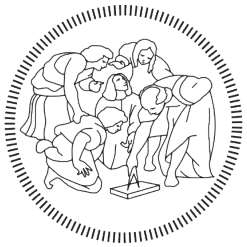

POLITECNICO
MILANO 1863

Scuola di Ingegneria Industriale e dell'Informazione
Laurea Magistrale in Engineering Physics

An atomistic multiscale approach for the simulation of electron-beam's irradiation induced effects on carbon nanotubes containing nickel nanoclusters

Master's Thesis of:
Luca Nicoli
Student ID: 905559

Supervisor:
Prof. Francesco Scotognella

Co-supervisor:
Prof. Timo Jacob

A.A. 2019 - 2020

Contents

List of Figures

Abstract

Although it is believed that the future provision of sustainable energy and materials may depend upon the catalysis of chemical reactions by nanosized clusters of transition metals, their atomistic behaviour is still largely unknown. The context of this thesis is the stop-frame filming of reacting molecules using a transmission electron microscope (TEM) both as stimulus and imaging tool i.e. a *ChemTEM* experiment. In particular, this work focuses in the development of a theoretical multiscale approach to simulate the electron-assisted metal-nanotube interaction, specifically for endohedrally confined nickel nanoparticles (55 atoms) in single walled carbon nanotubes. Calculations are carried out by means of reactive force field molecular dynamics methods. The goal is to cast light on the atomistic mechanisms at the basis of the nucleation of sp^2 hybridized carbons at the surface of the entrapped small metal cluster. The presence of rare single electron-atom collisions, during a *ChemTEM* experiment, allows modelling the electron irradiation of the specimen as a sequence of distinct impacts and subsequent relaxation events. The multiscale nature of our approach relies in simulating these two separate events, which have very different time scales, with two different, but joint, methods. The electron collisions are simulated by means of normal molecular dynamics tools, whereas the long time scale thermalization of the system after each collision, is simulated employing the *force biased Monte Carlo* acceleration algorithm. Based on this modelling, we developed an in-house code, specifically tailored for the simulation of this type of *ChemTEM* experiments, which allowed us to overcome the computational time issues related with such investigation. The results obtained by means of this code, show how the initial stages of the electron-assisted nucleation of a further nanotube from the surface of the endohedrally confined nickel particle, are similar to those reported for catalytic chemical vapour deposition (CVD) of hydrocarbon gas, dissociated at the nanocluster surface. However, the source of carbon in our case is the confining single walled nanotube. The strong interaction between the nanotube and the nickel cluster causes the latter to strongly bind with the vacancies defects in the nanotube wall, partially escaping from the endohedrally confining position. Depending on the fraction of nickels protruding outside the nanotube wall, two effects are observed: if it is large (e.g. more than 70 %), then the cluster-dissolved carbon monomers and dimers reattach to the outer nanotube; if it is small (e.g. less than 30 %), the dimers aggregate into trimers, and then into pentagon and hexagons. In the latter case, the so formed graphitic islands, segregated at the cluster's surface, aggregate into a carbon cap: this is the last stage before a new nanotube can start growing.

Abstract

Sebbene si ritenga che in futuro la fornitura di energia e materiali sostenibili dipenderà da particelle nanoscopiche di metalli di transizione, aventi la funzione di catalizzare reazioni chimiche, il loro comportamento atomistico è però ancora ampiamente sconosciuto. Questa lavoro di tesi si inserisce nel contesto della ripresa a "passo-uno" di molecole che reagiscono usando un microscopio elettronico a trasmissione (TEM) sia come strumento di stimolo che di imaging, ovvero un esperimento di *ChemTEM*. In particolare, l'elaborato si concentra sullo sviluppo di un approccio teorico multiscala per simulare l'interazione metallo-nanotubo indotta dalle collisioni elettroniche, specificamente per piccole (55 atomi) particelle di nichel confinate nella sezione di un nanotubo di carbonio a parete singola (SWNT). I calcoli sono stati effettuati mediante metodi di dinamica molecolare facenti uso di campi di forza reattivi. L'obiettivo è quello di gettare luce sui meccanismi atomistici alla base della nucleazione di piccoli reticoli di carbonio ibridizzato sp^2 sulla superficie della particella di metallo intrappolata. La presenza di singole collisioni elettrone-atomo separate nel tempo, durante un esperimento di *ChemTEM*, permette di modellizzare l'irradiazione elettronica del campione come una sequenza distinta di impatti e successivi eventi di rilassamento. La natura multiscala del nostro approccio risiede nel poter simulare questi due eventi separati, che possiedono scale temporali molto diverse, con tecniche distinte. Le collisioni elettroniche sono simulate tramite normali metodi di dinamica molecolare, mentre la lenta termalizzazione del sistema, dopo ciascuna collisione, tramite l'algoritmo di accelerazione noto come *force biased Monte Carlo*. Sulla base di questo modello per l'irradiazione elettronica, abbiamo sviluppato un codice appositamente studiato per la riproduzione di questo genere di esperimenti di *ChemTEM*, il quale ci ha permesso di superare i problemi di tempo di calcolo relativi alla loro simulazione. I risultati ottenuti con questo codice mostrano come gli stadi iniziali della nucleazione di un ulteriore nanotubo, indotta dalle collisioni elettroniche, dalla superficie di una particella di nichel confinata nella sezione di un nanotubo; sono simili a quelli riportati in letteratura riguardanti la deposizione chimica di vapore (CVD) di gas di idrocarburo, dissociato cataliticamente sulla superficie della nanoparticella. Tuttavia, nel nostro caso la fonte di carbonio è direttamente il nanotubo esterno. La forte interazione tra il nanotubo e la particella di nichel fa sì che quest'ultima si leghi fortemente ai difetti indotti dagli elettroni nella parete del nanotubo, scappando in parte dal confinamento endoedrico. A seconda della frazione di nichel che protrude dalla parete esterna del nanotubo, si possono osservare due effetti: se è il quantitativo di atomi è grande (ad esempio più del 70 %), i monomeri e i dimeri di carbonio disciolti nella particella di metallo vengono riassorbiti nel reticolo del nanotubo esterno; se è piccolo (ad esempio meno del 30 %), i dimeri si aggregano in trimeri, quindi in pentagoni ed esagoni. In quest'ultimo caso le isole grafitiche, così formatesi sulla superficie della particella, si aggregano in una calotta di carbonio: questo è l'ultimo stadio prima che un nuovo nanotubo possa iniziare a crescere.

1 Introduction

The challenge of microscopy is to image smaller and smaller objects with higher and higher resolution. Since the work of Ernst Abbe in 1873, the resolution $d = \frac{\lambda}{2nsin(\alpha)}$ is enhanced by lowering the wavelength λ, increasing the refraction index n and the opening angle α. This resulted in a scientific evolution from light optics, to x-rays and later on to electron microscopy.
The electron microscope and more specifically the transmission electron microscope (TEM) has become in recent years an indispensable tool to probe materials at the Ångström scale. In particular in this type of microscopy a finely focused (<10 nm diameter) electron beam with an acceleration voltage 50-150 kV is scanned across an electron-transparent specimen under vacuum and the intensities of the transmitted electrons are measured. In this way images can be formed by the variation of such intensity from point to point through the specimen. This instrument is undoubtedly the most useful tool for the investigation of nanostructures of all kinds and especially for those which are carbon derived: almost all types and subspecies of graphitic nanostructures have been indeed discovered and subsequently investigated using TEMs [1].
A recently developed experimental technique (*Thomas W. Chamberlain et al.* 2016), namely *ChemTEM* [17] aims to follow chemical transformations at the single-molecule level with the electron beam of an aberration corrected high resolution transmission electron microscope (AC-HRTEM) applied both as tunable source of energy and a sub-Ångström imaging probe. Such dual role of the electron beam enables to trigger and record transformation as soon as they happen. Moreover varying the accelerating voltage[1] different chemical processes can be qualitatively triggered [14], whilst tuning the electron dose rate might control the rate of reactions [15]. For all these reasons the *ChemTEM* methodology is well suited for recording continuous "movies" of chemical reactions by acquiring a time series of AC-HRTEM images for a single molecule, where the reaction is taking place simultaneously with the imaging process.
The tubular nature of carbon nanotubes has always triggered wide research interest: in recent years the development of techniques to confine foreign species within the CNTs channels [11] have paved the way for a whole new plethora of applications. In particular the confinement imposed by the nanotube (NT) can also create a "protected" environment to study all sort of reactions of the entrapped

[1] Typical values are between 20-300 kV.

species. Indeed all reactant molecules in single walled nanotubes (SWNT) exist in similar initial states because they are strictly ordered and stacked in the SWNT: this means that they all have approximately the same kinetic energy and orientation. Moreover, the entrapment restricts the movement of the molecules preventing certain conformations (especially those which are sterically demanding) and this often causes a deceleration in the chemical transformation that the confined species can undergo.
For all these reasons entrapment of the sample species within a single walled carbon nanotube's section, proved to give ideal conditions for the stop-frame filming of inter-molecular reactions during a *ChemTEM* experiment [17] .
However, carbon nanostructures are quite sensitive to electron irradiation in the typical TEM's operating voltage regimes [80] and therefore the confining carbon structure might inevitably undergo some structural changes during the irradiation, possibly interacting also with the confined species.
However this shouldn't be considered solely as a problem as the work of *Alexander S. Sinitsa et al.* [19] pointed out. Indeed in this case the intentional electron irradiation of nanostructures e.g. a metal nanocluster attached to sufficient carbon material (about 100 carbon atoms) allowed to produce new types of endohedral metallofullerenes[2] (EMF) [18] with huge advantages with respect to classical synthetic methods[3].
This therefore opened a completely new scenario: metal nanoparticles endohedrally confined in a SWNT section are excited by means of an electron beam and interacting with the carbon atoms of the encapsulating nanotube walls, they trigger reactions and morphological modifications.
Nanosized clusters of metals of the Periodic's table groups $9-11$ are very important in the context of catalysis of chemical reactions and they are believed to be crucial to the provision of sustainable materials for a number of different application [12]. Among these metals, nickel is considered of primary importance in the context of catalysis of carbon nanostructure's growth, such as carbon nanotubes [94]. However, the atomistic behaviour of nano-catalysts still remains largely unknown due to uncertainties associated with the highly labile metal nanoclusters changing their structure during the reaction [12].

[2]This type of nanomaterials which consist in a small (in the order of tenths of atoms) metal cluster endohedrally confined in a spherical carbon shell, hold promises for biological, medical and molecular electronics applications [18].

[3]These are mainly arc discharge reactor or laser evaporation [18] and both require high temperature, which leads to a loss of metal atoms by vaporisation or sintering [19].

The lack of such knowledge hinders the possibility of producing, with high efficiency, diameter controlled nanotubes [94]. Moreover, AC-HRTEM technology is currently still unable to reveal the atomistic mechanism of processes involved in a *ChemTEM* experiment. Therefore obtaining a detailed understanding of electron-assisted nickel-catalyzed carbon restructuring, can provide further insight into the atomistic mechanisms of nucleation of sp^2 hybridized carbons at the metal surface, which is considered to be the starting point for carbon nanotube's growth and is known to be crucial in determining the structure and property of such NTs [24],[25].

For all these reasons, this thesis work focuses in the development of a multiscale theoretical approach to simulate the electron assisted metal-nanotube interaction: targeting especially the simulation of the initial processes of growth of a carbon nanotube onto an endohedrally confined nickel nano-cluster (see figure 1). In order to do so the work is divided as follows:

1. In Chapter 2 a theoretical background of the simulation methods is given;
2. Chapter 3 focuses on a preliminary study to assess the validity of the force field used for the molecular dynamics simulations;
3. Chapters 4, 5, 6 focus on the physical modelling of a *ChemTEM* experiment;
4. Chapter 7 is devoted to the development of the in-house code for the simulation of a *ChemTEM* experiment, based on the modelling and on the parameters obtained in the previous chapters;
5. Chapter 8 presents the results obtained with such software when simulating the irradiation of a nickel cluster endohedrally confined in a carbon nanotube.

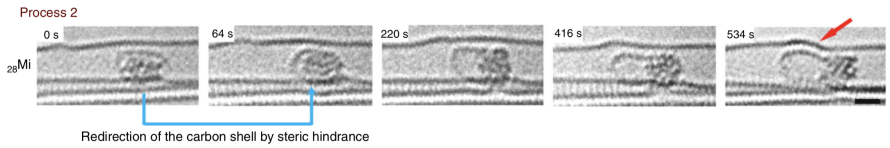

Figure 1: Time-series AC-HRTEM images illustrating a Ni nanocluster abstracts carbon atoms from a point of contact with the host NT and promotes the formation of a new carbon structure.
Image taken from [12].

2 Theoretical background and methods

2.1 Molecular Simulations

The simulation of a physical system of atoms and molecules is a very complex task. Considering a molecule, its Hamiltonian reads:

$$\hat{H}_{mol} = \hat{T}_e + \hat{T}_N + \hat{V}_{N-N} + \hat{V}_{e-e} + \hat{V}_{e-N} \tag{2.1}$$

Where:

1. $\hat{T}_e = \frac{-\hbar^2}{2m_e} \sum_{i=1}^{Ze} \nabla_i^2$ is the electrons kinetic energy operator;
2. $\hat{T}_N = -\hbar^2 \sum_{\alpha=1}^{Z} \frac{\nabla_\alpha^2}{M_\alpha}$ is the nuclei kinetic energy operator;
3. $\hat{V}_{e-e} = \sum_{i<j} k^2 \frac{e^2}{|\vec{r}_i - \vec{r}_j|}$ is the operator for the Coulomb interaction between electrons;
4. $\hat{V}_{N-N} = \sum_{\alpha<\beta} k^2 \frac{Z_\alpha Z_\beta e^2}{\left|\vec{R}_\alpha - \vec{R}_\beta\right|}$ is the operator for the Coulomb interaction between nuclei;
5. $\hat{V}_{e-N} = \sum_{\alpha,i} k^2 \frac{Z_\alpha e^2}{\left|\vec{R}_\alpha - \vec{r}_i\right|}$ is the operator of attractive interaction between nuclei and electrons.

Due to the extreme complexity of such Hamiltonian even for small systems, no analytical solution of the Schrödinger equation can be found. Thanks to the large difference between electronic and nuclei's motion energy levels, we can nevertheless make use of the Born-Oppenheimer approximation for which the molecular wave function can be split into a nuclear and an electronic part according to:

$$\Psi_{mol}(\vec{r}, \vec{R}) = \phi_N(\vec{R}) \psi_e(\vec{r}; \vec{R}) \tag{2.2}$$

Nuclear coordinates are treated parametrically in the electronic wave function, i.e. the electrons "see" a fixed nuclei configuration. Supposing that operators acting on nuclear coordinates don't act on the electronic wave function, then equation 2.1 can be reduced to a set of two coupled equations [26]:

$$\left(\hat{T}_e + \hat{V}_{e-e} + \hat{V}_{e-N}\right) \psi_e^g(\vec{r}; \vec{R}) = E_{el}^g(\vec{R}) \psi_e^g(\vec{r}; \vec{R}) \tag{2.3}$$

$$\left(\hat{T}_{N-N} + E_{el}^g + \hat{V}_{N-N}\right) \phi_N(\vec{R}) = E_{mol} \phi_N(\vec{R}) \tag{2.4}$$

Where E_{el}^g is the ground state electronic eigenvalue for a specific position of the nuclei. During the motion of the nuclei, the energy of the electrons might indeed vary without any change of state.
Even applying the Born-Oppenheimer approximation, the equations

are still extremely complex to solve: further approximations and numerical solutions are the only viable options to handle them in almost all cases.

2.2 Force field methods

The equations of quantum mechanics hereby presented are complex and computationally expensive to solve even in approximate forms: this leads to a dramatic reduction in the system size that can be treated in a finite computational time with this level of accuracy.
For this reason many other methods relying on more approximate models have been developed throughout the years in order to solve the equations of dynamics for physical and chemical systems. Among them, force field methods make use of classical-like forces to approximate the real interaction between atoms and molecules, leading to more simple classical equations rather than complex coupled equations of the quantum mechanics.
These methods rely in considering that in the nuclear Hamiltonian in equation 2.4 the electronic contribution enters just as E_{el}^{g} and it's still a function of the nuclear $\{\vec{R}\}$ set of coordinates. Thus the Hamiltonian 2.4 can be rewritten as:

$$\left(\hat{T}_{N-N} + \hat{V}_{tot}\right)\phi_N(\vec{R}) = E_{mol}\phi_N(\vec{R}) \tag{2.5}$$

Where $\hat{V}_{tot}$ embeds all the interaction terms between nuclei with a certain electronic ground distribution. Recasting equation 2.5 in a classical form:

$$H_N = T_N(\vec{R}) + V_{tot}(\vec{R}) \tag{2.6}$$

The core approximation of force field methods relies in consider the total *internuclear* potential energy V_{tot} (also called potential energy surface PES) as a sum of different energy terms, each one representing a different type of *interatomic* interaction:

$$V_{tot} = V_{bond} + V_{angle} + V_{torsion} + V_{Coulomb} + V_{vdW} + ... \tag{2.7}$$

Each potential energy term in this equation will have a certain functional shape, and will depend only upon the nuclear coordinates.
The electronic Hamiltonian 2.3 is therefore completely disregarded, whilst the contribution to the interatomic potentials of the electronic distribution is taken into account in the empirical functional shape of the different potential energy functions in equation 2.7.
To properly describe the behaviour of atoms of different species, these functions are dependent on a set of parameters which are in

general fit to experimental or computational data.
A force field consists of the equations chosen to model the potential energy and their associated parameters for specified interacting species.
It is important to bear in mind that molecular mechanics force fields are empirical: the separation of the potential energy in terms with a simple physical interpretation is not strictly correct, as there is no unique way to "translate" quantum mechanical effects into classical mechanics equations. As a consequence, there is no unique solution for the optimal set of functions and parameters. [27]
Thus different force fields might not just have different values for the same functions, but might have even different functional shapes for the same type of interaction.
Energy terms in most force fields can be divided into two classes: those associated with covalently bonded atoms e.g. V_{angle} and those associated with non-bonded atoms e.g. V_{vdW}.

A fundamental distinction between force fields has now to be done depending on how the covalent bonds are treated, i.e. fixed or varying. Classical force fields treat the valency of atoms as a fixed parameter, i.e. bonds cannot break and the description of the system is a purely non-reacting ensemble of molecules in which the nuclei don't experience anharmonicities in their potential. In practice, bond orders are a set of predetermined parameters (or they are computed at the beginning of the simulation) and they are not allowed to vary during the whole duration of the simulation. On the other hand reactive force fields treat the covalent bond as a function of interatomic distances, allowing the bond order, by means of a physically reasonable function, to vary smoothly between zero and its maximum value. In this way, as their name already suggests, reactive force fields are able to drive the system through reactions i.e. bond breaking and bond formation. Furthermore, other bond-dependent quantities scale with this continuous bond order as it varies between zero and its maximum value. Because chemical reactions may result in an atom converting from one type of configuration to another (e.g. one orbital hybridization or oxidation state to another), a universal description of each atom within the force field must adequately model all of these possible forms. [28]

2.3 The reactive force field *ReaxFF*

ReaxFF is a reactive force field developed by Adri C.T van Duin, Siddharth Dasgupta, Francois Lorant and William A. Goddard III [29][30]. Similar to empirical non-reactive force fields, the system potential energy is divided into various partial energy contributions:

1. E_{bond} the bond energy;
2. V_{val} the valence angle terms;
3. V_{tors} the torsion angle terms;
4. $V_{Coulomb}$ the Coulomb interactions;
5. V_{vdW} the van der Waals interactions;
6. E_{conj} the contribution of conjugation effects to the molecular energy;
7. E_{over}, E_{under} an energy penalty for atoms under or over coordinated.

A fundamental assumption of *ReaxFF* is that the bond order BO_{ij} between a pair of atoms can be obtained directly from the interatomic distance r_{ij} as given in the following equation 2.8 and plotted in figure 2:

$$\begin{aligned} BO'_{ij} &= BO^{\sigma}_{ij} + BO^{\pi}_{ij} + BO^{\pi\pi}_{ij} \\ &= exp\left[p_{bo,1}\cdot\left(\frac{r_{ij}}{r_0}\right)^{p_{bo,2}}\right] + exp\left[p_{bo,3}\cdot\left(\frac{r_{ij}}{r^{\pi}_0}\right)^{p_{bo,4}}\right] + exp\left[p_{bo,5}\cdot\left(\frac{r_{ij}}{r^{\pi\pi}_0}\right)^{p_{bo,6}}\right] \end{aligned} \tag{2.8}$$

Equation 2.8 consists of three exponential terms:

1. The σ bond ($p_{bo,1}$ and $p_{bo,2}$) which is unity below $\sim$ 1.5 Å but negligible above $\sim$ 2.5 Å ;
2. The first π bond ($p_{bo,3}$ and $p_{bo,4}$) which is unity below $\sim$ 1.2 Å but negligible above $\sim$ 1.75 Å ;
3. The second $\pi\pi$ bond ($p_{bo,5}$ and $p_{bo,6}$) which is unity below $\sim$ 1.0 Å but negligible above $\sim$ 1.4 Å ;

Each partial energy contribution will depend upon the bond order in such a way that pair of atoms far apart are going to have a null potential energy. For the sake of clarity in the next equation 2.9 the σ bond energies potential term E_{bond} is reported (see also figure 3).

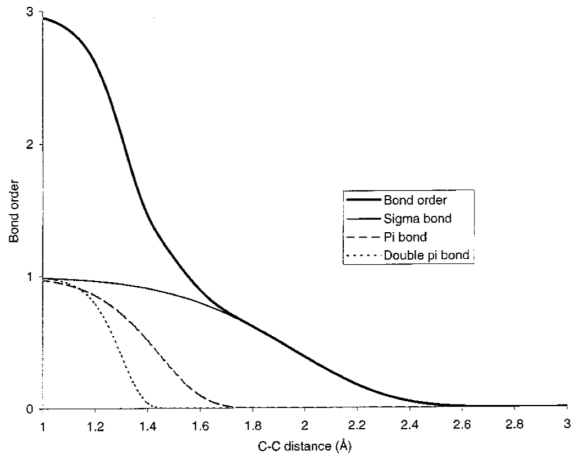

Figure 2: Interatomic distance dependency of the carbon-carbon bond order. Image taken from [29]

Analogous dependencies on BO are present for all potential terms of equation [29].

$$E^{\sigma}_{bond} = -D_e \cdot BO_{ij} \cdot exp\left[p_{be,1}\left(1 - BO_{ij}^{p_{be,2}}\right)\right] \tag{2.9}$$

In this way not just the bond order but also the potential energy surface are going to be continuous functions and therefore differentiable[4]. This is going to be the basis for the evaluation of forces in the molecular dynamics simulations described in section 2.5.1.

[4]For the sake of clarity, equation 2.8 defines the so called uncorrected bond order BO'_{ij}, whereas in the potential energy terms the corrected bond order BO_{ij} is employed (e.g. in equation 2.9). The latter one bears some corrections in order to reduce possible over-coordinations and to properly consider lone pairs. The issue is detailed addresses in the work of *Adri C. T. van Duin, Siddharth Dasgupta, Francois Lorant and William A. Goddard* [29].

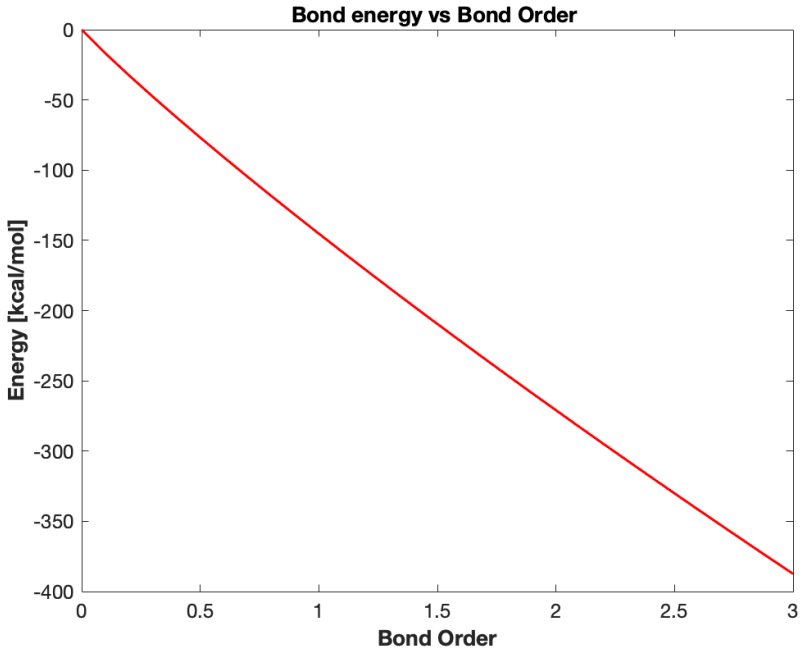

Figure 3: σ bond energy term eq. 2.9 for C-C interaction (*NiCH* force field [55])

2.4 Parameterization and validation of the force field

The many equations that make up a force field (FF) involve a number of parameters for which specific numeric values have to be obtained before being able to use this tool. One possibility is training the value of the parameters against experimental data. However, the amount of relevant experimental data available for fitting force field parameters is often scarce compared with the number of parameters that need to be established.
Thus, a second alternative is to use data from electronic structure calculations.
This second approach is a classic example of serial multiscale modelling: results obtained from electronic structure methods provide the input for a molecular model. [28] Parameters are typically fit by minimizing some measure of the error they produce when they are compared with the training set data.
A given parameterization can be validated using one or more of three different types of tests.
The first one uses the parameters to reproduce data that were not included in the training set, but which are of the same type (e.g. energies of formation from density functional theory (DFT) calcula-

tions) as those contained in the training set. If the parameters are able to reproduce these extra data, it suggests that their validity extends beyond the immediate boundaries of the training set.
A second way of using QM results to validate a FF parameterization is to extract novel structures or reaction pathways, which are observed during the course of simulations making use of certain values of the parameters and verify whether QM calculations indeed predict similar structures and energies for these snapshots.
The third type of test involves reproducing larger scale macroscopic properties, which are known experimentally, but cannot be directly reproduced using the electronic structure methods used to produce all or most of the training set (e.g. the melting point of a solid). This final validation procedure is useful in verifying if the parameters describe not only the atomistic phenomena predicted by electronic structure methods, but also higher order experimentally observable phenomena they are designed to model on the basis of the atomistic interactions [28]
The physical reasonableness of a force field formulation provides an important justification for applying the FF to systems which are not explicitly included in the training set. Nevertheless, it is important to know the limitations of an individual force field and caution is required when applying it to a system in which the atomic interactions deviate significantly from those contained in the training set.

2.5 Atomistic simulations

While the behaviour of some systems can be accurately characterized by exploring only the few points in phase space which correspond to selected local minima and the saddle-points connecting them, broader regions of phase space must be explored to reliably model other systems. Small molecules under ultra high vacuum (UHV) provide a good example of the former type because the vast majority of intermolecular interactions involve only a pair of molecules. Liquids provide a typical example of the latter type, because the relatively weak, but nevertheless significant interactions between molecules support a vast number of molecular configurations with similar energies. Average thermodynamic properties of such systems, rather than detailed snapshots, are typically what we want to extract from the simulations. A representative sample of configuration space is sufficient for this task.
Atomistic simulation techniques provide us with strategies for obtaining such representative samples [28].

2.5.1 Molecular Dynamics (MD)

The ergodic hypothesis implies that, regardless of the initial state we choose, if a thermodynamic variable is monitored as we follow the dynamical trajectory of the system, its value will converge to the appropriate statistical, expectation value as the time approaches infinity. In practice this means that, if followed over a sufficiently long time period, the system dynamics provide a thermodynamically representative sampling of a system. In molecular dynamics simulations this is typically accomplished by applying Newton's second law to determine the motion of each nucleus:

$$m_j \frac{\partial^2 \vec{R}_j}{\partial t^2} = -\nabla V_{\vec{R}=\vec{R}_j} \tag{2.10}$$

Where V represents the potential energy surface (PES), equation 2.7. As already stated before, in our case the PES is generated by the reactive force field *ReaxFF* partial energy contribution functions, which is continuous in the nuclear distances and therefore is differentiable. Of course analytical solutions are just available for very trivial systems. The dynamics of more complicated systems is therefore calculated by means of numerical algorithms (e.g. Verlet, leap-frog or velocity Verlet [28],[31]). These algorithms estimate the particle's position and velocity across finite time intervals. The shorter these time steps, the better the simulations approximate

the true trajectory but the higher computational time is required to explore the phase space. Meaningful trajectories typically require time steps which are at least an order of magnitude shorter than the fastest process in the system. Usually, these are considered to be vibrational frequencies of bonded atoms with an order of magnitude of $10^{-14}\ s^{-1}$, so that time steps of the order of 10^{-15} are necessary. Since the application of Newton's second law of motion eq. 2.10 conserves the energy, the natural ensemble for molecular dynamics (MD) simulations is the microcanonical NVE ensemble, in which the number of each type of particle (N), the system volume (V) and the energy of the system (E) are held constant. However, a *ChemTEM* experiment is carried out at constant temperature and due to the electronic collisions on the sample, definitely not at constant energy. Algorithms where the system temperature T is kept constant instead of E, therefore generating a canonical NVT ensemble, are required to simulate such experiments.

2.5.2 NVT ensemble and definition of Temperature

As we mentioned above, an NVE microcanonical ensemble is often not sufficient to properly describe the dynamics of a system of interest: indeed this does not correspond to the conditions under which most experiments are carried out. Therefore if someone is interested in simulating a system kept at constant temperature, an NVT simulation is required. The basic idea is to insert our system in a thermal bath kept at a reference temperature T_0, with which energy is exchanged in such a way to keep the average $\langle T \rangle$ of the system constant.
To do so we need first of all a proper definition of instantaneous T temperature of the system, which has to be compared with the reference one. In general in an MD simulation the current T of the system is related to the total kinetic energy through the particles' momenta:

$$\sum_{i=1}^{N} \frac{|\vec{p_i}|}{2m_i} = \frac{k_b T}{2}(3N - N_c) \tag{2.11}$$

with N_c the number of constraints on the system, so that $3N - N_c$ is the total number of degrees of freedom of the system.[32]
Defining the instantaneous T in this way it is ensured that the average $\langle T \rangle$ is identical to the macroscopical temperature T. [32] Now it has to be ensured that T converges to the required reference temperature T_0. The easiest way to do so is to implement a sort of

control on the velocities of the particle i.e. a velocity scaling.
A modification of the Newtonian MD scheme is therefore required and the methods used to do so i.e. to generate a thermodynamical ensemble at constant temperature are called thermostat algorithms. Several different thermostats have been developed and implemented in the MD simulations [32].
Throughout the current work we made extent use of the so called Nosé-Hoover chain thermostat [34] which is a slight modification of the so called Nosé-Hoover thermostat. The idea behind this algorithm is to extend the system with an external artificial variable $\tilde{\mathbf{s}}$ that plays the role of a time-scaling parameter. This variable is associated with a "mass" Q of units [energy x $(time)^2$], and a velocity $\dot{\tilde{\mathbf{s}}}$. The magnitude of Q determines the coupling between the reservoir and the real system and so influences the temperature fluctuations. It can be shown that the equations of motion sample a canonical ensemble in the real system [34]. The Nosé equations of motion are smooth, deterministic and time-reversible. However, because the time-evolution of the variable $\dot{\tilde{\mathbf{s}}}$ is described by a second-order equation, heat may flow in and out of the system in an oscillatory fashion, leading to nearly periodic temperature fluctuations [32].

2.5.3 Charge calculation: Electronegativity equalization method (EEM)

Electronegativity equalization method (EEM) [35],[36] is a fast empirical method for atomic charge calculation. *ReaxFF* has been developed around this method, in order to allow the calculation of geometry dependent charge distributions [29]. According to it, the effective (charge-dependent) electronegativity of an atom can be expressed as:

$$\chi_i = A_i + B_i q_i + \kappa \sum_{i \neq j} \frac{q_j}{r_{i,j}} \tag{2.12}$$

where χ_i denotes the electronegativity of atom i, A_i, B_i, κ are empirical parameters, q_i and q_j are the charges of atoms i and j respectively, and finally $r_{i,j}$ is the distance between atoms i and j [35]. According to Sanderson's Electronegativity Equalization Principle [37], the effective electronegativity of each atom in the molecule is equal to the molecular electronegativity $\bar{\chi}$:

$$\chi_1 = \chi_2 = ... = \chi_N = \bar{\chi} \tag{2.13}$$

The total charge Q of the molecule is equal to the sum of all the atomic charges:

$$\sum_{i=1}^{N} q_i = Q \tag{2.14}$$

Translating equation 2.12 into the system of linear equations describing one particular molecule gives us the final EEM system of $N+1$ equations with $N+1$ unknowns (q_1 , . . . , q_N , $\bar{\chi}$) used for calculating atomic charges:

$$\begin{pmatrix} B_1 & \frac{\kappa}{R_{1,2}} & \dots & \frac{\kappa}{R_{1,N}} & -1 \\ \frac{\kappa}{R_{2,1}} & B_2 & \dots & \frac{\kappa}{R_{2,N}} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\kappa}{R_{N,1}} & \frac{\kappa}{R_{N,2}} & \dots & B_N & -1 \\ 1 & 1 & \dots & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \\ \bar{\chi} \end{pmatrix} = \begin{pmatrix} -A_1 \\ -A_2 \\ \vdots \\ -A_N \\ Q \end{pmatrix} \tag{2.15}$$

The matrix of the equation system 2.15 is called EEM matrix.

2.6 Slow dynamics and acceleration algorithms

Reactive molecular dynamics is a powerful tool for exploring reaction pathways.
However, monitoring the evolution of a system between different equilibrium states is a computationally very demanding task. In general, the time window of the events that can be assessed through MD simulations is relatively limited and typically evolves in the range of picoseconds (ps) to nanoseconds (ns), depending on the dimension of the system. This is in general a very short time scale that often doesn't fit with experimentally observable events. This problem is tricky to overcome due to the fact that usually we both need to track the exact dynamics of the particle and follow it for a long time. For this reason, many approaches (e.g. [43],[41]) have been developed to speed up the dynamics of the system, keeping a physical motion in the configurational space.
The speed of the evolution of the system from two states is influenced by the height of the barriers in the PES involved in the transition from these states. In fact the larger the barrier, the less probable will be the transition according to Transition State Theory [40], which in a very simple formulation leads to the following formula 2.16 to model the thermally activated escape of any classical system from a metastable state A [40]:

$$k_{TST} = \frac{k_b T}{h} \frac{Q^{\#}}{Q_A} e^{-E_a/k_b T} \tag{2.16}$$

This equation models the rate k_{TST} of escape of A from a potential barrier, whose top represents the "transition state" (TS). Here E_a is the height of the barrier above the well, Q_A and $Q^{\#}$ are the partition functions in the well and in the TS respectively (the latter being a reduced partition function which sums over all coordinates orthogonal to the direction leading from reactant A to product B) and the temperature T provides the driving force to escape from the well[40].

Bearing in mind the experimental data [12] that we would like to reproduce, according to previous studies [38],[41], the defect healing at the SWCNT – catalyst interface needs to overcome a barrier of $E_a \approx 2.0$–2.5 eV, thus the timescale for such an event to occur is estimated to be $10^2 - 10^4$ μs. Nevertheless, as stated in section 2.5.1 a single MD time step is in the order of $\sim 10^{-15} s^{-1}$, the amount of steps to properly cover this time window is huge. We would therefore like to employ a method able to lead the system to overcome the large barriers involved with slow thermally activated reactions

of relaxation, without however modifying the PES to which the system is subjected.
Likewise a previous work of Erik C. Neyts [47] on free standing nickel particles, the method that we used is the so called force bias Monte Carlo or fbMC, which proved to be a reliable algorithm for the simulation of carbon nanotubes, in particular for sp^2 hybridized carbon atoms interacting with Ni [47] [42].

2.6.1 Force biased Monte Carlo fbMC

The essence of this method relies on an iterative modification of atomic positions taking into account the forces acting on them.
Indeed for every fbMC step each atom i is displaced over a distance $\xi_{i,j}\ \Delta_i$ in every Cartesian direction j, where $\xi_{i,j}$ is a stochastic variable distributed according to:

$$P(\xi_{i,j}) = \begin{cases} \frac{e^{\gamma_{i,j}(2\xi_{i,j}+1)} - e^{-\gamma_{i,j}}}{e^{\gamma_{i,j}} - e^{-\gamma_{i,j}}} & if\ \ \xi_{i,j} \in [-1, 0[\\ \frac{e^{\gamma_{i,j}} - e^{\gamma_{i,j}(2\xi_{i,j}-1)}}{e^{\gamma_{i,j}} - e^{-\gamma_{i,j}}} & if\ \ \xi_{i,j} \in]0, 1] \end{cases} \tag{2.17}$$

Where:

$$\gamma_{i,j} = \frac{F_{i,j}\Delta}{2k_bT} \tag{2.18}$$

Being $F_{i,j}$ the force acting on particle i in direction j .
In practice, every step a random coordinate pair (ξ_0, P_0) is chosen with $\xi_0 \in$ [-1,1] and $P_0 \in$ [0,1] for all the particles in all directions. If $P_{i,j}(\xi_0) > P_0$ the next interval of particle i will be around $x^*_{i,j} + \xi_0\Delta$. If $P_{i,j}(\xi_0) < P_0$ a new random pair (ξ_0, P_0) is generated for that degree of freedom and the acceptance is checked again.
The definition of γ given in equation 2.18 shows how the curve of conditioned probability $P_{i,j}(\xi_{i,j})$ (see figure 4) is influenced by the forces acting on the particle, highlighting how the system will statistically tend to states lower in energy. Thus this approach is able to describe the evolution of the system dynamics based on a simple probabilistic approach disregarding the velocity distribution of the particles. This is in complete contrast with the classical MD dynamics formalism, hence this approach might seem highly unphysical. However, thanks to the shape of the probability density function, the probability of finding a particle in a well of the PES rather than in an hypersurface dividing two equilibrium states is much higher. Moreover, every step in a MC algorithm is not directly correlated to a physical event [44]. Therefore the dynamics described by the fbMC method is such of a system in an equilibrium state that once

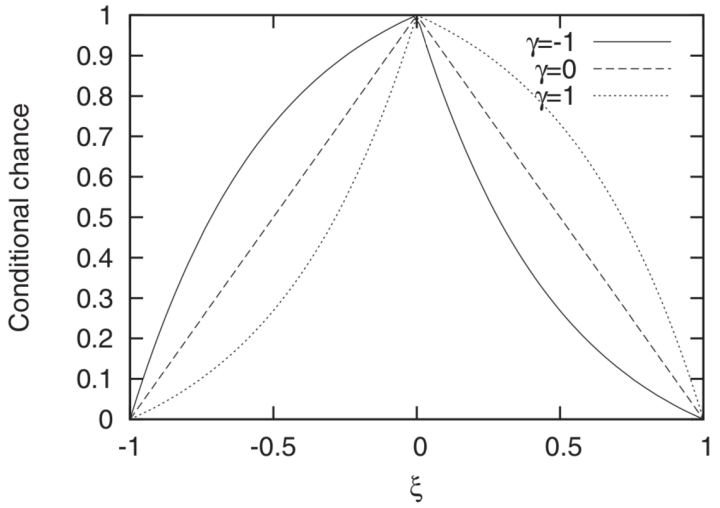

Figure 4: The function $P_{i,j}$ for different γ values (equation 2.17, figure taken from [44].

in a while moves to another stable state.
It is worth to emphasize again that the particles in this algorithm have no velocity: the temperature in this case has the effect of modifying the bias of the conditional probability Fig. 4 (the lower the temperature the higher the effect of the forces on the bias).

The probability distribution function $P_{i,j}$ is a key component of the fbMC algorithm (in appendix B it's reported its complete derivation according to the work of *Maarten J. Mees et al.* [44]). It is therefore mandatory to understand if this simulates the correct probability density function of a canonical ensemble. This is true if it meets the requirement of detailed balance[44], which given two states Λ and Λ' it reads:

$$P(\Lambda'|\Lambda)exp(-\beta U) = P(\Lambda|\Lambda')exp(-\beta U') \tag{2.19}$$

where U and U ' are the potential energies of the state Λ and Λ', respectively and $\beta = (k_b T)^{-1}$. This equation states that for a system in equilibrium the rate of transitions from a state Λ to a state Λ' has to be equilibrated by the reverse process. If this principle is not satisfied we would have a probability density flow and the system would not be in equilibrium.
Equation 2.19 in the fbMC framework is valid in the limit of infinitesimally small displacement Δ [44], i.e. for any practical simulation the detailed balance will be violated to some extent. Nevertheless, it widely recognized [44], [45], [46], [47], that displacement lengths

between 0.1 and 0.15 Å lead to physically meaningful results and can be considered as conservative choices [45]

Every step in a fbMC algorithm is just a probability-weighted displacement for all the particles of the system i.e. a change of the state in the configurational space. Nevertheless, there is no description of the time required from the system to reach that final state. Thus fbMC completely loses an absolute time-scale.
Some attempts have been made [44], deriving the timescale for this algorithm from a statistically relevant average time step related to a mean displacement $\langle \delta_{i,j} \rangle$ and a mean velocity as:

$$\langle \Delta t \rangle = \frac{\langle \delta_{i,j} \rangle}{\langle v_{i,j} \rangle} \tag{2.20}$$

This can give a first idea of the time spanned by the fbMC algorithm.
A more accurate description of the time step, with a direct comparison of the same processes with MD simulations [45] can also be done.
The time length of a fbMC step depends on the system temperature (the larger the temperature the smaller is the boost of the fbMC compared to a classical MD simulation) and on the displacement Δ. However, the magnitude of the displacement influences the PES since there is a higher deformation of the surrounding crystal. This leads to a lowering of the activation barriers [46] and therefore a loss of precision in the description of the dynamics of the system. For all these reasons it has been concluded [45] that it is inherently impossible to derive a universal timescale to describe the system evolution during a fbMC simulation.

In conclusion, the fbMC approach is a very simple method able to bring quickly (compared to normal MD) the system to an equilibrium state. Under the appropriate conditions, the PES is not too much modified and the detailed balance principle (stated in equation 2.19) is satisfied, leading the method to produce physically meaningful results. Furthermore, this method doesn't require any specific parameter of the system in analysis but just information about the temperature and the displacement to which the atoms are subjected. However, the defects of this method rely on the fundamental lack of an absolute timescale and on the risk of losing kinetic fidelity, of the processes to reach an equilibrium state, under large displacements.

2.7 Geometry optimization

After this long discussion regarding all the dynamical methods i.e. methods to track in time the evolution of a system it is worth talking rapidly about geometry optimization algorithms. These methods are used to find the set of coordinates in the configurational space which define a local minimum of the PES. In fact during an MD simulation the system moves in the configurational space and so on the PES. When the MD simulation stops, the system is in a state which is not necessarily the local minimum of energy: in fact it will lay on a state on the PES reachable with the thermal energy injected in the system through the thermostat. Therefore to make proper comparisons between energies of different geometries it is often required a method to bring the system down to the closest[5] local minimum of the potential energy surface in which the system is moving.
Thanks to the form of the interaction terms, as analyzed in section 2.3, the potential energy surface described by a reactive force field is continuous and differentiable.
For this reason throughout the whole thesis work a Quasi-Newton method namely L-BFGS (Limited-memory BFGS) [48] has been used for the calculation of the structures , since it requires just the evaluation of V^{PES} and ∇V^{PES}, of which the existence is guaranteed by the *ReaxFF* potential energy surface.
Quasi-Newton methods are in general faster and computationally cheaper compared to classical Newton methods since there is no need for the storage of the full Jacobian matrix [49], allowing us to calculate fast the energies of the closest minimum of the large structures of interest.

[5]Closest in terms of nuclei coordinates with respect to the initial state from which we apply the geometry optimization.

2.8 Softwares used

All molecular dynamics simulations were performed using the software "SCM: Software for Chemistry and Materials" version (2018.106) [50]. This huge suite embeds specific codes for all the methods exposed in this section.
The images of the results were all represented with the "Ovito" software version Version 2.9.0 [51].
Codes to handle SCM data (sections 7.1 and 7) where written on *Python* [52].
To analyze the results and to plot graphs Matlab (R2017b) was used [53].

2.9 General computational details

All molecular dynamics simulations were performed with a step size of 0.25 fs and the Nosé-Hoover chain thermostat with a damping constant of 100 fs. Specific values for: the size of the simulation box, the temperature and the number of MD steps; are reported in each section where this method is employed.
All geometry optimization calculations were perfomred using a L-BFGS method, with 0.1 eV/Å convergence criterion. Specific value of the simulation box size are reported in each section where this method has been used

I hope that this chapter briefly but precisely exposed the theoretical background and the motivations for the use of the computational methods described.
In the next chapter the force field used, and the analysis of its validity for the system of interest are going to be presented.

3 The Force Field and its evaluation

The force field required to study a specific system needs to take into account all the interactions that could occur. In our case, we decided to make use of *NiCH* [55] force field developed by *Mueller, van Duin and Goddard III* in 2010[6].
The complete set of parameters of the used force field is reported in appendix D.
Due to the intrinsic empirical nature of molecular dynamics force fields based methods, an accurate analysis of its validity for the system that we want to study is mandatory before starting. This means that each interaction of interest has to be well described by the value of parameters of the potential functions.

3.1 Evaluation of Nickel-Nickel interactions

We began considering the interaction between nickel atoms. In order to be systematic we investigated the following properties:

1. The bulk face centered cubic *fcc* cohesive energy;
2. The surface energies;
3. The stable geometries of clusters at different sizes;
4. The binding energy per atom as function of the size of the cluster;

3.1.1 Ni bulk cohesive energy

To find the bulk cohesive energy of nickel it was constructed a bulk *fcc* unitary cell, with different lattice constants; for all structures we performed a single-point energy calculation[7]. The cohesive energy will simply be the total energy divided by the number of atoms in the unit cell.
Results are shown in table 1 .

In this way, we were able to find that for a nickel crystal in *fcc* configuration this force field yield to a minimum state with lattice constant of 3.61 Å and energy per atom of -4.50 eV. This result is in good agreement with experimental value of -102.4 kcal/mol i.e. -4.44 eV [56]. This value for the lattice constant will be used in the following section for the calculation of surface energies.

[6]The name of a force field indicates all the atoms' species for which the interactions have been parameterized and trained.

[7]This means that the energy for each structure was computed using a geometry optimization approach with 0 iterations i.e. obtaining as output the energy for that precise lattice constant.

Lattice constant [Å]	E_{coh} [kcal/mol]
3.52	-102.95
3.57	-103.57
3.60	-103.69
3.61	-103.71
3.62	-103.70

Table 1: Calculated bulk cohesive energies for different lattice constants.

3.1.2 Ni surface energies

The calculation of the surface energies was carried out for low index surfaces, i.e. for those whose Miller indices are between {100} and {332}. Thus we built slabs of 1x200 atoms of Nickel, starting from an *fcc* configuration with Ni-Ni distance found before of 3.61 Å. 20 Å of vacuum were left on each side of the slab. Subsequently, we drove a geometry optimization calculation using the L-BFGS method with a convergence criterion of 0.1 (kcal/mol)/Å. The surface energy γ identified by its Miller indices (*i,j,k*) was then calculated using the formula:

$$\gamma_{ijk} = \frac{E_{slab,ijk} - (N_{atoms} E_{bulk})}{2A_{ijk}} \tag{3.1}$$

Where $E_{slab,ijk}$ is the calculated total energy of the relaxed $\{ijk\}$ slab, E_{bulk} is the total energy per atom of the nickel fcc bulk unit cell, N_{atoms} is the total number of atoms in each *fcc* unit cell and A_{ijk} is the surface area of the slab model. The results obtained, compared with reference DFT calculations [57],[58], are shown in table 2.

It has to be noticed that the calculated surface energies approximate the theoretical trend [57] and the mismatch for the lowest energetic surfaces are acceptable: $\gamma_{111} = 0.08$ J/m^2, $\gamma_{332} = 0.1$ J/m^2, $\gamma_{322} = 0.1$ J/m^2, $\gamma_{221} = 0.11$ J/m^2. The highest mismatch found is for high energetic surfaces like $\gamma_{210} = 0.19$ J/m^2, $\gamma_{310} = 0.21$ J/m^2 which are in any case very unfavourable.

However as it's visible from tab. 2 the stability order is maintained. The remarkable exception is the surfaces of the {100} family, for which we have the highest mismatch between theory and calculations and for which our calculations bring them to have a very low energy.

Surface	Calculated	Tran et al. [57]	Hong et al. [58]
100	1.97	2.21	2.23
110	2.13	2.29	2.29
111	1.84	1.92	2.02
210	2.21	2.40	–
211	2.09	2.24	–
221	2.05	2.17	–
310	2.19	2.40	–
311	2.13	2.30	2.31
320	2.21	2.39	–
321	2.17	2.32	–
322	2.02	2.12	–
331	2.11	2.23	–
332	1.99	2.09	–

Table 2: Surface Energies results, expressed in [J/m^2]

3.1.3 Stable geometries and binding energies as function of the cluster size

The goal of the force field validation procedure is to understand whether the set of potential energy functions eq. 2.7 is able to reproduce (within an acceptable range) remarkable physical properties of the system. Having so far tested general nickel properties, we now need to validate the force field against more system specific properties.
To do so, we have studied the optimal geometries and the binding energies for nickel clusters (Ni_n) of different dimensions i.e. all the clusters made up by 2 to 23 atoms plus the cluster made by 55 atoms i.e. (Ni_{2-23} and Ni_{55}). The choice for these values was driven from the abundance of previous studies both theoretical and experimental for clusters of these dimensions e.g. [61], [60], [62], [63], [64], [65]. Thus we could have a reliable benchmark for the results of our force fields.
To evaluate the binding energies, we had first to:

1. Build the clusters;
2. Sample the configurational space.

Indeed several minimum energy structures might be present for each cluster, therefore once a geometry is built, we need to sample the configurational space of the cluster, in order to find the global minimum of the PES. This was done by means of an artificial thermal annihilation routine which is described in table 3. Moreover building the cluster shouldn't be done randomly, since this might lead to a bias in the reachable geometries. In order to build a reasonable

initial guess on the structure for each cluster, we employed results of the literature. Here a problem arises immediately: for many clusters the minimum energy geometry seemed to depend on the method used to study it. For example for Ni_7 the minimum energy structure is debated, references [63] and [60] consider it to be a capped octahedron whereas references [62] and [64] suggest a structure belonging to D_{5h} symmetry group. Lacking therefore of an universal reference for all geometries, we decided to build our structures based on the work of *Mark S. Stave and Andrew DePristo* [62] solely for practical reason: indeed in this work all the calculated geometries were reported (and not just the energies), and therefore we could use it as a guideline to build our structures. As mentioned before, in order to sample all the configurational space, and therefore evaluate also the other minimum energy structures, we performed a thermal annihilation procedure based on the setup shown in table 3.
Each calculation ran for 370000 MD steps of 0.25 fs with an NVT Nosé-Hoover chains thermostat. Each cluster was placed inside a cubic box with length of 100Å to avoid possible interaction within the periodic image representation. Multiple simulations were done for each cluster, enabling us to find several minima for each system. All the minimum energy structures found in this fashion were then fed through a geometry optimization (L-BFGS) with a convergence criterion of 0.1 (kcal/mol)/Å on the forces.

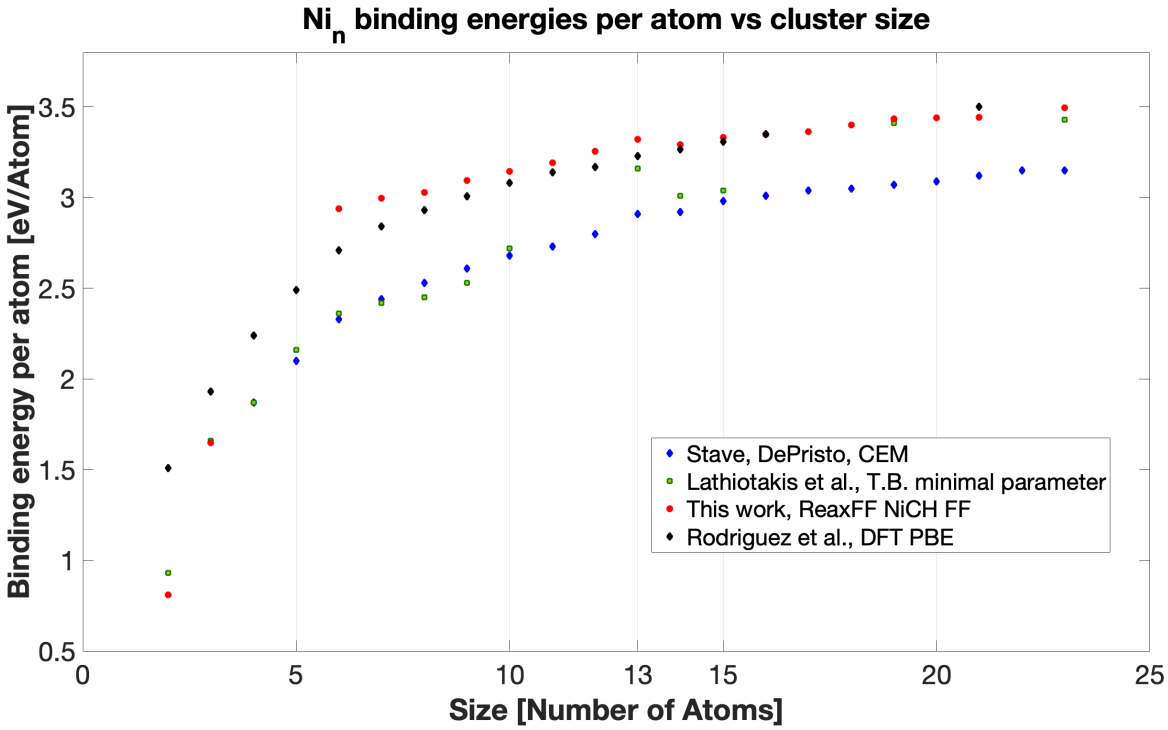

Figure 5: Binding energy in eV/atom as function of the cluster size, in the legend are also reported the methods used in the reference studies:
1) *Stave and DePristo* [62], CEM stands for corrected effective medium;
2) *Lathiotakis et al.* [64], tight binding minimal parameter;
3) *Rodriguez et al.* [63], DFT PBE.

The lowest final energy for each cluster was then used to calculate the binding energy (B.E.) of the cluster which reads:

$$E_b = E_{Ni_n} - E_{Ni_{n-1}} - E_{Ni} \tag{3.2}$$

Where E_{Ni_n}, $E_{Ni_{n-1}}$ and E_{Ni} are the energies of a cluster with n, $n-1$ and one atom respectively.
It has to be noted that within the force field representation the energy of an isolated atom is zero since all the energy terms are zero due to a null bond order.
Binding energies for the most stable geometries can be found in table 4 and are plotted as function of the cluster size in the figure 5.
The force field seems to overestimate the binding energy (Fig. 5) if compared with reference results, nevertheless the trend is remarkably good considering that small clusters where not part of the training set of this FF.

MD step	T [K]
0	1
10000	298
40000	1000
200000	800
250000	600
280000	300
320000	50
350000	1

Table 3: Simulated thermal annealing setup for the MD simulations, the damping constant for the thermostat was always set at 100 fs.

Number of atoms	Calc. symmetry	Ref. symmetry [62]	Calc. B.E.
2	–	–	0.809
3	–	–	1.648
6	D_{4v}	O_h	2.94
7	D_{5v}	D_{5h}	3.00
8	D_{2d}	D_{2d}	3.03
9	C_{2v}	D_{3h}	3.10
10	C_{3v}	C_{3v}	3.15
11	C_{2v}	C_{2v}	3.19
12	C_{5v}	C_{5v}	3.26
13	I_h	I_h	3.32
14	C_{3v}	C_{2v}	3.29
15	C_{2v}	D_{6d}	3.33
16	C_s	D_{3h}	3.35
17	–	T_d	3.36
18	C_{5v}	C_{2v}	3.40
19	C_{5h}	D_{5h}	3.44
20	C_{2v}	D_{3d}	3.44
21	–	C_s	3.45
22	–	D_{6h}	3.47
23	D_{3h}	D_{3h}	3.50
55	I_h	I_h	3.74

Table 4: Calculated binding energies (calc. B.E.) per atom, for each stablest nickel cluster. Energies are reported in eV/Atom.
For each cluster it's reported the calculated and the reference [62] symmetry group of the minimum energy structure. If the cell is filled with "–", no specific symmetry group was found for that cluster.

The problem related with uncertainties in the literature regarding the most stable geometry was not present for Ni_{13} and Ni_{55}. These two clusters are part of the so called *magic clusters* series[8] because they show a very high binding energy and a well defined icosahedral (see figure 6) structure. It is assumed [64] that Jahn-

[8]In analogy with the magic nuclei in nuclear physics.

Teller distortions of electronically degenerate configurations play a dominant role in transition metals' cluster geometry.
Although no explicit electron dependency is present in the *ReaxFF* description, the force field is able to reproduce for Ni_{13} and Ni_{55} the icosahedral minimum structure. Even more surprisingly, our reactive force field is able to recognize the extremely high stability of the Ni_{13} *magic cluster*: indeed from table 4 and from figure 5 it is visible how the binding energy for such cluster is higher with respect to the neighbouring Ni_{12} and Ni_{14}.
In table 5 are reported the calculated energies of Ni_{55} for different studies, showing how the binding energy obtained by means of the *ReaxFF* potentials with the *NiCH* force field is completely within the range of reported values for this cluster.

References	Method	Binding Energy [eV/Atom]
N.D. Yilmazer et al. [66]	DFT B3LYP	3.51
Lathiotakis et al. [64]	TBMD	4.27
P.L Rodriguez-Kessler et al. [63]	DFT PBE	3.93
Luo [67]	TBMD	3.55
This study	*ReaxFF*	3.74

Table 5: Values of binding energy of Ni_{55} reported by different reference works. TBMD stands for tight binding molecular dynamics.

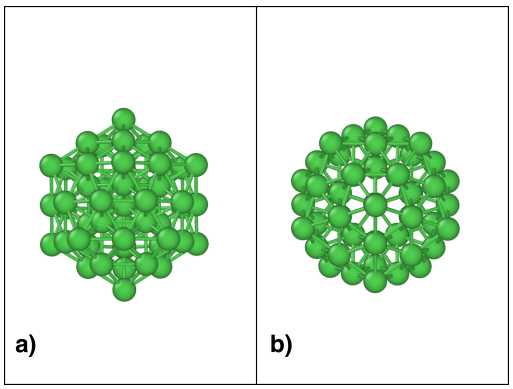

Figure 6: Top **a)** and side **b)** view of the icosahedral structure of Ni_{55}

Thus the overestimation of the binding energy is no more so evident increasing the cluster size.
The results presented in this section somehow allow us to think that even tough the energy for the surface (100) calculated in section 3.1 was not in agreement with reference results [57], [58], the force field it is nevertheless still able to lead to reasonable results for small sized nickel clusters.

Therefore, after all these considerations, we assumed that the behaviour of the force field *NiCH* regarding the nickel-nickel interactions was satisfactory.

3.2 Evaluation of Carbon-Carbon interaction interactions

Following the scheme presented in the previous sections, also the evaluation of the parameters for the carbon-carbon interaction was done by comparing both general and system specific properties to reference results.
In particular, we focused our attention on:

1. Evaluation of the graphite formation energy;
2. Evaluation of CNTs' stability as function of their chirality and diameter;

3.2.1 Evaluation of the graphite formation energy

As a preliminary general study we decided to evaluate the graphite formation energy. To do so, we followed the single point calculation routine mentioned before in section 3.1.1 for the evaluation of the Nickel bulk energy. Here we had two parameters to vary, namely the in plane vector $\vec{a}$ and the out of plane vector $\vec{c}$ between two identically stacked graphene sheets[9], of the primitive cell of the graphite. Here are presented the values obtained for such parameters and the related formation energy:

1. $a_1 = a_2 = 2.48$ Å the in plane vectors to be compared with ref. [68]: 2.44 Å;
2. $c = 6.21$ Å the spacing between two planes to be compared with ref. [68]: 6.71 Å;
3. $E_c = -7.93$ eV formation energy for a 500 atom graphite slab to be compared with ref. [69]: -7.35 eV.

The detailed results for $\vec{a}$ and $\vec{c}$ vectors can be find in table 6

c \ a	2.46	2.47	2.48	2.49	2.51
6.20	-715.024	-724.267	-731.347	-730.756	-730.614
6.21	-715.025	-724.267	-731.347	-730.756	-730.614
6.22	-715.024	-724.266	-731.347	-730.755	-730.613

Table 6: Unit cell graphite energy computed as a single point calculation. Energies are reported in kcal/mol. Lengths are in Å.
Highlighted in blue is the most stable combination.

[9] Graphite has an ABAB stacking, thus $\vec{c}$ is related to the distance between two A or B identically stacked graphene sheets.

The results are far from what can be found in literature [68],[69]. Nonetheless, the force-field wasn't specifically trained to represent graphite properties[10] but rather carbides systems [55] and was successfully applied for the growth of nanotubes [42]. Therefore we kept note of this flaw of the FF and we continued our analysis on more system specific properties.

3.2.2 Evaluation of the CNT stability as function of its diameter

The relative energies of different nanotubes were used as system specific property to evaluate the carbon-carbon interaction. The construction of the NTs was done thanks to the "Atomic Simulation Environment" (ASE) [70],[71] (version 3.17.0) using the package "NanoCap" [72] (version 1.0b15) which allowed us to construct a nanotube with the desired chirality, length and either with free ends or two carbon caps at the end of the NT.
For this analysis, we built several series of fragments with different chirality, terminated with hydrogen atoms, in order to properly compare the results with those of *Hedman et al.* [73]. The aim of this study was to try to reproduce the relative energies between nanotubes belonging to the same NT series i.e. the NTs belonging to a specific K chirality family where $K = n + m$.
The results of the reference study of *Hedman et al.* [73] are reported in figure 7. The fragments that we analyzed belonged to the series of NT from $K = 10$ to $K = 18$ covering all the possible chiralities, i.e from n = m to n = K and $m = 0$.
Each fragment was then relaxed by means of a 50 Kelvin MD simulations with a Nosé-Hoover thermostat for 50000 MD steps of 0.25 fs in a squared cell of 27 Å of length and a subsequent geometry optimization with a convergence criterion of 1 (kcal/mol)/Å and the L-BFGS optimization procedure.
Unfortunately due to the H termination not all the calculations were able to converge, in fact even if the energies were basically converged (there were fluctuations with a maximum of 1 kcal/mol) the gradients were not going to zero. Therefore for some system we enlarged the convergence criterion to 3 (kcal/mol)/Å.
The resulting energies were compared with the energy of the most stable nanotube of the series and they are reported in the following Figure: 8 with a direct comparison with the theoretical results from [73] in the Figure 7.

[10]The FF was trained against DFT-Perdew-Burke-Ernzerhof values (DFT-PBE) [55]. The incorrect description of the lattice parameter c is a result of the false description of the dispersion interactions between the graphene layers within the PBE functional.

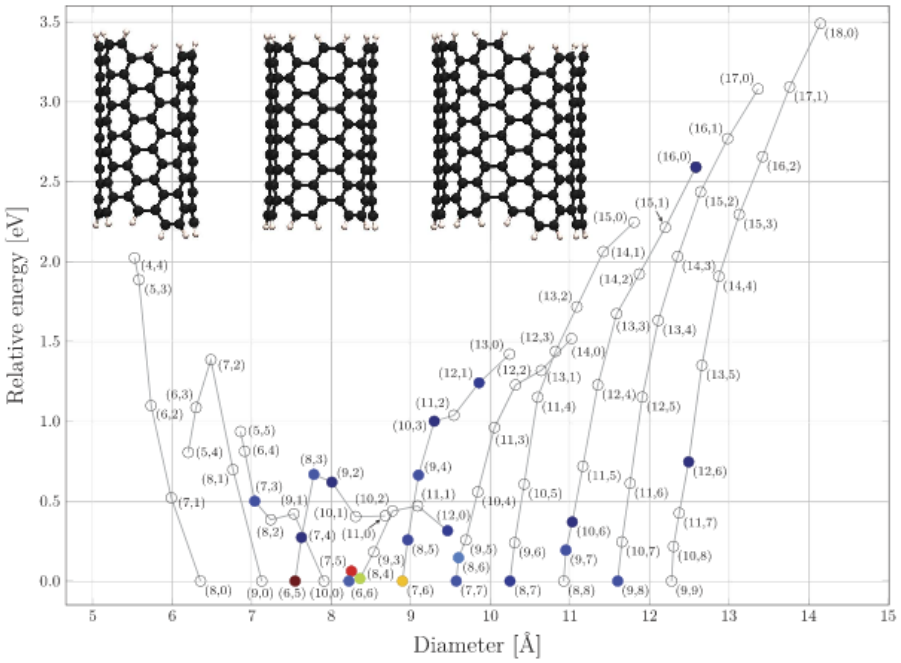

Figure 7: **Reference** relative energies between SWNTs of each (n + m) series from (n + m) = 8 to 18 plotted against the nanotube segment diameter. The zero line thus represents the most stable tube in each series. [73]

A comparison of the energy windows (ΔE) i.e. the difference between the most stable and least stable SWNT within the series are reported in the following table 7.

Series K =	10	12	14	16	18
ΔE_{Ref} [eV]	0.94	0.47	1.52	2.59	3.489
ΔE_{Calc} [eV]	-0.73	1.3	1.86	2.41	2.967

Table 7: Energy windows (ΔE) of the difference between the most stable and least stable SWNT within each series.

From the images and more quantitatively from table 7 it is visible that the force-field doesn't perform well for small diameter nanotubes e.g. for the series $K = 10$. Way more encouraging is the trend reported for the larger NT i.e. for those of the series of (16,0) or (18,0). Armchair nanotubes resulted in the stablest chirality for all series.
The dependency of results on the size, can be explained considering the fact that in small diameter nanotubes, quantum effects are extremely important. In fact for NTs with larger diameter the curvature is small, therefore for symmetry reasons the atomic eigen-

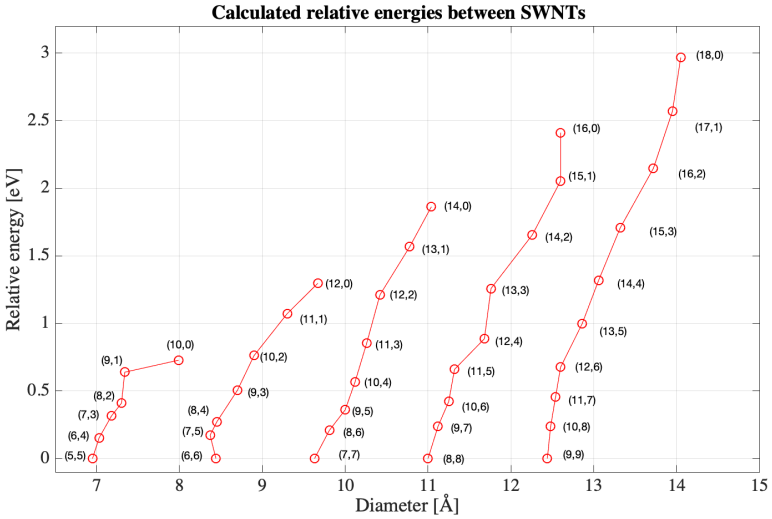

Figure 8: **Calculated** relative energies between SWNTs of each (n + m) series from (n + m) = 10 to 18 plotted against the nanotube segment diameter. The zero line thus represents the most stable tube in each series.

functions (that can be used for a tight binding description of the system) are the ones of a graphenic lattice i.e. sp_2 and p_z orbitals. For smaller nanotubes, due to the high curvature, the Hamiltonian for each atom loses the operation of mirror symmetry with respect to a plane passing from that atom: thus the eigenfunctions cannot be anymore those of before. This in turn leads to an hybridization [74] of the σ^* and of the π^* states, which also translates in an asymmetric electronic distribution inside and outside the nanotube [74] as shown in Figure: 9.
Since the force field was not trained to represent such distorted orbitals [55], it therefore looks natural that is not able to properly describe these kinds of system.
Even if the graphite formation energy is not so well represented from the force field, the results regarding the carbon-carbon interaction in nanotubes are nevertheless encouraging. Carbon NTs weren't indeed considered in the training set of the *NiCH* force field [55]: the trend for large K is therefore surprising. We concluded that also the carbon-carbon interaction is well represented for our purposes.
Knowing that small diameter nanotubes are a weak spot of our force field, we therefore decided to work with large nanotubes i.e. $K > 18$ in order to avoid the artefacts of the force field .

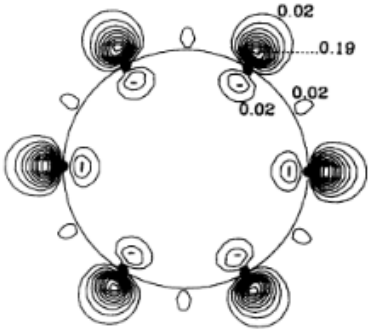

Figure 9: Contour plot of the charge density for a state at the Γ point for a (6,0) tube. The contours are in a plane perpendicular to the axis of the tube which contains six carbon atoms. The numbers quoted are in units of e^-/(a.u.). The circle represents a cross section of the cylinder on which the atoms lie. Image taken from [74]

3.3 Evaluation of Carbon-Nickel interaction

The last step in the process of evaluation of the applicability of such force field for a system of interest, is the study of the carbon-nickel interaction. In particular, we analyzed the binding energy of a small nickel cluster Ni_n with $n = 1, 2$ on a single vacancy defect site on a graphene sheet i.e. Ni_n/SV-GS.
The binding energy E_b is in this case defined as :

$$E_b = E_{sub+Ni} - E_{sub} - E_{Ni_n} \tag{3.3}$$

Where E_{sub+Ni_n}, E_{sub} and E_{Ni_n} represent the total energy of the Ni_n/SV-GS, of the defected graphene sheet and of the nickel cluster, respectively. Calculated results can be found in the following table 8 compared to reference results from the work of *Gao et al.* [76]. The

	Ni_1	Ni_2
Gao et al. [76]	-6.69	-8.05
This work	-10.10	-12.49

Table 8: Comparison of the binding energies for Ni_1/SV-GS and Ni_2/SV-GS. Reference results from *Gao et al.* [76] are calculated by means of DFT-PBE. All energy values are reported in eV

force field seems to overestimate both binding energies. Moreover, the energy difference between Ni_1 and Ni_2 is of 2.39 eV whereas from reference [76] is 1.36 eV . This difference is pretty large.
Calculated minimum energy geometries are found in figure 10.
In the case of Ni_2 the calculated geometry is extremely similar to

that of the reference [76], with two atoms symmetrically placed with respect to the perfectly planar graphene sheet.
In the case of Ni_1 (figures 10 a)-c)), results by means of the force field show the graphene sheet to be more distorted, with the carbons more lifted towards the nickel with respect to reference results [76]. This is likely caused by the parameterization of the force field, which probably embeds a strong interaction between nickel and carbons. This might also be the reason for the discrepancies in the binding energies in table 8.
Previous results on on nickel carbides [55] and on the growth of nanotubes on nickel clusters [47] proved nevertheless the the validity of the Ni-C interaction parameters and showed that even if this is slightly stronger than expected the predictive power of such tool is still relevant.

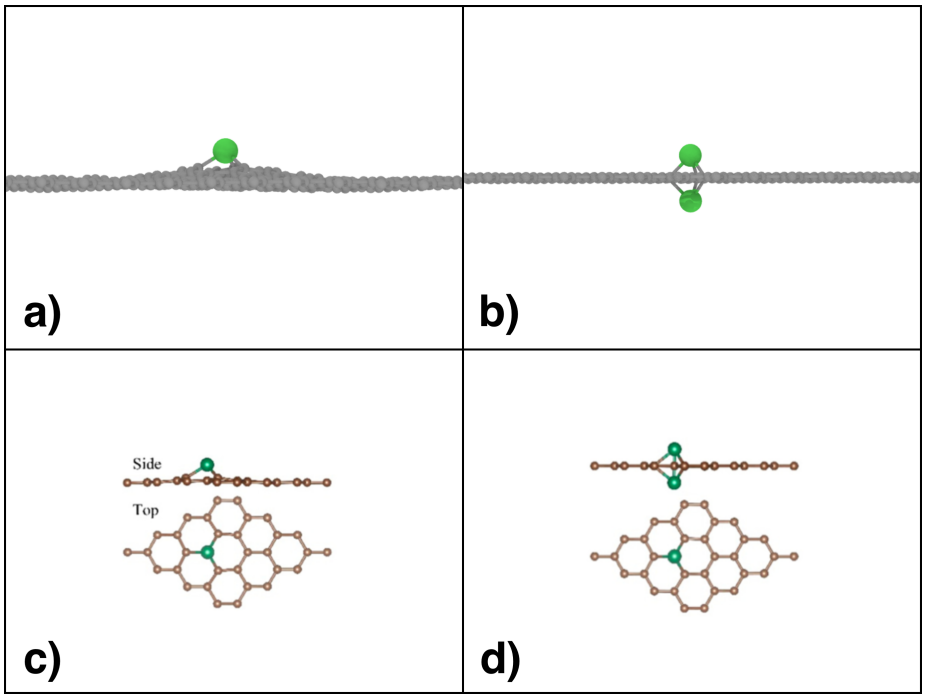

Figure 10: The lowest-energy geometry structures of Ni clusters on graphene's single vacancy. On the top line are the results calculated by means of *NiCH* force field whereas on the bottom line results from *Gao et al.* [76].
Images **a)-c)** are related to Ni_1 whereas **b)-d)** to Ni_2

3.4 A short but required comment

Force field based methods are the only tools on which we can rely on to simulate with atomic resolution large systems. If they are well trained they can serve as a very good descriptive tool allowing us to gain information regarding a variety of effects such as diffusion, intercalation and more in general reaction mechanisms.

Yet they have to be handled with care!

Their fundamental approximation of separating the potential energy in terms with a simple physical interpretation is not strictly correct: thus we cannot expect the accuracy of *ab initio* methods. Moreover , due to their intrinsic empirical nature, each force field is extremely training-set-dependent: hence, if one digs deep, problems with its applicability will always arise.

Force field methods have therefore to be regarded as a needed compromise.
Thus good sense is mandatory when analyzing results obtained with this tool.

In this spirit, the procedure of evaluation of the validity of the force field parameters for our system is not just allowing us to determine whether a certain FF is eligible or not to be used, but is also allowing us to determine its strengths and weaknesses. Knowing them is crucial to properly interpret the results and distinguish among possible artefacts or physically relevant events.

In conclusion of this chapter, we decided that the performances of the *NiCH* force field [55] were after all acceptable for our purposes.

4 Modelling a *ChemTEM* experiment of endohedrally confined metal particles

As already discussed in the introduction (see chapter 1), *ChemTEM* is a recently developed technique that aims to follow chemical transformations at the single-molecule level, with the electron beam of a transmission electron microscope (TEM) applied both as a tunable source of energy and a sub-Ångström imaging probe. The goal of this chapter is to provide a description of the relevant physical events taking place during such experiments on endohedrally trapped metal particles, to find a proper way to model and therefore to simulate them. In order to better understand the experimental results that we aim to reproduce, a brief digression on the endohedral confinement of metal clusters is required.

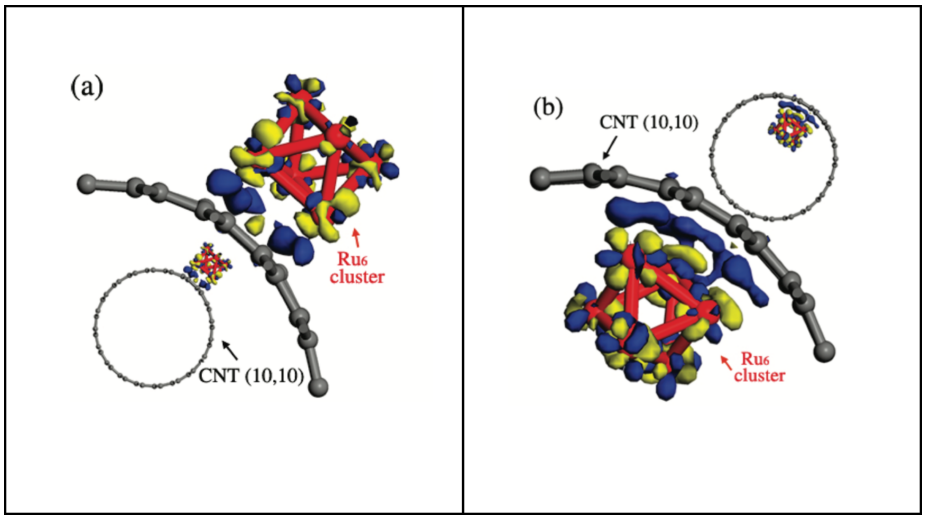

Figure 11: The differential electron density isosurfaces of (a) Ru_6-out and (b) Ru_6-in. The insets show the cross-section of a full nanotube model. The gray balls represent the carbon atoms and red bars the RuRu bonds. The blue and yellow areas suggest enriched and depleted electron density, respectively, with respect to free-standing clusters.
Image adapted from *Pan X. and Bao X.* [11].

4.1 Endohedral confinement of metal particles in a SWNT

Carbon nanotubes (CNT) have been into the scientific community's spotlight since their discovery in 1991 [8], due to their unique properties [9]. From then on, extensive studies have been carried out to fully understand and exploit all their amazing mechanical, electrical and catalytic properties [10]. The tubular nature of carbon nanotubes has always triggered wide research interest and in recent years the development of techniques to confine foreign species within the CNTs channels [11] has paved the way for a whole new plethora of applications. The internalization of different metal species within the channel of nanotubes has been reported to induce a so called confinement effect, which translates into a variation of the behaviour of the incorporated entity. Indeed the curvature of CNT walls causes the π electron density of the graphene layers to shift from the concave inner to the convex outer surface, which results in an electric potential difference. As a result, the molecules and nanomaterials on the exterior walls of CNTs likely display different properties (see for example image 11) and chemical reactivities from those confined within CNTs [11]. Thus a variety of intriguing effects arise upon such an *endohedral* confinement of foreign species within the nanotube channel, allowing both a change in the properties of the confined species and the potential electronic participation of the interior walls of the nanotube into e.g. the catalysis [13].
Several modes of reaction can be identified when considering catalytic processes of filled carbon nanotubes. More in general they can be classified into five distinct classes [13]:

1. CNT acts as a nanoreactor and the products escape the tube;
2. CNT acts as a nanoreactor and the products in this case remain inside the nanotube;
3. Heterogeneous catalysis mediated by endohedrally confined metal supported nanoparticles, catalysis takes place inside the nanotube;
4. Catalysis mediated by endohedrally confined metal supported nanoparticles, catalysis takes place outside the CNT;
5. Magnetic reactions, when the confined metal species turns out to be magnetic.

Detailed reviews of all these mechanisms can be found in the work of *Daniel Iglesias and Michele Melchionna* [13] and of *Xiulian Pan and Xinhe Bao* [11].

In addition to all this, as already stated in the introduction (chapter 1) the confinement imposed by the nanotube creates a "protected" environment perfectly suited for the study of reactions by means of TEM imaging.

4.2 Main interaction mechanisms between a SWNT and a confined metal particle under electron irradiation

Making use of all the concepts presented so far, *Kecheng Cao et al.* [12] turned their attention on studying the atomic scale dynamics of nanometre-sized metal clusters endohedrally confined in SWNTs[11] under electron irradiation. In particular, they focused on the evaluation of the interaction of the host NT with the nanocluster, thanks to the excess energy provided by the electron beam. It was proven that under e^- irradiation the metals interact with the nanotube by means of two types of fundamental mechanisms [12],[77],[19]:

1. Defect formation and propagation due to C-C bond breaking and Metal-C bond formation (Electron beam induced ejection EBIE);
2. Rearrangement of one form of carbon into another (Electron beam induced restructuring EBIR).

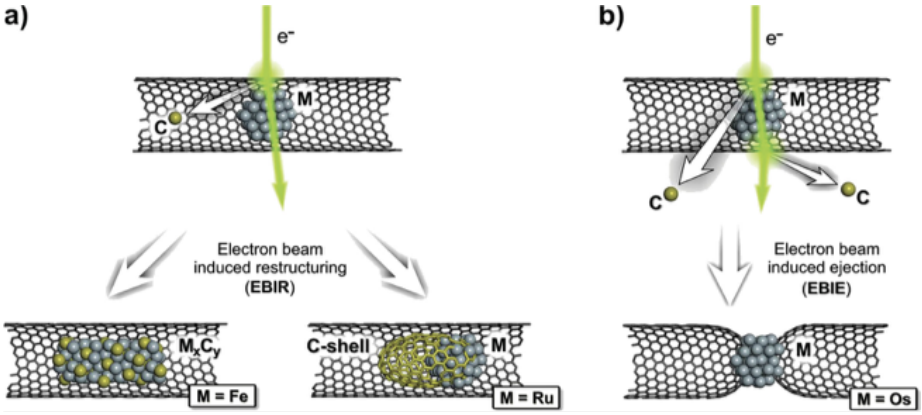

Figure 12: A simple representation of the effects promoted by the electron beam. Image adapted from [77].

Each electronic collision in principle could trigger an event of EBIR or of EBIE. However since a *ChemTEM* experiment might span hundreds of seconds (see for example Fig. 14 or Fig. 15), the number of collisions is very high and therefore the systems might undergo very complex dynamics and structural changes. In particular:

1. If EBIE is promoted over EBIR, the nanotubes are extensively damaged: Fe and Ni promote the rupture of the nanotube see figure 14;

[11] Their study was carried out for 14 technologically important metals.

2. If EBIR is promoted over EBIE, then three classes of processes can be identified depending on the metal considered, see figure 15.

Particularly interesting is the case of nickel, for which, if EBIR prevails over the EBIE process, there might be the formation of a new nanotube growing, inside the external NT section, from the surface of the entrapped nickel cluster (see figure 13).

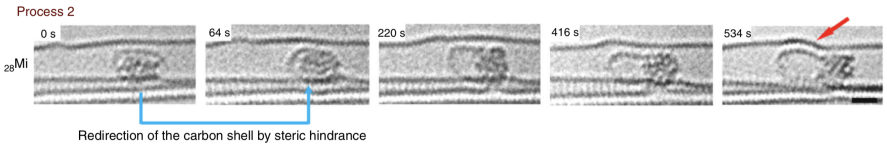

Figure 13: Time-series AC-HRTEM images illustrating a Ni nanocluster which abstracts carbon atoms from a point of contact with the host NT and promotes the formation of a new carbon structure. The direction in which the carbon shell grows is dictated by steric hindrance imposed by the host nanotube (indicated with blue arrows). The loss of carbon atoms from the host NTs leads to structural deformation (indicated with red arrow).
Image taken from [12].

4.3 Computational approaches

Considering the images of the actual *ChemTEM* experiment carried out by *Kecheng Cao et al.* [12] (figure 14 and 15), the very first thing that has to be noticed is the extremely long time scale (compared to MD) of such experiments which spans from tenths to hundreds of seconds[12].
Since the time step of our atomistic simulations is in the order of ~ 0.25 fs , also assuming that the CPU we are using would be able to complete each step calculation in 1 ns, simulating 100 s would require more than 10 years of computational time. The long time required by the experiments to show interesting complex effects, comes from the rarity (compared again to the MD scale) of the carbon displacement events induced by an electron collision. Indeed, considering as an example that the calculated area of a primitive cell of graphene is 0.051 nm^2 [39] and that in the primitive cell there are two atoms, then it can be easily calculated that on half of this area (i.e. where one carbon atom is present) with the reported [12] dose of the e-beam of $\sim 1.1 \times 10^6\ e^- nm^{-2} s^{-1}$, ca. 28 electrons every millisecond are impinging i.e. 1 electron every ca. 35 μs.
The experimental results demonstrated that the nickel clusters were able to lower the barrier for electron beam induced displacements (namely knock-on defects) and reactions between carbon and nickel atoms, promoting EBIE or EBIR processes [12],[77]. However, we need also to take into account the cross section for these effects: indeed only a fraction of the electrons colliding can effectively induce a defect (in particular those whose energy transfer to the target atom is sufficiently high). Since the beam energy threshold to generate electron-induced defects in a perfect SWNT is ~ 86 keV [12], for a beam energy of 80 keV (as the one used in the experiment of *Kecheng Cao et al.* [12]) just a small fraction of electrons colliding will generate a remarkable effect.
Therefore we can assume that for an eligible carbon atom, events either of EBIE or EBIR will happen even less frequently than 28 times a millisecond *i.e.* in a time window of width $\tau >> 35\ \mu s$.

According to previous studies [38],[41], the defect healing at the SWCNT – Ni catalyst interface needs to overcome a barrier of $E_a \approx$ 2.0–2.5 eV, thus the timescale for such event to occur is estimated to be 10^2-10^4 μs at room temperature. This time length is therefore within the time window of two different electron-carbon atom effective scattering events.

The presence of rare discrete events which are spaced in time of the same amount if not more than the time required for Ni-C systems to thermodynamically relax holds the key to simulate this kind of *ChemTEM* experiments, overcoming the computational time issues presented here and in section 2.6.

Following this idea *Skowron S. T., Lebedeva I. V., Popov A. M. and Bichoutskaia E.* developed an algorithm named *CompuTEM* [21], [23], [20] which proved the possibility of realistically model chemical reactions induced by an electron beam. Thanks to the accurate modelling of the electron-sample interaction, several mechanisms have been already elucidated: the work of *Alexander S. Sinitsa et al.* [19] described theoretically the formation of the endohedral nickel metallo-fullerenes and *Irina V. Lebedeva et al.* [22] explained the process of electron-assisted nanotube cutting catalyzed by a nickel cluster adsorbed on the outer shell of the NT. Nevertheless up to now, no atomistic explanation can be found for the most intriguing process a endohedral nickel cluster shows: the catalyzed growth of an endohedral secondary nanotube (again see figure 13).

Thus following the concepts of *Skowron S. T. et al.* [23] we supposed that a *ChemTEM* [17] experiment can be simulated by splitting it into two different disjoint parts:

1. *Electron collision induced displacement* and the subsequent fast dynamics of the bonds rearrangement (Fast dynamics);
2. *Relaxation* of the system into its thermodynamical minimum (Slow dynamics).

The multiscale nature of our approach is related to the simulation of these two parts, whose temporal scale is very different, with distinct methods. The complete system's dynamics will be obtained by the serial application of these two approaches.

The next two chapters are going to describe in detail the modelling of these two separate parts.

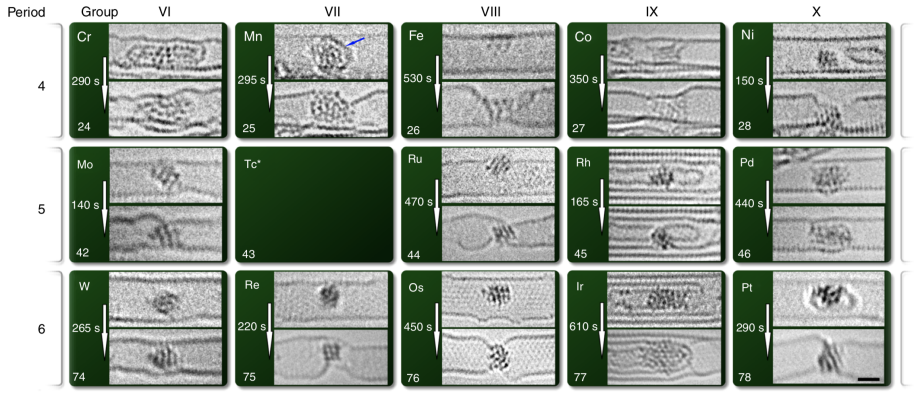

Figure 14: Examples of *ChemTEM* experiments on different transition metal clusters endohedrally confined in SWNT.
In this case, EBIE is promoted with respect to EBIR.
Image from [12].

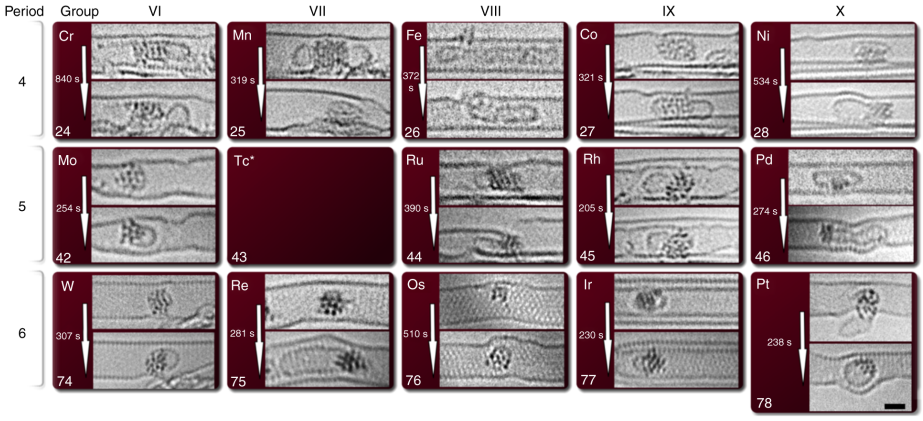

Figure 15: Examples of *ChemTEM* experiments on different transition metal clusters endohedrally confined in SWNT.
In this case, EBIR is promoted with respect to EBIE.
Image from [12].

5 Fast Dynamics: electron-carbon collisional events

When molecular dynamics tools for the simulation of extended systems, by means of reactive force field, were presented in section 2.2, it was often underlined how the electronic Hamiltonian equation 2.3 was completely disregarded. The electrons are therefore not described, whilst the nuclei move according to the Newton equation eq. 2.10 on a potential energy surface generated by semi-empirical potentials.
Hence lacking of a description of the electrons one might ask how to describe the damages induced by their interactions with the nuclei. The goal of this section is to answer this question presenting the modelling of the electron collisions in a simulation environment where electrons are indeed absent.
The first part is devoted to the description of the physical relevant events that occur during e^- beam's irradiation with a special focus on *knock-on* effects. The following parts are devoted to the development of a model and its implementation as an algorithm in order to simulate such events in our SCM [50] environment.

5.1 Radiation effects in solids

When a highly energetic particle such as an electron or ion strikes the atoms of a target, different mechanisms of energy or momentum transfer take place.
The most important primary radiation effects are [80]:

1. electronic excitation or ionization of individual atoms;
2. collective electronic excitations, e.g. plasmons;
3. breakage of bonds or cross-linking;
4. generation of phonons, leading to heating of the target;
5. displacement of atoms in the bulk of the target;
6. sputtering of atoms from the surface.

   Secondary effects are:

7. emission of photons, e.g. x-rays or visible light,
8. emission of secondary or Auger electrons, leading to a charging of the target.

The energy of the colliding electrons is an extremely important parameter since the different phenomena show different energy dependencies. When dealing with carbon, it is useful to group these contributions into those that lead to a displacement of atoms, namely *knock-on* effects, and those that do not, namely *excitations*. Generally, increasing particle energy excitations decrease in importance, whereas knock-on effects increase [80].

5.1.1 Excitations

The *excitation of phonons*, leading to a heating of the specimen, is mainly due to inelastic scattering of the projectile by electrons. Phonon generation by collisions with nuclei at energy transfers in the meV range are of less importance. In small objects such as carbon nanostructures, which are typically of some 10–100 nm in size, electron irradiation is expected not to heat the specimen by more than by a few degrees [80].
Electronic excitations such as intraband or interband (electron–hole pair) excitations or excitons are energy transfers in the eV range. These excitations are of significance in insulators and to a less extent in semiconductors. The excited states can cause local atomic bonding instabilities and rearrangement, leading to bond breakage; this phenomenon is commonly known as radiolysis. Metals are immune to this type of damage [80].
The dissipation of *plasmons* causes heating but little damage [80].
Ionization is of importance in insulators and semiconductors where the lifetime of the excited electrons is long enough to cause irreversible bond breaking. In a metal, ionization is quenched instantaneously (10^{15} s), local perturbations in electric charge are removed in that time scale.
The generation of *X-rays* or *Auger electrons* behaves similarly to ionization damage [80].

5.1.2 Atom displacements: knock-on

Atom displacements occur by knock-on collisions of highly energetic electrons or ions with the nuclei of the atoms of the specimen.
The knock-on displacement event occurs within a very short time. The time scales of the effects taking part during the production of atomic defects are [78]:

1. 10^{-21} s: energy transfer from the particle to the nucleus (primary knock-on);
2. 10^{-13} s: interatomic collisions (cascade);
3. 10^{-11} s: dissipation of epithermal energy (stable defects and clusters);
4. $\geq 10^{-11}$ s: thermal migration of point defects.

It is recognized [80], [12] [79] that radiation-induced modification of carbon nanostructures under electron irradiation is dominated by knock-on.
In order to properly implement such events in our simulation environment an accurate description of their physics has first to be taken into account.

5.2 Physics of knock-on events

As already mentioned before a knock-on displacement event occurs within a very short time: in particular the energy transfer happens in ca. 10^{-21} s.
The collision of a fast electron with a nucleus can be treated by using a simple Coulomb potential. This is justified because the screening effect of the surrounding electrons can be neglected [80]. Considering the scattering of an electron with a nucleus, due to the huge difference in masses there is a small transfer of energy and therefore momentum transfer arises almost entirely from the change in direction of the electron.
The angular dependence on the scattering angle Ω of the energy T transferred to the nucleus is given by:

$$T(\Omega) = T_{max} cos^2(\Omega) \tag{5.1}$$

Where Ω is the emission angle i.e. the angle between the initial direction of the colliding electron and the direction of displacement of the displaced atom (Fig. 16) and T_{max} is the maximum transferred energy by an head-on collision i.e. $\Omega = 0$ and the electron is scattered backwards. It appears clear from figure 16 a) not just that the probability of missing the target is much higher than the probability of a direct head-on collision, but also that large angle scattering events will have a predominant role.
Momentum conservation gives the maximum energy transfer of a relativistic particle (energy E, mass m) to a nucleus (transferred energy T, mass M, $\Omega = 0$):

$$T_{max} = \frac{2ME(E + 2mc^2)}{(M + m)^2c^2 + 2ME} \tag{5.2}$$

We call threshold energy E_d, the minimum energy which is transferred to an atom in order to generate a vacancy-interstitial pair that doesn't spontaneously recombine. The threshold energy shows an anisotropy regarding the direction of the knock-on with respect to the crystal lattice. Directions that would require e.g. a very strong deformation of the lattice are highly unfavourable for the knock-on, resulting in very high E_d so that e.g. for graphene we have a difference of up to 757 eV [79] for in plane or out of plane energy thresholds.

The important quantity to consider to interpret experiments is the displacement cross section. The displacement rate p of each atom is given by:

$$p = j\sigma_d \tag{5.3}$$

Where σ_d is the displacement cross section and j is the beam current density.
In order to obtain the displacement cross section, it's required the calculation of the Coulomb scattering cross section. Since we are dealing with very energetic particles relativistic corrections are required. The theoretical cross section for Coulomb scattering between a relativistic electron and a nucleus has been derived by Mott [81],[82] as a solution of the Dirac equation. McKinley and Feshbach found an approximate formula [83] accurate up to middle Z elements, which rewritten as function of the emission energy T it reads:

$$\sigma(T) = \left(\frac{Ze^2}{4\pi\epsilon_0 2m_0c^2}\frac{T_{max}}{T}\right)^2 \frac{1-\beta^2}{\beta^4} \left[1 - \beta^2\frac{T}{T_{max}} + \pi\frac{Ze^2}{\hbar c}\beta\left(\sqrt{\frac{T}{T_{max}}} - \frac{T}{T_{max}}\right)\right] \quad (5.4)$$

Where $\beta = v/c$, Z is the atomic number of the nucleus and m_0 is the electron rest mass. Displacement cross section can be obtained by integrating the cross section for Coulomb scattering inside the

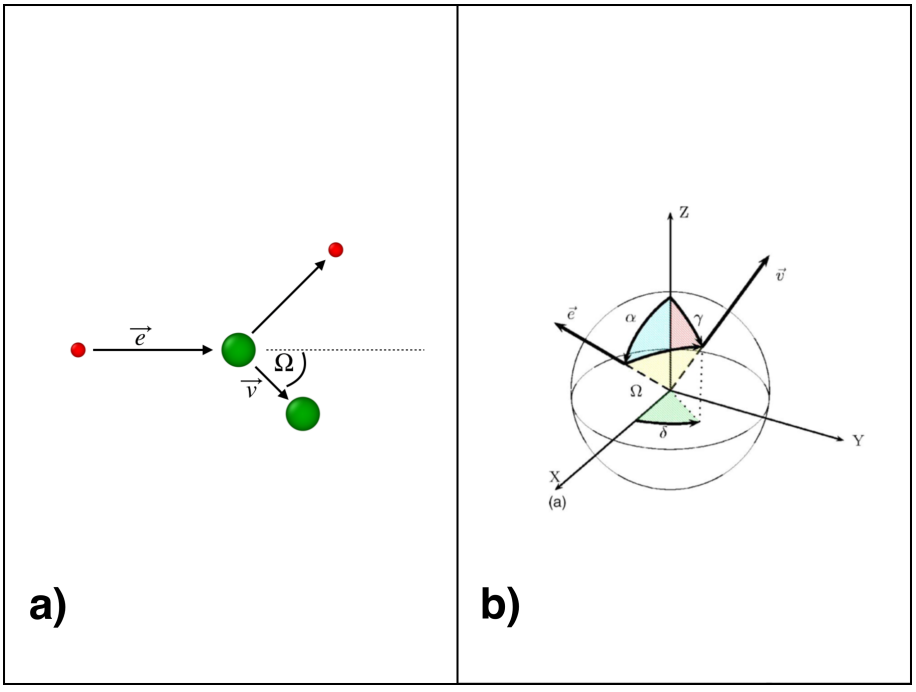

Figure 16: Schematic 2D **a)** and 3D **b)** representations of the angles involved in the process of direct knock-on. The electron incidence direction is $\vec{e}$ whereas $\vec{v}$ is the atom emission direction. Image **b)** adapted from [79].

energy domain S over which emission conditions are satisfied:

$$\sigma_d = \int_{S(T>E_d)} \sigma(T) \frac{4\pi}{T_{max}} dT \tag{5.5}$$

$\sigma(T)$ is a very general quantity for a certain atom Z, whereas σ_d completely depends upon the environment in which the atom Z is. Displacement cross section can be calculated only if the full $E_d(\gamma, \delta)$ is known, where γ and δ are the zenith and azimuth angles according to figure 16 b).

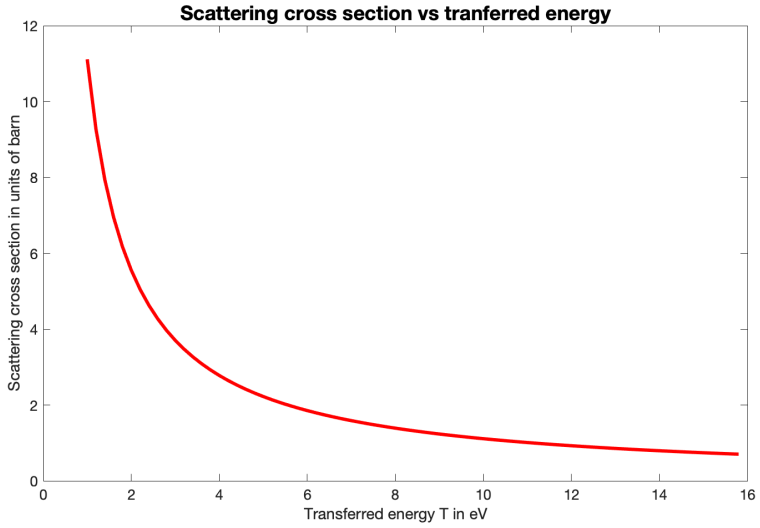

Figure 17: Cross section for Coulomb scattering as function of the transferred energy T

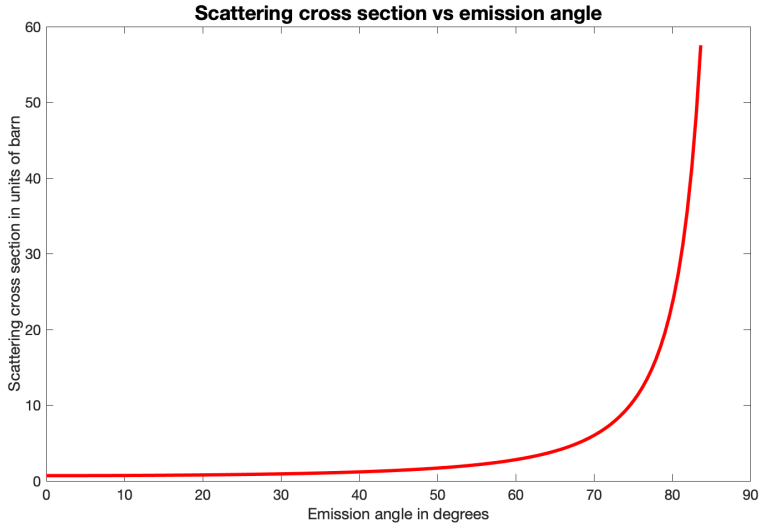

Figure 18: Cross section for Coulomb scattering as function of the emission angle Ω

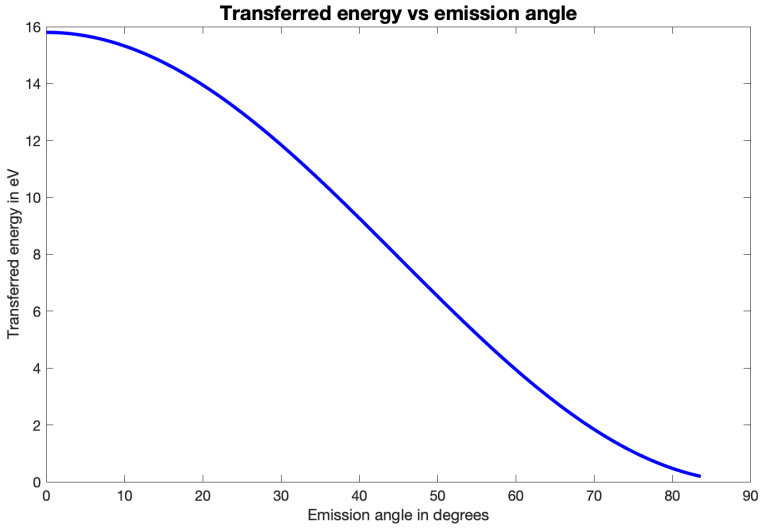

Figure 19: Transferred energy T as function of the emission angle Ω

5.3 Knock-on and atomistic simulations: fundamental hypotheses

Now that the physics of knock-on effects have been elucidated, we still need to model such events into something that our SCM [50] atomistic simulation environment can handle.
In order to do so, we based our work on four fundamental hypotheses:

1. Thanks to the large time difference between the characteristic time of interaction between the nucleus and the colliding electron c.a. 10^{-21} s and our MD step 0.25 fs, we assumed that a reasonable approximation is to treat *collisions as punctual events*, i.e. the target nucleus all of a sudden acquires an excess kinetic energy T due to the collision with the electron;

2. We assumed that each colliding electron *interacts just once*;

3. Due to the large mass of a nickel atom, the energy transferred to it by an electronic collisions is very small[12]. Thus we assumed that colliding electrons *interact just with carbon atoms.*

4. After the interaction the e^- *escapes the specimen*;

The amount of transferred energy T will depend on an angle Ω between the direction of the electron beam and the direction of emission of the target atom according to equation 5.1.
Due to the ever changing nature of the lattice environment for each target atom it is impossible to determine a unique $E_d(\gamma, \delta)$ and therefore σ_d cannot be used[13].
Therefore the model needs to rely only on the Coulomb scattering cross section $\sigma(T)$. Thus the probability of choosing a certain emission angle Ω for the target atom will have to take into account the dependency of $\sigma(T)$ (equation 5.4) from it. In this way higher angle collisions are going to be promoted more than lower angles, resembling the physical behaviour depicted in figure 18.
With these assumptions we can simulate the effect of scattering-mediated energy transfer with no need for a proper description of the electron.

[12] According to *Kecheng Cao et al.* [12], T_{max} for Ni is 3.23 eV, which is not sufficient to displace irreversibly an atom in a cluster.

[13] Already assuming a perfect nanotube in absence of any cluster inside, thermal fluctuations would induce some temporal variation of the E_d anisotropy. Moreover considering the presence of former knock-on defects, the relaxation and reconstruction of the lattice, the possible presence of nickel atoms in different positions with respect to the target atom and the dependency on time of all these effects it results clear that during a *ChemTEM* experiment, rather than a single steady $E_d(\gamma, \delta)$ we will have a set of $E_{d_j}(\gamma, \delta, t)$ for each *j-th* atom.

The following sections are devoted to explain how this model has been "translated" into a language understandable by our simulation software.

5.3.1 Collision event algorithm

Electron irradiation of nanotubes by means of TEM experiments is usually performed in nontilted case where the tube axis lies perpendicular to the direction of the electron beam [79]. Thus being $\alpha = 0$ (Fig.16 b)), we have $\Omega = \gamma$ and the knock-on model will depend only upon the two characteristic γ and δ emission and azimuth angles respectively.
The main idea is to create a *Python* code that handles the SCM input data, generating a new set of velocities for a specific target atom.
The algorithm of the code is designed according to the following steps:

1. Target atom selection;
2. Generation of the characteristic angles (γ, δ);
3. Generation of the excess (x,y,z) velocities for the target atom.

First of all, a list of all the carbon atoms that are within a sphere of 11 Å from the center of mass of the Nickel cluster is created. Afterwards a random atom is chosen from this list as target atom for the collision. At this point random angles γ^* and δ^* are chosen with $0 \leq \gamma^* \leq \gamma_{max}$ and $0 \leq \delta^* \leq 360$[14], respectively. A random probability p is then generated and the emission angle γ^* is accepted if:

$$p \leq \frac{\sigma(T(\gamma^*))}{\sigma(T(\gamma_{max}))} \tag{5.6}$$

Angle δ^* always has unitary acceptance.
If the condition of eq. 5.6 is not met, a new pair of $(\gamma^{**}, \delta^{**})$ is generated and the acceptance for the emission angle is tested again. This is repeated until the criterion is met.
Condition 5.6 ensures a choice of angles that better mimic the process of Coulomb scattering for which larger emission angles have an higher cross section $\sigma(T)$ (Fig.18).
Once the angles (γ, δ) are determined, they are used to generate the excess velocities in a set of three orthonormal axes in which vector $\tilde{Z}$ is parallel to the direction of the e^- beam.

[14] Here γ_{max} is the maximum angle for which the emission condition is met, i.e. $T(\gamma_{max}) = E_d$. However, as stated before 5.3, the threshold energy is a quantity which is not well defined for our system. For this reason, we assumed for simplicity that the minimum energy that we wanted to consider was 10 eV lower than the $T_{max} = 15.8$ eV of the experiment. In such a way according to equation 5.2, $\gamma_{max} \sim 45$ and we assumed this to be an upper bound of the true maximum emission angle i.e. the maximum angle for which a knock-on is observed.

These velocities are generated simply by projection:

$$\begin{cases} V_{\tilde{Z}} = V_{max} cos(\gamma) \\ V_{\tilde{X}} = V_{max} sin(\gamma) cos(\delta) \\ V_{\tilde{Y}} = V_{max} sin(\gamma) sin(\delta) \end{cases} \tag{5.7}$$

Where from momentum transfer $V_{max} = \sqrt{\frac{2T(\gamma)}{M}}$.
These velocities are added to the initial velocities of the target atom and the system is ready to be simulated by means of a normal MD simulation.

It has to be clarified that the set of e^- beam's axis $(\tilde{X}, \tilde{Y}, \tilde{Z})$ where $\tilde{Z}$ represents the electron beam's direction, it is not necessarily parallel to the standard cartesian (X,Y,Z) set of the SCM simulation box. During the simulation the nanotube might indeed undergo some roto-translations [15] which have anyway no impact on the dynamics of the system in a periodic boundary condition empty box simulation. If the direction $\tilde{Z}$ would be fixed in our (X,Y,Z) frame, then the e^- beam would hit each time different areas of the nanotube and with different angles, failing in reproducing a *ChemTEM* experiment.
Thus the direction $\tilde{Z}$ need to stay orthogonal to the same side of the tube throughout the simulation.
The solution to this problem is found by generating a set of system specific "fake" coordinates namely $\tilde{z}$ and $\tilde{y}$ constructed on the flight as quasi-radial [16] normalized coordinates i.e. each one is built as normalized distance vector between the mean positions of a pair of specific atoms (Fig.20 a)−c)). These 8 atoms were chosen because they met two criteria: they were sufficiently far from the "hot" region where the cluster is present and the e^- beam is inducing damages; and their relative position could vary very little, i.e. just due to thermal fluctuations.
By means of Grahm-Schmidt orthonormalization two orthonormal radial coordinates $(\tilde{Z}, \tilde{Y})$ are constructed and by means of a cross

[15] The fbMC seemed to be the main source of this effect.
[16] They are not radial due to the way they are constructed, see next.

product the axial ($\tilde{X}$) coordinate is generated too:

$$\begin{cases} \tilde{Z} = \tilde{z} \\ \\ \tilde{Y} = \tilde{y} - \frac{\langle \tilde{y}, \tilde{Z} \rangle}{\langle \tilde{Z}, \tilde{Z} \rangle} \tilde{Z} \\ \\ \tilde{X} = \tilde{Z} \times \tilde{Y} \end{cases} \tag{5.8}$$

A transformation matrix $M_{Z->\tilde{Z}}$ is then designed each time such that:

$$M \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \tilde{X} \\ \tilde{Y} \\ \tilde{Z} \end{bmatrix} \tag{5.9}$$

This set $(\tilde{X}, \tilde{Y}, \tilde{Z})$ moves quasi-coherently [17] with the system and it is used to generate the e^- beam's coherent velocities defined in equation 5.7.
Since all the data files handeled by the SCM software require the expression of the coordinates with respect to the standard (X,Y,Z), the inverse transformation $\tilde{M} = M_{\tilde{Z}->Z} = M^{-1}$ is then applied to the excess velocities (equations 5.7) obtaining the velocities in the common cartesian reference frame.
These are then overwritten onto the input files and the simulation can start.

[17] The set moves quasi-coherently because the axes are generated by means of distance vectors between mean positions of pair of atoms. Thus between two different collisional events, thermal fluctuations might induce a slight modification of the directions determined by these vectors. Anyway, the variations are extremely small and we assumed them irrelevant.

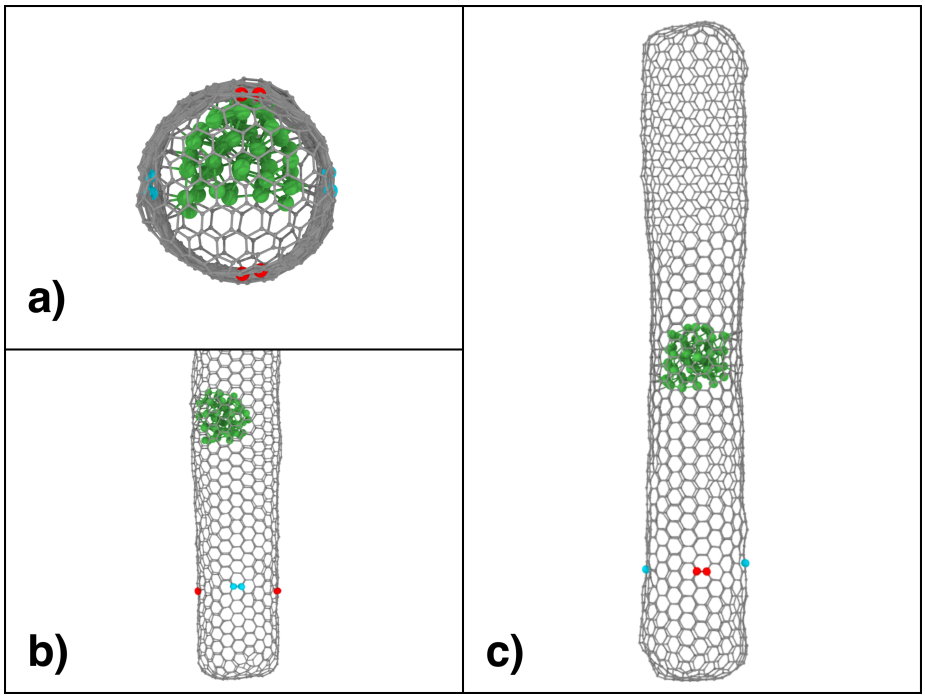

Figure 20: Different viewports of reference system: highlighted in red are the atoms related to direction $\tilde{z}$ and in blue atoms related to $\tilde{y}$ direction.

a) top view;
b) side view;
c) cross section view.

5.3.2 Determination of the maximum transferred energy T_{max}

Known the kinetic energy of the colliding relativistic electron, equation 5.2 gives us the maximum transferred energy for a head-on collision.
In the case of an 80 keV beam, $T_{max} = 15.8$ eV [12] which
is ≈ 6 eV lower than the estimated [85] displacement threshold energy E_d of ca. 22 eV for an (11,11) carbon nanotube.
E_d is related to the minimum amount of energy required to displace an atom, thus it depends upon the strength of the interactions. In theoretical calculations this will depend upon the way interactions are parameterized, i.e. in our case upon the force field. Thus using in our code as maximum transferrable energy a value of 15.8 eV, might not be the best choice to represent the 80 keV beam's effects with our *NiCH-ReaxFF* representation.
We will therefore need to find the *NiCH* force field's equivalent T_{max}.
In order to identify this quantity we first need to determine the equivalent displacement energy threshold.
For this purpose calculations for a perfect (11,11) nanotube at different transferred energies have been carried out. The velocities were added keeping the emission angle null i.e. $\gamma = 0$. Calculations led to estimate this value as ca. 26 eV which is higher as reported values of ≈ 22 eV [85], ≈ 17 eV [12] and ≈ 20.5 eV [79] .
Knowing the equivalent displacement energy threshold, we can now consider a lower value as equivalent maximum transferred energy.
As a first choice we assumed $T_{max} = 21$ eV for our calculations.

In this chapter we developed and parameterized a physically based algorithm able to reproduce the effect of kinetic energy transfer, from a relativistic electron to a carbon atom in a nanotube.

Now we need to simulate the dynamics of the system within the large time window between two different electronic collisions. This is going to be the goal of the next chapter.

.

6 Slow Dynamics: long time scale thermodynamically driven relaxation effects

As already discussed in chapter 4 the fundamental hypothesis of our modelling of a *ChemTEM* experiment, is the possibility of splitting the dynamics of the system under electron irradiation into: an out of equilibrium fast dynamics part (collisions) and a thermodynamically-driven slow dynamics part (relaxation).
The previous chapter was devoted to the development and parameterization of an algorithm to simulate the electron-carbon collision events.
The goal of this chapter is to find the optimal parameters for the method used to simulate the slow relaxation effects.

The algorithm used is the so called force biased Monte Carlo (fbMC) for which a theoretical background has been presented in section 2.6.1. As already discussed in that section the heart of fbMC algorithm relies in an iterative displacement of each atom along a random direction, and the acceptance of this new position is tested against the forces acting on the particle during that specific displacement. Thus the evolution of the system dynamics is based on a simple probabilistic description. The system simulated by fbMC is mainly located in an equilibrium state which once in a how is left to evolve toward another equilibrium configuration. Thus many iterations are able to bring the system around a local minimum of the PES.
fbMC doesn't allow us to have a proper description of time [45] and therefore the dynamics of the system has to be view as a pseudo-dynamics i.e. each Monte Carlo (MC) step has not to be considered as a true MD step. The trajectories of the particles moving along the PES are nonetheless of physical relevance if Δ is small enough [45],[44] and the principle of detailed balance equation 2.19 is satisfied.
From an initial state A in the configurational space, fbMC algorithm will mostly lead the system into state B whose properties are a combination of the following:

1. Highest gradients to escape;
2. Highest distance in the configurational space from other local minima;

3. Lowest amount of unfavourable MC steps [18] required to reach it from A.

For this reason even if fbMC is able to efficiently sample the configurational space, yet it is not truly guaranteed that each time it is applied, the system will end in the potential energy surface's global minimum. Nevertheless, this method has been chosen since it has been successfully used in past works [47],[42] for the simulation of relaxation processes during the growth of carbon nanotubes onto a nickel cluster catalyst (Fig.21). Thus due to the evident similarities among those systems and ours, we assumed that this method could bring us reliable results.

The next section is devoted to the determination of the unknown parameters required by the fbMC algorithm, namely:

1. Maximum displacement Δ;
2. Equivalent Temperature T;
3. Number of MC steps.

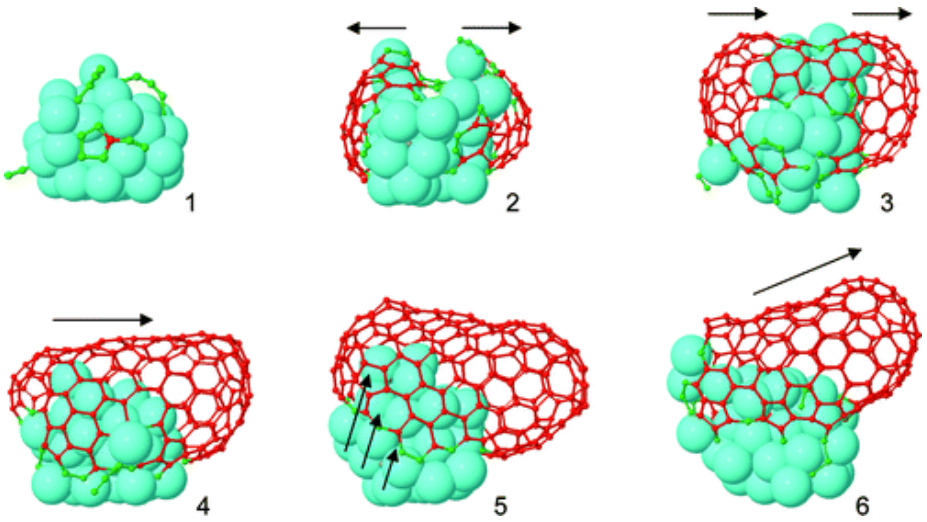

Figure 21: Simulated growth of a defect free carbon nanotube by means of a mixed MD-fbMC approach. Image taken from [42]

[18] This means a step whose displacement is chosen against the forces.

6.1 Extended analysis of fbMC parameters for a 4-fold vacancy in a (11,11) nanotube

The goal of this section is to justify the choice of the values for the fbMC's parameters, namely the temperature, the displacement and the number of steps. To do so, we needed a proper benchmark, i.e. test system whose known dynamics would allow us to properly address the performances of the selected parameters.

Due to an absence of prior references regarding nickel clusters encapsulated into carbon nanotubes, we decided to switch our attention to the more known [75], [86], [88] process of relaxation of 4-fold (TV) vacancy in CNTs. It is reported [87] that barrier for the relaxation of a 2-fold vacancy (DV) in a nanotube is $E_a = 0.224$ eV. Given the transition rate according to equation 2.16, even if we would consider the ratio $\frac{Q^{\#}}{Q_A} = 1$, the time required for such healing would be of 920 ps, which is completely within the estimated time window of two distinct electron collisional events. The study a 4-fold vacancy was chosen since we were not interested in any estimation of the energies involved, but rather we wanted to track in time the evolution of the system. Indeed it is reported [75], [86] that for all chiralities the minimum energy structure is always the so called (5775) shown in figure 22. The target geometry is therefore known. Moreover, since in this case more atoms need to rearrange their position with respect to the relaxation of a DV, we might expect that the healing of a TV would require more time than the relaxation of a 2-fold vacancy defect, and therefore would allow us to analyze a larger time window. Thus since the target (5775) minimum energy structure for a 4-fold vacancy relaxation is known, we decided to test the parameters against this process.

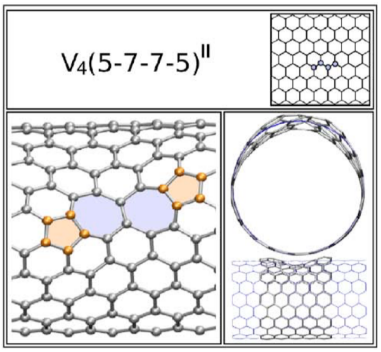

Figure 22: Healed 4-fold vacancy in a (10,10) nanotube showing the typical (5775) geometry. Image taken from [86]

6.1.1 Evaluation of the optimal T and Δ

To find the best T and Δ a systematic study comprising more than 500 calculations, in order to have statistical relevance, was carried out.
The calculations were performed for different values of the parameters:

1. Displacement Δ: 0.08 Å, 0.10 Å and 0.12 Å;
2. Temperature T: 900 K, 1000 K, 1100 K and 1200 K ;

The analysis of the optimal amount of MC steps has been split through the study.
At first, a fixed total amount of 500000 Monte Carlo (MC) steps was chosen.
In order to address the presence of possible number of steps' dependent effects the total amount of Monte Carlo steps was then divided into five fbMC subroutines of different MC length:

1. 1 fbMC routine of 500000 steps;
2. 2 fbMC routines of 250000 steps;
3. 5 fbMC routines of 100000 steps;
4. 10 fbMC routines of 50000 steps;
5. 50 fbMC routines of 10000 steps.

Each routine was separated from the next one by 25000 MD steps. Indeed although it is true that fbMC leads the system in proximity of an equilibrium state in the PES, nevertheless each *i-th* atom is constantly and recursively moved around with a mean displacement $\langle \delta_{i,j} \rangle \approx \frac{\Delta}{3}$ [44]. This translates in the accumulation of a certain amount of potential energy for each atom which has to be released to properly reach the minimum energy structure (see figure 27 for an example). The 25000 MD steps, seemed to be a conservative choice for the time required from the system to relax this accumulated strain and restore the proper velocities distribution.
Each simulation began and ended with an equilibrating MD routine again of 25000 MD steps.
The temperature for the molecular dynamics routines was set to the fbMC temperature.
For each setup comprising the set (Δ, T, Nroutines) 10 calculations were run to have a better statistical representation.
The results coming from these simulations were then fed through

an MD routine that smoothly i.e. with a temperature coefficient of -0.001343 K/MD step, brought each system from its starting temperature down to 5 K [19] . These were considered the final outcomes and therefore each energy value in this chapter is evaluated at 5 K.

The setup just described and depicted in figure 23 has been used to produce every result of this chapter.

The complete set of results is shown, for the sake of readability, at the end of the section in Fig. 26.
The partial results (averaged over all MC step distribution) as function of Δ and T are presented in the following table 9.

Δ \ T	900 K	1000 K	1100 K	1200 K
0.08 [Å]	-250893.98	-250902.58	-	-250926.76
0.10 [Å]	-250897.96	- 250928.51	-250938.74	-250951.03
0.12 [Å]	-250973.53	-250992.36	-251024.04	-251050.14

Table 9: Average final energy of the 4-fold vacancy initial structure at 5 K as function of fbMC displacement Δ and T. Average is done with respect to the different N fbMC subroutines. Energies are reported in kcal/mol.

The trend is pretty clear, showing (without any surprise) that increasing temperature and displacement there is an increase of average relaxation, with the minimum average energy reached for highest T and highest Δ.
Looking at the full set of outcomes in figures 26, the results for Δ = 0.12 Å and 1200K seem to suggest that the single MC routine of

[19] The advantage of choosing such a T as reference for comparison of the energies rather than the standard procedure of comparing the geometry relaxed structures, was that we could scan a larger amount of different setups in a shorter computational time without losing accuracy. Indeed structures which were not completely healed might still have strong gradients and therefore not all the final geometries would easily converge by means of L-BFGS geometry optimization. Although converging to 5 K the systems were just close to the equilibrium point, thermal fluctuations were very small i.e. max. $\sim$ 1 kcal/mol compared to energy differences among differently relaxed structures, which are in the order of tenths of kcal/mol. Thus the results were still comparable and we assumed that a further processing by means of geometry optimization would not lead to substantial variation of the behaviour, costing however time and computational power.

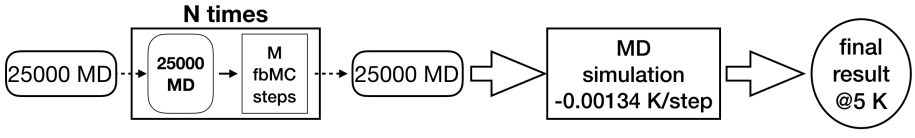

Figure 23: Sequence of calculations performed to reach each result of this section.

500000 MC doesn't give the best performances among all routines at that displacement and temperature. Nevertheless, 10 results were not sufficient to do an accurate statistics, therefore we weren't able to understand if this might have been caused by statistical fluctuations or by the presence of some sort of other effects dependent upon the intermediate MD steps in between different fbMC subroutines.

6.1.2 Identifying the best combination of Monte Carlo steps

In order to clarify whether there might be a different outcome by inserting some MD steps between the application of the fbMC routines, 30 more calculations for a number N of subroutines $N = 1$ and $N = 10$ (i.e. 1×500000 MC and 10×50000 MC) with
$T = 1200$ K and $\Delta = 0.12$ Å were performed to gain statistical relevance.
Results show that in average 1×500000 MC relax to -251033.08 kcal/mol and 10×50000 MC to -251062.75 kcal/mol. Moreover in the following figure 24 are plotted the distribution of the results along the different minimum energy structures. All the results could indeed be grouped, with an uncertainty of ~ 1 kcal/mol into specific common energy values which were related to specific geometries. This is in line with the idea that fbMC simulates the dynamics of a system in equilibrium that once in a while moves towards another accessible minimum [44]. It is interesting to notice that the force field is able to distinguish the minimum energy structure for a 4-fold vacancy, i.e. the reconstructed (5775) geometry figures 22 and 24, which is in agreement with references [86] and [75].
Moreover, from Fig.24 it results clear that inserting some MD steps in between the application of Monte Carlo subroutines, favours not just the average relaxation of the system, but it increases the probability that the system would relax in the minimum energy structure.

One possible explanation for these results is that in the fbMC routine the movement of the *i-th* atom at the $(N-1)^{th}$ MC step is modifying the PES for the movements of the other atoms at the N^{th} step. This means that all the atoms are moved according to the past potential energy surface PES_{N-1}, which then is suddenly refreshed (PES_N). Although this in general should not be an issue, in carbon nanotubes, the healing of defects takes place through rotations of bonds [41], i.e. the simultaneous movement of two atoms. Thus since in fbMC, PES and therefore forces are updated for each atom after the $(N-1)^{th}$ step has been completed for the whole system, there might be the chance that this effect is not properly considered.
The introduction of some MD steps in between might therefore lead the system to a small rearrangement of the atoms and to a different fbMC starting geometry, thus enabling to overcome this issue and allowing a better relaxation.
Nevertheless, this supposition has not been proved yet and further and deeper studies regarding the behaviour of fbMC algorithm are

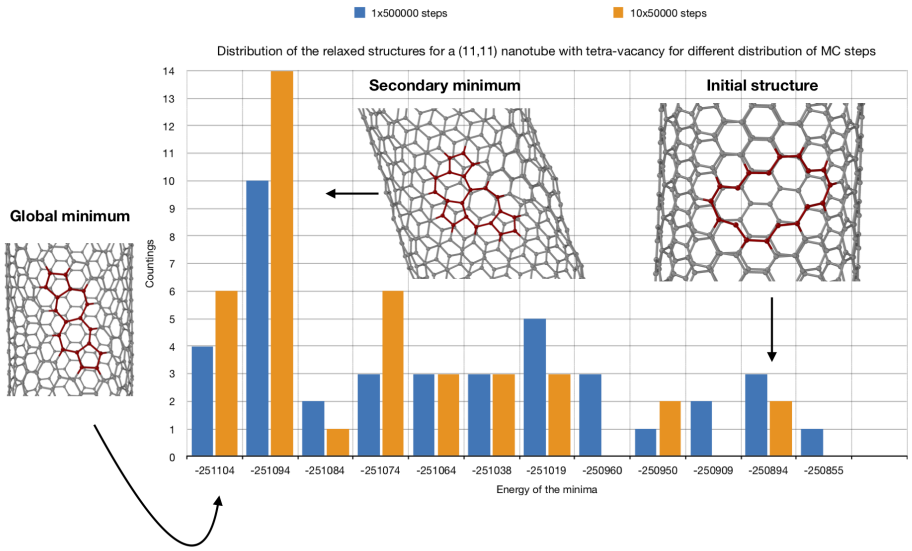

Figure 24: Distribution in energy of the results at 5 K for the different amount of Monte Carlo subroutines for the relaxation of a 4-fold vacancy defect. In blue the single 1×500000 MC routine and in orange the 10×50000 MC steps routine. The three most important geometries are reported and in red are highlighted the atoms directly taking part to the surface reconstruction. Energies are reported in kcal/mol.

for sure required.

Although the reasons for this are not completely clear, results shown in Fig.24 suggest that to better relax the system, i.e. to reach faster the minimum energy structure, we will have to find the best combination of total Monte Carlo steps and the best distribution of them along the simulation, i.e. the best set M and N where N is the number of force biased subroutines and M is the number of Monte Carlo steps for each subroutine.
To do so, we started analyzing the relaxation trend for one single subroutine with variable MC steps, i.e. $N = 1$ and $M = 10000$, 50000, 100000, 150000, 250000, 400000 and 500000.
The temperature was set at 1200 K and the displacement to 0.12 Å. For each setup 20 simulations were run to gain statistical relevance.

Results are shown in the following figure 25.

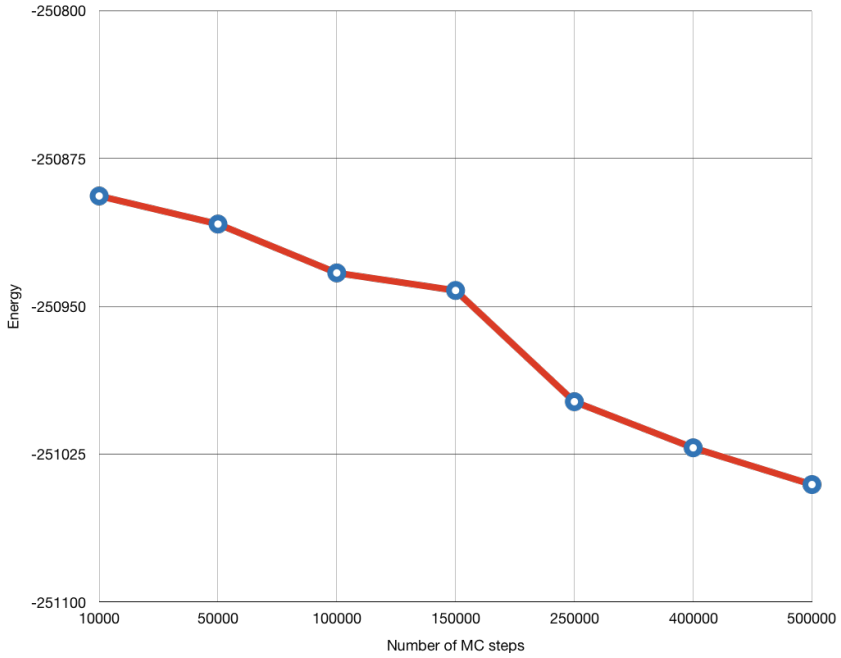

Figure 25: Average final energy at 5 K for a single fbMC routine as function of a different amount of M MC steps. Energies are reported in kcal/mol.

It can be seen that average energy is a monotonically decreasing function of the number of MC steps and therefore that none of these setups seems to have reached the equilibrium probability distribution in the configurational space [20] .

At this point, we switched our attention towards the dependency of the average energy on the amount of MC subroutines. Considered the fact that this whole study has been carried out in order to have a physically meaningful and justified fbMC setup that would help us simulating long time scales effects with a reduced computational cost with respect to a normal MD simulation, it appears clear that we aim towards maximizing the average energy relaxation, whilst minimizing as much as possible the total amount of MC and MD steps. for the sake of clarity, the system that we aim to simulate to

[20]Indeed I would expect that, given a certain temperature and increasing the number of MC steps $M \to \infty$, we will reach a steady value for the ensemble average energy, populating the minima in the configurational space according to the probability distribution function for that temperature. Being able to see a variation in the average energy means that we still didn't reach the thermodynamical equilibrium distribution probability in the configurational space. More about the probability distribution function of the fbMC algorithm is reported in appendix B.

N \ M	75000	100000	150000	250000
2×	-250933	-250953	-251002	-251029
3×	-	-250994	-251054	-
4×	-250944	-251039	-	-
5×	-	-251057	-	-
6×	-251007	-	-	-

Table 10: Final 4-fold vacancy average energies at 5K as function of N number of fbMC subroutines and M number of MC steps for each subroutine. Energies are reported in [kcal/mol].

replicate the *ChemTEM* experiments of *Kecheng Cao et al.* [12], is an (11,11) capped carbon nanotube with encapsulated a 55 nickel atoms cluster for a total of 1854 atoms[21], each MC step on average has a computational time cost of ~ 29 ms whereas an MD step would cost ~ 23.1 ms [22]. We assumed 550000 to be an upper bound for the total amount of MC steps and 200000 for the total MD steps that we could afford to pay. For these reasons, we decided to consider just subroutines of $M = 75000$, 100000, 150000, and 200000 number of MC steps for our study for different N number of subroutines.
Once again for each setup 20 calculations were performed in order to have statistical relevance.
The results of the calculations are reported in table 10 from which it can be seen that the best performance is reached by repeating three times a fbMC subroutine of 150000 steps. Indeed the average energy reached by it is -251054 kcal/mol, lower than the average -251033 kcal/mol reached for 1×500000 MC but higher than the calculated -251063 kcal/mol for a 10×50000 MC setup [23]. Nevertheless, among all the most performing setups (table 11) the 3×150000 MC seems to require the least amount of total estimated computational time ~ 15360 s.

[21] In appendix A the complete description of the reference system used to replicate these experiments is reported.
[22] Computational time cost is evaluated with respect to the Intel(R) Core(TM) i7-6700 CPU 3.40GHz processor which the computer I was provided was endowed.
[23] These energies were previously calculated and reported in this section.

	1×500000	2×250000	3×150000	5×100000	10×50000
Energy	-251033	-251029	-251054	-251057	-251063
Time [s]	15655	16232	15360	17965	20852

Table 11: Estimated required total computational time for the most performing (in terms of average energy) fbMC setups. Estimations are based on the approximation that we require $\sim$ 23.1 ms each MD step and $\sim$ 29 ms each MC step. Energies are reported in kcal/mol. Estimated computational time is reported in seconds.

It has to be stressed that the study presented in this chapter was done on a test system different from the one of the experiment that we aim to simulate. Gradients and minimum energy structures will be completely different in presence of the nickel. In the absence of any reference would be nevertheless impossible to properly address the performances of the different possible fbMC setups, leaving us with no justification for the parameters chosen. Therefore these results have to be considered as a guideline, a reasonable justification for the values used to simulate the experiment.

For all these reasons we decided to simulate the time window in between two different electronic collisional events by means of a force biased Monte Carlo algorithm with the following setup:

1. MC displacement $\Delta = 0.12$ Å;
2. MC temperature $T = 1200$ K;
3. three subroutines of 150000 MC steps separated by 25000 MD steps (3×150000 MC);
4. two equilibrating MD routines of 25000 MD steps, one before the application of the 3×150000 MC and one after.

$\Delta = 0.08$Å	900 K	1	2	3	4	5	6	7	8	9	10	Average	Group average
	1x												
	2x	-250894,15	-250894,05	-250895	-250894	-250894	-250894,81	-250894,28	-250894,55	-250895	-250895	-250894,39	-250893,98
	5x	-250894	-250894	-250894	-250894.67	-250895.48	-250894	-250895.23	-250894	-250895	-250895	-250894,33	
	10x	-250895	-250893	-250894	-250894	-250891	-250894	-250894	-250894	-250895	-250891	-250893,5	
	50x	-250894	-250893	-250894	-250894	-250894	-250891	-250895	-250894	-250894	-250894	-250893,7	
	1000 K												
	1x	-250894	-250894	-250894	-250894	-250894	-250895	-250894	-250900	-250894	-250894	-250894,7	-250902,58
	2x	-250998	-250894	-250894	-250894	-250998	-250891	-250895	-250894	-250894		-250916,89	
	5x	-250894	-250894	-250894	-250894	-250894	-250894	-250895	-250998	-250895	-250894	-250904,6	
	10x	-250894	-250894	-250894	-250894	-250894	-250894	-250894	-250894	-250895	-250894	-250894,11	
	50x												
	1200 K												
	1x	-250894	-251074	-250894.61	-250895	-250894	-250949	-251017	-250894	-250894	-250895	-250934	-250926,76
	2x	-250895	-250895	-250895	-250998	-251001	-250894	-250894	-250894	-250999	-250894	-250925,9	
	5x	-250999	-251074	-250894	-250894	-250998	-250998	-250998	-250894	-250895	-251000	-250964,4	
	10x	-250894	-250895	-250894	-250894	-250894	-250894	-251000	-250894	-250894	-250894	-250904,7	
	50x	-250894	-250895	-250894	-250895	-251000	-250894	-250894	-250894	-250894	-250894	-250904,8	

$\Delta = 0.10$Å	900 K	1	2	3	4	5	6	7	8	9	10	Average	Group average
	1x	-250894	-250893	-250949	-250895	-250894	-250894	-250894	-250894	-250891	-250894	-250899,2	-250897,96
	2x	-250894	-250894	-250894	-250894	-250901	-250894	-250894	-250893	-250892	-250895	-250894,5	
	5x	-250895	-250894	-250894	-250894	-250943	-250894	-250894	-250893	-250894	-250895	-250899	
	10x	-250894	-250894	-250894	-250895	-250894	-250894	-250949	-250894	-250895	-250894	-250899,7	
	50x	-250890	-250894	-250894	-250894	-250932	-250894	-250894	-250894	-250894	-250894	-250897,4	
	1000 K												
	1x	-250894	-250894	-250894	-250894	-250894	-250949	-250949	-250894	-251017	-250894	-250917,3	-250928,51
	2x	-251094	-250894	-250894	-251000	-250931	-250894	-250891	-251017	-250894	-250894	-250940,3	
	5x	-250894	-250891	-250894	-250892	-251017	-251017	-250894	-250894	-250894	-250895	-250918,2	
	10x	-250894	-250894	-250894	-250894	-251017	-250895	-251021	-251017	-250894	-250894	-250935,56	
	50x	-251017	-250894	-250894	-250894	-251017	-250898	-250894	-250894	-250894	-251016	-250931,2	
	1100 K												
	1x	-251017	-250894	-250894	-250949	-250894	-251000	-251017	-250894	-250894	-250894	-250934,7	-250938,74
	2x	-250894	-250890	-250894	-251018	-251017	-251074	-251094	-250894	-250894	-250998	-250966,7	
	5x	-251018	-250894	-250894	-250895	-250894	-250895	-250895	-250894	-251074	-250998	-250935,1	
	10x	-250998	-250894	-250894	-251017	-250894	-251017	-250948	-250894	-250895	-251074	-250952,5	
	50x	-250899	-250894	-250895	-250894	-250894	-250894	-250998	-250891	-250894	-250894	-250904,7	
	1200 K												
	1x	-250998	-251018	-250898	-250895	-251104	-250894	-250894		-250894	-251038	-250961,64	-250951,03
	2x	-251000	-250894	-250894	-250999	-250998	-250894	-251094	-250999	-250891	-250894	-250955,7	
	5x	-250894	-251018	-251017	-251020	-251017	-250898	-250894	-251017	-250962	-250894	-250963,1	
	10x	-250894	-250894	-251017	-251017	-250895	-250894	-250999	-251017	-250894	-251036	-250955,7	
	50x	-250949	-251093	-250894	-250894	-250894	-250888	-250894	-250894	-250895	-250895	-250919	

$\Delta = 0.12$Å	900 K	1	2	3	4	5	6	7	8	9	10	Average	Group average
	1x	-250895	-251000	-250894	-250858	-250955	-251001	-250894	-251094	-250934	-251063	-250958,8	-250973,53
	2x	-251063	-250894	-251074	-250894	-250894	-251104	-251093	-251093	-251094	-251017	-251022	
	5x	-251094	-250857	-250894	-251017	-250955	-250894	-251094	-251017	-250894	-251017	-250973,3	
	10x	-251017	-250894	-250894	-250932	-251094	-250931	-251017	-250894	-251104	-250945	-250957,56	
	50x	-250894	-250899	-250908	-250997	-251017	-250894	-250949	-250893	-251094	-250953	-250956	
	1000 K												
	1x	-250894	-251094	-251094	-251038	-250894	-251094	-250950	-251094	-250900	-251094	-251014,6	-250992,36
	2x	-250894	-251094	-250949	-251003	-251073	-250963	-250894	-251017	-251074	-250955	-250991,6	
	5x	-251019	-250947	-251017	-251017	-250954	-250894	-250894	-251104	-250895	-251017	-250975,8	
	10x	-250894	-251094	-251064	-250962	-251017	-251093	-250949	-250894	-251017	-251017	-251000,1	
	50x	-250894	-250894	-251017	-251002	-251017	-251093	-251074	-250895	-250894	-251017	-250979,7	
	1100 K												
	1x	-251094	-251094	-251016	-251105	-251105	-251104	-251086	-251094	-251017	-251060	-251077,5	-251024,04
	2x	-251060	-251094	-251017	-251039	-251000	-251036	-251103	-251017	-251073	-250894	-251033,3	
	5x	-251105	-250949	-250894	-251094	-251073	-251104	-251094	-250894	-251016	-251017	-251024	
	10x	-250949	-251017	-251017	-250963	-251021	-251093	-250898	-250894	-251094	-250949	-250989,5	
	50x	-250932	-251074	-250895	-251104	-250895	-251093	-251059	-250891	-251063	-250953	-250995,9	
	1200 K												
	1x	-250894	-251038	-251019	-251094	-251021	-251093	-250962	-251094	-251084	-251093	-251039,2	-251050,14
	2x	-251093	-251104	-250962	-250894	-251104	-251063	-251093	-251094	-250962	-251093	-251046,2	
	5x	-251075	-250962	-250949	-250894	-251094	-251104	-251094	-251104	-250894	-251093	-251026,3	
	10x	-251093	-251104	-251094	-251093	-251094	-251074	-251104	-251064	-251094	-251104	-251091,8	
	50x	-251094	-250950	-251039	-251094	-251016	-251093	-251104	-251084	-251104	-250894	-251047,2	

Figure 26: Complete set of results obtained for different displacement, different temperatures and different distribution of the total amount of 500000 MC steps. In light green are reported the structures that hit the minimum energy structure. Energies are reported in kcal/mol.

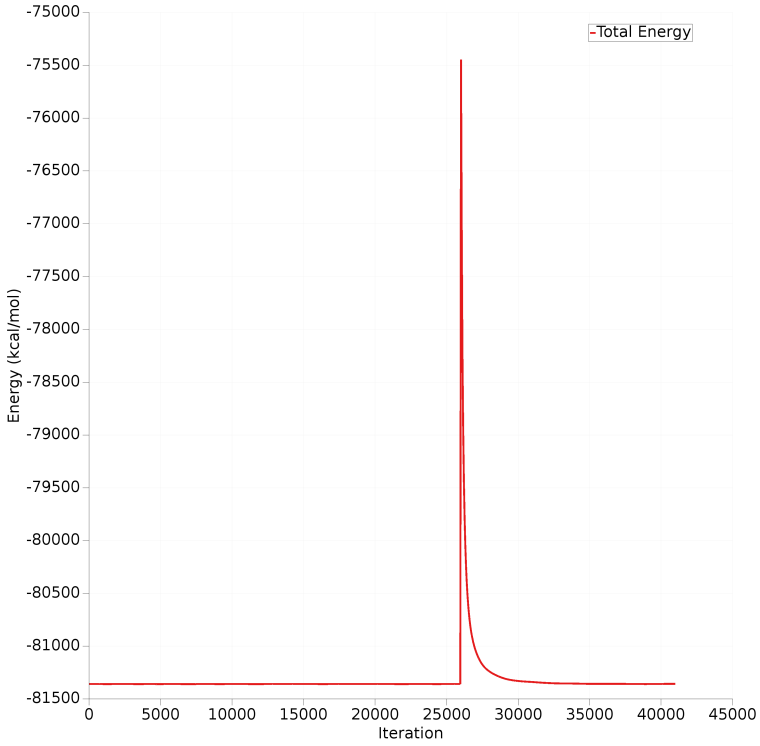

Figure 27: Example of total energy as function of time with the application of a fbMC routine at the 26000^{th} step. In the SCM environment, due to the absence of a fbMC's intrinsic timescale [45], the energy of the system during its application is not reported.
The spike in the energy right after its application is caused by the fact that during the fbMC routine each atom of the system is displaced on average of $\frac{\Delta}{3}$, thus each atom is not in the most stable position. After the application of the fbMC the energy steadily decreases due to the release of this strain.

7 Development of a *ChemTEM* simulation tool for experiments with endohedrally confined nickel particles

The previous chapters were focused on the development of an algorithm to simulate the electronic collisional event (chapter 5) and the optimization of the force biased Monte Carlo's parameters to thermally relax the system after each collision (chapter 6).
This chapter is devoted to the description of the algorithm developed to joint these two parts, generating a fully autonomous simulation tool able to reproduce in sequence many collisions and the corresponding dynamics, i.e able to reproduce a *ChemTEM* [17] experiment for endohedrally confined nickel particles.
The basic idea is to consider the sequence $collision \rightarrow relaxation$ as the building block for an iterative algorithm.

The algorithm has been implemented as a *Python* code in such a way that it could interact and handle the source files of the SCM package. The complete code is reported in appendix C.

The idea is to have four building blocks which are :

1. A normal molecular dynamics simulation that has the goal of restoring the equilibrium of the system. Its duration has been chosen to be of 25000 MD steps for a total amount of 6.25 ps of simulated time;

2. The *electron beam* block;

3. Another normal molecular dynamics simulation with the goal again of restoring all the velocities, and therefore temperature, to the equilibrium values. This block has been chosen to be of 40000 MD steps i.e. 10 ps;

4. The force biased Monte Carlo routine with the setup presented in the previous chapter i.e. 3×150000 MC steps, for a total amount of four MD separating subsections of 25000 MD steps.

After the simulation of one block, final geometry and velocities are passed as input files for the next simulation block. After the termination of all four blocks, the simulation is restarted using as input the final atoms' positions and velocities coming from the fbMC block. Each iteration of the four blocks should therefore hypothetically simulate one event of collision and the subsequent time window before the next one.

In all MD simulations, the temperature is set to 1200 K by means of a Nosé-Hoover thermostat with a damping constant of 100 fs and an MD step of 0.25 fs.
Such high temperature has been chosen because, as it's reported in the supplementary material of the work of *Kecheng Cao et al.* [12], temperatures in the range of $870 - 1370$ K are needed for activating transition metal catalysts for the transformation and growth of carbon structures. Therefore a parallelism can be drawn between the processes triggered and promoted by the 80 keV e^-beam in nanotubes and analogous processes initiated thermally at $870 - 1370$ K in bulk, even though technically the temperature of the materials in the AC-HRTEM experiments were essentially ambient, e.g. $290 - 300$ K [12].

This algorithm is inspired and follows the main ideas exposed in the work of *Skowron S. T. et al.* [23].

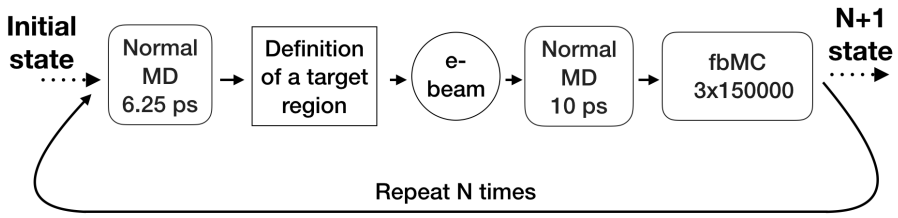

Figure 28: Scheme of the iterative algorithm implemented on *Python* that aims to simulate a *ChemTEM* experiment.

7.1 Electron beam block

The workflow of the simulation (see figure 28) is trivial.
Nevertheless, the *electron beam* block needs to be further explained. As always, indeed, we need to find a trade off between accuracy and computational time cost. In our simulations force biased Monte Carlo is the bottleneck since it is extremely time consuming: as reported in section 6.1 even for our optimized setup it would require a total time of $\sim$ 15300 s. Thus before starting it, we need to be sure that this is really going to be useful.
Let's explain this better.
As stated in section 5.3 we are not able to make use of the σ_d, the displacement cross section, since we aren't able to properly describe for each atom the emission energy threshold E_d. For this reason, the algorithm simulating the electron-carbon collisions has been developed in such a way to consider just the Coulomb scattering cross section $\sigma(T)$, disregarding the probability that this event would effectively lead to a knock-on defect. Moreover, the acceptance criterion for the emission angle γ (equation 5.6) will select with higher probability larger angles, thus lower transferred energies according to eq. 5.1. Therefore once the target atom and the set of angles (γ, δ) are selected, we are still not ensured that this will lead to the generation of a knock on defect. Blindly passing this result to the fbMC routine might cost a lot of time without giving us any information.
Thus we need to introduce a second acceptance criterion [24] , this time on the event itself.
Indeed we would like to verify if a knock-on event has happened: if this is not the case, then the algorithm is restarted and a new target atom and a new set of emission angles are generated. Of course, due to system's complexity, this selection cannot be made a priori only given (γ, δ).
For this reason after having selected a target atom (this time within a sphere of $r = 11$ Å from the nickels' centre of mass) and having modified its velocities according to sec. 5.3.1 a very short MD simulation of 2000 MD steps i.e. 500 fs is started.
The idea is therefore to check after such cheap MD simulation if an *effective* modification of the geometry has happened[25].

[24] The first acceptance criterion is the one regarding the emission angle γ for which we cycle over new values for γ each time condition 5.6 is not satisfied.

[25] If this is not the case, as stated above, a new target atom and new velocities are generated. This is repeated until an *effective* collisional event took place or until the system is not able to produce any defect anymore i.e. this procedure is repeated over 100 times without producing any knock-on (in this case an error message is produced whilst the simulation is stopped).

The short simulation time i.e. 2000 MD steps was chosen since it is the shortest (in time) conservative choice [26] that allow us to fast identify whether an *effective* event took place.

7.1.1 Effective collisional events: Events detection algorithm

The procedure just presented is based on the possibility of determining whether a collision led to an event which is worth to track in time by means of fbMC or not i.e. an *effective* event. The key point is thus the definition of event "worth" to be followed: in fact not all collisions lead to a knock-on, but not all knock-on lead, after thermalization of the system, to a modification of the structure. It might indeed happen that an atom is kicked in a position but after some nanoseconds it regains its initial position: also in this case we would have a loss of computational time that we want to avoid.
It has to be clarified immediately and this will be stressed more throughout the whole discussion, that this selection procedure is purely artificial and has no physical parallelism. The true dynamics of the system could be simulated by considering every single event of collision, even if that would not lead to any modification of the system. Indeed when electrons collide just a very small fraction of them generate a knock-on defect. Nevertheless this procedure is needed in order to have the chance to simulate the evolution of the system in a reasonable amount of time. The event's acceptance criterion has to be chosen with extreme care, since it could bias the dynamics of the system leading to a complete artificial behaviour.
The acceptance algorithm is based on recognizing whether an atom has *effectively* changed its neighbouring environment. Now due to the always changing geometry of the specimen under irradiation and the very complex nature of it (i.e. a mixed system of a large amount of carbon and nickel atoms) it is extremely difficult to define a unique environment for each carbon atom [27] . Thus it is difficult to determine general conditions for which an atom is effectively displaced.
Forced by this reason we decided to introduce a simplification: each event of collision will be classified according to the *effective* processes occurring in the pristine state of the system, i.e. when the nanotube has no defects yet. The pristine reference system used is an (11,11) CNT, endowed with a vacancy defect for a total of 1799 carbon atoms, and a nickel cluster endohedrally confined of 55

[26] The excess kinetic energy was already redistributed among other atoms in this amount of time and the target atom could experience a well defined equilibrium state.

[27] This is the same reason for which we cannot uniquely define the displacement energy threshold.

atoms. A complete description of such reference geometry is given in appendix A.
Thus we call *effective* processes those processes that would lead the system from initial state A (reference system Fig.42)to final state B with $A \neq B$ even after the application of the force biased algorithm [28].
We have to be aware that in such a way the dynamics at the beginning of the experiment might be well described, whereas some bias might be introduced when the geometry of the system is far from the initial one.
In order to identify such *effective* processes for our reference system in figure 42, we conducted the following:

1. Preliminary study on a small set of targets with the usage of the expensive fbMC in order to identify the most likely *effective* events;

2. Definition and implementation on *Python* of geometrical based conditions in order to automatically classify each collisional event;

3. A study on a large sample of target atoms and collisions in order to gain some statistics for the so classified *effective* events.

The first preliminary study was conducted on a reduced set of atoms, i.e. those carbons within a sphere of 6.5 Å from the centre of mass of the nickel cluster. In this way, we could identify four target atoms. For each atom of this set, 30 different collision events were simulated according to sec. 5.3.1. With the velocities so generated, an MD simulation of 2000 MD steps was then ran for each setup (γ, δ, *target*). The results were then fed into the usual 3x150000 fbMC routine [29].

We could observe that among all possible final geometries of the short MD simulation just four led to a modified fbMC's final geometry with respect to the reference system Fig. 42.
These were the following:

1. *Ejection* event Fig.29 a), i.e. the target atom is ejected from the side of the nanotube. After the application of the fbMC

[28] It might indeed happen that the simulated collision would lead to a temporary modification of the geometry i.e. $A \rightarrow A^*$, which doesn't heal during the normal MD simulation after the collision but that experiences a reverse process $A^* \rightarrow A$ during the fbMC. Since such a collision event doesn't induce any modification on the system, it is useless for our purpose and the computational time required to simulate it would be wasted.

[29] The time cost of the force biased Monte Carlo routine had forced us to use a very small set of target carbons for this study.

algorithm the single vacancy defect is stabilized by a nickel atom Fig.29 b);

2. *Dissolution* event Fig.29 c), i.e. the target atom is detached from the nanotube and is dissolved in the nickel cluster. The carbon is never adsorbed onto the surface of the cluster but rather it is dissolved in the volume of the cluster mostly occupying subsurface sites, where it can maximize the interaction with nickel atoms. This behaviour is consistent with previous results [93], [91]. During the application of the fbMC algorithm the carbon atom simply diffuses inside the nickel cluster, moving from one site to a neighbouring one. The single vacancy defect in the nanotube wall is stabilized by a nickel atom;

3. *pre-dissolution* event see figure 29 d) i.e. the target atom is displaced inside the nanotube section, placed below the plane identified by the former neighbouring C atom, being still connected only to one of them. The vacancy defect in the side of the nanotube is stabilized by a nickel atom, which moreover hinders the displaced C atom from regaining its previous position in the lattice. This event might evolve either towards the dissolution of this atom inside the cluster or in the restoration of the initial unperturbed state;

4. *Double atom pre-dissolution* event which was observed just once and no image is reported. This process is similar to the pre-dissolution, but in this case two neighbouring atoms are displaced towards the inside of the tube. They remain connected between each other and one of the two is still connected to one of its carbon neighbours.

It has to be once again stressed that these mechanisms were observed during the perturbation of the pristine system (again Fig. 42). For sure, in the presence of an highly modified geometry with respect to the one of the system analyzed here, these processes might lead to a different outcome after the fbMC, as it might be that other mechanisms for the knock-on formation would arise.
We will therefore have to be careful when analyzing results for highly deformed geometries.

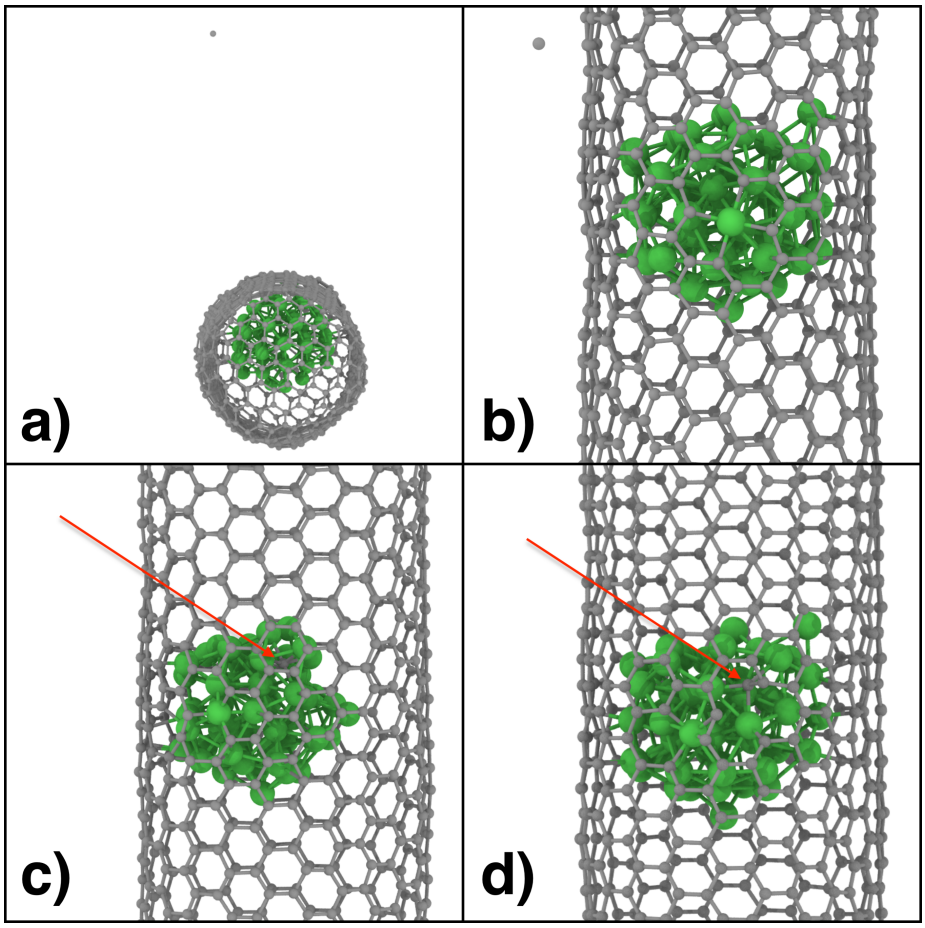

Figure 29: Snapshots of the three most probable collision-induced events:

Image **a)** show the ejection of a carbon atom from the nanotube. In this case, the electron beam is displacing atoms away from the cluster.
Image **b)** shows the zoom on the area where the atom of image **a)** has been ejected. In this specific case, the carbon atom has been ejected from the side of a pre-existent single vacancy defect, thus it is visible that the nickel binds to the so formed 2-fold vacancy defect, stabilizing it.
Image **c)** shows the dissolution of a carbon in the cluster. It is visible that the atom (red arrow) is not dissolved on its surface but rather in a sub-superficial site. The nickel atom stabilizes the defect in the side of the nanotube.
Image **d)** shows the pre-dissolution of a carbon. In this case, the target atom (red arrow) is still connected to one carbon.

As already mentioned among these mechanisms the least probable one was the *double atom pre-dissolution* which was observed just once and therefore it is not considered in the following discussion [30].
The other three mechanisms can be identified by means of atom's specific information, namely:

1. BO final bond order[31];
2. ΔR position variation;
3. C_{nn} final number of carbon atoms nearest neighbours (n.n.);
4. ΔNi_{nn} variation of the total number of nickel atoms n.n.;
5. K number of unchanged C nearest neighbours.

In this fashion, we could define some criteria in order to classify on the fly, i.e. after each short MD simulation, the collision event:

1. An *ejection* mechanism E is detected when the final bond order of the target atom is below a certain threshold of 0.3 ;
2. A *dissolution* mechanism D is detected when: the number of nickel nearest neighbours is either unchanged or increased and either $\Delta R > 3.5$ Å or the target atom has no carbon n.n. or it is no more connected to any old carbon nearest neighbour;
3. A *Pre-dissolution* mechanism PD is detected when: the number of nickel nearest neighbours has increased and $\Delta R \leq 3.5$ Å and the target atom is connected to just one old carbon nearest neighbour;

This can be summarized by means of boolean variables (E, D, PD) as follows:

$$E = 1 \quad \textbf{if} \quad BO < 0.3 \tag{7.1}$$

$$D = 1 \quad \textbf{if} \quad (\Delta R > 3.5\text{Å} \ \textbf{or} \ C_{nn} = 0 \ \textbf{or} \ K = 0) \ \textbf{and} \ (\Delta Ni_{nn} >= 0) \tag{7.2}$$

$$PD = 1 \quad \textbf{if} \quad \Delta R \leq 3.5\text{Å} \ \textbf{and} \ C_{nn} = 1 \ \textbf{and} \ \Delta Ni_{nn} > 0 \tag{7.3}$$

[30] Moreover the modelling of this mechanism by means of the geometrical information that we can get from the SCM's output files it's cumbersome and could easily lead to a mistakes, i.e. processes which are none of the four aforementioned are recognized as *double atom pre-dissolution*, thus leading to a loss of computational time.
Since the probability of this event seemed very low, we decided to ignore this effect.

[31] Equation 2.8 defines the so called uncorrected bond order BO'_{ij}, whereas both in the potential energy terms and here, the corrected bond order BO_{ij} is employed. BO_{ij} bears some corrections in order to reduce possible over-coordinations and to properly consider lone pairs. The issue is detailed addresses in the work of *Adri C. T. van Duin, Siddharth Dasgupta, Francois Lorant and William A. Goddard* [29].

Once we determined the events that most likely would lead to a variation of the final geometry after the force biased Monte Carlo, and a way to automatically classify them we decided to gain some statistical information about them.
In order to do so, a set of 39 target atoms were selected as those carbons placed within a sphere of radius 7.5 Å from the center of mass of the nickel cluster. For each atom 200 collisions were simulated according to section 5.3.1 and with the velocities so generated an MD simulation of 2000 steps was started. Each final geometry was then used to classify the corresponding collision event by means of the conditions set by equations 7.1. The direction of the electron beam was set in order to displace atoms towards the nickel cluster. Results are reported in the following graphs 30 (a)-(d).
It can be noticed that all mechanisms are more efficient for low emission angles, and their number of counts decreases increasing γ. The *ejection* event has overall a very low chance of happening figure 30 (b), due to the directionality of the beam. A null probability is detected in the case of $\gamma = 0$ whereas some events are detected for slightly higher emission angles i.e. within 5° and 15°. For such angles there is the chance for the atoms of getting ejected from the side of the nanotube. Increasing the angle, the transmitted energy decreases, thus the probability for this mechanism drops down to zero again.
The *dissolution* Fig. 30 (c) event is in general more likely than the *pre-dissolution* event Fig. 30 (d), the latter one being dominant just for very large emission angles i.e. from ca. 40°. During the *pre-dissolution* event not all the bonds with the target's carbon nearest neighbours are broken as for the *dissolution*, the required energy for the former is therefore lower than the latter. Thus considering that the higher the emission angle, the lower the transmitted energy the trend of these two mechanisms seems physically reasonable.
Moreover, the *dissolution* event shows a local peak in the number of counts for $\gamma = 10°$. This might be caused by a lack of sufficient samples (even tough the total counts of such event is 1018). Nevertheless, we should not exclude another possible reason: the energy transferred for this angle seems to be still over the barrier for the dissolution (at least for some atoms), however there might be a larger amount of atoms for which the direction of emission towards the nickel is identified by a γ larger than 5 °. Therefore even if the energy transferred is lower, having a larger amount of possible targets increases the efficienty of this process.

The *dissolution* event is in general more likely to appear with a total probability of 0.1305 whereas the *pre-dissolution* event has just a total probability of 0.0643 to happen [32].
In figure 30 (a) the number of *other* events is reported. As it is visible increasing the emission angle the probability of recording an event which is not one of the three aforementioned increases, suggesting that most of these events are going to be just small displacements of the target atom (which therefore wouldn't be helpful for our goal). Nonetheless, even for null emission angle the probability of not having an *effective* event is the highest, suggesting that just the few carbons closer to nickel atoms are going to take part to these electron collision-assisted reactions.
Altogether the classification method seems to provide a very reasonable distribution of the dependencies of each effect from the emission angle [33]. Thus we concluded that for the pristine state of the system it's working correctly.

It is better to stress once again the fact that these conditions would represent the most probable knock-on mechanisms strictly for an unperturbed system. The true dynamics should take into account every single possible collision event and outcome of the short MD simulation. Nevertheless, it will be shown in the next chapter that even though this might seem a too simple approach, it still leads to very reasonable results at the advantage of overcoming computational time limitations.

[32]The probabilities are calculated with respect to the total of 7800 collisions analyzed.

[33]If some effect would have been misclassified, the dependencies of the probability as function of the angle wouldn't have any physical justification.

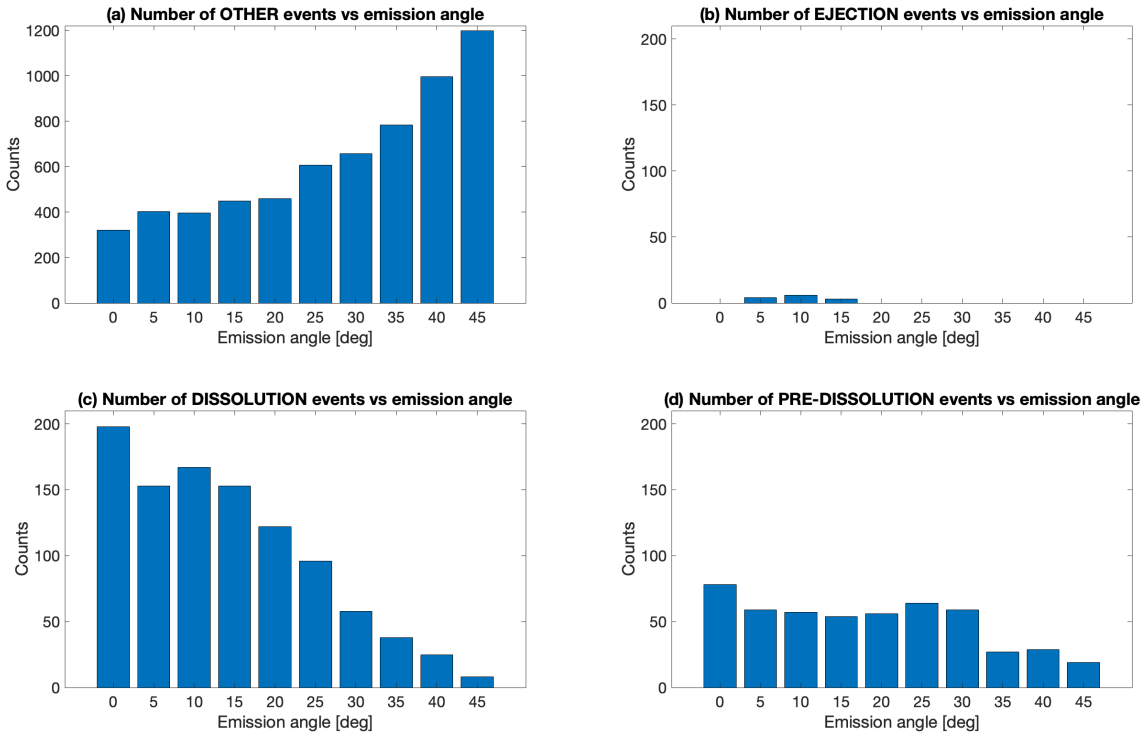

Figure 30: Barplots of the counts of the different effects as function of the emission angle γ for carbon displacements pointing towards the nickel cluster:
(a) Number of counts of *non-effective* events;
(b) Number of counts of *ejection* events;
(c) Number of counts of *dissolution* events;
(d) Number of counts of *pre-dissolution* events.
.

To recap, due to computational time issues we needed to implement a further acceptance routine for the collisions, based on classifying each event with respect to a set of known *effective* processes taking place in the pristine system. In practice: each time a collision event has to be simulated the code will follow section 5.3.1 in order to generate the set of initial velocities of the target particle, a 2000 MD steps simulation is then run and if one or more of the criteria in equations 7.1 is satisfied, then we consider that an effective collision has happened. Thus the final geometry and the velocities distribution are passed to the next MD section and then to the fbMC routine following the scheme depicted in section 7. If none of these requirements is met, then a new target atom is selected and the procedure is repeated for a maximum of 100 times.

If at this point the requirements are still not satisfied the simulation is stopped and an error message is produced.
The scheme of this algorithm is shown in figure 31.

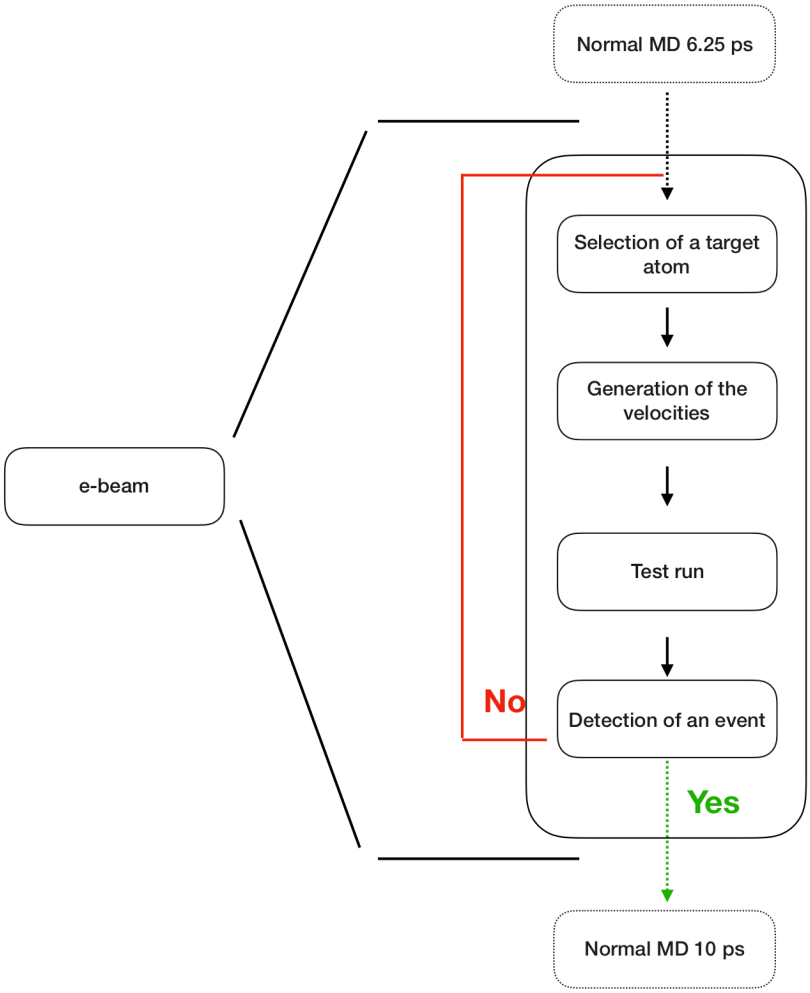

Figure 31: *Electron beam* building block: schematic representation of the algorithm to automatically evaluate the presence of *effective* collisional events.

8 Results and discussion

In this chapter, we will present and discuss the results obtained by our *ChemTEM* simulation code developed in the previous sections, when trying to replicate the experiments carried out by *Kecheng Cao et al.* [12] for a nickel cluster. All the calculations have been carried out starting from the reference geometry described in appendix A.
The first section is devoted to analyzing the dynamics of the reference system (as always described in appendix A) in the absence of any external excitation and will prove that no remarkable effect is visible. The next sections are going to be devoted to the analysis and discussion of the results obtained enabling the carbon atoms to interact with the electron beam. A detailed analysis of the mechanisms involved in the evolution of the system is going to be crucial to properly understand the phenomena involved in a *ChemTEM* experiment [17]. At the end of this chapter, we hope therefore to have assessed a way to properly simulate electron-assisted carbon restructuring effects in a TEM experiment.

8.1 CNT-Ni_{55} interaction in the absence of any external excitation

In order to properly understand the effect of the electronic beam, we first of all carried out a NVT molecular dynamics simulation on our reference system in absence of any external excitation, i.e. with no electronic collisions, so to have a comparison.
In order to test long time-scale effects, the classical setup $\Delta = 0.12$ Å, $T = 1200$ K and 3×150000 MC steps (sec.6.1) was used. 10 simulations were run to gain statistical relevance.
Each calculation was performed in a cubic box with the side of 300 Å.
Among all simulations nothing remarkable has happened.
The presence of the single vacancy defect in the nanotube wall, allowed the cluster to start the calculation already adsorbed (see appendix A). Thus no particular motion is registered throughout the whole simulation. This is consistent with most experimentally observed processes in which the nickel clusters are often observed to be already adsorbed on pre-existing defects in the nanotube sidewall at the start of TEM experiements (vacancy type defects are very common in SWNTs as defect-free nanotubes are virtually non-existent)[22]. Although as discussed in section 2.6.1, using a fbMC approach makes us lose a proper description of time, thanks to the detailed analysis on fbMC behaviour done in section 6.1, we are still quite confident in saying that that without any added external excitation no substantial interaction is taking place between the cluster and the nanotube for a sufficient amount of time.

8.2 Results in the presence of an electronic excitation

Due to the intrinsic bidimensional nature of TEM images, the position of the cluster within the nanotube is not fully recoverable. Nevertheless, results by means of this model, highly depend on the position of the NT wall in contact with the cluster and the directionality of the electron beam. Indeed, considering the reference system (figure 42), while electrons providing kinetic energy inwards with respect to the tube promote events of dissolution, electrons colliding in the opposite direction would promote sputtering of the carbon atoms from the surface, thus leading to a different evolution of the system, nevertheless having the same TEM image. As already discussed in chapter 4, a previous work of *Lebedeva et al.* [22] already addressed the mechanisms behind the cutting of a nanotube catalyzed by a nickel cluster under electron irradiation. Therefore since we are focusing our attention to the the processes related to the electron beam induced restructuring i.e. EBIR, we considered the direction of the beam such that the colliding electrons would promote displacement of the carbon atoms towards the nickel cluster.
Many simulations have been run following the scheme depicted in section 7, leading to different results.
Each calculation was performed in a cubic box with the side of 300 Å.
Due to computational time costs, we could analyze just the first stages of the interaction between the nanotube and the nickel cluster $\sim 50 - 70$ iterations i.e. effective events.
The important quantities that we analyzed were:

1. Number of carbon *monomers* and *dimers*;
2. Number of carbon atoms with more than one carbon nearest neighbour, in the following discussion called *structured* carbons;
3. The *total charge evolution* of the nickel cluster and of the carbon monomers and dimers;
4. The *average charge evolution* of the nickel cluster and of the carbon monomers and dimers.

A close examination of the evolution of all these quantities and of the geometry of the system allowed us to recognize four different distinct stages in the initial interaction between the nickel and the carbon nanotube under irradiation, namely:

1. Carbon monomers formation and rearrangement: formation of a carbide;
2. Aggregation of monomers into dissolved carbon dimers;
3. *Either* the dimers aggregate into trimers and quickly into pentagons or hexagons;
4. *Or* the carbon structures forming are broke apart and reabsorbed by the carbon nanotube.

8.3 First stage: Carbon monomers formation and rearrangement

The initial behaviour of the system is similar among all simulations. In these first $\sim 15 - 20$ iterations the collisions allow the atoms which are in contact with the nickel, to dissolve inside the cluster. This is clearly visible analyzing the evolution of the carbon monomers and of the *structured* carbon percentage in time, examples of these quantities are reported in figure 32(a) and figure 32(b) respectively. Indeed the growth of the number of carbon monomers tracks the decrease in the percentage of *structured* carbons in the system, pointing out that these carbons come from the shell of the nanotube.
It is moreover interesting to visualize the charge evolution of the nickel cluster within this time window figure 32(c). Indeed, as discussed in section 7.1.1 the dissolved carbon monomers tend to place themself in subsurface sites maximizing their coordination with nickel atoms. Therefore due to the higher electronegativity of the carbons with respect to the nickel, the monomers will be partially negatively charged wheres the nickel atoms will have on average a positive charge. Increasing the number of monomers, leads to an increase in the total charge of the nickel cluster. This is in agreement with previous results [91]. Moreover, the average charge of the monomers figure 32(d) stays almost constant throughout this period, pointing out that even increasing the number of dissolved atoms, their environment doesn't change much.
These two observations suggest that the dissolved monomers, partially negatively charged, repel each other, occupying all the available sub-surface sites within the cluster.
These intuitions are confirmed by the analysis of the geometry of the system within this time window figure 33 a) where it is visible the formation of the first defects in the side of the nanotube and the dissolution of the monomers. Moreover, the defects in the side of the tube are occupied by nickel atoms that stabilize them and often pop out from the nanotube wall figure 33 b), resembling the behaviour of nickel atoms adsorbed onto graphene vacancies defects [92], [99].
Although the system is utterly different from the free standing nickel particle structure usually considered for the growth of carbon nanotubes [94], [41],[42], [47], this initial stage behaviour is analogous for the two systems, resembling the so called *alloying stage* [47] believed to be the first step for the nucleation of a nanotube [93], [91].

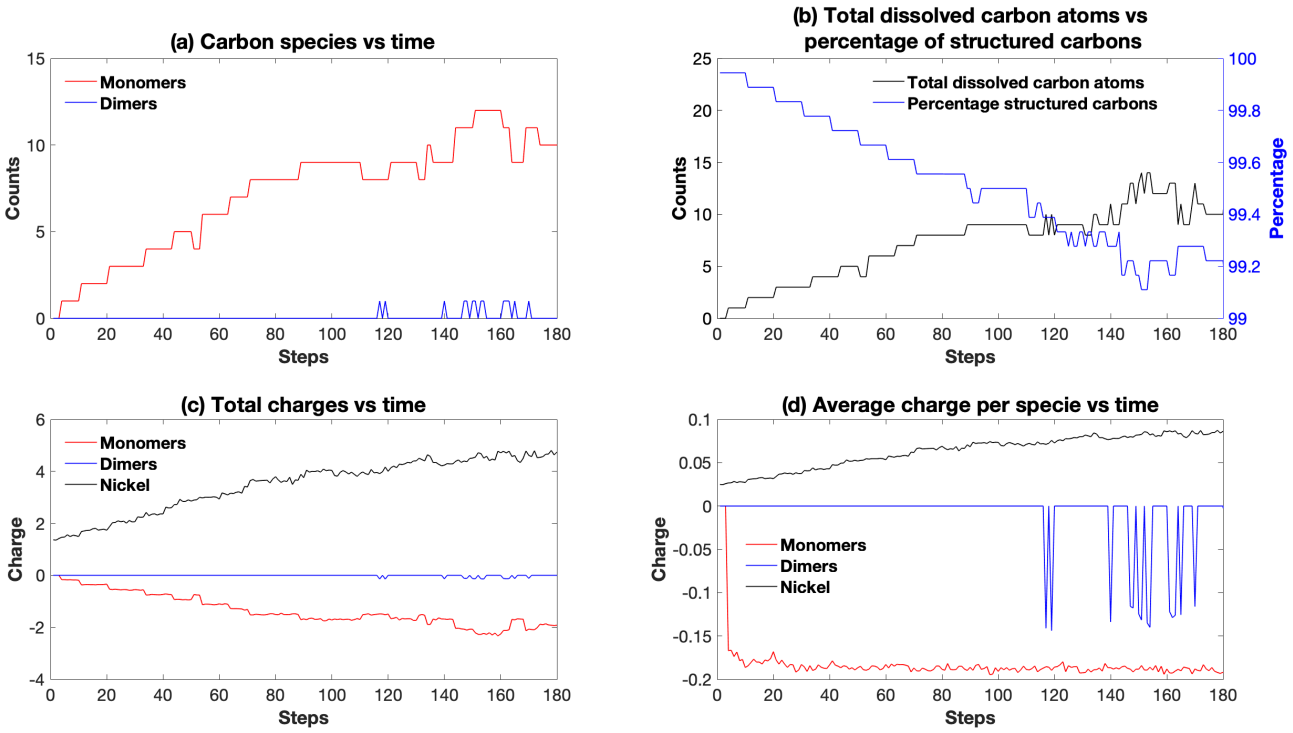

Figure 32: Example of the evolution during the first 18 iterations of the different quantities of interest.

(a) Carbon monomers and dimers as function of time.
(b) Total carbon atoms dissolved in the cluster compared with the percentage of *structured* carbon.
(c) Total charge of the different species as function of time.
(d) Average charge per species as function of time.

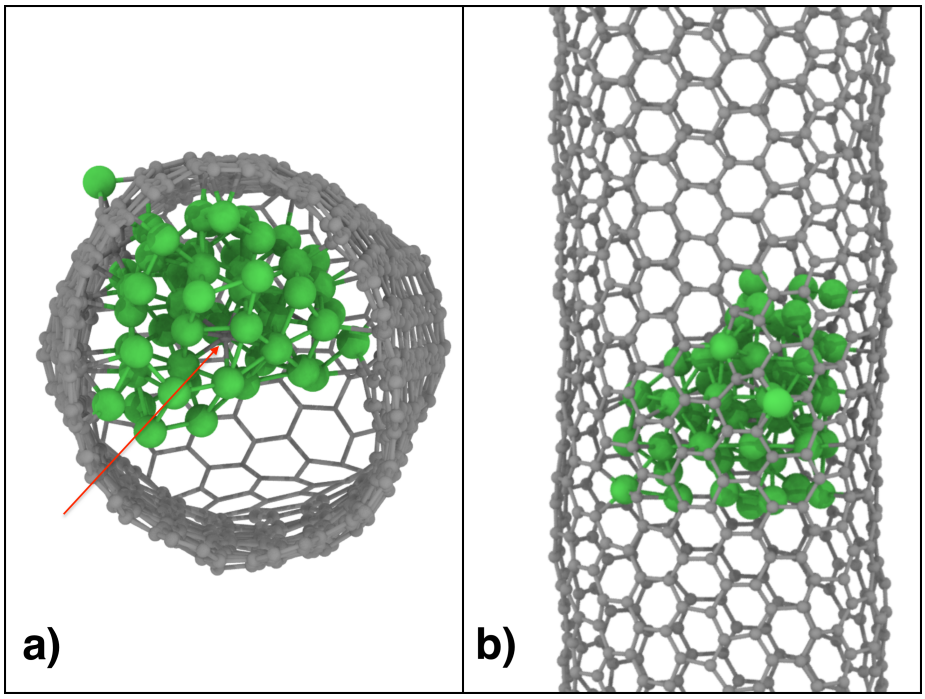

Figure 33: Snapshots of an example system geometry after one collisional event.

In snapshot **a)** a carbon monomer dissolved in the nickel cluster is visible (red arrow). The popping out of nickel atoms in nanotube defects seems to be supported by experimental data [20]. In this snapshot, one of the two caps has been cut for a better representation.
In snapshot **b)** is visible the top view of the same system. Two holes in the nanotube lattice are visible, one is caused by the dissolution of the monomer, the other one was present in every starting geometry to let the cluster attach to a specific point in the nanotube.
The stabilizing effect of the nickel atoms is visible: no reconstruction of the single vacancy defect is indeed taking place.

8.4 Aggregation of monomers into dimers

When the content of carbon monomers in the cluster reaches 20% − 25%, some monomers tend to aggregate into dimers. The first appearance of stable dimers is after ca. 20 iterations [34] and it's characterized by a sudden drop of the content of monomers figure 34(a) with very little changes in the percentage of *structured* carbons in the system. Thus the formation of this species is mostly due to the aggregation of monomers in the carbide. Also this stage finds its analogous for a free standing nickel particle, in the so called *supersaturation stage* [96], [97]. Moreover, the analysis of the average charge of a dimer figure 34(d) shows first of all that a dimer carries less negative charge than a monomer and secondly that this is more variable compared to the same quantity for the monomers. This suggests that the environment for the dimers might change significantly during the simulations. Indeed dimers segregate mainly at the surface of nickel figure 35 a) on which they can migrate from site to site. Nevertheless, due to the small size of the cluster [35] and the strong interaction with the carbon shell, the nickel atoms are highly mobile. Thus the environment for each dimer might change substantially during its migration leading to a more variable average charge with respect to the subsurface dissolved monomers.
In agreement with previous results for a free standing nickel cluster, dimers were observed both in subsuperficial and in superficial states [47].
Moreover, it is interesting to note that at this point a substantial amount of carbon atoms has been withdrawn from the outer carbon nanotube ca. 20 − 25 atoms. The nickel particle tends to bind very strongly to this defected area protruding outside the nanotube wall and resulting in a partial escape of the cluster from its initial endohedral confinement figure 35 b). A previous work of *Gao et al.* [99] proved the very high strength of NI-C σ-bonds, higher than Ni-Ni bonds or Ni-C π bonds and showed how nickel clusters would tend to grow symmetrically with respect to a large defect on a graphene sheet, suggesting that this would rationalize the experimental findings that Ni nanoparticles are very stable on the CNT supports [98].

[34] Often dimers also appear for fewer iterations. Nevertheless, their life is very short i.e. they quickly disappear during the fbMC routine in favour of monomers.

[35] This is due to the Gibbs Thomson effect [95] for which decreasing the particle size decreases the melting point according to:

$$\Delta T = \frac{2T_M\sigma_{SL}}{\rho_S \Delta H}\frac{1}{r}$$

Where T_M is the melting point of the bulk system, σ_{SL} is is the solid-liquid interface energy, ρ_S is the number density of the solid phase, ΔH is the latent heat of melting and r is the radius of the cluster.

Our results showing the nickel cluster's preference for symmetrical disposition with respect to the defected nanotube wall, cast light on the atomistic mechanisms of interaction between the host tube and the metal particle supporting the ideas of *Gao et al.* [99].
Eventually, it is important to note that carbon dimers are observed to generate both in the outer protruding side of the cluster and in the endohedrally confined half figure 35 c).

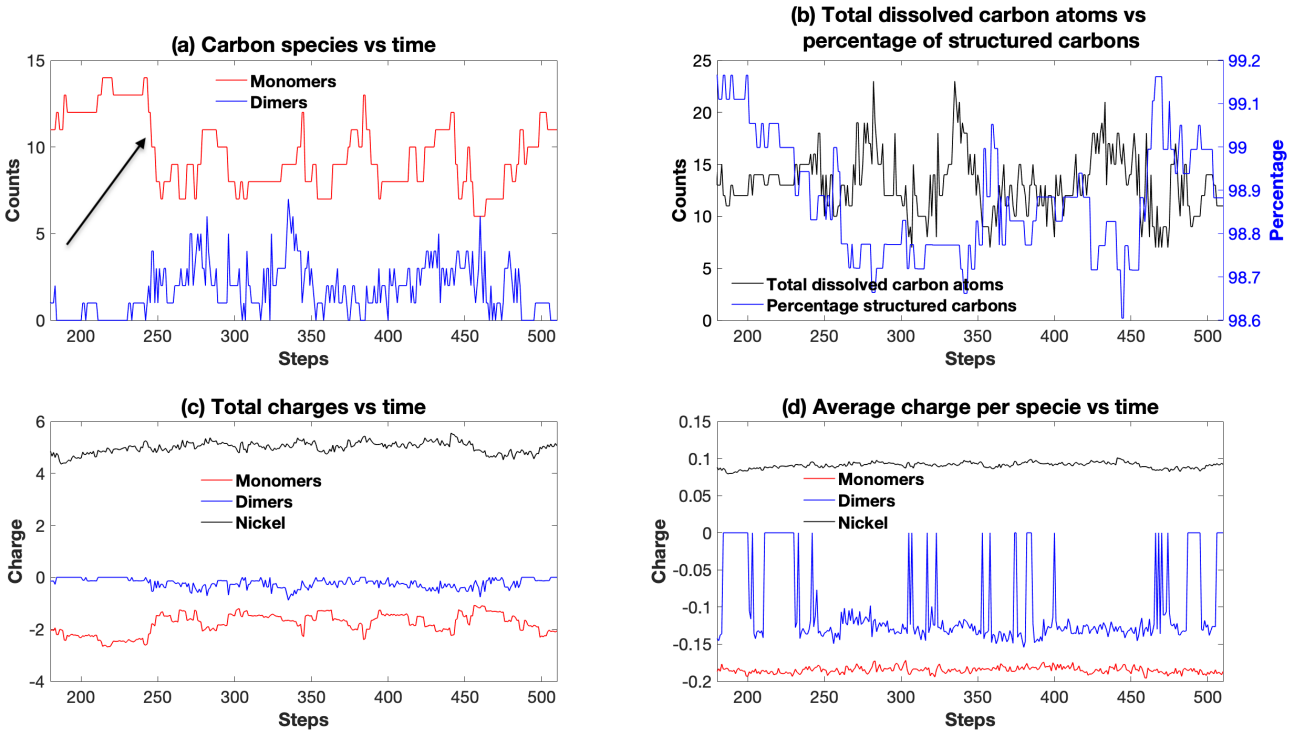

Figure 34: Example of the evolution during the supersaturation stage of the different quantities of interest.

(a) Carbon monomers and dimers as function of time. The arrow indicates the moment for which stable dimers start to form.
(b) Total carbon atoms dissolved in the cluster compared with the percentage of *structured* carbon. The moment in which monomers condense into dimers is related to a small decrease of the total carbon content and an increase of structured carbons. This might be due to the reattachment of some monomers to the nanotube lattice.
(c) Total charge of the different species as function of time.
(d) Average charge per species as function of time. The charge per dimer is visibly more changing than the charge per monomer.

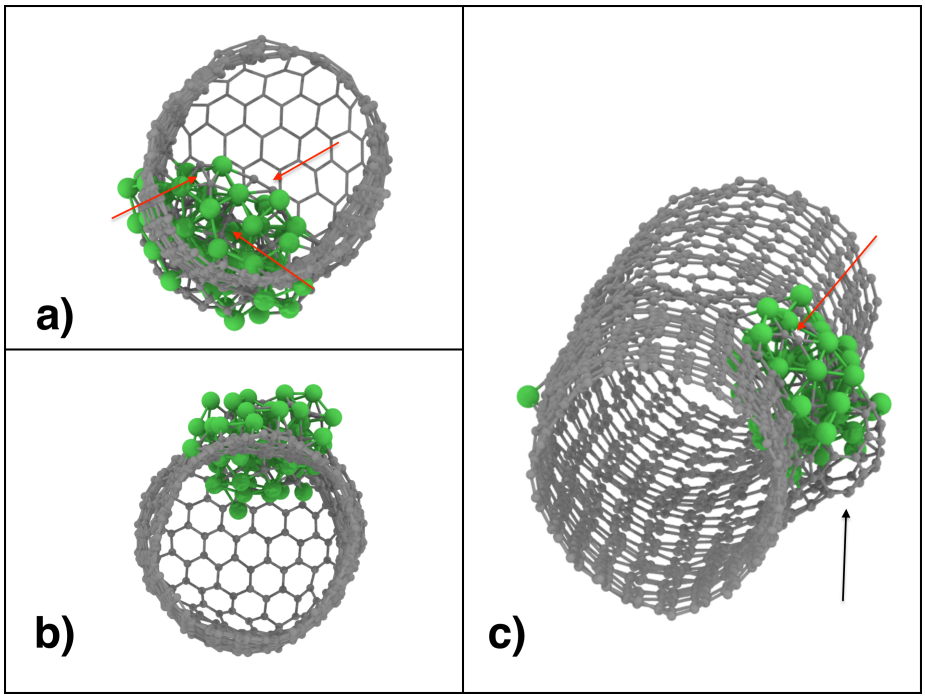

Figure 35: Snapshots of different simulations around the 30^{th} iteration.

In snapshot **a)** three dimers are visible (red arrows) of which two segregated on the surface whereas the third one is in a subsuperficial site.
In snapshot **b)** the partial escape of the nickel cluster from the endohedrally confined position is visible. The cluster is placed almost symmetrically with respect to the nanotube wall.
In snapshot **c)** highlighted by the red arrow is visible a dimer formed on the outer surface of the cluster. The partial protrusion of the nickel particle might cause the lift and modification of part of the host carbon nanotube shell (black arrow).

8.5 Behaviour of the system after many iterations

The main stages in the metal-catalyzed growth mechanisms of carbon nanotubes on a free standing metal particle by chemical vapour deposition, are believed to be the following [94],[42],[97],[47]:

1. *Alloying stage*: impinging gas phase carbon atoms easily adsorb on the surface of the particle and subsequently dissolve into the cluster;

2. *Supersaturation stage*: the dissolved carbon atoms start to form C-C bonds in the form of dimers, causing a drop in the atomic carbon content in the cluster;

3. *Ring formation*: the addition of single atoms to short carbon chains rearrange into polygonal structures, eventually leading to the formation of pentagons or hexagons;

4. *Graphitic islands formation*: formation of concatenated rings;

5. *Cap formation* and *lift off*: coalescence of many graphitic islands. Once a certain critical dimension is reached the so formed carbon cap is lifted off from the surface of the nickel cluster.

6. *Growth of the nanotube*: dissolved carbon atoms diffuse towards the metal-cap interface where they are incorporated in the side of the tube.

Up to this point, our system showed no substantial deviation from these stages.
So far indeed the confining carbon nanotube acted mainly as a reservoir of atoms that the electronic beam was dissolving with a constant rate in the cluster, allowing the observation of the *alloying* and *supersaturation* stages.
Nevertheless, after many electron collisions, the presence of the nanotube might lead to strong deviation from the free standing particle's behaviour.
Indeed as previously discussed, once the defect reaches a substantial dimension, it happens that the cluster starts protruding outside the nanotube wall.

The fraction of nickel atoms that remain endohedrally confined, seemed to determine two different evolution paths for the system:

1. If a small number of nickel atoms (e.g. less than 30 %) remain endohedrally confined, i.e. the protrusion is really evident, the excess carbon atoms dissolved in the cluster diffuse towards the nickel nanotube interface, reattaching to the NT's wall;

2. If a large fraction of nickel atoms (e.g. more than 70 %) remains endohedrally confined, the system evolves towards the formation of graphitic islands.

In the next subsections these two different evolution paths are going to be addressed and discussed in detail.

8.5.1 Case 1: small fraction of nickel atoms endohedrally confined

If the defect is very large, i.e. many carbons are randomly removed from the same area, the protrusion is prominent leaving a substantial number of nickel atoms unwrapped by the carbon shell an example in figure 36.
The tendency of the cluster of moving outwards might cause a lift of the encapsulating outer shell, generating an evident carbon protrusion on the side of the nanotube.
Moreover, when the nickels are partially escaped, some monomers and dimers dissolved on both the inner and outer side of the cluster might migrate towards the nickel carbon nanotube interface where they condense in the nanotube lattice. This causes the system to retain a constant percentage of *structured* carbons figure 37 (b). If polygonal structures were previously formed onto the surface of the cluster, it might happen that they rip apart into smaller chains or single carbons, rejoining the carbon nanotube an example in figure 38 a) − d).
Although at this point, the system dynamics seemed quite steady without any major feature in the graphs, due to limited computational resources, we couldn't span longer time, i.e. simulate more iterations. Thus this process might also be an intermediate step for some further and different stage that we could not observe.
Further studies should therefore address the behaviour of such partially escaped endohedrally confined nickel particles, after many electronic collisions i.e. ca. 100-120.

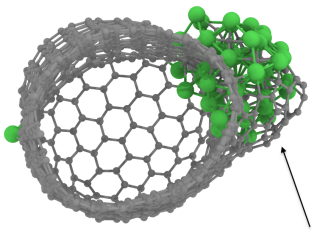

Figure 36: Example of a system in which the cluster is protruding very much outside of the nanotube walls. Moving outwards the cluster has lifted part of the nanotube walls (black arrow).

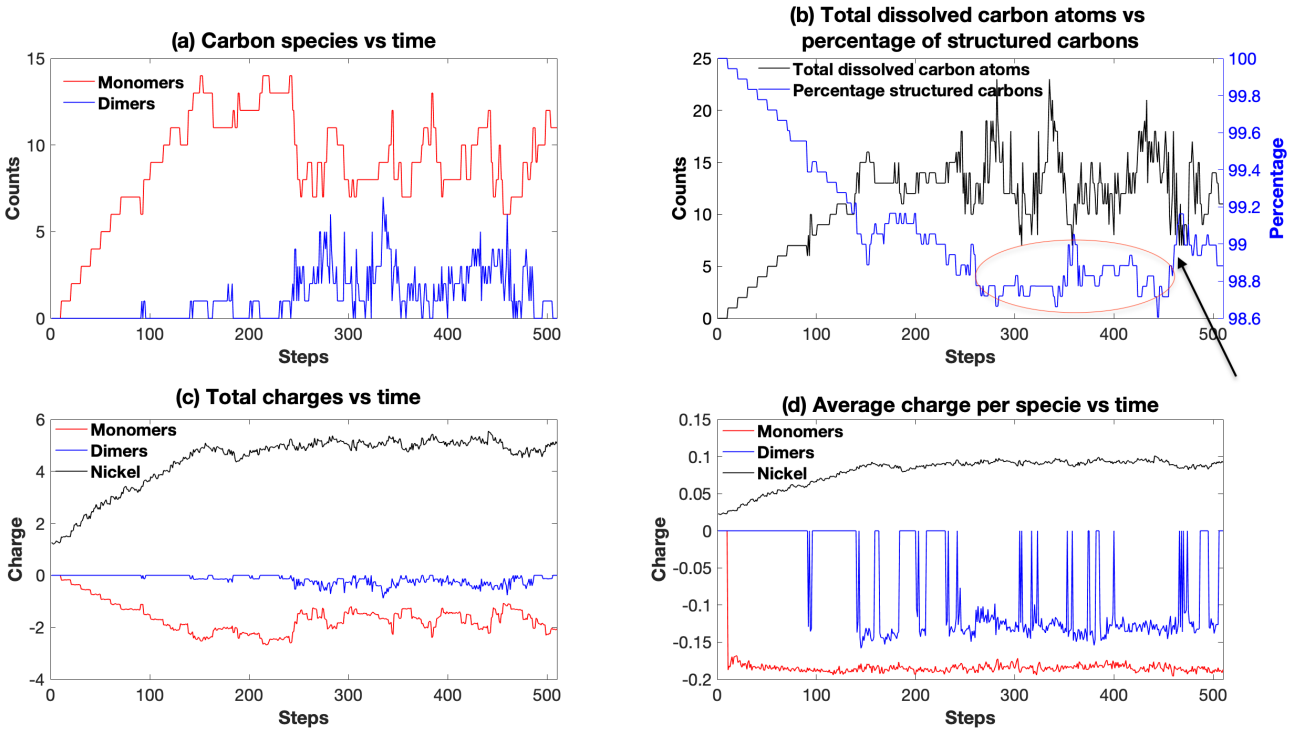

Figure 37: Example of the evolution of the when a small amount of atoms remains endohedrally confined.

(a) Carbon monomers and dimers as function of time.
(b) Total carbon atoms dissolved in the cluster compared with the percentage of *structured* carbon. Highlighted by the red ellipses the region after the formation of dimers in which the system retains ca. a constant amount of structured carbons. The black arrow shows the moment of reattachment of a broken carbon pentagon to the outer shell.
This process is shown in the next images Fig. 38 a)-d)
(c) Total charge of the different species as function of time.
(d) Average charge per species as function of time.

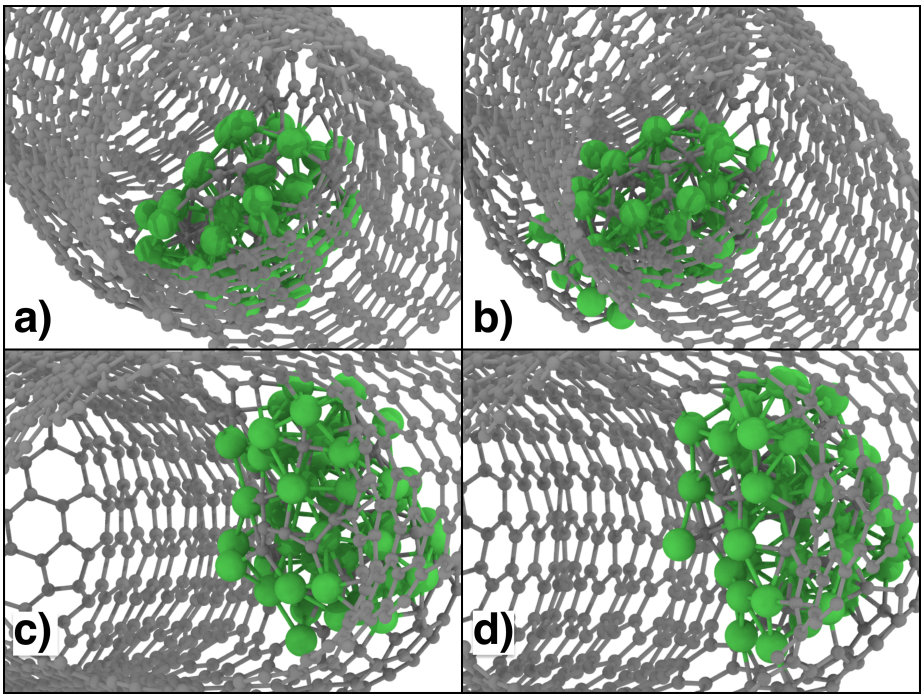

Figure 38: Snapshots of the geometry of the system described by the graphs in figure 37 for iterations between the 37^{th} and 50^{th}.
This is the case of a system with a large number of nickels escaping from the endohedral confinement.

In snapshot **a)** the presence of an endohedrally confined pentagon on the surface of the cluster is visible
In snapshot **b)** the pentagon is broken.
In snapshot **c)** the atoms that were composing the pentagon are now forming linear chains.
In snapshot **d)** these carbons are finally reattached to the side of the nanotube.
The NT throughout these steps rotates on his axis and the electron beam moves accordingly (see section 7.1).
One of the caps of the nanotube has been cut in order to have a better view on the effects taking place at the surface of the endohedrally confined cluster.

8.5.2 Case 2: large fraction of nickel atoms remain endohedrally confined

If the nickel cluster remains mostly confined in the nanotube (e.g. more than 70 % of the nickel atoms), i.e. the protrusion is not so evident, then the dynamics of the system seemed to be different from the previous case.
Indeed this geometry has two advantages for the formation of carbon structures onto the cluster: first of all the endohedral position of the nickels keep favouring the dissolution of carbon atoms from the outer shell, and secondly the segregated dimers are going to be inside the nanotube section, and their aggregation into rings is going to happen mostly away from the carbon shell of the confining NT, leading with high probability to the formation of graphitic islands.
Supporting these suppositions are the data that we collected
Fig.39(a)-(d).
Indeed when the concentration of total dissolved carbon atoms in the nickel cluster reaches ca. 37%, then the carbon dimers start to form longer chains and rings. This is characterized by a simultaneous drop of the total amount of dissolved carbons and an increase in the percentage of *structured* carbons Fig.39(b). Analysis of the number of monomers and dimers dissolved in the cluster show how carbon dimers content decreases whereas the carbon monomers don't show such an evident modification 39(a). Thus we can conclude that the formation of *structured* carbons comes mainly from the aggregation of carbon dimers. Figures 40 a) and b) show the snapshots of the system's geometry at the beginning and end of such a transformation, these points are marked on the graphs with two lines. The formation of a first graphitic island composed by a pentagon a hexagon and a heptagon is visible in figure 40 b). After this point if the electron beam continues to dissolve carbons inside the cluster, the graphitic island evolves towards a cap that eventually lifts-off from the nickel cluster 41 a) and b). The speed of formation of the graphitic island depends on the number of dissolution events happened during the simulations. Indeed if a large amounts of collisions lead to an ejection event, then the amount of carbons fed into the system is less, and the formation of dimers or long chains is delayed [36]. Although the structure of the system and the way of providing carbons is completely different, the stages that our system follows are extremely similar to those which a free standing nickel particle that catalyzes the growth of a nanotube from carbons in gas phase

[36] For example the simulation that leads to the graphs and pictures of this subsection has seen just two ejection events out of 70 collisions simulated. Similar results have been found for another simulation after ca. 100 collisional events, of which over 30 led to an ejection.

undergoes [94], [42], [97],[47].
This might be therefore seen as further proof of the goodness of previously developed atomistic models for the mechanisms of growth of nanotubes, moreover showing the potentiality of the *ChemTEM* [17] approach for a controlled modification of the specimen.

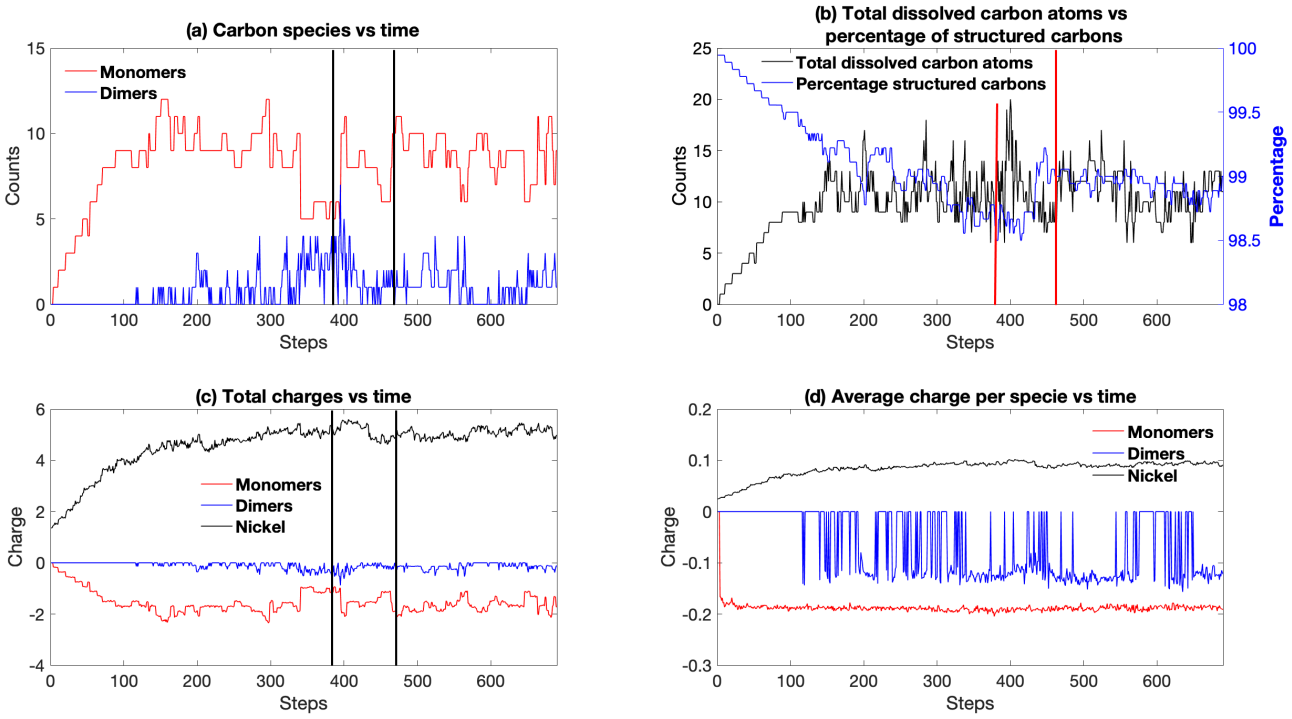

Figure 39: Example of the evolution of the system.
The two lines identify the points at which snapshots a) and b) of figure 40 are taken.

(a) Carbon monomers and dimers as function of time.
(b) Total carbon atoms dissolved in the cluster compared with the percentage of *structured* carbon. Between the two lines a simultaneous net decrease of the total amount of dissolved carbons and an increase of *structured* carbons are visible.
(c) Total charge of the different species as function of time.
(d) Average charge per species as function of time.

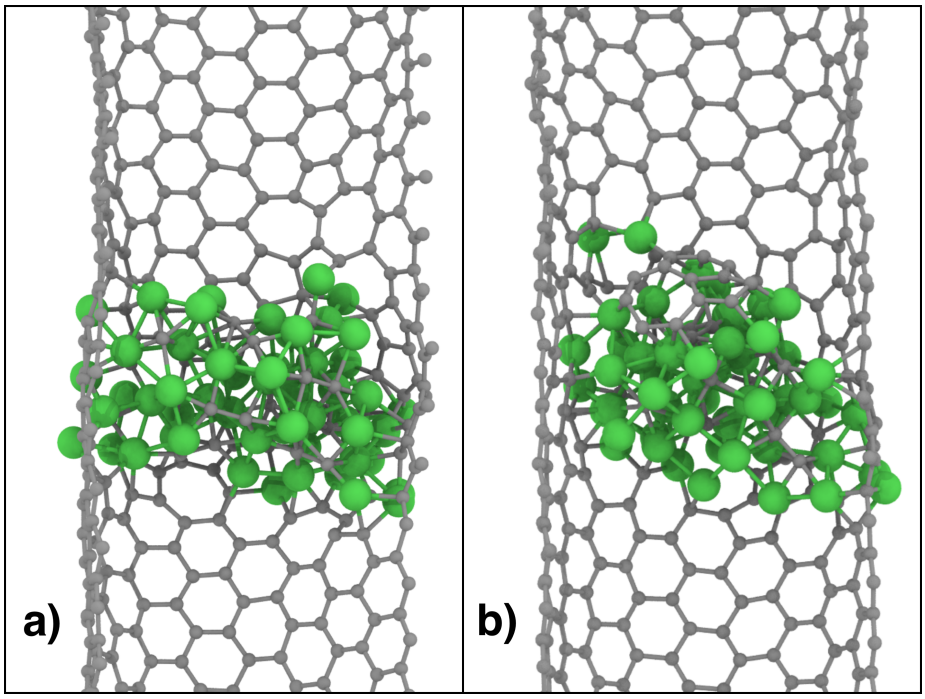

Figure 40: Snapshots of the geometry of the system described by the graphs in figure 39 at iteration 37 and 48 i.e. at the beginning and end of the graphitic island nucleation process.

In snapshot **a)** the presence of a high number of surface segregated dimers is visible.
In snapshot **b)** it is visible the formation of a graphitic island composed by a pentagon, a hexagon and a heptagon.
The upper section of the nanotube has been cut in order to have a better view of the effects taking place at the surface of the endohedrally confined cluster.

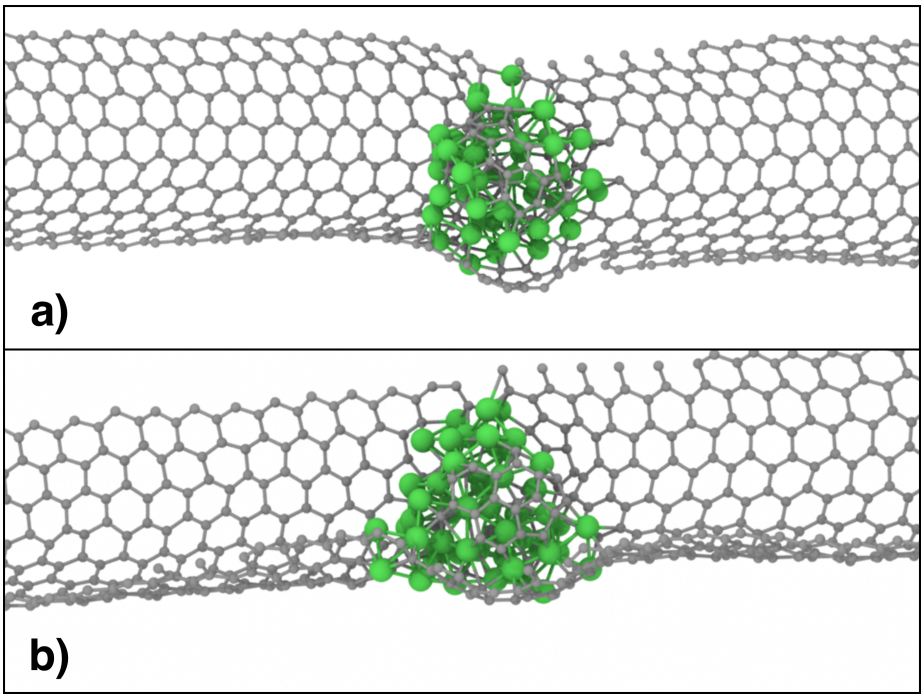

Figure 41: Snapshots of the geometry of the system shown in figure 40 after 58 and 70 iterations **a)** and **b)** respectively.

In snapshot **a)** a large graphitic island of three pentagons and two hexagons is visible.
In snapshot **b)** the graphitic island has rearranged forming a carbon cap of three hexagons and two pentagons. The cap is lifted from the surface of the cluster.
The upper section of the nanotube has been cut in order to have a better view of the effects taking place at the surface of the endohedrally confined cluster.

9 Conclusions

This thesis work focused its attention on the development of a multiscale theoretical approach to simulate the electron assisted metal-nanotube interaction. The goal was to cast light on the atomistic mechanisms at the basis of the nucleation of sp^2 hybridized carbons at the metal surface of an endohedrally confined nickel cluster in a SWNT. The simulation of the dynamics of the nuclei was done by means of reactive force field molecular dynamics.
The multiscale nature of our approach relies in splitting the dynamics of the system under electron irradiation as: an *electronic collision* part, short in time and simulated by means of molecular dynamics; and a *relaxation* part, which spans a longer time and it's simulated by means of the force biased Monte Carlo method.
The theoretical modelling of the physics of the electronic collisions was crucial in this dissertation and allowed us to identify three fundamental mechanisms for an irreversible knock-on displacement of a carbon atom in the system of interest, namely: *ejection*, *predissolution* and *dissolution* event.
Considering these three fundamental mechanisms and taking into account the thermalization of the system after the electronic collision (the *relaxation* part), we implemented on *Python* an atomistic *ChemTEM* simulation tool.
By means of such in-house code, we were able to identify four main stages in the initial dynamics of interaction between the host nanotube and the nickel cluster. The first two stages are the so called *alloying* and *supersaturation* stages for which the electronic collisions cause the carbon atoms to dissolve inside the nickel cluster, forming monomers and subsequently dimers respectively. These stages are also common to the well known free standing nickel particle that catalyses the growth of a CNT [94]. The presence of the encapsulating nanotube becomes critical once the defect in the side of the NT is large. Indeed the cluster binds very strongly to the vacancies in the CNT wall, partially protruding outside of it when the defect reaches a certain dimension. Depending on the number of nickels that remain trapped in the section of the NT, two different effects were observed to be promoted. If a small fraction of nickel atoms remained confined (e.g. less than 30 %) then the excess carbon monomers and dimers dissolved in the cluster reattach to the nanotube wall. If a large fraction of nickels remains endohedrally confined (e.g. less than 70 %), then the surface-segregated carbon dimers aggregate, first forming trimers, pentagons and graphitic islands all the way up to a nanotube cap. This is considered to be

the last stage before the nanotube starts growing [47].
Due to computational time constraints we could not sample further the dynamics of the system. The development of less computational expensive acceleration algorithms or a better parallelization of the force biased Monte Carlo algorithm might help in the future reaching further stages of the evolution of such systems.

In conclusion, thanks to the in-house developed software, we were able to simulate and explain the initial processes of growth of a carbon nanotube onto an endohedrally confined nickel nanocluster under electron irradiation.

Appendices

A Reference system for the simulated *ChemTEM* experiments

The reference system we used for all the simulated *ChemTEM* calculations is the following: a capped (11,11) carbon nanotube consisting in 1800 carbon atoms on which a single vacancy was induced by removing an atom approximately from the middle point of the tube; a 55 atoms nickel cluster endohedrally confined and placed in contact with the nanotube walls in the proximity of the vacancy defect. The nanotube was built with the "Atomic Simulation Environment" [70],[71] (version 3.17.0) using the package "NanoCap" [72] (version 1.0b15). The vacancy is induced in the side of the nanotube in order to let the cluster adsorb on such defect. This is consistent with most experimentally observed processes in which the nickel clusters are often observed to be already adsorbed on pre-existing defects in the nanotube sidewall at the start of the cutting transformation (vacancy type defects are very common in SWNTs as defect-free nanotubes are virtually non-existent) [22].

Each calculation on this system was performed in a cubic box with the side of 300 Å.

Once the system has been built, a 300 K molecular dynamic simulation of 100000 steps has been carried out in order to equilibrate the system. This relaxed geometry was then used as input for all the calculations during simulated *ChemTEM* experiments. Images of the reference system are reported in the following figures 42 a)-c).

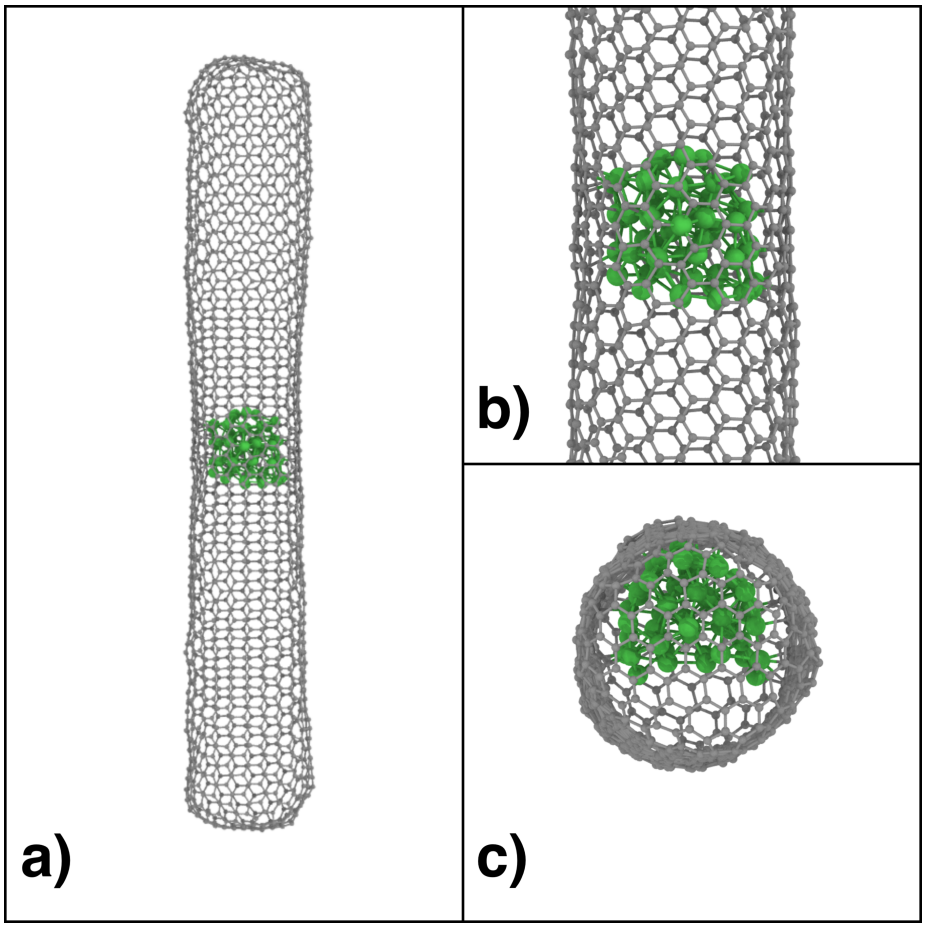

Figure 42: Reference system for the simulated *ChemTEM* experiments seen from different viewports.

b) zoom on the pre-induced vacancy defect.

B Derivation of the fbMC probability distribution

The derivation that follows is taken from the work of *Maarten J. Mees, Geoffrey Pourtois, Erik C. Neyts, Barend J. Thijsse and Andre Stesmans*[44]. In a canonical ensemble at temperature T, containing N particles and with a Hamiltonian H, the probability density function ρ that describes the system is given by:

$$\rho = \frac{exp(-\beta H)}{\int exp(-\beta H)\, d\Gamma} \tag{B.1}$$

with Γ representing the phase space in which all the possible states of particles are described and $\beta = (k_B \mathrm{T})^{-1}$.
The Hamiltonian of the N interacting particles associated with mass m_i can take the form

$$H = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} + U(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) \tag{B.2}$$

where $\mathbf{p}_i$ is the momentum vector of the particle i and the potential U accounts for both the particle's interaction and the presence of a possible external applied potential. During its evolution the system can adopt a state associated with a different set of coordinates such as $\mathbf{x}^* = (\mathbf{x}_1^*, \mathbf{x}_2^*, ..., \mathbf{x}_N^*)$, where a reformulation of U based on a Taylor expansion in the neighbourhood of this state leads to

$$\begin{aligned} U(\mathbf{x}) &= U(\mathbf{x}^*) + (dU)(\mathbf{x}^*) \cdot (\mathbf{x} - \mathbf{x}^*)^T + \mathcal{O}(\mathbf{x}^2) \\ &= U(\mathbf{x}^*) - \sum_{i=1}^{N} \mathbf{F}_i \cdot (\mathbf{x} - \mathbf{x}^*) + \mathcal{O}(\mathbf{x}^2) \end{aligned} \tag{B.3}$$

with $\mathbf{F}_i$ being the force acting on the particle i. If the Taylor expansion is truncated to its first-order term, meaning that $\mathcal{O}(\mathbf{x}^2)$ is negligible for all $x_{i,j} \in [x_{i,j}^* - \Delta/2\ ,\ x_{i,j}^* + \Delta/2]$ with Δ being the range of atomic displacement for which the expansion is valid and j being the x, y and z directions, then the Hamiltonian in the neighbourhood of $\mathbf{x}^*$ takes the form:

$$\tilde{H} = U(\mathbf{x}^*) + \sum_{i=1}^{N} \sum_{j=1}^{3} \left(\frac{p_{i,j}^2}{2m_i} - F_{i,j}(x_{i,j} - x_{i,j}^*) \right) \tag{B.4}$$

By combining equations B.1 and B.4 the probability density function $\tilde{\rho}$ in coordination space in the neighbourhood of $\mathbf{x}^*$ takes the form:

$$\tilde{\rho} = exp\big[-\beta U(\mathbf{x}^*)\big] \frac{\prod_{i,j} exp\big[\beta F_{i,j}(x_{i,j} - x_{i,j}^*)\big]}{\int_\nu exp\big[-\beta U(\mathbf{x})\big] d\mathbf{x}} \tag{B.5}$$

Equation B.5 will be used to derive the necessary probability function used to drive the fbMC algorithm[37]. Before going into the details of this derivation, it is useful to gain some insights about the MC-algorithm mechanics. We assume that at a given time the particles are located in the (small) coordinate space intervals $\left([x^*_{i,j} - \Delta/2\ ,\ x^*_{i,j} + \Delta/2]\right)_{i,j}$ which we label as a volume ν_i given by:

$$\nu_i = \bigcup_{i,j} [x^*_{i,j} - \Delta/2\ ,\ x^*_{i,j} + \Delta/2] \tag{B.6}$$

Note that this volume corresponds to the one described by ρ. Suppose now that, during a short but unspecified time, the particles are allowed to move in this volume ν_i. The question arises of being able to identify the probability that, during this time interval, the particles move to another volume ν_f which has the same form ν_i, but which is shifted with respect to $\mathbf{x}^*$. Assuming that this probability can be quantified, all the necessary ingredients to use a MC algorithm would be present to choose the next volume ν_f. The latter will become the new initial volume ν_i bound to a new coordinate x^*. Upon the repetition of the procedure, successive volumes are generated in which all particles are captured and the dynamics of the system is described. For such an approach to be operational, one naturally needs to develop an explicit formulation for the probability of going from ν_i to ν_f. It turns out that the concept of conditional probability is perfectly suited for this task. The conditional probability of going for ν_i to ν_f is defined as:

$$P(state\nu_f | state\nu_i) = \frac{P((\mathbf{x} \in \nu_i) \cap (\mathbf{x} \in \nu_f))}{P(\mathbf{x} \in \nu_i)} \tag{B.7}$$

Using to our advantage the fact that $P(\mathbf{x} \in A) = \int_A \rho(\mathbf{x}) d\mathbf{x}$ and that $\rho = \tilde{\rho}$ on ν_i, the conditional probability can be reformulated as:

$$\begin{aligned} P(state\nu_f | state\nu_i) &= \frac{\int_{\nu_i \cap \nu_f} \rho(\mathbf{x}) d\mathbf{x}}{\int_{\nu_i} \rho(\mathbf{x}) d\mathbf{x}} \\ &= \frac{\int_{\nu_i \cap \nu_f} \tilde{\rho}(\mathbf{x}) d\mathbf{x}}{\int_{\nu_i} \tilde{\rho}(\mathbf{x}) d\mathbf{x}} \\ &= \frac{\prod_{i,j} \int_{\nu_i \cap \nu_f} e^{\beta F_{i,j}(x_{i,j} - x^*_{i,j})} d\mathbf{x}}{\prod_{i,j} \int_{\nu_i} e^{\beta F_{i,j}(x_{i,j} - x^*_{i,j})} d\mathbf{x}} \end{aligned} \tag{B.8}$$

[37] In the original paper from *Maarten J. Mees et al.* [44] this method is referred as time stamped force biased Monte Carlo or tfMC.

In order to obtain an approximate timescale, Δ need to have a mass dependence, therefore we call $\Delta_i = \Delta\sqrt{\frac{m_{min}}{m_i}}$ the normalized maximum atomic displacement for each particle i with $m_{min} = min\{m_i | i = 1, 2, .., N\}$. For notational reasons we define $\gamma_{i,j}$ as

$$\gamma_{i,j} = \frac{F_{i,j}\Delta_i}{2k_BT} \tag{B.9}$$

and write the coordinate $\mathbf{x}_f^*$ of the final volume on the basis of its initial coordinate $\mathbf{x}_i^*$ through a translation vector $\boldsymbol{\Delta\xi}$ as:

$$\mathbf{x}_f^* = \mathbf{x}_i^* + \boldsymbol{\Delta\xi} \tag{B.10}$$

and with $\boldsymbol{\xi}$ being a vector defined as $\boldsymbol{\xi} = (\xi_{1,1}, \xi_{1,2}, ..., \xi_{N,3})$, where each component $\xi_{i,j}$ takes values in $[-1, 1]$ and with $\boldsymbol{\Delta} = (\Delta_1, \Delta_2, .., \Delta_N)$. Equations B.9 and B.10 allow us to rewrite eq. B.8 as $P(state\nu_f | state\nu_i) = \prod_{i,j} P_{i,j}(\xi_{i,j})$ with $P_{i,j}(\xi_{i,j})$ given by:

$$P(\xi_{i,j}) = \begin{cases} \frac{e^{\gamma_{i,j}(2\xi_{i,j}+1)} - e^{-\gamma_{i,j}}}{e^{\gamma_{i,j}} - e^{-\gamma_{i,j}}} & if \;\; \xi_{i,j} \in [-1, 0[\\ \frac{e^{\gamma_{i,j}} - e^{\gamma_{i,j}(2\xi_{i,j}-1)}}{e^{\gamma_{i,j}} - e^{-\gamma_{i,j}}} & if \;\; \xi_{i,j} \in]0, 1] \end{cases} \tag{B.11}$$

Equation B.11 defines the location of the particles in the coordination space throughout the iterations.

C In-house *ChemTEM* simulation tool: complete *Python* script

In this appendix, the complete code developed in chapters 5 and 7 for the simulation of a *ChemTEM* experiment is reported. The images are snapshots taken from "TextWrangler" [100] code visualization tool.

~/Desktop/GS_experiment_in.py

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Tue Nov 12 15:09:05 2019

@author: Luca Nicoli
"""

import os, subprocess
from shutil import copyfile, rmtree
import math
import random
import numpy

################################################################################################
#SETUP PHASE

#definition of the names of the folders
home = os.getcwd()
runfolder1 = "setup_files/step_1/runfolder/"
runfolder2 = "setup_files/step_2/runfolder/"
runfolder3 = "setup_files/step_3/runfolder/"
runfolder4 = "setup_files/step_4/runfolder/"
results1 = "results/temp_current_iteration/step_1/"
results2 = "results/temp_current_iteration/step_2/"
results3 = "results/temp_current_iteration/step_3/"
results4 = "results/temp_current_iteration/step_4/"

general_run = "cycle1_"

#number of times the simulation is repeated
N = 70

#radius for the selection of the target atoms
r_cutoff_tregion = 11

past_events_list = []

#list of the atoms required to set the fictitious basis of the space
reference_atoms = [274,277,252,253,262,265,287,288]

#END SETUP PHASE
################################################################################################

os.chdir(home + "/results/")
for h in range(0,N):

################################################
#Creation of the current iteration results folder
    #if not os.path.isdir('temp_current_iteration'):
    os.mkdir("temp_current_iteration")
    os.chdir("temp_current_iteration/")
    os.mkdir("step_1")
    os.mkdir("step_2")
    os.mkdir("step_3")
    os.mkdir("step_4")
    os.chdir(home)
```

```
################################################################################################
# STEP 1

    #Normal MD simulation
    os.chdir(runfolder1)
    subprocess.call("reaxff", shell=True)
    os.remove("geo")

# END OF STEP 1
################################################################################################

################################################################################################
# PROCEDURE TO GENERATE THE LIST OF TARGET ATOMS

    xyz = open("fort.90", "r")
    n = 0
    X_cm = 0
    Y_cm = 0
    Z_cm = 0

    #get the number of atoms in the system from molfra file
    molfra = open("molfra.out", "r")
    lines_molfra = molfra.readlines()
    atoms = int(lines_molfra[len(lines_molfra)-2].split()[4])
    molfra.close()

    lines = xyz.readlines()
    linee = lines[5:atoms+5]

    #Find the center of mass of the cluster
    for line in linee:
        if (line.split()[2] == 'Ni'):
            n = n + 1
            X_cm = X_cm + float(line.split()[3])
            Y_cm = Y_cm + float(line.split()[4])
            Z_cm = Z_cm + float(line.split()[5])

    X_cm = X_cm/n
    Y_cm = Y_cm/n
    Z_cm = Z_cm/n

    N_region = 0
    NOT_region = 0
    Not_region_atoms = []
    Region_atoms =[]
    R = []
    VR = []

    #Get the last molsav of the former simulation
    molsavList = [i for i in os.listdir() if i.startswith('molsav-')]
    molsavList.sort()
    molsav_name = str(molsavList[-1])

    molsav = open(molsav_name, "r")
    lineeMolsav = molsav.readlines()
    r = 0
    position = 0
```

~/Desktop/GS_experiment_in.py

```
116
117     #Based on the distance from the nickel cluster:
118     # 1)Set a list of target atoms, namely Region_atoms;
119     # 2)Set a list of the other atoms, namely Not_region_atoms;
120     # 3)Save in R the position of the atoms
121     # 4)Save in VR the velocities of the Region_Atoms
122
123     for line in linee:
124         lineVelocities = lineeMolsav[int(line.split()[1]) + 3 + atoms + 1]
125         lineAccelerations = lineeMolsav[int(line.split()[1]) + 3 + atoms + 1 + atoms + 1]
126         R.insert(position, line)
127         position = position + 1
128
129         if (line.split()[2] =='C'):
130             r = math.sqrt(((float(line.split()[3]) - X_cm)** 2) + ...
131             ((float(line.split()[4]) - Y_cm) ** 2) + ...
132             ((float(line.split()[5]) - Z_cm) ** 2))
133             if (r <= r_cutoff_tregion):
134                 Region_atoms.insert(N_region, int(line.split()[1]))
135                 #occhio al numero di linee nei due files
136                 VR.insert(N_region, lineVelocities)
137                 N_region = N_region + 1
138             else:
139                 Not_region_atoms.insert(NOT_region, int(line.split()[1]))
140                 NOT_region = NOT_region + 1
141
142         elif (line.split()[2] =='Ni'):
143             Not_region_atoms.insert(NOT_region, int(line.split()[1]))
144             NOT_region = NOT_region + 1
145
146     xyz.close()
147     molsav.close()
148
149 # END OF PROCEDURE TO GENERATE THE LIST OF TARGET ATOMS
150 ##############################################################################################
151
152
153     #copying the results in the results folder
154     os.chdir(home)
155     #New_path = copyfile(runfolder1 + "xmolout", results1 + "xmolout")
156     New_path = copyfile(runfolder1 + "thermolog", results1 + "thermolog")
157     New_path = copyfile(runfolder1 + "summary.txt", results1  + "summary.txt")
158     New_path = copyfile(runfolder1 + "reaxout.kf", results1  + "reaxout.kf")
159     New_path = copyfile(runfolder1 + molsav_name, results1 + "molsav")
160
161     #copying the files in the next step's runfolder
162     New_path = copyfile(runfolder1 + "fort.90", runfolder2 + "geo")
163
164     #initialization of the current folder for the next iteration
165     os.chdir("setup_files/step_1/")
166     rmtree("runfolder/")
167     os.mkdir("runfolder")
168     New_path = copyfile("control","runfolder/control")
169     New_path = copyfile("ffield","runfolder/ffield" )
170     os.chdir(home)
171
172 ##############################################################################################
```

```
172 ################################################################################################
173 # STEP 2
174 # COLLISION EVENT SIMULATION
175
176     os.chdir(runfolder2)
177
178     Max_energy = 21 # Maximum transferred energy
179     past_events_list = []
180
181     ################################################################################################
182     #PROCEDURE TO DEFINE THE FICTITIOUS BASIS OF THE SPACE
183     #This basis is going to be normalized but not necessarily orthogonal
184     geo = open("geo", "r")
185     lines_geo = geo.readlines()
186
187     #From the average position of the reference atoms I get the position of my reference
188
189     #Dz vector:
190     Dzx = (float(lines_geo[int(reference_atoms[0]) + 3].split()[3]) + ...
191     float(lines_geo[int(reference_atoms[1]) + 3].split()[3]))/2
192     Dzx = Dzx - (float(lines_geo[int(reference_atoms[2]) + 3].split()[3]) + ...
193     float(lines_geo[int(reference_atoms[3]) + 3].split()[3]))/2
194
195     Dzy = (float(lines_geo[int(reference_atoms[0]) + 3].split()[4]) + ...
196     float(lines_geo[int(reference_atoms[1]) + 3].split()[4]))/2
197     Dzy = Dzy - (float(lines_geo[int(reference_atoms[2]) + 3].split()[4]) + ...
198     float(lines_geo[int(reference_atoms[3]) + 3].split()[4]))/2
199
200     Dzz = (float(lines_geo[int(reference_atoms[0]) + 3].split()[5]) + ...
201     float(lines_geo[int(reference_atoms[1]) + 3].split()[5]))/2
202     Dzz = Dzz - (float(lines_geo[int(reference_atoms[2]) + 3].split()[5]) +...
203      float(lines_geo[int(reference_atoms[3]) + 3].split()[5]))/2
204
205     DZ = math.sqrt((Dzx)**2 + (Dzy)**2 + (Dzz)**2)
206     #normalization
207     Dzx = Dzx/DZ
208     Dzy = Dzy/DZ
209     Dzz = Dzz/DZ
210
211     #Uy vector:
212     Uyx = (float(lines_geo[int(reference_atoms[4]) + 3].split()[3]) + ...
213     float(lines_geo[int(reference_atoms[5]) + 3].split()[3]))/2
214     Uyx = Uyx - (float(lines_geo[int(reference_atoms[6]) + 3].split()[3]) + ...
215     float(lines_geo[int(reference_atoms[7]) + 3].split()[3]))/2
216
217     Uyy = (float(lines_geo[int(reference_atoms[4]) + 3].split()[4]) + ...
218     float(lines_geo[int(reference_atoms[5]) + 3].split()[4]))/2
219     Uyy = Uyy - (float(lines_geo[int(reference_atoms[6]) + 3].split()[4]) + ...
220     float(lines_geo[int(reference_atoms[7]) + 3].split()[4]))/2
221
222     Uyz = (float(lines_geo[int(reference_atoms[4]) + 3].split()[5]) + ...
223     float(lines_geo[int(reference_atoms[5]) + 3].split()[5]))/2
224     Uyz = Uyz - (float(lines_geo[int(reference_atoms[6]) + 3].split()[5]) + ...
225      float(lines_geo[int(reference_atoms[7]) + 3].split()[5]))/2
226
227     UY = math.sqrt((Uyx)**2 + (Uyy)**2 + (Uyz)**2)
228     #normalization
```

```
    #normalization
    Uyx = Uyx/UY
    Uyy = Uyy/UY
    Uyz = Uyz/UY

    #Grahm-Schmidt orthogonalization of Uy onto Dz generating Dy
    g = (Dzx*Uyx + Dzy*Uyy + Dzz*Uyz)/math.sqrt((Dzx)**2 + (Dzy)**2 + (Dzz)**2)
    Dyx = Uyx - g*Dzx
    Dyy = Uyy - g*Dzy
    Dyz = Uyz - g*Dzz
    DY = math.sqrt((Dyx)**2 + (Dyy)**2 + (Dyz)**2)
    #normalization
    Dyx = Dyx/DY
    Dyy = Dyy/DY
    Dyz = Dyz/DY

    #Dx vector orthogonal to the others:
    Dxx = (Dzy*Dyz)-(Dzz*Dyy)
    Dxy = -1*((Dzx*Dyz)-(Dzz*Dyx))
    Dxz = (Dzx*Dyy)-(Dyx*Dzy)
    DX = math.sqrt((Dxx)**2 + (Dxy)**2 + (Dxz)**2)
    #normalization
    Dxx = Dxx/DX
    Dxy = Dxy/DX
    Dxz = Dxz/DX

    #Transformation matrix from cartesian to {D} basis coordinates
    M_x_D = numpy.array([[Dxx, Dxy, Dxz], [Dyx, Dyy, Dyz], [Dzx, Dzy, Dzz]])

    #Inverse transformation matrix
    M_D_x = numpy.linalg.inv(M_x_D)

    #Determination of the alpha angle between Dz and Dy. It is in rad
    #alpha = math.acos((Dzx*Dyx + Dzy*Dyy + Dzz*Dyz)/(DX*DY))

    geo.close()
    # END OF PROCEDURE TO DEFINE THE FICTITIOUS BASIS OF THE SPACE
    ###############################################################################
    ###############################################################################

    # Now we have two nested cycles:
    # cycle 1: it is quitted if the target atom undergoes an event;
    # cycle 2: it is quitted once we accept the angle for the generation of the velocities
    #          for the target atom.

    #CYCLE 1
    cycle_1 = 0
    exit_condition_1 = 0
    infinite_loop = 0
    while (exit_condition_1 == 0 and infinite_loop < 30):

        # Generation of a sub_folder related to what happens to the selected atom
        current_run = str(general_run + str(cycle_1+1) + "/")
        current_run = str(runfolder2 + current_run)
        os.chdir(home)
        os.mkdir(current_run)
        os.chdir(current_run)
```

```
        infinite_loop = infinite_loop + 1

        cycle_1 = cycle_1 + 1
        ref_var = 0

        # GET A TARGET ATOM, RANDOMLY FROM THE LIST
        seed = int(random.randrange(0, len(Region_atoms)-1, 1))
        target_atom_identifier = Region_atoms[seed] - 1

        #pay attention target_atom_identifier starts from 1

        #CYCLE 2
        cycle_2 = 0
        exit_condition_2 = 0
        while exit_condition_2 == 0:
            cycle_2 = cycle_2 + 1

            ## generate randomly an angle theta between 0 and 90 and convert it in rad
            theta = (random.randrange(0, 50 , 5))
            theta_rad = theta*(math.pi/180)

            ## calculate the normalized Transmitted energy for this angle
            Norm_transmitted_energy = (math.cos(theta_rad))**2
            Trasmitted_energy = Norm_transmitted_energy*Max_energy

            ## generate a random probability
            rndm_P = random.randrange(0,1000,1)
            rndm_P = rndm_P/1000

            #Generate a folder for this angle
            current_attempt = str( "attempt_" + str(cycle_2) + "/")
            os.mkdir(current_attempt)
            os.chdir(current_attempt)

            #Write these information in the event detect file
            angle = open("event_detect.txt", "a")
            angle.write("Target atom:  " + str(target_atom_identifier + 1) + "\n")
            angle.write("\n")
            angle.write("Theta = " + str(theta) + " Probabilities [angle/rndm]:  " ...
            + str(Norm_transmitted_energy) + "   " + str(rndm_P) + "\n")


            voltage = 80*10**3
            nm = 1*10**-9
            j = 1.1*10**6*(nm**-2)
            eps = 8.9 * 10**-12
            q = 1.6*10**-19
            barn = 1*10**-28
            mo = 9.10956*10**-31
            c = 299792458
            c2 = c**2
            v = sqrt(2*voltage/mo)
            beta = v/c
            hbar = 1.0545718*10**-34
            Z = 6
            Tmax = 15.8
            theta_max = 50
```

```
            #maximum emission angle
            theta_max = 50
            theta_max_rad = theta_max*math.pi/180

            #Scattering cross section
            def sigma(T):
                return (((Z*q**2)/(4*math.pi*eps*2*mo*c2)*(Tmax/T))**2)*...
                ((1-beta**2)/(beta**4))*(1-((beta**2)*(T/Tmax))+ ...
                math.pi*(Z*q**2)/(hbar*c)*beta*(math.sqrt(T/Tmax) -(T/Tmax)));
           #normalized scattering cross section
            def n_sigma(T):
                return sigma(T)/sigma((Max_energy*(math.cos(theta_max_rad))**2))

            #EVALUATION OF ACCEPTANCE CRITERION TO EXIT CYCLE 2
            ## if P_generated > P_angle then restart the cycle 2
            if (rndm_P <= n_sigma(Trasmitted_energy)):
                exit_condition_2 = 1
            else:
                angle.close()
                os.chdir(home)
                os.chdir(current_run)

        ## generate a random phi angle and convert it to radiants
        phi = (random.randrange(0, 355 , 5))
        phi_rad = phi*(math.pi/180)
        #Write these information in the event detect file
        angle.write("Phi Angle = " + str(phi) + "\n")
        angle.write("\n")

        ###############################################################################
        ###############################################################################
        #GENERATION AND TRANSFORMATION IN THE CARTESIAN BASE OF THE ADDED VELOCITIES
        conv = 6.219054408*10**-8  #conversion from energy to speed
        V_tot = math.sqrt((Max_energy*Norm_transmitted_energy)/conv)*10**10
        #in (Angstrom/s)

        #Added velocity components in the fictitious base
        Added_D_velocity_vector = []

        #Added Vdx
        Added_D_velocity_vector.insert(0, V_tot*(math.sin(theta_rad))*(math.cos(phi_rad))

        #Added Vdy
        Added_D_velocity_vector.insert(1, V_tot*(math.sin(theta_rad))*(math.sin(phi_rad))

        #Added Vdz
        Added_D_velocity_vector.insert(2, -1*V_tot*(math.cos(theta_rad)))

        #Transformation of these components in the cartesian base
        Added_v_vec = numpy.dot(M_D_x, Added_D_velocity_vector)

        ###############################################################################
        ###############################################################################

        #Get the initial velocities in the cartesian coordinates for the target atom
        string = VR[seed].split()
        initial_vx = float(string[0].split('D+')[0])*(10**int(string[0].split('D+')[1]))
```

```
        initial_vy = float(string[1].split('D+')[0])*(10**int(string[1].split('D+')[1]))
        initial_vz = float(string[2].split('D+')[0])*(10**int(string[2].split('D+')[1]))

        #Calculate the final velocities for the target atom: syntax required for vels file
        final_vx = '%.15f'%((Added_v_vec[0] + initial_vx)/10**15)
        final_vy = '%.15f'%((Added_v_vec[1] + initial_vy)/10**15)
        final_vz = '%.15f'%((Added_v_vec[2] + initial_vz)/10**15)

        #Write these information in the Event detect file
        angle.write("initial vels atom number: " + str(target_atom_identifier + 1 ) +"\n")
        angle.write(str(initial_vx) + " " + str(initial_vy) + " " + str(initial_vz) +"\n")
        angle.write("\n")
        angle.write("added vels: \n")
        angle.write(str(Added_v_vec[0]) + " " + str(Added_v_vec[1]) + " " + ...
         str(Added_v_vec[2]) + "\n")
        angle.write("\n")
        angle.write("final vels: \n")
        angle.write(str(final_vx) + " " + str(final_vy) + " " + str(final_vz) + "\n")
        angle.write("\n")
        angle.write("\n")
        angle.close()

        #generate a name for the current folder: in order to copy the file for the MD
        here = str(current_run + current_attempt)

        os.chdir(home)

        #copying the required files for the MD in the folder named before
        New_path = copyfile(runfolder2 + "geo", here + "geo")
        New_path = copyfile(runfolder2 + "control", here + "control")
        New_path = copyfile(runfolder2 + "ffield", here + "ffield")

        #go in that folder to start the run
        os.chdir(here)

        ###############################################################################
        # VELS SYNTAX PROCEDURE
        #Procedure to write properly the string for the velocity of the target atom in the
        #vels file
        if (float(final_vx) < 0):
            if(len(final_vx) < 18):
                string_vx = str("-") + str(final_vx).split('-')[1] + str("1") +str("D+15")
            else:
                string_vx = str("-") + str(final_vx).split('-')[1] + str("D+15")
        else:
            if(len(final_vx) < 17):
                string_vx = str(" ") + str(final_vx) + str("1") + str("D+15")
            else:
                string_vx = str(" ") + str(final_vx) + str("D+15")

        if (float(final_vy) < 0):
            if(len(final_vy) < 18):
                string_vy = str("-") + str(final_vy).split('-')[1] + str("1") +str("D+15")
            else:
                string_vy = str("-") + str(final_vy).split('-')[1] + str("D+15")
        else:
            if(len(final_vy) < 17):
```

```
                string_vy = str(" ") + str(final_vy) + str("1") + str("D+15")
            else:
                string_vy = str(" ") + str(final_vy) + str("D+15")

        if (float(final_vz) < 0):
            if(len(final_vz) < 18):
                string_vz = str("-") + str(final_vz).split('-')[1] + str("1") +str("D+15")
            else:
                string_vz = str("-") + str(final_vz).split('-')[1] + str("D+15")
        else:
            if(len(final_vz) < 17):
                string_vz = str(" ") + str(final_vz) + str("1") + str("D+15")
            else:
                string_vz = str(" ") + str(final_vz) + str("D+15")
        # END OF VELS SYNTAX PROCEDURE
        ##############################################################################

        #String of the new cartesian velocities of the target atom
        target_atom_string = str("  ") + str(string_vx) + str("  ") + str(string_vy) +...
         str("  ") + str(string_vz) + "\n"

        ##############################################################################
        # VELS FILE
        new_molsav = open("vels","w+")
        new_molsav.write(lineeMolsav[0])
        new_molsav.write(lineeMolsav[1])
        new_molsav.write(lineeMolsav[2])
        new_molsav.write(lineeMolsav[3])
        #Generate the Vels file from the velocities that we saved before
        #Adding the right velocity to the target atom
        for j in range(4, 4 + atoms):
            new_molsav.write(lineeMolsav[j])
        new_molsav.write(lineeMolsav[4 + atoms])
        for j in range(4 + atoms + 1, 4 + atoms+1 + atoms):
            if ((int(j) - 1 - atoms - 3) == target_atom_identifier + 1):
            #insert the velocity for the target atom
                new_molsav.write(target_atom_string)
            else:
                new_molsav.write(lineeMolsav[j])

        #write the accelerations
        for j in range(4+atoms+1+atoms, len(lineeMolsav) ):
            new_molsav.write(lineeMolsav[j])

        molsav.close()
        new_molsav.close()
        # END VELS FILE
        ##############################################################################

        ##############################################################################
        # PROCEDURE TO SAVE THE INFORMATION REGARDING INITIAL CONNECTIVITY OF THE ATOM
        initial_connections = open("geo", "r")
        lines_initial_geo = initial_connections.readlines()
        old_nn_list = []
        old_position = []
        new_position = []
        old_nn_list = lines_initial_geo[target_atom_identifier + 1 + atoms +5].split()[2:]
```

```
        old_Ni_nn = 0
        old_C_nn = 0

        #Skip atom is required to exit cycle 1, INITIALIZATION
        skip_atom = 0

        #Save the number of C and Ni nearest neighbors before MD
        for i in old_nn_list:
            line = lines_initial_geo[int(i) + 4]
            if (line.split()[2] == "Ni"):
                old_Ni_nn = old_Ni_nn + 1
            elif (line.split()[2] == "C"):
                old_C_nn = old_C_nn + 1
        old_position.insert(0, float(R[target_atom_identifier].split()[3]))
        old_position.insert(1, float(R[target_atom_identifier].split()[4]))
        old_position.insert(2, float(R[target_atom_identifier].split()[5]))

        #Ceck whether this target atom has already undergone an event
        for m in past_events_list: #in past events list atoms have their geo name

            if((target_atom_identifier + 1) == int(m)): #this atom was already moved
                #therefore I consider it just if there is a ejection event
                skip_atom = 1

        initial_connections.close()
        # END OLD CONNECTIVITY PROCEDURE
        ################################################################################

        #MD SIMULATION FOR THIS VELS AND THIS TARGET ATOM
        subprocess.call('reaxff', shell=True)

        ################################################################################
        # EVENT DETECTION ALGORITHM
        # Ceck whether I had some events:
        # 1) Ejection;
        # 2) Dissolution or pre-dissolution

        remove_list = [k for k in os.listdir() if k.startswith('remove.')]
        new_C_nn = 0
        new_nn_list = []
        mantained_C_nn = 0

        #Boolean variables
        Ejection = 0
        Dissolution = 0
        Pre_dissolution = 0

        #Calculate the total amount of atoms from molfra
        molfra = open("molfra.out", "r")
        lines_molfra = molfra.readlines()
        Tot_atoms = int(lines_molfra[len(lines_molfra)-2].split()[4])
        molfra.close()

        event = open("event_detect.txt", "a")
        past_events_list.sort()

        #EJECTION
```

```
        if (len(remove_list) != 0): # I had an ejection event

            event.write("**** Detected an Ejection event ****\n")
            event.write("\n")
            event.write("Past events list:  " + str(past_events_list))

            Ejection = 1
            skip_atom = 0
            ###############################################################################
            #PROCEDURE TO GET THE INDEXES OF THE ATOMS REMOVED
            #we generate just 1 remove file
            rem_file = open(str(remove_list[0]), "r")
            lines_rem = rem_file.readlines()

            #get the number of removed atoms
            n_removed = int(lines_rem[0].split()[0])
            removed_list = []
            new_list = []

            #get the index of those removed atoms
            variable = 5
            for m in range(0, n_removed):
                removed_list.insert(n_removed, int(lines_rem[variable].split()[0]))
                variable = variable + 3

            #sort both lists
            removed_list.sort()
            past_events_list.sort()

            #Ceck if I am removing a new atom or a dissolved atom
            #This is specific for internal dissolution
            #In the other case no need for this
            for ll in removed_list:
                for mm in past_events_list:
                    if (int(ll) == int(mm)):
                        #I am removing an atom that was dissolved
                        #I don't want this
                        Ejection = 0
                        skip_atom = 1
                        break

            # END REMOVED ATOMS INDEXES
            ###############################################################################


            #If the atom removed was not in the List, I can continue
            if (skip_atom == 0):

                #Insert the number of the target atom in the event list and sort it
                past_events_list.insert(len(past_events_list), target_atom_identifier + 1)
                past_events_list.sort()

                ###############################################################################
                # PROCEDURE TO UPDATE THE PAST EVENTS LIST INDEXES

                #I have a new_list that is the one where I update the numbers
                new_list[0:] = past_events_list[0:]
```

```
                past_events_lenght = len(past_events_list)

                for rem_indx in range(0, len(removed_list)):
                    for pst_indx in range(0,len(past_events_list)):

                        #Now if target atom is removed, I remove it from past events list
                        if(removed_list[rem_indx] == int(past_events_list[pst_indx])):
                            del new_list[pst_indx]
                            del past_events_list[pst_indx]

                        #If the removed atom has an index below some past events atom
                        #I need to update their index new_index = new_index -1
                        if (pst_indx < len(past_events_list)):
                            if(removed_list[rem_indx] < int(past_events_list[pst_indx])):
                                for jj in range(pst_indx, len(new_list)):
                                    value = new_list.pop(jj)
                                    new_list.insert(jj, value - 1)
                                    value = 0
                                break
                                #I update the past events list
                past_events_list[0:] = new_list[0:]

                # END UPDATE INDEXES
                ###########################################################################

                ###########################################################################
                # PROCEDURE TO UPDATE THE FICTIOUS BASIS INDEXES
                ref_var = 1
                ref_file = open("Reference_atoms_info.txt", "a")
                ref_file.write("Atoms fictious basis: " + str(reference_atoms) + "\n")
                ref_file.write("\n")
                #I have a new_list that is the one where I update the numbers
                new_reference = []
                new_reference[0:] = reference_atoms[0:]
                for rem_indx in range(0, len(removed_list)):
                    for ref_indx in range(0,4):

                        #Now if a basis atom is removed
                        #I will skip this CYCLE 1
                        if(removed_list[rem_indx] == int(reference_atoms[ref_indx])):
                            Ejection = 0
                            skip_atom = 1
                            break

                        #If the removed atom has an index smaller than a reference atom
                        #I need to update the reference atom index new_index = new_index-1
                        if(removed_list[rem_indx] < int(reference_atoms[ref_indx])):
                            value = new_reference.pop(ref_indx)
                            new_list.insert(ref_indx, value - 1)
                            value = 0

                reference_atoms[0:] = new_reference[0:]
                ref_file.write("New fictious basis: " + str(reference_atoms)+ "\n")
                ref_file.write("\n")
                ref_file.close()
                # END UPDATE BASIS INDEXES
                ###########################################################################
```

```
                # END UPDATE BASIS INDEXES
                ###########################################################################



                #If I had an ejection write the information
                event.write("\n")
                event.write("Atoms Ejected:  " + str(removed_list))
                event.write("\n")
                event.write("\n")

                ###########################################################################

            rem_file.close()
# END EJECTION
##############################################################################################



##############################################################################################
# DISSOLUTION
        else:
            final_geometry_file = open("fort.90", "r")

            # Get the new connectivity information
            lines_final_geo = final_geometry_file.readlines()
            new_nn_list = lines_final_geo[target_atom_identifier + 1 +Tot_atoms + 5]...
            .split()[2:]
            new_Ni_nn = 0
            for nn in new_nn_list:
                line = lines_final_geo[int(nn) + 4]
                if (line.split()[2] == "Ni"):
                    new_Ni_nn = new_Ni_nn + 1
                elif (line.split()[2] == "C"):
                    new_C_nn = new_C_nn + 1
                    for onn in old_nn_list:
                        if (int(nn) == int(onn)):
                            mantained_C_nn = mantained_C_nn + 1

            # Get position information
            new_position.insert(0,float(lines_final_geo[target_atom_identifier + 5]...
            .split()[3]))
            new_position.insert(1,float(lines_final_geo[target_atom_identifier + 5]...
            .split()[4]))
            new_position.insert(2,float(lines_final_geo[target_atom_identifier + 5]...
            .split()[5]))
            position_variation = math.sqrt((new_position[0]-old_position[0])**2 +...
            (new_position[1]-old_position[1])**2 + (new_position[2]-old_position[2])**2)

            final_geometry_file.close()

            # Now I have information regarding the variation of the Number of Ni nearest n
            # Information regarding the variation of the number of C nn
            # Information regarding how many of this C atoms were nn
            # Information regarding the position variation

            #I write these information
```

```
            #I write these information
            event.write("Ni nn [old/new]:   " + str(old_Ni_nn) + " / " + ...
            str(new_Ni_nn) + " \n")
            event.write("C nn [old/new]:   " + str(old_C_nn) + " / " + ...
            str(new_C_nn) + " \n")
            event.write("mantained C nn:   " + str(mantained_C_nn) + " \n")
            event.write("\n")
            event.write("Old C nn: " + str(old_nn_list) + "\n")
            event.write("\n")
            event.write("New C nn: " + str(new_nn_list) + "\n" )
            event.write("position [old]:   " + str(old_position[0]) + " " + ...
            str(old_position[1]) +" " +str(old_position[2]) + "\n" )
            event.write("position [new]:   " + str(new_position[0]) + " " +...
            str(new_position[1]) +" " +str(new_position[2]) + "\n" )
            event.write("position variation:   " + str(position_variation) + "\n")

            #Dissolution event and new_Ni_nn != 0
            if((position_variation > 3.5 or new_C_nn == 0 or mantained_C_nn == 0) and ...
            ((new_Ni_nn >old_Ni_nn)):
                Dissolution = 1
                event.write("\n")
                event.write("\n")
                event.write("**** Detected a Dissolution event ****\n")
                event.write("\n")
                event.write("\n")
            #Pre-dissolution event
            if(((position_variation <= 3.5) and mantained_C_nn == 1) and ...
            new_Ni_nn > old_Ni_nn  ):
                Pre_dissolution = 0
                event.write("\n")
                event.write("\n")
                event.write("**** Detected a Pre-dissolution event ****\n")
                event.write("\n")
                event.write("\n")
# END DISSOLUTION
##########################################################################################

        if (skip_atom == 0and(Pre_dissolution == 1 or Dissolution == 1 or Ejection == 1)):

            # If one of those events is detected go further with fbMC
            exit_condition_1 = 1

            #If I had an ejection the list was already updated
            #Now I have to update as well in the case of other events
            if (Ejection == 0):
                past_events_list.insert(len(past_events_list), target_atom_identifier + 1)
                past_events_list.sort()

            event.write("Updated list of atoms which have already undergone an event: \n")
            event.write(str(past_events_list))
            event.write("\n")
            event.write("\n")

        else:
            event.write("\n")
            event.write("\n")
            event.write("**** No event detected ****\n")
```

```
            event.write("\n")
            event.write("\n")

        event.close()
    #END EVENT DETECTION ALGORITHM
    ###############################################################################



    #generating the new geo file
    os.remove('geo')
    os.rename(r"fort.90",r"geo")

    #copying the results in the results folder
    molsavList = [i for i in os.listdir() if i.startswith('molsav-')]
    molsavList.sort()
    os.chdir(home)

    #New_path = copyfile(runfolder2 + "xmolout", results2 + "xmolout")
    New_path = copyfile(here + "thermolog", results2 + "thermolog")
    New_path = copyfile(here + "summary.txt", results2 + "summary.txt")
    New_path = copyfile(here + "reaxout.kf", results2  + "reaxout.kf")
    New_path = copyfile(here + "molfra.out", results2 + "molfra.out")
    New_path = copyfile(here + "event_detect.txt", results2 + "event_detect.txt")
    New_path = copyfile(here + "xmolout", results2 + "xmolout")
    if (ref_var == 1):
        New_path = copyfile(here + "Reference_atoms_info.txt", results2 + ...
        "Reference_atoms_info.txt")
    #copying the input files in the next step's runfolder
    New_path = copyfile(here + str(molsavList[-1]), runfolder3 + "vels")
    New_path = copyfile(here + "geo", runfolder3 + "geo")

    #initializaion for the next iteration
    os.chdir("setup_files/step_2/")
    rmtree("runfolder/")
    os.mkdir("runfolder")
    New_path = copyfile("control","runfolder/control")
    New_path = copyfile("ffield","runfolder/ffield" )
    os.chdir(home)

###############################################
#Step 3

    #MD equilibration
    os.chdir(runfolder3)
    subprocess.call('reaxff', shell=True)

    #generating the new geo file
    os.remove("geo")
    os.rename(r"fort.90",r"geo")

    #copying the results in the results folder
    molsavList = [i for i in os.listdir() if i.startswith('molsav-')]
    molsavList.sort()
    os.chdir(home)
    #New_path = copyfile(runfolder3 + "xmolout", results3 + "xmolout")
    New_path = copyfile(runfolder3 + "thermolog", results3 + "thermolog")
```

```
    os.chdir(home)
    #New_path = copyfile(runfolder3 + "xmolout", results3 + "xmolout")
    New_path = copyfile(runfolder3 + "thermolog", results3 + "thermolog")
    New_path = copyfile(runfolder3 + "summary.txt", results3 + "summary.txt")
    New_path = copyfile(runfolder3 + "reaxout.kf", results3  + "reaxout.kf")
    #copying the input files in the next step's runfolder
    New_path = copyfile(runfolder3 + str(molsavList[-1]), runfolder4 + "vels")
    New_path = copyfile(runfolder3 + "geo", runfolder4 + "geo")


    #initializaion for the next iteration
    os.chdir("setup_files/step_3/")
    rmtree("runfolder/")
    os.mkdir("runfolder")
    New_path = copyfile("control","runfolder/control")
    New_path = copyfile("ffield","runfolder/ffield" )
    os.chdir(home)


################################################
#Step 4

    #local-fbMC routine
    os.chdir(runfolder4)
    subprocess.call('reaxff', shell=True)

    #generating the new geo file
    os.remove("geo")
    os.rename(r"fort.90",r"geo")

    #copying the results in the results folder
    molsavList = [i for i in os.listdir() if i.startswith('molsav-')]
    molsavList.sort()
    os.chdir(home)
    New_path = copyfile(runfolder4 + "xmolout", results4 + "xmolout")
    New_path = copyfile(runfolder4 + "thermolog", results4 + "thermolog")
    New_path = copyfile(runfolder4 + "summary.txt", results4 + "summary.txt")
    #New_path = copyfile(runfolder4 + "xmoloutMC", results4 + "xmoloutMC")
    New_path = copyfile(runfolder4 + "reaxout.kf", results4  + "reaxout.kf")
    #copy the files in the step_1 folder to repeat
    New_path = copyfile(runfolder4 + str(molsavList[-1]), runfolder1 + "vels")
    New_path = copyfile(runfolder4 + "geo", runfolder1 + "geo")

    #initializaion for the next iteration
    os.chdir("setup_files/step_4/")
    rmtree("runfolder/")
    os.mkdir("runfolder")
    New_path = copyfile("control","runfolder/control")
    New_path = copyfile("ffield","runfolder/ffield" )
    os.chdir(home)

################################################
#Renaming the current iteration folder with it's actual number
    os.chdir("results/")
    current_iteration = str("Iteration_") + str(h + 1)
    os.rename("temp_current_iteration", current_iteration)

```

D *NiCH* force field parameters

```
Reactive MD-force field: Ni/C/H April 2009
 39       ! Number of general parameters
   50.0000 !p(boc1)
    9.5469 !p(boc2)
   26.5405 !p(coa2)
    1.7224 !p(trip4)
    6.8702 !p(trip3)
   60.4850 !kc2
    1.0588 !p(ovun6)
    4.6000 !p(trip2)
   12.1176 !p(ovun6)
   13.3056 !p(ovun8)
  -70.5044 !p(trip1)
    0.0000 !Lower Taper-radius (swa)
   10.0000 !Upper Taper-radius (swb)
    2.8793 !Not used
   33.8667 !p(val7)
    6.0891 !p(lp1)
    1.0563 !p(val9)
    2.0384 !p(val10)
    6.1431 !Not used
    6.9290 !p(pen2)
    0.3842 !p(pen3)
    2.9294 !p(pen4)
   -2.4837 !Not used
    5.7796 !p(tor2)
   10.0000 !p(tor3)
    1.9487 !p(tor4)
   -1.2327 !not used
    2.1645 !p(cot2)
    1.5591 !p(vdW1)
    0.1000 !Cutoff for bond order*100  (cutoff)
    2.1365 !p(coa4)
    0.6991 !p(ovun4)
   50.0000 !p(ovun3)
    1.8512 !p(val8)
    0.5000 !Not used
   20.0000 !Not used
    5.0000 !not used
    0.0000 !not used)
    2.6962 !p(coa3)
  3    ! Nr of atoms; atomID;ro(sigma); Val;atom mass;Rvdw;Dij;gamma;ro(pi);Val(e)
            alfa;gamma(w);Val(angle);p(ovun5);n.u.;chiEEM;etaEEM;n.u.
            ro(pipi);p(lp2);Heat increment;p(boc4);p(boc3);p(boc5),n.u.;n.u.
            p(ovun2);p(val3);n.u.;Val(boc);p(val5);n.u.;n.u.;n.u.
 C    1.3831   4.0000  12.0000   1.8814   0.1923   0.9000   1.1363   4.0000
      9.7821   2.1317   4.0000  30.0000  79.5548   5.9666   7.0000   0.0000
      1.2071   0.0000 186.1720   9.0068  34.9357  13.5366   0.8563   0.0000
     -2.8983   2.5675   1.0564   4.0000   2.9663   0.0000   0.0000   0.0000
 H    0.8873   1.0000   1.0080   1.5420   0.0598   0.6883  -0.1000   1.0000
      8.1910  30.9706   1.0000   0.0000 121.1250   3.5768  10.5896   1.0000
     -0.1000   0.0000  54.0596   1.3986   2.1457   0.0003   1.0698   0.0000
    -15.7683   2.1488   1.0338   1.0000   2.8793   0.0000   0.0000   0.0000
```

```
 Ni   1.8201   2.0000  58.6900   1.9449   0.1880   0.8218   0.1000   2.0000
     12.1594   3.8387   2.0000   0.0000   0.0000   4.8038   7.3852   0.0000
     -1.0000   0.0000  95.6300  50.6786   0.6762   0.0981   0.8563   0.0000
     -3.7733   3.6035   1.0338   8.0000   2.5791   0.0000   0.0000   0.0000
  6      ! Nr of bonds; at1;at2;De(sigma);De(pi);De(pipi);p(be1);p(bo5);13corr;n.u.;p(bo6)
  ,p(ovun1)
                       p(be2);p(bo3);p(bo4);n.u.;p(bo1);p(bo2)
  1  1 143.3883  96.3926  76.4404  -0.7767  -0.4710   1.0000  34.9900   0.5108
         0.4271  -0.1116   9.0638   1.0000  -0.0840   6.7452   1.0000   0.0000
  1  2 181.9084   0.0000   0.0000  -0.4768   0.0000   1.0000   6.0000   0.7499
        12.8085   1.0000   0.0000   1.0000  -0.0608   6.9928   0.0000   0.0000
  2  2 168.2342   0.0000   0.0000  -0.2191   0.0000   1.0000   6.0000   1.0062
         6.1152   1.0000   0.0000   1.0000  -0.0889   6.0000   0.0000   0.0000
  1  3  83.5810   9.0383   0.0000   0.2531  -0.2000   1.0000  16.0000   0.0529
         1.4085  -0.1113  13.3900   1.0000  -0.1436   4.5683   1.0000   0.0000
  2  3 114.7566   0.0000   0.0000  -0.8939   0.0000   1.0000   6.0000   0.1256
         0.1054   1.0000   0.0000   1.0000  -0.1196   5.0815   0.0000   0.0000
  3  3  91.2220   0.0000   0.0000  -0.2538  -0.2000   0.0000  16.0000   0.2688
         1.4651  -0.2000  15.0000   1.0000  -0.1435   4.3908   0.0000   0.0000
  3    ! Nr of off-diagonal terms. at1;at2;Dij;RvdW;alfa;ro(sigma);ro(pi);ro(pipi)
  1  2   0.1188   1.4017   9.8545   1.1203  -1.0000  -1.0000
  1  3   0.0800   1.7085  10.0895   1.5504   1.4005  -1.0000
  2  3   0.0366   1.7306  11.1019   1.2270  -1.0000  -1.0000
 18    ! Nr of angles. at1;at2;at3;Thetao,o;p(val1);p(val2);p(coa1);p(val7);p(pen1);
 p(val4)
  1  1  1  72.7917  38.5829   0.7209   0.0000   0.1409  17.4509   1.0670
  1  1  2  72.1533  14.2108   6.2512   0.0000   0.0100   0.0000   1.1022
  2  1  2  73.2608  24.9703   3.7807   0.0000   0.1335   0.0000   3.0461
  1  2  2   0.0000   0.0000   6.0000   0.0000   0.0000   0.0000   1.0400
  1  2  1   0.0000   7.5000   5.0000   0.0000   0.0000   0.0000   1.0400
  2  2  2   0.0000  27.9213   5.8635   0.0000   0.0000   0.0000   1.0400
  1  3  1  62.5000  16.6806   0.7981   0.0000   0.9630   0.0000   1.0711
  1  1  3  87.6241  12.6504   1.8145   0.0000   0.6154   0.0000   1.5298
  3  1  3 100.0000  40.4895   1.6455   0.0000   0.0100   0.0000   1.7667
  1  3  3   5.0994   3.1824   0.7016   0.0000   0.7465   0.0000   2.2665
  2  3  2 106.3969  30.0000   0.9614   0.0000   1.9664   0.0000   2.2693
  2  2  3   0.0000  26.3327   4.6867   0.0000   0.8177   0.0000   1.0404
  3  2  3   0.0000  60.0000   1.8471   0.0000   0.6331   0.0000   1.8931
  2  3  3  30.3748   1.0000   4.8528   0.0000   0.1019   0.0000   3.1660
  2  3  3 180.0000 -27.2489   8.3752   0.0000   0.8112   0.0000   1.0004
  1  3  2  97.5742  10.9373   2.5200   0.0000   1.8558   0.0000   1.0000
  1  2  3   0.0000   0.2811   1.1741   0.0000   0.9136   0.0000   3.8138
  2  1  3  84.0006  45.0000   0.6271   0.0000   3.0000   0.0000   1.0000
 12    ! Nr of torsions. at1;at2;at3;at4;;V1;V2;V3;p(tor1);p(cot1);n.u;n.u.
  1  1  1  1  -0.5000  53.0886  -0.1335  -6.2875  -1.9524   0.0000   0.0000
  1  1  1  2  -0.4614  29.0459   0.2551  -4.8555  -2.7007   0.0000   0.0000
  2  1  1  2  -0.2833  31.2867   0.2965  -4.8828  -2.4652   0.0000   0.0000
  0  1  2  0   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0  2  2  0   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
  0  1  1  0   0.0000  50.0000   0.3000  -4.0000  -2.0000   0.0000   0.0000
  1  1  1  3   0.0000   5.0000   0.4000  -6.0000   0.0000   0.0000   0.0000
  3  1  1  3   0.0000  44.3024   0.4000  -4.0000   0.0000   0.0000   0.0000
  2  1  1  3   0.0000  21.7038   0.0100  -4.0000   0.0000   0.0000   0.0000
  2  1  3  1   0.0000   5.2500   0.0100  -6.0000   0.0000   0.0000   0.0000
  1  1  3  1   0.0000   5.1676   0.0100  -5.9539   0.0000   0.0000   0.0000
  1  1  3  2   0.0000   5.1676   0.0100  -5.9539   0.0000   0.0000   0.0000
  0    ! Nr of hydrogen bonds. at1;at2;at3;r(hb);p(hb1);p(hb2);p(hb3)
```

References

[1] R.Gauvin, *A Review of Transmission Electron Microscopy for the Characterization of Materials*, Proc. SPIE 10291, Materials Characterization and Optical Probe Techniques: A Critical Review, 102910C, **1997**

[2] Jens Leschner, *High-resolution electron tomography on beam-sensitive carbon materials*, Open Access Repositorium der Universität Ulm, Dissertation, **2011**

[3] Terrones M, Terrones H., *The carbon nanocosmos: novel materials for the twenty-first century*, Philos Trans R Soc Lond Ser A: Math Phys Eng Sci, 361(1813):2789–806, **2003**

[4] Harris PJF *New perspectives on the structure of graphitic carbons*, Crit Rev Solid State Mater Sci, 30(4):235–53, **2005**

[5] Chuvilin A, Khlobystov AN, Obergfell D, Haluska M, Yang S, Roth S, et al., *Observations of chemical reactions at the atomic scale: dynamics of metal-mediated fullerene coalescence and nanotube rupture*, Angew Chem Int Ed, 49(1):193–6, **2010**

[6] Warner JH, Schäffel F, Zhong G, Rümmeli MH, Bühner B, Robertson J, et al., *Investigating the diameter-dependent stability of single-walled carbon nanotubes*, ACS Nano, 3(6):1557–63, **2009**

[7] Dahmen U, Erni R, Radmilovic V, Kisielowski C, Rossell M-D, Denes P., *Background, status and future of the transmission electron aberration-corrected microscope project*, Philos Trans R Soc Lond Ser A: Math Phys Eng Sci, 367(1903):3795–808, **2009**

[8] Sumio Iijima, *Helical microtubules of graphitic carbon*, Nature volume 354, pages56–58, **1991**

[9] Tasis D., Tagmatarchis N., Bianco A. and Prato M. *Chemistry of carbon nanotubes*, Chem. Rev. , 106, 1105–1136, **2006**

[10] Valentin N Popov, *Carbon nanotubes: properties and application*, Materials Science and Engineering: R: Reports Volume 43, Issue 3, Pages 61-102, **2004**

[11] Pan X. and Bao X. *The effects of confinement inside carbon*, Acc. Chem. Res. , 44, 553–562, **2011**

[12] Cao, Kecheng and Zoberbier, Thilo and Biskupek Johannes and Botos Akos and McSweeney Robert L. and Kurtoglu, Abdullah and Stoppiello, Craig T. and Markevich, Alexander V. and Besley, Elena and Chamberlain, Thomas W. and Kaiser, Ute and Khlobystov, Andrei N., *Comparison of atomic scale dynamics for the middle and late transition metal nanocatalysts*, Nature Communications volume 9, Article number: 3382, **2018**

[13] Daniel Iglesias and Michele Melchionna, *Enter the Tubes: Carbon Nanotube Endohedral Catalysis*, Catalysts, 9, 128, *2019*

[14] Thilo Zoberbier, Thomas W. Chamberlain, Johannes Biskupek, Mikhail Suyetin, Alexander G. Majouga, Elena Besley, Ute Kaiser and Andrei N. Khlobystov, *Investigation of the Interactions and Bonding between Carbon and Group VIII Metals at the Atomic Scale*, Small ,12, No. 12, 1649–1657, **2016**

[15] Thomas W. Chamberlain, Johannes Biskupek, Stephen T. Skowron, Peter A. Bayliss, Elena Bichoutskaia, Ute Kaiser and Andrei N. Khlobystov, *Isotope Substitution Extends the Lifetime of Organic Molecules in Transmission Electron Microscopy*, Small , 11, No. 5, 622–629, **2015**

[16] Kecheng Can, Thomas W. Chamberlain, Johannes Biskupek, Thilo Zoberbier, Ute Kaiser and Andrei N. Khlobystov, *Direct correlation of carbon nanotube nucleation and growth with the atomic structure of Rhenium Nanocatalysts simulated and imaged by the electron beam*, Nano Lett, 18, 6334-6339, **2018**

[17] Thomas W. Chamberlain, Johannes Biskupek, Stephen T. Skowron, Alexander V. Markevich, Simon Kurasch,§Oliver Reimer, Kate E. Walker, Graham A. Rance, Xinliang Feng, Klaus Müllen, Andrey Turchanin, Maria A. Lebedeva, Alexander G. Majouga, Valentin G. Nenajdenko, Ute Kaiser, Elena Besley and Andrei N. Khlobystov, *Stop-Frame Filming and Discovery of Reactions at the Single-Molecule Level by Transmission Electron Microscopy*, ACS Nano, 11, 25092520, **2017**

[18] Yang S. and Wang C.R., *Endohedral Fullerenes: From Fundamentals to Applications* World Scientific: Singapore, **2014**

[19] Alexander S. Sinitsa, Thomas W. Chamberlain, Thilo Zoberbier, Irina V. Lebedeva, Andrey M. Popov, Andrey A. Knizhnik, Robert L. McSweeney, Johannes Biskupek, Ute Kaiser and Andrei N. Khlobystov, *Formation of Nickel Clusters Wrapped in*

Carbon Cages: Toward New Endohedral Metallofullerene Synthesis, Nano Lett. , 17, 10821089, *2017*

[20] Chuvilin A., Kaiser U., Bichoutskaia E., Besley N. A., Khlobystov A. N., *Direct Transformation of Graphene to Fullerene*, Nat. Chem. 2, 450453, **2010**

[21] Santana A., Zobelli A., Kotakoski J., Chuvilin A., Bichoutskaia E., *Inclusion of Radiation Damage Dynamics in High-Resolution Transmission Electron Microscopy Image Simulations: the Example of Graphene*, Phys. Rev. B: Condens. Matter Mater. Phys. 87, 094110, **2013**

[22] Irina V. Lebedeva, Thomas W. Chamberlain, Andrey M. Popov, Andrey A. Knizhnik, Thilo Zoberbier, Johannes Biskupek, Ute Kaiserf and Andrei N. Khlobystov, *The atomistic mechanism of carbon nanotube cutting catalyzed by nickel under an electron beam*,Nanoscale, 6, 14877, **2014**

[23] Skowron S. T., Lebedeva I. V., Popov A. M., Bichoutskaia E., *Approaches to Modelling Irradiation-Induced Processes in Transmission Electron Microscopy* Nanoscale , 5, 66776692, **2013**.

[24] Ziwei Xu, Lu Qiuabc and Feng Ding, *The kinetics of chirality assignment in catalytic single-walled carbon nanotube growth and the routes towards selective growth*, Chem. Sci., 9, 3056, **2018**

[25] Xiao Wang , Maoshuai He, Feng Ding, *Chirality-controlled synthesis of single-walled carbon nanotubes—From mechanistic studies toward experimental realization*, Materials Today, Volume 21, Issue 8, Pages 845-860, **2018**

[26] Peter Atkins and Ronald Friedman, *Molecular Quantum Mechanics*, Fifth edition, Oxford university press, **2011**.

[27] Monticelli L., Tieleman D.P., *Force Fields for Classical Molecular Dynamics* In: Monticelli L., Salonen E. (eds) Biomolecular Simulations. Methods in Molecular Biology (Methods and Protocols), vol 924. Humana Press, Totowa, NJ, **2013**

[28] Jonathan E. Mueller, Donato Fantauzzi, and Timo Jacob, *Multiscale modelling of Electrochemical Systems*, Electrocatalysis: Theoretical Foundations and Model Experiments, First Edition, Edited by Richard C. Alkire, Ludwig A. Kibler, Dieter M. Kolb, and Jacek Lipkowski, **2013**

[29] Adri C. T. van Duin, Siddharth Dasgupta, Francois Lorant and William A. Goddard, *ReaxFF: A Reactive Force Field for Hydrocarbons*, J. Phys. Chem. A , 105, 41, 9396-9409, **2001**

[30] Kimberly Chenoweth, Adri C. T. van Duin, William A. Goddard, *ReaxFF Reactive Force Field for Molecular Dynamics Simulations of Hydrocarbon Oxidation*, J. Phys. Chem. A, 112, 5, 1040-1053, **2008**

[31] Loup Verlet, *Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules*, Phys. Rev. 159, 98, **1967**

[32] Hünenberger P. H., *Thermostat Algorithms for Molecular Dynamics Simulations*, Advances in Polymer Science, 105–149 **2005**

[33] Greiner W. , *Thermodynamik und Statistische Mechanik*, Verlag Harri Deutsch, **1993**

[34] Glenn J. Martyna, Michael L. Klein and Mark Tuckerman, *Nose Hoover chains: The canonical ensemble via continuous dynamics*, J. Chem. Phys. 97 (4), **1992**

[35] Wilfried J. Mortier, Swapan K. Ghosh and S. Shankar, *Electronegativity-equalization method for the calculation of atomic charges in molecules*, J. Am. Chem. Soc., 108, 15, 4315-4320, **1986**

[36] Geert O. A. Janssens, Bart G. Baekelandt, Helge Toufar, Wilfried J. Mortier and Robert A. Schoonheydt, *Comparison of Cluster and Infinite Crystal Calculations on Zeolites with the Electronegativity Equalization Method (EEM)*, J. Phys. Chem., 99, 10, 3251-3258, **1995**

[37] Sanderson R.T. *Polar Covalence*, Academic Press; New York, **1983**

[38] F. Ding and B. I. Yakobson , *Energy-Driven Kinetic Monte Carlo Method and Its Application in Fullerene Coalescence* J. Phys. Chem. Lett., 5 , 2922 -2926, **2014**

[39] Jean-Noel Fuchs, Mark Oliver Goerbig, *Introduction to the Physical Properties of Graphene*, **2008**

[40] Michael J. Murphy,Gregory A. Voth and Amy L. R. Bug, *Classical and Quantum Transition State Theory for the Diffusion of Helium in Silica Sodalite*, J. Phys. Chem. B,101,491-503, **1997**

[41] Ziwei Xu, Tianying Y and Feng Ding, *Atomistic simulation of the growth of defect-free carbon nanotubes*, Chem. Sci.,6, 4704-4711, **2015**

[42] Erik C. Neyts, Adri C. T. van Duin, Annemie Bogaerts, *Changing Chirality during Single-Walled Carbon Nanotube Growth: A Reactive Molecular Dynamics/Monte Carlo Study*, J. Am. Chem. Soc., 133, 43, 17225-17231, **2011**

[43] Kristof M. Bal and Erick C. Neyts, *Merging Metadynamics into Hyperdynamics: Accelerated Molecular Simulations Reaching Time Scales from Microseconds to Seconds.* Journal of Chemical Theory and Computation, **2015**

[44] Maarten J. Mees, Geoffrey Pourtois, Erik C. Neyts, Barend J. Thijsse, Andre Stesmans, *Uniform-acceptance force-bias Monte Carlo method with time scale to study solid-state diffusion*, Phys. Rev. B 85, 134301, **2012**

[45] Kristof M. Bal and Erik C. Neyts, *On the time scale associated with Monte Carlo simulations*, The Journal of Chemical Physics, **2014**

[46] Maria Timonova, Jasper Groenewegen, and Barend J. Thijsse, *modelling diffusion and phase transitions by a uniform-acceptance force-bias Monte Carlo method*, Phys. Rev. B 81, 144107, **2010**

[47] Erik C. Neyts, Yasushi Shibuta, Adri C. T. van Duin and Annemie Bogaerts, *Catalyzed Growth of Carbon Nanotube with Definable Chirality by Hybrid Molecular Dynamics-Force Biased Monte Carlo Simulations*, ACS Nano, 4, 11, 6665-6672, **2010**

[48] Jorge Nocedal , *Updating quasi-Newton matrices with limited storage*, Math. Comp. 35, 773-782, **1980**

[49] John Dennis, Jorge Moré, *Quasi-Newton Methods, Motivation and Theory*, SIAM Review, Society for Industrial and Applied Mathematics, 19 (1), pp.46-89, **1977**

[50] Amsterdam Density Functional program, Theoretical Chemistry, Vrije Universiteit, Amsterdam, URL: http://www.scm.com.

[51] Ovito (Open Visualization Tool), Version 2.9.0, Copyright (C) 2017, Alexander Stukowski, http://www.ovito.org/

[52] Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org

[53] MATLAB, 2010. version 9.3.0713579 (R2017b), Natick, Massachusetts: The MathWorks Inc., https://it.mathworks.com/

[54] D.Fantauzzi, J.E. Mueller, L. Sabo, A.C.T. van Duin and T. Jacob *Surface Buckling and Subsurface Oxygen: Atomistic Insights into the Surface Oxidation of Pt(111)*, ChemPhysChem, Vol.16: 2797-2802, **2015**

[55] J.E. Mueller, A.C.T van Duin, W.A. Goddard, *Development and Validation of ReaxFF Reactive Force Field for Hydrocarbon Chemistry Catalyzed by Nickel*, The Journal of Physical Chemistry C, **2010**

[56] Kittel C., *Introduction to Solid State Physics*, 8th ed.; Wiley: New York, **2005**

[57] R. Tran, Z. Xu, B. Radhakrishnan, D. Winston, W. Sun, K.A. Persson, S. Ping Ong, *Surface energies of elemental crystals*, Scientific Data, **2016**

[58] S. Hong, Young-Han Shin J. Ihm, *Crystal Shape of a Nickel Particle Related to Carbon Nanotube Growth*, Japanese journal of Applied Physics, **2002**

[59] Charles L. Cleveland and Uzi Landman,*The energetics and structure of nickel clusters: Size dependence*, The Journal of Chemical Physics, **1991**

[60] E.K. Parks, L. Zhu, J. Ho and S.J. Riley, *The structure of small nickel clusters I.* $Ni_3 - Ni_{15}$, The Journal of Chemical Physics, **1994**

[61] E.K. Parks, L. Zhu, J. Ho and S.J. Riley, *The structure of small nickel clusters II.* $Ni_{16} - Ni_{28}$, The Journal of Chemical Physics, **1995**

[62] Mark S. Stave and Andrew DePristo, *The structure of* Ni_n *and* Pd_n *clusters:* $4 \leq N \leq 23$, J.Chem.Phys. 97, 3386, **1992**

[63] P. L. Rodríguez-Kessler and A. R. Rodríguez-Domínguez, *Stability of Ni Clusters and the Adsorption of CH4: First-Principles Calculations*, J. Phys. Chem. C, 119, 22, 12378-12384, **2015**

[64] N. N. Lathiotakis, A. N. Andriotis, M. Menon, and J. Connolly, *Tigth binding molecular dynamics study of Ni clusters*, J. Chem. Phys. 104, 992, **1996**

[65] V. G. Grigoryan and M. Springborg, *A theoretical study of the structure of Ni clusters Ni_N*, Phys. Chem. Chem. Phys.,3, 5135-5139, **2001**

[66] Nusret Duygu Yilmazer, Mehmet Ferdi Fellah, Isik Önal, *Ni_{55} nanocluster: a density functional theory study of the binding energy of nickel and ethylene adsorption* , Turk J Chem, 36 , 55 – 67, **2012**

[67] Chenglin Luo, *Energies and structural properties of nickel clusters determined by tight-binding simulations: Ni_4-Ni_{55}*, Modelling Simul. Mater. Sci. Eng. 10, 13–20, **2002**

[68] D. Chung, *Review Graphite*, Journal of Material Science 37, 1475, **2002**

[69] Hyeondeok Shin, Sinabro Kang, Jahyun Koo, Hoonkyung Lee, and Yongkyung Kwon, *Cohesion Energetics of Carbon Allotropes : Quantum Monte Carlo Study*, J. Chem. Phys. 140, 114702, **2014**

[70] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E. Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N. Groves, Bjørk Hammer, Cory Hargus, Eric D. Hermes, Paul C. Jennings, Peter Bjerre Jensen, James Kermode, John R. Kitchin, Esben Leonhard Kolsbjerg, Joseph Kubal, Kristen Kaasbjerg, Steen Lysgaard, Jón Bergmann Maronsson, Tristan Maxson, Thomas Olsen, Lars Pastewka, Andrew Peterson, Carsten Rostgaard, Jakob Schiøtz, Ole Schütt, Mikkel Strange, Kristian S. Thygesen, Tejs Vegge, Lasse Vilhelmsen, Michael Walter, Zhenhua Zeng, Karsten Wedel Jacobsen *The Atomic Simulation Environment—A Python library for working with atoms* J. Phys.: Condens. Matter Vol. 29 273002, **2017**

[71] ASE version 3.17.0, https://wiki.fysik.dtu.dk/ase/

[72] Nanocap, M. Robinson copyright 2020, version 1.0b15, http://nanocap.sourceforge.net

[73] Daniel Hedman, Hamid Reza Barzegar, Arne Rosén, Thomas Wagberg and J. Andreas Larsson, *On the Stability and Aboundance of Single Walled Carbon Nanotubes*, Nature - Scientific Reports, **2015**

[74] X. Blase, Lorin X. Benedict, Eric L. Shirley and Steven G. Louie, *Hybridization Effect and Metallicity in Small Radius Carbon Nanotubes*, Phys. Rev. Lett. 72, 1878, **1993**

[75] GuiXiao Jia, Lei Li, TongWei Wu, XiaoXia Wang and ShengLi An, *Curvature, vacancy size and chirality effects ofmono- to octa-vacancies in zigzag single-walledcarbon nanotubes*, New J.Chem., 40,8625, **2016**

[76] Zhengyang Gao, Ang Lia, Xiaoshuo Liua, Chuanzhi Maa, Xiang Lia, Weijie Yanga, Xunlei Dingb, *Density functional study of the adsorption of NO on Ni_n (n = 1,2,3 and 4) T clusters doped functionalized graphene support*, Applied Surface Science, Volume 481, Pages 940-950, **2019**

[77] Thilo Zoberbier, Thomas W. Chamberlain, Johannes Biskupek, Mikhail Suyetin, Alexander G. Majouga, Elena Besley, Ute Kaiser, Andrei N.Khlobystov, *Investigation of the Interactions and Bonding between Carbon and Group VIII Metals at the Atomic Scale*, Wiley Small Journal, 12, No.12, 1649–1657, **2016**

[78] P. Jung,*1.7.1.1 Electron irradiation*, Landolt-Börnstein - Group III Condensed Matter 25 (Atomic Defects in Metals) ,H. Ullmaier (ed.) Springer Materials, **1991**

[79] A. Zobelli, A. Gloter, C. P. Ewels, G. Seifert, and C. Colliex, *Electron knock-on cross section of carbon and boron nitride nanotubes*, Phys. Rev. B 75, 245402, **2007**

[80] Florian Banhart, *Irradiation effects in carbon nanostructures*, Rep. Prog. Phys. 62, 1181–1221, **1999**

[81] Nevill Francis Mott, *The scattering of fast electrons by atomic nuclei*, 124 Proc. R. Soc. Lond. A, **1929**

[82] Nevill Francis Mott, *The polarisation of electrons by double scattering*, 135 Proc. R. Soc. Lond. A, **1932**

[83] William A. McKinley, Jr. and Herman Feshbach, *The Coulomb Scattering of Relativistic Electrons by Nuclei*, Phys. Rev. 74, 1759 , **1948**

[84] Tiago Botari, Ricardo Paupitz, Pedro Alves da Silva Autreto, Douglas S.Galvao, *Graphene healing mechanisms: A theoretical investigation*, Carbon, Volume 99, Pages 302-309, **2016**

[85] F. Banhart, J. X. Li and A. V. Krasheninnikov, *Carbon nanotubes under electron irradiation: Stability of the tubes and their action as pipes for atom transport*, PHYSICAL REVIEW B 71,241408, **2005**

[86] J. Kotakoski, A. V. Krasheninnikov and K. Nordlund, *Energetics, structure, and long-range interaction of vacancy-type defects in carbon nanotubes: Atomistic simulations*, PHYSICAL REVIEW B 74, 245420, **2006**

[87] Jianhui Yuan and K.M. Liew, *Effects of vacancy defect reconstruction on the elasticproperties of carbon nanotubes*, CARBON, 47, 1526–1533, **2009**

[88] J.E.Padilha, R.G.Amorim, A.R.Rocha, A.J.R.da Silva and A.Fazzio, *Energetics and stability of vacancies in carbon nanotubes*, Solid State Communications, Volume 151, Issue 6, March, Pages 482-486, **2011**

[89] Tetsuya Morishita, *Fluctuations formulas in molecular-dynamics simulations with the weak coupling heath bath*, J. CHem. Phys. 113, 2976, **2000**

[90] Recep Zan, Quentin M. Ramasse, Ursel Bangert, Konstantin S. Novoselov, *Graphene Reknits Its Holes*, Nano Lett. 12, 8, 3936-3940 **2012**

[91] Satoru Fukuhara, Fuyuki Shimojo and Yasushi Shibuta ,*Conformatio and catalytic activity of nickel–carbon cluster for ethanoldissociation in carbon nanotube synthesis:Ab initiomolecular dynamicssimulation*, Chemical Physics Letters 679 164–171165, **2017**

[92] Banhart F., Kotakoski J., Krasheninnikov AV., *Structural defects in graphene*, ACS Nano , 5, 1, 26-41, **2011**

[93] Zhimin Yang, Qiang Wang, Xiaoye Shan, Shuo-Wang Yangb and Hongjun Zhu, *Theoretical investigation on carbon nucleation on nickel carbides at initial stages of single-walled carbon nanotube formation*, Phys. Chem. Chem. Phys.,16, 19654-19660, **2014**

[94] Umedjon Khalilov, Annemie Bogaerts Erik C. Neyts, *Atomic scale simulation of carbon nanotube nucleation from hydrocarbon precursors*, Nat Commun 6, 10306, **2015**

[95] Y. Shibuta, T. Suzuki, *Melting and nucleation of iron nanoparticles: a molecular dynamics study*, Chem. Phys. Lett., 445, p. 265 **2007**

[96] M Meyyappan, Lance Delzeit, Alan Cassell and David Hash, *Carbon nanotube growth by PECVD: a review*, Plasma Sources Sci. Technol. 12 205–216, **2003**

[97] Feng Ding, Kim Bolton, Arne Rosén, *Nucleation and Growth of Single-Walled Carbon Nanotubes: A Molecular Dynamics Study*, J. Phys. Chem. B, 108, 45, 17369-17377, **2004**

[98] Ali Rinaldi, Jean-Philippe Tessonnier, Manfred E. Schuster, Raoul Blume, Frank Girgsdies, Qiang Zhang, Timo Jacob, Sharifah Bee Abd Hamid, Dang Sheng Su, Robert Schlögl, *Dissolved Carbon Controls the Initial Stages of Nanocarbon Growth*, Angew. Chem. Int. Ed. ,50, 3313 –3317, **2011**

[99] Wang Gao, Jonathan E. Mueller, Josef Anton, Qing Jiang, and Timo Jacob, *Nickel Cluster Growth on Defect Sites of Graphene:A Computational Study*, Angew. Chem. Int. Ed. ,52, 14237 –14241, **2013**

[100] TextWrangler, Copyright 1992-2020 Bare Bones Software, http://www.barebones.com

Acknowledgements

I would like to thank my supervisor Professor Scotognella not just for the opportunity to develop my thesis work at the Institute of Electrochemistry at Ulm University, but also for all the extremely valuable advice he had give me throughout the years of my Master degree.
I want to extend my gratitude to my co-supervisor Professor Jacob for giving me the opportunity of working on such interesting project.
I also want to thank the Institut of Electrochemistry for having hosted me and supported financially together with Universitäts Klinikum Ulm.
I wish to show my gratitude also to M.Sc. Christoph Jung, M.Sc. Daniel Gaissmeier and M.Sc. Matthias Van Der Borg for their active support throughout the whole work.
I would like to thank all my office colleagues, especially Dr. Samuel Bertolini Da Silva and M.Sc. Changbin Im, for their "technical assistance" and the nice coffee breaks.
I would also like to thank Professor Puppin for its patience and its advice.
I wish to show my gratitude also to Riccardo Piergiacomi that helped me a lot figuring out how to write this thesis.
Un ringraziamento speciale va a Giovanni Floreale, Andrea Fiamberti e Carlo Modica: la vostra amicizia è stata di grande ispirazione per me. Se questi anni al Politecnico di Milano sono stati non solo formativi a livello professionale, ma anche a livello personale è anche e sopratutto merito vostro.
Un grazie va anche a tutta la mia famiglia (nessuno escluso).
Vorrei ringraziare dal profondo del mio cuore i miei genitori per il loro inestimabile supporto durante questi anni, la possibilità di poter scegliere liberamente la mia strada è una grandissima fortuna che mi avete permesso di avere e per cui vi sarò sempre riconoscente.
Un grazie speciale va anche a mio fratello per avermi sempre ed incondizionatamente spalleggiato, comprendendo i sacrifici da me fatti per giungere fino a qui.
Per ultimo vorrei lasciare una nota a me stesso, quando in futuro, forse, rileggerò questo lavoro: ricorda, sono state la determinazione, la dedizione la curiosità ma sopratutto la positività, che ti hanno permesso di superare ogni ostacolo, arrivando fino a qui, felice.