



**“Simulación de sistemas metálicos amorfos y
porosos de elementos nobles”**

T E S I S

QUE PARA OBTENER EL GRADO DE
DOCTOR EN CIENCIA E INGENIERÍA DE
MATERIALES

P R E S E N T A

M. en C. César Ulises Santiago Cortés

Tutor:

Dr. Ariel Alberto Valladares Clemente

México, D. F.

2011



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Cuando agradeces a una persona lo que haces es crear una pequeña grieta en tu presa de agradecimientos. . . y que ahora se rompe; y empieza por aquí, salpicando a unos cuantos; y sigue por allá, creando ríos; y más allá, cambiando el clima y modificando paisajes; y terminará Dios sabe dónde, quizá del otro lado del mundo, haciendo desaparecer poblaciones. Así que, comencemos la catástrofe y veamos qué sucede. En primer lugar, y porque todavía hay clases sociales, agradezco a mi director de tesis, Dr. Ariel Alberto Valladares; simplemente esto no estaría escrito sin él. Agradezco el apoyo económico a CONACyT, uno de los organismos más importantes para el desarrollo de México. Y también, agradezco el apoyo económico de la DGAPA-UNAM a través del proyecto IN112211-3. Agradezco a DGTIC-UNAM, antes DGSCA-UNAM, por permitir que esto:

```
1  #!/bin/bash
2  #kanbalam
3  IP=132.248.202.193
4  if [ $# == 0 ]; then
5      ssh $IP -l cusc
6  else
7      ssh $IP -l $1
8  fi
```

no todos los días, resulte en:

```
1  ssh: The host is unreachable.
```

o peor aún en:

```
1  Permission denied, please try again.
```

Agradezco a M.T. Vázquez y O. Jiménez por conseguir lo inconseguible; toda aquella información científica escondida debajo de alguna piedra que, hasta entonces, estaba perdida en el mundo. Agradezco al Dr. Gerardo Vazquez Fonseca por aceptar ser parte de mi comité tutorial, y al Dr. Alexander Valladares, que también aceptó, pero sobre todo porque de él nació la propuesta de mezclarnos con los fractales. Agradezco a la Dra. Rubiciela Vargas de la Universidad Autónoma Metropolitana campus Iztapalapa, UAM-I, por su orientación positiva en todo lo negativo de Fukui. Y por supuesto, agradezco a la Universidad Nacional Autónoma de México, UNAM; que ha marcado un hito en mi vida, y que ha traído a mi existencia personas increíbles (a las que por cierto, también agradezco).

. . .y el resto de lo queda en la presa, que todavía es bastante, prefiero que fluya de mi boca, aunque me lleve toda la vida.

Lista de publicaciones

C. Romero, J.C. Noyola, U. Santiago, R.M. Valladares, A. Valladares, A.A. Valladares, A New Approach to the Computer Modeling of Amorphous Nanoporous Structures of Semiconducting and Metallic Material: A Review, *Materials*, 3 (2010) 467-502.

C.U. Santiago-Cortés, L.M. Mejía-Mendoza, R.M. Valladares, A. Valladares, Ariel A. Valladares, Computational alternatives to generate amorphous nanoporous structures using ab initio molecular dynamics, *J. Non-Cryst. Solids* (2011), doi:10.1016/j.jnoncrysol.2011.10.021

Contenido

Lista de Abreviaturas	11
Introducción	13
1 Antecedentes	17
1.1 Nanometales	19
2 Fundamento teórico	27
2.1 Dinámica molecular	27
2.1.1 Tamaño de paso	29
2.2 Teoría de funcionales de la densidad	30
2.3 Funciones de Fukui	34
2.4 Función de distribución de pares	36
2.5 Número de coordinación mediante la RDF	39
3 Desarrollo	41
3.1 Expansión de red	43
3.2 Desaleación	47
3.3 Desaleación progresiva	49
3.4 Generación de superceldas	52
3.5 Distribución de velocidades	55
3.6 Elementos simulados	58
3.7 Características de la simulación	60
4 Resultados	65
4.1 Funciones de distribución de pares para los sistemas estudiados	65
4.1.1 Ausencia de un mínimo cero en metales	71
4.1.2 Bimodalidad en el pico de segundos vecinos	73
4.1.3 Caída de la PDF debida a la porosidad	75
4.2 Ángulos	78
4.3 Topología	84
4.3.1 Estructuras desaleadas: Cu	85
4.3.2 Estructuras desaleadas: Ag	90
4.3.3 Estructuras desaleadas: Au	95
4.3.4 Estructuras expandidas: Cu, Ag, Au	100
4.4 Densidad de estados electrónicos	114
4.5 Densidad de estados vibracionales	119
4.6 Funciones de Fukui	121
4.7 Fractales	125

5	Conclusiones	131
	Referencias	133
A	Apéndices	139
A.1	Ejemplo de un archivo <code>.input</code> con parámetros para la dinámica molecular	139
A.1.1	Descripción de los parámetros utilizados por <code>dmol3</code> .	139
A.1.2	Archivo <code>.input</code> .	140
A.2	Ejemplo de un archivo <code>.input</code> con parámetros para la optimización	142
A.3	Número de átomos en una celda desaleada	142
A.4	Programa <code>celdas_asimetricas</code>	144
A.5	Número de átomos en una celda asimétrica desaleada	145
A.6	Programa <code>aleaciones</code>	146
A.7	Programa <code>dvmd</code>	155
A.8	Programa <code>pdf</code>	162
A.9	Programa <code>angulos</code>	168
A.10	Programa <code>msd</code>	177
A.11	Fundamento teórico del desplazamiento medio cuadrático	185
A.12	Validez del desplazamiento medio cuadrático para simulaciones <i>ab initio</i>	190
A.13	Programa <code>gausseame</code>	193
A.14	Programa <code>vibraciones</code>	197

Lista de Figuras

1.1	Estructuras nanoporosas...	23
1.2	Micrografías de Au nanoporoso...	25
2.1	Esquema de la función de distribución de pares para un material monoatómico. ...	36
2.2	Condiciones periódicas para una supercelda. ...	38
2.3	Alternativa para calcular la PDF. ...	38
2.4	Diagramas esquemáticos de RDF para evaluar el número de coordinación. ...	39
3.1	Forma esquemática de la técnica de expansión de red...	44
3.2	Forma esquemática de la técnica de expansión de red...	44
3.3	Forma periódica esquemática de la técnica de expansión de red...	46
3.4	Forma esquemática de la técnica de desaleación.	47
3.5	Forma esquemática de la técnica de desaleación asimétrica.	48
3.6	Forma esquemática de la técnica de desaleación progresiva.	50
3.7	Desaleación progresiva...	51
3.8	Supercelda de la aleación sustitucional Au-Ag al 50%...	53
3.9	Supercelda de Au desaleada al 70%...	54
3.10	Esquema de la obtención de sistemas desaleados. ...	55
3.11	PDF para Ag y Au desaleados ...	56
3.12	Cualquier punto, elegido de forma aleatoria...	57
3.13	Forma esquematizada del método aceptación-rechazo...	58
4.1	Función de distribución de pares, PDF, para el sistema Cu...	66
4.2	Función de distribución de pares, PDF, para el sistema Ag...	67
4.3	Función de distribución de pares, PDF, para el sistema Au...	68
4.4	Distancias de transición de primeros a segundos vecinos...	72
4.5	PDF esquemática...	73
4.6	Esquema que muestra la posible distribución geométrica correspondiente a la bimodalidad...	74
4.7	Capa de coordinación icosaedrales...	75
4.8	PDF de superceldas de Cu de arista de 23.5973 Å ...	77
4.9	Curvas PDF para los sistemas Cu ₄₀ y Ag ₄₀ a 300 K. (a) Comparación entre dCu ₄₀ y eCu ₄₀ . (b) Comparación entre dAg ₄₀ y eAg ₄₀ . En los detalles de cada figura aparece una línea gruesa continua que sirve como guía y es obtenida eliminando 95% de las frecuencias más altas en un suavizado de Fourier.	79
4.10	Comparación de los valores de ángulos planos para el sistema Cu...	81
4.11	Comparación de los valores de ángulos planos para el sistema Ag...	82
4.12	Comparación de los valores de ángulos planos para el sistema Au...	83
4.13	Estructura topológica del sistema dCu ₁₀ , número de átomos 259...	85
4.14	Estructura topológica del sistema dCu ₂₀ , número de átomos 274...	85

4.15	Estructura topológica del sistema dCu30, número de átomos 259...	86
4.16	Estructura topológica del sistema dCu40, número de átomos 240...	86
4.17	Estructura topológica del sistema dCu50, número de átomos 250...	87
4.18	Estructura topológica del sistema dCu60, número de átomos 240...	87
4.19	Estructura topológica del sistema dCu70, número de átomos 269...	88
4.20	Estructura topológica del sistema dCu80, número de átomos 256...	88
4.21	Estructura topológica del sistema dCu90, número de átomos 230...	89
4.22	Estructura topológica del sistema dAg10, número de átomos 259...	90
4.23	Estructura topológica del sistema dAg20, número de átomos 274...	90
4.24	Estructura topológica del sistema dAg30, número de átomos 259...	91
4.25	Estructura topológica del sistema dAg40, número de átomos 240...	91
4.26	Estructura topológica del sistema dAg50, número de átomos 250...	92
4.27	Estructura topológica del sistema dAg60, número de átomos 240...	92
4.28	Estructura topológica del sistema dAg70, número de átomos 269...	93
4.29	Estructura topológica del sistema dAg80, número de átomos 256...	93
4.30	Estructura topológica del sistema dAg90, número de átomos 230...	94
4.31	Estructura topológica del sistema dAu10, número de átomos 259...	95
4.32	Estructura topológica del sistema dAu20, número de átomos 274...	95
4.33	Estructura topológica del sistema dAu30, número de átomos 259...	96
4.34	Estructura topológica del sistema dAu40, número de átomos 240...	96
4.35	Estructura topológica del sistema dAu50, número de átomos 250...	97
4.36	Estructura topológica del sistema dAu60, número de átomos 240...	97
4.37	Estructura topológica del sistema dAu70, número de átomos 269...	98
4.38	Estructura topológica del sistema dAu80, número de átomos 256...	98
4.39	Estructura topológica del sistema dAu90, número de átomos 230...	99
4.40	Estructura topológica del sistema eCu10, número de átomos 256...	101
4.41	Estructura topológica del sistema eCu20, número de átomos 256...	101
4.42	Estructura topológica del sistema eCu30, número de átomos 256...	102
4.43	Estructura topológica del sistema eCu40, número de átomos 256...	102
4.44	Estructura topológica del sistema eCu50, número de átomos 256...	103
4.45	Estructura topológica del sistema eCu60, número de átomos 256...	103
4.46	Estructura topológica del sistema eCu70, número de átomos 256...	104
4.47	Estructura topológica del sistema eCu80, número de átomos 256...	104
4.48	Estructura topológica del sistema eCu90, número de átomos 256...	105
4.49	Estructura topológica del sistema eAg10, número de átomos 256...	105
4.50	Estructura topológica del sistema eAg20, número de átomos 256...	106
4.51	Estructura topológica del sistema eAg30, número de átomos 256...	106
4.52	Estructura topológica del sistema eAg40, número de átomos 256...	107
4.53	Estructura topológica del sistema eAg50, número de átomos 256...	107
4.54	Estructura topológica del sistema eAg60, número de átomos 256...	108
4.55	Estructura topológica del sistema eAg70, número de átomos 256...	108
4.56	Estructura topológica del sistema eAg80, número de átomos 256...	109

4.57	Estructura topológica del sistema eAg90, número de átomos 256...	109
4.58	Estructura topológica del sistema eAu20, número de átomos 256...	110
4.59	Estructura topológica del sistema eAu30, número de átomos 256...	110
4.60	Estructura topológica del sistema eAu40, número de átomos 256...	111
4.61	Estructura topológica del sistema eAu50, número de átomos 256...	111
4.62	Estructura topológica del sistema eAu60, número de átomos 256...	112
4.63	Estructura topológica del sistema eAu70, número de átomos 256...	112
4.64	Estructura topológica del sistema eAu80, número de átomos 256...	113
4.65	Estructura topológica del sistema eAu90, número de átomos 256...	113
4.66	Forma esquemática de la densidad de estados...	114
4.67	Densidad de estados electrónicos para el sistema Cu...	116
4.68	Densidad de estados electrónicos para el sistema Ag...	117
4.69	Densidad de estados electrónicos para el sistema Au...	118
4.70	Densidad de estados vibracionales para el sistema Cu...	119
4.71	Densidad de estados vibracionales para el sistema Ag...	120
4.72	Densidad de estados vibracionales para el sistema Au...	120
4.73	Plano xz del sistema eAg30...	122
4.74	Cortes transversales a $f^0(\mathbf{r})$ en los planos xz y yz del sistema eAg30...	123
4.75	Cortes paralelos al plano yz de la FF que atraviesan el centro...	125
4.76	Cálculo de la dimensión fractal para las estructuras...	127
4.77	Estructura topológica del sistema pC, después de la dinámica molecular partiendo de una estructura expandida de diamante...	130
4.78	Estructura topológica del sistema pC, después de la dinámica molecular partiendo de una estructura expandida de grafito...	130
A.1	Representación esquemática del desplazamiento en las posiciones atómicas...	187
A.2	Intervalos de tiempo posibles en un sistema de 12 pasos de simulación.	188
A.3	Intervalos de tiempo posibles en un sistema de 12 pasos de simulación.	188
A.4	MSD para Ge con time-step de 13 fs y 515 intervalos de tiempo.	190
A.5	MSD para Ge con <i>time step</i> de 13 fs y 515 intervalos de tiempo...	191
A.6	Cuatro curvas MSD para Si...	191
A.7	Cuatro curvas MSD para Si...	192
A.8	(a) Cuatro curvas MSD para Si...	192

Lista de Abreviaturas

BCC – Cúbica centrada en el cuerpo

cFF – Función de Fukui condensada

CS – Cúbica simple

DEE – Densidad de estados electrónicos

DEV – Densidad de estados vibracionales

DFT – Teoría de funcionales de la densidad

DM – Dinámica molecular

DVMB – Distribución de velocidades de Maxwell-Boltzmann

ER – Expansión de red

FCC – Cúbica centrada en las caras

FF – Función de Fukui

HCP – Hexagonal compacta

KS – Kohn y Sham

LDA – Aproximación de la densidad local

MC – Monte Carlo

PDF – Función de distribución de pares

RDF – Función de distribución radial

SC – Supercelda

SMNP – Sistemas metálicos nanoporosos

Introducción

Simular significa representar o hacer parecer algo que no es o no sucede en realidad.¹ Y efectivamente, en las simulaciones computacionales se busca, mediante algoritmos, representar y reproducir un fenómeno real que no ha sucedido en realidad. Computacionalmente se pueden simular desde cosas tan sencillas como el resultado de lanzar una moneda hasta fenómenos tan complejos como la interacción de muchos cuerpos o sistemas caóticos como el clima; se pueden simular comportamientos sociales o personales, tendencias en el mercado bursátil, incluso se habla de simular la realidad misma, en la actualidad no creemos que esto sea posible pero muchos videojuegos muestran los avances en este sentido. No hay límites, el único límite es la imaginación... y por supuesto, el algoritmo (modelo matemático) y los recursos computacionales. Pero dejando a un lado los recursos computacionales, la semejanza de lo simulado con la realidad es función exclusiva del modelo matemático. Entre mayor sea la semejanza de lo simulado con la realidad y con las implicaciones de ésta, mejor será el modelo empleado. Pero de nada serviría crear una *imagen* fiel de la realidad, sino es posible manipularla y hacer predicciones con ella; después que un modelo ha sido probado y ha demostrado su capacidad para generar una simulación congruente con la realidad (y sus implicaciones), es entonces cuando se utiliza para esto. De hecho, éste es el objetivo principal de toda simulación, la predicción y manipulación de esta imagen de la realidad.

En la simulación de sistemas físicos, la manipulación puede utilizarse para entender la propia realidad, esto es, la simulación es una herramienta que permite estudiar y comprender la naturaleza a un nivel de interacción tal, imposible de conseguir de otra forma, al llevar al extremo los parámetros experimentales; como la temperatura o la presión.

Hay muchas formas de abordar una simulación, por ejemplo, podría pensarse en aquellas simulaciones que sólo intentan reproducir una parte básica y aislada de la realidad como el resultado de la tirada de un par de dados, o aquellas en que, basadas en leyes físicas, intentan reproducir la tirada de los dados en sí misma, o yendo más allá, simulaciones que traten de reproducir el comportamiento de los átomos que conforman los dados. Ahora bien, podría abordarse una simulación desde un punto determinístico en el que existe cierto

¹ Fuente: L.F. Lara, Diccionario del español usual en México, El Colegio de México, México, 1996.

conjunto de ecuaciones que describen el proceso que se quiere simular; o bien, desde el punto de vista estocástico en el se introduce una variable aleatoria y se establecen las tendencias (ecuaciones) en su comportamiento. Y así podrían abordarse de otras tantas maneras, como aquellas simulaciones que se basan en datos experimentales y las que se basan en una teoría pura, y de aquí en teorías clásicas o cuánticas; todo depende de lo que se quiera simular.

Aquellas simulaciones conocidas como *ab initio* o cuánticas *ab initio* son un ejemplo de simulaciones de *teoría pura*, que no utilizan datos experimentales, que se basan exclusivamente en la teoría cuántica para simular el comportamiento de un sistema a nivel atómico. Los algoritmos creados con las ecuaciones de la teoría cuántica suelen ser muy demandantes computacionalmente, y esta demanda aumenta, casi siempre, cúbicamente con el número de átomos simulados. Por esta razón, en las técnicas *ab initio*, se simulan normalmente menos de 500 átomos.

El desarrollo tecnológico orientado a la manipulación y creación de sistemas nanométricos, de tanto interés económico y académico, se vale de las simulaciones *ab initio* para desarrollar, estudiar y comprender estructuras y fenómenos que se manifiestan a esa escala. Así es como en este trabajo nos valemos también de la simulación *ab initio* para estudiar cierto tipo de estructuras denominadas de forma genérica: sistemas metálicos nanoporosos, SMNP. Los SMNP son estructuras nanométricas continuas que tienen extensa aplicación efectiva y *potencial* en diferentes áreas tecnológicas lo que vuelve fundamental estudiar sus propiedades y características de la forma más extensamente posible.

Aunque ya antes se habían simulado estas estructuras con métodos no *ab initio*, sobre todo para estudiar su mecanismo de formación estructural, no hay reportado un estudio *ab initio* como el que realizamos en este trabajo. Aquí, no estudiamos el mecanismo de formación, más bien, analizamos el tipo de estructura formada mediante dos métodos computacionales de generación de superceldas; no encontramos una diferencia significativa entre ellos al analizar la estructura topológica de los sistemas creados. Los SMNP que simulamos están compuestos ya de Cu, ya de Ag o ya de Au, tres elementos representativos de los SMNP. Analizamos, además, la reactividad de los sistemas creados, utilizando la teoría de funciones de Fukui; lo que nos ha llevado a obtener resultados controversiales. Así mismo, presentamos resultados de los cálculos de la estructura electrónica y un estudio vibracional, y otro de ángulos planos. Empleamos el concepto de dimensión fractal para

caracterizar estas estructuras. Finalmente, presentamos el código fuente de los principales programas de cómputo que hemos desarrollado para este trabajo.

Los resultados presentados sólo son predictivos, no por ser producto de una simulación, sino porque no hemos encontrado resultados experimentales que describan sistemas semejantes a los que aquí simulamos. Los resultados en este trabajo son un buen ejemplo de la predicción y manipulación de la imagen de la realidad que sólo es posible con la simulación. Esperamos con esto, contribuir al campo de conocimiento de estos materiales y de los materiales nanoporosos en general.

Finalmente, en los apéndices, al final de este trabajo, presentamos el código fuente de algunos programas de cómputo que hemos desarrollado y que nos han servido de herramienta durante el desarrollo de este trabajo. Ninguno de estos programas está limitado a ser utilizado en los sistemas metálicos nanoporosos, pues su función es tan general como el análisis de ángulos, cálculo de PDF y RDF, análisis vibracional, etcétera.

1 Antecedentes

Nanomateriales, nanopartículas, nanocapas, nanoestructuras... todos ellos son términos que están presentes en lo que algunos investigadores han denominado una nueva era *nano* donde el desarrollo tecnológico está basado en la manipulación de la materia a escala nanoscópica, manipulación que tiene por objetivo práctico producir eficientemente energía, químicos y materiales.^[1] La ciencia de las nanoestructuras o nanociencia es en realidad una ciencia interdisciplinaria, pero el denominador común es su objeto de estudio: estructuras en las que, *al menos una* de sus dimensiones es típicamente menor a 100 nm.^[2, 3]

Es decir, la nanociencia pretende manipular la materia a escala atómica y molecular. En este sentido, la nanociencia no tiene límites, potencialmente toda la materia, es decir, todo aquello constituido por átomos puede ser manipulado: metales, no metales, semi-metales; cualquier elemento de la tabla periódica; cualquier compuesto, cualquier solución, cualquier estructura. En la actualidad, y a pesar de la gran actividad en el campo de la nanociencia, quizá no se estudie toda la materia existente, pero es tanta la cantidad y diversidad de materia que se estudia que se ha intentado dar algún tipo de orden. Una de las clasificaciones más reconocida es la que dio la Agencia de Protección Ambiental de los Estados Unidos de Norteamérica.^[4] Según esta agencia, todos los nanomateriales se encuentran dentro de alguno de los siguientes grupos:

- Materiales basados en carbono: compuestos principalmente de carbono, comúnmente se habla de esferas huecas, elipsoides y tubos. *Fulereno* es el nombre que reciben los nanomateriales a base de carbón de forma esférica o elipsoidal, y *nanotubo* los de forma cilíndrica. Tanto los fulerenos como los nanotubos tienen muchas aplicaciones potenciales entre las cuales se incluye la mejora de capas y recubrimientos materiales resistentes y ligeros, y aplicaciones en electrónica.
- Materiales dendriméricos: son polímeros de tamaño nanométrico construidos a partir de unidades ramificadas. La superficie de los dendrímeros tiene numerosas cadenas terminales que pueden ser adaptadas para realizar funciones químicas específicas. Además, debido a que los dendrímeros tridimensionales contienen cavidades interiores en el que otras moléculas podrían ser incluidas, pueden ser útiles para la administración de fármacos.

- Materiales basados en metal: por ejemplo, puntos cuánticos,² oro, plata, cobre, óxidos metálicos.
- Materiales compuestos: Combinan nanopartículas de diferentes materiales y tamaños, que a su vez pueden ser colocados en otros materiales en bulto. Las nanopartículas, como las arcillas nanométricas, ya se añaden a los productos que van desde piezas de automóviles a materiales de embalaje.

Pero ¿qué es lo que hace tan diferente a la materia cuando se manipula a escala nanométrica? Esta pregunta quizá no sea ni corta ni fácil de responder. Se sabe que cuando se manipula la materia a esta escala, se manifiestan variaciones en las propiedades eléctricas, ópticas, magnéticas y químicas.^[5] Y pueden darse dos argumentos para que efectivamente esto sea así; el primero es: en los materiales nanométricos la relación área superficial–volumen es muy grande comparada con la relación correspondiente en los materiales en bulto, es decir, existe una gran cantidad de enlaces sueltos, o bien, átomos subcoordinados, por unidad de volumen; y el segundo, debido a las dimensiones y topología de estos materiales, es apreciable el efecto del cambio en confinamiento de los potenciales, es decir, los límites entre los cuales se encuentra confinado un potencial es de orden nanométrico. Por ejemplo, en el caso de nanopartículas metálicas cuyo tamaño se aproxima o es menor al recorrido libre medio de los electrones (para metales a temperatura ambiente es del orden de 10^{-5} cm o menor^[6]) la dispersión en la superficie se manifiesta. Cuando el tamaño de las partículas metálicas se reduce, el principio de incertidumbre implica que la energía cinética intrínseca sea:

$$E_k = \left(\frac{1}{2m_e} + \frac{1}{2m_h} \right) p^2 \approx \frac{\hbar^2}{m^*r^2} \quad (1.1)$$

donde r es el diámetro de la partícula m_e y m_h son las masas de los electrones y de los huecos. Un exceso de electrones induce a cargas imagen en la superficie de la partícula que generan un potencial coulombico atractivo que es proporcional a $e^2/\epsilon r$. Y El radio de Bohr para nanopartículas

² Un punto cuántico es un cristal semiconductor, de empaquetamiento compacto, compuesto de cientos o miles de átomos, y cuyo tamaño es del orden de hasta cientos de nanómetros.^[4] Y en general si una de las longitudes de la nanoestructura tridimensional es nanodimensional, la estructura recibe el nombre de *pared cuántica*; si dos lados son de longitud nanométrica, la estructura se le llama *alambre cuántico*. Así, en un *punto cuántico* las tres dimensiones de la estructura son de orden nanométrico.^[7]

$$r_B = \frac{\hbar^2 \epsilon}{m^* e^2} \quad (1.2)$$

puede definirse como aquel para el cual la energía coulombica es igual a la energía cinética. El efecto del tamaño cuántico predomina sobre el efecto de tamaño clásico cuando la partícula es más pequeña que el radio de Bohr.^[8]

Las aplicaciones de los nanomateriales son muchas,^[7] por mencionar algunas están: varios tipos de sensores usados en la industria, por particulares, en la defensa, y en aplicaciones biomédicas. Por ejemplo, sensores de presión piezoresistivos, sensores de presión absoluta para motores y presión arterial, cabezales de impresión de tinta, biosensores, sensores de gas. Otras aplicaciones incluyen, detectores de infrarrojo, aplicaciones en la ciencia espacial, aplicaciones en misiles, picosatélites,³ celdas de combustible, productos hidráulicos y neumáticos, sistemas de almacenamiento magnético, etcétera. Una de las áreas de producción de la empresa alemana de nanotecnología, PlasmaChem,^[9] es la fabricación de cúmulos metálicos de tamaño entre 10 y 300 nm, según esta empresa, este tipo de materiales se aplican en la metalurgia de polvos, nanolíquidos magnéticos, juntas de soldadura, pegamentos reforzados y materiales compósitos, componentes primarios para intercambiadores de calor porosos, aditivos para aceite de motor y nanofiltros reutilizables de metal. Una aplicación potencial en nanoelectrónica es usar moléculas individuales o nanotubos para almacenar bits de información. Esta lista de aplicaciones no es para nada exhaustiva, pero da una idea del campo donde la nanociencia y nanotecnología es y puede ser aplicada.

1.1 NANOMETALES

Pretender estudiar todos los materiales es algo imposible, aunque es cierto que la nanociencia es multidisciplinaria, los grupos de trabajo, normalmente, se especializan en algún tipo de material o bien en diferentes materiales con ciertas características en común. En este trabajo se tratará principalmente con nanomateriales metálicos por lo que en los siguientes párrafos se realizará una breve descripción de ellos.

³ Satélites cuyo masa está entre 0.1 kg y 1kg.

Existen varios métodos de fabricación de nanometales, dependiendo del producto final deseado, que podrían clasificarse en varios grupos:^[10]

1. Físico-mecánicos y fase-vapor

- Molienda mecánica y mecanoquímica (Mechanical and mechanochemical milling), utilizando el molino de bolas, por ejemplo. Es un tipo de técnica que involucra la erosión controlada⁴ de precursores (partículas o polvo metálico) para producir estructuras finas. Si el polvo es de un sólo elemento se llama «molienda mecánica», pero si el polvo es de dos o más elementos se le llama «aleado mecánico». La distribución del tamaño de las partículas obtenidas por este medio, suele ser bastante ancha comparada con la distribución producida por métodos químicos.
- Técnicas de electrohilado (Electrospinning). Está basada en la aplicación de altos voltajes para la preparación de nanofibras y otras nanoestructuras elongadas. El proceso consiste en aplicar altos voltajes a una solución polimérica que es expulsada de un capilar. Esto resulta en fibras que son colectadas en el contraelectrodo en forma de paquetes alineados. Es posible adicionar nanopartículas metálicas a la solución polimérica para obtener nanofibras compósitas metal-polímero.
- Pirólisis de rocío (Spray pyrolysis).^[11] Se utiliza para producir partículas de tamaño micro y nanométrico. Una solución acuosa, del precursor correcto, es atomizada sobre una superficie caliente donde los constituyentes *pueden* reaccionar para formar un compuesto químico. O bien las gotas, mediante un gas, son llevadas dentro de un horno tubular y el precursor es convertido en nanopartículas esféricas.
- Descarga de arco (Arc-discharge). Se aplica un arco eléctrico dentro de un gas inerte o reactivo, o bien, dentro de un líquido, generando un plasma que da lugar a la formación de átomos metálicos y su condensación en partículas de diferentes estructuras y composición.
- Condensación de vapor metálico (Metal vapor condensation). En esta técnica se evapora un metal para después condensarlo bajo condiciones inertes, dando lugar a partículas metálicas.

⁴ También conocida como «desgaste mecánico».

- Condensación de vapor metal-orgánico (Metal-Organic chemical vapor condensation). Se emplean especies metalorgánicas como precursores para la preparación de nanopartículas metálicas. El método se basa en la evaporación y descomposición del precursor y su posterior condensación.

2. Químico

- Ruta química sin surfactantes.⁵ Consiste simplemente en la reducción directa de un precursor en la solución apropiada.
- Ruta química con surfactantes. Éste es todo un conjunto de técnicas que consisten en la reducción de sales metálicas en presencia un ambiente micelar (surfactantes) estabilizante. Los diferentes tipos de esta técnica se distinguen por el tipo de surfactante.
- Ruta química basada en agentes limitantes poliméricos y dendriméricos (Wet-chemical routes based on polymer and dendrimer capping agents). De alguna forma esta técnica es similar a la anterior, en que las partículas metálicas, reducidas desde un apropiado precursor, son encapsuladas, esta vez, con cadenas poliméricas.
- Rutas en fluidos calentados y comprimidos. Se trata de reacciones heterogéneas que ocurren en presencia de solventes acuosos o mineralizadores bajo altas presiones y altas temperaturas, condiciones para disolver y recristalizar materiales que son relativamente insolubles bajo condiciones ordinarias.

3. Electroquímico. Los procesos electroquímicos, a pesar de no ser usados extensamente para producir nanopartículas metálicas,⁶ generan partículas de alta pureza y permiten un control estricto de la morfología de las partículas variando la densidad de corriente o el potencial.

⁵ La palabra surfactante resulta de la unión de tres palabras inglesas *surface-active-agent*, es decir un agente que actúa en la superficie. Así, los surfactantes son compuesto que modifican la relación entre dos superficies, o bien, interfases. Los surfactantes tienen la propiedad de absorberse en la superficies o interfases y alterar la energía superficial o interfacial. Existen tres tipos principales de surfactantes: detergentes, agentes humectantes y emulsificantes.^[12]

⁶ Aunque sí para generar otro tipo de estructuras como los sistemas metálicos nanoporosos que se revisarán más adelante.

- Electrólisis con ánodo de sacrificio en presencia de surfactantes. En esta técnica se reduce un metal en presencia de un estabilizador (surfactante) para evitar la adhesión de los iones en el electrodo. Con el surfactante pueden crearse partículas completas surfactadas.
- Electro erosión (Electrochemical milling). Esta técnica consiste en erosionar, de forma muy localizada, una superficie por vía electroquímica. Las partículas «arrancadas» mediante esta erosión pueden ser de tamaño nanométrico, aunque la superficie erosionada también queda horadada nanométricamente.

Las técnicas descritas anteriormente están enfocadas, principalmente a la generación de nanopartículas y no de estructuras «continuas» nanoestructuradas. Para este último tipo de estructuras existen otro tipo de técnicas como el recocido de listones amorfos,^[13] la deformación plástica severa^[14, 15] (por ejemplo, torsión bajo alta presión y extrusión cíclica). Al principio, lo que se buscaba en este tipo de técnicas era modificar la microestructura de tal forma que se obtuviera un refinamiento de grano, lo que implicaba un aumento en la resistencia pero una disminución en la ductilidad, lo cual no siempre era deseable, sobre todo en aplicaciones estructurales. El punto es, estas técnicas han llegado a un refinamiento tal que pueden conseguirse tamaños de grano ultrafinos (de orden nanométrico), lo que provoca alta resistencia pero también alta ductilidad.^[14]

Un tipo más de materiales «continuos» nanoestructurados son los llamados sistemas metálicos nanoporosos.⁷ A grandes rasgos estos materiales son materiales metálicos que poseen poros de tamaño nanométrico en su interior. Los sistemas metálicos nanoporosos tienen una aplicación efectiva y potencial en diferentes áreas tecnológicas. En las aplicaciones se incluyen la catálisis para las celdas de combustible, actuadores electroquímicos⁸ — electrochemical actuators^[18]—, sensores,^[19] control de microfluidos,^[19, 20] biosensores,^[21] intercambiadores de calor,^[17] y almacenamiento de hidrógeno.^[22] Pero más allá de su aplicación directa, los metales nanoporosos poseen propiedades interesantes por sí mismos como la dependencia de la resistencia mecánica con el tamaño de *ligamento*^[23] (ligament⁹),

⁷ De acuerdo a las recomendaciones de la IUPAC los poros se clasifican según su tamaño como: microporos (<2 nm), mesoporos (2–50 nm) y macroporos (>50 nm), aunque generalmente se emplea el término *nanoporo* y *nanoespacio* para describir poros de escala nanométrica.^[1, 16]

⁸ La *actuación electroquímica* es el resultado de variar la magnitud del esfuerzo superficial —surface stress— de los *ligamentos* (ver la siguiente nota) de un metal poroso mediante la aplicación de un voltaje electroquímico. El *esfuerzo superficial* es una cantidad termodinámica que cuantifica la energía necesaria para crear una nueva área superficial mediante la deformación de una superficie existente.^[17]

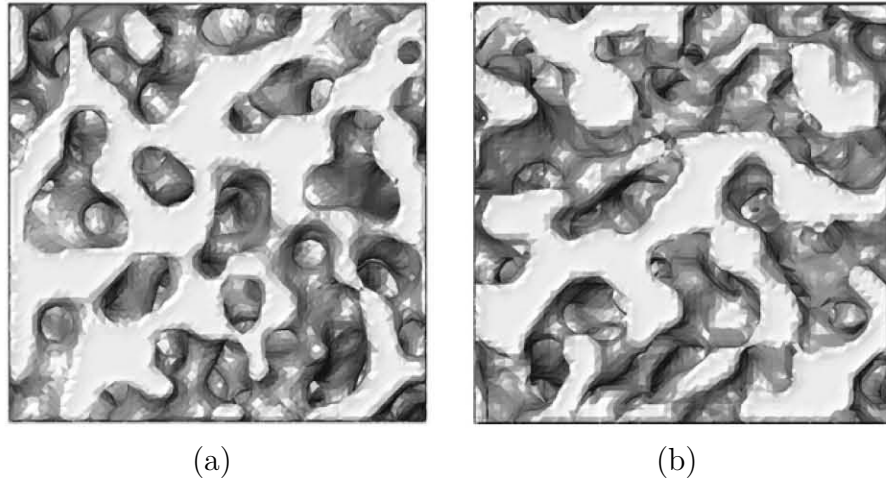


Figura 1.1 Estructuras nanoporosas —simuladas con (a) el modelo de *phase-field* (b) análisis Berk— de Au, las regiones claras son los ligamentos y las oscuras los poros. *D. A. Crowson et al., Scripta Materiala., 56, (2007) 919–922.*

propiedades electrocatalíticas, piezoelectricidad,^[24, 25, 26] terapia termal, electroquimiluminiscencia^[25] y la transición dúctil–frágil al cambiar la escala microestructural,^[27] entre otras.

Todo esto es paradójico si se considera que históricamente la porosidad en metales y aleaciones (sistemas metálicos), a cualquier escala de magnitud, se ha considerado casi siempre indeseable, pues se le asocia con la corrosión —asistida por esfuerzos, por picadura, desaleación, etcétera.^[28, 29, 30, 31] Pero la paradoja sigue, en la actualidad no sólo ya no es indeseable, bajo ciertos contextos, sino que existen métodos^[19, 22, 25, 26] para producir estructuras metálicas porosas con un control relativamente estricto en el tamaño^[24] y, en ocasiones, en la forma de la porosidad. Los métodos de preparación para los *materiales* nanoporosos se dividen en dos grupos: procesos de acumulación [de partículas](build-up) y lixiviación (leaching).^[16] Dentro de los procesos de lixiviación se encuentra la electro–disolución (disolución selectiva [electroquímica]) que en aleaciones es realmente una desaleación. Pero la mayoría de las publicaciones hechas recientemente acerca de la fabricación de nanoporos metálicos dirigen su atención a la lixiviación y más específicamente a la desaleación.^[26, 32] La aglomeración de partículas integra algunos de

⁹ Si con un plano cortásemos una estructura porosa, todo aquello que se intersecta con el plano son los ligamentos. En la **figura 1.1** se muestran dos ejemplos. En este contexto también empleamos el término *backbone*.

los métodos descritos arriba.^[33] Por su parte, la desaleación consiste en sumergir una aleación (generalmente una aleación binaria y de una sola fase) en algún tipo de solución química,^[34] el elemento menos noble será disuelto mientras que el más noble quedará formando una especie de esqueleto que no corresponde exactamente a la estructura ni a la posición de los átomos más nobles antes de que la disolución suceda;^[17] aunque, se ha probado que la estructura «predominantemente»¹⁰ se preserva.^[35] Justamente esta discrepancia entre las posiciones y estructura del elemento más noble antes y después de la desaleación es lo que ha llevado a Erlebacher,^[29] y a otros, a estudiar el mecanismo de formación; pues consideran que los aspectos morfológicos de las estructuras formadas mediante la desaleación y su subyacente mecanismo no es claro.^[17, 29, 36]

Entre los años 2001 y 2004 Erlebacher *et al.*^[29, 37, 38] proponen «[...] un modelo [de medio] continuo que es consistente con los experimentos y simulaciones teóricas de la disolución de aleaciones, y demuestra que la nanoporosidad en metales es debida a un proceso de formación de un patrón dinámico intrínseco».^[29] El patrón intrínseco al que alude Erlebacher es el característico de la descomposición espinodal.¹¹ En este mismo artículo Erlebacher aplica su modelo a la aleación Ag–Au mediante una simulación de Monte Carlo obteniendo la conclusiones mencionadas arriba. Aunque consistente, el modelo de Erlebacher no considera lo que le sucede en la microestructura durante la desaleación, es decir, considera la aleación como una red rígida donde no hay recristalización ni esfuerzos inducidos por la deformación.^[35, 36] Otra simulación numérica hecha sobre un sistema desaleado, empleando el método de Monte Carlo, fue en el 2004 y sólo se simula la superficie de una aleación durante la desaleación,^[39]

En el año 2007 Crowson *et al.* se plantean una pregunta: «si comenzamos con una aleación de Ag–Au contenida en un volumen L^3 , y entonces removemos la Ag de la aleación mediante una disolución selectiva, ¿qué gobierna la dimensión del volumen final con arista L^*

¹⁰ Específicamente Van Petegen *et al.*^[35] dice: [...] para esta aleación [Au–Ag] la preservación de la estructura granular original resulta en una microestructura no relajada con una deformación-rms (root mean square strain) de alrededor del 0.1% que es del mismo orden de magnitud observada en algunos materiales nanocristalinos con tamaño de grano equivalente.

¹¹ Una aleación de dos componentes, digamos A y B, que es homogénea a altas temperaturas, al enfriarse puede descomponerse de forma espontánea en una mezcla de dos fases heterogéneas. Esta mezcla es una estructura fina en la que alternan de forma más o menos regular las dos fases separadas. Este tipo de descomposición se caracteriza por no tener una barrera energética de nucleación y la interfase entre las regiones ricas en A y B es difusa más que abrupta.^[40] Una simulación de este tipo de descomposición puede verse en: <http://math.gmu.edu/~sander/movies/spinum.html>.

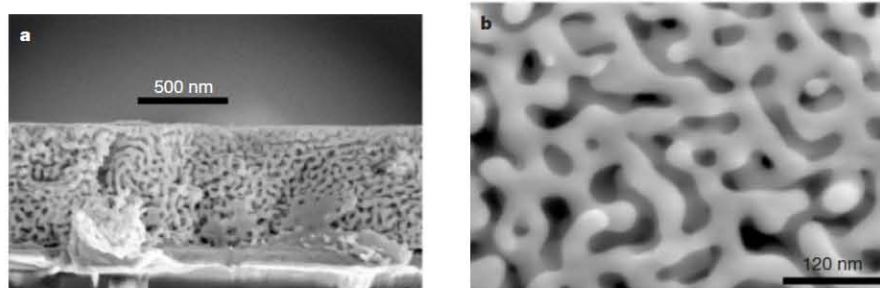


Figura 1.2 Micrografías de Au nanoporoso hechas por disolución selectiva a partir de la aleación Au₃₂Ag₆₈. La porosidad es abierta y el espaciado de los ligamentos es de 5 a 10 nm. *J. Erlebacher et al., Nature, 410, (2001) 450–453.*

de la estructura nanoporosa de Au resultante?»^[17] Intentan responder esta pregunta mediante simulaciones computacionales empleando potenciales interatómicos del tipo *átomo embebido* (Embedded atom method potentials). En este tipo de simulaciones se realizan dinámicas moleculares integrando las ecuaciones de Newton, obteniendo así las posiciones y velocidades de los átomos. Típicamente pueden simularse entre 5,000 y 100,000 átomos.^[41] Estos mismos autores en el año 2009 emplean esta técnica para determinar la estabilidad mecánica de los metales nanoporos con respecto a pequeños tamaños de ligamentos.^[18] Un año antes, en el 2008 se publicó un artículo donde se simula numéricamente la desaleación de una aleación Ag–Au empleando el método del elemento finito con la finalidad de describir la formación de la nanoporosidad.^[42] Y hasta aquí, no se han hecho más simulaciones, con carácter morfológico, en este campo de la desaleación, por ende, de las estructuras metálicas nanoporosas; hasta donde sabemos. Otro tipo de simulaciones que nos interesa estudiar son aquellas en las que se investiga la reactividad, estructura electrónica, propiedades vibracionales u ópticas, y que no han sido reportadas hasta la fecha. Por esto, en este trabajo simulamos de forma computacional y desde primeros principio, *ab initio*, la estabilidad, características y viabilidad de la generación de estructuras nanoporosas en los metales, para lo cual elegimos tres metales Au, Ag y Cu. Nos proponemos, además, determinar la estructura electrónica así como su reactividad empleando la teoría de funciones de Fukui.

2 Fundamento teórico

En esta sección se da un resumen de la teoría atrás de las simulaciones computacionales *ab initio* que se realizan en este trabajo.

2.1 DINÁMICA MOLECULAR

La dinámica molecular, DM es una simulación computacional en la que se describe la evolución temporal de un conjunto de partículas¹² mediante la integración de las ecuaciones que rigen su movimiento. En detalle, en una DM se calcula la fuerza resultante de la interacción de un conjunto de partículas, luego el sistema se deja evolucionar, esto es, integrando las ecuaciones de movimiento para determinar la posición, velocidad y aceleración en un intervalo de tiempo,¹³ Δt , al término del cual, el conjunto se «congela» y se determinan las fuerzas de interacción de las partículas, una vez hecho esto el conjunto se «descongela» dejando evolucionar nuevamente un tiempo Δt , al cabo del cual el sistema se congela y se calculan las fuerzas... Este ciclo se repite hasta que la suma de los intervalos de tiempo alcanza cierto límite. Las ecuaciones de movimiento normalmente son clásicas, pero la fuerza de interacción o potencial puede calcularse de forma cuántica o clásica; para el caso cuántico, que es el que nos interesa, debe resolverse la ecuación de Schrödinger o alguna otra equivalente como las ecuaciones de Kohn–Sham. Normalmente se considera el sistema como un sistema conservativo que cumple con las ecuaciones de movimiento de Newton, pero no siempre es así.

En la dinámica puede considerarse cualquier ensamble estadístico: microcanónico, canónico, gran canónico, isoentálpico-isobárico, etcétera. Cada paso de simulación representa una configuración particular con la que pueden realizarse mediciones de forma estadística como la energía potencial, la energía cinética, la temperatura, la presión, curvas calorimétricas, el desplazamiento medio cuadrático, etcétera. Pero ¿cómo es que se hace todo esto? La forma más simple de visualizar la DM es, definiendo las posiciones de conjunto de partículas dentro de una caja de dimensiones conocidas y asignándoles una velocidad. Las

¹² Átomos, moléculas o ambos.

¹³ Time step.

posiciones iniciales dentro de la caja pueden ser arbitrarias, producto de una simulación anterior, ordenadas según un arreglo cristalino, todo depende del tipo de sistema que se simule y el propósito con el que se haga. Y el conjunto de velocidades, de la misma forma, puede ser arbitrario, siguiendo un propósito específico,¹⁴ cero, o bien siguiendo algún tipo de distribución, por ejemplo, la distribución de Maxwell–Boltzmann. Justamente en la asignación de velocidades y posiciones es donde entra la aleatoriedad, lo que genera configuraciones arbitrarias y simula, en consecuencia, un proceso real.

Este es el paso inicial de la DM, lo que sigue es controlar la evolución del sistema, es decir, calcular la temperatura, presión, volumen pero sobre todo las fuerzas que actúan sobre cada partícula. La temperatura se controla mediante lo que se conoce como *reescalamiento de velocidades*, útil sobre todo cuando durante la DM se desea aumentar o disminuir la temperatura. Por ejemplo, en el algoritmo de Verlet^[43] se reemplaza el cálculo de la velocidad a una temperatura dada:

$$\mathbf{v}(t + \Delta t/2) = \mathbf{v}(t) + 1/2\mathbf{a}(t)\Delta t \quad (2.1)$$

por

$$\mathbf{v}(t + \Delta t/2) = \sqrt{\frac{T_o}{T(t)}}\mathbf{v}(t) + 1/2\mathbf{a}(t)\Delta t \quad (2.2)$$

donde T_o es la temperatura deseada y $T(t)$ es la temperatura instantánea. Justamente, éste es un caso donde la energía del sistema no se conserva. Las propiedades del sistema no deben calcularse cuando se hace una variación de este tipo pues se introduce un parámetro artificial que desestabiliza el sistema. Las fuerzas del sistema, como se dijo arriba, pueden calcularse de forma clásica o cuántica, en este trabajo lo que nos interesa es el cálculo cuántico en especial utilizando la teoría de funcionales de la densidad por lo que en la **sección 2.2** se presenta un pequeño resumen de esta teoría; pero antes, en la sección siguiente se discute brevemente la forma en la que puede seleccionarse el tamaño de paso empleado en una simulación.

¹⁴ Por ejemplo, asignar cierta velocidad que permita predecir el punto de choque de dos partículas.

2.1.1 Tamaño de paso

El intervalo de tiempo que se deja evolucionar al sistema, tamaño de paso o *time step*, Δt , debe elegirse lo suficientemente pequeño como para impedir que los átomos que participan la dinámica no se traslapen pero lo suficientemente grande para que exista una evolución del sistema y se maximice el tiempo de cómputo.

La elección del Δt , hasta donde sabemos, se basa en criterios de prueba y error. Sin embargo el programa FAST STRUCTURE^[44] reporta una sencilla fórmula que depende del número de masa atómica.

$$\Delta t \propto \sqrt{\frac{A}{5}} \quad (2.3)$$

donde A es el número de masa atómica¹⁵ del elemento más ligero que participa en la dinámica molecular. Tal y cómo está, esta ecuación da resultados adimensionales pero FAST STRUCTURE los considera como número de femto segundos. El grupo de trabajo utiliza, casi siempre, un valor de Δt igual a cuatro veces el definido por FAST STRUCTURE; de cualquier forma esto no asegura que no haya traslapes durante la dinámica; por supuesto, tampoco lo asegura un tamaño de paso menor. Sin embargo está fórmula da un punto de partida en la elección del Δt . Otra manera de elegir el Δt es hacer diferentes dinámicas variando únicamente el Δt y observando el comportamiento de la curva de energía del sistema. Una curva suave es generada por un Δt en el que no hubo traslapes ni explosiones, entonces un criterio de elección será elegir el Δt más grande para el cual la curva de energía se mantenga suave.

No existe una fórmula que permita calcular de forma eficiente el tamaño de paso. Pero de existir, ésta debería considerar algunos parámetros generales como la masa, la velocidad, la temperatura, las distancias interatómicas, y quizá hasta el tamaño de las partículas.

En definitiva, la elección de un Δt no puede ser arbitraria, un valor muy pequeño equivale a un cálculo ineficiente debido a que se realizan integraciones de las ecuaciones de movimiento innecesarias, pues el sistema evoluciona muy poco entre una integración y otra. Un valor demasiado grande, puede ocasionar el cálculo de fuerzas de repulsión enormes, que se traducen en la explosión del sistema, por el acercamiento entre los átomos.

¹⁵ $A = Z + N$; donde Z es el número de protones y N número de neutrones.

2.2 TEORÍA DE FUNCIONALES DE LA DENSIDAD

Una de las aproximaciones mecanocuánticas más importantes (y más populares) para calcular las propiedades moleculares o de sólidos es la teoría de funcionales de la densidad, DFT.¹⁶ «La DFT intenta calcular la energía electrónica,¹⁷ E_0 , y otras propiedades moleculares a partir de la densidad electrónica, ρ_0 .»^[45] En un artículo publicado en 1964, P. Hohenberg y W. W. Kohn,^[46] probaron que la función de onda, la energía en el estado base y todas las propiedades electrónicas, pueden determinarse a partir de una función de la posición: la densidad de la probabilidad electrónica del estado base, $\rho_0(x, y, z)$ o $\rho_0(\mathbf{r})$, es decir, la energía es una funcional de la densidad, $E_0(\rho_0)$. La primera suposición que hacen Hohenberg y Kohn para demostrar esto, es que los núcleos atómicos permanecen fijos,¹⁸ de aquí se define el llamado *potencial externo*, $v(\mathbf{r}_i)$, que es el debido a los núcleos y que actúa sobre el electrón i ,

$$v(\mathbf{r}_i) = - \sum_{\alpha} \frac{Z_{\alpha}}{r_{i\alpha}}, \quad (2.4)$$

ellos comprobaron que ρ_0 determina el potencial externo y, por supuesto, el número de electrones mediante:

$$\int \rho_0(\mathbf{r}) d\mathbf{r} = n. \quad (2.5)$$

Del hamiltoniano puramente electrónico,

$$\hat{H}_{\text{electrónico}} = -\frac{1}{2} \sum_i \nabla^2 - \sum_{\alpha} \sum_i \frac{Z_{\alpha}}{r_{i\alpha}} + \sum_j \sum_{i>j} \frac{1}{r_{ij}} \quad (2.6)$$

que contiene los términos de energía cinética electrónica, $\bar{T} = -1/2 \sum_i \nabla^2$, atracciones electrón-núcleo, $\bar{V}_{Ne} = -\sum_{\alpha} \sum_i Z_{\alpha}/r_{i\alpha}$, y repulsiones electrón-electrón, $\bar{V}_{ee} = \sum_j \sum_{i>j} 1/r_{ij}$; $E = \bar{T} + \bar{V}_{Ne} + \bar{V}_{ee}$ y debido a que cada uno de estos términos es una funcional de ρ_0 ,

$$E[\rho_0] = \bar{T}[\rho_0] + \bar{V}_{Ne}[\rho_0] + \bar{V}_{ee}[\rho_0]; \quad (2.7)$$

pero $\hat{V}_{Ne} = \sum v(\mathbf{r}_i)$, entonces puede demostrarse que

¹⁶ Density Functional Theory.

¹⁷ Tanto la energía como las propiedades moleculares y la densidad electrónica son del estado base, a menos que se indique lo contrario.

¹⁸ Aproximación de Born Oppenheimer.

$$\bar{V}_{Ne} = \langle \psi_0 | \sum v(\mathbf{r}_i) | \psi_0 \rangle = \int \rho_0(\mathbf{r})v(\mathbf{r})d\mathbf{r} \quad (2.8)$$

De aquí se conoce \bar{V}_{Ne} pero $\bar{T}[\rho_0]$ y $\bar{V}_{ee}[\rho_0]$ no, así

$$E[\rho_0] = \int \rho_0(\mathbf{r})v(\mathbf{r})d\mathbf{r} + \bar{T}[\rho_0] + \bar{V}_{ee}[\rho_0] = \int \rho_0(\mathbf{r})v(\mathbf{r})d\mathbf{r} + F[\rho_0] \quad (2.9)$$

con $F[\rho_0] \equiv \bar{T}[\rho_0] + \bar{V}_{ee}[\rho_0]$. Lo que hay que calcular ahora es la funcional $F[\rho_0]$. El primer paso para ello lo dan los mismos Hohenberg–Kohn con su teorema variacional; ellos demostraron que para una densidad electrónica, $\rho(\mathbf{r})$, que cumpla con $\int \rho(\mathbf{r}) = n$ y $\rho(\mathbf{r})d\mathbf{r} > 0$, es válida la desigualdad

$$E[\rho_0] \leq E[\rho] \quad (2.10)$$

es decir, la densidad electrónica del estado base minimiza la funcional de energía. El segundo paso se debe a Kohn y Sham,^[47] quienes idean un método para obtener ρ_0 y de ahí, E_0 . El método en sí es exacto, pero al emplearse funcionales aproximadas, como se hace, se vuelve aproximado. A grandes rasgos, el método de Kohn y Sham consiste en sustituir el sistema real por un sistema equivalente donde los electrones no interactúan entre sí, y el sistema no interactuante se aproxima al sistema real mediante el hamiltoniano $\hat{H}_\lambda \equiv \hat{T} + \sum_i v_\lambda(\mathbf{r}_i) + \lambda \hat{V}_{ee}$, donde el parámetro λ toma valores en el intervalo¹⁹ de 0 a 1. v_λ es un potencial externo que hace que el sistema con \hat{H}_λ genere una densidad electrónica igual a la densidad electrónica del sistema real. Kohn y Sham reescriben con estas ideas la **ecuación 2.9** de la siguiente forma²⁰

$$E[\rho_0] = \int \rho(\mathbf{r})v(\mathbf{r})d\mathbf{r} + \bar{T}[\rho] + \frac{1}{2} \iint \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{r_{1,2}}d\mathbf{r}_1d\mathbf{r}_2 + \Delta\bar{T}[\rho] + \Delta\bar{V}_{ee}[\rho] \quad (2.11)$$

donde se define $\Delta\bar{T}[\rho]$ como la diferencia entre la energía cinética electrónica del sistema real y el equivalente:

$$\Delta\bar{T}[\rho] \equiv \bar{T}[\rho] - \bar{T}_e[\rho] \quad (2.12)$$

y $\Delta\bar{V}_{ee}[\rho]$ como la diferencia entre las repulsiones electrón–electrón del sistema real y la repulsión electrostática interelectrónica.

$$\Delta\bar{V}_{ee}[\rho] \equiv \bar{V}_{ee}[\rho] - \frac{1}{2} \iint \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{r_{1,2}}d\mathbf{r}_1d\mathbf{r}_2 \quad (2.13)$$

¹⁹ Los límites son: 0 para el sistema no interactuante y 1 para el sistema real.

²⁰ El subíndice en ρ se omite.

con estas dos últimas ecuaciones se define la funcional de la energía de correlación e intercambio, $E_{\text{ci}}[\rho]$:

$$E_{\text{ci}}[\rho] \equiv \Delta \bar{V}_{ee}[\rho] + \Delta \bar{T}[\rho] \quad (2.14)$$

entonces se tiene que

$$E[\rho_0] = \int \rho(\mathbf{r})v(\mathbf{r})d\mathbf{r} + \bar{T}[\rho] + \frac{1}{2} \iint \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{r_{1,2}} d\mathbf{r}_1 d\mathbf{r}_2 + E_{\text{ci}}[\rho] \quad (2.15)$$

La funcional $E_{\text{ci}}[\rho]$ es la que debe aproximarse, entre mejor sea la aproximación más preciso será el cálculo. Con estas ecuaciones se calculan los orbitales de Kohn–Sham,²¹ ψ_i^{KS} , de los cuales se obtiene:

$$\rho = \rho_e = \sum_{i=1}^n |\psi_i^{\text{KS}}|^2 \quad (2.16)$$

el índice i hace referencia a todos los orbitales ocupados. En otras palabras, las funciones ψ_i^{KS} se obtienen de la ecuación monoeléctronica de Schödinger, $\hat{H}\psi_i^{\text{KS}} = \epsilon_i^{\text{KS}}\psi_i^{\text{KS}}$, en la que se ha substituido el potencial externo, $v(\mathbf{r}_i)$, por una función llamada *potencial efectivo* (ver **ecuación 2.15** y **2.21**) definido como:

$$v_{\text{ef}}(\mathbf{r}) = v(\mathbf{r}) + \iint \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{r_{1,2}} + v_{\text{ci}}(\mathbf{r}); \quad (2.17)$$

donde

$$v_{\text{ci}}(\mathbf{r}) = \frac{\delta E_{\text{ci}}}{\delta \rho} \quad (2.18)$$

Y aquí es donde surge la pregunta ¿cómo obtener $E_{\text{ci}}[\rho]$? Una forma es mediante la aproximación de la densidad local,²² LDA. Si ρ varía lentamente con la posición, la energía de correlación e intercambio está dada por

$$E_{\text{ci}}^{\text{LDA}} = \int \rho(\mathbf{r})\epsilon_{\text{ci}} d\mathbf{r} \quad (2.19)$$

ϵ_{ci} es la energía de intercambio más la de correlación, $\epsilon_{\text{ci}}(\rho) = \epsilon_c(\rho) + \epsilon_i(\rho)$, tomando la derivada funcional de $E_{\text{ci}}^{\text{LDA}}$ se obtiene el potencial de correlación e intercambio, v_{ci}

$$v_{\text{ci}}^{\text{LDA}} = \frac{\delta E_{\text{ci}}^{\text{LDA}}}{\delta \rho} = \epsilon_{\text{ci}}(\rho(\mathbf{r})) + \rho(\mathbf{r}) \frac{\partial \epsilon_{\text{ci}}(\rho)}{\partial \rho} \quad (2.20)$$

²¹ Funciones de onda monoeléctronicas.

²² Local Density Approximation.

que se utiliza para resolver

$$\left[-\frac{1}{2}\nabla_1^2 - \sum_{\alpha} \frac{Z_{\alpha}}{r_{1,\alpha}} + \int \frac{\rho(\mathbf{r}_2)}{r_{1,2}} d\mathbf{r}_2 + v_{\text{ci}}^{\text{LDA}} \right] \psi_i^{\text{KS}}(1) = \epsilon_i^{\text{KS}} \psi_i^{\text{KS}}(1). \quad (2.21)$$

La energía de intercambio tiene la forma

$$\epsilon_i = -\frac{1}{3} \left(\frac{3}{\pi} \right)^{1/3} [\rho(\mathbf{r})]^{1/3}; \quad (2.22)$$

para la energía de correlación suele emplearse una función calculada por Vosko, Wilk y Nusair, vWN.^[48]

$$E_c^{\text{vWN}}(\mathbf{r}) = A \left\{ \ln \frac{x^2}{X(x)} + \frac{2b}{Q} \tan^{-1} \frac{Q}{2x+b} - \frac{bx_0}{X(x_0)} \left[\ln \frac{(x-x_0)^2}{X(x)} + \frac{2(b+2x_0)}{Q} \tan^{-1} \frac{Q}{2x+b} \right] \right\} \quad (2.23)$$

con $x = \sqrt{\mathbf{r}_s}$, $X(x) = x^2 + bx + c$, $Q = (4c - b^2)^{-1/2}$ y $\mathbf{r}_s = (3/4\pi\rho(\mathbf{r}))^{1/3}(1/a_0)$.^[49]

En la práctica se emplea directamente el procedimiento elaborado por Kohn y Sham y una simplificación de éste hecha por J. Harris.^[50] La simplificación consiste en expandir la energía total de Kohn–Sham alrededor de una densidad $\rho_x(\mathbf{r})$ que es la suma de las densidades electrónicas traslapadas poco interactuantes²³ y se toman en cuenta sólo términos lineales de la expansión. Cada densidad se obtiene mediante el esquema de Kohn–Sham. En otras palabras, se calcula la densidad electrónica alrededor de cada átomo hasta cierta distancia fija, ésta será la contribución de cada átomo. En el enfoque de Harris para determinar la densidad electrónica en un punto x se suman las densidades individuales más una pequeña corrección debido a las perturbaciones mutuas de los átomos. Así, por ejemplo, si en el punto x se traslapan dos densidades se tendrá:

$$\rho_x^{\text{Harris}}(\mathbf{r}) = \rho_1(\mathbf{r}) + \rho_2(\mathbf{r}) + \delta\rho(\mathbf{r}) \quad (2.24)$$

La adición de $\delta\rho(\mathbf{r})$ es un intento de acercar $\rho_x^{\text{Harris}}(\mathbf{r})$ a la densidad real. Por tanto,

$$\rho_x^{\text{real}}(\mathbf{r}) \approx \rho_x^{\text{Harris}}(\mathbf{r}) \quad (2.25)$$

y cuando $\delta\rho(\mathbf{r})$ es muy pequeño

²³ Sin embargo, en este mismo artículo Harris, muestra que pueden obtenerse resultados razonables incluso para enlaces fuertemente covalentes.

$$\rho_x^{\text{real}}(\mathbf{r}) \approx \rho_1(\mathbf{r}) + \rho_2(\mathbf{r}) \quad (2.26)$$

Realmente, el sistema computacional utilizado en este trabajo emplea un método desarrollado por Z. Lin y J. Harris^[51] en 1993. Ellos emplean una expansión de orbitales moleculares esféricos de la densidad electrónica para generar la así llamada *funcional de Harris* (símil de la **ecuación 2.11**) empleando una densidad de prueba arbitraria. La dinámica del sistema se introduce mediante un lagrangiano que depende de los grados de libertad iónicos y electrónicos. Igualmente se emplea la aproximación Born–Oppenheimer en la DM. Al resolver el lagrangiano, se obtienen las ecuaciones dinámicas del sistema. Con el esquema de Lin y Harris se dan la coordenadas iniciales, se evalúa la funcional de Harris y finalmente se calculan las fuerzas de los cuerpos interactuantes, y así para cada paso de la simulación.

La ventaja de utilizar este desarrollo, es que la funcional se define con funciones atómicas y puede evaluarse con cualquier densidad de referencia, es decir, la densidad no es generada por el potencial externo y, en consecuencia, no es un método autoconsistente como el de Kohn y Sham; esta aproximación puede ahorrar tiempo en el cálculo de optimización de geometría y dinámica molecular^[51] y los resultados topológicos calculados son comparables con los datos experimentales.^[52]

2.3 FUNCIONES DE FUKUI

Una función de Fukui,^[53] FF, es un descriptor de sitios de reactividad, se define como la derivada del potencial químico, $\mu[n, v]$, con respecto al potencial externo pero, mediante las relaciones de Maxwell, puede representarse en términos de la densidad electrónica:

$$f^\alpha(\mathbf{r}) = \left. \frac{\partial \mu}{\partial v(\mathbf{r})} \right|_n = \left. \frac{\partial \rho(\mathbf{r})}{\partial n} \right|_{v(\mathbf{r})}. \quad (2.27)$$

La FF es una función normalizada:

$$\int f^\alpha(\mathbf{r}) d\mathbf{r} = 1 \quad (2.28)$$

El significado de $f^\alpha(\mathbf{r})$ es directo, mide que tan sensible es un sistema a la variación del potencial químico, en un punto, ocasionada por una perturbación externa. La $f^\alpha(\mathbf{r})$

expresada con $\rho(\mathbf{r})$ es la variación de la densidad electrónica cuando hay un cambio local en el número de electrones, así, el valor de la pendiente, que esta función determina, indica si una la región, centrada en el punto (\mathbf{r}) , puede sufrir un ataque electrofílico o nucleofílico. Si la pendiente es negativa, $f^-(\mathbf{r})$, está propensa a un ataque electrofílico; en cambio, si es positiva, $f^+(\mathbf{r})$, está propensa a sufrir un ataque nucleofílico.^[54] Esta última interpretación proviene de la siguiente aproximación:

$$f^+(\mathbf{r}) \approx \rho_{\text{LUMO}}(\mathbf{r}) \quad (2.29)$$

$$f^-(\mathbf{r}) \approx \rho_{\text{HOMO}}(\mathbf{r}) \quad (2.30)$$

El promedio de estas dos cantidades mediría la reactividad hacia un agente inocuo.

$$f^{\circ}(\mathbf{r}) \approx \frac{1}{2}[f^-(\mathbf{r}) + f^+(\mathbf{r})] \quad (2.31)$$

Las aproximaciones de las **ecuaciones 2.29** y **2.30** son llamadas también *funciones de frontera*^[67] por su relación con los orbitales de frontera. Las FFs son de gran utilidad porque permiten determinar, de forma cualitativa, la susceptibilidad de una molécula a reaccionar con un agente.

Las FFs surgen de forma natural de la teoría de funcionales de la densidad.^[53] En otras palabras, las FFs miden el cambio de la densidad electrónica, $\rho(\mathbf{r})$, debido a una cambio en el número de electrones. Entre más grande sea el valor de las FFs mayor será la susceptibilidad de la molécula a un ataque electro o nucleofílico, es decir las moléculas son más reactivas donde $f(\mathbf{r})$ sea grande.^[55] En el manual de DMol³^[56] presenta las siguientes ecuaciones como una aproximación discreta de las anteriores, pero el significado continúa siendo el mismo.

$$f^+(\mathbf{r}) \approx \frac{1}{\Delta N}[\rho_{N+\Delta}(\mathbf{r}) - \rho_N(\mathbf{r})] \quad (2.32)$$

$$f^-(\mathbf{r}) \approx \frac{1}{\Delta N}[\rho_N(\mathbf{r}) - \rho_{N-\Delta}(\mathbf{r})] \quad (2.33)$$

Estas fórmulas usan una aproximación de diferencias finitas en las que se utiliza $\Delta N = 1$, un electron es removido pero pueden utilizarse otros valores. Y de la misma forma la **ecuación 2.32** mide los cambios en la densidad cuando la molécula gana electrones, ataque nucleofílico, inversamente, la **ecuación 2.33** mide los cambios en la densidad cuando la molécula pierde electrones, ataque electrofílico.

2.4 FUNCIÓN DE DISTRIBUCIÓN DE PARES

La topología de un cristal se caracteriza por las posiciones atómicas regulares y periódicas. Es decir, en los sistemas cristalinos, es posible determinar, al menos teóricamente, cada una de las posiciones atómicas cuando se definen algunos pocos parámetros de posición y distancia. Por otro lado en los sistemas gaseosos o líquidos, y en general en los sistemas no periódicos, no es posible tal certidumbre. La forma de describir la topología de los sistemas no periódicos es estadística y probabilística, esto es, mediante funciones de distribución. Dos de las funciones comúnmente empleadas son: la función de distribución radial, RDF,²⁴ y la función de distribución de pares, PDF.²⁵ Estas dos funciones son equivalentes a través de una transformación. Existen otras funciones de distribución que tienen relevancia en ciertos contextos.

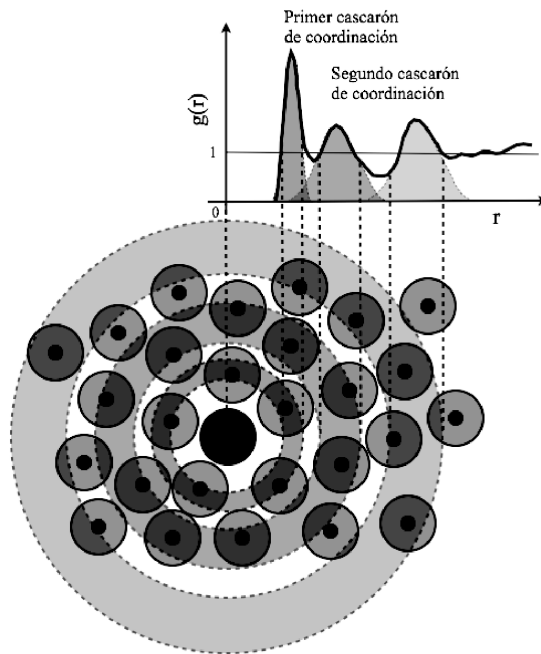


Figura 2.1 Esquema de la función de distribución de pares para un material monoatómico.
Cortesía de Zaahel Mata, zaahel@gmail.com.

La idea básica, y más sencilla, de la PDF es la de determinar la probabilidad de encontrar un átomo a la distancia r de otro átomo en la posición $r = 0$, ver **figura 2.1**. En los

²⁴ También conocida como $J(r)$. $J(r) = 4\pi r^2 \rho_o g(r)$.

²⁵ También conocida como $g(r)$.

sistemas reales²⁶ la PDF se obtiene de los datos de difracción y su definición tiene una justificación matemáticamente estricta.[57, 58] Nosotros consideramos que, dado que sólo trabajamos con sistemas obtenidos mediante simulación computacional, es mejor describir la forma en la que se obtiene tal distribución de probabilidad para estos sistemas.

En las simulaciones computacionales se tiene un conocimiento exacto de la posición espacial de cada átomo simulado, con esta información lo que se utiliza es un algoritmo para determinar la función de distribución. En él se elige un átomo y se calcula la distancia de todos los demás átomos con respecto a éste. Se repite este cálculo con cada uno de los átomos de la muestra. Con estas distancias se calcula un histograma, este histograma representará la RDF. Para obtener la PDF es necesario dividir este histograma por la función $4\pi r^2 \rho_o$, donde ρ_o es la densidad numérica, es decir, el número de átomos por volumen. Si la muestra (supercelda) es lo suficientemente grande, podría no ser necesario considerar las condiciones periódicas, pero si la muestra es pequeña, como sucede casi siempre en las simulaciones *ab initio*, existen varias formas de manejar la periodicidad. La primera consiste en reproducir la supercelda en todas las direcciones, es decir, colocar 26 superceldas alrededor de la celda original, como se muestra en la **figura 2.2** y tomar como puntos de origen sólo aquellos átomos dentro de la celda original.

Pero ésta no es la única forma de establecer condiciones periódicas, la otra es empleando funciones «módulo» y trasladar sólo algunos átomos. Por ejemplo, en el cálculo de la RDF, en lugar de medir las distancias, medir regiones y dividir entre el volumen medido, como se muestra en la **figura 2.3**. El volumen medido corresponde al área entre dos esferas concéntricas, con origen en un átomo dado. Esta forma quizá tenga una correspondencia más directa con la definición de RDF. Los dos algoritmos son equivalentes. En el **apéndice A.8** se presenta un programa escrito en lenguaje C que utiliza la duplicación de las superceldas para calcular la PDF.

Ahora bien, suponiendo que se pueden utilizar condiciones periódicas infinitas, o bien que se cuenta con una supercelda casi homogénea en densidad e infinita, al contar todos aquellos átomos que se encuentren a una distancia entre r y $r + dr$, éste número irá aumentando conforme uno se aleja del origen; al ser un conteo radial este número deberá ser proporcional a r^2 , y de forma más precisa a $4\pi r^2$. Cuando el conteo se hace de esta

²⁶ No simulados computacionalmente.

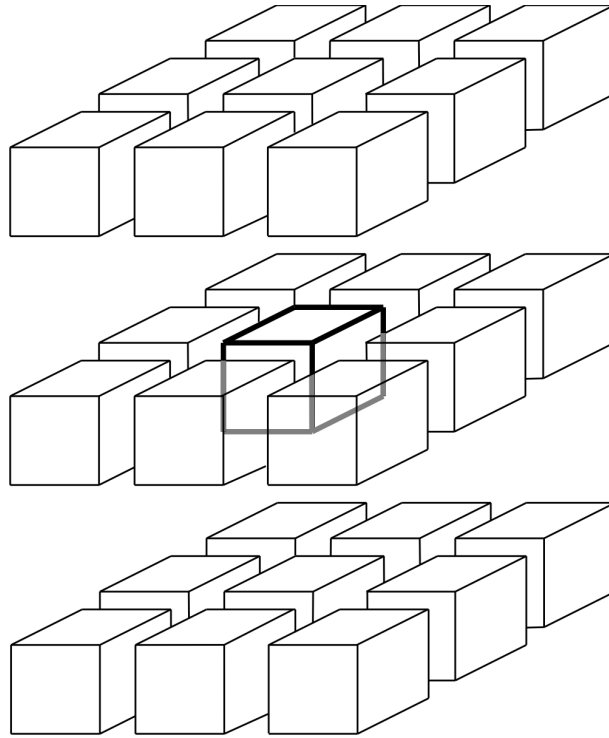


Figura 2.2 Condiciones periódicas para una supercelda. La celda central corresponde a la supercelda original. Las celdas están separadas para poder visualizar el arreglo.

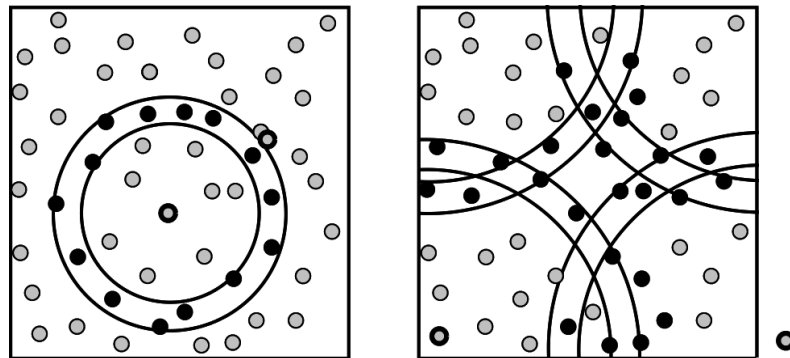


Figura 2.3 Alternativa para calcular la PDF. El casquete comprendido entre r y $r + dr$ obtenido para cada átomo se promedia. En el esquema del lado derecho se encuentra trasladado un átomo para reproducir las condiciones periódicas.

forma, lo que se obtiene es una RDF, mientras que sí el conteo se divide por $4\pi r^2$ se obtendrá una PDF, la PDF siempre se normaliza con la densidad numérica, para que cuando $r \rightarrow \infty$, $\text{PDF} \rightarrow 1$.

2.5 NÚMERO DE COORDINACIÓN MEDIANTE LA RDF

El número de coordinación es un parámetro especialmente importante para caracterizar materiales amorfos, debido a que, a diferencia de los materiales cristalinos el número de vecinos es variable. De especial importancia es el número de coordinación a primeros vecinos pues describe el ordenamiento a corto alcance y da una idea general de la estructura del material. De la definición se sabe que:

$$n = \int_{r_1}^{r_2} 4\pi r^2 g(r) dr \quad (2.34)$$

donde n es el número de coordinación que es estimado del área bajo la curva bajo el pico respectivo en la RDF. Generalmente el número de coordinación de los vecinos más cercanos es el que se utiliza en las discusiones de sistemas no cristalinos.

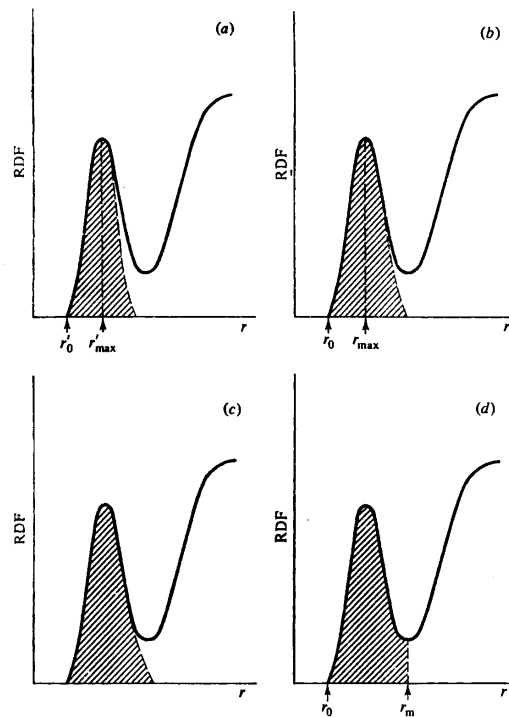


Figura 2.4 Diagramas esquemáticos de RDF para evaluar el número de coordinación. Tomado de Y. Waseda, *The structure of non-crystalline materials*, McGraw-Hill, EUA, 1980, pp. 49.

El problema en sí para el cálculo del número de coordinación es definir el intervalo de integración. Según Waseda^[57] hay cuatro métodos para calcular el número de coordinación a primeros vecinos basados en la RDF; todos consisten en definir qué porción del área del primer pico hay que tomar. La primera aproximación consiste en integrar hasta el máximo del primer pico de la curva $rg(r)$ y multiplicar por dos. La segunda aproximación consiste en integrar hasta el máximo del primer pico (curva $r^2g(r)$) y multiplicarlo por dos; la tercera aproximación consiste en la extrapolación de la curva $4\pi r^2g(r)$ para hacer el primer pico simétrico; y por último en la cuarta aproximación se integra hasta el mínimo de la RDF (entre el primero y el segundo pico). El cuarto método normalmente sobreestima el número de coordinación pues se integra un poco más allá del primer pico, pero éste como todos los demás, tienen su propia justificación y son igualmente válidos. En la **figura 2.4** se presenta un esquema de estas cuatro aproximaciones.

3 Desarrollo

En este trabajo empleamos dos técnicas para la generación de estructuras metálicas porosas:

1. Expansión de red. Nosotros hemos empleado este término como la traducción del término en inglés *lattice expansion*.
2. Desaleación. De la misma forma, ésta es la traducción que hemos empleado para el término en inglés *dealloying*.

Dada una supercelda con tamaño de arista a' (SC- a'), la primera técnica consiste en crear, a partir de ésta, otra supercelda con tamaño de arista a (SC- a) con la condición, que siempre debe cumplirse, de que $a > a'$; la SC- a es el resultado de modificar el volumen de SC- a' aumentando proporcionalmente la distancia entre los átomos.²⁷ Este procedimiento ocasiona que al calentar (recocer) la SC- a , los átomos tenderán a formar algún tipo de estructura diferente. Se espera que esta técnica, como se ha comprobado simulacionalmente genere estructuras porosas e incluso amorfas.^[59]

La segunda técnica es realmente la translación, al contexto simulacional, de una idea utilizada para generar estructuras nano y mesoporosas experimentalmente: la disolución selectiva de aleantes o *desaleación* (*dealloying*). En el campo experimental la disolución se realiza vía electroquímica, disolviendo con alguna sustancia, por ejemplo, ácido nítrico para la plata, uno de los aleantes, manteniendo el otro intacto.^[32, 34] La disolución selectiva de aleantes provoca que el aleante no disuelto conserve su posición espacial,²⁸ es decir, se genera un esqueleto del aleante no disuelto que corresponde a la posición de éste dentro de la aleación. Así, mediante un programa, escrito para este trabajo, se crea una supercelda del tamaño adecuado y aleatoriamente se elimina²⁹ cierta cantidad de átomos.³⁰ Por ejemplo, si se deseara crear una estructura desaleada de 100 átomos

²⁷ Esta técnica se identificará como *expansión de red*, ER.

²⁸ Esto no es del todo cierto, experimentalmente se observa la difusión superficial del aleante no disuelto. Siendo la velocidad de difusión dependiente de las condiciones experimentales.^[29, 34]

²⁹ Ver **sección 3.4**

³⁰ En lo que resta del texto, nos referiremos a esta técnica como *desaleación* mientras que para referirnos al procedimiento experimental utilizaremos el término *disolución selectiva*.

al 60% de porosidad, deberá empezarse con una supercelda de 250 átomos.³¹ *Sí bien la técnica que nosotros llamamos desaleación tiene como referencia la técnica experimental de disolución selectiva, nuestra técnica no pretende reproducir simulacionalmente a ésta.* Es decir, las condiciones bajo las cuales se generan y se simulan nuestras estructuras no reproducen el fenómeno experimental; comenzando porque la disolución selectiva parte de una aleación y nosotros, en la mayor parte de este trabajo (ver **sección 3.3**), partimos de elementos puros. En suma, en este trabajo hemos tomado la idea de la disolución selectiva para generar una estructura que sirva como precursor para formar un material amorfo y nanoporoso. Creamos computacionalmente un sistema sustitucional de dos componentes y eliminamos uno, entonces, este sistema «desaleado», por ser inestable, lo sometemos a una dinámica molecular *ab initio*. Hay tres diferencias esenciales entre el desaleado experimental y el precursor que nosotros formamos: 1. No consideramos una desaleación progresiva y la consecuente difusión atómica; 2. No consideramos distorsiones en la red debido a los aleantes. 3. Podemos crear estructuras en las que no participa ningún metal, como estructuras de Si o C puros.

Ahora bien, suponiendo que nosotros partimos también de una aleación y que «eliminamos» todos los átomos de uno de los aleantes, estos átomos se encuentran posicionados aleatoriamente, por el programa **aleaciones**, dentro de la aleación; lo que no siempre sucede en la realidad, incluso para aquellos aleantes que cumplen con las condiciones de solubilidad de Hume–Rothery, es decir, puede existir cierto arreglo espacial ordenado que no puede ser predicho;^[60] ni con estas condiciones. De cualquier forma, nosotros no podemos asegurar que para una aleación sustitucional real, aun para una tan soluble como la de Ag–Au, el ordenamiento de los aleantes sea totalmente aleatorio. Pero nuestra mejor suposición es que sí lo sea.

Estas dos técnicas de construcción tienen en común el disminuir la densidad de las superceldas y el controlar el porcentaje de porosidad. Y aunque, la primera técnica, no tenga un símil directo en el plano experimental, no debe ni puede descartarse como un método útil para generar estructuras porosas a nivel simulacional, pues lo que interesa aquí es producir estructuras metálicas porosas *al margen de la técnica experimental que se emplee para ello*.

³¹ Considerando que todos los átomos de la supercelda poseen el mismo tamaño volumétrico.

Con las superceldas generadas por medio de estas dos técnicas se realizará una dinámica molecular, cuyas características se detallan en la **sección 3.7**.

El número de átomos en este tipo de simulaciones no puede ser mayor de varios cientos, por limitaciones de tiempo, computacionales y de código, por lo que planeamos trabajar con superceldas de 256 átomos para la técnica ER, esto corresponde a 4 veces la arista de una celda FCC. En el caso de la técnica de desaleación,³² el número de átomos empleados trataremos se acerque a los valores anteriores, para fines comparativos, pues la construcción de las celdas en este caso es función de la porosidad requerida, ver **sección 3.2**. Por ejemplo, para una aleación FCC desaleada al 50% los valores más cercanos serán de 128 y 250 átomos que corresponden a 4 (256 átomos) y 5 (500 átomos) veces la arista de una celda unitaria, respectivamente. Debido al límite en el número de átomos, consideramos que con nuestras estructuras sólo podemos representar estructuras reales cuyo tamaño de poro sea de unos cuantos nanómetros. En las siguientes secciones se darán más detalles de las características de la simulación.

Efectuada la dinámica molecular en cada estructura, se buscará asegurar el mínimo local de energía, para cada átomo, mediante un proceso de optimización; la optimización es, a grandes rasgos, un método de minimización no lineal aplicado a la superficie de energía de la estructura final de la dinámica molecular.

3.1 EXPANSIÓN DE RED

En este trabajo intentamos simular una estructura metálica porosa y, de ser posible, amorfa. La primera técnica que utilizaremos es llamada expansión de red, ER.^[59] Esta técnica es parte del procedimiento utilizado tradicionalmente en el grupo de trabajo para generar estructuras porosas y amorfas de Si, C, Ge, entre otros. Como mencionamos anteriormente, la técnica consiste en modificar artificialmente la densidad del material con el que se trabaja, esto es, se varía el volumen de la estructura conservando la misma cantidad de átomos. Se pueden imaginar dos formas esenciales de lograr esto, la primera consiste en mantener las distancias atómicas de la estructura y variar sólo el tamaño de la supercelda, como se muestra en la **figura 3.1**;

³² En algunos casos empleamos la desaleación asimétrica para cumplir con este objetivo, ver **sección 3.2**.

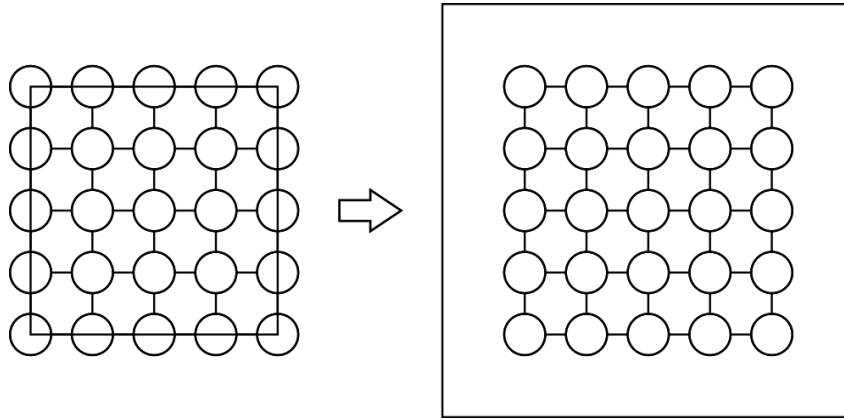


Figura 3.1 Forma esquemática de la técnica de expansión de red, manteniendo las distancias de enlace y variando el tamaño de la supercelda.

y la segunda es variando el tamaño de la supercelda al mismo tiempo que se varían las distancias interatómicas, **figura 3.2**.

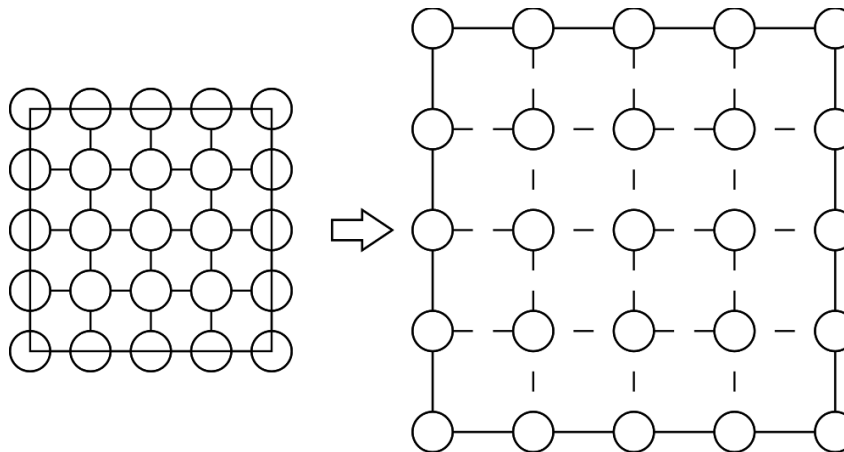


Figura 3.2 Forma esquemática de la técnica de expansión de red, variando las distancias interatómicas y el tamaño de la supercelda.

El tamaño de la arista, ya sea de la celda unitaria o de la supercelda, se calcula mediante la fórmula:

$$l = \left[\frac{m}{x\rho_0} \right]^{1/3} = \left[\frac{m}{\rho} \right]^{1/3} \quad (3.1)$$

donde l es la longitud final de la arista de la celda, m es la suma de la masa de todos los átomos de la celda, ρ_0 es la densidad de la celda antes de ser expandida y x es la variación

de la densidad, es decir, si se quiere expandir hasta generar un 50% de vacuidad,³³ la celda deberá de tener el 50% de su densidad original y entonces x debe ser igual a 0.50. Ahora bien, podría simplemente substituirse $x\rho_o$ por la densidad deseada, ρ , y así obtener una l determinada. Como puede apreciarse, esta fórmula tiene su origen en la definición de la densidad.

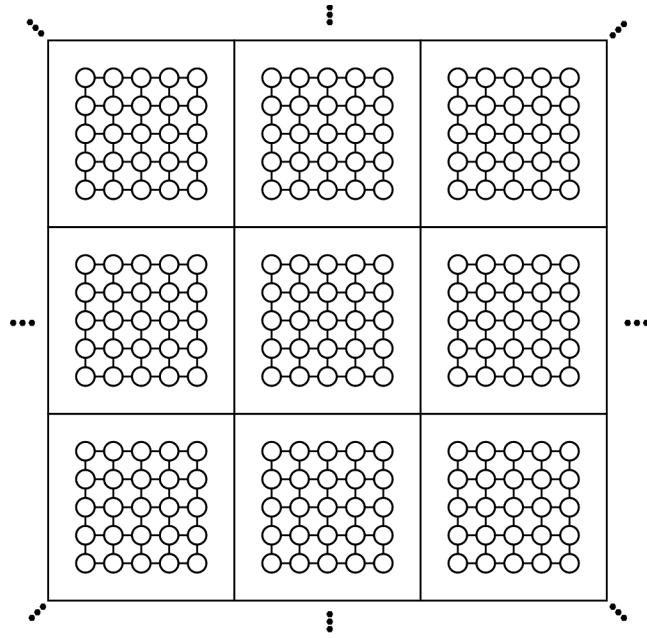
Si se consideran las condiciones periódicas, el primer tipo de ER generará una superestructura con grupos aislados de átomos (cúmulos) y entre más se disminuya la densidad, más aislados estarán, ver **figura 3.3(a)**. En el segundo tipo de ER, se genera una estructura homogénea en la que, si la densidad es lo suficientemente baja, cada átomo de la estructura estará desenlazado, en la **figura 3.3(b)** se muestra este caso.

La forma de la técnica de ER que empleamos en este trabajo es la segunda: variar el volumen variando las distancias interatómicas. La estructura generada mediante esta segunda variación de la técnica de ER es la estructura que se somete a la dinámica molecular. Finalmente, con esta técnica puede controlarse de forma exacta la cantidad de átomos de la estructura final atendiendo, claro está, la cantidad de átomos permitidos en la supercelda, es decir, el número de átomos siempre será igual a:

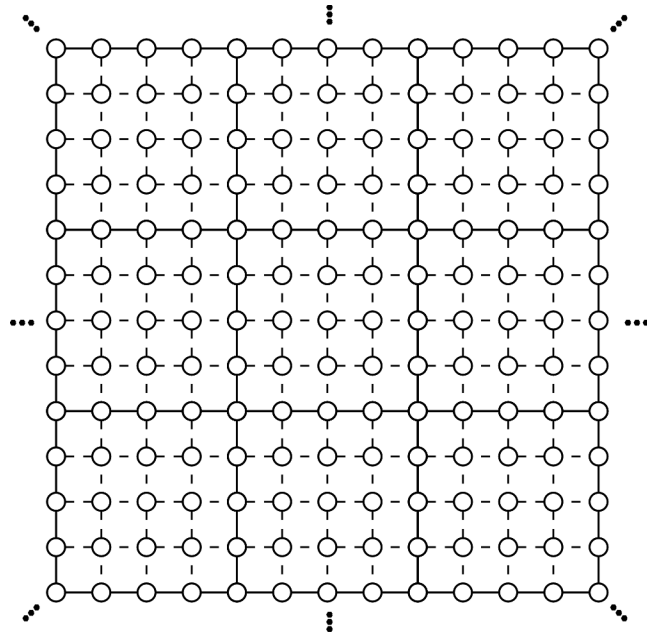
$$\#\text{átomos ER} = n^3 d \quad (3.2)$$

donde n es el número de celdas unitarias por lado y d es el número de átomos por celda unitaria.

³³ Debido tanto a razones energéticas durante la creación de superficies dentro de la supercelda, y la estructura y densidad del «backbone», no puede asegurarse que la estructura total corresponda a un 50% de porosidad con respecto a una estructura sin un solo poro; la relación que asumimos en general es % porosidad ≈ 100 (1-fracción de densidad)%.



(a)



(b)

Figura 3.3 Forma periódica esquemática de la técnica de expansión de red, a) Grupos de átomos aislados. b) Estructura homogénea con posibles átomos desenlazados.

3.2 DESALEACIÓN

La segunda técnica que empleamos en este trabajo la hemos llamado «desaleación».³⁴ Consiste en generar una supercelda y remover aleatoriamente cierta cantidad de átomos y así obtener la densidad deseada. De esta forma puede hablarse de porcentaje de desaleación o porosidad o de variación de la densidad,³⁵ **figura 3.4**.

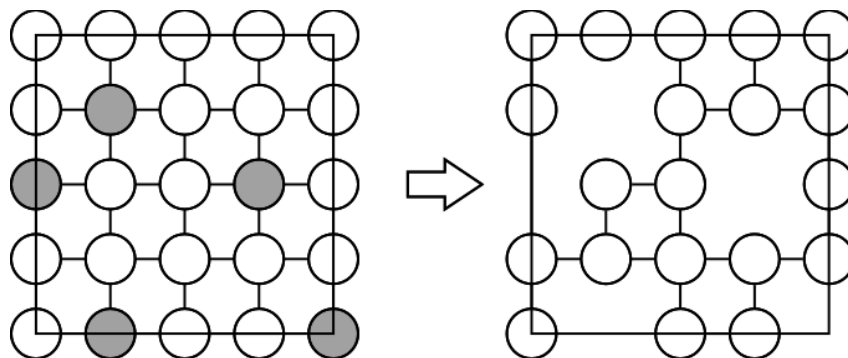


Figura 3.4 Forma esquemática de la técnica de desaleación.

De la misma manera que sucede con la técnica de ER, en la desaleación hay dos formas de obtener estructuras desaleadas. La primera consiste en establecer el porcentaje de desaleación y la cantidad de átomos «aproximados» que se desea y después calcular las dimensiones de la supercelda de la que se necesita partir. Por ejemplo, si se desea generar una estructura FCC desaleada al 20% con aproximadamente 200 átomos, es necesario partir de una supercelda, formada de 64 ($4 \times 4 \times 4$) celdas unitarias FCC, con 256 átomos; la supercelda ya desaleada contendrá 204 átomos. En el **apéndice A.3** se presenta una tabla para superceldas con estructura FCC en la que se muestra la cantidad de átomos resultantes como función del número de celdas unitarias, el porcentaje de desaleación y el número de átomos por celda unitaria. Los valores en esta tabla se calculan mediante la fórmula

³⁴ Una vez más, en este punto, es importante resaltar que si bien la técnica que nosotros hemos llamado «desaleación» tiene rasgos en común con la técnica de *disolución selectiva* o *desaleación experimental*, estas dos no son equivalentes, entre otras razones porque no todas las aleaciones son sustitucionales y en caso de serlo, el ordenamiento atómico no puede ser completamente al azar pues está sujeto al ambiente electrónico de los aleantes, incluso para aleantes tan parecidos como la Ag y el Au. En el caso de aleaciones no sustitucionales o multifásicas, aleaciones que no consideramos en este trabajo, las posiciones atómicas son aún más restringidas y menos aleatorias.

³⁵ Ver **nota 33**.

$$\#\text{átomos desaleados} = (1 - x)n^3d \quad (3.3)$$

donde n es el número de celdas unitarias por lado, x es la fracción de desaleación y d es el número de átomos por celda unitaria, cuando la construcción es a partir de celdas FCC, $d = 4$, cuando es a partir de BCC, $d = 2$ y cuando es a partir de CS, $d = 1$.

En el caso de la ER la cantidad de átomos siempre es proporcional a n^3 para cualquier porcentaje de porosidad. Lo que permite comparar fácilmente las estructuras con diferentes porosidades. Para esta primera forma de desaleación esto no es posible. Por ejemplo, si se quiere comparar las celdas desaleadas con aproximadamente 108 átomos, las únicas celdas posibles serían con 86, 100, 150, 102, 128, 153, 75, 86, 97 átomos (ver **apéndice A.3**) para 90—10% de desaleación, es decir, en el caso extremo, se estaría comparado una celda de 150 átomos con otra de 75. La forma de solucionar esto es no restringiendo las dimensiones de la supercelda a ser proporcionales a n^3 ; sino, hacerla proporcional a $l \times n \times m$. Lo que podría acarrear otro problema: podría darse el caso en que una de las dimensiones sea mucho menor a las otras dos, o bien, una de ellas mucho mayor a las otras dos. Esto es, se podría provocar estructuras anisotrópicas de forma directa y no aleatoria (por la desaleación y la dinámica molecular). Pero antes de continuar, mostraremos de forma esquemática en que consiste esta solución a la que llamaremos «desaleación asimétrica» y que constituye la segunda forma de desaleación. En la figura se observa que la estructura inicial antes de la desaleación es de dimensiones 6×4 y no de 4×4 , 6×6 o bien de 5×5 (n^2) como en la figura anterior

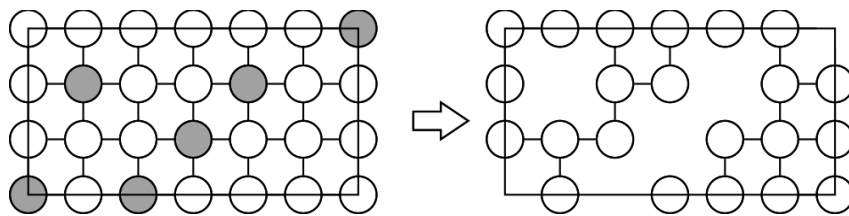


Figura 3.5 Forma esquemática de la técnica de desaleación asimétrica.

La manera de disminuir esta anisotropía es utilizando aquellos valores de l , m y n para los que se cumpla que $|l - m| \approx |m - n| \approx |l - n|$. En el **apéndice A.4** se muestra un programa escrito en lenguaje **Python** con el cuál se determina, de forma poco elegante, las dimensiones de la supercelda dado el sistema cristalino, el porcentaje de desaleación y el número de átomos deseados después del desaleado. En el **apéndice A.5** se presenta

una tabla, calculada con este programa, en la que se dan las dimensiones de la supercelda como función del número de átomos, y el porcentaje de desaleación para una estructura FCC.

Cualquiera de las dos formas de desaleación, ya sea mediante una supercelda proporcional a n^3 o proporcional a $l \times n \times m$, presenta algunas particularidades en comparación con las superceldas generadas mediante la ER. La primera es que el número de átomos no es fijo al variar el porcentaje de porosidad. La segunda particularidad es que casi nunca³⁶ los átomos se encuentran desenlazados; a excepción de porcentajes de porosidad muy altos, en los que, estadísticamente, no todos los átomos pueden estar enlazados y estar distribuidos en el volumen de la supercelda al mismo tiempo.

3.3 DESALEACIÓN PROGRESIVA

La idea de la desaleación está inspirada en la técnica experimental del mismo nombre que consiste en disolver, con algún agente corrosivo, el aleante más activo de los que componen una aleación. Experimentalmente la desaleación ocurre siempre sobre la superficie, al desalearse la superficie se generan «nuevas» superficies que a su vez pueden ser desaleadas. Durante la creación de la nueva superficie, ocurre una difusión superficial (reacomodo) de los átomos no desaleados tanto pasivos como activos. Siguiendo esta descripción lo *ideal*, en la simulación, sería realizar un «desaleado progresivo», es decir, retirar los átomos superficiales, realizar una dinámica molecular u optimización estructural, y después, retirar los átomos que han quedado en las «nuevas» superficies y nuevamente realizar dinámica molecular u optimización, y así hasta optimizar la última estructura en la que no quede ningún átomo del aleante pasivo. Este procedimiento tiene varias desventajas, la primera es que, debido a que se debe desalea desde la superficie, sería necesario trabajar con clústers, cosa que no pretendemos en este trabajo; segunda, si se quisiera no trabajar con clústers, sino con estructuras más grandes, es necesario aumentar el número de átomos, a mucho más de 500 lo que haría prohibitivo el cálculo computacional para simulaciones *ab initio*, como las que se realizan en este trabajo; tercero, la otra opción sería trabajar con estructuras periódicas, lo que tiene el inconveniente de eliminar toda superficie inicial y cualquier intento de reproducir el desaleado experimental ocurriría dentro del volumen

³⁶ «Casi nunca» significa que puede darse el caso en el que hayan, debido a la aleatoriedad, átomos aislados.

y no desde la superficie; finalmente, cuarto, ¿cómo definir una superficie?, ¿una superficie está definida por una capa monoatómica, dos, tres o más? ¿y si los átomos no están perfectamente ordenados?, ahora bien, se podría definir una superficie en función de la interacción de una especie química que no pertenezca a ella, lo que nos regresa al punto dos.

Ahora bien, quizá el punto tres sea la opción más viable ya que no se requieren muchos más átomos de los que normalmente se utilizan en este tipo de simulaciones (*ab initio*) y porque tampoco se trabajaría con clústeres. De esta forma, el procedimiento esquematizado sería el siguiente, **figura 3.6**:

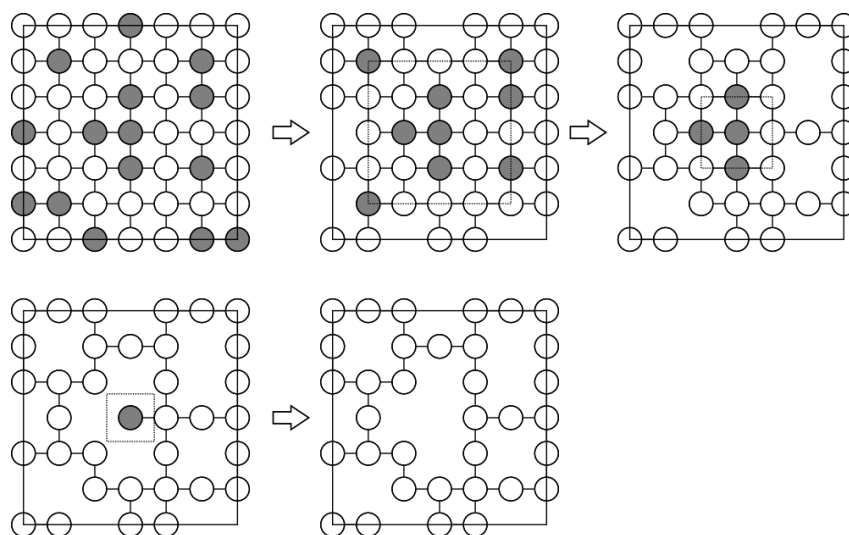


Figura 3.6 Forma esquemática de la técnica de desaleación progresiva.

En este esquema no se toma en cuenta el reajuste atómico, lo que realmente es relevante, pero se representa la eliminación de los átomos desde el borde de la supercelda y hasta el su centro, capa por capa. En cada etapa de eliminación (desaleado) deberá realizarse una dinámica molecular o una optimización, lo que haría sumamente lenta la simulación. Ahora, suponiendo que no existiese la limitante de tiempo computacional, el reajuste de los átomos podría impedir la definición de capas atómicas. Una solución práctica sería quitar, en cada etapa, todos aquellos átomos que se encuentre a cierta distancia del límite de la celda, tal como se muestra en la siguiente figura.

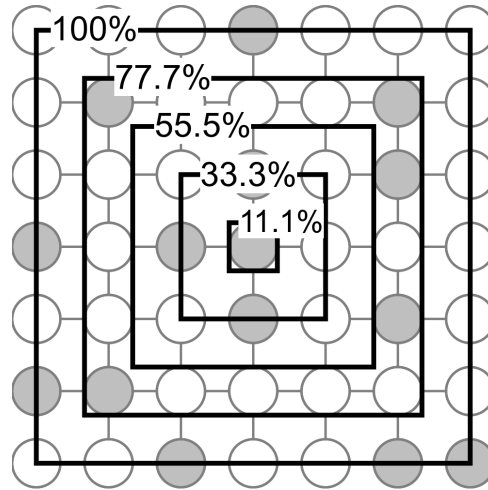


Figura 3.7 Desaleación progresiva. Cada cuadro marca la distancia, porcentaje, del centro de la celda a una de sus caras.

En la figura se muestran cuadros concéntricos, la arista de cada cuadro se encuentra a la distancia, en porcentaje, marcada en la figura, del apotema. Lo ideal sería que cada cuadro se encontraría, separado de otro, a una distancia del orden del tamaño atómico. Esta técnica no la empleamos por considerarla inconsistente y poco factible para simulaciones *ab initio*.

La implementación de esta técnica nos regresa al problema planteado en la **nota 33**, el porqué una estructura con, digamos, 50% de su densidad no genera una estructura 50% porosa. Es sabido que los átomos en una superficie no se comportan de la misma manera que aquellos dentro del volumen; ya sea que no tengan un ordenamiento definido, que la distancia de enlace sea mayor o menor en comparación a los otros, etcétera. Lo importante es saber cómo definir aquellos átomos que conforman un cuerpo sólido de aquellas que se encuentran en la superficie y delimitan poro, para saberlo, en primer lugar, sería necesario definir los límites de la estructura, es decir una superficie, y en base a esto, el volumen sólido. Ahora, para definir una superficie es necesario definir el tamaño de un átomo en particular y el volumen que se considere ocupe en el espacio. En ocasiones se definen las superficies como las definidas por las envolventes de los radios de solvatación. Pero podrían adoptarse otras *convenciones*. Nosotros preferimos no entrar en estos detalles al momento de definir porosidad y simplemente adoptamos la convención de definir el porcentaje de porosidad como igual al porcentaje de disminución de la densidad para una especie pura,

es decir, consideramos, para el caso de una estructura desaleada, como la densidad base aquella del elemento que formará la estructura final.

3.4 GENERACIÓN DE SUPERCELDAS

Las estructuras iniciales para la dinámica molecular, tanto las generadas por la técnica de ER como las del método de desaleado, se construyen mediante un programa que hemos desarrollado especialmente con este fin. El código de dicho programa se presenta en el **apéndice A.6**, también su uso se presenta en este apéndice; esta sección sólo describe el fundamento en el que está basado y sus características principales. El programa, que hemos llamado **aleaciones**, puede generar una supercelda, a partir de las estructuras cristalinas más comunes, FCC, BCC, HCP, diamante, . . . , tomando como base celdas unitarias individuales; esto es, puede generar estructuras cristalinas de dimensiones $l \times m \times n$, en las que l , m y n son los números de celdas unitarias por lado de la supercelda. Los átomos constituyentes de esta supercelda son de dos tipos como máximo, la ubicación espacial de ambos tipos de átomos dentro de la celda es aleatoria, por supuesto, dentro de las posiciones que permite la red. La aleación generada mediante este programa es de tipo sustitucional, generar otro tipo de aleaciones requeriría modificar el programa. Así mismo, el motivo de la estructura cristalina es un sólo átomo, esto es, también se requiere modificar el programa para manejar otro tipo de motivos. En la siguiente figura se muestra la supercelda de una aleación generada con este programa.

Como puede observarse, al aumentar únicamente el parámetro de red pueden generarse superceldas expandidas de aleaciones o de elementos puros, si se hace el soluto igual al disolvente; todas las estructuras de las que partimos en este trabajo son de este último tipo: el soluto se hace igual al disolvente, esto equivale a decir, que nuestro método de desaleación consiste, en realidad, de la eliminación aleatoria de cierta cantidad de átomos dentro de la red. Para generar estructuras desaleadas sólo debe eliminarse uno de los aleantes, cosa que puede hacerse directamente en el programa. En la **figura 3.9** se presenta una de tales estructuras desaleadas.

Nosotros concebimos originalmente la técnica de desaleado siguiendo los siguientes pasos:

1. Crear una aleación sustitucional aleatoria.

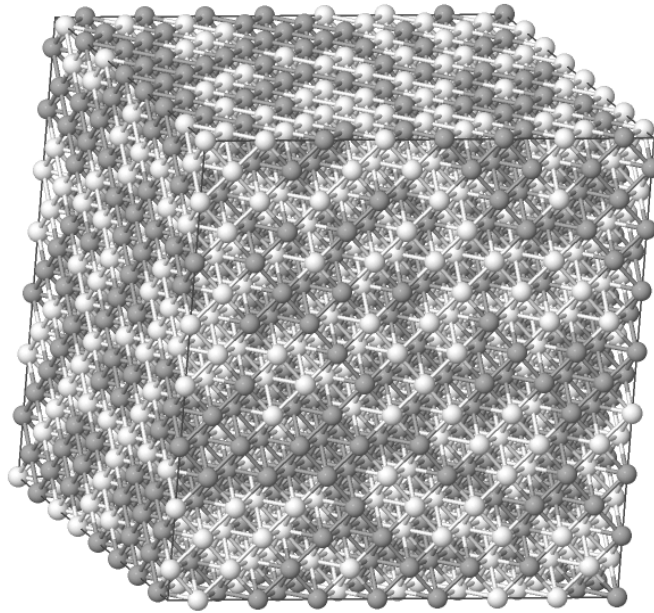


Figura 3.8 Supercelda de la aleación sustitucional Au-Ag al 50%, generada con el programa `aleaciones`. Formada a partir de 8 celdas unitarias, de tipo FCC, por lado con un parámetro de red igual a 4.0857 \AA cada una (32.6801 \AA arista de la supercelda). 2048 átomos totales.

2. Realizar dinámica molecular.
3. Buscar el mínimo de energía.
4. Desalear: de lo que se obtienen dos subsistemas, cada uno formado con un aleante.
5. Realizar dinámica molecular acada uno de estos dos subsistemas.
6. Buscar mínimo de energía de cada subsistema.

Pero ¿cuál es la diferencia entre ésta y el desaleado en el que se elimina uno de los componentes de una aleación sustitucional generada por el programa `aleaciones`? La diferencia principal consite en permitir que los átomos dentro de la aleación ocupen posiciones deteminadas por el sistema y no asignadas por nosotros. Nosotros no consideramos esta aproximación en nuestro tabajo principalmente por dos razones, la primera es el número de átomos utilizados, el número de átomos de la aleación original deberá ser mayor

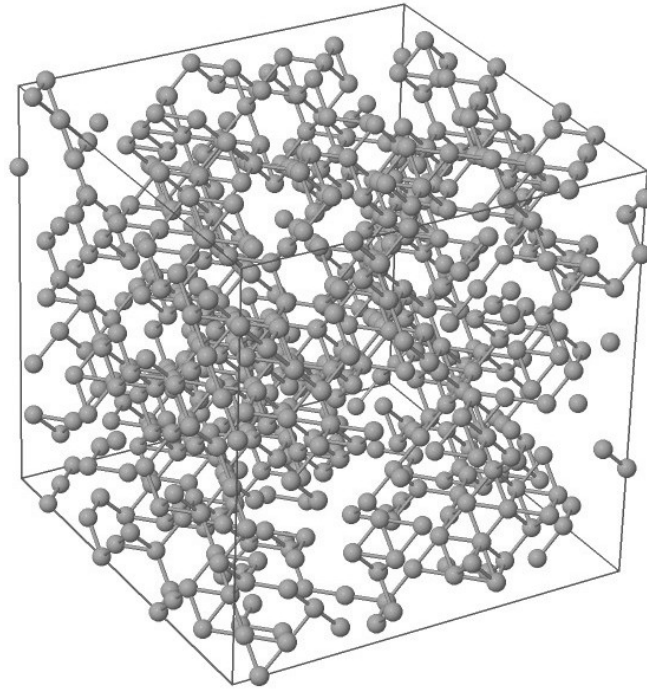


Figura 3.9 Supercelda de Au desaleada al 70% a partir de una aleación Ag-Au generada con el programa aleaciones. Formada a partir de 8 celdas unitarias, de tipo FCC, por lado, con un parámetro de red igual a 4.0857 Å. cada una (32.6801 Å arista de la supercelda). 614 átomos totales.

de 500, sobre todo bajas desaleaciones, si se quiere tener sistemas desaleados de alrededor de 250 átomos. La segunda razón es la movilidad atómica a 300 K, que es la temperatura que nosotros utilizamos en nuestras simulaciones, no es lo suficiente grande para modificar significativamente las posiciones atómicas de la aleación original. Para mostrar esto último nosotros generamos dos aleaciones Ag-Au50% idénticas de 256 átomos; con la primera de ellas obtenemos dos sistemas, uno de Au y otro de Ag, cada uno de 128 átomos y realizamos 500 pasos de dinámica molecular a 300 K y optimizamos; con la segunda la segunda también realizamos 500 de DM y optimización, al término de la cual obtenemos dos sistemas, uno de Ag y otro de Au de 128 átomos cada uno, y de la misma forma se realizan 500 pasos de DM y luego optimización. Es decir, al final se obtiene 4 subsistemas dos de Au y otros dos de Ag, dos provienen de una aleación sin DM y los otros dos de una DM. El esquema de la **figura 3.10** resume esto.

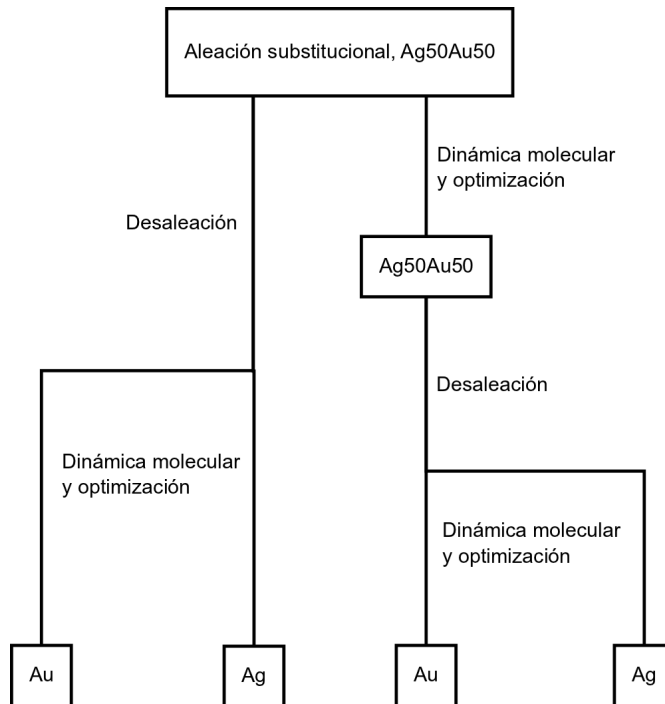


Figura 3.10 Esquema de la obtención de sistemas desaleados. La rama izquierda del diagrama corresponde al método utilizado en este trabajo. La rama derecha representa el método concebido originalmente. No se obtiene una diferencia significativa utilizando cualquiera de los métodos, aunque la rama derecha es un método más costoso computacionalmente.

Utilizando la función de distribución de pares, PDF, es como discernimos el comportamiento de las cuatro estructuras, y las presentamos a continuación.

No existen diferencias significativas en las estructuras obtenidas por ambos métodos, por lo que hemos elegido la desaleación según se muestra en la rama izquierda, pues entre otras cosas es menos costosa computacionalmente.

3.5 DISTRIBUCIÓN DE VELOCIDADES

En la dinámica molecular, tal y como está descrita en la **sección 2.1**, al principio del primer paso de simulación, a los átomos de la estructura les debe ser asignada una velocidad aleatoria de acuerdo a la distribución de celeridades Maxwell-Boltzmann, DVMB. El

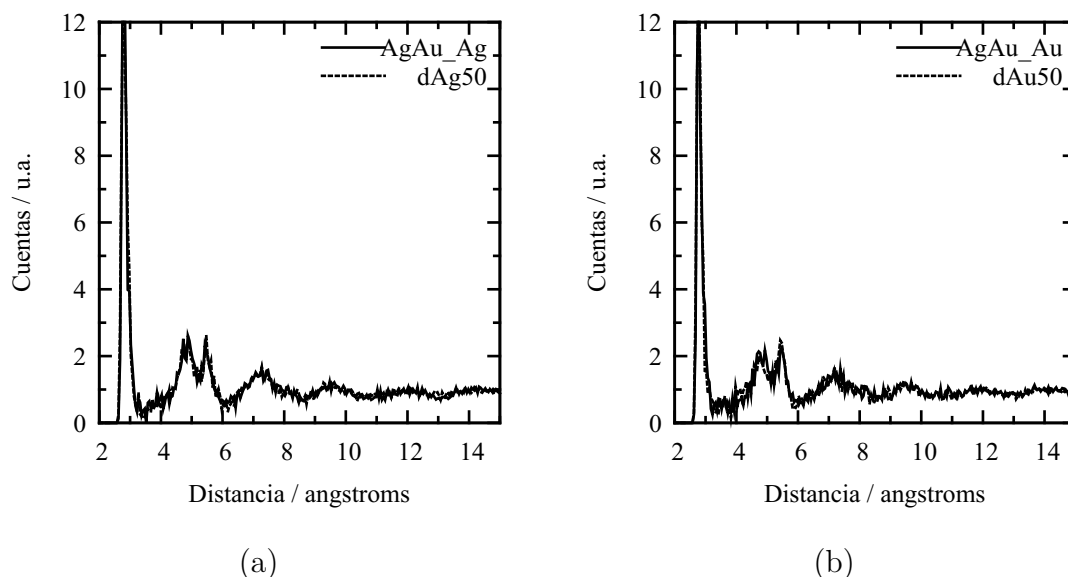


Figura 3.11 PDF para Ag y Au desaleados a 300 K generados a partir de una aleación sometida a dinámica molecular (AgAu_Ag y AgAu_Au) y una sin dinámica molecular (Ag y Au).

grupo de paquetes informáticos, **Materials Studio**® versión 3.2, que nosotros utilizamos para realizar nuestras simulaciones, no asigna estas velocidades dentro de sus rutinas, es decir, nosotros, el usuario, debemos asignárselas manualmente. Para esto hemos desarrollado un programa llamado **dvmb**, ver **apéndice A.7**, que genera velocidades aleatorias cuyo módulo siguen la DVMB. En la parte final del **apéndice A.1** se muestra la salida de este programa para un supercelda de 230 átomos.

El algoritmo empleado para este programa es bastante sencillo, consiste en la generación de 5 números aleatorios, tres de ellos siguen una distribución uniforme y se utilizan para generar las coordenadas de una dirección aleatoria. **Figura 3.12**.

Los otros dos números son utilizados en un método tipo Monte Carlo,^[66] MC, llamado de aceptación–rechazo,³⁷ la finalidad de este método generar una variable aleatoria con una función de densidad de probabilidad, f.d.p., arbitraria, es decir, generar un número aleatorio que siga una distribución arbitraria.³⁸ Textualmente la referencia anterior la define como:

³⁷ Acceptance–Rejection Method

³⁸ Generalmente una distribución poco convencional. Nosotros aplicamos este método por ser fácil de programar y muy intuitivo.

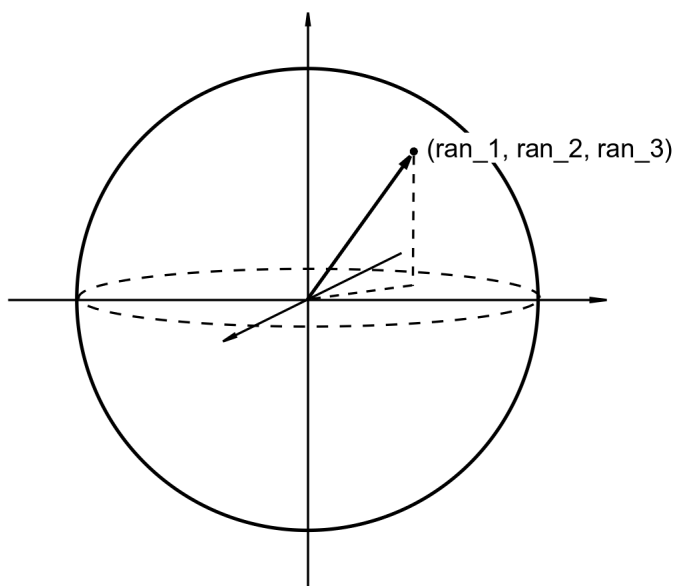


Figura 3.12 Cualquier punto, elegido de forma aleatoria, de la superficie de la esfera representa una dirección posible. ran_1, ran_2 y ran_3 representan un número aleatorio dentro del rango continuo $[-1, 1]$.

$\{f(z), a \leq z \leq b\}$ es una f.d.p.³⁹ con factorización:

$$f(z) = cg(z)h(z),$$

donde

$$h(z) \geq 0; \int_a^b h(z)dz = 1; c = \sup_z [f(z)/h(z)]; \text{ y } 0 \leq g(z) \leq 1.$$

Z denota una variable aleatoria con una f.d.p. $\{h(z)\}$ y U es otra variable aleatoria de con una f.d.p. uniforme en el rango $[0, 1]$. Si $U \leq g(Z)$, entonces Z tiene una f.d.p $\{f(z)\}$.

La función $h(z)$ es un f.d.p. de la que se sabe como obtener la variable aleatoria y $g(z)$ la f.d.p. de la que se desea obtener. De aquí, el algoritmo es directo, se generan dos números aleatorios, el primero, Z , que pertenezca a $h(z)$ y el segundo, U , que pertenezca a una distribución uniforme, con estos se calcula $g(Z)$ y se repite esto hasta que se encuentre un U y un Z tales que $U \leq g(Z)$. Este número Z será un número que pertenece a $\{g(z)\}$. Ahora, lo más sencillo, aunque no lo más eficiente, es utilizar una $h(z)$ que sea una

³⁹ Función de densidad de probabilidad.

distribución uniforme, haciéndolo así, se generan dos números aleatorios con distribución uniforme. Todo esto se puede imaginar como una curva con forma de $g(z)$ dentro de un recuadro, $h(z)$. Entonces, cualquier punto dentro de este recuadro tendrá las coordenadas $(Z, h(z))$, todos aquellos puntos contenidos en el área delimitada por $g(z)$ son los que siguen la distribución $g(z)$. En el programa DVMB hacemos $g(z)$ igual a la distribución de Maxwell-Boltzmann. En la **figura 3.13** se muestra esquemáticamente este algoritmo.

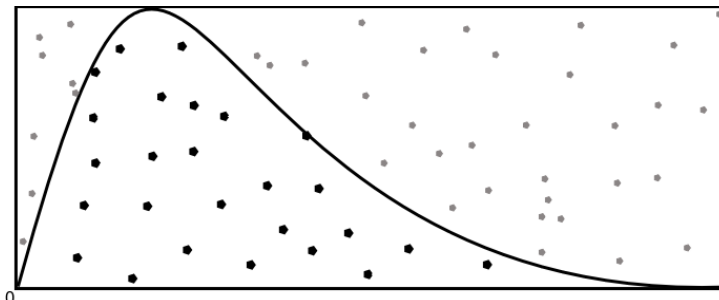


Figura 3.13 Forma esquematizada del método aceptación-rechazo, empleado en el programa DVMB. El objetivo es generar una variable Z con una distribución $g(z)$ de Maxwell-Boltzmann. La densidad está acotada superiormente por una cantidad H y si U sigue una distribución uniforme en el intervalo $[0, 1]$, aceptaremos el valor de U como parte de $\{g(z)\}$ si $U \leq g(z)$.

3.6 ELEMENTOS SIMULADOS

Los elementos químicos que nosotros hemos elegido para realizar nuestras simulaciones son: Au, Ag y Cu. La estructura cristalina de estos tres metales es FCC y esa es la que hemos empleado. Creamos superceldas con cada uno de estos elementos variando la porosidad (desaleado) desde 10% y hasta el 90% de 10 en 10%. La base de cálculo en nuestras simulaciones es de 250 átomos, es decir, para la técnica de expansión de red hemos utilizado 256 átomos; para la técnica de desaleado hemos intentado mantener este número, como se indica arriba, mediante desaleación «simétrica»⁴⁰ y la desaleación asimétrica. En la **tablas 3.1, 3.2, 3.3 y 3.4** se resumen las características geométricas de todas

⁴⁰ En la que se mantienen el volumen de la supercelda proporcional a n^3 .

las celdas utilizadas.⁴¹ La densidad base que hemos utilizado para nuestros cálculos es 8.96 g cm^{-3} para el cobre, 10.49 g cm^{-3} para la plata y 19.3 g cm^{-3} para el oro.^[61] Y los parámetros de red de la celda unitaria para las superceldas desaleadas son: 3.6149, 4.0853, 4.0782 Å para Cu,^[62] Ag^[63] y Au,^[64] respectivamente. Para facilitar el referirnos a los sistemas simulados establecemos las convenciones que se enumeran a continuación.

- Los sistemas desaleados son precedidos por una «d»: dCu, dAg, dAu.
- Los dos sistemas en los que se utiliza la técnica ER son precedidos por una «e»: eCu, eAg, eAu.
- De aquí se establecen los sistemas base Cu: dCu, eCu; base Ag: dAg, eAg; y base Au: dAu, eAu.
- Cuando se nombra un sistema en particular éste va seguido de un número que indica el porcentaje de porosidad de dicho sistema, así, el sistema Cu desaleado con 70% de porosidad será: dCu70.
- Escribimos Cu10, Cu20, Cu30...para referirnos a los sistemas con 10, 20, 30...sin hacer distinción de la técnica empleada.

En total se tienen 54 celdas diferentes. Se hace dinámica molecular con cada una a temperatura de 300 K y 500 pasos de simulación. Ver **apéndice A.1** para más detalles de los parámetros de simulación. Teniendo la estructura optimizada se analiza su topología, mediante las funciones de distribución de pares y distribuciones de ángulos planos, y se calcula la densidad de estados electrónicos. Teníamos planeado utilizar el desplazamiento cuadrático medio para caracterizar nuestras simulaciones pero hemos encontrado algunas *particularidades* en su empleo para el tipo de sistemas que utilizamos. Estas *particularidades* las analizamos en el **apéndice A.12**.

Una forma, utilizada por nuestro equipo de trabajo para obtener materiales amorfos, en particular de semiconductores, es generando un precursor cuya red no corresponde a la red más estable para las condiciones de temperatura y presión con las que se realiza la

⁴¹ Hemos realizado otras simulaciones, de las que se hablará en su momento, pero que no constituyen la parte principal de este trabajo.

Tabla 3.1 Dimensiones y número de átomos por supercelda para cada elemento para cada métodos: desaleado y expansión de red.

% Porosidad	Desaleado asimétrico		Expansión de red	
	Átomos	Dimensión	Átomos	Dimensión
10	230	$4 \times 4 \times 4$	256	$4 \times 4 \times 4$
20	256	$4 \times 4 \times 5$	256	$4 \times 4 \times 4$
30	269	$4 \times 4 \times 6$	256	$4 \times 4 \times 4$
40	240	$4 \times 5 \times 5$	256	$4 \times 4 \times 4$
50	250	$5 \times 5 \times 5$	256	$4 \times 4 \times 4$
60	240	$5 \times 5 \times 6$	256	$4 \times 4 \times 4$
70	259	$6 \times 6 \times 6$	256	$4 \times 4 \times 4$
80	274	$7 \times 7 \times 7$	256	$4 \times 4 \times 4$
90	259	$4 \times 9 \times 9$	256	$4 \times 4 \times 4$

dinámica molecular. Por ejemplo, se parte de una estructura tipo FCC diamante o cúbica simple para generar silicio amorfo a presión atmosférica.

En este trabajo realizamos una prueba para comparar esta aproximación y la empleada partiendo de una estructura de *equilibrio*. Generamos dos estructuras tipo diamante de plata una la desaleamos y otra la expandimos al 50% de tal forma que generamos una estructura con densidad igual a 5.245 g/cm^3 y la comparamos con otras dos estructuras desaleada y expandida con la misma densidad cuyo precursor es una estructura FCC. Realizamos 500 pasos de dinámica molecular y optimizamos, los parámetros de simulación son los mismos que se emplean en todas las simulaciones de este trabajo. Las PDF muestran resultados semejantes para ambos sistemas, es decir, no es apreciable una diferencia significativa con un precursor tipo diamante y otro tipo FCC. Decidimos emplear la estructura FCC para el resto del trabajo.

3.7 CARACTERÍSTICAS DE LA SIMULACIÓN

En todas nuestras simulaciones hemos mantenido el volumen de la supercelda constante, es decir, la densidad durante la simulación no varía. Dos de las consecuencias más importantes de esto son: primero, se mantiene el nivel de porosidad constante; segundo, no hay

Tabla 3.2 Características de los sistemas Cu simulados, el prefijo «d» indica la técnica de desaleación y el prefijo «e» la de la expansión de red.

Nombre del Sistema	% porosidad	Densidad / g cm ⁻³	Número de átomos
dCu10	10	8.0278	230
eCu10		8.0598	256
dCu20	20	7.1482	256
eCu20		7.1599	256
dCu30	30	6.2594	269
eCu30		6.2698	256
dCu40	40	5.3612	240
eCu40		5.3699	256
dCu50	50	4.4677	250
eCu50		4.4799	256
dCu60	60	3.5741	240
eCu60		3.5799	256
dCu70	70	2.6785	259
eCu70		2.6799	256
dCu80	80	1.7845	274
eCu80		1.7900	256
dCu90	90	0.8928	259
eCu90		0.8900	256

manera, cuantitativa, de saber si una estructura de baja porosidad colapsa, o bien, una de alta llena todo el espacio. Ver resultados para 10% y 90% de porosidad, **sección 4.3**. En nuestros cálculos de dinámica molecular utilizamos el módulo `Dmol3` del programa de simulación y modelado MATERIALS STUDIO MODELING v3.2.^[65] En los cálculos se utiliza la aproximación local de la densidad, LDA, con la parametrización de Vosko, Wilk y Nusair, VWN. Los electrones de coraza se tratarán mediante un pseudopotencial denominado DSPP (Density functional Semi-core PseudoPotentials).^[56] Se considerarán condiciones periódicas para todos los sistemas. En el **apéndice A.1** se presenta un archivo de entrada para el módulo `Dmol3`. Los parámetros mostrados en ese apéndice corresponden a los utilizados en todas nuestras muestras, a menos que se indique otra cosa. En el **apéndice A.2** se muestran los parámetros utilizados durante la optimización.

Tabla 3.3 Características de los sistemas Ag simulados, el prefijo «d» indica la técnica de desaleación y el prefijo «e» la de la expansión de red.

Nombre del Sistema	% porosidad	Densidad / g cm ⁻³	Número de átomos
dAg10	10	9.4410	230
eAg10		9.4409	256
dAg20	20	8.4069	256
eAg20		8.3918	256
dAg30	30	7.3612	269
eAg30		7.3428	256
dAg40	40	6.3050	240
eAg40		6.2939	256
dAg50	50	5.2541	250
eAg50		5.2449	256
dAg60	60	4.2033	240
eAg60		4.1959	256
dAg70	70	3.1500	259
eAg70		3.1469	256
dAg80	80	2.0986	274
eAg80		2.0979	256
dAg90	90	1.0501	259
eAg90		1.0490	256

Tabla 3.4 Características de los sistemas Au simulados, el prefijo «d» indica la técnica de desaleación y el prefijo «e» la de la expansión de red.

Nombre del Sistema	% porosidad	Densidad / g cm ⁻³	Número de átomos
dAu10	10	----	230
eAu10		----	256
dAu20	20	15.4307	256
eAu20		15.4393	256
dAu30	30	13.5119	269
eAu30		13.4995	256
dAu40	40	11.5730	240
eAu40		11.4997	256
dAu50	50	9.6442	250
eAu50		9.5997	256
dAu60	60	7.7153	240
eAu60		7.6998	256
dAu70	70	5.7820	259
eAu70		5.6998	256
dAu80	80	3.8521	274
eAu80		3.7999	256
dAu90	90	1.9274	259
eAu90		1.9000	256

4 Resultados

4.1 FUNCIONES DE DISTRIBUCIÓN DE PARES PARA LOS SISTEMAS ESTUDIADOS

A continuación se presentan las curvas de distribución de pares para dCu, eCu, dAg, eAg, dAu y eAu. Éstas las presentamos en tres grupos, **figuras 4.1**, **4.2** y **4.3**, cada uno de estos grupos está dividido a su vez en grupos de porcentaje de porosidad, tanto para el sistema desaleado como para el expandido. Estas curvas fueron calculadas con un programa que desarrollamos para este trabajo que hemos llamado pdf, escrito en lenguaje C.⁴² El código de este programa se presenta en el **apéndice A.8**. El intervalo del histograma y el ancho del muestreo en la PDF fue de 0.05 Å.

La primera característica general que hay que notar es que a bajas porosidades, 10% y 20%, puede observarse cierta estructura cristalina; sobre todo para los sistemas desaleados; existe también un desplazamiento de los picos para los sistemas expandidos con respecto a los sistemas desaleados, pero de ella se hablará más adelante. Esta «conservación» de la red cristalina para sistemas desaleados puede explicarse si se considera que al desalearse se mantienen grandes regiones con estructura perfectamente cristalina. Esto es, si imaginamos un sistema cristalino con algunas vacancias, este sistema se mantendrá cristalino pero existirán ciertas distorsiones debido a las vacancias, al aumentar el número de vacancias la distorsión aumenta, pero las vacancias pueden agruparse y formar un hueco en la red. Esto es lo que observamos en las PDF para los sistemas desaleados con 10% y 20% de porosidad; y puede corroborarse al observar su estructura topológica en las **figuras 4.13**, **4.14**, **4.22**, **4.23** y **4.31**. Por supuesto, debe existir un límite en el que la energía superficial de este hueco deforme tanto la red, que el *backbone* ya no pueda considerarse como cristalino. Esto último «puede» ser cierto para sistemas tan pequeños como los nuestros, porque, para sistemas mayores, lo que podría esperarse son sólo distorsiones de la red en la región próxima a un hueco. En la **referencia 35** se realiza un estudio experimental, empleando una técnica de rayos-X, con el que se determina la

⁴² El lenguaje Python a pesar de ser un lenguaje muy versátil, es un lenguaje *interpretado*, lo que lo pone en desventaja, en cuanto a velocidad, cuando se le compara con otros lenguajes, *compilados*, como C o Fortran.

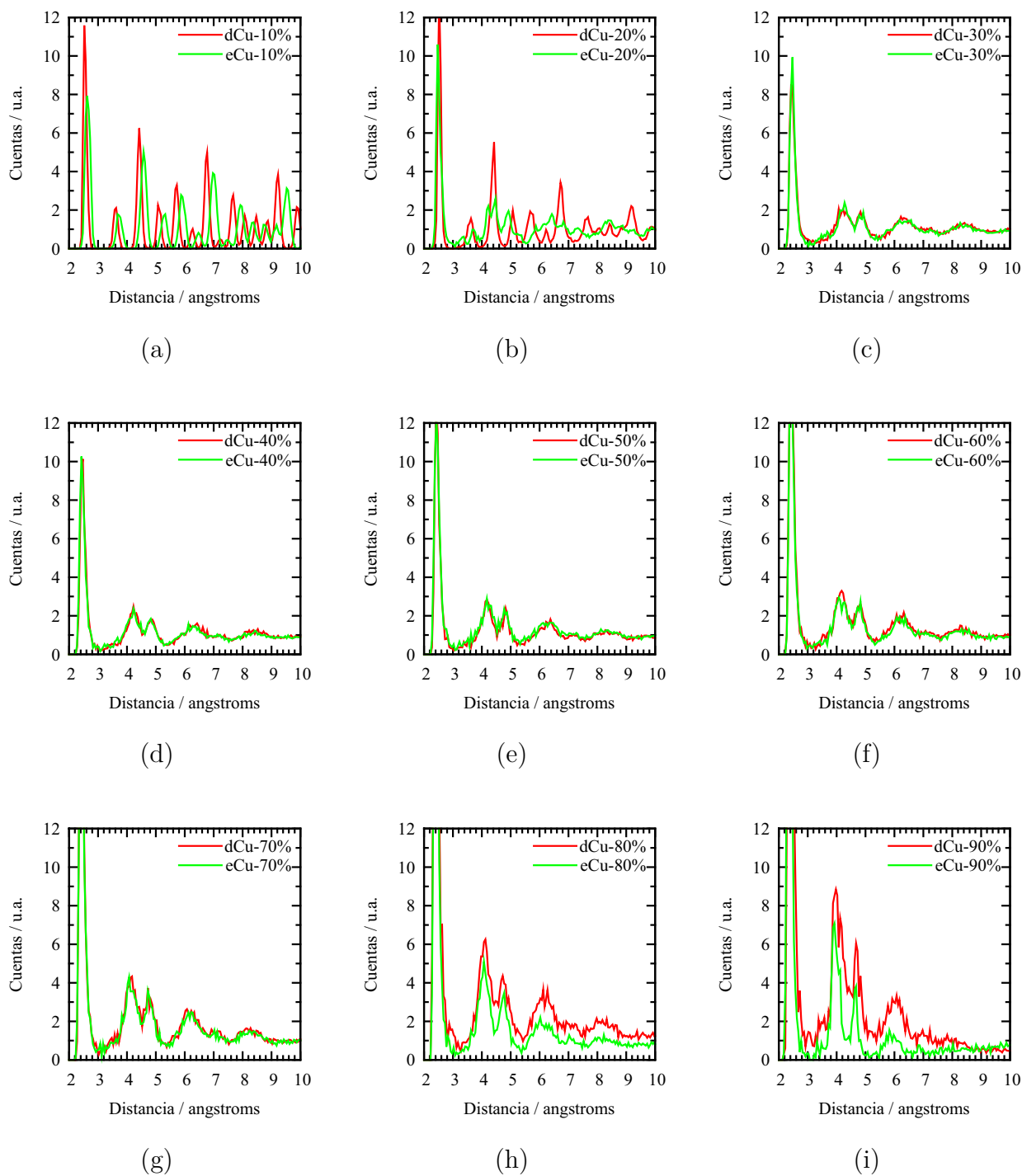


Figura 4.1 Función de distribución de pares, PDF, para el sistema Cu (desaleado y expandido). (a) 10% porosidad (b) 20% porosidad (c) 30% porosidad (d) 40% porosidad (e) 50% porosidad (f) 60% porosidad (g) 70% porosidad (h) 80% porosidad (i) 90% porosidad.

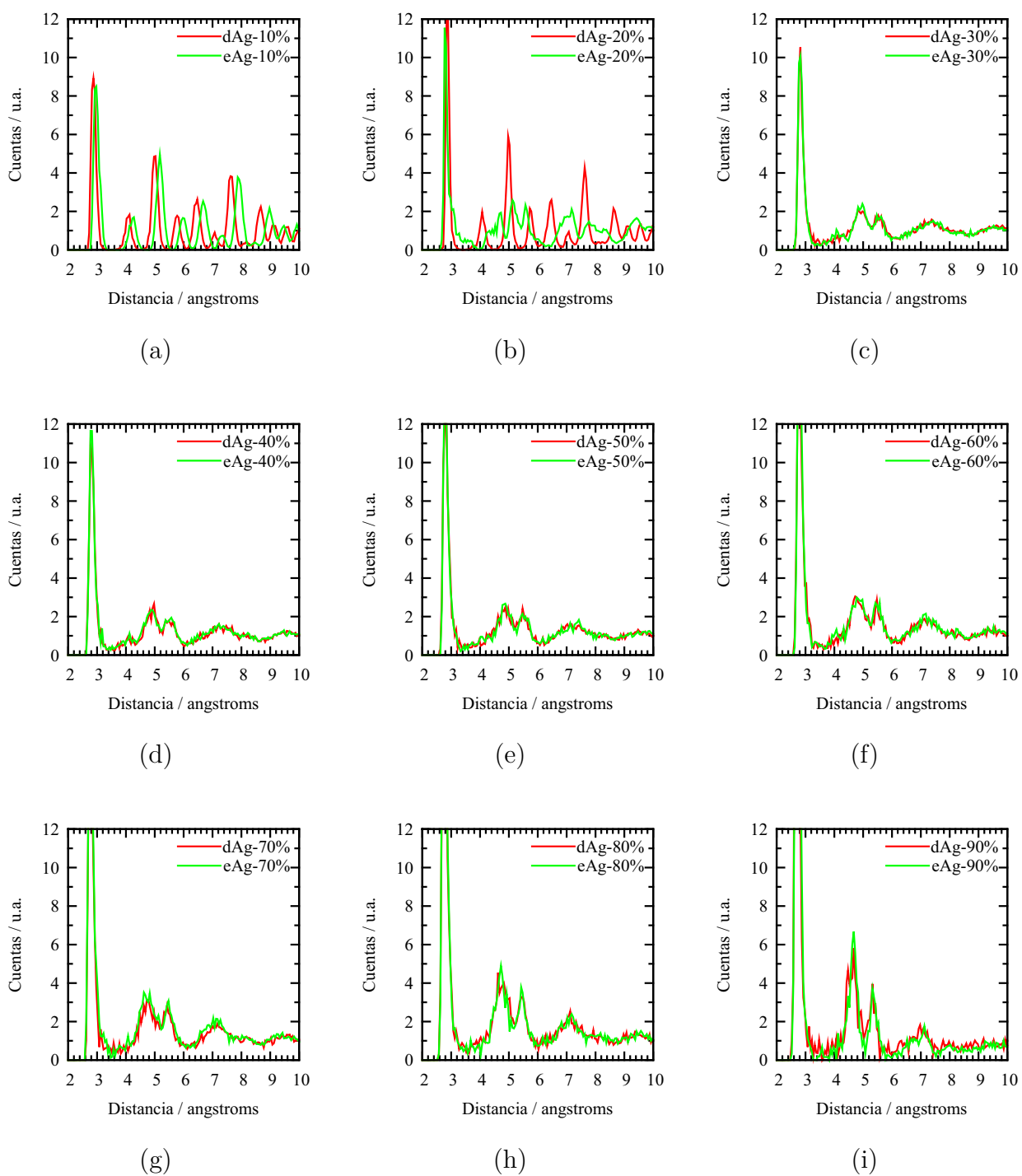


Figura 4.2 Función de distribución de pares, PDF, para el sistema Ag (desaleado y expandido). (a) 10% porosidad (b) 20% porosidad (c) 30% porosidad (d) 40% porosidad (e) 50% porosidad (f) 60% porosidad (g) 70% porosidad (h) 80% porosidad (i) 90% porosidad.

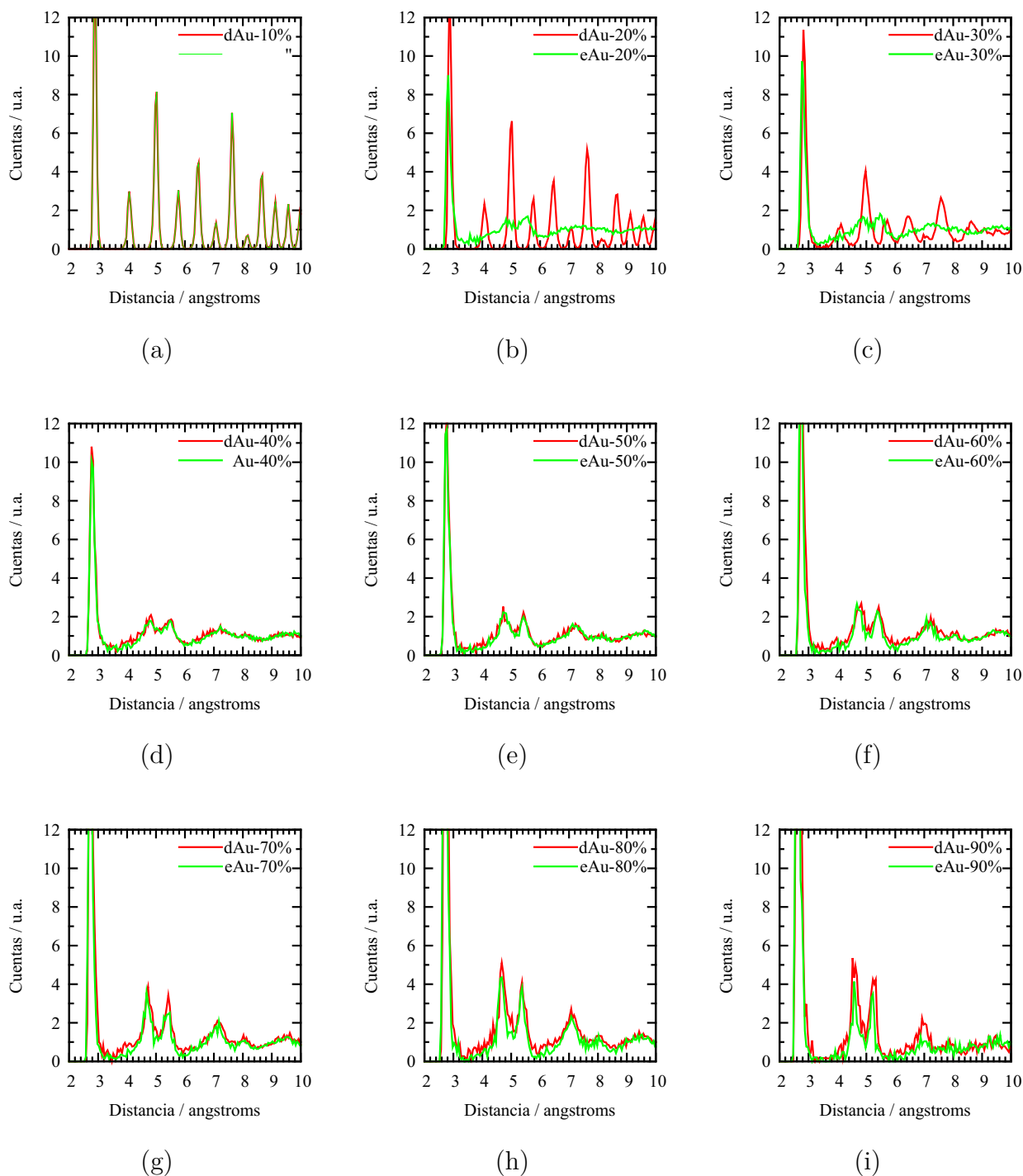


Figura 4.3 Función de distribución de pares, PDF, para el sistema Au (desaleado y expandido). (a) 10% porosidad (b) 20% porosidad (c) 30% porosidad (d) 40% porosidad (e) 50% porosidad (f) 60% porosidad (g) 70% porosidad (h) 80% porosidad (i) 90% porosidad.

cristalinidad de una aleación Au–Ag en función del tiempo de desaleado, ellos aseguran que la estructura cristalina de la aleación se mantiene prácticamente invariable pero hay distorsiones en la red debido a la aparición de nanoporos de Au.

El caso de los sistemas generados con expansión de red, para estas mismas porosidades, 10% y 20%, es diferente. En la ER no partimos de una celda con vacancias o ausencia de átomos,⁴³ sino que aumentamos la distancia interatómica y conservamos el orden de la red. Si aplicásemos la técnica de expansión de red de forma continua, se tendría una estructura cristalina que se va tensando a medida que se expande la red, después de cierto límite los átomos ya no podrían estar más tensados (su interacción será muy débil) y podrán considerarse como desenlazados; el desplazamiento de cada pico observado en las **figuras 4.1(a), 4.2(a) y 4.3(a)** puede explicarse de esta manera, en esta estructura cristalina la interacción atómica es lo suficientemente fuerte como para considerar a sus átomos como enlazados (y se comportan como tal), pero la distancia no es la distancia normalmente observada, en las **figuras 4.40 y 4.49** puede verse la estructura cristalina *tensada* como un simple cristal. En nuestras simulaciones no permitimos que el volumen de la supercelda varíe, utilizamos un ensamble canónico, lo que impide que la estructura se contraiga, y las distancias de enlace se reduzcan. La tolerancia de enlace se pierde (átomos desenlazados) cuando la expansión de red genera una porosidad mayor al 10%, por lo que durante la dinámica molecular la estructura no se comporta como un cristal tensado y genera una PDF cuya forma no es propia de un cristal, ver **figuras 4.1(b), 4.2(b) y 4.3(b)**, y **4.41, 4.50 y 4.58**.

En las **tablas 4.1, 4.2 y 4.3** se presentan los valores numéricos de los puntos más representativos de las curvas PDF presentadas arriba. En la sexta columna de estas tablas se presenta el valor de inicio de todas las PDF. Los valores reportados y comúnmente aceptados para el radio metálico de estos tres metales son: $r_{\text{Cu}}=1.28 \text{ \AA}$, $r_{\text{Ag}}=1.44 \text{ \AA}$ y $r_{\text{Au}}=1.44 \text{ \AA}$, de aquí que la distancia a primeros vecinos será, $2r$, 2.56, 2.88 y 2.88 \AA para el Cu, la Ag y el Au respectivamente. Puede observarse de estas tablas que este valor corresponde aproximadamente al valor esperado,⁴⁴ pero el valor de inicio de PDF muestra un valor menor para todos los sistemas indicando que existen átomos, dentro de todos ellos, que se encuentran a una distancia a primeros vecinos menor de lo que cabría

⁴³ Y el *backbone* cristalino.

⁴⁴ Con algunas particularidades de las que se hablará más adelante.

Tabla 4.1 Puntos críticos más importantes de las PDF para los sistemas Cu. Los valores de la bimodalidad se encuentran en las columnas con encabezado «Máx. 2do. pico» y «Máx. 3er. pico»; los valores entre paréntesis son los esperados según la **figura 4.6**.

Nombre del Sistema	Máx. 1er pico	Mín. cero	Máx. 2do. pico	Máx. 3er. pico	Inicio
dCu10	2.525	2.78	3.625	4.425	2.325
eCu10	2.625	2.925	3.675	4.575	2.375
dCu20	2.525	2.925	3.625	4.452	2.275
eCu20	2.475	3.075	3.675	4.475	2.225
dCu30	2.475	3.275	4.275(4.287)	4.825(4.950)	2.225
eCu30	2.475	3.025	4.225(4.287)	4.925(4.950)	2.225
dCu40	2.475	3.075	4.225(4.287)	4.825(4.950)	2.225
eCu40	2.425	3.075	4.225(4.200)	4.825(4.850)	2.225
dCu50	2.425	3.025	4.175(4.200)	4.825(4.850)	2.225
eCu50	2.425	3.075	4.175(4.200)	4.775(4.850)	2.225
dCu60	2.425	3.275	4.175(4.200)	4.825(4.850)	2.175
eCu60	2.425	2.975	4.075(4.200)	4.825(4.850)	2.225
dCu70	2.425	3.025	4.175(4.200)	4.725(4.850)	2.225
eCu70	2.425	3.125	4.075(4.200)	4.725(4.850)	2.175
dCu80	2.375	3.285	4.125(4.114)	4.725(4.750)	2.225
eCu80	2.375	3.075	4.075(4.114)	4.775(4.750)	2.225
dCu90	2.375	2.925	3.975	4.675	2.125
eCu90	2.375	3.075	3.925	4.675	2.125

esperar. Hemos determinado que aquellos átomos con menor distancia a primeros vecinos se encuentra en las proximidades de un poro; si este «acercamiento» interatómico está relacionado con la superficie es de esperar que los picos a primeros vecinos se desplacen a la izquierda conforme aumenta la superficie (aumenta la porosidad⁴⁵). Esta tendencia puede observarse claramente en estas tres tablas; en las que el lugar del máximo del primer pico disminuye al aumentar la porosidad. También el desplazamiento de valor de inicio de la PDF hacia la izquierda se corresponde con el aumento de la superficie y la disminución del tamaño de *backbone*, ver las estructuras topológicas de los sistemas Cu70-90, Ag70-90 y Au70-90.

⁴⁵ Entre más «tortuosa» o dendrítica sea la forma de los poros formados más superficie se creará.

Tabla 4.2 Puntos críticos más importantes de las PDF para los sistemas Ag. Los valores de la bimodalidad se encuentran en las columnas con encabezado «Máx. 2do. pico» y «Máx. 3er. pico»; los valores entre paréntesis son los esperados según la **figura 4.6**.

Nombre del Sistema	Máx. 1er pico	Mín. cero	Máx. 2do. pico	Máx. 3er. pico	Inicio
dAg10	2.875	3.325	4.125	5.025	2.675
eAg10	2.975	3.375	4.275	5.175	2.725
dAg20	2.875	3.375	4.075	4.975	2.625
eAg20	2.775	3.925	4.725	5.125	2.625
dAg30	2.825	3.325	4.825(4.893)	5.525(5.650)	2.575
eAg30	2.825	3.325	4.975(4.893)	5.475(5.650)	2.575
dAg40	2.775	3.575	4.975(4.806)	5.575(5.550)	2.575
eAg40	2.775	3.475	4.875(4.806)	5.375(5.550)	2.575
dAg50	2.825	3.375	4.875(4.893)	5.475(5.650)	2.575
eAg50	2.775	3.325	4.875(4.806)	5.475(5.550)	2.575
dAg60	2.825	3.275	4.875(4.893)	5.475(4.650)	2.575
eAg60	2.775	3.325	4.875(4.806)	5.475(5.550)	2.575
dAg70	2.775	3.525	4.752(4.806)	5.425(5.550)	2.575
eAg70	2.775	3.475	4.625(4.806)	5.475(5.550)	2.575
dAg80	2.775	3.625	4.625(4.806)	5.475(5.550)	2.525
eAg80	2.775	3.575	4.725(4.806)	5.425(5.550)	2.525
dAg90	2.675	3.575	4.675	5.325	2.475
eAg90	2.675	3.275	4.675	5.325	2.475

4.1.1 Ausencia de un mínimo cero en metales⁴⁶

Hay dos características en las funciones de distribución de pares, PDF, que nosotros presentamos y que están presentes en muchos sistemas metálicos (ver por ejemplo la referencia [57]). La primera es la bimodalidad del pico de segundos vecinos; la segunda es la ausencia de un mínimo cero entre el primer (primeros vecinos) y los segundos picos (segundos vecinos). De la bimodalidad hablaremos en la siguiente sección. En esta sección hacemos un pequeño razonamiento que nos permite justificar la segunda.

⁴⁶ Con «mínimo cero» nos referimos al mínimo entre los picos de primeros y segundos vecinos en una PDF cuya altura sea cero.

Tabla 4.3 Puntos críticos más importantes de las PDF para los sistemas Au. Los valores de la bimodalidad se encuentran en las columnas con encabezado «Máx. 2do. pico» y «Máx. 3er. pico»; los valores entre paréntesis son los esperados según la **figura 4.6**.

Nombre del Sistema	Máx. 1er pico	Mín. cero	Máx. 2do. pico	Máx. 3do. pico	Inicio
dAu10	2.875	3.225	4.075	5.025	2.675
eAu10					
dAu20	2.875	3.225	4.075	5.025	2.675
eAu20	2.825	3.525	4.825	5.575	2.625
dAu30	2.825	3.425	4.125	4.975	2.575
eAu30	2.775	3.325	4.875(4.806)	5.475(5.550)	2.575
dAu40	2.775	3.625	4.825(4.806)	5.525(5.550)	2.575
eAu40	2.775	3.675	4.825(4.806)	5.475(5.550)	2.575
dAu50	2.775	3.325	4.725(4.806)	5.425(5.550)	2.575
eAu50	2.775	3.375	4.725(4.806)	5.425(5.550)	2.525
dAu60	2.725	3.425	4.825(4.720)	5.425(5.450)	2.575
eAu60	2.725	3.225	4.675(4.720)	5.375(5.450)	2.575
dAu70	2.725	3.475	4.725(4.720)	5.425(5.450)	2.525
eAu70	2.725	3.625	4.675(4.720)	5.475(5.450)	2.525
dAu80	2.725	3.625	4.675(4.720)	5.375(5.450)	2.525
eAu80	2.725	3.375	4.675(4.720)	5.375(5.450)	2.525
dAu90	2.475	3.225	4.525	5.225	2.425
eAu90	2.425	3.125	4.575	5.225	2.475

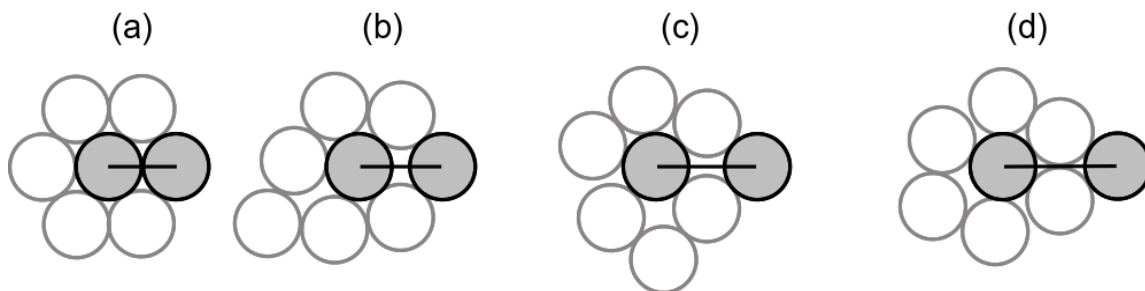


Figura 4.4 Distancias de transición de primeros a segundos vecinos para un sistema sin enlaces direccionales. Forma esquemática.

Si consideramos un átomo metálico como una esfera, que en el mejor de los casos puede

ser rígida,⁴⁷ y si consideramos, además, que podemos formar sistemas compactos; podemos colocar alrededor de esta primera esfera, y en cualquier dirección, una, dos, tres, ... y hasta doce esferas. Siendo posible debido a que en los metales no existen enlaces direccionales. Esto trae como consecuencia, ver **figura 4.4**, que no exista una transición clara entre los átomos que constituyen los primeros vecinos y aquellos que constituyen los segundos. En esta figura se presenta de forma esquemática un átomo (esfera) rodeado arbitrariamente por otros átomos, esta configuración permite el posicionamiento, en dirección, casi arbitraria de un vecino. En la **figura 4.5** se encuentra una PDF esquemática donde se muestran las posiciones correspondientes de la figura anterior. Estas dos figuras no consideran la contracción en la distancia de primeros vecinos de la que se habló arriba y que corresponde a la primera mitad del pico de primeros vecinos.

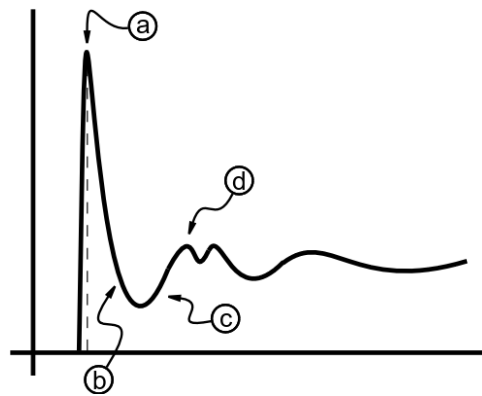


Figura 4.5 PDF esquemática, la ausencia de un mínimo cero. Las letras circuladas corresponden a los estados mostrados en la **figura 4.4**.

4.1.2 Bimodalidad en el pico de segundos vecinos

En todas las funciones de distribución de pares, PDF, que nosotros presentamos existe una bimodalidad en el pico de segundos vecinos. Esta bimodalidad es característica de la mayoría de los metales amorfos. Nosotros hemos considerado algunas situaciones geométricas dentro de las estructuras metálicas FCC que permiten identificar esta bimodalidad.

⁴⁷ Las variaciones en la distancia a primeros vecinos en cada sistema que nosotros hemos estudiado no permite considerar esferas totalmente rígidas. Ver **tablas 4.1, 4.2 y 4.3**

Al considerar el plano más denso de un sistema compacto puede observarse que existen al menos dos distancias a segundos vecinos muy próximas, ver **figura 4.6**, la primera de ellas (la más larga) corresponde a un átomo localizado directamente atrás de un primer vecino, parte (c), la segunda (la más cercana) está en el intersticio dejado por dos átomos de primeros vecinos, parte (b). La distancia más larga para este sistema es igual⁴⁸ a $4r$ y la más corta igual a $4r \sin(60^\circ)$, es decir, hay una diferencia del 13 %.

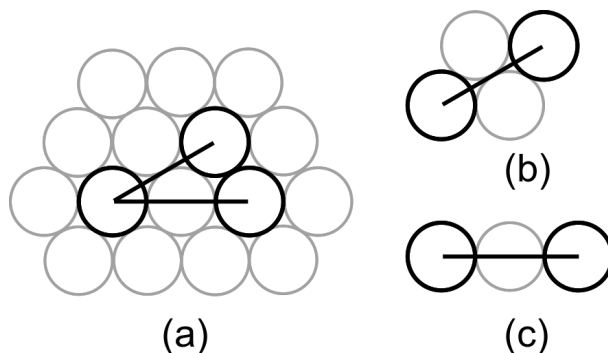


Figura 4.6 Esquema que muestra la posible distribución geométrica correspondiente a la bimodalidad en el pico de segundos vecinos.

Estos son los valores que habrían de esperarse en un arreglo perfecto, en un arreglo imperfecto, como el nuestro, éstos dos posiciones deben variar debido a que no se tiene un ordenamiento totalmente compacto. En las **tablas 4.1**, **4.2** y **4.3** se presentan los valores de los picos a segundos vecinos obtenidos en la simulación y los calculados (encerrados en paréntesis). En promedio ha una diferencia de hasta el 2% en estos valores para todos los sistemas. Lo que nos permite asegurar que efectivamente estas posiciones corresponden a átomos cercanos a las posiciones que hemos descrito. Esta misma descripción la hace S. R. Elliott^[68] al hablar de empaquetamientos aleatorios densos de esferas.⁴⁹ Existen otras interpretaciones más complejas para explicar la bimodalidad del pico a segundos vecinos para metales,^[69, 70] estas consisten en describir estructuras poliédricas, principalmente icosaedrales, que conforman una capa de coordinación para un átomo dado. Ver **figura 4.7**. Estos estudios se han realizado para aleaciones, aunque también se han buscado este tipo de estructuras icosaedrales para elementos puros.^[71]

⁴⁸ donde $2r$ es la distancia a primeros vecinos.

⁴⁹ Dense random packing of spheres.

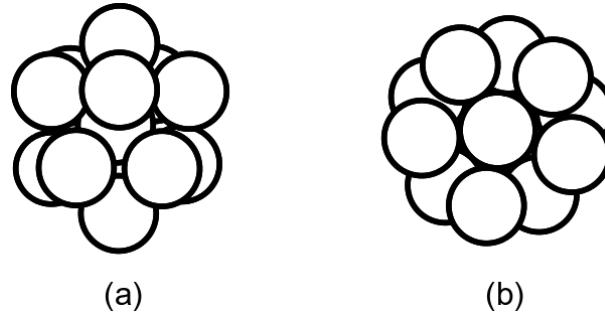


Figura 4.7 Capa de coordinación icosaedral con la que se ha explicado la bimodalidad en aleaciones. (a) Vista frontal. (b) Vista superior.

La existencia de estructuras icosaedrales obliga a que el número de coordinación sea al menos doce, pero esto también se espera de un sistema compacto, **tabla 4.4**. Por otro lado, a distribución de ángulos planos indica que posiblemente puedan existir estas estructuras.

Existen al menos cuatro métodos para estimar el número de coordinación, ver **sección 2.5**. El que nosotros empleamos es un método que genera valores sobreestimados; utilizamos aquel en el que el límite de integración es el valor del mínimo entre los picos de primeros y segundos vecinos, hemos utilizado este método pues consideramos que al no haber una transición clara entre la distancia de primeros y segundos vecinos, lo más sensato será elegir una distancia intermedia entre estos y aquellos. En la **tabla 4.4** se puede apreciar la sobrecoordinación para los sistemas eCu10 y eAg10, pues ya hemos establecido que estos dos sistemas corresponden a estructuras FCC *perfectas*, y el número de coordinación debe ser 12 exactamente; en la tabla se calculan valores de 12.23 y 12.203, respectivamente. La disminución del número de coordinación se corresponde con el aumento de porosidad, pero de esto se hablará más adelante en la **sección 4.3**.

4.1.3 Caída de la PDF debida a la porosidad

Algunas de las PDF calculadas muestran una característica que hemos asociado con la porosidad del material. Esta característica consiste en una caída por debajo de uno a distancias intermedias. Es algo que no se observa en las estructuras no porosas; aunque, tampoco se observa en todas las estructuras porosas. En esta sección haremos un análisis para explicar el porqué de esta caída.

Tabla 4.4 Números de coordinación obtenidos para los sistemas simulados.

Nombre del Sistema	Número de coordinación	Sistema	Número de coordinación	Nombre del Sistema	Número de coordinación
dCu10	11.041	dAg10	10.927	dAu10	10.971
eCu10	12.230	eAg10	12.203	eAu10	---
dCu20	10.136	dAg20	10.104	dAu20	9.698
eCu20	11.216	eAg20	10.391	eAu20	9.793
dCu30	11.007	dAg30	10.365	dAu30	9.076
eCu30	10.576	eAg30	10.439	eAu30	8.794
dCu40	9.934	dAg40	10.261	dAu40	8.986
eCu40	9.980	eAg40	10.237	eAu40	8.393
dCu50	9.444	dAg50	9.790	dAu50	8.013
eCu50	10.111	eAg50	9.633	eAu50	7.552
dCu60	9.656	dAg60	9.317	dAu60	7.847
eCu60	9.331	eAg60	9.33	eAu60	6.801
dCu70	9.704	dAg70	8.127	dAu70	7.256
eCu70	9.165	eAg70	8.930	eAu70	6.237
dCu80	9.693	dAg80	7.716	dAu80	6.400
eCu80	7.834	eAg80	7.523	eAu80	5.713
dCu90	7.221	dAg90	4.978	dAu90	4.123
eCu90	7.834	eAg90	4.909	eAu90	3.288

Suponiendo que nos ubicamos en una posición atómica i dentro de una estructura y calculamos la relación entre el volumen de un casquete de radio r y espesor dr , $V(r)$, —centrado en esa posición— y el número de átomos dentro de ese casquete, N_r ; y esta relación la definimos como $R_i(r) = N_r/V(r)$, para cada valor de r . Si luego repetimos este proceso para cada una de las posiciones atómicas de la estructura, el promedio de todas estas $R_i(r)$ será igual a la PDF de la estructura:

$$\text{PDF} = \sum_i R_i(r) \quad (4.1)$$

Es decir, la $R_i(r)$ es una PDF para el átomo i . Entonces, ya que el cálculo de la PDF es estadístico, es lógico pensar que, en promedio, hay una disminución en el número de átomos dentro del intervalo que abarca la caída por debajo de 1, Δr_{debajo} . O, en otras palabras, la mayoría de las $R_i(r)$ registran la ausencia de átomos (porosidad) alrededor

de Δr_{debajo} . Esto significa que Δr_{debajo} podría dar un tamaño promedio o estimado de la porosidad. Para comprobar esto utilizamos una estructura amorfa artificial⁵⁰ de 1000 átomos de cobre,⁵¹ la arista de esta supercelda (estructura) mide 23.5973 Å. Ahora, eliminamos todos los átomos que se encuentren hasta una distancia r desde el centro de la celda, las distancias que utilizamos son 4, 6, 8, 10, 12 y 14 Å, es decir, generamos un hueco esférico de radio r ; y calculamos la PDF de estas estructuras, **figura 3.11**.

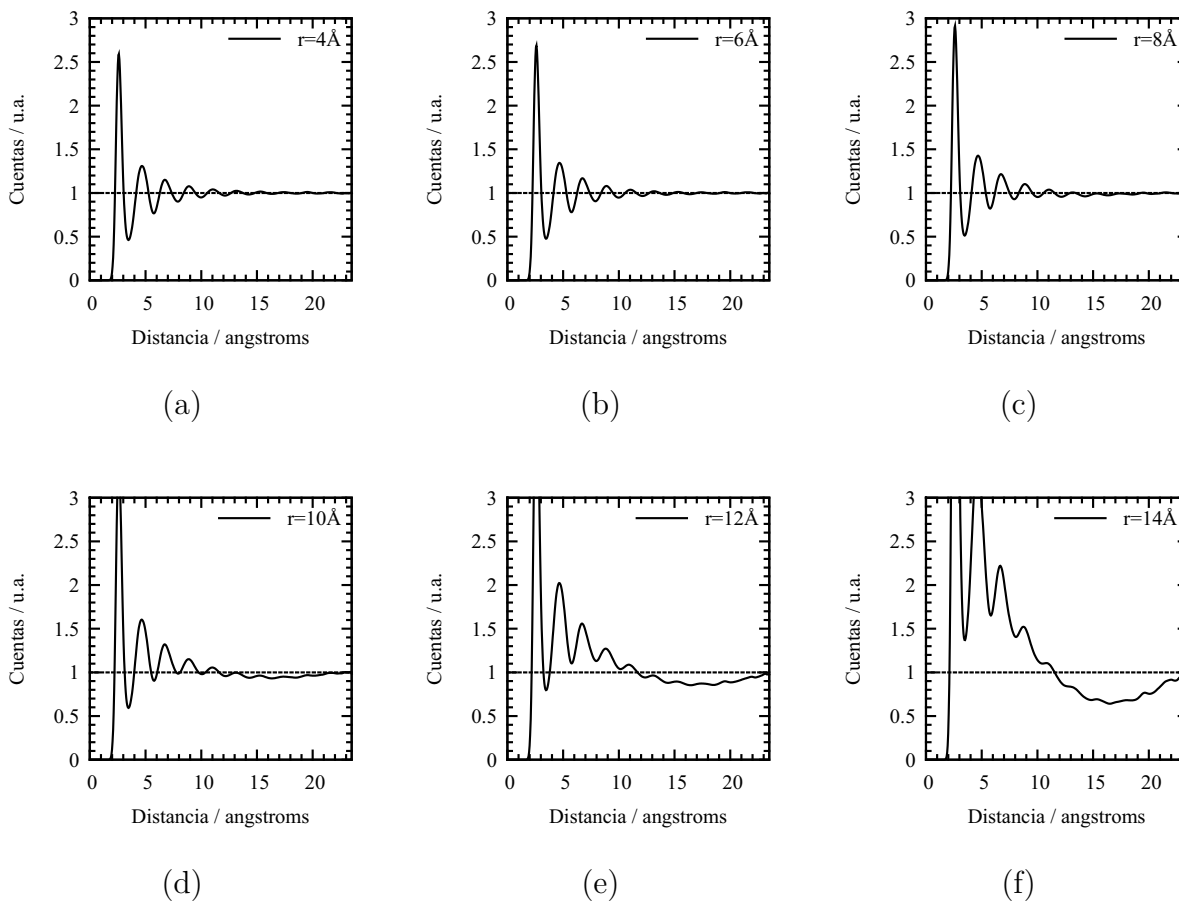


Figura 4.8 PDF de superceldas de Cu de arista de 23.5973 Å y con un hueco esférico en su interior de radio (a) 4 Å, (b) 6 Å, (c) 8 Å, (d) 10 Å, (e) 12 Å y (f) 14 Å.

⁵⁰ Una estructura no obtenida por dinámica molecular, en la que los átomos son colocados «aleatoriamente» respetando ciertos atributos como distancia de enlace, ángulo, etcétera; para evitar traslapes o estructuras altamente inestables.

⁵¹ En este caso, el metal que se emplea no es importante, sólo tiene relevancia para determinar distancia de vecinos y dimensiones de la estructura.

Se observa que la caída es difícil de ver cuando la porosidad es baja así, también es difícil determinar los límites del valle que está por debajo de 1, Δr_{debajo} . Nosotros sólo hemos podido estimarlo para los radios 8, 10, 12 y 14 Å, obteniendo valores de Δr_{debajo} de 8, 9, 11 y 11.5 Å respectivamente. Lo que nos sugiere que esta caída, efectivamente, está relacionada con el tamaño de la porosidad y podría servir como medida del tamaño de poro. Sin embargo, presenta algunos inconvenientes: el primero es la dificultad para determinar los límites del valle. Debido al carácter estadístico de la PDF no puede determinarse la forma de la porosidad. Si la porosidad es pequeña comparada con el tamaño del backbone, no es observable la caída. De esta última afirmación puede sacarse una conclusión relevante, si la oquedad está dispersa,^{52,53} en toda la estructura como en la porosidad dendrítica, la caída no será observable; pero si la oquedad está concentrada en una región de tal forma que el tamaño de poro sea mayor que el backbone,⁵⁴ la caída será observable y el tamaño del poro medible. A continuación presentamos el detalle de 4 PDFs de las muestras simuladas, eCu40, dCu40, eAg40 y dAg40, en las que se aprecia la caída en la PDF.

Como bien se observa, la Δr_{debajo} es más clara en las muestras de Ag que en las de cobre, al observar su estructura, **sección 4.3**, se ve que en la Ag la porosidad se encuentra menos dispersa que en el Cu. Esto puede confirmarse en la **figura 4.76d** de la sección de fractales; la dimensión fractal de las muestras de Ag40 está por debajo de 2.8 mientras que para el Cu40 está por arriba. Ahora bien, eAg40 es menos dispersa y es más fácil observar la caída en que dAg40 que es más dispersa pues $d_{\text{eAg40}} < d_{\text{dAg40}}$.⁵⁵ Nosotros hemos escrito un artículo^[72] en el que presentamos mayor información y de una forma más generalizada, considerando otros sistemas; aunque no en él no incorporamos la dimensión fractal.

4.2 ÁNGULOS

En esta sección se presentan los ángulos planos calculados para todas las estructuras, el cálculo de estos ángulos se realiza mediante un programa escrito en lenguaje Python y presentado en el **apéndice A.9**, llamado **angulos**. El programa consiste en tomar triadas

⁵² Tamaño de poro comparable con el ancho del backbone.

⁵³ Dimensión fractal alta, ver **sección 4.7**.

⁵⁴ Dimensión fractal baja, ver **sección 4.7**.

⁵⁵ Dimensión fractal, **sección 4.7**.

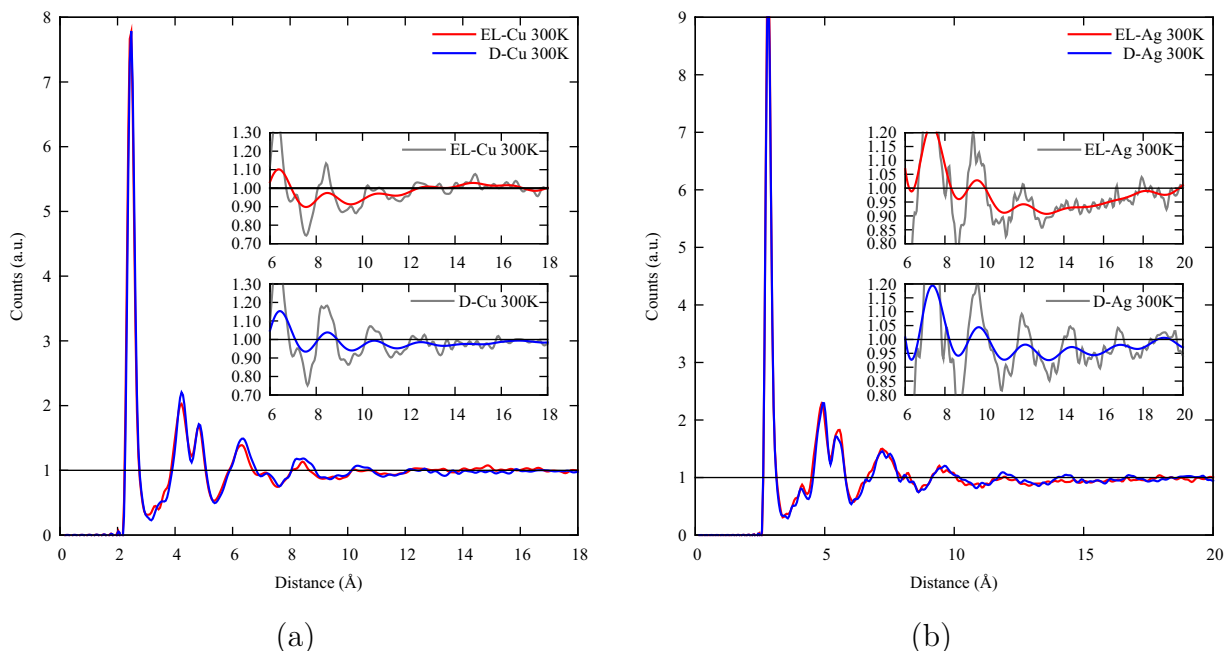


Figura 4.9 Curvas PDF para los sistemas Cu40 y Ag40 a 300 K. (a) Comparación entre dCu40 y eCu40. (b) Comparación entre dAg40 y eAg40. En los detalles de cada figura aparece una línea gruesa continua que sirve como guía y es obtenida eliminando 95% de las frecuencias más altas en un suavizado de Fourier.

de átomos. Cada triada posee un átomo central cuya distancia a los otros dos átomos debe ser menor o igual a una distancia determinada por el usuario. En nuestro estudio la distancia que hemos empleado es la distancia del mínimo entre los picos de primeros y segundos vecinos. Estos valores se reportan en las **tablas 4.1, 4.2 y 4.3**.

El sistema cristalino FCC (dodecaedro) tiene cuatro ángulos principales a primeros vecinos: 60, 90, 120° y estos son los picos que se observan en los sistemas con bajas porosidades. Sobretudo para los sistemas Au. El oro al ser el metal más pesado de entre los estudiados tiende a moverse más lento, a una temperatura dada, que los otros dos metales, y es prudente esperar que retenga su estructura inicial por más tiempo. Es por eso que en las curvas PDF el dAu30 tiene una estructura todavía cristalina, cuando el dCu30 y el dAg30 ya no.

En un icosaedro ideal los ángulos principales son: 60, 63.5, 108, y 116.5°. De ser cierto que existen grupos icosaedrales dentro de una estructura metálica amorfa, estos ángulos deberían aparecer en las distribuciones de ángulos planos. A bajas porosidades y cuando

Tabla 4.5 Valores máximos de las distribuciones de ángulos planos para los sistemas Cu, Ag y Au.

Nombre del Sistema	Ángulos	Sistema	Ángulos	Nombre del Sistema	Ángulos
dCu10	60, 90, 120, 177	dAg10	60, 90, 120, 177	dAu10	59, 90, 120, 178
eCu10	60, 90, 120, 177	eAg10	60, 90, 120, 177	eAu10	---
dCu20	60, 90, 120, 178	dAg20	60, 90, 120, 178	dAu20	60, 89, 120, 178
eCu20	59, 90, 121, 178	eAg20	60, 116, 178	eAu20	60
dCu30	60, 113	dAg30	60, 117	dAu30	60, 91, 120, 177
eCu30	60, 87, 117	eAg30	60, 91, 120	eAu30	60, 116
dCu40	59, 117	dAg40	60, 90, 120	dAu40	60, 118
eCu40	60, 116	eAg40	60, 94, 120	eAu40	60, 118
dCu50	60, 118	dAg50	60, 116	dAu50	60, 117
eCu50	59, 117	eAg50	60, 91, 116	eAu50	60, 117
dCu60	59, 117	dAg60	59, 116	dAu60	60, 120
eCu60	59, 117	eAg60	60, 92, 118	eAu60	60, 120
dCu70	60, 117	dAg70	59, 116	dAu70	60, 118
eCu70	60, 113	eAg70	59, 116	eAu70	60, 118
dCu80	60, 109	dAg80	60, 118	dAu80	60, 118
eCu80	60, 109	eAg80	60, 117	eAu80	60, 118
dCu90	60, 108	dAg90	60, 113	dAu90	60, 120
eCu90	60, 108	eAg90	60, 117	eAu90	60, 120

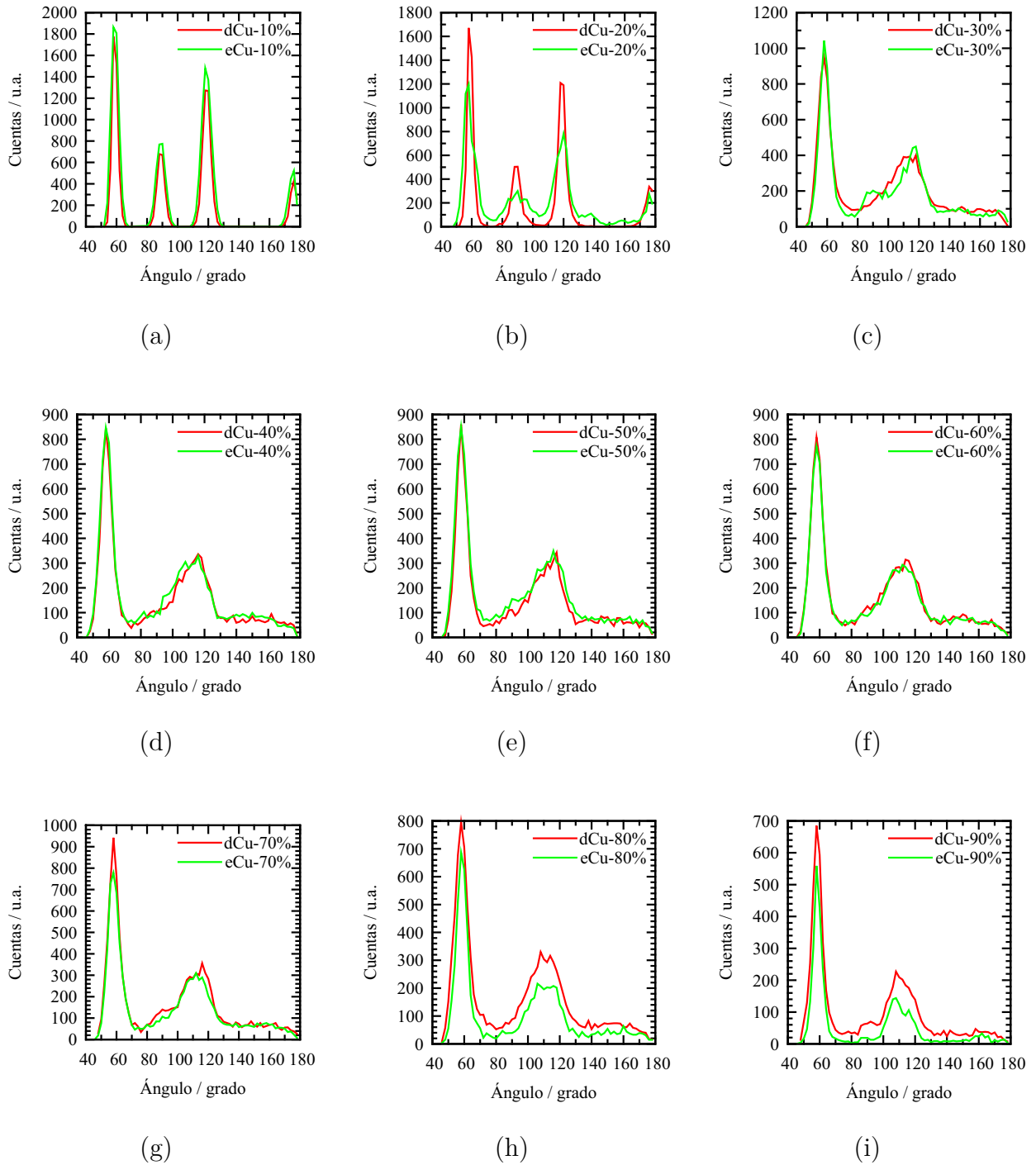


Figura 4.10 Comparación de los valores de ángulos planos para el sistema Cu (desaleado y expandido). (a) 10% porosidad (b) 30% porosidad (c) 30% porosidad (d) 40% porosidad (e) 50% porosidad (f) 60% porosidad (g) 70% porosidad (h) 80% porosidad (i) 90% porosidad.

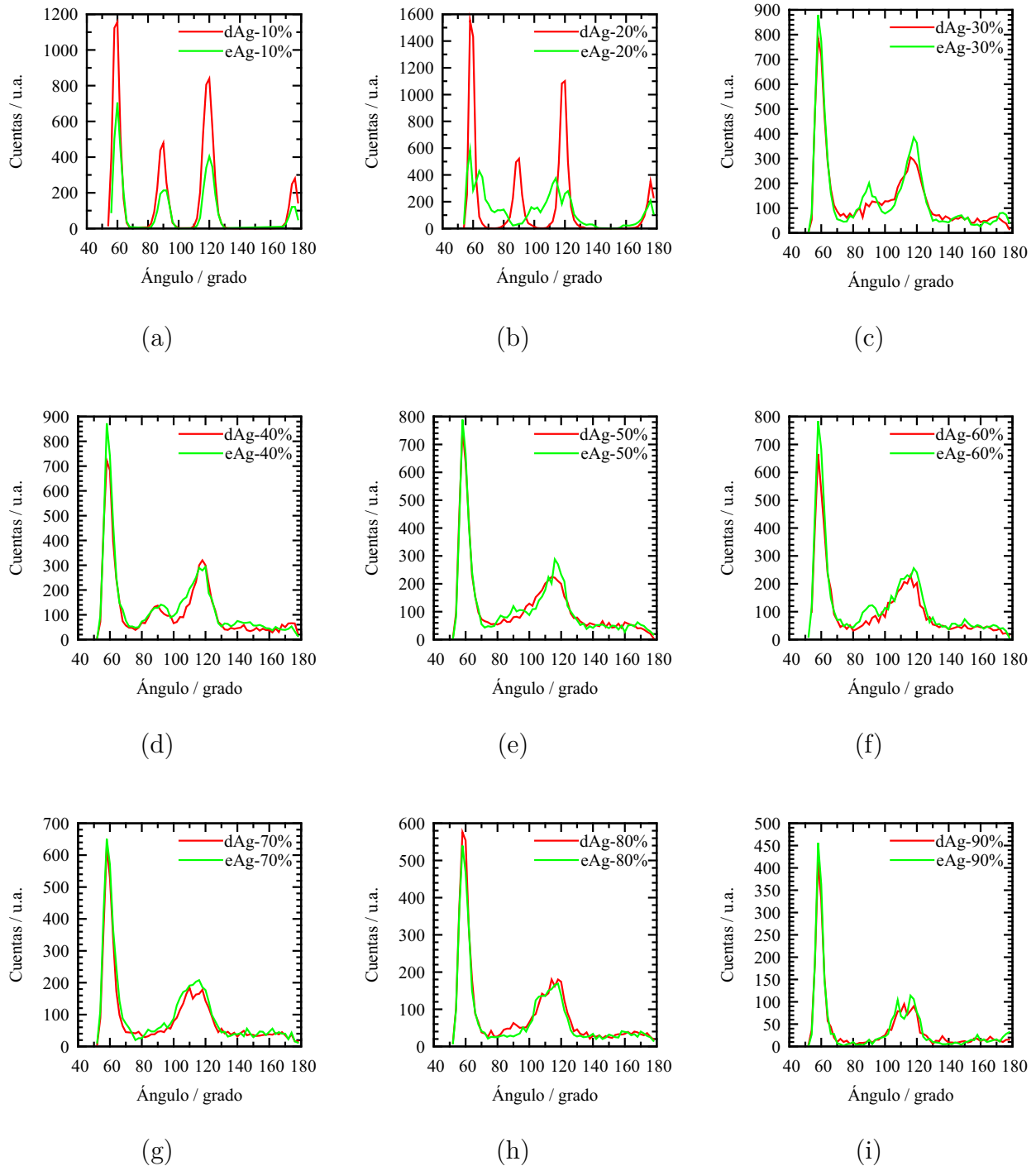


Figura 4.11 Comparación de los valores de ángulos planos para el sistema Ag (desaleado y expandido). (a) 10% porosidad (b) 20% porosidad (c) 30% porosidad (d) 40% porosidad (e) 50% porosidad (f) 60% porosidad (g) 70% porosidad (h) 80% porosidad (i) 90% porosidad.

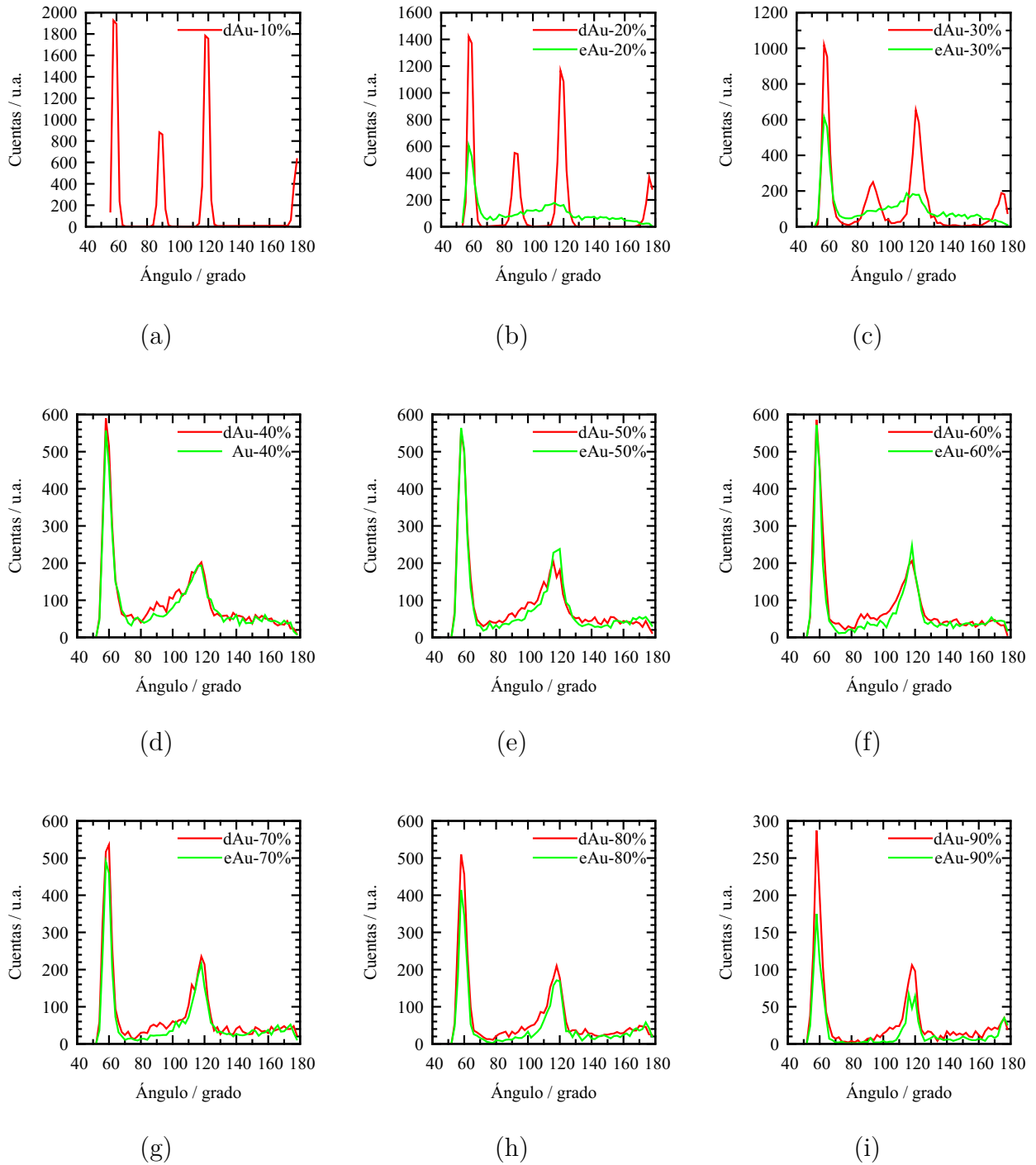


Figura 4.12 Comparación de los valores de ángulos planos para el sistema Au (desaleado y expandido). (a) 10% porosidad (b) 20% porosidad (c) 30% porosidad (d) 40% porosidad (e) 50% porosidad (f) 60% porosidad (g) 70% porosidad (h) 80% porosidad (i) 90% porosidad.

el sistema se mantiene cristalino, según la RDF, los ángulos 60, 90 y 120° prevalecen. En las otras estructuras se mantiene el ángulo $\tilde{60}^\circ$ y aparecen ángulos alrededor de 116.5, y el ensanchamiento de los picos entre 100 y 120° indica la presencia de ángulos con valor de 108°, es decir, creemos que efectivamente debe existir cierta estructura icosaedral. Creemos, así mismo, que la estructura dodecaedral también está presente pues sigue habiendo valores para ángulos de 120°. Sin embargo no podemos hablar de icosaedros ni de dodecaedros perfectos, ni del dominio dodecaedral sobre el icosaedral. En suma, la deformación de la red ocasionada por la porosidad genera un arreglo tal que aparecen ángulos propios de estructuras dodecaedrales e icosaedrales.

4.3 TOPOLOGÍA

En esta sección se presentan las imágenes topológicas de todos los sistemas. El orden de presentación es el siguiente: dCu, dAg, dAu, eCu, eAg y eAu. Cada una de estas estructuras se ha generado de forma independiente, es decir, una estructura con porosidad baja no es el precursor de otra estructura con porosidad más alta. Nosotros observamos una evolución en las estructuras mostradas, pero esa evolución es alcanzada de forma independiente. La forma en la que presentamos estas imágenes, para mejor apreciación de la estructura, es multiplicando por tres la supercelda simulada en cada dirección. La simetría que se observa es, por tanto, ficticia y sólo debida a la presentación de las imágenes.

Nuestras estructuras no son simétricas, en primer lugar porque en los cálculos hechos con DMol³, ver **apéndice A.1**, no utilizamos simetría. Lo que sí utilizamos son las llamadas condiciones periódicas de frontera,⁵⁶ PBC. Con ellas se simula un sistema infinito, en el que no existe una superficie en el límite de la supercelda, lo que sucede durante el cálculo es que si, debido a la simulación, un átomo sale por el lado derecho de la supercelda otro átomo equivalente es introducido a ella por el lado izquierdo; es decir, una especie cercana al límite de la supercelda siente un potencial como si existiesen otras superceldas más allá de ese límite. Si no utilizásemos las PBC lo que tendríamos serían grupos de átomos aislados, en los que la única interacción posible se daría sólo dentro de la supercelda; es decir, se estaría hablando de *clusters* o grupos atómicos con superficie.

⁵⁶ Periodic boundary condition.

4.3.1 Estructuras desaleadas: Cu

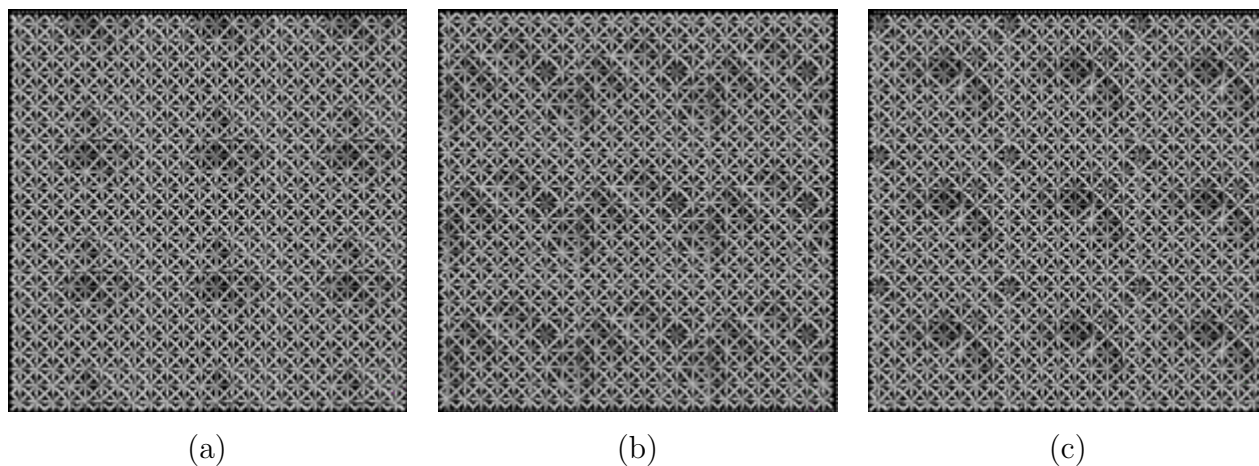


Figura 4.13 Estructura topológica del sistema dCu10, número de átomos 259, densidad 8.02780 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

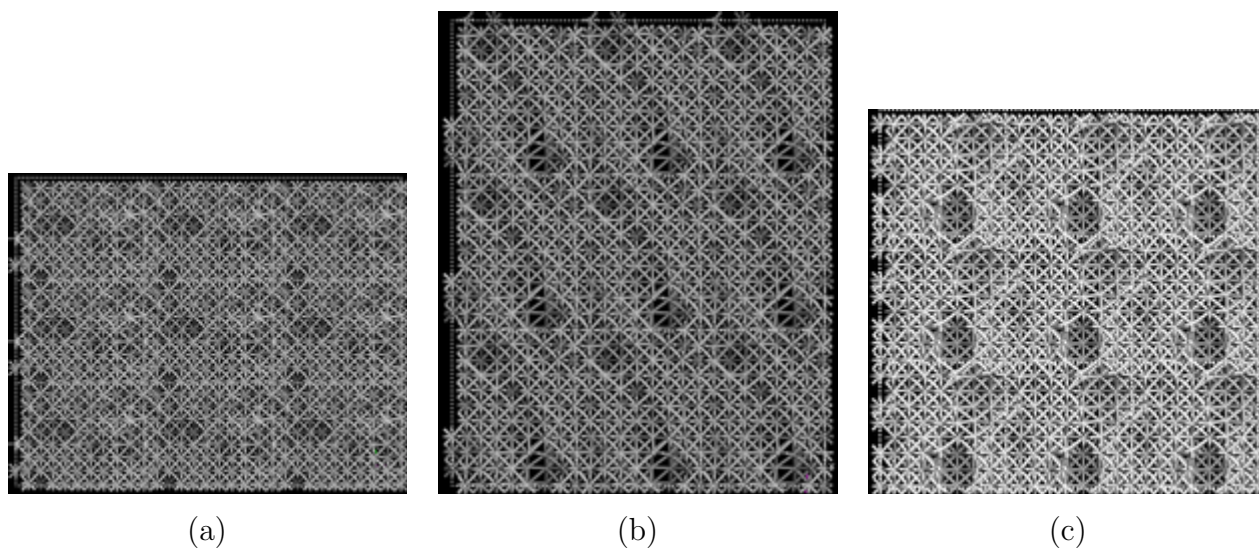


Figura 4.14 Estructura topológica del sistema dCu20, número de átomos 274, densidad 7.14823 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

La primera estructura corresponde al sistema dCu10, en el se observa una estructura casi cristalina, confirmando lo mostrado en la PDF de la **figura 4.1**. Se observa, además, pequeñas cavidades, dispersas por la supercelda, generadas en el proceso de desaleado y durante la dinámica molecular, estas cavidades rompen con la estructura cristalina y generan un ensanchamiento de los picos en la PDF. El aumento en la porosidad, sistema dCu20, no rompe aún con la cristalinidad del *backbone* pero existe mayor distorsión debido

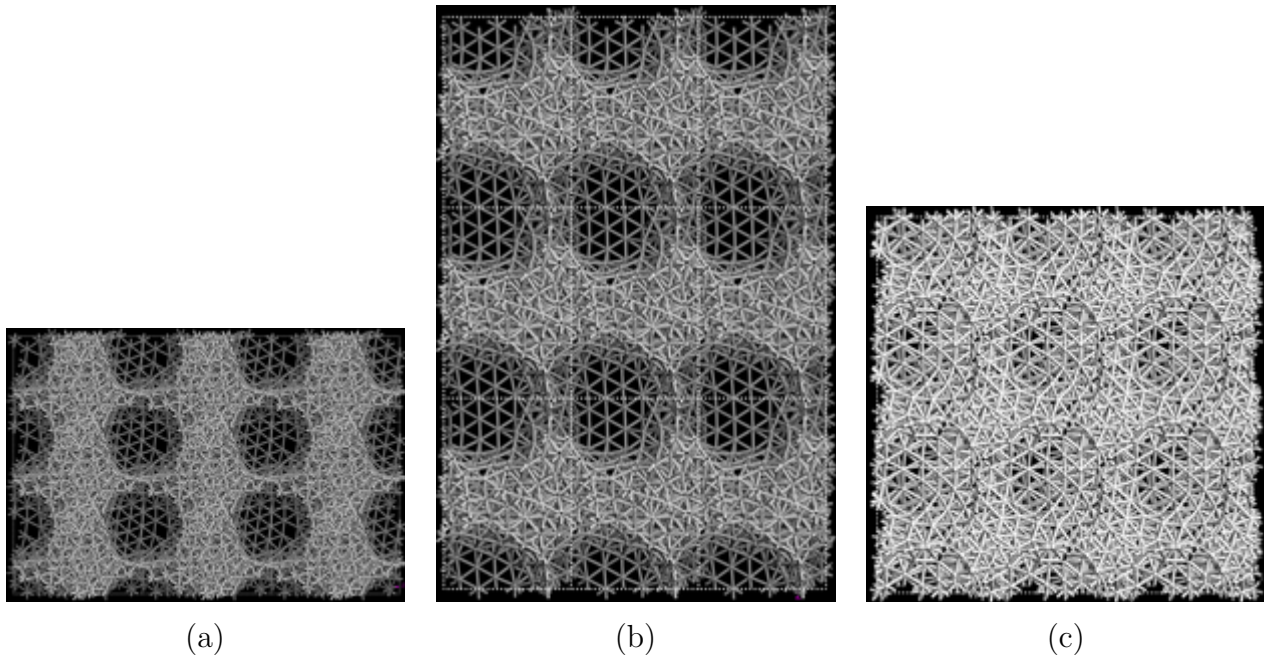


Figura 4.15 Estructura topológica del sistema dCu30, número de átomos 259, densidad 6.25936 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

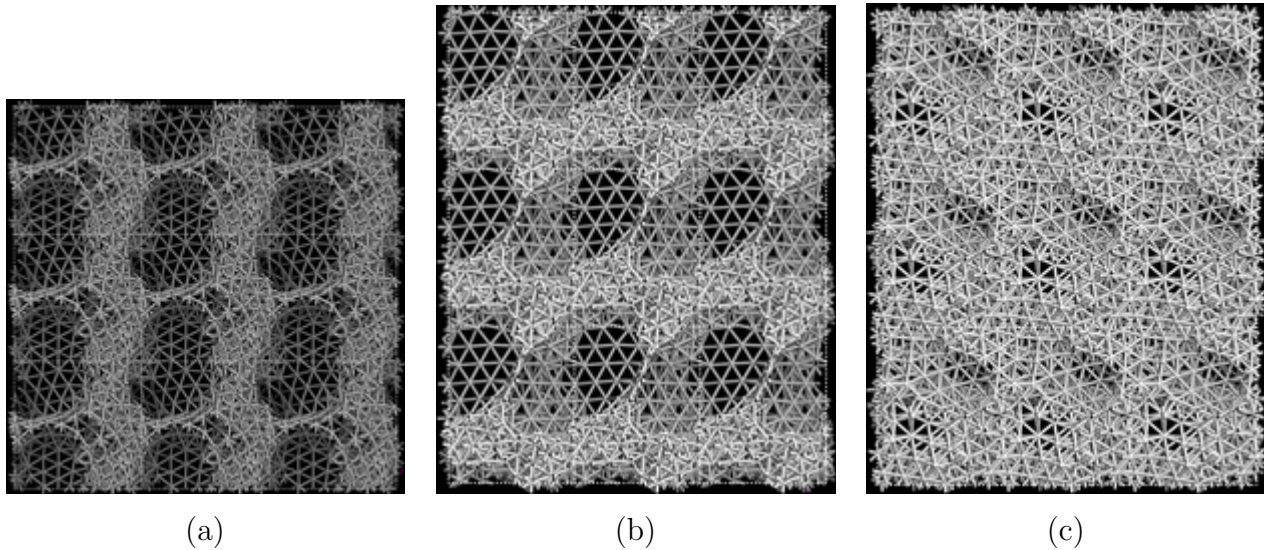


Figura 4.16 Estructura topológica del sistema dCu40, número de átomos 240, densidad 5.36118 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

a las cavidades formadas, lo que conlleva a un ensanchamiento aún mayor de los picos en la PDF con respecto a los picos de la PDF del sistema anterior. Es hasta la porosidad del 30% donde puede considerarse al sistema como un sistema no cristalino, en este sistema, dCu30, las pequeñas cavidades coalescen para formar una aún mayor, esta cavidad está

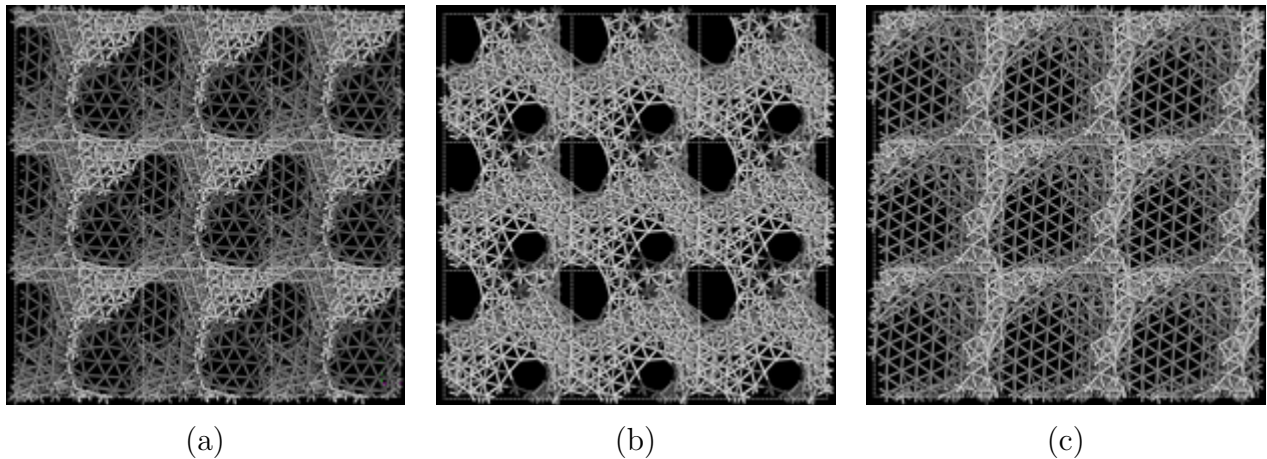


Figura 4.17 Estructura topológica del sistema dCu50, número de átomos 250, densidad 4.46765 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

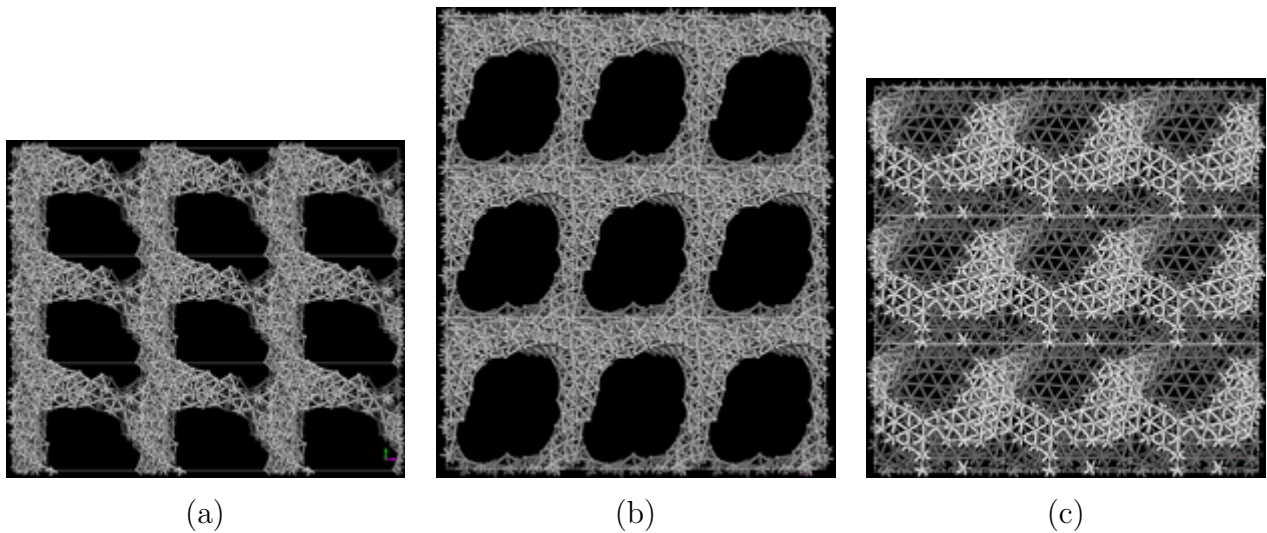


Figura 4.18 Estructura topológica del sistema dCu60, número de átomos 240, densidad 3.57412 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

restringida a la supercelda, por lo que no puede hablarse de poros en la estructura; aunque pueden observarse monocapas hexagonales que separan una cavidad de otra. La forma de estas cavidades nos ha llevado a acuñar el término *cazuelas* para referirnos a ellas de forma genérica. En el sistema dCu40 siguen existiendo cavidades, *cazuelas*, que no atraviesan la estructura, aunque podemos observar menor regularidad en la estructura hexagonal que las limita. En la estructura dCu50 por primera vez las cavidades atraviesan la estructura y forman lo que podríamos llamar un poro real, al menos en una dirección. Las estructura dCu60, dCu70 y dCu80 son estructuras totalmente porosas, con poros «tubulares». Al

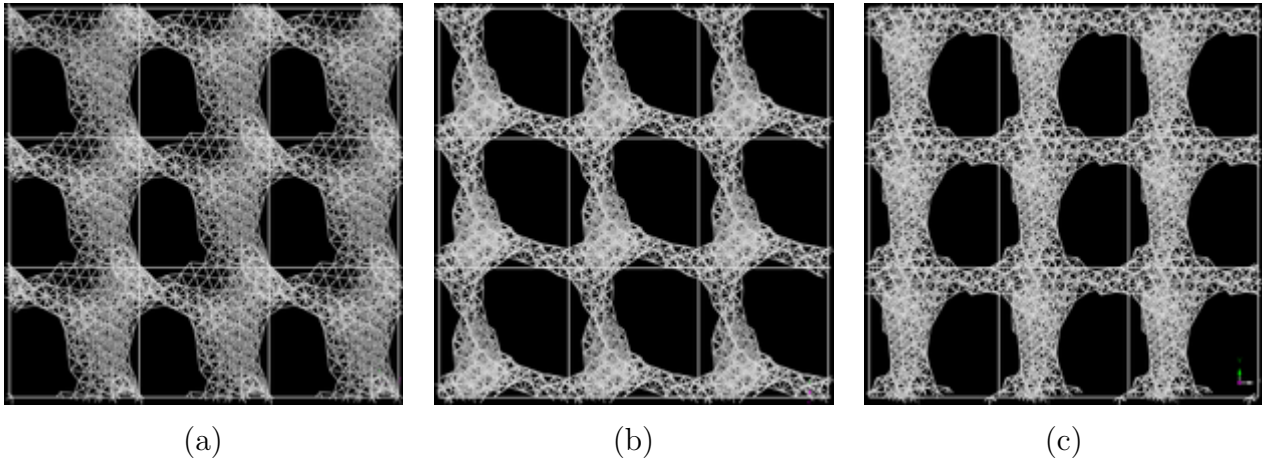


Figura 4.19 Estructura topológica del sistema dCu70, número de átomos 269, densidad 2.67852 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

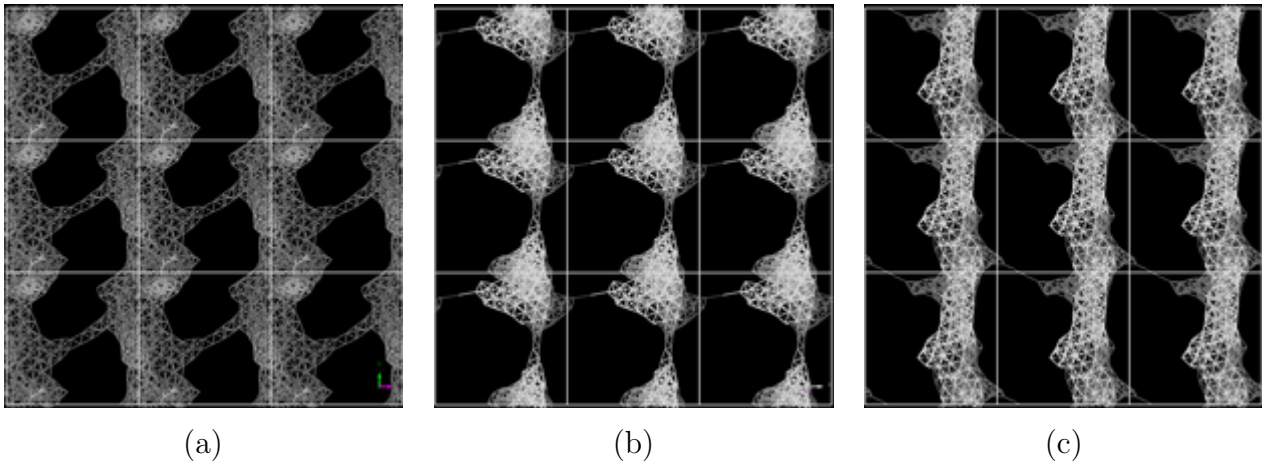


Figura 4.20 Estructura topológica del sistema dCu80, número de átomos 256, densidad 1.78445 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

observar la estructura dCu80, nos surgió una pregunta, ¿que tan estables mecánicamente son estas estructuras? lo preguntamos porque observamos capas discontinuas de átomos —**figura 4.20(a)**— unidas por delgadas estructuras —**figura 4.20(b)–(c)**—. Nosotros impedimos que el volumen de la estructura varíe, de no ser así, probablemente esta estructura colapsaría; no lo sabemos, nosotros hemos establecido una metodología y nos hemos apegado a ella. Además creemos que si liberásemos la restricción del volumen, obtendríamos alguna de las estructuras anteriores de menor porosidad, lo que nos llevaría a establecer esta estructura como inexistente. El último sistema desaleado de Cu, dCu90, es un sistema que, simplemente está en el límite de nuestra metodología. Es un sistema

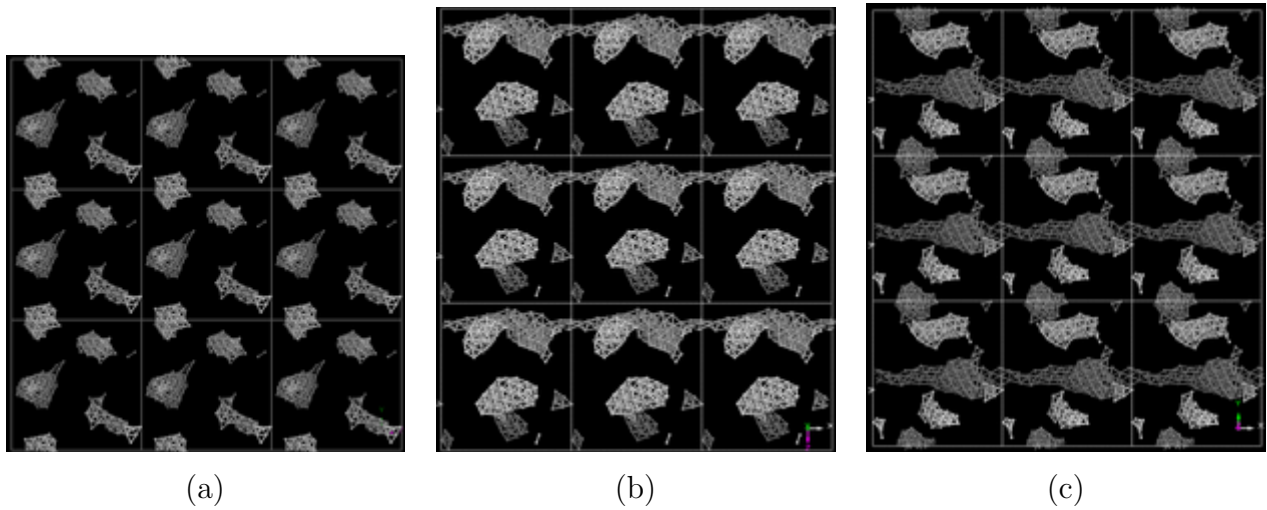


Figura 4.21 Estructura topológica del sistema dCu90, número de átomos 230, densidad 0.892840 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

que presenta grupos aislados de átomos dentro de la supercelda, lo relevante de presentar ésta y todas las otras estructuras con 90 % de porosidad es comprobar que efectivamente la estructura a corto alcance se mantiene, es decir, al observar las PDF y las curvas de distribución de ángulos planos, apreciamos estructuras similares. Esto podría dar pie a afirmar que estos grupos de átomos son la configuración base que conforma el *backbone* de todas las estructuras.

4.3.2 Estructuras desaleadas: Ag

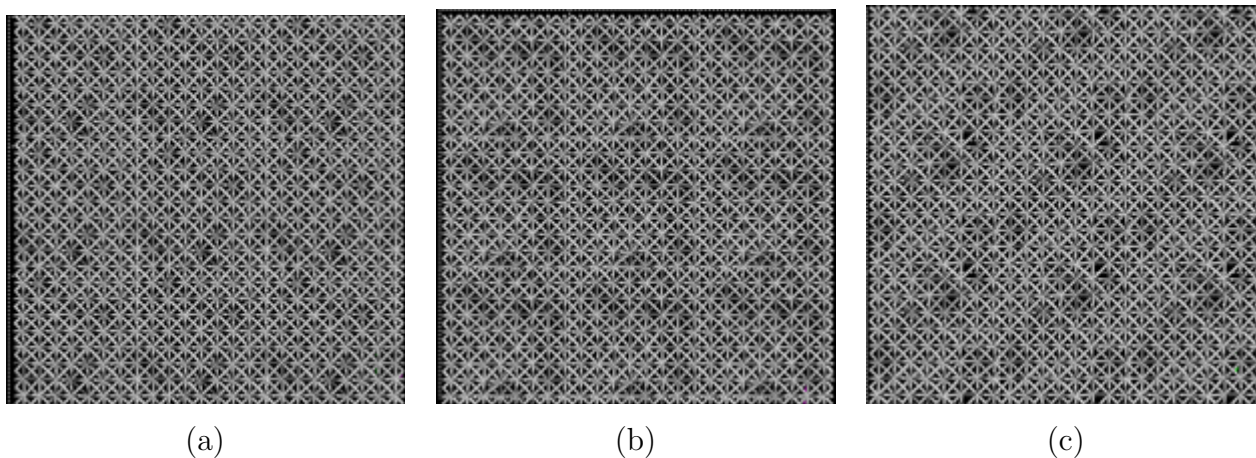


Figura 4.22 Estructura topológica del sistema dAg10, número de átomos 259, densidad 9.44100 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

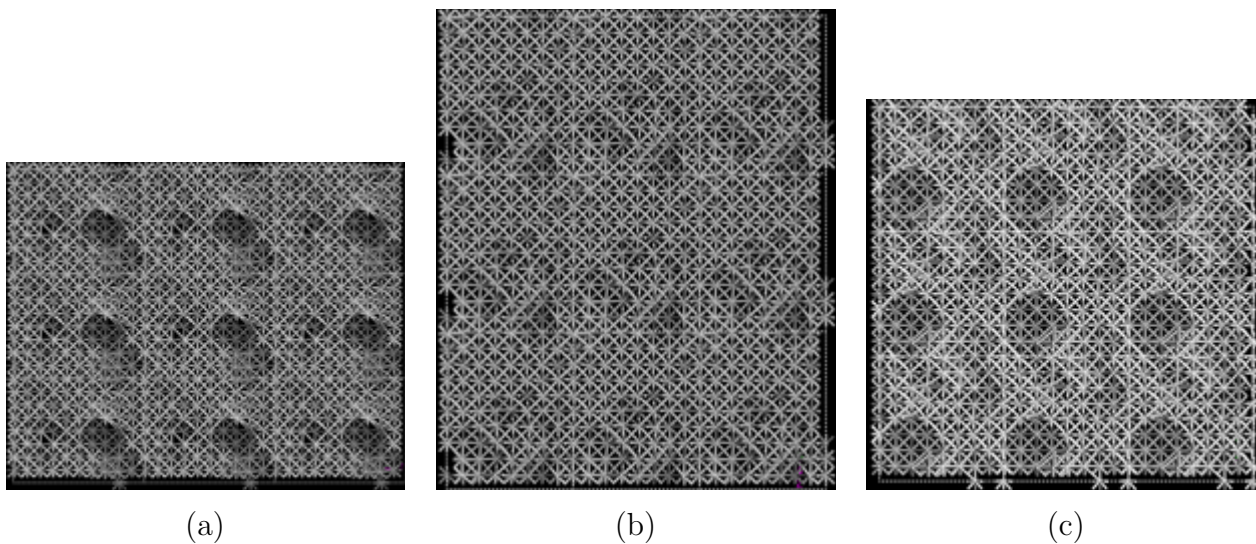


Figura 4.23 Estructura topológica del sistema dAg20, número de átomos 274, densidad 8.40659 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

La evolución del sistema dAg es similar a la presentada para el sistema dCu con ligeras variaciones como la aparición de poros verdaderos desde el 40 % de porosidad, la presencia de una estructura «colapsable» ya en dAg60. y quizá la variación más importante es la formación de estructuras menos aglomeradas (con más superficie) lo que se ve reflejado en la disminución en el número de coordinación, **tabla 4.4**.

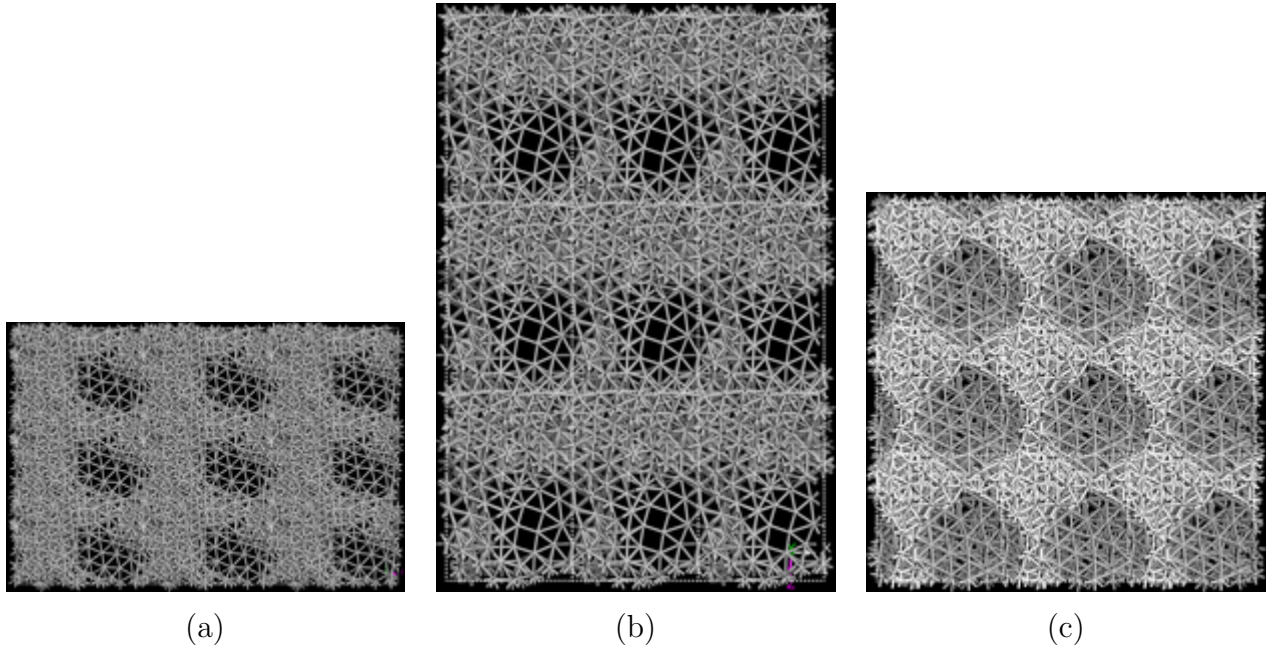


Figura 4.24 Estructura topológica del sistema dAg30, número de átomos 259, densidad 7.36124 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

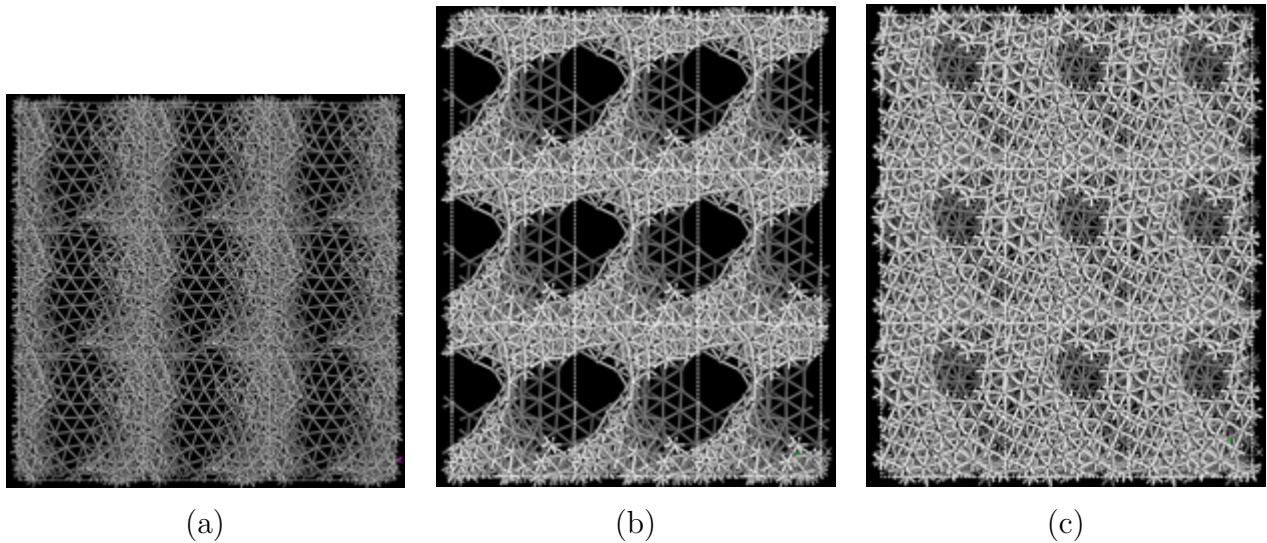


Figura 4.25 Estructura topológica del sistema dAg40, número de átomos 240, densidad 6.30495 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

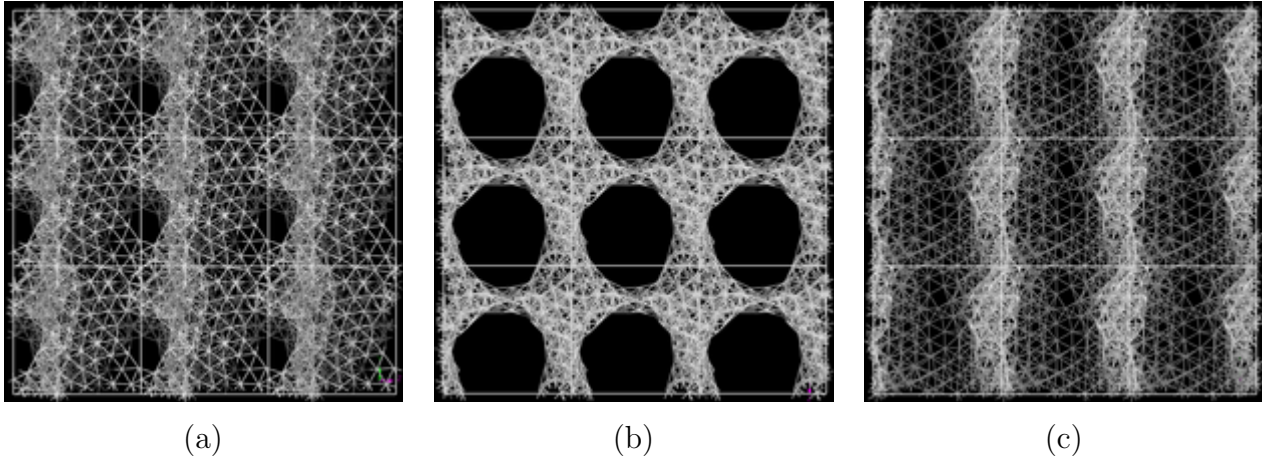


Figura 4.26 Estructura topológica del sistema dAg50, número de átomos 250, densidad 5.25412 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

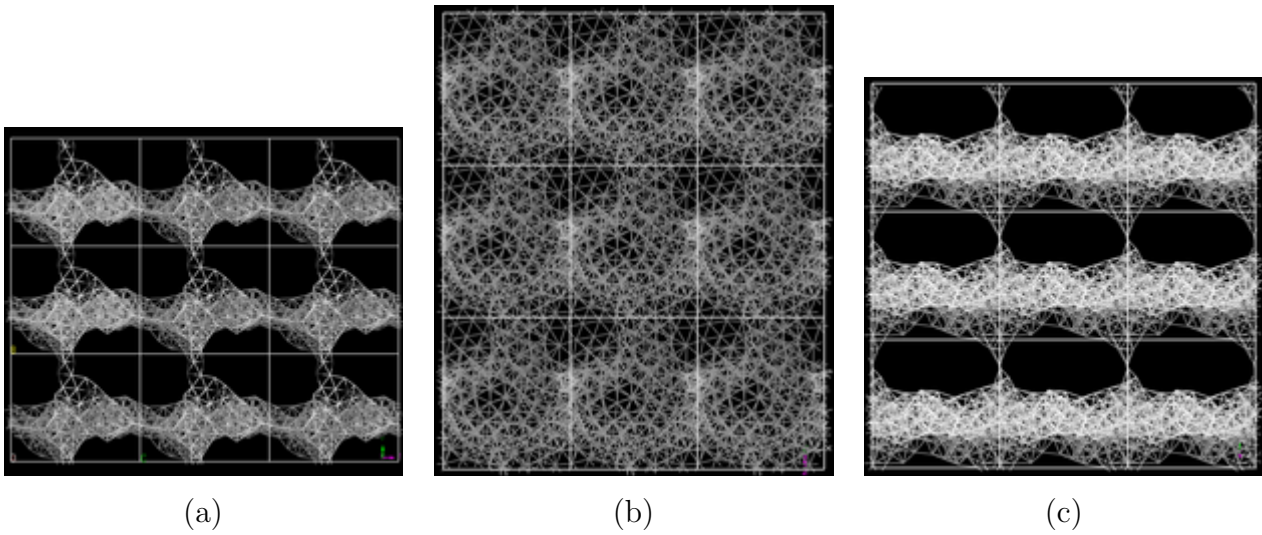
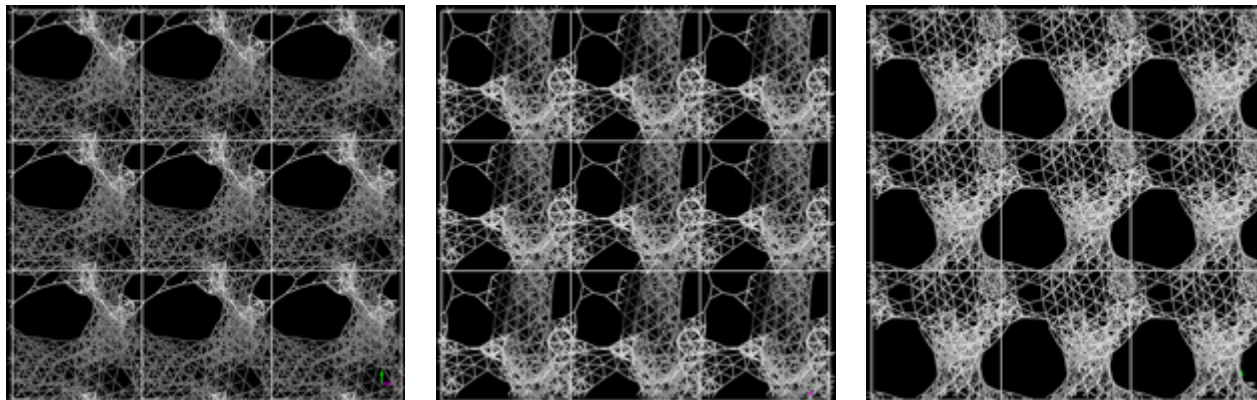


Figura 4.27 Estructura topológica del sistema dAg60, número de átomos 240, densidad 4.20330 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

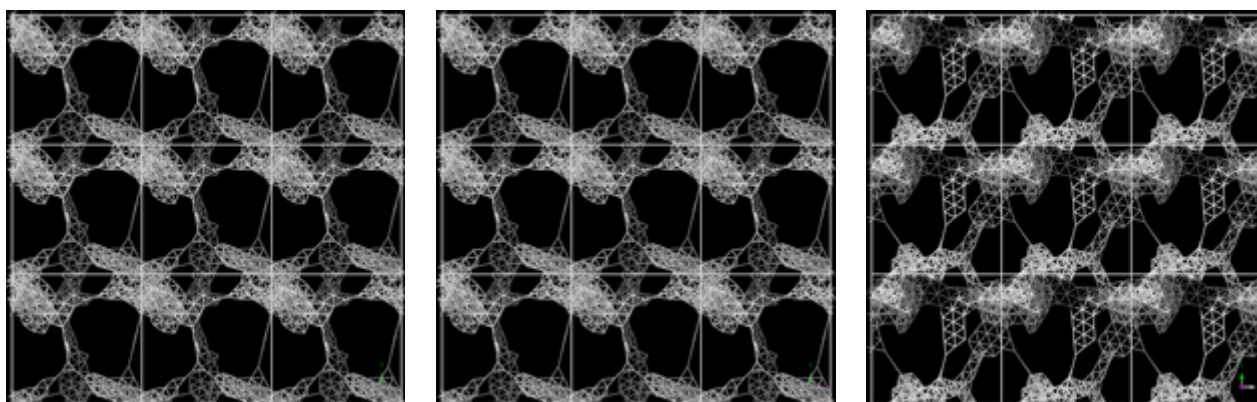


(a)

(b)

(c)

Figura 4.28 Estructura topológica del sistema dAg70, número de átomos 269, densidad 3.15004 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.



(a)

(b)

(c)

Figura 4.29 Estructura topológica del sistema dAg80, número de átomos 256, densidad 2.09859 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

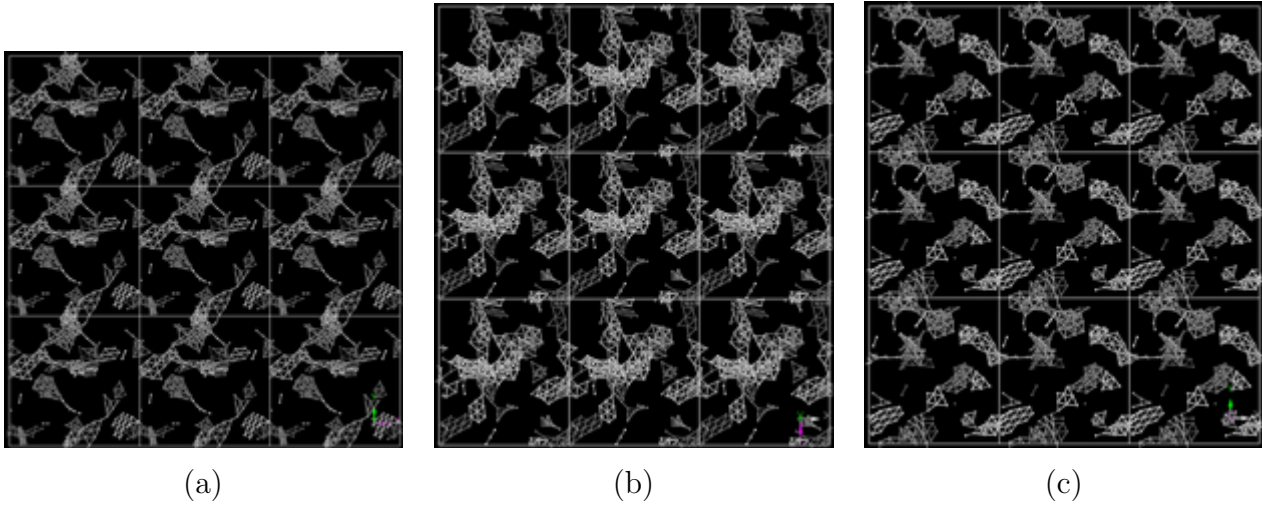


Figura 4.30 Estructura topológica del sistema dAg90, número de átomos 230, densidad 1.05010 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

4.3.3 Estructuras desaleadas: Au

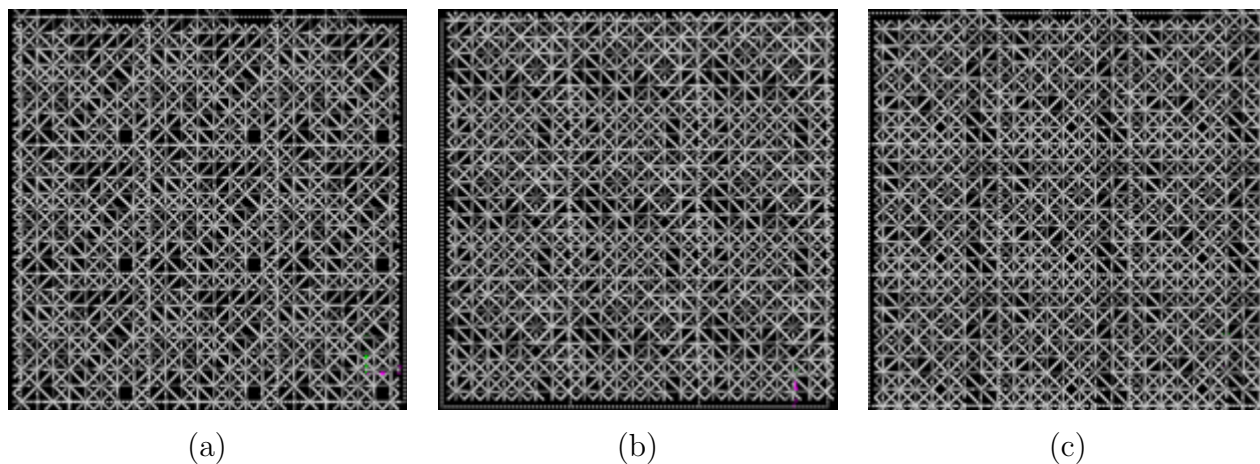


Figura 4.31 Estructura topológica del sistema dAu10, número de átomos 259, densidad 17.3294 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

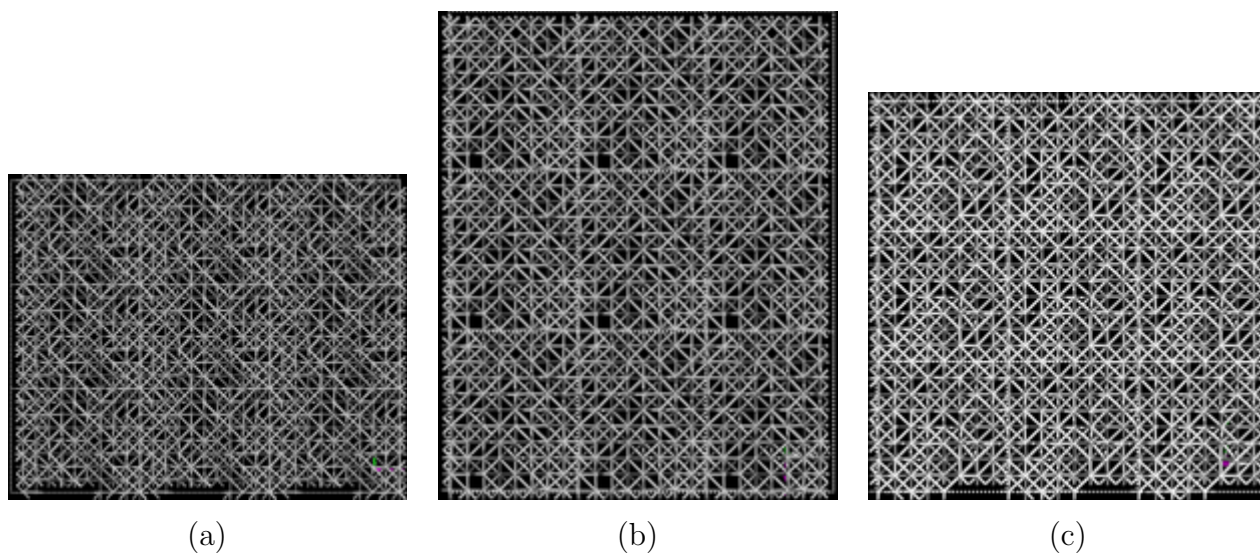


Figura 4.32 Estructura topológica del sistema dAu20, número de átomos 274, densidad 15.4307 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

El oro al ser el elemento más pesado de los tres estudiados tiene menor movilidad a la misma temperatura, por lo que puede justificarse el porqué la estructura cristalina no desaparece con la porosidad de 30% como lo hacen los sistemas dCu30 y dAg30. Por lo demás la estructura topológica del sistema dAu tiene un comportamiento similar al de la estructura dAg.

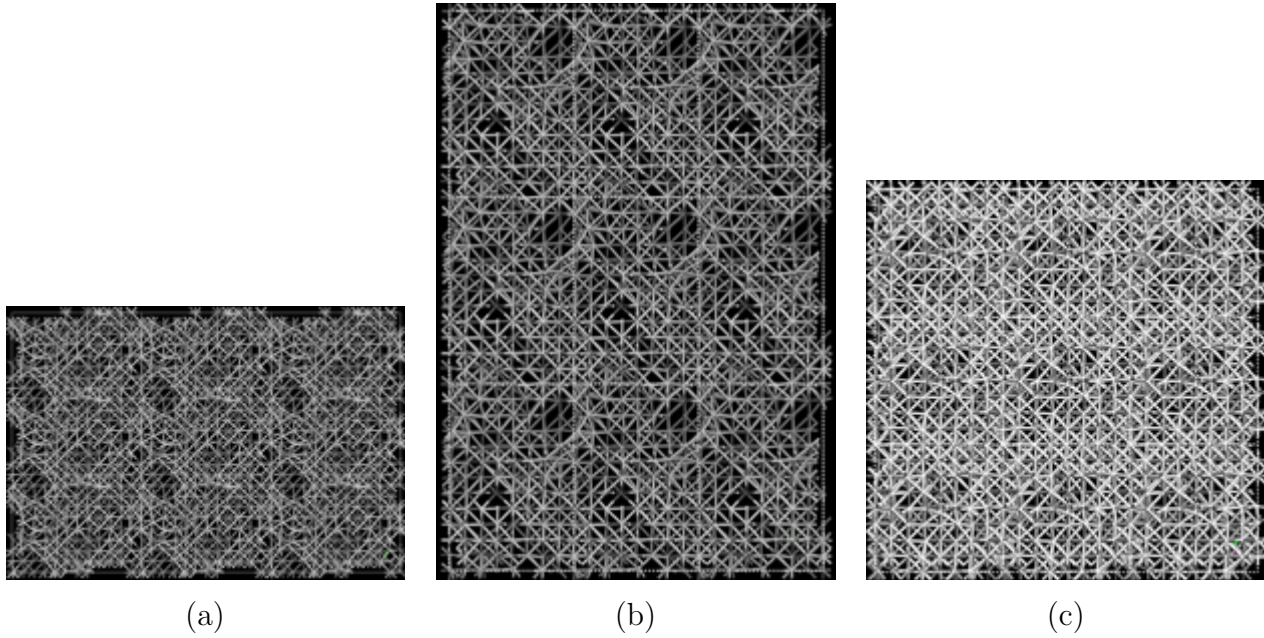


Figura 4.33 Estructura topológica del sistema dAu30, número de átomos 259, densidad 13.5119 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

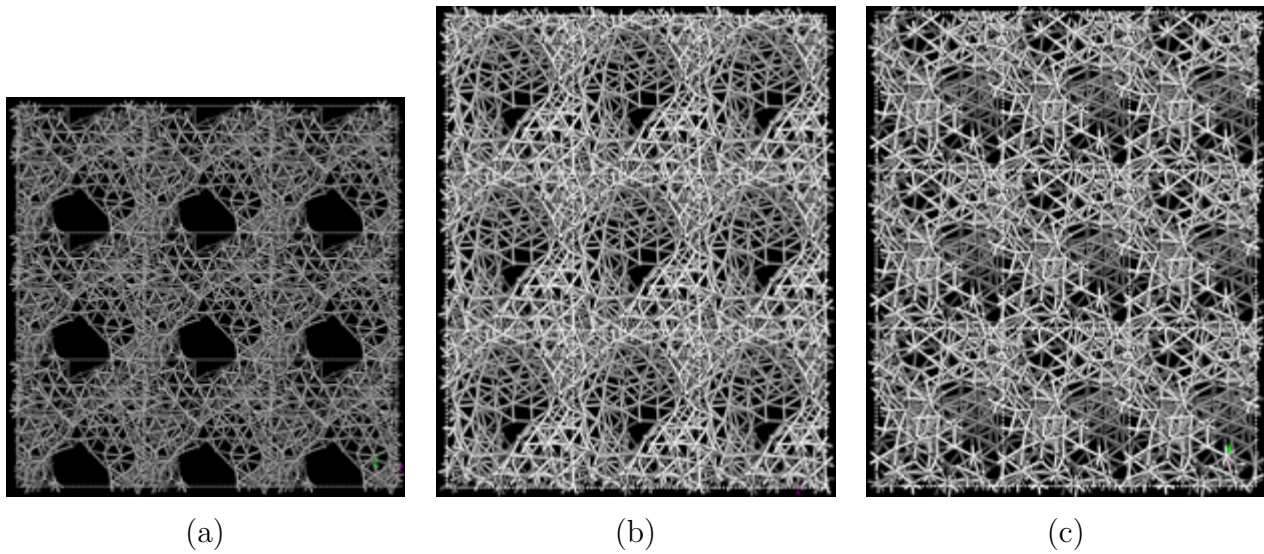


Figura 4.34 Estructura topológica del sistema dAu40, número de átomos 240, densidad 11.5730 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

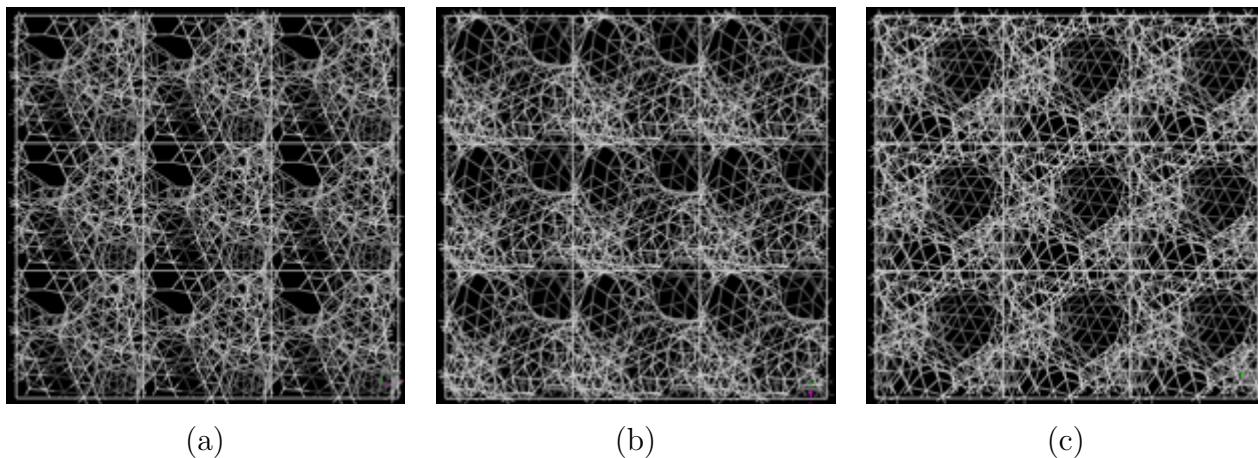


Figura 4.35 Estructura topológica del sistema dAu50, número de átomos 250, densidad 9.64418 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

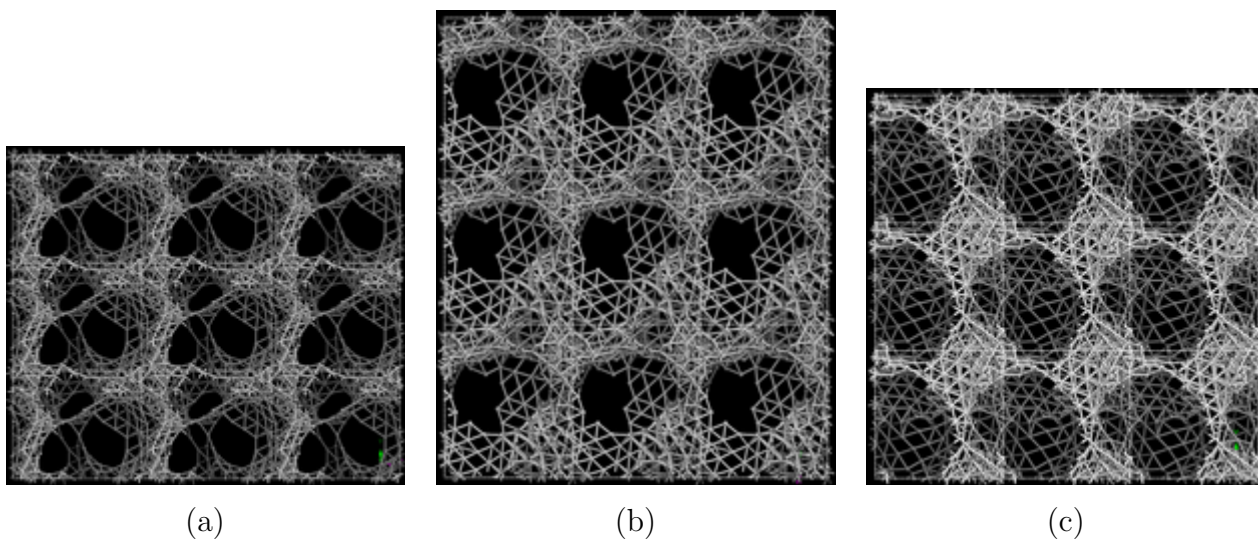


Figura 4.36 Estructura topológica del sistema dAu60, número de átomos 240, densidad 7.71534 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

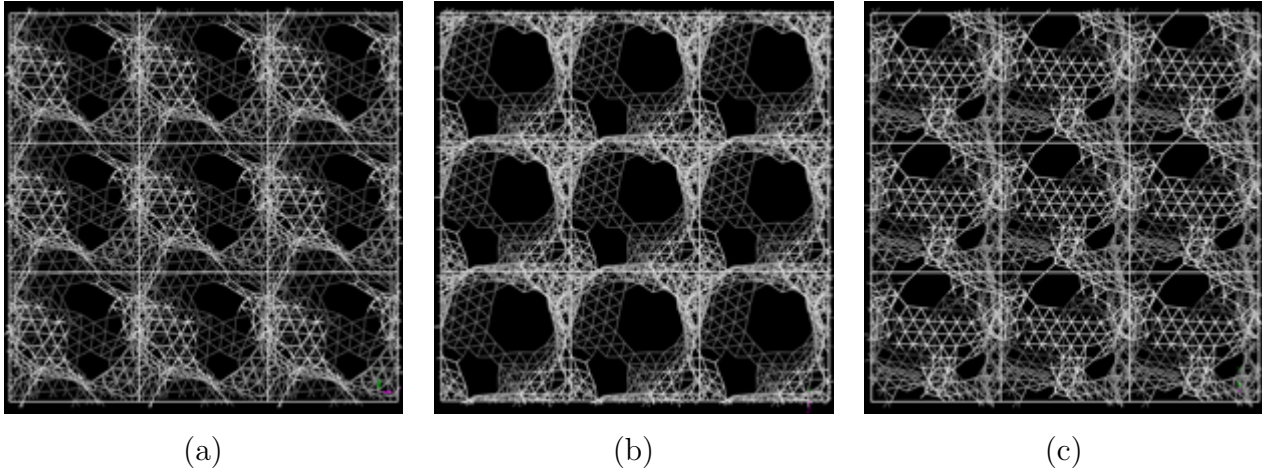


Figura 4.37 Estructura topológica del sistema dAu70, número de átomos 269, densidad 5.78204 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

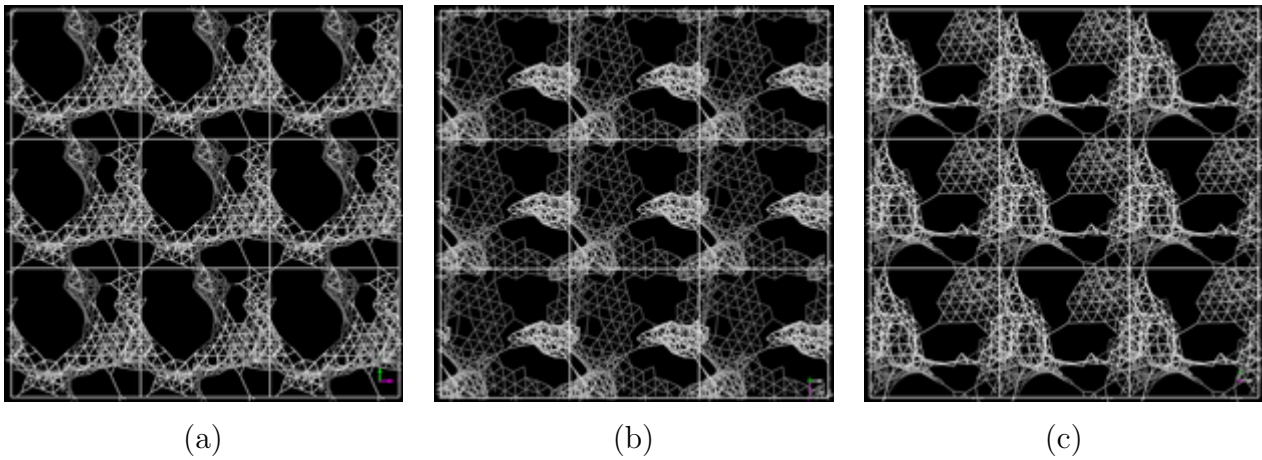


Figura 4.38 Estructura topológica del sistema dAu80, número de átomos 256, densidad 3.85205 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

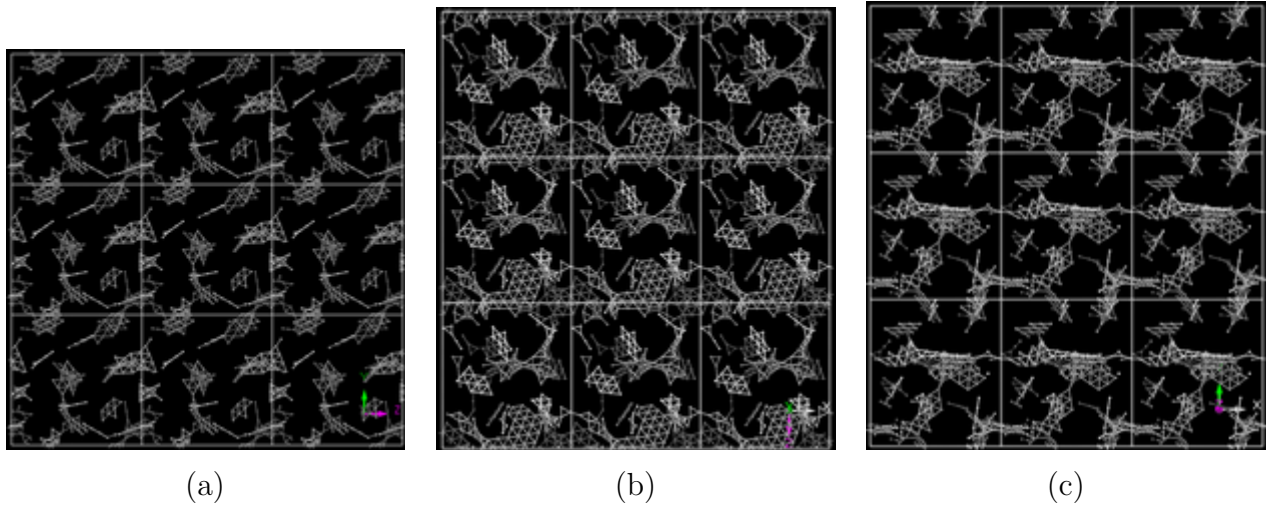
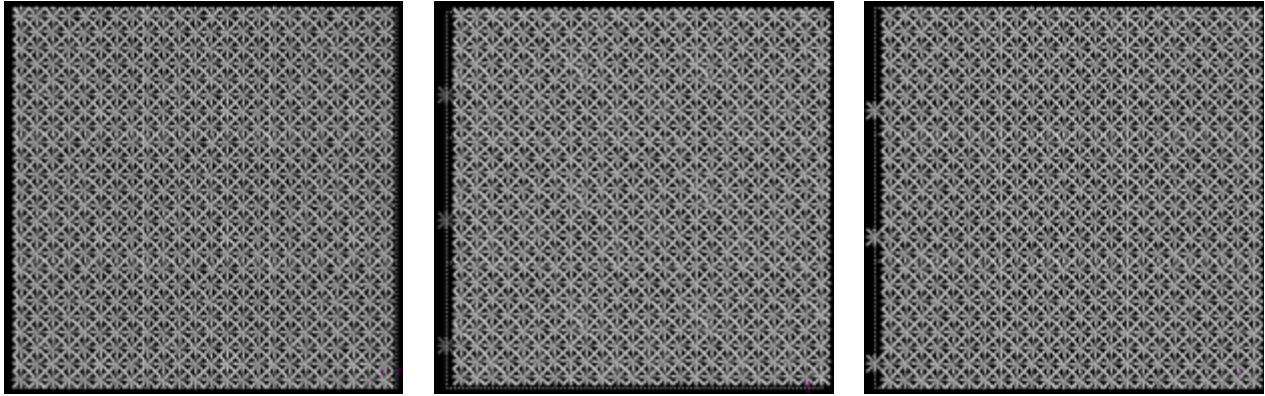


Figura 4.39 Estructura topológica del sistema dAu90, número de átomos 230, densidad 1.92735 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

4.3.4 Estructuras expandidas: Cu, Ag, Au

¿Cuál es la diferencia principal en cuanto a la formación de poros mediante la técnica de desaleado y la técnica de expansión de celda? Los poros en el desaleado se forman a partir de los lugares dejados por los átomos eliminados, lugares que por cierto fueron asignados aleatoriamente. En la ER, en cambio, la localización de los poros, en principio, está condicionada a la dirección de las velocidades iniciales, pues los átomos en la mayoría de los sistemas expandidos no se encuentran a distancia de enlace. Por ejemplo, si dos átomos avanzan en direcciones contrarias, es probable que el punto del cual se alejan dé origen a un poro. Ahora, el *backbone* se forma, también de diferente manera, en el desaleado ya se encuentra prácticamente constituido y durante la dinámica hay un reacomodo que lleva a la estructura final. En la ER no existe tal estructura, el *backbone* se forma por el acercamiento aleatorio de átomos, una vez dado el acercamiento, la dinámica continúa con el reacomodo de átomos. Sin embargo, de forma general no se observa una diferencia significativa en las estructuras obtenidas de estas dos técnicas.

Como se mencionó en la **sección 2.4**, las estructuras obtenidas con la ER y 10% de porosidad son prácticamente cristalinas, pero esta estructura es una estructura tensada pues los átomos se comportan como si estuvieran enlazados pero se encuentran más separados de lo normal. Es por esto que la **figura 4.40** muestra una estructura homogénea y cristalina. Las PDF de las estructuras expandidas revelan que el comportamiento cristalino no se mantiene más allá del 10% de porosidad, esto es una consecuencia debido, como se dijo arriba, de la forma en la que se enlazan los átomos para formar el *backbone*. Pero más allá del 20% de porosidad para el Cu y para la Ag y 30% para el Au, el comportamiento topológico de los sistemas tanto desaleado como expandido, es similar.

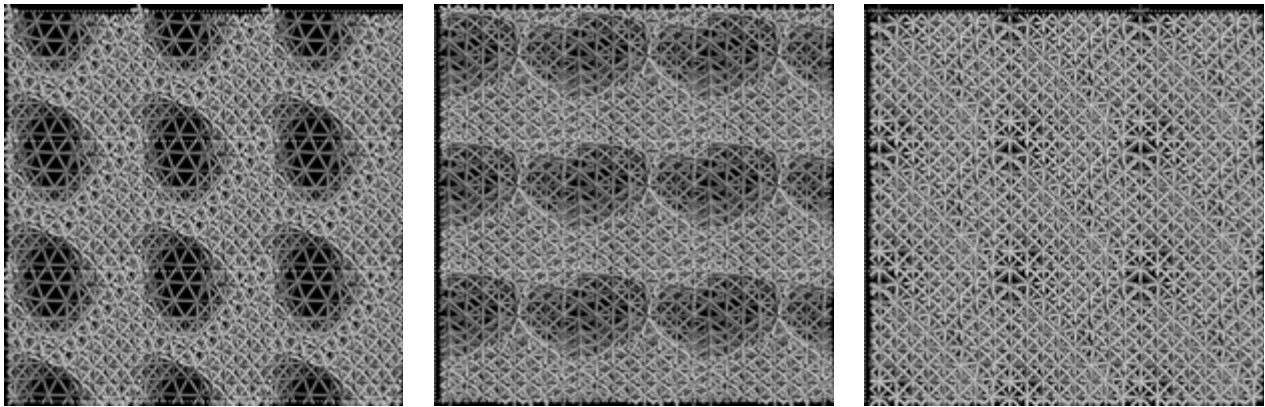


(a)

(b)

(c)

Figura 4.40 Estructura topológica del sistema eCu10, número de átomos 256, densidad 8.05975 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.



(a)

(b)

(c)

Figura 4.41 Estructura topológica del sistema eCu20, número de átomos 256, densidad 7.15985 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

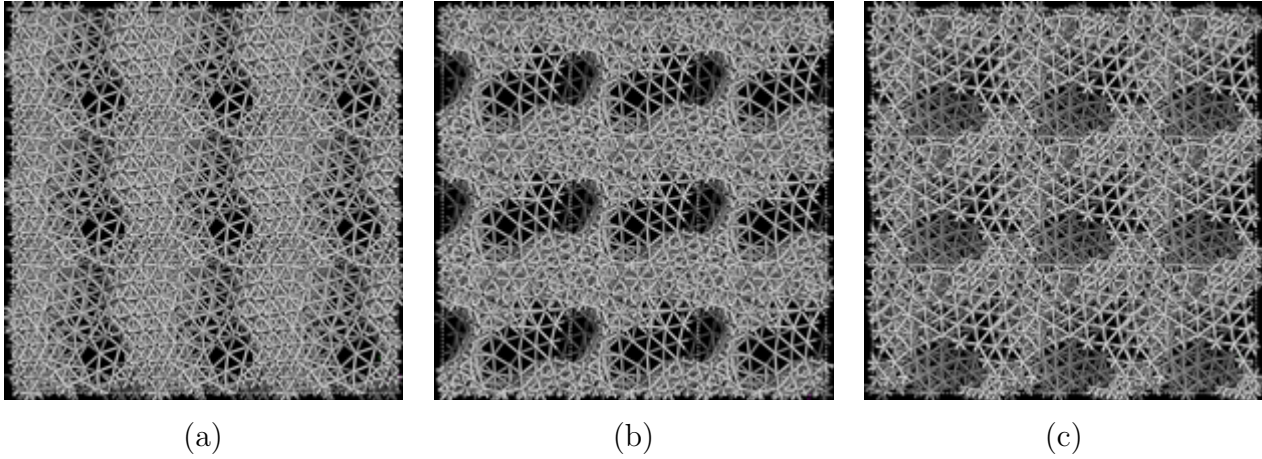


Figura 4.42 Estructura topológica del sistema eCu30, número de átomos 256, densidad 6.26982 g/cm³; (a) dirección x, (b) dirección y, (c) dirección z.

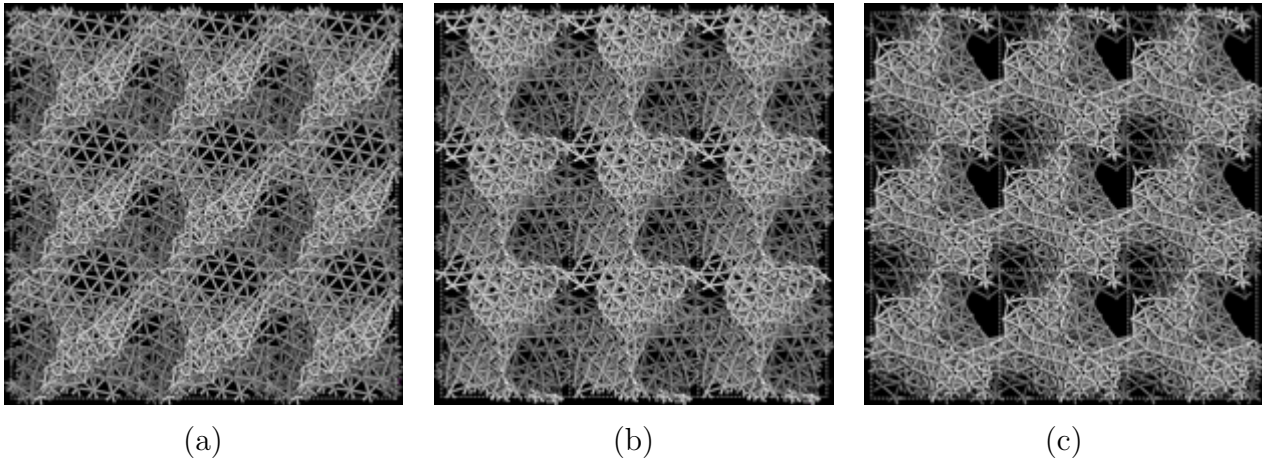


Figura 4.43 Estructura topológica del sistema eCu40, número de átomos 256, densidad 5.36986 g/cm³; (a) dirección x, (b) dirección y, (c) dirección z.

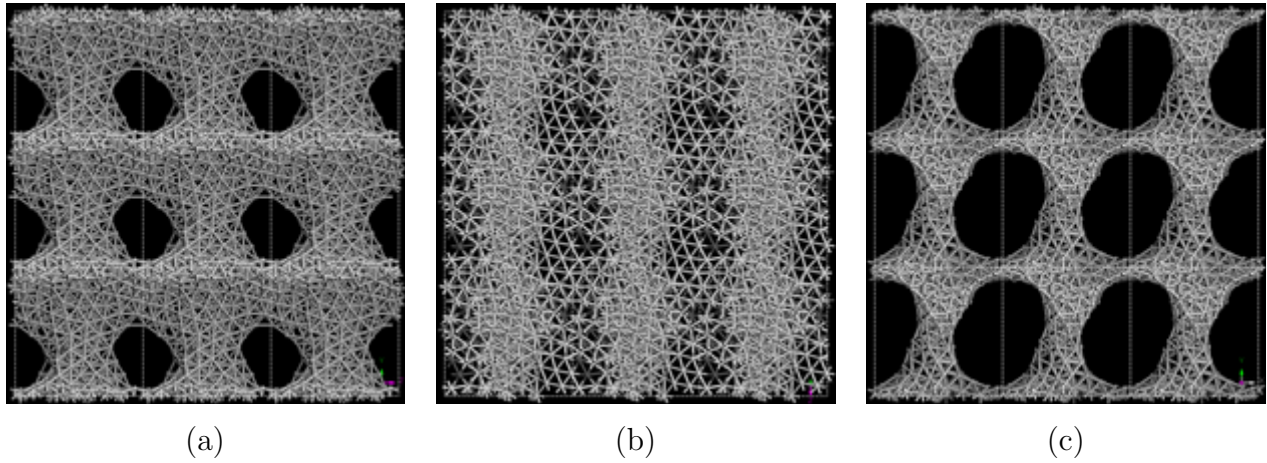


Figura 4.44 Estructura topológica del sistema eCu50, número de átomos 256, densidad 4.47992 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

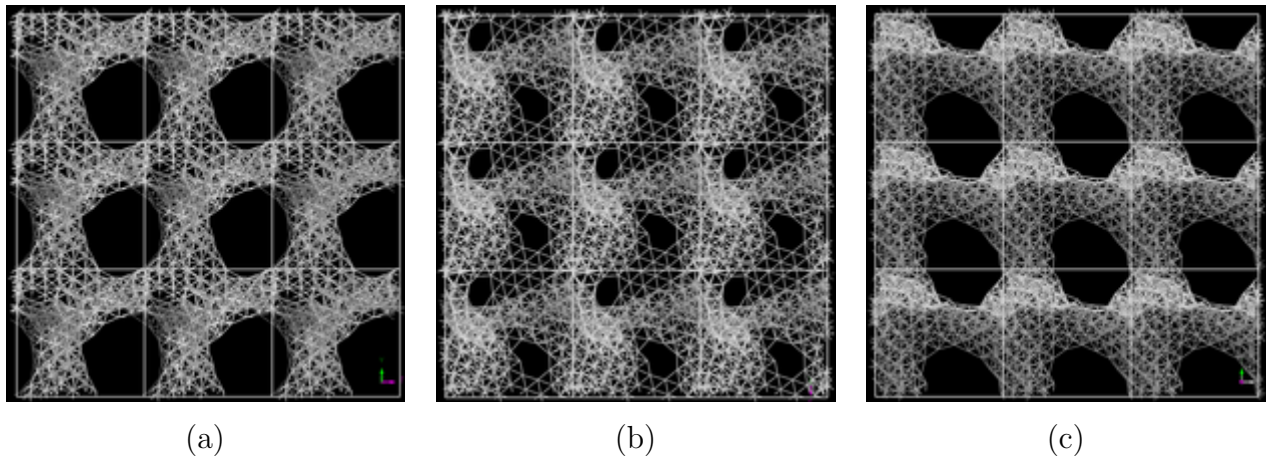
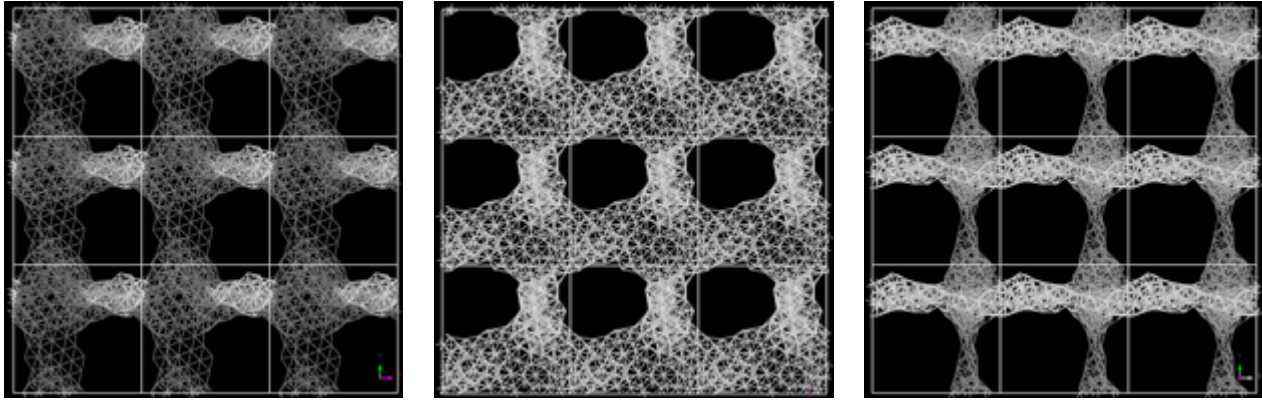


Figura 4.45 Estructura topológica del sistema eCu60, número de átomos 256, densidad 3.57991 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

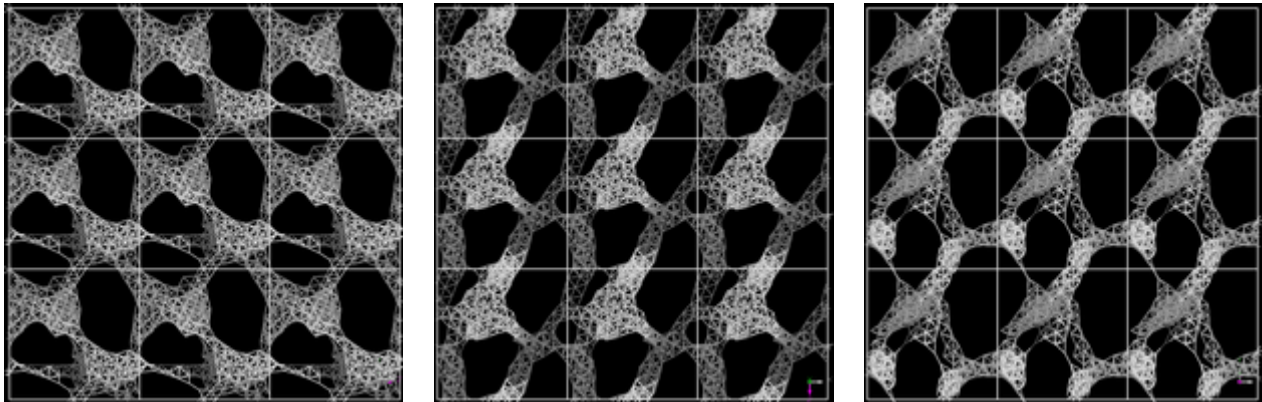


(a)

(b)

(c)

Figura 4.46 Estructura topológica del sistema eCu70, número de átomos 256, densidad 2.67994 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.



(a)

(b)

(c)

Figura 4.47 Estructura topológica del sistema eCu80, número de átomos 256, densidad 1.78996 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

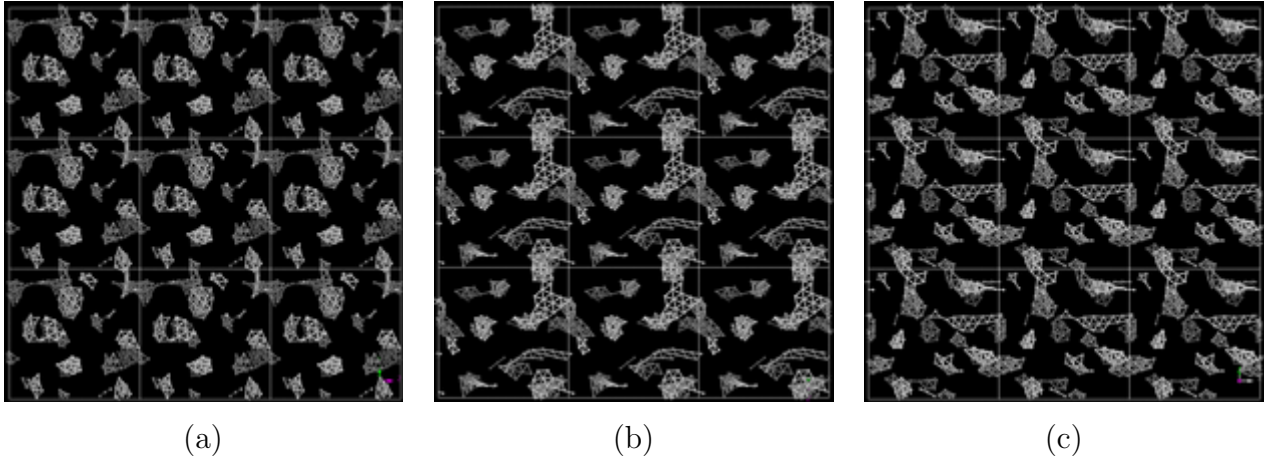


Figura 4.48 Estructura topológica del sistema eCu90, número de átomos 256, densidad 0.88998 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

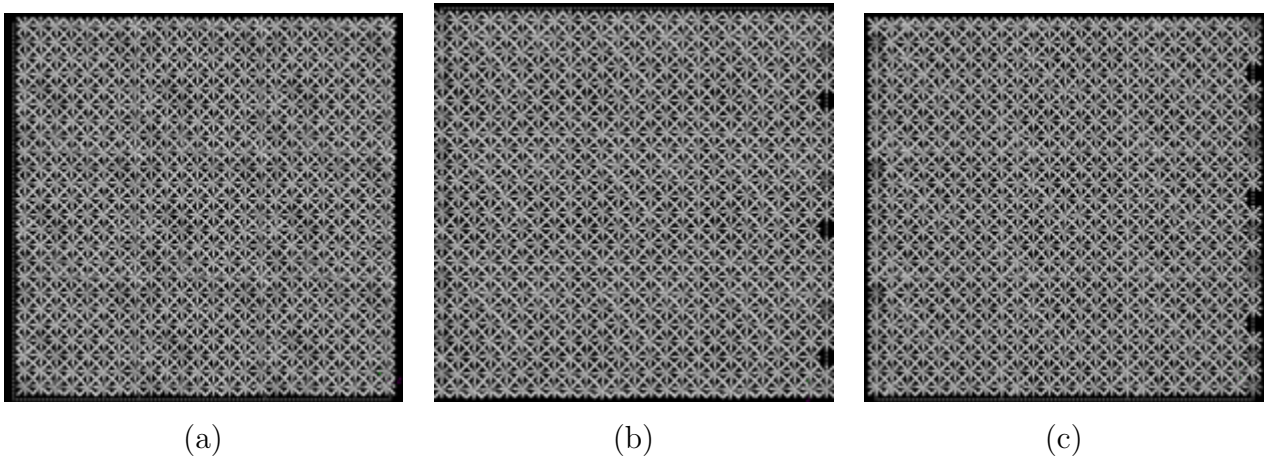
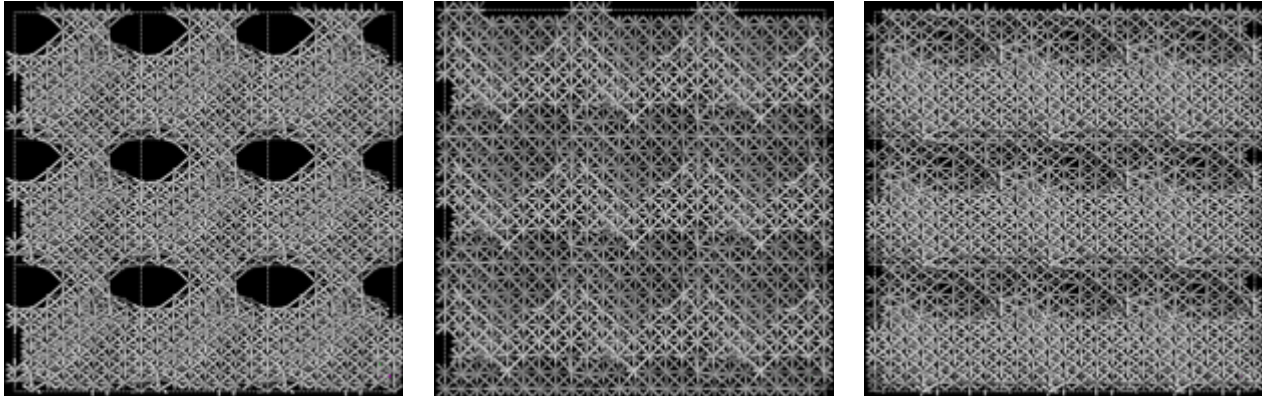


Figura 4.49 Estructura topológica del sistema eAg10, número de átomos 256, densidad 9.44085 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

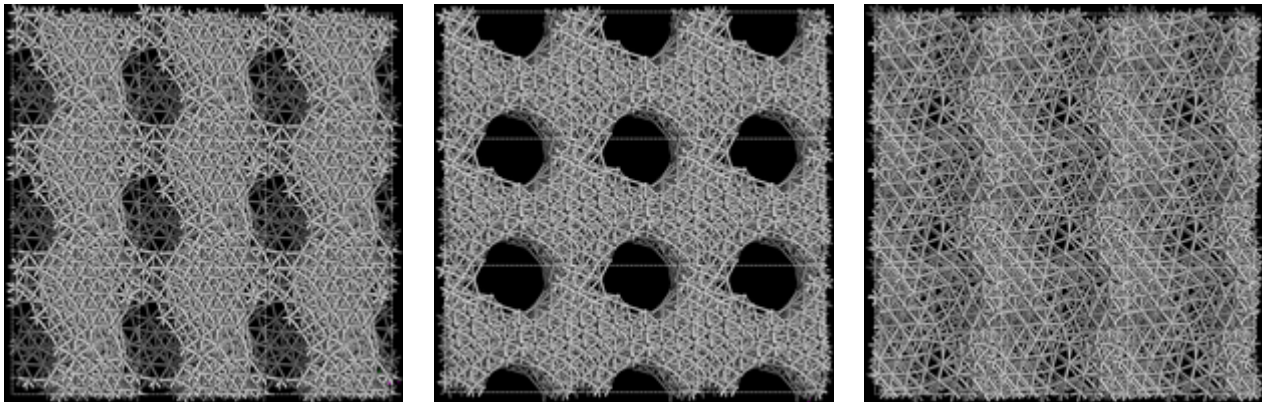


(a)

(b)

(c)

Figura 4.50 Estructura topológica del sistema eAg20, número de átomos 256, densidad 8.39175 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.



(a)

(b)

(c)

Figura 4.51 Estructura topológica del sistema eAg30, número de átomos 256, densidad 7.34277 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

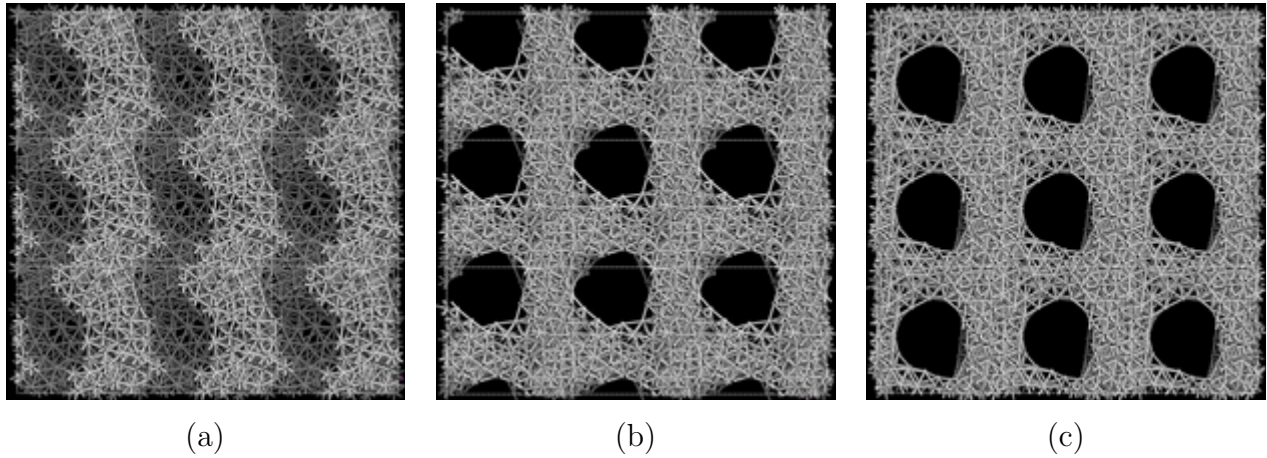


Figura 4.52 Estructura topológica del sistema eAg40, número de átomos 256, densidad 6.29386 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

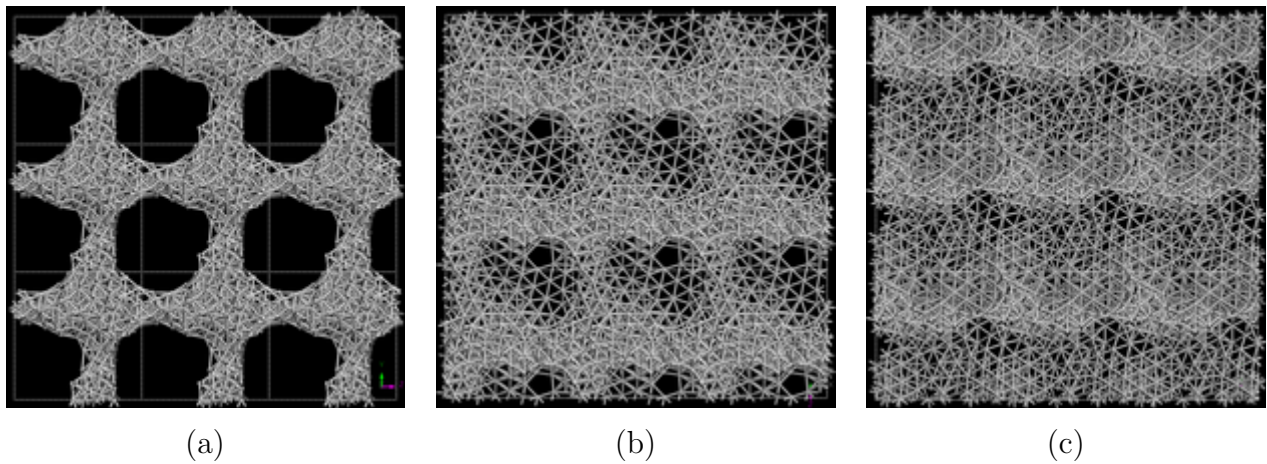
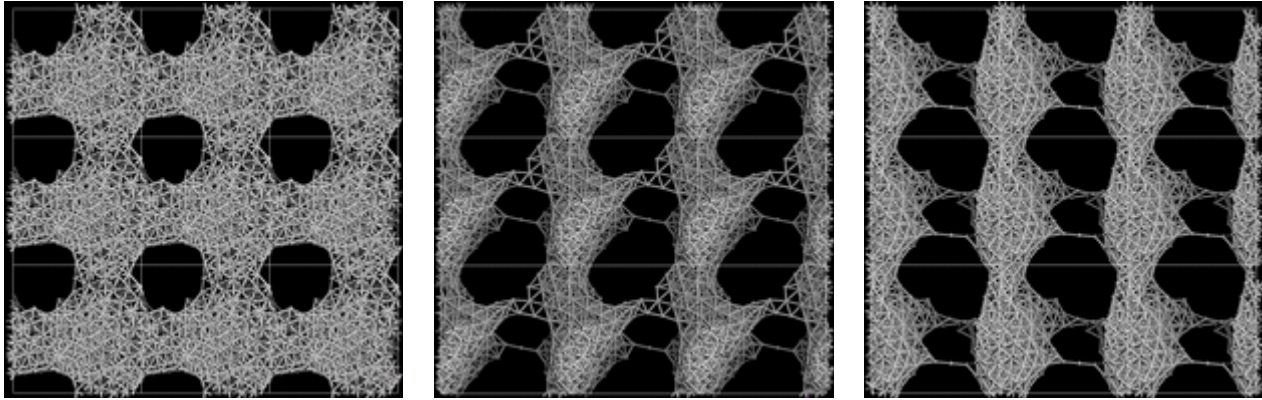


Figura 4.53 Estructura topológica del sistema eAg50, número de átomos 256, densidad 5.24489 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

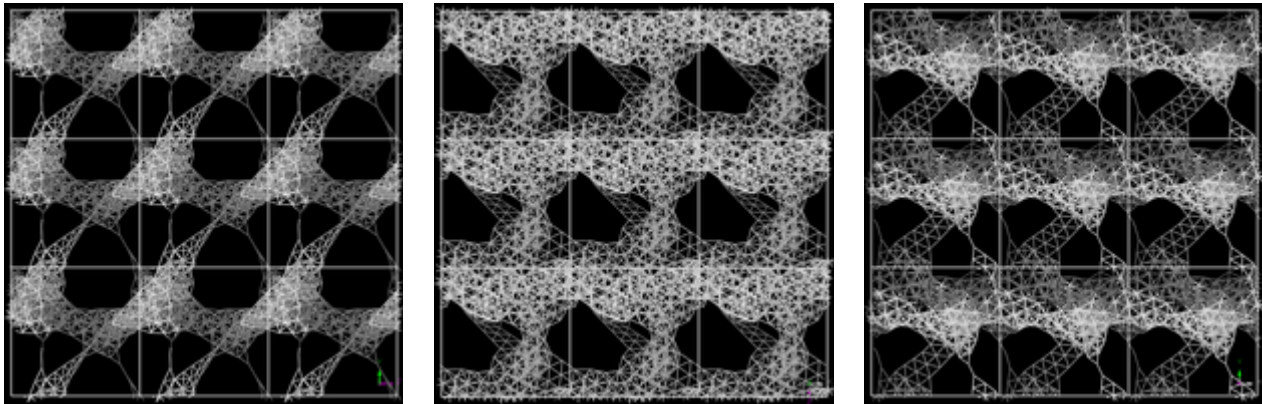


(a)

(b)

(c)

Figura 4.54 Estructura topológica del sistema eAg60, número de átomos 256, densidad 4.19587 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.



(a)

(b)

(c)

Figura 4.55 Estructura topológica del sistema eAg70, número de átomos 256, densidad 3.14690 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

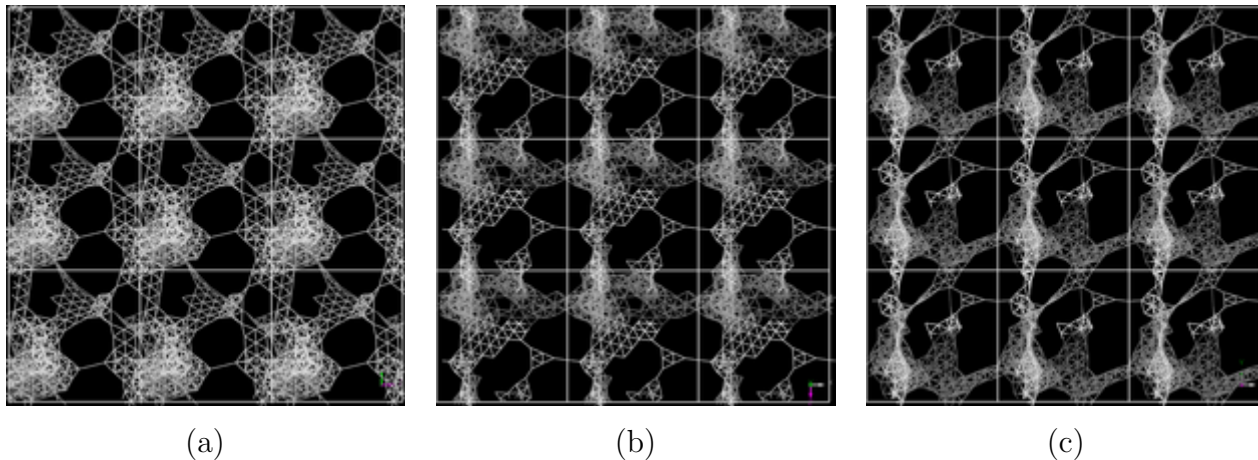


Figura 4.56 Estructura topológica del sistema eAg80, número de átomos 256, densidad 2.09794 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

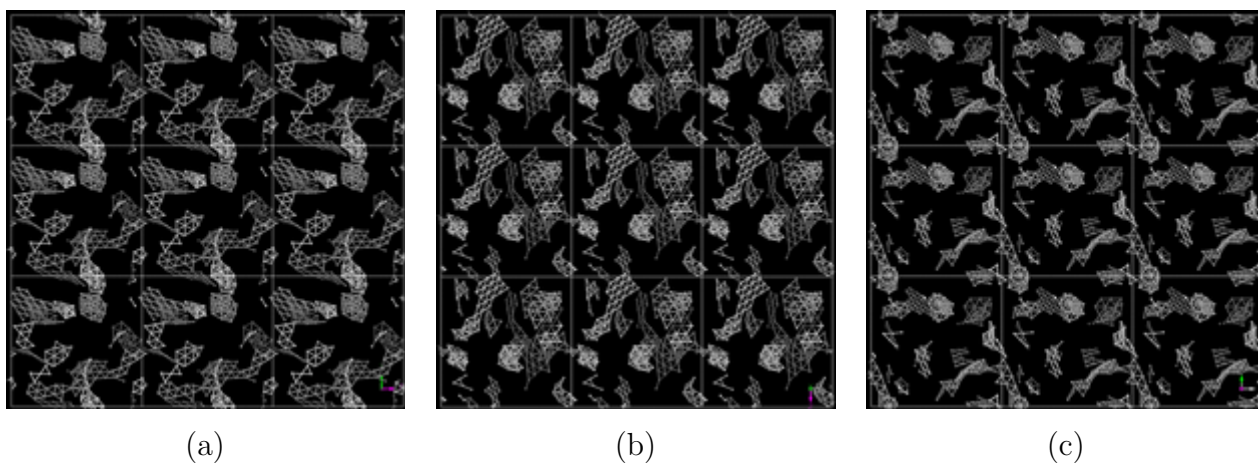
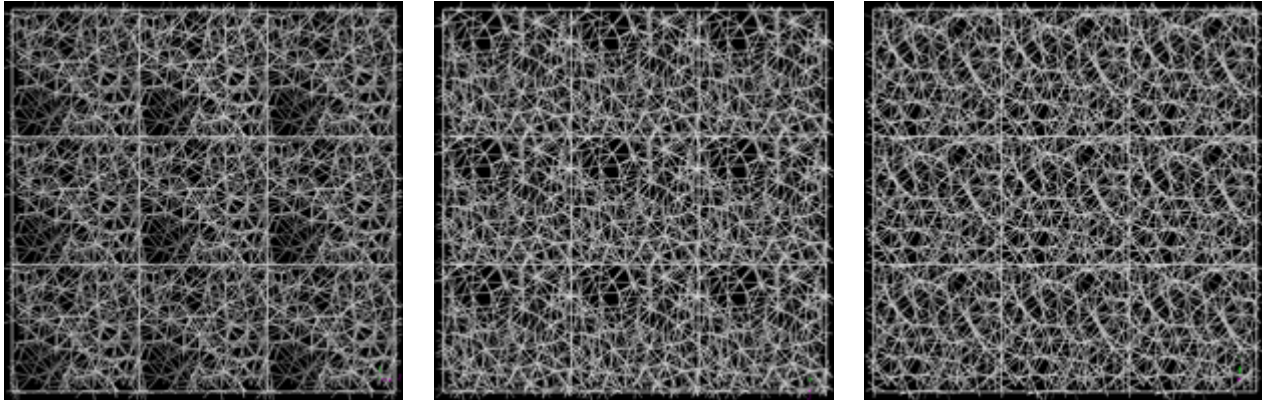


Figura 4.57 Estructura topológica del sistema eAg90, número de átomos 256, densidad 1.04898 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

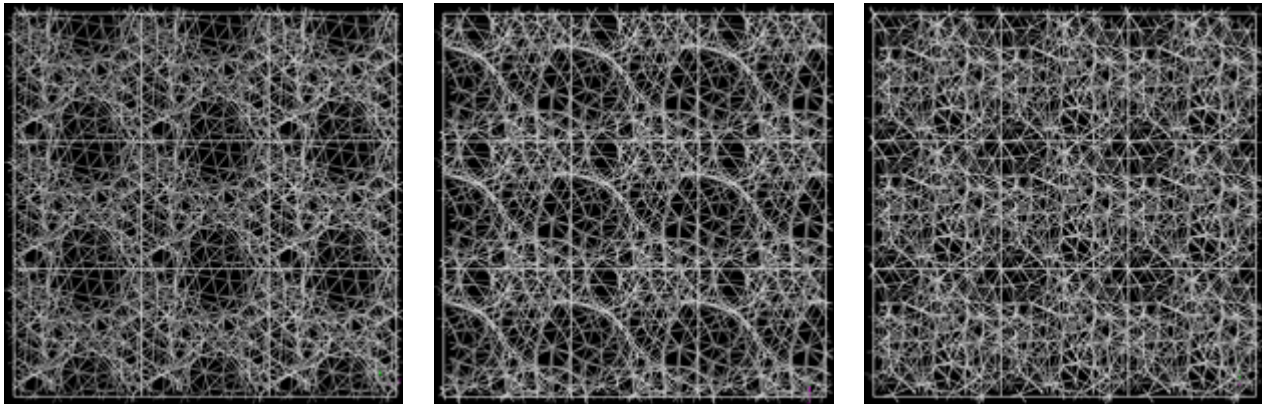


(a)

(b)

(c)

Figura 4.58 Estructura topológica del sistema eAu20, número de átomos 256, densidad 15.4397 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.



(a)

(b)

(c)

Figura 4.59 Estructura topológica del sistema eAu30, número de átomos 256, densidad 13.4995 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

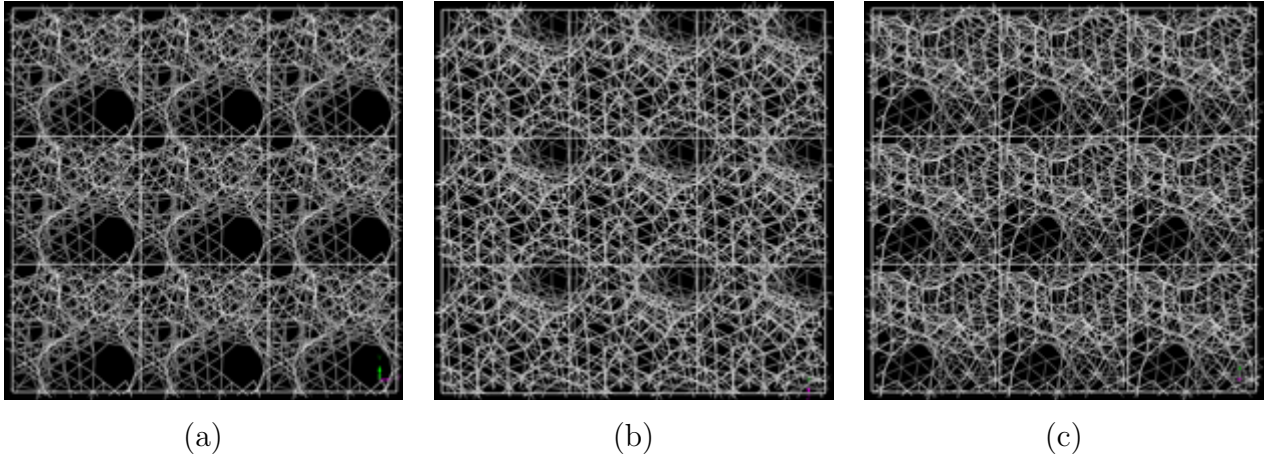


Figura 4.60 Estructura topológica del sistema eAu40, número de átomos 256, densidad 11.4997 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

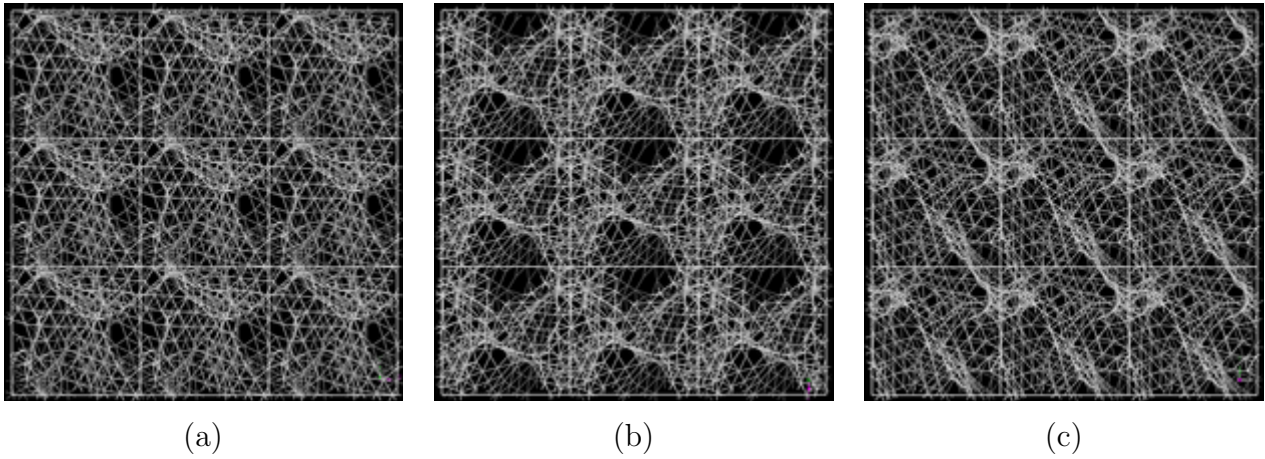
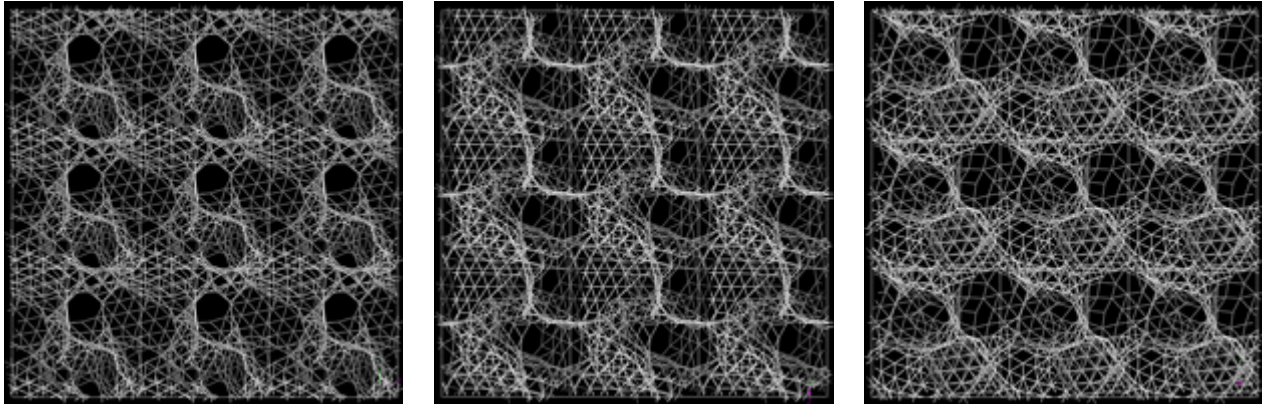


Figura 4.61 Estructura topológica del sistema eAu50, número de átomos 256, densidad 9.59974 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

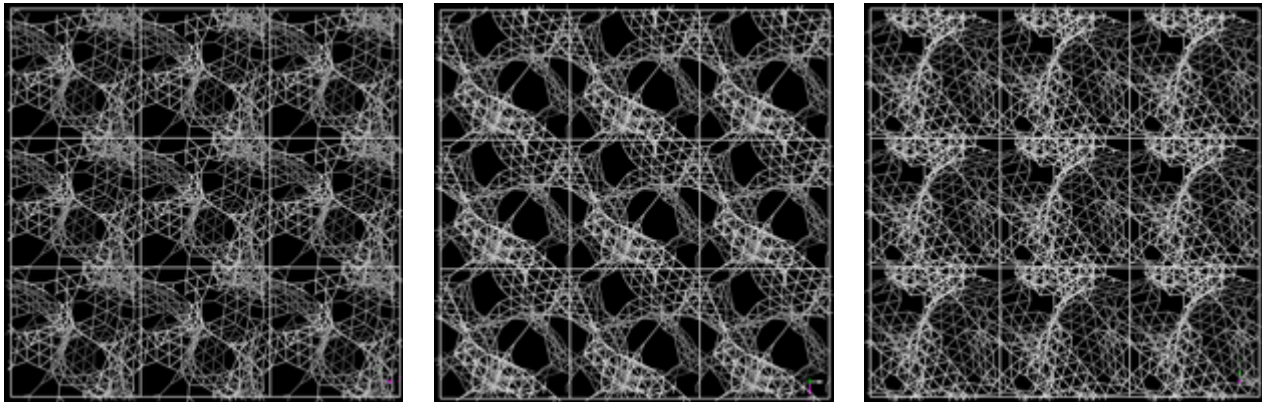


(a)

(b)

(c)

Figura 4.62 Estructura topológica del sistema eAu60, número de átomos 256, densidad 7.69976 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.



(a)

(b)

(c)

Figura 4.63 Estructura topológica del sistema eAu70, número de átomos 256, densidad 5.69982 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

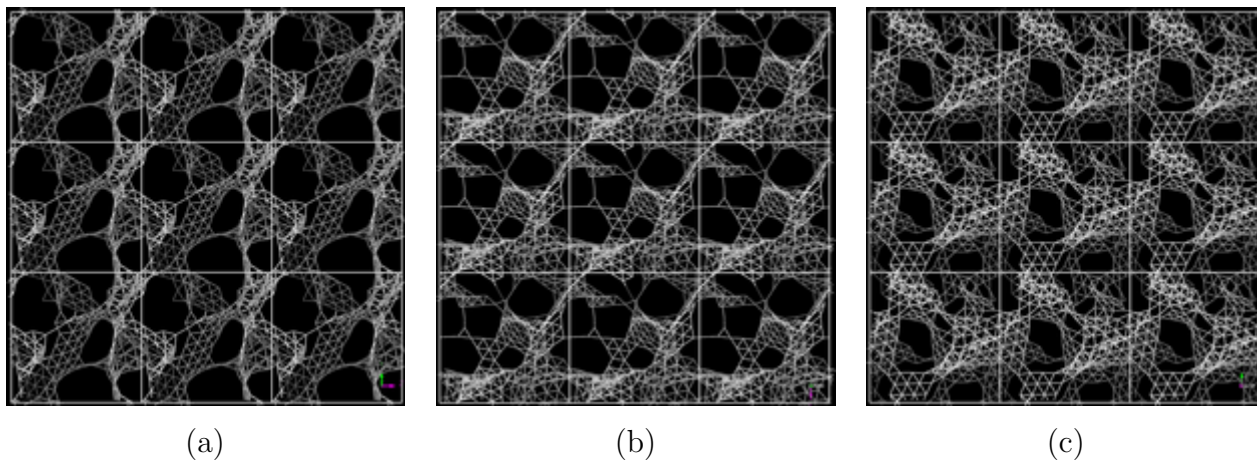


Figura 4.64 Estructura topológica del sistema eAu80, número de átomos 256, densidad 3.79992 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

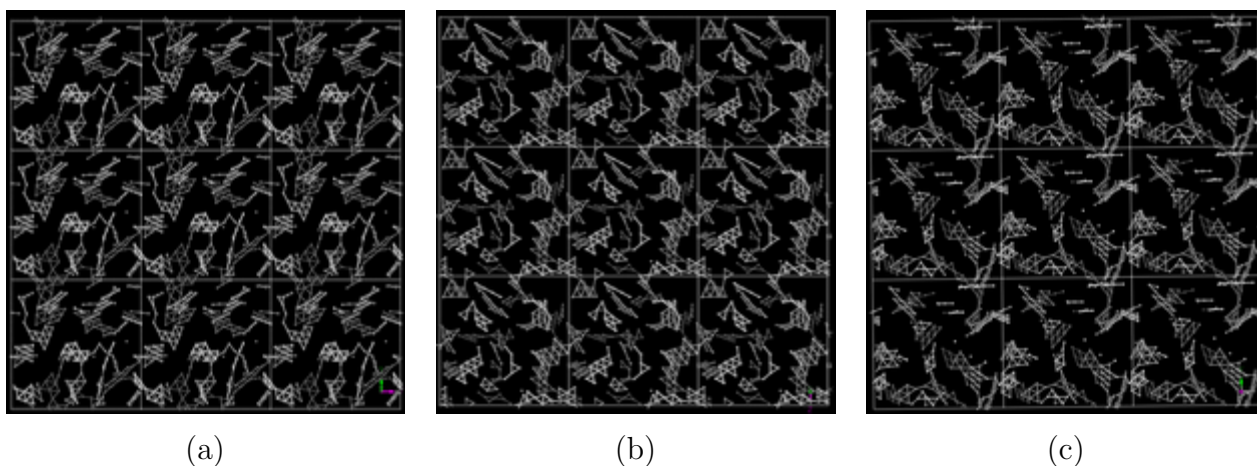


Figura 4.65 Estructura topológica del sistema eAu90, número de átomos 256, densidad 1.89995 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

4.4 DENSIDAD DE ESTADOS ELECTRÓNICOS

En esta sección se presentan las densidades de estados electrónicos, DEE, para los sistemas estudiados. La configuración electrónica de los tres metales estudiados es Cu:[Ar]3d¹⁰4s¹; Ag:[Kr]4d¹⁰5s¹ y Au:[Xe]4f¹⁴5d¹⁰6s¹. Existen otras combinaciones en la forma de representar la configuración electrónica pero nosotros hemos decidido utilizar ésta por los resultados que hemos obtenido en la DEE; la banda d completamente llena. Lo relevante es que las bandas electrónicas más externas son del tipo d y s, para la banda d el número de estados por átomo es 5, mientras que para la banda s es 2, así se observa que la densidad de estados es mucho más grande en la banda d que en la banda s.^[73] La **figura 4.66** esquematiza las dos bandas, d y s, más externas para estos metales.

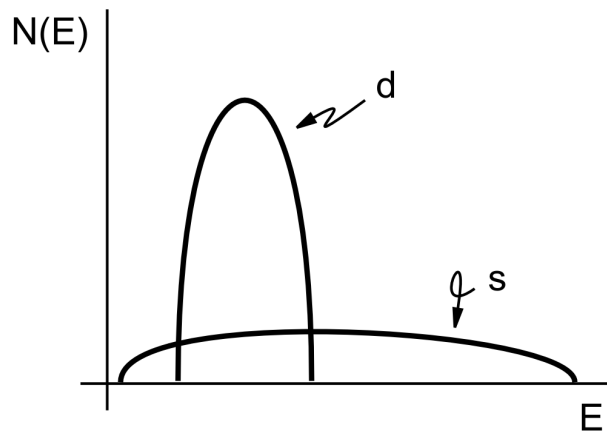


Figura 4.66 Forma esquemática de la densidad de estados electrónicos, $N(E)$, para la banda d y s del Cu, Ag y Au.

En el caso de estos tres metales, la banda d está completamente llena y la banda s parcialmente llena, como se observa en las **figuras 4.67**, **4.68** y **4.69**. En estas mismas figuras se indica con una línea negra vertical el nivel de Fermi; que en la mayoría de los casos indica claramente el llenado parcial de la banda s y el llenado completo de la banda d. No observamos algún comportamiento extraño en las DEE de todos los metales; a excepción de aquellos sistemas con alta porosidad, 80 y 90%. Es decir, el llenado parcial de la banda s es un comportamiento constante y característicamente metálico, el nivel de porosidad no parece afectar, en primera instancia, el comportamiento metálico de los sistemas. ¿Pero qué es lo que sucede cuando hay altas porosidades (80 y 90%)? la formación de las bandas es una característica de los sólidos; y por extensión, sólidos continuos. Lo que observamos

con estas porosidades son cúmulos aislados, o casi aislados (ver sistemas dCu80 y eCu80), formados por unos cuantos átomos. Esta *discontinuidad* debe modificar la estructura de bandas.

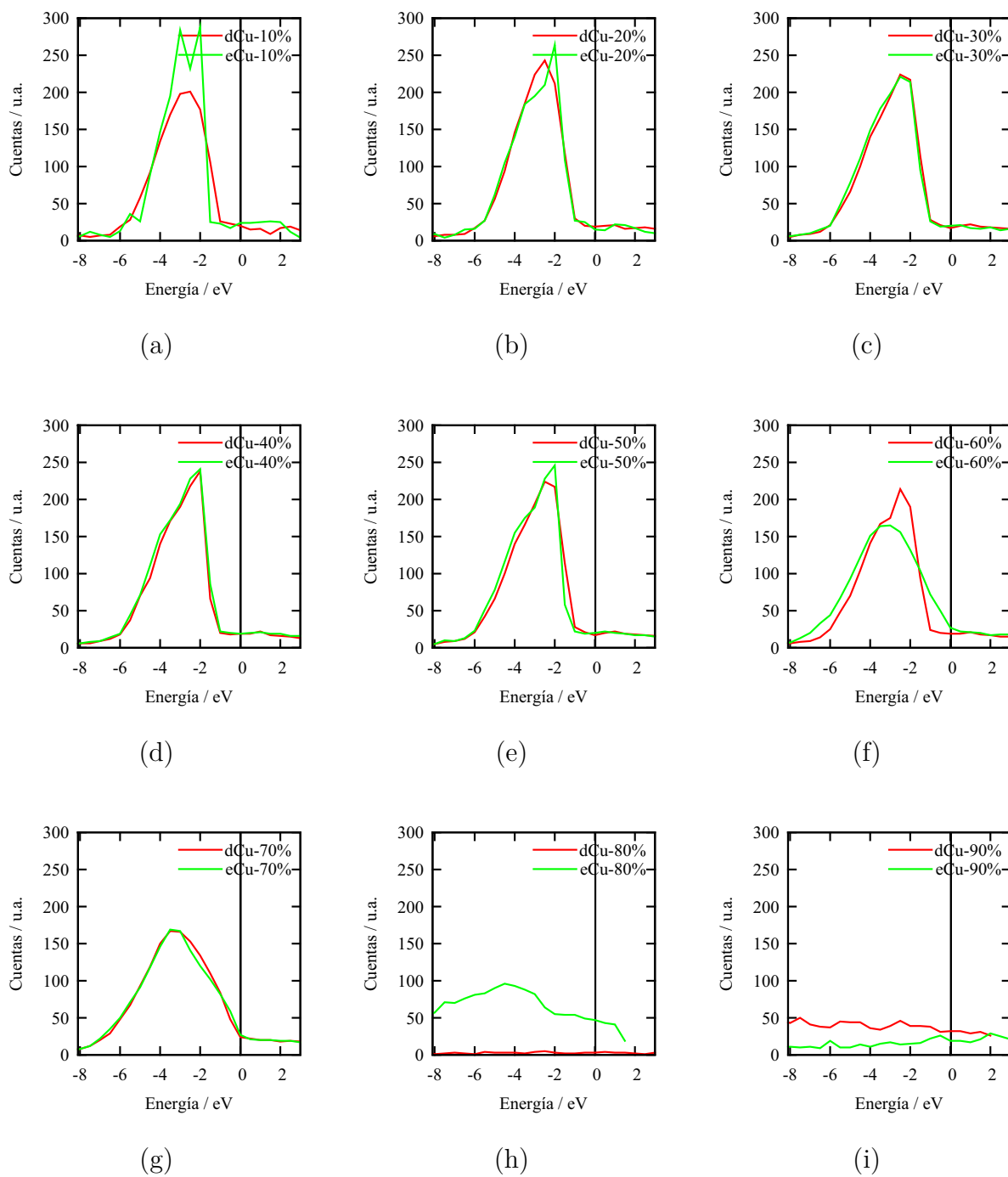


Figura 4.67 Densidad de estados electrónicos para el sistema Cu (a) dCu10 y eCu10, (b) dCu20 y eCu20, (c) dCu30 y eCu30, (d) dCu40 y eCu40, (e) dCu50 y eCu50, (f) dCu60 y eCu60, (g) dCu70 y eCu70, (h) dCu80 y eCu80, (i) dCu90 y eCu90.

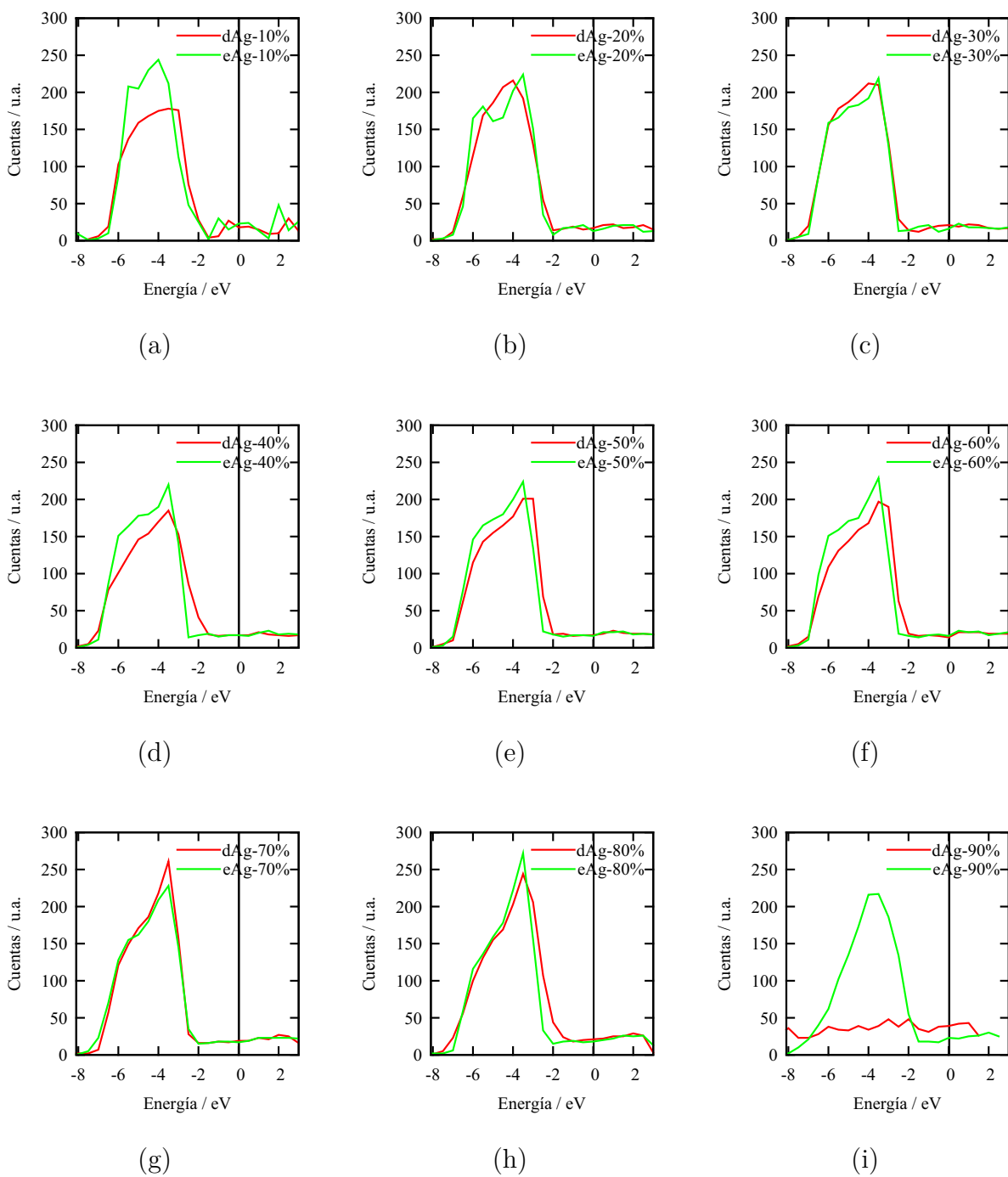


Figura 4.68 Densidad de estados electrónicos para el sistema Ag (a) dAg10, (b) dAg20 y eAg20, (c) dAg30 y eAg30, (d) dAg40 y eAg40, (e) dAg50 y eAg50, (f) dAg60 y eAg60, (g) dAg70 y eAg70, (h) dAg80 y eAg80, (i) dAg90 y eAg90.

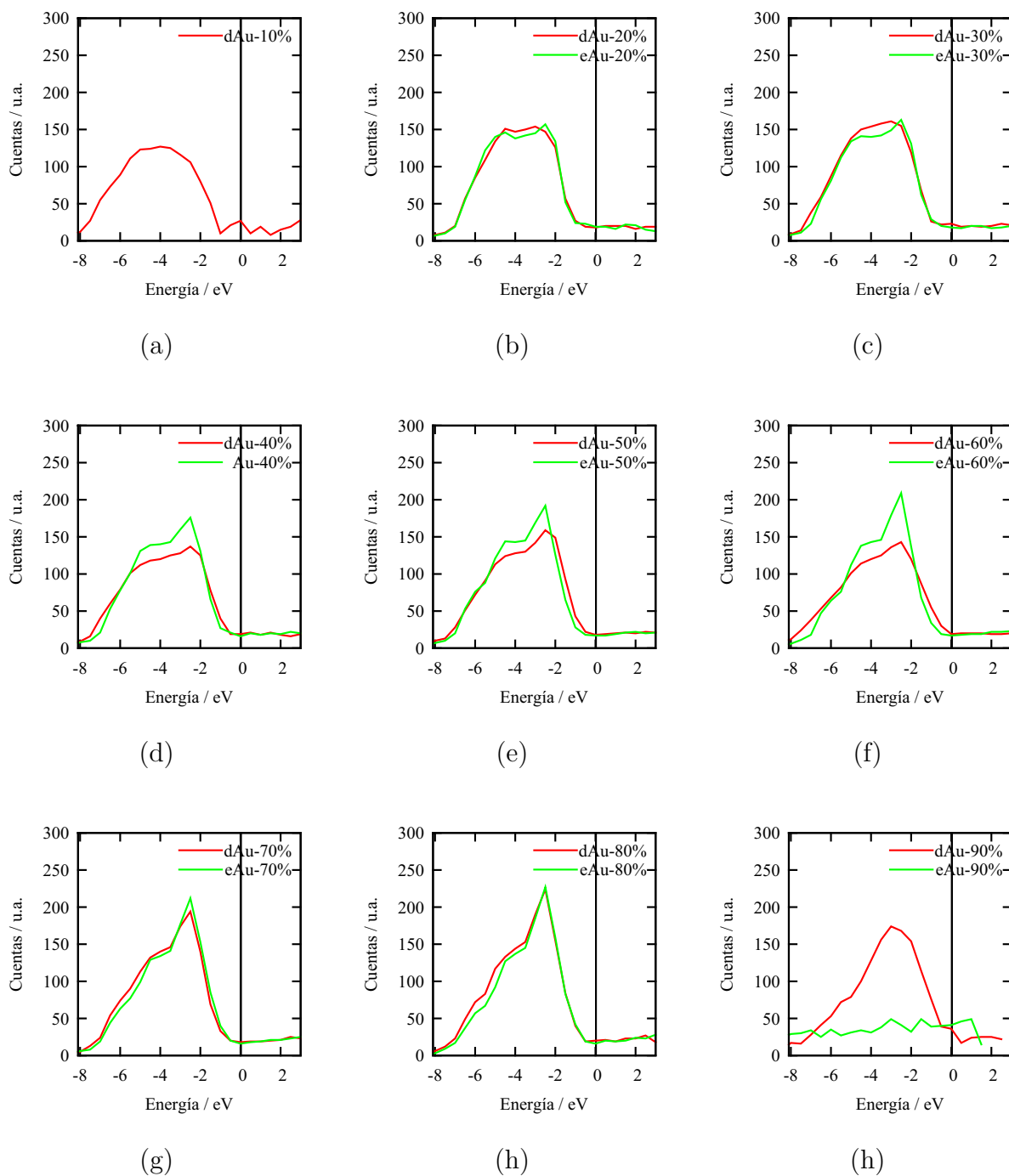


Figura 4.69 Densidad de estados electrónicos para el sistema Au (a) dAu10 y eAu10, (b) dAu20 y eAu20, (c) dAu30 y eAu30, (d) dAu40 y eAu40, (e) dAu50 y eAu50, (f) dAu60 y eAu60, (g) dAu70 y eAu70, (h) dAu80 y eAu80, (i) dAu90 y eAu90.

4.5 DENSIDAD DE ESTADOS VIBRACIONALES

Computacionalmente el cálculo de densidad de estados vibracionales, DEV, es muy costoso, por ello hemos elegido sólo aquellas estructuras que, consideramos son más representativas: estructuras desaleadas y expandidas con porosidades de 30, 50 y 70%. El tamaño de paso usado en las diferencias finitas para la evaluación el Hessiano fue de 0.01 bohr. Las DEV se muestran en las **figuas 4.70, 4.71 y 4.72**. La primera característica general a notar es la presencia de frecuencias vibracionales negativas, pero, no es raro encontrar frecuencias negativas en los sistemas amorfos simulados, debido a que pueden existir inestabilidades dinámicas. Aunque, esto es un problema de la simulación computacional, pues experimentalmente no puede concebirse el cálculo de una frecuencia negativa. La segunda característica general es la contracción del espectro hacia bajas frecuencias al aumentar la masa atómica, lo que es de esperarse pues la masa está directamente relacionada con la fuerza. No se observan diferencias significativas entre los espectros calculados para los sistemas desaleados y expandidos, a excepción del sistema Au30. En estos sistema, Au30, no son sistemas equivalentes pues el sistema desaleado indica una estructura cuasi-cristalina y el expandido una estructura amorfa, ver **figura 4.3c**. El hecho de que durante la dinámica molecular los sistemas base Cu hayan tenido mayor movimiento y los sistemas base Au, menor; esto es, la evolución de la estructura sea menor para los sistemas Au; puede indicar el porqué se observa mayor estructura en los DEV.

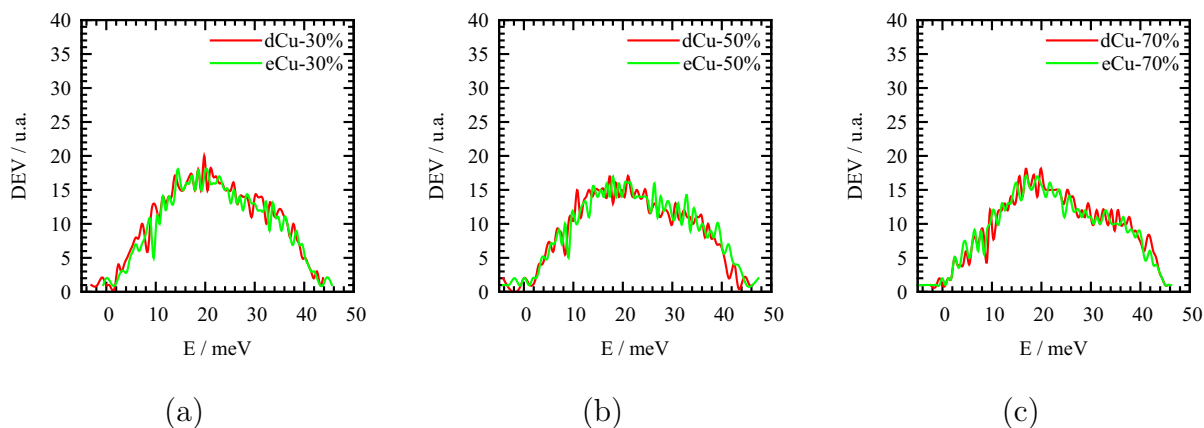


Figura 4.70 Densidad de estados vibracionales para el sistema Cu (a) dCu30 y cCu30 (b) dCu50 y cCu50, (c) dCu70 y cCu70.

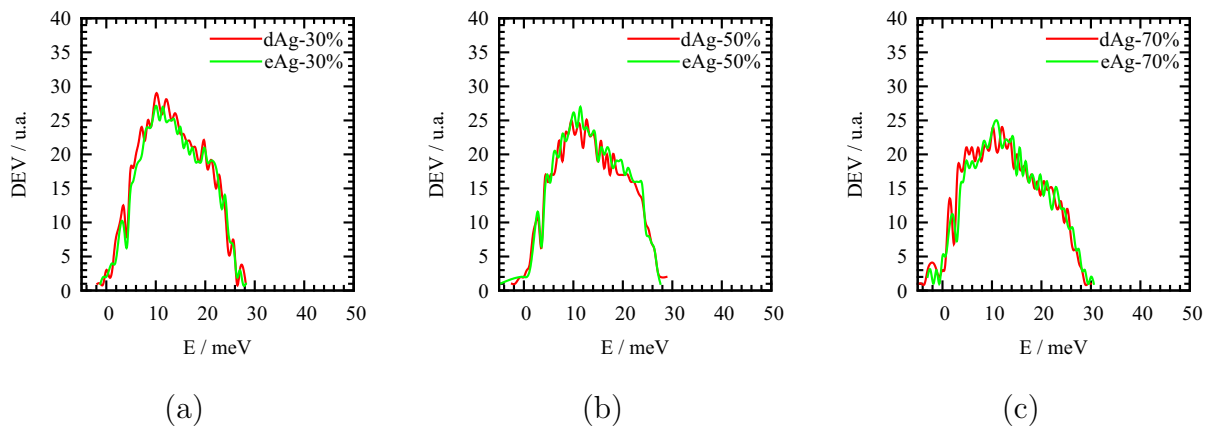


Figura 4.71 Densidad de estados vibracionales para el sistema Ag (a) dAg_{30} y cAg_{30} (b) dAg_{50} y cAg_{50} , (c) dAg_{70} y cAg_{70} .

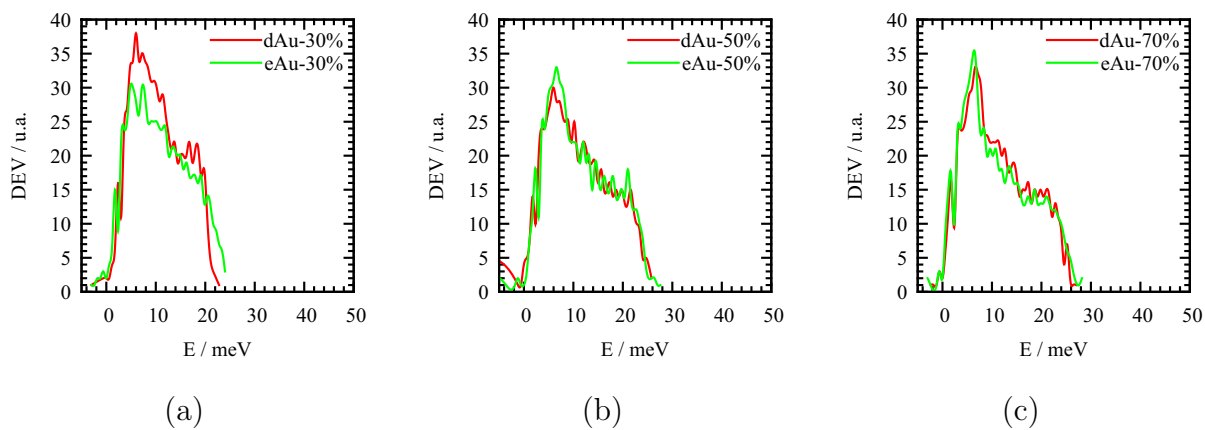


Figura 4.72 Densidad de estados vibracionales para el sistema Au (a) dAu_{30} y cAu_{30} (b) dAu_{50} y cAu_{50} , (c) dAu_{70} y cAu_{70} .

4.6 FUNCIONES DE FUKUI

Nosotros empleamos la teoría de funciones de Fukui, FF, para evaluar cualitativamente la reactividad local en algunas de nuestras muestras. Esto es, empleamos las FFs en los sistemas Cu, Ag y Au para las porosidades de 30, 50 y 70 %. Hemos elegido estos sistemas porque consideramos que son los que mejor representan este trabajo. Los índices de Fukui o FF son evaluadas de forma volumétrica, es decir, las FFs son evaluadas punto a punto a partir de la densidad de carga electrónica que tiene un valor para cada punto en el espacio. Esta característica *volumétrica* hace un tanto complicado representar los valores de las funciones en dos e incluso tres dimensiones; y aunque pueden utilizarse curvas de nivel en dos dimensiones e isosuperficies en tres dimensiones, es necesario gran cantidad de imágenes para tener una apreciación completa de los resultados. En esta sección presentaremos sólo algunas imágenes representativas de entre todas las posibles de todos los sistemas y nos limitaremos a describir los resultados que hemos observado. Pero, ¿cómo se visualizan, entonces, los cálculos hechos empleando las FFs? Es una imagen semejante a la presentada para la densidad de carga, es decir, una estructura envuelta por una «nube» formada punto a punto y en el que cada uno de ellos es un valor del índice de Fukui como función de la posición.

La siguiente figura, **figura 4.73**, muestra los valores de la FF vistos desde el plano xz del sistema eAg30. La figura realmente es la composición de dos imágenes, la parte derecha de la imagen muestra la estructura topológica simple, mientras que la parte izquierda es la estructura más la nube de la FF, $f^0(\mathbf{r})$. Mostramos $f^0(\mathbf{r})$ y no $f^+(\mathbf{r})$ o $f^-(\mathbf{r})$ porque su comportamiento es similar y representa un comportamiento promedio; cuando $f^0(\mathbf{r})$ no sea representativa de $f^+(\mathbf{r})$ o $f^-(\mathbf{r})$ se indicará. Como consecuencia del enunciado anterior, tanto la reactividad nucleofílica⁵⁷ como la electrofílica⁵⁸ están presentes en los mismos lugares dentro de la estructura; tal como se verá más adelante esto es característico de todas nuestras estructuras. Y como cabría esperar, los valores más altos para la FF se encuentran sobre las superficies; en este caso, las superficies que delimitan el poro. Estos valores máximos se mantiene prácticamente constantes en toda la superficie, sólo existen pequeñas variaciones en la reactividad de un punto a otro sobre la misma.

⁵⁷ Nucleofílica: una especie que al reaccionar, cede electrones.

⁵⁸ Electrofílica: una especie que al reaccionar, acepta electrones.

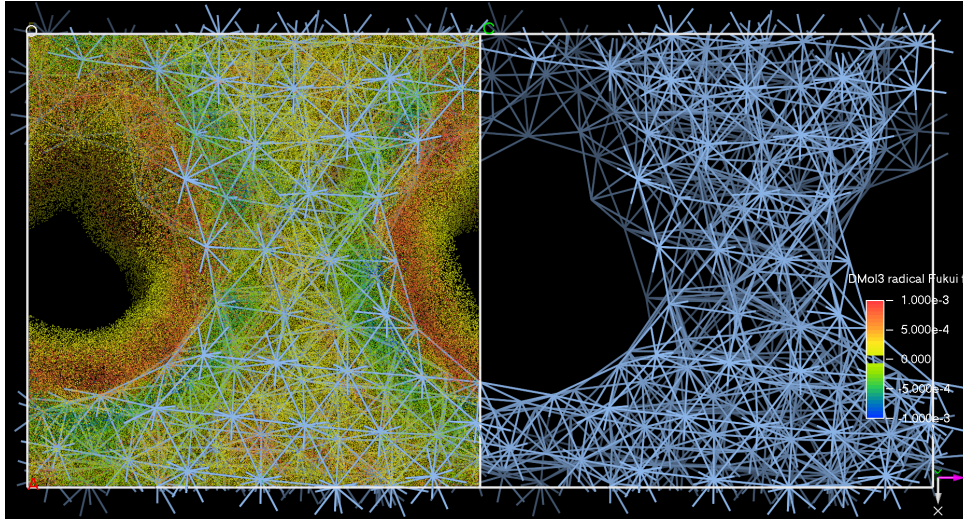


Figura 4.73 Plano xz del sistema eAg30; el lado izquierdo de la figura muestra la estructura topológica más la nube de la FF, $f^0(\mathbf{r})$. El lado derecho, y solo para fines comparativos, muestra la estructura topológica simple.

Ahora bien, en este sistema pueden observarse valores **negativos** en la FF. Existe una tendencia en su ubicación, los valores negativos se localizan del lado opuesto a los valores positivos, es decir, si sobre la superficie del poro se encuentran los valores positivos, justo por debajo de ésta se encontrarán los valores negativos. Este comportamiento lo hemos observado en todas nuestras muestras. La **figura 4.74** muestra dos cortes transversales justo a la mitad de la celda y paralelos al plano xz y yz. En los cortes se pueden apreciar las regiones positivas sobre la superficie del poro y negativas por debajo de ella. Otra característica relevante es que el valor negativo de la función se encuentra siempre justo por debajo de la superficie, esto es, alejado de la superficie, hacia el interior del *backbone* la función vuelve a ser positiva. La región positiva por encima de la superficie se extiende en promedio 3 Å para la plata, lo mismo que la región negativa por debajo de la superficie.

El problema de fondo en estos resultados se resume en dos simples preguntas: ¿tiene sentido un valor negativo en la FF? y si lo tiene, ¿qué significa que esté presente en nuestras muestras? Si se considera siempre la densidad electrónica como una función positiva, dada la implementación con diferencias finitas de las FFs

$$f^+(r) = \frac{1}{\Delta N} (\rho_{N+\Delta N}(r) - \rho_N(r)) \quad (4.2)$$

$$f^-(r) = \frac{1}{\Delta N} (\rho_N(r) - \rho_{N-\Delta N}(r)) \quad (4.3)$$

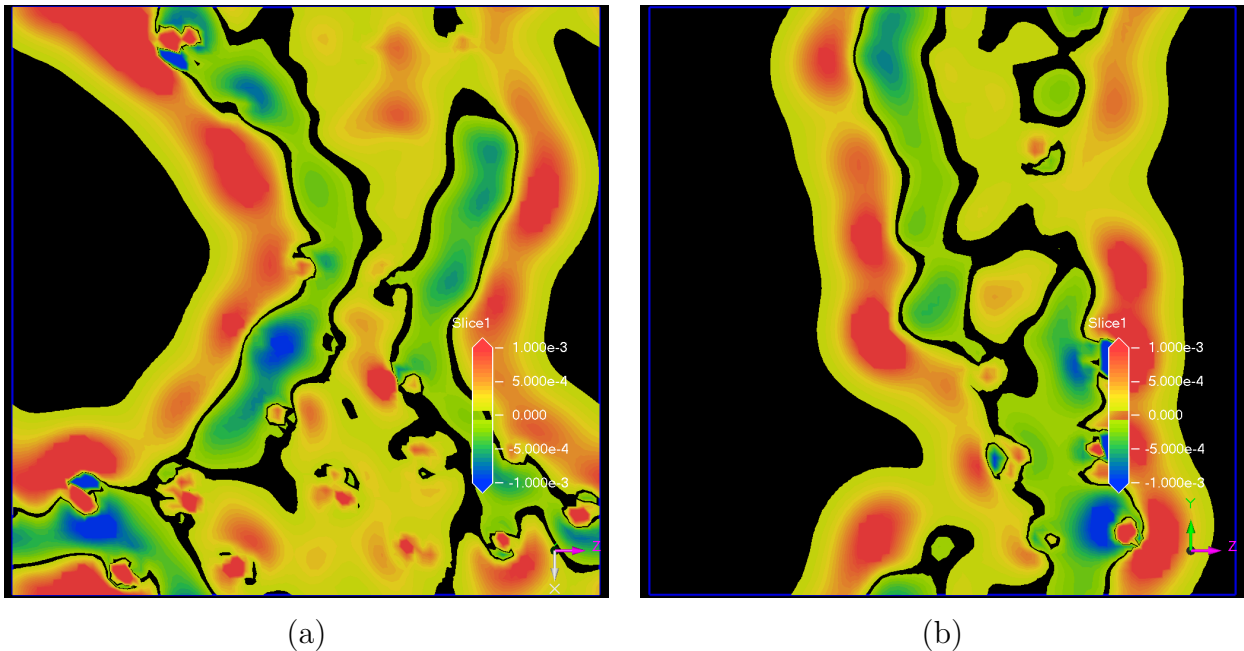


Figura 4.74 Cortes transversales a $f^0(\mathbf{r})$ en los planos xz y yz del sistema eAg30. Puede apreciarse el valor positivo de $f^0(\mathbf{r})$ sobre la superficie del poro y negativo por debajo de ella. También, puede observarse las regiones negativas al rededor de los núcleos atómicos.

o bien la definición de la función condensada de Fukui, cFF .

$$f_k^+ = q_k^{\text{anión}} - q_k \quad (4.4)$$

$$f_k^- = q_k - q_k^{\text{catión}} \quad (4.5)$$

(donde q es el valor de la carga asignada a un átomo en la molécula mediante el análisis de Mülliken o Hirshfeld) el valor de la FF siempre debería ser positiva. Intuitivamente el valor negativo en la FF, en un contexto molecular, significa que su carga formal (estado de oxidación) incrementa cuando un electrón se adiciona a la molécula,^[75] —un átomo de la molécula se oxida (reduce) cuando la molécula en general se reduce (oxida)^[76]— o en otras palabras, significa que cuando se agrega un electrón a la molécula, en algunos puntos, la densidad electrónica se reduce y cuando se remueve un electron de la molécula, en algunos puntos la densidad electrónica crece.^[77] Por supuesto esto va en contra de la intuición. En el contexto de sólidos que, cuando una especie cargada negativamente se acerca a la superficie, se crea una concentración electrónica (negativa) en la región por donde la carga se aproxima. Y a pesar de esto, la existencia de valores negativos en el cálculo de las FF, no se ha descartado, al menos no matemáticamente.^[77]

Muchos de los análisis a moléculas reportados utilizan la cFF más que la FF debido a que se prefiere asociar propiedades como el carácter iónico, la polaridad, la reactividad química, entre otras, a los átomos dentro de la molécula más que a un punto en el espacio. Y para realizar esta asignación se emplea lo que se conoce como análisis de población electrónica, APE. Y aunque, considerado artificial, pues no se puede determinar experimentalmente la carga de un átomo que forma una molécula, existen varios métodos para realizar un APE. Tres de los más representativos son el análisis de Mülliken,^[78] el análisis de Hirshfeld^[79] y el análisis de Singh–Kollman.^[80]

Con estos análisis se han dado argumentos del porqué las FFs pueden resultar en valores negativos. Ya que el análisis de Mülliken trabaja con funciones de onda y el de Hirshfeld con densidad de carga se cree que Hirshfeld sólo puede dar valores positivos, sin embargo un estudio realizado por Bultinck *et al.*^[77] que consiste en deformar moléculas sencillas como la del agua y calcular el valor de la FF, afirma que también, con este análisis es posible encontrar valores negativos. Ellos establecen la presencia de valores negativos cuando la distancia de enlace disminuye un 35%, y cuando el ángulo se modifica drásticamente;^[81]

Otro argumento dado para la presencia de los valores negativos es el tamaño de la ΔN cuando se emplean diferencias finitas. Se recomienda usar valores fraccionados para evitar esto, por ejemplo, 0.1. En suma, varios autores^[76] han argumentado, basados en consideraciones conceptuales, que la existencia de valores negativos en la FF son producto del tipo de análisis de población que se emplea y no algo inherente a la estructura analizada, a menos que, como se dijo en el párrafo anterior, la estructura sea altamente deformada o comprimida; que no es nuestro caso.

En el 2006 P. Ayers^[76] publicó un artículo exponiendo las condiciones necesarias y suficientes para la existencia de valores negativos en las FF. Algunos de sus razonamientos llegan a afirmar que es «natural» que en las regiones próximas al núcleo atómico la FF sea negativa. Por ejemplo, cuando se agrega carga negativa a un átomo, la densidad electrónica, al menos cerca del núcleo, disminuirá, lo que implica un negativo en $f^+(\mathbf{r})$; justo como sucede cuando se crea un anión, el radio aniónico es mayor que su equivalente atómico. Lo mismo en la creación de un catión, la densidad electrónica cerca del núcleo aumenta lo que implica una $f^-(\mathbf{r})$ negativa. Este comportamiento lo observamos en todas nuestras muestras, la **figura 4.75** muestra tres cortes de la FF que atraviesan el algún átomo. Puede observarse el valor negativo (en azul) alrededor de los núcleos.

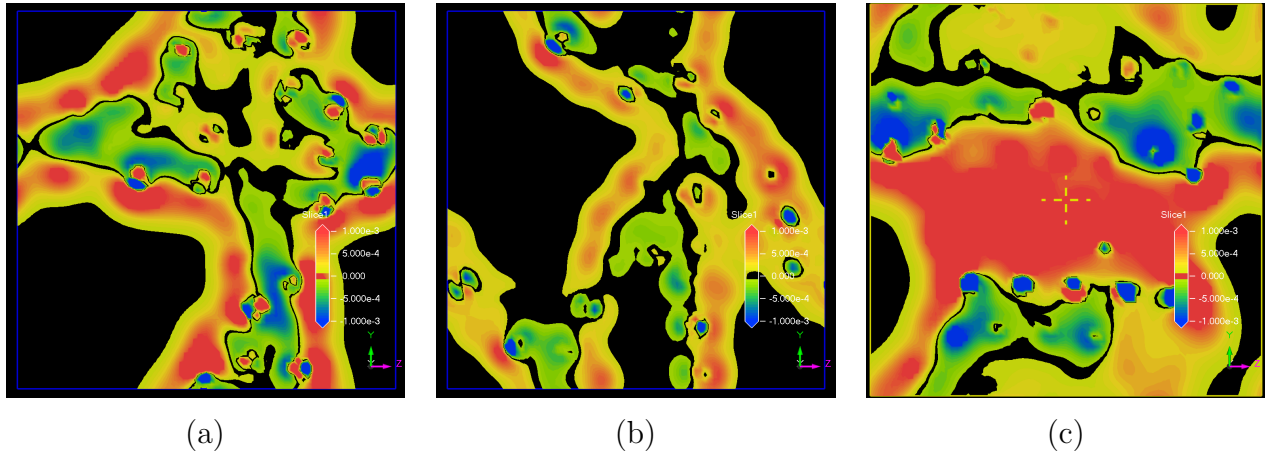


Figura 4.75 Cortes paralelos al plano yz de la FF que atraviesan el centro de al menos un átomo de la muestra, todos ellos presentan valores negativos alrededor de cada núcleo atómico. (a) Sistema eAg50, (b) Sistema eAu50, (c) Sistema eCu50.

De cualquier forma, nosotros no hemos podido adaptar ninguna de estas explicaciones a nuestros sistemas y tampoco hemos podido dar una respuesta razonable del porqué disminuye la densidad electrónica por debajo de la superficie del poro al extraer carga y porqué aumenta al adicionar carga; mientras que en la superficie del poro sucede lo contrario. La deformación de la red en nuestras muestras no parece ser lo suficientemente grande para generar valores negativos según lo presentado por Bultinck *et al.*;[77] ni es aplicable la transferencia HOMO–LUMO pues no podemos establecer una diferencia razonable entre dos átomos de la misma especie dentro de nuestras estructuras. Sin embargo, hay una diferencia esencial en los sistemas presentados por los autores anteriores y nuestros sistemas: los nuestros son sistemas periódicos. No sabemos de que forma esto pueda modificar el comportamiento de las FFs, pero al parecer tiene alguna influencia.

4.7 FRACTALES

¿Cómo medir la distribución de la oquedad dentro de una estructura porosa, es decir, la forma y distribución del espacio vacío dentro de una estructura? Es evidente que las estructuras porosas no llenan el espacio en tres dimensiones y que normalmente existe toda una distribución de poros u oquedades de diferentes formas y tamaños dispersos aparentemente de manera aleatoria dentro de la estructura. De manera natural esta forma de describir una estructura porosa hace pensar en el concepto de *fractal*, esto es, asociar

un valor de dimensión fractal a una estructura de este tipo. Utilizar la dimensión fractal para describir una estructura porosa no es nuevo, por ejemplo, Schaefer y Keefer en 1986^[82] y Kieffer y Angell en 1988^[83] la utilizan para describir y analizar estructuras altamente porosas de óxido de silicio. Comúnmente se piensa en estructuras fractales del tipo autosemejantes —aquellas en las que un patrón se repite de forma continua al variar la escala— pero existe al menos otro tipo de fractales, los llamados: no autosemejantes. A grandes razgos, en este tipo de fractales al variar la escala hay un entorno parecido pero no existe patrón que se repita de forma sistemática. En los fractales, en cuanto a objetos matemáticos, la variación de escala puede ser infinita porque el fractal en sí mismo tiene estructura al infinito. Ahora bien, en la naturaleza existen muchos objetos que se han clasificado como fractales, por ejemplo, las costas del mar, las galaxias, el brocoli, los ríos, los caracoles, las montañas, etcétera; sin embargo, los fractales de la naturaleza no pueden tener un resescalamiento al infinito, y aunque quizá esto no tenga relevancia práctica, digamos, al calcular la dimensión fractal de una costa, si lo tiene cuando se trata con estructuras fractales cerca de una escala atómica. En otras palabras, la disminución en la escala de una estructura fractal de la naturaleza necesariamente llevará al límite atómico en el que no es posible disminuir la escala y mantener la estructura fractal. Quizá una forma más práctica y aplicable de pensar en la dimensión fractal sea como un índice que indica el grado en el que una estructura llena el espacio (el espacio en una, dos o tres dimensiones). Este índice, en consecuencia, podría discriminar dos estructura con la misma cantidad de átomos y la misma densidad pero cuya estructura llena (se distribuye) de forma diferente en el espacio.

A las estructuras que nosotros hemos generado al ser estructuras porosas que no llenan todo el espacio y cuyas formas son un tanto indefinidas, es fácil aplicarles el concepto de dimensión fractal. Nosotros no podemos asegurar, a menos no con tan pocos átomos que nuestras estructuras sean fractales pero sí podemos aplicar el concepto de dimensión fractal para caracterizarlas. Empleamos directamente el método utilizado por Schaefer y Keefer ^[82] en cada una de nuestras muestra, obtenemos los resultados mostrados en las gráficas de la **figura 4.76**. Este método está relacionado directamente con el cálculo de la dimensión de Minkowski-Bouligand^[84] (o de conteo de cajas⁵⁹). El cálculo de la dimensión Minkowski-Bouligand consiste en recubrir un fractal con cajas cuya arista sea de tamaño ϵ , y se cuenta el número de cajas que cubren la estructura fractal. El recubrimiento se

⁵⁹ Cajas, esferas, cubos, círculos, ...el tipo de figura con la que se recubre no es importante.

hace de forma iterativa con cajas de ϵ cada vez más pequeña, en el límite donde ϵ tiende a cero, se calcula la dimensión fractal. Si $N(\epsilon)$ es el número de cajas que cubren al fractal entonces la dimensión de Minkowski-Bouligand será:

$$\dim = \lim_{\epsilon \rightarrow 0} \frac{N(\epsilon)}{1/\epsilon} \quad (4.6)$$

Es decir, la pendiente en la parte más recta de la gráfica $N(\epsilon)$ vs. $1/\epsilon$ será la dimensión fractal. Kieffer y Angell grafican en log-log el número de iones $N(r)$ que pueden ser encontrados dentro de una esfera de radio r centrada en un punto arbitrario. Y calculan la pendiente.

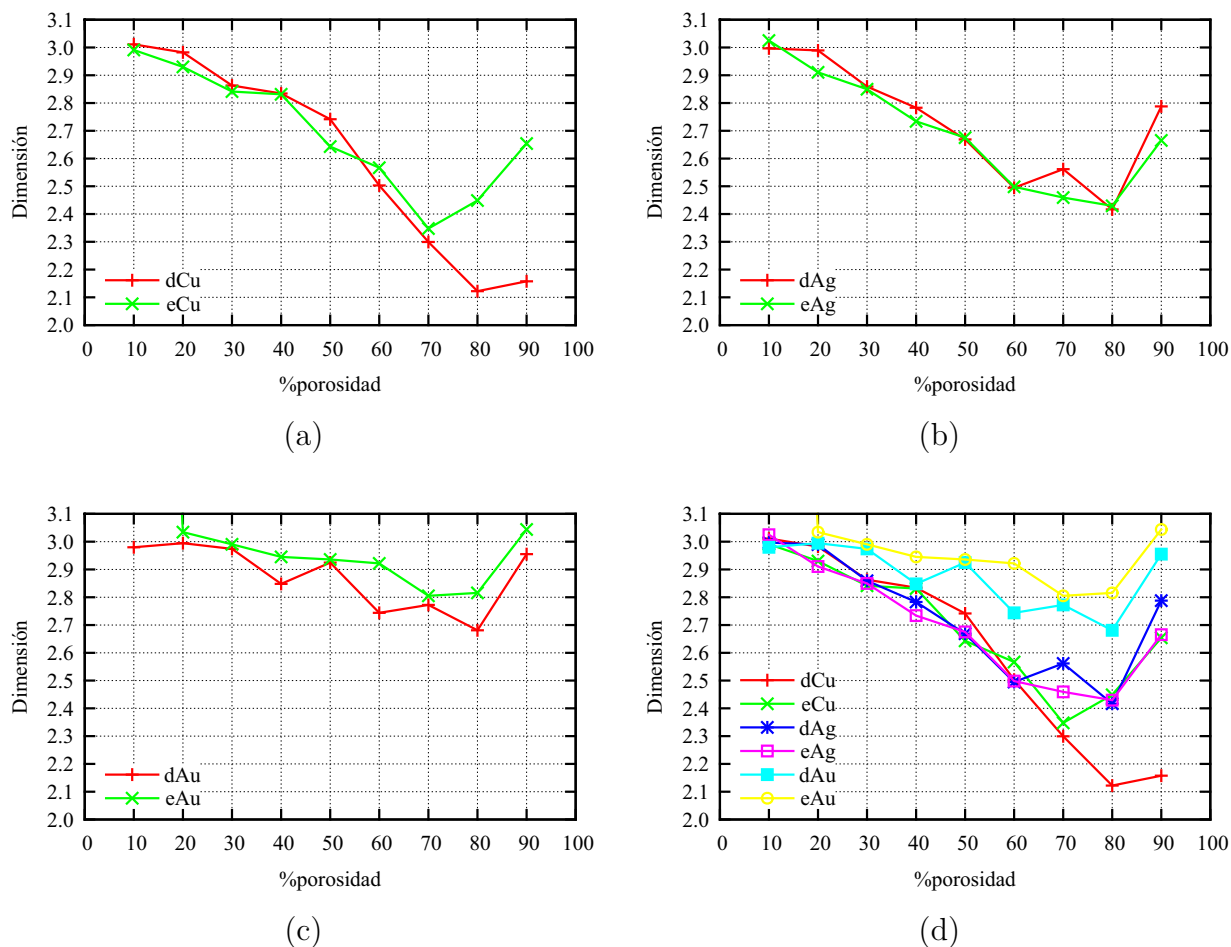


Figura 4.76 Cálculo de la dimensión fractal para las estructuras (a) Cu (b) Ag (c) Au (d) Comparación de la dimensión fractal para Cu, Ag y Au.

Tabla 4.6 Valores de la dimensión fractal calculados para los sistemas simulados

%porosidad	Cu		Ag		Au	
	Desaleado	Expandido	Desaleado	Expandido	Desaleado	Expandido
10	3.0111	2.9909	2.9969	3.0250	2.9799	---
20	2.9822	2.9300	2.9893	2.9104	2.9945	3.0342
30	2.8631	2.8412	2.8592	2.8497	2.9741	2.9905
40	2.8343	2.8313	2.7828	2.7339	2.8480	2.9451
50	2.7415	2.6431	2.6688	2.6752	2.9247	2.9357
60	2.5031	2.5667	2.4948	2.4983	2.7438	2.9219
70	2.2996	2.3478	2.5617	2.4595	2.7724	2.8049
80	2.1221	2.4490	2.4169	2.4297	2.6811	2.8155
90	2.1577	2.6539	2.7876	2.6652	2.9550	3.0440

En las gráficas calculadas, **figura 4.76**, claramente se observa la tendencia de disminuir la dimensión al aumentar la porosidad. Los datos de estas figuras se presentan en la **tabla 4.6** Esto es fácilmente justificable: al aumentar la porosidad la «eficiencia» de llenado disminuye, es decir, el espacio se llena cada vez menos (de forma menos eficiente). Cada una de estas gráficas presenta los resultados tanto de las estructuras desaleadas como de las expandidas. La dimensión se encuentra entre 3 y 2, lo que también es de esperar. Puede observarse que el último en todas las gráficas no sigue esta tendencia y aumenta ligeramente su valor, esto realmente corresponde a un cambio evidente en todas las estructuras. Este último punto corresponde a la porosidad de 90%, con esta porosidad (en todas las muestras) no se crea una estructura continua, sino cúmulos, esto rompe con la tendencia de las estructuras creadas con porosidades menores. En otras palabras, las estructuras formadas con 90% de porosidad son estructuras diferentes, la forma de llenar el espacio es diferente (son fractales diferentes), ver **sección 4.3**.

Existe una característica más en los resultados del cálculo de la dimensión fractal: para una misma porosidad, y diferente elemento, la dimensión varía sistemáticamente de la siguiente manera: $d_{Cu} < d_{Ag} < d_{Au}$. ¿Cómo explicar esto? Hay que recordar que el *time step* es el mismo para los tres elementos, lo que se traduce en un mismo tiempo de simulación, pero al ser elementos de diferente peso atómico, la movilidad entre ellos es diferente, de hecho, la movilidad sigue el orden inverso dado arriba para la dimensión.

Así, si consideramos, estos tres metales como formados por esferas rígidas, podríamos suponer que la variación se debe a que, debido a la movilidad, las estructuras con átomos más pesados se encuentran en estados *intermedios*. Pero en este punto surge otra pregunta ¿en qué difieren las estructuras *intermedias* (Au y Ag) de la estructura más avanzada o *final*? La respuesta es la distribución de los átomos dentro de la estructura. Al comparar estructuras topológicas de la **sección 4.3** puede notarse que cuando los poros son grandes; es decir, hay grandes espacios vacíos dentro de la celda, la dimensión calculada es menor. Mientras que cuando la oquedad no está en una región, sino que se encuentra distribuída en la estructura, la dimensión es mayor. En otras palabras, cuando la estructura tiende a ocupar o distribuirse por todo el espacio (poros pequeños) la dimensión es alta (más cerca de 3); pero cuando los átomo no tratan de ocupar todo el espacio, sino que se concentran sólo en ciertas regiones, la dimensión es baja (tiende a 2). El caso modelo de esto es para las estructuras eAu70 y eCu70, **figuras 4.63** y **4.46**.

Para reforzar esta afirmación, utilizamos dos dinámicas realizadas por Cristina Romero Rangel, miembro del grupo de trabajo. Ella realiza dos dinámicas moleculares empleando C, partiendo de dos estructuras expandidas diferentes: grafito y diamante; ambas estructuras tienen la misma densidad (misma porosidad) de 1.38 g/cm, la diferencia entre ellas es únicamente la estructura inicial. Después de 300 pasos de simulación a temperatura 1000K se obtienen las estructuras mostradas en las **figuras 4.77** y **4.78**. En la primera, la de grafito, la oquedad se encuentra concentrada principalmente en ciertas regiones (poros grandes) formando capas y en la segunda, la de diamante, la oquedad está distribuída en toda la estructura. Al calcular la dimensión fractal, lo que obtenemos $d_{\text{grafito}}=1.571$ y $d_{\text{diamante}}=2.0202$, confirmando de esta manera lo dicho arriba.

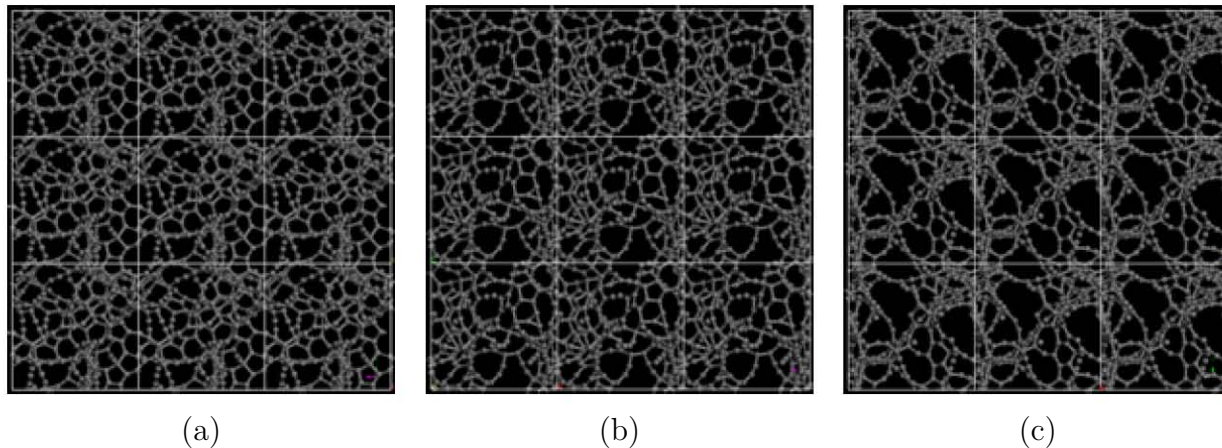


Figura 4.77 Estructura topológica del sistema pC, después de la dinámica molecular partiendo de una estructura expandida de diamante, densidad 1.38 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

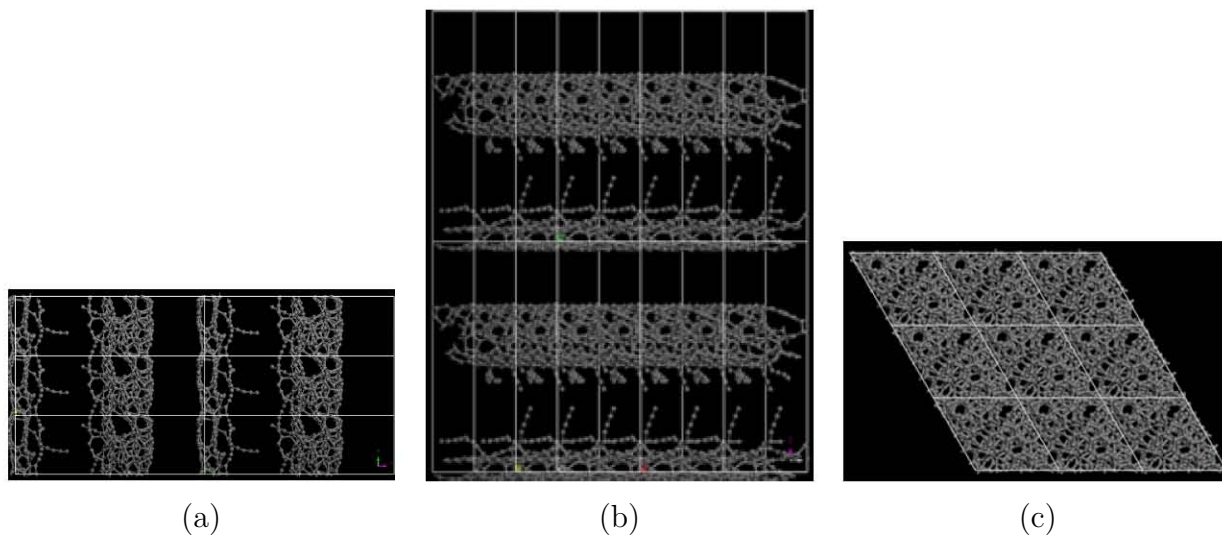


Figura 4.78 Estructura topológica del sistema pC, después de la dinámica molecular partiendo de una estructura expandida de grafito, densidad 1.38 g/cm^3 ; (a) dirección x, (b) dirección y, (c) dirección z.

5 Conclusiones

- A la temperatura utilizada los dos métodos de generación de superceldas son equivalentes, excepto para porosidades bajas. Con el método de desaleado la estructura porosa final tiene relación directa con la estructura inicial; esto es, las regiones vacías originales (aleatorias) son los precursores de los poros finales. Con el método de expansión de celda no hay poros iniciales como tal.
- A muy bajas porosidades y a la temperatura utilizada el método de expansión de celdas genera estructuras cristalinas *tensadas* que son estructuras estables bajo los parámetros de simulación utilizados, en particular para el volumen expandido.
- Las PDFs calculadas son más características de sistemas amorfos que de sistemas cristalinos.
- El corrimiento a la izquierda del pico de primeros vecinos al aumentar la porosidad lo atribuimos a que aumenta el número de átomos con menor número de coordinación, con respecto al bulto, los cuales se ubican en la «superficie» de los poros.
- La bimodalidad en el pico a «segundos vecinos», característico en las PDFs de sistemas metálicos, así como la ausencia de un mínimo entre primeros y segundos vecinos, se explica utilizando un modelo de esferas rígidas y considerando que el enlace no direccionado, característico en los metales, permite una transición continua entre la distancia a primeros vecinos y los siguientes (ausencia de mínimo), y dos distancias a «segundos» vecinos (bimodalidad).
- La caída por debajo de 1 en la PDF de las estructuras porosas es observable cuando existen poros grandes en esas estructuras (baja dimensión fractal).
- El tamaño de la caída, Δr_{debajo} , puede dar un estimado del tamaño promedio de los poros grandes (comparables con el tamaño del backbone).
- La distribución de ángulos indica que los sistemas metálicos conservan un empaquetamiento compacto en el *backbone*.

- La porosidad parece no afectar el comportamiento electrónico característico de los metales de transición, banda d llena y banda s parcialmente llena; sin embargo, al romperse la continuidad del sólido (formación de cúmulos) la estructura electrónica no tiene una forma clara.
- Las funciones de Fukui aplicadas a nuestras muestras indican un backbone aparentemente anisotrópico en cuanto a reactividad con excepción de aquellos puntos donde existen átomos con bajo número de coordinación con respecto al resto de los átomos. Esto es, los sistemas simulados muestran un comportamiento tanto electrofílico como nucleofílico.
- La presencia de valores negativos en los valores de las funciones de Fukui, no es del todo clara, pues no se puede justificar su presencia. La existencia de valores negativos es considerada por algunos como algo ficticio e inherente al cálculo; mientras que otros la consideran como algo real. Y, aunque, se han dado, en la literatura, argumentos en pro y en contra; no se ha alcanzado un consenso que establezca si un valor negativo en la función de Fukui, efectivamente, representa una realidad física.
- La dimensión fractal puede ser utilizada para caracterizar las estructuras porosas. Un valor alto en la dimensión fractal corresponde a estructuras con porosidad dispersa; mientras que, un valor bajo corresponde a estructuras con porosidad localizada. Consideramos que hacen falta más estudios que indiquen si la dimensión fractal puede emplearse para tipificar una estructura porosa en general, o es sólo una medida relativa y característica al sistema medido. En otras palabras, dos valores iguales de dimensión fractal, medidas sobre dos sistemas independientes, ¿representan un mismo tipo de estructura porosa?

Referencias

1. R. J. White, R. Luque, V. L. Budarin, J. H. Clark, D. J. Macquarrie, Supported metal nanoparticles on porous materials: Methods and applications, *Chem. Soc. Rev.*, **38** (2009) 481–494.
2. <http://www.nano.gov> Página web de la National Nanotechnology Initiative.
3. C. C. Koch (ed.), Nanostructured materials. Processing, properties, and applications., William Andrew Publishing, 2 ed., Nueva York, pág. xvii, (2007).
4. <http://www.epa.gov> Página web de la United States Environmental Protection Agency.
5. P. Moriarty, Nanostructured materials, *Rep. Prog. Phys.*, **64** (2001) 297–381.
6. E. H. Sondheimer, The mean free path of electrons in metals, *Adv. Phys.*, **50** (2001) 499–537.
7. B. Bhushan (ed.), Springer handbook of nanotechnology, 2 ed., Springer, Nueva York, (2007) 1–9.
8. J. T. Lue, A review of characterization and physical property studies of metallic nanoparticles, *Adv. Phys.*, **62** (2001) 1599–1612.
9. <http://www.plasmachem.com>
10. Kumar, C. S. S. R. (ed.), Metallic Nanomaterials, Wiley-VCH, Germany, (2009) 3–70.
11. J. B. Mooney, S. B. Radding, Spay pyrolysis processing, *Ann. Rev. Mater. Sci.*, **12** (1982) 81–101.
12. M. J. Rosen, Surfactants and interfacial phenomena, 3 ed., Wiley, New Jersey, (2007) 1–33.
13. A. Calk, D. Wexler, D. Oleszak, J. Bystrzycki, Formation of amorphous and nanostructural powder particles from amorphous metallic glass ribbons using ball milling and electrical discharge milling, *Solid State Phenom.*, **101-102** (2005) 111–116.
14. R. Z. Valiev, Recent progress in developing bulk nanostructured SPD with unique properties, *Solid State Phenom.*, **101-102** (2005) 3–12.
15. K. J. Kurzydłowski, M. Richert, J. Richert, J. Zasadziński y M. Suś-Ryszkowska, Effect of non-conventional large deformations on the formation of nanostructures in materiales, *Solid State Phenom.*, **101-102** (2005) 31–36.
16. K. Okada, K. J. D. MacKenzie en: H. Hosono, Y. Mishima, H. Takezoe, K. J. D. MacKenzie (Eds.), Nanomaterials: from research to applications, Elsevier, 1 ed., Gran Bretaña, pp. 352, (2006).

17. D. A. Crowson, D. Farkas, S. G. Corcoran, Geometric relaxation of nanoporous metals: The role of surface relaxation, *Scripta Mater.*, **56** (2007) 919–922.
18. D. A. Crowson, D. Farkas, S. G. Corcoran, Mechanical stability of nanoporous metals with small ligament sizes, *Scripta Mater.*, **61** (2009) 497–499.
19. Y. Ding, J. Erlebacher, Nanoporous metals with controlled multimodal pore size distribution, *J. Am. Chem. Soc.*, **125** (2003) 7772–7773.
20. Y. Song, J. Hormes, C. S. S. R. Kumar, Microfluidic Synthesis of Nanomaterials, *Small*, **4** (2008), 698–711.
21. K. Hu, D. Lan, X. Li, S. Zhang, Electrochemical DNA biosensor based on nanoporous gold electrode and multifunctional encoded DNA–Au Bio Bar codes, *Anal. Chem.*, **80** (2008) 9124–9130.
22. R. Klajn, T. P. Gray, P. J. Wesson, B. D. Myers, V. P. Dravid, S. K. Smoukov, B. A. Grzybowski, Bulk synthesis and surface patterning of nanoporous metals and alloys from supraspherical nanoparticle aggregates, *Adv. Funct. Mater.*, **18** (2008) 2763–2769.
23. J. Biener, A. M. Hodge, J. R. Hayes, C. A. Volkert, L. A. Zepeda–Ruiz, A. V. Hamza, F. F. Abraham, Size effects on the mechanical behavior of nanoporous Au, *Nano Lett.*, **6** (2006) 2379–2382.
24. L.-Y. Chen, J.-S. Yu, T. Fujita, M.-W. Chen, Nanoporous copper with tunable nanoporosity for SERS applications, *Adv. Funct. Mater.*, **19** (2009) 1221–1226.
25. X. Gu, L. Xu, F. Tian, Y. Ding, Au–Ag Alloy Nanoporous Nanotubes, *Nano Res*, **2** (2009) 386–393.
26. M. Hakamada, M. Mabuchi, Preparation of nanoporous Ni and Ni–Cu by dealloying of rolled Ni–Mn and Ni–Cu–Mn alloys, *J. Alloys Compd.*, (2009), doi: 10.1016 / j.jallcom.2009.06.031.
27. A. Mathur, J. Erlebacher, Size dependence of effective Young’s modulus of nanoporous gold, *Appl. Phys. Lett.*, **90** (2007) 061910.
28. A. Dursun, D. V. Pugh, S. G. Corcoran, Dealloying of Ag–Au alloys in halide-containing electrolytes. Effect on critical potential and pore size, *J. Electrochem. Soc.*, **150** (2003) B355–B360.
29. J. Erlebacher, M. J. Aziz, A. Karma A., N. Dimitrov, K. Sieradzki, Evolution of nanoporosity in dealloying, *Nature*, **410** (2001) 450–453.
30. M. G. Fontana, Corrosion Engineering, McGraw-Hill, New York, (1986).
31. D. A. Jones, Principles and prevention of corrosion, 2 ed., Prentice Hall, New York, (1996).

32. Y. Ding, Y.-J. Kim, J. Erlebacher, Nanoporous gold leaf: Ancient technology/advanced material, *Adv. Mater.*, **16** (2004) 1887–1900.
33. F. H. (Sam) Froes, O. N. Senkov, E. G. Baburaj, Synthesis of nanocrystalline materials – an overview, *Adv. Mater.*, **A301** (2001) 44–53.
34. Z. Zhang, Y. Wang, Z. Qi, W. Zhang, J. Qin, J. Frenzel, Generalized fabrication of nanoporous metals (Au, Pd, Pt, Ag, and Cu) through chemical dealloying, *J. Phys. Chem. C*, **113** (2009) 12629–12636.
35. S. Van Petegem, S. Brandstetter, R. Maass, A. M. Hodge, B. S. El-Dasher, J. Biener, B. Schmitt, C. Borca, H. Van Swygenhoven, On the microstructure of nanoporous gold: an x-ray diffraction study, *Nano Lett.*, **9** (2009) 1158–1163.
36. Y. Sun, T. J. Balk, Evolution of structure, composition, and stress in nanoporous gold thin films with grain-boundary cracks, *Metall. Mater. Trans. A*, **39A** (2008) 2656–2665.
37. J. Erlebacher, K. Sieradzki, Pattern formation during dealloying, *Scripta Mater.*, **49** (2003) 991–996.
38. J. Erlebacher, An atomistic description of dealloying porosity evolution, the critical potential, and rate-limiting behavior, *J. Electrochem. Soc.*, **151** (2004) C614–C626.
39. W. Fan, X. G. Gong, Monte Carlo simulation of surface de-alloying of Au/Ni(110), *Surf. Sci.*, **562** (2004) 219–225.
40. D. A. Porter, K. E. Easterling, Phase transformations in metals and alloys, 2 ed., CRC Taylor and Francis Group, USA, (1992) pp. 263.
41. <http://xmd.sourceforge.net/about.html>
42. C. Eilks, C. M. Elliott, Numerical simulation of dealloying by surface dissolution via the evolving surface finite element method, *J. Comput. Phys.*, **227** (2008) 9727–9741.
43. L. Verlet, Computer "experiments" on classical fluids. I. Thermodynamical properties of lennard-jones molecules, *Phys. Rev.*, **159** (1967) 98; L. Verlet, Computer "experiments" on classical fluids. II. Equilibrium correlation functions, *Phys. Rev.*, **165** (1967) 201.
44. FAST STRUCTURE SIMMULATED ANNEALING, Users Guide, Release 4.0.0, San Diego, Molecular Simulations, Inc., Septiembre (1996).
45. I. N. Lavine, Química cuántica, 5 ed., Pearson Educación, Madrid, (2001).
46. P. Hohenberg, W. W. Kohn, Inhomogeneous Electron Gas, *Phys. Rev.*, **136** (1964) B864–B871.

47. W. Kohn, L. J. Sham, Self-consistent equations including exchange and correlation effects, *Phys. Rev.*, **140** (1965) A1133–A1138.
48. S. H. Vosko, L. Wilk, M. Nusair, Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis, *Can. J. Phys.*, **58** (1980) 1200–1211.
49. R. M. Dreizler, E. K. U. Gross, Density functional theory. An approach to the quantum many-body problem, Springer–Velag, Germany, (1990) 176–179.
50. J. Harris, Simplified method for calculating the energy of weakly interacting fragments, *Phys. Rev. B*, **31** (1985) 1770–1779.
51. Z. Lin, J. Harris, A localized-basis scheme for molecular dynamics, *J. Phys.: Condens. Matter*, **5** (1993) 1055.
52. MSI, Cerius² 3.5 Quantum mechanics – Chemisty: ADF, DMol³, Gaussian, MOPAC, Zindo, *Molecula Simulations Inc.*, San Diego, (1997); Accelrys, Materials studio getting started, release 4.0, *Accelrys Software Inc.*, San Diego,(2006).
53. R. G. Parr, W. Yang, Density–Functional theory of atoms and molecules, Oxford University Press, New York, (1989) 99–101.
54. R. R. Contreras, P. Fuentealba, M. Galván, P. Pérez, A direct evaluation of regional Fukui functions in molecules, *Chem. Phys. Lett.*, **304**(1999) 405–413.
55. P. W. Ayers, W. Yang, L. J. Bartolotti, Fukui Function, in: P. K. Chattaraj (ed.), Chemical reactivity theory: a density functional view, CRC press, USA, (2009) 255–267.
56. Accelrys, MS Modeling Getting Started, User guide, Release 3.2, Accelrys Software Inc.: San Diego, 2005.
57. Y. Waseda, The structure of non-crystalline materials, McGraw-Hill, USA, (1980) caps. 1-4.
58. R. M. White, T. H. Geballe, Long range order in solids, Academic press, EUA, (1979) pp. 354-359.
59. A. A. Valladares, A. Valladares, R. M. Valladares, Computer Modeling of Nanoporous Materials: An *ab initio* Novel Approach for Silicon and Carbon, *Mater. Res. Soc. Symp. Proc.* **988** (2007).
60. R. Tilley, Understanding solids: The science of materials, John Wiley & Sons, LTD, (2004) 154–157.
61. G. W. C. Kaye, T. H. Laby, Tables of physical and chemical constants, 15 ed, Longman, London, (1993).

62. M. E. Straumanis, L. S. Yu, Lattice parameters, densities, expansion coefficients and perfection of structure of Cu and of Cu–In α phase *Acta Cryst.*, **A25** (1969) 676–682.
63. L-g. Liu, W. A. Bassett, Compression of Ag and phase transformation of NaCl, *J. Appl. Phys.* **44** (1973) 1475.
64. A. Maeland, T. B. Flanagan, Lattice spacings of gold-palladium alloys *Can. J. Phys.* **42** (1964) 2364–2366.
65. B. Delley, An all-electron numerical method for solving the local density functional for polyatomic molecules, *J. Chem. Phys.* **92** (1990) 508; B. Delley, From molecules to solids with the DMol3 approach, *J. Chem. Phys.* **113** (2000) 7756.
66. G. S. Fisheman, Monte Carlo: concepts, algorithms and applications, Springer-Verlag, USA (1995) 171-174.
67. K. H. Lee, J. Kang, S. S. Park, J. Lee, S. K. Lee, D. K. Bohme, Preferential site of attack on fullerene cations: Frontier orbitals and rate coefficients, *J. Phys. Chem. A.*, **110**(2006) 11730.
68. S. R. Elliott, Physics of amorphous materials, 2 ed., Longman Group Limited, United Kingdom, (1990) 152–157.
69. J. Hafner, Bond–Angle distribution functions in metallic glasses, *J. Phys.*, **46** (1985) C9-69–C9-78.
70. L. Hui, W. Guanghou, Z. Jijun, B. Xiufang, Cluster structure and dynamics of liquid aluminum under cooling conditions, *J. Chem. Phys.*, **116** (2002) 10809–10815.
71. A. Di Cicco, A. Trapanati, S. Faggioni, A. Filipponi, Is There Icosahedral Ordering in Liquid and Undercooled Metals?, *Phys. Rev. Lett.*, **91** (2003) 135505.
72. C.U. Santiago-Cortés, L.M. Mejía-Mendoza, R.M. Valladares, A. Valladares, A.A. Valladares, Computational alternatives to generate amorphous nanoporous structures using ab initio molecular dynamics, *J. Non-Cryst. Solids* (2011), doi:10.1016/j.jnoncrysol.2011.10.021
73. N. F. Mott, H. Jones, The theory of the properties of metals and alloys, Dover, New York, (1958) 189-192.
74. P. K. Chattaraj(ed.), Chemical reactivity theory: A density functional view, CRC Press USA, 2009.
75. P. W. Ayers, Variational principles for describing chemical reactions: Condensed reactivity indices, *J. Chem. Phys.*, **116** (2003) 8731-8744.
76. P. W. Ayers, R. C. Morrison, R. K. Roy, Can one oxidize an atom by reducing the molecule that contains it?, *Phys. Chem. Chem. Phys.*, **8** (2006) 3387–3390.

77. P. Bultincka, R. Carbó-Dorca, W. Langenaeker, Negative Fukui functions: New insights based on electronegativity equalization, *J. Chem. Phys.*, **118** (2003) 4349-4356.
78. S. R. Mulliken, Electronic population analysis LCAO-MO molecular wave functions, *J. Chem. Phys.*, **23**, (1955) 1833.
79. F.L. Hirshfeld Bonded-atom fragments for describing molecular charges densities, *Theor. Chim. Acta B.*, **44**, (1977) 129.
80. C. U. Singh, P. A. Kollman, An approach to computing electrostatic charges for molecules, *J. Comput. Chem.*, **5**, (1984) 129.
81. P. Bultinck, R. Carbó-Dorca, Negative and infinite Fukui functions: The role of diagonal dominance in the hardness matrix, *J. Chem. Phys.*, **34** (2003) 67-74.
82. D. E. Schaefer, K. D. Keefer, Structure of random porous materials: silica aerogel, *Phys. Rev. Lett.*, **56** (1986) 2199-2202.
83. J. Kieffer, C. A. Angell, Generation of fractal structures by negative pressure rupturing of SiO₂ glass, *J. Non-Cryst. Solids*, **106** (1986) 336-342.
84. K. Falconer, *Fractal Geometry: mathematical foundations and applications*, 2ed, Wiley (2003).

A Apéndices

A.1 EJEMPLO DE UN ARCHIVO `.INPUT` CON PARÁMETROS PARA LA DINÁMICA MOLECULAR

En este apéndice se muestra un archivo `.input` utilizado para realizar diámicas con el módulo `dmol3` y se describen los parámetros utilizados.

A.1.1 Descripción de los parámetros utilizados por `dmol3`.

1. **Calculate.** Dice a `dmol` que tipo de cálculo se desea realizar; los valores posibles son: `energy`, `gradient`, `optimize`, `frequency` y `optimize_frequency`.
2. **Symetry.** Se utiliza para establecer el grupo puntual de la molecula simulada; los valores posibles son: `off`, `auto`, `c1`, `cs`, `D2`...
3. **Charge.** Especifica la carga molecular total; el valor admitido es un número real.
4. **Basis.** Permite especificar el conjunto de base atómico. Es el número `dy` tipo de orbitales atómicos usados en la expansión de orbitales atómicos; `min`, `dn`, `dnp`, `dnd` y `user`.
 1. `min`. Un orbital atómico por cada orbital que es ocupado en el átomo libre.
 2. `dn`. Aproximadamente 2orbitales atómicos por cada 1 ocupado en el átomo libre.
 3. `dnp`. Usa la base `dn` con funciones de polarización, es decir, funciones con momento angular, uno más alto que el orbital más alto ocupado (por ejemplo, 2p para el H, 3d para C). en un átomo libre.
 4. `dnd`. Como `dnp` más orbitales d.
 5. `user`. Permite personalizar la base utilizada.
5. **Pseudopotential.** Especifica la forma del potencial del *core* o la corrección relativista usada en el cálculo; `dspp` (cálculos con DFT), `ecp` (calculos con Hartre-Fock), `vpsr` y `none`.
6. **Functional.** Especifica la correlación local y las funcionales gradiente-correctadas para intercambio y correlación; los valores posibles son: `vwn`, `pwc`, `bp`...
7. **Harris.** Utilizar la aproximación de Harris.

8. `Integration_grid`. Controla el número de puntos para la integración numérica para el cálculo de los elementos de matriz; valores posibles: `xfine`, `fine`, `medium`, `coarse`, `xcoarse` y `user`.
9. `Occupation`. Especifica la ocupación de los orbitales; valores permitidos: `Fermi`, `Terminal`, ...
10. `Cutoff_Global`. Permite especificar los radios de corte en espacio real para el cálculo de las bases atómicas.
11. `Kpoints`. Especifica la configuración de puntos K utilizados para el muestreo de la zona de Brillouin.

A.1.2 Archivo `.input`.

```

# Task parameters
Calculate                Simulated_Annealing
Symmetry                 off

# Electronic parameters
Charge                   0
Basis                    dnd
Pseudopotential         dspp
Functional                vwn
Harris                   on
Integration_grid         xfine
Occupation                fermi
Cutoff_Global            5.0000 angstrom
Kpoints                  off

#MD and Simulated Annealing Keywords
MD_Time_Step             7.000

MD_SimAnn_Panel          1
  1      500 MD_NVT      300.00      0.00

# Print options
Print                    eigval_last_it

Max_Memory                10240
# Velocidades según la distribución de Maxwell-Boltzmann
# random number seed : 1282604344.6834011
# número de átomos    : 64
# temperatura         : 300 K
# velocidad máxima    : 15000 m/s (150.0 angstrom/ps)
# num                 |V|                V_x                V_y                V_z
MD_Velocity User_Defined 64
1                1.984515      -0.141099      -0.733597      -0.664776      # Ag

```

2	1.309107	-0.939274	-0.266459	0.216252	# Ag
3	2.804993	-0.542144	0.167783	-0.823364	# Ag
4	1.646514	-0.310891	0.492193	-0.813076	# Ag
5	2.677534	-0.649835	-0.284694	0.704744	# Ag
6	1.691025	-0.067120	0.907540	-0.414568	# Ag
7	1.975529	0.459105	-0.881208	-0.112672	# Ag
8	3.893852	-0.657491	0.473095	-0.586418	# Ag
9	0.729179	0.717662	-0.464699	0.518669	# Ag
10	1.999482	-0.266808	0.296229	0.917094	# Ag
11	3.691594	0.235735	0.721683	0.650847	# Ag
12	3.676361	-0.248112	-0.887663	-0.387936	# Ag
13	6.115678	0.503139	-0.408140	-0.761757	# Ag
14	1.337062	0.543451	-0.769050	-0.336486	# Ag
15	1.902453	0.352634	-0.632766	0.689388	# Ag
16	2.228372	-0.073053	-0.460869	-0.884456	# Ag
17	1.771916	0.490050	0.720622	0.490463	# Ag
18	1.681923	-0.036647	-0.712502	0.700712	# Ag
19	1.701035	-0.250784	0.100851	0.962776	# Ag
20	0.878402	0.644263	0.708272	-0.288574	# Ag
21	2.373958	-0.927402	0.181359	0.327162	# Ag
22	3.910344	-0.301828	-0.652306	0.695268	# Ag
23	1.802795	0.259406	-0.913853	-0.312379	# Ag
24	3.697824	0.829070	-0.497275	0.255656	# Ag
25	2.022455	-0.699989	-0.712363	0.050540	# Ag
26	2.874603	-0.972939	0.198189	0.118791	# Ag
27	2.933209	0.793760	-0.606945	-0.039529	# Ag
28	1.670817	-0.675289	0.051216	0.735773	# Ag
29	2.905648	-0.609076	-0.781465	-0.135420	# Ag
30	2.431641	0.451764	-0.502724	0.737006	# Ag
31	1.733556	-0.525192	0.599628	-0.603837	# Ag
32	2.844338	-0.470451	-0.769070	-0.432674	# Ag
33	2.342102	-0.257553	-0.627853	-0.734484	# Ag
34	2.266320	0.178167	0.755508	0.630448	# Ag
35	3.955597	-0.399837	0.905940	0.139295	# Ag
36	2.044197	0.623397	-0.780324	0.049717	# Ag
37	1.408843	-0.564220	0.422272	-0.709466	# Ag
38	3.330959	-0.026483	0.037941	0.998929	# Ag
39	1.842873	-0.559560	-0.795404	0.232864	# Ag
40	2.220258	-0.196143	0.633546	-0.748430	# Ag
41	3.104835	0.304298	0.085727	0.948711	# Ag
42	0.825232	0.715694	-0.688948	-0.114598	# Ag
43	1.968117	-0.668026	-0.529262	0.523090	# Ag
44	3.337718	-0.351977	0.869909	-0.345500	# Ag
45	2.085015	0.175358	-0.979459	0.099549	# Ag
46	2.924040	0.249077	0.548652	0.798086	# Ag
47	2.372206	0.555075	0.647939	-0.521600	# Ag
48	2.142502	-0.133341	-0.119534	-0.983835	# Ag
49	1.777429	0.551604	-0.834077	-0.006908	# Ag
50	2.364872	0.316234	-0.911856	0.261753	# Ag
51	3.714132	-0.331049	0.591406	-0.735286	# Ag
52	1.682115	-0.601621	0.653368	0.459524	# Ag
53	1.543410	-0.757227	-0.231835	0.610622	# Ag
54	3.198723	0.902400	0.382076	0.199228	# Ag
55	2.533965	-0.067176	0.495384	0.866073	# Ag
56	2.649479	-0.807019	-0.551846	-0.210203	# Ag
57	4.003465	-0.853380	0.391379	0.344332	# Ag

58	3.191683	0.389751	-0.526097	-0.755855	# Ag
59	4.167524	-0.272990	-0.744271	0.609538	# Ag
60	1.420956	-0.712900	0.223361	0.664743	# Ag
61	2.642238	-0.568993	-0.612112	-0.549150	# Ag
62	2.010177	0.980469	-0.185830	0.064399	# Ag
63	3.194103	0.367291	-0.603743	0.707525	# Ag
64	2.048784	-0.871522	-0.263102	0.413796	# Ag

A.2 EJEMPLO DE UN ARCHIVO .INPUT CON PARÁMETROS PARA LA OPTIMIZACIÓN

```
# Task parameters
Calculate                optimize
Opt_energy_convergence  2.0000e-005
Opt_gradient_convergence 4.0000e-003 A
Opt_displacement_convergence 5.0000e-003 A
Opt_iterations           1500
Opt_max_displacement     0.3000 A
Symmetry                 off

# Electronic parameters
Spin_polarization        restricted
Charge                   0
Basis                    dnd
Pseudopotential          dspp
Functional                vwn
Harris                   on
Aux_density              monopole
Integration_grid         xfine
Occupation               fermi
Cutoff_Global            5.0000 angstrom
Kpoints                  off
Max_Memory               10240

# Print options
Print                    eigval_last_it
```

A.3 NÚMERO DE ÁTOMOS EN UNA CELDA DESALEADA

n = Número de veces la arista; %d = porcentaje de desaleación

	n									
%d	1	2	3	4	5	6	7	8	9	10
100%	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

99%	0.04	0.32	1.08	2.56	5.00	8.64	13.72	20.48	29.16	40.00
98%	0.08	0.64	2.16	5.12	10.00	17.28	27.44	40.96	58.32	80.00
97%	0.12	0.96	3.24	7.68	15.00	25.92	41.16	61.44	87.48	120.00
96%	0.16	1.28	4.32	10.24	20.00	34.56	54.88	81.92	116.64	160.00
95%	0.20	1.60	5.40	12.80	25.00	43.20	68.60	102.40	145.80	200.00
94%	0.24	1.92	6.48	15.36	30.00	51.84	82.32	122.88	174.96	240.00
93%	0.28	2.24	7.56	17.92	35.00	60.48	96.04	143.36	204.12	280.00
92%	0.32	2.56	8.64	20.48	40.00	69.12	109.76	163.84	233.28	320.00
91%	0.36	2.88	9.72	23.04	45.00	77.76	123.48	184.32	262.44	360.00
90%	0.40	3.20	10.80	25.60	50.00	86.40	137.20	204.80	291.60	400.00
89%	0.44	3.52	11.88	28.16	55.00	95.04	150.92	225.28	320.76	440.00
88%	0.48	3.84	12.96	30.72	60.00	103.68	164.64	245.76	349.92	480.00
87%	0.52	4.16	14.04	33.28	65.00	112.32	178.36	266.24	379.08	520.00
86%	0.56	4.48	15.12	35.84	70.00	120.96	192.08	286.72	408.24	560.00
85%	0.60	4.80	16.20	38.40	75.00	129.60	205.80	307.20	437.40	600.00
84%	0.64	5.12	17.28	40.96	80.00	138.24	219.52	327.68	466.56	640.00
83%	0.68	5.44	18.36	43.52	85.00	146.88	233.24	348.16	495.72	680.00
82%	0.72	5.76	19.44	46.08	90.00	155.52	246.96	368.64	524.88	720.00
81%	0.76	6.08	20.52	48.64	95.00	164.16	260.68	389.12	554.04	760.00
80%	0.80	6.40	21.60	51.20	100.00	172.80	274.40	409.60	583.20	800.00
79%	0.84	6.72	22.68	53.76	105.00	181.44	288.12	430.08	612.36	840.00
78%	0.88	7.04	23.76	56.32	110.00	190.08	301.84	450.56	641.52	880.00
77%	0.92	7.36	24.84	58.88	115.00	198.72	315.56	471.04	670.68	920.00
76%	0.96	7.68	25.92	61.44	120.00	207.36	329.28	491.52	699.84	960.00
75%	1.00	8.00	27.00	64.00	125.00	216.00	343.00	512.00	729.00	1000.00
74%	1.04	8.32	28.08	66.56	130.00	224.64	356.72	532.48	758.16	1040.00
73%	1.08	8.64	29.16	69.12	135.00	233.28	370.44	552.96	787.32	1080.00
72%	1.12	8.96	30.24	71.68	140.00	241.92	384.16	573.44	816.48	1120.00
71%	1.16	9.28	31.32	74.24	145.00	250.56	397.88	593.92	845.64	1160.00
70%	1.20	9.60	32.40	76.80	150.00	259.20	411.60	614.40	874.80	1200.00
69%	1.24	9.92	33.48	79.36	155.00	267.84	425.32	634.88	903.96	1240.00
68%	1.28	10.24	34.56	81.92	160.00	276.48	439.04	655.36	933.12	1280.00
67%	1.32	10.56	35.64	84.48	165.00	285.12	452.76	675.84	962.28	1320.00
66%	1.36	10.88	36.72	87.04	170.00	293.76	466.48	696.32	991.44	1360.00
65%	1.40	11.20	37.80	89.60	175.00	302.40	480.20	716.80	1020.60	1400.00
64%	1.44	11.52	38.88	92.16	180.00	311.04	493.92	737.28	1049.76	1440.00
63%	1.48	11.84	39.96	94.72	185.00	319.68	507.64	757.76	1078.92	1480.00
62%	1.52	12.16	41.04	97.28	190.00	328.32	521.36	778.24	1108.08	1520.00
61%	1.56	12.48	42.12	99.84	195.00	336.96	535.08	798.72	1137.24	1560.00
60%	1.60	12.80	43.20	102.40	200.00	345.60	548.80	819.20	1166.40	1600.00
59%	1.64	13.12	44.28	104.96	205.00	354.24	562.52	839.68	1195.56	1640.00
58%	1.68	13.44	45.36	107.52	210.00	362.88	576.24	860.16	1224.72	1680.00
57%	1.72	13.76	46.44	110.08	215.00	371.52	589.96	880.64	1253.88	1720.00
56%	1.76	14.08	47.52	112.64	220.00	380.16	603.68	901.12	1283.04	1760.00
55%	1.80	14.40	48.60	115.20	225.00	388.80	617.40	921.60	1312.20	1800.00
54%	1.84	14.72	49.68	117.76	230.00	397.44	631.12	942.08	1341.36	1840.00
53%	1.88	15.04	50.76	120.32	235.00	406.08	644.84	962.56	1370.52	1880.00
52%	1.92	15.36	51.84	122.88	240.00	414.72	658.56	983.04	1399.68	1920.00
51%	1.96	15.68	52.92	125.44	245.00	423.36	672.28	1003.52	1428.84	1960.00
50%	2.00	16.00	54.00	128.00	250.00	432.00	686.00	1024.00	1458.00	2000.00
49%	2.04	16.32	55.08	130.56	255.00	440.64	699.72	1044.48	1487.16	2040.00
48%	2.08	16.64	56.16	133.12	260.00	449.28	713.44	1064.96	1516.32	2080.00
47%	2.12	16.96	57.24	135.68	265.00	457.92	727.16	1085.44	1545.48	2120.00
46%	2.16	17.28	58.32	138.24	270.00	466.56	740.88	1105.92	1574.64	2160.00
45%	2.20	17.60	59.40	140.80	275.00	475.20	754.60	1126.40	1603.80	2200.00
44%	2.24	17.92	60.48	143.36	280.00	483.84	768.32	1146.88	1632.96	2240.00

43%	2.28	18.24	61.56	145.92	285.00	492.48	782.04	1167.36	1662.12	2280.00
42%	2.32	18.56	62.64	148.48	290.00	501.12	795.76	1187.84	1691.28	2320.00
41%	2.36	18.88	63.72	151.04	295.00	509.76	809.48	1208.32	1720.44	2360.00
40%	2.40	19.20	64.80	153.60	300.00	518.40	823.20	1228.80	1749.60	2400.00
39%	2.44	19.52	65.88	156.16	305.00	527.04	836.92	1249.28	1778.76	2440.00
38%	2.48	19.84	66.96	158.72	310.00	535.68	850.64	1269.76	1807.92	2480.00
37%	2.52	20.16	68.04	161.28	315.00	544.32	864.36	1290.24	1837.08	2520.00
36%	2.56	20.48	69.12	163.84	320.00	552.96	878.08	1310.72	1866.24	2560.00
35%	2.60	20.80	70.20	166.40	325.00	561.60	891.80	1331.20	1895.40	2600.00
34%	2.64	21.12	71.28	168.96	330.00	570.24	905.52	1351.68	1924.56	2640.00
33%	2.68	21.44	72.36	171.52	335.00	578.88	919.24	1372.16	1953.72	2680.00
32%	2.72	21.76	73.44	174.08	340.00	587.52	932.96	1392.64	1982.88	2720.00
31%	2.76	22.08	74.52	176.64	345.00	596.16	946.68	1413.12	2012.04	2760.00
30%	2.80	22.40	75.60	179.20	350.00	604.80	960.40	1433.60	2041.20	2800.00
29%	2.84	22.72	76.68	181.76	355.00	613.44	974.12	1454.08	2070.36	2840.00
28%	2.88	23.04	77.76	184.32	360.00	622.08	987.84	1474.56	2099.52	2880.00
27%	2.92	23.36	78.84	186.88	365.00	630.72	1001.56	1495.04	2128.68	2920.00
26%	2.96	23.68	79.92	189.44	370.00	639.36	1015.28	1515.52	2157.84	2960.00
25%	3.00	24.00	81.00	192.00	375.00	648.00	1029.00	1536.00	2187.00	3000.00
24%	3.04	24.32	82.08	194.56	380.00	656.64	1042.72	1556.48	2216.16	3040.00
23%	3.08	24.64	83.16	197.12	385.00	665.28	1056.44	1576.96	2245.32	3080.00
22%	3.12	24.96	84.24	199.68	390.00	673.92	1070.16	1597.44	2274.48	3120.00
21%	3.16	25.28	85.32	202.24	395.00	682.56	1083.88	1617.92	2303.64	3160.00
20%	3.20	25.60	86.40	204.80	400.00	691.20	1097.60	1638.40	2332.80	3200.00
19%	3.24	25.92	87.48	207.36	405.00	699.84	1111.32	1658.88	2361.96	3240.00
18%	3.28	26.24	88.56	209.92	410.00	708.48	1125.04	1679.36	2391.12	3280.00
17%	3.32	26.56	89.64	212.48	415.00	717.12	1138.76	1699.84	2420.28	3320.00
16%	3.36	26.88	90.72	215.04	420.00	725.76	1152.48	1720.32	2449.44	3360.00
15%	3.40	27.20	91.80	217.60	425.00	734.40	1166.20	1740.80	2478.60	3400.00
14%	3.44	27.52	92.88	220.16	430.00	743.04	1179.92	1761.28	2507.76	3440.00
13%	3.48	27.84	93.96	222.72	435.00	751.68	1193.64	1781.76	2536.92	3480.00
12%	3.52	28.16	95.04	225.28	440.00	760.32	1207.36	1802.24	2566.08	3520.00
11%	3.56	28.48	96.12	227.84	445.00	768.96	1221.08	1822.72	2595.24	3560.00
10%	3.60	28.80	97.20	230.40	450.00	777.60	1234.80	1843.20	2624.40	3600.00
9%	3.64	29.12	98.28	232.96	455.00	786.24	1248.52	1863.68	2653.56	3640.00
8%	3.68	29.44	99.36	235.52	460.00	794.88	1262.24	1884.16	2682.72	3680.00
7%	3.72	29.76	100.44	238.08	465.00	803.52	1275.96	1904.64	2711.88	3720.00
6%	3.76	30.08	101.52	240.64	470.00	812.16	1289.68	1925.12	2741.04	3760.00
5%	3.80	30.40	102.60	243.20	475.00	820.80	1303.40	1945.60	2770.20	3800.00
4%	3.84	30.72	103.68	245.76	480.00	829.44	1317.12	1966.08	2799.36	3840.00
3%	3.88	31.04	104.76	248.32	485.00	838.08	1330.84	1986.56	2828.52	3880.00
2%	3.92	31.36	105.84	250.88	490.00	846.72	1344.56	2007.04	2857.68	3920.00
1%	3.96	31.68	106.92	253.44	495.00	855.36	1358.28	2027.52	2886.84	3960.00
0%	4.00	32.00	108.00	256.00	500.00	864.00	1372.00	2048.00	2916.00	4000.00

A.4 PROGRAMA CELDAS_ASIMETRICAS

El programa `celdas_asimetricas` genera las dimensiones de una supercelda, lo más simétrica posible, para las que se cumpla un porcentaje de desaleación y un número de átomos dado.

```

1  #!/usr/bin/python

2  def factores(num_t, numero_en_celda, desaleado, tol):
3      lista = []
4      aux_l = []
5      dic = {}
6      aux = 0

7      for i in range(1, 20):
8          for j in range(1,20):
9              for k in range(1,20):
10                 real = int(i*j*k*numero_en_celda*desaleado)
11                 if ( num_t*(1-tol) <= real <= num_t*(1+tol)):
12                     aux_l = [i,j,k]
13                     aux_l.sort()
14                     aux= tuple(aux_l)
15                     if not dic.has_key(aux):
16                         #print real
17                         dic[aux] = real
18                         #print i, j, k, real
19                         lista += [aux]
20                 else:
21                     continue
22         print lista[-1], dic[lista[-1]]

23 for i in range(1, 10):
24     print i*0.1
25     factores(250, 4, i*0.1, 0.1)
26     print

```

A.5 NÚMERO DE ÁTOMOS EN UNA CELDA ASIMÉTRICA DESALEADA

Este apéndice presenta los resultados de ejecutar el programa `celdas_asimetricas` con cuatro números objetivo de átomos: 64, 108, 250 y 500. Los número entre paréntesis muestran las dimensiones de una celda unitaria desaleada una cantidad $d\%$, el número fuera del parentesis es el número de átomos resultante para esa geometría y esa cantidad de desaleación. La nomenclatura de la tabla es la siguiente: n =Número de veces la arista; $\%d$ =porcentaje de desaleación; (l, m, n) =dimensiones de la supercelda; $\#a$ =número de átomos de la celda; $\#A$ =número de átomos objetivo.

#A			
64	108	250	500

%d	(1, m, n), #a	(1, m, n), #a	(1, m, n), #a	(1, m, n), #a
90	(5, 5, 7), 70	(6, 7, 7) 117	(8, 9, 9) 259	(11, 11, 11) 532
80	(4, 4, 5), 64	(5, 5, 5) 100	(7, 7, 7) 274	(8, 9, 9) 518
70	(3, 3, 6), 64	(4, 4, 6) 115	(6, 6, 6) 259	(7, 8, 8) 537
60	(2, 4, 5), 64	(4, 4, 4) 102	(5, 5, 6) 240	(7, 7, 7) 548
50	(2, 4, 4), 64	(3, 3, 6) 108	(5, 5, 5) 250	(6, 6, 7) 504
40	(3, 3, 3), 64	(3, 4, 4) 115	(4, 5, 5) 240	(6, 6, 6) 518
30	(2, 3, 4), 67	(3, 3, 4) 100	(4, 4, 6) 268	(5, 6, 6) 504
20	(2, 2, 5), 64	(3, 3, 4) 115	(4, 4, 5) 256	(5, 5, 6) 480
10	(2, 3, 3), 64	(2, 4, 4) 115	(4, 4, 4) 230	(5, 5, 6) 540

A.6 PROGRAMA ALEACIONES

```

1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-

3  from __future__ import division

4  __author__ = 'Ulises Santiago <ulises_osa@yahoo.com.mx>'
5  __date__   = "Mayo2009"

6  ##-----
7  from numpy import *
8  from math import *
9  import getopt
10 import sys
11 import random
12 import time

13 deg2rad = pi/180.
14 ##-----
15 def uso(nombre_programa):
16     texto = "uso: " + nombre_programa + " " [OPCIONES]

17     DESCRIPCION
18     Programa para crear una aleacion substitucional. Los atomos del soluto
19     se substituyen de forma aleatoria.

20     OPCIONES
21     -h --help
22         Muestra esta ayuda y sale.

23     -a --parametro_de_red_a=numero
24         Tamano del parametro de red para un sistema cristalino cubico.
25         Por defecto se utiliza el parametro de red de la plata 4.0857
26         (angstrom).

```

```

27     -b --parametro_de_red_b=numero
28         Tamano del parametro de red para un sistema cristalino cubico.
29         Por defecto se utiliza el parametro de red de la plata 4.0857
30         (angstrom).

31     -c --parametro_de_red_c=numero
32         Tamano del parametro de red para un sistema cristalino cubico.
33         Por defecto se utiliza el parametro de red de la plata 4.0857
34         (angstrom).

35     -x --concentracion_soluto=numero
36         Concentracion del soluto. Debe darse en fraccion.

37     -d --disolvente=nombre
38         Elemento del disolvente en la aleacion. Por defecto es Ag.

39     -e --soluto=nombre
40         Elemento del soluto en la aleacion. Por defecto es Au.

41     -n --veces_la_arista=numero
42         Numero de veces el parametro de red que se empleara para formar
43         una arista de la super celda. El volumen de la supercelda sera
44         n^3. Por defecto este numero se ajusta a 10.

45     -o --output=nombre del sistema
46         Establece el nombre del sistema, este sera el nombre base de los
47         archivos de salida, car y xyz. Si esta opcion no se utiliza el
48         nombre del sistema sera 'AgAu'.

49     -s --sistema_base=<fcc|bcc|sc|diamante|...>
50         Indica el sistema cristalino base a considerar para la construccion
51         del cumulo. Por defecto se utilizara el sistema fcc.

52     -f --sin_disolvente
53         Genera la estructura de la aleación sólo con el soluto

54     -g --sin_soluto
55         Genera la estructura de la aleación sólo con el disolvente

56     -k --densidad
57         La densidad deseada de la aleación, esta opción elimina los
58         parametros de red

59     ""

60     print texto

```

```

61  ##-----
62  masas_atomicas = {
63      'H' : 1.0079, 'He': 4.0026, 'Li': 6.9410, 'Be': 9.0122,
64      'B' : 10.8110, 'C' : 12.0107, 'N' : 14.0067, 'O' : 15.9994,
65      'F' : 18.9984, 'Ne': 20.1797, 'Na': 22.9898, 'Mg': 24.305 ,
66      'Al': 26.9815, 'Si': 28.0855, 'P' : 30.9738, 'S' : 32.065 ,
67      'Cl': 35.453 , 'Ar': 39.948 , 'K' : 39.0983, 'Ca': 40.078 ,
68      'Sc': 44.9559, 'Ti': 47.867 , 'V' : 50.9415, 'Cr': 51.9961,
69      'Mn': 54.9381, 'Fe': 55.845 , 'Co': 58.93332, 'Ni': 58.6934,
70      'Cu': 63.546 , 'Zn': 65.409 , 'Ga': 69.723 , 'Ge': 72.64 ,
71      'As': 74.9216, 'Se': 78.96 , 'Br': 79.904 , 'Kr': 83.798 ,
72      'Rb': 85.4678, 'Sr': 87.62 , 'Y' : 88.9059, 'Zr': 91.224 ,
73      'Nb': 92.9064, 'Mo': 95.94 , 'Tc': 98 , 'Ru': 101.07 ,
74      'Rh': 102.9055, 'Pd': 106.42 , 'Ag': 107.8682, 'Cd': 112.411 ,
75      'In': 114.818 , 'Sn': 118.71 , 'Sb': 121.76 , 'Te': 127.6 ,
76      'I' : 126.9045, 'Xe': 131.293 , 'Cs': 132.9055, 'Ba': 137.327 ,
77      'La': 138.9055, 'Ce': 140.116 , 'Pr': 140.9077, 'Nd': 144.242 ,
78      'Pm': 145. , 'Sm': 150.36 , 'Eu': 151.964 , 'Gd': 157.25 ,
79      'Tb': 158.9254, 'Dy': 162.5 , 'Ho': 164.9303, 'Er': 167.259 ,
80      'Tm': 168.9342, 'Yb': 173.04 , 'Lu': 174.967 , 'Hf': 178.49 ,
81      'Ta': 180.9479, 'W' : 183.84 , 'Re': 186.207 , 'Os': 190.23 ,
82      'Ir': 192.217 , 'Pt': 195.084 , 'Au': 196.9666, 'Hg': 200.59 ,
83      'Tl': 204.3833, 'Pb': 207.2 , 'Bi': 208.9804, 'Po': 209. ,
84      'At': 210. , 'Rn': 222. , 'Fr': 223. , 'Ra': 226. ,
85      'Ac': 227. , 'Th': 232.0381, 'Pa': 231.0359, 'U' : 238.0289,
86      'Np': 237. , 'Pu': 244. , 'Am': 243. , 'Cm': 247. ,
87      'Bk': 247. , 'Cf': 251. , 'Es': 252. , 'Fm': 257. ,
88      'Md': 258. , 'No': 259. , 'Lr': 262. , 'Rf': 261. ,
89      'Db': 262. , 'Sg': 266. }

90  def num_atomos_sistema( multiplicador, sistema='fcc'):

91      if sistema == 'fcc':
92          atomos_celda = 4
93      elif sistema == 'bcc':
94          atomos_celda = 2
95      elif sistema == 'diamante':
96          atomos_celda = 8
97      elif sistema == 'sc':
98          atomos_celda = 1
99      elif sistema == 'grafito':
100         atomos_celda = 4
101      else:
102         print "no implementado"
103         atomos_celda = 0
104         sys.exit(1)

105         return int(atomos_celda * multiplicador**3)

106  ##-----
107  def num_atomos_concentracion(num_total, fraccion):
108      numero = int(round(num_total * fraccion))
109      fraccion = 1.*numero/num_total
110      return numero, fraccion

```

```

111 ##-----
112 def calcula_arista(densidad, soluto, disolvente, atomos_s, atomos_d):

113     m_disolvente = masas_atomicas[disolvente] #g/mol Cu
114     m_soluto     = masas_atomicas[soluto] #g/mol Zr
115     NA=6.022e23 #mol^-1
116     #densidad #g/cm^3

117     #Cu64Zr36
118     m = m_disolvente*atomo_s_d + m_soluto*atomo_s #g.atomo/mol
119     m /= NA #g

120     L = m / densidad #cm^3
121     L = L**(1./3)
122     L *= 1.e8
123     return L

124 ##-----
125 def calcula_densidad(soluto, disolvente, atomos_s, atomos_d, volumen):
126     m_disolvente = masas_atomicas[disolvente] #g/mol
127     m_soluto     = masas_atomicas[soluto] #g/mol
128     NA=6.022e23 #mol^-1
129     m = m_disolvente*atomo_s_d + m_soluto*atomo_s #g.atomo/mol
130     m /= NA #g
131     return m/volumen

132 ##-----
133 def vector_translacion(sistema):

134     if sistema=='fcc':
135         return [[ 0.0,  0.0,  0.0],
136                [ 0.0,  0.5,  0.5],
137                [ 0.5,  0.0,  0.5],
138                [ 0.5,  0.5,  0.0]
139                ]
140     elif sistema == 'bcc':
141         return [[ 0.0,  0.0,  0.0 ],
142                [ 0.5,  0.5,  0.5 ]
143                ]
144     elif sistema == 'romboedral':
145         return [[ 0.0 ,  0. ,  0.0 ],
146                [ 2./3.,  1./3.,  1./3.],
147                [-1./3.,  1./3.,  1./3.],
148                [-1./3., -2./3.,  1./3.]
149                ]
150     elif sistema == 'base_centrada':
151         return [[ 0.0 ,  0. ,  0.0 ],
152                [ 1./2.,  1./2.,  0.0 ]
153                ]
154     elif sistema == 'diamante':
155         return [[ 0.0 ,  0.0 ,  0.0 ],
156                [ 0.0 ,  0.5 ,  0.5 ],

```



```

157         [ 0.5 , 0.0 , 0.5 ],
158         [ 0.5 , 0.5 , 0.0 ],
159         [ 1./4., 1./4., 1./4.],
160         [ 1./4., 3./4., 3./4.],
161         [ 3./4., 1./4., 3./4.],
162         [ 3./4., 3./4., 1./4.]
163     ]
164     elif sistema=='grafito':
165         return [[ 0.0 , 0.0 , 0.25],
166                [ 0.0 , 0.0 , 0.75],
167                [ 1./3., 2./3., 0.25],
168                [ 2./3., 1./3., 0.75]
169                ]
170     elif sistema == 'sc':
171         return [[0., 0., 0.]]
172     else:
173         print 'Sistema cristalino no registrado'
174         sys.exit(1)
175     ##-----
176 def vector_abc(a, b, c, alfa, beta, gama):
177     alfa *= deg2rad
178     beta *= deg2rad
179     gama *= deg2rad
180     v_a = array([a, 0, 0])
181     v_b = array([b*cos(gama), b*sin(gama), 0])
182     v_c = array([c*cos(beta),
183                 c*cos(alfa)*sin(gama),
184                 c*sin(acos(sqrt(
185                     cos(beta)**2 +
186                     cos(alfa)**2 *
187                     sin(gama)**2)
188                 ) ) ])
189
190     return v_a, v_b, v_c
191
192     ##-----
193 class Crea_car:
194     def __init__(self, nombre_archivo, abc, angulos):
195         cabecera = "!BIOSYM archive 3\nPBC=ON\n"
196         cabecera += "aleaciones, Ulises Santiago <ulises_osa yahoo.com.mx>"
197         cabecera += "\n!DATE "
198         cabecera += time.ctime()
199         cabecera += "\nPBC "
200
201         for i in abc:
202             cabecera += "%9.4f "%i
203
204         for i in angulos:
205             cabecera += "%9.4f "%i
206
207         cabecera += "(P1)\n"

```

```

203         self.__archivo_car = open(nombre_archivo + ".car", 'w')
204         self.__archivo_car.write(cabecera)

205     def agrega_coordenadas(self, tipo_atomo, num, coordenadas):
206         self.__archivo_car.write('%-7s'%(tipo_atomo + str(num)))
207         self.__archivo_car.write('%12.9f  %12.9f  %12.9f  '%coordenadas)
208         self.__archivo_car.write('XXXX 1      xx      %-2s  0.000\n'%tipo_atomo)

209     def finaliza_archivo(self):
210         self.__archivo_car.write('end\nend\n')
211         self.__archivo_car.close()

212     ##-----
213     class Crea_xyz:
214         def __init__(self, nombre_archivo):
215             cabecera = " "*50
216             cabecera += "\n" + nombre_archivo + "\n"
217             self.__archivo_car = open(nombre_archivo + ".xyz", 'w')
218             self.__archivo_car.write(cabecera)
219             self.__nombre_archivo = nombre_archivo + ".xyz"
220             self.__cont = 0

221         def agrega_coordenadas(self, tipo_atomo, coordenadas):
222             self.__cont += 1
223             self.__archivo_car.write('%-7s'%(tipo_atomo))
224             self.__archivo_car.write('%12.9f  %12.9f  %12.9f\n'%coordenadas)

225         def finaliza_archivo(self):
226             self.__archivo_car.close()
227             self.__archivo_car = open(self.__nombre_archivo, 'r+')
228             self.__archivo_car.write('%d'%self.__cont)
229             self.__archivo_car.close()

230     ##-----
231     def a_vegard(x):
232         return (1-x)*4.0857 + x*4.0782

233     ##-----
234     def main():

235         ## fcc
236         sistema_base = 'fcc'
237         sistema_cristalino = array(vector_translacion(sistema_base))
238         #parametro_de_red = 4.0857 ## angstroms
239         #parametro_de_red_b = 4.0857 ## angstroms
240         #parametro_de_red_c = 4.0857 ## angstroms

241         nombre_sistema = 'AgAu'
242         tamaño_arista = 3 ## veces parámetro de red
243         disolvente = 'Ag'

```

```

244     soluto                = 'Au'
245     concentracion        = 0.20
246     parametro_de_red_a    = 4.0857 # a_vegard(concentracion)
247     sin_soluto           = False
248     sin_disolvente        = False
249     con_densidad          = False

250     densidad              = 0.0

251     ### argumento, ver documentación del módulo 'getopt'
252     try:
253         opciones, argumentos = getopt.getopt(sys.argv[1:], \
254             "hfga:b:c:n:o:x:s:d:e:k:", ["help", "sin_soluto", "sin_disolvente",
255             "parametro_de_red_a=", "parametro_de_red_b=", \
256             "parametro_de_red_c=", "veces_la_arista=", \
257             "output=", "concentracion=", "sistema_base=", \
258             "disolvente=", "soluto=", "densidad="])
259     except getopt.error, msg:
260         print msg
261         print "Para ayuda use --help"
262         sys.exit(2)

263     ## si se elije la opción de ayuda
264     for (opcion, valor) in opciones:
265         if opcion in ("-h", "--help"):
266             uso(sys.argv[0])
267             sys.exit(0)

268     #####

269     ## procesamiento de opciones
270     for (opcion, valor) in opciones:
271         if opcion in ("-a", "--parametro_de_red_a"):
272             parametro_de_red_a = float(valor)

273         elif opcion in ("-b", "--parametro_de_red_b"):
274             parametro_de_red_b = float(valor)

275         elif opcion in ("-c", "--parametro_de_red_c"):
276             parametro_de_red_c = float(valor)

277         elif opcion in ("-n", "--tamano_arista"):
278             tamano_arista = float(valor)

279         elif opcion in ("-o", "--output"):
280             nombre_sistema = valor

281         elif opcion in ("-s", "--sistema_base"):
282             sistema_base = valor
283             sistema_cristalino = array(vector_translacion(valor))

```

```

284     elif opcion in ("-x", "--concentracion"):
285         concentracion = float(valor)
286         #parametro_de_red = a_vegard(concentracion)

287     elif opcion in ("-d", "--disolvente"):
288         disolvente = valor

289     elif opcion in ("-e", "--soluto"):
290         soluto = valor

291     elif opcion in ("-f", "--sin_disolvente"):
292         sin_disolvente = True

293     elif opcion in ("-g", "--sin_soluto"):
294         sin_soluto = True

295     elif opcion in ("-k", "--densidad"):
296         con_densidad=True
297         densidad = float(valor)
298     else:
299         assert False, "Opcion inmanejable: " + opcion

300 num_atomos_total = num_atomos_sistema(ceil(tamano_arista), sistema_base)
301 num_solutos, fraccion = num_atomos_concentracion(num_atomos_total, concentracion)

302 if con_densidad:
303     parametro_de_red_a = calcula_arista(densidad, soluto, disolvente, \
304                                     num_solutos, num_atomos_total - \
305                                     num_solutos) / ceil(tamano_arista)
306     parametro_de_red_b = parametro_de_red_a
307     parametro_de_red_c = parametro_de_red_a

308 if sistema_base != 'grafito':
309     v_a, v_b, v_c = vector_abc(1., 1., 1., 90, 90, 90)
310     parametro_de_red_c = parametro_de_red_b= parametro_de_red_a
311     archivocar = Crea_car(nombre_sistema,\
312                          (parametro_de_red_a*ceil(tamano_arista),\
313                           parametro_de_red_a*ceil(tamano_arista),\
314                            parametro_de_red_a*ceil(tamano_arista)),\
315                          (90, 90, 90))
316 else:
317     v_a, v_b, v_c = vector_abc(1., 1., 1., 90, 90, 120)
318     parametro_de_red_b = parametro_de_red_a
319     archivocar = Crea_car(nombre_sistema,\
320                          (parametro_de_red_a*ceil(tamano_arista),\
321                           parametro_de_red_b*ceil(tamano_arista),\
322                            parametro_de_red_c*ceil(tamano_arista)),\
323                          (90, 90, 120))

324 archivoxyz = Crea_xyz(nombre_sistema)

```

```

325     solutos = []
326     for i in range(num_solutos):
327         while 1:
328             aleatorio = random.randint(0, num_atomos_total-1)

329             if aleatorio not in solutos:
330                 solutos += [aleatorio]
331                 break

332     a_total     = 0
333     a_soluto    = 0
334     a_disolvente = 0
335     frac        = 0.0

336     for x in arange(tamano_arista):
337         for y in arange(tamano_arista):
338             for z in arange(tamano_arista):
339                 for i in sistema_cristalino:
340                     ## hace que los átomos se encuentre dentro de la súper celda
341                     ## y no más allá debido a los vectores de translación

342                     vec = v_a*(x+i[0]-frac + 2*frac*random.random()) \
343                         + v_b*(y+i[1]-frac + 2*frac*random.random()) \
344                         + v_c*(z+i[2]-frac + 2*frac*random.random())

345                     if ((vec[0] >= tamano_arista) or
346                         (vec[1] >= tamano_arista) or
347                         (vec[2] >= tamano_arista)):
348                         continue

349                     vec *= array([parametro_de_red_a, parametro_de_red_b, parametro_de_red_c])

350                     if a_total not in solutos:
351                         a_disolvente += 1
352                         if not sin_disolvente:
353                             archivar.agrega_coordenadas(disolvente, a_disolvente, (vec[0],
354 vec[1], vec[2]))
355                             archivoxyz.agrega_coordenadas(disolvente, (vec[0], vec[1], vec[2]))

356                     else:
357                         a_soluto += 1
358                         if not sin_soluto:
359                             archivar.agrega_coordenadas(soluto, a_soluto, (vec[0], vec[1],
360 vec[2]))
361                             archivoxyz.agrega_coordenadas(soluto, (vec[0], vec[1], vec[2]))

362                     a_total += 1

363     print "Nombre del sistema      : %s"%nombre_sistema
364     print "Sistema base           : %s"%sistema_base

```

```

365     print "Atomos totales           : %d (%d)"%(a_total, num_atomos_total)
366     print "Atomos disolvente (%2s)   : %d (%d)"%(disolvente, a_disolvente, num_atomos_total-
367 num_solutos)
368     print "Atomos soluto      (%2s)   : %d (%d)"%(soluto, a_soluto, num_solutos)
369     print "Concentracion real      : %f (%f)"%(a_soluto/a_total, fraccion)
370     print "Veces celda unitaria     : %f"%ceil(tamano_arista)
371     print "Parametro de red a       : %f A"%parametro_de_red_a
372     print "Parametro de red b       : %f A"%parametro_de_red_b
373     print "Parametro de red c       : %f A"%parametro_de_red_c
374     print "parametro de supercelda a: %f A"%(parametro_de_red_a*tamano_arista)
375     print "parametro de supercelda b: %f A"%(parametro_de_red_b*tamano_arista)
376     print "parametro de supercelda c: %f A"%(parametro_de_red_c*tamano_arista)
377     print "densidad                 : %f gcm-1"calcula_densidad(soluto, disolvente, \
378                                     num_solutos, \
379                                     num_atomos_total-num_solutos,\
380                                     ((parametro_de_red_a*tamano_arista*\
381                                     parametro_de_red_b*tamano_arista*\
382                                     parametro_de_red_c*tamano_arista)*1e-24))

383     parametro_de_red_a*ceil(tamano_arista),\
384     parametro_de_red_b*ceil(tamano_arista),\
385     parametro_de_red_c*ceil(tamano_arista)

386     archivocar.finaliza_archivo()
387     archivoxyz.finaliza_archivo()

388 if __name__ == "__main__":
389     main()

```

A.7 PROGRAMA DVMD

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-

3  __author__ = 'Ulises Santiago <ulises_osa@yahoo.com.mx>'
4  __all__    = []

5  from random import seed, uniform
6  from math import log, sqrt, pi, exp
7  import time
8  import sys
9  import re
10 import getopt

11 ##-----
12 def uso(nombre_programa):
13     texto = "\nuso: " + nombre_programa + " " + "[OPCIONES] archivo_car

```

```

14 DESCRIPCION
15 Programa que genera un conjunto de velocidades que siguen la distribu--
16 cion de probabilidad de Maxwell-Boltzmann.

17 La distribución de probabilidades de Maxwell-Boltzman es funcion de masa
18 de los atomos, por lo que debe indicarse el tipo de atomo que se encuen-
19 tra en la distribuicion. Debe darse el nombre del archivo donde se en---
20 cuentra una lista de los atomos de los que se desea una velocidad alea--
21 toria que cumpla con distribucion de probabilidades de Maxwell-Boltzmann
22 Por ejemplo, si son 3 atomos de silicio y dos de germanio, el archivo
23 debera listarlos como sigue:

24 Si
25 Si
26 Si
27 Ge
28 Ge

29 Las velocidades que se generan apareceran en ese mismo orden. El archivo
30 que se proporciona puede ser un archivo tipo car o cualquier otro en el
31 que al principio de la linea se encuentre un nombre de atomo valido

32 OPCIONES
33 -h --help
34 Muestra esta ayuda y sale.

35 -s --semilla=<numero>
36 Establece la semilla para el generador de numeros aleatorios. El
37 numero puede ser cualquier real positivo o negativo. Si no se
38 especifica se utiliza el reloj del sistema con la función
39 time.time().

40 -t --temperatura=<numero>
41 Temperatura del sistema en kelvin. Por defecto sera 300 K.

42 -c --celeridad_maxima=<numero>
43 Celeridad maxima que podra aparecer en la distribucion. Por
44 defecto es 15000 m/s.

45 -o --archivo_salida=<archivo>
46 Nombre del archivo de salida, por defecto se añade la extension dpmb
47 Si no se da un nombre el nombre por defecto sera el nombre bases del
48 archivo de atomos.

49 -m --unidad_m_s
50 Por defecto las unidades de velocidad son angstrom/ps, si se
51 desean las unidades m/s debe darse esta opcion.

52 ##### no funciona todavía #####
53 -i --todos_iguales=<nombrenumero>

```

```

54         Si todos los atomos son iguales, puede emplearse esta opcion, indi-
55         cando el tipo de atomo y el numero de ellos. Por ejemplo, Si64.
56         ""
57         print texto

```

```

58     ##-----
59     masas_atomicas = {
60         'H' : 1.0079, 'He': 4.0026, 'Li': 6.9410, 'Be': 9.0122,
61         'B' : 10.8110, 'C' : 12.0107, 'N' : 14.0067, 'O' : 15.9994,
62         'F' : 18.9984, 'Ne': 20.1797, 'Na': 22.9898, 'Mg': 24.305 ,
63         'Al': 26.9815, 'Si': 28.0855, 'P' : 30.9738, 'S' : 32.065 ,
64         'Cl': 35.453 , 'Ar': 39.948 , 'K' : 39.0983, 'Ca': 40.078 ,
65         'Sc': 44.9559, 'Ti': 47.867 , 'V' : 50.9415, 'Cr': 51.9961,
66         'Mn': 54.9381, 'Fe': 55.845 , 'Co': 58.93332, 'Ni': 58.6934,
67         'Cu': 63.546 , 'Zn': 65.409 , 'Ga': 69.723 , 'Ge': 72.64 ,
68         'As': 74.9216, 'Se': 78.96 , 'Br': 79.904 , 'Kr': 83.798 ,
69         'Rb': 85.4678, 'Sr': 87.62 , 'Y' : 88.9059, 'Zr': 91.224 ,
70         'Nb': 92.9064, 'Mo': 95.94 , 'Tc': 98 , 'Ru': 101.07 ,
71         'Rh': 102.9055, 'Pd': 106.42 , 'Ag': 107.8682, 'Cd': 112.411 ,
72         'In': 114.818 , 'Sn': 118.71 , 'Sb': 121.76 , 'Te': 127.6 ,
73         'I' : 126.9045, 'Xe': 131.293 , 'Cs': 132.9055, 'Ba': 137.327 ,
74         'La': 138.9055, 'Ce': 140.116 , 'Pr': 140.9077, 'Nd': 144.242 ,
75         'Pm': 145. , 'Sm': 150.36 , 'Eu': 151.964 , 'Gd': 157.25 ,
76         'Tb': 158.9254, 'Dy': 162.5 , 'Ho': 164.9303, 'Er': 167.259 ,
77         'Tm': 168.9342, 'Yb': 173.04 , 'Lu': 174.967 , 'Hf': 178.49 ,
78         'Ta': 180.9479, 'W' : 183.84 , 'Re': 186.207 , 'Os': 190.23 ,
79         'Ir': 192.217 , 'Pt': 195.084 , 'Au': 196.9666, 'Hg': 200.59 ,
80         'Tl': 204.3833, 'Pb': 207.2 , 'Bi': 208.9804, 'Po': 209. ,
81         'At': 210. , 'Rn': 222. , 'Fr': 223. , 'Ra': 226. ,
82         'Ac': 227. , 'Th': 232.0381, 'Pa': 231.0359, 'U' : 238.0289,
83         'Np': 237. , 'Pu': 244. , 'Am': 243. , 'Cm': 247. ,
84         'Bk': 247. , 'Cf': 251. , 'Es': 252. , 'Fm': 257. ,
85         'Md': 258. , 'No': 259. , 'Lr': 262. , 'Rf': 261. ,
86         'Db': 262. , 'Sg': 266. }

```

```

87     ##-----
88     ## constante de boltzmann
89     k = 1.381e-23     ## J/K

```

```

90     ## unidad de masa atómica
91     uma = 1.6606e-27     ## kg

```

```

92     ##-----
93     def vec_unitario():
94         """Crea un vector unitario con dirección aleatoria.
95         """
96         x1 = uniform(-1, 1)
97         x2 = uniform(-1, 1)
98         x3 = uniform(-1, 1)
99         r = sqrt(x1*x1 + x2*x2 + x3*x3)
100        r_ = 1./r
101
102        return x1*r_, x2*r_, x3*r_

```



```

102  ##-----
103  def DPMB_celeridades(T, m, c):
104      """Funcion de distribucion de probabilidad de celeridades de
105      Maxwell-Boltzmann.
106      """
107      return 4*pi*(m/(2*pi*k*T))**(3./2.)*c*c*exp(-m*c*c/(2*k*T))

108  ##-----
109  def celeridad_mas_probable(T, m):
110      """Valor mas probable de la celeridad en la distribución de
111      Maxwell-Boltzmann.
112      """
113      return sqrt(2*k*T/m)

114  ##-----
115  def multi_d_aleatorios_MB(temperatura, num_puntos, masa, max_x=15000):
116      """Da, de forma aleatoria, un conjunto de celeridades que siguen la
117      distribucion de probabilidad de Maxwell-Boltzmann, DPMB, (a una temperatura
118      dada).

119      Para cada celeridad se da un vector unitario de dirección aleatoria. El
120      tamaño del conjunto es igual al numero de 'num_puntos'. La temperatura debe
121      darse en kelvin y la 'masa' en unidades atómicas. La variable 'max_x' es el
122      limite superior del rango de celeridades considerado, el limite inferior es
123      cero.
124      """
125      ## método de aceptación-rechazo (acceptance-rejection)
126      T = temperatura                ## temperatura en kelvin
127      m = masa*uma                   ## masa en kg
128      cmp = celeridad_mas_probable(T, m) ## celeridad más probable
129      y = DPMB_celeridades(T, m, cmp) ## altura máxima de la curva de la DPMB
130      celeridades = []              ## lista de celeridades que cumplen con
131      ## la DPMB

132      ## Este ciclo se repite hasta completar el número de 'puntos' dentro de la
133      ## distribución.
134      cont = 1
135      while cont <= num_puntos:
136          aleatorio_x = uniform(0, max_x) ## escoge una celeridad de forma uniforme
137          ## en el rango [0, max_x)
138          aleatorio_y = uniform(0, y) ## escoge una probabilidad de forma uniforme
139          ## en el rango [0, y), siendo 'y' la altura
140          ## máxima de la DPMB.
141          ## Con estas dos variables, aleatorio_x
142          ## aleatorio_y, se escoge, de forma aleatoria
143          ## y uniforme, un punto dentro del rectángulo:
144          ## [0, 0]--[0, y)--(max_x, y)--(max_x, 0]

145      ## Si el punto escogido en la parte anterior se encuentra debajo o sobre
146      ## la curva de la DPMB. Es uno de los puntos buscados
147      if aleatorio_y <= DPMB_celeridades(T, m, aleatorio_x):
148          celeridad += [aleatorio_x]
149          cont += 1

```

```

150     return celeridades

151  ##-----
152  def mono_d_aleatorios_MB(temperatura, masa, max_x=15000):
153      """Da, de forma aleatoria, una celeridad que siguen la distribucion de
154      probabilidad de Maxwell-Boltzmann, DPMB, (a una temperatura dada).

155      La temperatura debe darse en kelvin y la 'masa' en unidades atomicas. La
156      variable 'max_x' es el limite superior del rango de celeridades considerado,
157      el limite inferior es cero. La celeridad dada tiene unidades de m/s.
158      """
159      ## método de aceptación-rechazo (acceptance-rejection)
160      T = temperatura          ## temperatura en kelvin
161      m = masa*uma            ## masa en kg
162      cmp = celeridad_mas_probable(T, m)  ## celeridad más probable
163      y = DPMB_celeridades(T, m, cmp)    ## altura máxima de la curva de la DPMB

164      ## El ciclo para cuando se encuentra una celeridad dentro de la distribución
165      while True:
166          aleatorio_x = uniform(0, max_x) ## escoge una celeridad de forma uniforme
167                                  ## en el rango [0, max_x)
168          aleatorio_y = uniform(0, y) ## escoge una probabilidad de forma uniforme
169                                  ## en el rango [0, y), siendo 'y' la altura
170                                  ## máxima de la DPMB.
171                                  ## Con estas dos variables, aleatorio_x
172                                  ## aleatorio_y, se escoge, de forma aleatoria
173                                  ## y uniforme, un punto dentro del rectángulo:
174                                  ## [0, 0]--[0, y)--(max_x, y)--(max_x, 0]

175          ## Si el punto escogido en la parte anterior se encuentra debajo o sobre
176          ## la curva de la DPMB, el ciclo para. Es el punto buscado.
177          if aleatorio_y <= DPMB_celeridades(T, m, aleatorio_x):
178              break

179      return aleatorio_x

180  ##-----
181  def nombre_base():
182      """Da el nombre base del sistema.
183      """
184      nombre = sys.argv[-1]
185      nombre = re.match(".*(?:=\car)|.*", nombre, re.IGNORECASE).group(0)

186      return nombre

187  ##-----
188  def atomos(nombre_archivo):
189      """Lee los nombres de los atomos del archivo. Devuelve una lista.
190      """
191      ## lista de átomos
192      lista_atomos = []

```

```

193     ## Abre el archivo en modo de sólo lectura.
194     archivo_entrada = open(nombre_archivo, "r")

195     ## patrón de un átomo
196     patron_atomo = re.compile('^[A-Z][a-z]?(?=[\d+|\s+])')

197     for linea in archivo_entrada:
198         ## busca el patron EleNumXYZ en la línea
199         patron_encontrado = re.match(patron_atomo, linea)
200         ## si se encuentra el patrón en la línea
201         if patron_encontrado != None:
202             atomo = patron_encontrado.group(1)
203             lista_atomos += [atomo]

204     archivo_entrada.close()

205     return lista_atomos

206     ##-----
207     def main():
208         """Función principal.
209         """
210         ## argumento, ver documentación del módulo 'getopt'
211         try:
212             opciones, argumentos = getopt.getopt(sys.argv[1:], \
213                 "hmt:c:o:s:i:", ["help", "unidad_m_s", "temperatura=",
214                 "celeridad_maxima=", "archivo_salida=",
215                 "semilla=", 'todos_iguales='])
216         except getopt.error, msg:
217             print msg
218             print "Para ayuda use --help"
219             sys.exit(2)

220         ## si se elije la opción de ayuda
221         for (opcion, valor) in opciones:
222             if opcion in ("-h", "--help"):
223                 uso(sys.argv[0])
224                 sys.exit(0)

225         ## valores por defecto
226         ## nombre base del archivo del salida
227         nombre_archivo_salida = nombre_base()
228         max_x                = 15000        ## celeridad máxima
229         temperatura          = 300          ## 300 K
230         semilla              = time.time()  ## tiempo del sistema
231         unidad_m_s           = 0           ## La unidad de celeridad sera de
232                                 ## angstrom/ps
233         todos_iguales        = 0           ## si todos los atomos son iguales
234         factor = 0.01        ## factor de conversión de m/s a
235                                 ## angstrom/ps

```

```

236 # procesamiento de opciones
237 for (opcion, valor) in opciones:
238     if opcion in ("-o", "--archivo_salida"):
239         nombre_archivo_salida = valor
240     elif opcion in ("-c", "--celeridad_maxima"):
241         max_x = float(valor)
242     elif opcion in ("-t", "--temperatura"):
243         temperatura = float(valor)
244     elif opcion in ("-s", "--semilla"):
245         semilla = float(valor)
246     elif opcion in ("-m", "--unidad_m_s"):
247         unidad_m_s = 1
248         factor = 1
249     elif opcion in ("-i", "--todos_iguales"):
250         todos_iguales = 1
251         patron_atomo = re.compile('[A-Z][a-z]?(\d+)')
252         patron_encontrado = re.match(patron_atomo, valor)
253         try:
254             tipo = patron_encontrado.group(1)
255             n_atomos = int(patron_encontrado.group(2))
256         except:
257             print "Tipo de atomo o numero no reconocido"
258             sys.exit(1)
259     else:
260         assert False, "Opcion inmanejable: " + opcion

261 ## establece la semilla del generador de números aleatorios
262 seed(semilla)

263 # procesamiento de argumentos
264 nombre_archivo = argumentos[0]
265 lista_atomos = atomos(nombre_archivo)

266 ## lista de celeridades
267 celeridades = []

268 for atomo in lista_atomos:
269     celeridad = mono_d_aleatorios_MB(temperatura, \
270                                     masas_atomicas[atomo], \
271                                     max_x)
272     celeridades += [celeridad]

273 archivo_salida = open(nombre_archivo_salida + ".dvmb", 'w')

274 ## cabecera del archivo de salida
275 archivo_salida.write('# Velocidades según la distribución de Maxwell-Boltzmann\n')
276 archivo_salida.write('# random number seed : ' + str(repr(semilla)) + '\n')
277 archivo_salida.write('# número de átomos : ' + str(len(lista_atomos)) + '\n')
278 archivo_salida.write('# temperatura : ' + str(temperatura) + ' K\n')
279 archivo_salida.write('# velocidad máxima : ' + str(max_x) + ' m/s'
280                       + ' (' + str(max_x*0.01) + ' angstrom/ps)\n')
281 archivo_salida.write("# %-4s%15s %15s %15s\n"%(

```

```

282         'num', '|V|', 'V_x', 'V_y', 'V_z')
283     )
284     archivo_salida.write('# MD_Velocity User_Defined %s\n'%str(len(lista_atomos)))

285     cont = 1
286     for celeridad in celeridades:
287         v_unitario = vec_unitario()
288         archivo_salida.write("%-4d %15.6f %15.6f %15.6f %15.6f  # %s\n"%(
289             cont, celeridad*factor,
290             v_unitario[0],
291             v_unitario[1], v_unitario[2],
292             lista_atomos[cont-1])
293         )
294     cont += 1

295 ##-----
296 if __name__ == '__main__':
297     main()

```

A.8 PROGRAMA PDF

```

1  #include <string.h>
2  #include <math.h>
3  #include <regex.h> /* expresiones regulares */

4  # define PI 3.141592653589793238462643383

5  const char* nombre_programa;
6  char *nombre_base(char * );
7  void imprime_uso(void);

8  int main (int argc, char **argv){

9      int siguiente_opcion;          /*variable ocupada en getopt*/

10     FILE *archivo_entrada;         /* archivo de coordenadas */
11     float vec[3];                  /* dimensiones de la celda */

12     int n_atomos = 0;              /* número de átomos */
13     float dr = 0.05;               /* tamaño de intervalo de medición */
14     float maximo = 0.;             /* el valor máximo los vectores de red */
15     float faux = 0.0;              /* variable auxiliar */
16     float volumen = 0.0;           /* almacena el volumen de la celda */
17     float vaux[3] = {0, 0, 0};     /* un vector auxiliar */
18     float inv_dr;                  /* inverso de la variable dr */

```

```

19     float inv_rho_4pi, rho_4pi, rho; /* variables auxiliares */
20     int i, j, iaux;                 /* más variables auxiliares */
21     char *nom_archivo;              /* nombre del archivo de coodenadas */
22     char *nom_base;                 /* nombre base del achivo de coordendas */

23     /*número de intervalos para las intensidades*/
24     int n_intensidades;
25     /*Almacena las coordenadas de las 27 celdas, variable dinámica*/
26     float (*coor_ext)[3];
27     /*Almacena las intesidades de cada distancia, variable dinámica*/
28     int *intensidades;
29     /*coordenadas originales, variable dinámica*/
30     float (*coordenadas)[3];

31     /*****OPCIONES*****/
32     /* Una cadena que lista las opciones cortas válidas */
33     const char* const op_cortas = "ho:a:b:c:n:d:" ;

34     /* Una estructura de varios arrays describiendo los valores largos */
35     const struct option op_largas[] =
36     {
37     { "help",          0,  NULL,  'h'},
38     { "output",        1,  NULL,  'o'},
39     { "tamanio_de_arista_a",1,  NULL,  'a'},
40     { "tamanio_de_arista_b",1,  NULL,  'b'},
41     { "tamanio_de_arista_c",1,  NULL,  'c'},
42     { "numero_de_atomos",  1,  NULL,  'n'},
43     { "dr",             1,  NULL,  'd'},
44     { NULL,            0,  NULL,  0  }
45     };

46     /* El nombre del fichero que recibe la salida del programa */
47     const char* fichero_salida = NULL ;

48     /* Guardar el nombre del programa para incluirlo a la salida */
49     nombre_programa = argv[0];

50     /* Si se ejecuta sin parámetros ni opciones */
51     if (argc == 1){
52         imprime_uso();
53         exit(EXIT_SUCCESS);
54     }

55     while (1){
56         /* Llamamos a la función getopt */
57         siguiente_opcion = getopt_long (argc, argv, op_cortas, op_largas, NULL);

58         if (siguiente_opcion == -1)
59             break; /* No hay más opciones. Rompemos el bucle */

```

```

60     switch (siguiente_opcion){
61         case 'h' : /* -h o --help */
62             imprime_uso();
63             exit(EXIT_SUCCESS);

64         case 'a' : /* -a o --tamano_de_arista_a */
65             faux = atof(optarg) ;
66             vec[0] = faux;
67             break;

68         case 'b' : /* -b o --tamano_de_arista_b */
69             faux = atof(optarg) ;
70             vec[1] = faux;
71             break;

72         case 'c' : /* -c o --tamano_de_arista_c */
73             faux = atof(optarg) ;
74             vec[2] = faux;
75             break;

76         case 'n' : /* -n o --numero_de_atomos */
77             n_atomos = atoi(optarg) ;
78             break;

79         case 'd' : /* -d o --dr */
80             dr = atof(optarg) ;
81             break;

82         case 'o' : /* -o ó -output */
83             fichero_salida = optarg; /* optarg contiene el argumento de -o */
84             break;

85         case '?' : /* opción no valida */
86             imprime_uso(); /* código de salida 1 */
87             exit(1);

88         case -1 : /* No hay más opciones */
89             break;

90         default : /* Algo más? No esperado. Abortamos */
91             abort();
92     }
93 }

94 /* Si no se da o es cero una de las coordenadas */
95 for ( i=0; i<3; i++){
96     if (vec[i] > maximo){
97         maximo = vec[i];
98     }
99 }

```

```

100  /* si no se proporciona ningun tamaño de celda */
101  if (maximo <= 0){
102      fprintf(stderr, "Dimensiones de la celda inválidas.\n");
103      exit (EXIT_FAILURE);
104  }

105  /* si alguno de los valores de la celda no se proporciona se llena con
106   * se asigna el valor máximo de los sí proporcionados */
107  for ( i=0; i<3; i++){
108      if (vec[i] <= 0 ){
109          vec[i] = maximo;
110      }
111  }

112  /*****TERMINAN OPCIONES*****/

113  /**/
114  if (n_atomos==0){
115      fprintf(stderr, "Mal número de atomos\n");
116      exit(EXIT_FAILURE);
117  }

118  inv_dr= 1./dr;
119  n_intensidades = (int) (vec[0]*inv_dr);
120  coor_ext      = calloc(n_atomos*27*3, sizeof(float));
121  intensidades  = calloc(n_intensidades, sizeof(int));
122  coordenadas   = calloc(n_atomos*3, sizeof(float));
123  nom_archivo   = argv[argc-1];
124  nom_base      = nombre_base(nom_archivo);
125  volumen       = vec[0]*vec[1]*vec[2];

126  //printf("%d %f %f %s\n", n_intensidades, inv_dr, volumen, nom_base );

127  for (j=0; j<3; j++){
128      for (i=0; i<n_atomos; i++){
129          coordenadas[i][j] = 0.0;
130      }
131      for (i=0; i<n_atomos*27; i++){
132          coor_ext[i][j] = 0.0;
133      }
134      vaux[j] = 0.0;
135  }

136  for (i=0; i< n_intensidades; i++){
137      intensidades[i] = 0;
138  }

139  if ((archivo_entrada =
140      fopen(nom_archivo, "r")) == NULL){
141      fprintf(stderr, "No se puede abrir el archivo.\n");

```



```

142     exit (EXIT_FAILURE);
143 }

144 j= 0;
145 while (!feof(archivo_entrada)){
146     fscanf(archivo_entrada, "%f%f%f",
147         &coordenadas[j][0],
148         &coordenadas[j][1],
149         &coordenadas[j][2]
150     );
151     j += 1;
152 }

153 fclose(archivo_entrada);

154 /*extendiende la celda en todas las direcciones*/
155 j = 0;
156 for (i=0; i<n_atomos; i++){
157     for (int x=-1; x<2; x++){
158         for (int y=-1; y<2; y++){
159             for (int z=-1; z<2; z++){
160                 coor_ext[j][0] = coordenadas[i][0] + vec[0]*x;
161                 coor_ext[j][1] = coordenadas[i][1] + vec[1]*y;
162                 coor_ext[j][2] = coordenadas[i][2] + vec[2]*z;
163                 j += 1;
164             }
165         }
166     }
167 }

168 for( i=0; i<n_atomos; i++){
169     for (j=0; j<n_atomos*27; j++){
170         vaux[0] = coordenadas[i][0]-coor_ext[j][0];
171         vaux[1] = coordenadas[i][1]-coor_ext[j][1];
172         vaux[2] = coordenadas[i][2]-coor_ext[j][2];
173         faux = sqrt(vaux[0]*vaux[0] + vaux[1]*vaux[1] + vaux[2]*vaux[2]);
174         if (faux < vec[0]){
175             iaux = (int)(faux*inv_dr);
176             intensidades[iaux] += 1;
177         }
178     }
179 }

180 //rho = n_atomos/volumen;
181 //inv_rho_4pi = 1./(4*PI*rho*n_atomos*(dr*dr)*dr);
182 inv_rho_4pi = volumen/(4*PI*n_atomos*n_atomos*(dr*dr)*dr);

183 for (i=1; i< n_intensidades; i++){
184     faux = intensidades[i] * inv_rho_4pi / (i*i);
185     printf("%f %f\n", dr*i+dr*0.5, faux );
186 }

```

```

187     return 0;
188 }

189 void imprime_uso ()
190 {
191     printf("Uso: %s opciones [ argumentos ...]\n", nombre_programa);
192     printf("    -h --help           Muestra esta ayuda\n"
193           "    -o --output_archivo   Escribe la salida al archivo\n"
194           "    -a --tamano_de_arista_a Tamaño de la arista\n"
195           "    -b --tamano_de_arista_b Tamaño de la arista\n"
196           "    -c --tamano_de_arista_c Tamaño de la arista\n"
197           "    -n --numero_de_atomos  Número de átomos\n"
198           "    -d --dr                Tamaño del intervalo de muestreo\n"
199           );
200 }

201 char *nombre_base(char * nombre){
202     /*
203      *Función para determinar el nombre base de un archivo
204      */
205     int longitud; /*longitud de la cadena*/
206     int aux = 0;
207     int i;
208     char *seed_name='\0';

209     longitud = strlen(nombre);

210     /*en caso de no encontrar un punto, copia la cadena completa*/
211     aux = longitud;

212     /*recorre la cadena de forma inversa y para hasta encontrar un punto*/
213     for (i=longitud; i>0; i-- ){
214         if (*(nombre + i) == '.'){
215             aux = i;
216             break;
217         }
218     }

219     /*asigna memoria adecuada*/
220     seed_name = calloc(aux+1, sizeof(char));

221     for (i=0; i<aux; i++){
222         *(seed_name + i) = *(nombre + i);
223     }

224     return seed_name;
225 }

226 int lee_archivo(float *a, float *b, float *c, float *alfa, float *beta, \
227               float *gamma, float (*coordenadas)[3]){

```

```

228     return 0;
229 }

```

A.9 PROGRAMA ANGULOS

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-

3  __author__ = 'Ulises Santiago <ulises_osa@yahoo.com.mx>'
4  __date__   = "Enero2009"

5  from math import acos
6  from numpy import *
7  import re
8  import getopt
9  import sys
10 import time

11 ##-----
12 def uso(nombre_programa):
13     texto = "\nuso: " + nombre_programa + " " [OPCIONES] archivo_car

14     DESCRIPCION
15     Programa para calcular los angulos planos entre los atomos de una
16     supercelda. Considera condiciones periodicas.

17     OPCIONES
18     -h --help
19         Muestra esta ayuda y sale.

20     -o --archivo-salida=<nombre_del_archivo_de_salida>
21         Establece el nombre del archivo de salida, si esta opcion no se
22         utiliza, el nombre del archivo por de fecto sera el nombre base del
23         archivo car.

24     -r --distancia_de_enlace=<distancia_del_enlace>
25         Distancia maxima a la cual se considera que dos atomos estan en-
26         lazados. El valor por defecto será de 1.55.

27     -e --archivo_distancias_enlace=<archivo>
28         Cuando existe mas de un tipo de atomo en el archivo car, puede
29         definirse un archivo de distancias de enlace. Por ejemplo, el --
30         archo que puede emplearse para el sistema C, Si, Mg debera con-
31         tener la siguiente informacion:

```

```

32             Si Si  2.29
33             Mg Mg  2.85
34             C  C   1.55
35             C  Si  1.86
36             Si Mg  2.57
37             C  Mg  2.07

38             si en el archivo no se especifican todos los enlaces encontrados
39             por el programa en el archivo car, la distancia empleada sera la
40             dada en la opcion -r o el valor por defecto (1.55).

41             """

42             print texto
43             ##-----

44             class A_donde:
45                 """A dónde debe trasladarse un átomo para cumplir con la periodicidad.

46                 La coordenada trasladadas son del átomo 'otro'.
47                 """
48                 def __init__(self, abc, radio, origen, otro):
49                     self.__abc      = abc          # Tupla de dimensiones de celda.
50                     self.__radio    = radio        # Radio de corte
51                     self.__origen    = origen       # Coordenada fija, átomo central
52                     self.__otro     = otro         # Coordenada del átomo a trasladar
53                     self.__xyz      = [0., 0., 0.]  # dirección de traslado para cada
54                                                         # eje. Los valores posibles para
55                                                         # cada coordenada son: -1, 0, 1.
56                     self.__coor_xyz = self.__a_donde(0)

57             def __a_donde(self, num):
58                 """Determina si el átomo 'otro' se encuentran alrededor del átomo central.

59                 Las coordenadas del átomo central se encuentran en '__origen'.
60                 """
61                 # La profundidad máxima de recursión es 3.
62                 # num = 0, 1, 2 corresponde a las coordenadas x, y, z respectivamente
63                 if num < 3:
64                     #
65                     if self.__otro[num] - self.__radio <= self.__origen[num] <= \
66                        self.__otro[num] + self.__radio:
67                         return self.__a_donde(num+1)

68                 elif self.__otro[num] - self.__radio <= self.__origen[num] + \
69                    self.__abc[num] <= self.__otro[num] + self.__radio:
70                     self.__xyz[num] -= 1.
71                     return self.__a_donde(num+1)

```

```

72         elif self.__otro[num] - self.__radio <= self.__origen[num] - \
73             self.__abc[num] <= self.__otro[num] + self.__radio:
74             self.__xyz[num] += 1.
75             return self.__a_donde(num + 1)

76         else:
77             return None
78     else:
79         x = self.__otro[0] + self.__xyz[0] * self.__abc[0]
80         y = self.__otro[1] + self.__xyz[1] * self.__abc[1]
81         z = self.__otro[2] + self.__xyz[2] * self.__abc[2]

82     return array([x, y, z])

83     def XYZ(self):
84         return self.__coor_xyz

85     ##-----
86     def distancia_vec(vector_1, vector_2):
87         """Determina la distancia entre dos vectores
88         """
89         vector = vector_2 - vector_1
90         return sqrt(vdot(vector, vector))
91     ##-----
92     def enlazados(coordenada_1, coordenada_2, d_maxima):
93         """¿Estas dos coordenadas pueden considerarse enlazadas?
94         """
95         if distancia_vec(coordenada_1, coordenada_2) <= d_maxima:
96             return True
97         else:
98             return False
99     ##-----

100     def calcula_angulo(v1, v2, v3, grados = True):
101         """Calcula el ángulo entre los vectores a-b y c-b.
102         """
103         argumento = vdot(v1-v2 ,v3-v2) / (distancia_vec(v1,v2)*distancia_vec(v3,v2))

104         # por error de redondeo el valor absoluto del argumento puede ser ligeramente
105         # mayor a uno, en esta sección esto se corrige.
106         if not abs(argumento) < 1:
107             if argumento < -1: argumento = -1
108             elif argumento > 1: argumento = 1

109         angulo = acos( argumento )

110         # si el resultado debe darse en grados
111         if grados:
112             return 180.0/pi * angulo
113         # el resultado se da en radianes
114         else:

```

```

115         return angulo
116     ##-----
117     def triadas_angulos(dic, central, triadas):
118         """Calcula los ángulos entre todos los átomos enlazados con
119         un átomo central.
120         """
121         origen = dic[central[0]][central[1]]["original"]
122         tipos_atomos = dic[central[0]][central[1]]["adyacentes"].keys()
123         adyacentes = []

124         for tipo in tipos_atomos:
125             for num in dic[central[0]][central[1]]["adyacentes"][tipo].keys():
126                 vec = dic[central[0]][central[1]]["adyacentes"][tipo][num]
127                 adyacentes += [(tipo, vec)]

128         n = len(adyacentes)

129         if n < 2: return 1

130         # para cada átomo enlazado con el átomo central
131         for izquierda in range(0, n):
132             # selecciona otro átomo de la misma lista, sin repetir
133             for derecha in range(izquierda+1, n):
134                 aux = [adyacentes[izquierda][0], adyacentes[derecha][0]]
135                 aux.sort()
136                 nombre_triada = aux[0] + '-' + central[0] + '-' + aux[1]
137                 angulo = calcula_angulo(adyacentes[izquierda][1], origen, \
138                 adyacentes[derecha][1] )

139                 if triadas.has_key(nombre_triada):
140                     triadas[nombre_triada] += [angulo]
141                 else:
142                     triadas[nombre_triada] = [angulo]

143     ##-----

144     def administra_coordenadas(nombre_archivo):
145         """Lee las coordenadas atómicas del archivo tipo CAR.

146         Determina el tipo y número de cada átomo presente en el archivo CAR. Crea un
147         diccionario de diccionarios donde la primera entrada corresponde al tipo y -
148         la segunda al número de átomo, los valores almacenados son las coordenadas.
149         """

150         ## Abre el archivo ARC en modo de sólo lectura.
151         archivo_entrada = open(nombre_archivo, "r")

152         ### Expresión regular para un número real
153         #num_real = '([+-]?\.?\d+\.\d*(?:[eE] [+-]?\d+)?)'

```

```

154     ## Expresión regular para encontrar un elemento químico seguido de un número
155     ## y tres números reales (coordenadas) separados por espacios.
156     ## Esta expresión regular generará 5 grupos de búsqueda:
157     ##     grupo 1: nombre del elemento químico
158     ##     grupo 2: número del elemento químico
159     ##     grupo 3: coordenada x
160     ##     grupo 4: coordenada y
161     ##     grupo 5: coordenada z
162     patron_EleNumXYZ = re.compile('^[A-Z][a-zA-Z?](\d+)' + \
163     '\s+([-+]?\.\d+\.\d*(?:[eE][-+]?[d+])?|\*)' + \
164     '\s+([-+]?\.\d+\.\d*(?:[eE][-+]?[d+])?|\*)' + \
165     '\s+([-+]?\.\d+\.\d*(?:[eE][-+]?[d+])?|\*)')
166     ## El patrón anterior contempla la explosión de la celda, en tal caso
167     ## un(os) asterisco(s) deberá(n) aparecer en lugar de la coordenada

168     ## patrón encontrar los parámetros de la celda, a, b, c, alfa, beta, gama
169     patron_parametros_celda = re.compile('^[!PBC]' + \
170     '\s+([-+]?\.\d+\.\d*(?:[eE][-+]?[d+])?|\*)' + \
171     '\s+([-+]?\.\d+\.\d*(?:[eE][-+]?[d+])?|\*)' + \
172     '\s+([-+]?\.\d+\.\d*(?:[eE][-+]?[d+])?|\*)' + \
173     '\s+([-+]?\.\d+\.\d*(?:[eE][-+]?[d+])?|\*)' + \
174     '\s+([-+]?\.\d+\.\d*(?:[eE][-+]?[d+])?|\*)' + \
175     '\s+([-+]?\.\d+\.\d*(?:[eE][-+]?[d+])?|\*)')

176     ## Diccionario principal
177     posicion_atomos = {}

178     ## Lee una por una las líneas del archivo de entrada
179     for linea in archivo_entrada:
180         ## busca el patron EleNumXYZ en la línea
181         patron_encontrado = re.match(patron_EleNumXYZ, linea)
182         ## si se encuentra el patrón en la línea
183         if patron_encontrado != None:
184             elemento = patron_encontrado.group(1)
185             numero = patron_encontrado.group(2)
186             X = patron_encontrado.group(3)
187             Y = patron_encontrado.group(4)
188             Z = patron_encontrado.group(5)

189             ## crea un diccionario para almacenar lo números de átomo y sus
190             ## coordenadas
191             if posicion_atomos.has_key(elemento): pass
192             else: posicion_atomos[elemento] = {}

193             ## coordenada
194             posicion_atomos[elemento][int(numero)] \
195                 = {'original' : array([float(X), float(Y), float(Z)]),
196                   'adyacentes': {}}
197         }
198     # si el patrón no se encontró en la línea
199     else:
200         parametros_celda = re.match(patron_parametros_celda, linea)
201         # si se encuentran los parámetros de la celda.

```

```

202         if parametros_celda != None:
203             a     = float(parametros_celda.group(1))
204             b     = float(parametros_celda.group(2))
205             c     = float(parametros_celda.group(3))
206             alfa  = float(parametros_celda.group(4))
207             beta  = float(parametros_celda.group(5))
208             gama  = float(parametros_celda.group(6))

209     archivo_entrada.close()
210     print "Archivo CAR leído..."

211     return posicion_atomos, a, b, c, alfa, beta, gama
212     ##-----
213     def tipos_de_atomos(posiciones):
214         """Regresa una lista con los nombres de los tipos de átomos presentes en el
215         archivo CAR.
216         """
217         return posiciones.keys()

218     ##-----
219     def cantidad_de_cada_tipo(tipos_atomos, posicion_atomos):
220         """Determina cuántos átomos hay de cada tipo.
221         """
222         num_atomos = []

223         for tipo_atomo in tipos_atomos:
224             num_aux = len(posicion_atomos[tipo_atomo].keys())
225             num_atomos += [num_aux]

226         return num_atomos

227     ##-----
228     def numero_total_de_atomos(atomos_de_cada_tipo):
229         """Da el número total de átomos considerados.
230         """
231         return sum(atomos_de_cada_tipo)

232     ##-----
233     def nombre_base():
234         """Da el nombre base del sistema.
235         """
236         nombre = sys.argv[-1]
237         nombre = re.match(".*(?=\car)|.*", nombre, re.IGNORECASE).group(0)

238         return nombre

239     ##-----
240     def main():
241         """Función principal.
242         """

```



```

243     ## argumento, ver documentación del módulo 'getopt'
244     try:
245         opciones, argumentos = getopt.getopt(sys.argv[1:], \
246             "ho:r:e:c:", ["help", "archivo_salida=", \
247                 "distancia_de_enlace=", "archivo_distancias_enlace=",
248                 "ocupar-nombre-base="])
249     except getopt.error, msg:
250         print msg
251         print "Para ayuda use --help"
252         sys.exit(2)

253     ## si se elije la opción de ayuda
254     for (opcion, valor) in opciones:
255         if opcion in ("-h", "--help"):
256             uso(sys.argv[0])
257             sys.exit(0)

258     ## valores por defecto
259     ## condiciones_periodicas = 1             #True
260     nombre_archivo_salida = nombre_base() # nombre base del archivo del entrada
261     distancia_de_enlace   = 1.55
262     distancias_de_enlace = {}
263     distancias = []

264     # procesamiento de opciones
265     for (opcion, valor) in opciones:
266         if opcion in ("-o", "--archivo_salida"):
267             nombre_archivo_salida = valor
268             #elif opcion in ("-p", "--condiciones_periodicas"):
269             #    condiciones_periodicas = int(valor)
270         elif opcion in ("-r", "--distancia_de_enlace"):
271             distancia_de_enlace = float(valor)

272         elif opcion in ("-e", "--archivo_distancias_enlace"):

273             enlaces = open(valor, 'r')

274             ## crea las entradas para los enlaces. Existen dos entradas para
275             ## cada pareja que forman un enlace, por ejemplo, si en el archivo
276             ## se registra Si - Mg 2.57, puede accederse a esta información
277             ## mediante [Si][Mg] o bien [Mg][Si].
278             for linea in enlaces:
279                 entrada = linea.split()
280                 distancias_de_enlace[entrada[0]] = {}
281                 distancias_de_enlace[entrada[1]] = {}

282             enlaces.close()
283             enlaces = open(valor, 'r')

284             ## obtiene la información del enlaces
285             for linea in enlaces:

```

```

286         entrada = linea.split()
287         distancia = float(entrada[2])
288         distancias_de_enlace[entrada[0]][entrada[1]] = distancia
289         distancias_de_enlace[entrada[1]][entrada[0]] = distancia

290         distancias += [distancia]

291     enlaces.close()

292     else:
293         assert False, "Opcion inmanejable: " + opcion

294     # procesamiento de argumentos
295     nombre_archivo = argumentos[0]
296     posicion_atomos, a, b, c, alfa, beta, gama = \
297         administra_coordenadas(nombre_archivo)
298     tipos_atomos = tipos_de_atomos(posicion_atomos)
299     num_tipos_atomos = len(tipos_atomos)
300     num_atomos = cantidad_de_cada_tipo(tipos_atomos, posicion_atomos)
301     num_atomos_total = numero_total_de_atomos(num_atomos)
302     distancias += [distancia_de_enlace]
303     max_distancia_enlace = max(distancias)
304     print a,b,c

305     ## Éstas son "dos parejas" de ciclos que recoren, cada una, el tipo y el nú-
306     ## mero de átomo, de esta forma cada átomo se compara con todos los demás.
307     for atomo in tipos_atomos:
308         for numero in posicion_atomos[atomo].keys():
309             for atomo_o in tipos_atomos:
310                 for numero_o in posicion_atomos[atomo_o].keys():
311                     try:
312                         ## si se dieron los enlaces a través de un archivo
313                         radio = distancias_de_enlace[atomo][atomo_o]
314                     except KeyError:
315                         ## el valor por defecto
316                         radio = distancia_de_enlace

317     ## Determina si un átomo "puede" estar enlazado a otro de
318     ## forma directa o mediante una translación en la supercelda
319     ## (condiciones periódicas) genera las coordenadas de tal -
320     ## translación.
321     cambio_coor = A_donde((a,b,c), max_distancia_enlace, \
322         posicion_atomos[atomo][numero]['original'], \
323         posicion_atomos[atomo_o][numero_o]['original'])

324     ## Si están enlazados los dos átomos, sin importar que sea
325     ## a través de una translación
326     if cambio_coor.XYZ() != None:
327         if enlazados(posicion_atomos[atomo][numero]['original'],\
328             cambio_coor.XYZ(), radio): #distancia_de_enlace):
329             x, y, z = cambio_coor.XYZ()

```

```

330             ## comprueba si el átomo no hace referencia a sí mismo
331             hacer = True
332             if numero == numero_o and atomo == atomo_o:
333                 hacer = False

334             ## sino es él mismo
335             if hacer:
336                 ## registra el átomo con el que se enlaza
337                 if posicion_atomos[atomo][numero]\
338                     ["adyacentes"].has_key(atomo_o):
339                     posicion_atomos[atomo][numero]["adyacentes"] \
340                         [atomo_o][numero_o] = array([x, y, z])
341                 else:
342                     posicion_atomos[atomo][numero]["adyacentes"] \
343                         [atomo_o] = {numero_o: array([x, y, z])}

344             ## Es una variable de almacenamiento, se pasa por referencia a la función
345             ## triadas_angulos. Las triadas sólo son de "tipos" de átomo.
346             triadas = {}
347             for atomo in tipos_atomos:
348                 for numero in posicion_atomos[atomo].keys():
349                     triadas_angulos(posicion_atomos, (atomo,numero), triadas)

350             ## crea un diccionario que almacenará achivos, principal y los de las difen-
351             ## tes triadas
352             archivos = {'total' : open(nombre_archivo_salida + '-total.txt', "w" )}

353             ## cabecera del archivo total
354             archivos['total'].write("# ANGULOS PLANOS\n")
355             archivos['total'].write("# numero y tipo de atomos\n")
356             cont = 0
357             for i in tipos_atomos:
358                 archivos['total'].write("#\t%-2s\t%d\n"%(i, num_atomos[cont]))
359                 cont += 1

360             ## archivos para las diferentes triadas
361             for i in triadas :
362                 archivos[i] = open(nombre_archivo_salida + '-' + i + '.txt', "w" )
363                 archivos[i].write("# ANGULOS PLANOS\n")
364                 archivos[i].write("# triada      " + i )
365                 archivos[i].write("\n# distancia de enlace\n" )

366             elementos = i.split('-')
367             #elementos.sort()

368             try:
369                 radio = distancias_de_enlace[elementos[0]][elementos[1]]
370             except KeyError:
371                 radio = distancia_de_enlace
372             archivos[i].write("#\t%2s - %s\t\t%f\n"%(elementos[0],elementos[1], radio))

```

```

373     try:
374         radio = distancias_de_enlace[elementos[2]][elementos[1]]
375     except KeyError:
376         radio = distancia_de_enlace
377     archivos[i].write("#\t%-2s - %-2s\t\t%f\n"%(elementos[2], elementos[1], radio))

378     for i in triadas.keys():
379         for j in triadas[i]:
380             archivos[i].write( str(j) +'\n' )
381             archivos['total'].write( str(j) + '\n' )

382     archivos[i].close()

383     archivos['total'].close()

384     _archivos = archivos.keys()
385     _archivos.sort()
386     print "Archivos generados:"
387     for archivo in _archivos:
388         print "\t%s-%s.txt"%(nombre_archivo_salida, archivo)

389 if __name__ == '__main__':
390     t1 = time.clock()
391     main()
392     t2 = time.clock()
393     print t2-t1, 's'

```

A.10 PROGRAMA MSD

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-

3  __author__ = 'Ulises Santiago <ulises_osa@yahoo.com.mx>'

4  from numpy import *
5  import re
6  import getopt
7  import sys

8  ##-----
9  def uso(nombre_programa):
10     texto = "uso: " + nombre_programa + " " [OPCIONES] archivo_arc

```

```

11     DESCRIPCION
12     Programa para calcular el despalazamiento cuadratico medio (msd) para un
13     sistema de n elementos constituyentes.

14     OPCIONES
15     -h --help
16         Muestra esta ayuda y sale.

17     -o --archivo-salida=<nombre_del_archivo_de_salida>
18         Establece el nombre del archivo de salida, si esta opcion no se
19         utiliza el nombre del archivo por defecto sera 'msd.salida'.

20     -i --paso_inicial=<numero_de_paso_inicial>
21         Define el numero de paso inicial desde el cual comenzaran ha ---
22         realizarse los calculos. Si no se especifica, el numero de paso
23         inicial sera '0'.

24     -f --paso_final=<numero_de_paso_final>
25         Define el numero de paso final hasta el cual han de realizarse
26         los calculos. Si no se especifica, el numero de paso final sera
27         el numero de la ultima linea valida en el archivo de entrada.

28     -t --intervalo_tiempo=<intervalo_de_tiempo>
29         Establece el intervalo de tiempo considerado en cada paso de ---
30         ejecucion. El intervalo de tiempo por defecto es de 1 fs.
31     """

32     print texto

33     ##-----
34     def norma_cuadrada(vector):
35         """Calcula la norma al cuadrado de un vector.
36         """
37         norma = 0.0
38         for i in vector:
39             norma += i*i
40         return norma
41     ##-----
42     def ordena_posiciones(nombre_archivo, paso_final=-1):
43         """Lee las coordenadas atómicas del archivo tipo ARC.

44         Determina el tipo y número de cada átomo presente en el archivo ARC. Crea un
45         diccionario de diccionarios donde la primera entrada corresponde al tipo y -
46         la segunda al número de átomo, los valores almacenados son una lista ordena-
47         da de coordenadas. Cada coordenada corresponde a la posición de ese átomo en
48         un intervalo de tiempo determinado. Por ejemplo, para acceder a la posición
49         del átomo 'Al28' en el en el paso (intervalo) 53:

50         posicion_atomos['Al'][28][53]

```

```

51     y para el átomo 'Si105' en el paso cero:

52         posicion_atomos['Si'][105][0]
53     """

54     ## Abre el archivo ARC en modo de sólo lectura.
55     archivo_entrada = open(nombre_archivo, "r")

56     ### Expresión regular para un número real
57     #num_real = '([+-]?\.\d+\.\d*(?:[eE][+-]?\d+)?)'

58     ## Expresión regular para encontrar un elemento químico seguido de un número
59     ## y tres números reales (coordenadas) separados por espacios.
60     ## Esta expresión regular generará 5 grupos de búsqueda:
61     ##     grupo 1: nombre del elemento químico
62     ##     grupo 2: número del elemento químico
63     ##     grupo 3: coordenada x
64     ##     grupo 4: coordenada y
65     ##     grupo 5: coordenada z
66     patron_EleNumXYZ = re.compile('^[A-Z][a-z]?(\d+)' + \
67                                   '\s+([+-]?\.\d+\.\d*(?:[eE][+-]?\d+)?)|\s+' + \
68                                   '\s+([+-]?\.\d+\.\d*(?:[eE][+-]?\d+)?)|\s+' + \
69                                   '\s+([+-]?\.\d+\.\d*(?:[eE][+-]?\d+)?)|\s+')
70     ## El patrón anterior contempla la explosión de la celda, en tal caso
71     ## un(os) asterisco(s) deberá(n) aparecer en lugar de la coordenada

72     ## patrón para contar pasos
73     patron_pasos_date = re.compile('^\!DATE')

74     ## Diccionario principal
75     posicion_atomos = {}

76     num_pasos = 0

77     ## Lee una por una las líneas del archivo de entrada
78     for linea in archivo_entrada:
79         ## busca el patron EleNumXYZ en la línea
80         patron_encontrado = re.match(patron_EleNumXYZ, linea)
81         ## si se encuentra el patrón en la línea
82         if patron_encontrado != None:
83             elemento     = patron_encontrado.group(1)
84             numero       = patron_encontrado.group(2)
85             X            = patron_encontrado.group(3)
86             Y            = patron_encontrado.group(4)
87             Z            = patron_encontrado.group(5)

88             ## si ya había encontrado este tipo de átomo
89             if posicion_atomos.has_key(elemento):
90                 ## si ya había un registro de este tipo y numero de átomo
91                 if posicion_atomos[elemento].has_key(int(numero)):

```

```

92         posicion_atomos[elemento][int(numero)] \
93         += [array([float(X), float(Y), float(Z)])]
94     ## si no había un registro de este tipo y numero de átomo
95     else:
96         posicion_atomos[elemento][int(numero)] \
97         = [array([float(X), float(Y), float(Z)])]
98     ##si es la primera vez que encuentra este tipo de átomo
99     else:
100         ## crea un diccionario para almacenar lo números de átomo
101         posicion_atomos[elemento] = {}
102         ## la primera coordenada de este tipo y número de átomo
103         posicion_atomos[elemento][int(numero)] \
104         = [array([float(X), float(Y), float(Z)])]
105     # si el patrón no se encontró en la línea
106     else:
107         # si no se condicionó la lectura del archivo.
108         if paso_final == -1:
109             paso = re.match(patron_pasos_date, linea)
110             # si la línea marca un paso nuevo (^!DATE)
111             if paso != None:
112                 num_pasos += 1
113             continue
114         # si se condicionó a terminar la lectura del archivo en un paso espe-
115         # cífico
116         else:
117             paso = re.match(patron_pasos_date, linea)
118             # si la línea marca un paso nuevo (^!DATE)
119             if paso != None:
120                 num_pasos += 1

121         # si se revasa el número máximo de pasos a leer
122         if num_pasos > paso_final+1: break

123     #continua leyendo el archivo para contar los pasos que faltan
124     for linea in archivo_entrada:
125         paso = re.match(patron_pasos_date, linea)
126         # si la línea marca un paso nuevo (^!DATE)
127         if paso != None:
128             num_pasos += 1

129     archivo_entrada.close()
130     #print "Archivo ARC leído..."
131     return posicion_atomos, num_pasos

132     ##-----
133     def tipos_de_atomos(posiciones):
134         """Regresa una lista con los nombres de los tipos de átomos presentes en el
135         archivo ARC.
136         """
137         return posiciones.keys()

138     ##-----
139     def cantidad_de_cada_tipo(tipos_atomos, posicion_atomos):

```

```

140     """Determina cuántos átomos hay de cada tipo.
141     """
142     num_atomos = []

143     for tipo_atomo in tipos_atomos:
144         num_aux = len(posicion_atomos[tipo_atomo].keys())
145         num_atomos += [num_aux]

146     return num_atomos

147     ##-----
148     def numero_total_de_atomos(atomos_de_cada_tipo):
149         """Da el número total de átomos considerados.
150         """
151         return sum(atomos_de_cada_tipo)

152     ##-----
153     def numero_de_pasos(posicion_atomos, tipos_atomos):
154         """Determina el número de pasos (intervalos de tiempo) encontrados en el ar-
155         chivo ARC.

156         Éste lo determina por la longitud de la lista de coordenadas para el primer
157         átomo en el paso uno.
158         """
159         return len(posicion_atomos[tipos_atomos[0]][1])

160     ##-----
161     def desglosa(maximo, paso):
162         """
163         Desglosa todos los intervalos de tiempo posibles de una lista cuyo último -
164         elemento es el número 'maximo', tomados de 'paso' en 'paso'.

165         Ejemplo: para una lista con max=10 y paso=3, todos los intervalos posibles
166         son: (0,3), (3,6), (6,9), (1,4), (4,7), (7,10), (2,5), (5,8), (3,6), (6,9)
167         """
168         try:
169             for i in xrange(paso+1):
170                 L = xrange(i, maximo+1, paso)
171                 try:
172                     cont = 0
173                     while 1:
174                         yield L[cont], L[cont+1]
175                         cont += 1
176                 except IndexError:
177                     pass
178             except ValueError:
179                 yield (0,0)

180     ##-----
181     def main():

```



```

182     ## argumento, ver documentación del módulo 'getopt'
183     try:
184         opciones, argumentos = getopt.getopt(sys.argv[1:], \
185             "ho:i:f:t:", ["help", "archivo_salida=", "paso_inicial=", \
186                 "paso_final=", "intervalo_tiempo="])
187     except getopt.error, msg:
188         print msg
189         print "Para ayuda use --help"
190         sys.exit(2)

191     ## si se elije la opción de ayuda
192     for (opcion, valor) in opciones:
193         if opcion in ("-h", "--help"):
194             #print __doc__
195             uso(sys.argv[0])
196             sys.exit(0)

197     ## valores por defecto
198     paso_inicial = 0
199     paso_final = -1
200     nombre_archivo_salida = "msd.salida"
201     intervalo_de_tiempo = 1.0

202     # procesamiento de opciones
203     for (opcion, valor) in opciones:
204         if opcion in ("-o", "--archivo_salida"):
205             nombre_archivo_salida = valor
206         elif opcion in ("-i", "--paso_inicial"):
207             paso_inicial = int(valor)
208         elif opcion in ("-f", "--paso_final"):
209             paso_final = int(valor)
210         elif opcion in ("-t", "--intervalo_tiempo"):
211             intervalo_de_tiempo = float(valor)
212         else:
213             assert False, "Opcion inmanejable: " + opcion

214     # procesamiento de argumentos
215     nombre_archivo = argumentos[0]
216     posicion_atomos, num_pasos_total = ordena_posiciones(nombre_archivo, paso_final)
217     tipos_atomos = tipos_de_atomos(posicion_atomos)
218     num_tipos_atomos = len(tipos_atomos)
219     num_atomos = cantidad_de_cada_tipo(tipos_atomos, posicion_atomos)
220     num_pasos = numero_de_pasos(posicion_atomos, tipos_atomos)
221     num_atomos_total = numero_total_de_atomos(num_atomos)

222     paso_final = num_pasos

223     archivo_salida = open(nombre_archivo_salida, "w" )

224     # cabecera del archivo
225     archivo_salida.write("# DESPLAZAMIENTO CUADRATICO MEDIO\n")
226     archivo_salida.write("# tipos de atomo      " )

```

```

227     for i in tipos_atomos:
228         archivo_salida.write(i + " ")
229     archivo_salida.write("\n")
230     #archivo_salida.write("# número de tipos      " + str(len(tipos_atomos)) + '\n')
231     archivo_salida.write("# numero de atomos      ")
232     cont = 0
233     for i in tipos_atomos:
234         archivo_salida.write("%d"%(num_atomos[cont]))
235         cont += 1
236     archivo_salida.write("\n")
237     archivo_salida.write("# numero de intervalos " + str(paso_final-paso_inicial) + '\n')
238     archivo_salida.write("# pasos en archivo      " + str(num_pasos_total) + '\n')
239     archivo_salida.write("# paso inicial          " + str(paso_inicial) + '\n')
240     archivo_salida.write("# paso final            " + str(paso_final-1) + '\n')

241     ## etiqueta las columnas del archivo de salida
242     archivo_salida.write("#")
243     archivo_salida.write("-"*(15 + 21*(1+len(tipos_atomos))))
244     archivo_salida.write("\n#%15s  "%tam_intervalo")

245     for i in tipos_atomos:
246         archivo_salida.write("%20s "%("MSD_"+ str(i)))

247     archivo_salida.write("%20s\n"%MSD_total")
248     archivo_salida.write("#")
249     archivo_salida.write("-"*(15 + 21*(1+len(tipos_atomos))))
250     archivo_salida.write("\n")

251     TIEMPOS = range(paso_inicial, paso_final+1)
252     longitud = paso_final-paso_inicial

253     # registra el número de intervalos de diferente longitud
254     cont_0 = 0

255     # para cada uno de los intervalos de diferente longitud, i
256     for i in xrange(longitud):
257         cont_2 =0
258         #registra el msd correspondiente a un intervalo fijo para cada átomo
259         promedios = []
260         # para cada tipo de átomo en el archivo
261         for atomo in xrange(num_tipos_atomos):
262             tipo_atomo = tipos_atomos[atomo]
263             promedio = 0.0
264             # registra el número de intervalos de igual longitud
265             cont_1 = 0
266             # para cada átomo de un mismo tipo
267             for num in xrange(1,num_atomos[atomo]+1):
268                 # para cada intervalo posible de tamaño i
269                 for (h, k) in desglosa(longitud-1, i):
270                     promedio += norma_cuadrada(\
271                         posicion_atomos[tipo_atomo][num][TIEMPOS[k]] - \
272                         posicion_atomos[tipo_atomo][num][TIEMPOS[h]])

```

```

273             cont_1 += 1
274             cont_2 += 1

275         #promedio /= cont_1
276         #print "%7d %7d %20.10g"%(cont_1/num_atomos[atomo], cont_1, promedio),
277         promedios += [promedio]

278         # si es el primer átomo de la lista
279         if tipo_atomo == tipos_atomos[0]:
280             archivo_salida.write("%15.6g"%((cont_0)*intervalo_de_tiempo))
281             #archivo_salida.write(str(cont_0*intervalo_de_tiempo) + " ")
282             archivo_salida.write("%20.10g" % (promedio/cont_1))
283             #archivo_salida.write(str(promedio)+" ")

284         promedio_total_aux = sum(promedios) / cont_2
285         archivo_salida.write("%20.10g\n"%(promedio_total_aux))
286         #archivo_salida.write(str(promedio_total_aux) + "\n")
287         print "%7d %20.10g"%(i,promedio_total_aux)

288         cont_0 += 1
289         print ""
290         archivo_salida.close()
291         #-----
292         def tiempo(t):
293             dia    = 86400
294             hora   = 3600
295             minuto = 60

296             num_dias  = t//dia
297             t -= num_dias*86400
298             num_horas = t//hora
299             t -= num_horas*3600
300             num_minutos= t//minuto
301             t -= num_minutos*60

302             print "%2d días %2d hr %2d min %6.4f s"%(num_dias, num_horas, num_minutos, t)
303         #-----

304         if __name__ == "__main__":

305             import time
306             t1 = time.clock()
307             main()
308             t2 = time.clock()
309             #print t2-t1, "segundos"
310             tiempo(t2-t1)
311             print "Fin."

```

A.11 FUNDAMENTO TEÓRICO DEL DESPLAZAMIENTO MEDIO CUADRÁTICO

En esta sección se describe el procedimiento utilizado en el programa `msd` para el cálculo del desplazamiento cuadrático medio, MSD.

- Las palabras «estado», «paso» y las expresiones «número de intervalo de tiempo» y «número de paso» hacen referencia a una misma cosa: al número de paso en la dinámica molecular. Así, para referirse al número de paso 10 puede decirse paso 10, estado 10 o bien número de intervalo 9.⁶⁰ Esto equivale a dar las posiciones atómicas después de 117 fs en una dinámica con un *time step* de 13 fs.
- Un archivo `.car` es un archivo que contiene las posiciones atómicas de una dinámica en un estado determinado, y aunque después de cada dinámica se genera un único archivo del tipo `.arc` que contiene todas las posiciones atómicas de todos los estados de la dinámica, es más cómodo trabajar con las posiciones de cada estado por separado (archivos `.arc`) por lo que así se hará en esta discusión.
- Debe tomarse en cuenta que la definición de MSD no considera el desplazamiento en sí, sino el cuadrado del módulo de éste, ya que no se desea realizar una suma vectorial la cual puede llevar a la anulación del desplazamiento en una dirección con respecto al desplazamiento en otra dirección. Por tanto se utilizará el término desplazamiento como sinónimo del cuadrado de la distancia entre dos posiciones ya sean atómicas o entre estados. Así, al decir el desplazamiento de los átomos del estado uno al estado dos, debe interpretarse como: el cuadrado de la distancia que hay entre los átomos del estado uno con respecto a los átomos del estado dos.

Para describir el cálculo del MSD considérese una dinámica hipotética con duración de 110 fs donde las posiciones atómicas se registran, en un archivo `.car`, cada 10 fs. De esta manera se tendrán 12 archivos tipo `.car` incluyendo el de las posiciones iniciales (paso uno, P_1). Con esta información, el cálculo de MSD sólo requiere de la aplicación de la fórmula:

$$MSD = \langle r^2(t) \rangle = \left\langle \frac{1}{N} \sum_{i=0}^N |r_i(t) - r_i(0)|^2 \right\rangle$$

⁶⁰ Se considera el paso uno como aquél de las posiciones atómicas iniciales.

donde $|r_i(t) - r_i(0)|^2$ se refiere a la diferencia en posiciones, elevado al cuadrado, de un átomo i en el intervalo de tiempo $t - 0 = t$. Esta notación tiene el inconveniente de aparentar que los intervalos de tiempo siempre se toman desde $t = 0$, siendo que lo importante no es el origen o fin del intervalo sino el *tamaño* del intervalo en sí. De esta forma, el intervalo sería mejor representado como $t_f - t_o = \Delta t$ y quizá la fórmula de MSD debería haberse escrito como:

$$MSD_{\Delta t} = \left\langle \overline{\Delta r^2(t)} \right\rangle = \left\langle \frac{1}{N} \sum_{i=0}^N |r_i(t_f) - r_i(t_o)|^2 \right\rangle$$

Aquí puede verse un promedio sobre todos los átomos, N , que se encuentran entre los límites del intervalo de tiempo, Δt . Ahora bien, los paréntesis angulares denotan otro promedio, éste hace referencia al promedio sobre todos los intervalos de tiempo cuyo tamaño sea igual a Δt . Por lo que la fórmula podría reescribirse nuevamente así:

$$MSD_{\Delta t} = \frac{1}{n} \sum_{\Delta t} \overline{\Delta r^2(t)} = \frac{1}{n} \frac{1}{N} \sum_{\Delta t} \sum_{i=0}^N |r_i(t_f) - r_i(t_o)|^2,$$

donde la suma sobre Δt denota la suma sobre todos los intervalos del *mismo tamaño* que puedan formarse (con la información con que se cuente). El número de estos intervalos será igual a n . Al observar con cuidado esta última ecuación se notará que $MSD_{\Delta t}$ sólo considera el promedio sobre intervalos de un mismo tamaño, por lo que si se desea obtener una curva que describa el desplazamiento cuadrático medio a cualquier intervalo, tendrá que calcularse el MSD para los diferentes tamaños posibles, es decir, $MSD_{\Delta t_1}$, $MSD_{\Delta t_2}$, $MSD_{\Delta t_3}$, etcétera.

Realmente el MSD sólo debe responder a la pregunta: *¿en promedio, cuánto se desplaza⁶¹ un átomo en un intervalo de tiempo determinado?* En este ejemplo, y por la naturaleza de la información disponible, tal intervalo deberá ser múltiplo de 10 fs, y la pregunta será entonces: *¿en promedio, cuánto se desplaza un átomo en un intervalo de 10 fs?, ¿cuánto en uno de 20 fs?, ¿cuánto en uno de 30 fs?, ¿cuánto en otro de 40 fs?, y así sucesivamente.* Lo primero es, determinar el desplazamiento promedio entre los átomos de dos estados

⁶¹ El cuadrado de la distancia.

que difieran 10, 20, 30, ..., y 110 fs —la forma de hacerlo se muestra en la **figura A.1**, donde, de forma esquemática, se promedia el desplazamiento de cada átomo después de un intervalo de tiempo igual a 60 fs (6 x 10 fs), la línea curva que une al P₁ con el P₇ (parte inferior) representa el desplazamiento promedio del estado 1 al estado 7—. Podría tomarse como estado inicial al P₁ y de ahí determinar el desplazamiento después de 10 fs (P₂), 20 fs (P₃), 30 fs (P₄), ..., 110 fs (P₁₂), aunque podría tomarse como estado inicial al P₂ y de ahí determinar el desplazamiento después de 10 fs (P₃), 20 fs (P₄), 30 fs (P₅), ..., 100 fs (P₁₂), o bien tomar al P₃ y determinar el desplazamiento después de 10 fs (P₄), ...y así hasta llegar al P₁₁ y determinar el desplazamiento después de 10 fs (P₁₂) únicamente. Todos ellos son intervalos de tiempo con los que se puede promediar. La **figura A.2** muestra todas las posibilidades.

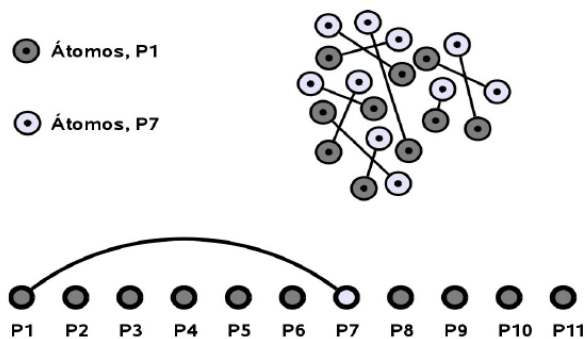


Figura A.1 Representación esquemática del desplazamiento en las posiciones atómicas en un intervalo de tiempo de 60 fs. Cada círculo representa todas las posiciones atómicas en un paso determinado.

En la **figura A.3** se han organizado los desplazamientos de la figura anterior según el tamaño del intervalo tiempo. Para responder la pregunta planteada al MSD *¿en promedio, cuánto se desplaza un átomo en un intervalo de 10, 20, 30, ... y 110 fs?*, es necesario promediar todos los intervalos de 10, 20, 30, ... y 110 fs. En esta misma figura puede observarse que el promedio de cada intervalo, hasta el de 60 fs (P₁–P₇, segunda columna de la figura), se hace con todos los estados disponibles pero después de este intervalo (60 fs,) se promedia con un número cada vez menor de estados.

El significado que puede dársele a esto es simple: *el valor de MSD después de $\Delta t = 60$ fs desde el punto de vista estadístico, carece de significado, pues no se consideran todos los*

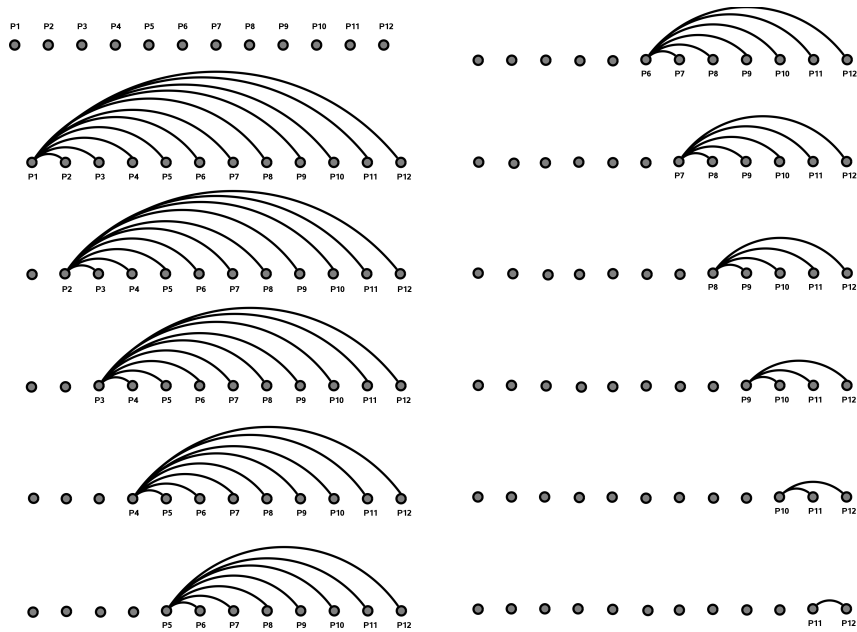


Figura A.2 Intervalos de tiempo posibles en un sistema de 12 pasos de simulación.

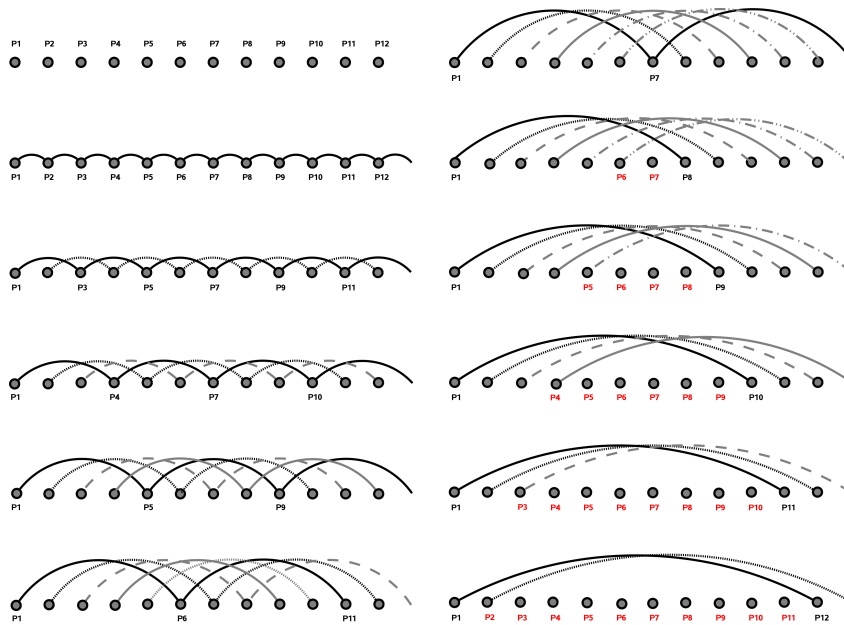


Figura A.3 Intervalos de tiempo posibles en un sistema de 12 pasos de simulación.

estados de la muestra. Esta oración puede generalizarse diciendo que para un número M de diferentes intervalos de tiempo posibles, ordenados de menor a mayor, el cálculo de MSD es más representativo para los primeros $M/2$ intervalos. En la siguiente tabla (**tabla A.1**)

se resume la estadística de este ejemplo marcándose con un asterisco aquellos intervalos que no son representativos por no considerar los 12 estados posibles.

Tabla A.1 Estadística de una dinámica con 12 estados y un *time step* de 10 fs (Los asteriscos señalan aquellos intervalos en los que no se consideran los 12 estados).

Veces <i>time step</i>	Intervalos	Número de intervalos	Estados considerados
1	P1–P2, P2–P3, P3–P4, P4–P5, P5–P6, P6–P7, P7–P8, P8–P9, P9–P10, P10–P11, P11–P12	11	12
2	P1–P3, P3–P5, P5–P7, P7–P9, P9–P11, P2–P4, P4–P6, P6–P8, P8–P10, P10–P12	10	12
3	P1–P4, P4–P7, P7–P10, P2–P5, P5–P8, P8–P11, P3–P6, P6–P9, P9–P12	9	12
4	P1–P5, P5–P9, P2–P6, P6–P10, P3–P7, P7–P11, P4–P8, P8–P12	8	12
5	P1–P6, P6–P11, P2–P7, P7–P12, P3–P8, P4–P9, P5–P10	7	12
6	P1–P7, P2–P8, P3–P9, P4–P10, P5–P11, P6–P12	6	12
7	P1–P8, P2–P9, P3–P10, P4–P11, P5–P12	5	10*
8	P1–P9, P2–P10, P3–P11, P4–P12	4	8*
9	P1–P10, P2–P11, P3–P12	3	6*
10	P1–P11, P2–P12	2	4*
11	P1–P12	1	2*

A continuación se muestra la curva de MSD generada a partir de una muestra de Ge con time-step de 13 fs, **figura A.4**, el número de pasos de la dinámica es igual a 515, $M = 515$. Por lo dicho en la sección anterior esta curva es más representativa hasta el intervalo $M/2 = 258$ o en términos de tiempo $258 \times 3354\text{fs}$.

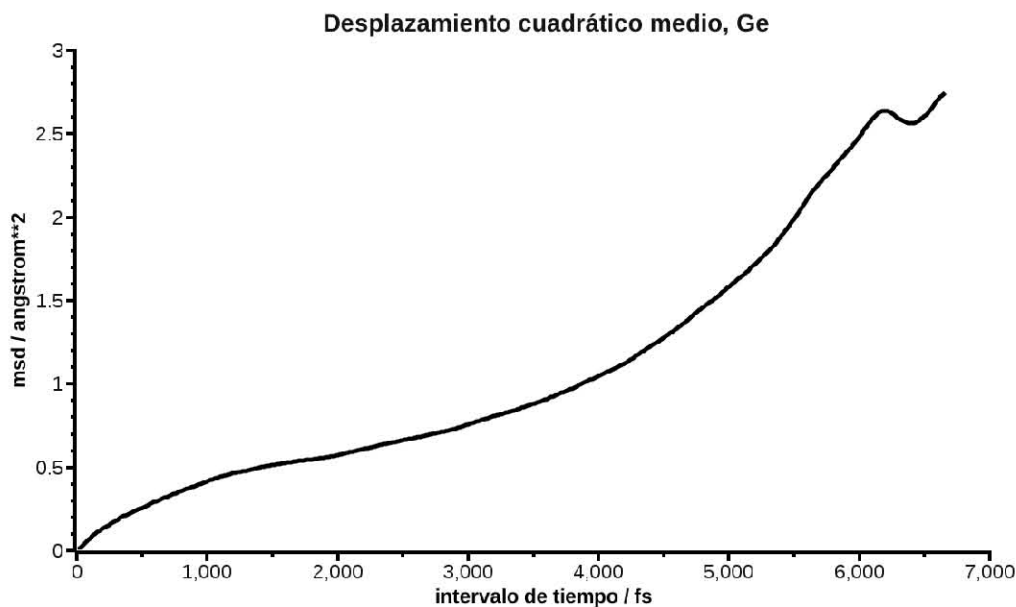


Figura A.4 MSD para Ge con time-step de 13 fs y 515 intervalos de tiempo.

La **gráfica A.5** siguiente muestra la misma curva superpuesta a una serie de curvas que representan todas las mediciones de desplazamiento tomando en cada una de ellas un t_o constante y variando t_f . Éstas son las curvas generadas con el procedimiento mostrado en la **figura A.2**. El promedio de todas estas curvas genera la curva de MSD.

A.12 VALIDEZ DEL DESPLAZAMIENTO MEDIO CUADRÁTICO PARA SIMULACIONES *ab initio*

El desplazamiento medio cuadrático es, como se indica en la sección anterior, un parámetro estadístico, su validez por tanto, debe ser sensible al número de átomos y estados considerados. Nosotros realizamos una serie de pruebas con un elemento ligero, como el silicio, para determinar qué tan reproducible y que tan aceptables son los resultados obtenidos. En la primera prueba realizamos dinámica molecular, NVT, a 1000 K, a 4 muestras de silicio con estructura diamante cuya densidad fue de 2.33 g cm^{-3} (sin porosidad, sólo se pretendía amorfizar). Cada muestra tenía un time step diferente pero pretendíamos que el tiempo de simulación fuera el mismo, 800 fs. Así que utilizamos time steps de 1, 2, 3 y 4 fs, lo que conduce a 800, 400, 267 y 200 pasos de simulación. Lo que nosotros

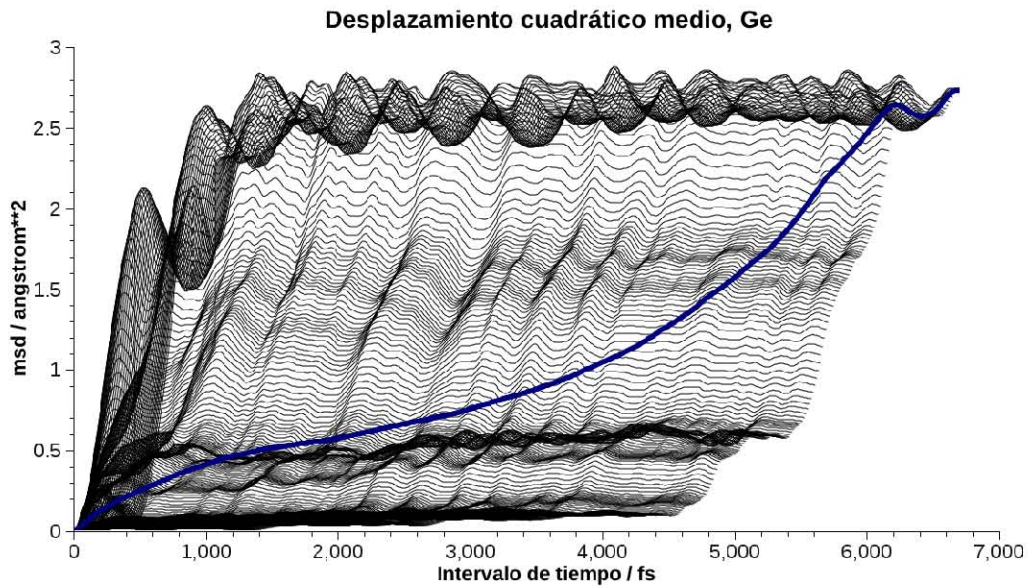


Figura A.5 MSD para Ge con *time step* de 13 fs y 515 intervalos de tiempo (curva principal). Las curvas de fondo son curvas de desplazamiento considerando un estado fijo.

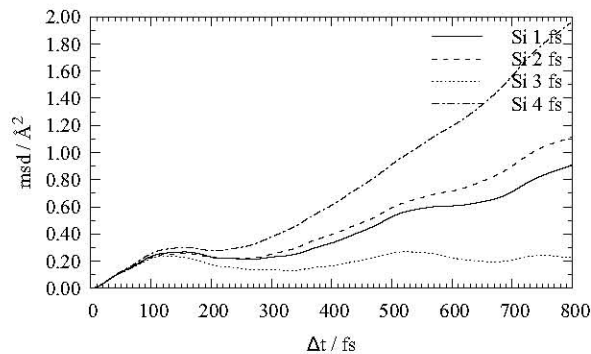


Figura A.6 Cuatro curvas MSD para Si, el tiempo total de simulación es 800 fs. Los intervalos son variables.

esperábamos eran ligeras variaciones en las curvas de MSD. La **figura A.6** muestra los resultados obtenidos.

Esta gráfica muestra que no sólo no son semejantes los resultados si no que no existe una tendencia aparente en su comportamiento. Repetimos las simulaciones para 3 y 4 fs y obtuvimos las siguientes gráficas, **figura A.7**.

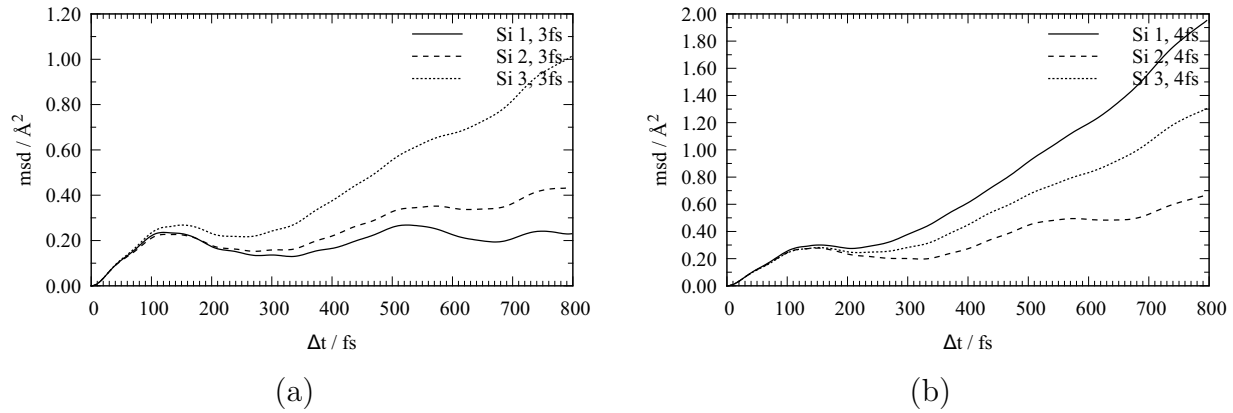


Figura A.7 Cuatro curvas MSD para Si, el tiempo total de simulación es 800 fs. Los intervalos son variables. (a) 3 fs, (b) 4 fs.

De estas gráficas podemos concluir que al parecer hay cierta aleatoriedad en los resultados. ¿A qué se debe? La única diferencia en estas últimas simulaciones es: las velocidades aleatorias asignadas al comienzo de la dinámica. La suma vectorial de las velocidades iniciales, SVV , al parecer determina la posición de la curva de MSD. Aquellas simulaciones con menor SVV generan curvas por debajo de aquellas que provienen de una simulación con mayor SVV . También hicimos un par de simulaciones más en las que hacíamos el número de paso igual para todas, lo que cambia el tiempo de simulación total, a 800, 1600, 2400 y 3200 fs, esperando que al haber un mismo número de pasos, las curvas de MSD fueran más semejantes. Los resultados los presentamos en la **figura A.8**

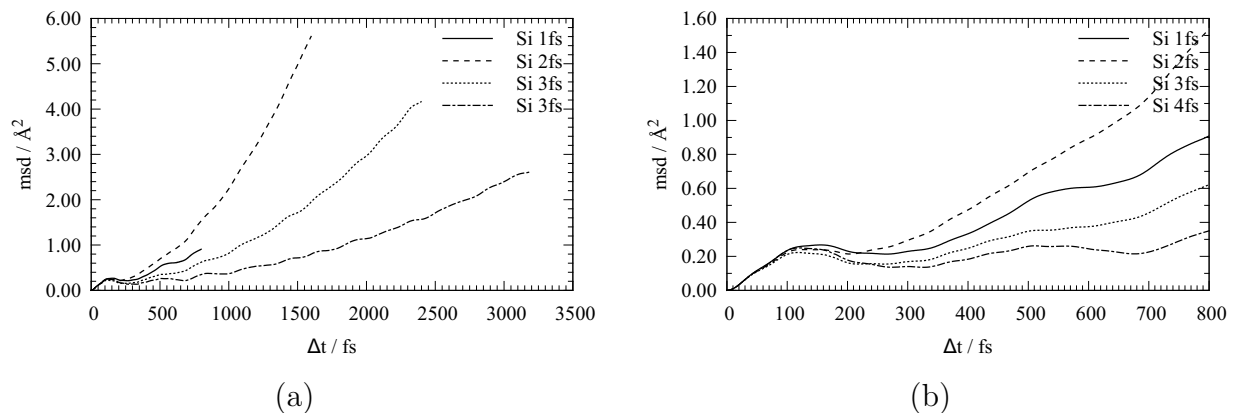


Figura A.8 (a) Cuatro curvas MSD para Si, el número de pasos es igual a 800. Los intervalos son variables. (b) Detalle de la figura (a).

De la misma forma, la posición de la curva sigue siendo sensible a la SVV . Aunque se aprecia mayor semejanza al principio, lo que indica que posiblemente aumentando el número de

pasos aumente la semejanza. El otro punto es, las velocidades aleatorias asignadas siguen la distribución de Maxwell-Boltzmann y su dirección es aleatoria, lo que indica que con un número suficiente de velocidades elegidas la SVV debería tender a cero, esto haría que las curvas de MSD no dependieran de la SVV, el número de átomos empleado en las simulaciones *ab initio* no permite observar una $SVV \approx 0$, por lo que las curvas MSD no son representativas para un número tan pequeño de átomos.

A.13 PROGRAMA GAUSSEAME

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3  from __future__ import division
4  from math import pi, exp, ceil
5  from numpy import *
6  import getopt
7  import sys

8  __author__ = 'Ulises Santiago <ulises_osa@yahoo.com.mx>'
9  __all__    = []

10 def uso(nombre_programa):
11     texto = "\nuso: " + nombre_programa + " " [OPCIONES] archivo-de-datos

12     DESCRIPCION
13     -----

14     OPCIONES
15     -h --help
16         Muestra esta ayuda y sale.

17     -s --sigma=numero
18         Desviacion estandar considerada para cada gaussian. Por defecto
19         s=0.1.

20     -n, --veces_sigma=numero
21         Numero de desviaciones estandar, apartir de la media, que se consi-
22         deran para evaluar cada gaussiana. Por ejemplo, si se elige n=2, con
23         una media igual a m la gaussina considerada estara definida dentro
24         del rango [m-2s, m+2s]. Por defecto n=4.

25     -i, --intervalos_por_sigma=numero
26         Cada distancia igual a sigma se divide entre 'i' intervalos de mues-
27         treo. Entre más grande sea estevalor mayor numero de puntos habra de
28         la gaussiana. Por defecto i=10.
```

```

29     -o, --archivo_de_salida=nombre
30     Nombre del archivo donde se almacenan los datos de salida. Si esta
31     opcion no se emplea, el nombre por defecto sera el nombre del archi-
32     vo de entrada mas ".gausseado".

33     -e, --escala_valores=numero
34     Este parametro escala la altura de la gaussiana. En el escalamiento
35     se escala igualmente la desviacion estandar y el numero de intervalos
36     por desviacion estandar. El valor por defecto es 1 (sin escalamiento).
37

38     -p, --pauli= (0|1)
39     Debe el espectro cumplir con el principio de exclusion de Pauli. Por
40     defecto es falso (0).

41     """
42     print texto

43     #-----
44     def normal(x, mu, sigma, altura=1):
45         Z = (x - mu) / sigma
46         exponente = -0.5 * Z*Z
47         return altura * exp(exponente)
48     ##-----

49     def main():
50         ## argumento, ver documentación del módulo 'getopt'
51         #-----
52         try:
53             opciones, argumentos = getopt.getopt(sys.argv[1:], \
54                 "hs:n:i:o:e:p:", ["help", "sigma=", "veces_sigma=", \
55                     "intervalos_por_sigma=", "archivo_de_salida=", \
56                     "escala_valores=", "pauli="])
57         except getopt.error, msg:
58             print msg
59             print "Para ayuda use --help"
60             sys.exit(2)

61         for (opcion, valor) in opciones:
62             if opcion in ("-h", "--help"):
63                 uso(sys.argv[0])
64                 sys.exit(0)

65         ## nombre del archivo de datos
66         n_archivo_de_datos = argumentos[0]
67         archivo_de_datos = open(n_archivo_de_datos, "r")
68         escala = 1

69         ## valor por defecto
70         sigma = 0.1     ## desviación estándar
71         intervalos_sigma = 10.0     ## intervalos en los que se divide cada sigma

```

```

72     num_s = 4.
73     pauli = 0
74     parametros = []
75     n_archivo_de_salida = n_archivo_de_datos + ".gausseado"

76     ### si se elije la opción de ayuda
77     for (opcion, valor) in opciones:
78         if opcion in ("-h", "--help"):
79             uso(sys.argv[0])
80             sys.exit(0)
81         elif opcion in ("-s", "--sigma"):
82             sigma = float(valor)
83         elif opcion in ("-n", "--veces_sigma"):
84             num_s = float(valor)
85         elif opcion in ("-i", "--intervalos_por_sigma"):
86             intervalos_sigma = float(valor)
87         elif opcion in ("-o", "--archivo_de_salida"):
88             n_archivo_de_salida = valor
89         elif opcion in ("-e", "--escala_valores"):
90             escala = float(valor)
91         elif opcion in ("-p", "--pauli"):
92             pauli = int(valor)

93     #-----
94     n_archivo_de_parametros = n_archivo_de_datos + ".gausseanas"
95     archivo_de_parametros = open(n_archivo_de_parametros, 'w')
96     #-----

97     num_sigma = num_s*sigma           ## número de veces la desviación
98                                     ## estándar, extensión de la gaussiana
99     tol_g = 1.*sigma/intervalos_sigma ## tamaño de cada intervalo
100    histograma = {}                   ## almacena las coordenadas de las
101                                     ## gaussianas, de todas

102    for linea in archivo_de_datos:
103        if linea[0]=='#': continue
104        alturas = []
105        ef = linea.split()
106        mu = float(ef[0])              ## energía media
107        num_columnas = len(ef)         ## resto de las columnas

108        ## cada una de las alturas para cada línea
109        alturas = [ float(ef[i]) for i in range(1, num_columnas) ]

110    CEROS = True
111    for altura in alturas:
112        ## alguna de las alturas es diferente de cero termina
113        if altura != 0:
114            CEROS = False
115            break

```

```

116     ## si todas las alturas para esa energía son cero
117     if CEROS:
118         continue

119     archivo_de_parametros.write("%15.6g %15.6g"%(mu, sigma))

120     for altura in alturas:
121         archivo_de_parametros.write(" %15.6g"%(altura*escala))

122     archivo_de_parametros.write("\n")
123     parametros += [[mu, sigma] + [altura*escala for altura in alturas]]

124     archivo_de_datos.close()
125     archivo_de_parametros.close()

126     indice = 0
127     ## De la primera energía toma 5 desviaciones estándar hacia la izquierda y
128     ## de la última 5 hacia la derecha
129     for x in arange(parametros[ 0][0] - num_sigma*escala, \
130                   parametros[-1][0] + num_sigma*escala, tol_g):

131         ## sumas para cada una de las columnas de alturas
132         sumas = [0.0 for i in range(num_columnas-1)]

133         for j in range(num_columnas-1):
134             ## para cada columna de alturas suma cada gaussiana correspondiente
135             ## a una energía, número de filas = número de gaussianas sumadas
136             for i in parametros:
137                 sumas[j] += normal(x, i[0], i[1], i[j+2])

138         CEROS = True
139         for i in range(len(sumas)):
140             if sumas[i] > 1e-7:
141                 CEROS=False
142             else:
143                 sumas[i] = 0

144         ## si la sumas de toda la fila es cero
145         if CEROS:
146             continue

147         ## si cada altura debe ser un numero par
148         if pauli:
149             for i in range(len(sumas)):
150                 sumas[i] = floor(sumas[i])
151                 if sumas[i]%2!=0:
152                     sumas[i] += 1

153     histograma[indice] = [x] + [suma for suma in sumas]

```

```

154         indice += 1

155     entradas = histograma.keys()
156     entradas.sort()

157     archivo_de_salida = open(n_archivo_de_salida, "w")

158     indice = 0
159     for i in range(len(histograma.keys())):
160         archivo_de_salida.write( "%15.6f"%histograma[indice][0])
161         for j in range(1, num_columnas):
162             archivo_de_salida.write( " %15.6f"%(histograma[indice][j]))
163         indice += 1
164         archivo_de_salida.write("\n")

165     archivo_de_salida.close()

166     print "desviación estándar: %-10.5g" %sigma
167     print "veces_sigma:          %-10.5g" %(num_sigma/sigma)
168     print "intervalos_sigma:     %-10.5g" %intervalos_sigma
169     print "tol_g:                %-10.4g" %tol_g
170     print "escala:               %-10.4g" %escala

171 if __name__ == "__main__":
172     main()

```

A.14 PROGRAMA VIBRACIONES

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-

3  from __future__ import division
4  from angulos import *
5  from numpy import *
6  from math import floor
7  import re
8  import getopt
9  import sys
10 import time

11 __author__ = 'Ulises Santiago <ulises_osa@yahoo.com.mx>'

12 def main():
13     """Función principal.
14     """
15     ## argumento, ver documentación del módulo 'getopt'
16     try:

```



```

17     opciones, argumentos = getopt.getopt(sys.argv[1:], \
18         "ho:r:e:c:", ["help", "archivo_salida=", \
19             "distancia_de_enlace=", "archivo_distancias_enlace=",
20             "ocupar-nombre-base="])
21 except getopt.error, msg:
22     print msg
23     print "Para ayuda use --help"
24     sys.exit(2)

25     ## si se elije la opción de ayuda
26     for (opcion, valor) in opciones:
27         if opcion in ("-h", "--help"):
28             uso(sys.argv[0])
29             sys.exit(0)

30     ## valores por defecto
31     ## condiciones_periodicas = 1          # True
32     nombre_archivo_salida = nombre_base() # nombre base del archivo del entrada
33     distancia_de_enlace   = 2.45
34     distancias_de_enlace = {}
35     distancias = []

36     # procesamiento de opciones
37     for (opcion, valor) in opciones:
38         if opcion in ("-o", "--archivo_salida"):
39             nombre_archivo_salida = valor
40             #elif opcion in ("-p", "--condiciones_periodicas"):
41             #    condiciones_periodicas = int(valor)
42         elif opcion in ("-r", "--distancia_de_enlace"):
43             distancia_de_enlace = float(valor)

44         elif opcion in ("-e", "--archivo_distancias_enlace"):

45             enlaces = open(valor, 'r')

46             ## crea las entradas para los enlaces. Existen dos entradas para
47             ## cada pareja que forman un enlace, por ejemplo, si en el archivo
48             ## se registra Si - Mg 2.57, puede accederse a esta información
49             ## mediante [Si][Mg] o bien [Mg][Si].
50             for linea in enlaces:
51                 entrada = linea.split()
52                 distancias_de_enlace[entrada[0]] = {}
53                 distancias_de_enlace[entrada[1]] = {}

54             enlaces.close()
55             enlaces = open(valor, 'r')

56             ## obtiene la información del enlaces
57             for linea in enlaces:
58                 entrada = linea.split()
59                 distancia = float(entrada[2])

```

```

60         distancias_de_enlace[entrada[0]][entrada[1]] = distancia
61         distancias_de_enlace[entrada[1]][entrada[0]] = distancia

62         distancias += [distancia]

63         enlaces.close()

64     else:
65         assert False, "Opcion inmanejable: " + opcion

66     # procesamiento de argumentos
67     nombre_archivo = argumentos[0]
68     posicion_atomos, a, b, c, alfa, beta, gama = \
69         administra_coordenadas(nombre_archivo)

70     tipos_atomos = tipos_de_atomos(posicion_atomos)
71     num_tipos_atomos = len(tipos_atomos)
72     num_atomos = cantidad_de_cada_tipo(tipos_atomos, posicion_atomos)
73     num_atomos_total = numero_total_de_atomos(num_atomos)
74     distancias += [distancia_de_enlace]
75     max_distancia_enlace = max(distancias)
76     #print a,b,c

77     ## Éstas son "dos parejas" de ciclos que recoren, cada una, el tipo y el nú-
78     ## mero de átomo, de esta forma cada átomo se compara con todos los demás.
79     for atomo in tipos_atomos:
80         for numero in posicion_atomos[atomo].keys():
81             for atomo_o in tipos_atomos:
82                 for numero_o in posicion_atomos[atomo_o].keys():
83                     try:
84                         ## si se dieron los enlaces a través de un archivo
85                         radio = distancias_de_enlace[atomo][atomo_o]
86                     except KeyError:
87                         ## el valor por defecto
88                         radio = distancia_de_enlace

89     ## Determina si un átomo "puede" estar enlazado a otro de
90     ## forma directa o mediante una translación en la supercelda
91     ## (condiciones periódicas) genera las coordenadas de tal -
92     ## translación.
93     cambio_coor = A_donde((a,b,c), max_distancia_enlace, \
94         posicion_atomos[atomo][numero]['original'], \
95         posicion_atomos[atomo_o][numero_o]['original'])

96     ## Si están enlazados los dos átomos, sin importar que sea
97     ## a través de una translación
98     if cambio_coor.XYZ() != None:

```

```

99         if enlazados(posicion_atomos[atomo][numero]['original'],\
100                     cambio_coor.XYZ(), radio): #distancia_de_enlace):
101             x, y, z = cambio_coor.XYZ()

102         ## comprueba si el átomo no hace referencia a sí mismo
103         hacer = True
104         if numero == numero_o and atomo == atomo_o:
105             hacer = False

106         ## sino es él mismo
107         if hacer:
108             ## registra el átomo con el que se enlaza
109             if posicion_atomos[atomo][numero]["adyacentes"].has_key(atomo_o):
110                 posicion_atomos[atomo][numero]["adyacentes"] \
111                     [atomo_o][numero_o] = array([x, y, z])
112             else:
113                 posicion_atomos[atomo][numero]["adyacentes"] \
114                     [atomo_o] = {numero_o: array([x, y, z])}

115         ## almacena el tipo de átomo y su coordinación
116         coordinacion={}
117         ## almacena el número de coordinación y de ahí qu el elemento la tiene
118         coordinacion_inv={}

119         for i in tipos_atomos:
120             coordinacion[i] = {}

121         for atomo in tipos_atomos:
122             for numero in posicion_atomos[atomo].keys():
123                 coordinacion[atomo][numero] = 0
124                 for atomo_o in posicion_atomos[atomo][numero]["adyacentes"].keys():
125                     coordinacion[atomo][numero] += \
126                         len(posicion_atomos[atomo][numero]["adyacentes"][atomo_o])

127                 #coordinacion_inv[coordinacion[atomo][numero]] = {}
128                 if not coordinacion_inv.has_key(coordinacion[atomo][numero]):
129                     coordinacion_inv[coordinacion[atomo][numero]] = {}

130                 try:
131                     coordinacion_inv[coordinacion[atomo][numero]][atomo] += [numero]
132                 except KeyError:
133                     coordinacion_inv[coordinacion[atomo][numero]][atomo] = []
134                     coordinacion_inv[coordinacion[atomo][numero]][atomo] += [numero]

135         #print coordinacion
136         for i in coordinacion_inv:
137             print i
138             for j in coordinacion_inv[i]:
139                 print j, len(coordinacion_inv[i][j] )
140             print coordinacion_inv[i][j]
141         print

```

```

142     ## desplazamientos
143     ## tipo -> numero -> v_vibracion -> vector
144     desplazamientos = {}

145     leer_modos_1 = re.compile("^\s+vibrational")
146     leer_modos_2 = re.compile("^\s+\d+\s+(?:[-+]?\.?\d+\.\.?d*(?:[eE] [-+]?d+)?)" +
147         "\s+(?:[-+]?\.?\d+\.\.?d*(?:[eE] [-+]?d+)?)")
148     leer_vibraciones_1 = re.compile("^s+Frequencies")

149     vibraciones = []

150     archivo = open(nombre_archivo_salida + '.outmol', 'r' )

151     for linea in archivo:
152         if re.match(leer_modos_1, linea) != None:
153             break

154     num_modos = 0
155     for linea in archivo:
156         vibracion = re.match(leer_modos_2, linea)
157         if re.match(leer_modos_2, linea) != None:
158             vibraciones += [float(vibracion.group(1))]
159             num_modos += 1
160         elif re.match(leer_vibraciones_1, linea) != None:
161             break

162     patron_vibraciones = \
163         re.compile('\s+(\w+)?\s+([xyz])' +
164             '\s+(?:[-+]?\.?\d+\.\.?d*(?:[eE] [-+]?d+)?)*9 )

165     coordenada = {'x':0, 'y':1, 'z':2}
166     desplazamientos = {}

167     for tipo in tipos_atomos:
168         desplazamientos[tipo] = {}
169         for numero in posicion_atomos[tipo].keys():
170             desplazamientos[tipo][numero] = {}
171             for vibracion in xrange(1, num_modos+1):
172                 desplazamientos[tipo][numero][vibracion] = zeros(3)

173     cont_elemento = {}
174     for i in tipos_atomos:
175         cont_elemento[i] = 0

176     ## número total de átomos
177     cont = 0
178     vibracion_inicial = 1

```

```

179 #patron_vibraciones = \
180 # re.compile('(\w+)?\s+([xyz])(?:\s+([-+]?\.?\d+\.\d*(?:[eE] [-+]?[d+]?))){1,}')

181 for linea in archivo:
182     parametros_celda = re.match(patron_vibraciones, linea)

183     if parametros_celda != None:
184         ## si en el primer grupo está el nombre de un elemento
185         if parametros_celda.group(1) != None:

186             tipo = parametros_celda.group(1)
187             cont_elemento[tipo] += 1
188             cont += 1
189     else:
190         for i in tipos_atomos:
191             cont_elemento[i] = 0
192         continue

193     cont1 = 0
194     for vibracion in xrange(vibracion_inicial, vibracion_inicial+9):
195         if parametros_celda.group(3+cont1) == None:
196             cont1 += 1
197             continue

198         desplazamientos[tipo][cont_elemento[tipo]]\
199         [vibracion][coordenada[parametros_celda.group(2)]] =\
200         parametros_celda.group(3+cont1)
201         cont1 += 1

202     if num_modos < cont*3:
203         cont = 0
204         vibracion_inicial += cont1

205     a_supertotal = open(nombre_archivo_salida + "_" + "suma_total_v.dat", "w")

206     archivos = {}
207     for modo in range(1, num_modos+1):
208         numerador_total = 0.0
209         numeradores = {}
210         denominadores = {}
211         for coord in coordinacion_inv:
212             #if coord==0: continue
213             for X in coordinacion_inv[coord]:
214                 for i in coordinacion_inv[coord][X]:
215                     for Y in posicion_atomos[X][i]["adyacentes"].keys():
216                         for j in posicion_atomos[X][i]["adyacentes"][Y].keys():
217                             coordinacion_Y= coordinacion[Y][j]
218                             dif_vec = desplazamientos[X][i][modo] - \
219                                     desplazamientos[Y][j][modo]
220                             numerador = dot(dif_vec, dif_vec)

```

```

221         nombre_1 = str(X) + str(coord) + '-' + str(Y) + \
222                 str(coordinacion_Y)
223         nombre_2 = str(Y) + str(coordinacion_Y) + '-' + \
224                 str(X) + str(coord)

225         if numeradores.has_key(nombre_1) or \
226             numeradores.has_key(nombre_2):
227             try:
228                 numeradores[nombre_1] += numerador
229             except KeyError:
230                 numeradores[nombre_2] += numerador

231         else:
232             numeradores[nombre_1] = numerador

233         numerador_total += numerador

234     for nombre in numeradores:
235         denominadores[nombre] = numeradores[nombre]/numerador_total

236     for nombre in denominadores:
237         try:
238             for i in range(int(denominadores[nombre]*100)):
239                 archivos[nombre].write("%g\n"%vibraciones[modo-1])
240                 a_supertotal.write("%g\n"%vibraciones[modo-1])
241         except KeyError:
242             archivos[nombre] = open(nombre_archivo_salida + "_" + nombre + \
243                                     '_v.dat', 'w')
244             for i in range(int(denominadores[nombre]*100)):
245                 archivos[nombre].write("%g\n"%vibraciones[modo-1])
246                 a_supertotal.write("%g\n"%vibraciones[modo-1])

247     for nombre in denominadores:
248         archivos[nombre].close()

249     a_supertotal.close()

250 if __name__ == '__main__':
251     main()

```