

UNIVERSITY OF



VALLADOLID

FACULTY OF SCIENCES

MASTER DISSERTATION

INTERUNIVERSITY MASTER IN MOLECULAR
NANOSCIENCE AND NANOTECHNOLOGY

**Characterization and dynamics
of Si self-interstitial clusters by
self-learning kinetic Monte Carlo
simulations**

Author:

Mr. Diego Calvo Ruiz

Supervisor:

Dr. Pedro López Martín

Valladolid, July 15th, 2016

TITLE: **Characterization and dynamics of Si self-interstitial cluster by self-learning kinetic Monte Carlo simulations**

AUTHOR: **Mr. Diego Calvo Ruiz**

CO-ADVISOR: **Dr. Pedro López Martín**

CO-ADVISOR: **Dr. María Luz Rodríguez Méndez**

DEPARTAMENT: **Electricity and Electronics**

READING DATE: **July 15th, 2016**

GRADE POINT:

Abstract

Si is a semiconductor material whose relevance in the industry is undeniable, being implemented in every generation of the transistor scaling over the last decades thanks to their excellent properties and easy production. During the fabrication process it is common to deal with the diffusion of impurity atoms in Si, which is critically influenced by intrinsic defects such as self-interstitials and vacancies. Point defects tend to aggregate forming small clusters and extended defects and therefore, the dopant diffusivity is enhanced and leakage currents are increased in the final device.

The aim of this work is to study the energetic characteristics of small Si interstitials clusters from an atomistic point of view, determining their formation enthalpies and energy barriers for each cluster size. To do so, we have run simulations with the kinetic Activation-Relaxation Technique. We have characterized each geometrical configuration based on energetic and visual criteria, classifying the small clusters in chainlike, compact or $\{111\}$ configurations. The transition barriers between these structures have been also determined. This information is useful to understand the behaviour of small clusters in crystal and how they can evolve to extended defects. This study can also be applied to other semiconductor materials.

Keywords

Silicon, crystalline, small clusters, chainlike, compact, formation enthalpy, energy barrier, kART, simulation.

Resumen

La evolución de la industria micro y nano-electrónica se ha basado en el escalado continuado de los dispositivos, siguiendo la conocida Ley de Moore. La construcción de transistores con dimensiones cada vez más reducidas y con un menor consumo, requiere de un estricto control de la profundidad de los dopantes, y de la existencia de corrientes de baja intensidad. Los procesos de implantación iónica utilizados en la fabricación de dispositivos dan lugar a la formación de defectos de intersticiales en Si. Estos defectos afectan negativamente a las prestaciones de los dispositivos, pues son responsables del incremento en la difusión de los dopantes, de la degradación de la movilidad o del aumento de las corrientes de fuga. Durante los recocidos térmicos los defectos experimentan un mecanismo de Ostwald ripening, en el que los más grandes y estables absorben intersticiales liberados por los defectos de menor tamaño. Este crecimiento, motivado por la reducción de la entalpía de formación de los defectos con el tamaño, hace que los clusters de intersticiales se transformen en defectos extensos. Entre ellos destacan los defectos tipo $\{113\}$, que son cadenas en la dirección $\langle 110 \rangle$ a lo largo de planos $\{113\}$, y los lazos de dislocación en planos $\{111\}$ y $\{100\}$.

El papel fundamental que juegan los defectos en el comportamiento de los dispositivos ha propiciado el desarrollo de modelos para su formación, crecimiento y disolución, utilizados ampliamente en los simuladores de procesos de la industria microelectrónica. En ellos se asume que la energía de los defectos extensos se reduce monótonamente con su tamaño y para clusters pequeños se asigna una energía concreta a cada tamaño, y se considera que existen unos números mágicos, es decir, determinados tamaños que son especialmente estables y con una energía inferior. En todo caso, los modelos de evolución de defectos existentes hoy en día fallan en determinadas situaciones. Los clusters pequeños de intersticiales presentan un escenario mucho más rico que el considerado por los modelos clásicos. Un cluster de un determinado tamaño puede encontrarse en diferentes configuraciones, cada una caracterizada por una simetría y una entalpía de formación.

En particular, se han estudiado configuraciones tipo cadena alargadas en la dirección $\langle 110 \rangle$, configuraciones compactas en los planos $\{100\}$ o configuraciones simétricas en los planos $\{111\}$, que podrían ser las precursoras de defectos extensos tipo $\{113\}$ o de lazos de dislocación en los planos $\{111\}$ o $\{100\}$, respectivamente. La prevalencia en las primeras fases de un recocido térmico de una determinada configuración de clusters de intersticiales, podría condicionar el crecimiento de los defectos y favorecer la formación de un tipo concreto de defecto extenso. La inclusión de estos tres caminos de crecimiento permitiría el desarrollo de modelos de evolución de defectos más precisos. Para ello es necesario no solo conocer las entalpías de formación de cada configuración de clusters de intersticiales, sino las barreras de energía que controlan las transiciones entre las diferentes configuraciones. Así, en este trabajo se han determinado las barreras de energía para las transiciones estructurales entre las configuraciones tipo cadena, compacta y simétrica en el plano $\{111\}$ de clusters de intersticiales de Si, con tamaños de 2 a 5 intersticiales. Hasta la fecha, este es el único estudio en el que se calculan estas barreras, parámetros necesarios para la implementación de modelos más precisos de evolución de defectos.

El pequeño tamaño de los clusters de intersticiales hace muy difícil su caracterización experimental, por lo que se utilizan técnicas de simulación para su estudio. La técnica de dinámica molecular clásica (DMC) es especialmente adecuada, pues abarca las dimensiones espaciales y temporales características de la evolución de defectos. Así, se ha empleado esta técnica para obtener la entalpía de formación de los clusters de intersticiales, pero no permite determinar barreras de transición. La Técnica de Activación-Relajación cinética (*kinetic Activation-Relaxation Technique*, kART) ha sido diseñada para poder caracterizar los puntos de transición o *saddle points* en la evolución entre dos mínimos de energía del sistema, y obtener la barrera de energía de esa transición. Además, al incorporar el método de Monte Carlo cinético permite simular tiempos macroscópicos, inalcanzables con DMC, y acceder de este modo a barreras de energía elevadas.

Para ello, kART combina la Técnica de Activación-Relajación (*Activation-Relaxation Technique*, ART) con una clasificación topológica de eventos bajo un algoritmo de Monte Carlo cinético. En primer lugar, realiza un desplazamiento aleatorio de los átomos y lleva a cabo la clasificación de las diferentes topologías encontradas en la red y la generación de eventos a partir de ellas, cada uno con su barrera de energía asociada. Un evento lo componen los mínimos de energía inicial y final, y el punto de transición que ha de superarse, que determina la altura de la barrera. Seguidamente se procede a refinar aquellos eventos con barreras de energía pequeñas, pues serán los que presenten mayor probabilidad de ser seleccionados al ser aplicado el método Monte Carlo con posterioridad.

Todas las simulaciones se han llevado a cabo en celdas a 0 K y a 0 GPa, si bien se ha utilizado una temperatura de 500 K para el algoritmo Monte Carlo. Asimismo, se han modificado los parámetros adecuados de kART para lograr transiciones entre configuraciones. Se ha utilizado SEARCH_FREQUENCY a 50 para poder encontrar una mayor cantidad de eventos por topología e INCREMENT_SIZE a 0,3 Å para permitir mayores desplazamientos de los átomos. Se ha empleado el potencial interatómico Tersoff 3 para calcular las energías de la red aprovechando la compatibilidad de kART con las librerías del simulador de DMC LAMMPS (*Large-scale Atomic/Molecular Massively Parallel Simulator*).

Las configuraciones iniciales de los clusters han sido obtenidas gracias al trabajo de varios autores. Partiendo de una red cristalina perfecta, se inserta el número correspondiente de intersticiales en posiciones cercanas para cada tamaño de cluster, y se deja evolucionar el sistema a 1200 K durante 25 ns, mediante simulaciones de DMC. En este trabajo hemos utilizado como punto de partida los estados de mínima energía de cada una de las configuraciones obtenidas de esta forma. Para las simulaciones de kART del di- y el tri-intersticial se ha utilizado una celda cúbica de 6a de lado, con un total de $1728 + n$ átomos, mientras que para el tetra-, el penta- y el hexa-intersticial se ha empleado una celda cúbica con 8a de lado, conteniendo $4096 + n$ átomos, siendo n es el número de intersticiales insertados y a el parámetro de red del Si (5.432 Å). Como recursos computacionales se ha dispuesto de los clusters de ordenadores Linux del grupo de investigación MMM (*Multiscale Materials Modeling Group*). El coste CPU que han conllevado las simulaciones es elevado, alcanzándose un total de 6500 h.

Procediendo a analizar los resultados obtenidos, primero es necesario destacar que a lo largo de una simulación un cluster puede recorrer distintas configuraciones (cadena, compacta, simétrica en $\{111\}$) y presentar varias topologías y niveles de energía dentro de una misma configuración. Las transiciones entre los estados de mínima energía de las diferentes configuraciones no han sido nunca directas, sino que para evolucionar de una a otra se han atravesado estados de mayor energía. La barrera de energía para la transición entre dos configuraciones viene determinada por el punto de transición de mayor energía que se visita, y se calcula como la diferencia de energía entre el punto de transición y el estado de mínima energía de la configuración de partida.

Una vez determinadas las entalpías de formación de cada configuración para cada tamaño, se puede observar cómo tanto la configuración cadena como la $\{111\}$ disminuyen en energía con el tamaño, lo que favorece el crecimiento de estos tipos de defectos al reducirse la energía total del sistema, dando lugar eventualmente a la formación de defectos extensos. Esta tendencia no se cumple en todos los tamaños de la configuración compacta. Debido a la gran estabilidad (baja energía) del tetra-intersticial compacto el crecimiento al penta-intersticial no es favorable, pues da lugar a un aumento de energía. La configuración $\{111\}$ es la más energética de todas las estudiadas, aunque las diferencias con respecto al resto de configuraciones se reducen con el tamaño. En todo caso, conviene recordar que los lazos de dislocación habituales en Si se encuentran en planos $\{111\}$, lo que indica que debe existir la posibilidad de que los defectos $\{111\}$ aumenten en tamaño o bien que se llegue a ellos desde otras configuraciones de defectos.

Si analizamos las barreras de energía entre las configuraciones compacta y cadena, en el di-intersticial las barreras son similares, mientras que en el tri-intersticial la menor entalpía de formación y una barrera más elevada hacen que la configuración cadena prevalezca. Los clusters compactos con 4, 5 o 6 intersticiales son menos energéticos que las configuraciones cadena y además poseen barreras de salida elevadas. La baja energía del tetra-intersticial compacto unido a la elevada barrera de salida hacen que este defecto pueda ralentizar el crecimiento de los clusters. En cuanto a la configuración $\{111\}$, en general existen barreras reducidas para abandonarla pero hay que superar barreras elevadas para llegar a esta configuración desde los clusters tipo cadena o compacto. Todo esto indica que en los tamaños estudiados la configuración $\{111\}$ es inestable y difícil de alcanzar desde otras configuraciones, salvo para el caso del penta- y hexa-intersticial donde su barrera de salida es elevada, favoreciendo que permanezca en ella.

Así, se pueden extraer varias conclusiones importantes de este trabajo. Nuestros datos muestran que el tetra-intersticial compacto es muy estable y con barreras elevadas para su evolución, mientras que los clusters con simetría en planos $\{111\}$ son energéticos y en general evolucionan rápidamente a otras configuraciones. Hemos observado la existencia de barreras significativas incluso para abandonar configuraciones de alta energía, lo que aumenta su vida media y retrasa su transformación estructural en otras configuraciones, posibilitando así su crecimiento y abriendo nuevos caminos para la evolución de los defectos. Las barreras de energía proporcionadas en este trabajo son parámetros necesarios para el desarrollo de nuevos modelos de evolución de defectos de intersticiales en Si.

A mis padres

Acknowledgments

Foremost, I would like to express my sincere gratitude to my supervisor Dr. Pedro López Martín for the continuous support of my study and research, for his patience, motivation and enthusiasm. His guidance has helped me in all the research and writing time of this thesis.

I also thank the members of MMM research group, showing a mayor gratitude for the advices that Dr. Lourdes Pelaz gave me during this stage. In addition, I would like to express my appreciation to Dr. Maria Luz Rodríguez Méndez for the excellent welcome and constant concern about the master.

Special thanks to my friends and partners Javier Santos, Luis Antonio Domínguez and Antonio Pesquera. Their invaluable help and company have made easier the achievement of all proposed aims.

Y en muy especial medida, desearía dar las gracias a mis padres, Rosario y José Luis, así como a Andrea, tanto por su comprensión como por su esfuerzo y apoyo incondicional. Ellos me han soportado estoicamente en todos los momentos difíciles y son, sin ninguna duda, una parte esencial presente en este proyecto sin la cual no hubiera sido posible.

Lastly, I would like to extend my gratitude to everyone who has been involved in the development of this work in one way or another.

My sincere thanks to all of them.

Contents

1	Introduction	1
1.1	Motivation	2
1.2	State of art	7
1.3	Aims	13
1.4	Computational resources	13
1.5	Memory structure	17
2	Simulation method and tools	19
2.1	Introduction	20
2.1.1	Deterministic and stochastic methods	21
2.1.2	Classification according to system size and simulation time	21
2.2	Kinetic Activation-Relaxation Technique	26
2.2.1	Theoretical basis	26
2.2.1.1	Lanczós	28
2.2.1.2	DIIS	30
2.2.2	Comparison with similar techniques	31
2.2.2.1	MD extensions	31
2.2.2.2	Based on kMC algorithm	31

2.2.3	Review of the technique	37
2.2.3.1	ART nouveau	37
2.2.3.2	Operating mode of kART	41
2.2.3.3	Crucial implementation details	46
2.3	Molecular Dynamics	51
2.3.1	Brief description	51
2.3.2	Interatomic potentials	54
2.3.2.1	Stillinger-Weber potential	55
2.3.2.2	Tersoff potential	55
2.3.2.3	Environment-dependent interatomic potential	56
2.3.2.4	Comparative	56
2.3.3	Implementation with kART	57
2.4	Used tools	57
2.4.1	Visualization tools	57
2.4.2	Processing tools	59
2.4.3	Additional software	60
2.4.3.1	Communication tools	60
2.4.3.2	Graphic tools	61
3	Simulation procedure	63
3.1	Initial configurations collection	64
3.2	Relevant simulations parameters	65
3.3	Considered output files	67
3.4	Data processing	69
3.5	Determination of formation enthalpies and energy barriers	72

4	Configurations and diffusion mechanisms of Si self-interstitial	73
4.1	Introduction	74
4.2	Stillinger-Weber potential	75
4.2.1	Simulation methodology	76
4.2.2	Formation enthalpies	77
4.2.3	Energy barriers	77
4.2.4	Comparison with other authors	79
4.3	Tersoff potential	81
4.3.1	Simulation methodology	81
4.3.2	Tersoff singularities	82
4.3.2.1	Analysis of possible barriers	82
4.3.2.2	Simulations explanation	84
4.3.2.3	Flowchart of energy minima	87
4.3.3	Formation enthalpies	91
4.3.4	Energy barriers	91
4.3.5	Comparison with other authors	92
4.4	Conclusions	94
5	Energy properties of small interstitials clusters	95
5.1	Di-interstitial	96
5.1.1	Summary of simulations	96
5.1.2	Analysis of states and possible barriers	96
5.1.3	Formation enthalpies and energy barriers	99
5.1.4	Transition mechanism	100
5.2	Tri-interstitial	101

5.2.1	Summary of simulations	102
5.2.2	Formation enthalpies and energy barriers	102
5.2.3	Transition mechanism	104
5.3	Tetrainterstitial	105
5.3.1	Summary of simulations	105
5.3.2	Formation enthalpies and energy barriers	106
5.3.3	Transition mechanism	108
5.4	Pentainterstitial	110
5.4.1	Summary of simulations	110
5.4.2	Formation enthalpies and energy barriers	110
5.4.3	Transition mechanism	112
5.5	Hexainterstitial	114
5.5.1	Summary of simulations	114
5.5.2	Formation enthalpies and energy barriers	115
5.6	Conclusions	117
6	Growth and evolution of small interstitial clusters	119
6.1	Formation enthalpies	120
6.2	Energy barriers	123
6.3	Emission and capture of interstitials	126
6.4	Pressure effect	129
6.4.1	Summary of simulations	129
6.4.2	Formation enthalpies and energy barriers	130
6.5	Conclusions	133

7	Conclusions and future research lines	135
7.1	Conclusions	136
7.2	Future research lines	137
A	Library of defects configurations	139
A.1	I_2	141
A.1.1	Compact configurations	141
A.1.2	Chainlike configurations	146
A.2	I_3	149
A.2.1	Compact configurations	149
A.2.2	Chainlike configurations	152
A.2.3	$\{111\}$ configurations	155
A.3	I_4	156
A.3.1	Compact configurations	156
A.3.2	Chainlike configurations	158
A.3.3	$\{111\}$ configurations	163
A.4	I_5	164
A.4.1	Compact configurations	164
A.4.2	Chainlike configurations	167
A.4.3	$\{111\}$ configurations	172
A.5	I_6	173
A.5.1	Compact configurations	173
A.5.2	Chainlike configurations	175
A.5.3	$\{111\}$ configurations	177

B	Examples of input files	179
B.1	Input file in kART	179
B.2	Initial atomic positions in kART	186
B.3	Input file in LAMMPS	187
C	Analysis programs (C/C++)	189
	Bibliography	221

List of Figures

1.1	Projected scaling results from last ITRS documents	3
1.2	Number of published articles in scientific journals and patents about <i>Silicon</i> and <i>Defects</i> from 1980 to the present day according to <i>Google Scholar</i>	4
1.3	TEM image of {311} defects seen through cross-section high-resolution micrograph and WBDF TEM images of dislocation loops from samples implanted with 150 keV Ge ⁺ ions to a dose of 2 · 10 ¹⁵ ions/cm ² after annealing at 900° in a N ₂ ambient	5
1.4	Schematic diagram of the behaviour of Si interstitial defects as a function of implant dose and thermal budget	8
1.5	Formation energy of the different types of interstitial agglomerates as a function of size	9
1.6	Predicted minimum energy configurations for (I ₅₋₁₂) defects in Si for [1-11] and [001] orientations with the associated symmetry (Grey: Bulk atoms, Gold: Displaced atoms)	10
1.7	Evolution map for self-interstitial aggregation as a function of cluster size and temperature/pressure	12
1.8	Detailed technical information of Beta cluster by <i>Ganglia cluster Toolkit</i>	14
1.9	Detailed technical information of Gamma cluster by <i>Ganglia cluster Toolkit</i>	15
1.10	Picture of Beta and Gamma clusters and their UPSs taken in the Simulation Laboratory of the Higher Technical School of Telecommunications Engineering	15

1.11	Instance of activity report of each Beta node by <i>Ganglia cluster Toolkit</i> .	16
1.12	Instance of aggregate load in Beta cluster by <i>Ganglia cluster Toolkit</i>	16
2.1	Connexion between theory, experiment and simulation	20
2.2	Simulation techniques and experimental methods depending on size and time scales	25
2.3	Sample of an energy landscape showing a saddle point bewteen two adjacent energy minima linked by the minimum energy path	28
2.4	Simplified workflow for on-the-fly kMC algorithms	35
2.5	Sketch of minimum energy pathways from various first order saddle points to the local energy minimum in ART	38
2.6	Perpendicular hyperplane to the initial energy minimum designed with Matlab	39
2.7	Example of topological analysis in kART using NAUTY	42
2.8	Refinement of generic events into specific events in kART	44
2.9	Example of events reconstruction in kART	45
2.10	Example of kART trapping between two states with low energy barrier between them	46
2.11	Construction of a basin formed by state A and states A and B with their possible events.	48
2.12	Meaning of PBC in a 2D cell	50
2.13	Representation of Tersoff potential for R=2.95 and D=0.15	52
2.14	Simplified calculating method in MD simulations	53
2.15	Sample of representation with RasMol	58
2.16	Screenshot of <i>File Transfer Client</i>	60
2.17	Sample of flowchart with GVedit	61

3.1	Snapshot of <i>allconf</i> output file of kART	67
3.2	Snapshot of <i>Energies.dat</i> output file of kART	67
3.3	Snapshot of <i>Diffusion.dat</i> output file of kART	68
3.4	Example of energy minima identification in formation enthalpy evolution at 500 K and 0 GPa beginning from a compact I ₃ configuration	69
3.5	Example of configurational identification in formation enthalpy evolution at 500 K and 0 GPa with <i>MIN_SIG_BARRIER</i> at 0.6 eV beginning from a compact I ₃ configuration	71
4.1	XY, XZ, and YZ projections of the different Si self-interstitial configurations	74
4.2	First 20 steps of the energy evolution of our SW based simulation beginning from a T configuration and labelling all the encountered configurations .	76
4.3	Relative frequency of energy barriers for E → DE, DE → E and DE ↔ DE transitions using the SW potential	78
4.4	Diffusion mechanisms for the self-interstitial using the SW potential . . .	79
4.5	Diffusion mechanisms for the self-interstitial of this work compared with P. Ganster <i>et al.</i> and M. Trochet <i>et al.</i> results	80
4.6	First 30 steps of the energy evolution of the Tersoff based simulation at 500 K beginning from a T configuration, labelling the visited minima . .	81
4.7	Grouping of same energy possible events from the lowest energy T configuration taking it as a reference and frequency of the barriers for these events	83
4.8	Grouping of same energy possible events from the second and the third lowest energy T configuration taking them as a reference	84
4.9	Steps 660 through 690 of the energy evolution of Tersoff based simulation with <i>MIN_SIG_BARRIER</i> to 0.7 eV at 500 K beginning from a T configuration and labelling all the encountered configurations	85

4.10	Frequency of the energy barriers for the possible events from T configuration with INCREMENT_SIZE set to 0.3 Å at 500 K and an example of our problem with low returning energy barriers (b)	85
4.11	Steps 160 through 190 of the energy evolution of Tersoff based simulation with MIN_SIG_BARRIER to 0.3 eV and INCREMENT_SIZE to 0.3 Å at 1000 K beginning from a T configuration and labelling all the encountered configurations	86
4.12	First 30 steps of the energy evolution of Tersoff based simulation with MIN_SIG_BARRIER to 0.9 eV at 500 K beginning from a E configuration and labelling all the encountered configurations	87
4.13	Flowchart of energy minima ranked by energy and connected with their respective minimum energy barriers at 500 K	89
4.14	Flowchart of energy minima ranked by energy and connected with their respective minimum energy barriers at 1000 K	90
4.15	Diffusion mechanism for the Si self-interstitial using the Tersoff potential	92
4.16	Diffusion mechanisms for the Si self-interstitial of this work compared with Luis A. Marqués <i>et al.</i> [88] and Iván Santos <i>et al.</i> [90] results	93
5.1	Formation enthalpies and relative frequency of energy minima for each I ₂ configurations and their role in transitions	97
5.2	Frequency of the energy barriers for the possible events from Ch8, Co4, Co7 and Co9 I ₂ energy minima and final states associated	98
5.3	Transition path between compact and chainlike I ₂ configurations indicating the formation enthalpies, topologies in [-1 0 0] plane and all energy barriers found	99
5.4	DAs of chainlike and compact I ₂ configurations in [-1 0 0] plane	100
5.5	Chainlike to compact I ₂ configuration evolution in [-1 0 0] plane over eight kMC steps	101

5.6	Transition path between compact, chainlike and $\{111\}$ I_3 configurations indicating the formation enthalpies, topologies in $[-1\ 0\ 0]$ plane for the first two and in $[-1\ 1\ -1]$ for the third and all energy barriers found	103
5.7	DAs of chainlike, compact and $\{111\}$ I_3 configurations in $[-1\ 0\ 0]$ plane and $[-1\ -1\ -1]$ plane	104
5.8	Compact to chainlike I_3 configuration evolution in $[-1\ 0\ 0]$ and $[0\ 1\ 0]$ planes over four selected kMC steps	105
5.9	Transition path between compact, long and short chainlike and $\{111\}$ I_4 configurations indicating the formation enthalpies, topologies in $[-1\ 0\ 0]$ plane for the first three and in $[-1\ 1\ -1]$ for the fourth and all energy barriers found	107
5.10	DAs of short and long chainlike, compact and $\{111\}$ I_3 configurations in $[-1\ 0\ 0]$ plane and $[-1\ -1\ -1]$ plane	107
5.11	Short chainlike to compact I_4 configuration evolution in $[0\ 0\ -1]$ plane over eight selected kMC steps	109
5.12	Short chainlike to long chainlike I_4 configuration evolution in $[0\ 0\ -1]$ plane over eight selected kMC steps	109
5.13	Transition path between compact, long and short chainlike and $\{111\}$ I_5 configurations indicating the formation enthalpies, topologies in $[-1\ 0\ 0]$ plane for the first three and in $[-1\ 1\ -1]$ for the fourth and all energy barriers found	111
5.14	DAs of short and long chainlike, compact and $\{111\}$ I_3 configurations in $[-1\ 0\ 0]$ plane and $[-1\ -1\ -1]$ plane	112
5.15	Short chainlike to compact I_5 configuration evolution in $[-1\ 0\ 0]$ plane over eight selected kMC steps	113
5.16	Short chainlike to long chainlike I_5 configuration evolution in $[-1\ 0\ 0]$ plane over eight selected kMC steps	114
5.17	Transition path between chainlike and $\{111\}$ I_6 configurations indicating the formation enthalpies, topologies in $[-1\ 0\ 0]$ plane for the first and in $[-1\ 1\ -1]$ for the second and all energy barriers found	116

5.18	DAs of chainlike, compact and $\{111\}$ I_6 configurations in $[-1\ 0\ 0]$ plane and $[-1\ -1\ -1]$ plane	116
6.1	Energy per interstitial for each configuration of small interstitials clusters	120
6.2	Annealing at 850 K of a cell which contains different types of big defects ($\{111\}$ RLDs and (110) chains) and compact I_4 and his agglomerates in 170 ns	121
6.3	Comparison of energy per interstitial for each configuration of small interstitials clusters with other authors	122
6.4	Evolution of energy barriers in transitions between compact and chainlike configuration for each cluster size	124
6.5	Evolution of energy barriers in transitions between $\{111\}$ and compact and chainlike configuration for each cluster size	125
6.6	Mean time for transformation, emission and capture of an interstitial for chainlike, compact and $\{111\}$ configurations	128
6.7	Transition path between compact and chainlike I_3 configurations showing the formation enthalpies and all energy barriers found at -3 GPa and 3 GPa	131
6.8	Energy barriers from compact to chainlike I_3 configurations as a function of pressure	131
6.9	Transition path between compact and chainlike I_3 configurations indicating the formation enthalpies and all energy barriers found at -3 GPa and 0 GPa	132
6.10	Evolution of the energy difference between compact and chainlike configuration for all cluster sizes and pressures we have studied	132
A.1	First minimum energy configuration for compact I_2	141
A.2	Second minimum energy configuration for compact I_2	142
A.3	Third minimum energy configuration for compact I_2	143
A.4	Fourth minimum energy configuration for compact I_2	144
A.5	Eighth minimum energy configuration for compact I_2	145

A.6	First minimum energy configuration for chainlike I_2	146
A.7	Third minimum energy configuration for chainlike I_2	147
A.8	Fourth minimum energy configuration for chainlike I_2	148
A.9	First minimum energy configuration for compact I_3	149
A.10	Second minimum energy configuration for compact I_3	150
A.11	Ninth minimum energy configuration for compact I_3	151
A.12	First minimum energy configuration for chainlike I_3	152
A.13	Fifth minimum energy configuration for chainlike I_3	153
A.14	Tenth minimum energy configuration for chainlike I_3	154
A.15	{111} configuration for I_3	155
A.16	First minimum energy configuration for compact I_4	156
A.17	Second minimum energy configuration for compact I_4	157
A.18	First minimum energy configuration for short chainlike I_4	158
A.19	Second minimum energy configuration for short chainlike I_4	159
A.20	First minimum energy configuration for long chainlike I_4	160
A.21	Third minimum energy configuration for long chainlike I_4	161
A.22	Fifth minimum energy configuration for long chainlike I_4	162
A.23	{111} configuration for I_4	163
A.24	First minimum energy configuration for compact I_5	164
A.25	Third minimum energy configuration for compact I_5	165
A.26	Seventh minimum energy configuration for compact I_5	166
A.27	First minimum energy configuration for short chainlike I_5	167
A.28	Sixth minimum energy configuration for short chainlike I_5	168
A.29	Twelfth minimum energy configuration for short chainlike I_5	169

A.30	First minimum energy configuration for long chainlike I_5	170
A.31	Second minimum energy configuration for long chainlike I_5	171
A.32	{111} configuration for I_5	172
A.33	First minimum energy configuration for compact I_6	173
A.34	Second minimum energy configuration for compact I_6	174
A.35	First minimum energy configuration for chainlike I_6	175
A.36	Third minimum energy configuration for chainlike I_6	176
A.37	First {111} configuration for I_6	177
A.38	Second {111} configuration for I_6	178

List of Tables

2.1	Considered entities according to the size and simulation time for different models	21
2.2	Technical comparison between k-ART and Self-evolving atomistic kMC .	36
2.3	Tersoff 3 parameters	55
4.1	Formation enthalpies averaged for each self-interstitial configurations predicted by the SW potential	77
4.2	Energy barriers averaged for each transition by using the SW potential .	77
4.3	Formation enthalpies for several Si self-interstitial configurations as found in the literature compared with our work using the SW potential	80
4.4	Formation enthalpies averaged for each Si self-interstitial configurations predicted by the Tersoff potential	91
4.5	Formation enthalpies for several Si self-interstitial configurations as found in the literature compared with our work using the Tersoff potential . . .	93
5.1	Summary of simulations which have been run for di-interstitial	96
5.2	Summary of simulations which have been run for tri-interstitial	102
5.3	Summary of simulations which have been run for tetrainterstitial	106
5.4	Summary of simulations which have been run for pentainterstitial	110
5.5	Summary of simulations which have been run for hexainterstitial	115

6.1	Statistical analysis of all energy barriers for each cluster size	123
6.2	Estimated time needed at 1200 K to see the transitions of figure 6.4 . . .	125
6.3	Estimated time needed at 1200 K to see the transitions of figure 6.5 . . .	126
6.4	Summary of simulations which have been run for tri-interstitial at -3 and 3 GPa	129
6.5	Summary of simulations which have been run for tetrainterstitial at -3 and 3 GPa	130

Chapter 1

Introduction

*There is plenty of room at the bottom.
~ Richard Feynman ~*

THE AIM OF THIS WORK IS to study the growth and transitions of small self-interstitial clusters in crystalline Si, analysing their formation enthalpy and the energy barriers between different configurations, which are characterized by a certain spatial arrangement of atoms.

Si is at the heart of the semiconductor industry, being implemented in every generation of the transistor scaling over the last decades. During the fabrication process of devices dopant atoms are usually introduced by ion implantation, which is followed by the subsequent annealing to remove the damage. This thermal process implies the diffusion of impurity atoms, which is critically influenced by intrinsic point defects such as self-interstitials and vacancies. Point defects tend to aggregate forming extended defects and therefore, the dopant diffusivity is enhanced and leakage currents can exist in the final device. Both are common problems to be addressed and investigated.

To carry out this work, atomistic simulations have been run using kART (*kinetic Activation-Relaxation Technique*). It is an off-lattice, self-learning, on-the-fly method which identifies and evaluates activation barriers using ART nouveau and a topological description of events, enabling us to access the time and size scales typical of defect evolution.

1.1 Motivation

Looking back at when the integrated circuit was invented, the price of a single Si transistor was about \$10. Today, however, one can buy more than 50 thousand million transistors with the same budget. In fact, a current transistor represents less cost than the ink for one letter of newsprint. And this is only an example of the economic impact of the device scaling. This progress could not be achieved without the work of very capable engineers and scientists who have reinvented and improved every step of the fabrication process, from the preparation of pure Si materials to the final packaging operation.

After elapsing some years from the first stages of semiconductor device development when the bipolar transistor was the most widespread device, Si MOSFET (*Metal Oxide Semiconductor Field Effect Transistor*) became the preferred device type in the 1970s. It has eventually grown to dominate the use of semiconductor devices in integrated circuits due to the large scale of integration and the development of CMOS (*Complementary MOS*) technology [1].

The competitive drive for improved performance and cost reduction of CMOS based devices has resulted in the scaling of each generation of circuit elements to ever-smaller dimensions, as it can be seen in figure 1.1. Hence, MOSFET dimensions have shrunk from a gate length of 5 μm in its origin to 10 nm today, i.e., in only 45 years. Furthermore, according to what Intel has already forecast, the 5 nm node will be reachable in about 2020.

According to the ITRS (*International Technology Roadmap for Semiconductors*) [2] predictions, we are attending to the end of Moore's Law due to the nanometer effects which have appeared in this level, highlighting the sub-threshold leakage current, the threshold voltage variation and the carrier mobility degradation in the channel, the direct tunnelling leakage current and the depletion in the gate, and the parasitic resistance and capacitance between source and drain. Although some alternatives have been proposed, such as spintronics, carbon based devices or quantum electronics, the semiconductor industry is struggling to extend de miniaturization of CMOS devices, fighting against processing and physics limits. What strikes more about it is that they are achieving their aim thanks to the effort and creativity of each generation, despite of each one used to be seen as the last one.

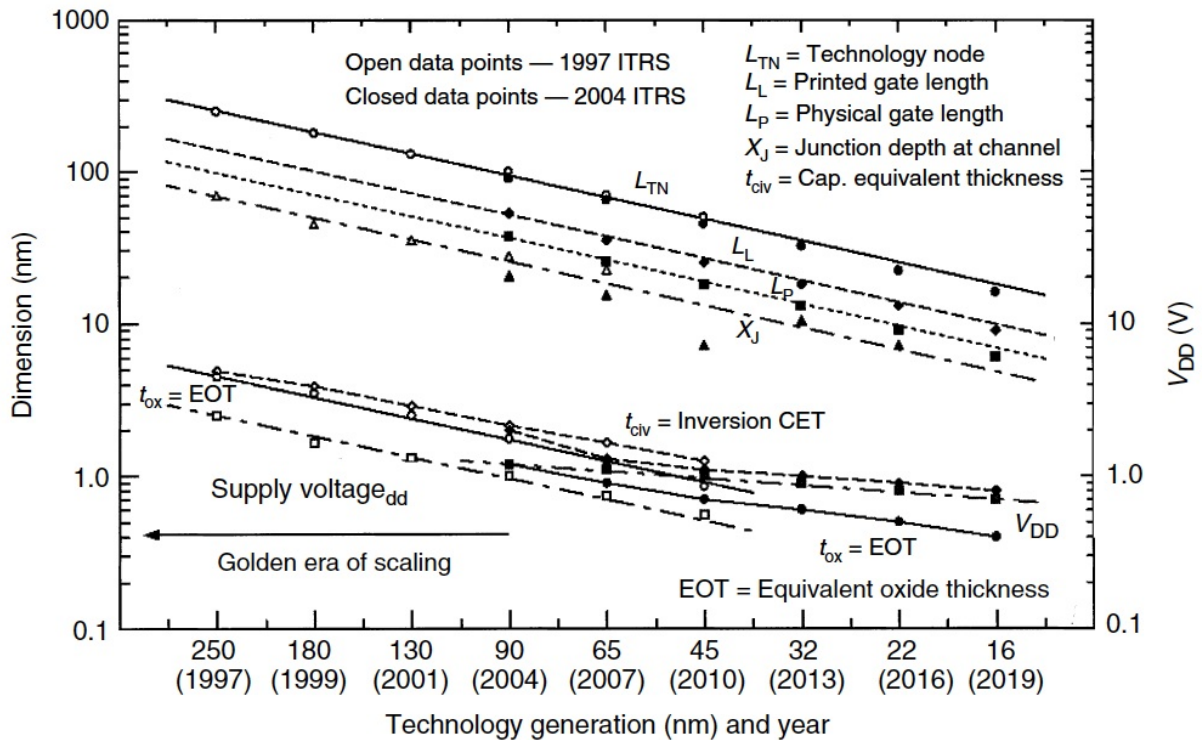


Figure 1.1: Projected scaling results from last ITRS documents [1].

In any case, Si is the main material in the microelectronics industry due to its terrestrial abundance and the great stability and ease of production of its oxide, and therefore a deeper understanding of it is essential for the fabrication of next generations of semiconductor devices. Ion implantation is the main doping technique in Si based integrated circuit manufacturing, however, it gives rise to a damage in the semiconductor lattice that has to be removed with a subsequent annealing. During this thermal budget it has become notorious the diffusion of dopant atoms, one of the main problems in the manufacture of sub-micron devices, limiting how small they can be made [1].

A knowledge of diffusion on the microscopic scale in situations far from equilibrium is required in order to understand these effects. The diffusion of impurity atoms in Si is critically influenced by intrinsic point defects such as self-interstitials and vacancies, which nucleate into extended defects, ripen and eventually dissolve in the annealing. Consequently, it is of great importance to improve our understanding of the behaviour of these defects, which play a crucial role on the dopant diffusion during device processing. Additionally, figure 1.2 verify the interest in this area, whose number of papers have been increased since 1980.

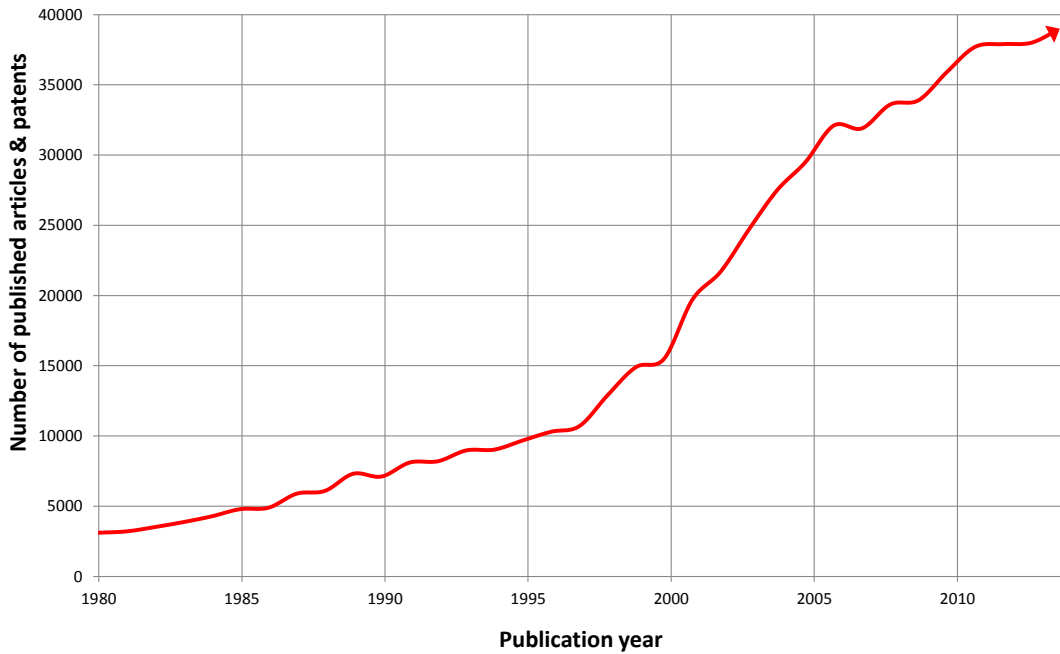


Figure 1.2: Number of published articles in scientific journals and patents about *Silicon* and *Defects* from 1980 to the present day according to *Google Scholar*.

The extended defects can present various types of atomic arrangement, being all of them widely known in the literature and they are represented in figure 1.3. Among them, it should be emphasized the $\{311\}$ defects [3], which are chains in the $\langle 111 \rangle$ direction in $\{311\}$ planes, and the dislocation loops in planes $\{111\}$ [4]. In addition, dislocation loops have been observed in planes $\{100\}$ at high temperatures annealing [5]. Although they are very frequent in Ge, Si has presented them recently [6]. These defects are energy favourable, having lower free energy than individual interstitials. It gives them metastability, being dissolved with continued annealing at high temperature. However, as the dissolution takes place, they release excess interstitials into the lattice, enhancing the dopant diffusivity and also giving rise to leakage currents in the final device, which are common problems in the manufacturing process [7,8].

Conversely, implantation damage has recently developed a new field of Si based optoelectronic devices. Although the Si indirect band-gap has generally limited the optical applications, optically active small defect clusters and extended defects are being explored for efficient light emission in Si, which has attracted much attention. Converting Si into a light-emitter semiconductor will make optoelectronics take advantage of the Si microelectronic industry technology, and will result in a large reduction of the fabrication costs of their devices [9].

Since the size of today devices is nanometer scale and their profiles are very abrupt, the current requirements demanded by the industry need a minimum movement of dopant atoms due to thermal processes. Hence, it is needed to predict the right temperature and time to vanish these defects. To do so, one has to know the physical behaviour of these defects and also their geometrical configurations, which is especially important for extended defects.

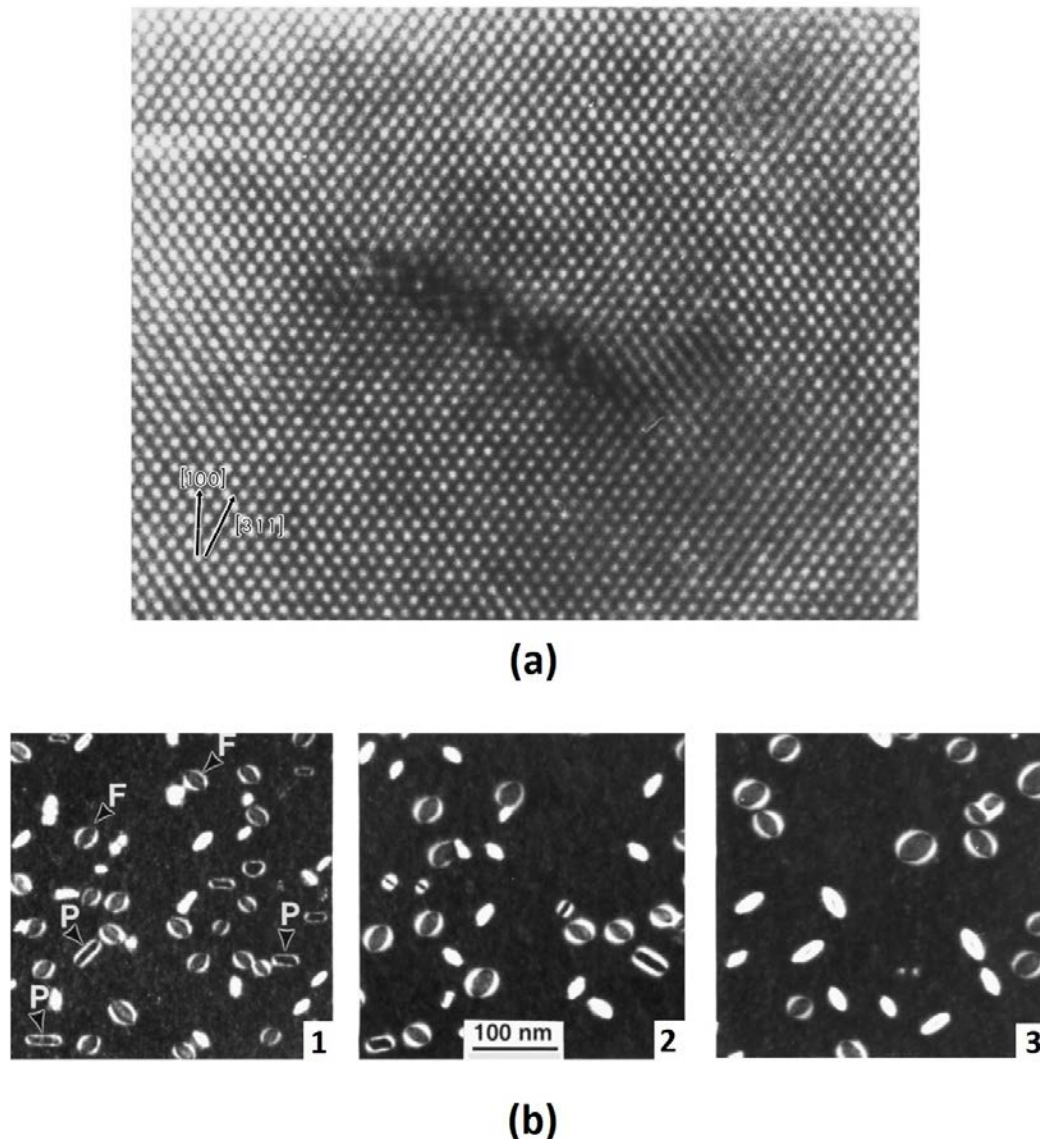


Figure 1.3: TEM image of $\{311\}$ defects seen through cross-section high-resolution micrograph (a) and WBDF TEM images of dislocation loops from samples implanted with 150 keV Ge^+ ions to a dose of $2 \cdot 10^{15}$ ions/cm² after annealing at 900° in a N_2 ambient in (1) $t=10\text{s}$, (2) $t=100\text{s}$ and (3) $t=400\text{s}$ (F: Faulted, P: Perfect) (b) [8].

There has been a lot of driven force to understand this kind of defects. The $\{311\}$ s are usually associated with the transient enhanced diffusion of dopant atoms such as Boron, presenting stability thanks to their spatial configuration. On the other hand, when the doses and the temperature are even higher one can find dislocation loops, which can be faulted or perfect. The first type consist of a circular stacking surrounded by a dislocation line whereas the second one refers to an extra plane of atoms instead of the stacking and are elongated through $\langle 110 \rangle$ direction, having less formation enthalpy than the faulted configuration.

The growing mechanisms of the previous defects are quite curious. When the temperature goes higher, the morphology of the defects is evolving. Firstly, the point defects, thanks to their high mobility, group into small clusters, characterized by being almost motionless. Then it happens the maturity process: the small clusters begin to dissolve and vanish whereas the big ones end up in a $\{311\}$ configuration. Additionally, it is believed that in case the $\{311\}$ s are next to the surface and the doses are high, they can evolve to an imperfect dislocation loop and, eventually, to a perfect one. However, the transition between both configurations remains unclear, giving rise to a lot of controversy [8].

At this point we need to point out that it is likely that certain small clusters configurations are the previous stage of the bigger ones we have already described and illustrated. In this sense (110) chainlike defects can be the precursor of the $\{311\}$ defects, compact configurations can be the herald of $\{100\}$ dislocation loops and symmetrical in $\{111\}$ plane can be the precursor of $\{111\}$ dislocation loops. Although significant experimental and theoretical efforts have been made to determine the structure and stability of small self-interstitial clusters as well as their growth to larger extended defects, it is still unclear. In any case, these small aggregations can also play an important role in the already mentioned transient enhanced diffusion and, in the end, they may act as a restriction in the device scaling [10].

Unfortunately, the small clusters of interstitials cannot be detailed experimentally using structural techniques, such as TEM (*Transmission Electron Microscopy*). Their instability together with the technical difficulties to measure many defects at the same time, makes the interpretation of the results difficult. Hence, the identification of such defects mostly relies on spectroscopic techniques as electron paramagnetic resonance, deep level transient spectroscopy, photoluminescence and infrared absorption. Nevertheless, one can address the problem thanks to the modelling of materials and the computational

advances we can dispose nowadays, being able to provide a more detailed description. It is here when the relevance of the atomistic simulation techniques should be stressed.

Regarding to the interest of our technique, kART, which is detailed in chapter 2, we should mention that is a novel powerful tool in the field. Standard kinetic Monte Carlo, which was developed in the 1970's and applied to materials science in the end of the 1980's, is limited to on-lattice configurations. This meant very limited applications in the study of semiconductors, alloys, interfaces and, in general, complex systems where it is not possible to identify diffusion mechanisms beforehand and off-lattice positions and elastic effects are important. This context gives rise to kART, a very solid code that can now produce exciting new science.

In fact, kART is the only kinetic Monte Carlo method that can be applied to disordered or complex materials such as ion-bombarded crystal, amorphous semiconductors and glasses at the moment. As such, the method opens new fields of simulations and it is attracting considerable attention that should be growing with the publication of recent atomistic simulations of the evolution of complex systems over time scales of 1 second or longer, more than 10 million times longer than anything available until now. Summarizing the technique very briefly, kART is a powerful both on-the-fly and off-lattice kinetic Monte Carlo method for generating events and calculating energy barriers while retaining a complete description of long-range elastic interactions due to the discretization of events even for disordered systems.

Thus, in the light of the above it should be clear that self-interstitial clustering in Si has attracted the curiosity of a large number of researchers around the world. In our case, to conclude this section, we should stress that this area is one of the research lines of the group in which has been developed this work (see section 1.4 for more information).

1.2 State of art

The current bibliography about small clusters in Si is not quite rich if we compare it with what has been done about extended defects. This section is going to outline the main studies which have analysed the formation enthalpy and the energy barrier between different configurations of small clusters, giving an insight into extended defects transitions due to their interest in our work.

We have already mentioned that, under certain conditions, which are not clear by now, the transition between $\{311\}$ s to dislocation loops takes place, as it can be seen in figure 1.4. Although it is experimentally obvious that the implantation dose must be high in order to achieve the dislocation loops structure and the transition is not instantaneous, one can find several kinetic Monte Carlo simulation models which affirm that either the defect size or a fixed energy threshold determine this configurational change.

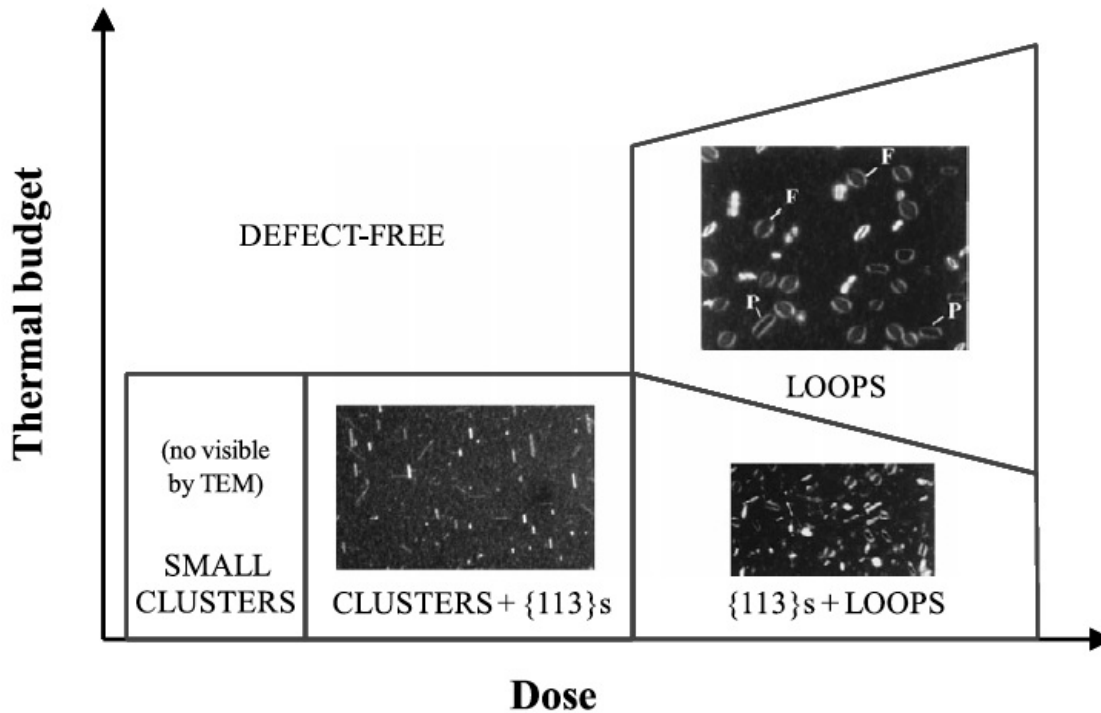


Figure 1.4: Schematic diagram of the behaviour of Si interstitial defects as a function of implant dose and thermal budget [9].

In this sense, some authors have proposed models which assume that the transformation rate does not depend on defect size, describing correctly the nucleation and evolution of the dislocation loops in amorphizing implants but predicting their formation also for low-dose implants due to this independence [11]. On the other hand, others support that the transformation rate must depend on a fixed threshold, agreeing with the experimental dose and describing properly the overall evolution of the $\{311\}$ and dislocation loops populations [12]. This idea, however, implies that the transformation is almost instantaneous and eludes whether or not the defects nucleate. Furthermore, there is a third approach that tries to combine the advantages from the previous ones, i.e., a size dependent energy barrier between both configurations [13].

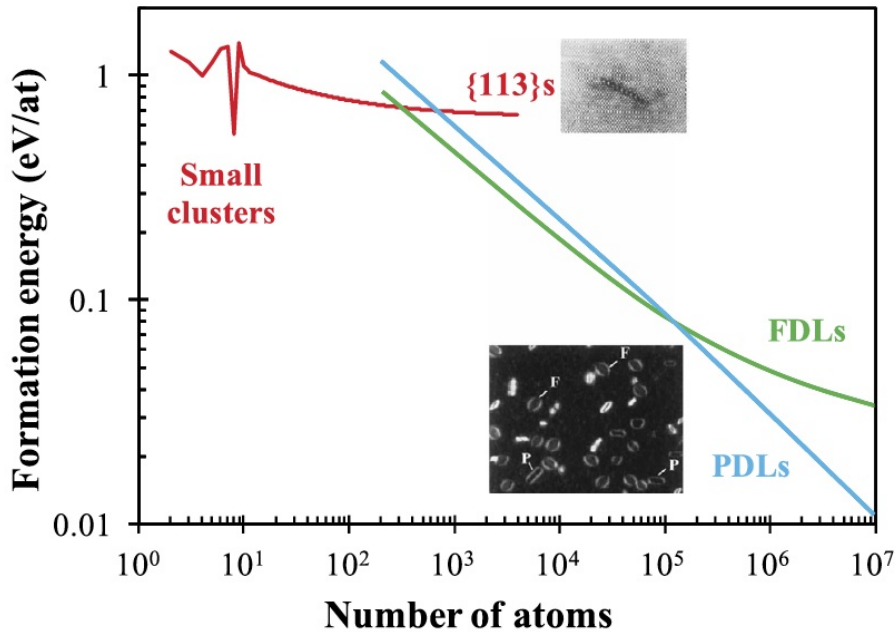


Figure 1.5: Formation energy of the different types of interstitial agglomerates as a function of size [9].

Regarding the formation enthalpy of each extended defect configuration, figure 1.5 shows the results of Cristiano *et al.* [14], who determined it as a function of size by an Ostwald Ripening model. It should be noted that the microstructural evolution of the material can be extracted from it. N. Cowern *et al.* [15] carried out the calculation of the energies of small clusters with the same model. It was one of the first attempts to characterize energetically these defects, though, it does not take into account the different structures which appear for the same number of interstitials. In any case, the energy oscillations of small aggregations are due to the fact that specific small cluster sizes have been marked as magic numbers, which correspond to stable configurations where atoms remain fourfold coordinated [9].

In this sense, there has been a large number of authors who have determined the minimum energy configurations of small interstitial clusters in Si by first principles calculations [16–19], highlighting the work of Kim *et al.* [20] and A. Bongiorno *et al.* [21]. They were pioneers in distinguishing between chainlike and compact configurations for clusters sizes smaller than 10 interstitials and also proposed a theory about the growth of chainlike to rodlike $\{311\}$ configurations, which is believed to stabilize as it elongates in the chain direction. Moreover, the stable configuration is found to be compact for small clusters, elongated for medium clusters, and planar for large clusters.

However, Sangheon Lee and Gyeong S. Hwang are the ones who have performed the most detailed analysis of the structure and energy of these defects, discretizing between chainlike and compact configurations, through a combination of CRN-MMC (*Continuous Random Network model based Metropolis Monte Carlo*), TBMD (*Tight Binding Molecular Dynamics*) and DFT (*Density Functional Theory*) calculations [22–24].

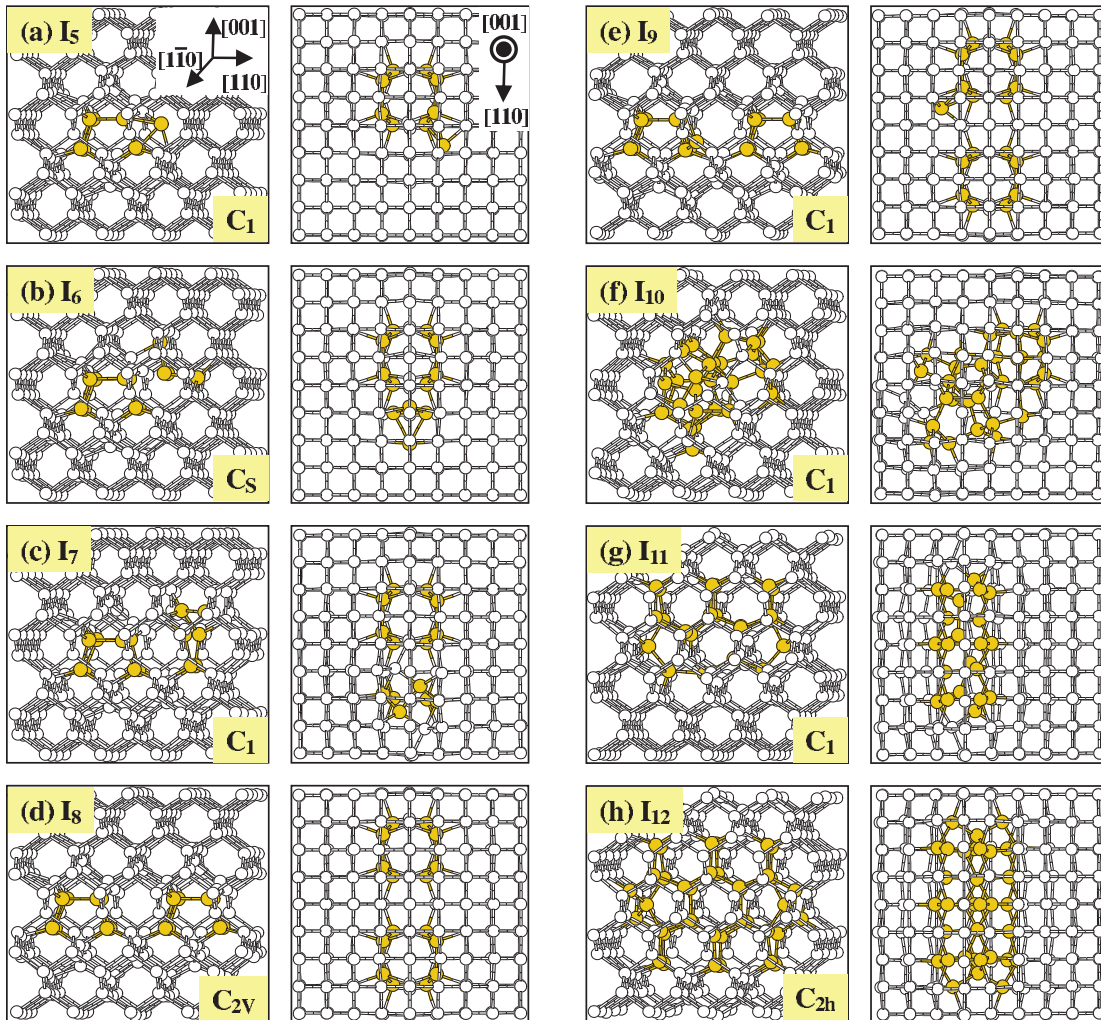


Figure 1.6: Predicted minimum energy configurations for (I_{4-12}) defects in Si for $[1-11]$ and $[001]$ orientations with the associated symmetry (Grey: Bulk atoms, Gold: Displaced atoms) [23].

After analysing the formation energies and the structures of self interstitial clusters up to a size of 16 atoms of both chainlike and compact configurations, they conclude that small clusters tend to favour compact structures, but the compact geometry is no longer

energetically favourable when the cluster size is greater than 10 interstitials. Additionally, this study also emphasizes that tetrainterstitial (I_4) and octainterstitial (I_8) compact clusters would kinetically or/and thermodynamically inhibit the formation of chainlike clusters. The higher symmetry of fourfold configurations can be seen in figure 1.6, which also shows minimum energy configurations for each size from 5 to 12 interstitials.

Based on their results, they also try to forecast the compact to elongated transition, giving two possible growth mechanisms. The first one may involve the evolution from the fourfold compact to chainlike elongated structures at I_{10} configuration, followed by the capture of additional interstitials at $\langle 110 \rangle$ edges. The second one, however, is based on the elongated fourfold I_{11} and I_{12} . Since these configurations are quite well ordered and present high thermal stability, the interconversion between them and the chainlike ones is not likely to happen. Furthermore, they also study the growth of these small defects to $\{311\}$ s, assuring that the $\{311\}$ core structure becomes favoured when the cluster size is greater than 20 atoms, below which the fourfold I_{12} seems to be prevail.

Once having characterized the small clusters, R.J. Bondi *et al.* were interested in the variation of their energies under different strain fields (uniaxial and biaxial) through DFT calculations [25, 26]. They focused on clusters smaller than 4 atoms and simulated the cell with positive and negative strain ($-4\% \leq \varepsilon \leq 4\%$), revealing that the minimum energy configuration is a function of the strain conditions presented in the system. In addition, they confirm qualitatively that interstitial clusters with fourfold coordination generally introduce less changes in and around the Si band-gap than clusters without complete fourfold coordinated bond topology by analysing its total density of states.

On the other hand, S.S. Kapur *et al.* have studied the formation energy of the small defects identified in figure 1.6 in terms of large-scale Molecular Dynamics simulations at 1100 K based on the empirical Environment-Dependent Interatomic Potential [27–29], taking into account the vibrational and configurational entropies. As such, launching conjugated gradient minimization at constant volume or pressure they get the energy values. Furthermore, they have computed the probability distribution functions of cluster formation energies at 1100 K for different pressures ($-1\% \leq \varepsilon \leq 1\%$, i.e., from -3 to 3 GPa). On this basis and considering also the large vibrational entropy of each cluster size, they have outlined the growth mechanism as figure 1.7 shows. An important idea which can be extracted from it is that the evolution of cluster morphology with size depends strongly on both temperature and stress, whose effects have entropic means.

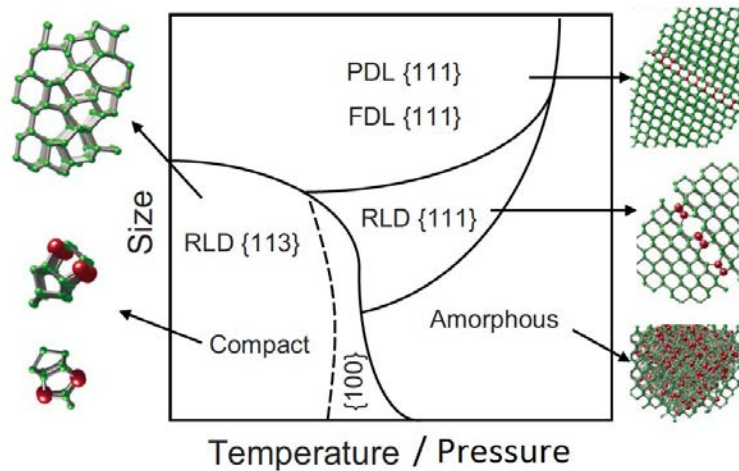


Figure 1.7: Evolution map for self-interstitial aggregation as a function of cluster size and temperature/pressure [29].

In this approach, they also used the Tersoff potential and contrasted it with Environment-Dependent Interatomic Potential, concluding that the potential landscape associated with self-interstitial clusters of the last one may be rougher than the Tersoff one. In other words, it contains a larger number of local minima, making it more difficult for a single configuration to dominate even if it is entropically favourable.

C.Y. Chuang *et al.* have recently made a comprehensive work obtaining the formation enthalpies of clusters whose size ranges from 2 to 150 interstitial through molecular dynamics based on the Erhart-Albe parametrization of the Tersoff potential [30]. Their simulation procedure was the same as the previous one but extending their work for temperatures from 1500 to 2500 K.

In the light of what has been done until now, defects play a key role in the behaviour of devices. Indeed, they have given rise to the development of models to analyse their formation, growth and dissolution. The microelectronic industry makes use of them in their process simulators, assuming that the energy of extended defects is reduced monotonously when the defect is getting bigger. For the small clusters they used the magic numbers of N. Cowern *et al.* [15] we have already mentioned. In addition, these simulators consider that a small cluster is transformed to a {113} defect when it reaches a given size, and the same takes place for the transition of {113} defect to a dislocation loop [12]. Therefore, current models fail to reproduce certain situations, such as very superficial implantations in Ge (dislocation loops are formed before other defects) [31].

The prevalence of a certain configuration in the first stages of an annealing can influence the growth of defects, favouring the formation of a given extended defect. If one includes this idea in the simulators more accurate models of defects evolution could be developed. To do so, it is not only necessary to know the formation enthalpies of each configuration, but also the energy barriers of the transitions between the different configurations. Low energy barriers would make easy the change of configuration while high ones would make it difficult. As a result, the evolution of defects would be determined by the initial configuration of the cluster for the second case. In any case, despite the fact that there has been some studies which have determined the energy of small interstitial aggregations in Si and predict their growth as we have described in this section, one can not find an approach which calculates the energy barriers between the configurations of small interstitial clusters.

1.3 Aims

The aims of this work can be outlined as it is showed below:

- Characterization of chainlike, compact and $\{111\}$ configurations of small interstitials clusters (up to 6 atoms), determining their formation enthalpy and topology and the energy barriers of their associated transitions.
- Study of the variation of formation enthalpies and energy barriers of small clusters with different pressures, either positive or negative.
- Implementation of Tersoff interatomic potential through a kART scheme.

1.4 Computational resources

It should be stressed that this work has been made within the acknowledged MMM (*Multiscale Materials Modelling*) [32] research group of the Electricity and Electronics Department of University of Valladolid. Semiconductor defects is one of the research lines MMM group is studying, as well as the area of irradiation of materials, technological processes involving ion beam processing and annealing and the development of new models and computational methods to improve predictive capabilities and applications

of computer simulation to science and engineering. To do so, a wide range of atomistic modelling techniques, including fundamental ab-initio calculations are combined, Tight-Binding and Classical Molecular Dynamics, and Kinetic Monte Carlo simulations.

The computational resources which have made it possible to undertake this study belong to this group and it is owned by the mentioned institution. Two Linux clusters have been used to run the simulations: Beta [33], which consists of 15 nodes of 2 Intel Xeon E5450 quadcore each one with 8 GB RAM, and Gamma [34], which consists of 6 nodes of 2 Intel Xeon X5650 hexacore each one with 192 GB RAM and also 8 nodes of 2 Intel Xeon E5-2660 v2 decacore each one with 256 GB RAM. Furthermore, Beta has a total disk memory of 8 TB whereas Gamma have 10 TB for the data storage. Both of them have a UPS (*Uninterruptible Power Supply*) in case of a power supply failure. Beta cluster technical specifications can be seen in figure 1.8 and Gamma features are showed in figure 1.9. Additionally, figure 1.10 displays a picture of both clusters.

All the described resources have been essential in this work. The simulations which we are going to refer to in the following chapters have required high computational cost due to the fact that we have reproduced the behaviour of relative big cells during long times, nanoscale speaking. In this sense, the considerable operational requirements of kART simulations adds to the previous statement. As an example, figure 1.11 shows the activity for a day, when after several simulations were launched in nodes from 0 to 6 and from 8 to 15. Similarly, figure 1.12 presents the load distribution for a week, so that we can see the work of each node during our simulations.

Legend			
Node Name	1-min load	beta.local 0.13	
cpu: CPU clock (GHz) (num CPUs)		Last heartbeat 19s	
mem: Total Memory (GB)		cpu: 2.40G (8) mem: 15.60G	
compute-0-15.local 9.01	compute-0-11.local 8.02	compute-0-7.local 8.07	compute-0-3.local 8.00
Last heartbeat 10s	Last heartbeat 5s	Last heartbeat 18s	Last heartbeat 18s
cpu: 3.00G (8) mem: 7.81G	cpu: 2.99G (8) mem: 7.81G	cpu: 2.99G (8) mem: 7.81G	cpu: 2.99G (8) mem: 7.81G
compute-0-14.local 8.08	compute-0-10.local 8.07	compute-0-6.local 8.00	compute-0-2.local 8.00
Last heartbeat 27s	Last heartbeat 8s	Last heartbeat 3s	Last heartbeat 4s
cpu: 2.99G (8) mem: 7.81G	cpu: 2.99G (8) mem: 7.81G	cpu: 2.99G (8) mem: 7.81G	cpu: 3.00G (8) mem: 7.81G
compute-0-13.local 8.06	compute-0-9.local 8.04	compute-0-5.local 8.08	compute-0-1.local 8.00
Last heartbeat 31s	Last heartbeat 26s	Last heartbeat 19s	Last heartbeat 23s
cpu: 2.99G (8) mem: 7.81G	cpu: 2.99G (8) mem: 7.81G	cpu: 3.00G (8) mem: 7.81G	cpu: 2.99G (8) mem: 7.81G
compute-0-12.local 8.00	compute-0-8.local 8.03	compute-0-4.local 8.00	compute-0-0.local 8.00
Last heartbeat 16s	Last heartbeat 5s	Last heartbeat 4s	Last heartbeat 18s
cpu: 2.99G (8) mem: 7.81G	cpu: 2.99G (8) mem: 7.81G	cpu: 2.99G (8) mem: 7.81G	cpu: 2.99G (8) mem: 7.81G

Figure 1.8: Detailed technical information of Beta cluster by *Ganglia cluster Toolkit* [33].

Legend			
Node Name	1-min load	gamma.local	
cpu: CPU clock (GHz) (num CPUs)		Last heartbeat 40s	
mem: Total Memory (GB)		cpu: 2.40G (8) mem: 7.75G	
compute-0-13.local	0.01	compute-0-9.local	0.00
Last heartbeat 15s		Last heartbeat 16s	
cpu: 2.20G (20) mem: 252.32G		cpu: 2.20G (20) mem: 252.32G	
compute-0-12.local	0.00	compute-0-8.local	0.00
Last heartbeat 16s		Last heartbeat 19s	
cpu: 2.20G (20) mem: 252.32G		cpu: 2.20G (20) mem: 252.32G	
compute-0-11.local	0.00	compute-0-7.local	0.00
Last heartbeat 14s		Last heartbeat 16s	
cpu: 2.20G (20) mem: 252.32G		cpu: 2.20G (20) mem: 252.32G	
compute-0-10.local	0.05	compute-0-6.local	0.00
Last heartbeat 43s		Last heartbeat 40s	
cpu: 2.20G (20) mem: 252.32G		cpu: 2.20G (20) mem: 252.32G	
compute-0-5.local	0.00	compute-0-4.local	0.00
Last heartbeat 22s		Last heartbeat 43s	
cpu: 2.66G (12) mem: 189.23G		cpu: 2.66G (12) mem: 189.23G	
compute-0-1.local	0.00	compute-0-3.local	0.01
Last heartbeat 36s		Last heartbeat 15s	
cpu: 2.66G (12) mem: 189.23G		cpu: 2.66G (12) mem: 189.23G	
compute-0-0.local	0.00	compute-0-2.local	0.00
Last heartbeat 24s		Last heartbeat 26s	
cpu: 2.66G (12) mem: 189.23G		cpu: 2.66G (12) mem: 189.23G	

Figure 1.9: Detailed technical information of Gamma cluster by *Ganglia cluster Toolkit* [34].



Figure 1.10: Picture of Beta and Gamma clusters and their UPSs taken in the Simulation Laboratory of the Higher Technical School of Telecommunications Engineering.

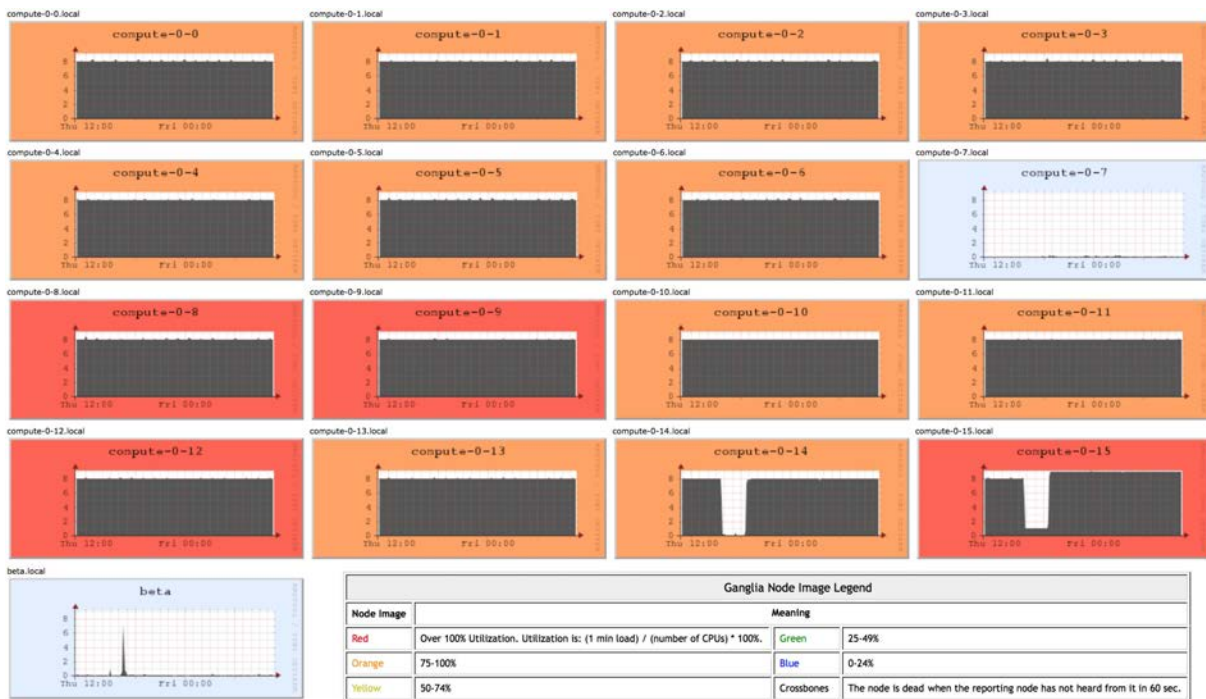


Figure 1.11: Instance of activity report of each Beta node by *Ganglia cluster Toolkit* [33].

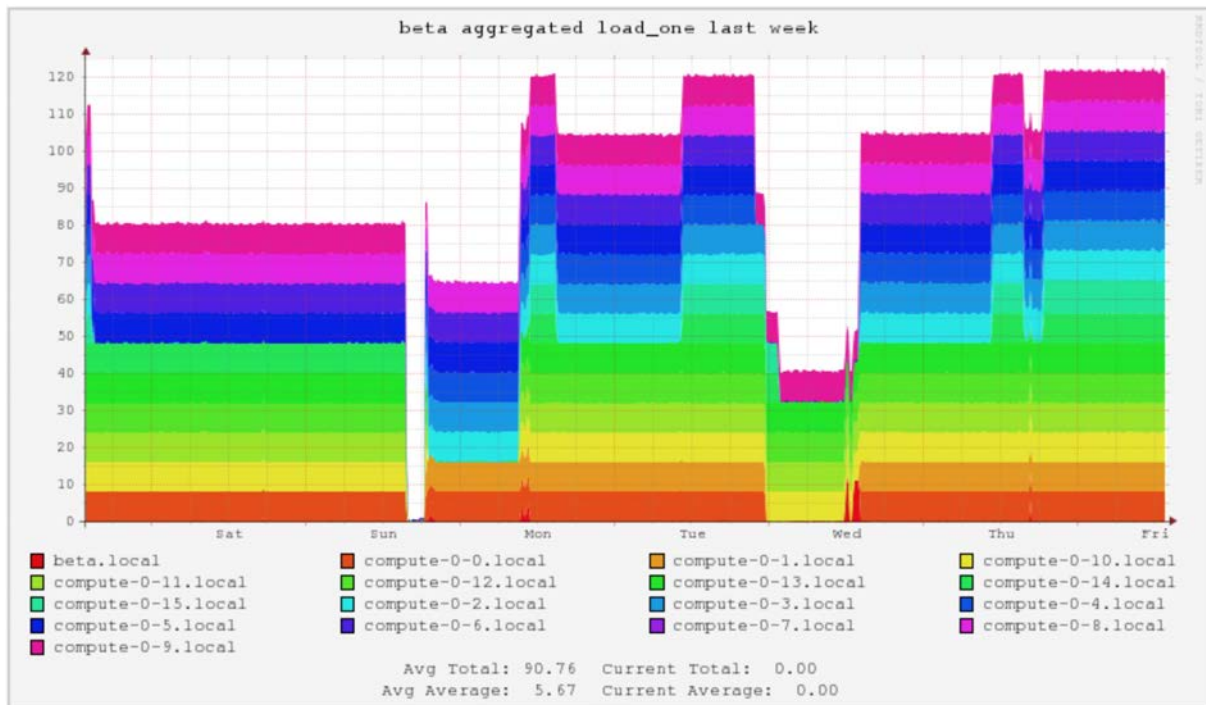


Figure 1.12: Instance of aggregate load in Beta cluster by *Ganglia cluster Toolkit* [33].

1.5 Memory structure

Our contributions have diversified into various topics, which have served to structure this report:

- Firstly we have motivated this Master Dissertation and reviewed the state of the art of small and extended interstitial clusters in crystalline Si. These issues, as well as the formal presentation of the study, are the subject of Chapter 1.
- Since all our work is to be performed using the same simulation technique, a detailed explanation of it is given in Chapter 2, comparing it with other similar methods. The description of software is also included.
- Chapter 3 describes the methodology we have used to run the simulations: the main features of the simulation cells, parameters used to run a kART simulation, its data processing and how we have carried out the energetic characterization.
- Before working with small clusters, Chapter 4 presents a checking of the simulation technique, inserting only a self-interstitial atom.
- Afterwards the obtained results of interstitials clusters whose size ranges from 2 to 6 atoms are presented in Chapter 5.
- Once having characterized each cluster aggregation, Chapter 6 compares the evolution of the analysed properties with the size and how pressure modify them.
- Eventually, Chapter 7 includes the main conclusions which can be extracted from this work and some possible future research lines are raised.

Chapter 2

Simulation method and tools

*Our virtues and our failings are inseparable, like force
and matter. When they separate, man is no more.
~ Nikola Tesla ~*

THIS PART AIMS to describe the theoretical aspects about the simulation technique we have used. We should realize and keep in mind all the approximations which are implemented and verify they match with our work context in order to be sure we have made a correct use of the simulation technique. Hence, we are going to define what is the simulation method we have chosen, highlighting its special features as well as important technical parameters to consider. In addition, the method is going to be contextualized, mentioning related techniques and their connexion with ours. This idea is essential due to the fact that we have run our simulations by embedding other technique in our used method in order to be able to use the desired interatomic potential.

Likewise, the processing programs we have developed and the visualization tools we have used should not be avoid. Both utilities have had a great importance owing to the type of data we have worked with and have played a major role in the conclusions which can be extracted about them. Additionally, the end of the chapter will include other tools which we have frequently used, having made easier our work.

2.1 Introduction

While computer simulation methods were seen only as a powerful but complementary tool in many fields of science some years ago, nowadays they are a well-established research area almost present in every sector. The possibility of avoiding the typical approaches of analytical problems gave rise to a widespread use, whose applications continue expanding. Thus, the computational techniques allow us to study the behaviour of complex systems that could not be solved by analytical techniques [35].

In fact, certain properties of complex systems can only be studied through simulation under some circumstances. The reason of the previous statement lies in the impossibility of measuring specific aspects by analytical techniques despite how sophisticated they may be. Besides this idea, computing a great deal of data is no feasible if we use these traditional methods. In any case, we should consider the simulation techniques in the middle of theory and experiment, filling the gap between them, as it can be seen in figure 2.1. They provide an accurate comparison between model and experiment, letting us determine the validity of the first one. As a result, one can experiment through the computer.

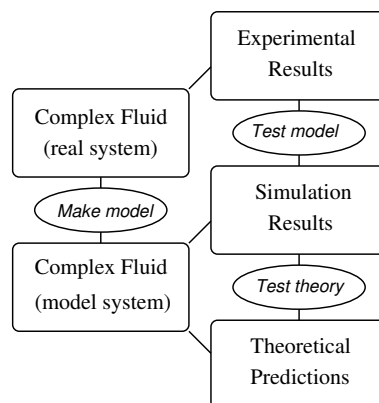


Figure 2.1: Connexion between theory, experiment and simulation [35].

In this sense, we have to compare directly the theoretical results with the experimental ones to check the simulation methods. This affirmation, however, cannot be true due to the fact that there is no detailed information during an experiment, measuring macroscopic properties. Statistical mechanics raise to solve this issue.

2.1.1 Deterministic and stochastic methods

Taking into account that in order to obtain a mean value of a certain property of our system we have to propagate the system through a path in the phase space (positions and momentums), there are two possibilities to implement this propagation [36]:

- **Deterministic methods.** They propagate the system through the phase space by making use of their intrinsic dynamic, calculating the movements equations and integrating them every time step. MD (*Molecular Dynamics*) is the most extended method.
- **Stochastic methods.** They evaluate only the configurational part of the problem, integrating the moments of the particles every time. Thus, moment transitions have to be introduced to pass from one configuration to another. To do so, they are based on a probabilistic evolution. Consequently, an important advantage they have is to be able to simulate systems without intrinsic dynamic. Considering the research efforts in this area, MC (*Monte Carlo*) is the most common technique and we are going to give some details about it in next section.

2.1.2 Classification according to system size and simulation time

A convenient way to address the issue of modelling materials is to separate all the techniques according to the entity whose behaviour is described. Commonly speaking there is a four level approach depending on this minimum element to consider. They can be seen in the table 2.1.

Method	Entity described	Number of units	Length scale	Time scale
Electronic models	electron	10 - 100	0.1nm - 1nm	-
Atomistic models	atom	10^2 - 10^9	0.1nm - 100nm	fs - μ s
Mesoscopic models	nanoparticle, grain	10^6 - unlimited	100nm - mm	ms - s
Continuum approaches	continuum unit cell	unlimited	nm - m	s - ks

Table 2.1: Considered entities according to the size and simulation time for different models [37].

It should be noted that when we decrease or increase the system size the models should be adjusted to their respective systems, reproducing the right behaviour by using

the corresponding physical or chemical expressions. At the same time, the simulation time should not be very long, making it feasible. In this way, a certain time scale is linked to the systems which can be simulated.

The following points summarize the main aspects of each approach [37]:

- **Electronic models.** They follow quantum mechanics laws, which enable them to characterize the system by knowing the wave function and their Hamiltonian operator. We can find different variants inside this model, highlighting textitab-initio, MB (*Many Body*), HF (*Hartree-Fock*), and TB (*Tight Binding*). In this point, we have to emphasize that *ab-initio* models are a very widely used approach for isolated systems at 0 K, being able to develop models to upper scales by using DFT (*Density Functional Theory*) [38].
- **Atomistic models.** This is the point in which our work is located. They are based on statistical mechanics, statistical thermodynamics and probability theory. The interatomic forces between the atoms are determined by the interatomic potential, which is parametrized through analytical expressions obtained from a lower abstraction level such as *ab-initio* techniques. They avoid the interaction of electrons inside the network but they are able to reproduce systems of ~ 100 nm size ($\sim 10^9$ atoms) during $\sim \mu s$ time. Hence, being strict we cannot argument that they can simulate macroscopic systems, but instead they are likely to reveal the microscopic origin of them getting macroscopic properties similar to the experimental ones. There is a long list of techniques to emphasize in this category: MD and MC (we have already mentioned and we are going to give some details below), atomistic spin models and statistical atomistic models.
 - **MD** is a simulation technique which calculates the atom displacements in a dynamic system. It implies that the system is influenced by certain physical forces during a particular period. To do so, it solves the Newton's equations of movevement for a multimass system with interactions between themselves. Using a numerical integration scheme, the dynamics of each atom in the system under the effect of the interatomic forces are calculated for each time step. The length of the chosen time step is a compromise between a value large enough to allow a long simulation time for the purpose of the study and a value small enough to capture all the required fundamental frequencies of motion in the system and conserve the energy between integration steps. This is

typically in the range of 1-10 fs, but depends on the range of motions allowed in the system. Multiple time scale calculations, where rapidly changing forces are updated more frequently than slowly changing forces, are also possible in order to reduce the required computational time. In addition, the control of the temperature and the pressure is done by either stochastic or numerical methods.

The main advantage of this technique is that we can obtain the trajectory of the system during the simulation, from which dynamical and structural quantities can be obtained, as well as macroscopic properties (e.g. temperature, pressure, work, heat, free energy, and entropy) as other methods usually do. Furthermore, within this technique we can find several variations, such as CMD (*Classical Molecular Dynamics*), MD *ab-initio* or QM/MM methods (*Quantum Mechanics/Molecular Mechanics*).

In any case, we have to bear in mind that its aim is to determine the applied force to each atom of the system by keeping constant some physical magnitudes: the number of atoms, the volume and the energy, which is called NVE (*Microcanonical ensemble*), the number of particles, the temperature and the volume, which is called μ VT (*Gran canonical ensemble*), the number of atoms, the pressure and the temperature, which is called NPT (*Isothermal-isobaric ensemble*) or other variations.

The development of this technique dates back to 1960, when J. Alder y Thomas E. Wainwright took advantage of the technological advances of their time and simulated perfectly elastic collisions between hard spheres. Since then MD have become a very remarkable simulation technique in several areas, including physics and chemistry, and they have given rise to interesting applications in many other fields. In fact, it is being widely used to study protein structures, nuclei acids, biomolecules and complex fluids. Looking for the causes of this diverse range of application it can be attributed to the fact that depending on the interatomic potential which is being used it can be adapted to almost every kind of system [36, 39].

- **MC** is a simulation method which does not follow the Newton's equations at all, but instead it proceeds in a stochastic way which depends on a sequence of random numbers which are generated during the simulation. Thus, the purpose is to use these random numbers to determine the events of a system and its particles, each of them with an associated probability.

Taking into account that with a different sequence of random numbers the simulation will not give identical results, but it will yield values which agree with those obtained from the first sequence to within some statistical error, a very large number of problems can be addressed with this technique.

It should be noted that although we may not be moving within the phase space along the same path we may be attempting to sample a region of the phase space in order to estimate certain properties of the model. In order to find the cause of the last statement, we should consider that the task of equilibrium statistical mechanics is to calculate thermal averages of interacting many-particle systems, taking proper account of statistical fluctuations and their effects in such systems. Moreover, since the accuracy of a Monte Carlo estimate depends upon the thoroughness with which phase space is probed, a substantial improvement may be obtained by simply running the calculation a little longer to increase the number of samples [40].

Among this category we have kMC (*kinetic Monte Carlo*) technique, which keep the same statistical principles but using a variable time step, which enable us to reach longer time scales ($\sim h$). It is quite useful to reproduce the behaviour of materials in technological processes such as interactions between dopant atoms during annealing, being that the main reason for what we have used this method in this work. In addition, we have to mention that BCA (*Binary Collision Approximation*) method is also very common to study ion implantation.

Looking back about its origin, the modern version of the Markov Chain Monte Carlo method was invented in the late 1940s by Stanislaw Ulam, while he was working on nuclear weapons projects at the Los Alamos National Laboratory in collaboration with John Von Neumann. It is thought that Ulam while playing solitaire tried to calculate the likelihood of winning based on the initial layout of the cards and he named MC after a casino in Monaco.

- **Mesoscopic models.** They focus on describing the behaviour of entities like nanoparticles, grains, agglomerates, large biomolecules or macropins. In this sense, molecular models belong to this class. The method consists of grouping more than one atom together in an entity whose behaviour is described by an equation in order to allow the simulation of larger systems and for longer times, although it implies loosing some atomistic details. Additionally, there are two main techniques inside

these models: Statistical Mesoscopic Models and Micromagnets Models, although their implementation is not as widespread as MD and MC.

- **Continuum models.** They can simulate bigger structures reaching macroscopic levels without accuracy. All the previous abstraction levels take into account that the materials are composed by molecules that separated by empty gaps. However, these models consider the matter as a continuum. In other words, they assume the matter distributes filling all the gaps. Therefore, they lose touch with reality but can simulate certain physical phenomena.

Figure 2.2 shows the contextualization of some simulation methods we have described in comparison with the corresponding experimental techniques of the same size scale. Anyway it is necessary to stress that almost every physical process cannot be reproduced by a single model because it would be tough to follow the real behaviour accurately. What researchers in this area usually do is a multiscale approach due to the intrinsic nature of the physical and chemical phenomena. Therefore, these methods are characterized by combining themselves to simulate a system properly, being this idea the fundamental key of the MMM [32] research group.

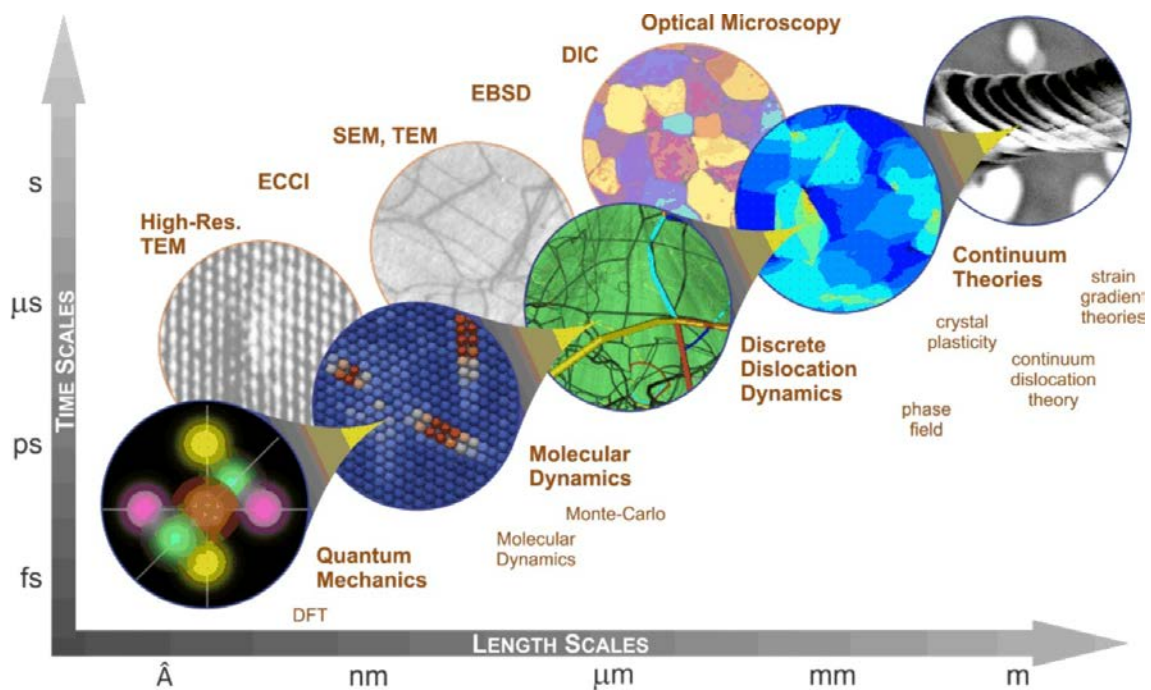


Figure 2.2: Simulation techniques and experimental methods depending on size and time scales [41].

2.2 Kinetic Activation-Relaxation Technique

After giving a general view to the simulation methods we can find in the literature, we are going to describe in detail kART (*Kinetic Activation-Relaxation Technique*), which is the key in this work. Firstly it should be clarified why we have chosen this simulation method and its role in sampling the energy in complex materials, indicating a brief summary of the theoretical background behind it. Lately we will focus on describing our technique.

2.2.1 Theoretical basis

The calculation of transition rates for diffusion events has been a common problem in condensed matter physics. During this kind of events the configuration of atoms is changed in some way during the transition, obtaining the interaction between the atoms from an approximate solution of the Schrödinger equation describing the electrons or from an otherwise determined potential energy function. It should be noted that although a classical mechanics treatment of the motion is usually sufficient, the transitions of interest are typically many orders of magnitude slower than the vibrations of the atoms [42].

Hence, a direct simulation of the classical dynamics is not useful. If we follow the traditional approach for this context, it is likely that we cannot access to rare events whose energy barrier is about 1 eV and occur many times per second. This kind of phenomena would involve the track of too many vibrational periods ($\sim 10^{10}$) and a direct classical dynamics simulation, which necessarily has to faithfully track all this vibrational motion, would present an infeasible computational cost. In order to reach longer time scales, as it has been already introduced in kMC methods, purely statistical methods based on the TST (*Transition State Theory*) are used.

Giving an insight into this theory, TST relies on the Born-Oppenheimer approximation and two basic assumptions. The first one points that the rate should be slow enough that a Boltzmann distribution is established and maintained in the reactant state. The second one establishes that a dividing surface of dimensionality $D-1$, where D is the number degrees of freedom in the system, can be identified such that a reacting trajectory going from the initial state to the final state only crosses the dividing surface once. The dividing

surface must, therefore, represent a bottleneck for the transition. The TST expression for the rate constant r can be written as [42]:

$$r = \frac{\langle |v| \rangle}{2} \frac{Q^\ddagger}{Q_R} \quad (2.1)$$

Where $\langle |v| \rangle$ is the average speed, Q^\ddagger is the configurational integral for the transition and Q_R is the configurational integral for the initial state.

Consequently, the TST rate estimate gives an approximation for the rate of escape from the initial state without considering the final state. However, it can be determined by short time simulations of the dynamics starting from the dividing surface, which also give us an estimate of the correction to TST which is needed to improve the approximation. This is what is widely called as dynamical corrections.

In any event, the previous expression is a general equation which is not usually apply for crystals whose atoms are tightly packed and the relevant temperatures are low compared with the melting temperature. To achieve this, the harmonic approximation to TST takes it into account and can typically be used in studies of diffusion and reactions in crystals. In addition, it estimates the rates more accurate. Following this approach, the search for the optimal transition state then becomes a search for the lowest energetic saddle points at the edge of the potential energy basin corresponding to the initial state. Thus, the rate constant for transition through the region around each saddle point can be obtained from the energy and frequency of normal modes at the saddle point and the initial state [42]:

$$r^{hTST} = \frac{\prod_i^{3N} -\nu_i^{init}}{\prod_i^{3N-1} \nu_i^\ddagger} e^{-(E^\ddagger - E^{init})/k_B T} \quad (2.2)$$

Being E^\ddagger is the energy of the saddle point, E^{init} is the local potential energy minimum regarding to the initial state, and ν_i are the corresponding normal mode frequencies.

As it happened before, the mechanism of the transition is reflected in the saddle point and when it is found one can follow the gradient of the energy downhill, both forward and backward, and map out the minimum energy path. Nevertheless, it should be stressed that finding a relevant saddle point is quite tough, being one of the most challenging tasks in theoretical studies of transitions in condensed matter. In addition, when a search is launched it is not sufficient to find a saddle point, but rather one needs to find the highest

saddle point along the minimum energy path, in order to get an accurate estimate of the rate from harmonic approximation to TST. Figure 2.3 gives an excellent representation of this idea.

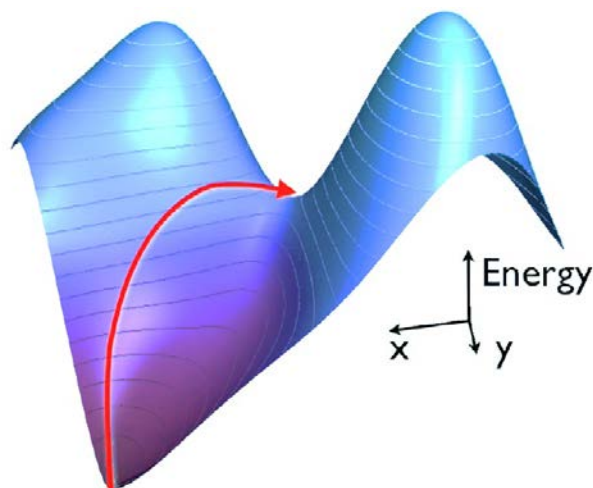


Figure 2.3: Sample of an energy landscape showing a saddle point and adjacent energy minima linked by the minimum energy path [43].

Many methods have been proposed for finding the minimum energy paths and saddle points. They basically involve some kind of maximization of one degree of freedom and minimization in other degrees of freedom. The key is to find a good and inexpensive estimate with the more freedom as possible. Here we are going to describe briefly Lanczós [44] and DIIS (*Direct Inversion in the Iterative Subspace*) [45], which are the methods used by kART in order to converge to a first-order saddle point, which is a position on the potential energy surface corresponding to a minimum in all directions but one.

2.2.1.1 Lanczós

kART makes use of Lanczós algorithm to determine the lowest eigenvalue of the Hessian $\partial^2 E / \partial r_i \partial r_j$ and its corresponding eigenvector u_i necessary in its first two operation stages: leaving the harmonic well and activation and convergence to a first-order saddle point (see section 2.2.3.1 for more information). In the first phase, when this eigenvalue falls below a certain negative threshold, the system is considered to have left the harmonic well and we move to the activation regime. In the second one, Lanczós is used to obtain a direction of negative curvature.

Because of the recursive application of the method, it was shown that only 13–16 force evaluations provide an accurate value of the lowest eigenvalue, without having to compute and diagonalize the $3N$ -dimensional Hessian at every step. Actually it is not a direct implementation of Lanczós but a modification which reduce systematically the number of force evaluations during the activation stage while converging with a much greater precision onto the saddle point [46].

Expressing the fundamentals of the method mathematically, it is basically an adaptation of power methods to find the most useful eigenvalues and eigenvectors of n th order linear system with a limited number of operations, which should be lower than the order. Its aim is to calculate the tridiagonal and symmetric matrix $T_{mm} = V_m^* A V_m$. Once calculated, one can solve its eigenvalues $\lambda_i^{(m)}$ and their corresponding eigenvectors $u_i^{(m)}$ by a complementary algorithm such as the widely known QR ($A = QR$, where Q is an orthogonal matrix and R is a triangular matrix).

To do so keeping the stability the following algorithm is used:

$v_1 \leftarrow$ random vector with norm 1.

$v_0 \leftarrow 0$

$\beta_1 \leftarrow 0$

for $j = 1, 2, \dots, m - 1$

$w'_j \leftarrow A v_j$

$\alpha_j \leftarrow w'_j \cdot v_j$

$w_j \leftarrow w'_j - \alpha_j v_j - \beta_j v_{j-1}$

$\beta_{j+1} \leftarrow \|w_j\|$

$v_{j+1} \leftarrow w_j / \beta_{j+1}$

endfor

$w_m \leftarrow A v_m$

$\alpha_m \leftarrow w_m \cdot v_m$

return

Where the diagonal elements are denoted by $\alpha_j = t_{jj}$, and the off-diagonal elements are denoted by $\beta_j = t_{j-1,j}$. It should be noted that $t_{j-1,j} = t_{j,j-1}$, due to its symmetry. As a result, we get the α_j and β_j from each iteration, constructing the tridiagonal matrix:

$$T_{mm} = \begin{pmatrix} \alpha_1 & \beta_2 & & & & & & 0 \\ \beta_2 & \alpha_2 & \beta_3 & & & & & \\ & \beta_3 & \alpha_3 & \ddots & & & & \\ & & \ddots & \ddots & \beta_{m-1} & & & \\ & & & \beta_{m-1} & \alpha_{m-1} & \beta_m & & \\ 0 & & & & \beta_m & \alpha_m & & \end{pmatrix} \quad (2.3)$$

2.2.1.2 DIIS

An important acceleration of the technique was due to the combination of DIIS method with the previous one in the activation phase. This enable kART to improve the convergence and can significantly decrease the computational costs of this stage. It should be stressed that DIIS is much less demanding than Lanczós in terms of force evaluation per step: it only demands one, a crucial advantage when calculating the energy and forces is costly [46].

If we define the method in a mathematical approach, at a given iteration, the algorithm combines previous iterations e_i to construct a linear combination of approximate error vectors e_{m+1} :

$$\mathbf{e}_{m+1} = \sum_{i=1}^m c_i \mathbf{e}_i. \quad (2.4)$$

Their coefficients c_i are determined so to best approximate by equating zero to the derivatives of L with respect to the coefficients:

$$L = \|\mathbf{e}_{m+1}\|^2 - \lambda \left(\sum_i c_i - 1 \right) = \sum_{ij} c_j B_{ji} c_i - \lambda \left(\sum_i c_i - 1 \right) \quad (2.5)$$

Where $B_{ij} = \langle \mathbf{e}_j, \mathbf{e}_i \rangle$. Then, these coefficients are used to extrapolate the function variable for the next iteration.

2.2.2 Comparison with similar techniques

Over the last decades there has been an incremental effort to get an effective method for sampling the energy landscape of complex materials, especially in the field of semiconductors. In this area, the aim is to understand the long-time kinetics of defects diffusion inside a lattice containing from many tens to many thousands atoms.

Thus, before kART was launched the following methods were available [47–49].

2.2.2.1 MD extensions

Some of the proposed methods are variations of MD. For instance, the Voter’s hyperdynamics [50], which provides an accelerated scheme that incorporates directly thermal effects, the Laio and Parrinello’s metadynamics [51], which focuses on computing free-energy barriers for specific mechanisms, and the basin-hopping method by Wales and Doye [52] and Goedecker [53].

Although the traditional approach would point that MD simulations is the right way to address this kind of problem, we have to keep in mind that the computational cost is extremely high and the simulation times it can reach are too short for our interest. We should not forget that many problems in condensed matter and materials science involve stochastic processes associated with the diffusion of atoms over barriers that are high with respect to temperature. Therefore, we would not be able to overpass them under normal conditions, which is what can be done with MD [47].

2.2.2.2 Based on kMC algorithm

They treat the thermal contributions in a quasi-harmonic fashion, in accordance with the transition state theory so that they are really aimed to search for transition states and the computation of energy barriers. Consequently, applying this approach we can get access to events which have small rates, in other words, we can avoid the MD problem by considering these processes independent and neglecting the thermal motion of the atoms. This is the key of kMC based methods.

In kMC simulations, the clock is then advanced using Poisson’s law, a move is selected with the appropriate rate, and a dynamical trajectory is constructed. While time steps in kMC are dominated by the lowest energy barriers, it is possible, under the right

conditions, to simulate experimental time scales. Anyway, the time scale is determined by the fastest activated processes, reaching time scales of \sim ms, much longer than those accessible with MD. That is the point why kMC has been extensively and successfully used over the past 20 years.

Nevertheless, it presents a few drawbacks to bear in mind [49]:

- The systems investigated must be discretized and mapped onto a fixed lattice in order to define the various diffusion mechanisms that need to be considered at a given moment.
- The simulations simply consist in operating a diffusion event picked at random, updating the list of possible moves in the new configuration, which is not suitable for certain systems whose complexity is high. For instance, it fails when the systems undergo significant lattice deformations or when long-range elastic effects are important.

Lifting these limitations has been an active field of research for many years. Generally speaking we can classify all the solutions in two categories: ones which implement continuum approximations for the long-range strain deformations and others which are based on an on-the-fly evaluation of the energy barriers. The way the first category works to determine the barriers is by using extrapolation schemes from these long-range contributions. However, there is a need for setting up a catalogue of all possible activation mechanisms [48].

In any event, regardless of whether we chose one or another, we should be sure about where the displacement can take place. Whereas in MD it is assumed an off-lattice approximation, it can be distinguished two approaches with kMC methods:

- **On-lattice.** They limited the displacements of the atoms to the lattice we are working with. In other words, a fixed grid is given and atoms can only move from one site to another. Such an on-lattice approach is no the best choice for treating defects.
- **Off-lattice.** They operate can outside the lattice by defining a set of mobile sites in space on which events occur. Each site's neighbourhood of nearby sites is defined by a cut-off distance and forces are calculated through a potential. The problem

with on-lattice kMC methods is that rules are set up for defining movements and their energy barriers, being unlimited in complexity in principle.

Thus, the proposed solution which implement continuum approximations we have already mentioned above tend to be on-lattice. Therefore, it is not the right option for our work, being preferable an on-the-fly and off-lattice technique approach. In this sense we have two possible approaches which fulfil both requirements [47]:

- **Closed-ended methods.** They require the knowledge of both the initial and final states. Hence we do not let the system evolve freely and it adds to the fact that there are some complex systems whose list of states are almost empty beforehand. This is the case, for example, for the NEBM (*Nudged-Elastic Band Method*) [54] and the GSM (*Growing-String Method*) [55].
- **Open-ended methods.** To solve the previous problem, there has been proposed some methods which gives freedom to the evolution of the systems, achieving transitions without indicating final energy state a priori. They tend to follow the prescriptions proposed by Cerjan and Miller [56] and Simons *et al.* [57], highlighting the ART (*Activation-Relaxation Technique*) [58–60], also known as ART nouveau, the eigenvector-following method [61], an hybrid version [62] and the dimer method [63].

When the final state of the transition is known, both the initial and final coordinates of the atoms can be used as boundary conditions in the search. When only the initial state is known, the problem is more challenging and the search for the saddle point represents also a search for the optimal transition mechanism [42].

In any case, whereas all the methods described above have the advantage of being more flexible due to the off-lattice approach, the vast of them are currently inefficient as they do not take advantage of the knowledge of previously encountered events, and are therefore only useful for small systems with very few barriers. In other words, they are not self-learning methods. It should be stressed that once a method is available for finding saddle points and adjacent minima, there are several ways to treat this information. Additionally, in the end researchers have a strong interest in understanding the underlying kinetics controlling the complex systems. To this purpose, it is no longer sufficient to collect these saddle points: they must be ordered and connected in some fashion to reconstruct at least a reduced representation of the energy landscape.

In this way, there are many options available to address the problem, highlighting kART [49] and Wales [64] approach. The second method is based on constructing a connectivity matrix that links together minima through saddle points after starting from a large catalogue of events, and then applying a master equation to solve the kinetics. Thus, it benefits from the discrete path sampling in terms of providing a complete solution to the system's kinetics. However, the matrix increases rapidly with the system's complexity, making it difficult to address the kinetics of large and complex problems.

Hence, a more straightforward manner to generate kinetics with these methods is to apply an on-the-fly kMC procedure. The main point here is that from a given local minimum, a finite number of open-ended event searches are launched and the resulting barriers are used to estimate a probable time scale over which an event will take place. In addition, to be feasible an extensive sampling of activated events is performed before each kMC step. In this sense, some authors proposed several open-ended methods but all of them are characterized by not recycling unaffected events. There is a widespread effort for years in order to implement cataloguing for these events, which most of the time involve off-lattice positions.

It should be noted that the need for such a method is very strong, as is confirmed by the multiplications of other algorithms that also address various limitations of standard kMC published recently. Besides kART, there are other research groups which have recently implemented a simple on-the-fly kMC approach with sampling of events after every step but no making use of previously found events, such as the already mentioned Henkelman and Jónsson approach [63] but adding it kinetics, the self-learning kMC [65], the self-evolving atomistic kMC [66], the local-environment kMC [67] and the molecular dynamics saddle-search adaptive kMC [68]. All of them are based on the scheme showed in figure 2.4.

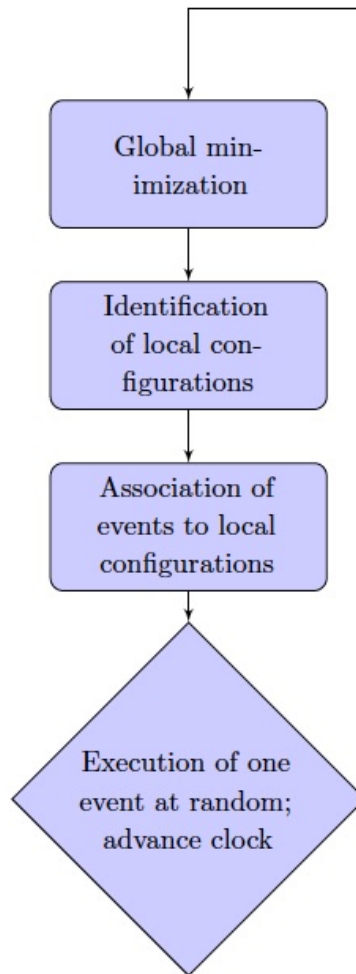


Figure 2.4: Simplified workflow for on-the-fly kMC algorithms [69].

In any case, the challenge, for all these methods, is threefold [47]:

1. Ensuring there are not relevant event missing in the catalogue.
2. Devising a classification which can correctly discriminate events even in complex environments, for example in materials with defects.
3. On-the-fly updating of the energy barrier in order to fully consider elastic deformations.

That is the point where kART has come onto the scene, developed by the same research group as ART. It is a powerful both on-the-fly and off-lattice kMC method

based on ART algorithm for generating events and calculating barriers, achieving speed-ups as large as 4000 over standard MD for complex systems, while retaining a complete description of the relevant physics, including long-range elastic interactions due to the discretization of events even for disordered systems. The gain in efficiency is achieved through a topological classification of atomic environments, which allows configurations and events to be recognized and stored efficiently, and used again as the simulation proceeds [49].

Task	kART	Self-evolving atomistic kMC
Identification of local configurations	Local configurations are well described by topology of connectivity graph	Local configs are well described by distances of atoms from the perfect lattice
Association of events to local configurations	Events can be catalogued using topology	Only a subset of atoms are necessary to handle elastic effects on saddle-point
	Events can be mapped using graph canonical representation	Sometimes neglect checks for duplicity of events
	Most important events are sampled when encountering a new topology	Neglect detailed balance. (Formally, a large number of searches will still lead to micro-reversibility.)
	Expansive, but incomplete, catalogue is rarely resampled	Catalogue is resampled with uncorrelated random searches at every kMC step to compensate limited number of searches
Event execution	Harmonic version of TST Constant prefactor	Harmonic version of TST Constant prefactor
	Poisson process Uncorrelated processes	Poisson process Uncorrelated processes
	Mean-passage time correctly describes kinetics of flickers	

Table 2.2: Technical comparison between k-ART and Self-evolving atomistic kMC [69].

Apart from kART, if we have to emphasize one alternative methods described above, it would be the self-evolving atomistic kMC. It has a lot of characteristics shared with kART, as it can be seen in table 2.2, but it samples a relatively small number of such transition states each step, identifying local configurations and also associating events to these configurations in a different way. In any case, the way this method works does not match our interest. It simply identifies defects by comparing the atomic positions of atoms to those of a perfect lattice or can be based on energy or stress criteria. The topology-based approach implemented by kART, however, is more flexible and demands less *a priori* knowledge of the system. At the same time, by precisely identifying defects, kART can make saddle-searches more effective by centring them on these positions, increasing the success rate of finding interesting events [69].

On the other hand, one can consider inserting kART in another method, such as the well-known BEP (*Bell Evans Polanyi*) principle. However, it has been recently shown that pure BEP simulations become trapped rapidly in relatively high energy configurations, while KMC runs manage to find ever lower energy states. In addition, if these issue is corrected then the kinetics of the system is lost [70].

2.2.3 Review of the technique

As we have already introduced in the previous section, kART is an on-the-fly, off-lattice and open-ended method which combines ART with a self-learning topological classification of atomic environments behind a kMC algorithm. In this section we are going to explain ART in detail and describe all the essential steps that kART follow during a simulation, giving extra information to the most influential parts.

2.2.3.1 ART nouveau

In this section we are going to explain the first-order saddle point search carried out every step in kART. The ART nouveau method is an algorithm which has been developed over the last 15 years. In short, it incorporates a search in line with the eigenvector-following and the dimer method, through the use of the Lanczós algorithm (see section 2.2.1.1 for more information about how it proceeds).

The method is structured in three steps which form an event [47]:

1. **Leaving the harmonic well.** Taking as a point of origin a local energy minimum, the aim is to identify a direction of negative curvature on the energy landscape because it would imply the presence of a nearby first order saddle point. In this point it is recommended to keep in mind that, mathematically speaking, a first order saddle point is characterized by conforming $\partial E/\partial r = 0$ and the Hessian matrix with one negative eigenvalues.

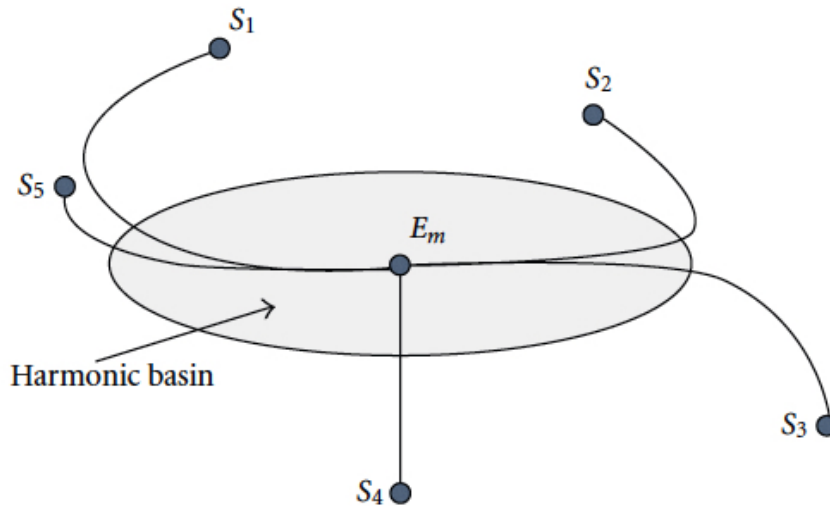


Figure 2.5: Sketch of minimum energy pathways from various first order saddle points to the local energy minimum in ART [47].

The critical issue here is to select the right direction for leaving the harmonic well, which is defined as a region of the energy landscape surrounding a local minimum with only positive curvature, as figure 2.5 shows. In this sense, it cannot make use of the Hessian matrix given by Lanczós, but instead it uses random directions of deformation. Nevertheless, as following this approach tend to favour certain events, ART let us be focused on particular subsets of a system, which has a strong interest when working with defects.

This procedure can be divided into 4 steps:

- (a) Selection a random atom from the whole network or a specific region.
- (b) Identification of this atom and its neighbours by using a cut-off distance and displacement of them along a random direction.
- (c) Minimisation in the perpendicular hyperplane in order to avoid collisions and allow the full system to react the deformation carried out in the previous step.

It should be slight to not fall in the well again. This plane can be seen in figure 2.6 for the initial position.

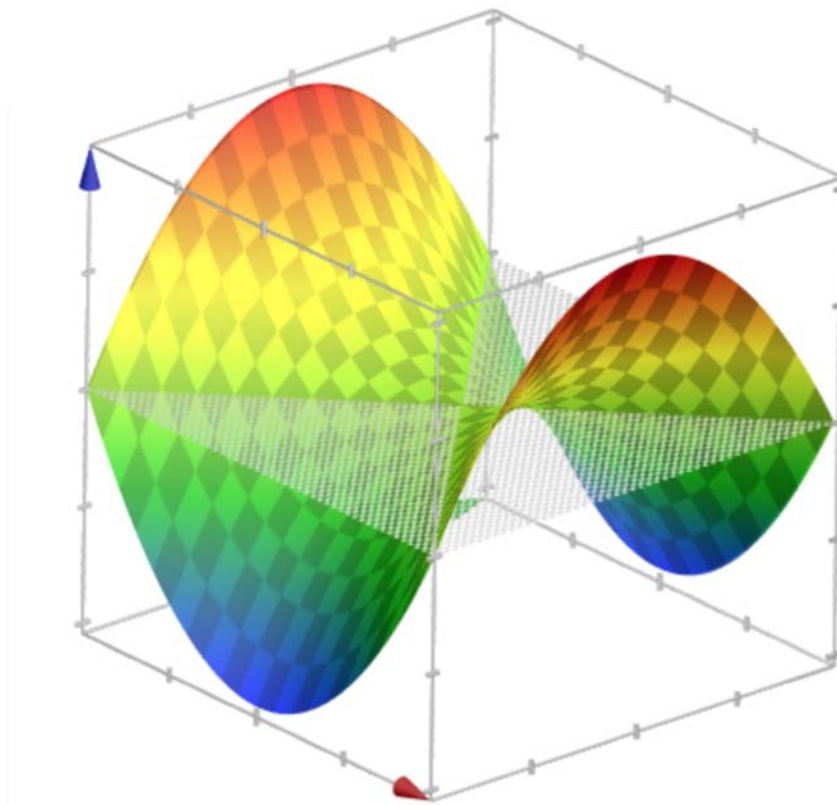


Figure 2.6: Perpendicular hyperplane to the initial energy minimum designed with Matlab.

- (d) Launching of Lanczós algorithm to determine the lowest eigenvalue of the Hessian matrix. At the time this eigenvalue falls below a certain negative threshold, the system is considered to have left the harmonic well and we move to the activation regime.

2. Activation and convergence to a first-order saddle point. The task of this step can be seen as straightforward due to the difficulties of the previous one. At first sight, what it only needs to do is bringing the system to a first-order saddle point by pushing it along the selected direction while the energy is minimised in the perpendicular hyperplane to ensure we are on the right way.

Nevertheless, if we translate this idea into computational methods it can be expensive due to the requirement of partial knowing of the Hessian matrix. Here Lanczós

plays an essential role again because has the advantage of building on the previous results to identify the next lowest eigendirection, decreasing considerably the computational cost. It is affirmed that a 15x15 matrix would be enough to ensure a stable activation, irrespective of the system size, which would only need 16 force evaluations. In any case, depending on the system this situation can be altered so that the number of evaluations decrease.

Although the problem seems to be solved, we have only addressed it when the activation takes place, i.e., when pushing the system next to the initial minimum. The dual approach of activation with Lanczós and minimization in the perpendicular hyperplane is not optimal near the saddle point, so in order to converge to a first order saddle point, it is preferable to use other algorithm instead the hyperplane minimisation.

In this sense, the authors of ART have chosen DIIS, an integrated algorithm that can converge efficiently on inflection points (see [2.2.1.2](#) for to know more details). This algorithm enable ART to accelerate the convergence phase and it also increases the accuracy.

In summary, this procedure can be divided into 4 steps:

- (a) Determination of the direction of negative curvature through Lanczós algorithm.
- (b) Push the system along this direction keeping a displacement which is decreasing with the number of interactions.
- (c) Relaxation of the system in the perpendicular hyperplane, decreasing the number of iterations when we are closed to the energy minimum in order to not fall again into the harmonic well.
- (d) Depending on if DIIS is or not used, we have two possibilities:
 - In case DIIS is avoid, the first three steps are repeated until the system reach to a first order saddle point, which will occurs when total force falls below a predefined threshold. Alternatively, if the lowest eigenvalue becomes positive, it would imply that the system has found its way back into the harmonic well.
 - As long as DIIS is applied, Lanczós algorithm will be launched until the negative eigenvalue has reached a minimum and has gone up for 4 sequential iterations. Once completed this task, DIIS will be in charge of reaching the convergence criterion. Finally, a Lanczós calculation is then performed to ensure we are actual in the aimed saddle point.

- 3. Relaxation into a new minimum.** At this point, the purpose is to push again the system to in order to converge into a new energy minimum. The nudge should be slighter than the previous ones, being typically pushed over a distance of 0.15 times the distance between the saddle and the initial minimum along the eigenvector. To conclude, the system is brought into the new minimum by using any controlled algorithm which ensures the convergence to the closest minimum such as FIRE (*Fast Inertial Relaxation Engine*) [71]. In this way, the algorithm selection does not present the importance as the ones implemented before.

In light of the the three steps which conform an event we have described above, we already know the initial minimum, the transition state and the final minimum, so that it is possible to characterize the local energy landscape. In any case, with this approach we get energy points of the landscape without connexion between them. Thus, by inserting ART within a Metropolis scheme or to generate a kinetic trajectory using a kMC approach all points generated would be fully connected. This idea will be discussed in the following section.

2.2.3.2 Operating mode of kART

If one want to evaluate the dynamics of a system ART is not enough. In this sense, the bias for selecting a specific barrier over the others should be known besides finding saddle points. Thus, a very efficient solution would be sampling all possible events with ART to serve in a kMC scheme.

Summarizing the technique in two lines, kART uses ART nouveau for generating new events and update energy barriers, coupled with a topological classification for local environments, which allows the construction of a discrete event catalogue. Anyway, the kART procedure is described further below, classifying its operation method in three steps which are launched in a loop [47]:

- 1. Topological characterization and generation of events.** Starting from a local energy-minimum, the first task is to analyse the topology associated with each atom. This topological description of local environments has to be accurate due to its later use for mapping events uniquely, establishing the basis to describe environments of any chemical and geometrical complexity.

To do so, a central atom is selected and all its neighbours within a given cut-off radius, connecting them. Then it follows a predefined procedure which forms a graph which is analysed by a software which returns a unique tag associated with the topology and an ordered list of cluster atoms. Regarding to this issue, the used program is NAUTY [72], which provides the permutation key needed to reconstruct a specific geometry from the generic topology and a set of reference positions. As it can be seen in figure 2.7, a truncated graph (b) is extracted from the complete lattice (a) and this graph is analysed through NAUTY (c), which returns a unique key and the associated topology (d).

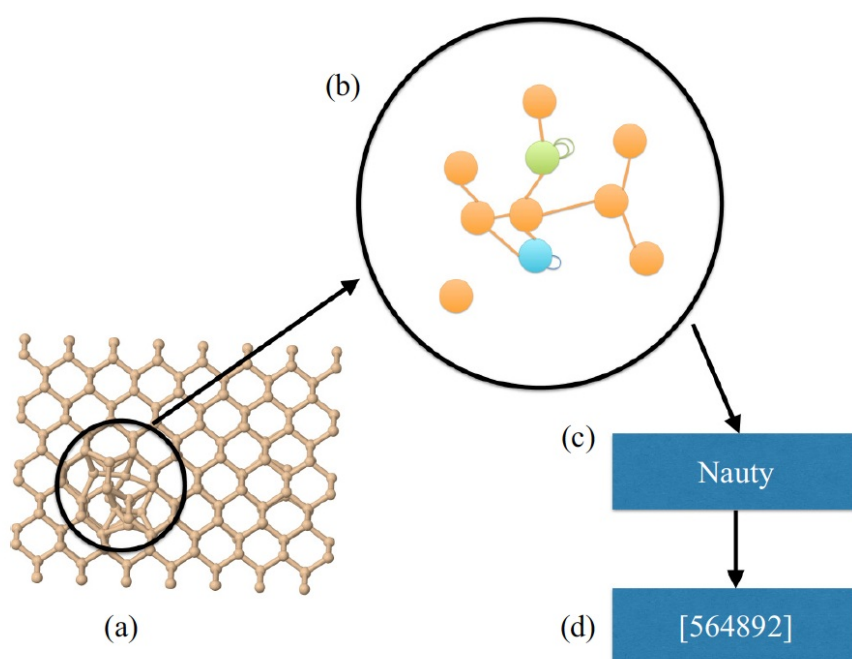


Figure 2.7: Example of topological analysis in kART using NAUTY [73].

The size of the truncated region are selected to ensure that the configuration is uniquely defined through this network, that is, the connectivity graph must lead to a unique structure once relaxed with a given interatomic potential. In the case of crystalline Si, for example, it is defined the local environment around an atom by a sphere of radius 5.0 \AA , which includes about 40 atoms considering 2.8 \AA as a cut-off distance to link two atoms.

The critical point here is to ensure that the topological description and the real-space geometry of this local region follow a unique correspondence. To do so, one can suppose that the graph has to be embedded into a well-defined surrounding

geometry. This idea is fulfilled the majority of conformations, but it is not true when the following cases take place so that they have to be addressed:

- One topology is associated with the same geometry: it is likely to happen when working with very flat minima or saddle points, overcounting events then. It is solved by comparing other known parameters such as the barrier height, the absolute total displacement and the direction of motion of the main atom both between initial and final state and between initial state and the saddle point.
- A single topology corresponds to more than one geometry: quite common for highly symmetric systems when two topologically identical events which differ only in the direction of motion. In order to solve this issue, several extra searches of ART nouveau are launched to reconstruct the geometry, ensuring that motion takes place in different directions.
- A topology is associated with fundamentally different geometries: although is strange to occurs, it can be an internal failure which is automatically solved. Since it leads to incorrect saddle-point reconstruction that cannot be converge, the algorithm modifies the first neighbour cut-off by itself until the multiple geometries are separated into unique topologies.

Once the topological characterization has been described, now we are going to focus on the event generation. In this sense, as long as a new topology is identified, ART nouveau is launched from the central atom in order to determine the associated saddle point. This procedure is done a certain number of times depending on the system. An estimate number between 25 and 50 searches for every \log_{10} times we see a topology is given by the manual, saving all the associated states (initial and final energy minimum and saddle point) to each one. Hence, it should be noted that event searches are not limited to new topologies. All these events are called generic events and stored and accumulated in the catalogue. kART make use of them to reconstruct specific events associated with a given configuration, being able to decrease the computational cost in further studies.

One of the major advantages of this approach is that this catalogue of generic events and topologies can be saved to an external disk, made available to reuse it on similar systems. Even multiple catalogues from different kART simulations can be merged, creating a larger database that can be used increase the speed in next simulations [48].

2. Reconstruction of the geometry and refinement of low energy events.

Although it is assumed that all atoms sharing the same topology will have access to the same generic events, a small adjustment to the energy barrier due to local variations in position or long ranged elastic interactions need to be done, forming then what are called the specific events.

These phenomenons affect every barrier differently so that they give rise to favoured directions even for formally isotropic defects. To do so, kART should reconstruct every generic event for each atom and relax it again, taking into account only kinetically relevant events to avoid an excessive computational cost. In this sense, only the lowest energy barrier events (up to a given threshold typically 99.9% or 99.99% of the total rate) are fully reconstructed and refined for each KMC step, representing one to ten percent of all barriers in the catalogue (depending on the system). This idea is showed in figure 2.8.

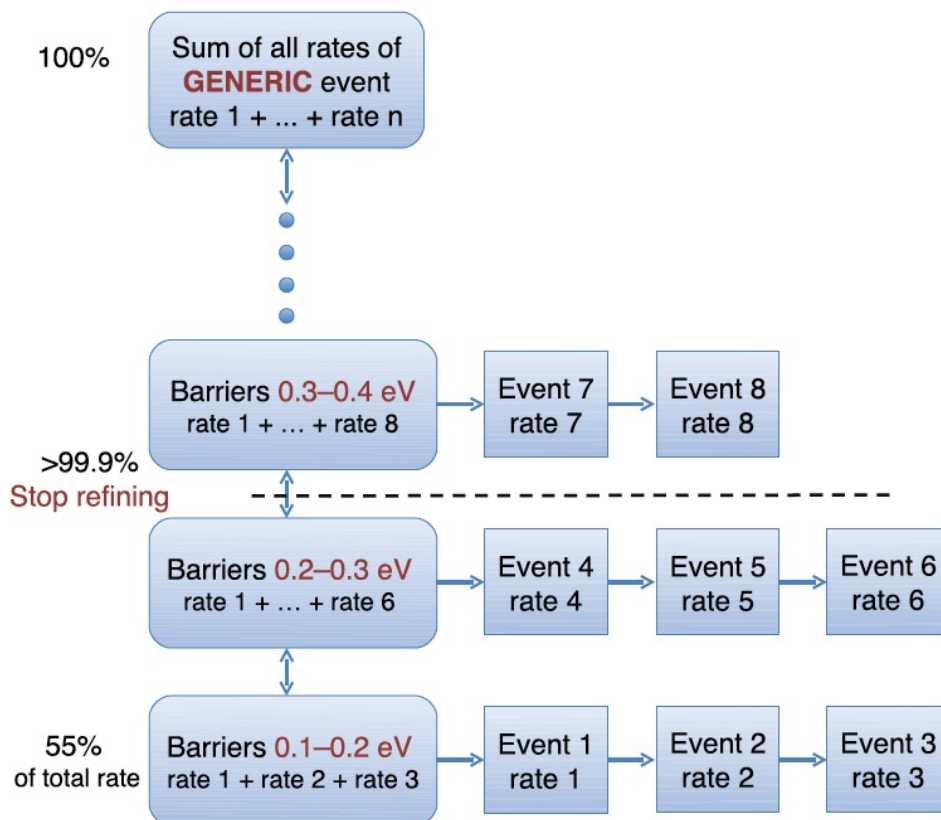


Figure 2.8: Refinement of generic events into specific events in kART [48].

In any case, the local reconstruction takes place in two steps:

- (a) The geometric transformation necessary to map the initial state onto the current configuration and to the atomic displacements between initial state and saddle point is determined by using the reference atomic positions of the generic event.
- (b) The system is relaxed into the final state by pushing it over the saddle point, considering the one-to-one correspondence between topology and geometry while mapping is carried out.

Figure 2.9 illustrate this concept.

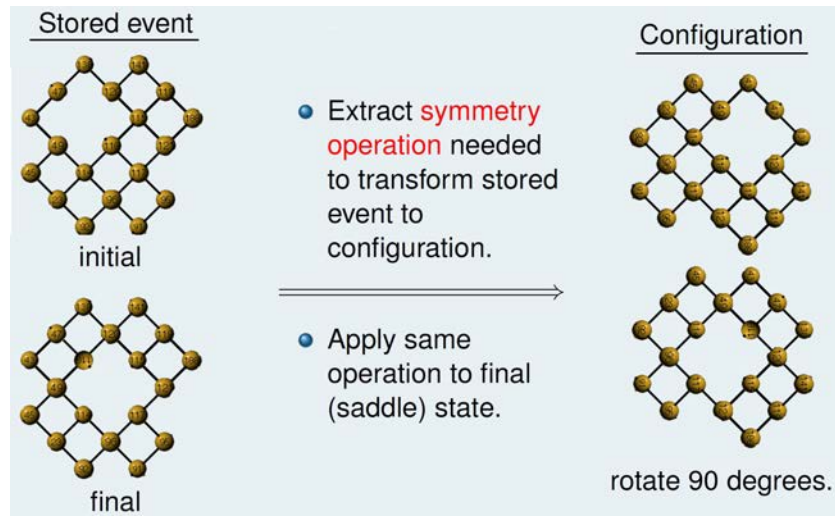


Figure 2.9: Example of events reconstruction in kART [43].

3. **Applying the kMC procedure.** Having the information about all the possible events, one can make a list of these events which serves to a kMC algorithm whose rates are determined according to transition state theory:

$$r_i = \tau_0 e^{\left(\frac{-\Delta E_i}{k_B T}\right)} \quad (2.6)$$

Where r_i is the rate associated with event i , τ_0 is the attempt frequency ($\sim 10^{13} s^{-1}$), ΔE_i is the barrier height ($E_{saddle} - E_{initialminimum}$) and k_B is the product of the Boltzmann constant and the temperature, respectively. Since τ_0 varies only weakly with the chosen pathway for the vast majority of systems, combining the previous

equation to the rate of harmonic TST (see equation 2.2), it can be extracted the Poisson distributed time step Δt over which an event will take place:

$$\Delta t = \frac{-\ln \mu}{\sum_i r_i} \quad (2.7)$$

Where μ is a random number between $[0,1]$ which ensures a Poisson distribution. Thus, the simulation clock is pushed forward by this Δt and an event is selected randomly according to the KMC procedure [74], i.e., considering a weight proportional to its rate. Then the atoms are moved accordingly and, finally, the whole system is relaxed into a minimum energy configuration. Hence, the next step can be launched.

2.2.3.3 Crucial implementation details

Despite of the fact that we have already described the method, there are some relevant aspects which are missing and they should be commented [47, 48, 75]:

- **Handling low-energy barrier.** Since we are using a KMC approach, dynamics are dominated by the lowest energy barriers of the system. Therefore, in the event of a energy landscape consisted of basins with numerous states connected by very low energy barriers compared to those needed to leave these basins, the algorithm would become trapped into computing non-diffusive events, as figure 2.10 shows. This issue affects directly to the attainable simulated time, which will be lower due to the higher total rate of this kind of barriers, and also increase the CPU cost without yielding meaningful physics.

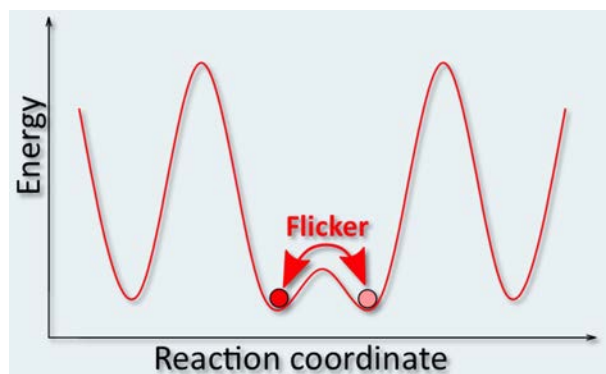


Figure 2.10: Example of kART trapping between two states with low energy barrier between them [43].

To address this problem the researchers who developed kART have created bac-MRM (basin-autoconstructing Mean Rate Method) [48], which is based on the widely known MRM [76]. We are going to describe both of them due to their essential role in this work. In short, MRM separates the trajectory into transient states and absorbing states and accelerates the simulation. Mathematically speaking, firstly it determines the probability to exit the basin to state x , we calculate the transition probability matrix $\underline{\mathbf{T}}$, whose components are:

$$T_{ji} = \frac{R_{i \rightarrow j}}{\sum_k R_{i \rightarrow k}} = \tau_i^{-1} R_{i \rightarrow j} \quad (2.8)$$

Where $R_{i \rightarrow j}$ is the associated rate to go from basin state i to basin state j , $\sum_k R_{i \rightarrow k}$ is over all basin and exit states k and τ_i^{-1} is the mean residence time in state i each time it is visited. Taking into account that the occupation probability vector of all basin states after in-basin jump m , $\Theta(m)$, is given by repeated application of $\underline{\mathbf{T}}$ to the initial occupation probability, then the sum of the occupation probabilities over all possible number of jumps is:

$$\Theta^{sum} = \sum_{m=0}^{\infty} \underline{\mathbf{T}}^m \Theta(0) \quad (2.9)$$

That gives the average number each basin state is visited and also the mean residence time before leaving the basin can be calculated:

$$\tau_i = \tau_i^{-1} \Theta_i^{sum} \quad (2.10)$$

And these residence times are then used to accelerate the basin exit so that the following KMC step is then determined using these accelerated rates:

$$\langle R_{i \rightarrow j} \rangle = \frac{\tau_i}{\sum_k \tau_k} R_{i \rightarrow j} \quad (2.11)$$

Hence, MRM can accelerate the simulation by averaging over all possible jumps between transient states, yielding the correct probability to exit a basin to a certain absorbing state.

On the other hand, the way bac-MRM works is adapting this state based method to an event based one (note the difference in terminology between bac-MRM and

MRM). In this sense, bac-MRM defines a basin event as an event which presents smaller energy difference between the forward barrier $b_{for} = E_{saddle} - E_{initial}$ and the backward barrier $b_{back} = E_{saddle} - E_{final}$ than a threshold defined by the user. This is a key point in our work because by modifying this threshold we have been able to avoid the mentioned flickers.

In addition, creates several basins which are formed by these events. Then, once an event is selected that matches the description of a basin event, the event is executed and then added to the current basin of events. Consequently, basins are identified on-the-fly by the barrier heights separating the basin state.

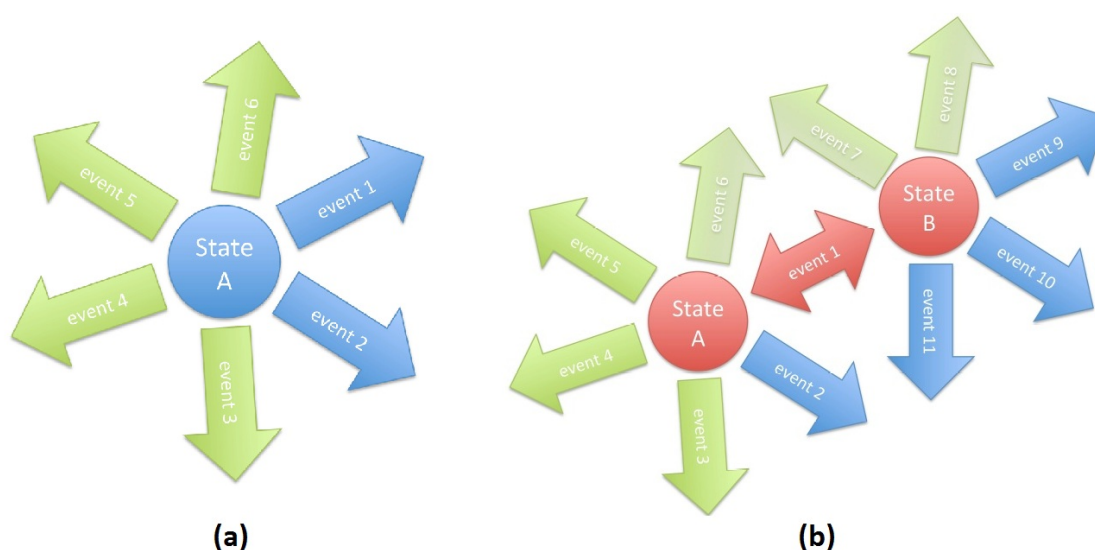


Figure 2.11: Construction of a basin formed by state A (a) and states A and B (b) with their possible events (Green: ordinary events, Blue: potential basin events, Red: basin events) [43].

A concreted mode of operation can be seen in figure 2.11. Firstly, after generating the list of possible events from the initial minimum (a), the ones with low barrier are marked as potential basin events, storing their associated atomic displacement. If such an event is picked for execution (event 1 in the figure), it is added to the current network of basin events and removed from the tree of available events, executing in following the normal procedure of kART. Once done this step, the system is in state B so that kART event finding algorithm is started from this state, keeping all events from previous basin states in the tree and, therefore, being able to be picked (b). Here is when MRM come onto the scene, modifying the rates of leaving both states according to the equation 2.11 and also adjusting the total rate.

In this point the next step can lead to a new basin state, such as events 2, 9, 10 or 11, which would be added according to the procedure described above, or out of the basin, e.g., events 3, 4, 5, 6, 7 and 8, which would imply a standard kART move. However, if an event is found to lead to an already explored basin state, such as event 1, it is rejected. In addition, this event is also removed from the tree, and added to the basin, adjusting the rates as needed. It is common that kART chooses basin events until all basin events are added to the basin. Note that it can be chosen all possible events inside the basin despite of unsharing the initial state. Consequently, following this method basins are explored on-the-fly but as far as necessary, and what we really benefit from it is that no state is intentionally visited twice. In any case, thought all the events inside a basin are basin events and share the mentioned properties, it should be stressed that there are several slight different states inside the same basin. Hence, before the next KMC step, events originating from all states from the current basin could be selected and the continuity of the simulation can be lost. Despite of this drawback, bac-MRM yields the correct distribution of exit states depending on the basin internal rates and the point of entry into the basin without consuming CPU resources in practice.

- **Optimizing kART for large scale systems.** Although working with large systems can be seen as a problem for kART, this idea is wrong due to the nature of activated mechanism. The diffusion phenomenons tend to occur in a local region inside the lattice, so that they affect to regions composed of 10-100 atoms with induced forces propagated up to a few nm. Hence, the issue can be simply addressed by an algorithm which construct neighbour list and calculate the forces locally.

To do so, after defining a cut-off radius to consider a nearby atom as a first neighbour (typically ~ 2.8 Å for crystalline Si, as we have already mentioned in the topological characterization in section 2.2.3.2) and also for second neighbour (~ 4 Å), all the atoms become labelled in order to know the region around each one. Then, when an event is being launched, local forces are first computed on all atoms involved in the event plus their first and second neighbours. In addition, in case surrounding atoms around the initial one have a force which exceeds a certain threshold, their first and second neighbours are added to the list. Hence, it is assured that forces are computed only on the relevant atoms in the event. In any case, whereas the event generation is local, it should be noted that a global minimization if performed after each KMC move in order to take into account all elastic effects.

- **Box size and boundary conditions.** Although this point can experiment changes in the near future, kART allows only parallelepiped boxes with PBC (*Periodic Boundary Conditions*) in all directions. It should be noted that the simulation box, also known as simulation cell, is the three-dimensional region where all the particles can displace.

This kind of periodic conditions implies that the cell is repeated infinitely in axis X, Y and Z. Hence, our initial cell is surrounded by multiple image cells whose particles are moved keeping solidarity respect to the initial one. However, there is no need for all of them to move, but only our cell is simulated as well as those particles which enter and leave from our region. In this sense, the affected particles of our initial cell would have image particles which would follow the movements of the original one. This idea can be checked in figure 2.12.

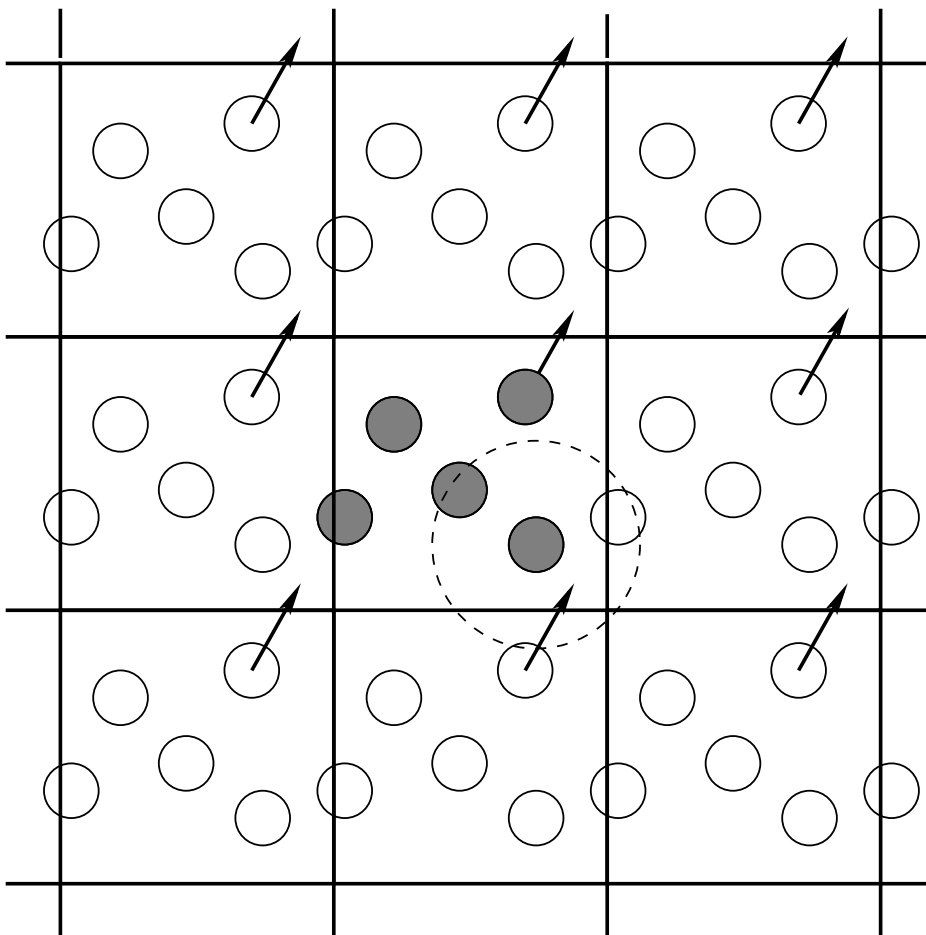


Figure 2.12: Meaning of PBC in a 2D cell [77].

Thus, following this method we are able to avoid losing particles in the boundaries, keeping the number of particles constant and also removing the well-known boundary effects in the nanoscale. Expressing this idea mathematically, if we consider a cubic system with side L , every particle i and its image i' will verify:

$$\vec{r}_{i'} = \vec{r}_i + \vec{n}L \quad (2.12)$$

$$\vec{p}_{i'} = \vec{p}_i \quad (2.13)$$

Where $\vec{n} = (n_1, n_2, n_3)$ is a vector of arbitrary whole numbers, getting an infinite value of possible combinations of images with share the same moment as the original. Thus, when we apply PBC the cells are embedded into a infinite volume and, therefore, the behaviour of the material is reproduced accurately. It should be mentioned that, obviously, when one have to determine the forces of the system it is needed to consider the dependencies between all of them:

$$U(\vec{r}_1, \vec{r}_2, \vec{r}_3, \dots, \vec{r}_N) = \sum_{i < j} v(\vec{r}_{ij}) + \sum_{\vec{n}} \sum_{i < j} v(|\vec{r}_{i'} - \vec{r}_j|) \quad (2.14)$$

Where the first term represents the used potential and the second one indicates the dependency regarding to the image particles. In any case, to make this problem feasible, the short range interactions are restricted to the surrounding particles, defining a cut-off radio which can also cover images particles [36].

2.3 Molecular Dynamics

As we have work integrating a MD simulation program into kART, we are going to mention certain aspects of MD, focusing later on defining the interatomic potential we have used and also giving an insight into their role in this paper.

2.3.1 Brief description

According to forcefields models from an atomic point of view, a system can be described as a collection of atoms which are influenced by the interaction forces. The vast majority of these interactions are determined by a potential energy U , which acts as a

pair potential. The analytical expression of U is determined by the used potential which is Tersoff in our case (see section 2.3.2). This potential calculates the potential energy between pairs of atoms in accordance with two functions, one attractive and one repulsive, which are limited softly with a cut-off function f_C . This function is showed in figure 2.13 and its expression for a pair of atoms i y j with positions r_i y r_j is the following:

$$f_C(r_{ij}) = \begin{cases} 1, & r_{ij} < R_{ij} - D_{ij} \\ \frac{1}{2} - \frac{1}{2} \text{sen} \left(\frac{\pi}{2} \frac{(r_{ij} - R_{ij})}{D_{ij}} \right), & R_{ij} - D_{ij} \leq r_{ij} < R_{ij} + D_{ij} \\ 0, & r_{ij} \geq R_{ij} + D_{ij} \end{cases} \quad (2.15)$$

Where the parameters R y D are selected to include the first neighbour nucleus for a certain structure.

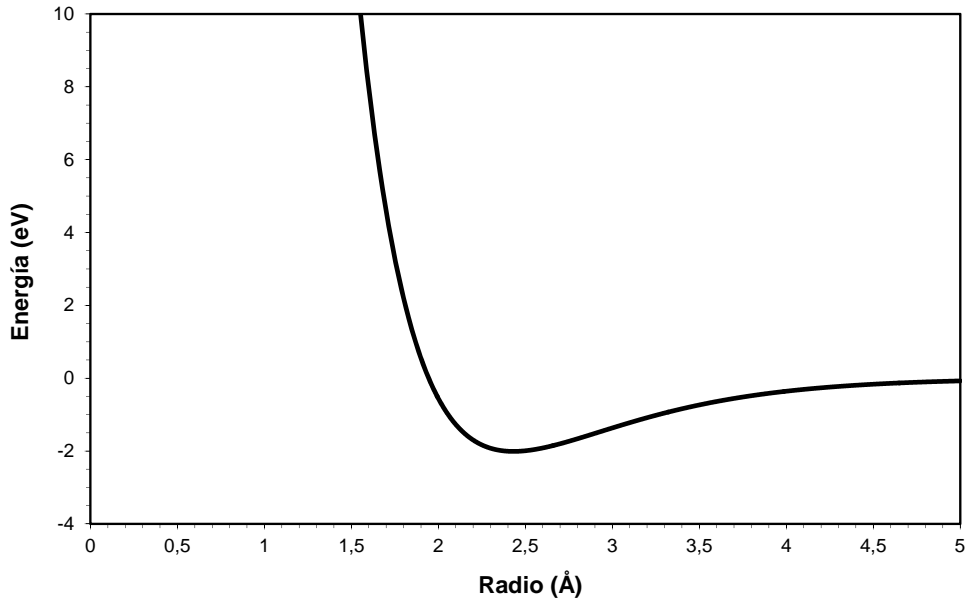


Figure 2.13: Representation of Tersoff potential for $R=2.95$ and $D=0.15$.

Taking into account that U depends on the positions \vec{r}_i of each particles, which are totally N , the potential energy can be expressed as:

$$U = U(\vec{r}_1, \vec{r}_2, \vec{r}_3, \dots, \vec{r}_N) \quad (2.16)$$

Thus, the forces over every N particles can be obtain deriving U in all the axis:

$$F_{ij} = \frac{\partial U}{\partial r_{ij}}(\vec{r}_1, \vec{r}_2, \vec{r}_3, \dots, \vec{r}_N) \quad \text{para } i = 1, 2, 3, \dots, N \quad j = x, y, z \quad (2.17)$$

Hence, it can be determined the trajectories followed by the particles over time by integrating the Newton's equations of move. To do so, starting with an initial configuration os positions and momentums:

$$m_i \frac{d\vec{r}_i}{dt} = \vec{p}_i \quad (2.18)$$

$$m_i \frac{d\vec{p}_i}{dt} = \vec{F}_i \quad (2.19)$$

Where m_i y \vec{p}_i are the mass and the linear momentum of the i th particle, respectively.

In this sense, the MD method determine the trajectories in the phase space (positions and momentums) inside a system whose particles follow the classical mechanic laws. This calculus cannot be implemented analytically, but instead the move equations can be discretized and solved by a computer iteratively, using an integration algorithm, as figure 2.14 shows. Consequently, it should be highlighted that there is an inherent error due to the discretization process, whose value depends on the memory resources and the integration algorithm. The information about the physical magnitudes of the system can be inferred by doing time-weighted averages over the simulation period [36].

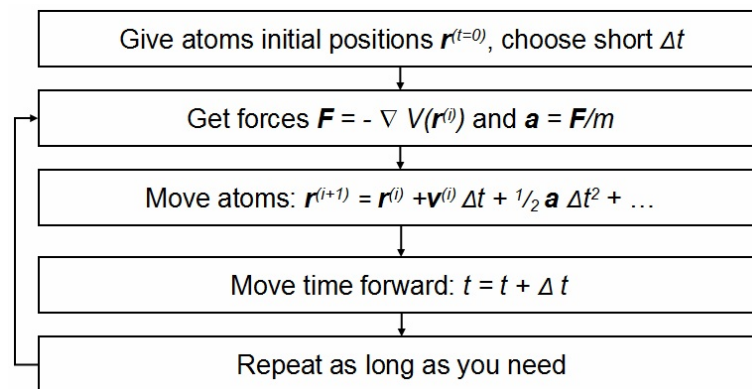


Figure 2.14: Simplified calculating method in MD simulations [77].

2.3.2 Interatomic potentials

As we have already explained, MD needs a numerical method to determine the forces between the atoms and, therefore, solve the move equations. Although we have mentioned some of them, such as DFT or TB, interatomic potentials are used from an atomistic point of view due to the size of the cell and the dynamic context. Hence, these kinds of potentials do not spend time calculating the fundamental electronic state, instead they reproduce the interaction between the atoms through empirical equations. In fact, they are a combination of empirical functions which depend on atom positions and parametric settings, giving a full description of the energies of system and the forces applied over each particle [36].

All empiric potentials follow the classical laws of mechanics so that the kinetic and potential energies can be expressed as [38]:

$$V_T = \sum_i^N V_1(r_i) + \sum_{i,j>i}^N V_2(r_{i,j}) + \sum_{i,j>i,k>j>i}^N V_3(r_{i,j,k}) + \dots \quad (2.20)$$

Being m_i and v_i the mass and velocity of atom i , respectively, $V_1(r_i)$ the effect of an external field on the system, $V_2(r_{i,j})$ the pair potential that only depends on the distance among the atoms i and j , and $V_3(r_{i,j,k})$ the triplets of interacting particles. It should be noted that the second term, $V_2(r_{i,j})$, is the one with the highest contribution to the total potential and third and higher order terms can be ignored for most applications. It should be noted that once the total energy has been evaluated ($E_T = K_T + V_T$), it is possible to calculate the stability of one system structure over one other, determine energy changes as a function of system deformation, etc. However, the accuracy of these properties is related to the empirical potential used.

Focusing now on semiconductor materials simulations, there are three widespread interatomic potentials: Stillinger-Weber potential, Tersoff potential and Environment-dependent interatomic potential. The first two ones were aimed to Si although some current versions support other compounds, represent the pioneering potentials for Si, whereas the third one has a wide range of materials. Here we are going to discuss about their differences and their possible applications, and also explain why we have mainly chosen Tersoff in our work.

2.3.2.1 Stillinger-Weber potential

It was proposed in 1985 down to the increasing interest in studies of computer simulations. Some previous pair potentials, such as the Lennard-Jones potential, were unable to stabilize the structure of diamond of Si under normal conditions and also to reproduce its molten phases, so Frank H. Stillinger and Thomas A. Weber decided to develop their own potential, widely known as SW (*Stillinger-Weber Potential*) [78]. Fundamentally is a three-body potential which considers the angular dependencies of atoms ($\phi_3 \propto (\cos \theta_{ijk} + \frac{1}{3})^2$). This potential reproduces the behaviour of liquid and crystalline phases of semiconductor materials accurately, having been also applied to study extended defects.

2.3.2.2 Tersoff potential

A year later J. Tersoff published their approach [79], which is known as Tersoff 1. He realized that the interatomic potentials which had been come out to date did not describe the properties of non-tetrahedral structures of Si. Thus, their fundamental idea was that the force of an atom bond depends mainly on its position and its neighbours. Following this principle, the more coordination number it has, the less force the bond will be. At the same time, he also included several body effects, being able to be modelled as a Morse potential.

Parameter	Value for SI
A (eV)	$1.8308 \cdot 10^3$
B (eV)	$4.7118 \cdot 10^2$
λ_1 (\AA^{-1})	2.4799
λ_2 (\AA^{-1})	1.7322
α	0.0
β	$1.0999 \cdot 10^{-6}$
n	$7.8734 \cdot 10^{-1}$
c	$1.0039 \cdot 10^5$
d	$1.6218 \cdot 10^1$
h	$-5.9826 \cdot 10^{-1}$
λ_3 (\AA^{-1})	1.7322
R (\AA)	2.85
D (\AA)	0.15

Table 2.3: Tersoff 3 parameters [80].

Nevertheless, Tersoff 1 potential cannot stabilize the diamond network of Si through MD, so that it gave rise to other versions. To do so, J. Tersoff also developed Tersoff 2 and Tersoff 3 [80] by using different parametrizations from the first one, whose values can be seen in table 2.3. In this sense, Tersoff 3 is the most accurate and extended interatomic potential to reproduce Si. The only problem it presents is that it requires using equivalences because it gets a melting temperature higher than the experimental value. In any case this is not a serious drawback since it is possible to make a temperature rescaling between Tersoff temperatures, T_{T3} , and real temperatures, T_{real} , both in K, using the expression [38]:

$$T_{T3} = 6.95 \times 10^{-4} \times T_{real}^{-2} + 2.66 \times 10^2 \times T_{real} + 3.79 \times 10^2 \quad (2.21)$$

2.3.2.3 Environment-dependent interatomic potential

Some years after the previous potentials were published, in 1997, the EDIP (*Environment dependent interatomic potential*) were proposed [81]. It was based on using the *ab-initio* innovations to arrive at a functional form that describes the dependence of chemical bonding on the local coordination number by using two-body and three-body terms which depend on the local atomic environment. Thus, bond order, hybridization, metallization, and angular stiffness are all described in qualitative agreement with theory, keeping the form as simple as possible and reproducing the essential physics with little more complexity than existing potentials which share the same aim.

2.3.2.4 Comparative

Over the last two decades, new functional forms have been proposed in an attempt to improve the description of Si. There has been different studies published which contrast the existing empirical potentials for Si [82], having found that, although SW and EDIP potentials are good at describing the liquid phase epitaxial regrowth in Si, Tersoff potential is the one which provides better description of the solid phase epitaxial regrowth and both point defects and structures different from perfect diamond, and several phenomena related to ion irradiation in Si [38]. Hence, since we were aimed to work with extended defects, Tersoff 3 has been the chosen potential. However, Chapter 4 implements also SW potential in order to check the simulation technique, comparing its results with Tersoff potential ones.

2.3.3 Implementation with kART

One of the flexible features of kART is that can be compiled with the LAMMPS (*Large-Scale Atomic/Molecular Massively Parallel Simulator*) [83] library, giving it accesses to a wide range of efficiently programmed forcefields. In addition, kART also includes a modified version of the Fortran interface to the library as the original one has some limitations.

In short, LAMMPS is the most extended software for MD parallel simulations nowadays. It is an open source program under GNU GPL *GNU General Public License* developed in the 1990s decade due to a CRADA (*Cooperative Research And Development Agreement*) between Sandia and LLNL (*Lawrence Livermore National Laboratory*) laboratories, which belong to the DOE (*U.S. Department of Energy*). At this point we need to highlight that their wide range of application, it can be implemented in solid state materials, semiconductors, and even in biological materials such as biomolecules or polymers. Hence, it is a widespread software used in a large number of publications.

Nevertheless, we should emphasize that the role of LAMMPS in this work is reduced to calculated the energy of the system with the given interatomic potential. In this way, kART does not support Tersoff by itself, but what we have done is introduced it thanks to LAMMPS.

2.4 Used tools

This section will present the applications we have used and also certain programs written in C/C++ developed by ourself, which have enabled us to analyse the data results by carrying out automatic calculations and extract straightforward conclusions from the files originated from a kART simulation.

2.4.1 Visualization tools

Among all the different possibilities available to represent networks at an atomic level, we have mainly used RasMol in order to transform the atomic positions given from a simulation into 3D structures so that we are going to describe its main features and how we have made use of it in the following lines.

RasMol was created by Roger Sayle in 1992 and is an open source program with GNU GPL since version 2.7.3 was launched onto the market. Furthermore, the scope of RasMol is not only limited to make representations of molecules for informational purposes, but also a tool used by thousands of researchers around the world to write its papers with high quality resolution [84].

Their operation mode is quite simple: the software can generate a 3D network from a given input file where the atomic positions are indicated, following a predefined format such as the well-known PDB (*Protein Data Bank*) or CIF (*Crystallographic Information Framework*). Furthermore, the represented atomic network can be customized according to the user preferences. It can be chosen the size and colour of atoms, the addition of bonds and their colour, the zoom level, etc. (see the manual for more information). Figure 2.15 shows a sample of what can be done with this tool.

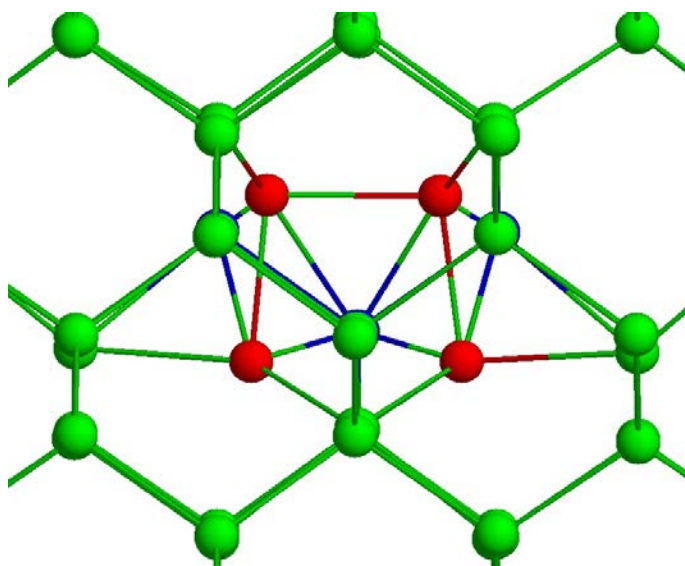


Figure 2.15: Sample of representation with RasMol: extended configuration of self-interstitial over XZ axis in Si (Red: displaced atoms, Blue: empty spaces, Green: bulk atoms).

However, the most relevant feature of RasMol is the network rotation by dragging with the mouse. This option is quite interesting when we need to see defects in detail. In addition to what is already mentioned, the possibility of scripting has also played key role in this work. An script has enabled us to generate consecutive pictures of what RasMol has in its screen the times we need. We have benefit from it a large number of times:

- Representation of all steps from a given simulation in one go. We have frequently work with simulations of ~ 1000 steps.
- Representation of all orientations planes from each step at once. As it can be seen in the appendix A. We have usually worked with 11 planes: $[0\ 0\ -1]$, $[0\ 1\ -1]$, $[0\ 1\ 0]$, $[0\ 1\ 1]$, $[-1\ 0\ -1]$, $[-1\ 0\ 0]$, $[-1\ 0\ 1]$, $[-1\ 1\ -1]$, $[1\ 1\ -1]$, $[-1\ 1\ -1]$ and $[-1\ -1\ -1]$.
- Making short films using the pictures given by RasMol as a frames in a video editor (~ 30 fps).

2.4.2 Processing tools

kART generates a great amount of data which should be analysed if we want to extract some interesting conclusion from them. Thus, we have developed several programs written in C/C++ which process the main output files of kART: *allconf* (a heavy file which contains all the atomic positions at every KMC step), *Energies.dat* (file which contains the total energy value from each step among other data) and *Diffusion.dat* (file which contains the total MSD (*Mean Square Displacement*) from each step among other data). For more information, see section 3.3.

Taking this files as inputs in our programs, we have calculated certain parameters and characterized each step. Below is an enumeration of the more representative ones:

- **Energy tree generator.** It consists of analysing the energy of each KMC state, visited or not, taking into account all the happened or possible transitions so that it can be created a flowchart which have all the states sorted by energy and linked according to these transitions.
- **Film generator.** It discretizes all the atomic positions from each step, letting RasMol to give one picture per step and making a film then.
- **Linux script generator.** It is a high flexible program capable of making Linux scripting for several purposes: launching certain simulation in one go, merging of several images into one thanks to ImageMagick, movement of thousands of files at once, etc.
- **Characterization of each kART step.** This is done by matching the input formation enthalpy with an internal database and also link each step with the

suitable information provided by the simulation program in different files. It can be applied to I_2 , I_3 , I_4 and I_5 cells at several pressure values. For more explanation, see section 3.4.

The appendix C includes the code of some of them. It should be noted we have not attached all the codes due to their length.

2.4.3 Additional software

2.4.3.1 Communication tools

We have to do remote access in order to communicate with the clusters, either Beta or Gamma. In order to make easy this task, we have used two programs: X-Win32 y SSH Secure in its versions *Shell Client* y *File Transfer Client*. Whereas X-Win32 and *Shell Client* let the user execute commands in a remote machine, *File Transfer Client* is aimed to file transference due to its simplicity, as figure 2.16 shows.

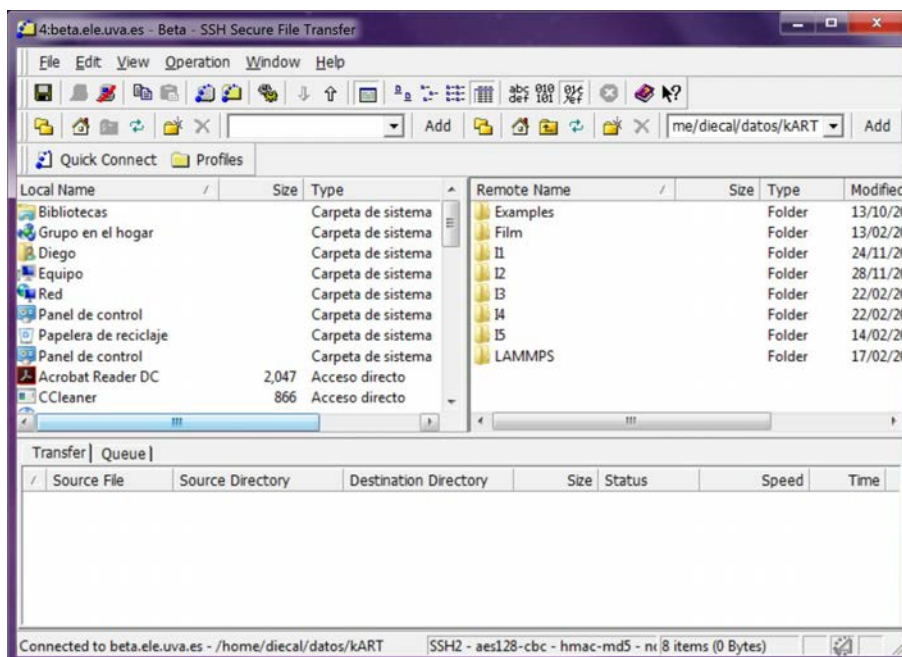


Figure 2.16: Screenshot of *File Transfer Client*.

2.4.3.2 Graphic tools

Our simulation results have made us do certain flowcharts in order to interpret them rightly. GVedit is a Graphviz tool [85] for creating, viewing, editing and processing DOT files. It allows users to set attributes of graphs with dialogue boxes and save them for future use. It should be noted that DOT is a declarative language in which you express nodes and their relations in a graph. You can label the nodes and their edges (links between nodes) and you have an array of styling and shaping tools at hand. Figure 2.17 shows an instance of what can be done with this software.

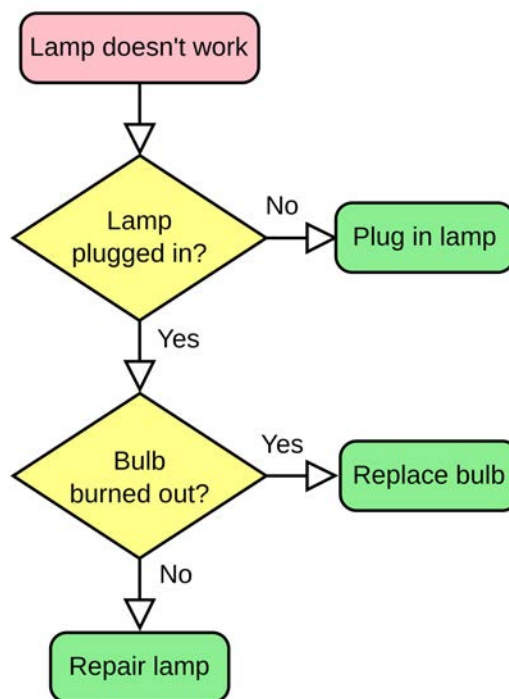


Figure 2.17: Sample of flowchart with GVedit.

Chapter 3

Simulation procedure

*It is important to do everything with passion,
it embellishes life enormously.
~ Lev Landau ~*

THE AIM OF THIS CHAPTER IS to detail the methodology we have used to determine the formation enthalpy of each configuration and the energy barriers of the associated transitions for each cluster size we have analysed in order to ensure the reproducibility of our results.

Firstly, the origin of the initial atomic coordinates of each structure is going to be described, indicating the physical conditions of the cells. Afterwards we are going to give an outlook of the standard simulation conditions we have used, explaining in depth the parameters we have mainly modified and their implications. Then it is necessary to point the output files we have worked with and what we have done with this information to obtain the mentioned energy values. In this sense, due to the nature of our results we have determined the energy barriers assuming certain considerations which have to be cleared.

It should be noted that complement content to this chapter is available in the appendices [A](#), [B.1](#), [B.2](#), [B.3](#) and [C](#). They are code examples which can be reused if one has the appropriate software and adapt them to their hardware resources.

3.1 Initial configurations collection

The initial atomic coordinates we have used for each configuration of all the clusters sizes under analysis ($I_{2 \leq n \leq 6}$, being n the excess of interstitials) are originated from a recent work of Santos *et al.* [86]. Their simulation methodology, in relation with our study, was the following:

- They have run CMD simulations based on Tersoff 3 potential for exploring the configurational landscape of interstitials clusters whose size are up to 5 atoms through LAMMPS code. The temperature was set to 1200 K over 25 ns with a simulation time step of 0.5 fs, fixing PBC in the three spatial directions. The size of the cells depended on the cluster size so that the pressure introduced by the excess interstitials is negligible. For $n = 2, 3$ it has been used a $6a$ Å cubic cell, containing $1728 + n$ atoms, whereas for $n = 4, 5, 6$ the cubic cell had a lateral size of $8a$ Å, holding $4096 + n$ atoms. Note that a is the lattice parameter of crystalline Si, 5.432 Å, and both cells were oriented in $X = (1\ 0\ 0)$, $Y = (0\ 1\ 0)$ y $Z = (0\ 0\ 1)$.
- Once running the annealing for each cluster size, a large variety of defect configurations (over 100) were extracted by discretizing in energy and, therefore, spatial arrangement of atoms. In this sense, they found not only several possible structures for chainlike, compact and 1 $\{111\}$ configurations, but also others atomic distributions less frequent.
- To enable us to work at 0 K, all the previous configurations were quenched and subjected to conjugate gradient energy minimization. This minimization consist of relaxing the cell through an algorithm which search the more stable configuration iteratively. The stop criteria can be several causes, such as energy tolerance. Hence, the minimum energy configurations were obtained.

Afterwards, we selected the minimum energy configuration for each cluster size, chainlike, compact or $\{111\}$ one, and, beginning from these chosen configurations, we have launched our kART simulations with the considerations given in the next section. It should be noted that these minimum energy clusters can be seen in appendix A, which also shows more energetic ones. The only configuration we have extracted from other author is the symmetrical in $\{111\}$ plane for I_3 , where the Carvalho [87] *et al.* proposal has been used instead the mentioned one.

3.2 Relevant simulations parameters

The explanation of all the parameters which can be modified by the user in kART is available in the official manual [75]. Almost every simulation we have run sets common standard parameters, whose more representative ones are presented below:

- **Maximum number of kMC steps (*NBRE_KMC_STEPS*):** 1000. It is long enough to be able to see transitions and ensure the simulation time is not too long.
- **Temperature (*TEMPERATURE*):** 500 K. Although the energy values we have obtained in this work are at 0 K, this is a parameter which has a great importance in the kMC algorithm, as we have already described. Increasing the temperature implies that it is likely to end up walking through high energy configurations with no physical interest.
- **Calculation of the system energy (*ENERGY_CALC*):** LAM. It is necessary to be able to implement Tersoff 3 potential through LAMMPS code.
- **Cut-off radius for local displacements (*RADIUS_INITIAL_DEFORMATION*):** 2.5 Å. Since we have chosen to carry out a local random deformation in generating an event, an arbitrary atom and its neighbours are selected and moved in a random direction, with the rest of the system allowed to react. To do so, the previous value sets the possible neighbouring area to perform the local moves, being enough to our work although it barely exceeds the first neighbour distance of crystalline Si (2.35 Å). It is worth noting that one can get a more diversified set of barriers from a certain event by increasing this parameter.
- **Maximum number of relaxation steps inside the basin (*MAX_PERP_MOVES_BASIN*):** 2. Simulating the system to this number of possible steps in the hyperplane perpendicular to the direction of deformation assures head-on collisions do not take place and that the total energy remains under control. Additionally, the probability of successful events is maximized for this value.
- **Threshold below which the system is reputed to have left the harmonic basin (*EIGENVALUE_THRESHOLD*):** -1. This value lets the system balance to maximise the number of successful events while making sure that the system does not jump basins.

- **Threshold at which we consider that the system is converged at the saddle point (*EXIT_FORCE_THRESHOLD*):** 0.05 eV/Å. We keep it in its default value due to the fact that the convergence has always been right.

For more information about where the previous parameters are applied within the kART simulations, see section 2.2.3. Furthermore, kART let the user set more than 120 parameters. Thus, the rest of the parameters were kept in their default values on a first approach. However, after doing a large number of tests without interesting results, we have concluded that the following variables are the ones which can lead us to transitions in the energy landscape given by Tersoff 3 when their values are modified:

- **Maximum height of direct barrier and inverse barrier for an event to be considered inside a basin (*MIN_SIG_BARRIER*):** from 0.3 to 2.1 eV. This parameter plays a key role in our simulations, giving us the possibility to reach more energetic configurations if we increase it. In other words, the higher value this parameter is set, the more events will be added to the basin and therefore, these events will be executed only once. Hence, we have swept this parameter to higher values (by increasing it 0.3 eV each test) in case we do not see an interest transition during a simulation. It should be noted that both direct and inverse barriers must be under the threshold to be considered inside the basin and the increment of this parameter means that the computational cost upscale critically.
- **Overall scale for the increment moves in activation (*INCREMENT_SIZE*):** 0.3 Å. Its defect value has been incremented in 0.2 Å so as to enable the simulation to push the system outside the harmonic basin. In addition, this parameter also contributes to the convergence phase because it is used again as a length step when the system is leaded to the saddle point.
- **Minimum number of event searches when a new topology is identified (*SEARCH_FREQUENCY*):** 50. We have also increased its value so as to give more chances to the system to determine the associated saddle point for each topology. However, it also implies consuming more hardware resources.

In addition to what have already been mentioned, the appendix B.1 presents a real simulation script we have used in order to carry out a simulation under the previously described conditions. It contains a brief description of all the possible parameters accepted by kART.

3.3 Considered output files

The atomic coordinates for each kMC step has been extracted from the file *allconf*, represented in figure 3.1. It presents a periodic structure for each step, giving firstly the number of atoms, the kMC step and the cell size, and then all the atomic positions. They are given in XYZ format, being compatible with many visualization programs.

```

1731
0 32.28000000 32.28000000 32.28000000
Si 0.669462844720163 0.670216031625203 0.660835501514896
Si 0.661559494734538 3.36041654093824 3.34121707394665
Si 3.35395183623319 0.667904414237269 3.34360384580219
Si 3.35978860848424 3.36419542921836 0.661324577939361
Si 2.01261005339684 2.01755135425811 2.00190761417251
Si 1.99959245325065 4.71131140556313 4.68147142498879
Si 4.69439291755654 2.01488440155650 4.68616104846837
Si 4.70222686133573 4.71190985769079 2.00443288756216
Si 6.05268143734544 0.675389226236889 0.663252435694677
Si 6.04377109194092 3.36323162809729 3.34836005488856
Si 8.73959179480643 0.676149531116672 3.35347088770987
Si 8.74164747538430 3.37022073365723 0.666744682947005
Si 7.39582632468700 2.02171659521116 2.00788862125634
Si 7.38459157533717 4.71043739578626 4.693934628488984
...

```

Figure 3.1: Snapshot of *allconf* output file of kART (Number of atoms: 1731, kMC step: 0, Cell size: 32.280 Å).

On the other hand, the energies and the temporal aspects for each kMC step has been extracted from *Energies.dat*. This file, which is exemplified in figure 3.2 lets you see the number of kMC steps, CPU times, simulated times, initial and final energy values of the minima, activated barrier executed and even topological information. Hence, it contains the basic information about the system evolution.

# KMCs	CPU time	Old energy	New energy	barrier	Total rate	time step	Sim time	Selec Ev	Init Topo	Sad Topo	Fin Topo
0	0.0000E+00	0.0000	-7995.5105	0.0000	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0
1	0.1402E+04	-7995.5105	-7995.1807	0.3772	0.5562E+10	0.1365E-10	0.1365E-10	636160	460222	1163852	1021236
2	0.2460E+04	-7995.1807	-7995.5105	0.0474	0.0000E+00	0.1188E-13	0.1366E-10	1082868	1041932	4661	199960
3	0.2578E+04	-7995.5105	-7995.1807	0.3773	0.2201E+10	0.1215E-08	0.1228E-08	636160	460222	1163852	1021236
4	0.2860E+04	-7995.1807	-7995.5105	0.0474	0.0000E+00	0.2247E-12	0.1229E-08	1082868	1041932	4661	199960
5	0.2946E+04	-7995.5105	-7995.1807	0.3773	0.6976E+10	0.1430E-10	0.1243E-08	636160	460222	1163852	1021236
6	0.3154E+04	-7995.1807	-7995.5105	0.1336	0.0000E+00	0.8147E-13	0.1243E-08	37574	1021236	338987	460222
7	0.3280E+04	-7995.5105	-7995.1808	0.3774	0.3759E+10	0.4037E-09	0.1647E-08	636160	460222	1163852	1021236
8	0.3470E+04	-7995.1808	-7995.1282	0.0533	0.4971E+13	0.1186E-12	0.1647E-08	100277	1021236	967509	967509
9	0.4984E+04	-7995.1282	-7995.1282	0.0012	0.1344E+14	0.4098E-13	0.1647E-08	590623	946708	967509	967509
10	0.5473E+04	-7995.1282	-7995.1809	0.0001	0.2193E+14	0.4660E-13	0.1647E-08	1255548	967509	967509	1021236
11	0.5614E+04	-7995.1282	-7995.5106	0.0604	0.0000E+00	0.2213E-12	0.1647E-08	1162066	967509	152055	865827
12	0.5828E+04	-7995.5106	-7995.1808	0.4669	0.4177E+10	0.1190E-09	0.1766E-08	42075	199960	933635	1041932
13	0.6029E+04	-7995.1808	-7995.5106	0.1371	0.0000E+00	0.8851E-13	0.1766E-08	37574	1021236	571787	460222
14	0.6121E+04	-7995.5106	-7995.1808	0.3777	0.5287E+10	0.4583E-09	0.2224E-08	636160	460222	1163852	1021236
15	0.6315E+04	-7995.1808	-7995.5106	0.0479	0.0000E+00	0.2257E-12	0.2225E-08	1082868	1041932	4661	199960

Figure 3.2: Snapshot of *Energies.dat* output file of kART.

Finally, we have got the MSD for each kMC step from *Diffusion.dat*, which provides not only the total MSD but also the MSD with the initial configuration as a reference, as it can be seen in figure 3.3.

#	Elapsed Time	Sqr Displ.	KMC step	Event delr
#	*****	*** Total ***	*****	*****
0.00000000E+00	0.00000000	0	0.00000000	
0.13650086E-10	0.73274086	1	0.85600343	
0.13661968E-10	0.00000081	2	0.85600284	
0.12282999E-08	0.73273984	3	0.85592870	
0.12285246E-08	0.00000135	4	0.85592810	
0.12428276E-08	0.73273882	5	0.85589717	
0.12429091E-08	0.00113746	6	0.84903434	
0.16466482E-08	0.74019019	7	0.85526249	
0.16467668E-08	0.65366549	8	1.08077012	
0.16468078E-08	0.65036590	9	1.12991440	
0.16468544E-08	0.72528253	10	1.07326624	
...				

Figure 3.3: Snapshot of *Diffusion.dat* output file of kART.

Furthermore, we have also worked with other files in order to solve certain issues or expand our knowledge about certain simulation aspects. A brief description of some of them is presented below:

- *KMC_log.txt* It provides a global summary of all simulation details, including parameters, code version, the date of the compilation and a number of parameters used in the simulation as well as the basic simulation time as a function of events.
- *sortieproc.xx*. They are a collection of files, one for each core running the simulation, giving explicit details about each step of the simulation. The master is always labelled 0 (*sortieproc.0*) and all other cores are slaves which are labelled from 0 to $N - 1$, where N is the number of cores being used.
- *selec_ev.dat*. It provides information on the event chosen by the code at each kMC step.
- *Gen_ev.dat* and *Spe_ev.dat*. They give information on the creation of generics and specific events, respectively
- *Topologies* and *topos.list*. They are an exhaustive list of all the topology keys observed during the simulation.
- *event_list_conf_x.dat*. They are numerous files inside a folder called *EVLIST_DIR*. Each one is related to a certain configuration, containing all the associated topologies and possible events from it. This has a great importance due to the fact that one can access not only the event which has been selected in a kMC step, but also the other events which have been not chosen. Hence, one can map all the possible events from a certain configuration, predicting what are the most likely ones beforehand.

3.4 Data processing

Taking into account that our aim was to characterize the energy evolution of the system in terms of formation enthalpy and energy barriers, firstly we have built an internal database which links univocally each structural configuration for each cluster size with its formation enthalpy. To do so, we have run simulations setting `MIN_SIG_BARRIER` to 0.3 in order to see all the interested low energy minima. Figure 3.4 shows an example of this task beginning from a compact I_3 configuration. Afterwards, by looking at the formation enthalpy of the minima which appear in these simulations, we have labelled them numerically, according to their energy value from lower to upper values. This approach is illustrated in appendix A, where the most likely structures for each configuration for each cluster size are represented.

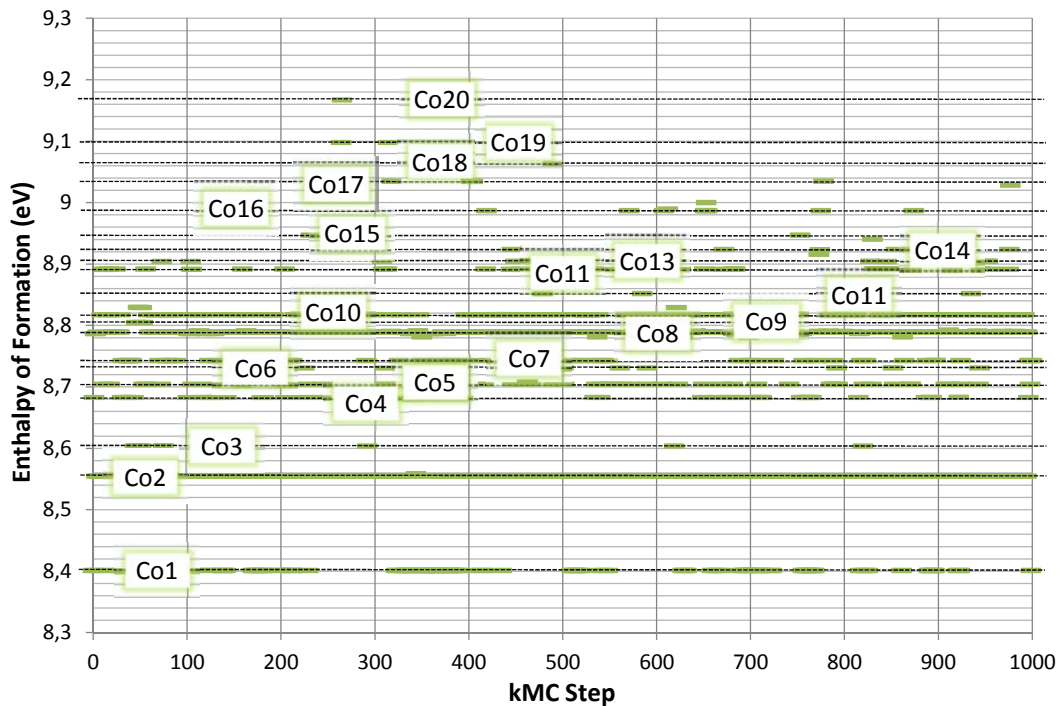


Figure 3.4: Example of energy minima identification in formation enthalpy evolution at 500 K and 0 GPa beginning from a compact I_3 configuration.

It should be noted that a single configuration can have several structures, each one with a different energy and spatial arrangement of atoms. The difference between two structures which share the same configuration, such as chainlike, tend to be in the position of certain atoms. In any case, no matter how its frequency is, we have considered all energy minima which have appeared through the mentioned simulations. In this sense,

in order to discretize in energy two states whose energy values are close, we have used a threshold of 0.001 eV to consider a structure as a new one. Overall, we have identified for each cluster size at 0 GPa:

- 24 structures for I_2 : 14 compact, 9 chainlike and 1 {111} configurations.
- 39 structures for I_3 : 20 compact, 18 chainlike and 1 {111} configurations.
- 19 structures for I_4 : 4 compact, 5 short chainlike, 9 long chainlike and 1 {111} configurations.
- 53 structures for I_5 : 16 compact, 26 short chainlike, 10 long chainlike and 1 {111} configurations.
- 21 structures for I_6 : 7 compact, 6 chainlike and 8 {111} configurations.

Hence, we have considered over 130 different chainlike, compact and {111} energy minima for 0 GPa simulations. Additionally, we have identified other configurations which are less frequent at the same pressure and we have also applied the same process at -3 GPa and +3 GPa, resulting in more than 500 elements in our database. Here we have to emphasize that when the cluster size increases ($I_{4 \leq n \leq 6}$), some intermediate structures between chainlike, compact and {111} configurations have appeared.

Once our internal list of possible formation enthalpies and labels for each configuration is built, we have run many simulations modifying the *MIN_SIG_BARRIER* parameter. In this sense, we have developed a C code program, which can be seen in appendix C, which let us process the cited output files automatically by comparing with our database. The way it works is simple: every kMC step is analysed by extracting the corresponding information from the files mentioned in the previous section. In this process, the energy comparison between a value of our internal list and the new simulation result is done rounding both to a meV. When the energies matching takes place, it is assigned the suitable label to the new energy.

As a result, one can make the figure 3.5 almost instantly. It should be noted that because of the variation of the minimum basin barrier new configurations which does not match with our database also appear, labelling them as “Other” to not overload our collection. In any case, whereas the number of unmatched visited configurations tend to be small if we compare it with the identified ones, we have always drawn the energy evolution graph based on visual identification of these new found minima.

Giving an insight into how we have done the graph, we have coloured the same kind of structures, e.g., compact structures, with the same colour. In addition, the energy evolution line cover initial and final energy minima and their barriers, which have their own symbol to emphasize them among the minima. Hence, continuity basin problems are detected easily due to the fact that a final energy minimum does not present the same energy as the following initial minimum. A vertical line indicates this issue in the corresponding step.

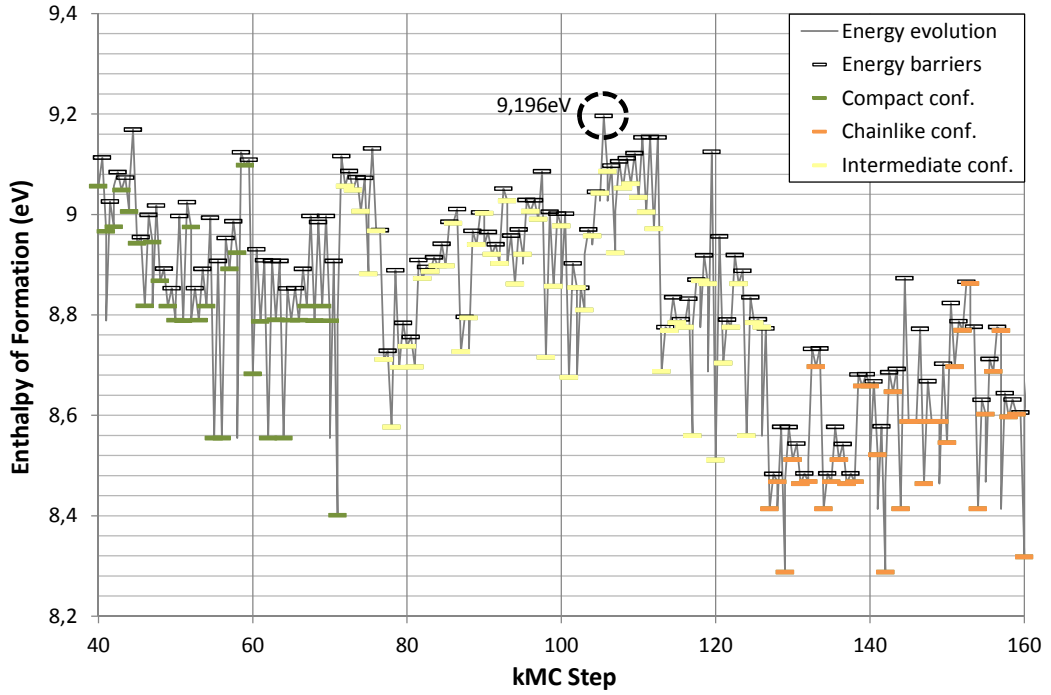


Figure 3.5: Example of configurational identification in formation enthalpy evolution at 500 K and 0 GPa with *MIN_SIG_BARRIER* at 0.6 eV beginning from a compact I_3 configuration.

In this way, the program we have mentioned above also prints an output file for each kMC step which is compatible with RasMol, making the visual identification an easy task. It is worth noting that this program only represents a cubic region around the displaced atoms and the empty sites which have been identified by focusing on the zone of the cell in which the defects are, as one can check in the appendix A. To do so, we select a cut-off distance of 7 Å for $I_{2 \leq n \leq 4}$ defects and 12 Å for I_5 and I_6 defects due to their longer size.

3.5 Determination of formation enthalpies and energy barriers

The calculation of the formation enthalpy H_F is simple. Once the potential energy of a perfect cell which contains a certain number of atoms is determined, it is rescaled including the number of inserted interstitials. Its difference with the potential energy of each kMC step from other simulation which has defects gives the aimed energy. Thus, one can indicate this idea as the following expression:

$$H_F = E_{P,defects\ cell} - \frac{N+n}{N} E_{P,ref} \quad (3.1)$$

Being N the number of atoms of the reference cell and n the number of inserted interstitials. It is worth noting that, following this procedure, both perfect cell and cell with defects are averaged to the same number of atoms $N+n$. It should be highlighted that the states of minimum energy provided by kART match with other values obtained by CMD techniques.

Regarding the way we have determined the energy barriers in our simulations, our approach needs to be explained. Since we have worked with an interatomic potential whose potential energy surface is rough, it is possible to have some intermediate minima between two different configurations, as it has already been shown. Hence, we have determined every barrier between two known configurations considering the most energetic barrier which has been overcome. In this sense, we have added an extra constriction based on the idea that both energy minima which are implied in the transition must be the lowest energy structure for their respective configuration, i.e., the lowest chainlike or compact or $\{111\}$ configuration.

In other words, if we are calculating the transition from a compact to a chainlike configuration, we would consider all the barriers between the lowest energy minima we have characterized for both compact and chainlike configurations, selecting the higher one. This method can be seen in figure 3.5, where the chosen barrier is marked with a discontinuous circle.

Chapter 4

Configurations and diffusion mechanisms of Si self-interstitial

*The most exciting phrase to hear in science, the one that heralds new discoveries, is not “Eureka!” but rather “hmm... that’s funny...”.
~ Isaac Asimov ~*

THIS CHAPTER PRESENTS a brief checking of kART through self-interstitial simulations by using different interatomic potentials, validating the method with the results obtained by other authors under similar conditions. Although this topic is distanced from the rest of this dissertation, it has enabled us to gain experience on this technique.

Native point defects in Si, both vacancies and self-interstitials, have been an important field of theoretical and experimental research for several decades. An attempt to outline the basic knowledge about Si self-interstitial is made before going to the main part. In this sense, the spatial configurations of this defect are also presented. It should be emphasized that it is the building block of small clusters.

kART is directly compatible with the SW potential and therefore, our first task was to verify the technique with this potential. Once the results we are going to summarize here had been obtained, we concluded that the program worked correctly. Afterwards, we also explored the energy landscape through the Tersoff potential, using it with the given considerations (see section 2.3.3). This issue was the most relevant part for the next part of the thesis. We experienced for the first time the kART behaviour when the Tersoff potential is used (there are no studies about both in the literature).

4.1 Introduction

The study of the Si self-interstitial properties is a quite active field of research due to its particular relevance. It has been shown that this kind of defect has been related to several well-known topics: as the origin of rodlike defects observed in Czochralski single-crystal growth, as an interacting particle with dopant atoms in the already mentioned transient-enhanced diffusion and as a key factor in the understanding of the amorphous phase formation. Although many authors have struggled to reveal the interstitial contribution to Si self-diffusion, this phenomenon is far from being fully understood. In this sense, theoretical studies have been carried out to determine its configurations and energies. There are first principles, TB and CMD calculations which come to different conclusions about its properties regarding to the lowest formation energy configuration and to the microscopic description of the interstitial-mediated self-diffusion mechanism [88].

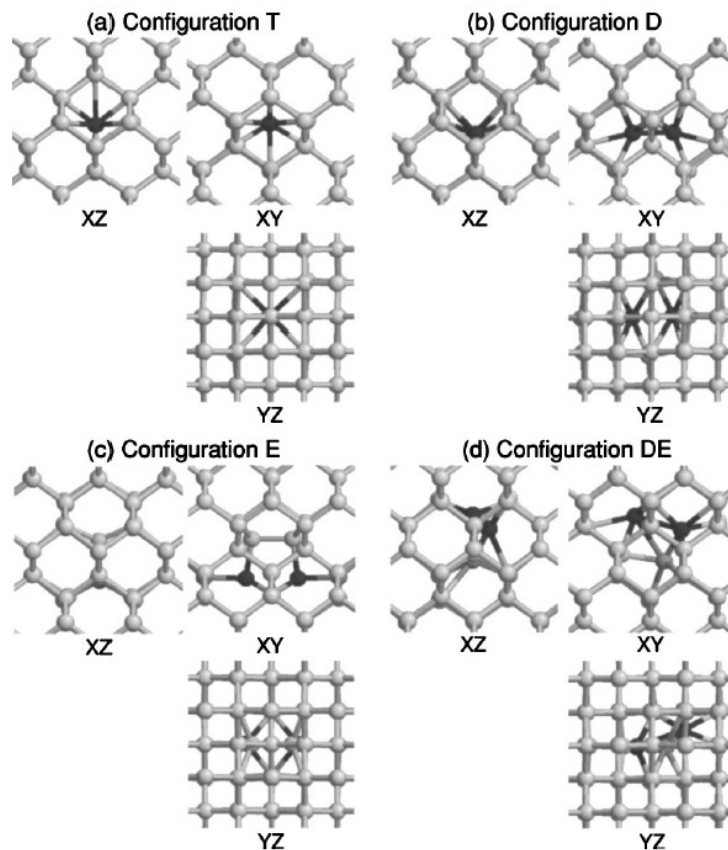


Figure 4.1: XY, XZ, and YZ projections of the different Si self-interstitial configurations (Gray scale represents potential energies, where darker tones correspond to higher values). [88].

Figure 4.1 shows the typical self-interstitial configurations. The so-called tetrahedral interstitial (T) only consists in 1 DA (*Displaced Atom*) which occupies a tetrahedral interstitial site, without any ES (*Empty Site*) surrounding, while the dumbbell interstitial (D) is formed by 2 DAs which are oriented along the $\langle 110 \rangle$ direction, sharing a common lattice site, and one ES. Despite of the fact that these two configurations are the most studied ones, it is been demonstrated that the extended interstitial (E), which consists of 4 DAs and 3 ESs that lie on a $\langle 110 \rangle$ plane plays also a relevant role. This configuration is highly delocalized, appearing during recrystallization from a planar crystal-amorphous interface in MD simulations. In addition, certain authors also claim the existence of a fourth configuration known as dumbbell extended (DE), which consists of 3 DAs and 2 ESs, less symmetrical and stable than the previous configurations.

For MD simulations using the Tersoff 3 potential, T is the minimum energy configuration while implementing SW makes D lower in energy, in accordance also with TB and first principles simulations. Nevertheless, it should be noted the high variability respect to the formation energy value for each configuration in the literature. The same issue takes place with the self-interstitial diffusion mechanism. The most accepted theory consist in moving the T interstitial between two neighbouring tetrahedral sites, being the D configuration the saddle point in the transition, for the methods which consider T as the lowest energy configuration. Regarding the other ones which have determined D as the most stable configuration, the mechanism is analogue but acting D as a reference and T as a saddle point. In any case, many transitions have been seen through diverse techniques which imply other configurations and mechanisms proposed by their authors, highlighting the transition between D and E configurations.

4.2 Stillinger-Weber potential

This section is dedicated to sum up all the results obtained by using the SW potential to determine the energetic aspects of the Si self interstitial. Since this chapter is quite different from the ones in which we analyse small interstitial clusters, we are going to detail the setting of our simulations before showing the formation enthalpies obtained for each self interstitial configuration and the determined energy barriers. Moreover, a brief comparison with other values found in the literature is presented.

4.2.1 Simulation methodology

Starting from a 513 atoms cubic cell of 21.7238 Å side length with a lattice parameter of $a = 5.432$ Å and oriented in $X = (1\ 0\ 0)$, $Y = (0\ 1\ 0)$ y $Z = (0\ 0\ 1)$, which contains a self-interstitial in a T configuration, we have run a kART simulation by using the SW potential (ENERGY_CALC = SWP). The temperature has been set to 500 K during 1000 kMC steps. The parameters have been set by default except the SEARCH_FREQUENCY = 50. Note that it means that INCREMENT_SIZE and MIN_SIG_BARRIER parameters are set to 0.1 Å and 0.1 eV, respectively.

Regarding how we have characterized each configuration, the approach here is different from what we have already mentioned for small clusters (see section 3.4 for more information). Every kMC step has been processed determining its DAs and ESs, and then, classifying them in T, D, DE or E thanks to their difference in the number of DAs. Taking this idea into account, figure 4.2 shows the energy evolution of our system in terms of formation enthalpies and their corresponding energy barriers for the first 20 kMC steps of the simulation. As it can be seen, T configuration is so unstable that it disappears in the first step and never comes back, evolving mainly to E and DE configurations, and sometimes to D configuration.

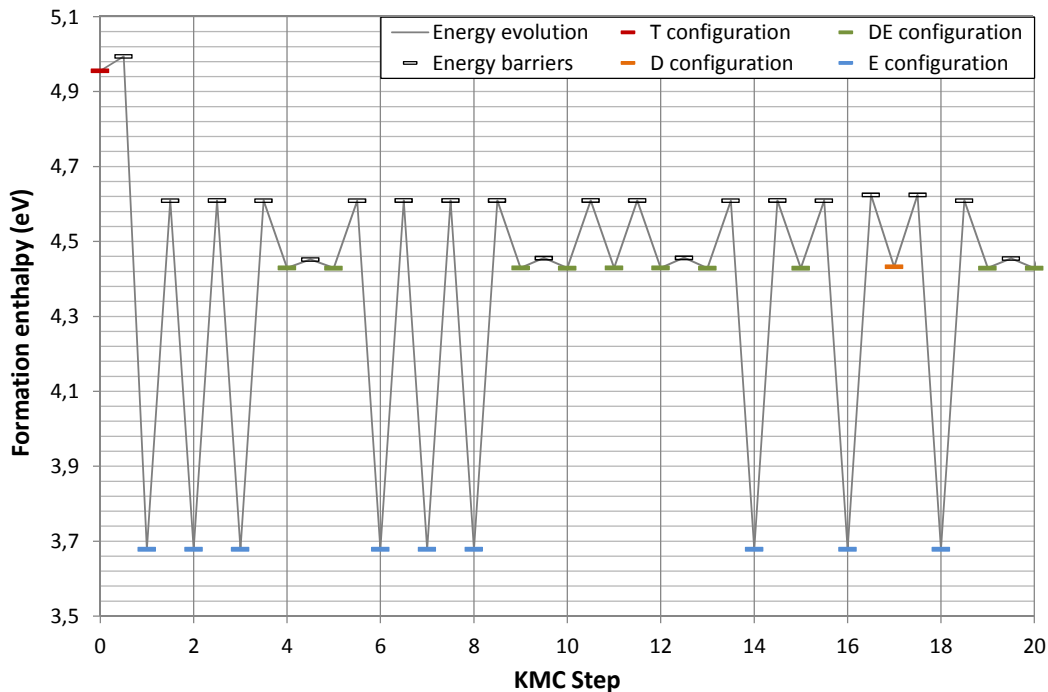


Figure 4.2: First 20 steps of the energy evolution of our SW based simulation beginning from a T configuration and labelling all the encountered configurations.

4.2.2 Formation enthalpies

Computing the formation enthalpy of each self-interstitial configuration over the 1000 steps, one can make the table 4.1. Since we have used SW to calculate the energies, E is the lowest energy configuration, followed by D and DE, whose values are so close. The energy of T configuration, however, is higher than the previous ones. It should be noted how accurate are the average values.

Type of configuration	T	D	DE	E
Number of times	1	58	535	407
Percentage	0.10%	5.80%	53.50%	40.70%
Formation enthalpy average (eV)	4.955	4.433	4.429	3.679
Standard deviation (eV)	0	2.691E-04	2.326E-04	5.142E-05

Table 4.1: Formation enthalpies averaged for each self-interstitial configurations predicted by the SW potential.

4.2.3 Energy barriers

We are going to do a similar approach with the obtained energy barriers, characterizing each value and discretizing every possible transition, as table 4.2 shows. The energy barrier between T and E configuration is extremely low, justifying what happened in the initial step.

Transition	Number of times	Energy barrier average (eV)	Standard deviation (eV)
T → E	1	0.038	0
D → E	31	0.179	0.005
D → DE	25	0.177	0.003
DE → E	252	0.066	0.072
DE → D	9	0.185	0.007
E → D	47	0.937	0.007
E → DE	237	0.864	0.078
DE ↔ DE	273	0.055	0.075
E ↔ E	122	0.934	0.035

Table 4.2: Energy barriers averaged for each transition by using the SW potential.

In fact, analysing in detail all the $E \rightarrow DE$, $DE \rightarrow E$ and $DE \leftrightarrow DE$ transitions, we can see that there are multiple energy barriers for each one. This idea is showed in figure 4.3, where it is represented all the chosen barriers for each transition over the 1000 steps in terms of its frequency. Hence, for $E \rightarrow DE$ transition there are three selected barriers, 0.773, 0.931 and 0.945 eV, for $DE \rightarrow E$ there are two possible barriers, 0.021 and 0.181 eV and for $DE \leftrightarrow DE$ transitions there are also three chosen barriers, 0.001, 0.0021 and 0.181 eV.

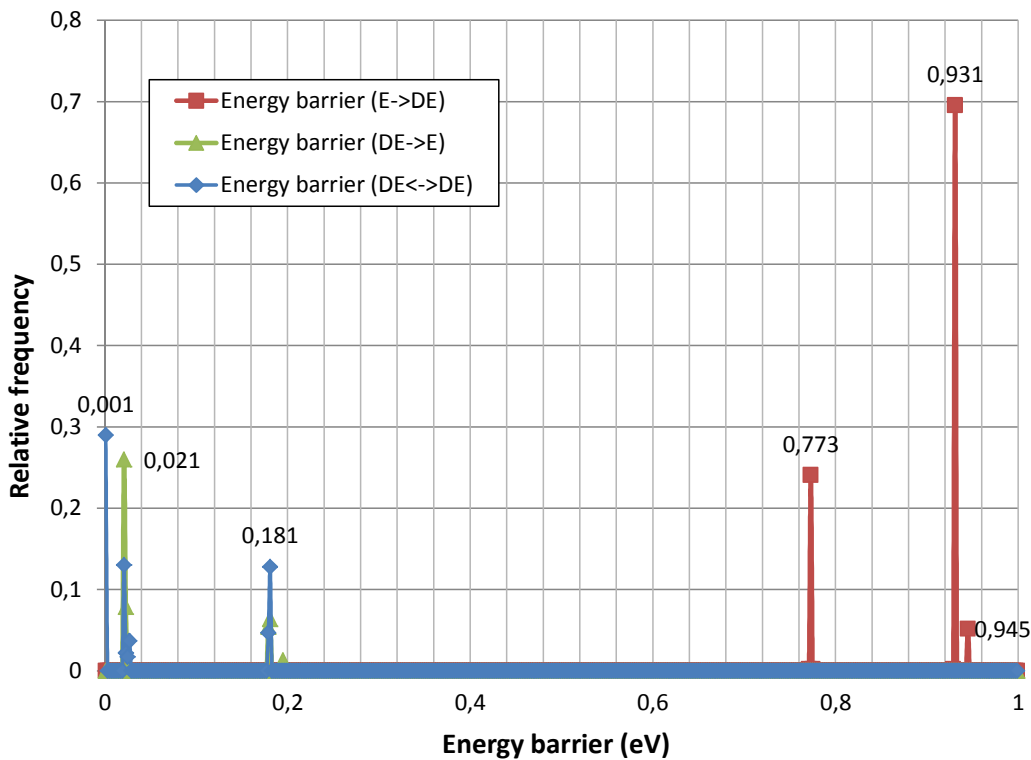


Figure 4.3: Relative frequency of energy barriers for $E \rightarrow DE$, $DE \rightarrow E$ and $DE \leftrightarrow DE$ transitions using the SW potential.

Considering the energy barriers which have appeared more often during the simulation, e.g., the ones with more relative frequency, and the formation enthalpy of each configuration referenced to the E configuration energy, we have drawn the transition diagram showed in figure 4.4. It worth noting that this diagram summarizes all our results. We can extract from it the most relevant saddle points our simulation has passed and therefore, the possible diffusion mechanisms of the self-interstitial. In addition, we have included the transition associated to the T configuration although it has only happened in the first step.

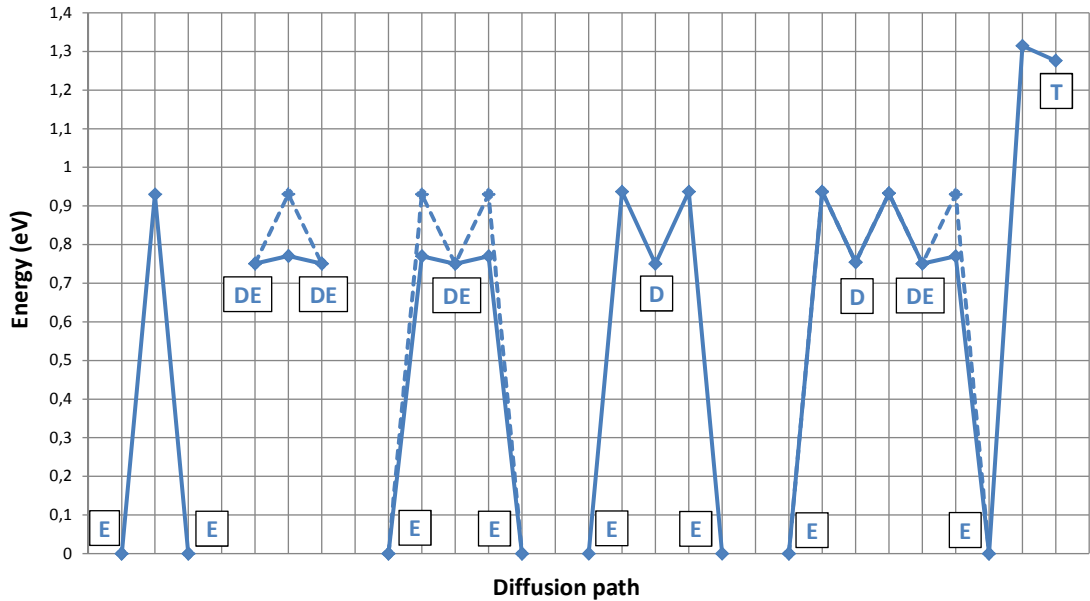


Figure 4.4: Diffusion mechanisms for the self-interstitial using the SW potential (Continuous lines: events passing by the lowest saddle points, Dot lines: events passing by the highest saddle points).

4.2.4 Comparison with other authors

We can find in the literature a large list of authors who have worked with the SW potential to determine the formation enthalpy of each self-interstitial configuration, but the variability of their results is quite notorious [88]. Here we are going to compare the achieved results with other references which have used ART or kART with the SW potential to carry out similar simulations. In this sense, it should be emphasised the contributions of P. Ganster *et al.* [89] and M. Trochet *et al.* [73].

The first one used ART to determine the self-interstitial diffusion mechanism whereas the second one used kART instead. As it can be seen in table 4.3, our values are coincident with those of Ganster *et al.* in terms of formation enthalpies, disagreeing with the M. Trochet *et al.* contribution, whose D and E energies calculations are the same but inverted with our results. As a result, when drawing the diffusion path, this author considered as a reference the D instead of the E configuration, contrary to what we have found, and he also found a low energy barrier between two E configurations. The diffusion path diagram represented in figure 4.5 illustrates this idea, showing also the equivalence of our work with the diffusion mechanism predicted by Ganster *et al.*, except for the lower energy barrier we have determined between E and DE configurations.

Reference	Formation enthalpy (eV)			
	T	D	DE	E
P. Ganster <i>et al.</i> (ART) [89]	-	4.44	-	3.68
M. Trochet <i>et al.</i> (kART) [73]	-	3.685	-	4.435
Our work	4.955	4.432	4.429	3.679

Table 4.3: Formation enthalpies for several Si self-interstitial configurations as found in the literature compared with our work using the SW potential.

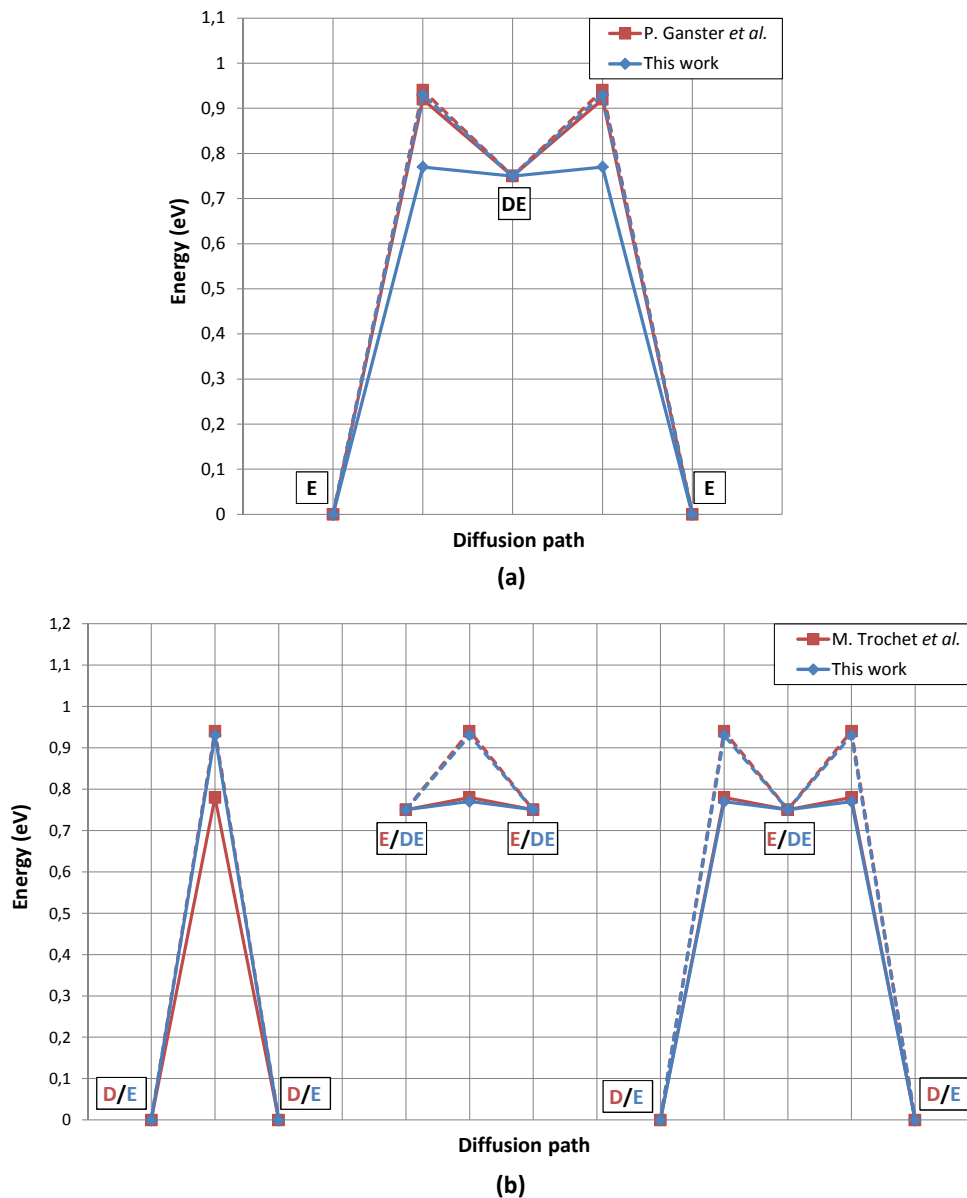


Figure 4.5: Diffusion mechanisms for the self-interstitial of this work compared with P. Ganster *et al.* [89] (a) and M. Trochet *et al.* [73] (b) results.

4.3 Tersoff potential

The structure of this section is similar to what has been already mentioned with the SW potential, but considering certain relevant aspects which appears when using the Tersoff 3 potential instead. Similarly, we are going to present the obtained results and contrast them with values from other authors after show how we have set the simulations.

4.3.1 Simulation methodology

The initial cell we have used is equivalent that described in section 4.2.1. However, here we have actually worked with several cells, each one containing self-interstitials not only with T configuration but also with D and E configurations. Additionally, we have not run a single 1000 steps kART simulation, but instead more than 20 simulations have been launched. To do so, we have used the Tersoff potential (ENERGY_CALC = LAM) and 50 number of searches per topology (SEARCH_FREQUENCY = 50) in all simulations, but we have modified MIN_SIG_BARRIER and INCREMENT_SIZE to diverse values in each one, among other parameters. The temperature has also been changed from 500 K to 1000 K for a particular simulation.

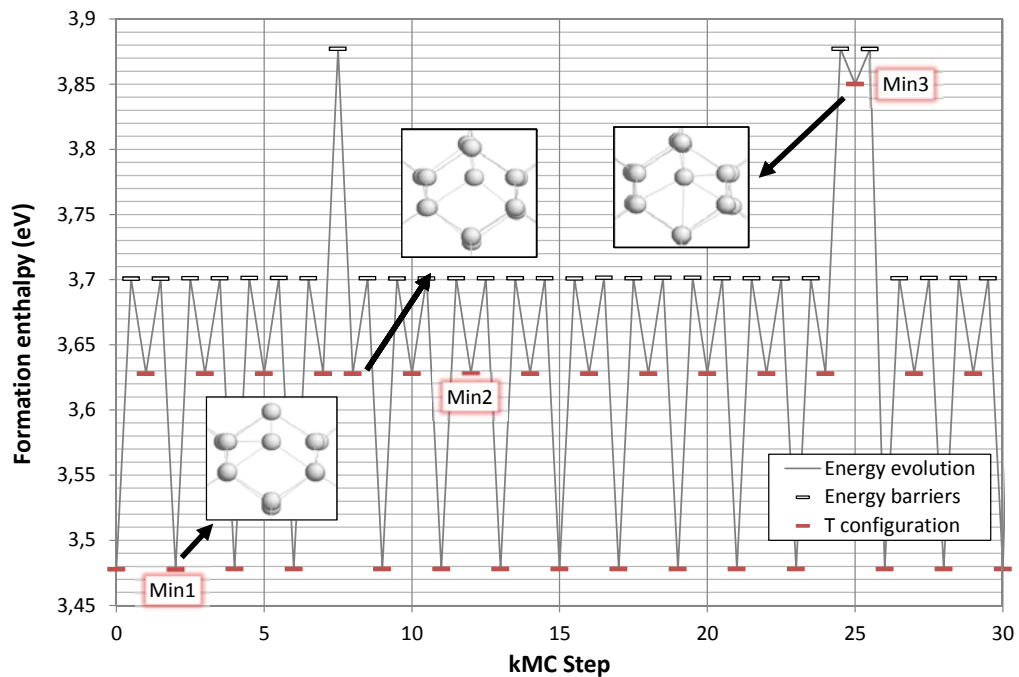


Figure 4.6: First 30 steps of the energy evolution of the Tersoff based simulation at 500 K beginning from a T configuration, labelling the visited minima.

On the other hand, the characterization of the self-interstitial configurations has not been changed. We have classified every kMC step according to its number of DAs and ESs. However, the Tersoff potential based simulations do not have a unique energy minimum per self-interstitial configuration as it happened with the SW potential. In fact, each configuration presents different energy minima, some more symmetrical than others. This idea can be seen in figure 4.6, which represents the first 30 kMC steps of a simulation in which it has been set the same parameter values as what we have previously explained with the SW potential (only modifying the number of searches), starting again from the T configuration (labelled as “Min1”).

4.3.2 Tersoff singularities

The scheme showed in figure 4.6 remains unaltered over the entire simulation, i.e., the system has not been capable of passing through D, DE or E configurations with the same simulation conditions as the SW ones, it only evolves over T configurations with slight symmetrical differences. Hence, it is obvious that we should focus on modifying certain parameters in order to see interesting transitions. In any case, firstly we are going to analyse in detail what happened. Due to the fact that kART let the user direct access to the possible events for each kMC step, we can figure out why the transitions to other configurations did not take place, and even draw a flowchart of energy minima whose transitions are interconnected.

4.3.2.1 Analysis of possible barriers

Computing all the possible events from the lowest energy minimum from figure 4.6 (“Min1”) we can make a graphic which summarizes all the options the kMC algorithm has to run over the next step, as it is represented in figure 4.7a. However, it should be noted that some events share the initial and final energy and their energy barrier. In fact, more than 40 events are available from the initial configuration, so it means that certain barriers present more weight, in terms of frequency, than others. This idea is showed in figure 4.7b. Here it is worth remembering that the kMC calculations consider not only the energy value of the barrier to select the next step but also its frequency, although in this case the energy separation between the lowest energy barrier and the next one is too high to be in competition.

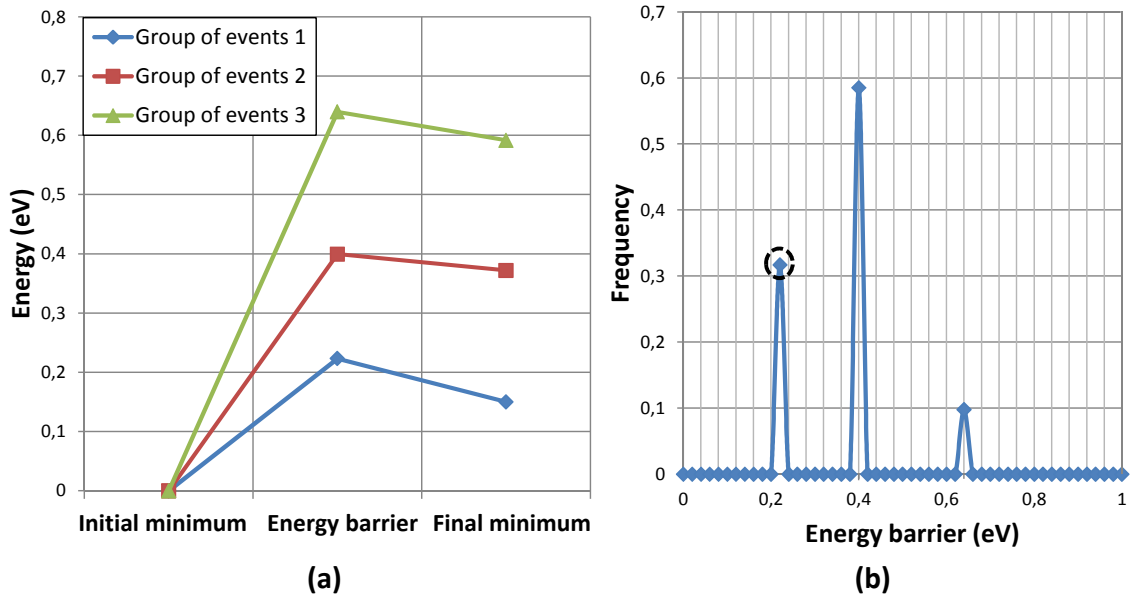


Figure 4.7: Grouping of same energy possible events from the lowest energy T configuration taking it as a reference (a) and frequency of the barriers for these events (b).

Keeping this analysis in mind, now it is easy to interpret the behaviour showed in figure 4.6. Over the vast majority of the kMC steps the lowest energy barrier (circled in the previous figure), which leads the system to “Min2”, is chosen despite the fact that it has lower frequency (0.33) than the higher energy one (0.59), which leads the system to “Min3”. As such, this barrier is selected only a few times during the simulation whereas the 0.64 eV energy barrier, which would carry the system to an upper energy minimum (“Min4” following the current nomenclature), is not chosen at any time. Furthermore, we have also analysed the possible barriers from the other visited energy minima (“Min2” and “Min3”), but the thing gets more complicated. The number of possible different events increase much and their energy barriers do likewise, as it can be seen in figure 4.8. In any case, we continue selecting the lower energy ones.

Hence, one can instantly think of modifying the `MIN_SIG_BARRIER` parameter to higher values in order to solve this issue, being able to pass over higher energy barriers and see transitions because it would add more events to the basin and these events would be executed only once. Another solution could be to modify the parameters involved in the calculation of events, such as `INCREMENT_SIZE` or `BASIN_FACTOR`, widening then the map of possible transitions which take place. Additionally, starting from a different Si self-interstitial configuration, such as D or E, might change the situation.

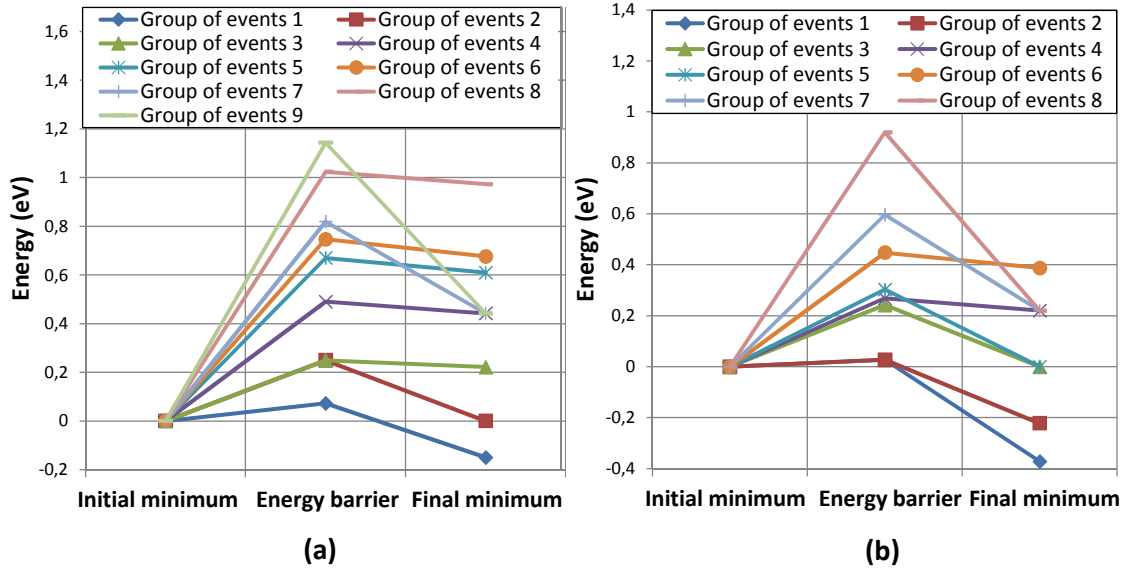


Figure 4.8: Grouping of same energy possible events from the second (a) and the third (b) lowest energy T configuration taking them as a reference.

4.3.2.2 Simulations explanation

Following the previous reasoning, in a first step we have launched simulations setting MIN_SIG_BARRIER to 0.3, 0.5, 0.7 and 0.9 eV and therefore, higher energy minima of the T configuration have been visited, as figure 4.9 shows for the 0.7 eV attempt. In addition, although we have visited for the first time a energy minimum with 2 DAs and 1 ES, actually it is not exactly a D configuration. In this sense, it also seems to have several energy minima for a D configuration. Consequently, by modifying this parameter we are capable of visiting new energy minima higher in energy, but any of them is one of the known configurations.

Other approach is to set INCREMENT_SIZE to higher values to enable the system to be pushed outside the harmonic basin with larger atom displacements so that the map of events is increased. Figure 4.10a illustrates this concept. In any case, whereas the available events from the initial T configuration is increased, we continue selecting the low energy ones which we have already visited, even when we combine this parameter with the previous one.

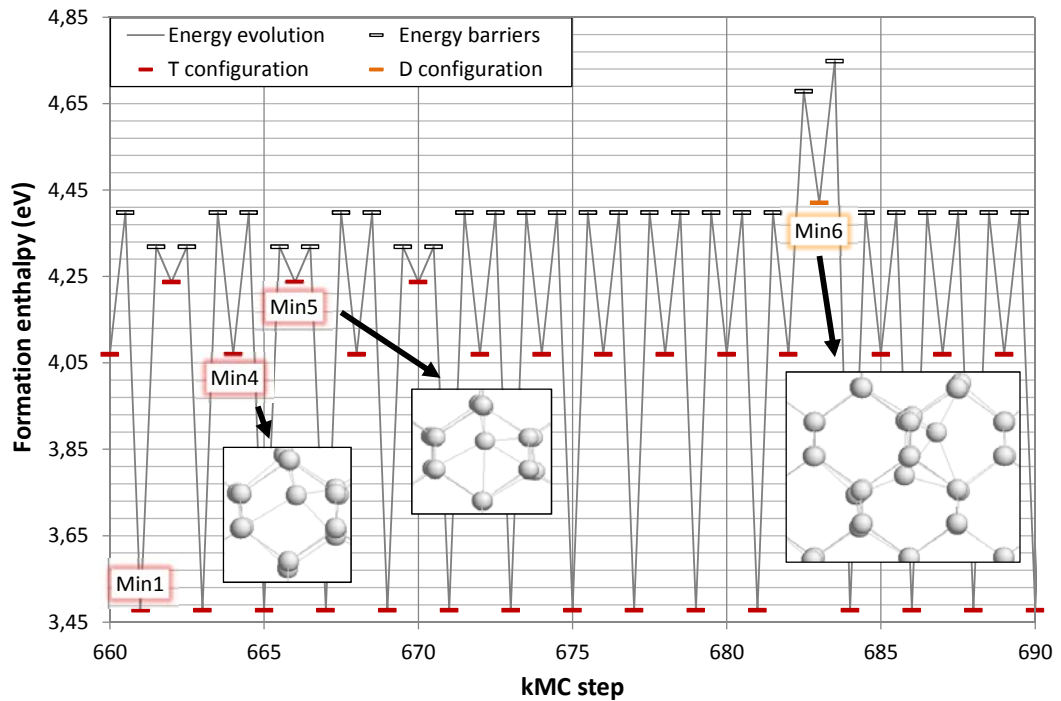


Figure 4.9: Steps 660 through 690 of the energy evolution of Tersoff based simulation with MIN_SIG_BARRIER to 0.7 eV at 500 K beginning from a T configuration and labelling all the encountered configurations.

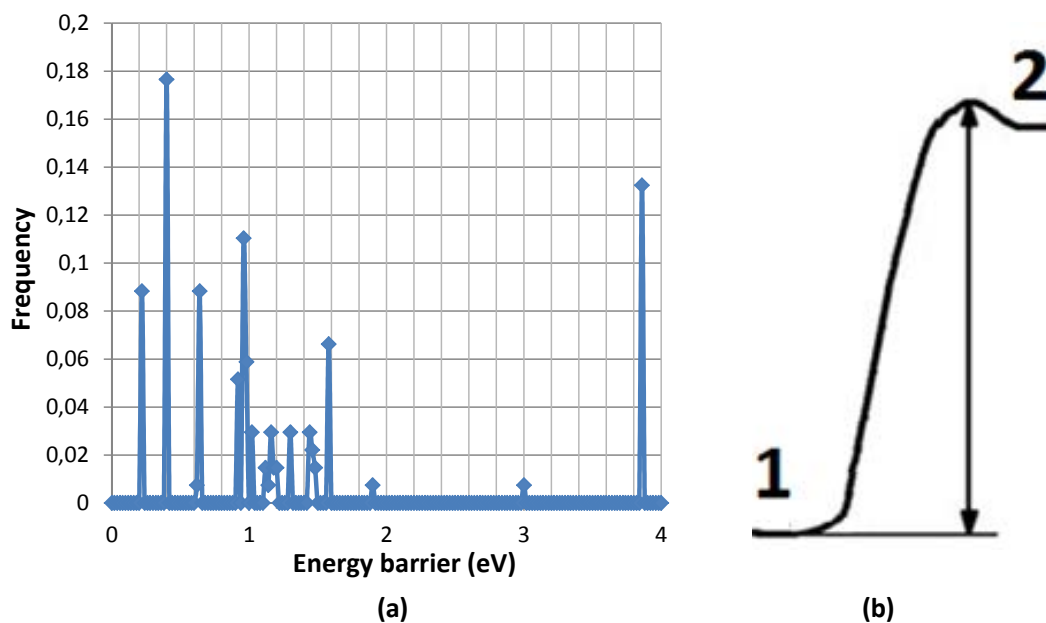


Figure 4.10: Frequency of the energy barriers for the possible events from T configuration with INCREMENT_SIZE set to 0.3 Å at 500 K (a) and an example of our problem with low returning energy barriers (b).

It should be stressed that we are not facing a flicker problem because we are able to reach higher energy states interconnected without low energy barriers by modifying these parameters. The return barriers, however, are quite low, as figure 4.10b represents. Hence, the unique remaining solution for seeing a transition beginning from a T configuration is increasing the temperature so that is what we have carried out. Nevertheless, we have to consider the drawbacks of doing this task because the possible energy minima for the same configuration are increased, making difficult the labelling of each one. As a result, this is the only point in this memory in which this parameter is changed from 500 to 1000 K.

Thus, we have run a simulation at 1000 K with $\text{MIN_SIG_BARRIER} = 0.3 \text{ eV}$ and $\text{INCREMENT_SIZE} = 0.3 \text{ \AA}$, identifying D and E configurations after starting from a T configuration, as it can be seen in figure 4.11. We visit E configuration in two steps and D in three steps, but actually we do not pass over DE configurations; the steps characterized with 3 DAs + 2 ES are E configuration slightly displaced.

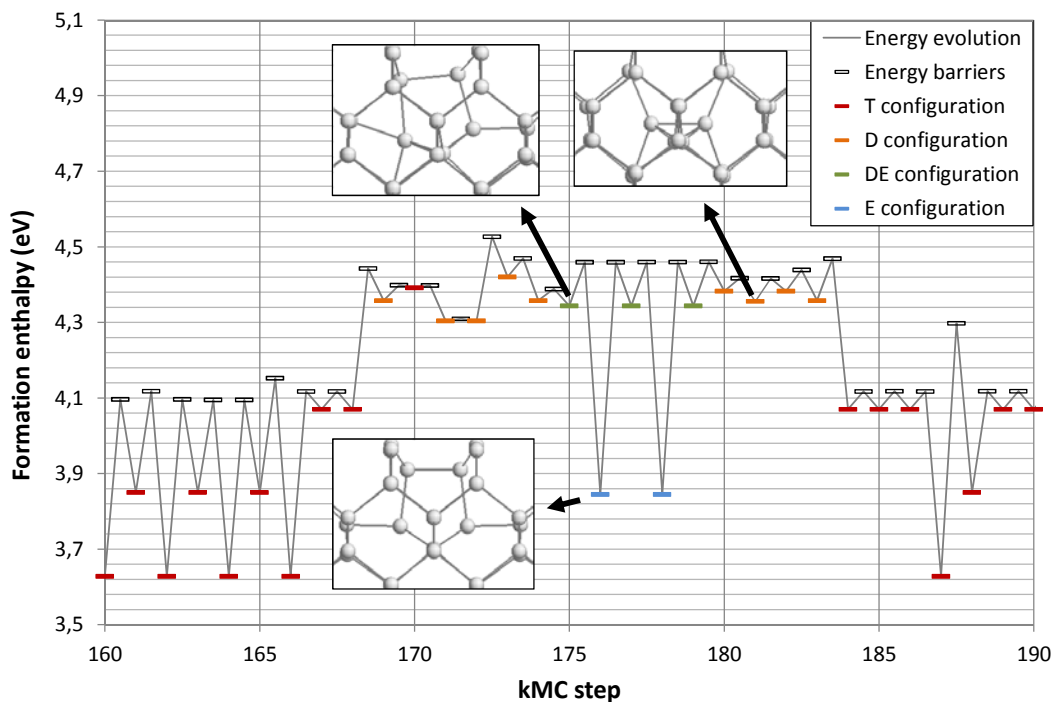


Figure 4.11: Steps 160 through 190 of the energy evolution of Tersoff based simulation with MIN_SIG_BARRIER to 0.3 eV and INCREMENT_SIZE to 0.3 \AA at 1000 K beginning from a T configuration and labelling all the encountered configurations.

Though the previous work has been necessary to go away from energy minima which have T configurations, starting directly from E or D configurations we can end up visiting a T configuration without increasing the temperature. In this sense, when we have begun from a D configuration keeping all the simulation parameters by default except for `SEARCH.FREQUENCY = 50`, we have fallen in the first steps to T configurations through an energy barrier of 0.18 eV, i.e., this configurations seems to be unstable. However, if we start from an E configuration it requires more effort: `MIN_SIG_BARRIER` must be set to 0.9 eV to fall into T configuration. This phenomenon is showed in figure 4.12.

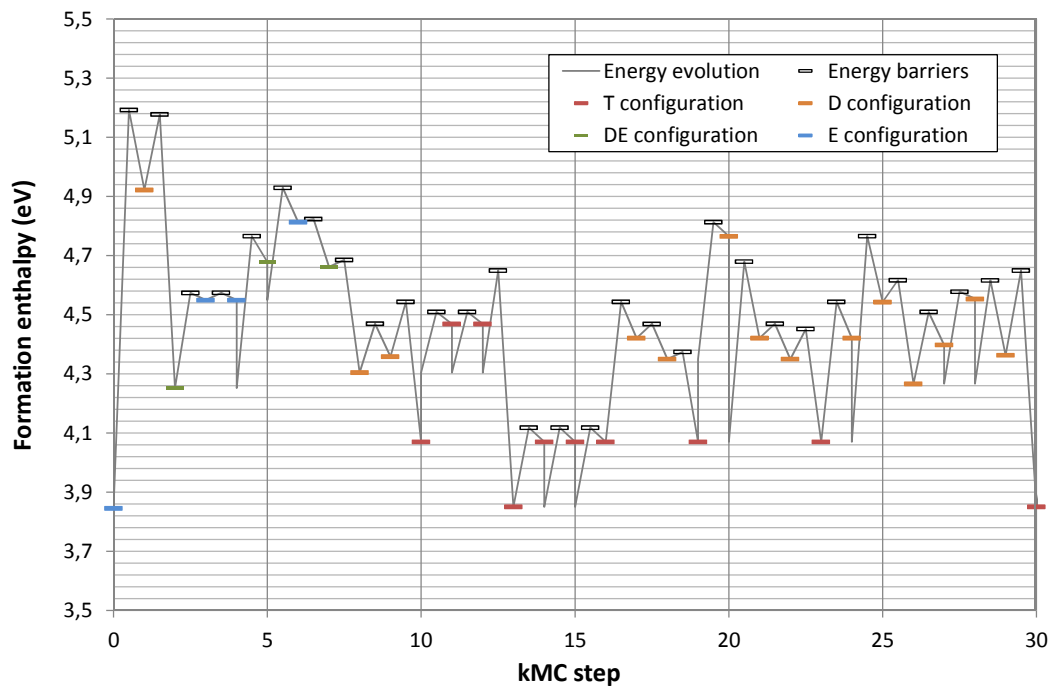


Figure 4.12: First 30 steps of the energy evolution of Tersoff based simulation with `MIN_SIG_BARRIER` to 0.9 eV at 500 K beginning from a E configuration and labelling all the encountered configurations.

4.3.2.3 Flowchart of energy minima

Once the previous analysis is done, we have also tried to extract more information about the resources kART give to us. In this sense, a C program which considers each energy minimum (either visited or not) over the simulation, has been developed. It is based on taking into account all the observed transitions of each kMC step by reading the `Energies.dat` output file, but also storing the other events which are not selected from

the `event_list_conf_x.dat` (x is the kMC step) output files in `EVLIST_DIR` folder. Hence, we have created a database with all these events, i.e., a list of elements which have three energy values: initial energy minimum, energy barrier and final energy minimum. When an event is not in the database, then it is added, being possible to process the output files from several simulations which have led to different energy minima. That occurs, for example, when we go through higher energy states by modifying `MIN_SIG_BARRIER` parameter.

Furthermore, not only have we characterized each event, but also we are able to draw all of them at the same time indicating their connections in a flowchart. Thus, we can have direct access to all the observed transitions and even the ones which have not taken place, seeing the energy pathway to follow in order to get a certain transition between two different energy minima. On the negative side, due to the fact that the list of generated events by kART for each step is so large and some of them share the initial and final energy but with different energy barrier values, one cannot make a draw based on lots of simulations output files.

One example of this affirmation is presented in figure 4.13. It only shows the visited and not visited energy minima for the simulations we have mentioned above at 500 K: 4 simulations starting from the T configuration (keeping constant `INCREMENT_SIZE` to 0.3 Å by modifying `MIN_SIG_BARRIER` from 0.3 to 0.9 eV), one simulation taking E configuration as the initial energy minimum setting default parameters, and beginning from D configuration with default values again. Additionally, we have not represented all the energy barriers which connect the same energy minima, only the lowest ones are considered. By looking more carefully to it, we can determine how difficult is the transition from a T configuration to a E configuration at 500 K. The energy path to follow in order to achieve this transition covers a lot of different energy minima whose energy values are high and have lower return energy barriers to come back to states visited previously instead of continuing this way. But what is really relevant is that they are not connected at all, so that this explains why we have not been able to see the T to E transition at this temperature.

On the other hand, one can apply the same procedure to the case in which the transition from T to E configuration happened, i.e., at 1000 K with `INCREMENT_SIZE` and `MIN_SIG_BARRIER` to 0.3 Å and 0.3 eV, respectively, obtaining the diagram showed in figure 4.14. We can see that, though difficult, the transition is likely to happen, justifying why we have witnessed it at 1000 K

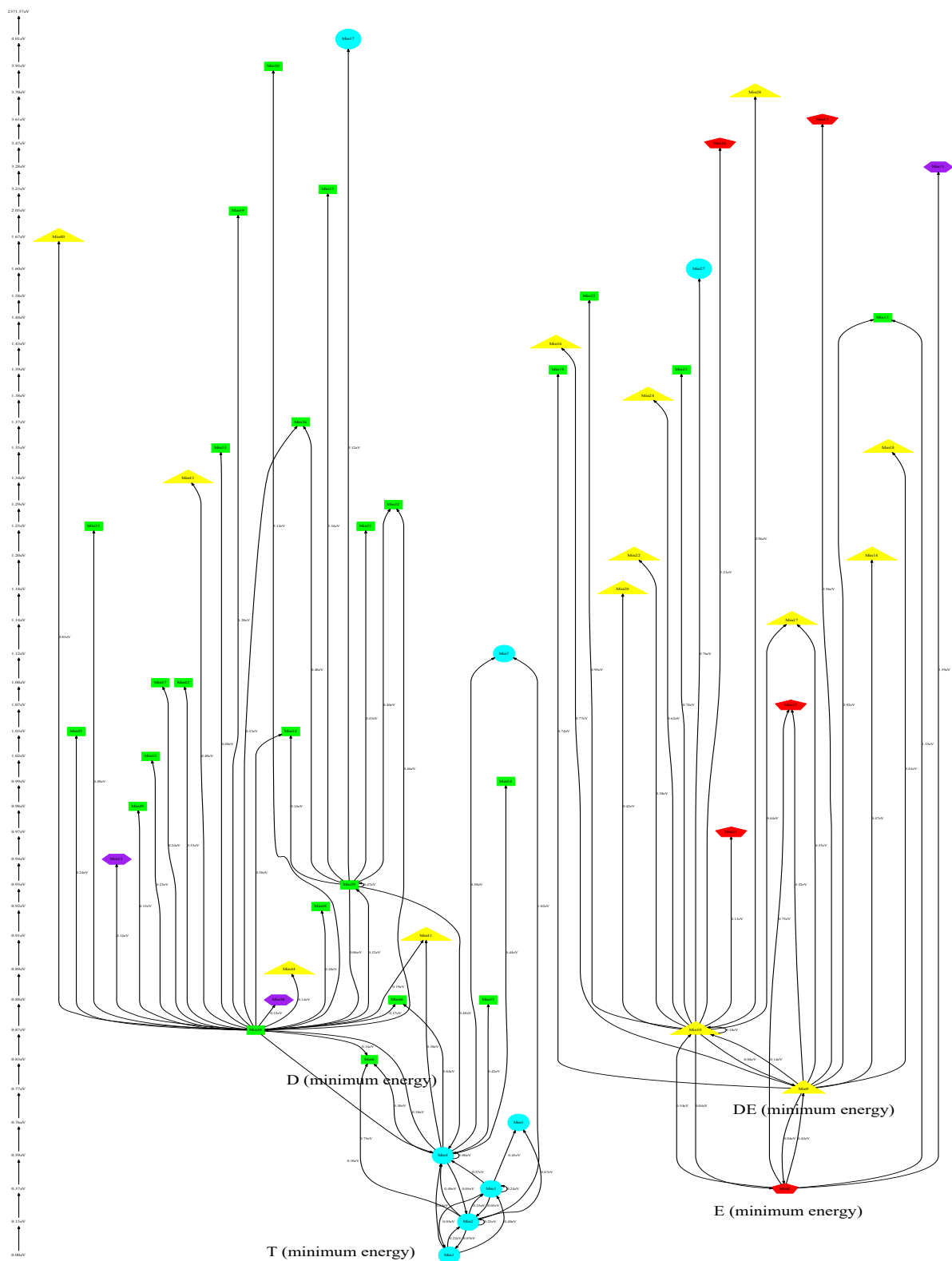


Figure 4.13: Flowchart of energy minima ranked by energy and connected with their respective minimum energy barriers at 500 K (T: cyan circle, D: green box, DE: yellow triangle, E: red pentagon, Other configurations with more than 4 DAs: purple hexagon).

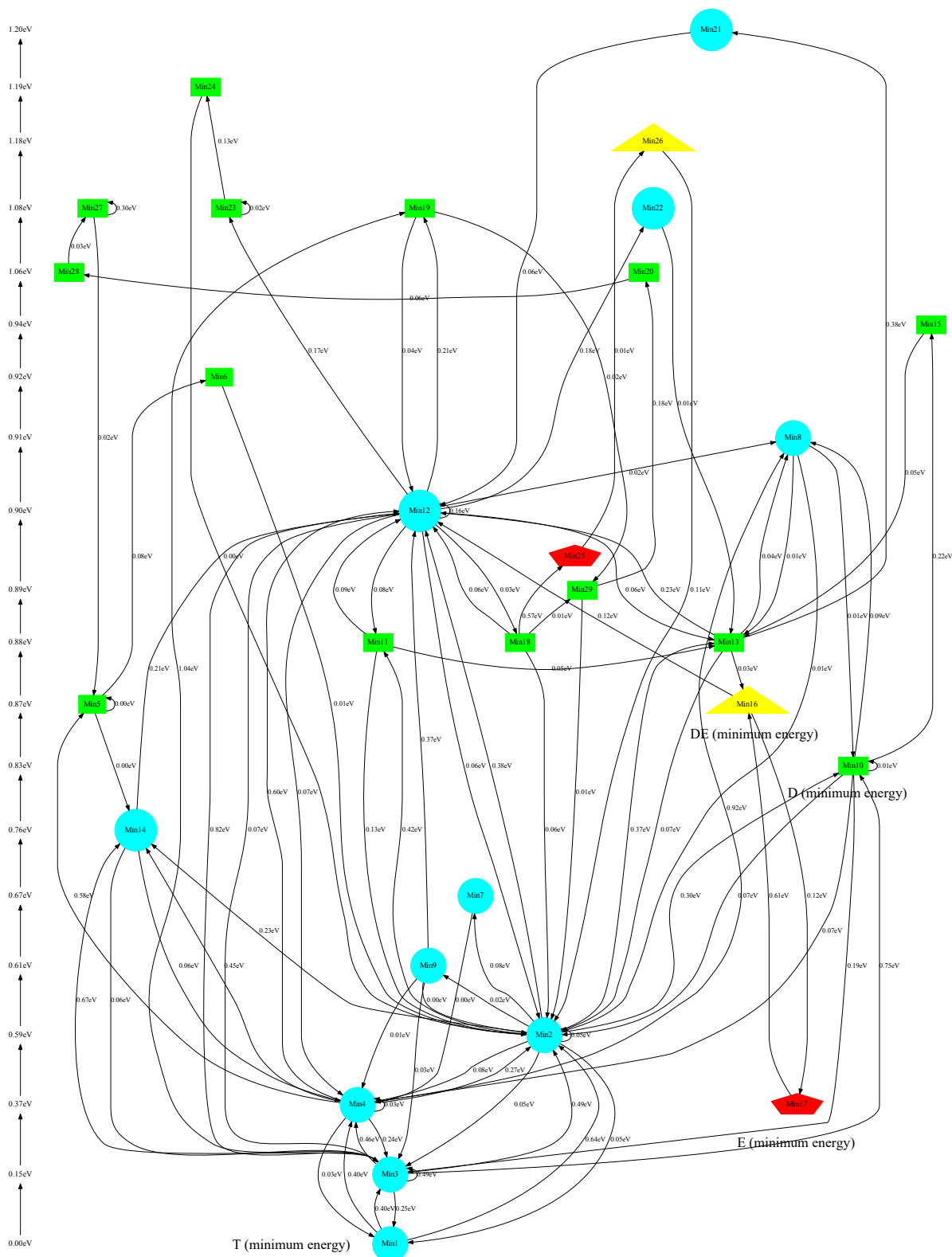


Figure 4.14: Flowchart of energy minima ranked by energy and connected with their respective minimum energy barriers at 1000 K (T: cyan circle, D: green box, DE: yellow triangle, E: red pentagon, Other configurations with more than 4 DAs: purple hexagon).

4.3.3 Formation enthalpies

We have determined the formation enthalpy of each configuration taking the lowest energy minimum which presents each of these structures. To do so, since we have launched simulations starting from T, D and E configurations and setting parameters by default except for the number of searches, i.e., keeping the same conditions than the SW potential case, we can sum up the obtained energy values as table 4.4 shows. For the case of DE, however, it has been selected the formation enthalpy of a displaced E which presents 3 DAs and 2 ESs. It should be noted that the formation enthalpy of T configuration is lower than the E one, contrary to what SW has determined.

Type of configuration	T	D	DE	E
Number of times	443	1	418	573
Formation enthalpy average (eV)	3.478	4.356	4.344	3.845
Standard deviation (eV)	4.661E-05	0	3.399E-05	5.768E-05

Table 4.4: Formation enthalpies averaged for each Si self-interstitial configurations predicted by the Tersoff potential.

4.3.4 Energy barriers

Regarding the determination of the diffusion mechanism for the Si self-interstitial, here we do not have a great amount of transitions to consider as it was the case for SW potential based simulations. In fact, we have basically taken into account the few transitions which happened at 1000 K. Noting this clarification, the predicted energy barriers between the different configurations, referenced to the T configuration formation enthalpy, are showed in the figure 4.15.

Although this diagram is not as rich as the SW one statistically speaking, one can also extract important ideas. For instance, the direct transition between T and E configuration has not been seen so a unique diffusion mechanism through D and DE configurations is likely to happen. In addition, it is demonstrated that the energy barrier is as high as the ones obtained by modifying MIN_SIG_BARRIER parameter, being the problem not a question of this kind.

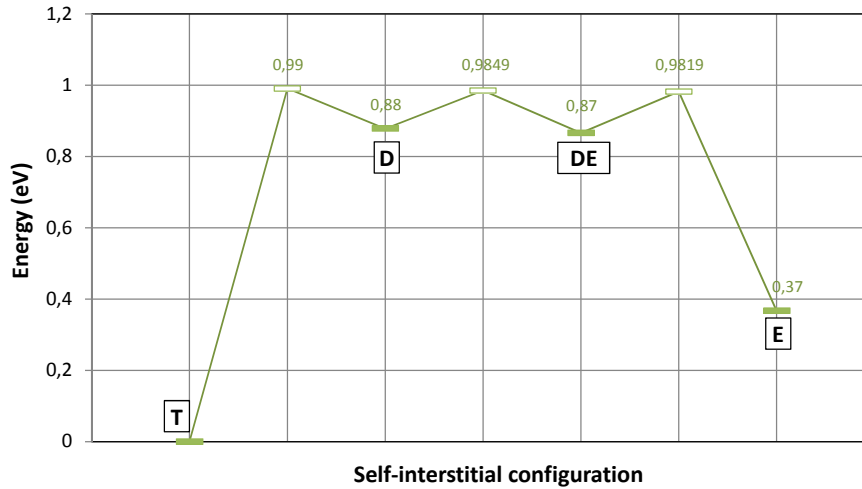


Figure 4.15: Diffusion mechanism for the Si self-interstitial using the Tersoff potential.

4.3.5 Comparison with other authors

As we have already mentioned when comparing with the SW based simulations results, one can find many works dedicated to study the formation enthalpy of Si self-interstitial configurations by using the Tersoff potential. Similarly, the achieved results are characterized for being quite different between themselves. However, there is no work in the literature which have study this issue by using Tersoff in a kART scheme. Thus, we are going to compare our formation enthalpies values with the achieved results of Luis A. Marqués *et al.* [88], who have calculated them with MD simulations based on the Tersoff potential, and with Iván Santos *et al.* [90], who have used NEBM.

One can see the formation enthalpy we have determined for T agrees with the results of both authors in the table 4.5. For the case of the D and DE configurations, MD simulations seem to overestimate it according to our work, although it should be remembered that our DE configuration was not exactly what it should be because it was a distorted E configuration. Lastly, the predicted formation enthalpy for E configuration is also not distanced from the reference values, maybe closer to the NEBM results.

Reference	Formation enthalpy (eV)			
	T	D	DE	E
Luis A. Marqués <i>et al.</i> (MD with Tersoff) [88]	3.45	4.7	5.08	3.94
Iván Santos <i>et al.</i> (NEBM) [90]	3.46	4.39	-	3.83
Our work	3.48	4.36	(4.34)	3.85

Table 4.5: Formation enthalpies for several Si self-interstitial configurations as found in the literature compared with our work using the Tersoff potential.

To compare the diffusion path, figure 4.16 represents the energy barriers obtained by NEBM from T to D configuration and to go away from E configuration. In addition, it is also showed the formation enthalpies of the MD approach, but as the author has not determined the energy barriers (he only estimated them) we have indicated them with the same values as the enthalpies. Contrasting our energy barriers with the ones obtained by NEBM, it is clear that we have found lower energy paths, being the energy barriers in the same tendency. It should be noted that kART does not need to presuppose a path unlike NEBM.

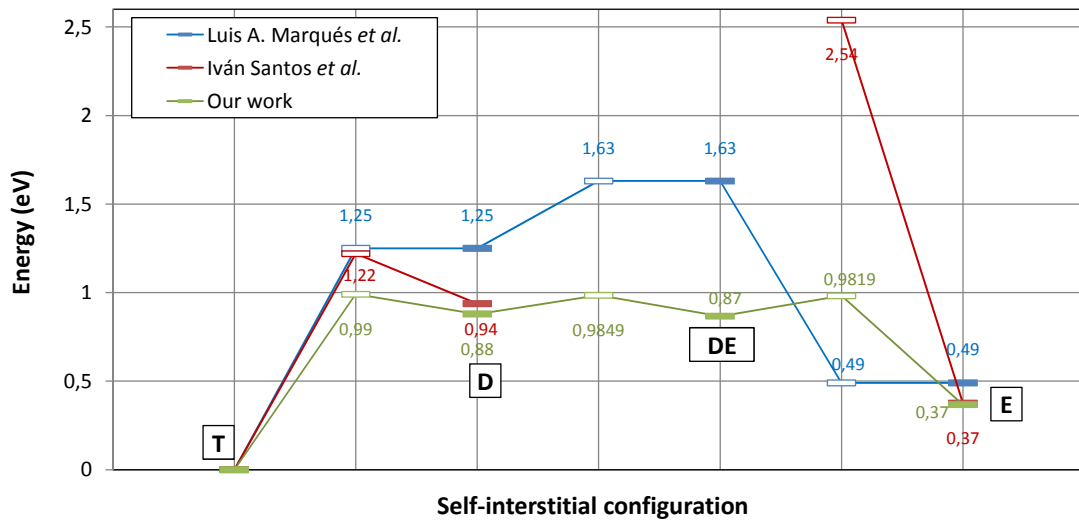


Figure 4.16: Diffusion mechanisms for the Si self-interstitial of this work compared with Luis A. Marqués *et al.* [88] and Iván Santos *et al.* [90] results.

4.4 Conclusions

In this chapter we have studied the Si self-interstitial configurations by using two interatomic potentials: the SW and the Tersoff 3. The formation enthalpy of each configuration and the energy barriers of the possible transitions between them have been determined in both cases.


SW based simulation have verified that E is the lowest energy configuration with 3.679 eV, in accordance with the results obtained by other authors with this potential. Additionally, the D and DE configurations seem to be lower than the T configuration in energy so they are more implicated than T in the diffusion mechanisms. In this sense, we have found five diffusion paths which involve energy barriers higher than 0.7 eV in order to go away from the E configuration. We have also seen cases in which there are two possible barriers for the same transition, as it is the case for the DE to DE transition and DE to E transition. In any case, all the transitions observed during the simulations have been direct, i.e., there has not been different energy minima for the same configuration involved. Other authors have also studied this idea with ART and kART simulations, being our results similar to their ones.

On the other hand, using Tersoff has lead us to different results. Firstly the lowest formation enthalpy achieved, 3.478 eV, has been for the T configuration, being the E configuration 0.367 eV higher in energy. However, we have had to launch several simulations in order to see transitions due to the fact that the energy landscape given by Tersoff is rougher than the predicted by SW. In this sense, we have encountered different energy minima for the same configuration, which means that the landscape presents small energy wells to be visited if one wants to see different transitions. Comparing the obtained energy barriers with the ones achieved by NEBM, kART gives lower values for all the studied transitions.

Chapter 5

Energy properties of small interstitials clusters

*You can understand perfectly,
if you give your mind to it.
~ William Thomson (Lord Kelvin) ~*

 ONCE WE HAVE CHECKED our simulation technique, we are going to characterize small clusters of interstitials. This chapter aims to calculate the formation enthalpies of each configuration determining also the energy barriers associated with the the transitions between them. In addition, their topologies will be also indicated.

To do so, many simulations have been launched at 0 K and 0 GPa for each cluster size, following the approach explained in chapter 3. Since the method we have carried out is quite similar for all sizes, all sections share a common structure. Firstly, we outline the computational cost the simulations have involved, taking into account all the attempts we have launched in order to see the interested transitions. Lately we detail the simulation conditions for which the transitions have taken place, presenting the energy barrier diagram between the visited configurations (chainlike configuration, compact configuration or {111} configuration, mainly).

It should be noted that the bigger the cluster size is, the more efforts are necessary to see transitions. The stability of certain configurations makes going away from them a difficult task. In addition, the atom rearrangement required to change the topology is more complex with higher energy barrier, being an infrequent event of kART.

5.1 Di-interstitial

This part is focused on analysing the energetic aspects of chainlike and compact configurations for I_2 . After giving an insight on how we have got the transitions, we are going to analyse the barriers of certain states implied. Eventually the energy barrier diagram is shown.

5.1.1 Summary of simulations

We have run 8 simulations starting from a compact configuration and 9 simulations starting from a chainlike configuration, consuming almost 900 h of CPU time as it can be seen in table 5.1. However, we have not always seen transitions between compact and chainlike configurations, only the 35.29 % of the simulations have presented them.

Number of simulations	Total CPU time (h)	Success rate (%)	Total simulated time (ns)
17	854.57	35.29 %	546.68

Table 5.1: Summary of simulations which have been run for di-interstitial.

The setting of the minimum barrier for the basin and the overall scale for the increment moves in activation has not been a problem. One can see transitions between compact and chainlike configurations by setting `MIN_SIG_BARRIER` to 0.3 eV and `INCREMENT_SIZE` to 0.3 Å at 500 K. However, after setting this values, in the 0.3 % of kMC steps atoms were not grouped in a unique defect. That means that our system has not evolved to a known configuration but instead minima have appeared whose DAs are isolated from the others.

Moreover, we have also made several attempts at 1000 K, but the 55 % of steps consist of new configurations more disordered than those at 500 K and the number of non grouped steps increase to 25 %. In this sense, the data we can extract from them are not relevant and they are not included here.

5.1.2 Analysis of states and possible barriers

Considering the times each energy minimum has appeared during our simulations and its implication in transitions, we have drawn the graph which is represented in figure 5.1.

As it can be seen, Co2 is the most frequent minimum for compact configurations while Ch1 and Ch4 are the ones for chainlike configurations. In any case, what it should be emphasized here are the states implied in transitions due to the fact that they enable us to change from one configuration to another. In this sense, Co8 is the energy minimum which has been identified when a compact to chainlike transition takes place. On the other hand, Ch4, Ch7 and Ch9 are the ones visited when a chainlike to compact transition occurs. Both Co8 and Ch4 present an important weight so one can wonder what are the possible events starting from these minima.

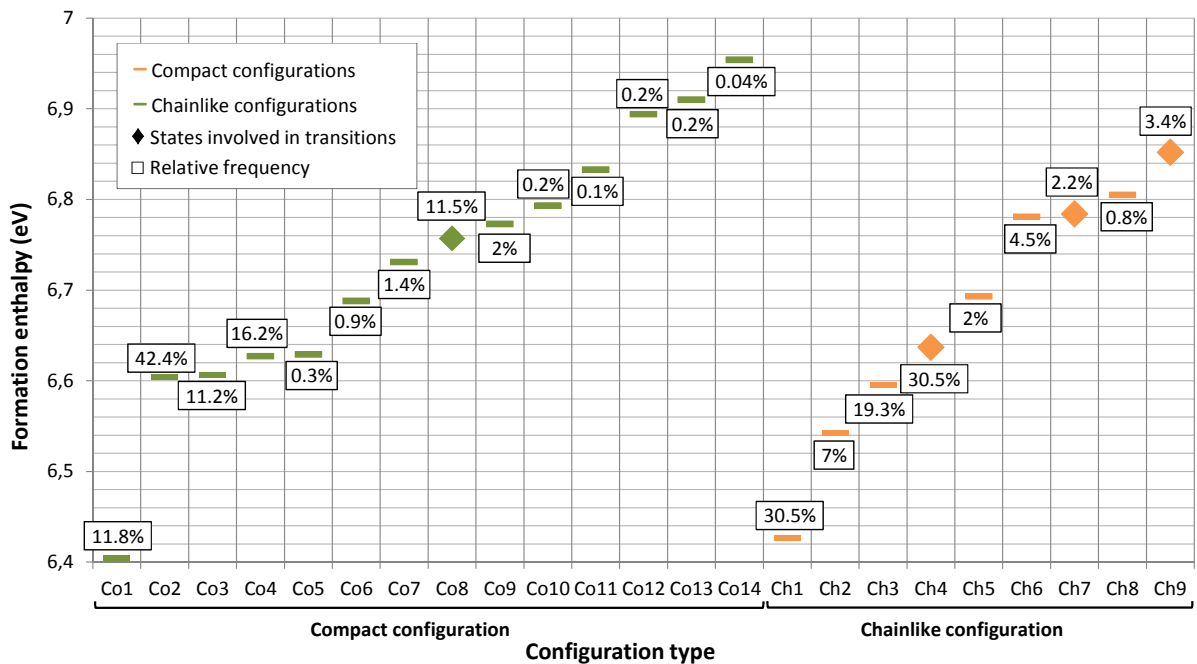


Figure 5.1: Formation enthalpies and relative frequency of energy minima for each I_2 configurations and their role in transitions.

Since kART let us access to information of the non-executed events, such as their coordinates and energy values, we can map the possible barriers for the previous minima indicating their final configuration, as it can be seen in figure 5.2. We have only analysed the final configurations of low energy barriers associated events because they are the ones which can be selected more frequently. Thus, this analysis tells us why these minima are the ones involved in transitions. For all of them the energy barrier related to the transition is either the lowest energy barrier available (the case of Co8 and Ch7) or at least is close to the lowest one (the case of Ch4 and Ch9). Then, the kMC algorithm gives more weight to these events and they can be selected.

It should be taken into account that the map of barriers for the rest of energy minima presented in figure 5.1 with rectangles lead the system to minima with the same configuration. Furthermore, the map of barriers analysed here might change from one kMC step to another, even if the initial minimum is the same. Hence, the barrier maps showed in figure 3 are not always the case for these minima and sometimes transitions are not possible.

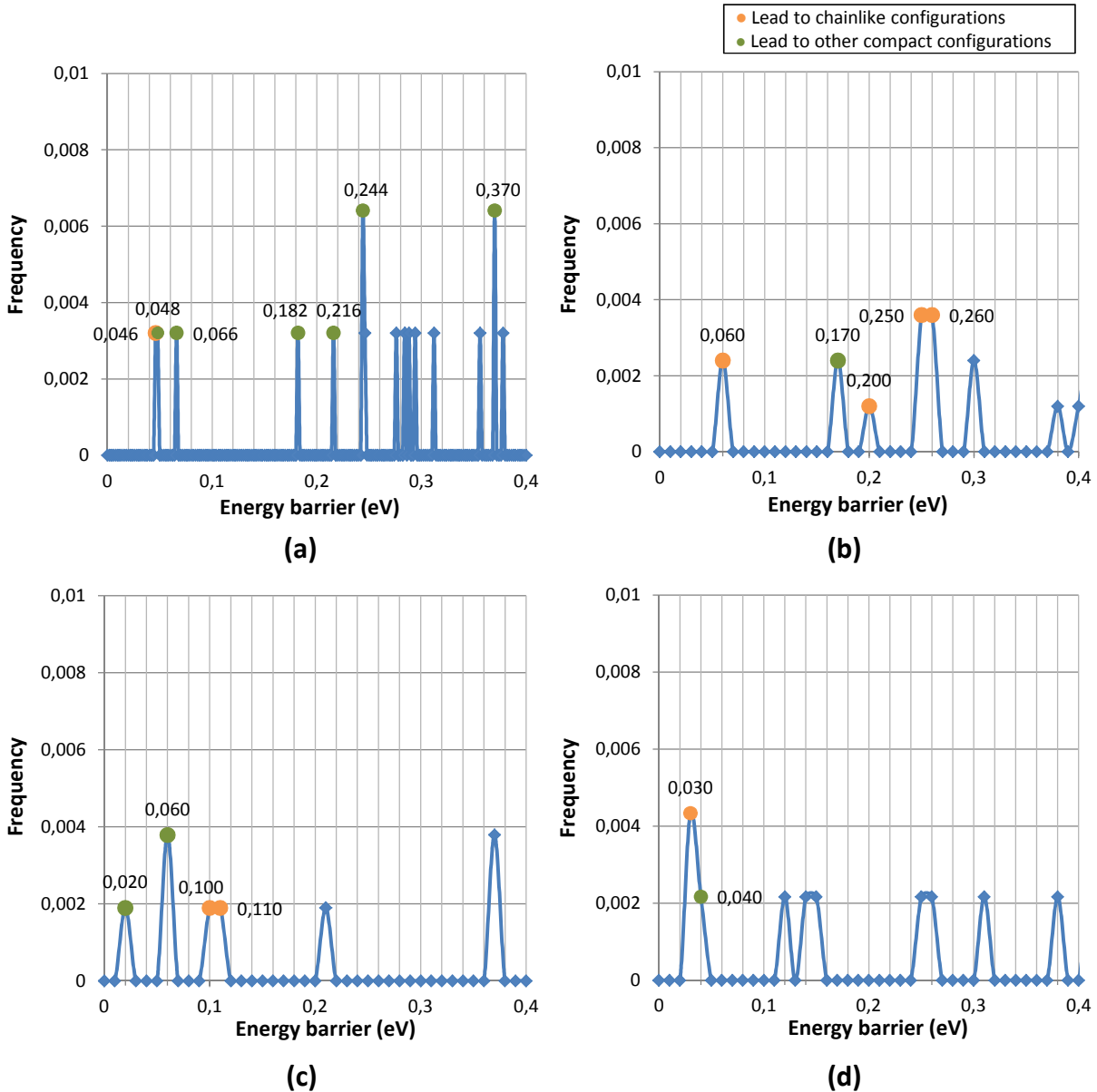


Figure 5.2: Frequency of the energy barriers for the possible events from Ch8 (a), Co4 (b), Co7 (c) and Co9 (d) I_2 energy minima and final states associated.

5.1.3 Formation enthalpies and energy barriers

We have determined that compact configuration is less energetic than chainlike configuration, speaking always in terms of the lowest energy minimum of each one. This idea is showed in figure 5.3. Moreover, considering all the transitions which have occurred during our simulations, we have seen 23 transitions from compact to chainlike configurations and 27 transitions from chainlike to compact configurations. All of them are represented as rectangles in the already mentioned figure. The lowest energy barrier (6.803 eV), however, is linked to the formation enthalpy of each configuration.

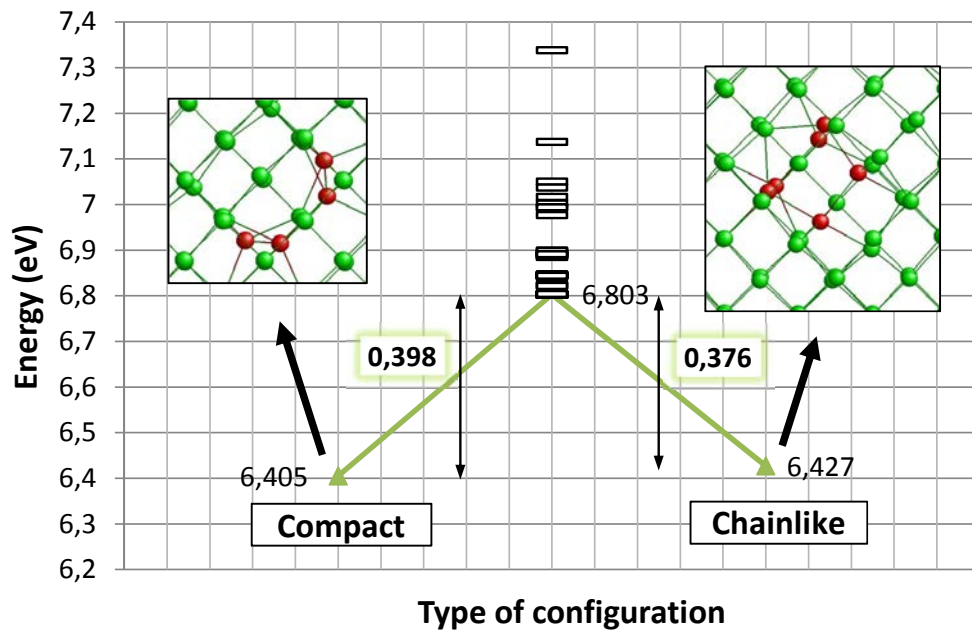


Figure 5.3: Transition path between compact and chainlike I_2 configurations indicating the formation enthalpies, topologies in $[-1\ 0\ 0]$ plane (DAs in red, bulk atoms in green) and all energy barriers found.

Consequently, the most stable configuration we have determined is compact, but it presents low energy difference with the chainlike formation enthalpy (22 meV). The energy barrier to go from chainlike to compact configuration and vice versa is relatively small, lower than 0.4 eV. It would imply that transitions between both structures are likely to happen without needing high temperature values or long periods of time. In any case, it should be noted that many energy barriers whose values are different have been found for the same transition. Hence, more than one energy path can be chosen in order to go from one configuration to the other.

The lowest energy barrier is smaller than the one calculated by Santos *et al.* [90] based on NEBM. Here we have determined a barrier of 0.398 eV from compact to chainlike configuration, referred to the compact one, while he calculated an energy barrier of 2.2 eV. Hence, NEBM seems to overestimate the energy barrier. In any case, the notorious difference between both values might be due to the nature of the techniques, as we have already explained in section 2.2.2.2.

It should be remembered that the topology of these energy minima as well as other frequent ones is represented in all space directions in the appendix A. Additionally, the figure 5.4 shows only the DAs of the lowest energy configurations. Compact defects presents 4 DAs and 2 ESs whereas chainlike defects have 6 DAs and 4 ESs. Since they are defects whose extension is not so long, the diffusion mechanism is likely by rearranging slightly the atoms. Hence, it is logical that the energy barrier was not high.

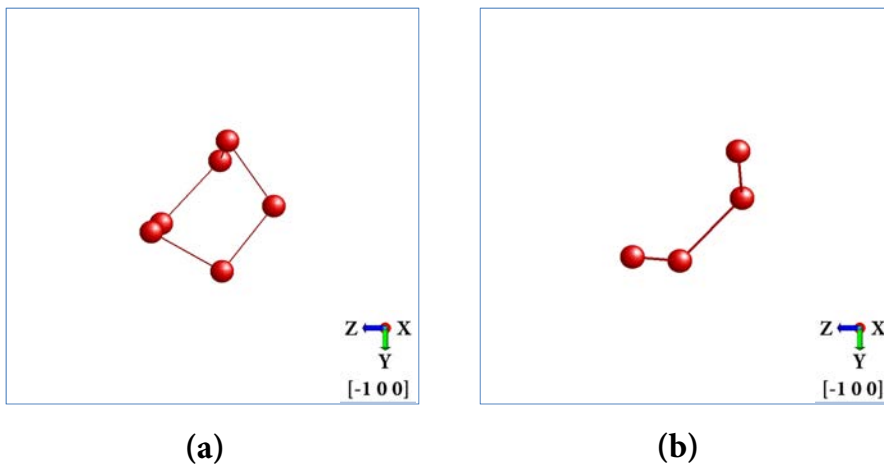


Figure 5.4: DAs of chainlike (a) and compact (b) I_2 configurations in $[-1\ 0\ 0]$ plane.

5.1.4 Transition mechanism

Developing the previous idea in depth, the figure 5.5 shows the atoms evolution when a transition from the chainlike to the compact configuration takes place. To do so, we have labelled each atom with an identifier, letting us see which of them move during the transition. It should be taken into account that each picture refers to a different energy minimum, without representing a saddle point. In addition, the first one represents the lowest energy minimum for a chainlike configuration while the last one represents the lowest energy minimum for a compact configuration.

The first step consists of a perfect chainlike configuration formed by 6 DAs. Lately, atoms 1 and 6 modify its position making atom 7 move from its bulk coordinates. Going a step forward we can see that atoms 5 and 7 come to a perfect cell position while the rest remains almost unaltered. The next two steps are quite similar to the compact configuration: atom 4 goes to the left and atom 5 goes to the right, whose position is occupied by atom 6. In addition, atoms 1 and 7 end up also in perfect cell coordinates. The compact structure is almost formed at this point. Finally, the last three steps show the movements of atoms 2, 3, 4, and 5, the ones which form the symmetrical compact defect.

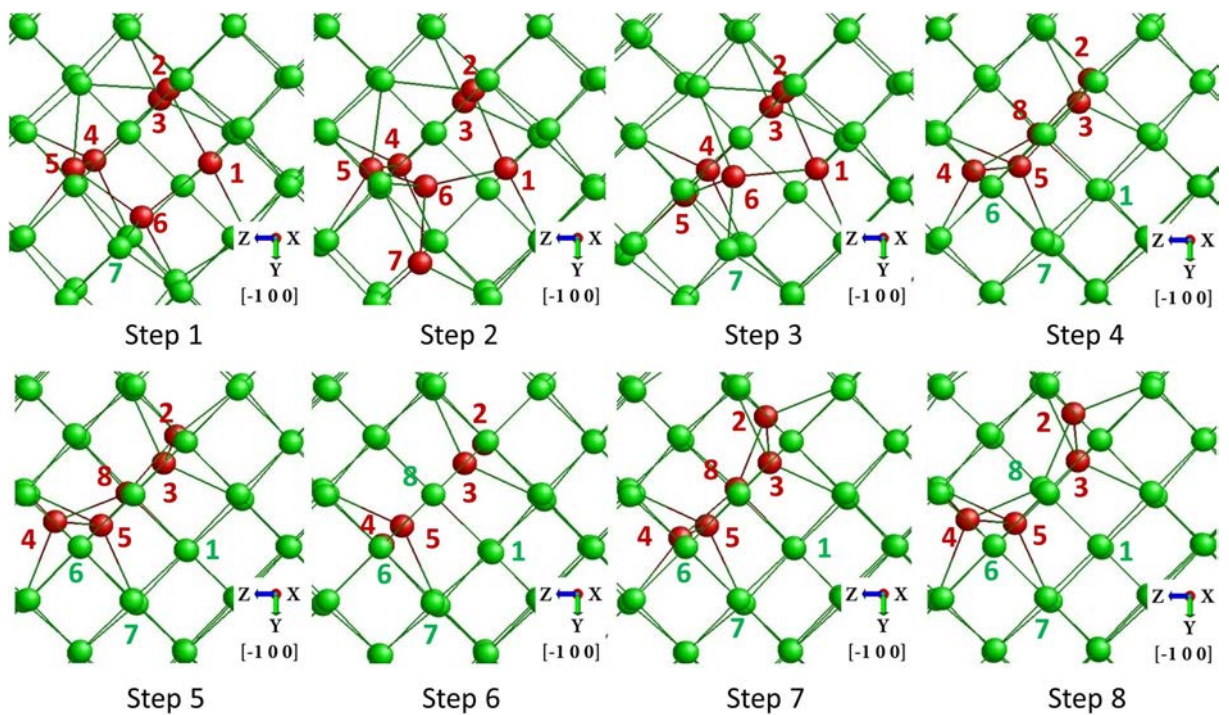


Figure 5.5: Chainlike to compact I_2 configuration evolution in $[-1\ 0\ 0]$ plane over eight kMC steps.

5.2 Tri-interstitial

We are going to continue our study with the analysis of compact, chainlike and $\{111\}$ configurations for I_3 . Following the same structure as before, an overview of the simulations we have run is presented and the formation enthalpies and energy barriers are indicated.

5.2.1 Summary of simulations

Here we have run 6 simulations starting from a compact configuration, 8 simulations beginning from a chainlike defect and 6 simulations starting from a $\{111\}$ configuration. Computing all of them the CPU time is close to 1000 h, but we have only seen transitions in 50.00 % of all attempts, as table 5.2 indicates.

Number of simulations	Total CPU time (h)	Success rate (%)	Total simulated time (ms)
20	1009.40	50.00 %	348.22

Table 5.2: Summary of simulations which have been run for tri-interstitial.

In this sense, setting MIN_SIG_BARRIER to 0.3 eV is not enough to see transitions between compact and chainlike configurations. In case the system starts from the first one, the minimum barrier of the basin should be set to 0.6 eV. If the system starts from the second one, however, the barrier needs to be set to 1.5 eV. The $\{111\}$ configuration does not present this issue because one can evolve to other known configuration with a 0.1 eV barrier. In any case, after setting this values there are no kMC steps in which the atoms are not grouped in a unique defect, as it happened in the previous section.

5.2.2 Formation enthalpies and energy barriers

According to our results, which are represented in figure 5.6, we can point out that the formation enthalpy of compact configuration is 0.45 eV higher that of the chainlike one. Hence, it does not follow the tendency of I_2 , where the chainlike configuration had lower energy. In addition, there are fewer transitions in this case. The system has evolved from a compact to a chainlike configuration 5 times and only 1 in the opposite direction. The transition from the $\{111\}$ to the compact configuration has happened once, but the opposite direction presents 4 samples.

The meaning of the symbols in the figure is the same as before, but here we have added a discontinuous line which indicates that we have not seen the transition in this direction. For instance, our system has gone from a $\{111\}$ configuration to a compact one, but not vice versa. We have estimated the energy barriers for those cases.

Hence, chainlike is the most stable configuration we have determined here and the barrier to fall into it from a compact configuration is high (0.84 eV). It means that this

transition would require high temperatures or long times to take place. In any case, the statistics here are not as numerous as what we did in I_2 analysis and they also present variability. Again, several energy paths can be chosen in order to go from chainlike to compact configuration and vice versa. The configuration symmetrical in $\{111\}$ plane, however, seems to be very unstable. It has negligible energy barriers and it has not appeared again by the evolution of our simulations.

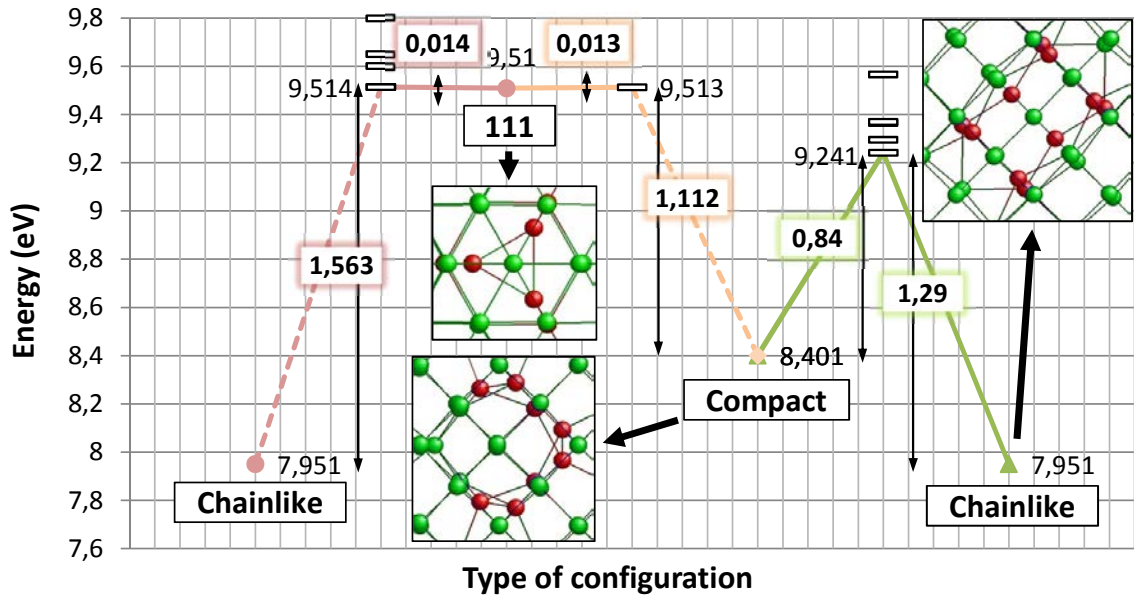


Figure 5.6: Transition path between compact, chainlike and $\{111\}$ I_3 configurations indicating the formation enthalpies, topologies in $[-1\ 0\ 0]$ plane for the first two and in $[-1\ 1\ -1]$ for the third (DAs in red, bulk atoms in green) and all energy barriers found.

Comparing these energy barriers with the work of Santos *et al.* [90] with NEBM, the barrier between compact and chainlike configuration is 3 eV below. Again, NEBM seems to overestimate this energy barrier.

It should be noted that the topology of these energy minima as well as other frequent ones are represented in the appendix A. Figure 5.7 represents the DAs of each configuration in order to see how the atoms have to move. Compact defects present 8 DAs and 5 ESs and chainlike and $\{111\}$ configurations have 10 DAs and 7 ESs. The extension of the chainlike defects along the network is large. As this configuration is more elongated than the I_2 case, it is logical that the energy barrier increases. This is going to happen every time we increase the defect size. The compact defect, however, grows in a circular way, having an atom left to complete the fourfold structure. We can also observe the high symmetry of $\{111\}$ configuration in that plane.

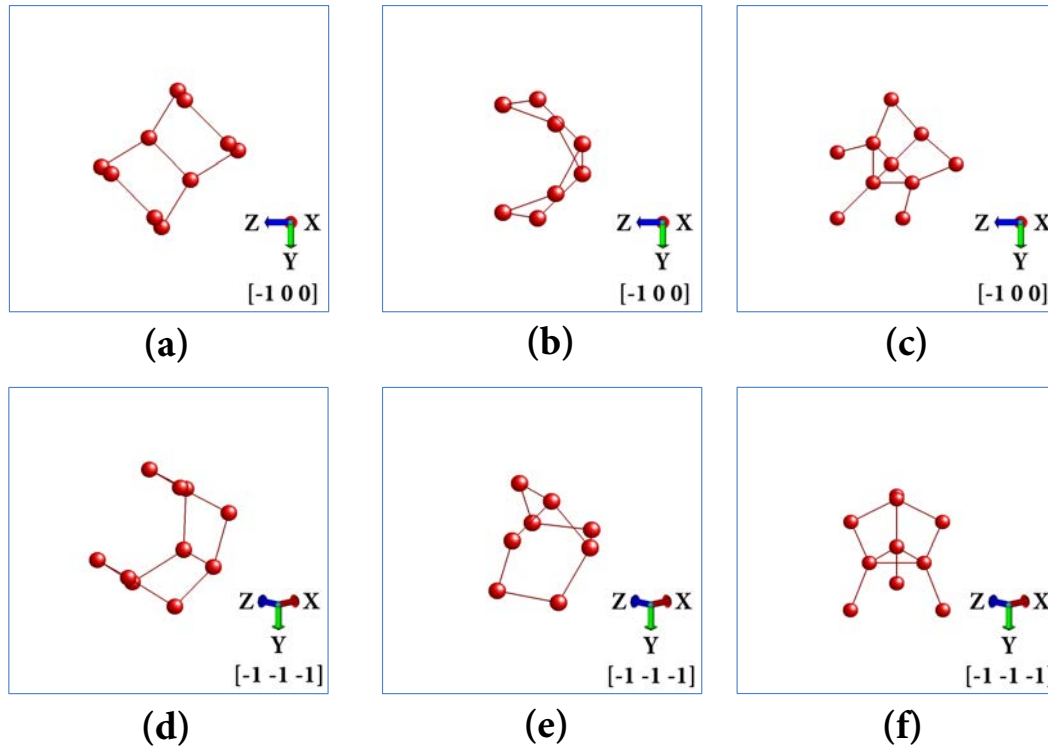


Figure 5.7: DAs of chainlike, compact and $\{111\}$ I_3 configurations in $[-1\ 0\ 0]$ plane (a, b, c) and $[-1\ -1\ -1]$ plane (d, e, f), respectively.

5.2.3 Transition mechanism

The transitions between compact and chainlike I_3 configurations are not as straight as it happened with the I_2 case. Here more than 50 kMC steps are needed to see these topologies change. Hence, we have only represented the most relevant ones in the figure 5.8. Note that labels from each plane refer to the same atoms and two orientations are represented at the same kMC step.

If we observe the evolution in $[-1\ 0\ 0]$ plane, firstly atoms 1, 2, 5 and 7 lost their compact position, displacing then atoms 8, 9 and 10, which were previously in perfect cell coordinates. In the following steps a trapezoid is formed by atoms 6, 8, 9 and 10. The last step involves a long exchange between atoms 9 and 10, although it seems to occur only in this direction because they keep close in the $[0\ 1\ 0]$ plane. In addition, atoms 3 and 5 open slightly to chainlike positions and atoms 2 and 12 fill their sites. The rest of displaced atoms move to the nearest perfect cell position.

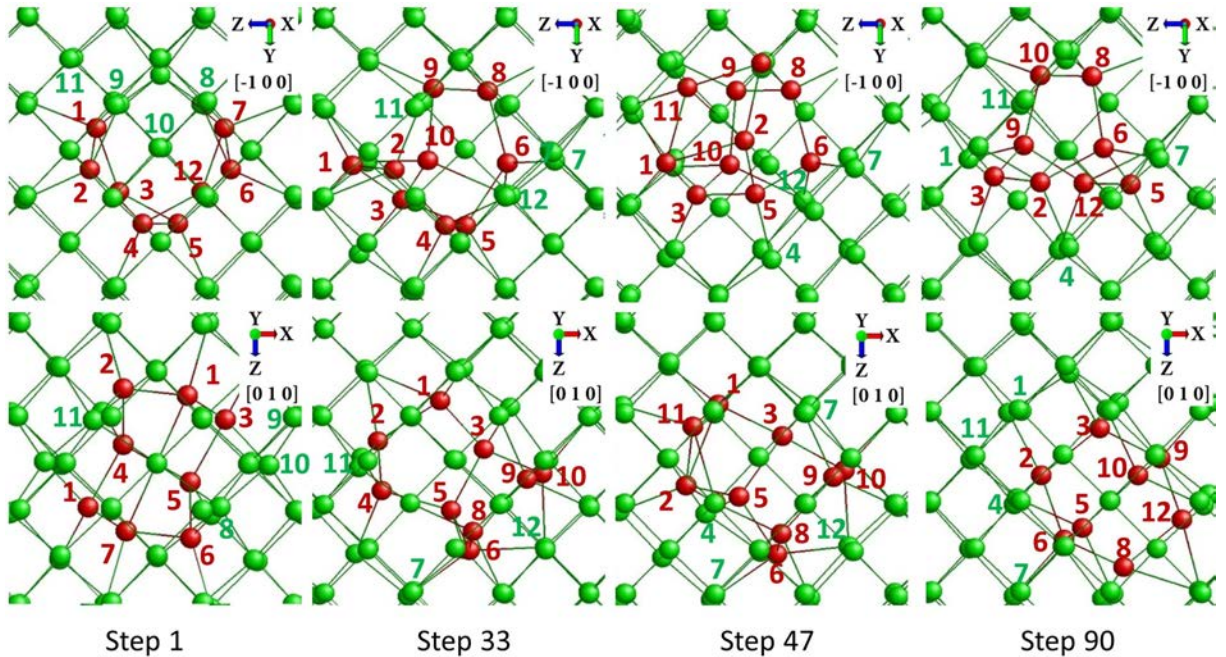


Figure 5.8: Chainlike to compact I_2 configuration evolution in $[-1\ 0\ 0]$ (up) and $[0\ 1\ 0]$ (down) planes over four selected kMC steps.

5.3 Tetrainterstitial

This section is aimed to study I_4 compact, chainlike and $\{111\}$ configurations. Firstly we are going to sum up the simulations we have run and then the formation enthalpies and energy barriers will be determined. Additionally, we will analyse in detail a pair of transitions.

5.3.1 Summary of simulations

The computational cost has increased at this point, as it can be seen in table 5.3. The facts that the network size is bigger and the transitions are harder to achieve have contributed to this issue. We have run 5 simulations starting from a compact configuration, 8 simulations beginning from a short chainlike defect, 4 simulations taking a long chainlike configuration as initial step and 8 simulations starting from a $\{111\}$ configuration. The CPU time is almost 3000 h, three times larger than the cluster sizes we have studied until now. In addition, transitions have been seen more frequently, 1 out of 2 attempts.

Number of simulations	Total CPU time (h)	Success rate (%)	Total simulated time (ms)
25	2859.24	52.00%	554426.70

Table 5.3: Summary of simulations which have been run for tetrainterstitial.

According to this work, if one wants to see a transition from the compact to the short chainlike configuration starting from the first one, even setting `MIN_SIG_BARRIER` to 2.1 eV is not enough. However, as kART let the user load a topological database from a previous simulation, it is likely to see it by setting this parameter to 1.5 eV and loading the catalogue of a simulation which has started from a long chainlike configuration. In addition, the transition from the long chainlike to the compact configuration can be achieved setting the mentioned parameter to 0.9 eV, and the same is needed to see transitions between long and short chainlike configurations. As it happened with the I_3 , the $\{111\}$ configuration does not present this issue because one can evolve to other known configuration with a 0.3 eV energy barrier. Moreover, there are no kMC steps in which the atoms are not grouped in a unique defect

5.3.2 Formation enthalpies and energy barriers

Contrary to what we have exposed in I_3 results, here the compact configuration is 1.1 eV lower than the short chainlike configuration and 1.6 eV lower than the long chainlike one. This agrees with the high stability of the fourfold structure, which has been called in the literature as Arai tetrainterstitial [16]. In addition, $\{111\}$ falls again into chainlike or compact configurations with low energy barriers. Regarding to the transitions, we have considered 23 samples: 3 transitions from short chainlike to compact configuration, 13 transitions between long and short chainlike configurations, 1 from compact to long chainlike configuration, 22 between $\{111\}$ and short chainlike configuration and 1 from $\{111\}$ to compact configuration. This can be checked in figure 5.9, taking into account what we have already explained for this kind of graphs.

It should be noted that the barrier between chainlike and compact configurations is high (2.28 eV). This means that this transition will be difficult to see, needing high temperatures or long times. In any case, Santos *et al.* [90] have also calculated the barrier between compact and chainlike configuration with NEBM, being 2.85 eV higher with respect to ours.

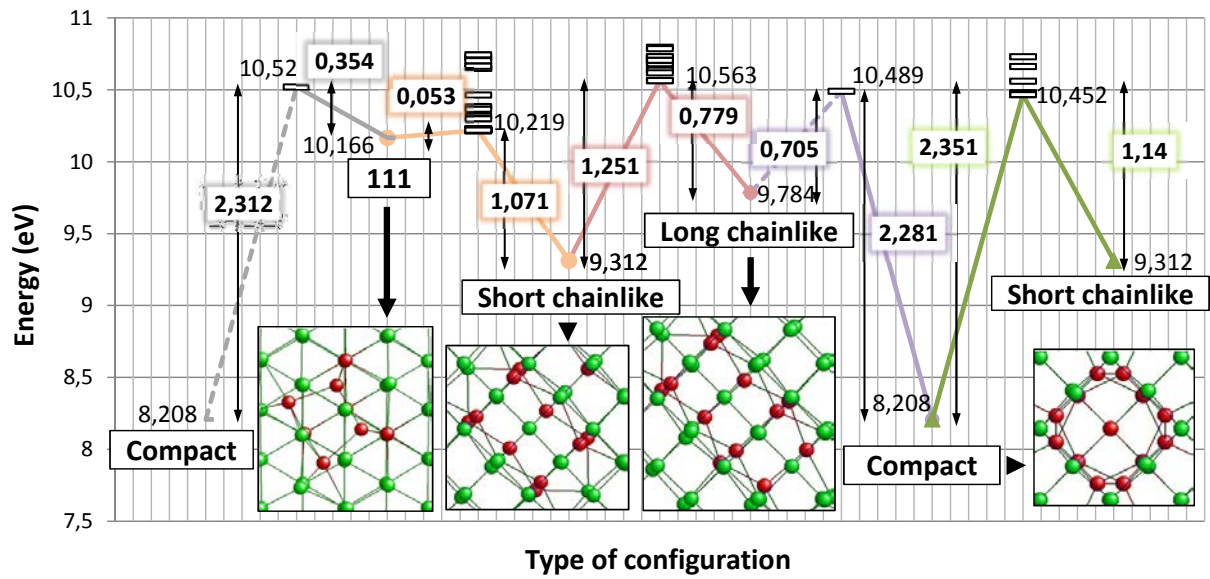


Figure 5.9: Transition path between compact, long and short chainlike and $\{111\}$ I_4 configurations indicating the formation enthalpies, topologies in $[-1\ 0\ 0]$ plane for the first three and in $[-1\ 1\ -1]$ for the fourth (DAs in red, bulk atoms in green) and all energy barriers found.

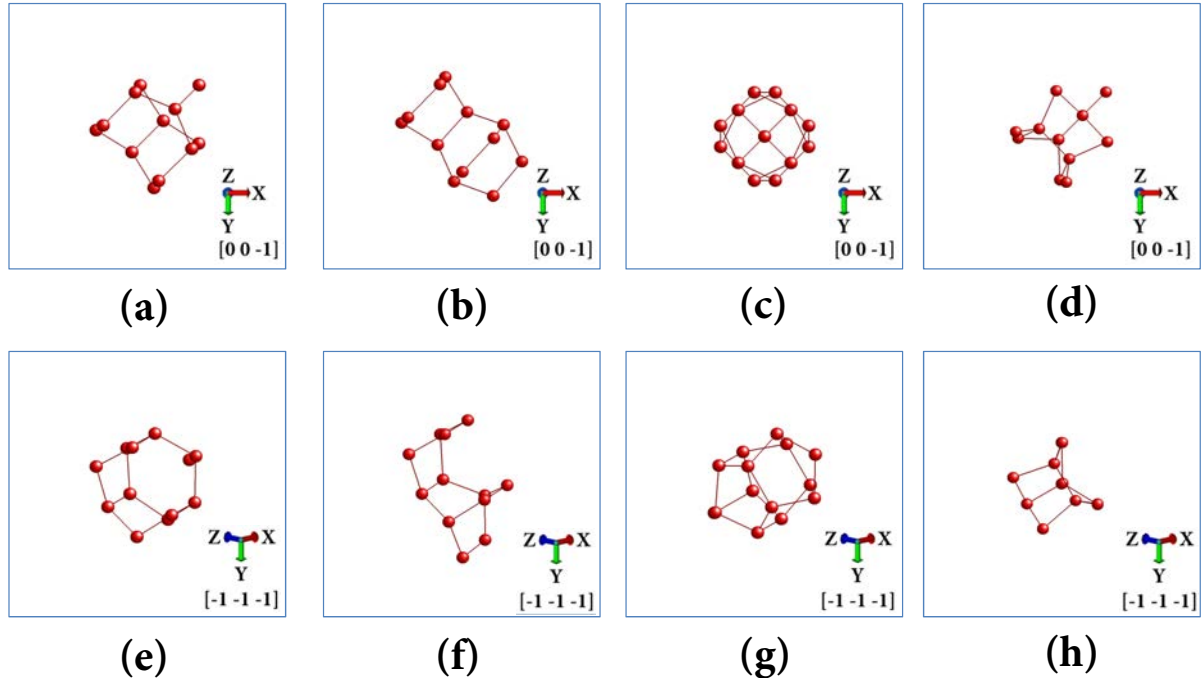


Figure 5.10: DAs of short and long chainlike, compact and $\{111\}$ I_3 configurations in $[-1\ 0\ 0]$ plane (a, b, c, d) and $[-1\ -1\ -1]$ plane (e, f, g, h), respectively.

The topology of these energy minima in several orientations is represented in the appendix A. Figure 5.10 shows the DAs of each configuration in order to see the arrangement of atoms when a transition takes place. Compact defects presents 14 DAs and 10 ESs, short and long chainlike configurations have 12 DAs and 8 ESs while $\{111\}$ defects are formed of 11 DAs and 7 ESs. Logically the extension of the long chainlike defects along the network continues growing with the cluster size. The short chainlike, however, only implies a DA in the lateral structure comparing it with the I_3 version. Additionally, as the fourfold is completed for the compact defect, evolving to other configuration is complex.

5.3.3 Transition mechanism

We are going to analyse two situations: a transition from the short chainlike to the compact configuration and a transition from the short chainlike to the long chainlike configuration. The first one involves more than 50 steps while the second one takes place in 20 steps. Hence, we will only represent the most relevant stages of the process. It is worth noting that these transitions occur in the same plane, unlike I_3 case.

As it can be seen in figure 5.11, the transition between short chainlike and compact configurations is quite complex. The initial steps consist of the approach of atom 12 to the chain structure, rearranging the bottom atoms, 8, 9, 10 and 11, and the central ones, 5 and 6. Lately, when we are in the middle of the transition, atom 7 displaces to upper positions, modifying the location of atoms 3 and 4. At this point, the compact structure can be foreseen. Finally, the atoms which are on the corner of the square (atoms 1, 2, 3, 4, 9, 10, 11 and 12) move slightly to the compact positions. In addition, atom 7 is displaced to a middle position, as it occurs to atoms 5, 8 and 16. Atom 6, however, goes to the central position of the compact configuration.

Now we are going to detail the transition between short and long chainlike configurations. Figure 5.12 illustrates this idea. As it has happened in the previous transition, the first stages consist of the approach of atom 12 to the chain structure, rearranging the upper atoms, 1, 2, 3 and 4, and the central ones, 5 and 6. In fact, atom 12 ends up filling the position of atom 4 in the step 17, making atom 4 go to upper sites and atom 13 leave its perfect lattice position. Thus, atom 4 and 13 form the top part of the long chainlike configuration at the end. The rest of the atoms remains almost unaltered if one compares them with the initial short chainlike positions.

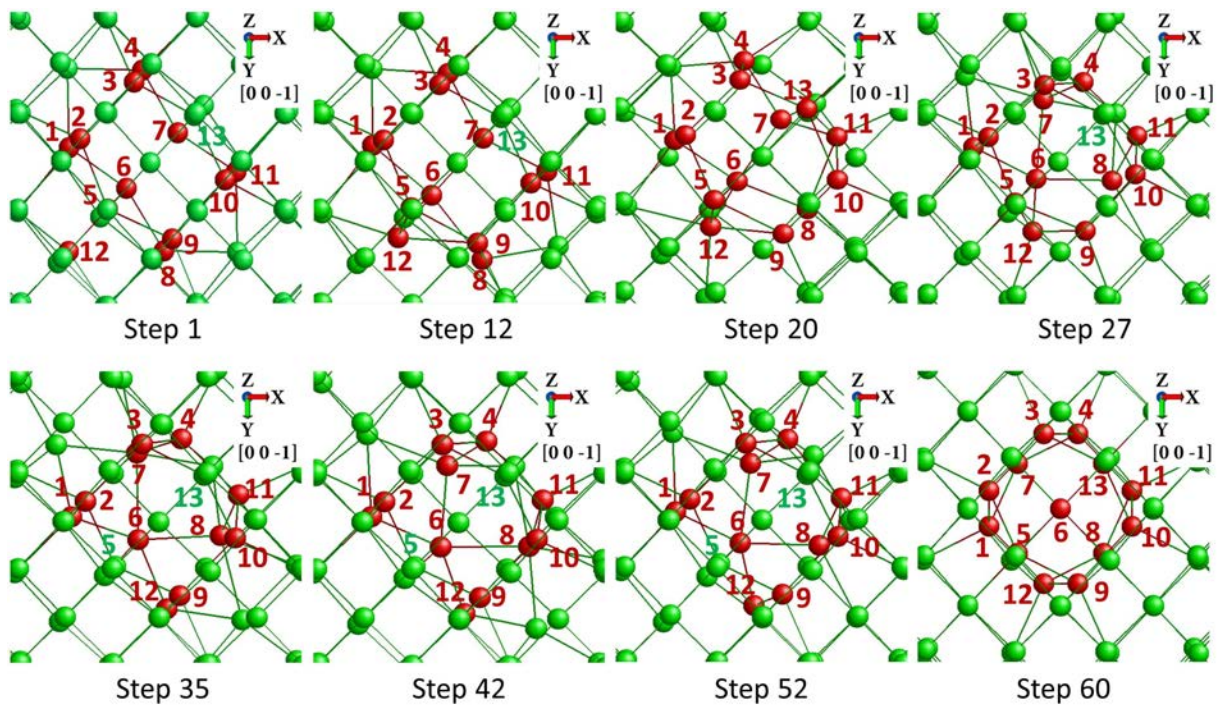


Figure 5.11: Short chainlike to compact I_4 configuration evolution in $[0\ 0\ -1]$ plane over eight selected kMC steps.

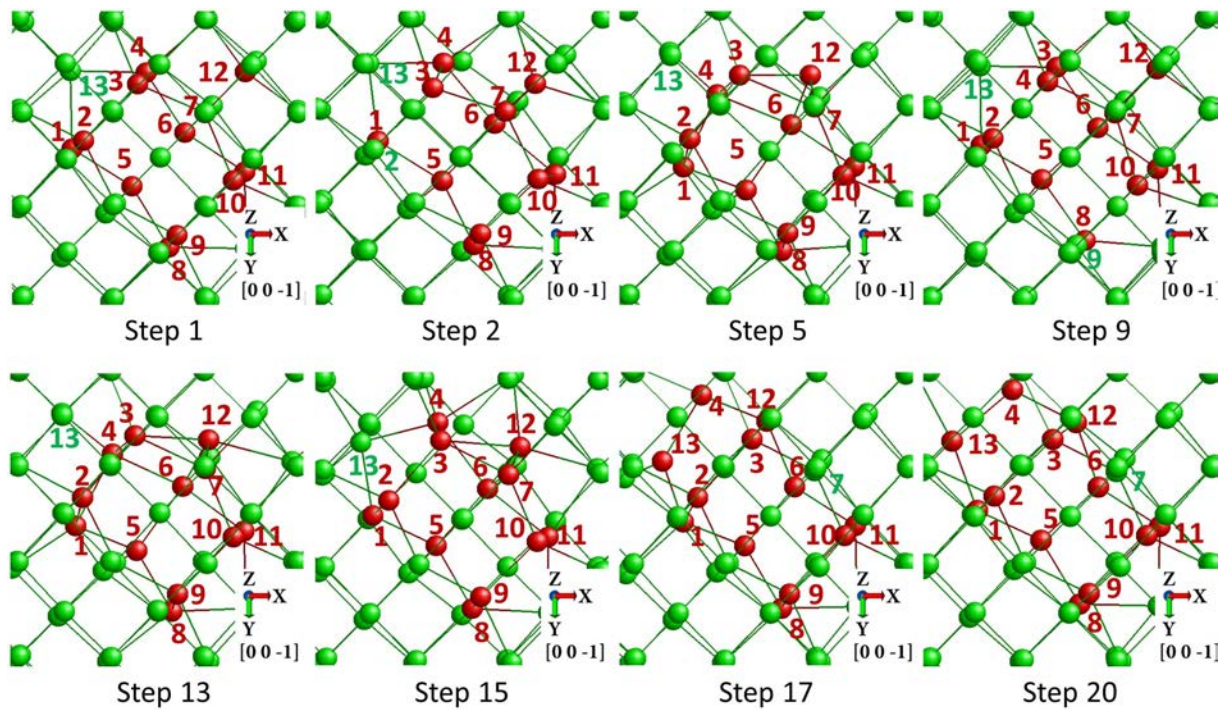


Figure 5.12: Short chainlike to long chainlike I_4 configuration evolution in $[0\ 0\ -1]$ plane over eight selected kMC steps.

5.4 Pentainterstitial

To continue with this chapter, we are going to analyse compact, short and long chainlike and $\{111\}$ configurations for I_5 . Following the same lines as before, an overview of the simulations is presented and the formation enthalpies and energy barriers are indicated. In addition, two transitions will be described in detail.

5.4.1 Summary of simulations

The computational cost has been lower than in the previous section thanks to the increment in success rate, as it can be seen in table 5.4. Transitions, though hard to achieve, have been seen more frequently (almost half of the attempts have been successful). In any case, the CPU time is also large, being close to 1700 h due to the network size and the high energy values we have worked with. We have run 2 simulations starting from a compact configuration, 3 simulations beginning from a short chainlike defect, 7 simulations taking a long chainlike configuration as initial step and 5 simulations starting from a $\{111\}$ configuration.

Number of simulations	Total CPU time (h)	Success rate (%)	Total simulated time (s)
17	1657.31	47.06%	17.46

Table 5.4: Summary of simulations which have been run for pentainterstitial.

Taking into account all our data, one can see transition from the compact to the short chainlike configuration starting from the first one setting `MIN_SIG_BARRIER` to 0.9 eV. Therefore, this transition seems to be easier than the I_4 case. However, here we have not seen the transition from the long chainlike to the compact configuration even setting this parameter to 1.8 eV. Additionally, transition between long and short chainlike has been seen when the minimum barrier of the basin is set to 1.5 eV, 0.6 eV higher than in the previous section. The same applies to $\{111\}$ configuration, which has needed to set this parameter to 0.6 eV to see transitions.

5.4.2 Formation enthalpies and energy barriers

Figure 5.6 shows the transition path we have determined for this cluster size. It should be highlighted that compact configuration is again the one with the lowest formation

enthalpy. The high stability of the fourfold structure justifies it (the extra atom respect to the I_4 compact configuration is away from the fourfold). However, here the difference between chainlike and compact energies (0.648 eV) is lower than what we calculated in the previous section. In addition, short chainlike configuration presents a 0.609 eV higher energy than the long version, again contrary to I_4 results. The $\{111\}$ configuration also differs from what we pointed in I_4 case. It evolves to the compact configuration, as it happened in the I_2 , and its barrier is significant, 0.627 eV. It is worth noting than the previous transitions paths for this configuration implied very low energy barriers, being a more unstable energy minimum.

Regarding to the transitions, we have considered 10 samples: 3 transitions from short chainlike to compact configuration, 3 transitions from long and to short chainlike configurations and 4 from $\{111\}$ to compact configuration. Hence, we have only seen transitions in one direction. The energy barrier to go away from the compact configuration is 1.811 eV, a high value. One would need very high temperatures or long periods of time to reproduce it. In any case, the statistics are poor and they also present variability. Once again, several energy paths can be chosen to go from one configuration to the other.

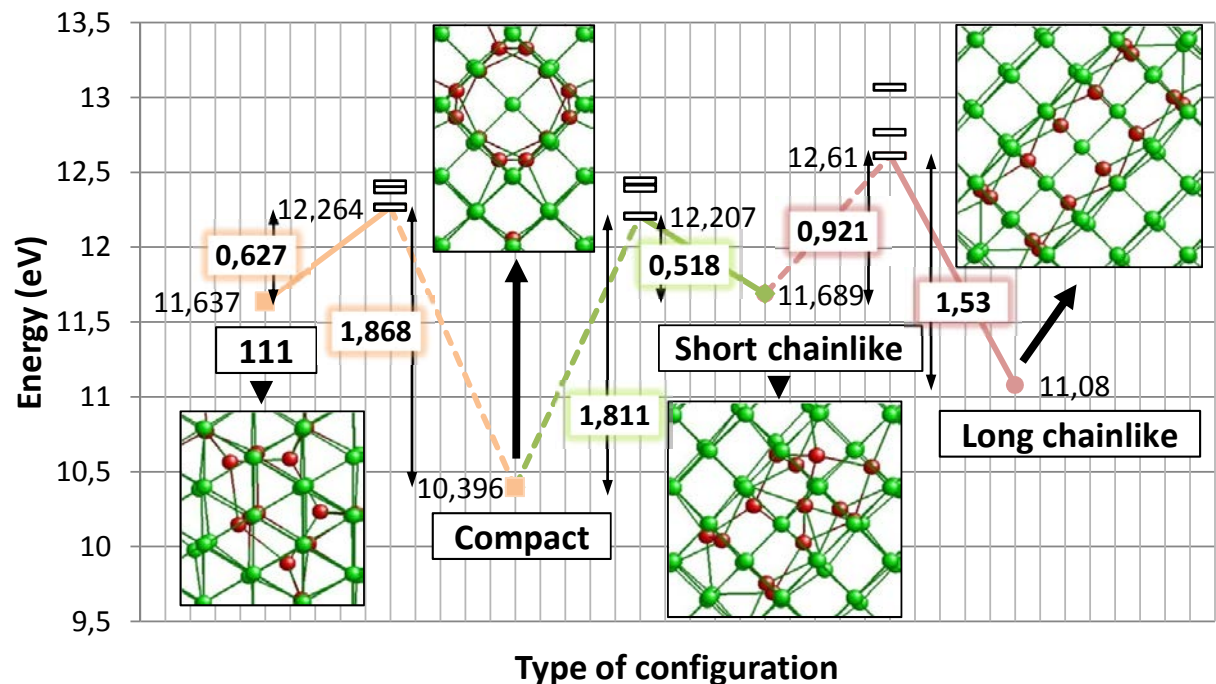


Figure 5.13: Transition path between compact, long and short chainlike and $\{111\}$ I_5 configurations indicating the formation enthalpies, topologies in $[-1\ 0\ 0]$ plane for the first three and in $[-1\ 1\ -1]$ for the fourth (DAs in red, bulk atoms in green) and all energy barriers found.

The completed topology of these energy minima as well as other frequent ones are represented in the appendix A. Furthermore, figure 5.14 represents the DAs of each configuration in order to see how the atoms have to rearrange during transition. Compact and $\{111\}$ defects present 15 DAs and 10 ESs, short chainlike configurations have 13 DAs and 8 ESs and long chainlike defects present 16 DAs and 11 ESs. The extension of the chainlike defects adds an atom respect to the I_4 case. The difference between all configurations is more significant, being logical that the energy barriers had also increased.

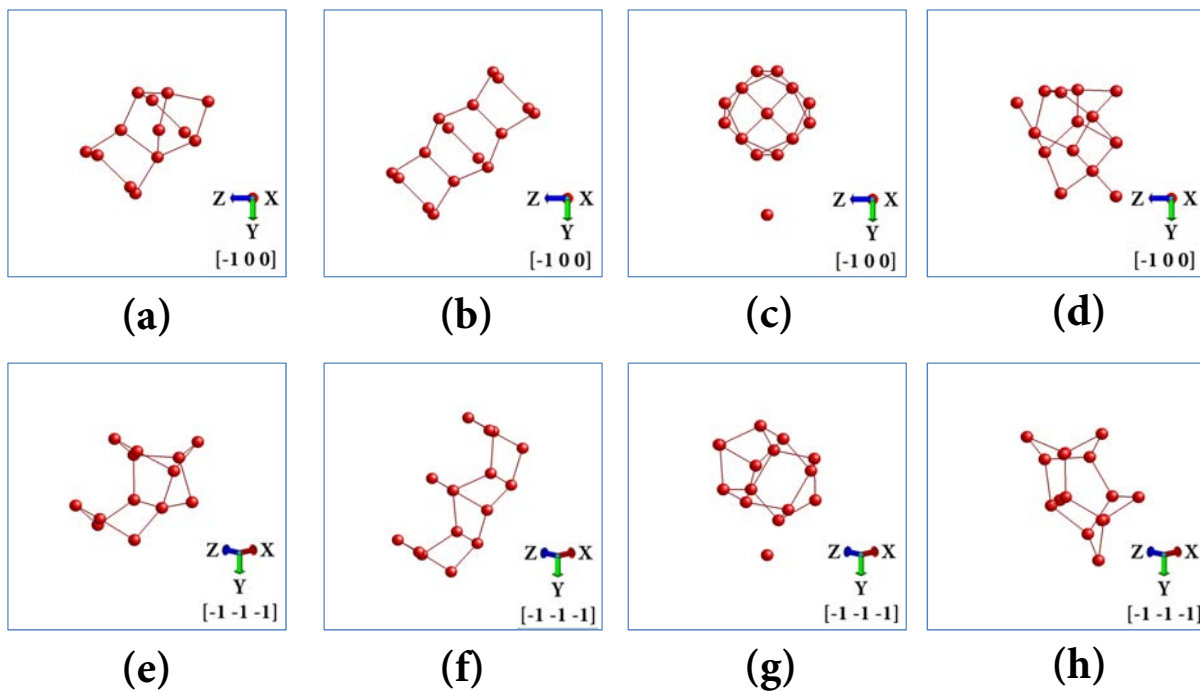


Figure 5.14: DAs of short and long chainlike, compact and $\{111\}$ I_3 configurations in $[-1\ 0\ 0]$ plane (a, b, c, d) and $[-1\ -1\ -1]$ plane (e, f, g, h), respectively.

5.4.3 Transition mechanism

We are going to analyse two different situations: a transition from the short chainlike to the compact configuration and a transition from the long chainlike to the short chainlike configuration. The first one involves more than 70 steps while the second one takes place in almost 150 steps. Therefore, only the most relevant stages of the diffusion will be represented. It should be noted that these transitions occur in the same plane, unlike I_3 case.

As it can be seen in figure 5.15, the transition between short chainlike and compact configurations is not simple. The first 20 steps consist of slight displacements of all atoms without any significant movement. When 29 steps have passed, however, a more symmetrical short chain is formed. It implies displacing the atoms 10 and 13 to upper positions and atom 7 to a perfect lattice one. In any case, this configuration is higher in energy than the initial structure. After that, atom 13 is displaced down, making atom 7 move from its previous position in step 42. The following stages are related to the formation of the compact structure. Atoms 1, 2, 3, 4, 11 and 12 get slightly displaced to form the bottom part of the fourfold structure, atom 5 goes up and form the upper part with atom 9, and atom 7 ends up in the central position of the compact configuration. Regarding the extra atom of the fourfold, atom 8 goes away to an upper position and atom 10 goes to a perfect lattice one.

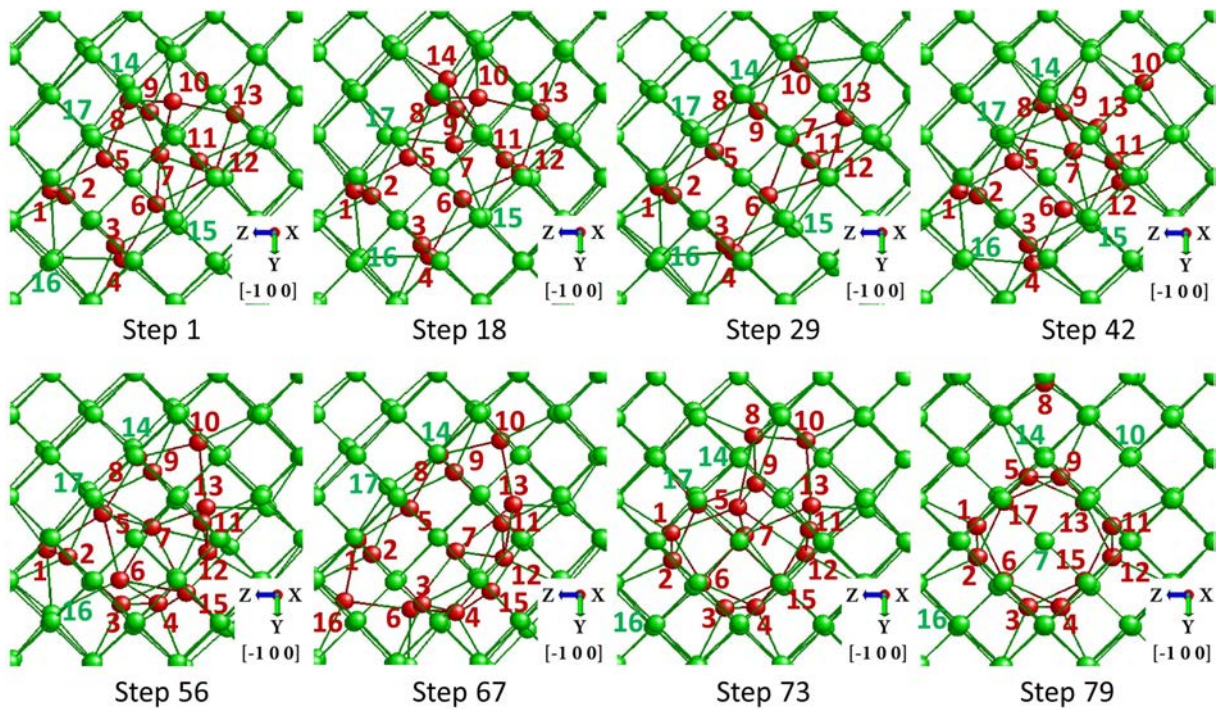


Figure 5.15: Short chainlike to compact I_5 configuration evolution in $[-1\ 0\ 0]$ plane over eight selected kMC steps.

Once the transition to a compact configuration is analysed, now we are going to detail the evolution from a long to a short chainlike configuration. As it can be extracted from figure 5.16, the atoms which form the bottom part of the chain, i.e., atoms 1, 2, 3, 4, 5, and 6, remain unaltered during all the process. The upper part, however, experiments several changes. Firstly atoms 13, 14 and 15 go to the nearest perfect lattice position and

atom 17 leaves the chain conformation. However, when more than 80 steps have elapsed, atom 7 comes back, atom 13 displaces to upper sites and atom 6 goes to the left. After 120 steps several changes take place: atom 13 returns to a perfect lattice position and atom 16 moves first to the right and lately goes down, filling atom 12 its position. Thus, the short chainlike configuration is formed.

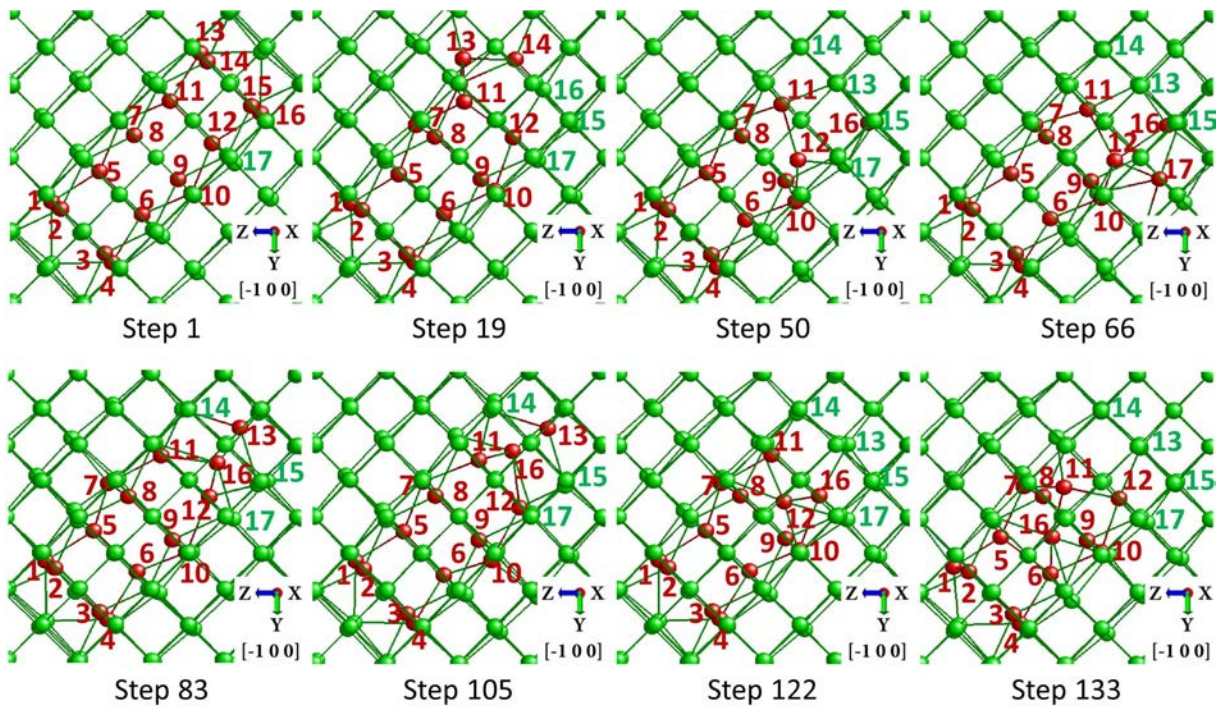


Figure 5.16: Short chainlike to long chainlike I_5 configuration evolution in $[-1\ 0\ 0]$ plane over eight selected kMC steps.

5.5 Hexainterstitial

To end with this chapter, here we are going to present some preliminary results we have obtained for cluster I_6 . It should be noted that the statistics are not as good as the previous ones.

5.5.1 Summary of simulations

We have run 2 simulations starting from a compact configuration, 5 simulations beginning from a chainlike defect and 5 simulations starting from a $\{111\}$ configuration.

Computing all of them the CPU time is close to 2800 h, but we have only seen transitions in 16.67 % of all attempts, as table 5.5 indicates.

Number of simulations	Total CPU time (h)	Success rate (%)	Total simulated time (s)
12	2767.53	16.67%	12.272

Table 5.5: Summary of simulations which have been run for hexainterstitial.

In this sense, setting `MIN_SIG_BARRIER` to 2.4 eV is not enough to see transitions between compact and chainlike configurations. The $\{111\}$ configuration can evolve to the chainlike configuration with a 1.2 eV limit of this parameter. In any case, after setting these values there are no kMC steps in which the atoms are not grouped in a unique defect, as it happened in the previous sections.

5.5.2 Formation enthalpies and energy barriers

According to our results, we can point out that the formation enthalpy of compact configuration is 10.396 eV, a lower value than those presented by the chainlike (11.08 eV) configuration and the $\{111\}$ configuration (11.689 eV). Here the system has evolved from a $\{111\}$ to a chainlike configuration twice, being the only transition we have seen. In any case, the energy diagram is represented in figure 5.17. The transition from a $\{111\}$ defect to a chainlike one involves a saddle point of 2.5 eV, a very high value. Hence, the $\{111\}$ configuration seems to be more stable than the previous sizes under analysis.

The topology of these energy minima as well as other frequent ones are represented in the appendix A. Figure 5.18 represents the DAs of each configuration in order to see how the atoms have to displace. Compact defects present 16 DAs and 10 ESs and chainlike configurations have 18 DAs and 12 ESs and $\{111\}$ configurations present 22 DAs and 16 ESs. The extension of the chainlike defects along the network is appreciable if we compare it with the I_5 case. The compact defect continues adding the extra atoms near the fourfold structure. We can also observe the high symmetry of $\{111\}$ configuration in that plane.

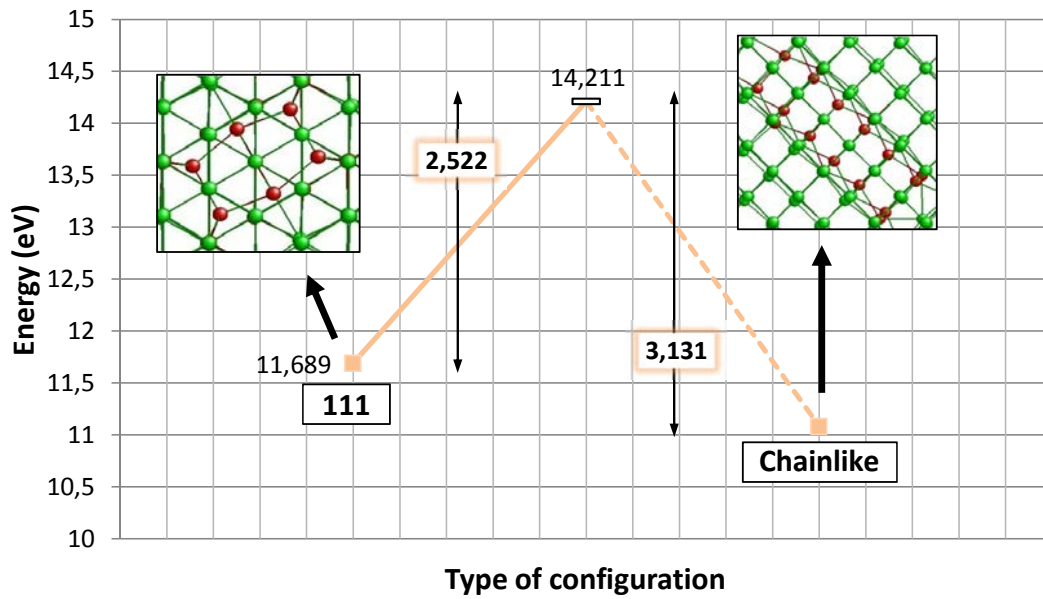


Figure 5.17: Transition path between chainlike and $\{111\}$ I_6 configurations indicating the formation enthalpies, topologies in $[-1\ 0\ 0]$ plane for the first and in $[-1\ 1\ -1]$ for the second (DAs in red, bulk atoms in green) and all energy barriers found.

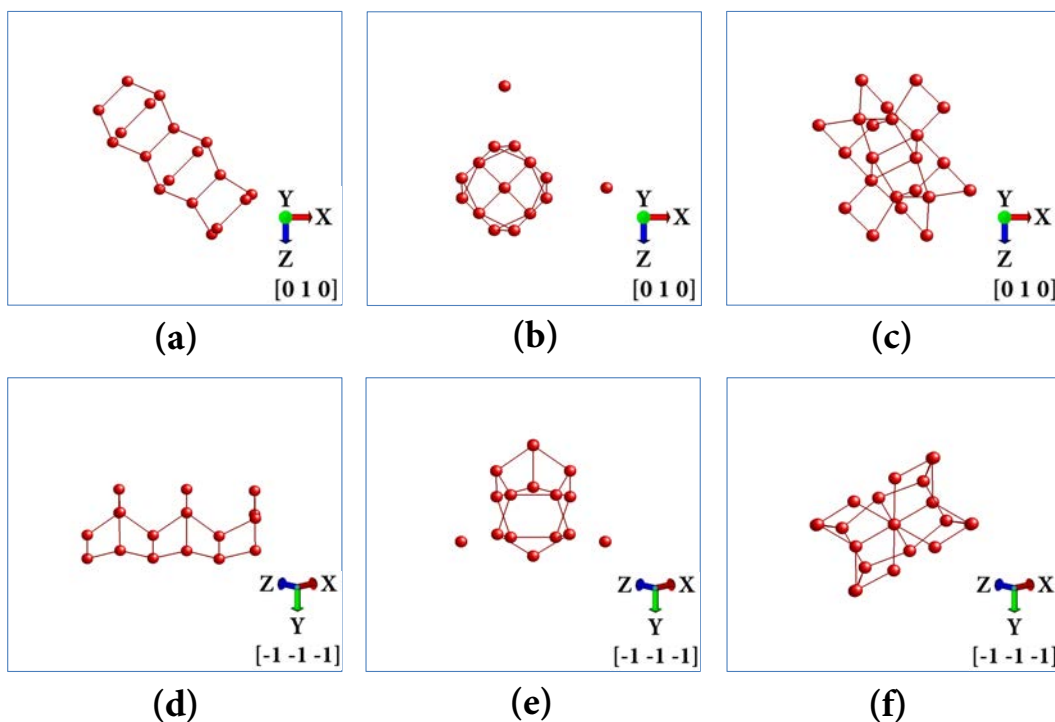


Figure 5.18: DAs of chainlike, compact and $\{111\}$ I_6 configurations in $[-1\ 0\ 0]$ plane (a, b, c) and $[-1\ -1\ -1]$ plane (d, e, f), respectively.

5.6 Conclusions

We have studied the Si self-interstitial clusters configurations from I_2 to I_6 by using Tersoff 3 interatomic potential in a kART scheme. The formation enthalpy of each configuration and the energy barriers of the possible transitions between them have been determined.

The most stable configuration for I_2 , I_4 , I_5 and I_6 is the compact one. For the I_3 , however, the chainlike configuration presents the lowest formation enthalpy. We have dealt with high energy barriers to go away from the lowest energy states, reaching almost 2 eV for the biggest cluster size. In any case, the amount of barriers which have been analysed is not quite large, existing different energy paths to go from one configuration to another. We have also studied the transition mechanism between the main configurations. The evolution is more complex as the cluster size increases, being logical that the energy barriers increase too.

Chapter 6

Growth and evolution of small interstitial clusters

*Knowing is not enough; we must apply.
Willing is not enough; we must do.
~ Johann Wolfgang von Goethe ~*

AFTER ANALYSING EACH SMALL INTERSTITIAL CLUSTER SIZE INDIVIDUALLY, we are going to detail certain aspects about how the formation enthalpies and energy barriers vary with the size and how they are influenced by changes in pressure. These studies will enable us to analyse the growth mechanisms of small interstitial clusters and their evolution.

The chapter is structured as follows. Firstly we are going to focus on the energetic aspects of each configuration when the cluster size increases, giving an insight about which of them are more likely to appear when interstitials aggregate. Lately the effect that pressure has on energy properties is going to be described, indicating the energetic variations for tri-interstitial and tetrainterstitial.

It should be noted that all the simulations we have launched for each size present the same number of interstitials, i.e., we have seen transitions between configurations which share the number of defects. However, real experiments may involve transitions between different cluster sizes due to the fact that atoms inside a material can reach the defect. Hence, its size can be increased or decreased, being necessary to study the capture and emission of interstitials.

6.1 Formation enthalpies

Firstly, figure 6.1 shows the formation enthalpy per interstitial for each configuration of small interstitials clusters. They refer to the formation enthalpy divided by the number of interstitials for each size, a straightforward method to let us interpret the growth mechanism. For those cases in which two possible structures can exist for the chainlike configuration (short and long), we have displayed the ones with the lowest energy values. Hence, the energy of short chainlike configuration is shown for I_4 and the energy of long chainlike configuration is represented for I_5 .

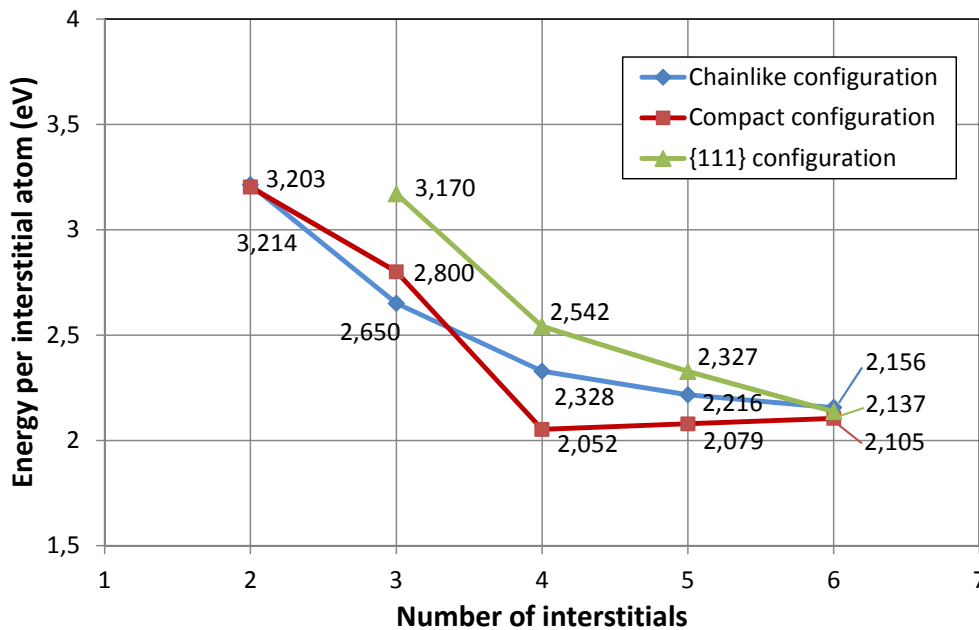


Figure 6.1: Energy per interstitial for each configuration of small interstitials clusters.

The most stable configuration is the compact one, except for I_3 case when the chainlike is lower in energy. The {111} configuration presents higher energy values for all sizes until the number of interstitial reaches 6. When the cluster increases, the formation enthalpy must decrease because it is the main reason of its aggregation. Although the {111} and chainlike configuration agree with the previous idea, it should be emphasized that the compact I_4 configuration has less energy per interstitial than compact I_5 and I_6 configurations. Therefore, it may act as an stopper in the growth mechanism of this kind of defects. In other words, the transition between compact I_4 and I_5 is not energetically favourable.

In fact, some simulations done by L.A. Marqués *et al.* [10] have shown this idea, as can be seen in figure 6.2. He carried out CMD simulations inserting many interstitials in a cell and letting them evolve to form extended defects. The results pointed out that compact I_4 is difficult to aggregate into others, being a very high stable configuration at low temperatures.

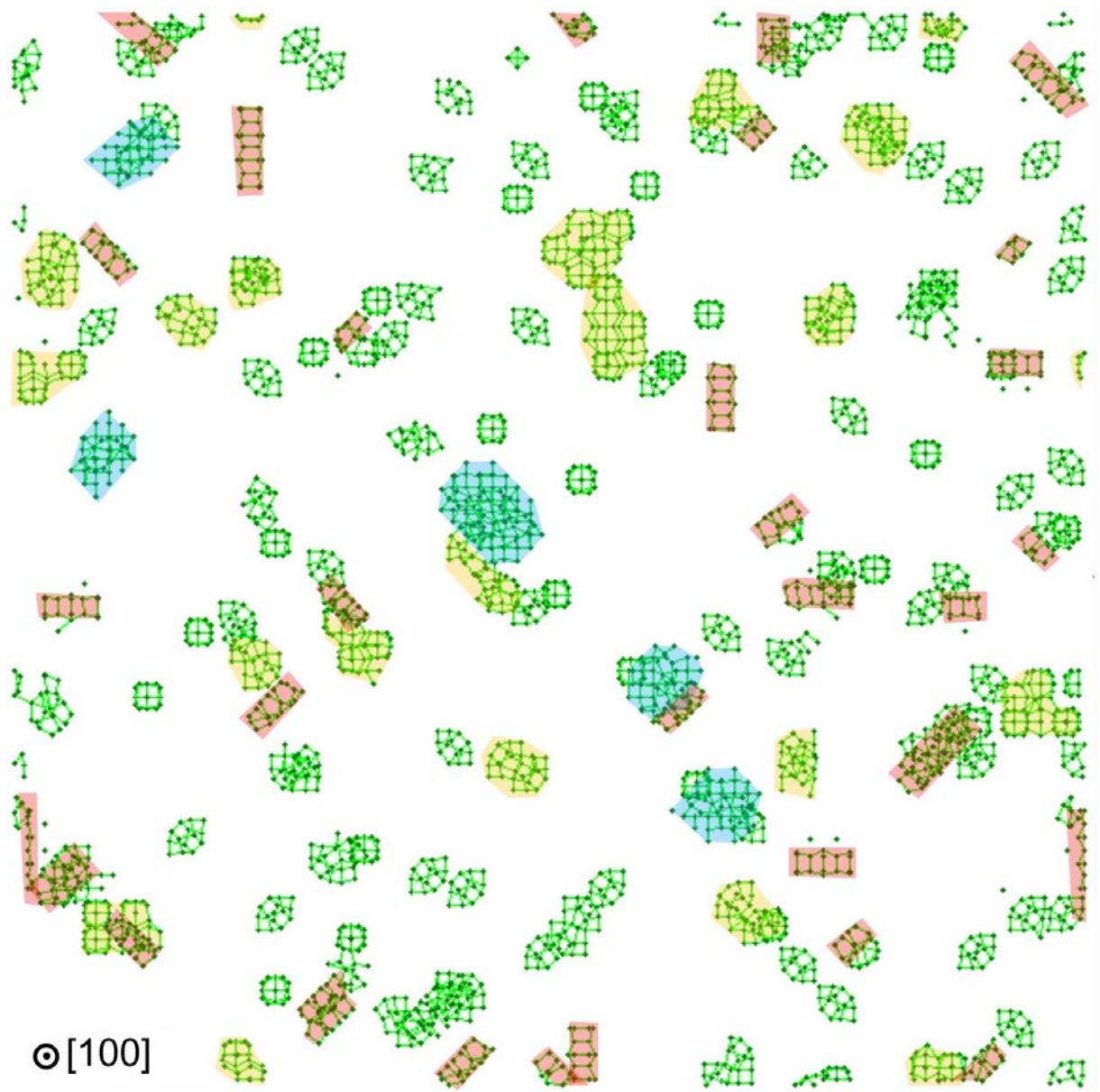


Figure 6.2: Annealing at 850 K of a cell which contains different types of big defects ($\{111\}$ RLDs in blue and (110) chains in red) and compact I_4 and his agglomerates (in yellow) in 170 ns.

Other authors in the literature have also studied the formation enthalpies of chainlike and compact configurations through several methods. Figure 6.3 shows the comparison with their results. S. Lee *et al.* [22, 23] carried out TBMD and DFT simulations while N. Cowern *et al.* [15] applied an Ostwald Ripening analysis focusing on the lowest energy configuration without indicating which one is. This last analysis is known as the magic sizes, a common term very extended in the field of simulation models for semiconductor defects.

Comparing our results with those obtained by S. Lee *et al.*, similarities can be found in the chainlike formation enthalpies. In addition, compact energies do not differ so much for big sizes, though the smaller ones (2 and 3 interstitials) present a large variation. It is difficult to determine why this difference appears. kART uses Tersoff 3 interatomic potential through LAMMPS to calculate the energy of the system. The energies depend on the potential and, therefore, it may change if other potential is used to describe the atomic network. The energies obtained by N. Cowern *et al.* should be understood as an average of all configurations, being close to the lowest energy configurations for each size (compact in general).

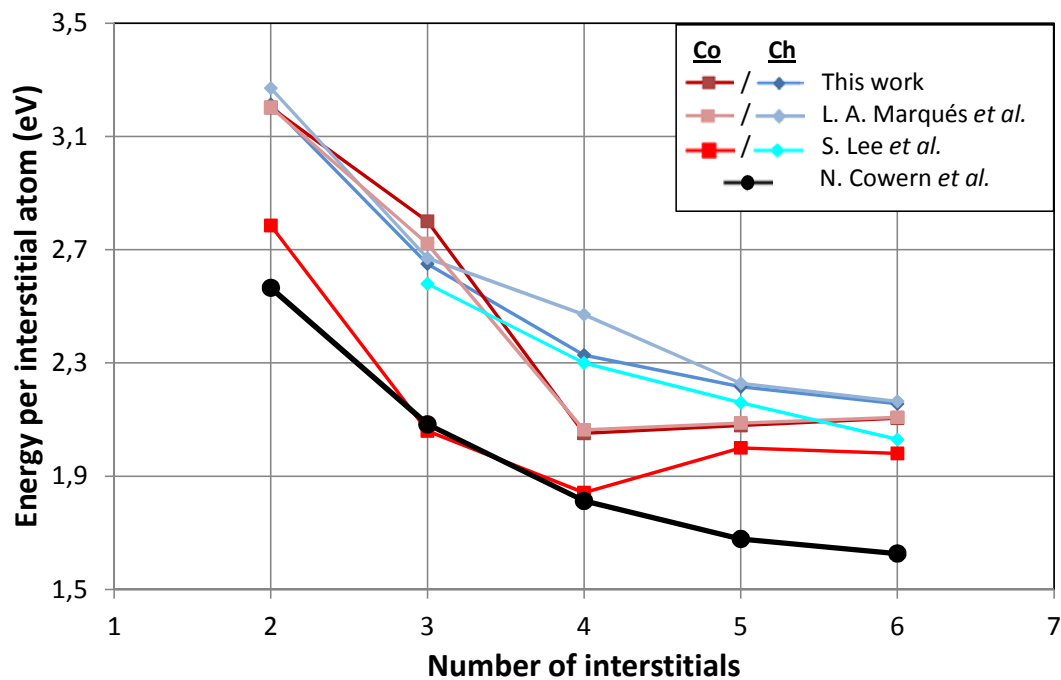


Figure 6.3: Comparison of energy per interstitial for each configuration of small interstitials clusters with other authors [15, 22, 23]

6.2 Energy barriers

Table 6.1 summarizes all the energy barriers we have determined in this work. Those transitions which have taken place in one direction but not in the opposite one are indicated in *italic*. In addition, we have not distinguished between long and short chainlike configurations for I_4 and I_5 cluster sizes, taking the lowest formation enthalpy configuration of both as a reference in these cases.

Cluster size	Transition	Minimum value (eV)	Average value (eV)	Standard deviation (eV)
I_2	Chainlike \rightarrow Compact	0.397	0.454	0.114
	Compact \rightarrow Chainlike	0.376	0.467	0.092
I_3	Chainlike \rightarrow Compact	1.29	1.29	0
	Compact \rightarrow Chainlike	0.894	0.999	0.117
	Chainlike \rightarrow 111	<i>1.563</i>	-	-
	111 \rightarrow Chainlike	0.014	0.132	0.119
	Compact \rightarrow 111	<i>1.112</i>	-	-
	111 \rightarrow Compact	0.003	0.003	0
I_4	Chainlike \rightarrow Compact	1.14	1.289	0.122
	Compact \rightarrow Chainlike	2.259	2.265	0.008
	Chainlike \rightarrow 111	0.907	0.938	0.087
	111 \rightarrow Chainlike	0.053	0.289	0.216
	Compact \rightarrow 111	<i>2.312</i>	-	-
	111 \rightarrow Compact	0.354	0.354	0
I_5	Chainlike \rightarrow Compact	1.53	1.669	0.123
	Compact \rightarrow Chainlike	<i>1.811</i>	-	-
	Compact \rightarrow 111	<i>1.868</i>	-	-
	111 \rightarrow Compact	0.627	0.697	0.081
I_6	Chainlike \rightarrow 111	<i>3.131</i>	-	-
	111 \rightarrow Chainlike	2.522	2.572	0.071

Table 6.1: Statistical analysis of all energy barriers for each cluster size.

Focusing now on the minimum energy barriers we have found during this work, figure 6.4 represents how they evolve with the cluster size in transitions between compact and chainlike configurations. We can see that, except for the I_2 case where the transition in both directions is likely, high energy barriers are needed to go from the lowest energetic configuration to the highest one for each size. Additionally, falling from the higher configurations also involves high energy barriers. As it happened with the analysis of

formation enthalpies, we can observe that the compact I_4 presents the most energetic barrier to going away from it (2.281 eV). This implies that the transition from compact to chainlike I_4 would take place every 4 ms in an annealing at 800 °C, a very high value comparing with the returning time (23 ns). Hence, this configurations may slow down the cluster growth not only because of its low energy but also for its high energy barrier to go away from it.

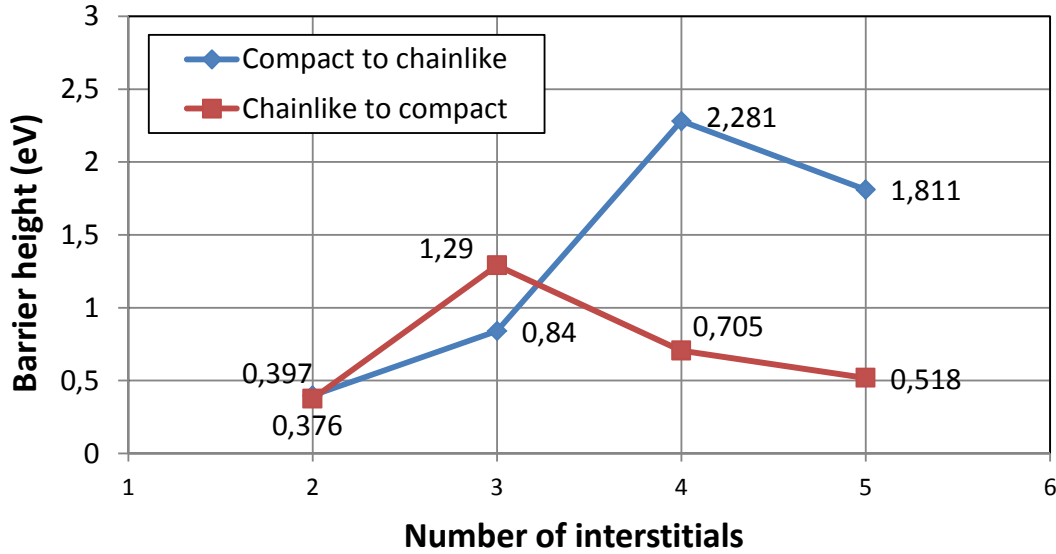


Figure 6.4: Evolution of energy barriers in transitions between compact and chainlike configuration for each cluster size.

It should be noted that for those cases in which two possible structures can exist for the chainlike configuration (short and long), we have displayed the ones which are the lowest energy values. If one thinks in terms of estimated time, these barriers can be translated into time by a simple conversion:

$$T_{est} = \frac{1}{f_0} e^{\frac{E_B}{k_B T}} \quad (6.1)$$

Being T_{est} the estimated time, f_0 the vibration frequency of the network (10^{13} Hz), E_B the energy value, k_B the Boltzmann constant and T the temperature in Kelvin. Hence, the previous energy values are converted as table 6.2 indicates at 1200 K. We can see that, although times for I_2 and I_3 energy barriers are accessible, the barriers to go away from the compact would involve several ms. This would be infeasible with an CMD approach.

Cluster size	Estimated time (ns)	
	Compact \rightarrow Chainlike	Chainlike \rightarrow Compact
I ₂	4.65E-03	3.79E-03
I ₃	3.37E-01	2.62E+01
I ₄	3.80E+05	9.14E-02
I ₅	4.04E+03	1.50E-02

Table 6.2: Estimated time needed at 1200 K to see the transitions of figure 6.4.

Regarding the transitions between $\{111\}$ and compact and chainlike configurations, figure 6.5 displays how they evolve when the cluster size gets bigger. The values to going away from $\{111\}$ are low, but they increase with the number of interstitials. This agrees with the idea showed with the formation enthalpies: $\{111\}$ becomes more stable when the cluster size increases. The return barriers, however, are very high for all cases, being even more energetic when the fourfold structure of the compact configuration is formed.

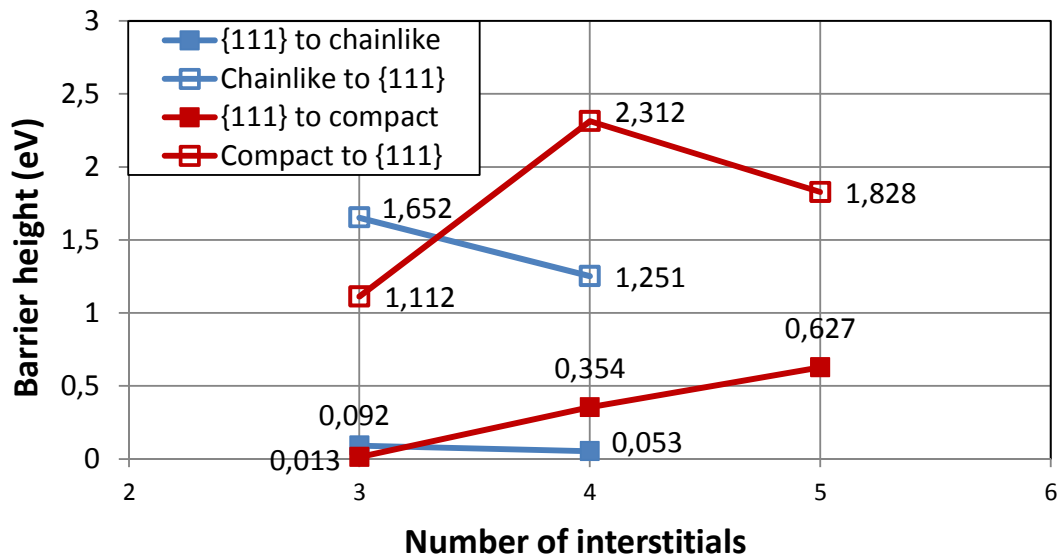
Figure 6.5: Evolution of energy barriers in transitions between $\{111\}$ compact and chainlike configuration for each cluster size.

Table 6.3 represents the previous energy barriers converted to time. Again, the barriers to go from compact to $\{111\}$ configuration would require a long time.

Cluster size	Estimated time (ns)			
	{111} → Compact	Compact → {111}	{111} → Chain.	Chain. → {111}
I ₃	1.13E-04	4.68E+00	2.43E-04	8.68E+02
I ₄	3.07E-03	5.13E+05	1.67E-04	1.80E+01
I ₅	4.30E-02	4.76E+03	-	-

Table 6.3: Estimated time needed at 1200 K to see the transitions of figure 6.5.

6.3 Emission and capture of interstitials

Until now we have only taken into account that a cluster of a certain size can change its configuration, but not the possibility of changing its size. Indeed, a cluster can do it by emitting or capturing interstitials. It should be remembered that after ion implantation occurs, an excess of interstitials exit which interact between themselves. Then they aggregate and create clusters, coexisting clusters with different sizes and configurations. In addition, this is a complex scheme where free interstitials can interact with the clusters. Thus, a cluster of a certain size and configuration can evolve in three different ways: reducing their size by emitting an interstitial, increasing their size by capturing an interstitial or changing to another configuration. The activation energy which controls each one of these processes lets us determine their frequency. In other words, it tells us the mean time needed for a process to take place.

The emission of an interstitial from a cluster requires a binding energy in order to separate the interstitial from the cluster and a migration energy to let the interstitial diffuse and go away from the cluster. Then, the binding energy of a cluster of n size, E_{bind,I_n} , is the difference between the formation enthalpy of the cluster of lower size, $H_{f,I_{n-1}}$, plus the formation enthalpy of a self-interstitial, $H_{f,I}$, respect to the formation enthalpy of the cluster, H_{f,I_n} , as equation 6.2 indicates.

$$E_{bind,I_n} = (H_{f,I_{n-1}} + H_{f,I}) - H_{f,I_n} \quad (6.2)$$

Knowing the formation enthalpy of each configuration and considering the formation enthalpy of a self-interstitial as 3.478 eV (from CMD simulations using Tersoff 3 interatomic potential [88]), the binding energy can be determined. The activation energy to emit an interstitial is, therefore, the addition of the binding energy and the migration energy of an interstitial (3.9 eV [91]). To estimate the associated time to the interstitial

capture for a certain cluster size and configuration, we suppose clusters are in equilibrium with free interstitials. Hence, other clusters with other sizes or configurations cannot technically exist. The concentration of free interstitials (C) is calculated from the binding energy of the cluster (E_{bind}):

$$C = C_o \exp\left(\frac{-E_{bind}}{kT}\right) \quad (6.3)$$

Being C_o the Si density ($5 \cdot 10^{22}$ atoms/cm³), k the Boltzmann constant and T the temperature in Kelvin. Consequently, the most stable clusters whose binding energies are the highest ones will be surrounded by few interstitials and the interacting probability will be low. Knowing the concentration of interstitials, the mean distance between them (d) can be determined and one can calculate the mean time associated to the interstitial capture using the Einstein equation:

$$\tau = \frac{d^2}{6D} = \frac{d^2}{6D_0 \exp(-E_m/kT)} = \frac{d^2}{6D_0} \exp(E_m/kT) \quad (6.4)$$

Where the diffusivity D has been expressed by a prefactor D_0 (5 cm²/s) and the migration energy E_m (0.9 eV), both parameters extracted from Ref. [91]. This calculation implies that the clusters are distributed uniformly. It also considers that the distance between an interstitial and a cluster matches the mean distance of interstitials. Although we are doing a simplification of the real scenario, the obtained values can provide an overview about the order of magnitude of the associated time to the processes.

Lastly, the transition between the different configurations are determined by the energy barriers we have already showed.

Figure 6.6 shows the mean time associated to the interstitial emission, the transformation in a different configuration and the capture of an interstitial for compact, chainlike and $\{111\}$ $I_{3 \leq n \leq 5}$ clusters. The temperature has been set to 800 K and we have only included those times lower than 1000 s. The first events to take place are the cluster transformation from $\{111\}$ configuration to other ones. The compact I_3 and chainlike I_4 evolve to other configurations is μs timescale. It should be emphasized that a long time is needed in order to transform the compact I_4 , agreeing with what we have already indicated about this cluster type.

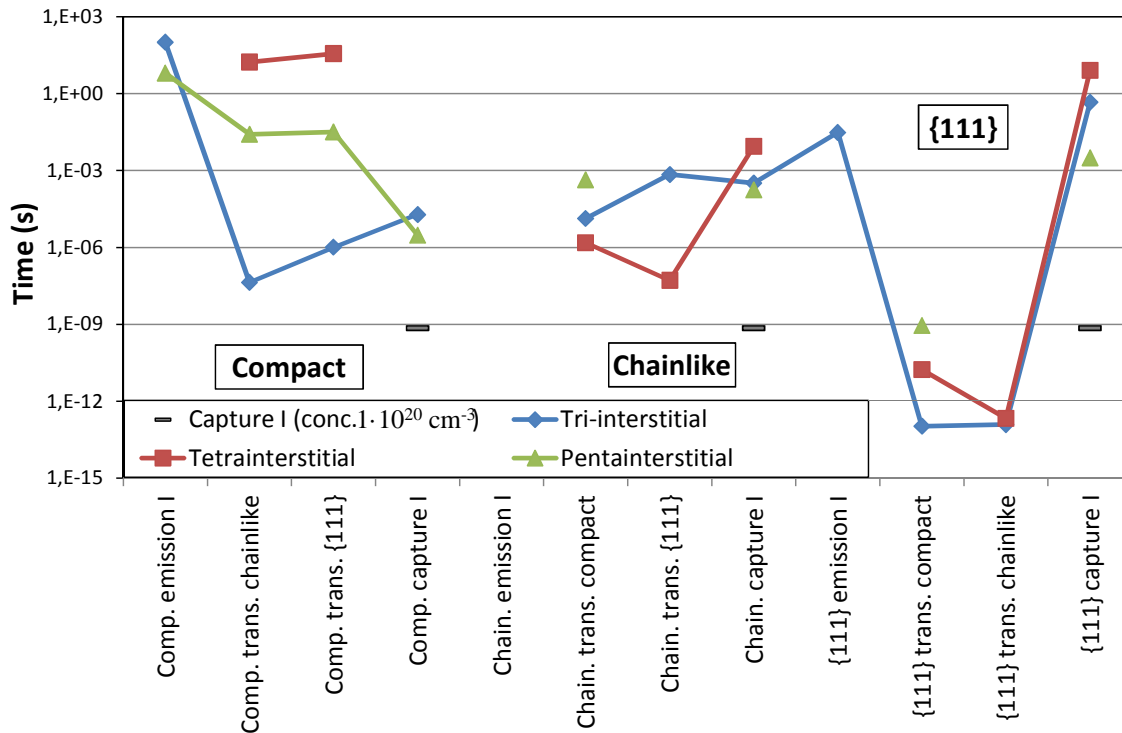


Figure 6.6: Mean time for transformation, emission and capture of an interstitial for chainlike, compact and $\{111\}$ configurations

As it has been showed, the needed times for a cluster in order to grow by capturing an interstitial are high due to the low concentration of interstitials which are surrounding the cluster. This concentration has been calculated from the binding energy of the cluster, considering there are no clusters with other sizes and configurations. The concentration of interstitials can be high after the ion implantation process or at the beginning of the thermal annealing. To illustrate this situation, we have calculated the time associated considering all the clusters are surrounded by a concentration of $5 \cdot 10^{20} \text{ cm}^{-3}$ interstitials, without taking into account the binding energy of the clusters. These conditions lead to a capturing time of 1 ns (grey rectangles in figure 6.6). As a result, this event will occur very frequently. Moreover, the timescale of the capture of an interstitial by $\{111\} I_5$ configuration is similar to the transformation to the compact, being likely to form $\{111\}$ clusters of larger sizes.

6.4 Pressure effect

This section is devoted to the study of what happens in formation enthalpies and energy barriers values of I_3 and I_4 when the pressure is modified, comparing the results with what we have already described for 0 GPa. It should be noted that I_4 results are preliminary.

6.4.1 Summary of simulations

Here we have run the same type of simulations as before but working with cells scaled for different pressures. We have used scaling factors in order to obtain atom coordinates at negative and positive pressure. To so do, instead of using a lattice parameter of 5.432 Å as we did for 0 GPa, we have used 5.492 and 5.380 Å to simulate -3 and 3 GPa, respectively. The computational cost for this section is summarized in tables 6.4 and 6.5. We have not used the {111} configurations as an initial minimum for this analysis. We have mainly focused on the pressure effect on chainlike and compact configurations.

For I_3 size we have run 10 simulations starting from a compact configuration (3 at -3 GPa and 7 at 3 GPa) and 9 simulations beginning from a chainlike configuration (5 at -3 GPa and 2 at 3 GPa). Computing all of them the CPU time reaches almost 1300 h, but we have only seen transitions in 29.41 % of all attempts. In this sense, the transition from compact to chainlike configuration requires a MIN_SIG_BARRIER equals to 0.9 eV to occur at -3 GPa while it only needs 0.6 eV value at 3 GPa.

Number of simulations	Total CPU time (h)	Success rate (%)	Total simulated time (ms)
17	1282.13	29.41 %	134.23

Table 6.4: Summary of simulations which have been run for tri-interstitial at -3 and 3 GPa.

For I_4 size we have run 2 simulations starting from a compact configuration (1 at -3 GPa and 1 at 3 GPa), 16 simulations beginning from a short chainlike configuration (6 at -3 GPa and 10 at 3 GPa) and 2 simulations beginning from a long chainlike configuration (1 at -3 GPa and 1 at 3 GPa). Computing all of them the CPU time is more than 1400 h. However, the success rate is quite low, 15.79 % of all attempts, due to the problems at 3 GPa to see the transition from short chainlike to compact configuration. Moreover, if one

wants to see a transition from short chainlike to compact configuration starting from the first one at -3 GPa, setting MIN_SIG_BARRIER to 2.1 eV is the key. Transitions between long and short chainlike are also possible by setting this limit to 1.5 eV. The transitions at 3 GPa, however, seems to be difficult to achieve because we have not observed them even setting the basin limit to 2.1 eV and restarting several simulations.

Number of simulations	Total CPU time (h)	Success rate (%)	Total simulated time (ms)
20	1435.02	15.79 %	34412.01

Table 6.5: Summary of simulations which have been run for tetrainterstitial at -3 and 3 GPa.

In any case, it should be noted that unknown configurations have appeared during this study. This is the main reason why the transitions between compact and chainlike configurations have been difficult. The system has got trapped in intermediate configurations in numerous attempts. Hence, the statistics of this section are even lower than what we have already shown in the previous chapter.

6.4.2 Formation enthalpies and energy barriers

The energetics properties of compact and chainlike configurations for I_3 and I_4 are analysed at -3 and 3 GPa, following the same analysis as we did for 0 GPa. We can know in advance that the formation enthalpies are going to decrease for negative pressures whereas they are going to increase for positive ones due to the variation in interatomic distances.

Firstly, we have determined that compact configuration is more energetic than chainlike configuration for tri-interstitial size at both pressures, as can be seen in figure 6.7. Regarding the energy barriers from compact to chainlike configuration, we have seen few transitions: two at -3 GPa and three at 3 GPa. In this sense, they present a visible difference because the barrier is higher when the pressure is negative. This idea is displayed in the figure 6.8, where it can be seen that the energy barrier decreases with the pressure. However, it should be emphasized that the statistics are poor and we have linked the lowest energy barriers reported in this graph.

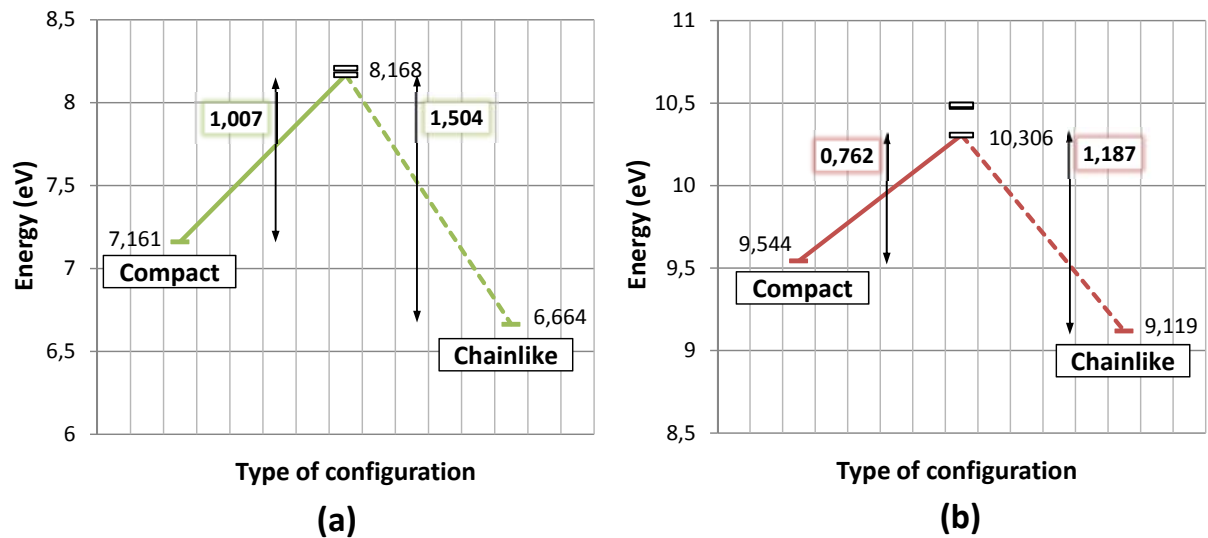


Figure 6.7: Transition path between compact and chainlike I_3 configurations showing the formation enthalpies and all energy barriers found at -3 GPa (a) and 3 GPa (b).

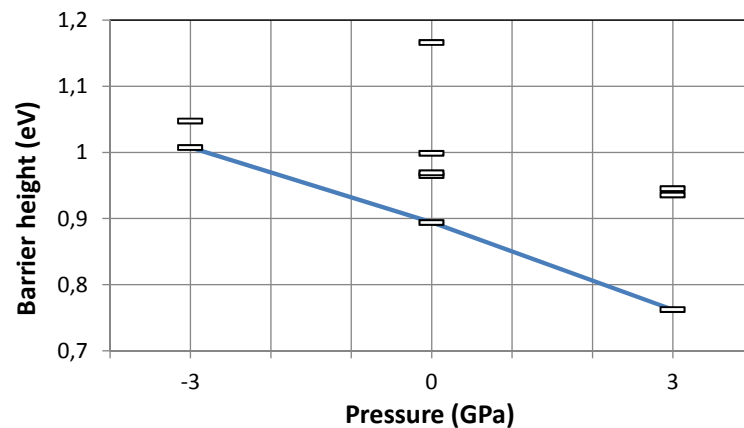


Figure 6.8: Energy barriers from compact to chainlike I_3 configurations as a function of pressure.

Secondly, we have determined that the compact configuration for I_4 is the most stable one at -3 and 3 GPa. Figure 6.9 (a) shows the energy barrier diagram for -3 GPa and (b) represents the comparison with the barriers at 0 GPa. As we have already mentioned in this section, we have not got transitions at 3 GPa. In any case, we can see that the tendency of I_3 is repeated because the barrier is higher when the pressure is negative. However, again it should be emphasized that the statistics are poor.

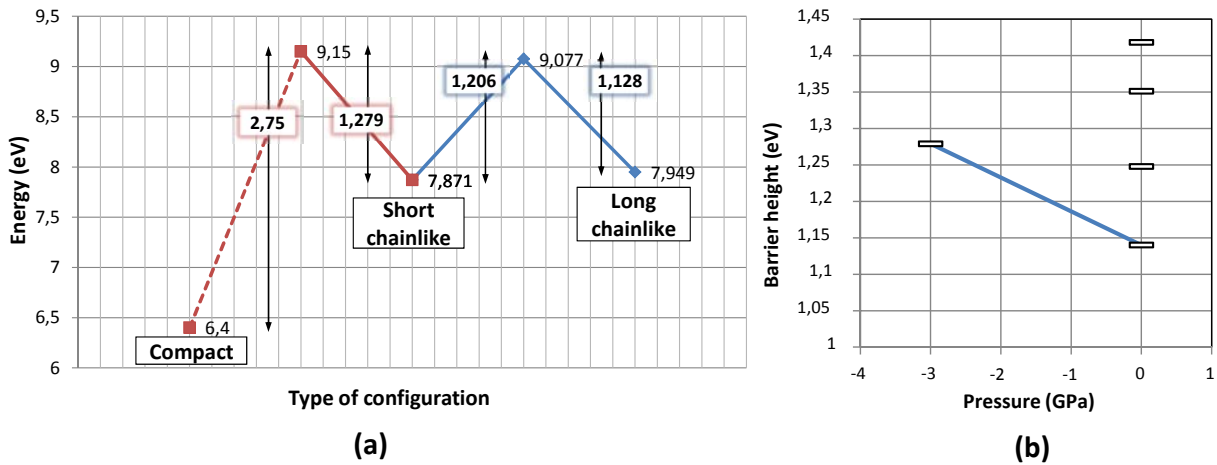


Figure 6.9: Transition path between compact and chainlike I_3 configurations indicating the formation enthalpies and all energy barriers found at -3 GPa (a) and 0 GPa (b).

Other relevant point to be analysed is the evolution of the energy difference between the chainlike and the compact configurations with the cluster size for the pressures we have studied. Figure 6.10 illustrates this concept. For those cases where long and short chainlike configurations are possible it has been showed the difference with each one. Although I_3 and I_4 long chainlike present almost the same difference, I_4 short chainlike is modified by the pressure. Then, the difference seems to be higher for negative pressures, meaning that the transition would be harder to achieve.

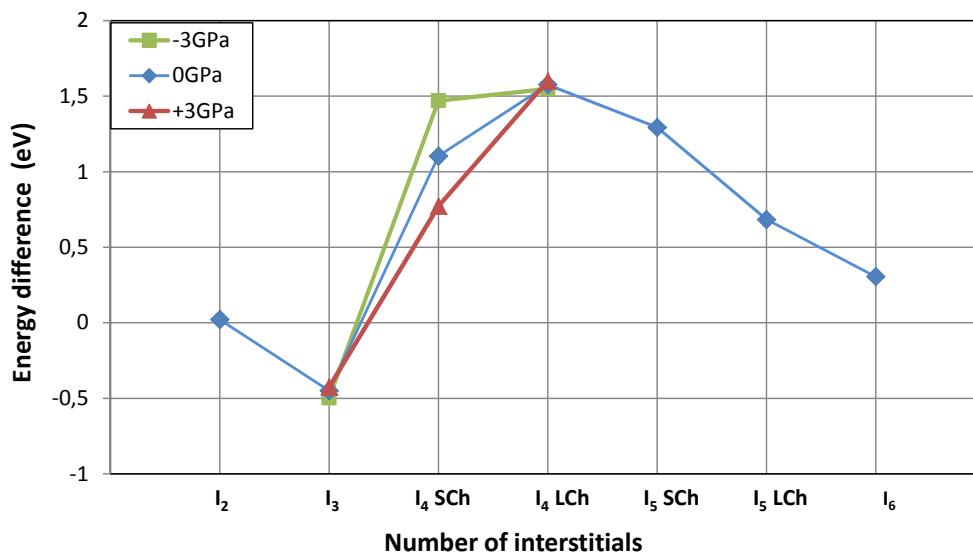


Figure 6.10: Evolution of the energy difference between compact and chainlike configuration for all cluster sizes and pressures we have studied.

6.5 Conclusions

We have analysed how the formation enthalpies and energy barriers of small interstitial cluster configurations change with the size and how they are influenced by changes in pressure.

Our results indicate that compact tetra-interstitial is very stable due to its low formation enthalpy and high energy barriers to go away from it. In fact, long times would be needed to dissolve this configuration in an annealing at 800 K, playing a key role on the growing mechanisms of small clusters. $\{111\}$ configuration, however, seems to be a quite energetic cluster, evolving quickly into other configurations because of their low energy barriers. In any case, significant barriers have been seen to abandon high energy configuration. This would increase their mean lifetime and delay their structural transformation in other configurations. Regarding the pressure analysis, we can conclude that the higher the pressure is, the lower the barriers would be, according to our small amount of data.

Chapter 7

Conclusions and future research lines

*If I have seen further, it is by
standing on the shoulders of giants.
~ Isaac Newton ~*

THE MOTIVATION OF THIS WORK has been to increase the knowledge of small interstitials clusters in crystalline Si, due to the key role they play on the formation of extended defects. To do so, we have identified several geometrical configurations of the clusters and analysed their energetics and dynamics, determining their formation enthalpies and the energy barriers of the associated transitions. The reported energy barriers can be used as input parameters for the development of more accurate damage generation and accumulation models for the process simulators of semiconductor industry. The analysis of their energetic landscape and their configurational transitions allows us to elucidate new growing mechanism for small clusters interstitial clusters.

The results of this master dissertation have been presented to the national science competition *Certamen Universitario Arquímedes 2016*, sponsored by the Spanish Ministry of Education.

7.1 Conclusions

We have determined the energy barriers which control the transition of compact, chainlike and $\{111\}$ configurations in clusters from 2 to 6 interstitials in crystalline Si.

Our data show that the compact tetrainterstitial is very stable and high energy barriers are required to evolve from it. The formation of compact tetrainterstitial clusters could slow down defect evolution and may act as an energy well for other configurations. This explains why this small defect is still observed when large extended defects are already formed. Symmetrical clusters in $\{111\}$ planes, however, are more energetic and they quickly evolve to other configurations. Although these defects are considered as the precursors of $\{111\}$ dislocation loops, our study indicates that at least for sizes up to 6 interstitials this configuration is not stable. Bigger clusters on $\{111\}$ planes may be more energetically favourable.

Significant barriers have been observed even for the structural transformation of high energy configurations, which increases their mean lifetime. A lattice in equilibrium conditions which contains interstitial clusters will be in the minimum energy state. The clusters population will be formed by those sizes and configurations which present the lowest formation enthalpies, with high energy barriers for their structural transformation. Hence, the evolution of defects would be limited by those very stable configurations.

Traditional defect evolution models reproduce this situation but they do not take into account that several topological configurations of clusters are possible. Our simulations indicate that even those high energy configurations have non-negligible energy barriers to evolve to lower energy ones. It implies that the system can stay some time in non-equilibrium scenarios. This opens new paths for the growing of small interstitial clusters, since these unstable configurations may grow by capturing diffusing interstitials.

7.2 Future research lines

This work can be continued following one of the next points:

- **Study bigger cluster sizes.** This work has focused on cluster whose sizes range from 2 to 6 interstitials. The analysis of bigger ones could reveal that other configurations are favoured.
- **Make more statistics.** The number of energy barriers we have presented has not been so long for certain situations. Widening this data can provide more accurate results.
- **Increase statistics.** The number of energy barriers we have obtained is small for certain situations. Widening these data can provide more accurate results.
- **Implement other interatomic potential.** The results can be checked by using other interatomic potentials apart from Tersoff 3, such as SW or EDIP.
- **Study different materials.** The methodology we have developed can be easily used to study defects in other semiconductors, like Ge, or binary materials such as SiGe or III-V.

Appendix A

Library of defects configurations

THIS APPENDIX PRESENTS the views in all directions for the most visited configurations during our simulations for each cluster size ($I_{2 \leq n \leq 6}$, being n the excess of interstitials) at 0 GPa. As it has been already mentioned, we have built a database which contains all the known configurations for each defect, characterizing the atoms geometry by discretizing in formation enthalpy values (see chapter 3 for more information about the procedure).

All the attached pictures have a common structure. From left to right, the first four snapshots show X axis projections ($[0\ 0\ -1]$, $[0\ 1\ -1]$, $[0\ 1\ 0]$, $[0\ 1\ 1]$), the next three snapshots show Y axis projections ($[-1\ 0\ -1]$, $[-1\ 0\ 0]$, $[-1\ 0\ 1]$) and the final four show projections along $\langle 111 \rangle$ axis ($[-1\ 1\ -1]$, $[1\ 1\ -1]$, $[-1\ 1\ -1]$, $[-1\ -1\ -1]$). The text box positioned on the bottom right corner points out the exact simulation step in which the snapshots have been taken and other parameters such as the temperature (TEMP.), the number of displaced atoms (DA) and the number of empty spaces (ES), the formation enthalpy (EF) and if we considered the defects as a group (GROUPED).

It should be taken into account that we are going focus on compact, chainlike and $\{111\}$ configurations for each cluster size due to their interest in this work, avoiding other types of defects which we have also found. In addition, the notation we have used is the following: CoX for the compact configurations and ChX for the chainlike ones, where X means their position when we sort all the same type of configurations by their energy values (from low values to high values). Hence, Co1 and Ch1 will be the lowest energy minima for the compact and the chainlike configurations, respectively.

Firstly we are going to detail I_2 clusters. According to our data, we have identified 14 compact and 9 chainlike energy minima. The most frequent states for the compact configuration are Co1 (11.80%), Co2 (42.43%), Co3 (11.18%), Co4 (16.22%) and Co8 (11.44%). Similarly, we have determined Ch1 (36.73%), Ch3 (18.37%) and Ch4 (32.14%) are the most visited energy minima for the chainlike configuration.

Secondly, I_3 clusters are going to be described. We have found 20 compact, 18 chainlike and 1 {111} cluster energy minima. Looking between them separately, Co1 (8.90%), Co2 (22.93%) and Co9 (18.97%) are the most likely energy states for the compact configuration and Ch5 (12.52%) and Ch10 (14.30%) for the chainlike configuration. In any case, we also show Ch1 (1.61%) due to their almost perfect symmetry. It should be noted the percentages are relative to each type of clustering.

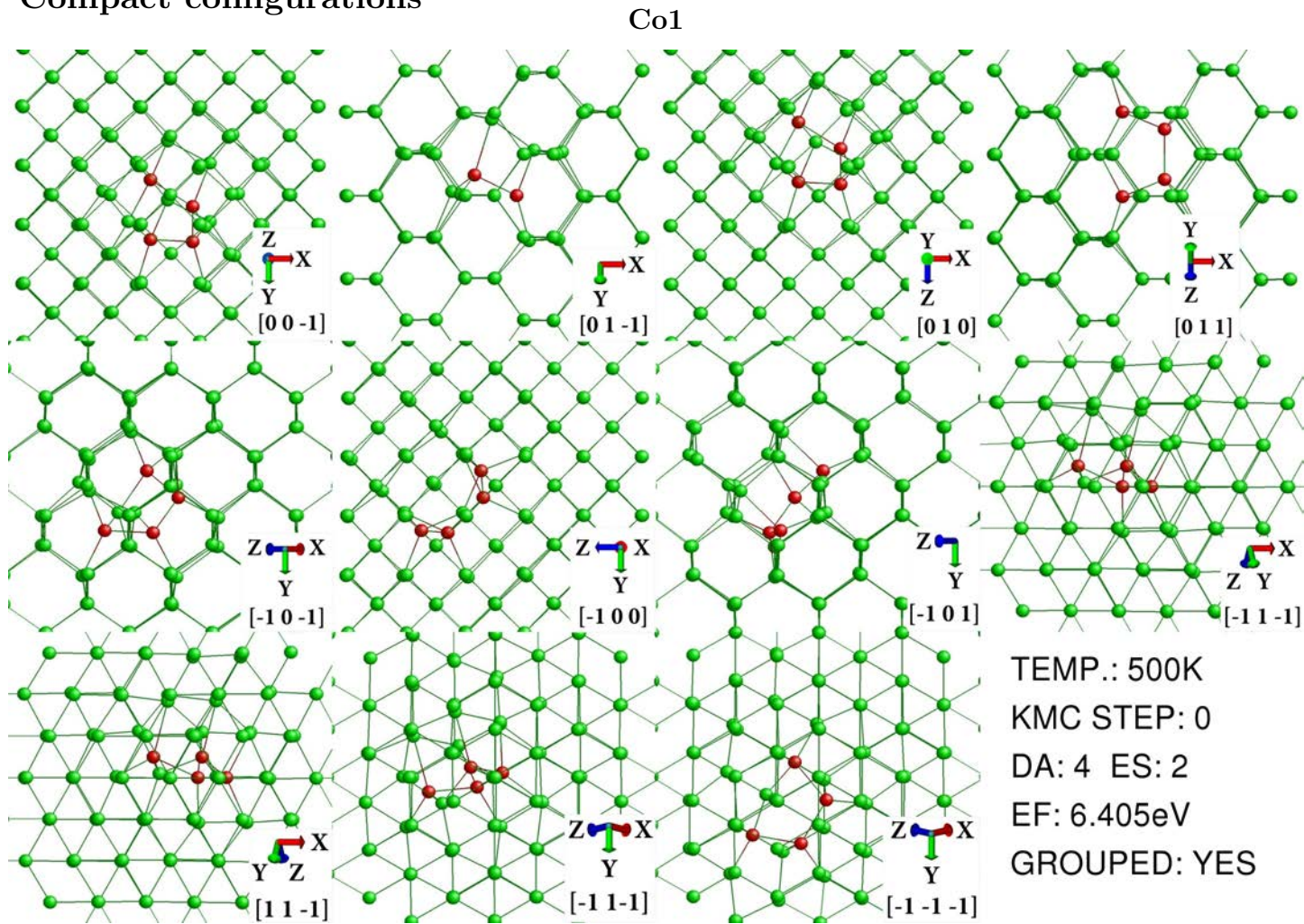
Thirdly, we are going to detail I_4 clusters. It should be kept in mind that, in this case and for I_5 cluster size, we have two possible configurations for the chainlike cluster, one more elongated than the other. The way we are going to call them are SChX (short chainlike) and LChX (long chainlike). According to our data, we have identified 4 compact, 5 short chainlike, 9 long chainlike and 1 {111} energy minima. The most frequent energy minima for the compact configuration are Co1 (25.63%) and Co2 (74.37%). Likewise, we have determined SCh1 (58.65%) and SCh2 (34.42%) are the most visited energy minima for the short chainlike configuration and LCh1 (31.74%), LCh3 (34.13%) and LCh5 (16.39%) are the most visited energy minima for the long chainlike configuration.

Fourthly, following the previous naming we are going to describe I_5 clusters. We have observed 16 compact and 10 short chainlike, 26 long chainlike energy minima and 1 {111}. Between this large number of variations, the relative frequency of the most frequent ones is the following: Co1 (43.29%), Co3 (17.41%) and Co7 (25.89%); SCh1 (16.45%), SCh6 (10.36%), and SCh12 (10.86%); LCh1 (24.93%) and LCh2 (36.08%).

And fifthly, I_6 clusters are going to be described. We have labelled 7 compact, 6 chainlike and 8 {111} energy minima. Co1 (47.84%) and Co2 (31.83%) are the most likely energy minima for the compact configuration, Ch1 (34.08%) and Ch3 (32.96%) for the chainlike configuration, and {111}_1 (33.54%) and {111}_2 (28.8%) for the {111} configuration.

A.1 I_2

A.1.1 Compact configurations

Figure A.1: First minimum energy configuration for compact I_2

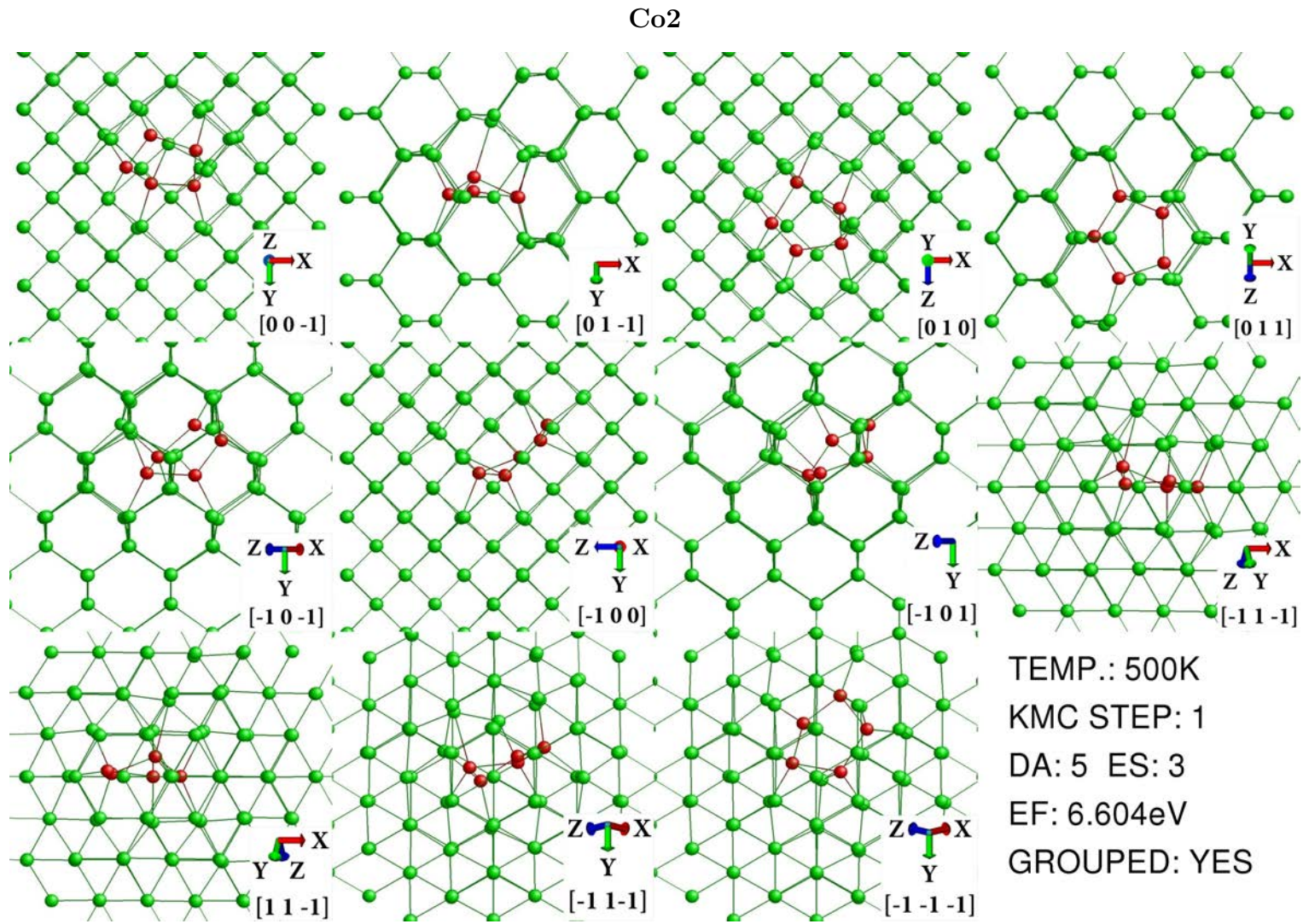


Figure A.2: Second minimum energy configuration for compact I_2

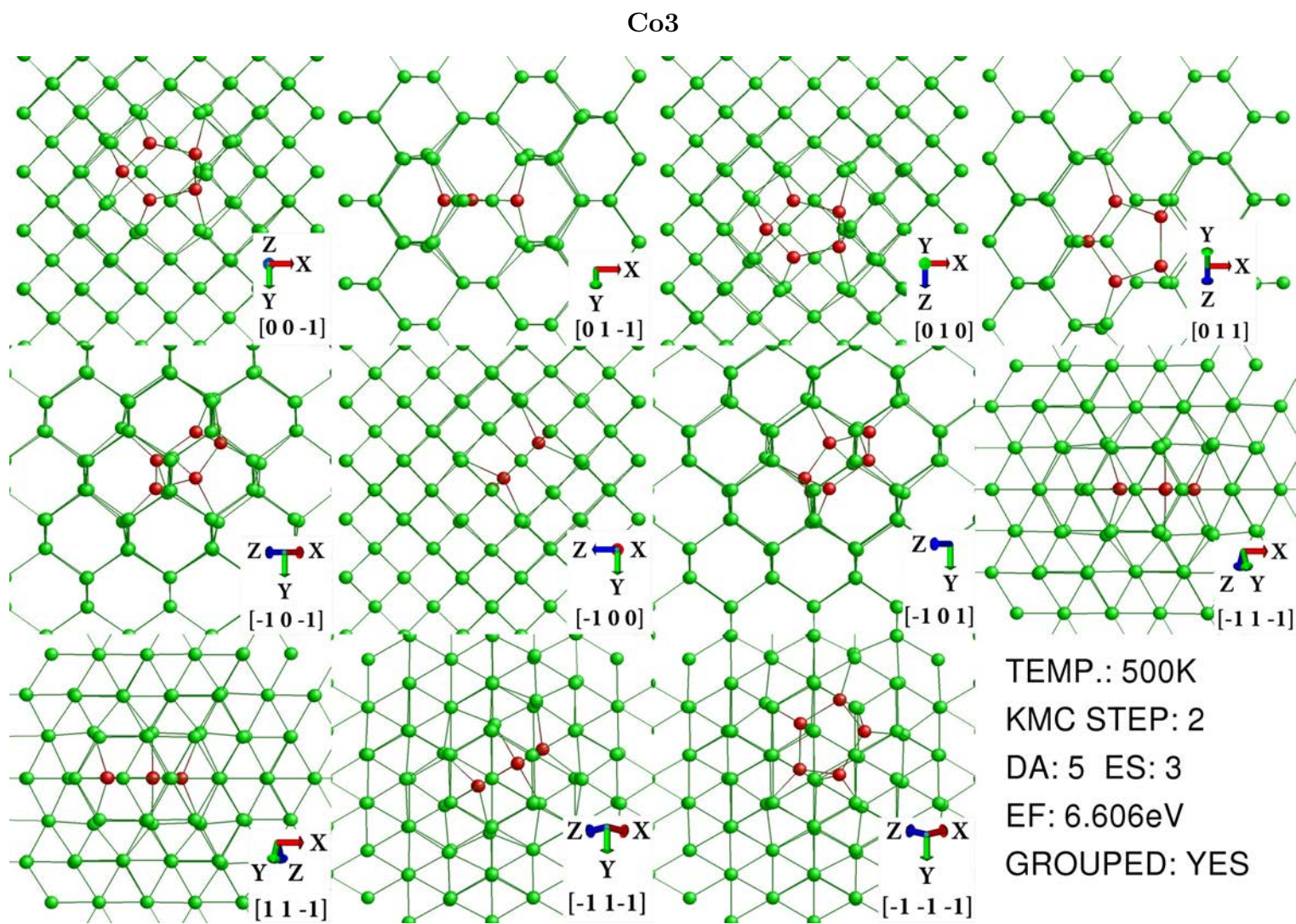


Figure A.3: Third minimum energy configuration for compact I_2

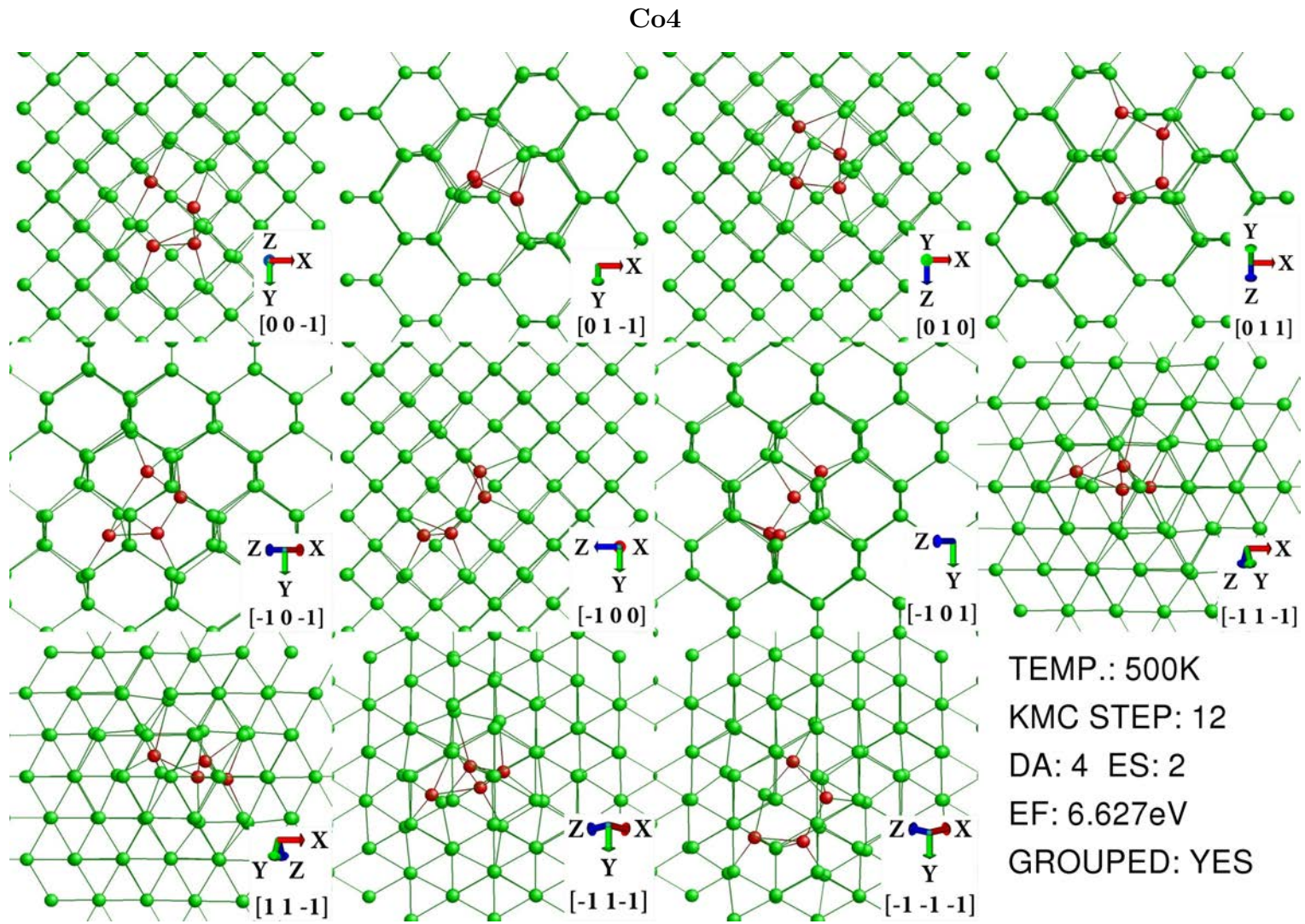
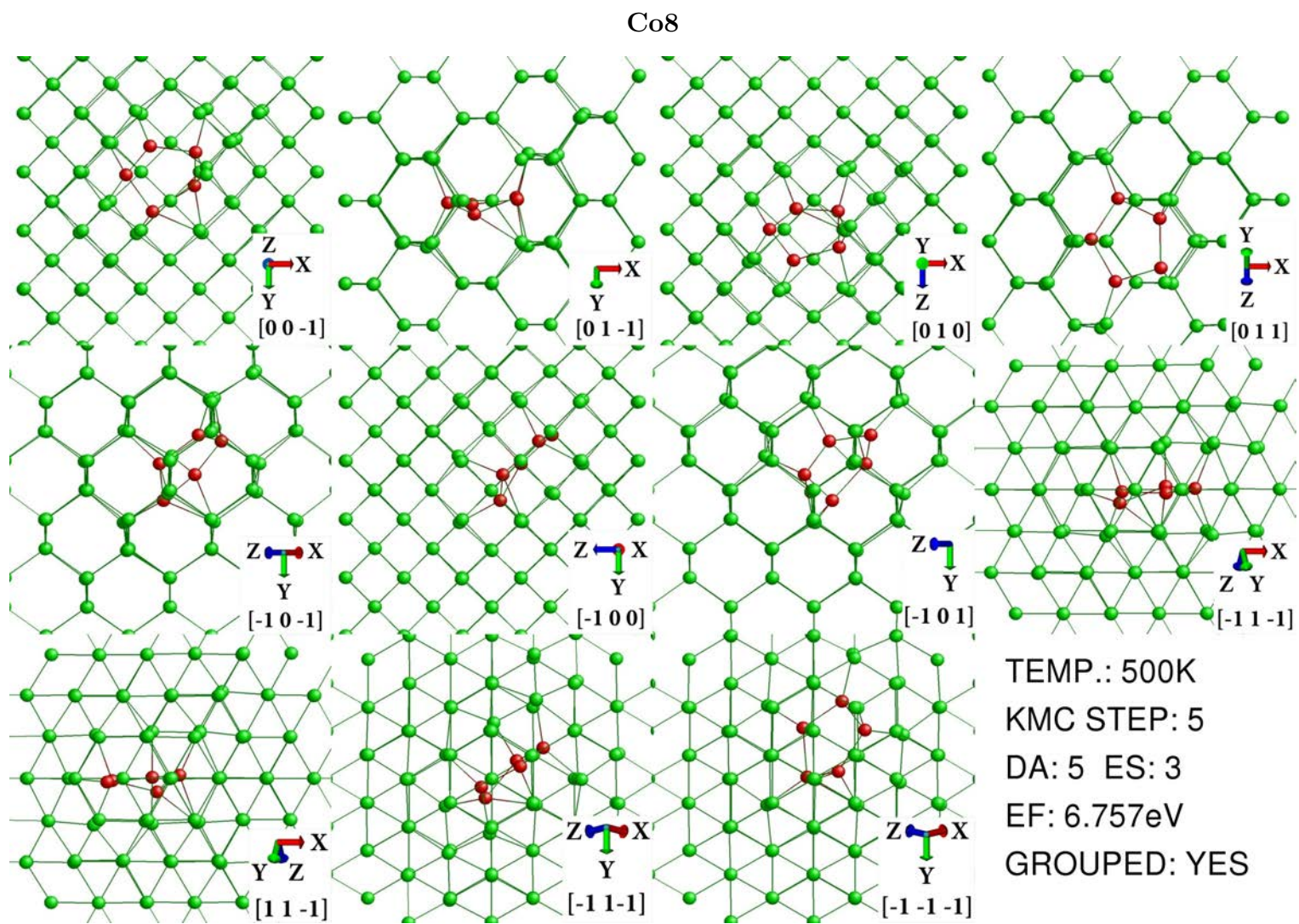
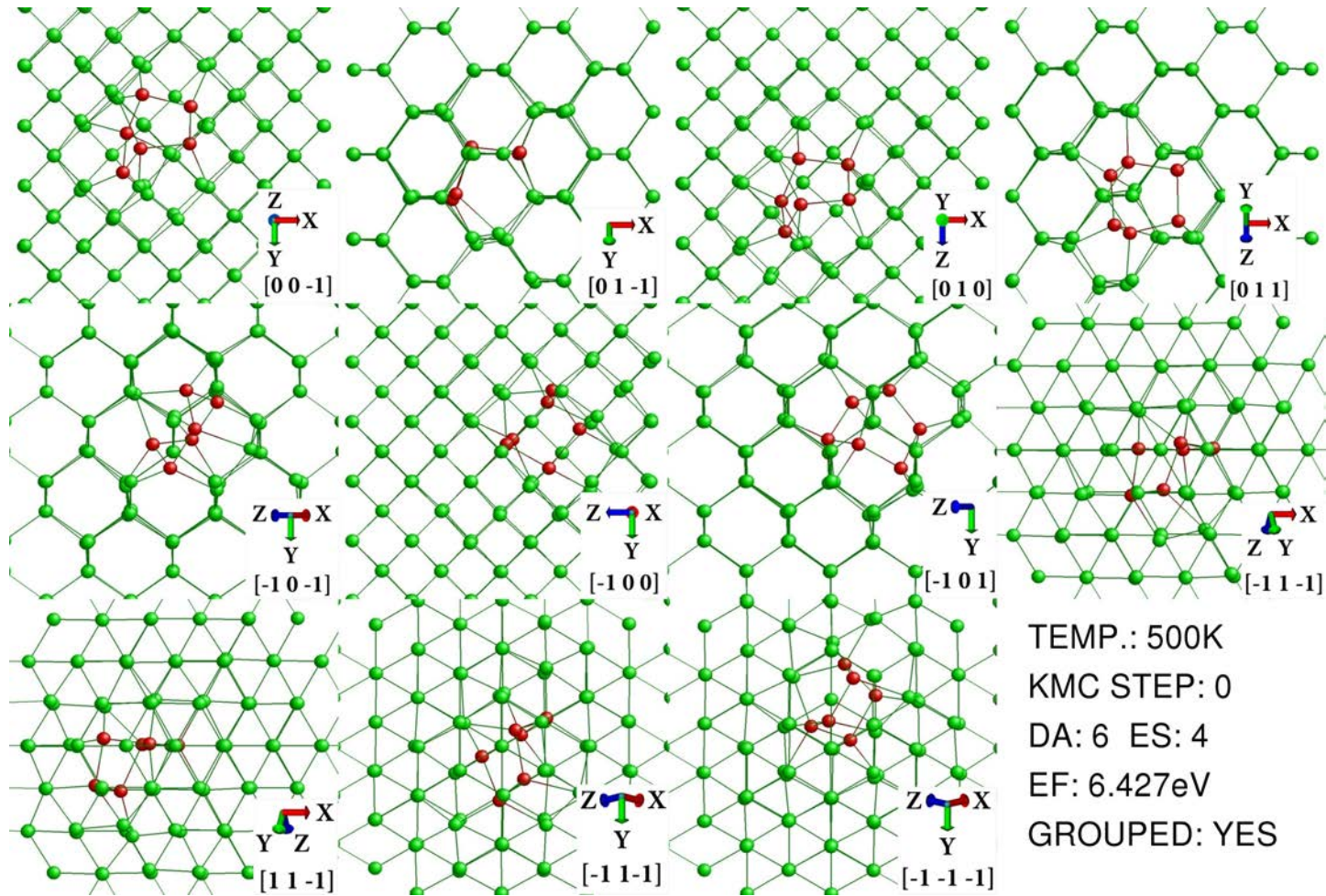


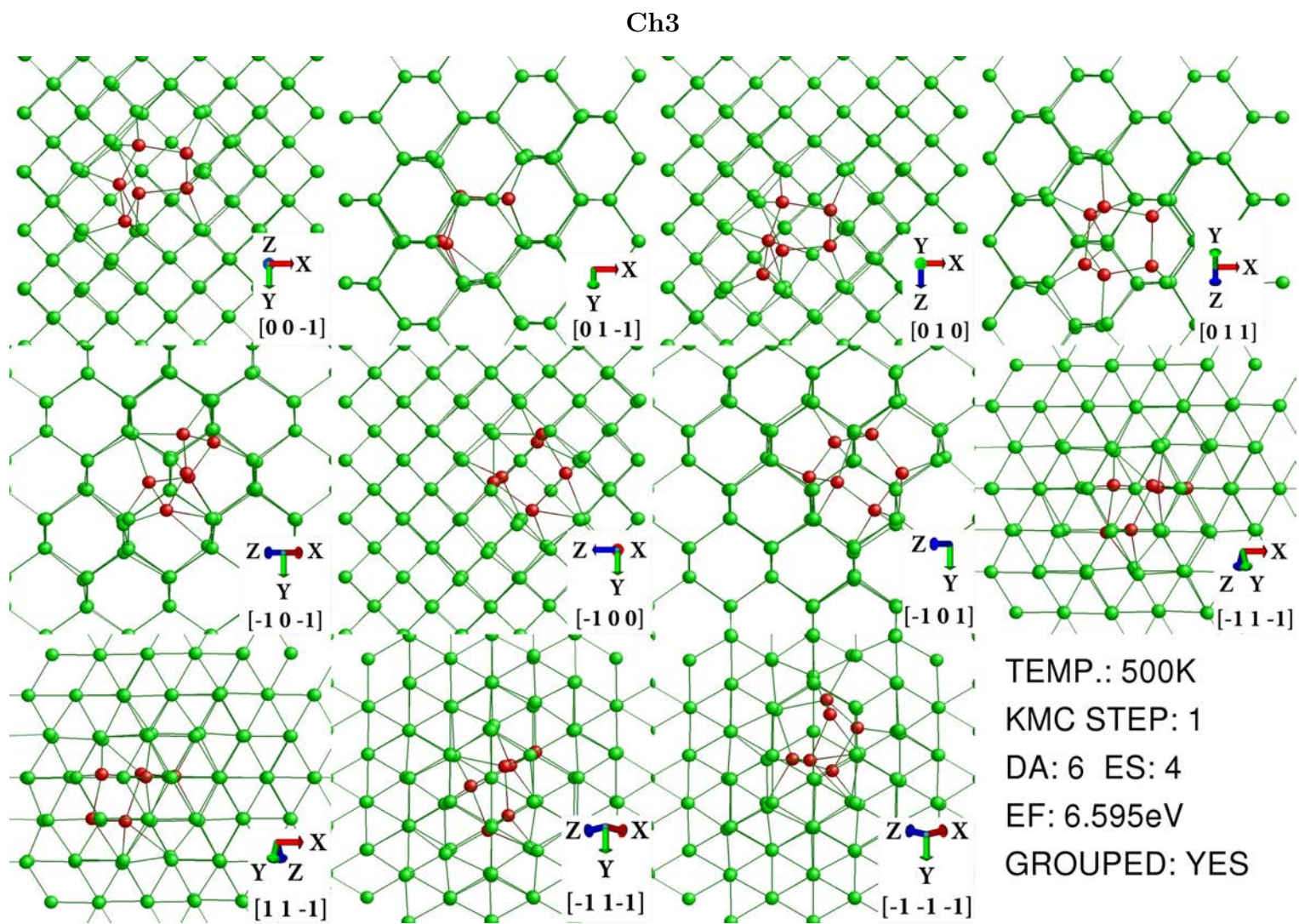
Figure A.4: Fourth minimum energy configuration for compact I_2

Figure A.5: Eighth minimum energy configuration for compact I_2

A.1.2 Chainlike configurations

Ch1

Figure A.6: First minimum energy configuration for chainlike I_2

Figure A.7: Third minimum energy configuration for chainlike I_2

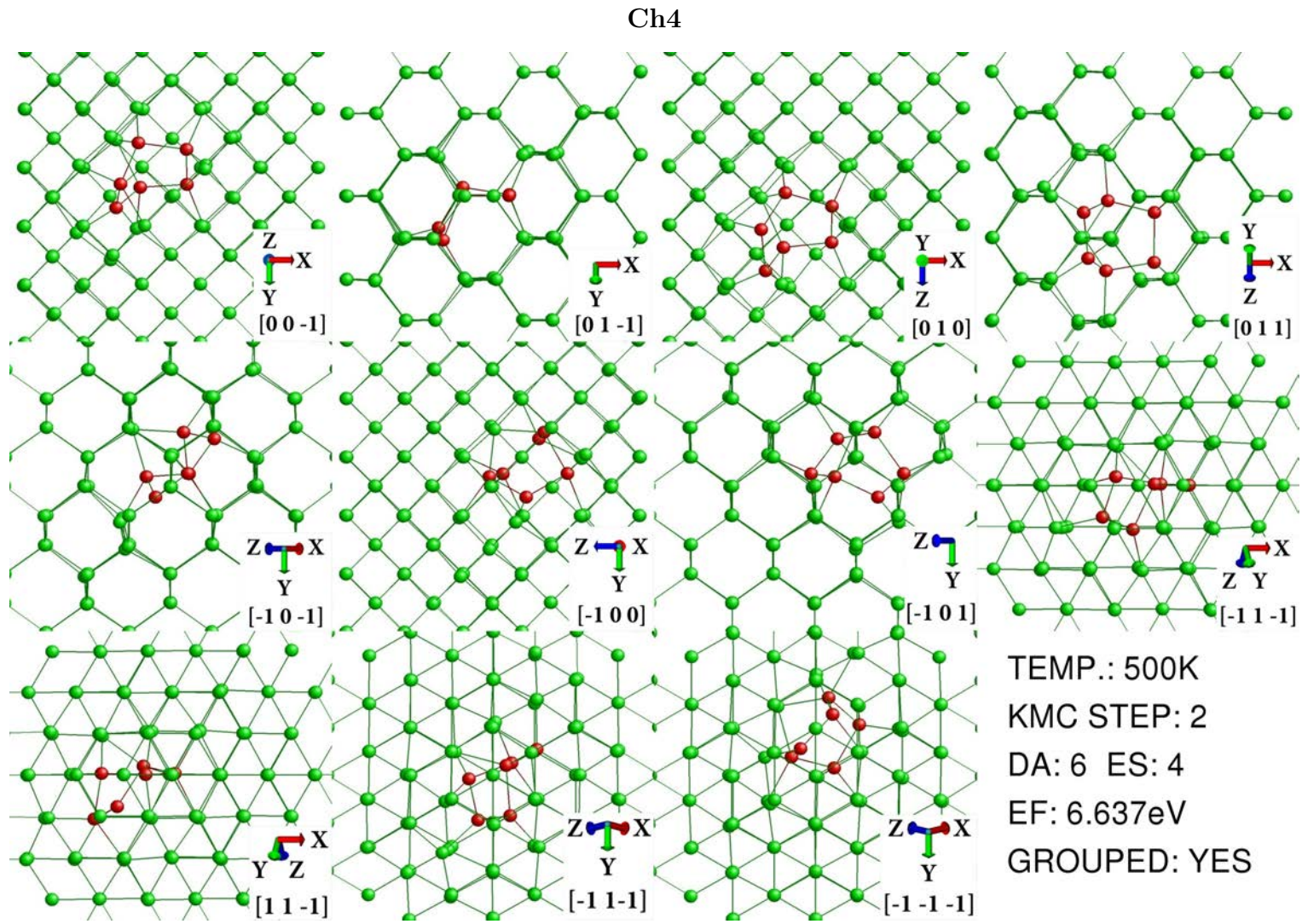
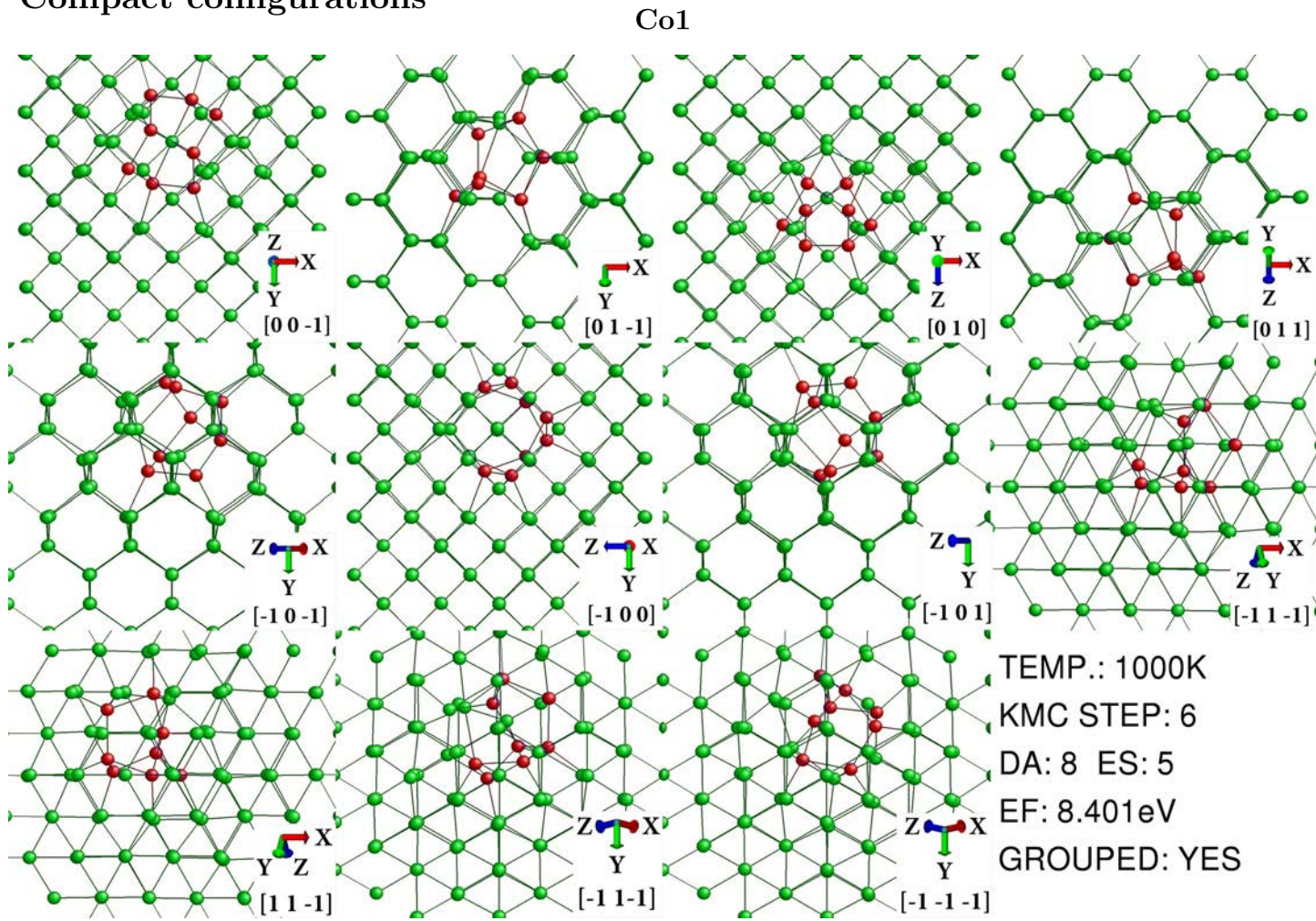


Figure A.8: Fourth minimum energy configuration for chainlike I_2

A.2 I_3

A.2.1 Compact configurations

Figure A.9: First minimum energy configuration for compact I_3

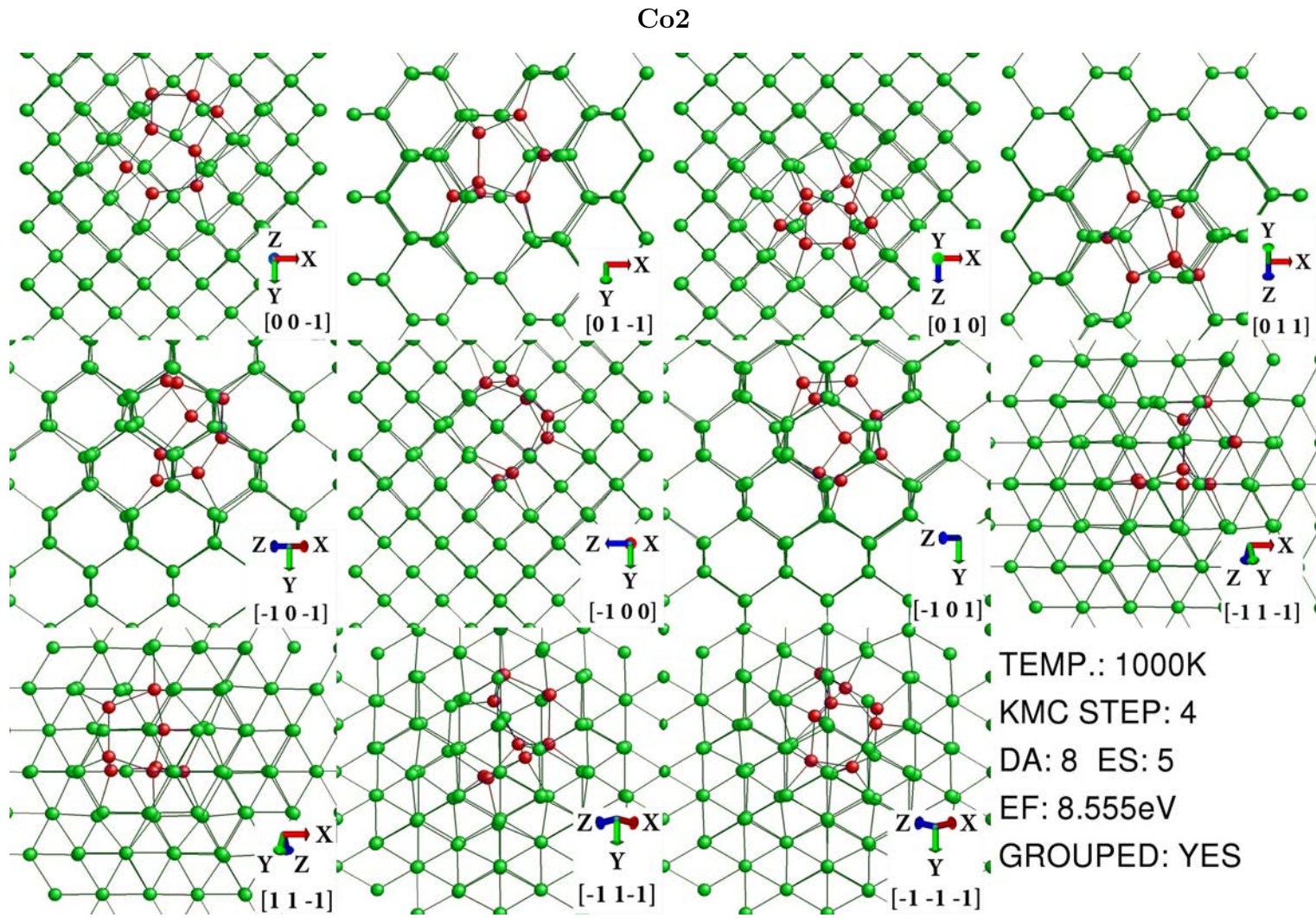


Figure A.10: Second minimum energy configuration for compact I_3

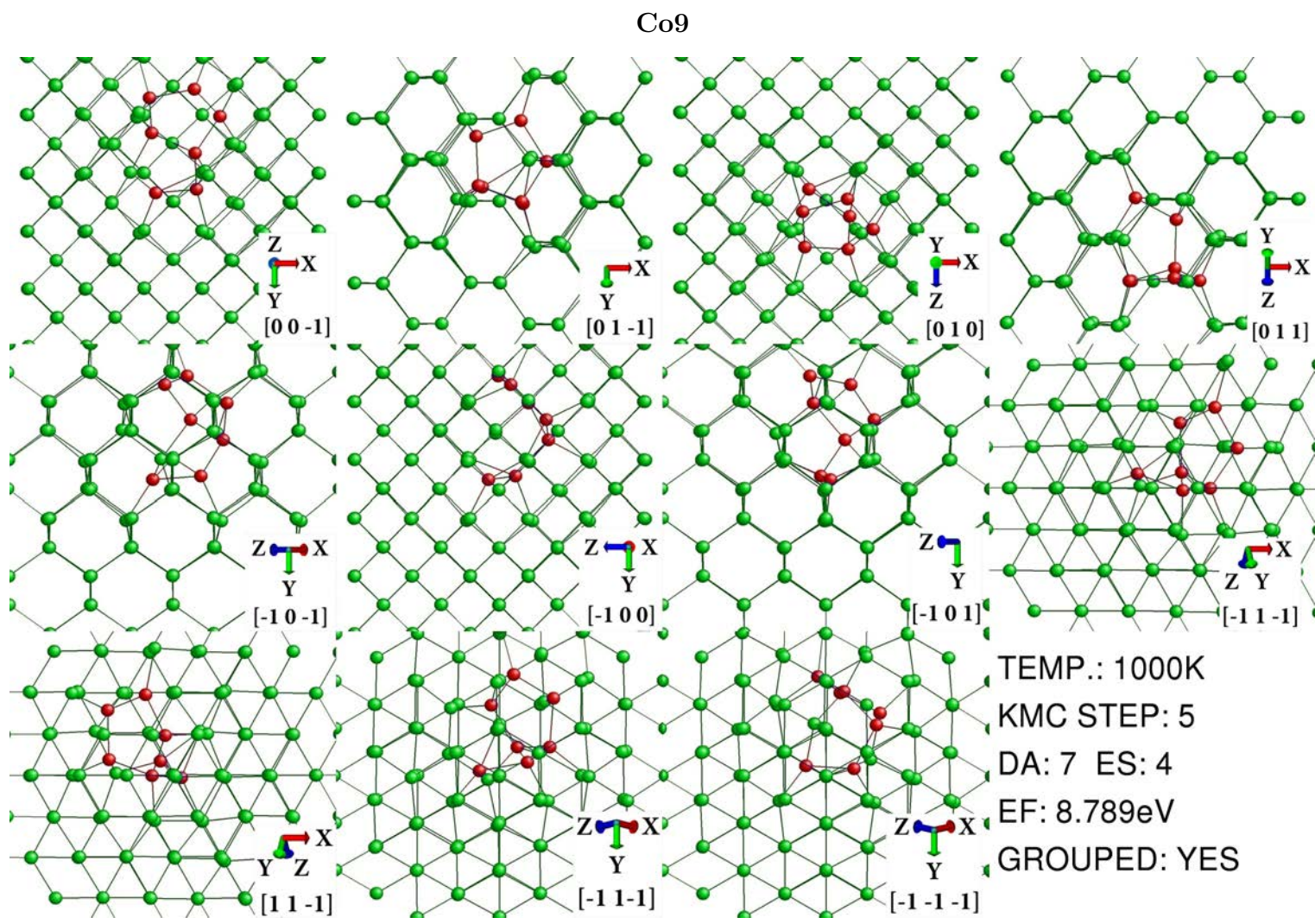
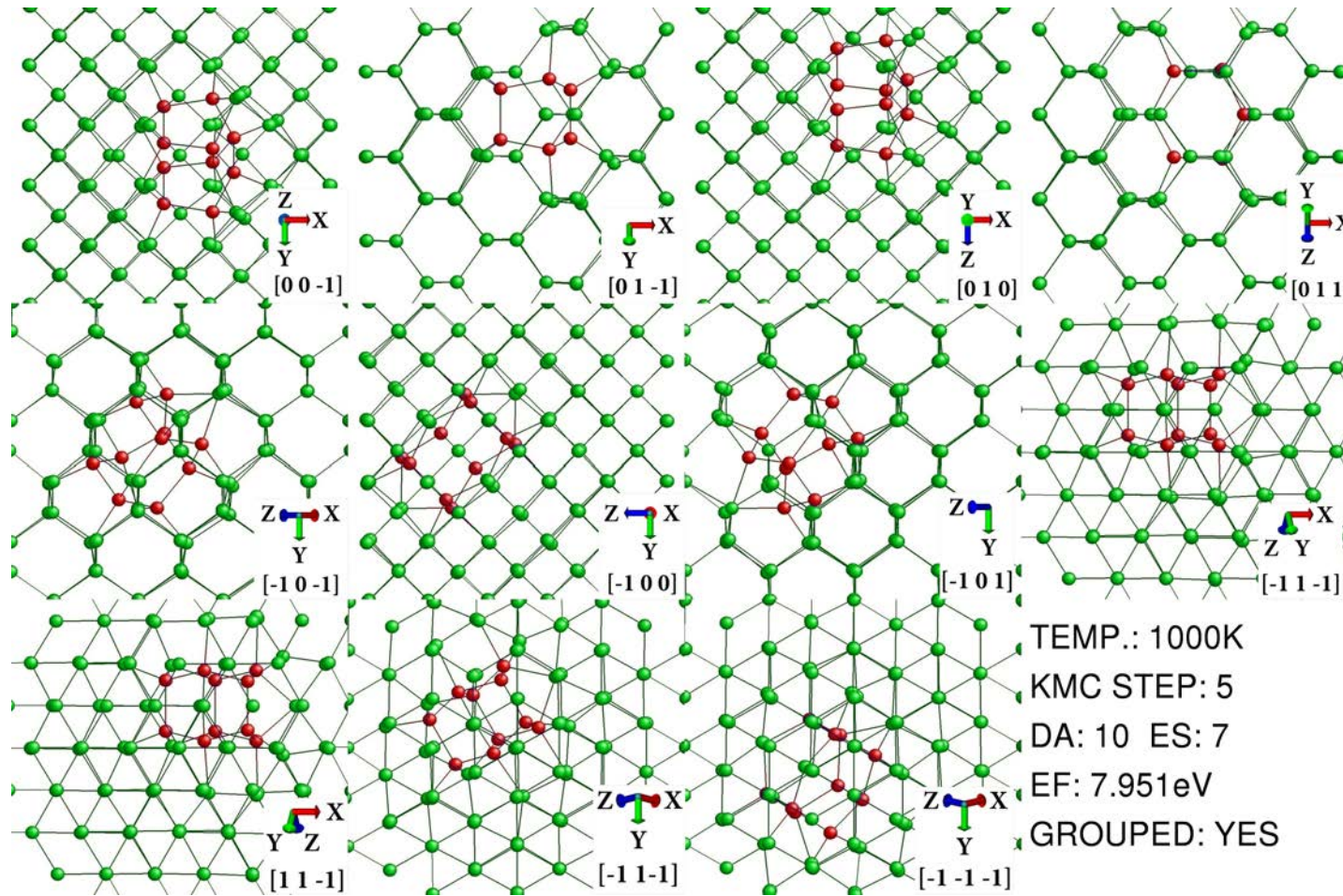
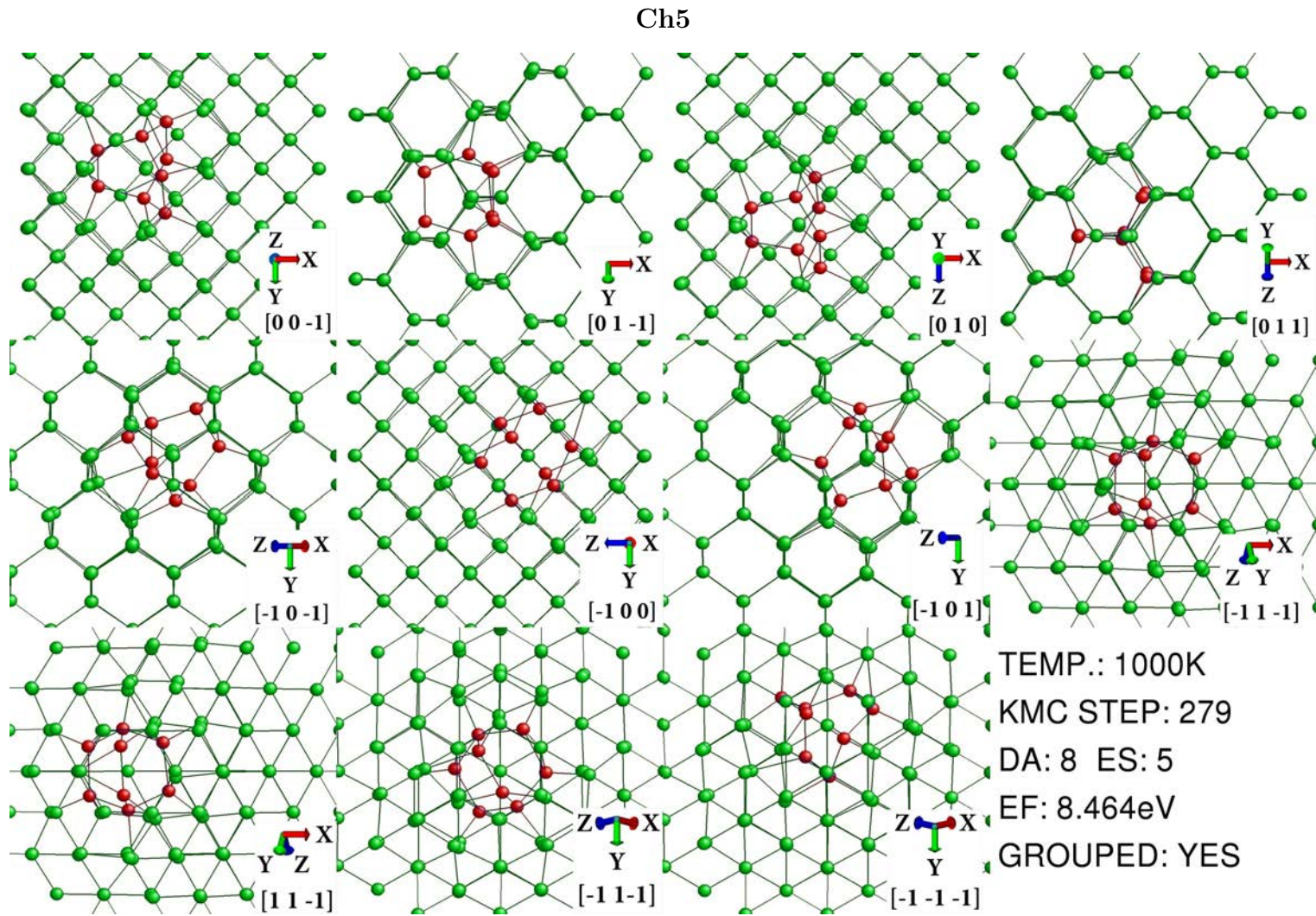


Figure A.11: Ninth minimum energy configuration for compact I_3

A.2.2 Chainlike configurations

Ch1

Figure A.12: First minimum energy configuration for chainlike I_3

Figure A.13: Fifth minimum energy configuration for chainlike I_3

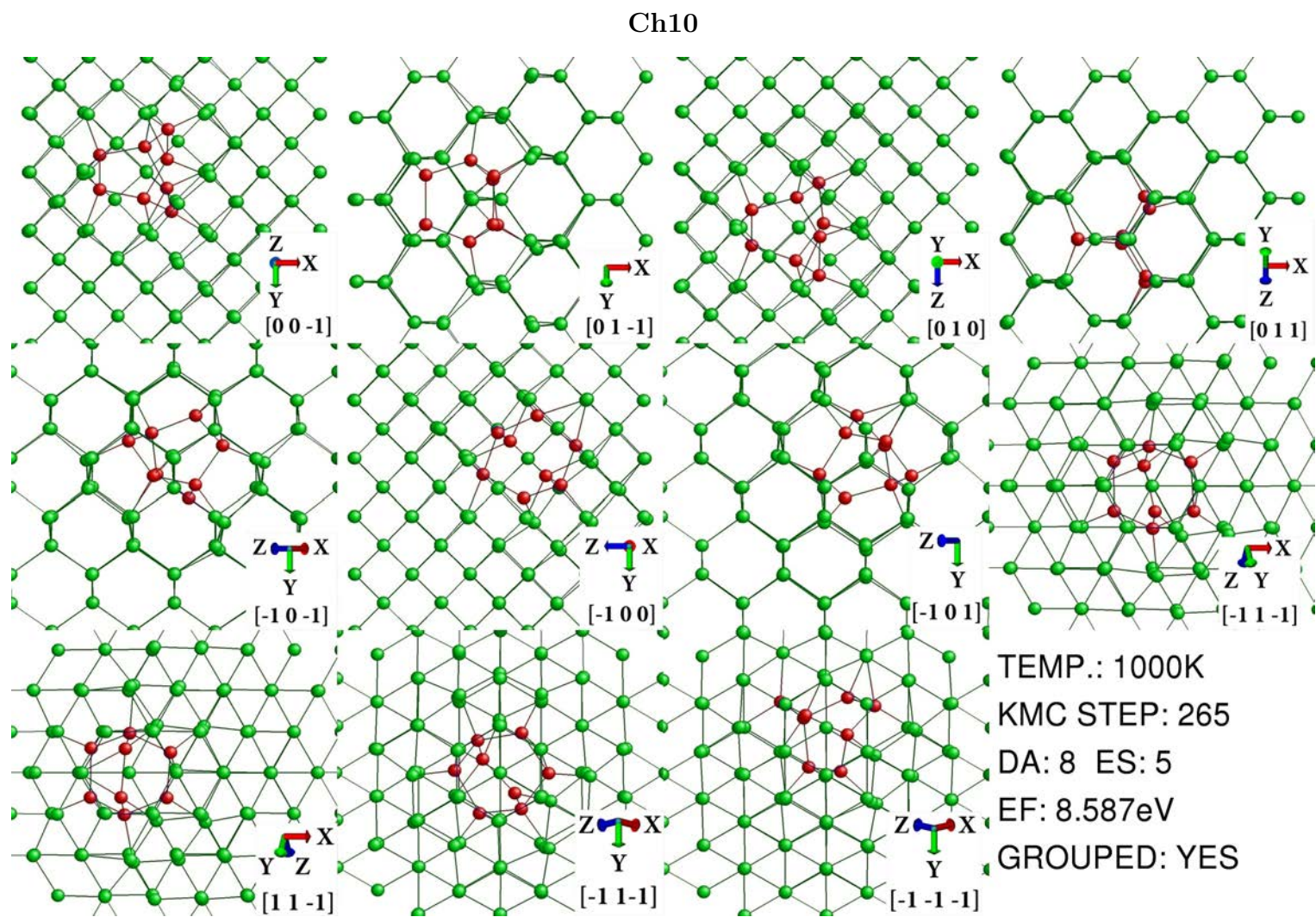
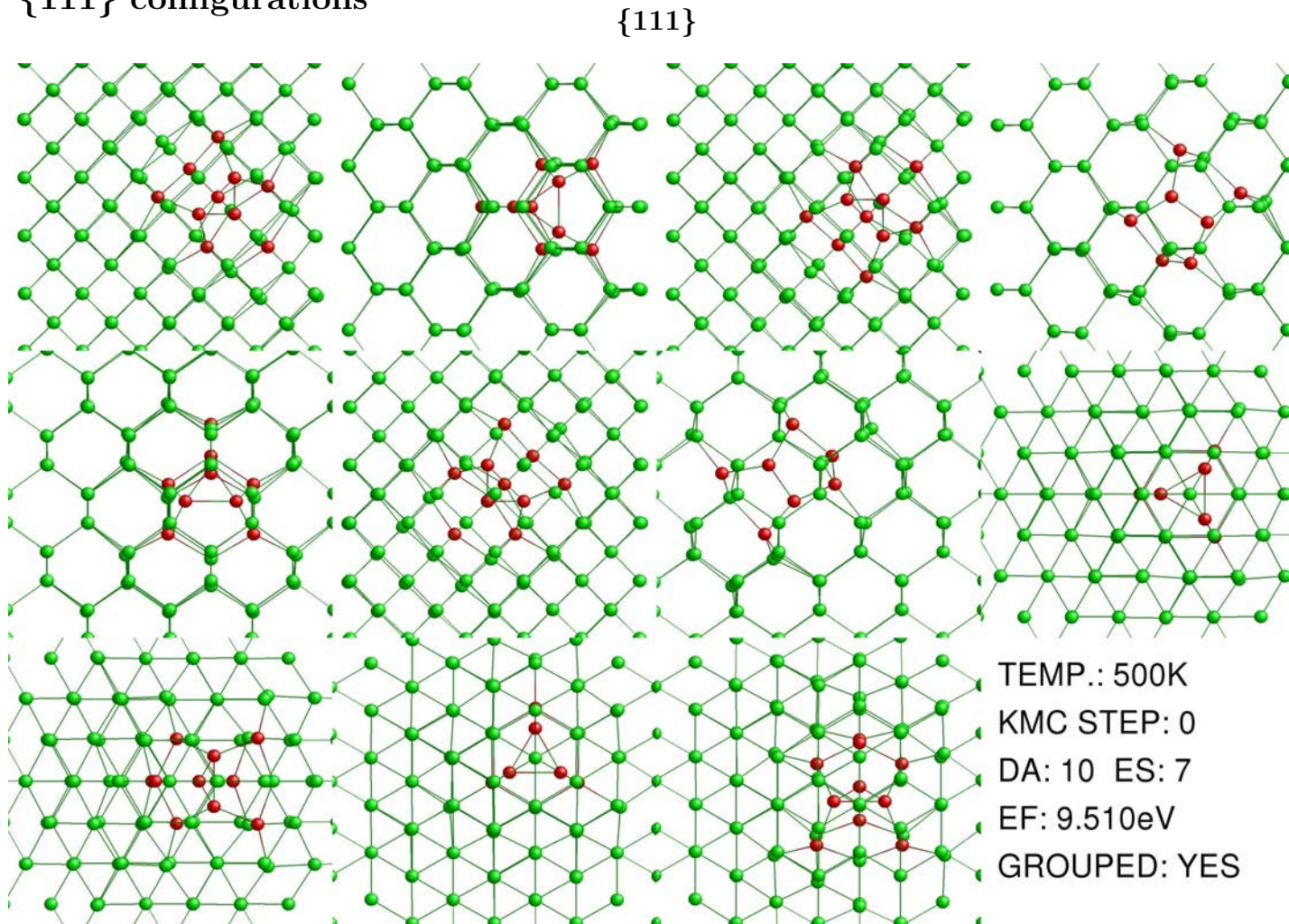
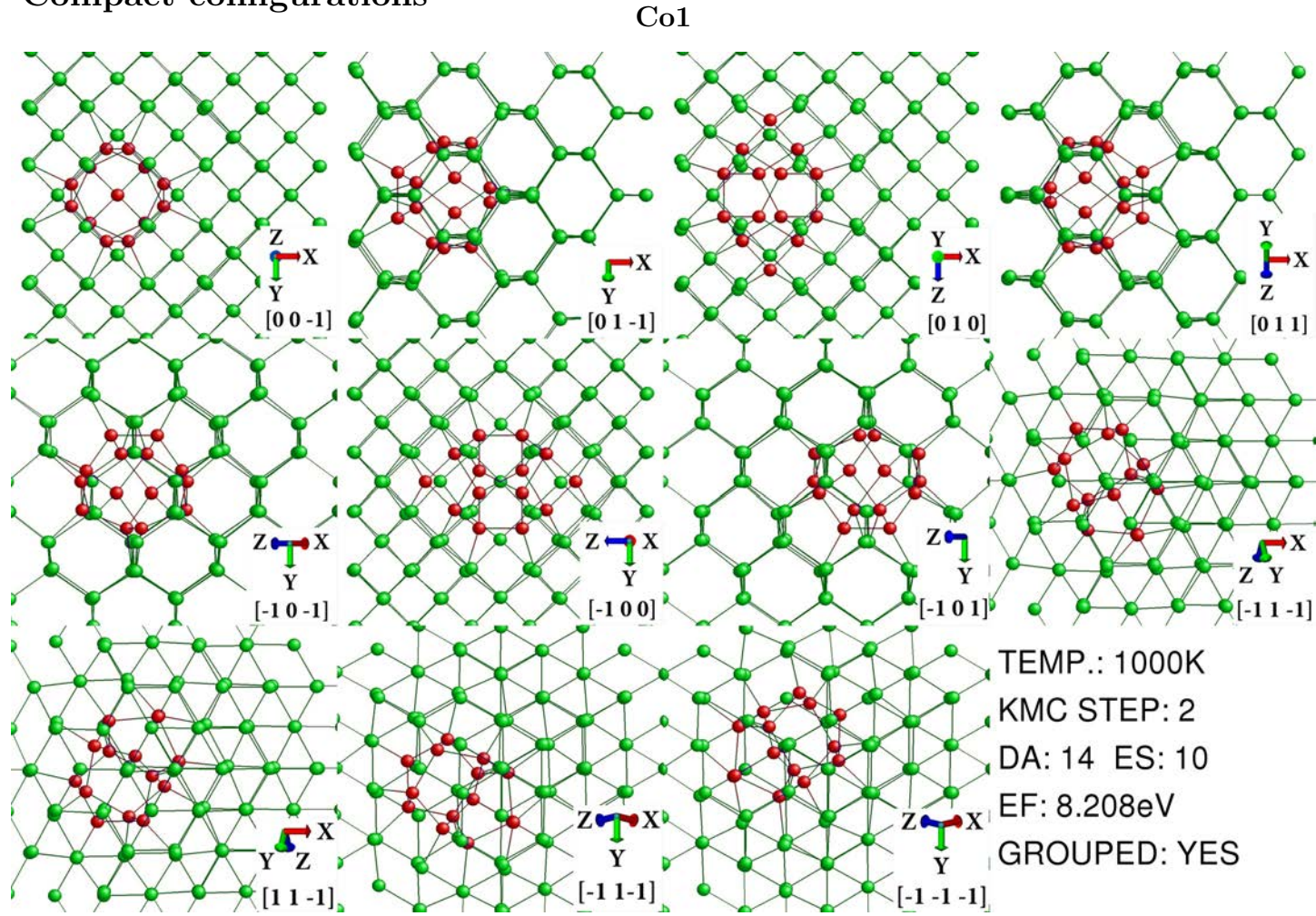


Figure A.14: Tenth minimum energy configuration for chainlike I_3

A.2.3 $\{111\}$ configurationsFigure A.15: $\{111\}$ configuration for I_3

A.3 I_4

A.3.1 Compact configurations

Figure A.16: First minimum energy configuration for compact I_4

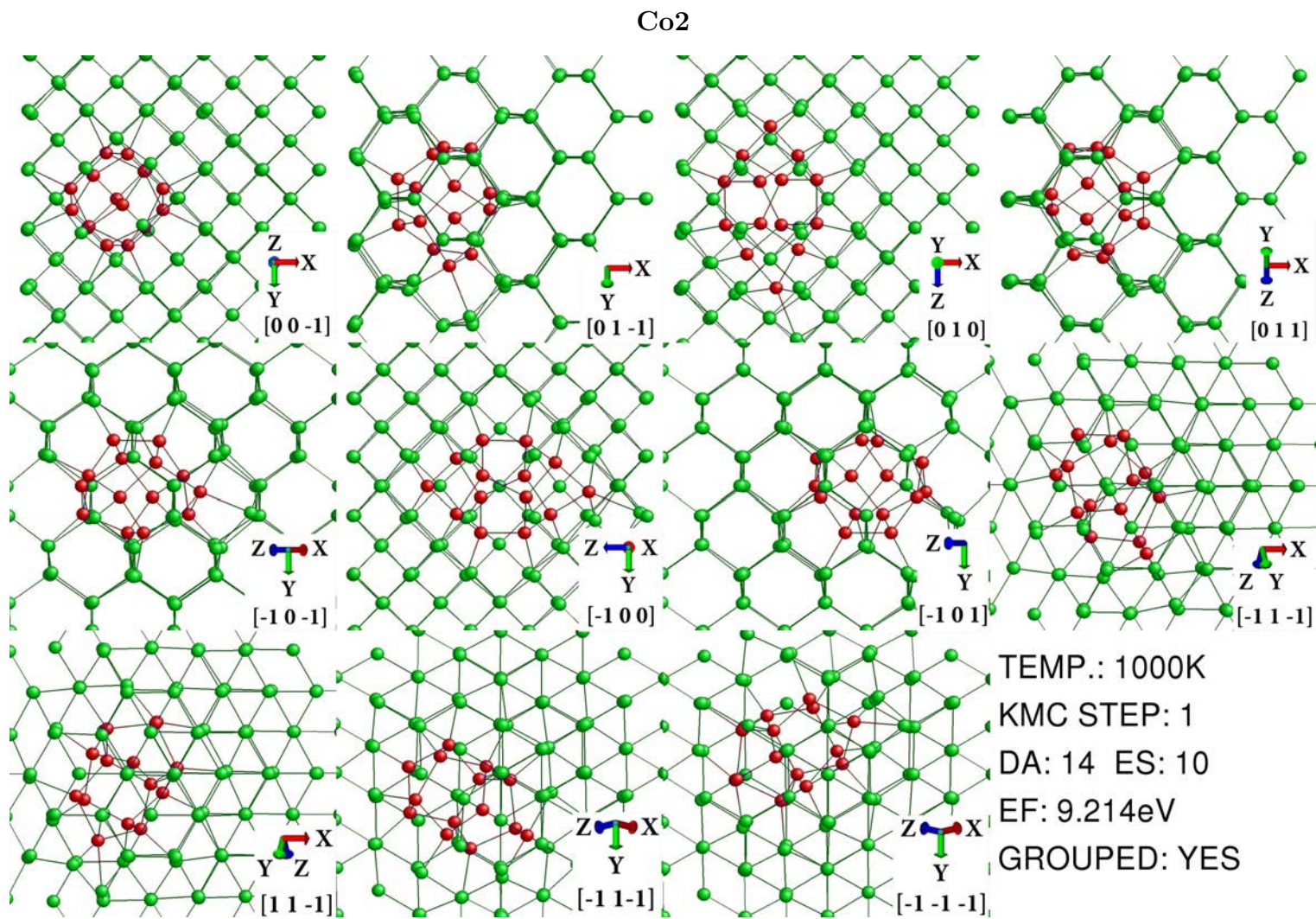
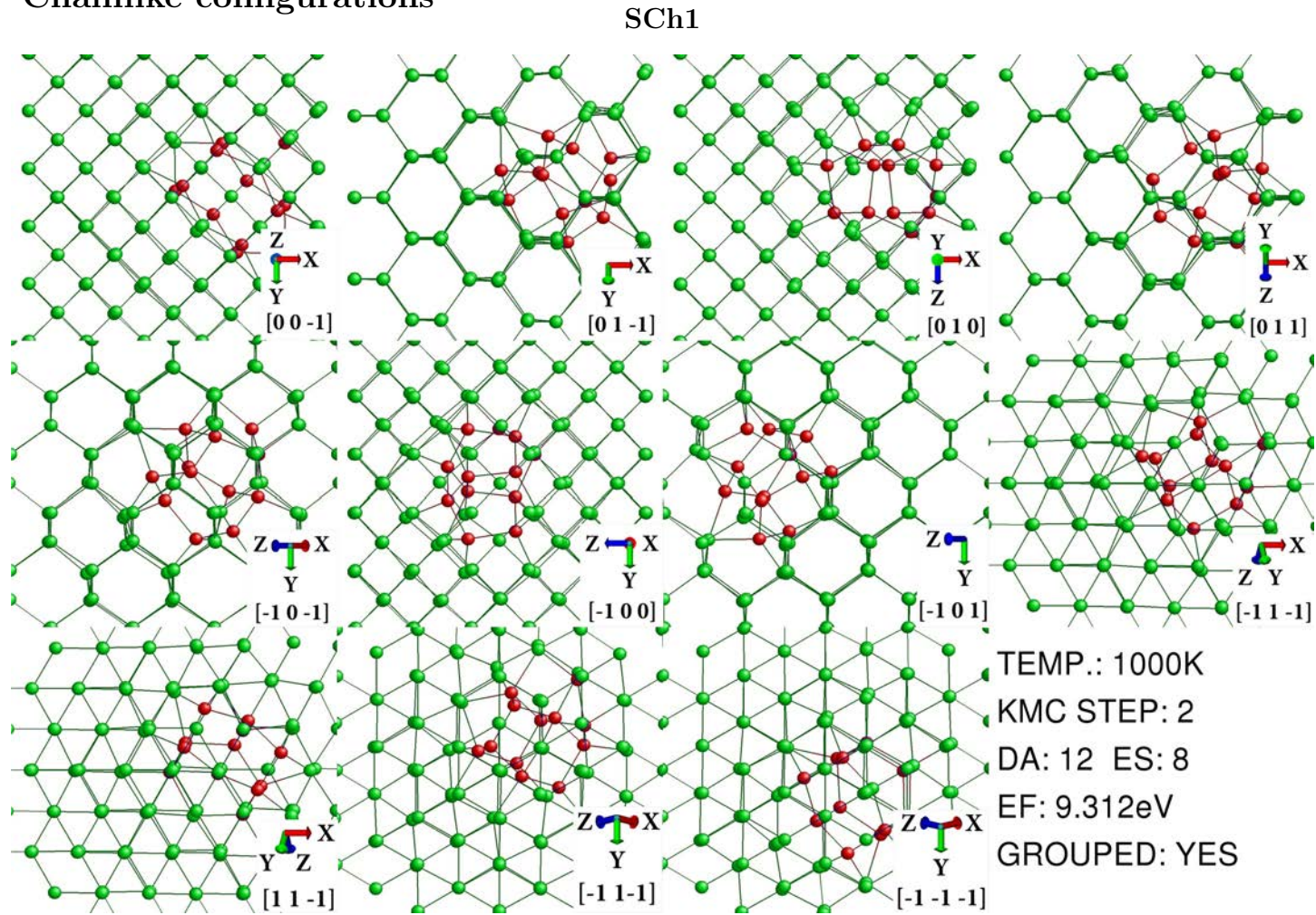
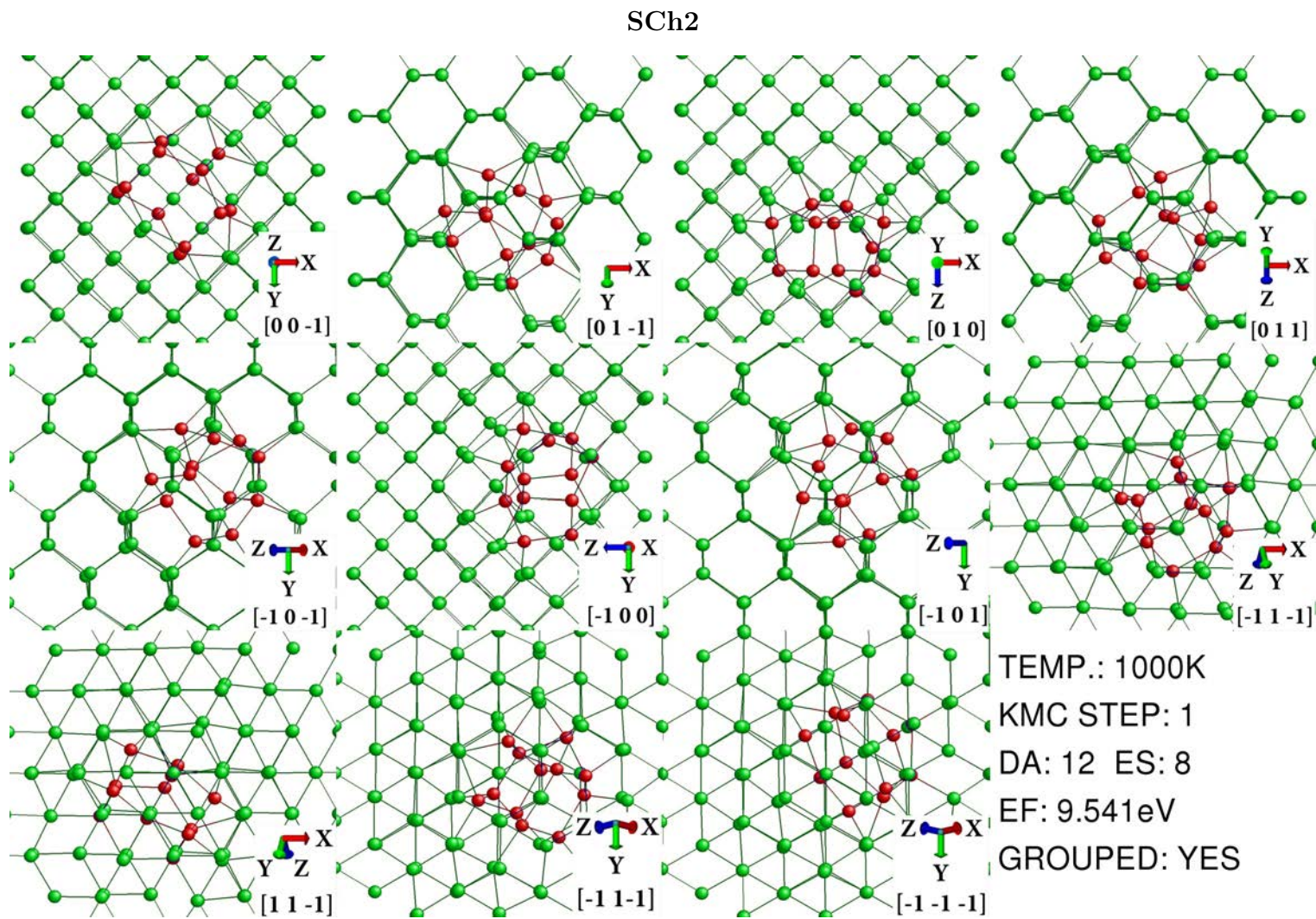


Figure A.17: Second minimum energy configuration for compact I_4

A.3.2 Chainlike configurations

Figure A.18: First minimum energy configuration for short chainlike I_4

Figure A.19: Second minimum energy configuration for short chainlike I_4

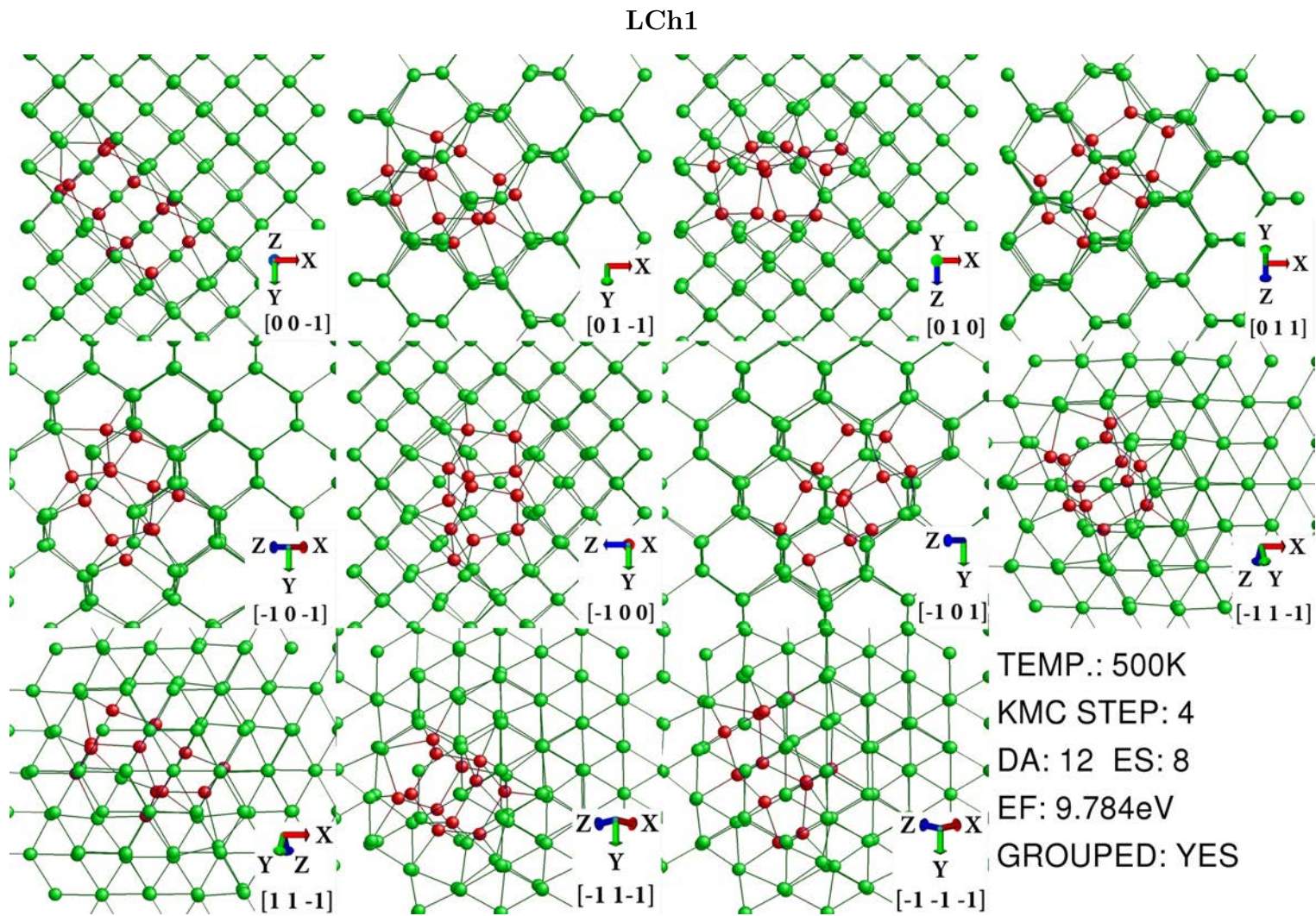
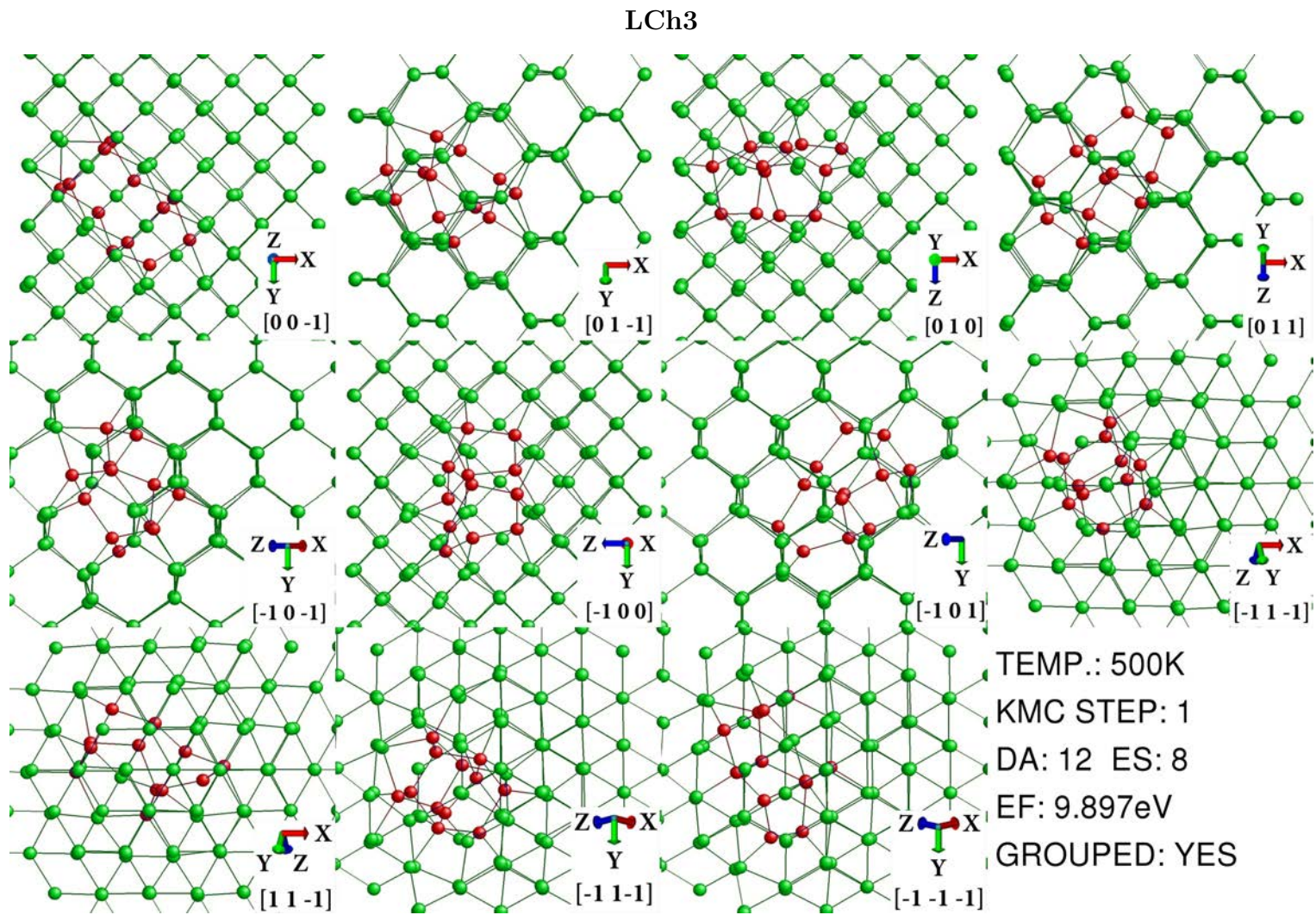


Figure A.20: First minimum energy configuration for long chainlike I_4

Figure A.21: Third minimum energy configuration for long chainlike I_4

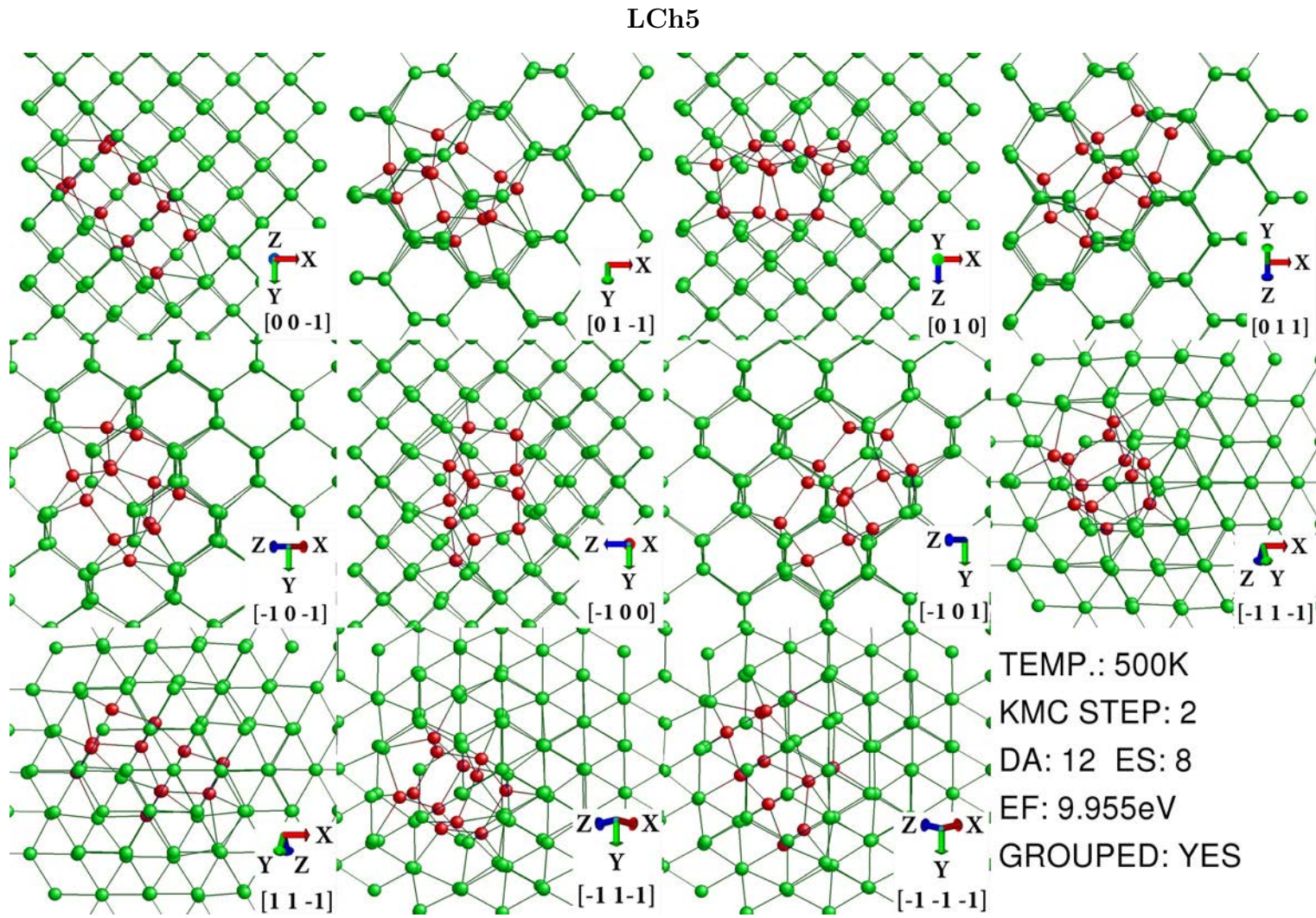
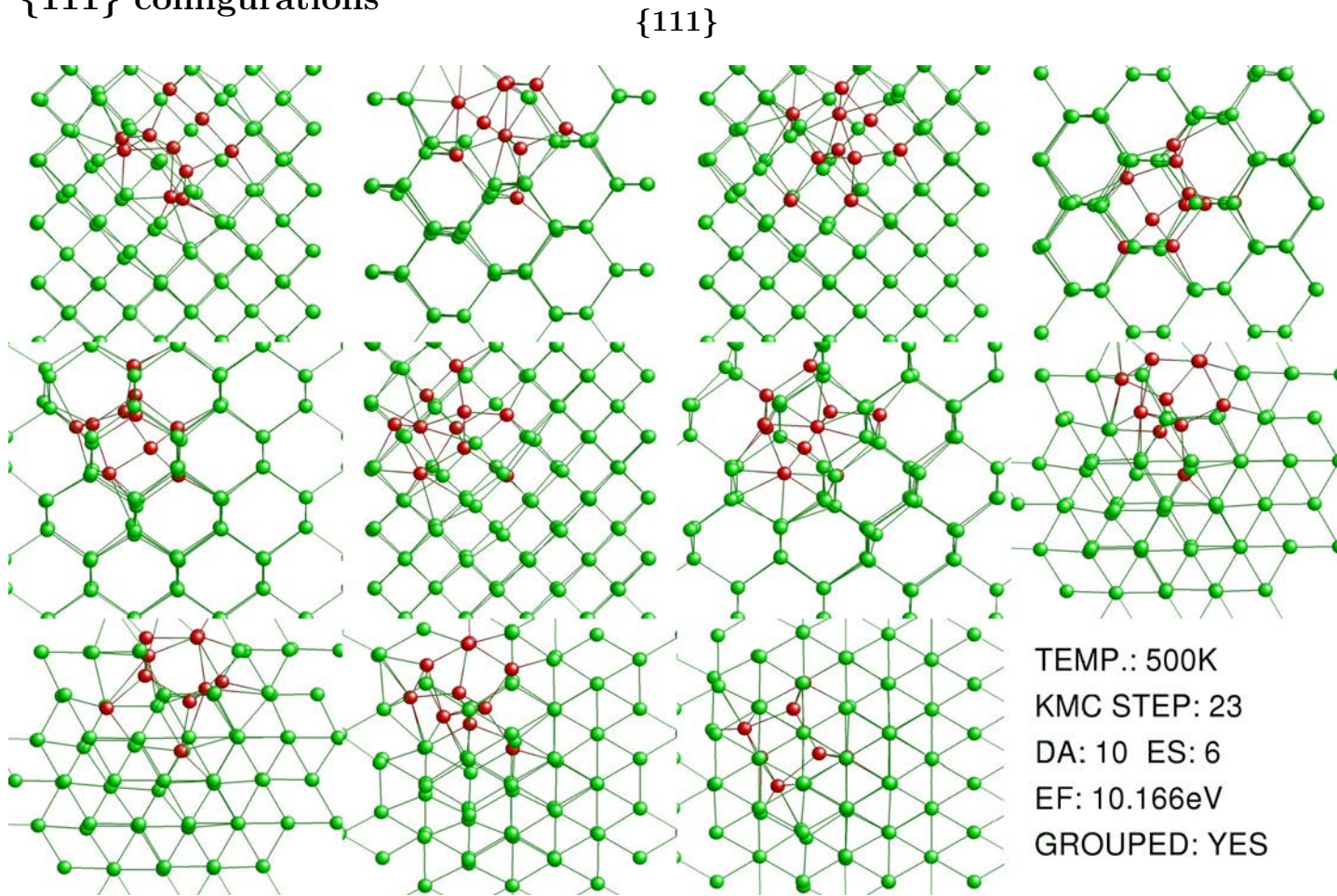


Figure A.22: Fifth minimum energy configuration for long chainlike I_4

A.3.3 $\{111\}$ configurationsFigure A.23: $\{111\}$ configuration for I_4

A.4 I_5

A.4.1 Compact configurations

Co1

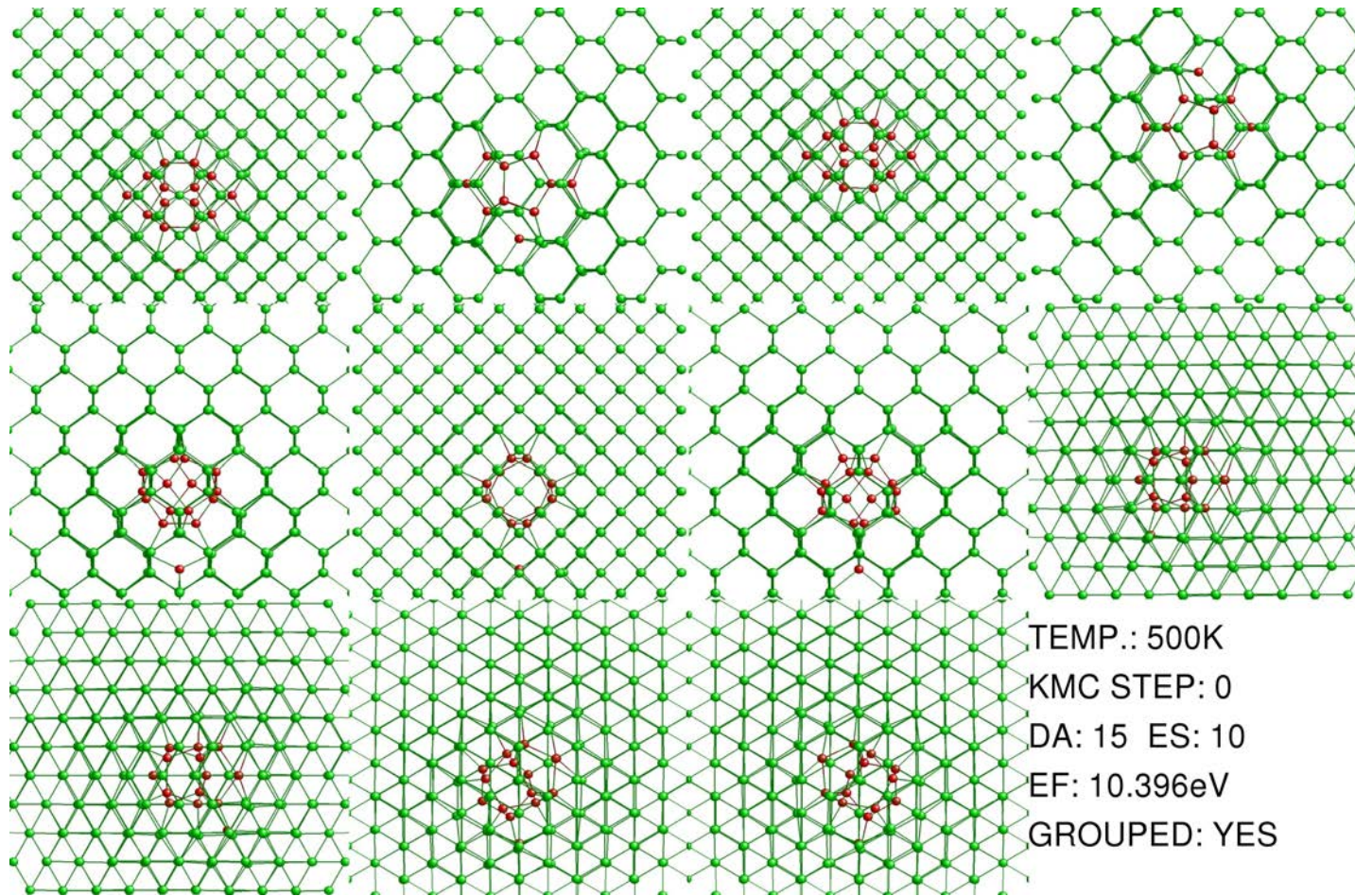


Figure A.24: First minimum energy configuration for compact I_5

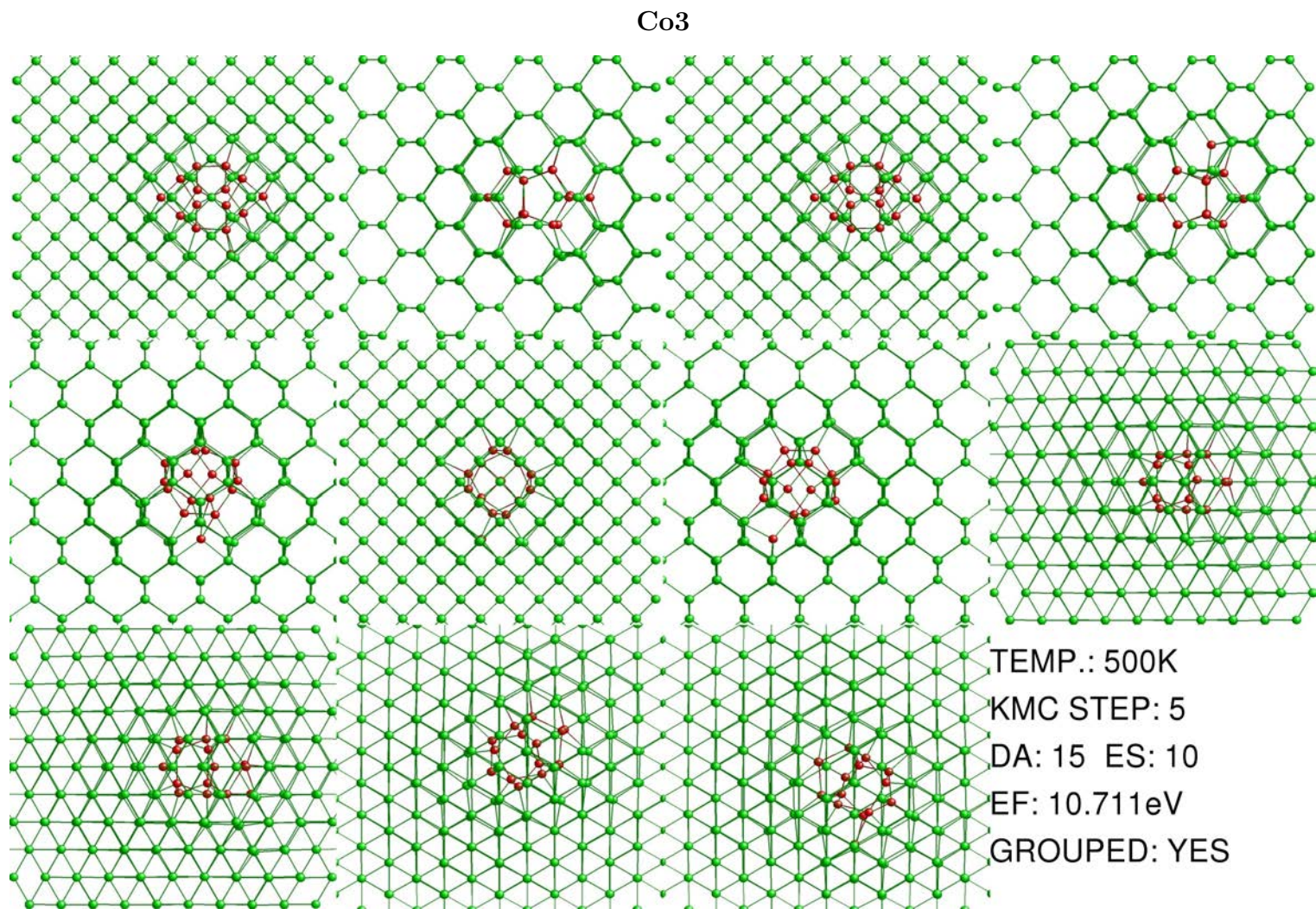


Figure A.25: Third minimum energy configuration for compact I_5

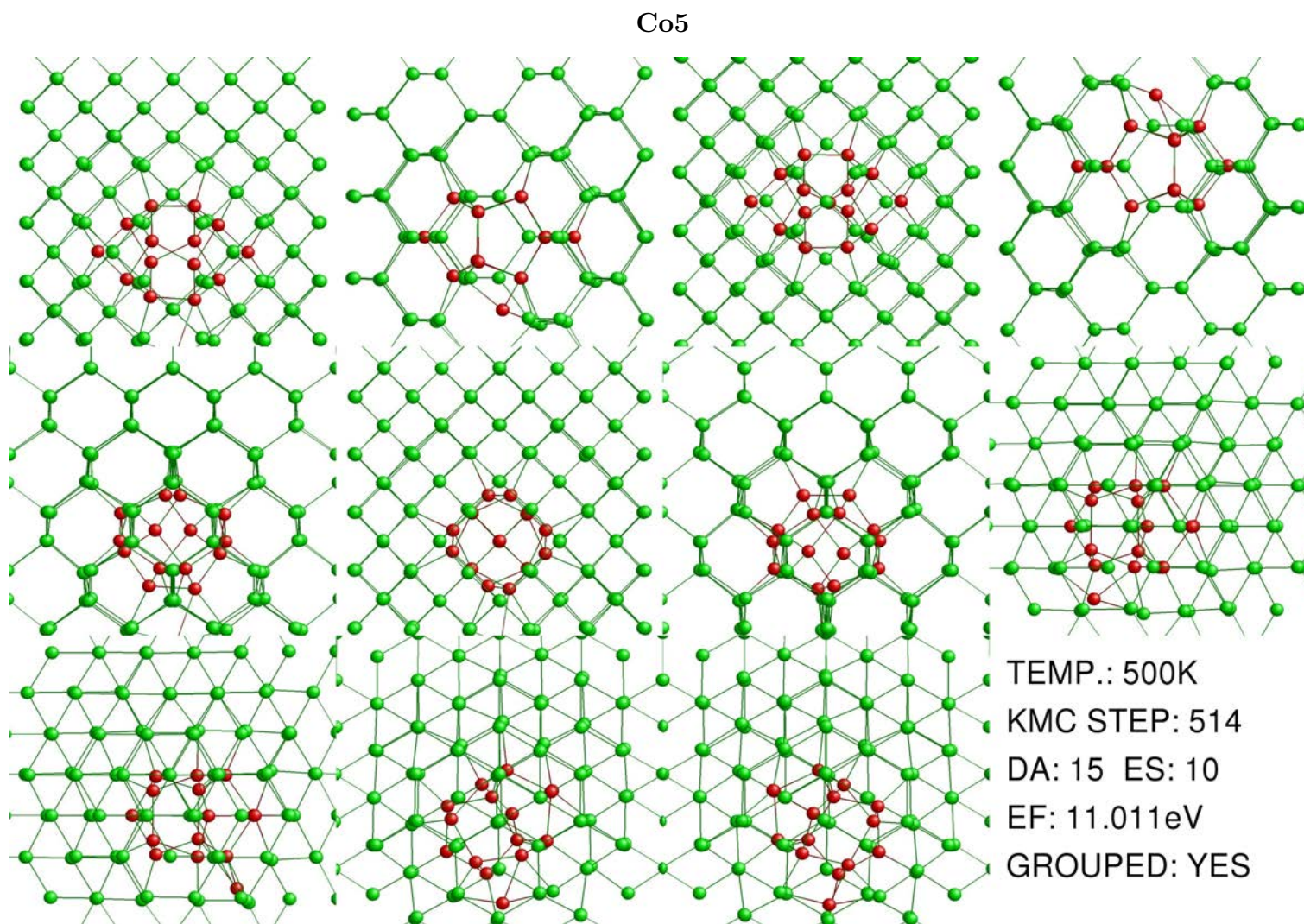
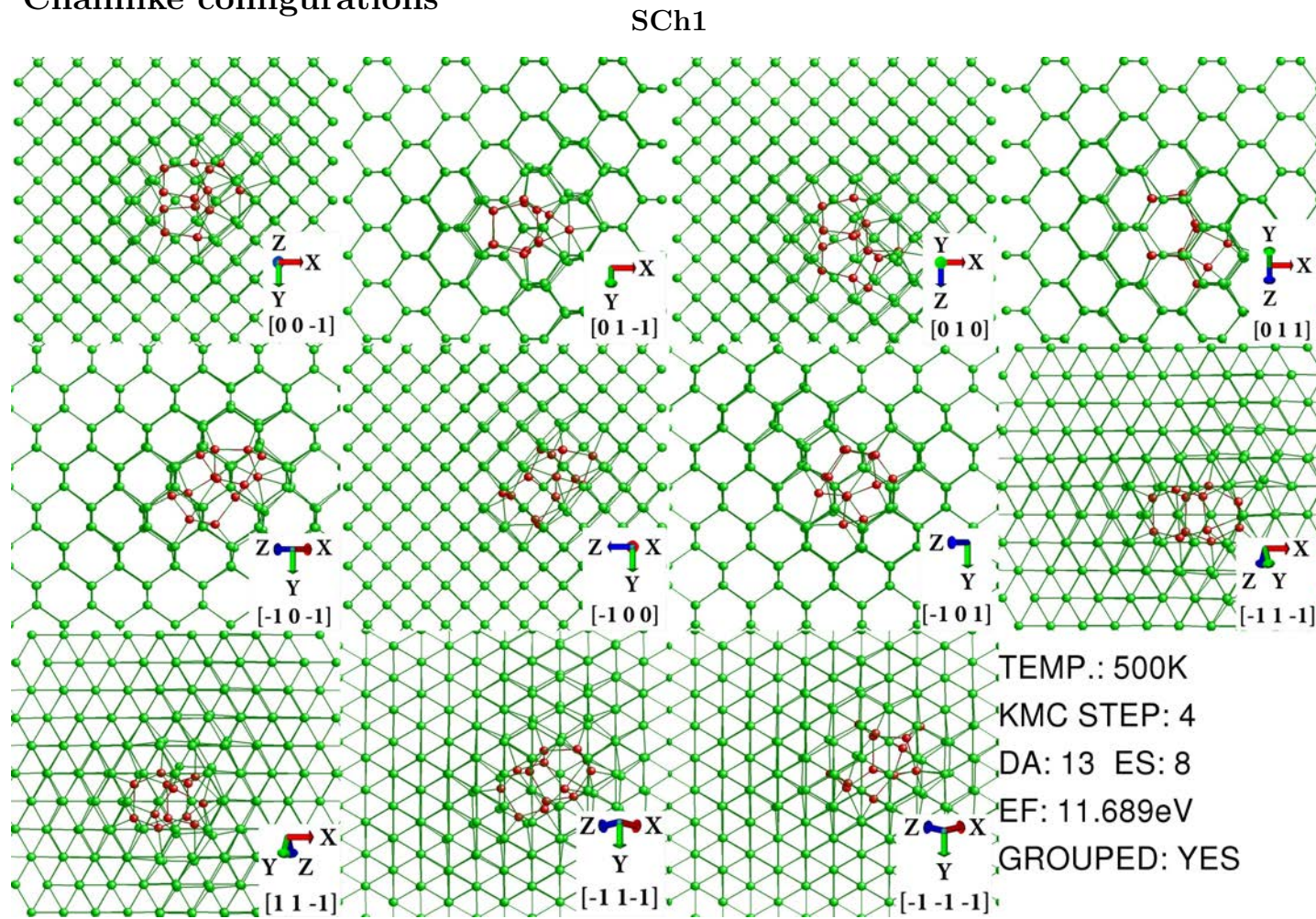


Figure A.26: Seventh minimum energy configuration for compact I_5

A.4.2 Chainlike configurations

Figure A.27: First minimum energy configuration for short chainlike I_5

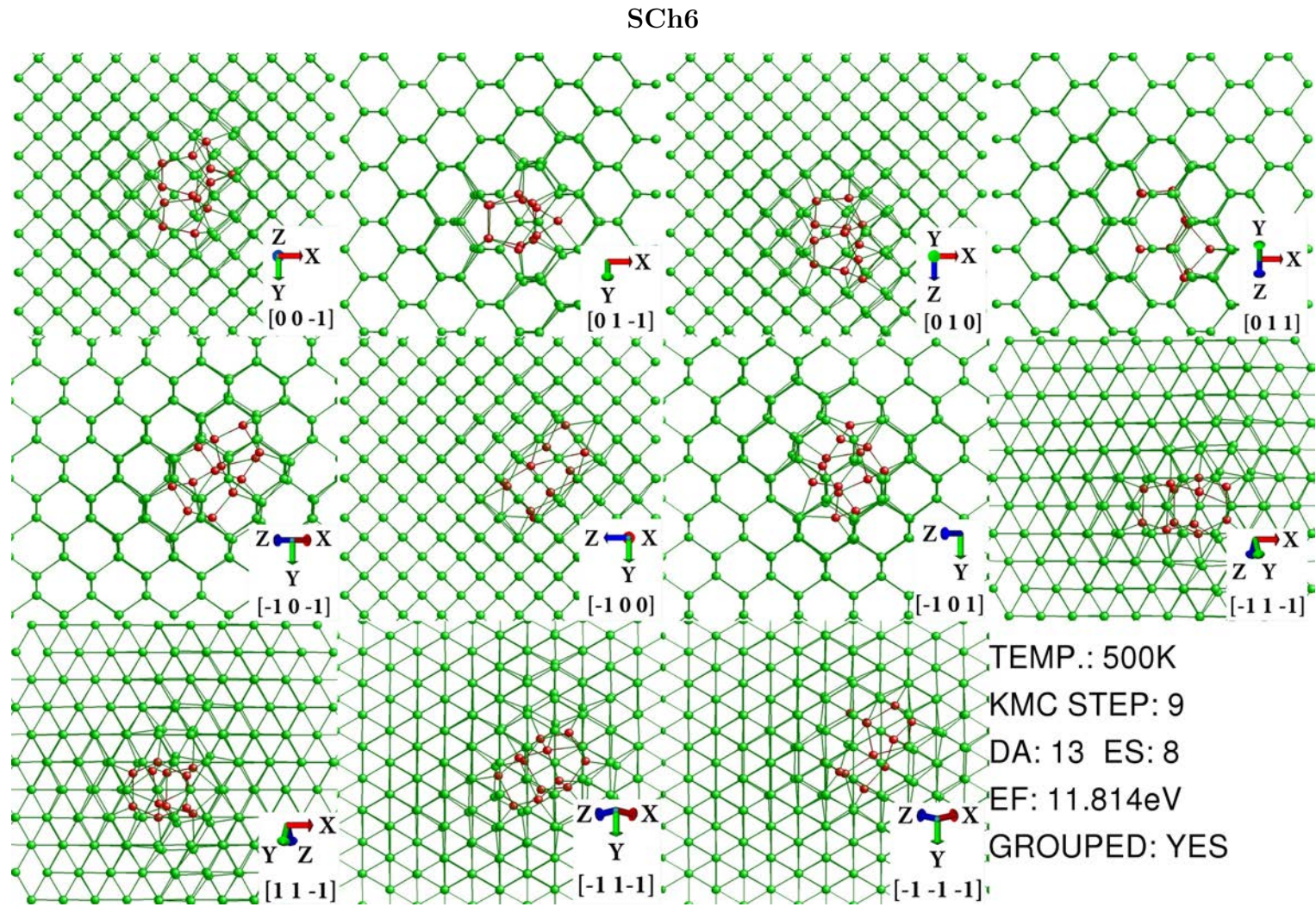
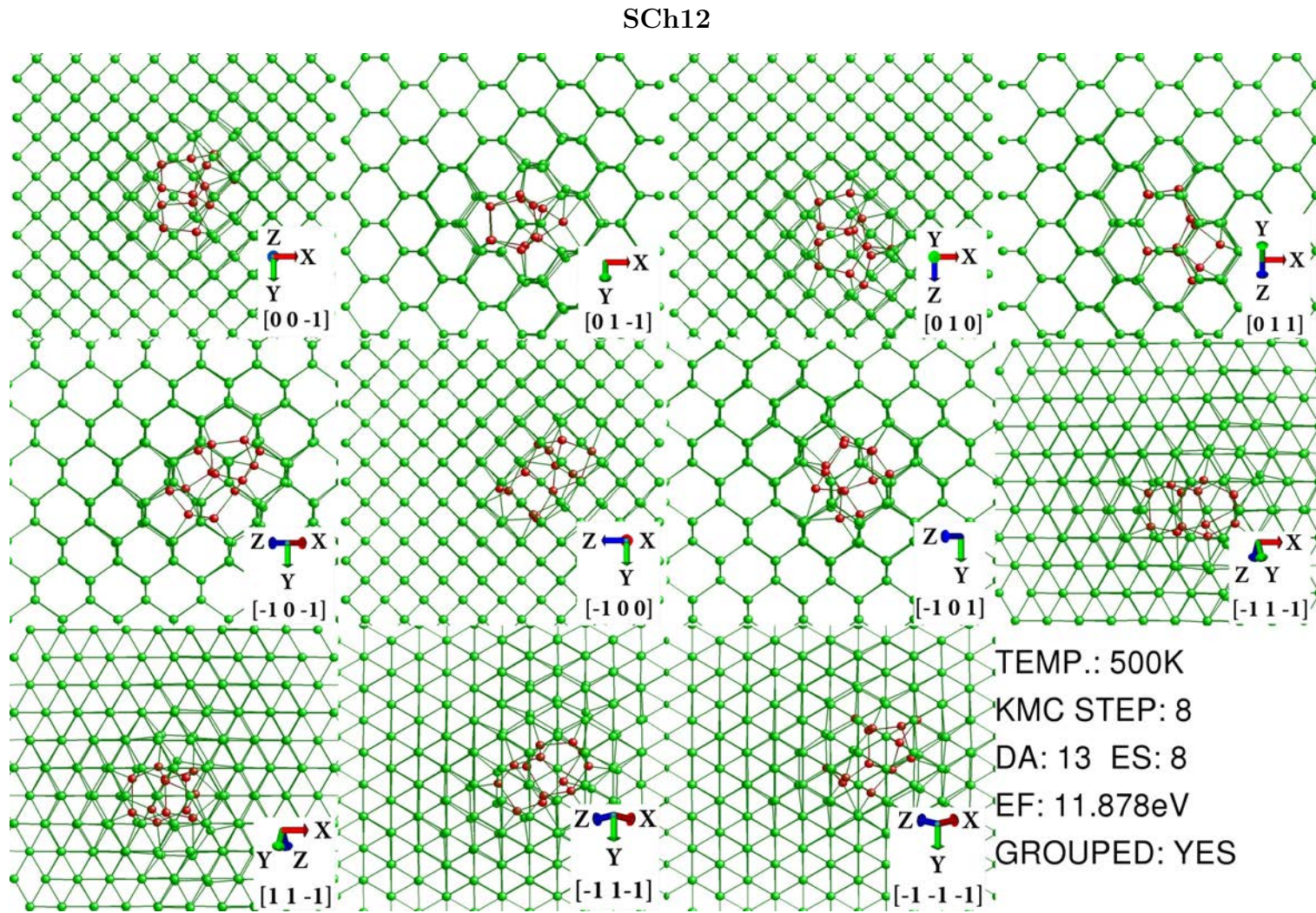


Figure A.28: Sixth minimum energy configuration for short chainlike I_5

Figure A.29: Twelfth minimum energy configuration for short chainlike I_5

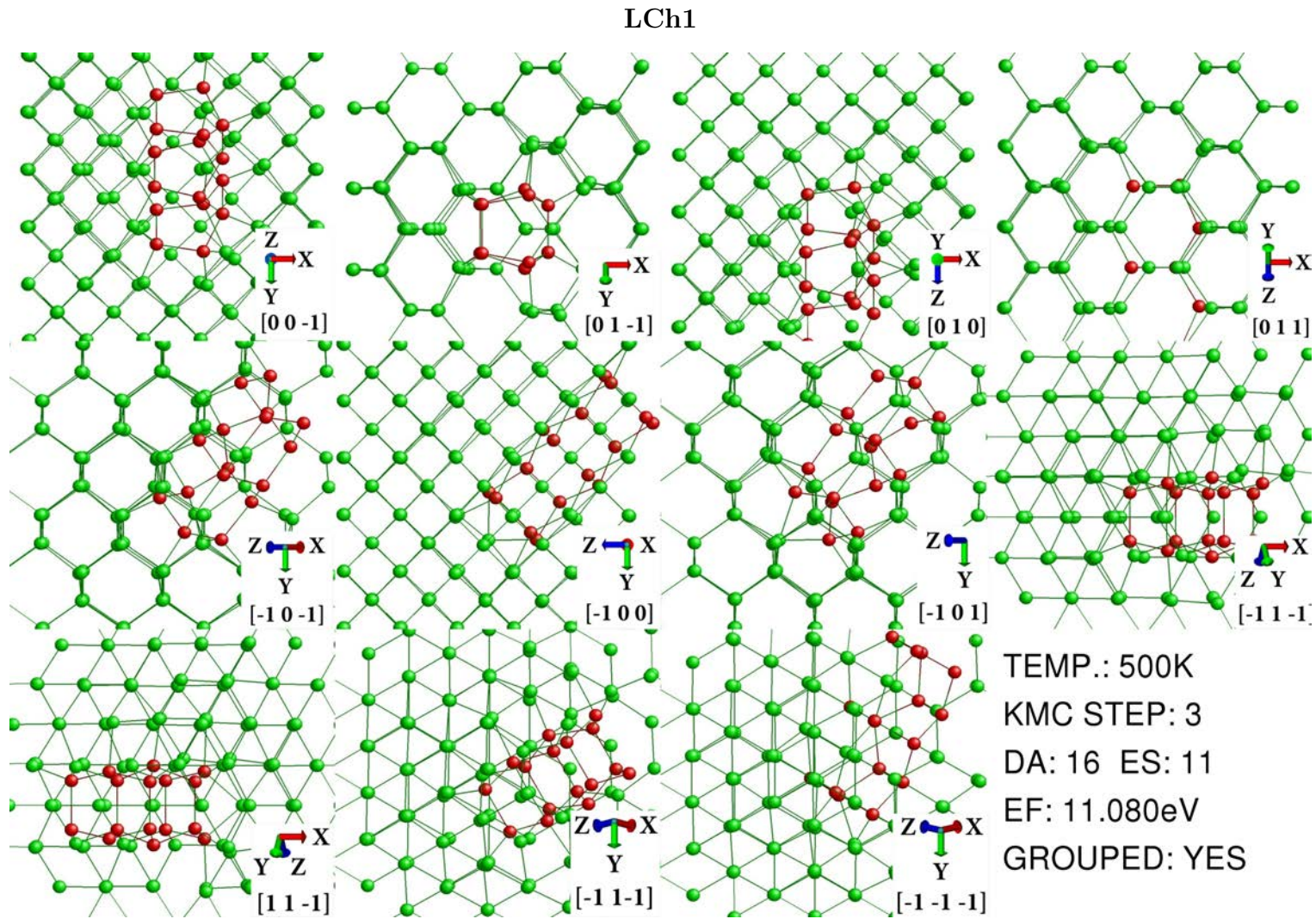
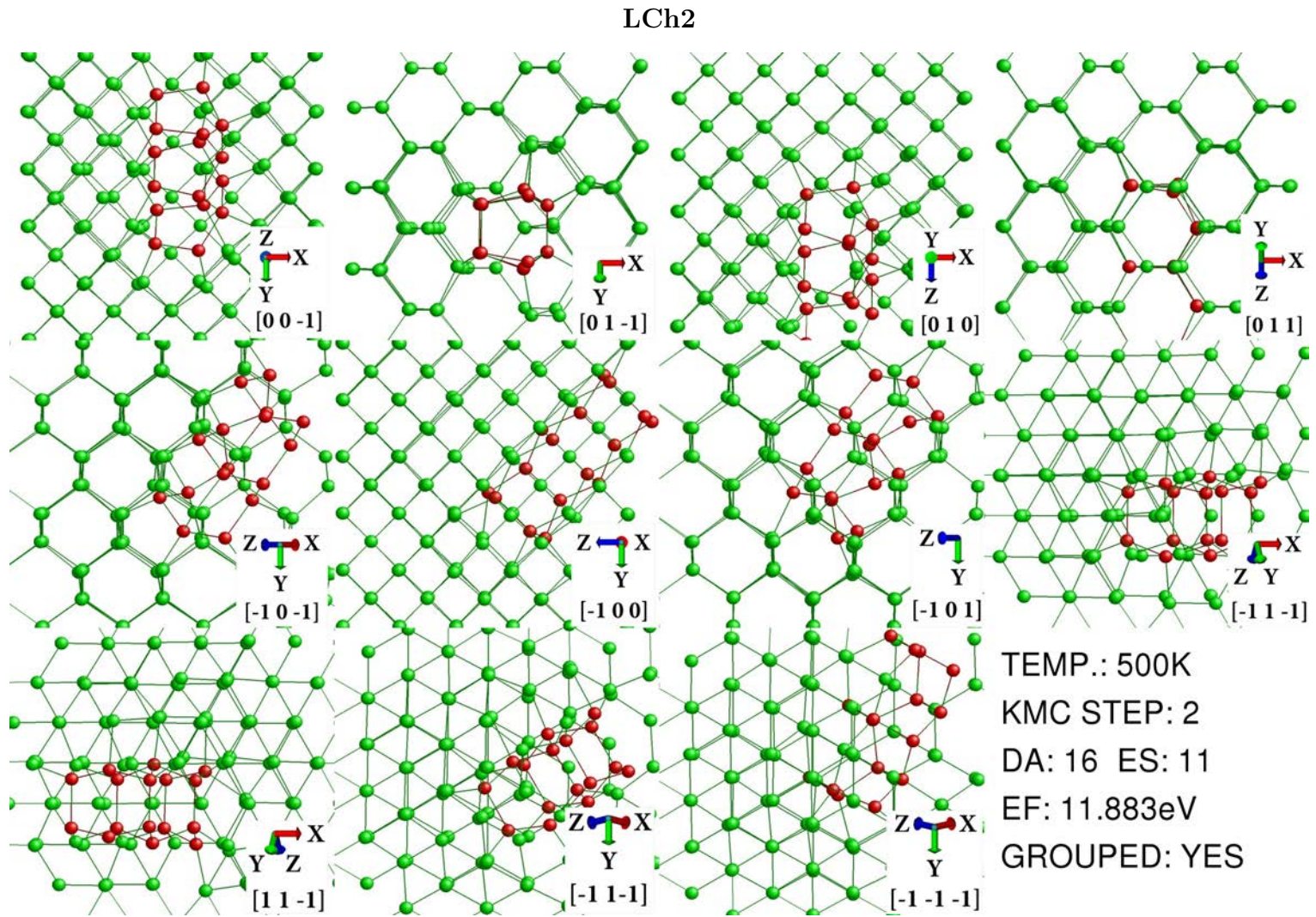
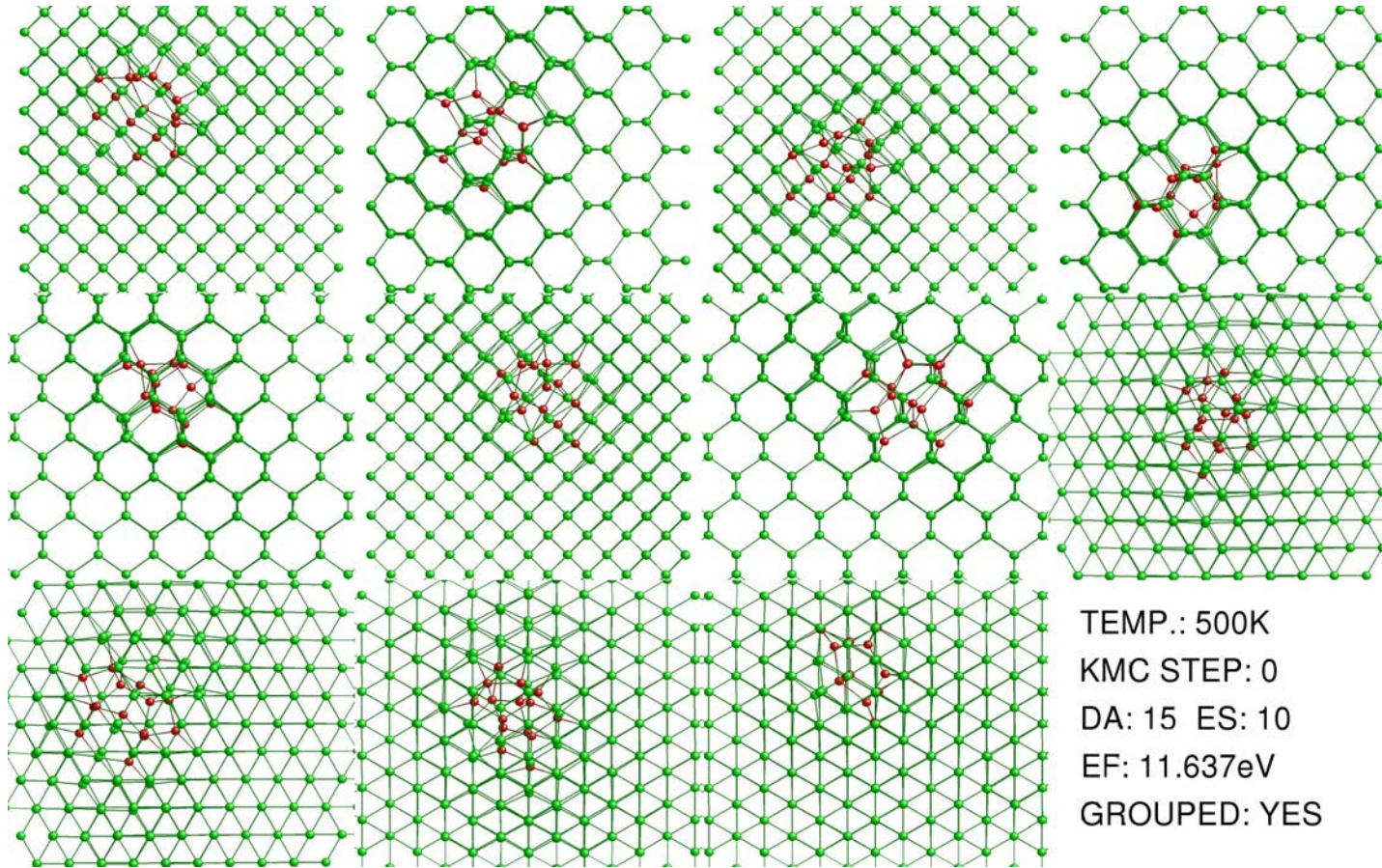


Figure A.30: First minimum energy configuration for long chainlike I_5

Figure A.31: Second minimum energy configuration for long chainlike I_5

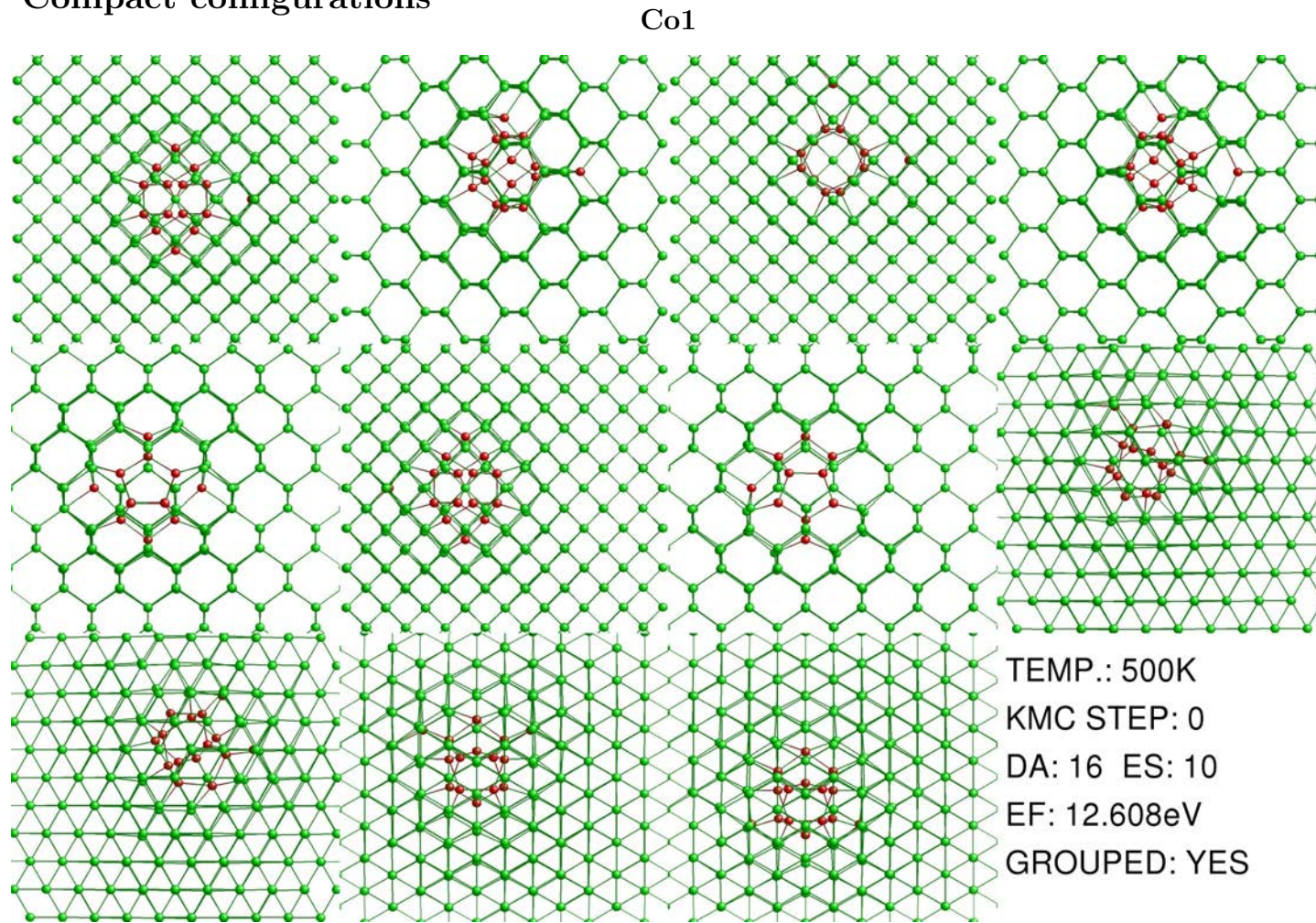
A.4.3 {111} configurations

{111}

Figure A.32: {111} configuration for I_5

A.5 I_6

A.5.1 Compact configurations

Figure A.33: First minimum energy configuration for compact I_6

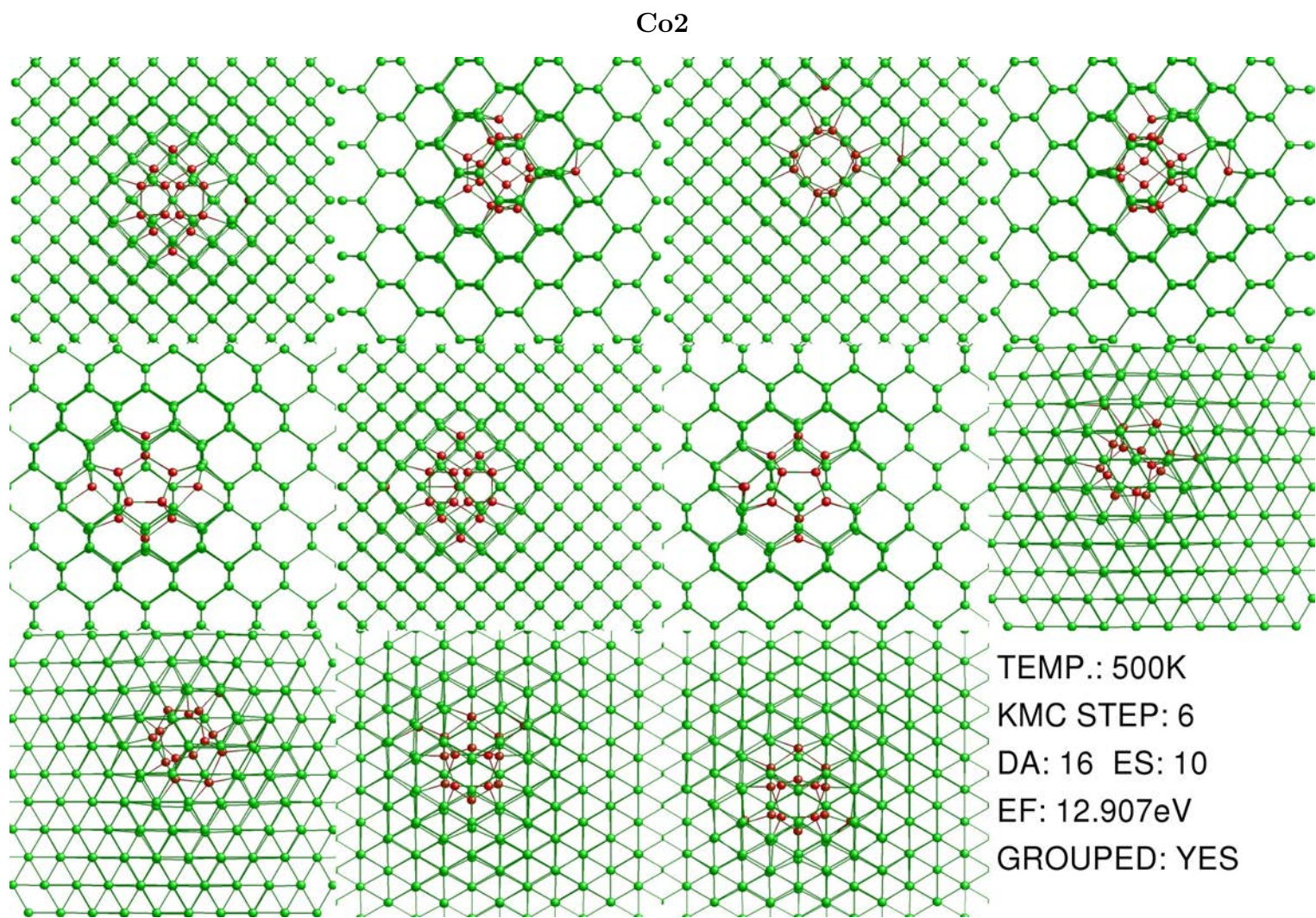
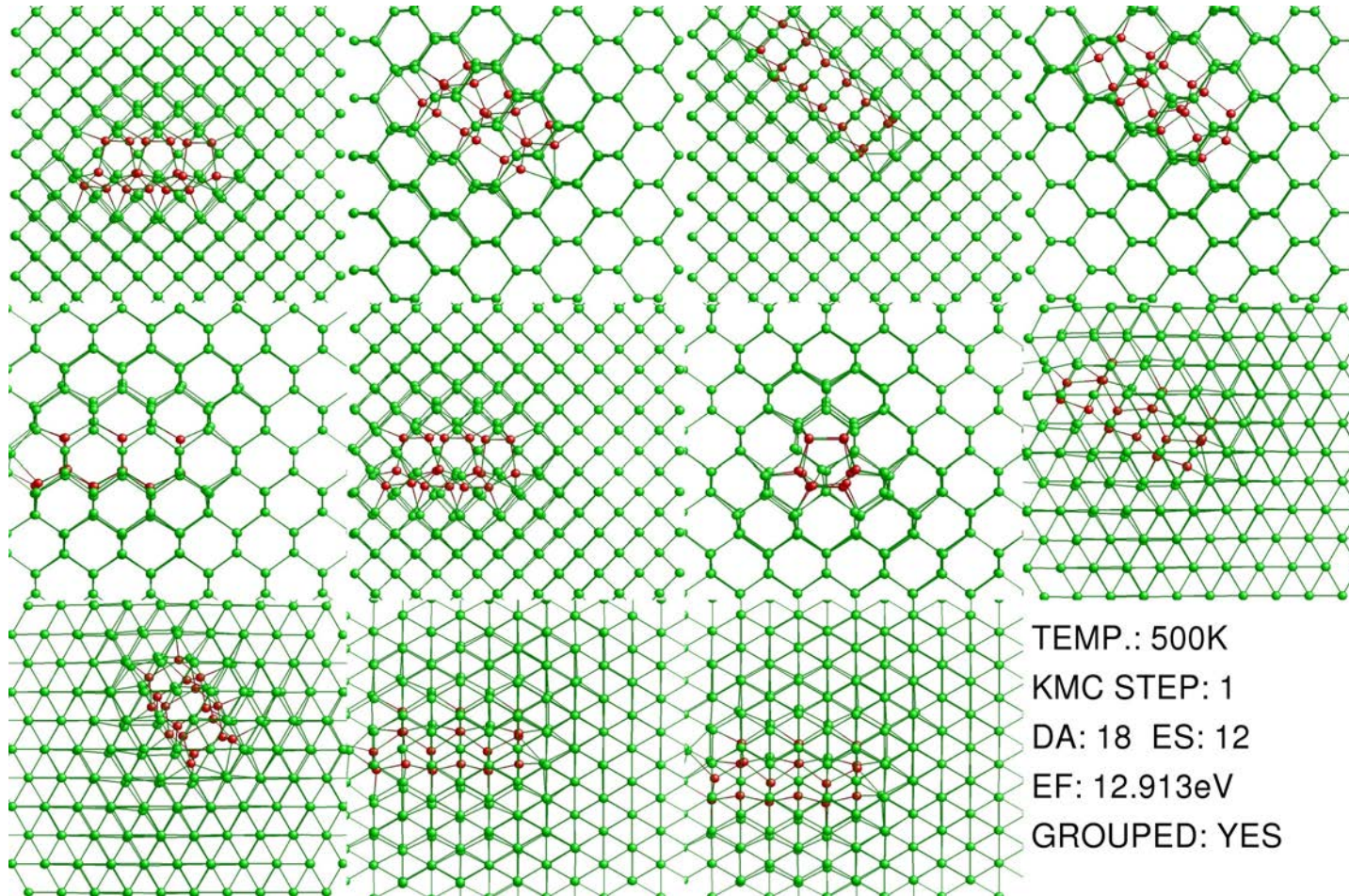


Figure A.34: Second minimum energy configuration for compact I_6

A.5.2 Chainlike configurations

Ch1

Figure A.35: First minimum energy configuration for chainlike I_6

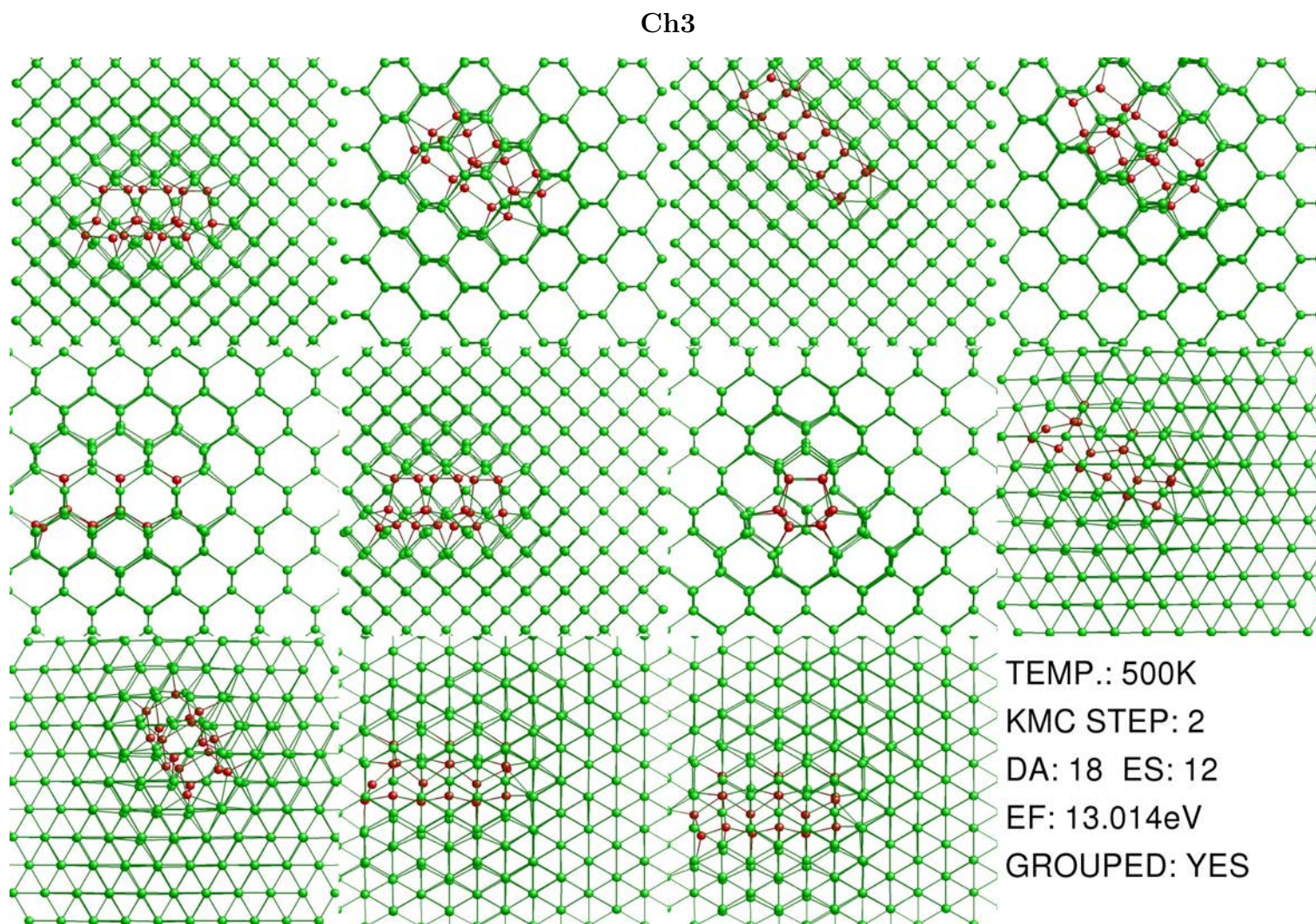
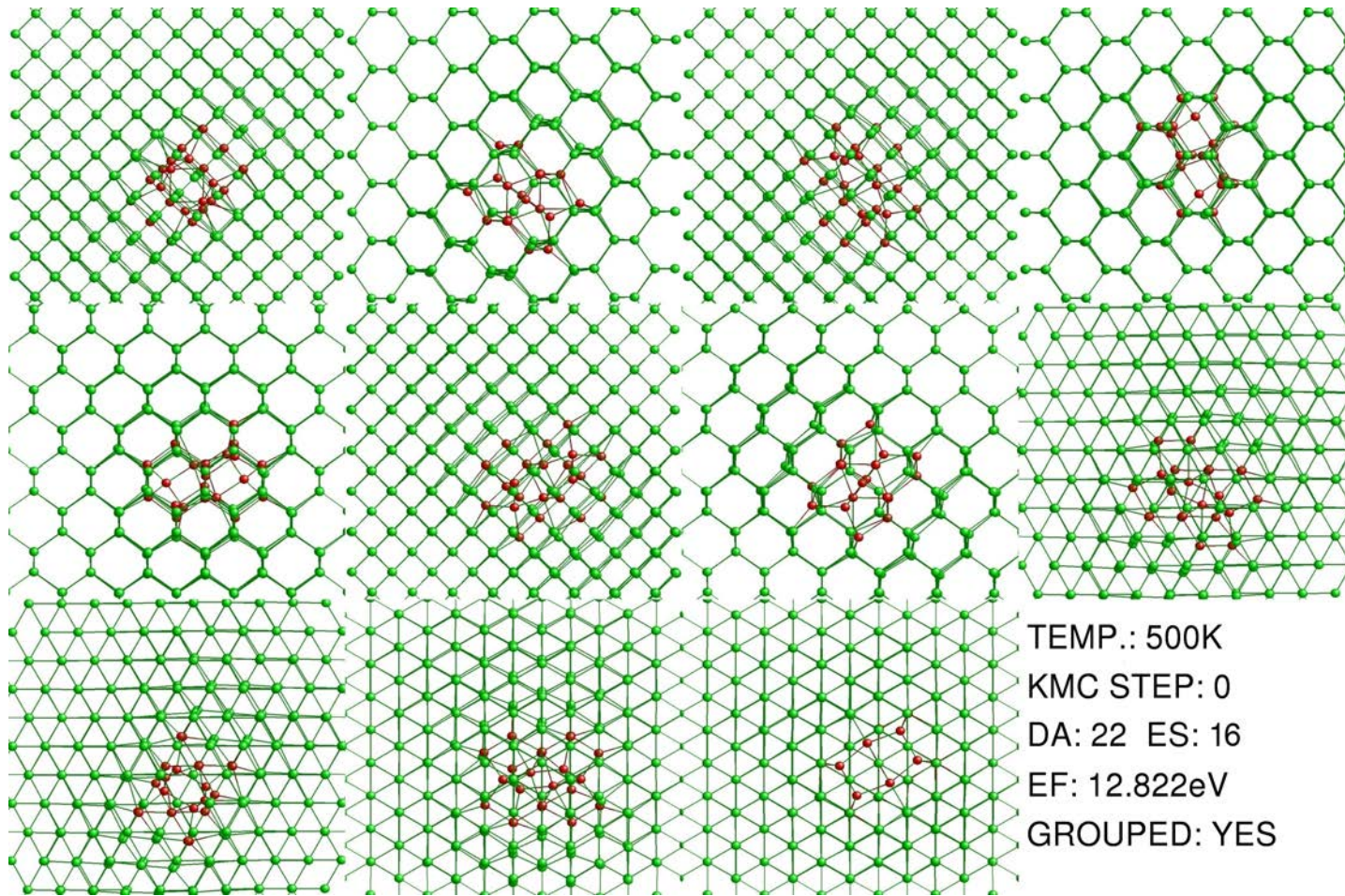


Figure A.36: Third minimum energy configuration for chainlike I_6

A.5.3 $\{111\}$ configurations $\{111\}_1$ Figure A.37: First $\{111\}$ configuration for I_6

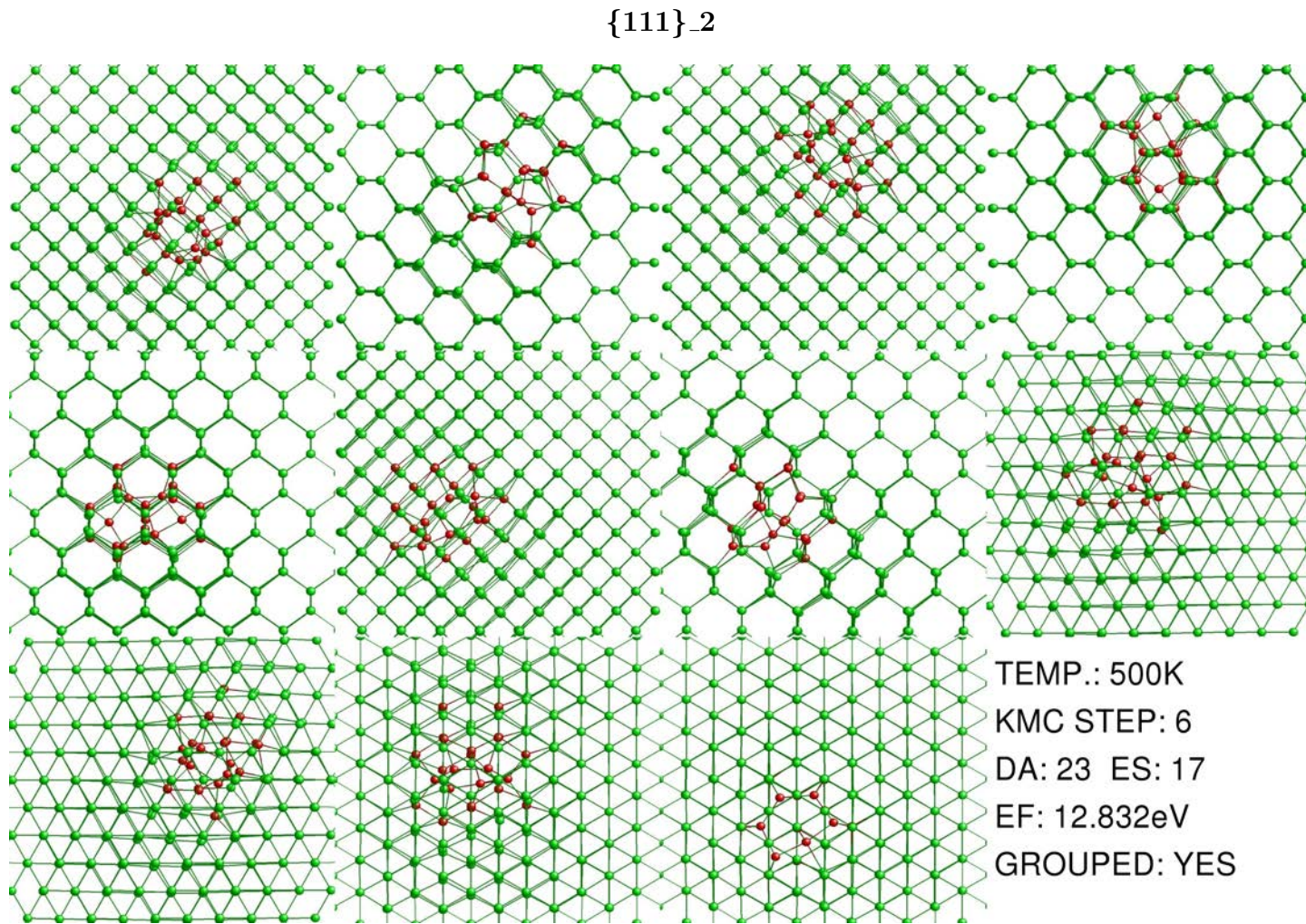


Figure A.38: Second $\{111\}$ configuration for I_6

Appendix B

Examples of input files

B.1 Input file in kART

Listing B.1: Sample of input file in kART with default parameter values including detailed explanations (KMC-long.sh).

```
1 #!/bin/csh
2 # IMPORTANT CONSIDERATIONS:
3 # - By convention variables must be defined in uppercase, if one is in
4 #   lowercase you will have errors at the moment executing the script.
5 # - To execute and test this script, put your executable at the end of
6 #   this file, and type in the simulation dir:
7 #   mpirun -np 2 KMC.sh (using ls -l be sure that KMC.sh has right
8 #                       permissions to execute: -rwxr-xr-x).
9 # - If you are in a cluster, use the standard procedure to submit jobs.
10
11
12 #----- kART PARAMETERS -----#
13
14 ##### Random number generator #####
15 #setenv RANDOM_SEED      -1103          # Random seed used to generate
16     random numbers (use only when debugging).
17
18 ##### Simulation Details #####
19 setenv NBRE_KMC_STEPS    1000          # The max number of KMC steps
20     to be executed.
21 setenv TOTAL_TIME        1.0000E1000  # The maximum simulation time
22     in seconds .
23 setenv ELAPSED_TIME      0.0           # The elapsed time before
24     first KMC step (will be added to the sim. time).
25 setenv STEP_KMC_MEASURE  1            # Number of steps before each
26     KMC measurement (i.e. writing to the .dat files).
27 setenv TEMPERATURE       500.0        # The simulated temperature in
28     kelvin.
29
```

```

24 ##### System size and types #####
25 setenv NUMBER_ATOMS      4101      # The total number of atoms.
26 setenv NSPECIES          1          # The number of different atom
    types.
27 setenv ATOMIC_TYPE_FILE  'Atomic_types' # Input file containing list of
    atomic types.
28 setenv SIMULATION_BOX    43.456     # The size of the simulation
    box (x, y and z).
29 # In case our box has different axes sizes:
30 #setenv X_BOX            # The size of the simulation
    box in the x dimension.
31 #setenv Y_BOX            # The size of the simulation
    box in the y dimension.
32 #setenv Z_BOX            # The size of the simulation
    box in the z dimension.
33
34 ##### Force field / Verlet neighbor list #####
35 setenv ENERGY_CALC      LAM        # Choose between EDP (
    Environment-Dependent Interatomic Potential) or SWP (Stillinger-Weber
    Potential with original parameters) or SWA (Stillinger-Weber
    Potential with modified parameters for amorphous silicon by Vink et.
    al) or LAM (LAMMPS).
36 setenv FORCE_CALC        TTL        # Choose TTL for total force
    calculation and PAR for partial force.
37 setenv ENERGY_TYPE     LAM        # Chose the potential to
    calculate the forces when ENERGY_CALC = LAM.
38 setenv INPUT_LAMMPS_FILE 'in.lammps' # LAMMPS input file when using
    ENERGY_TYPE = LAM to calculate the forces.
39 setenv ENERGY_UNITS_CONVERSION kcal2ev # Converts the energy units
    provided by the force code into a desired value for ART - affects
    only the parameters in ART and kART.
40 setenv UPDATE_VER_NEIB   TTL        # Choose TTL for total force
    calculation and PAR for partial force.
41 setenv NEIB_CALC         ALL        # Choose ALL or VER (Verlet
    Algorithm).
42 setenv UPDATE_TOPO       TTL        # Choose TTL or PAR.
43 setenv PAR_DISP_THRESH2  0.00001   # Maximum displacement squared
    which triggers an update of the neighbor list when using VER lists.
44
45 ##### Main Output options/files #####
46 setenv STATISTICS        .true.     # Write statistics about force
    and event calculation.
47 setenv OUTPUT_CONFIG_EVENTS .true.  # Create file with list of all
    the topologies and events after each KMC step.
48 setenv SAVE_FULL_EVENTS  .false.    # Save full events during
    catalog and after - one file per node (events.node_number).
49 setenv INI_FILE_NAME     '511.conf'  # The file name containing the
    initial configuration.
50 setenv KLOGFILE          'KMC_log.txt' # The file name containing a
    list of the parameters used.
51 setenv DISPL_FILE        'Diffusion.dat' # The file name used to store
    the squared displacement.

```

```

52 setenv CONFFILE          'allconf'          # Name of the file to store
    all the configurations visited.
53 setenv EV_FILE          'selec_ev.dat'      # File name for statistics
    about the selected events.
54 setenv GENERIC_EV_STAT  'Gen_ev.dat'       # File name for statistics
    about the GEN events.
55 setenv SPECIFIC_EV_STAT 'Spec_ev.dat'      # File name for statistics
    about the SPEC events.
56 setenv ENERGY_STAT    'Energies.dat'     # File name for statistics
    about the Energy.
57 setenv LOWEST_BARRIER  'MinBarr.dat'      # File name for statistics
    about the minimum barrier in the tree.
58 setenv BASIN_GRAPH      'basin_graph.gv'   # File name for storing
    statistics about basins.
59
60 ##### Restart options #####
61 setenv RESTART_KMC      .false.           # If true, restart from
    previous run.
62 setenv RESTART_FILE     "this_conf"       # The file name used to
    continue a simulation from where it was last stopped.
63 setenv RESTART_IMPORT   .false.           # Start NEW simulation with
    current KMC event catalogue (events.uft and topos.list).
64 setenv NEW_CATALOGUE    .false.           # Continue a simulation but
    will rebuild event catalogue from scratch.
65 setenv RESTART_MERGE    .false.           # Read multiple EVENT_DIR
    event files and combine them into a single catalogue.
66 setenv SIMINFO          'siminfo.list'     # List of directories to read
    for the RESTART_MERGE option.
67
68 ##### Basin parameters #####
69 setenv OSCILL_TREAT      BMRM              # Choose between BMRM, TABU or
    NON.
70 setenv MIN_SIG_BARRIER 0.3              # Maximum height of barrier
    and inv. barrier for an event to be considered inside a basin.
71 setenv BASIN_LOCAL      .false.           # If true, local basins are
    used.
72 setenv BASIN_RADIUS     10.0             # If using local basin, what
    is the radius of a local move.
73 setenv FLICKER_DIR      'FLICKERS'        # The name of the directory
    where the TABU event files are stored.
74 setenv EVENTS_MEMORY    100              # Number of event to keep in
    memory for TABU.
75 setenv BASIN_MEMORY     10              # Number of basins to keep in
    memory for TABU.
76
77 ##### Topology Params #####
78 setenv TOPO_RADIUS      6.0              # Radius for topology cluster.
79 setenv MAX_TOPO_CUTOFF  2.7              # Length-cutoff used by
    default to link two atoms.
80 setenv MIN_TOPO_CUTOFF  2.2              # Minimal length cutoff used
    when looking at secondary topologies.
81 setenv CRYST_TOPOID     973883          # Topo id of the crystalline-
    like topologies.

```

```

82 setenv CRYST_TOPO_RADIUS 4.0 # Radius for crystal-like
   topologies.
83 setenv TOPO_IGNORE_FILE 'Topo_ignore' # Topo ids to ignore (read at
   start of sim).
84 setenv TOPOLOGY_FILE 'Topologies' # Info about topologies (not
   used for restarts).
85 setenv TOPO_STAT_FILE 'topos.list' # Statistics about topologies
   found.
86 setenv EVENT_IGNORE_FILE 'Event_ignore' # File storing event ids to
   ignore (read at start of sim).
87
88 ##### Generic events parameters #####
89 setenv SEARCH_FREQUENCY 50 # Minimum number of attempts
   to find a GENERIC event per new topology encountered.
90 setenv THRES_INCREASE_FREQ 25 # Number of failed attempts
   encountered because increasing the EIGEN_THRESH.
91 setenv TYPE_EVENT_UPDATE SPEC # Choose between SPEC (
   specific) or GENE (generic).
92 setenv USE_LOG_SEARCH .true. # Search frequency is
   multiplied by logarithmic increasing function.
93
94 ##### Stop file #####
95 setenv STOP_FILE "stop_now" # Name of the file that if
   present in the simulation dir at the end of a KMC step will cause the
   program to stop.
96 setenv STOP_AFTER_BASIN .true. # Program will wait to exit a
   basin before stopping the sim when a stopfile is present.
97
98 ##### Debug event search parameters #####
99 setenv OUTPUT_NEB_GEN_EVENT .false. # Creates 3 files per generic
   event in format used by the lammps implementation of NEB use to check
   if initial-saddle-final configurations are well defined.
100 setenv CHECK_INI_SAD_CONNECTIVITY .false. # When GENERIC saddle is found
   , pushes the system towards the initial minimum and minimizes. If
   minimized config. not the same as the initial one, the saddle is
   rejected.
101
102 ##### Event analysis parameters #####
103 setenv BARRIER_CRITERIUM 5.0 # Maximum barrier (eV) to
   consider an event otherwise considered useless.
104 setenv BARRIER_CHECK 0.2 # Minimum change in barrier
   needed to launch a scalar product during analysis.
105 setenv DISPLACEMENT_THRESHOLD 0.15 # Minimum distance (A) for
   considering an atom displaced.
106 setenv INV_BARRIER_CRITERIUM 0.1 # Minimum inverse barrier to
   consider and analyse an event (eV).
107 setenv DEL_E_EVENT 0.1 # Minimum energy difference
   between the final states of an event to decide in they are different
   (eV).
108 setenv PARALLEL_CHECK 0.95 # Criterium for stating if two
   vectors are parallel from their scalar product.
109 setenv DISP_CHECK 0.15 # Criterium for comparing the
   displacement between two events.

```

```

110 setenv DELR_THRESH          1.0      # Criterium used to check if
      displacement associated with an event is large enough (used if
      barrier and inv. barrier smaller than min_sig_barr).
111
112 ##### Specific event parameters #####
113 setenv MAX_HISTO            20      # The number of histogram bins
      .
114 setenv REFINE_PUSH          0.1      # The ratio of the saddle-
      final distance pushed before minimizing a SPEC event.
115 setenv OUTPUT_SPECIFIC     .false.  # If true, will create a
      SPEC_EVENT_DIR and output two txt files for each SPEC event found.
116 setenv MAX_SPEC_EV_NBRE    40      # The min number of spec
      events per atom in memory (if more, array is doubled, tripled, etc.).
117 setenv REFINE_ATTEMPTS     2       # The number of attempts to
      refine a generic event.
118 setenv MIN_REFINE_BARRIER 0.1      # Minimum barrier to be
      refined, all barriers lower are cloned.
119 setenv BARRIER_CHANGE_MIN 0.1      # Minimum barrier needed to
      use BARRIER_CHANGE_THRESH in eV.
120
121 ##### Mapping parameters #####
122 setenv MAX_CoM_DEVIATION    0.08     # Maximum change in the
      displacement of CoM when mapping.
123 setenv MAPPING_E_CUT       5.0E-1   # Maximum difference in change
      in energy.
124 setenv MAPPING_DELR        1.0      # Maximum change in
      displacement when mapping.
125
126 ##### Steepest Descent minimization #####
127 setenv MAX_ITER_SD          2000     # Maximum number of iterations
      during SD minimization.
128 setenv FTHRESHOLD_SD       1.0d-1   # Force Threshold when it is
      running SD algorithm.
129 setenv STEPSIZE             1.0d-4   # Step size of SD algorithm.
130
131 ##### FIRE minimization #####
132 setenv DTMAX_FIRE           0.15     # Maximum step size the
      minimizer is permitted to use in FIRE (Fast Inertial Relaxation
      Engine). Consider the stability of the system when setting in time
      units.
133 setenv MAX_ITER_FIRE        1500     # Maximum number of iterations
      during FIRE minimization.
134 setenv NORM_CRITERIUM       0.0005   # Convergence criterium [eV/A
      ].
135 setenv FMAX_CRITERIUM       0.0005   # Force tolerance after each
      stage.
136
137 ##### FIRE perpendicular hyperplane minimization #####
138 setenv MAX_ITER_FIRE_PERP   15       # Maximum number of iterations
      during FIRE perpendicular hyperplane minimization.
139 setenv FOR_LEAVING_BASIN_USE_FIRE .false. # When relaxing perp for
      leaving harmonic basin if .true. choose FIRE algorithm else SD.
140

```

```

141 ##### Event catalogue file #####
142 setenv UFT_EVENTFILENAME 'events.uft' # The name of the uft event
      file.
143 setenv USE_TXT_EVENTFILE .false. # IF true, will create a
      directory containing event files.
144 setenv CONVERT_TO_UFT .false. # IF true, the program will
      read the txt event files and create a new uft file.
145 setenv CONVERT_TO_TXT .false. # IF true, the program will
      read the uft file and copy the catalogue to txt files (use for
      vizualization).
146 setenv EVENTS_DIR 'EVENTS_DIR' # The name of the directory
      where the event txt files are stored.
147 setenv EVLIST_DIR 'EVLIST_DIR' # The name of the directory
      where the event list per KMC step files are stored.
148 setenv MINSAD_DIR 'MINSAD_DIR' # The name of the directory
      where the local minima and saddle point txt files are stored.
149
150
151 #----- ART PARAMETERS -----#
152
153 ##### Direction inversion in iterative subspace #####
154 setenv USE_DIIS .false. # Use DIIS for the final
      convergence to saddle.
155 setenv DIIS_FORCE_THRESHOLD 0.2 # Force threshold for
      convergence.
156 setenv DIIS_MEMORY 6 # Number of vectors keeps in
      memory for algorithm.
157 setenv DIIS_MAXITER 150 # Maximum number of
      interactions for the DIIS scheme.
158 setenv DIIS_CHECK_EIGENVECTOR .true. # Check that the final state
      is indeed a saddle.
159 setenv DIIS_STEP_SIZE 0.005 # Step size for the position.
160
161 ##### ART options #####
162 setenv SADDLE_PUSH_PARAM 0.1 # The fraction of the initial-
      saddle distance used to push saddle config. away from initial minimum
      .
163 setenv TYPE_OF_EVENTS local # Initial move for events -
      global or local.
164 setenv RADIUS_INITIAL_DEFORMATION 2.5 # Cutoff for local-move (in
      angstroms).
165 setenv EIGENVALUE_THRESHOLD -1.0 # Eigenvalue threshold for
      leaving basin.
166 setenv EXIT_FORCE_THRESHOLD 0.05 # Threshold for convergence at
      saddle point.
167 setenv FINE_EXIT_FORCE_THRESHOLD 0.05 # finner Threshold for
      convergence at saddle point.
168 setenv MAX_PERP_MOVES_BASIN 2 # Maximum number of
      perpendicular steps leaving basin.
169 setenv MIN_NUMBER_KSTEPS 2 # Min. number of ksteps before
      calling lanczos.
170 setenv INCREMENT_SIZE 0.3 # Overall scale for the
      increment moves in activation.

```

```
171 setenv BASIN_FACTOR          2      # Factor multiplying the
      reference length Increment Size when pushing for leaving the harmonic
      basin.
172 setenv INITIAL_STEP_SIZE    0.01   # Size of initial displacement
      in Angstroms.
173 setenv MAX_PERP_MOVES_ACTIV  6      # Maximum number of
      perpendicular steps during activation.
174 setenv FORCE_THRESHOLD_PERP_REL 0.1  # Threshold for perpendicular
      relaxation.
175 setenv MAX_ITER_BASIN       15     # Maximum number of interaction
      for leaving the basin (kter).
176 setenv MAX_ITER_ACTIVATION   50     # Maximum number of interaction
      during activation (iter).
177 setenv NUMBER_LANCZOS_VECTORS 15    # Number of vectors included
      in lanczos procedure.
178 setenv LANCZOS_STEP          0.001  # Size of the step for the
      numerical derivative.
179 setenv CHECK_LANCZOS_STAB   .true.  # Check lanczos stability over
      200 steps, each iteration uses previous lanczos vector.
180
181 ##### ART Print out #####
182 setenv PRINT_DETAILS         .true.  # Prints the details of
      activation and minimization.
183 setenv KPRINT                5      # Prints details of leaving
      the basin.
184 setenv IPRINT                10     # Prints details of the
      convergence to saddle.
185 setenv MPRINT                100   # Prints details of the
      minimization process.
186 setenv LPRINT                1     #
187
188 ##### Input #####
189 setenv FILECOUNTER           filecounter # File tracking the file (
      event) number - facultative.
190 setenv REFCONFIG             refconfig.dat # initial configuration file.
191
192
193 ##### ---> Run the simulation <--- #####
194 unlimit stacksize
195
196 # BE CAREFUL, THIS IS A SERIAL VERSION OF THE CODE:
197 #.././SRC/KMCART_gnu2_briaree_v1458      # On briaree.
198 #.././SRC/KMCART_mac_gfortran_v1458     # On mac.
199 #.././SRC/KMCART_ubuntu_gfortran_v1458  # On linux.
200 #.././SRC/KMCART_briareelammps_v1458    # On briaree (linux).
```

B.2 Initial atomic positions in kART

Listing B.2: Sample of the file containing the initial atom configuration (513.conf).

```
1 # IMPORTANT CONSIDERATIONS:
2 # - kART is sensitive to spaces.
3 # - It is not necessary to expand/compress the cell.
4
5 run_id:          1                # Simulation ID.
6 total_energy:   -2220.9252        # Total energy of our system (kART
7                                     # calculates it without taking
8                                     # into account this value).
9 P    21.7238     21.7238     21.7238 # The size of the simulation box
10                                         # in the x, y and z dimension.
11 1 -10.184849 -10.190365 -10.172633
12 1 -10.167592 -7.502172 -7.443832
13 1 -7.463367 -10.193221 -7.454985
14 1 -7.476008 -7.466695 -10.173070
15 1 -8.818031 -8.840989 -8.817096
16 1 -8.825151 -6.121489 -6.090086
17 1 -6.110574 -8.830760 -6.097852
18 1 -6.119136 -6.109441 -8.813921
19 1 -4.756899 -10.182005 -10.173372
20 1 -4.759200 -7.468214 -7.457517
21 1 -2.044258 -10.185446 -7.457572
22 1 -2.041460 -7.466959 -10.172047
23 1 -3.399936 -8.824905 -8.814856
24 1 -3.399607 -6.111382 -6.100751
25 1 -0.687580 -8.830600 -6.097223
26 1 -0.680349 -6.111123 -8.815012
27 1 -10.212839 -4.748354 -10.168611
28 1 -10.212686 -2.042667 -7.463225
29 1 -7.482266 -4.754216 -7.457028
30 1 -7.482550 -2.036844 -10.174863
31 1 -8.844634 -3.395549 -8.815871
32 1 -8.839781 -0.686213 -6.106348
33 1 -6.118939 -3.397563 -6.101819
34 1 -6.119077 -0.682004 -8.817974
35 1 -4.760585 -4.751873 -10.172339
36 1 -4.760294 -2.039915 -7.460077
37 1 -2.040526 -4.753914 -7.458606
38 1 -2.040178 -2.038382 -10.173693
39 1 -3.399312 -3.396060 -8.816231
40 1 -3.399397 -0.683384 -6.103698
41 1 -0.680368 -3.397439 -6.101417
42 1 -0.680111 -0.680914 -8.817356
43 1 -10.163664 -10.213179 -4.723789
44 1 -10.163904 -7.487998 -1.998836
45 1 -7.459982 -10.193947 -2.017928
46
47 # ... (Continues until finishing the total number of atoms) ...
```


B.3 Input file in LAMMPS

Listing B.3: Sample of input file in LAMMPS compatible with kART including detailed explanations (in.lammps).

```

1 # IMPORTANT CONSIDERATIONS:
2 # - Pure Si crystal structure need to initialize lammps in kART.
3 # - Need only to create the box not creating atoms.
4
5 #----- Customize LAMMPS parametes: -----#
6 log      log_kART.lammps # If you want to see output of lammps (Be
7                          # careful: huge file).
8
9 units    metal          # Establish de working units. For style metal,
10                          # these are the units:
11                          # - mass = grams/mole
12                          # - distance = Angstroms
13                          # - time = picoseconds
14                          # - energy = eV
15                          # - velocity = Angstroms/picosecond
16                          # - force = eV/Angstrom
17                          # - torque = eV
18                          # - temperature = Kelvin
19                          # - pressure = bars
20                          # - dynamic viscosity = Poise
21                          # - charge = multiple of electron charge
22                          # - dipole = charge*Angstroms
23                          # - electric field = volts/Angstrom
24                          # - density = gram/cm^dim
25 atom_style  atomic     # Atom style (angle or atomic or dipole or
26                          # bond or charge or electron or molecule...).
27
28 atom_modify  map array # Modify certain attributes of atoms defined
29                          # and stored within LAMMPS, in addition to
30                          # what is specified by the atom_style command.
31                          # The map keyword points how atom ID lookup
32                          # is done for molecular atom styles. When the
33                          # array value is used, each processor stores a
34                          # lookup table of length N, where N is the
35                          # largest atom ID in the system.
36
37 #----- Methods to initialize coord in lammps -----#
38 #      1) Using a input file:
39 read_data  cSi_4100atoms_0K_minimized.datafile # Indicates file name.
40
41 #      2) Creating a box and putting atoms randomly in it by using
42 #          lammps commands:
43 #region   box block 0 21.712  0 23.036  0 23.036 # ID, xlo xhi ylo yhi
44                          # zlo zhi (box size).
45 #create_box      1 Box          # Number of atom types,
46                          # region-ID.
47 #create_atoms    1 random 4100 124675 Box # Type1, Number of atoms,

```

```
48                                     # seed, region-ID.
49
50 #----- Other LAMMPS parametes: -----#
51
52 mass          1 28.085              # Set the mass for all atoms of
53                                     # one or more atom types. In our
54                                     # case, set the Si mass.
55
56 pair_style    tersoff               # The tersoff style computes a
57                                     # 3-body Tersoff potential for the
58                                     # energy E of a system of atoms.
59
60 pair_coeff    * * Si.tersoff3 Si    # Specify the pairwise force field
61                                     # coefficients for one or more
62                                     # pairs of atom types. The number
63                                     # and meaning of the coefficients
64                                     # depends on the pair style.
65
66 #----- Speeding up the simulation: -----#
67 # Those lines increase the speed of calculating forces by a factor of
68 # almost 2:
69 neighbor      0.0 bin
70 neigh_modify  delay 0 every 1 check no one 50 page 550
```

Appendix C

Analysis programs (C/C++)

Listing C.1: kART processing program for 0GPa samples (kART_crystal_analysis.cpp).

```
1 /**-----
2 | Program name : kART_analysis.cpp |
3 -----
4 | Purpose: characterize each step of a kART simulation by matching |
5 |           the input formation enthalpy with an internal database |
6 |           and also link each step with the suitable information |
7 |           provided by the simulation program in different files. |
8 |           It can be applied to I2/I3/I4/I5/I6 cells at 0GPa. |
9 -----
10 | Input files: allconf (heavy file which contains all the atom |
11 |              coordinates from each step), Energies.dat (file |
12 |              which contains the total energy value from each |
13 |              step among other data), Diffusion.dat (file which |
14 |              contains the total mean square displacement (msd) |
15 |              from each step among other data) and perfectcell |
16 |              (file which contains the reference network without |
17 |              defects). |
18 | Output files: kART_data.txt (file which contains all the |
19 |               information we need from each step, from left to |
20 |               right: KMC step, simulated time, CPU time, |
21 |               number of groups, number of displaced atoms, number |
22 |               of empty spaces, net number, previous total energy |
23 |               value (eV), current total energy value (eV), energy |
24 |               barrier (eV), formation enthalpy (eV), identified |
25 |               configuration label and msd (A)) and min* files |
26 |               (one file per KMC step which contains a fragment of |
27 |               the cell in each step with the Rasmol format, |
28 |               assigning the crystal atoms and the displaced atoms |
29 |               different labels). |
30 -----
31 | Created: 18-11-2015 |
32 | Last reviewed: 3-5-2016 |
33 -----**/
```

```
34
35 #include <math.h> // Mathematical library.
36 #include <stdio.h> // Input/Output operations.
37 #include <string.h> // Memory manipulation.
38 #include <stdlib.h> // For using exit.
39 #include <assert.h> // For using assert.
40
41 // -----
42 // ***** Initial parameters *****
43 // -----
44 // Firstly, we define the network which we are working with
45 // (uncomment the pertinent one).
46 // #define I2 2
47 // #define I3 3
48 // #define I4 4
49 #define I5 5
50 // #define I6 6
51
52 // Depending on the first choice, the number of atoms changes. Noted
53 // it is showed the cell sizes which we are working with.
54 // I2
55 #ifdef I2
56     #define NATOMS 1730
57     #define NRETIC 1728
58     #define NDEFECT 3458
59 #endif
60 // I3
61 #ifdef I3
62     #define NATOMS 1731
63     #define NRETIC 1728
64     #define NDEFECT 3459
65 #endif
66 // I4
67 #ifdef I4
68     #define NATOMS 4100
69     #define NRETIC 4096
70     #define NDEFECT 8196
71 #endif
72 // I5
73 #ifdef I5
74     #define NATOMS 4101
75     #define NRETIC 4096
76     #define NDEFECT 8197
77 #endif
78 // I6
79 #ifdef I6
80     #define NATOMS 4102
81     #define NRETIC 4096
82     #define NDEFECT 8198
83 #endif
84
85 // We establish the number of cells we are going to divide the
86 // network.
```

```
87 #define NCELLSX 18
88 #define NCELLSY 18
89 #define NCELLSZ 18
90
91 // We indicate the number of steps of our simulation + 1 (to include
92 // the initial minimization).
93 #define NSTEPS 1001 // (+1)
94
95 // Depending on the first choice, the reference formation enthalpy
96 // changes. Note it is showed the formation enthalpy values of the
97 // cells which we are working with.
98 #ifdef I2
99     #define REFENERGY -8009.425151 // Scaled for I2
100 #endif
101 #ifdef I3
102     #define REFENERGY -8014.054877 // Scaled for I3
103 #endif
104 #ifdef I4
105     #define REFENERGY -18981.8749 // Scaled for I4
106 #endif
107 #ifdef I5
108     #define REFENERGY -18986.5046 // Scaled for I5
109 #endif
110 #ifdef I6
111     #define REFENERGY -18991.1344 // Scaled for I6
112 #endif
113
114
115 // -----
116 // ***** Common variables *****
117 // -----
118 #define MAXNEIGH 40 // Maximum number of neighbours by each
119 // atom.
120 #define PERBOUNDX 1 // Periodic boundary conditions in X axis
121 // (they are assumed in Y and Z).
122 #define FACTOR 1.0 // Factor to adapt our simulation box to
123 // a given density.
124 #define NETDIST 0.25 // Distance to associate an atom with an
125 // empty space in normalized units.
126 #define GROUPNUMBER 100 // Maximum number of groups.
127 #define DISTVEC 1.025 // First neighbour distance in network
128 // units (a/2).
129
130 // Cubes in which we are going to divide our network. They are
131 // chosen as the minimum value to cover the biggest simulation box.
132 #define DIMCUBEX 222
133 #define DIMCUBEY 222
134 #define DIMCUBEZ 222
135
136 // Beginning of the network in X, Y and Z axis in normalized units.
137 #define INIREDX 0.25
138 #define INIREDY 0.25
139 #define INIREDZ 0.25
```

```
140
141
142 // -----
143 // ***** Global variables *****
144 // -----
145
146 // Distances for Si.
147 double semired = 2.715475; // a/2 distance in A.
148 double a4 = 1.3577375; // a/4 (layer separation
149 // in 100 direction).
150 // Auxiliar variables.
151 int i, j, k, l, m, ic, jc, kc;
152 int atom, natoms, numgrupo, nretic, ndesplazados, nion;
153 int ax_2, ay_2, az_2, ax, ay, az;
154 int atomi, atomj, atomki, ndefectos, defectos[NDEFECT], ax1, ax2, ay1,
    ay2, az1, az2;
155 // Variables to store the neighbours of each atoms, groups and defects.
156 int vecinos[NDEFECT][MAXNEIGH], ord[NDEFECT], defecto[NATOMS], grupo[
    NDEFECT];
157 int ndesplgrupo[GROUPNUMBER], nretgrupo[GROUPNUMBER];
158 int redocupada[NRETIC], reticuloat[NATOMS];
159 // Parameters to divide the network in cubes.
160 int ford[NRETIC], numvecdesc[NATOMS], cuboxi, cuboyi, cubozi;
161 int despdx, despy, despz, ord2[NATOMS], vecinos2[NATOMS][MAXNEIGH], ford2
    [NATOMS];
162 // Variable which keeps the track.
163 int kARTstep;
164 // Dumb variable.
165 int intdjunk;
166 // Network parameters.
167 double sidex, sidey, sidez, prodrij_rik, mNETDIST, NETDIST2;
168 double sidex2, sidey2, sidez2, msidex2, msidey2, msidez2;
169 double longcubex, longcubey, longcubez, distx, disty, distz;
170 double junk, distvec, mdistvec, distvec2, factor[NCELLSX*4];
171 int intsidex, intsidey, intsidez, intsidex_ax_2, intsidey_ay_2,
    intsidez_az_2, count;
172 // Atom coordinates.
173 double xat[NATOMS], yat[NATOMS], zat[NATOMS];
174 // Product coordinates.
175 double xat_prod[NATOMS], yat_prod[NATOMS], zat_prod[NATOMS];
176 // Differential coordinates.
177 double xat_def[NDEFECT], yat_def[NDEFECT], zat_def[NDEFECT];
178 // Imagen coordinates.
179 int ix[NATOMS], iy[NATOMS], iz[NATOMS];
180 // Defects coordinates and normalized distances.
181 double xdef[NDEFECT], ydef[NDEFECT], zdef[NDEFECT], distnormx, distnormy
    , distnormz;
182 // Auxiliar variables for atom coordinates.
183 double xij, yij, zij, xik, yik, zik, rij, rik, rij_rik, fcrij, fa, fr,
    r2;
184 double x, y, z, xl[NATOMS], xred1[NRETIC], yred1[NRETIC], zred1[NRETIC],
    xred2[NRETIC], yred2[NRETIC], zred2[NRETIC];
185 // Mean coordinates of each defect.
```

```

186 double xcm[GROUPNUMBER], ycm[GROUPNUMBER], zcm[GROUPNUMBER];
187 // Variables to calculate the distance between atoms.
188 double dist, xdif, ydif, zdif;
189 // Parameter which store the formation enthalpy of each kART step.
190 double eFormation;
191 // Maximum and minimum values of the defects and crystal coordinates.
192 double xmin, xmax, ymin, ymax, zmin, zmax, deltax, deltax, deltax;
193 double xmin_def, xmax_def, ymin_def, ymax_def, zmin_def, zmax_def;
194 // Dumb variable.
195 double djunk;
196 // Label for clasifiying the configuration.
197 int atomIDvector[NATOMS], label, found;
198 // Parameters of each kART step.
199 double cpuTime[NSTEPS], simTime[NSTEPS], oldenergy[NSTEPS], newenergy[
    NSTEPS], barrier[NSTEPS], msd[NSTEPS];
200 // Input files names.
201 char posiniav[80], def[80], Si_word[20], line[200];
202 // Input files.
203 FILE *posiniaverag, *perfectcell, *inART;
204 // Output files.
205 FILE *outkART, *inenergies, *indiffusion, *error_file, *F_defectos;
206
207 // Declared functions.
208 void agrupa(int); // ID of the defect to group.
209 double round(double X, int k); // Round X with k decimals.
210
211
212 // -----
213 // ***** void main() *****
214 // -----
215
216 // First of all, we are going to read the input files and
217 // create some dynamic data arrays.
218
219 // Beginning of the program.
220 void main()
221 //void main (int argc, char **argv) // For Unix systems
222 {
223     // We create cube[][] dynamically to avoid memory problems.
224     int*** cube = new int**[DIMCUBEX];
225     for (i=0; i<DIMCUBEX; i++)
226     {
227         cube[i] = new int*[DIMCUBEY];
228         for (j=0; j<DIMCUBEY; j++)
229             cube[i][j] = new int[DIMCUBEZ];
230     }
231
232     // We build our energy database for each working cell.
233     #ifdef I2
234         double OurDatabase[41]={6.106, 6.122, 6.146, 6.163, 6.216,
            6.224, 6.246, 6.281, 6.282, 6.292, 6.32, 6.347, 6.373, 6.418,
            6.46, 6.481, 6.602, 6.406, 6.604, 6.606, 6.627, 6.629,
            6.688, 6.731, 6.757, 6.773, 6.793, 6.833, 6.894, 6.910,

```

```
        6.954, 6.427, 6.542, 6.595, 6.637, 6.693, 6.781, 6.784,
        6.805, 6.852, 6.454};
235 #endif
236 #ifdef I3
237     double OurDatabase[55]={8.096, 8.250, 8.284, 8.316, 8.347,
        8.492, 8.690, 8.691, 8.401, 8.555, 8.604, 8.682, 8.704,
        8.731, 8.743, 8.787, 8.789, 8.805, 8.817, 8.829, 8.852,
        8.892, 8.904, 8.924, 8.947, 8.987, 9.035, 7.951, 8.287,
        8.318, 8.414, 8.464, 8.468, 8.512, 8.521, 8.546, 8.587,
        8.597, 8.602, 8.647, 8.658, 8.697, 8.699, 8.742, 8.812,
        8.511, 8.559, 8.524, 8.577, 8.684, 8.696, 8.737, 8.788,
        8.867, 9.510};
238 #endif
239 #ifdef I4
240     double OurDatabase[38]={8.208, 9.214, 9.974, 10.694, 9.312,
        9.541, 9.619, 9.799, 9.850, 9.784, 9.814, 9.897, 9.903,
        9.955, 9.957, 10.002, 10.172, 10.24, 9.818, 9.864, 9.872,
        9.943, 9.968, 9.998, 10.011, 10.041, 10.071, 10.082, 9.828,
        9.849, 9.891, 9.894, 9.894, 9.914, 9.950, 9.960, 9.976,
        11.158};
241 #endif
242 #ifdef I5
243     double OurDatabase[72]={10.396, 10.711, 10.712, 10.827, 10.912,
        10.978, 11.011, 11.101, 11.211, 11.271, 11.301, 11.341,
        11.342, 11.534, 11.578, 11.755, 11.167, 11.168, 11.219,
        11.234, 11.278, 11.558, 11.595, 11.637, 11.642, 11.669,
        11.846, 11.267, 11.441, 11.487, 11.536, 11.732, 11.845,
        11.876, 11.876, 12.061, 11.080, 11.883, 11.899, 11.972,
        12.106, 12.171, 12.185, 12.257, 12.263, 12.272, 11.689,
        11.716, 11.736, 11.782, 11.787, 11.814, 11.831, 11.841,
        11.848, 11.850, 11.856, 11.878, 11.914, 11.926, 11.933,
        11.939, 11.953, 11.956, 11.975, 11.984, 11.999, 12.021,
        12.037, 12.048, 12.063, 12.089};
244 #endif
245 #ifdef I6
246     double OurDatabase[21]={12.608, 12.907, 13.078, 13.079, 13.111,
        13.166, 13.218, 12.913, 12.935, 13.014, 13.084, 13.305,
        13.341, 12.822, 12.832, 12.833, 12.845, 13.006, 13.029,
        13.109, 13.177};
247 #endif
248
249 // Initialize cell parameters (m means minus and 2 means square):
250 natoms = NATOMS;
251 nretic = NRETIC;
252 sidex = (xmax - xmin)/semired;
253 sidey = (ymax - ymin)/semired;
254 sidez = (zmax - zmin)/semired;
255 deltax = -xmin;
256 deltay = -ymin;
257 deltaz = -zmin;
258 sidex2 = sidex/2.0;
259 sidey2 = sidey/2.0;
260 sidez2 = sidez/2.0;
```



```
261     msidex2 = -sidex2;
262     msidey2 = -sidey2;
263     msidez2 = -sidez2;
264     mNETDIST = -NETDIST;
265     NETDIST2 = NETDIST*NETDIST;
266     distvec = DISTVEC;
267     mdistvec = -distvec;
268     distvec2 = distvec*distvec;
269
270     // Cell dimensions according to our current network:
271     // I2
272     #ifdef I2
273         xmin = ymin = zmin = 0.0;
274         xmax = ymax = zmax = 32.592;
275     #endif
276     // I3
277     #ifdef I3
278         xmin = ymin = zmin = 0.0;
279         xmax = ymax = zmax = 32.592;
280     #endif
281     // I4
282     #ifdef I4
283         xmin = ymin = zmin = 0.0;
284         xmax = ymax = zmax = 43.456;
285     #endif
286     // I5
287     #ifdef I5
288         xmin = ymin = zmin = 0.0;
289         xmax = ymax = zmax = 43.456;
290     #endif
291     // I6
292     #ifdef I6
293         xmin = ymin = zmin = 0.0;
294         xmax = ymax = zmax = 43.456;
295     #endif
296
297     // Number of cubes in each direction.
298     intsidex = (int) (1.5*sidex/NETDIST);
299     intsidey = (int) (1.5*sidey/NETDIST);
300     intsidez = (int) (1.5*sidez/NETDIST);
301
302     longcubex = sidex/intsidex;
303     longcubey = sidey/intsidey;
304     longcubez = sidez/intsidez;
305
306     ax = (int) (1+NETDIST/longcubex);
307     ay = (int) (1+NETDIST/longcubey);
308     az = (int) (1+NETDIST/longcubez);
309
310     distnormx = NETDIST - (ax-1.0)*longcubex;
311     distnormy = NETDIST - (ay-1.0)*longcubey;
312     distnormz = NETDIST - (az-1.0)*longcubez;
313
```

```
314     ax_2 = ax+ax;
315     ay_2 = ay+ay;
316     az_2 = az+az;
317
318     intsidex_ax_2 = intsidex + ax_2;
319     intsidey_ay_2 = intsidey + ay_2;
320     intsidez_az_2 = intsidez + az_2;
321
322
323     if ((intsidex+ax+ax)>DIMCUBEX)
324     {
325         printf ("Error: The X dimension in cube is so small.");
326         exit(0);
327     }
328     if ((intsidey+ay+ay)>DIMCUBEY)
329     {
330         printf ("Error: The Y dimension in cube is so small.");
331         exit(0);
332     }
333     if ((intsidez+az+az)>DIMCUBEZ)
334     {
335         printf ("Error: The Z dimension in cube is so small.");
336         exit(0);
337     }
338
339     // We initialize position arrays.
340     for (i=0; i<NRETIC; i++)
341         xred1[i] = yred1[i] = zred1[i] = xred2[i] = yred2[i] = zred2[i]
            =0.0;
342
343     for (i=0; i<NATOMS; i++)
344     {
345         xat[i] = yat[i] = zat[i] =0.0;
346         xat_prod[i] = yat_prod[i] = zat_prod[i] =0.0;
347     }
348
349     for (i=0; i<NDEFECT; i++)
350         xdef[i] = ydef[i] = zdef[i] = 0.0;
351
352
353     // We read Energies.dat file.
354     inenergies = fopen("Energies.dat", "r");
355
356     // We ignore the first 2 lines of the file.
357     fgets(line,200,inenergies);
358     fgets(line,200,inenergies);
359
360     for (i=0; i<NSTEPS; i++)
361         fscanf(inenergies, "%d %lf %lf %lf %lf %lf %lf %lf %d %d %d %d\n",
            &intdjunk, &cpuTime[i], &oldenergy[i], &newenergy[i], &
            barrier[i], &djunk, &djunk, &simTime[i], &intdjunk, &intdjunk
            , &intdjunk, &intdjunk);
362
```

```
363     fclose(inenergies);
364
365     // We read Diffusion.dat file.
366     indiffusion = fopen("Diffusion.dat", "r");
367
368     // We ignore the first 2 lines of the file.
369     fgets(line,200,indiffusion);
370     fgets(line,200,indiffusion);
371
372     for (i=0; i<NSTEPS; i++)
373         fscanf(indiffusion, "%lf %lf %d %lf\n", &djunk, &msd[i], &
            intdjunk, &djunk);
374
375     fclose(indiffusion);
376
377     // We read our reference network.
378     perfectcell = fopen("perfectcell", "r");
379
380     // We ignore the first 9 lines of the file.
381     for (j=0; j<9; j++)
382         fgets(line,200,perfectcell);
383
384     for (i=0; i<nretic; i++)
385     {
386         fscanf(perfectcell, "%d %d %lf %lf %lf \n", &atom, &djunk, &x, &
            y, &z );
387
388         // We situate each atom inside our cell (for unwrapped
389         // coordinates).
390         while(x < xmin)
391             x+=(xmax - xmin);
392         while(x > xmax)
393             x-=(xmax - xmin);
394         while(y < ymin)
395             y+=(ymax - ymin);
396         while(y > ymax)
397             y-=(ymax - ymin);
398         while(z < zmin)
399             z+=(zmax - zmin);
400         while(z > zmax)
401             z-=(zmax - zmin);
402
403         // We displace to make the cell begin at 0.
404         x += deltax;
405         y += deltay;
406         z += deltaz;
407
408         // We convert the coordinates to network units.
409         xred1[atom-1] = x/semired;
410         yred1[atom-1] = y/semired;
411         zred1[atom-1] = z/semired;
412
413         // Correction network.
```

```
414     xred2[atom-1] = xred1[atom-1];
415     yred2[atom-1] = yred1[atom-1];
416     zred2[atom-1] = zred1[atom-1];
417 }
418
419 fclose(perfectcell);
420
421 // We analyse each KMC step with the next loop:
422 for (kARTstep=0; kARTstep <= NSTEPS ; kARTstep++)
423 {
424     // We print some information because the processing time
425     // can be long.
426     printf("Processing file %d\n", nion);
427
428     // We open allconf input file and min* output file.
429     sprintf(posiniav, "allconf");
430     sprintf(def, "min%d", nion);
431
432     // If we are dealing with the first KMC step, we open the
433     // file allconf. Note we are going to keep this file open
434     // when the loop is executing because we are going to read
435     // one atoms configuration per step.
436     if(kARTstep==0)
437         posiniaverag = fopen(posiniav, "r");
438
439     // We assure there is no errors.
440     if (posiniaverag == NULL)
441     {
442         error_file = fopen("error.txt","a");
443         fprintf(error_file,"Error to open %s \n", posiniav);
444         fclose(error_file);
445     }
446     else
447     {
448         // We open our output files.
449         F_defectos = fopen(def,"w");
450         outkART= fopen("kART_data.txt", "a");
451     }
452
453     // We read allconf file taking into account its format.
454     fscanf(posiniaverag, "%d\n", &natoms);
455     fgets(line,200,posiniaverag);
456     for (i=0; i<natoms; i++)
457     {
458         fscanf(posiniaverag, "%s %lf %lf %lf\n", &Si_word, &x, &y, &
            z);
459
460         // The same assignments as when we read our perfect
461         // cell.
462         while(x < xmin)
463             x+=(xmax - xmin);
464         while(x > xmax)
465             x-=(xmax - xmin);
```

```

466         while(y < ymin)
467             y+=(ymax - ymin);
468         while(y > ymax)
469             y-=(ymax - ymin);
470         while(z < zmin)
471             z+=(zmax - zmin);
472         while(z > zmax)
473             z-=(zmax - zmin);
474
475         x += deltax;
476         y += deltay;
477         z += deltaz;
478
479         assert(atom<=NATOMS);
480
481         xat[i] = x/semired;
482         yat[i] = y/semired;
483         zat[i] = z/semired;
484     }
485
486     // In case the KMC step is the last one, we close the
487     // file. If not, we will read the next configuration in
488     // the next loop execution.
489     if(kARTstep==NSTEPS)
490         fclose(posiniaverag);
491
492
493     // We are going to assign the cubes to the perfect cell
494     // coordinates, but before we initialize ford and cube
495     // to -1.
496     for(i=0; i<nretic; i++)
497         ford[i] = -1;
498
499     for (ic=0; ic<=intside_x_ax_2; ic++)
500         for (jc=0; jc<=intside_y_ay_2; jc++)
501             for (kc=0; kc<=intside_z_az_2; kc++)
502                 cube[ic][jc][kc] = -1;
503
504     for (i=0; i<nretic; i++)
505     {
506         ix[i] = ax + (int) (xred2[i]/longcubex);
507         iy[i] = ay + (int) (yred2[i]/longcubey);
508         iz[i] = az + (int) (zred2[i]/longcubez);
509
510         atom = cube[ix[i]][iy[i]][iz[i]];
511
512         despx = despy = despz = 0;
513
514         if (atom == -1)
515         {
516             cube[ix[i]][iy[i]][iz[i]] = i;
517
518             // If there are periodic boundary conditions.

```

```
519         if (PERBOUNDX)
520         {
521             if (ix[i] >= intsidex)
522                 despx = -intsidex;
523             else if (ix[i] < ax_2)
524                 despx = intsidex;
525         }
526
527         if (iy[i] >= intsidey)
528             despy = -intsidey;
529         else if (iy[i] < ay_2)
530             despy=intsidey;
531
532         if (iz[i] >= intsidez)
533             despz = -intsidez;
534         else if (iz[i] < az_2)
535             despz = intsidez;
536
537         if (despx != 0)
538         {
539             cube[ix[i]+despx][iy[i]][iz[i]]=i;
540
541             if (despy != 0)
542             {
543                 cube[ix[i]][iy[i]+despy][iz[i]]=i;
544                 cube[ix[i]+despx][iy[i]+despy][iz[i]]=i;
545
546                 if (despz != 0)
547                 {
548                     cube[ix[i]][iy[i]][iz[i]+despz]=i;
549                     cube[ix[i]+despx][iy[i]][iz[i]+despz]=i;
550                     cube[ix[i]][iy[i]+despy][iz[i]+despz]=i;
551                     cube[ix[i]+despx][iy[i]+despy][iz[i]+despz]=
552                         i;
553                 }
554             }
555         else if (despz != 0)
556         {
557             cube[ix[i]][iy[i]][iz[i]+despz]=i;
558             cube[ix[i]+despx][iy[i]][iz[i]+despz]=i;
559         }
560     }
561 else
562
563
564     if (despy != 0)
565     {
566         cube[ix[i]][iy[i]+despy][iz[i]]=i;
567
568         if (despz != 0)
569         {
570             cube[ix[i]][iy[i]][iz[i]+despz]=i;
```

```

571             cube[ix[i]][iy[i]+despy][iz[i]+despz]=i;
572         }
573     }
574     else if (despz != 0)
575         cube[ix[i]][iy[i]][iz[i]+despz]=i;
576
577     }
578     else
579     {
580         if (ford[atom] == -1)
581             ford[atom] = i;
582         else
583         {
584             while(ford[atom] != -1)
585                 atom = ford[atom];
586             ford[atom] = i;
587         }
588     }
589 }
590
591 // We look for the closest perfect position.
592 for (i=0; i<NDEFECT; i++)
593     for (j=0; j<MAXNEIGH; j++)
594         vecinos[i][j] = -1;
595
596 for (i=0; i<nretic; i++)
597     redocupada[i] = -1;
598
599 for (i=0; i<natoms; i++)
600     reticuloat[i] = -1;
601
602 for (i=0; i<natoms; i++)
603     if( (xat[i]+yat[i]+zat[i]) != 0.0) // We avoid default
604         values.
605     {
606         cuboxi = ax + (int) (xat[i]/longcubex);
607         cuboyi = ay + (int) (yat[i]/longcubey);
608         cubozi = az + (int) (zat[i]/longcubez);
609
610         distx = xat[i]-(((int) (xat[i]/longcubex))*longcubex);
611         disty = yat[i]-(((int) (yat[i]/longcubey))*longcubey);
612         distz = zat[i]-(((int) (zat[i]/longcubez))*longcubez);
613
614         if (distx < distnormx)
615             ax1 = ax;
616         else
617             ax1 = ax-1;
618         if ((longcubex-distx) > distnormx)
619             ax2 = ax-1;
620         else
621             ax2 = ax;
622
623         if (disty < distnormy)

```

```

623         ay1 = ay;
624     else
625         ay1 = ay-1;
626     if ((longcubey-disty) > distnormy)
627         ay2 = ay-1;
628     else
629         ay2 = ay;
630
631     if (distz < distnormz)
632         az1 = az;
633     else
634         az1 = az-1;
635     if ((longcubez-distz) > distnormz)
636         az2 = az-1;
637     else
638         az2 = az;
639
640     ax1 = cuboxi-ax1;
641     ax2 = cuboxi+ax2;
642     ay1 = cuboyi-ay1;
643     ay2 = cuboyi+ay2;
644     az1 = cubozi-az1;
645     az2 = cubozi+az2;
646
647     for (ic=ax1; ic<=ax2; ic++)
648     {
649         for (jc=ay1; jc<=ay2; jc++)
650         {
651             for (kc=az1; kc<=az2; kc++)
652             {
653                 j = cube[ic][jc][kc];
654
655                 while (j != -1)
656                 {
657                     if (redocupada[j] == -1)
658                     {
659                         if (PERBOUNDX == 1)
660                         {
661                             xij = xat[i] - xred2[j];
662                             // Minimum image in X direction.
663                             if (xij <= msidex2)
664                                 xij += sidex;
665                             else if (xij > sidex2)
666                                 xij -= sidex;
667                         }
668
669                         if ((xij < NETDIST) && (xij >
670                             mNETDIST))
671                         {
672                             yij = yat[i] - yred2[j];
673                             // Minimum image in Y direction.
674                             if (yij <= msidey2)
675                                 yij += sidey;

```



```

675         else if (yij > sidey2)
676             yij -= sidey;
677
678         if ((yij < NETDIST) && (yij >
679             mNETDIST))
680         {
681             zij = zat[i] - zred2[j];
682             // Minimum image in X
683             direction.
684             if (zij <= msidez2)
685                 zij += sidez;
686             else if (zij > sidez2)
687                 zij -= sidez;
688
689             if ((zij < NETDIST) && (zij
690                 > mNETDIST))
691             {
692                 r2 = xij*xij + yij*yij +
693                     zij*zij;
694
695                 if (r2 <= NETDIST2)
696                 {
697                     // We save the
698                     defects ID.
699                     redocupada[j] = i;
700                     reticuloat[i] = j;
701                 }
702             }
703         }
704     }
705 }
706
707
708
709 // In the next lines we determine the neighbours of each atom.
710 for (i=0; i<natoms; i++)
711     ord2[i] = 0;
712
713 for (i=0; i<natoms; i++)
714     if( (xat[i]+yat[i]+zat[i]) != 0.0)
715     {
716         // Cube of i atom.
717         cuboxi = ax + (int) (xat[i]/longcubex);
718         cuboyi = ay + (int) (yat[i]/longcubey);
719         cubozi = az + (int) (zat[i]/longcubez);
720
721         distx = xat[i]-(((int) (xat[i]/longcubex))*longcubex);
722         disty = yat[i]-(((int) (yat[i]/longcubey))*longcubey);

```

```
723         distz = zat[i]-(((int) (zat[i]/longcubez))*longcubez);
724
725         if (distx < distnormx)
726             ax1 = ax;
727         else
728             ax1 = ax-1;
729         if ((longcubex-distx) > distnormx)
730             ax2 = ax-1;
731         else
732             ax2 = ax;
733
734         if (disty < distnormy)
735             ay1 = ay;
736         else
737             ay1 = ay-1;
738         if ((longcubey-disty) > distnormy)
739             ay2 = ay-1;
740         else
741             ay2 = ay;
742
743         if (distz < distnormz)
744             az1 = az;
745         else
746             az1 = az-1;
747         if ((longcubez-distz) > distnormz)
748             az2 = az-1;
749         else
750             az2 = az;
751
752         ax1 = cuboxi-ax1;
753         ax2 = cuboxi+ax2;
754         ay1 = cuboyi-ay1;
755         ay2 = cuboyi+ay2;
756         az1 = cubozi-az1;
757         az2 = cubozi+az2;
758
759         for (ic=ax1; ic<=ax2; ic++)
760         {
761             for (jc=ay1; jc<=ay2; jc++)
762             {
763                 for (kc=az1; kc<=az2; kc++)
764                 {
765                     j = cube[ic][jc][kc];
766
767                     while (j != -1)
768                     {
769                         if (j > i)
770                         {
771                             if (PERBOUNDX == 1)
772                             {
773                                 xij = xat[i] - xat[j];
774                                 if (xij <= msidex2)
775                                     xij += sidex;
```

```

776         else if (xij > sidex2)
777             xij -= sidex;
778     }
779
780     if ((xij < distvec) && (xij >
781         mdistvec))
782     {
783         yij = yat[i] - yat[j];
784
785         if (yij <= msidey2)
786             yij += sidey;
787         else if (yij > sidey2)
788             yij -= sidey;
789
790         if ((yij < distvec) && (yij >
791             mdistvec))
792         {
793             zij = zat[i] - zat[j];
794
795             if (zij <= msidez2)
796                 zij += sidez;
797             else if (zij > sidez2)
798                 zij -= sidez;
799
800             if ((zij < distvec) && (zij
801                 > mdistvec))
802             {
803                 r2 = xij*xij + yij*yij +
804                     zij*zij;
805
806                 if (r2 <= distvec2)
807                 {
808                     vecinos2[i][ord2[i]]
809                         = j;
810                     vecinos2[j][ord2[j]]
811                         = i;
812                     ord2[i]++;
813                     if (ord[i] ==
814                         MAXNEIGH)
815                     {
816                         printf ("The
817                             number of
818                             neighbours of
819                             %d is higher
820                             than %d\n",
821                             i, ord2[i]);
822                         exit (0);
823                     }
824                     ord2[j]++;
825                     if (ord[j] ==
826                         MAXNEIGH)
827                     {

```

```

815                                     printf ("The
                                     number of
                                     neighbours of
                                     %d is higher
                                     than %d\n",
                                     i, ord2[j]);
816                                     exit(0);
817                                     }
818                                     }
819                                     }
820                                     }
821                                     }
822                                     }
823                                     j = ford2[j];
824                                     }
825                                     }
826                                     }
827                                     }
828                                     }
829
830 // We calculate the number of neighbour displaced atoms.
831 for (i=0; i<natoms; i++)
832 {
833     numvecdesc[i] = 0;
834     for (j=0; j<ord2[i]; j++)
835         if (reticuloat[vecinos2[i][j]] == -1)
836             numvecdesc[i]++;
837 }
838
839 // Now we are going to assign the defects.
840 ndefectos = 0;
841 for (i=0; i<natoms; i++)
842     if( (xat[i]+yat[i]+zat[i]) != 0.0) // We avoid default
843         values.
844         if (reticuloat[i] == -1)
845         {
846             defectos[ndefectos] = i;
847             xdef[ndefectos] = xat[i];
848             ydef[ndefectos] = yat[i];
849             zdef[ndefectos] = zat[i];
850             ndefectos++;
851         }
852 ndesplazados = ndefectos;
853
854 for (i=0; i<nretic; i++)
855     if (redocupada[i] == -1)
856     {
857         defectos[ndefectos] = i;
858         xdef[ndefectos] = xred2[i];
859         ydef[ndefectos] = yred2[i];
860         zdef[ndefectos] = zred2[i];
861         ndefectos++;

```

```
862         }
863
864         // Variables for the total amount of empty spaces and
865         // displaced atoms of each defect.
866         int **numDefVec = new int*[ndefectos];
867         for (i=0; i<ndefectos; i++)
868         {
869             numDefVec[i] = new int[2];
870             for (j=0; j<2; j++)
871                 numDefVec[i][j]=0;
872         }
873
874         // Thus, we can obtain the neighbours of the defects
875         // and determine if they are empty spaces or displaced
876         // atoms.
877         for (i=0; i<ndefectos; i++)
878             ord[i] = 0;
879
880         for (i=0; i<ndefectos; i++)
881         {
882             for (j=i+1; j<ndefectos; j++)
883             {
884                 xij = xdef[i] - xdef[j];
885
886                 if (PERBOUNDX == 1)
887                 {
888                     if (xij <= msidex2)
889                         xij += sidex;
890                     else if (xij > sidex2)
891                         xij -= sidex;
892                 }
893
894                 if ((xij < distvec) && (xij > mdistvec))
895                 {
896                     yij = ydef[i] - ydef[j];
897
898                     if (yij <= msidey2)
899                         yij += sidey;
900                     else if (yij > sidey2)
901                         yij -= sidey;
902
903                     if ((yij < distvec) && (yij > mdistvec))
904                     {
905                         zij = zdef[i] - zdef[j];
906
907                         if (zij <= msidez2)
908                             zij += sidez;
909                         else if (zij > sidez2)
910                             zij -= sidez;
911
912                         if ((zij < distvec) && (zij > mdistvec))
913                         {
914                             r2 = xij*xij + yij*yij + zij*zij;
```

```

915
916         if (r2 <= distvec2)
917         {
918             vecinos[i][ord[i]] = j;
919             ord[i]++;
920             if (ord[i] == MAXNEIGH)
921             {
922                 printf ("The number of neighbours of
923                     %d is higher than %d\n", i, ord2
924                     [i]);
925                 exit(0);
926             }
927             vecinos[j][ord[j]] = i;
928             ord[j]++;
929             if (ord[j] == MAXNEIGH)
930             {
931                 printf ("The number of neighbours of
932                     %d is higher than %d\n", i, ord2
933                     [j]);
934                 exit(0);
935             }
936             // If i and j have displaced atoms.
937             if(j < ndesplazados)
938             {
939                 numDefVec[i][1]++;
940                 numDefVec[j][1]++;
941             }
942             // Else j is empty space for sure, and i
943             // can be empty space or displaced atoms
944             .
945             else
946             {
947                 numDefVec[i][0]++;
948                 if(i < ndesplazados) //i is DA.
949                 numDefVec[j][1]++;
950                 else //i is ES.
951                 numDefVec[j][0]++;
952             }
953         }
954     }
955 }
956
957
958 // We group the defects.
959 for (i=0; i<NDEFECT; i++)
960 grupo[i] = -1;
961
962 for (i=0; i<GROUPNUMBER; i++)

```

```

963     {
964         ndesplgrupo[i] = 0;
965         nretgrupo[i] = 0;
966         xcm[i] = 0.0;
967         ycm[i] = 0.0;
968         zcm[i] = 0.0;
969     }
970
971     for (i=0; i<ndefectos; i++)
972     {
973         if (grupo[i] == -1)
974         {
975             numgrupo++;
976             assert (numgrupo <GROUPNUMBER);
977             if (numgrupo >= GROUPNUMBER)
978             {
979                 printf("The number of groups exceeds the maximum
980                        value.");
981                 exit(0);
982             }
983             agrupa(i);
984         }
985
986
987         // Finally, we characterize each defect group of
988         // the current the kART step.
989         for (i=0; i<numgrupo+1; i++)
990         {
991             // We obtain the formation enthalpy.
992             eFormation=newenergy[kARTstep]-REFENERGY;
993
994             // We initialize auxiliary variables.
995             found=0;
996             label=1;
997
998             // We determine the configuration associated
999             // with the current step by comparing its
1000             // energy value.
1001             for (l=0; l<sizeof(OurDatabase)/8; l++)
1002             {
1003                 // If the energies matches, we save the
1004                 // vector position.
1005                 if (round(eFormation, 3)==OurDatabase[l])
1006                 {
1007                     found=1;
1008                     count=l;
1009                 }
1010             }
1011
1012             // If there is no matching.
1013             if (found==0)
1014                 label=0;

```

```
1015
1016 // We only print the first group (late we
1017 // are going to see if there are more than
1018 // one group).
1019 if(i==0)
1020 {
1021 // We print the corresponding label according to our
1022 // database.
1023 // I2
1024 #ifdef I2
1025 if((ndesplgrupo[i] - nretgrupo[i])!=2)
1026     fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
1027         lf %lf %lf NO_GROUP %lf %lf\n", kARTstep,
1028             simTime[kARTstep], cpuTime[kARTstep],
1029             numgrupo+1, ndesplgrupo[i], nretgrupo[i],
1030             ndesplgrupo[i] - nretgrupo[i], oldenergy[
1031                 kARTstep], newenergy[kARTstep], barrier[
1032                     kARTstep], eFormation, pow((newenergy[
1033                         kARTstep]-REFENERGY),2), msd[kARTstep]);
1034 else if(label==0)
1035     fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
1036         lf %lf %lf OTHER %lf %lf\n", kARTstep,
1037             simTime[kARTstep], cpuTime[kARTstep],
1038             numgrupo+1, ndesplgrupo[i], nretgrupo[i],
1039             ndesplgrupo[i] - nretgrupo[i], oldenergy[
1040                 kARTstep], newenergy[kARTstep], barrier[
1041                     kARTstep], eFormation, pow((newenergy[
1042                         kARTstep]-REFENERGY),2), msd[kARTstep]);
1043 else if(count<17)
1044     fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
1045         lf %lf %lf P%d %lf %lf\n", kARTstep, simTime[
1046             kARTstep], cpuTime[kARTstep], numgrupo+1,
1047             ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
1048             - nretgrupo[i], oldenergy[kARTstep],
1049             newenergy[kARTstep], barrier[kARTstep],
1050             eFormation, count+1, pow((newenergy[kARTstep]
1051                 ]-REFENERGY),2), msd[kARTstep]);
1052 else if(count<31)
1053     fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
1054         lf %lf %lf F%d %lf %lf\n", kARTstep, simTime[
1055             kARTstep], cpuTime[kARTstep], numgrupo+1,
1056             ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
1057             - nretgrupo[i], oldenergy[kARTstep],
1058             newenergy[kARTstep], barrier[kARTstep],
1059             eFormation, count-17+1, pow((newenergy[
1060                 kARTstep]-REFENERGY),2), msd[kARTstep]);
1061 else if(count<40)
1062     fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
1063         lf %lf %lf C%d %lf %lf\n", kARTstep, simTime[
1064             kARTstep], cpuTime[kARTstep], numgrupo+1,
1065             ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
1066             - nretgrupo[i], oldenergy[kARTstep],
1067             newenergy[kARTstep], barrier[kARTstep],
```



```

        eFormation, count-31+1, pow((newenergy[
        kARTstep]-REFENERGY),2), msd[kARTstep]);
1034     else
1035     fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf D1 %lf %lf\n", kARTstep, simTime[
        kARTstep], cpuTime[kARTstep], numgrupo+1,
        ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
        - nretgrupo[i], oldenergy[kARTstep],
        newenergy[kARTstep], barrier[kARTstep],
        eFormation, pow((newenergy[kARTstep]-
        REFENERGY),2), msd[kARTstep]);

1036     #endif
1037
1038     // I3
1039     #ifdef I3
1040     if((ndesplgrupo[i] - nretgrupo[i])!=3)
1041     fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf NO_GROUP %lf %lf\n", kARTstep,
        simTime[kARTstep], cpuTime[kARTstep],
        numgrupo+1, ndesplgrupo[i], nretgrupo[i],
        ndesplgrupo[i] - nretgrupo[i], oldenergy[
        kARTstep], newenergy[kARTstep], barrier[
        kARTstep], eFormation, pow((newenergy[
        kARTstep]-REFENERGY),2), msd[kARTstep]);
1042     else if (label==0)
1043     fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf OTHER %lf %lf\n", kARTstep,
        simTime[kARTstep], cpuTime[kARTstep],
        numgrupo+1, ndesplgrupo[i], nretgrupo[i],
        ndesplgrupo[i] - nretgrupo[i], oldenergy[
        kARTstep], newenergy[kARTstep], barrier[
        kARTstep], eFormation, pow((newenergy[
        kARTstep]-REFENERGY),2), msd[kARTstep]);
1044     else if (count<8)
1045     fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf P%d %lf %lf\n", kARTstep, simTime[
        kARTstep], cpuTime[kARTstep], numgrupo+1,
        ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
        - nretgrupo[i], oldenergy[kARTstep],
        newenergy[kARTstep], barrier[kARTstep],
        eFormation, count+1, pow((newenergy[kARTstep
        ]-REFENERGY),2), msd[kARTstep]);
1046     else if (count<27)
1047     fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf F%d %lf %lf\n", kARTstep, simTime[
        kARTstep], cpuTime[kARTstep], numgrupo+1,
        ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
        - nretgrupo[i], oldenergy[kARTstep],
        newenergy[kARTstep], barrier[kARTstep],
        eFormation, count-8+1, pow((newenergy[
        kARTstep]-REFENERGY),2), msd[kARTstep]);
1048     else if (count<45)

```

```

1049         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
           lf %lf %lf C%d %lf %lf\n", kARTstep, simTime[
           kARTstep], cpuTime[kARTstep], numgrupo+1,
           ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
           - nretgrupo[i], oldenergy[kARTstep],
           newenergy[kARTstep], barrier[kARTstep],
           eFormation, count-27+1, pow((newenergy[
           kARTstep]-REFENERGY),2), msd[kARTstep]);
1050     else if(count<47)
1051         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
           lf %lf %lf E%d %lf %lf\n", kARTstep, simTime[
           kARTstep], cpuTime[kARTstep], numgrupo+1,
           ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
           - nretgrupo[i], oldenergy[kARTstep],
           newenergy[kARTstep], barrier[kARTstep],
           eFormation, count-45+1, pow((newenergy[
           kARTstep]-REFENERGY),2), msd[kARTstep]);
1052     else if(count<54)
1053         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
           lf %lf %lf FC%d %lf %lf\n", kARTstep, simTime
           [kARTstep], cpuTime[kARTstep], numgrupo+1,
           ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
           - nretgrupo[i], oldenergy[kARTstep],
           newenergy[kARTstep], barrier[kARTstep],
           eFormation, count-47+1, pow((newenergy[
           kARTstep]-REFENERGY),2), msd[kARTstep]);
1054     else if(count<55)
1055         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
           lf %lf %lf 111 %lf %lf\n", kARTstep, simTime[
           kARTstep], cpuTime[kARTstep], numgrupo+1,
           ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
           - nretgrupo[i], oldenergy[kARTstep],
           newenergy[kARTstep], barrier[kARTstep],
           eFormation, pow((newenergy[kARTstep]-
           REFENERGY),2), msd[kARTstep]);

1056     #endif
1057
1058     // I4
1059     #ifdef I4
1060         if((ndesplgrupo[i] - nretgrupo[i])!=4)
1061             fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
           lf %lf %lf NO_GROUP %lf %lf\n", kARTstep,
           simTime[kARTstep], cpuTime[kARTstep],
           numgrupo+1, ndesplgrupo[i], nretgrupo[i],
           ndesplgrupo[i] - nretgrupo[i], oldenergy[
           kARTstep], newenergy[kARTstep], barrier[
           kARTstep], eFormation, pow((newenergy[
           kARTstep]-REFENERGY),2), msd[kARTstep]);
1062     else if(label==0)
1063         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
           lf %lf %lf OTHER %lf %lf\n", kARTstep,
           simTime[kARTstep], cpuTime[kARTstep],
           numgrupo+1, ndesplgrupo[i], nretgrupo[i],

```

```

        ndesplgrupo[i] - nretgrupo[i], oldenergy[
        kARTstep], newenergy[kARTstep], barrier[
        kARTstep], eFormation, pow((newenergy[
        kARTstep]-REFENERGY),2), msd[kARTstep]);
1064     else if(count<4)
1065         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf F%d %lf %lf\n", kARTstep, simTime[
        kARTstep], cpuTime[kARTstep], numgrupo+1,
        ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
        - nretgrupo[i], oldenergy[kARTstep],
        newenergy[kARTstep], barrier[kARTstep],
        eFormation, count+1, pow((newenergy[kARTstep]
        )-REFENERGY),2), msd[kARTstep]);
1066     else if(count<9)
1067         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf CC%d %lf %lf\n", kARTstep, simTime
        [kARTstep], cpuTime[kARTstep], numgrupo+1,
        ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
        - nretgrupo[i], oldenergy[kARTstep],
        newenergy[kARTstep], barrier[kARTstep],
        eFormation, count-4+1, pow((newenergy[
        kARTstep]-REFENERGY),2), msd[kARTstep]);
1068     else if(count<18)
1069         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf CL%d %lf %lf\n", kARTstep, simTime
        [kARTstep], cpuTime[kARTstep], numgrupo+1,
        ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
        - nretgrupo[i], oldenergy[kARTstep],
        newenergy[kARTstep], barrier[kARTstep],
        eFormation, count-9+1, pow((newenergy[
        kARTstep]-REFENERGY),2), msd[kARTstep]);
1070     else if(count<28)
1071         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf T%d %lf %lf\n", kARTstep, simTime[
        kARTstep], cpuTime[kARTstep], numgrupo+1,
        ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
        - nretgrupo[i], oldenergy[kARTstep],
        newenergy[kARTstep], barrier[kARTstep],
        eFormation, count-18+1, pow((newenergy[
        kARTstep]-REFENERGY),2), msd[kARTstep]);
1072     else if(count<37)
1073         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf FC%d %lf %lf\n", kARTstep, simTime
        [kARTstep], cpuTime[kARTstep], numgrupo+1,
        ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
        - nretgrupo[i], oldenergy[kARTstep],
        newenergy[kARTstep], barrier[kARTstep],
        eFormation, count-28+1, pow((newenergy[
        kARTstep]-REFENERGY),2), msd[kARTstep]);
1074     else if(count<38)
1075         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf 111 %lf %lf\n", kARTstep, simTime[
        kARTstep], cpuTime[kARTstep], numgrupo+1,

```

```

        ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
        - nretgrupo[i], oldenergy[kARTstep],
        newenergy[kARTstep], barrier[kARTstep],
        eFormation, pow((newenergy[kARTstep]-
        REFENERGY),2), msd[kARTstep]);
1076         #endif
1077
1078         // I5
1079         #ifdef I5
1080             if((ndesplgrupo[i] - nretgrupo[i])!=5)
1081                 fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
                    lf %lf %lf NO_GROUP %lf %lf\n", kARTstep,
                    simTime[kARTstep], cpuTime[kARTstep],
                    numgrupo+1, ndesplgrupo[i], nretgrupo[i],
                    ndesplgrupo[i] - nretgrupo[i], oldenergy[
                    kARTstep], newenergy[kARTstep], barrier[
                    kARTstep], eFormation, pow((newenergy[
                    kARTstep]-REFENERGY),2), msd[kARTstep]);
1082             else if(label==0)
1083                 fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
                    lf %lf %lf OTHER %lf %lf\n", kARTstep,
                    simTime[kARTstep], cpuTime[kARTstep],
                    numgrupo+1, ndesplgrupo[i], nretgrupo[i],
                    ndesplgrupo[i] - nretgrupo[i], oldenergy[
                    kARTstep], newenergy[kARTstep], barrier[
                    kARTstep], eFormation, pow((newenergy[
                    kARTstep]-REFENERGY),2), msd[kARTstep]);
1084             else if(count<16)
1085                 fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
                    lf %lf %lf F%d %lf %lf\n", kARTstep, simTime[
                    kARTstep], cpuTime[kARTstep], numgrupo+1,
                    ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
                    - nretgrupo[i], oldenergy[kARTstep],
                    newenergy[kARTstep], barrier[kARTstep],
                    eFormation, count+1, pow((newenergy[kARTstep
                    ]-REFENERGY),2), msd[kARTstep]);
1086             else if(count<27)
1087                 fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
                    lf %lf %lf FC%d %lf %lf\n", kARTstep, simTime
                    [kARTstep], cpuTime[kARTstep], numgrupo+1,
                    ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
                    - nretgrupo[i], oldenergy[kARTstep],
                    newenergy[kARTstep], barrier[kARTstep],
                    eFormation, count-16+1, pow((newenergy[
                    kARTstep]-REFENERGY),2), msd[kARTstep]);
1088             else if(count<36)
1089                 fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
                    lf %lf %lf CF%d %lf %lf\n", kARTstep, simTime
                    [kARTstep], cpuTime[kARTstep], numgrupo+1,
                    ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
                    - nretgrupo[i], oldenergy[kARTstep],
                    newenergy[kARTstep], barrier[kARTstep],

```

```

        eFormation, count-27+1, pow((newenergy[
        kARTstep]-REFENERGY),2), msd[kARTstep]);
1090     else if (count<46)
1091         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf CL%d %lf %lf\n", kARTstep, simTime
        [kARTstep], cpuTime[kARTstep], numgrupo+1,
        ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
        - nretgrupo[i], oldenergy[kARTstep],
        newenergy[kARTstep], barrier[kARTstep],
        eFormation, count-36+1, pow((newenergy[
        kARTstep]-REFENERGY),2), msd[kARTstep]);
1092     else if (count<72)
1093         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf CC%d %lf %lf\n", kARTstep, simTime
        [kARTstep], cpuTime[kARTstep], numgrupo+1,
        ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
        - nretgrupo[i], oldenergy[kARTstep],
        newenergy[kARTstep], barrier[kARTstep],
        eFormation, count-46+1, pow((newenergy[
        kARTstep]-REFENERGY),2), msd[kARTstep]);
1094     #endif
1095
1096     //I6
1097     #ifdef I6
1098         if((ndesplgrupo[i] - nretgrupo[i])!=6)
1099             fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf NO_GROUP %lf %lf\n", kARTstep,
        simTime[kARTstep], cpuTime[kARTstep],
        numgrupo+1, ndesplgrupo[i], nretgrupo[i],
        ndesplgrupo[i] - nretgrupo[i], oldenergy[
        kARTstep], newenergy[kARTstep], barrier[
        kARTstep], eFormation, pow((newenergy[
        kARTstep]-REFENERGY),2), msd[kARTstep]);
1100     else if (etiqueta==0)
1101         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf OTHER %lf %lf\n", kARTstep,
        simTime[kARTstep], cpuTime[kARTstep],
        numgrupo+1, ndesplgrupo[i], nretgrupo[i],
        ndesplgrupo[i] - nretgrupo[i], oldenergy[
        kARTstep], newenergy[kARTstep], barrier[
        kARTstep], eFormation, pow((newenergy[
        kARTstep]-REFENERGY),2), msd[kARTstep]);
1102     else if (count<7)
1103         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
        lf %lf %lf F%d %lf %lf\n", kARTstep, simTime[
        kARTstep], cpuTime[kARTstep], numgrupo+1,
        ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
        - nretgrupo[i], oldenergy[kARTstep],
        newenergy[kARTstep], barrier[kARTstep],
        eFormation, count+1, pow((newenergy[kARTstep
        ]-REFENERGY),2), msd[kARTstep]);
1104     else if (count<13)

```

```

1105         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
           lf %lf %lf C%d %lf %lf\n", kARTstep, simTime[
           kARTstep], cpuTime[kARTstep], numgrupo+1,
           ndesplgrupo[i], nretgrupo[i], ndesplgrupo[i]
           - nretgrupo[i], oldenergy[kARTstep],
           newenergy[kARTstep], barrier[kARTstep],
           eFormation, count-7+1, pow((newenergy[
           kARTstep]-REFENERGY),2), msd[kARTstep]);
1106     else if(count<21)
1107         fprintf(outkART,"%d %.15lf %lf %d %d %d %d %lf %
           lf %lf %lf 11%d %lf %lf\n", kARTstep,
           simTime[kARTstep], cpuTime[kARTstep],
           numgrupo+1, ndesplgrupo[i], nretgrupo[i],
           ndesplgrupo[i] - nretgrupo[i], oldenergy[
           kARTstep], newenergy[kARTstep], barrier[
           kARTstep], eFormation, count-13+1, pow((
           newenergy[kARTstep]-REFENERGY),2), msd[
           kARTstep]);

1108         #endif
1109     }
1110 }
1111
1112 fclose(outkART);
1113
1114
1115 // In addition, we save the coordinate files with
1116 // the crystal atoms and the displaced atoms. Empty
1117 // spaces are not represented as they are not our
1118 // aim.
1119
1120 j=0; // Atom counter for Rasmol.
1121
1122 // We adapt defect coordinates to network distance.
1123 xmax_def=xcm[0]*semired+(sidex*semired)/2;
1124 xmin_def=xcm[0]*semired-(sidex*semired)/2;
1125 ymax_def=ycm[0]*semired+(sidey*semired)/2;
1126 ymin_def=ycm[0]*semired-(sidey*semired)/2;
1127 zmax_def=zcm[0]*semired+(sidez*semired)/2;
1128 zmin_def=zcm[0]*semired-(sidez*semired)/2;
1129
1130 // We apply boundary conditions for each defect.
1131 for (i=0; i<ndefectos; i++)
1132 {
1133     xat_def[i]=xdef[i]*semired;
1134     while(xat_def[i] < xmin_def)
1135         xat_def[i]+=(xmax_def - xmin_def);
1136     while(xat_def[i] > xmax_def)
1137         xat_def[i]-=(xmax_def - xmin_def);
1138
1139     yat_def[i]=ydef[i]*semired;
1140     while(yat_def[i] < ymin_def)
1141         yat_def[i]+=(ymax_def - ymin_def);
1142     while(yat_def[i] > ymax_def)

```

```
1143         yat_def[i]-=(ymax_def - ymin_def);
1144
1145         zat_def[i]=zdef[i]*semired;
1146         while(zat_def[i] < zmin_def)
1147             zat_def[i]+=(zmax_def - zmin_def);
1148         while(zat_def[i] > zmax_def)
1149             zat_def[i]-=(zmax_def - zmin_def);
1150     }
1151
1152     // We calculate the distance between the
1153     // defect and the rest of the atoms because
1154     // we are going to represent only a fragment
1155     // of the cell in order to see the defects
1156     // properly. Note that we also apply boundary
1157     // to calculate de difference.
1158     for (i=0; i<natoms; i++)
1159     {
1160         xat_prod[i]=xat[i]*semired;
1161         while(xat_prod[i] < xmin_def)
1162             xat_prod[i]+=(xmax_def - xmin_def);
1163         while(xat_prod[i] > xmax_def)
1164             xat_prod[i]-=(xmax_def - xmin_def);
1165
1166         yat_prod[i]=yat[i]*semired;
1167         while(yat_prod[i] < ymin_def)
1168             yat_prod[i]+=(ymax_def - ymin_def);
1169         while(yat_prod[i] > ymax_def)
1170             yat_prod[i]-=(ymax_def - ymin_def);
1171
1172         zat_prod[i]=zat[i]*semired;
1173         while(zat_prod[i] < zmin_def)
1174             zat_prod[i]+=(zmax_def - zmin_def);
1175         while(zat_prod[i] > zmax_def)
1176             zat_prod[i]-=(zmax_def - zmin_def);
1177
1178         xdif=abs(xcm[0]*semired-xat_prod[i]);
1179         ydif=abs(ycm[0]*semired-yat_prod[i]);
1180         zdif=abs(zcm[0]*semired-zat_prod[i]);
1181
1182         // We represent a 7 angstroms cubic
1183         // box around a given defect position.
1184         if((xdif<7)&&(ydif<7)&&(zdif<7))
1185             j++;
1186     }
1187
1188     // Now we are going to print the atoms
1189     // according to the Rasmol format (see
1190     // http://www.openrasmol.org/doc/rasmol.html
1191     // for more information).
1192
1193     // Number of atoms (DA + crystal atoms).
1194     fprintf(F_defectos, "%d \n\n", j + ndesplazados);
1195
```

```
1196
1197     for (j=0; j<numgrupo+1; j++)
1198     {
1199         // We print the coordinates for displaced
1200         // atoms.
1201         for (i=0; i<ndesplazados; i++)
1202             if (grupo[i] == j)
1203                 fprintf(F_defectos, "si0 %lf %lf %lf -5\n", xat_def[
1204                     i], yat_def[i], zat_def[i]);
1205
1206         // We print the coordinates for displaced
1207         // atoms with other colour label.
1208         for (i=ndesplazados; i<ndefectos; i++)
1209             if (grupo[i] == j)
1210                 fprintf(F_defectos, "si0 %lf %lf %lf -5\n", xat_def[
1211                     i], yat_def[i], zat_def[i]);
1212
1213         // And we print the atoms which are crystal and
1214         // are within the nose radius with other label.
1215         for (i=0; i<natoms; i++)
1216         {
1217             xdif=abs(xcm[0]*semired-xat_prod[i]);
1218             ydif=abs(ycm[0]*semired-yat_prod[i]);
1219             zdif=abs(zcm[0]*semired-zat_prod[i]);
1220
1221             if((xdif<7)&&(ydif<7)&&(zdif<7))
1222                 fprintf(F_defectos, "si0 %lf %lf %lf 0\n", xat_prod[
1223                     i], yat_prod[i], zat_prod[i]);
1224         }
1225
1226         // We remove dynamic arrays for each kART step.
1227         for (i=0; i<ndefectos; i++)
1228         {
1229             delete [] numDefVec[i];
1230             numDefVec[i] = NULL;
1231         }
1232         delete [] numDefVec;
1233         numDefVec = NULL;
1234
1235     } // End of the global loop for each kART step.
1236
1237
1238     // We delete the dynamic arrays from the whole
1239     // kART simulation.
1240     for (i=0; i<DIMCUBEX; i++)
1241     {
1242         for (j=0; j<DIMCUBEY; j++)
1243         {
1244             delete [] cube[i][j];
1245             cube[i][j] = NULL;
1246         }
1247     }
```



```
1246     }
1247
1248     delete [] cube[i];
1249     cube[i] = NULL;
1250 }
1251 delete [] cube;
1252 cube = NULL;
1253
1254 }
1255
1256
1257 // Function to group atoms by neighbourhood.
1258 void agrupa (int n)
1259 {
1260     int j;
1261
1262     grupo[n] = numgrupo;
1263
1264     xcm[numgrupo] = xdef[n];
1265     ycm[numgrupo] = ydef[n];
1266     zcm[numgrupo] = zdef[n];
1267
1268     if (n<ndesplazados)
1269         ndesplgrupo[numgrupo]++;
1270     else
1271         nretgrupo[numgrupo]++;
1272
1273     for (j=0; j<ord[n]; j++)
1274         if (grupo[vecinos[n][j]] == -1)
1275             agrupa(vecinos[n][j]);
1276 }
1277
1278 // Function to round X with k decimals.
1279 double round(double X, int k)
1280 {
1281     return floor( pow(double (10),k)*X + 0.5) / pow(double (10),k) ;
1282 }
```

Bibliography

- [1] Yoshio Nishi and Robert Doering. *Handbook of Semiconductor Manufacturing Technology*. CRC Press, 2nd Edition 2008. ISBN: 978-1-57444-675-3.
- [2] Official page of the International Technology Roadmap for Semiconductors. <http://www.itrs2.net>.
- [3] H.J. Gossmann D. J. Eaglesham, P. A. Stolck and J. M. Poate. Implantation and transient b diffusion in si: The source of the interstitials. *Applied Physics Letters*, 65:2305, 1994. doi:10.1063/1.112725.
- [4] B. Colombeau M. Omri B. de Mauduit A. Claverie L. F. Giles F. Cristiano, J. Grisolia and N. E. B. Cowern. Formation energies and relative stability of perfect and faulted dislocation loops in silicon. *Journal of Applied Physics*, 87:8420, 2000. doi:10.1063/1.373557.
- [5] S. Takeda. Structure analysis of defects in nanometer space inside a crystal: creation and agglomeration of point defects in si and ge revealed by high-resolution electron microscopy. *Microscopy Research and Technique*, 40(4):313, 1998. doi:10.1002/(SICI)1097-0029(19980215)40:4<313::AID-JEMT6>3.0.CO;2-S.
- [6] K. Huet F. Mazzamuto G. Fisicaro A. La Magna M. Quillec N. Cherkashin H. Wang S. Duguay Y. Qiu, F. Cristiano and D.C. Blavette. Extended defects formation in nanosecond laser-annealed ion implanted silicon. *Nano Letters*, 14:1769, 2014. doi:10.1021/nl4042438.
- [7] J. Y. Cheng, D. J. Eaglesham, D. C. Jacobson, P. A. Stolck, J. L. Benton, and J. M. Poate. Formation of extended defects in silicon by high energy implantation of b and p. *Journal of Applied Physics*, 80(4):2105–2112, 1996. doi:10.1063/1.363103.
- [8] Ignacio Martín Bragado. *Atomistic simulation of processes for microelectronics*. Department of Electricity and Electronics, University of Valladolid, 2004. PhD thesis

http://www.ele.uva.es/~simulacion/update2005/Martin-Bragado_Tesis2004.pdf.

- [9] Maria Aboy, Iván Santos, Lourdes Pelaz, Luis A. Marqués, and Pedro López. Modeling of defects, dopant diffusion and clustering in silicon. *Journal of Computational Electronics*, 13(1):40–58, 2014. doi:10.1007/s10825-013-0512-5.
- [10] Luis A. Marqués, María Aboy, Manuel Ruiz, Iván Santos, Pedro López, and Lourdes Pelaz. Molecular dynamics simulation of the early stages of self-interstitial clustering in silicon. *Materials Science in Semiconductor Processing*, 42, Part 2:235–238, 2016. E-MRS Spring Meeting 2015 Symposium Z: Nanomaterials and processes for advanced semiconductor CMOS devices. doi:10.1016/j.mssp.2015.07.020.
- [11] Ibrahim Avci, Mark E. Law, Erik Kuryliw, Antonio F. Saavedra, and Kevin S. Jones. Modeling extended defect ($\{311\}$ and dislocation) nucleation and evolution in silicon. *Journal of Applied Physics*, 95(5):2452–2460, 2004. doi:10.1063/1.1645644.
- [12] Alp H. Gencer and Scott T. Dunham. A combined model for $\{311\}$ defect and dislocation loop evolution: Analytical formulation of kinetic precipitation model. *Journal of Applied Physics*, 91(5):2883–2889, 2002. doi:10.1063/1.1446223.
- [13] P. Castrillo, I. Martin-Bragado, R. Pinacho, M. Jaraiz, J.E. Rubio, K.R.C. Mok, F.J. Miguel-Herrero, and J. Barbolla. Physically based modeling of dislocation loops in ion implantation processing in silicon. *Materials Science and Engineering B: Solid-State Materials for Advanced Technology*, 124-125(SUPPL.):404–408, 2005. doi:10.1016/j.mseb.2005.08.119.
- [14] F. Cristiano, N. Cherkashin, X. Hebras, P. Calvo, Y. Lamrani, E. Scheid, B. de Mauduit, B. Colombeau, W. Lerch, S. Paul, and A. Claverie. Ion beam induced defects in crystalline silicon. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 216:46–56, Feb 2004. doi:10.1016/j.nimb.2003.11.019.
- [15] N.E.B. Cowern, G. Mannino, P.A. Stolk, F. Roozeboom, H.G.A. Huizing, J.G.M. van Berkum, F. Cristiano, A. Claverie, and M. Jaraíz. Energetics of Self-Interstitial Clusters in Si. *Physical Review Letters*, 82(22):4460–4463, 1999. doi:10.1103/PhysRevLett.82.4460.

- [16] S. Takeda N. Arai and M. Kohyama. Self-Interstitial Clustering in Crystalline Silicon. *Physical Review Letters*, 78(22):4265–4268, Jun 1997. doi:[10.1103/PhysRevLett.78.4265](https://doi.org/10.1103/PhysRevLett.78.4265).
- [17] Jeongnim Kim, Florian Kirchhoff, Wilfried G. Aulbur, John W. Wilkins, Furrukh S. Khan, and Georg Kresse. Thermally activated reorientation of di-interstitial defects in silicon. *Physical Review Letters*, 83(10):12, Sep 1999. arXiv:[9904031](https://arxiv.org/abs/9904031), doi:[10.1103/PhysRevLett.83.1990](https://doi.org/10.1103/PhysRevLett.83.1990).
- [18] Stefan Birner, Jeongnim Kim, David A. Richie, John W. Wilkins, Arthur F. Voter, and Thomas Lenosky. Accelerated dynamics simulations of interstitial-cluster growth. *Solid State Communications*, 120(7-8):279–282, Oct 2001. doi:[10.1016/S0038-1098\(01\)00391-X](https://doi.org/10.1016/S0038-1098(01)00391-X).
- [19] D.A. Richie, Jeongnim Kim, Stephen A. Barr, Kaden R.A. Hazzard, Richard Hennig, and John W. Wilkins. Complexity of small silicon self-interstitial defects. *Physical review letters*, 92(January):045501, Jan 2004. doi:[10.1103/PhysRevLett.92.045501](https://doi.org/10.1103/PhysRevLett.92.045501).
- [20] Jeongnim Kim, Florian Kirchhoff, John Wilkins, and Furrukh Khan. Stability of Si-Interstitial Defects: From Point to Extended Defects. *Physical Review Letters*, 84(3):503–506, Jan 2000. doi:[10.1103/PhysRevLett.84.503](https://doi.org/10.1103/PhysRevLett.84.503).
- [21] A. Bongiorno, L. Colombo, F. Cargnoni, C. Gatti, and M. Rosati. Evolution of energetics and bonding of compact self-interstitial clusters in Si. *EPL (Europhysics Letters)*, 50(5):608, Mar 2000. doi:[10.1209/epl/i2000-00313-4](https://doi.org/10.1209/epl/i2000-00313-4).
- [22] Sangheon Lee and Gyeong S. Hwang. Structure and stability of small compact self-interstitial clusters in crystalline silicon. *Physical Review B*, 77(8):085210, Feb 2008. doi:[10.1103/PhysRevB.77.085210](https://doi.org/10.1103/PhysRevB.77.085210).
- [23] Sangheon Lee and Gyeong S. Hwang. Growth and shape transition of small silicon self-interstitial clusters. *Physical Review B*, 78(4):045204, Jul 2008. doi:[10.1103/PhysRevB.78.045204](https://doi.org/10.1103/PhysRevB.78.045204).
- [24] Gyeong S. Hwang Sangheon Lee and Robert J. Bondi. Integrated atomistic modelling of interstitial defect growth in silicon. *Molecular Simulation*, 35(10-11):867–879, 2009. doi:[10.1080/08927020902929802](https://doi.org/10.1080/08927020902929802).

- [25] Sangheon Lee Robert J. Bondi and Gyeong S. Hwang. Theoretical characterization of silicon self-interstitial clusters in uniform strain fields. *Physical Review B*, 80(12):1–10, 2009. doi:[10.1103/PhysRevB.80.125202](https://doi.org/10.1103/PhysRevB.80.125202).
- [26] Sangheon Lee Robert J. Bondi and Gyeong S. Hwang. Prediction of the formation of stable periodic self-interstitial cluster chains $[(I_{\text{sub } 4})_{\text{sub } m}, m=1-4]$ in Si under biaxial strain. *Applied Physics Letters*, 94(26):264101, 2009. doi:[10.1063/1.3160545](https://doi.org/10.1063/1.3160545).
- [27] Sumeet S. Kapur and Talid Sinno. Entropic origins of stability in silicon interstitial clusters. *Applied Physics Letters*, 93(22), 2008. doi:[10.1063/1.3042096](https://doi.org/10.1063/1.3042096).
- [28] Sumeet S. Kapur and Talid Sinno. Detailed microscopic analysis of self-interstitial aggregation in silicon. I. Direct molecular dynamics simulations of aggregation. *Physical Review B*, 82(4):1–11, 2010. doi:[10.1103/PhysRevB.82.045205](https://doi.org/10.1103/PhysRevB.82.045205).
- [29] Sumeet S. Kapur and Talid Sinno. Detailed microscopic analysis of self-interstitial aggregation in silicon. II. Thermodynamic analysis of single clusters. *Physical Review B*, 82(4):1–11, 2010. doi:[10.1103/PhysRevB.82.045206](https://doi.org/10.1103/PhysRevB.82.045206).
- [30] Claire Y. Chuang, Andreas Sattler, and Talid Sinno. Thermodynamic and morphological analysis of large silicon self-interstitial clusters using atomistic simulations. *Journal of Applied Physics*, 117:135706, 2015. doi:[10.1063/1.4917049](https://doi.org/10.1063/1.4917049).
- [31] K.S. Jones F. Gutierrez and D.F. Downey. Defect evolution from low energy, amorphizing, germanium implants on silicon. *Materials Research Society Symposium Proceedings*, 669(J5.11):1–5, 2002. doi:[10.1557/PROC-669-J5.11](https://doi.org/10.1557/PROC-669-J5.11).
- [32] Official page of Multiscale Materials Modeling Group.
<http://www.ele.uva.es/~mmm/>.
- [33] Beta cluster web.
<http://beta.ele.uva.es/ganglia/>.
- [34] Gamma cluster web.
<http://gamma.ele.uva.es/ganglia/>.
- [35] Dieter W. Heermann. *Computer Simulation Methods in Theoretical Physics*. Springer, 2nd Edition 1990. ISBN: 978-3-540-52210-2.

- [36] Luís A. Marqués. *Simulation of sputtering and crystallization due to ionic implantation in Silicon*. Department of Electricity and Electronics, University of Valladolid, Oct 1993. PhD thesis.
- [37] Lulla Rosso and Anne F. de Baas. *What makes a material function? Let me compute the ways... - Modelling in FP7 NMP Programme Materials projects*. Springer, Directorate-General for Research and Innovation, European Commission, 3rd Version 2012. ISBN: 978-92-79-26597-6.
http://ec.europa.eu/research/industrial_technologies/pdf/modelling-brochure_en.pdf.
- [38] Iván Santos. *Multiscale modeling of dopant implantation and diffusion in crystalline and amorphous silicon*. Department of Electricity and Electronics, University of Valladolid, Mar 2010. PhD thesis
<http://www.ele.uva.es/~ivasan/research.html#PhD>.
- [39] Jarosaw Meller and Roman Petrenko. Molecular dynamics. *Encyclopedia of Life Science, Nature*, Mar 2010. doi:10.1002/9780470015902.a0003048.pub2.
- [40] David P. Landau and Kurt Binder. *A guide to Monte Carlo simulations in Statistical Physics*. Cambridge University Press, 3rd Edition 2009. ISBN: 978-0-521-76848-1.
- [41] Division of Engineering Californian Institute of Technology and Applied Science. *From atoms to the macroscale: the quasicontinuum method*.
http://www.kochmann.caltech.edu/research_QC.html.
- [42] G. Jóhannesson G. Henkelman and H. Jónsson. *Methods for finding saddle points and minimum energy paths*. Kluwer Academic, Theoretical Methods in Condensed Phase Chemistry 2000. ISBN: 978-0-306-46949-7.
http://theory.cm.utexas.edu/henkelman/pubs/henkelman00_269.pdf.
- [43] Peter Brommer. *On-the-fly, off-lattice KMC simulations on experimental time scales with k-ART*, International Workshop on "Beyond Molecular Dynamics: Long Time Atomic-Scale Simulations", Dresden 2012.
http://www.mpipks-dresden.mpg.de/~bemod12/Slides/Brommer_Bemod12.pdf.
- [44] Cornelius Lanczos. *Applied Analysis*. Prentice Hall, 1956. ISBN: 78-0-486-65656-4.

- [45] Péter Pulay. The activation-relaxation technique: Art nouveau and kinetic art. *Chemical Physics Letters*, 73(2):393–398, 1980. doi:[10.1016/0009-2614\(80\)80396-4](https://doi.org/10.1016/0009-2614(80)80396-4).
- [46] Eduardo Machado-Charry, Laurent Karim Béland, Damien Caliste, Luigi Genovese, Thierry Deutsch, Normand Mousseau, and Pascal Pochet. Optimized energy landscape exploration using the ab initio based activation-relaxation technique. *The Journal of Chemical Physics*, 135(3):034102, 2011. doi:[10.1063/1.3609924](https://doi.org/10.1063/1.3609924).
- [47] Normand Mousseau, Laurent Karim Béland, Peter Brommer, Jean-François Joly, Fedwa El-Mellouhi, Eduardo Machado-Charry, Mihai-Cosmin Marinica, , and Pascal Pochet. The activation-relaxation technique: Art nouveau and kinetic art. *Journal of Atomic, Molecular and Optical Physics*, 2012:1–14, 2012. doi:[10.1155/2012/925278](https://doi.org/10.1155/2012/925278).
- [48] Laurent Karim Béland, Peter Brommer, Fedwa El-Mellouhi, Jean-Fran çois Joly, and Normand Mousseau. Kinetic activation-relaxation technique. *Physical Review E*, 84(4):046704, Oct 2011. doi:[10.1103/PhysRevE.84.046704](https://doi.org/10.1103/PhysRevE.84.046704).
- [49] Fedwa El-Mellouhi, Normand Mousseau, and Laurent J. Lewis. Kinetic activation-relaxation technique: An off-lattice self-learning kinetic Monte Carlo algorithm. *Physical Review B*, 78(15):153202, 2008. doi:[10.1103/PhysRevB.78.153202](https://doi.org/10.1103/PhysRevB.78.153202).
- [50] A. F. Voter. Hyperdynamics: accelerated molecular dynamics of infrequent events. *Physical Review Letters*, 78(20):3908–3911, 1997. doi:[10.1103/PhysRevLett.78.3908](https://doi.org/10.1103/PhysRevLett.78.3908).
- [51] A. Laio and M. Parrinello. Escaping free-energy minima. *Proceedings of the National Academy of Sciences of the United States of America*, 99(20):12562–12566, 2002. doi:[10.1073/pnas.202427399](https://doi.org/10.1073/pnas.202427399).
- [52] D.J. Wales and J.P.K. Doye. Global optimization by basin- hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *Journal of Physical Chemistry*, 101(28):5111–5116, 1997. doi:[10.1021/jp970984n](https://doi.org/10.1021/jp970984n).
- [53] S. Goedecker. Minima hopping: an efficient search method for the global minimum of the potential energy surface of complex molecular systems. *Journal of Chemical Physics*, 120(21):9911–9917, 2004. doi:[10.1063/1.1724816](https://doi.org/10.1063/1.1724816).

- [54] G. Henkelman Blas P. Uberuaga and Hannes Jónsson. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *The Journal of Chemical Physics*, 113(22):9901–9904, 2000. doi:[10.1063/1.1329672](https://doi.org/10.1063/1.1329672).
- [55] Alexis T. Bell Anthony Goodrow and Martin Head-Gordon. Transition state-finding strategies for use with the growing string method. *The Journal of Chemical Physics*, 130(24), 2009. doi:[10.1063/1.3156312](https://doi.org/10.1063/1.3156312).
- [56] Charles J. Cerjan and William H. Miller. On finding transition states. *The Journal of Chemical Physics*, 75(6):2800–2806, 1981. doi:[10.1063/1.442352](https://doi.org/10.1063/1.442352).
- [57] Jeff Nichols, Hugh Taylor, Peter Schmidt, and Jack Simons. Walking on potential energy surfaces. *The Journal of Physical Chemistry*, 92(1):340–346, 1990. doi:[10.1063/1.458435](https://doi.org/10.1063/1.458435).
- [58] G.T. Barkema and Normand Mousseau. Event-based relaxation of continuous disordered systems. *Physical Review Letters*, 77:4358–4361, Nov 1996. doi:[10.1103/PhysRevLett.77.4358](https://doi.org/10.1103/PhysRevLett.77.4358).
- [59] Rachid Malek and Normand Mousseau. Dynamics of lennard-jones clusters: A characterization of the activation-relaxation technique. *Physical Review E*, 62:7723–7728, Dec 2000. doi:[10.1103/PhysRevE.62.7723](https://doi.org/10.1103/PhysRevE.62.7723).
- [60] Normand Mousseau and G.T. Barkema. Traveling through potential energy landscapes of disordered materials: The activation-relaxation technique. *Physical Review E*, 57:2419–2424, Feb 1998. doi:[10.1103/PhysRevE.57.2419](https://doi.org/10.1103/PhysRevE.57.2419).
- [61] J.P.K Doye and D.J. Wales. Surveying a potential energy surface by eigenvector-following. *Zeitschrift für Physik D Atoms, Molecules and Clusters*, 40(1):194–197, 2014. doi:[10.1007/s004600050192](https://doi.org/10.1007/s004600050192).
- [62] Lindsey J. Munro and David J. Wales. Defect migration in crystalline silicon. *Physical Review B*, 59:3969–3980, Feb 1999. doi:[10.1103/PhysRevB.59.3969](https://doi.org/10.1103/PhysRevB.59.3969).
- [63] G. Henkelman and Hannes Jónsson. A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives. *The Journal of Chemical Physics*, 111(15):7010–7022, 1999. doi:[10.1063/1.480097](https://doi.org/10.1063/1.480097).
- [64] D.J. Wales. Discrete path sampling. *Molecular Physics*, 100(20):3285–3305, 2002. doi:[10.1080/00268970210162691](https://doi.org/10.1080/00268970210162691).

- [65] Abdelkader Kara, Oleg Trushin, Handan Yildirim, and Talat S. Rahman. Off-lattice self-learning kinetic monte carlo: application to 2d cluster diffusion on the fcc(111) surface. *Journal of Physics: Condensed Matter*, 21(8):084213, 2009. doi:10.1088/0953-8984/21/8/084213.
- [66] Yury N. Osetsky Haixuan Xu and Roger E. Stoller. Simulating complex atomistic processes: On-the-fly kinetic monte carlo scheme with selective active volumes. *Physical Review B*, 84(13):132103, 2011. doi:10.1103/PhysRevB.84.132103.
- [67] Dhrubajit Konwar, Vijesh J. Bhute, and Abhijit Chatterjee. An off-lattice, self-learning kinetic monte carlo method using local environments. *The Journal of Chemical Physics*, 135(17), 2011. doi:10.1063/1.3657834.
- [68] Samuel T. Chill and G. Henkelman. Molecular dynamics saddle search adaptive kinetic monte carlo. *The Journal of Chemical Physics*, 140(21), 2014. doi:10.1063/1.4880721.
- [69] Laurent Karim Béland, Yuri N. Osetsky, Roger E. Stoller, and Haixuan Xu. Kinetic activation–relaxation technique and self-evolving atomistic kinetic monte carlo: Comparison of on-the-fly kinetic monte carlo algorithms. *Computational Materials Science*, 100, Part B:124 – 134, 2015. Special Issue on Advanced Simulation Methods. doi:10.1016/j.commatsci.2014.12.001.
- [70] Gawonou Kokou N'Tsouaglo, Laurent Karim Béland, Jean-François Joly, Peter Brommer, Normand Mousseau, and Pascal Pochet. Probing potential energy surface exploration strategies for complex systems. *Journal of Chemical Theory and Computation*, 11(4):1970–1977, 2015. doi:10.1021/ct501032v.
- [71] Erik Bitzek, Pekka Koskinen, Franz Gähler, Michael Moseler, and Peter Gumbsch. Structural relaxation made simple. *Physical Review Letters*, 97:170201, Oct 2006. doi:10.1103/PhysRevLett.97.170201.
- [72] Brendan D. McKay. *Practical graph isomorphism*, volume 30. Congressus Numerantium, Department of Computer Science, Vanderbilt University Tennessee, 1981. <http://users.cecs.anu.edu.au/~bdm/nauty/pgi.pdf>.
- [73] Mickaël Trochet, Laurent Karim Béland, Jean-François Joly, Peter Brommer, and Normand Mousseau. Diffusion of point defects in crystalline silicon using the kinetic activation-relaxation technique method. *Physical Review B*, 91(22):224106, 2015. doi:10.1103/PhysRevB.91.224106.

- [74] M.H. Kalos A.B. Bortz and J.L. Lebowitz. A new algorithm for monte carlo simulation of ising spin systems. *Journal of Computational Physics*, 17(1):10 – 18, 1975. doi:10.1016/0021-9991(75)90060-1.
- [75] Benjamin Bourassa *et atl*. Official manual of kinetic Activation-Relaxation Technique (k-ART).
- [76] Brian Puchala, Michael L. Falk, and Krishna Garikipati. An energy basin finding algorithm for kinetic monte carlo acceleration. *The Journal of Chemical Physics*, 132(13):134104, 2010. doi:10.1063/1.3369627.
- [77] Michael P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Oxford University Press, 1st Edition 1989. ISBN: 0-19-855375-7.
- [78] F.H. Stillinger and T.A. Weber. Computer simulation of local order in condensed phases of silicon. *Physical Review B*, 31:5262–5271, April 1985. doi:10.1103/PhysRevB.34.6987.
- [79] J. Tersoff. New empirical model for the structural properties of silicon. *Physical Review Letters*, 56(6):632–635, Erratum Phys. Rev. B 41, 3248 (1990), February 1986. doi:10.1103/PhysRevLett.56.632.
- [80] J. Tersoff. Empirical interatomic potential for silicon with improved elastic properties. *Physical Review B*, 38:9902–9905, Nov 1988. doi:10.1103/PhysRevB.38.9902.
- [81] Martin Z. Bazant, Efthimios Kaxiras, and J. F. Justo. Environment-dependent interatomic potential for bulk silicon. *Physical Review B*, 56:8542–8552, Oct 1997. doi:10.1103/PhysRevB.56.8542.
- [82] Stephen J. Cook and Paulette Clancy. Comparison of semi-empirical potential functions for silicon and germanium. *Physical Review B*, 47:7686–7699, Apr 1993. doi:10.1103/PhysRevB.47.7686.
- [83] Official page of LAMMPS.
<http://lammms.sandia.gov>.
- [84] Official page of RasMol.
<http://www.rasmol.org>.

- [85] Official page of Graphviz tools.
<http://www.graphviz.org>.
- [86] Iván Santos, María Aboy, Pedro López, Luis A. Marqués, and Lourdes Pelaz. Insights on the atomistic origin of X and W photoluminescence lines in c-Si from ab initio simulations. *Journal of Physics D*, 49(7):075109, Feb 2016. doi:
[10.1088/0022-3727/49/7/075109](https://doi.org/10.1088/0022-3727/49/7/075109).
- [87] J. Coutinho A. Carvalho, R. Jones and P. R. Briddon. Density-functional study of small interstitial clusters in si: Comparison with experiments. *Physical Review B*, 72(15):155208, Feb 2005. doi:[10.1103/PhysRevB.72.155208](https://doi.org/10.1103/PhysRevB.72.155208).
- [88] Luis A. Marqués, Lourdes Pelaz, Pedro Castrillo, and Juan Barbolla. Molecular dynamics study of the configurational and energetic properties of the silicon self-interstitial. *Physical Review B*, 71(8):1–12, 2005. doi:[10.1103/PhysRevB.71.085204](https://doi.org/10.1103/PhysRevB.71.085204).
- [89] Patrick Ganster, Laurent Karim Béland, and Normand Mousseau. First stages of silicon oxidation with the activation relaxation technique. *Physical Review B*, 86(7):075408, Aug 2012. doi:[10.1103/PhysRevB.86.075408](https://doi.org/10.1103/PhysRevB.86.075408).
- [90] Iván Santos and Sergio Retuerto. Study of the pressure effect on point and extended defects in semiconductors through atomistic simulations. Summer Residence, University of Valladolid. Oct 2015.
- [91] E. E. Haller H. Bracht and R. Clark-Phelps. Silicon self-diffusion in isotope heterostructures. *Physical Review Letters*, 81(2):393–396, Jul 1998. doi:[10.1103/PhysRevLett.81.393](https://doi.org/10.1103/PhysRevLett.81.393).

This dissertation was typeset
using the \LaTeX document
processing system originally
developed by L. Lamport,
based on \TeX typesetting
system created by D. Knuth.