



Computer models for the study of graphene

Federico D'Ambrosio

Eppur si muove!
And yet, it moves!

Galileo Galilei (1564-1642) [1]

The research reported in this thesis has been carried out under the auspices of the Research Institute of Information and Computing Sciences at Utrecht University.

DOI: 10.33540/1091

ISBN: 978-90-393-7463-4

Copyright 2022 by Federico D'Ambrosio

Computer models for the study of graphene

Computermodellen voor het bestuderen van grafeen
(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit Utrecht
op gezag van de
rector magnificus, prof.dr. H.R.B.M. Kummeling,
ingevolge het besluit van het college voor promoties
in het openbaar te verdedigen op

donderdag 21 april 2022 des middags te 12.15 uur

door

Federico D'Ambrosio

geboren op 16 september 1990
te Jesolo, Italië

Promotoren:

Prof. dr. G.T. Barkema

Prof. dr. H.L. Bodlaender

Contents

| | Page |
|--------------------------------------------------------------------|-----------|
| 1 Introduction | 1 |
| 1.1 Graphite and graphene | 3 |
| 1.2 Synthesis challenges. | 5 |
| 1.3 The model | 7 |
| 1.4 Limits of the model | 10 |
| 1.5 Thesis outline. | 11 |
| 2 Structural relaxation of polycrystalline graphene | 13 |
| 2.1 Introduction | 17 |
| 2.2 Methods | 19 |
| 2.3 Numerical simulations and results | 25 |
| 2.4 Discussion and outlook | 32 |
| 3 Discontinuous evolution of polycrystalline graphene | 33 |
| 3.1 Introduction | 37 |
| 3.2 Model. | 38 |
| 3.3 Numerical simulations and results | 41 |
| 3.4 Discussion and outlook | 43 |
| 4 Dynamic sampling from a discrete probability distribution | 45 |
| 4.1 Introduction | 49 |
| 4.2 Problem Statement and Assumptions | 52 |
| 4.3 Data Structures. | 54 |
| 4.4 Multilevel Methods | 62 |
| 4.5 Experimental Analysis | 70 |
| 4.6 Discussion and outlook | 77 |

| | |
|------------------------------------------------------------|-----------|
| 5 An Event-Based model for polycrystalline graphene | 81 |
| 5.1 Introduction | 85 |
| 5.2 Methods | 90 |
| 5.3 Conclusions | 96 |

Backmatter

| | |
|-------------------------|------------|
| Bibliography | 99 |
| Summary | 105 |
| Samenvatting | 109 |
| Curriculum Vitae | 113 |

CHAPTER 1

Introduction

1.1 Graphite and graphene



Figure 1.1: Close up of a pencil, with its core in graphite clearly shown. ("Pencil tip macro" by GuySie is licensed under CC BY-SA 2.0).

Inside most pencils, we can find a core made of an inexpensive, useful and common material called *graphite*. Graphite is made (at least in theory) exclusively of atoms of carbon, it does not have a three-dimensional crystal structure (which diamond, another allotrope of carbon, has) and instead consists of crystalline layers stacked on each other and linked together by a weak Van der Waals force. Due to this layered structure, the properties of graphite vary along different directions; for instance, it conducts electricity and heat quite well along the layers but badly between the layers [2]. This anisotropy also permits the layers to easily slide over each other, allowing the use of graphite as a lubricant and inside pencils, where the friction between the paper and the graphite core is enough to shred layers of the material, leaving a dark grey trace.

Each of these layers is a two-dimensional crystal, called *graphene*. Its carbon bonding is sp^2 (trigonal) hybridised, unlike diamond which is sp^3 (tetrahedral) hybridised, and each atom is connected to three other atoms in a hexagonal lattice. This layered structure has been reported more than 160 years ago [3] and graphene, a single crystal layer, has been the focus of theoretical studies for more than seventy years [4–7], both as a way to describe properties of carbon-based materials and as

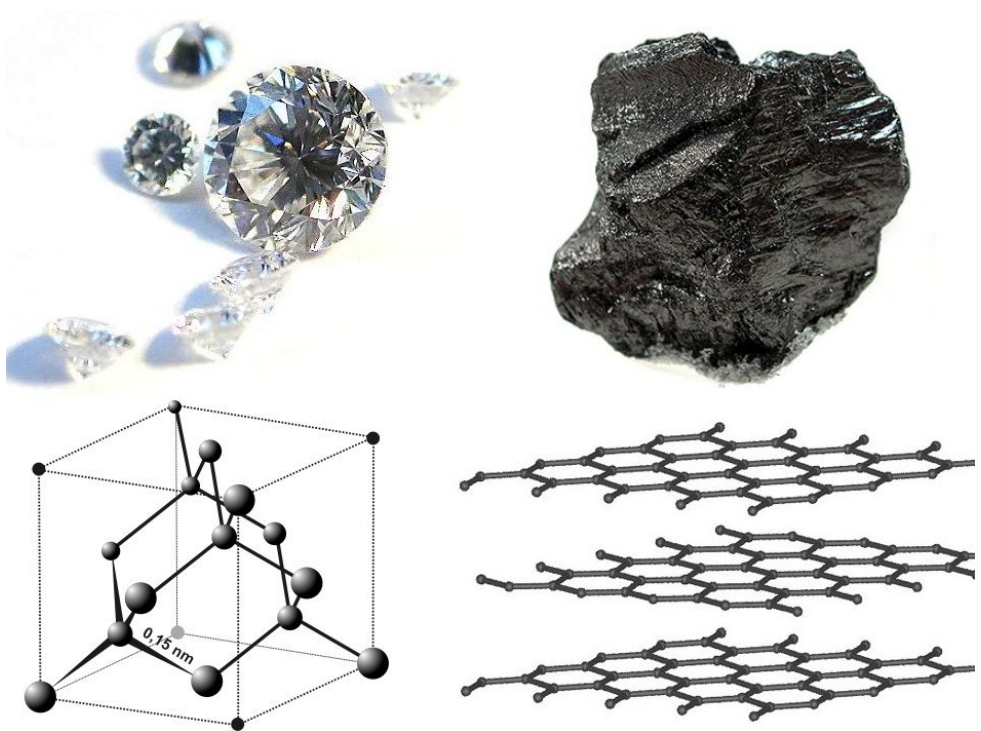


Figure 1.2: Structure of two allotropes of carbon. Diamond (left) has a three-dimensional crystalline structure given by the sp^3 hybridisation of the carbon bonding. Graphite (right) is composed of multiple two-dimensional crystalline layers (graphene) given by the sp^2 hybridisation of the carbon bonding, glued together by the weak Van der Waals interaction produced by a delocalised π orbital. ("File:Diamond and graphite2.jpg" by Itub, with derivative work by MaterialsScientist, is licensed under CC BY-SA 3.0).

a condensed-matter analogue of (2+1)-dimensional quantum electrodynamics [8–10]. At the same time, it was described as just an "academic" material [9] and two-dimensional crystals, in general, were believed to be thermodynamically unstable [11–13]. Quite unexpectedly, a single layer of graphene was isolated in 2004 by mechanical exfoliation with scotch tape [14], a discovery so remarkable that it was rewarded with the Nobel Prize in Physics just six years later, in 2010 [15].

Right after its isolation, graphene has immediately emerged as a remarkable material with surprising and unusual thermal, mechanical, optical and electronic properties [16–23]. As Geim, who shared the Nobel Prize with Novoselov for the isolation of graphene, summarises [17]:

Graphene is a wonder material with many superlatives to its name. It is the thinnest known material in the universe and the strongest ever measured. Its charge carriers exhibit giant intrinsic mobility, have zero effective mass, and can travel for micrometres without scattering at room temperature. Graphene can sustain current densities six orders of magnitude higher than that of copper, shows record thermal conductivity and stiffness, is impermeable to gases, and reconciles such conflicting qualities as brittleness and ductility.

Graphene and other 2D materials have repeatedly been predicted to revolutionise electronics and other industries, motivating not just scientists and engineers but also entrepreneurs and manufacturers [24]. Notwithstanding the intrinsic difficulties in synthesising graphene at a commercial scale [25], over 350 companies have been reported already producing related products [24, 26]. Graphene can now be found in batteries, printable electronics, photodetectors and both chemical and biological sensors [24]. New and innovative graphene products are expected to reach the market in this decade, such as biomedical technologies (for instance, biosensors, neural interfaces and drug delivery), electronics (for instance, low-cost printable electronics and flexible devices), sensors and imaging technologies (for instance, physical-chemical sensors, broadband CMOS cameras and spectrometers), telecommunication technologies and new graphene-based composite materials with different possible applications. [27].

1.2 Synthesis challenges

Nevertheless, there are currently still important issues with the production of graphene that require significant trade-offs between size, purity, yield and cost of graphene products [25]. In particular, the graphene that we are currently able to produce is in general polycrystalline (Figure 1.3), structural defects are extremely common [28] as confirmed by experimental observations [29–31], while some

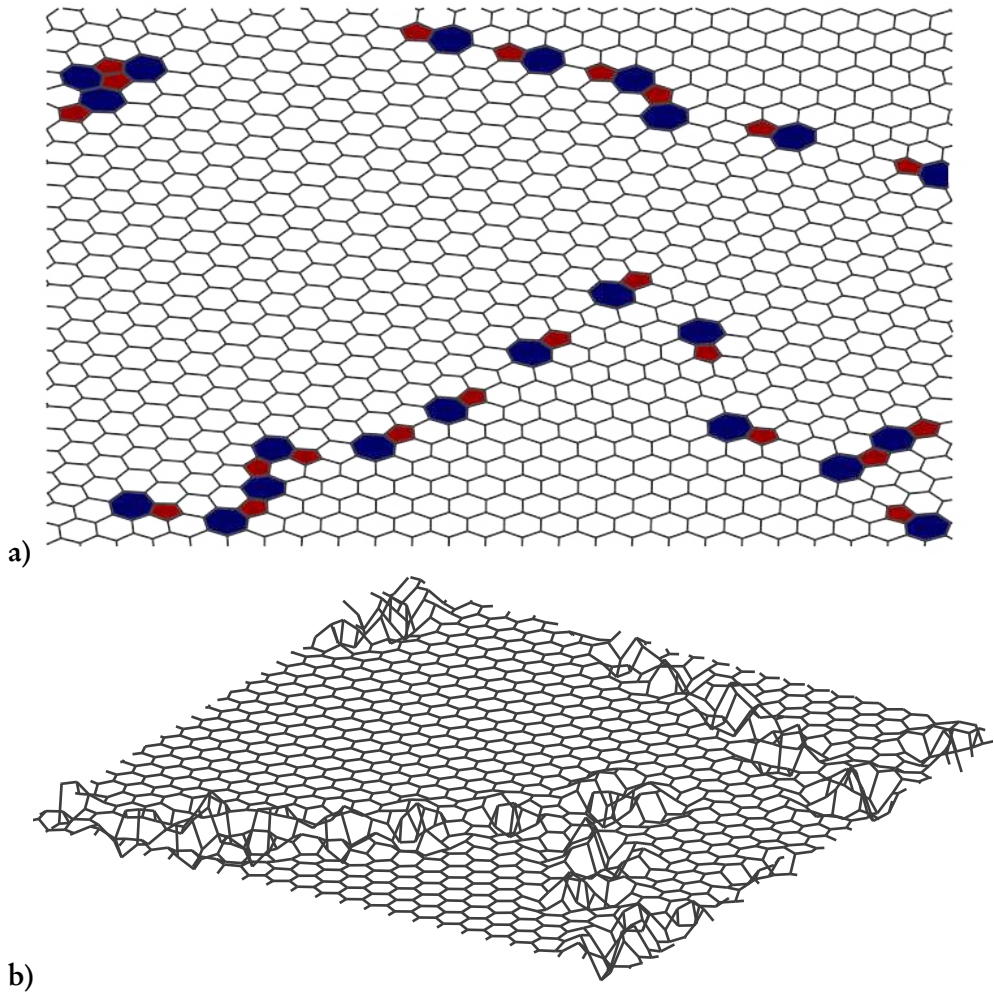


Figure 1.3: Polycrystalline graphene, in two (a) and three (b) dimensions, with an exaggerated z-axis. Note that the regions where the material is oriented in the same direction (crystalline domains) are separated by dislocation defects (in (a), pentagons are shown in red and heptagons in blue), and how the material buckles in the third dimension around them.

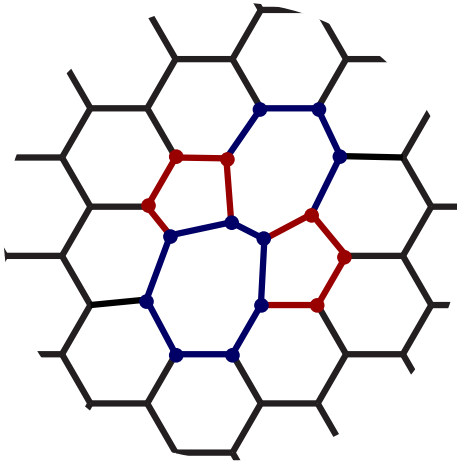


Figure 1.4: A Stone-Wales defect in graphene: four hexagons are replaced by two pentagons (red) and two heptagons (blue).

studies have been dedicated to their controlled production [32]. Defects have, of course, a negative effect on the properties expected from pristine graphene [33, 34] but they can also cause new effects that are otherwise absent [35–38]. Intrinsic defects (i.e. without foreign atoms) in particular, such as Stone-Wales (SW) [39] (Figure 1.4), are quite common since they can form both due to rapid quenching from a high temperature or irradiation [30].

While graphene is a two-dimensional material (i.e. a flat surface), it is embedded in the three-dimensional space that we all inhabit; atoms are therefore allowed to displace outside of the plane, in the third dimension [40]. When structural defects are present, the tension introduced in the material is relaxed via an out-of-plane buckling [28, 41, 42], as confirmed through electron microscopy observations [43, 44]. The study of the interaction between structural intrinsic defects and the collective behaviour of the material is therefore an interesting field of study to further our knowledge about this material.

1.3 The model

Amorphous materials, such as glasses [45] and amorphous silicon [46], have been studied with Continuous Random Network (CRN) models since their introduction almost 90 years ago by Zachariasen [45]. The strength of this type of model lies in its simplicity: the configuration of the sample is stored as a list of atoms, together with their position, and a list of the bonds between them. Since unsaturated carbon bonds are energetically very costly [28], we assume that each atom

is always perfectly coordinated. In the case of graphene, that means that all atoms are bonded to exactly three other atoms.

The atoms are placed on a rectangular box of size $L_x \times L_y$, while they are not limited in the third dimension. The rectangular box has periodic boundary conditions [47], to approximate an infinite system and prevent boundary effects, since from the point of view of an atom there are no boundaries. This is implemented with only a condition on the distance on the x - and y -component, without altering the coordinates of the atoms; if the distance along the x (or y) axis is more than $L_x/2$ ($L_y/2$), the size of the box is subtracted from it; if it is less than $-L_x/2$ ($-L_y/2$), it is added.

For this thesis, we have opted for the semi-empirical potential for polycrystalline graphene introduced by Jain et al. [48]:

$$E = \frac{3}{16} \frac{\alpha}{d^2} \sum_{i,j} (r_{ij}^2 - d^2)^2 + \frac{3}{8} \beta d^2 \sum_{j,i,k} \left(\theta_{j,i,k} - \frac{2\pi}{3} \right)^2 + \gamma \sum_{i,jkl} r_{i,jkl}^2. \quad (1.1)$$

The first term is the two-body term, which represents the tendency of the bonds to return to their ideal length $d = 1.42 \text{ \AA}$. The parameter $\alpha = 26.06 \text{ eV/\AA}$ is computed from density functional theory (DFT) calculations, the sum is over all bonds and r_{ij} is the length of the bond between the atoms i and j , after applying the previously described periodic boundaries.

The second term is the three-body term, which represents the tendency of the angles in the sample to return to the amplitude of angles in regular hexagons ($2\pi/3$ or 120°). The parameter $\beta = 5.511 \text{ eV/\AA}^2$ is computed from DFT calculations and the sum is over all the angles $\theta_{j,i,k}$ formed by the bonds j, i and i, k insisting on the same atom i .

The third term is the out-of-plane term, a term representing the tendency of planarity of the material. The parameter $\gamma = 0.517 \text{ eV/\AA}^2$ is computed from DFT calculations, $r_{i,jkl}$ is the distance between the atom i and the plane generated by its neighbours j, k, l . The sum is over all atoms.

This potential allows for graphene to buckle in the third dimension, albeit at an energetic cost, reproducing correctly the behaviour of polycrystalline graphene. It can of course be extended with other terms that might be useful in different settings. For instance, we can reproduce the interaction with a substrate on which the sample lies with a harmonic term

$$E_{\text{sub}} = K \sum_i z_i^2, \quad (1.2)$$

where the sum is over the distance of all atoms from the initial plane, which is placed conveniently at $z = 0$, and K is a constant set in such a way to constrain the buckling to the experimentally observed range of a given substrate. We can also strain the sample by adding a term

$$E_\sigma = -\sigma L_x L_y \quad (1.3)$$

with $L_x L_y$ the area of the periodic box and σ a parameter that allows the modeller to control the straining force on the sample.

To prevent biases in our sample, the simulation starts from a non-physical fully random configuration. The periodic box is initialised to a sensible size, slightly larger than a crystalline sample with the same number of atoms N , and $N/2$ points are randomly placed on it. We then identify the Voronoi vertices, points equidistant to at least three random points, and we place an atom on each of them. Each bond is the line made of points equidistant from the same couple of initial random points, provided there is no other random point closer than the two. The resulting diagram is perfectly coordinated (i.e. each atom is connected to three other atoms), although it is extremely chaotic and very far from a realistic configuration [48, 49].

While the semi-empirical potential does allow the sample to relax towards the energy minima given its structure (i.e. the list of bonds), it does not allow for changes in the bonds between atoms. The model discussed in this thesis uses the algorithm described by Wooten, Winer and Weaire (WWW) that allows for structural changes in the material while keeping detailed balance [46, 50]. In a WWW bond transposition, a string of four connected atoms is randomly selected and two of the bonds between them are switched. The system is then relaxed, usually through molecular dynamics, and the probability that this change is accepted is given by the Metropolis probability

$$P(\text{accept}) = \min \left\{ 1, \exp \left[\frac{E_b - E_a}{k_B T} \right] \right\}, \quad (1.4)$$

where E_b is the energy of the system before the bond transposition, E_a after the bond transposition, k_B the Boltzmann constant and T the temperature of the sample, a parameter set by the modeller. If a bond transposition reduces the total energy of the system (i.e. makes it more crystalline), it is always accepted, while one that does increase the energy of the system can still be accepted, albeit with a probability that decays exponentially with the energy difference. This way the system can go through sub-optimal metastates on its way to a more energetically convenient configuration. Through this process, our initial random sample evolves towards a more realistic polycrystalline configuration.

1.4 Limits of the model

After each attempted bond transposition, the system is relaxed through molecular dynamics, with a time scale that is dictated by the dynamics of the system; at each time step, the energy and the forces on all N atoms are recomputed, requiring a large number of calculations each. Thousands of these time steps are necessary to reach relaxation. To see the collective behaviour that we expect from polycrystalline graphene, we need a large system: N can be arbitrarily big. At this scale, the performance of the simulation becomes critical for any meaningful study therefore significant effort has been put into analysing and optimising its code. Van Dieten [51] wrote an optimised parallel implementation of this simulation, resulting in a performance boost of just under a factor 8 on an ordinary computer.

Once a sample has evolved to a somewhat realistic configuration, most of the attempted bond transpositions are rejected; it is quite normal for a well-relaxed sample to require hundreds or thousands of attempts before a further structural change is accepted. This is also a limit to how much we can allow the structure of a sample to relax, as the simulations grind to a halt for well-relaxed systems, rejecting almost all bond transpositions.

Barkema and Mousseau [52] designed a modified WWW algorithm for amorphous silicon that allows for the rejection of bond transpositions during the relaxation of the entire sample, arresting it once it becomes clear that it will not be accepted. Unfortunately, the assumptions necessary are not valid for polycrystalline graphene and a different algorithm is necessary to reduce the computational time dedicated to relaxing hopeless changes in the structure of the sample.

Alternatively, it would be tempting to get rid of the rejections altogether, where almost all the time is wasted on hopeless bond transpositions. After their rate (i.e. the expected number of realisations in a given unit of time) has been computed, we can treat the bond transpositions that are available in a given configuration as events and directly sample them one at a time.

We are faced with two challenges in implementing such an Event-Based model. In theory, after each change, all the pre-computed rates are invalidated as the initial state from which they are computed is no longer the current state of the system; the computational cost of this computation would be prohibitive for samples of non-trivial size. Even if we were able to limit this update to a limited fraction of all events, we still need a data structure that allows for both efficient sampling and updating (i.e. change in rate) of the events stored inside itself.

1.5 Thesis outline

In Chapter 2, we introduce two variations on the WWW algorithm where the final energy after the bond transposition is approximated by relaxing only the neighbours around the atoms involved in the structural change: the *early rejection*, which rejects hopeless moves; and the *early decision*, where a final decision on either acceptance or rejection is made.

In Chapter 3, we apply the computer model previously described to investigate the behaviour of polycrystalline graphene under stretching and show that it evolves in a discontinuous fashion: a tiny increase in the stretching force can lead to significant, avalanche-like displacement in the sample due to the collapse of the buckling in the sample, which causes vibrations in the material.

In Chapter 4, we consider different data structures for the problem of sampling from a dynamically changing discrete probability distribution (i.e. from a list of events with a given, but dynamic, rate), where some prior information is known on the distribution of the rates, in particular the maximum and minimum rate, and where the number of possible outcomes N is large.

In Chapter 5, we lay the groundwork for an Event-Based model for graphene, built on top of the results on the locality of the structural changes (Chapter 2) and data structures for dynamic sampling given some assumptions on the distribution of rates (Chapter 4).

CHAPTER 2

Efficient structural relaxation of polycrystalline graphene models

Abstract - Large samples of experimentally produced graphene are polycrystalline. For the study of this material, it helps to have realistic computer samples that are also polycrystalline. A common approach to produce such samples in computer simulations is based on the method of Wooten, Winer, and Weaire, originally introduced for the simulation of amorphous silicon. We introduce an *early rejection* variation of their method, applied to graphene, which exploits the local nature of the structural changes to achieve a significant speed-up in the relaxation of the material, without compromising the dynamics. We test it on a 3,200 atoms sample, obtaining a speedup between one and two orders of magnitude. We also introduce a further variation called *early decision* specifically for relaxing large samples even faster and we test it on two samples of 10,024 and 20,000 atoms, obtaining a further speed-up of an order of magnitude. Furthermore, we provide a graphical manipulation tool to remove unwanted artifacts in a sample, such as bond crossings.

This chapter is partially based on the following publication:

F. D'Ambrosio, J. Barkema, and G. T. Barkema, "Efficient structural relaxation of polycrystalline graphene models", *Nanomaterials* **11**, 1242 (2021).

2.1 Introduction

Graphene is a crystal of carbon atoms that form a three-coordinated honeycomb lattice. It is a material with a large set of exotic properties, both mechanical and electronic, and it has the particularity of being a two-dimensional crystal embedded in a three-dimensional space [16–23]. Large samples experimentally produced are usually polycrystalline, containing intrinsic [40, 54, 55], as well as extrinsic [56] lattice defects. These defects warrant a thorough study as they both have a significant detrimental effect on the properties expected from pristine graphene [33, 34], and they can also cause new effects that are otherwise absent [35–38].

In particular, structural defects are both prominent and common in graphene [28], as they can easily host lattice defects due to the flexibility of the carbon atoms in hybridization. Such defects can be frozen in the sample during the annealing process and have been experimentally observed [29–31]. Their controlled production in graphene has been explored [32].

Since unsaturated carbon bonds are energetically very costly [28], polycrystalline graphene samples can be studied with the use of continuous random networks (CRN) models [48], introduced by Zachariassen almost 90 years ago to represent the lack of symmetry and periodicity in glasses [45]. The rules of this type of model are quite simple: the only requirement is that each atom is always perfectly coordinated, i.e. their bonding needs are fully satisfied. Wooten, Winer, and Weaire (WWW) introduced an explicit algorithm to simulate the evolution of samples of amorphous Si and Ge, the so-called WWW algorithm that became the standard for this kind of model [46, 50]. In the WWW approach, a configuration consists of a list of the coordinates of all N atoms, coupled with an explicit list of the bonds between them.

We opted for the empirical potential for polycrystalline graphene recently proposed by Jain et al. [48]:

$$E = \frac{3}{16} \frac{\alpha}{d^2} \sum_{i,j} (r_{ij}^2 - d^2)^2 + \frac{3}{8} \beta d^2 \sum_{j,i,k} \left(\theta_{j,i,k} - \frac{2\pi}{3} \right)^2 + \gamma \sum_{i,jkl} r_{i,jkl}^2 \quad (2.1)$$

with r_{ij} the distance vector between the atoms i and j , $\theta_{j,i,k}$ the angle centered on the atom i between the atoms j and k , $r_{i,jkl}$ the distance between the atom i and the plane described by its neighbors j, k, l , and $d = 1.420 \text{ \AA}$ the ideal bond-length of graphene. The other parameters, extracted from DFT calculations [48], are $\alpha = 26.060 \text{ eV/\AA}^2$, $\beta = 5.511 \text{ eV/\AA}^2$ and $\gamma = 0.517 \text{ eV/\AA}^2$. The interaction

with the substrate on which the sample lays is simulated by an harmonic confining energy term in our potential

$$E_c = K \sum_{i=1} z_i^2 \quad (2.2)$$

where z_i is the z-coordinate of the atom with index i and K a prefactor that is determined empirically in order to constrain the maximum buckling height to the range of 4-8 Å, experimentally observed with scanning tunneling microscopy (TEM) [55].

The process starts with a completely random 2D sample with all atoms perfectly coordinated, in the case of graphene threefold connected, generated with the Voronoi diagram algorithm described in [48]. In order to generate an initial configuration with N atoms, we place $N/2$ random dots in a 2D square box, which is then surrounded by 8 copies of itself to implement periodic boundary conditions. We then compute the N vertices of the Voronoi diagram [49] of these random dots, which will be replaced by atoms, and connect them along the edges of the diagram to form the bonds by them. This highly-energetic configuration is then carefully relaxed with molecular dynamics.

The structure of the sample evolves through a series of bond transpositions involving four connected atoms with two bonds that are broken to create two new bonds. After each bond transposition, the system is relaxed; the move is accepted according to the Metropolis acceptance probability [57, 58]:

$$P(X'|X) = \min \left\{ 1, \exp \left[\frac{E(X) - E(X')}{k_b T} \right] \right\} \quad (2.3)$$

where X and X' are the configurations of the system respectively before and after the bond transposition, both the coordinates and the list of bonds. k_b is the Boltzmann constant, T is the temperature, and $E(Y)$ the energy of the configuration Y after complete relaxation. Relaxing the sample, even with an optimised molecular dynamics algorithm such as the FIRE algorithm [59], has a significant computational cost, which is wasted if the bond transposition is ultimately rejected. As the energy of the sample is gradually lowered through bond transpositions, the accepted ratio becomes smaller, often well below one per cent, and almost all computational time is wasted on proposed bond transpositions that are eventually rejected.

Barkema and Mousseau [52] developed a method for amorphous silicon, that al-

lows the early rejection of bond transpositions before completing the relaxation of the sample. It generates a stochastic energy threshold beforehand, given by

$$E_t = E_b - k_b T \ln(s) \quad (2.4)$$

with s a random number between zero and one. In the first ten relaxation steps, the sample is relaxed only locally up to the third neighbour shell. The energy is assumed to be harmonic around the minimum, therefore the final energy can be approximated as proportional to the square of the force

$$E(X') \approx E - c_f |F|^2 \quad (2.5)$$

with c_f an empirically determined constant and F the force vector. Once we are close enough to the minimum, we can immediately reject the bond transposition if, at any moment during the relaxation, $E - c_f |F|^2 > E_t$. The efficiency of this method is dependent on the quality of the assumption Eq. (5.11); for amorphous silicon, the type of model for which it has been developed, this approximation is generally valid after just a few relaxation steps.

In theory, this approach could also be applied to polycrystalline graphene. Unfortunately, the harmonic approximation of Eq. (5.11) is only valid very close to the minimum; as we show in Figure 2.1, the trajectory of the system in the phase-space fluctuates rapidly and erratically during the relaxation, instead of following the expected linear relation between the excess energy and squared force magnitude after a certain number of relaxation steps. Without this approximation, a very costly full relaxation is necessary after each attempted bond transposition. A different approach is needed.

In this work, we propose a new method where only the atoms up to a shortest-path distance l from the atoms involved in the bond transpositions are initially allowed to relax. The energy of the sample after this local relaxation is then used to predict the final energy and immediately reject hopeless bond transpositions, without requiring a full relaxation. We test this approach on a 3200 atoms sample, comparing the performance for different values of l . The quality of the results is also compared to those obtained only through global relaxation. We further propose a variation of this method for relaxing large samples and we test it by generating and relaxing a 20,000 atoms random sample.

2.2 Methods

The initial configuration of the sample is a disordered, perfectly three-fold coordinated, and two-dimensional random network. It is generated following the proce-

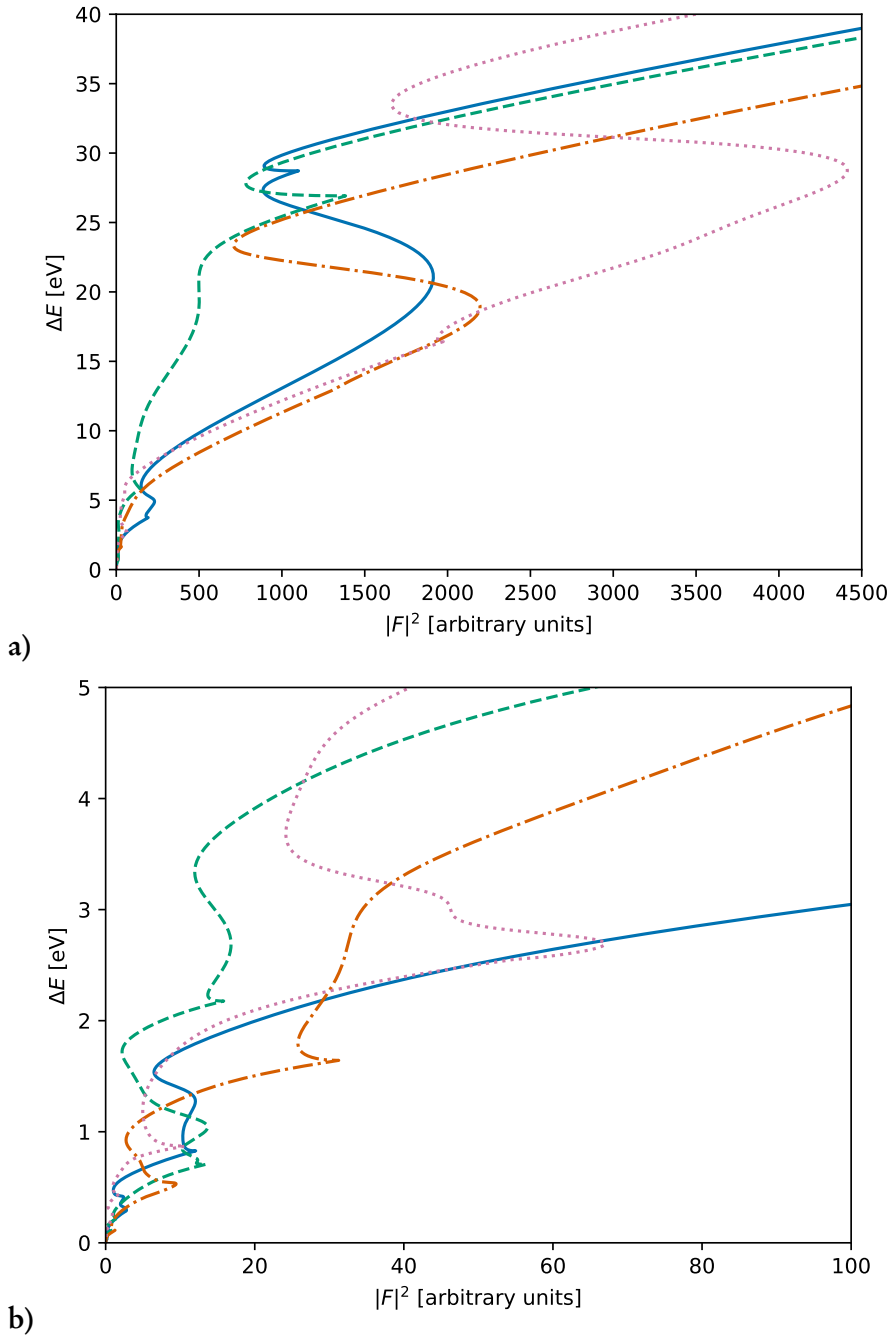


Figure 2.1: Some typical relaxation trajectories of a 3200 atoms sample, farther **a)** and closer **b)** to the origin. Even after thousands of iterations, the approximation of Eq. (5.11) cannot be applied to this system as it fluctuates rapidly in the phase space. ΔE is the energy difference with the relaxed (final) energy, $|F|^2$ the magnitude of the forces.

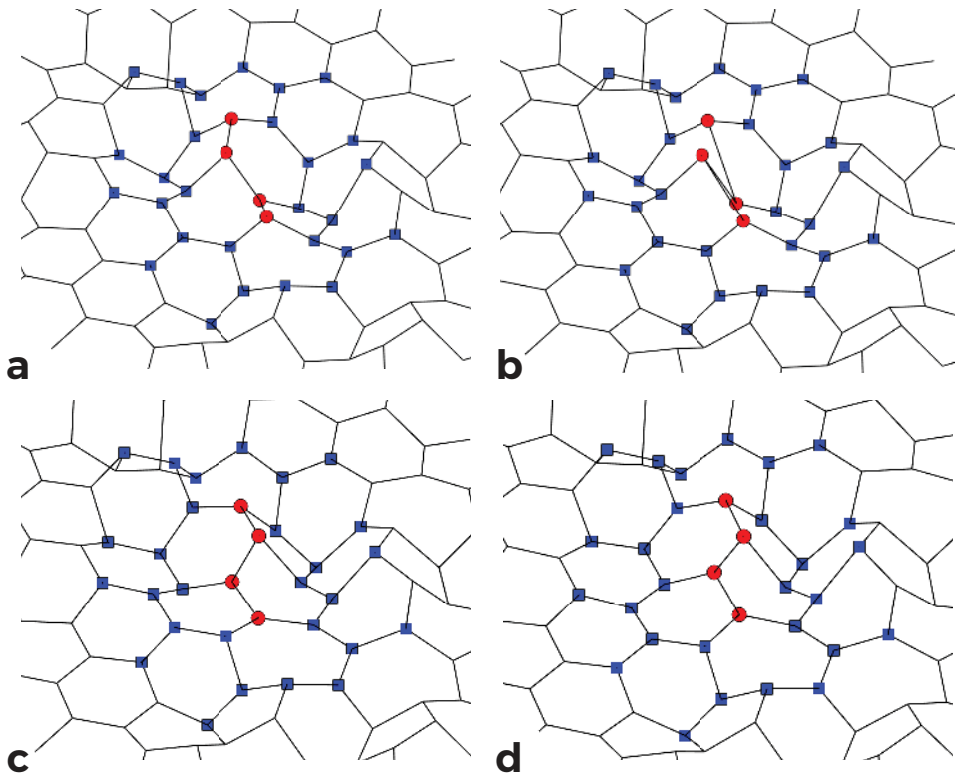


Figure 2.2: Successful bond transition on a sample of graphene: **a)** initial configuration **b)** bond transition, atoms involved are marked with red dots **c)** local relaxation, atoms involved are marked with blue squares **d)** final configuration after global relaxation.

ture described in [48].

The coordinates of the sample are relaxed to an energy minimum with molecular dynamics, following the FIRE technique [59]. After setting a temperature lower than the melting point of graphene, several bond transpositions are performed until it reaches reasonably low energy and a realistic configuration. Once this flat sample is sufficiently relaxed, every atom is placed at a random non-zero distance out of the two-dimensional plane and allowed to relax to a buckled three-dimensional configuration.

Our approach to the structural relaxation of graphene can be followed in Figure 2.2. Four consecutive atoms are randomly selected (Figure 2.2a) and the bonds between the first two and the last two are transposed (Figure 2.2b). The energy threshold is computed from Eq. (2.4) and we perform a local relaxation around the four atoms involved in the bond transposition; instead of limiting it to a certain number of relaxation steps, the atoms up to a shortest-path distance l from the transposed bonds are allowed to relax completely (Figure 2.2c). The list of atoms involved in the local relaxation is computed after each attempted bond transposition by iteratively exploring the network, starting from the four atoms involved in the bond transposition, and checking against duplicates. After local relaxation, attempted bond transpositions for which $E_l(X') - c_f|F|^2 > E_t$, with $E_l(X')$ the energy of the sample after local relaxation up to distance l , are immediately rejected. In contrast with the method from [52], the criterion is applied only once, instead of at each point of the relaxation (with possibly some upper bound on the force strength). As we note in Figure 2.3, the force strength after local relaxation (Figure 2.3b) is a good estimator for the final energy, especially in comparison to the force strength during the global relaxation (Figure 2.3a), which is used by the method in [52].

2.2.1 Early rejection

In the *early rejection* approach, the whole sample is otherwise allowed to relax (Figure 2.2d) and, if $E(X') < E_t$, the bond transposition is finally accepted. As less than one per cent of proposed moves are accepted in a relaxed sample, we expect the speed-up to be significant: most are rejected after relaxing a limited number of degrees of freedom.

The value of c_f is fine-tuned from empirical data collected from the simulation itself, targeting a higher bound on the rate of false negatives (i.e. bond transposition that are rejected erroneously), which in this work was fixed at 2% of the total number of attempts that should have been accepted. No transposition is accepted without complete relaxation, regardless of the result of the local minimization;

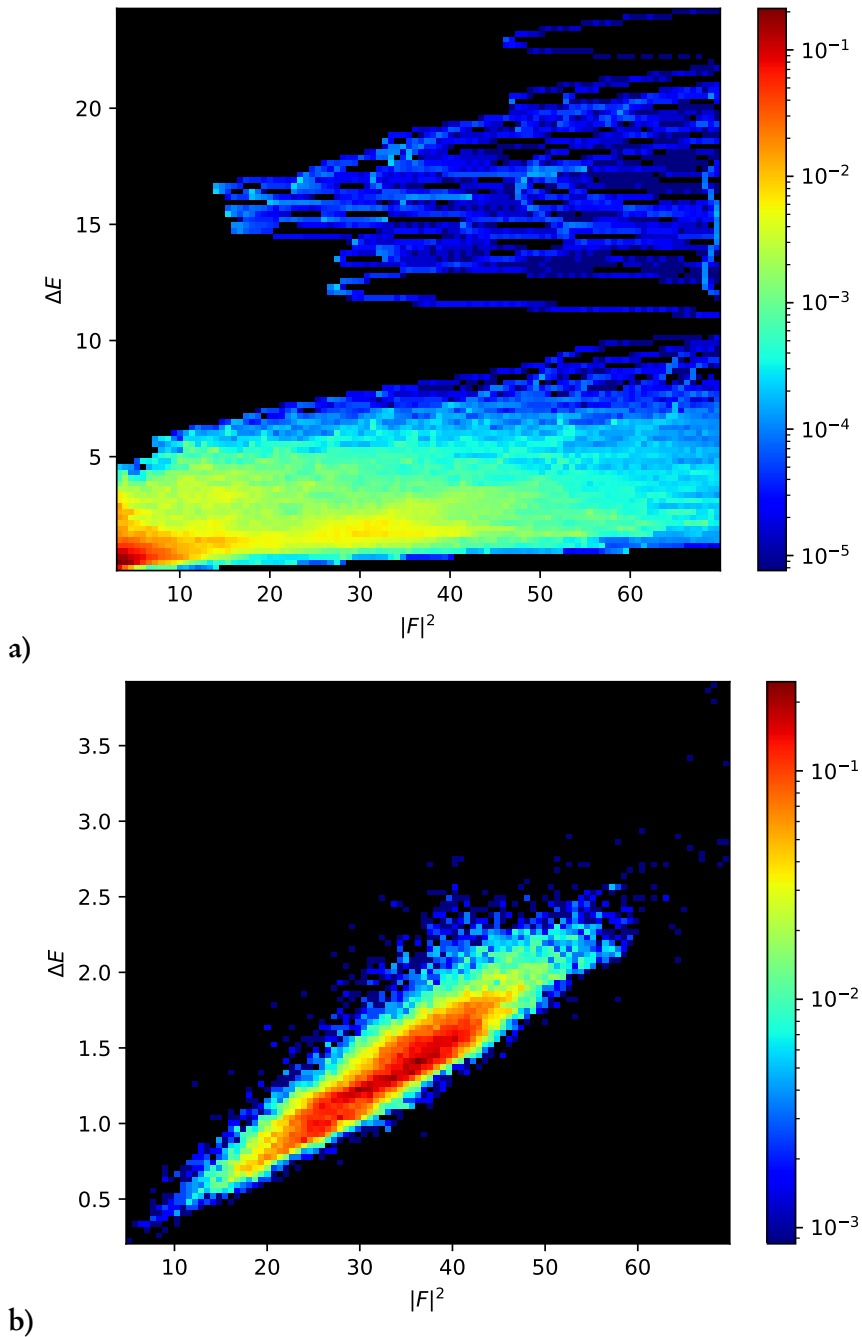


Figure 2.3: Two dimensional histograms of the energy difference with the relaxed (final) energy ΔE and the force strength $|F|^2$ over many relaxations of a 3200 atoms sample, from the values assumed during global relaxation, **a)**, and at the end of local relaxation, **b)**. Frequency is show in logarithmic scale. We can see that the values assumed at the end of local relaxation follow the harmonic approximation of Eq. 5.11.

therefore, no false positive (i.e. a bond transpositions is accepted erroneously) can be introduced by this technique.

2.2.2 Early decision

While the computational time required for each local relaxation is constant with regards to the size of the sample, the same cannot be said for the global relaxation after each tentatively accepted bond transposition. For large samples, due to the amount of computational time that this requires, the structural relaxation still grinds almost to a halt. This is particularly problematic for the initial structural relaxation of a large random sample, which requires a large number of bond transpositions to reach a realistic, more relaxed state.

We propose as an alternative for such cases the *early decision* approach: the decision on whether to reject or accept a bond transposition after the local relaxation is treated as final, without having to perform a global relaxation to accept it. The parameter c_f is still fine-tuned from empirical data but in this case, we opt for the value that best fits it. After a successful bond transposition, the system will not reach the energy it would have reached with global relaxation and the forces on atoms outside those involved in the last local relaxation will not go to zero. To correct for this issue, the energy threshold for accepting a bond transposition, see Eq. (2.4), will be computed replacing the current energy of the system (E_b) with an estimation of the energy that our current configuration would reach after a global relaxation according to Eq. (5.11). It can also be useful to set an upper value for the magnitude of the forces that, when reached, will trigger a global relaxation that will stop when the magnitude of the forces is comparable to those that are leftover after a single bond transposition, to reduce the time spent on these occasional global relaxations.

It must be noted that since we are replacing the energy of the relaxed system after a bond transposition in Eq. (2.3) with an estimate, the *early decision* method does not guarantee detailed balance, as opposed to the *early rejection* method. Nevertheless, this method is extremely powerful when performance is more critical than accuracy, for instance for the structural relaxation of a very large randomly generated sample when it is still far away from equilibrium. In these cases, detailed balance is not as critical and a large number of bond transpositions are required to reach a state closer to the equilibrium.

2.2.3 Manipulation tool

The initial random configuration can incorporate artifacts such as two bonds crossing each other. While in most cases these defects will gradually disappear as the sample relaxes to a more ordered configuration, some artifacts might be particu-

| l | $c_f [s^2 u^{-1}]$ | $\langle N_{loc} \rangle$ |
|-----|----------------------|---------------------------|
| 1 | $3.21 \cdot 10^{-3}$ | 13 |
| 2 | $4.63 \cdot 10^{-3}$ | 28 |
| 3 | $5.33 \cdot 10^{-3}$ | 53 |
| 4 | $9.08 \cdot 10^{-3}$ | 90 |

Table 2.1: Empirically determined values of the harmonic coefficient c_f and average number of atoms involved in local relaxation $\langle N_{loc} \rangle$ for different local relaxation distances l , in a sample of size $N = 3200$ atoms.

larly resilient and can persist even when the sample is otherwise sufficiently relaxed. Such defects have to be removed manually. We have developed a graphical tool called Graphene Editor ¹ to facilitate this work. This tool allows the user to upload and download a sample, explore it visually, add and remove bonds, move one or more atoms, replace a single atom with three connected atoms and vice-versa and check the consistency of the sample over the number of bonds for each atom and bond crossings.

2.3 Numerical simulations and results

2.3.1 Early rejection

A random sample with $N = 3200$ atoms was generated following the procedure described in the previous section. WWW bond transpositions are performed until the sample is relaxed to reasonably low energy, approximately 625 eV (less than 0.2 eV/atom). The values of c_f , seen in Table 2.1, for different values l of the local relaxation radius are chosen empirically, with the constraint of keeping the ratio of false negatives (successful bond transpositions that are nevertheless rejected) over successful bond transpositions under 2%, while still rejecting a large part of unsuccessful moves. The quantity c_f is expressed in units of seconds squared over the atomic mass unit. The average number of atoms involved in the local relaxation for different values of l is also shown in Table 2.1.

Starting from the same initial sample, we perform bond transpositions both using the usual WWW algorithm with full minimization and the *early rejection* method proposed here, with different values of l . The temperature is set to $T = 3000$ K for both samples. After each successful bond transposition, we record the energy, the elapsed time in CPU clocks, and the number of attempts since the last successful move. The simulation is stopped once the system reaches a final energy

¹Available at <https://github.com/jorisBarkema/Graphene-Editor>

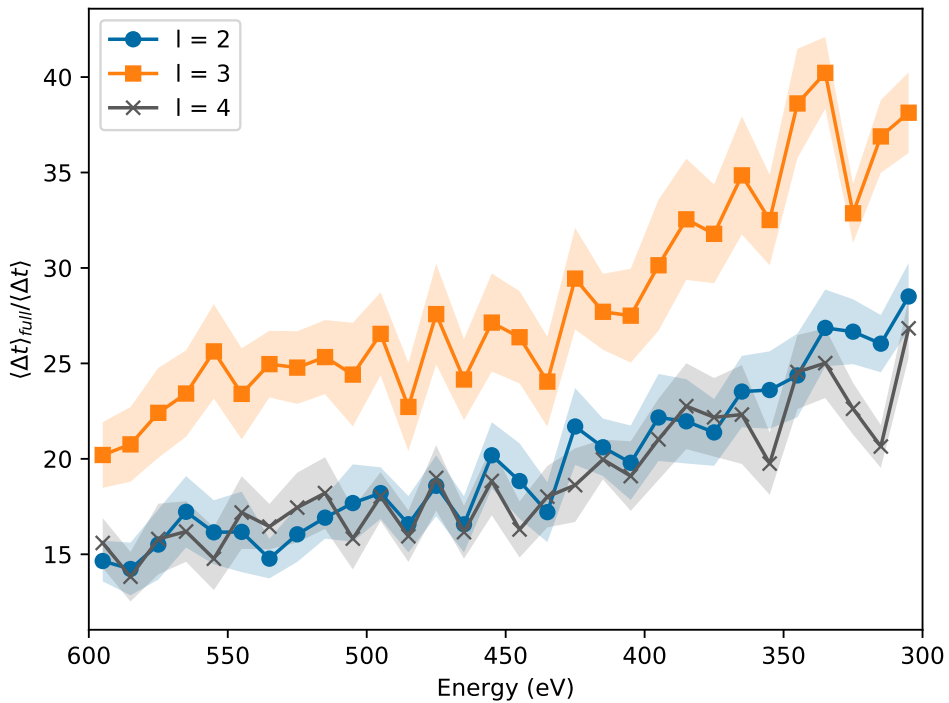


Figure 2.4: Average speed-up per accepted bond transposition, as ratio between CPU time required with full relaxation and *early rejection*, for different values of l : blue dots (2), orange squares (3) and grey crosses (4). The shaded area shows one standard deviation from the average. The speed improvement grows as the sample becomes more crystalline and its energy lowers, while best results are obtained for $l = 3$, with an improvement of a factor between 20 and 40.

| Atoms | Full relaxation | | Early rejection | |
|-------|-----------------|-------|-----------------|-------|
| Size | # | % | # | % |
| 5 | 56 | 3.50 | 60 | 3.75 |
| 6 | 1489 | 93.06 | 1480 | 92.50 |
| 7 | 54 | 3.38 | 60 | 3.75 |
| 8 | 1 | <0.01 | 0 | 0.00 |

Table 2.2: Ring statistics for the two final configurations of the 3200 atoms sample, relaxed with full relaxation (left) and Early Rejection (right). We note that they both have reached similar statistics, with around 93% of the rings being hexagons, 3 – 4% heptagons and pentagons, while octagons are too rare at this energy to compare between the two.

of $E_f = 200\text{eV}$, equivalent to 0.0625eV/atom . At least ten relaxation cycles are performed with the *early decision* method (with different values of l) and with complete relaxation after each bond transposition. As we note in Figure 2.4, the average CPU time per accepted bond transposition is improved by at least an order of magnitude. The speed-up grows as the sample grows larger crystalline domains and more random attempts are necessary per accepted bond transposition. Best results are obtained for $l = 3$, which leads to an efficiency improvement of a factor between 20 and 40.

The *early rejection* method does not alter the amount of relaxation obtained at the end of the process. As we note in Figure 2.5, both the level of separation between crystalline domains, i.e. the degree to which the defects are present on the borders between them, and the size of the domains are consistent. The ring statistics of the two final configurations, computed with the Ring Statistics Algorithm² [60] and reported in Table 2.2, are also consistent. In this final configuration, the ratio of false negatives is lower than 0.5%.

2.3.2 Early decision

As we noted in the previous section, while the *early rejection* technique is quite powerful for most samples, it is insufficient for very large samples; our attempt to relax a very large sample ($N = 20,000$) could not reach our initial energy target of 1eV/atom after more than a month, due to the computational time required by each global relaxation that takes place at least once per accepted bond transposition. In the *early decision* method, the decision on whether to accept a bond transposition or not takes place directly after performing a local relaxation, based on the estimated relaxed energy of the sample.

²<https://github.com/vitroid/CountRings>

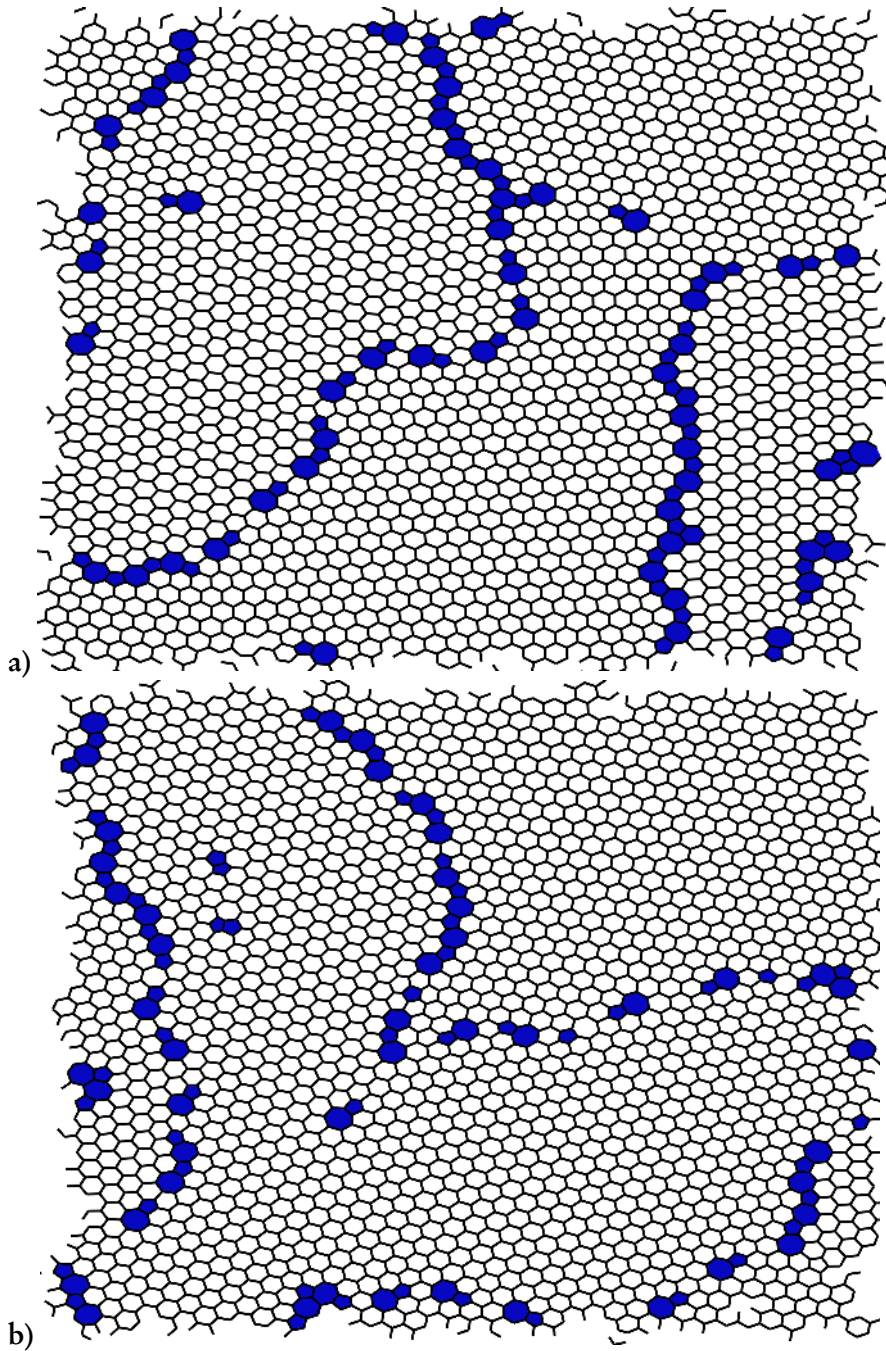


Figure 2.5: Final configurations of the sample at $E \approx 200$ eV obtained a) through only global relaxations b) through our local relaxation method with $l = 3$. Highlighted in blue are the defective (i.e. non-hexagonal) rings. The two samples are qualitatively indistinguishable: same level of separation between crystalline domains of similar sizes.

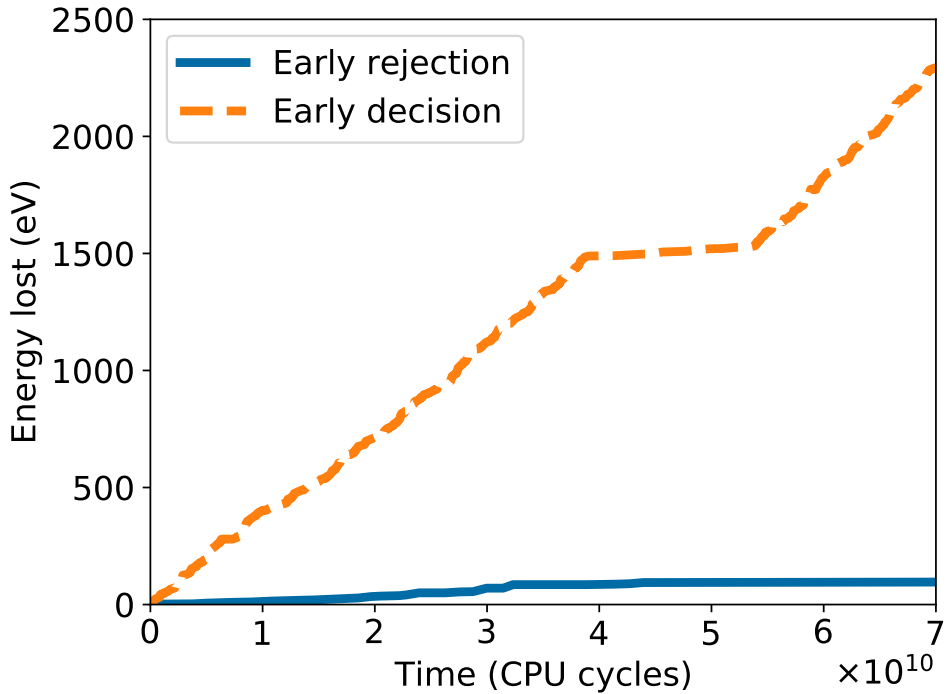


Figure 2.6: Structural relaxation of a large, randomly generated sample ($N = 20,000$), starting from an energy of 1.15 eV/atom with *early rejection* (blue solid line) and with *early decision* (orange dotted line) methods. The *early decision* method performs significantly faster, reaching a speed-up of a further order of magnitude. We also notice a plateau around $4 \cdot 10^{10}$ CPU cycles in the *early decision* line where, due to the forces accumulated from previous bond transpositions, our algorithm was incorrectly rejecting bond transpositions, slowing the evolution of the sample significantly.

| Atoms | 10,024 | | 20,000 | |
|-------|--------|-------|--------|-------|
| Size | # | % | # | % |
| 5 | 231 | 4.6 | 487 | 4.87 |
| 6 | 4554 | 90.86 | 9043 | 90.43 |
| 7 | 223 | 4.45 | 453 | 4.53 |
| 8 | 4 | 0.08 | 17 | 0.17 |

Table 2.3: Ring statistics for the two large samples of 10,024 and 20,000 atoms. We note that they both have reached similar statistics, with over 90% of the rings being hexagons, 4 – 5% heptagons and pentagons and less than 0.2% octagons.

We relaxed with both approaches a randomly generated sample of 20,000 atoms. We opted again for $l = 3$ for the local relaxation and c_f is set after fitting the data from $\tilde{100}$ global relaxations. The force magnitude thresholds are set in such a way that a global relaxation should be triggered each 50-100 successful bond transpositions and stop when the force magnitude reaches a value comparable with what is usually left after just one local relaxation. The temperature is set to $T = 3000$ K.

We initially performed the relaxation on a sample with energy of 1.15 eV/atom. As we can see in Figure 2.6 the *early decision* approach leads to a significant speed-up that we estimate to be around one further order of magnitude. The speed-up factor per bond transposition is stable during the relaxation at approximately 22. Both methods accept on average a bond transposition every seven attempts, but the *early decision* method is, as expected, less stable: there can be phases where it is not able to correctly estimate the correct decision to take. In these extreme cases, bond transpositions are erroneously rejected and the evolution of the sample slows down. This is especially the case when the magnitude of forces accumulated from previous bond transpositions become significantly large. We can see such a case in the plateau of the orange dotted line in Figure 2.6 and it underscores the importance of setting a correct threshold for the magnitude of forces accumulated before triggering a global relaxation.

Finally, we relaxed the 20,000 atoms sample and another sample of 10,024 atoms down to 1488.05 eV (0.074 eV/atom) and 695.51 eV (0.066 eV/atom) respectively. The temperature is initially set at 3000 K and then gradually reduced, in order to reach lower energies. The resulting samples, as we note in Figure 2.7, present large crystalline domains with defects accumulating on their boundaries, similarly to 2.5. As we note in Table 2.3, both samples have reached similar ring statistics, with less than 10% of defected rings and only a handful (less than 0.2%) defected by more than one atom (i.e. octagons). The ring statistics is computed with the

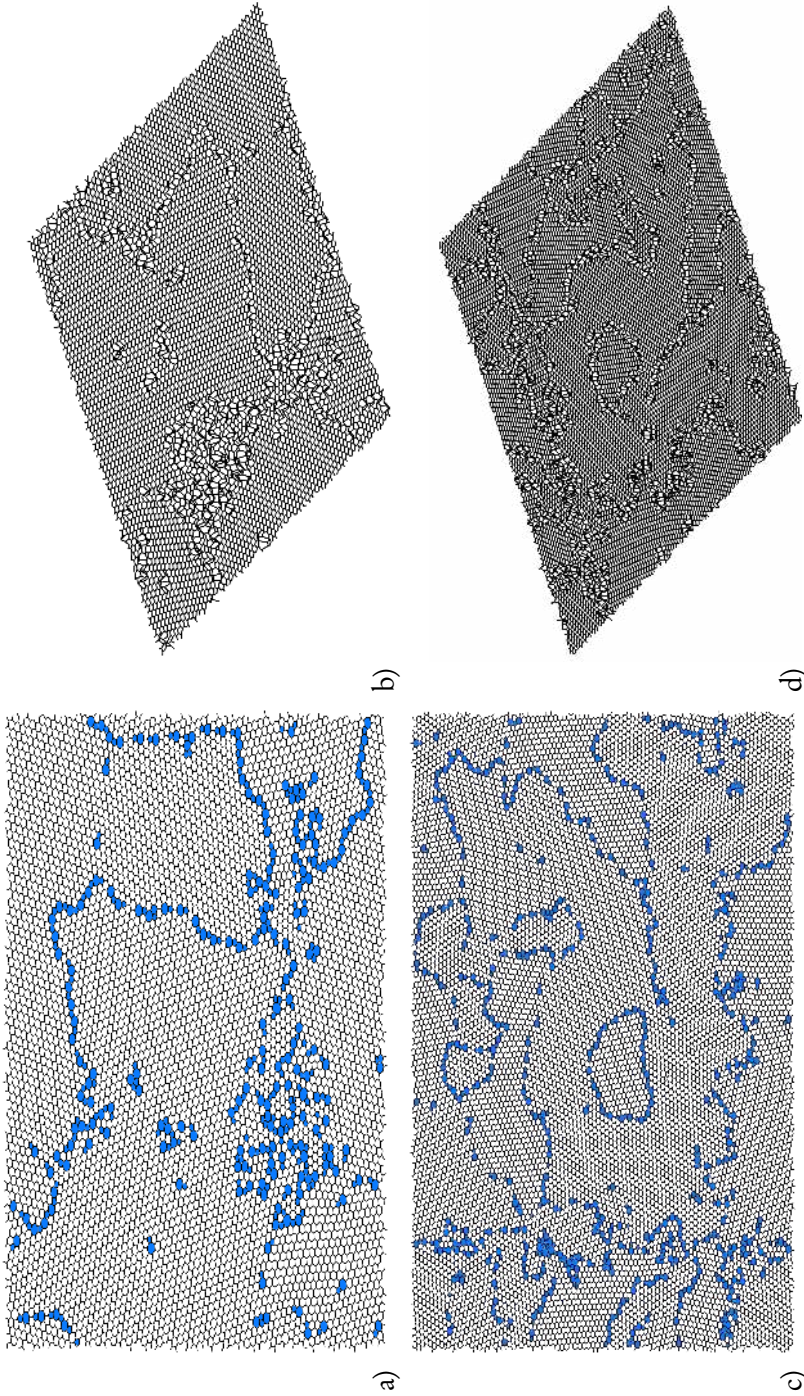


Figure 2.7: Final configurations of the 10,024 atoms sample in two (a) and three (b) dimensions and the 20,000 atoms sample in two (c) and three (d) dimensions, obtained through *early decision* local relaxation with $l = 3$. Defects (i.e. non-hexagonal rings) are highlighted in blue in the two-dimensional plots and clearly visible due to the buckling in the three-dimensional plots. The two samples are qualitatively very similar to those of Figure 2.5, with large domains surrounded by defects.

Ring Statistics Algorithm [60].

All the samples presented in this chapter are available online³.

2.4 Discussion and outlook

In summary, we introduced two techniques that, through local relaxation, can estimate the success of bond transpositions reducing or eliminating the need for relaxing the entire sample, which is extremely time-consuming. Both techniques significantly reduce the computational time required per accepted bond transposition: the *early rejection* method by immediately rejecting, without a global relaxation, hopeless attempts; the *early decision* method avoids global relaxations entirely, relying on the estimate of the energy of the relaxed sample.

The *early rejection* technique should be preferred for average-sized samples, especially if already well-relaxed since it gives an already significant speed-up while it guarantees that the dynamics are not compromised. Furthermore, its accuracy also improves as the energy of the sample is reduced. The *early decision* technique leads to an even larger speed-up but does allow for attempts to be erroneously rejected and should therefore be used when performance is a priority above accuracy, for instance when the sample is still very far from equilibrium and detailed balance is less critical. Since thousands of bond transpositions are required to reduce the energy of a few hundreds of electron volt, the cumulative speed-up obtained through either of these techniques can easily reach multiple orders of magnitude. These techniques open up the possibility of generating larger random samples with ordinary computers in an affordable amount of time.

Finally, our *manipulation tool* Graphene Editor makes those small manipulations that are often necessary as simple and quick as they can be.

³<https://github.com/federicodambrosio/graphene-samples>

CHAPTER 3

Discontinuous evolution of the structure of stretching polycrystalline graphene

Abstract - Polycrystalline graphene has an inherent tendency to buckle, i.e., develop out-of-plane, three-dimensional structure. A force applied to stretch a piece of polycrystalline graphene influences the out-of-plane structure. Even if the graphene is well relaxed, this happens in nonlinear fashion: Occasionally, a tiny increase in stretching force induces a significant displacement, in close analogy to avalanches, which in turn can create vibrations in the surrounding medium. We establish this effect in computer simulations: By continuously changing the strain, we follow the displacements of the carbon atoms that turn out to exhibit a discontinuous evolution. Furthermore, the displacements exhibit a hysteretic behaviour upon the change from low to high stress and back. These behaviours open up another direction in studying dynamical elasticity of polycrystalline quasi-two-dimensional systems, and in particular the implications on their mechanical and thermal properties.

This chapter is partially based on the following publication:

F. D'Ambrosio, V. Juričić, and G. T. Barkema, “Discontinuous evolution of the structure of stretching polycrystalline graphene”, *Phys. Rev. B* **100**, 161402 (2019).

3.1 Introduction

Graphene, a crystal of carbon atoms arranged in a honeycomb lattice, shows a plethora of exotic mechanical and electronic properties, and emerged as a paradigmatic example of a crystalline membrane embedded in the three-dimensional space [16–23]. In particular, it exhibits an intrinsic tendency to spontaneously buckle when it is polycrystalline, i.e, when it features many crystalline domains, or due to the presence of lattice defects, such as disclinations and Stone-Wales defects [28, 41, 42]. This effect arises due to the competition of stress, introduced either by mismatch between the domains or by the defects, and the tendency of the membrane to relieve the stress by bending, which yields a rather rich landscape of configurations for a relaxed membrane [61]. In addition, a graphene sheet can experience external stress, for instance when graphene is held by clamps which exert a pulling force, which further enriches the landscape of the ground state configurations.

It is well known how a polycrystalline or defected graphene sheet relaxes when subjected to a constant, static external stress [62], but the case of dynamic strain remained rather unexplored. The latter case can be experimentally relevant as external perturbations creating strain are in reality time-dependent. In particular, when external stress is gradually applied to a polycrystalline graphene sample, it is of a fundamental and practical importance to establish whether the change of the shape of the graphene membrane is continuous or it follows a discontinuous path in this rather complex configuration space.

In this chapter, by performing computer simulations, we show that the evolution of the shape of a polycrystalline graphene membrane is discontinuous: as the stress is uniformly increasing, occasionally a tiny increase in stretching force induces a significant displacement, analogous to avalanches, see Figure 3.2 and Supplemental Material online¹ for a video of the evolution of the sample due to the external stress that highlights this avalanche-like behaviour, which, as a result, can create vibrations in the surrounding medium. Furthermore, if the stretching force is then decreased again to its starting value, the system does not follow the same path in the configuration space, but rather exhibits a hysteretic behaviour while undergoing a cycle in the configuration space, see Figure 3.1. The change of the profile of the sample takes place through the creation and annihilation of ridges and vertices, with the elastic energy concentrated in these defects, as shown in Figure 3.3. Our findings should have implications for the dynamical elasticity in polycrystalline and defected sheets of graphene, and, in particular, we expect that various elastic moduli, phonon density of states, and thermal conductivity will be affected.

¹<https://journals.aps.org/prb/supplemental/10.1103/PhysRevB.100.161402/>

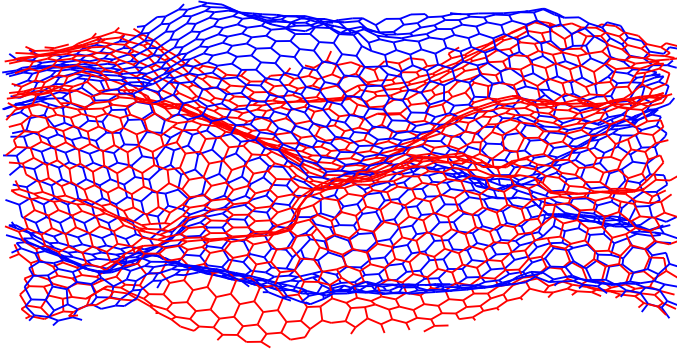
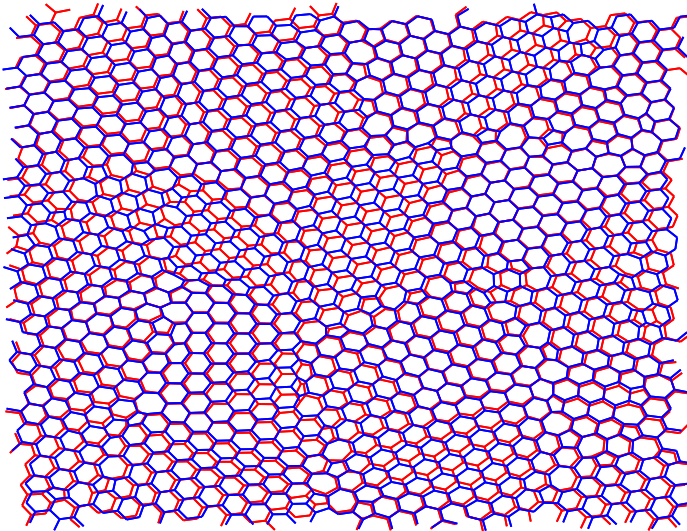
(a)**(b)**

Figure 3.1: A sample of relaxed buckled graphene with $N = 1600$ atoms, before and after a cycle in the configuration space at fixed topology. This is shown both from the side (a) and from the top (b). Its hysteretic behaviour is clearly visible: the states at the beginning and at the end of a cycle are dramatically different. The energies of the two states are 70.90 eV (red) and 71.63 eV (blue). The depicted sample has been selected because the hysteretic behaviour is particularly visible.

3.2 Model

We opted for the empirical graphene potential introduced in Ref. [48], which is based on Kirkwood's potential [63]. This potential has been used, for instance,

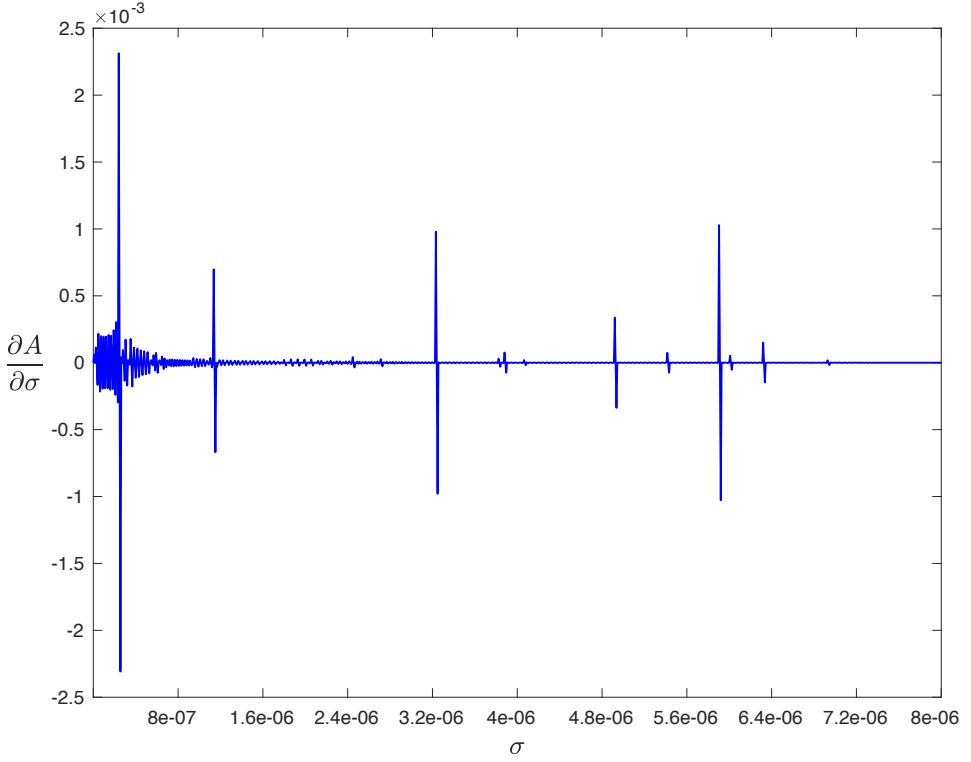


Figure 3.2: Evolution of the non-affinity parameter as the stretching force is gradually applied to a sample of relaxed graphene with $N = 3200$ atoms at fixed topology. The evolution of the shape of the sample during the straining is discontinuous, as occasionally a tiny increase in stretching force causes a significant displacement, which manifests through the discontinuity of the derivative of the non-affinity parameter A , with respect to the applied stress σ , analogous to avalanches.

for studying the long-range relaxation of structural defects [48], for probing crystallinity of graphene samples via their vibrational spectrum [64], for the study of twisted and buckled bilayer graphene [65] as well as of the shape of a graphene nanobubble [66]. In detail, for a two-dimensional hexagonal network for which out-of-plane deformations are also allowed, this potential can be written as

$$E_0 = \frac{3}{16} \frac{\alpha}{d^2} \sum_{i,j} (r_{ij}^2 - d^2)^2 + \frac{3}{8} \beta d^2 \sum_{j,i,k} \left(\theta_{j,i,k} - \frac{2\pi}{3} \right)^2 + \gamma \sum_{i,jkl} r_{ijkl}^2 \quad (3.1)$$

with $d = 1.420 \text{ \AA}$, the ideal bond length for graphene. A two-body bond stretching energy contribution is parametrised by $\alpha = 26.060 \text{ eV/\AA}^2$, $\beta = 5.511 \text{ eV/\AA}^2$ controls the bond shearing contribution, while $\gamma = 0.517 \text{ eV/\AA}^2$ corresponds to

the energy cost for the out-of-plane deformation of the graphene membrane [48]. Since the periodic boundary conditions apply, we can represent the contribution to the elastic energy due to external stress by an additional term of the form

$$E_\sigma = E_0 - \sigma L_x L_y = E_0 - \sigma S \quad (3.2)$$

where L_x and L_y are the lateral dimensions of the two-dimensional periodic box, S its surface and σ the parameter that controls the strength of the stretching force.

We first generate a polycrystalline flat sample by following the procedure described in Ref. [48]. A sample is allowed to relax to a lower energy configuration following the Fast Inertial Relaxation Engine (FIRE) algorithm [59]. The values of the parameters in this algorithm (N_{\min} , f_{inc} , f_{dec} , α_{start} and f_α) are taken as suggested in Ref. [59]. Further relaxation of the sample requires topological changes in the network, which we perform through so called bond transpositions [46]. In this procedure, four connected atoms are selected, two bonds are then broken and re-assigned between them. In the implementation we use an improved algorithm [52] that avoids complete relaxation before rejecting a bond transposition. After a random bond switch the complete sample is relaxed and the new configuration is accepted with the Metropolis probability

$$P = \min \left\{ 1, \exp \left(\frac{E_b - E_a}{k_B T} \right) \right\} \quad (3.3)$$

which includes the effect of thermal fluctuations: a move that increases the energy of the sample might be accepted, with a higher probability as temperature increases. However, note that no bond transposition is allowed while we perform the cycle in the configuration space. Temperature does therefore partially influence the evolution of the sample, and as long as there are still topological lattice defects around which the stress can accumulate, the discontinuous evolution persists.

Once this flat sample is sufficiently relaxed, every atom is placed at a random non-zero distance from the two-dimensional plane and allowed to relax to a buckled three-dimensional configuration. As topological defects increase the elastic energy of the sample, both globally and locally, they cause stress in the material that, once the material is allowed to relax in a three-dimensional configuration, is released by buckling [48]. Using such configuration of the graphene membrane, we perform a cycle in the configuration space by manipulating the stretching force σ in the empirical potential, given by Eq. (3.2), while the topology of the sample is kept fixed. During this process, we increase the stress in steps of size $\Delta\sigma$, and, after each step, the sample is allowed to relax. Once the value of the maximum stress σ_{max} is reached, it is decreased in the same way. In this work, $\Delta\sigma = 8 \times 10^{-9} \text{ u } \mu\text{s}^{-2}$ and $\sigma_{\text{max}} = 8 \times 10^{-6} \text{ u } \mu\text{s}^{-2}$, with u the atomic mass unit.

Between each stretching step, the atoms in the graphene membrane translate to a different position, and the relevant translations are only the non-affine (intrinsic) ones, which are defined in the reference frame fixed to the sample itself, and therefore they are not related to the expansion or contraction of the sample. In order to characterise the evolution of the graphene sample under the applied external stress, we thus use the non-affinity parameter defined in Ref. [67] as

$$A = \frac{\langle (r_i - r_{i,A})^2 \rangle}{L_x L_y} \quad (3.4)$$

where r_i is the in-plane position of the atom i , while $r_{i,A}$ is its expected position due to the expansion of the reference frame and $\langle \cdot \rangle$ is the average on all the atoms.

We can also extract the local energy distribution at a given moment with distinct energy contributions for the different terms in the total elastic energy, given by Eq. (3.1). This is computed by dividing the two body energy of every bond equally between the two atoms and assigning the three body and out of plane energy to the central atom.

3.3 Numerical simulations and results

We first adiabatically stretch and relax a sample with $N = 1600$ atoms in the way we previously described. After completing a cycle in the parameter space, the sample is in a different configuration, as it can be seen in Figure 3.1. By repeating this procedure a limited number of times, our sample ends up in a stable configuration. After repeating the cycle, the system reaches again a stable configuration, which is, however, different than the initial one.

The same procedure is then applied to a large sample containing $N = 3200$ atoms but in this case we further characterise the sample during a gradual application of the external stress.

We start with a stable configuration and perform a cycle during which we compute the non-affinity parameter, defined in Eq. (3.4) and its derivative with respect to the stretching parameter. Figure 3.2 shows that the evolution of the sample during the straining is clearly discontinuous. Significant changes in the configuration are caused by small increases in stretching force, which results in discontinuities in the derivative of the non-affine parameter.

We now analyse in more detail these discontinuities, in particular how the local energy distribution changes. As we can see in Figure. 3.3, the change of energy is not uniform in the sample. The energy difference, both positive and negative, is concentrated on and around the ridge defects in the sample [41, 61]. Separat-

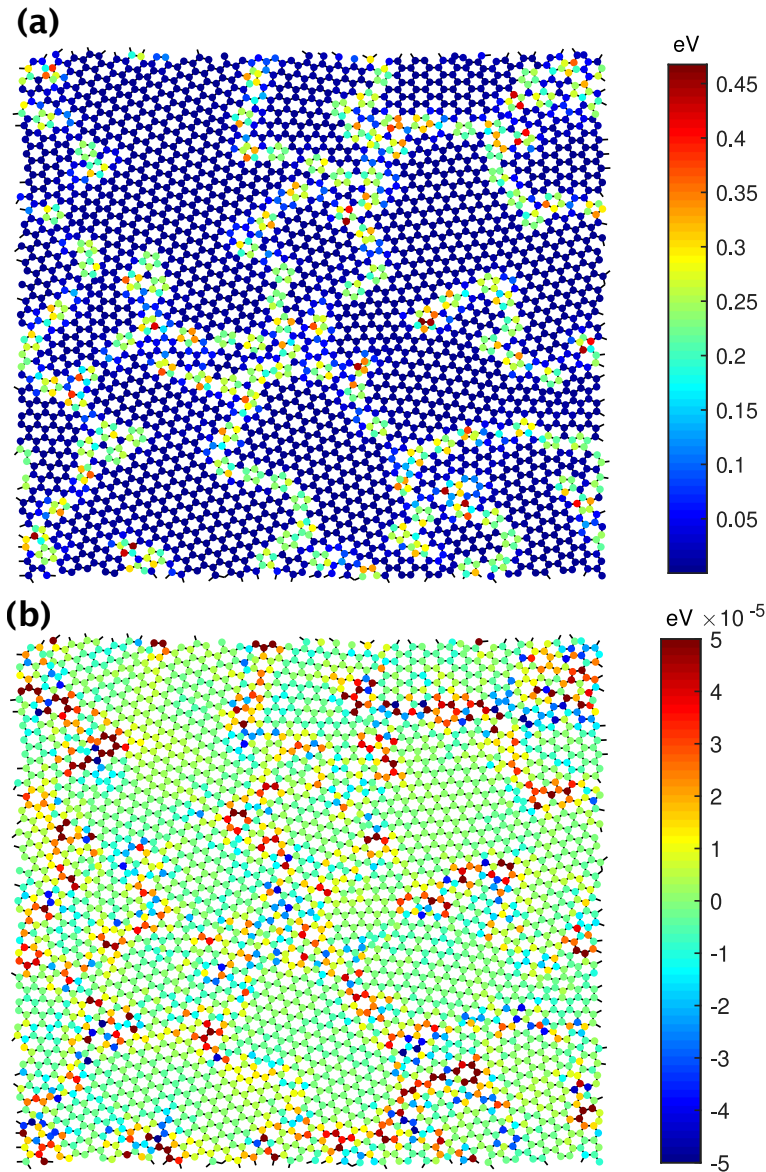


Figure 3.3: (a) Distribution of local energy in the graphene sample before a transition; (b) the local energy difference before and after the transition during stretching. The graphene sample contains $N = 3200$ atoms. The figure shows that the changes in the local energy distribution due to the transition are concentrated on and around ridges and vertices.

ing the different energy contributions, we notice that, at each transition during the straining, the sample increases its internal energy E_0 given by Eq. (3.1), even

though not all the terms contributing to it are necessarily positive. Nevertheless, the total elastic energy decreases during these transitions as the contribution of the stretching term, given by Eq. (3.2) is negative, indicating a sudden increase of the surface area of the sample. This is confirmed by subtracting the affine component corresponding to the surface expansion or contraction during the transition, computed through a fit of the data points immediately preceding the transition, which show a significant non-affine component (see Supplemental Material² for a breakdown of the affine and non-affine components of the surface changes during the transition). Finally, a minimal size of the sample in which this effect can be observable in our numerical experiment can be estimated as follows. In Figure 3.2 the highest peak is 2.7×10^{-3} , while the threshold for the observation of the discontinuity is 0.4×10^{-3} . Given that the sample we used in our simulations (Figure 3.3) contains about 15 grains, and assuming a linear scaling of the size of the discontinuity with the number of grains, yields an estimate of a critical size of about 3 crystalline grains.

3.4 Discussion and outlook

In summary, we established that the evolution of graphene sheets is discontinuous under gradually applied external stress. It proceeds through a series of avalanche-like processes in which the ridges and vertices are created and annihilated, with the energy concentrated in these defects. Furthermore, the behaviour of the graphene membrane is hysteretic: the system does not follow the same path back in the configuration space. Our results imply that if twisting is applied gradually to graphene bilayers and multilayers, the change of the shape of such structures should be discontinuous in nature, which could be possibly relevant to recent experiments [68–71]. Our results should also be pertinent to other two-dimensional polycrystalline materials, including phosphorene, MoS₂ and hexagonal boron nitride (h-BN) whose experimental realization has been recently reported in Refs. [72–74], as well as in van der Waals heterostructures [75]. A continuous application of stress to a graphene elastic membrane necessarily involves a creation and annihilation of the topological defects in the form of ridges and vertices, which can be consequential for the electronic and mechanical properties of these systems. On the other hand, our findings should motivate experimental studies of the dynamical elasticity in these systems, in particular the evolution of the elastic moduli, vibrational density of states and thermal conductivity with the adiabatically applied external stress. Finally, although we considered a specific case of monolayer graphene, we expect that our findings will be applicable to generic membranes embedded in three-dimensional space. In particular, within this approach it would be

²<https://journals.aps.org/prb/supplemental/10.1103/PhysRevB.100.161402/>

interesting to study the dynamical properties of origami metamaterials [76].

CHAPTER 4

Dynamic sampling from a
discrete probability
distribution with a known
distribution of rates

Abstract - In this chapter, we consider several efficient data structures for the problem of sampling from a dynamically changing discrete probability distribution, where some prior information is known on the distribution of the rates, in particular the maximum and minimum rate, and where the number of possible outcomes N is large.

We consider three basic data structures, the Acceptance-Rejection method, the Complete Binary Tree and the Alias method. These can be used as building blocks in a multi-level data structure, where at each of the levels, one of the basic data structures can be used, with the top level selecting a group of events, and the bottom level selecting an element from a group. Depending on assumptions on the distribution of the rates of outcomes, different combinations of the basic structures can be used. We prove that for particular data structures the expected time of sampling and update is constant when the rate distribution follows certain conditions. We show that for any distribution, combining a tree structure with the Acceptance-Rejection method, we have an expected time of sampling and update of $O(\log \log r_{\max}/r_{\min})$ is possible, where r_{\max} is the maximum rate and r_{\min} the minimum rate. We also discuss an implementation of a Two Levels Acceptance-Rejection data structure, that allows expected constant time for sampling, and amortized constant time for updates, assuming that r_{\max} and r_{\min} are known and the number of events is sufficiently large.

We also present an experimental verification, highlighting the limits given by the constraints of a real-life setting.

This chapter is partially based on the following publication:

F. D'Ambrosio, H. L. Bodlaender, and G. T. Barkema, "Dynamic sampling from a discrete probability distribution with a known distribution of rates", *Comput Stat*, 10 . 1007 / s00180-021-01159-3 (2021).

4.1 Introduction

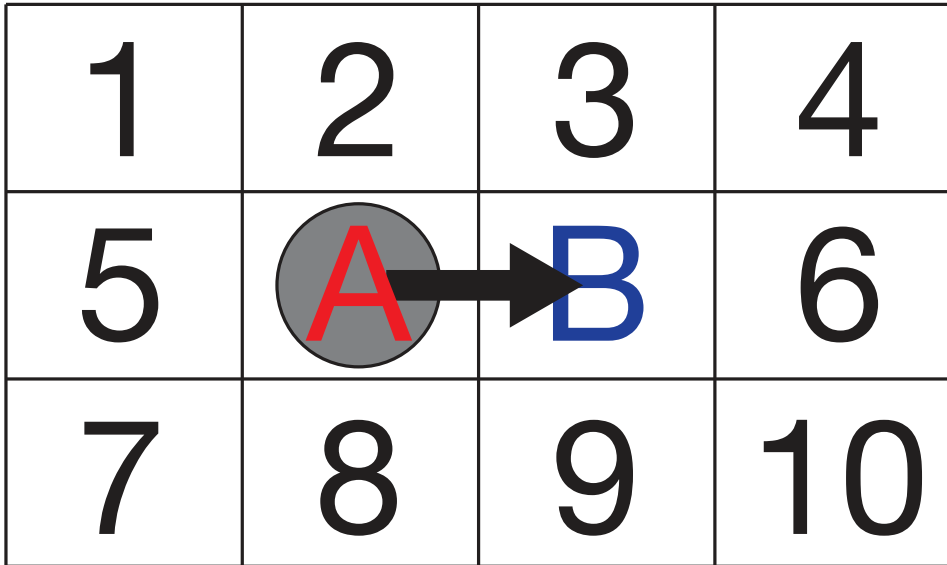


Figure 4.1: In the continuous time simulation of MBE growth on a metallic substrate, the hopping rate of an atom of copper from a position A to a position B is dependent on the occupational state of the ten surrounding sites: each move will influence the hopping rates of the surrounding atoms. It is critical to store these rates in a data structure that allows for updates.

4.1.1 The problem

In this chapter, we consider the design of data structures for the following problem. We have a dynamic discrete probability distribution, more precisely, we have a finite set of events, each with a rate. We have the following operations on the data structure: an event can be deleted, inserted, change its rate, and we want to randomly select an event, with each event selected with a probability proportional to its rate. This problem is well understood when the number of events is small, but in many applications, we need to sample from a very large collection of events.

In this chapter, we make one further assumption: we assume that the rates of possible events are distributed according to a known and unchanging probability distribution $\rho(r)$, i.e. the expected number of events with a rate between r and $r + \Delta r$ out of N total events can be computed as:

$$E[n_{r,r+\Delta r}] = N \int_r^{r+\Delta r} \rho(x) dx. \quad (4.1)$$

From this continuous distribution, which we call **rate distribution**, events are generated to populate and update the discrete distribution that we intend to sample. Knowledge about the rate distribution might come from theoretical knowledge about the underlying processes, direct observation, Monte Carlo simulations, etc. (For more details, see Section 4.2).

4.1.2 An illustrative example

To better understand the problem studied in this chapter, let us first introduce an example from a real life application: a continuous-time simulation of Molecular-Beam-Epitaxial (MBE) growth on a metallic substrate [47, 78], in the sub-monolayer regime. The set of energetically preferred positions of adatoms (atoms dropped on the surface) located on top of the metallic substrate forms a natural lattice with coordination number z (i.e. z denotes the number of neighbors of each site), typically a square lattice with $z = 4$ or a honeycomb lattice with $z = 3$. While new atoms are arriving on the substrate with a statistical rate determined by the beam intensity, the ones already present are hopping from one such preferred (lattice) position to a neighboring one, usually resulting in coalescence in islands. The hopping rate for an atom from site A to a neighboring site B depends on the atoms in the immediate vicinity of A and B . If site B is not occupied, the hopping rate is in very good approximation determined by the occupational state of the closest neighbors of A and B . In the case of $z = 4$, seen in Figure 4.1, this results in 2^{10} possible configurations, and, for each of these configurations, we can pre-compute the hopping rate [79]. The simulation then proceeds by two steps:

- a) the time is moved forward by a value Δt equal to the inverse of the sum of all the rates of all possible events;
- b) after this time increment, one event (hopping or arrival) is selected, with a probability proportional to its rate.

Therefore, we compute the rate of every possible move of every atom at every iteration and we sample a random event, employing a simple data structure: usually an array of size zN which contains at every index the sum of the rates of all the events up to that one. A random number between zero and the sum of all the rates is generated and we move through the array until we reach a value larger than our random number and we sample that event. This might work well, but it does not scale as the sampling time grows linearly with the number of possible events. With limited literature search we find better structures for our problem, for instance, Complete Binary Trees (see Section 4.3.2 and [80]) or, for a more flexible implementation, the Differential Search Tree from Maurer [81]), which sampling time grows logarithmically with the number of possible events, and even an op-

timal solution: the Alias method (see Section 4.3.4), proposed by Walker in 1974 [82, 83], an ingenious method that, employing two tables of the same size as the number of possible events, allows constant time sampling, regardless their number or their rate. Alternatively, if we assume that all rates can be written as multiples of a unit, these can be stored in an array and sampled in constant time by picking a random site of the array; the obvious downside is the size of such array. While it can be compressed with the method from Marsaglia [84], sampling from a compressed array requires $O(\log r_{\max})$, with r_{\max} the highest rate.

However, we see no significant improvement if we employ one of these structures. After each move, some of the configurations will have changed and we will have to rebuild the whole data structure from scratch, which costs a time that grows linearly with the number of possible moves zN , compromising the time saved with the sampling, even though only a limited number of possible moves have changed their rates. We can implement a (costly) update for the Complete Binary Tree (see Theorem 1) that requires $O(\log N)$ time, but that would still not scale well for larger numbers of atoms, and we would be tempted to optimize it in such a way that closer atoms are in the same branches, minimizing the number of operations required for the update, but as the atoms move they change neighbors, invalidating the optimization.

As the number of atoms necessary to study larger scale effects can be quite large, we would need a data structure that allows both optimal sampling and update of a random element. Unfortunately, and quite surprisingly, we were unable to find one for the general case. A quasi-optimal solution to the problem was given in 2003 by Matias et al. [85]. This method allows sampling in $O(\log^* N)$ time, with \log^* the iterated logarithm, and the update of an arbitrary item in $O(2^{\log^* N})$ worst-case time and $O(\log^* N)$ amortized expected time. Unfortunately, the method of Matias et al. is very complex to implement. A preliminary experimental study was done by van der Klundert [86]. Alternatively, the Acceptance-Rejection method (see section 4.3.3) does allow constant time updates, at the cost of performing samples in non-deterministic time, in which the expected value is dependent on the distribution of rates (see Theorem 2). Rajasekaran and Ross [87] and Hagerup et al. [88] developed different solutions that allow for expected constant time updates and samplings by imposing restrictions on the updates that are not in general satisfied in our example or in similar settings, where the ratio between the largest and smallest rate can be quite large or even arbitrarily large.

Our example is not unique. Similar problems have been described not just in material physics, but also chemistry [89] and biochemistry [90], and is in general relevant when we have an arbitrarily large number of possible events of known rate and their realization does not alter a significant fraction of them. It is there-

fore quite striking that we were not able to find in literature a general solution for such a relevant problem.

Since this is an intrinsically stochastic problem, it is sensible to ask whether the properties of the distribution of the rates of the possible outcomes, which can be determined either analytically or numerically assuming that the process that generates them is known, relates to the problem. An analysis from this point of view is also, to the best of our knowledge, missing in literature while there are some assumptions (see Section 4.2) that can be reasonable employed for large sets of applications that lead to some interesting solutions that we present in this chapter.

4.1.3 Our main contributions

Our main contributions are twofold. First, we identify several cases where assumptions on the distribution and/or the number of events lead to expected constant time for sampling an event; while insertions and deletions of events can be done in amortized constant time. In particular, the known Acceptance-Rejection method gives expected constant time for non-increasing distributions; our new two-level Cascade method gives expected constant time for two large classes of distributions, and our new two-level Acceptance-Rejection method gives expected constant time regardless of the distributions. In several cases, the result only holds for a sufficiently large number of events; in all cases, bounds for the smallest and largest rate of events have to be known. Second, we give an experimental evaluation of several of the data structures, both from existing literature and those introduced in this chapter.

4.1.4 Organization of this chapter

We start by defining our assumptions and the problem we are setting ourselves to solve (Section 4.2), then we will define and study the property of the data structures, both simple (Section 4.3) and multilevel (Section 4.4), that we employ to solve our problem. We perform an experimental analysis of our findings (Section 4.5). Some conclusions are given in Section 4.6.

4.2 Problem Statement and Assumptions

The data structures we study maintain an *Event Set* \mathbb{E} . The event set is a dynamic finite set (i.e. a finite set that can change over time). We call the elements of the Event Set *events*. Each event has a known, real, non-negative *rate*, that also can change over time; we denote the rate of event e_i by $r(e_i)$. The rate of an event represents the number of expected occurrences in some arbitrary time unit.

Our data structures support as operations the insertion of an element (with a given

rate), the deletion of an element, the change of the rate of an element, and a fourth operation: the *sampling* from the set of events. When we sample from the set of events, we randomly pick an event with a probability that is proportional to its rate. Thus, the probability that $e_i \in \mathbb{E}$ is sampled equals to

$$p(e_i) = \frac{r(e_i)}{\sum_{e \in \mathbb{E}} r(e)}. \quad (4.2)$$

We make a further assumption, namely that we know the distribution of the rates of the events. More precisely, we have a probability density function ρ such that the expected frequency of events with a rate between a and b equals $\int_a^b \rho(x) dx$. We assume that ρ is known and fixed. This of course does not guarantee that at all times the possible events will be distributed following ρ , but that as the number of possible events $N \rightarrow \infty$ it will tend to ρ . It is useful to think of $\rho(r)$ as the continuous distribution from which the rates of the elements of the Event Set, the events that are possible at each given time, are sampled. We also assume that the rate has known and finite maximum r_{\max} and minimum r_{\min} (i.e., $\int_{r_{\min}}^{r_{\max}} \rho(x) dx = 1$ and $\rho(r) = 0$ for $r \notin [r_{\min}, r_{\max}]$). We finally also note that, by definition, rates of possible events are strictly positive, and therefore also r_{\min} and r_{\max} are defined as positive.

For the cases where N is instead small, the Complete Binary Tree (see Section 4.3.2) is a good option, as it gives an $O(\log N)$ method that does not require assumptions on the distributions of rates.

In many practical cases, the assumptions may be approximations of the real situation. Often, in such cases, the predicted expected times for our data structures can be good approximations of the true behaviour.

Given these assumptions, our problem is the following:

Problem. *Given these assumptions, what is the most efficient method that allows for an event set \mathbb{E} :*

1. *sampling of an event (with each element selected with a probability that is proportional to its rate);*
2. *update of the rate of an arbitrary number of events;*
3. *removal or addition of an arbitrary number of events.*

Our problem statement represents a not-so-uncommon type of problems in dynamic simulations where the processes are only locally interdependent, i.e. the realisation of a process influences only up to a constant fraction of all possible

processes. An update of an event can be implemented by deleting the event and inserting a new event with the new rate; in several cases, we thus do not discuss updates of rates separately.

4.3 Data Structures

In this section, we describe several data structures for the problem studied in this chapter and briefly discuss dynamic arrays. After a short discussion of dynamic arrays, we review two basic well-known data structures: a Complete Binary Tree and the Acceptance-Rejection method. After that, we introduce three derivative methods which provide an efficient solution in different cases, depending on the probability distribution of the rates.

4.3.1 Dynamic arrays

In several cases, we store the events (or pointers to groups of events) as elements in an array. As we can add new elements to the data structure, the size of such an array can become too small. For this, we can use the standard data structure of *dynamic arrays*, also known as dynamic tables, see e.g., [91, Chapter 17.4]. Several standard programming languages have this data structure built-in, e.g., dynamic arrays are provided under the name of *vectors* in the C++ Standard Library. The main idea is that we use an array that is at least as large as needed, and copies all elements to an array of double size when the current array is too small. Occasionally, we have an operation that uses time, linear in the number of stored elements, but this happens infrequently, and the amortized time per insert (i.e., the total time divided by the number of operations) is bounded by a constant. For the details, we refer to e.g. [91, Chapter 17.4].

4.3.2 Complete Binary Trees

A data structure that is commonly used for event sampling is the Complete Binary Tree. Here, a *Complete Binary Tree* is a binary tree (i.e., a rooted tree with each node having at most two children), with all levels completely filled, except possibly the lowest, which is partially filled. (Complete Binary Trees are also sometimes known as *treaps*.) If we also impose that the lowest level is filled from the left, there is a simple implementation of Complete Binary Trees in arrays: we store the elements in an array $A[1 \dots n]$, with the parent of node $A[i]$ being $A[\lfloor i/2 \rfloor]$ ($i > 1$). See e.g. [91, Chapter 6.1], or [92].

While it would be tempting to group together in the same branch the events whose rate we might know to be correlated, for instance, the hopping rate of two spatially close atoms, we cannot assume that they will stay that way as the system evolves dynamically. The Complete Binary Tree has the advantage of an easier im-

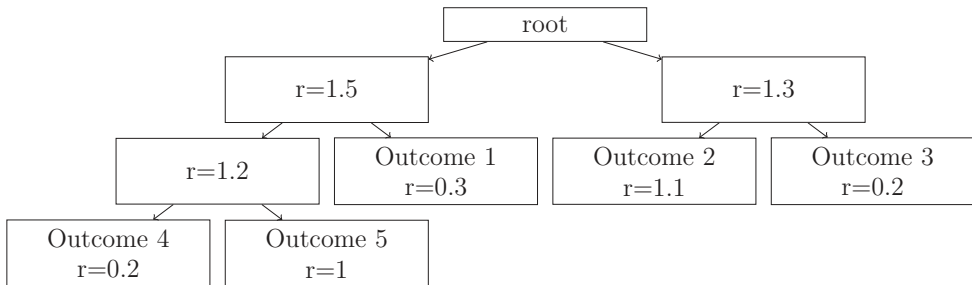


Figure 4.2: In the Complete Binary Tree, each node stores a variable called *rate*. Leaves, representing events, have the same rate as the corresponding event. Internal nodes have a *rate* equal to the sum of the rates of their children. An event is sampled by generating a random number between zero and the *rate* of the root (i.e. the sum of all rates): if this is smaller than the *rate* of the left node, we move to this node; otherwise, we subtract the left *rate* and move to the right node. This is repeated until we reach a leaf and the corresponding event is sampled.

plementation, and it minimizes the average depth of the tree over all binary trees to $d = \lfloor \log_2 N \rfloor$, with N the number of events.

A schematic representation of the structure of a Complete Binary Tree is shown in Figure 4.2. Each leaf represent an event and it is associated with its *rate*; internal nodes also have a *rate* associated with them and it is equal to the sum of the *rates* of their children. Sampling is intuitive: a random number between zero and the sum of the rates of all the events (r_{tot}) is generated and, starting from the root, if this is smaller than the *rate* of the left node we move in that direction; otherwise, we subtract the *rate* of that node and we move to the right. This is repeated at most d times until we reach a leaf. An update is performed by changing the *rate* of the corresponding leaf and updating the *rate* of the internal nodes between itself and the root. It is also possible to add or remove an event, by adding or removing a leaf with the usual methods, the *rate* of the affected internal nodes is updated. This is easiest in the array implementation: adding a new leaf just adds the element at the end of the array; in a deletion, we move the last element of the array to the position of the deleted element; in both cases, we update the rates of all nodes that are an ancestor of a replaced, inserted or deleted leaf. Under these assumptions, it is quite trivial to prove that all the operations that are interesting for us require logarithmic time. The following result can be easily derived from well-known insights and given here for completeness reasons.

Theorem 1 (Complete Binary Tree). *Given an Event Set \mathbb{E} of cardinality N represented as a Complete Binary Tree:*

- (a) *the sampling of an event can be performed in $O(\log N)$ time;*

- (b) *the update of the rate of an event can be performed in $O(\log N)$ time;*
- (c) *the addition or removal of an event can be performed in $O(\log N)$ time.*

Proof. (a) The sampling of an event requires a number of operations proportional to the number of nodes on the path between the root and the sampled leaf. For a Complete Binary Tree, this is at most $d = \lfloor \log_2 N \rfloor$ [93] and therefore it is $O(\log N)$.

- (b) In order to update the rate of an event, we perform a single operation on the leaf and then we update the internal nodes following backwards the same path as in (a). Therefore this is also $O(d) = O(\log N)$ operations.
- (c) First the leaf is deleted or added, which, for a binary heap, requires $O(\log N)$ time and then the rate of the nodes in the path from the deleted/added node to the root is updated. As we already mentioned, this costs also $O(\log N)$ time.

□

4.3.3 The Acceptance-Rejection method

One of the classic methods is the Acceptance-Rejection method. Here, we have an array of size N where each entry represents a possible outcome and its value is equal to its rate. Since the distribution is known, we assume that the maximum rate is also known. As no ordering is necessary, an element can be added and removed by simply adding or removing it from the array at any time, without any further preprocessing required. To sample an event, we randomly select an element and generate a random number between zero and the highest possible rate (r_{\max}), which is known (see Section 4.2); if this is larger than the value of the selected element, it is rejected and we draw a new one. Otherwise, it is accepted and sampled. The rate of an event is updated by simply changing the value of the corresponding element. A simple way to represent such data structure is as a histogram where each bin represents an element and their height is determined by their rate, up to the known maximum value r_{\max} . An example of such representation can be seen in Figure 4.3.

We can make the data structure dynamic by using a dynamic array instead of a (usual) array; see the discussion at the start of this section. A new event can be added at the end of the array, and an element can be removed by moving the last element of the array to its position.

In contrast with other methods, the sampling time does not depend on the cardinality of the Event Set (i.e. the number of possible events N) while the updating

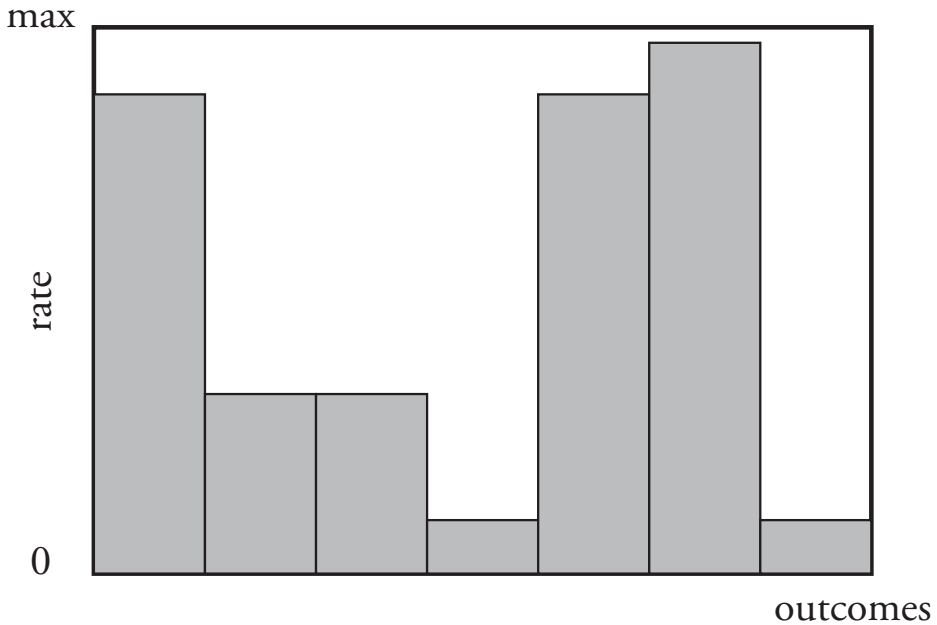


Figure 4.3: Acceptance-Rejection structure. Each element of the array represents the *rate* of a possible event. An event is sampled by selecting a random element and generating a random number between zero and the r_{\max} : if the latter is smaller than the former, the corresponding event is sampled; otherwise the process is repeated.

time is always trivial, but we have to investigate how the rate distribution affects the sampling time. As this is a stochastic method, it is sensible to look at the expected time. We give a simple analysis of this method below.

Theorem 2 (Acceptance-rejection). *Given an Event Set \mathbb{E} of cardinality N and largest rate r_{\max} , represented as an Acceptance-Rejection structure:*

- (a) *the sampling of an event can be performed in expected $O\left(\frac{r_{\max}}{E[r]}\right)$ time, with $E[r]$ the expected value of the rate according to the distribution $\rho(r)$;*
- (b) *the update, addition or removal of an event can be performed in constant time.*

Proof. (a) The probability of selecting an event with rate r is equal to the frequency of such events, which is expected to be $\rho(r)$. Then, a random number S is generated from a uniform distribution with support $[0, r_{\max}]$ and the event is accepted if $S \leq r$, the probability of which is $\frac{r}{r_{\max}}$. We can then integrate it over all possible values of r and write the probability of accepting

an event of any rate as:

$$P_{\text{sample}} = \frac{1}{r_{\max}} \int_{r_{\min}}^{r_{\max}} \rho(r) r dr = \frac{E[r]}{r_{\max}}. \quad (4.3)$$

Since this is a Bernoulli trial, the expected number of attempts before the first success is

$$E[n] = \frac{1}{P_{\text{sample}}} = \frac{r_{\max}}{E[r]}, \quad (4.4)$$

and the number of operations is proportional to the number of attempts.

- (b) The addition or removal of an event is performed by adding or removing an element to or from the vector. The rate of an event can be updated by simply changing the value of its element in the vector. All these actions require a constant number of operations, therefore they can be performed in constant time.

□

As the expected value of the rate cannot be smaller than the smallest possible rate, we can also say that

Corollary 1. *The sampling of an event can be performed in expected $O(r_{\max}/r_{\min})$ time, with r_{\min} the smallest rate in the Event Set.*

To avoid confusion, we can visualize the Event Set \mathbb{E} as a histogram of bins of equal width and height proportional to their rate, with the expected frequency given by the rate distribution $\rho(r)$. Note that, as we can see in Figure 4.4 the histogram does not look like the rate distribution. The sampling of an event is analogue to randomly shooting a dart on this area: if it lands inside a bin, that event is sampled; otherwise, it is rejected.

Let us step back to the result of Theorem 2 for sampling. We can easily imagine a worst-case, where all the events except one have a rate arbitrarily smaller than the largest and the sampling time, therefore, grows arbitrarily, and a best-case, where all the events have the same rate and the sampling time is constant. Is there a more general assumption we can introduce on the rate distribution $\rho(r)$ that would still guarantee expected constant time? We will show that assuming that the rate distribution is non-decreasing is sufficient to guarantee expected constant time.

The probability of selecting an outcome with a given rate is proportional to the

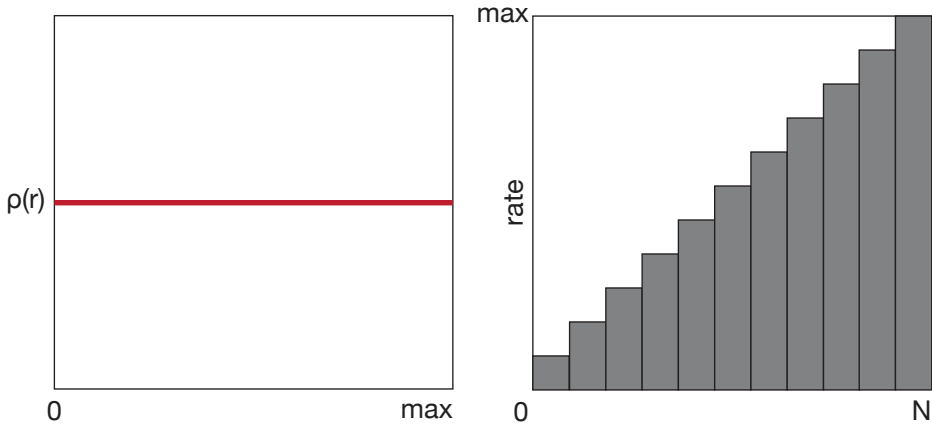


Figure 4.4: On the left, a uniform rate distribution; on the right, the visualization of the corresponding Event Set. In order to avoid confusion in the following proofs, it is important to remember this distinction. Since order does not affect sampling, rates are ordered for clarity.

number of elements with that rate. We can therefore write

$$\int_{r_{\min}}^{\bar{r}} \rho(r) dr = \int_{\bar{r}}^{r_{\max}} \rho(r) dr, \quad (4.5)$$

with $\rho(r)$ the rate distribution and \bar{r} the median of the distribution $\rho(r)$ [94], i.e. the real number for which

$$\int_{r_{\min}}^{\bar{r}} \rho(r) dr = \frac{1}{2} \quad \int_{\bar{r}}^{r_{\max}} \rho(r) dr = \frac{1}{2}, \quad (4.6)$$

which is guaranteed to be unique if $\rho(r) > 0$ in the open interval (r_{\min}, r_{\max}) . Since the possible outcomes are selected from an uniform distribution, this implies that the probability of selecting a possible outcome whose rate is at least \bar{r} or larger is

$$P_{\text{select}}(r \geq \bar{r}) \geq \frac{1}{2}. \quad (4.7)$$

Lemma 1. *If the rate distribution $\rho(r)$ is a non-decreasing function of r , its median is at least the middle of the interval $[r_{\min}, r_{\max}]$ (i.e., at least $\frac{r_{\min} + r_{\max}}{2}$).*

Proof. First, we rewrite Equation 4.5 as

$$\int_{r_{\min}}^{\bar{r}} \rho(r) dr - \int_{\bar{r}}^{r_{\max}} \rho(r) dr = 0. \quad (4.8)$$

Using that $\rho(\bar{r}) \geq \rho(r)$ for each $r \in [r_{\min}, \bar{r}]$ and $\rho(\bar{r}) \leq \rho(r)$ for each $r \in [\bar{r}, r_{\max}]$, as we assume that ρ is a non-decreasing function, it follows that

$$\rho(\bar{r}) \int_{r_{\min}}^{\bar{r}} dr - \rho(\bar{r}) \int_{\bar{r}}^{r_{\max}} dr \geq 0, \quad (4.9)$$

assuming that $\rho(\bar{r})$ is non-zero. Finally,

$$\bar{r} - r_{\min} - r_{\max} + \bar{r} \geq 0 \Rightarrow \bar{r} \geq \frac{r_{\max} + r_{\min}}{2}. \quad (4.10)$$

□

We are ready to prove the following theorem:

Theorem 3. *An Acceptance-Rejection structure with a non-decreasing rate distribution performs sampling of a possible outcome in expected constant time.*

Proof. Since the rates of possible events are strictly positive, we can write Equation 4.10 as:

$$\bar{r} \geq \frac{r_{\max} + r_{\min}}{2} \geq \frac{r_{\max}}{2}. \quad (4.11)$$

Remembering from the proof of Theorem 2a, we can write the probability of accepting an outcome with rate r , assuming that an outcome with rate $r \geq \bar{r}$ is already selected, is

$$P_{\text{accept}}(r | r \geq \bar{r}) = \frac{r}{r_{\max}} \geq \frac{\frac{r_{\max}}{2}}{r_{\max}} = \frac{1}{2}. \quad (4.12)$$

Remembering the result of Equation (4.7), the probability of successfully sampling an outcome with $r \geq \bar{r}$ therefore is

$$P_{\text{sample}}(r \geq \bar{r}) = P_{\text{select}}(r \geq \bar{r}) \cdot P_{\text{accept}}(r | r \geq \bar{r}) \geq \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}. \quad (4.13)$$

The probability of successfully sampling an outcome from a subset of the Event Set cannot be larger than the probability of sampling an outcome from the entire Event Set, which puts an upper boundary on the expected number of attempts before sampling an outcome

$$E[t_{\text{sample}}] = \frac{1}{P_{\text{sample}}} \leq \frac{1}{P_{\text{sample}}(r \geq \bar{r})} = 4 = O(1), \quad (4.14)$$

with P_{sample} the probability of successfully sampling an outcome from the entire Event Set. □

A visualization of this proof can be seen in Figure 4.5. This is a very powerful result: such a simple method allows constant time sampling for any Event Set with a non-decreasing rate distribution.

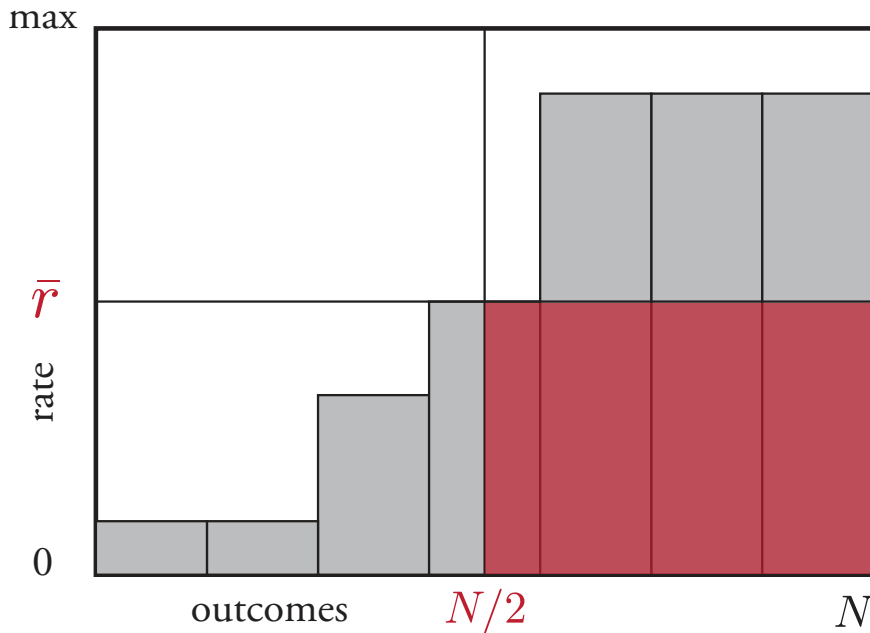


Figure 4.5: Each possible outcome is represented as a rectangle with unit width and height proportional to its rate. Since order does not affect sampling, rates are ordered for clarity. The probability of sampling an event is equal to the ratio between filled and total space in such a representation. From this geometric argument we can prove that, for a non-decreasing rate distribution, the probability of sampling an outcome is at least $\frac{1}{4}$ due to Lemma 1.

4.3.4 The Alias Method

The Alias method, introduced by Walker [82, 83] is a very ingenious solution to the static case of our problem. Each event is conceptually stored in a "bucket" of size r_{tot}/N ; if a bucket is not already full, the remaining space is assigned to another event, denoted as its *alias*, that is overfilling its bucket. The rate that has been assigned to the alias is then removed from its original bucket. This is repeated until each bucket is exactly full.

The buckets are represented as an array of size N , each element storing the fraction of the bucket assigned to the alias. To sample an event, an element and a random

number between zero and one are generated. If this is larger than the value stored in the element, the corresponding event is sampled; otherwise, we sample its alias.

As the number of steps required for sampling is fixed, the time required is constant. Unfortunately, except for some very particular cases, any update would be extremely costly and it would often require a complete rebuild of both tables, which takes at least $O(N)$ time. Nevertheless, we are presenting this method both for completeness and as a potential building block for multilevel methods.

4.4 Multilevel Methods

As we have seen, the Acceptance-Rejection method works better when the possible outcomes have a limited range of rates; if this is not the case, we can split the Event Set in multiple groups according to their rate, use one of the other methods to sample a group, and then the Acceptance-Rejection method to sample an element from that group [90]. We call such combinations of different methods *multilevel methods* and the structure that stores the groups *superstructure*. In this section, we present some of these combinations that have very powerful properties that will be shown in the next section.

4.4.1 Exponential grouping

All our two level methods employ the same data structure for the lower level.

The events are grouped according to their rates. Each group consists of all events with a rate in a specific interval. The sizes of these intervals grow exponentially, and hence we will refer in the successive subsection this grouping method by *exponential grouping*.

Fix some constant $c > 1$. A typical example would be to take $c = 2$. Different choices for c can affect the constant factors of the running time: larger values of c would slow down selection in the lower level of the data structure, but could speed up selection in the upper level of the data structure.

Number the groups starting at 1. The group with index i consists of all events with rate r in the interval

$$[c^{i-1} \cdot r_{\min}, \min\{c^i \cdot r_{\min}, r_{\max}\}), \quad (4.15)$$

adding the value r_{\max} to the last group, (i.e., all intervals except the last are right-open.)

For each group, we use a separate Acceptance-Rejection data structure to sample an event.

Lemma 2. *After a group is selected, sampling an event from that group can be done in $O(c)$ expected time.*

Proof. Note that the ratio between the largest and smallest rate of events from one group is bounded by $\frac{c^i \cdot r_{\min}}{c^{i-1} \cdot r_{\min}} = c$. Thus, the expected number of ‘rounds’ of the Acceptance-Rejection method until an event is selected from the group is bounded by c , which we assumed to be a constant. \square

Updating rates, inserting new events, and deleting events in the lower level data structure all can be done in constant time. An update can be performed by deleting the event with the old rate, and inserting an event with the new rate. We fix an array with an element for each group, that points to the Acceptance-Rejection data structure of that group. If we insert an element, with a constant number of arithmetic operations, we can determine its group, find the corresponding Acceptance-Rejection data structure, and add the event. To delete an event, we need a pointer to its location in its Acceptance-Rejection data structure, and delete it as in Theorem 2.

What remains is to build data structures to sample a group, where we need to select each group with a probability that is proportional to the total rate of all events in the group. For this, we have for each group a variable that maintains this total rate of all events in the group. Apart from that, we have different method to sample groups, which are discussed in the successive subsections.

4.4.2 Tree of Groups

Let us assume that the Event Set has an arbitrarily large cardinality but the range of rates is such that the number of groups required to cover it is limited. In such a case we can employ a Complete Binary Tree as a superstructure and obtain a very useful result: both update and sample are performed in $O\left(\log \log \frac{r_{\max}}{r_{\min}}\right)$ expected time. While this is not constant time, it is very small without requiring any further assumption on the rate distribution. A similar method, called SSA-CR (Stochastic Simulation Algorithm - Composition and Rejection), was introduced in [90].

The method thus works as follows. We group the events by the exponential grouping method (see Section 4.4.1). Each group is represented both as a leaf of a Complete Binary Tree (see Section 4.3.2), whose rate is given by the sum of the rates of all the events in the group, and as an Acceptance-Rejection structure where all its events are stored. This total rate can easily be maintained under insertions, deletions and updates; after such an operation the difference is added or subtracted from the group rate.

To sample an event, we first sample a group from the Complete Binary Tree in the previously described way (see Section 4.3.2) and the Acceptance-Rejection sampling (see Section 4.3.3) is performed inside it. Updates are trivial unless they require events to be moved to a different group; in that case, the relative element is removed from its group and added to the new one.

Theorem 4 (Tree of Groups). *Given an Event Set \mathbb{E} represented as a Tree of Groups:*

- (a) *the sampling of an event can be performed in $O\left(\log \log \frac{r_{\max}}{r_{\min}}\right)$ time;*
- (b) *the update, addition or removal of an event can be performed in $O\left(\log \log \frac{r_{\max}}{r_{\min}}\right)$ time.*

Proof. (a) Since the groups are stored in a Complete Binary Tree, the time to select a group grows logarithmically with the number of groups; the lower boundary of the i -th group is, by definition, $\frac{r_{\max}}{k^i}$; and the general lower boundary is r_{\min} , we can write the number of groups n as

$$\frac{r_{\max}}{k^n} = r_{\min} \Rightarrow n = \log_k \left(\frac{r_{\max}}{r_{\min}} \right). \quad (4.16)$$

Therefore, the time required to select a group from the Complete Binary Tree is $O(\log n) = O\left(\log \log \frac{r_{\max}}{r_{\min}}\right)$. Once we have selected a group, sampling an event from the group uses expected constant time (Lemma 2.)

- (b) Adding, removing and updating an event inside its group is performed in constant time (see Theorem 2 and Section 4.4.1). In order to maintain consistency it is necessary to update the rates in the Complete Binary Tree, which is performed in the same time as a sampling (see Theorem 1).

□

4.4.3 Cascade of Groups

We have previously shown that the Acceptance-Rejection is optimal for any non-decreasing rate distribution. While we would like to find a similar result for all decreasing rate distribution, therefore completing the solution for the general problem, we will split them into different subsets and attack them one at the time. Let us first consider those rate distributions that, according to some definition, decrease fast enough. For such rate distributions, most of the events will have lower values of rate; we must therefore store groups in a way that prioritize events with a lower rate.

Again, we use exponential grouping, see 4.4.1.

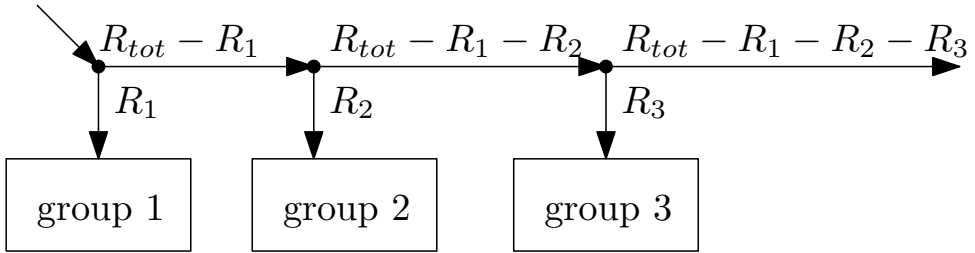


Figure 4.6: The first three groups in the Cascade of Groups structure. The numbers give the expected proportion of times the arrow is followed.

In the analysis below, we assume that r_{\max} is a multiple of r_{\min} . If this is not the case, we can have a slightly smaller last group. It is easy to see that the difference in expected running time is bounded by a constant.

Our data structure is as follows. We have a linked list [91, Chapter 10.2] with an element for each group, which has both a pointer to its Acceptance-Rejection data structure and the value of the sum of all the rate in the group (denoted, for the i th group, as R_i). See Figure 4.6 for visualization of such superstructure.

Sampling is, again, in two phases. A random number $rand \in [0, R_{\text{tot}}]$ is generated, with $R_{\text{tot}} = \sum R_i$; if $rand$ is larger than the sum of the rates in the first group R_1 , this is subtracted from $rand$ and we move to next group. This is repeated until a group is selected. A sample from the selected Acceptance-Rejection structure is then performed. Updates, addition or removal of events are performed inside the groups following the methods described in Section 4.3.3; R_{tot} and the sum of the rates in the involved group (or groups, for an event that changes group after an update) are also updated.

Let us start by introducing this useful Lemma for the Cascade of Groups:

Lemma 3. *In a Cascade of Group, if there is a constant $\alpha < 1$ such that, for each group, the expected sum of rates of a group is at most α times the expected rate of the previous group, then the expected time to select a group is $O\left(\frac{1}{1-\alpha}\right) = O(1)$.*

Proof. Once we reach the i th group in the Cascade of Group, the expected probability of selecting it is the expected rate of that group divided by the sum of the expected rate of that and all of the following groups. We call the expected rate of the i th group $E[R_i]$. Supposing we have g groups, the expected total rate of the

i th group and all following groups is at most

$$\begin{aligned} E \left[\sum_{j=i}^g R_j \right] &= \sum_{j=i}^g E[R_j] \leq \sum_{j=i}^g E[R_j] \cdot \alpha^{j-i} \leq \sum_{j=i}^{\infty} E[R_j] \cdot \alpha^{j-i} \\ &= E[R_i] \cdot \sum_{j=0}^{\infty} \alpha^j = \frac{E[R_i]}{1-\alpha}. \end{aligned} \quad (4.17)$$

The expected probability that when we are at a group i , take an element from that group is thus here at least

$$\frac{E[R_i]}{E[R_i] \cdot \left(\frac{1}{1-\alpha}\right)} = 1 - \alpha. \quad (4.18)$$

We can view the execution of the algorithm as an experiment that is repeated till the first success; with each round, we have a probability of success that is at least $1 - \alpha$. The expected number of steps before selecting a group is thus at most $O\left(\frac{1}{1-\alpha}\right) = O(1)$. \square

Following this Lemma, we can prove that the Cascade of Groups is a constant time solution if the rate distribution decreases fast enough:

Theorem 5. *Suppose we have a constant $c > 1$ such that for all $r \in [r_{\min}, r_{\max}/c]$:*

$$\rho(cr) \leq \rho(r)/c^\beta, \quad (4.19)$$

with $\beta > 2$, then the Cascade of Groups data structure gives expected constant time to sample an event.

Proof. We first relate the expected total rate of group i , $E[R_i]$, with the expected total rate of group $i - 1$, $E[R_{i-1}]$:

$$\begin{aligned} E[R_i] &= \int_{c^i r_{\min}}^{c^{i+1} r_{\min}} r \rho(r) dr = c \int_{c^{i-1} r_{\min}}^{c^i r_{\min}} c r' \rho(c r') dr' \\ &\leq c \int_{c^{i-1} r_{\min}}^{c^i r_{\min}} \frac{c}{c^\beta} r' \rho(r') dr' = c^{2-\beta} E[R_{i-1}], \end{aligned} \quad (4.20)$$

which follows by using the substitution $r' \rightarrow r/c$. Thus, we get the result by Lemma 3, and noting that $c^{2-\beta} < 1$, when $\beta > 2$.

Once inside a group, sampling takes expected constant time by Lemma 2. \square

4.4.4 Reversed Cascade of Groups

We can obtain a similar result when rates decrease sufficiently slow, just by reversing the superstructure. In the Reversed Cascade of Groups, we use exponential grouping (see Section 4.4.1), and again place these in a linked list (as for the Cascade of Groups), except that we link the groups in the reversed order, i.e., we start with the group with the events with the largest rate. Thus, if we have g groups, we first decide if we sample an element from g th group, then from $g - 1$ th group, etc.

Theorem 6. *Suppose we have a constant $c > 1$ such that for all $r \in [r_{\min}, r_{\max}/c]$:*

$$\rho(cr) \geq \rho(r)/c^\beta, \quad (4.21)$$

with $\beta < 2$, then the Reversed Cascade of Groups data structure gives expected constant time to sample an event.

Proof. Let $i < g$, i.e., group i is not the group with the events with largest rates. We again relate the expected total rate of group i , with the expected total rate of group $i - 1$. By substitution $r' \rightarrow r/c$, we obtain:

$$\begin{aligned} E[R_i] &= \int_{c^i r_{\min}}^{c^{i+1} r_{\min}} r \rho(r) dr = c \int_{c^{i-1} r_{\min}}^{c^i r_{\min}} c r' \rho(c r') dr' \\ &\geq c \int_{c^{i-1} r_{\min}}^{c^i r_{\min}} \frac{c}{c^\beta} r' \rho(r') dr' = c^{2-\beta} E[R_{i-1}]. \end{aligned} \quad (4.22)$$

All groups, except group g , thus fulfill the condition of Lemma 3. Visiting the first group costs constant time, and thus, with Lemma 3, and because $c^{\beta-2} < 1$ here, we see that the expected time to select a group is bounded by a constant. Again, the sampling inside a group costs expected constant time (Lemma 2). \square

We now have optimal solutions for small numbers of events (Tree of Groups, see Section 4.3.2), small range of rates (Tree of Groups, see Section 4.4.2), non-decreasing rate distributions (Acceptance-Rejection, see Section 4.3.3), fast decreasing rate distributions (Cascade of Groups, see Section 4.4.3) and slow decreasing rate distributions (Reverse Cascade of Groups, see Section 4.4.4). In the next Section, we will introduce an optimal solution of our problem for any rate distribution, if the number of events is significantly large.

4.4.5 Two Levels Acceptance-Rejection

We now discuss a Two Levels structure where both levels use the Acceptance-Rejection method.

Again, we group the events with exponential grouping (see Section 4.4.1).

The elements of the top-level Acceptance-Rejection structure are called *bins*. Each bin has a rate, and points to a group. We allow that groups have multiple bins, and the total rate of all the bins of a group equals the total rate of all events of the group.

By using multiple bins per group, we can obtain a constant expected time to sample an event, regardless of the rate distribution, and amortized constant time for insertions and deletions. However, insertions and deletions can require multiple pointer operations and can be slow in practice.

Suppose we are given the values of r_{\min} and r_{\max} . We now choose two values $B \geq r_{\max}$ and $c > 1$ that we consider to be constants. Let $g = \log_c \left\lceil \frac{r_{\max}}{r_{\min}} \right\rceil$ be the number of groups we obtain by using exponential grouping.

Now (as described in Section 4.4.1), each group at the lower level uses an Acceptance-Rejection data structure to sample an event from the group, but also the superstructure is an Acceptance-Rejection structure, where the bins play the role of events. Each group has at least one bin in the superstructure. Each bin has associated with it a non-negative real number, called *value*. For each group, all its bins have value B , except possibly the last (or the only) bin of the group.

A group is selected by randomly drawing a bin and a real number between 0 and B . If this random number is at most the value of the bin, then we select the bin and the group associated with it. Otherwise, we repeat this operation until a bin is selected. It is easy to see that the probability to select a group is proportional to the sum of the values of its bins, which is equal to the rate of the group.

The bins of a group have a pointer to the previous and next bin of the same group, and to an object that represents the group. That object has a pointer to the last bin of the group, a local variable equal to the rate of the group, and a pointer to the Acceptance-Rejection structure of the group.

An element can be added by inserting it in the Acceptance-Rejection structure of its group and then adding its rate to the value of the last bin of the same group. If this becomes larger than B , say it becomes $x > B$, then we create a new bin for the group, add it to the superstructure, set the value of the now second to last bin of the group to B , set the value of the new last element of the group to $x - B$, and set the pointers to and from the last and second to last bin of the group correctly.

Deleting an element is done by deleting it from the Acceptance-Rejection structure of its group, and subtracting its rate from the last bin of the group. Suppose the

rate of this last bin becomes γ . If γ is positive, it is simply updated. Otherwise, we delete the last bin of the same group and decrease the rate of the bin that has become the last of the group to $B + \gamma$.

Note that these operations ensure that all bins have a non-negative value that is at most B and that insertions and deletions involve a constant number of operations, and thus cost amortized constant time.

Lemma 4. *Suppose the total rate of all events is R_{tot} , and we have g groups. Then, the expected time to select a group is $O(1 + \frac{Bg}{R_{\text{tot}}})$.*

Proof. We have $m \geq g$ bins. All bins have a value at most B , and at most g bins have a value smaller than B , so we have that $R_{\text{tot}} > B(m - g)$. The expected value of a bin in this structure can be written as $E[b] = \frac{1}{m} \sum_{i=m}^1 b_i = R_{\text{tot}}/m$, with b_i the value of the i -th bin. We now can apply Theorem 2. As the event in this step is the selection of a bin, the expected value of a bin ($E[b]$) plays the role of $E[r]$, and the maximum value of a bin (B) plays the role of r_{max} , so, by Theorem 2, the expected time to select a bin and therefore a group is $O(\frac{B}{R_{\text{tot}}/m}) = O(Bm/R_{\text{tot}})$.

Now, observe that

$$\frac{Bm}{R_{\text{tot}}} = \frac{Bm - Bg}{R_{\text{tot}}} + \frac{Bg}{R_{\text{tot}}} < \frac{R_{\text{tot}}}{R_{\text{tot}}} + \frac{Bg}{R_{\text{tot}}} = 1 + \frac{Bg}{R_{\text{tot}}}. \quad (4.23)$$

The lemma now follows. □

Thus, recalling Lemma 2 and that $g = \lceil \log_c \frac{r_{\text{max}}}{r_{\text{min}}} \rceil$, we can state the following.

Theorem 7. *The Two Levels Acceptance-Rejection method, with $B \geq r_{\text{max}}$, $c > 1$, allows to perform*

1. *insertions and deletions in amortized constant time, and*
2. *sampling in expected time*

$$O\left(c + \frac{B \cdot \log_c \frac{r_{\text{max}}}{r_{\text{min}}}}{R_{\text{tot}}}\right), \quad (4.24)$$

when R_{tot} is the total rate of all current events in the data structure.

Interestingly, this means that the Two Levels Acceptance-Rejection method allows for constant time sampling if $R_{\text{tot}} > B \cdot g$. Therefore if there are enough events in the structure to satisfy this condition, we have a method than can be applied to any rate distribution, as long as r_{max} and r_{min} are known.

Two parameters can be set that influence the expected time, namely B and c . When we increase B , we can expect fewer operations that create or delete a bin, and thus would decrease the time needed for pointer operations, but it also means that the term $\frac{Bg}{R_{\text{tot}}}$ is larger, thus making the data structure viable only for larger total rate values R . When we increase c , we have fewer groups (as g is $\lceil \log_c \frac{r_{\text{max}}}{r_{\text{min}}} \rceil$), which decreases the time in the superstructure, but it increases the time to sample inside a group.

4.5 Experimental Analysis

We have performed an experimental analysis of the sample time of the previously described methods in order to confirm our asymptotic findings. We implemented these methods in the C++ language, building on top of C++11 Standard Library (in particular *random*). We computed the expected sample and the update time of each data structure by recording the time required with the *high_resolution_clock* method of the *chrono* library. Average and variance are computed according to the Walford method [95]. Our implementation is available in a GitHub repository¹.

Since rates can be re-scaled to different time units, the maximum rate is fixed to 1 in some arbitrary units, while r_{min} , the minimum possible rate, is a controllable parameter, together with N , the number of possible events, and the rate distribution.

We have included 5 monotonic rate distributions:

1. an increasing distribution, $\rho(x) = k * x$;
2. a uniform distribution, $\rho(x) = k$;
3. a decreasing distribution with $\beta < 2$, $\rho(x) = k/x$;
4. a decreasing distribution with $\beta = 2$, $\rho(x) = k/x^2$;
5. a decreasing distribution with $\beta > 2$, $\rho(x) = k/x^3$;

with k the appropriate normalization constant. We set the constant for exponential grouping (see Section 4.4.1) to $c = 2$ in all the Multilevel Methods. For each distribution, we vary the values of r_{min} and N and generate 100 random Event Sets for each of them. On each Event Set, we perform 10^4 samplings, 10^4 updates only on the Two Levels Acceptance-Rejection and 100 updates on all the others, and we compute the average of the CPU time required over all of them. The code is compiled and executed on the following system:

¹github.com/federicodambrosio/dynamic-sampling-code

Processor: AMD Ryzen 5 3600X
RAM: 16 GB
Storage: Crucial P2 1 TB M.2-2280 NVME Solid State Drive
Compiler: gcc 9.3.0.

4.5.1 Complete Binary Trees

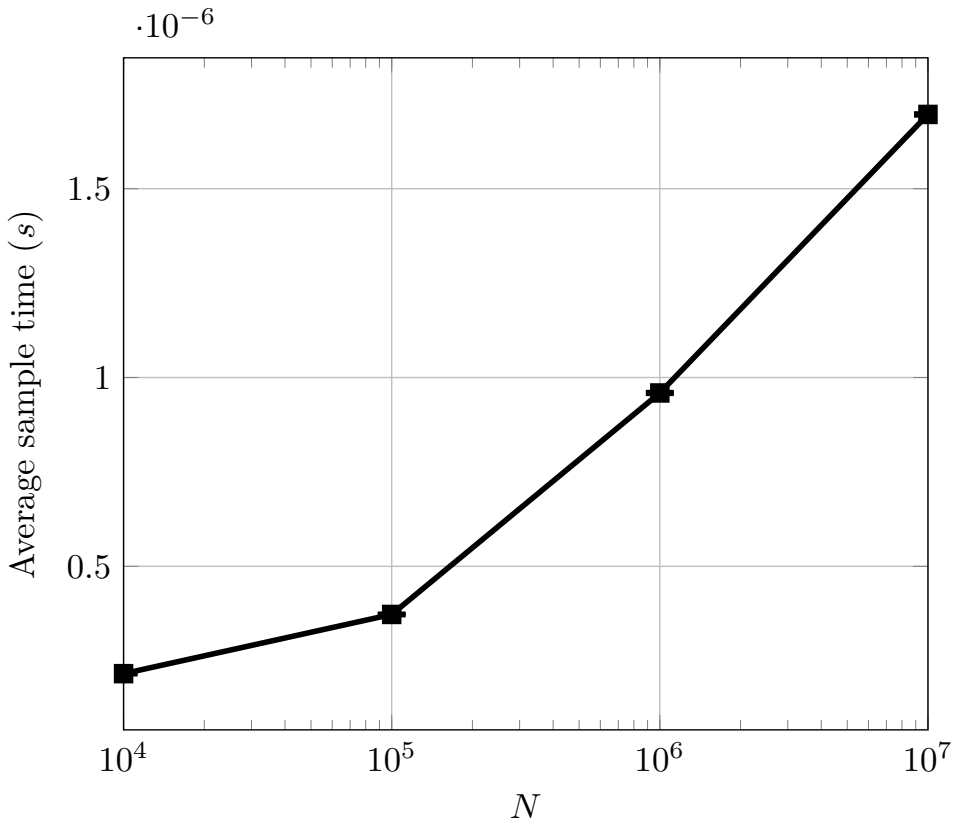


Figure 4.7: Average sample time from a Complete Binary Tree for different values of the number of events N , expressed in seconds. The x-axis is in logarithmic scale. The performance of this structure is sensitive only to the parameter N and it follows $O(\log N)$, although the performance degrades after $\approx 10^7$. Only a rate distribution (the uniform distribution) is shown, as it does not affect performance. The average update time, not shown, has a similar behaviour.

We opted for an object-oriented implementation of the Complete Binary Tree, slightly more complex than the heap-based implementation mentioned in Section 4.3.2 but more flexible. It is clear in Figure 4.7 that the sample time is proportional

to $\log(N)$, as we expected from Theorem 1. The average update time, not shown, follows the same pattern.

Performance degraded significantly when we tried to push the simulation to values of N larger than shown in Figure 4.7. One of the underlying assumptions of our work is that we have a vector-like structure that can access a random element in constant time. Once a data structure grows beyond the limits of the cache of the computer we are running our experiment on, we reach slower memory and this assumption is no longer valid.

Different implementations of this method that restrict its memory footprint can in theory allow for larger values of N before hitting the cache memory limits.

4.5.2 The Acceptance-Rejection method

We implemented the Acceptance-Rejection method with a dynamic maximum, i.e. the maximum value is set to the largest value encountered so far, which is clearly $\leq r_{\max}$. The Acceptance-Rejection performance appears insensitive to the range of rates for non-decreasing rate distributions, as we can see in Figure 4.8 and expected from Theorem 3. For decreasing rate distributions we notice that the performance degrades linearly with the ratio r_{\max}/r_{\min} , as expected from Lemma 1.

We note that it seems to be a correlation with the number of possible events N , if they follow a decreasing distribution. As more events are added to the structure, the discrete probability distribution that we sample from gets closer to the underlying rate distribution, in particular for small values of rate, which affects the performance of the structure. Similarly to what we mentioned in the previous section, we also expect further performance degradation for larger values of N once we hit the cache memory limit and we lose constant time access to the vector containing the events.

4.5.3 Tree of Groups

The Tree of Groups method has similar performance for all rate distributions and, as we can see in Figure 4.9, it does not seem to be significantly sensitive even to the range of the rates. The asymptotic behaviour of the sample time, which we expect to be $O\left(\log \log \frac{r_{\min}}{r_{\max}}\right)$, is too small to be noticeable even for extremely large values of $\frac{r_{\max}}{r_{\min}}$.

There is a minor correlation with the number of events N for the decreasing distribution that we can explain with the same arguments of the previous subsection. Nevertheless, since the complexity grows with the logarithm of the ratio the statistical effect for small values of N is significantly less dramatic than previously.

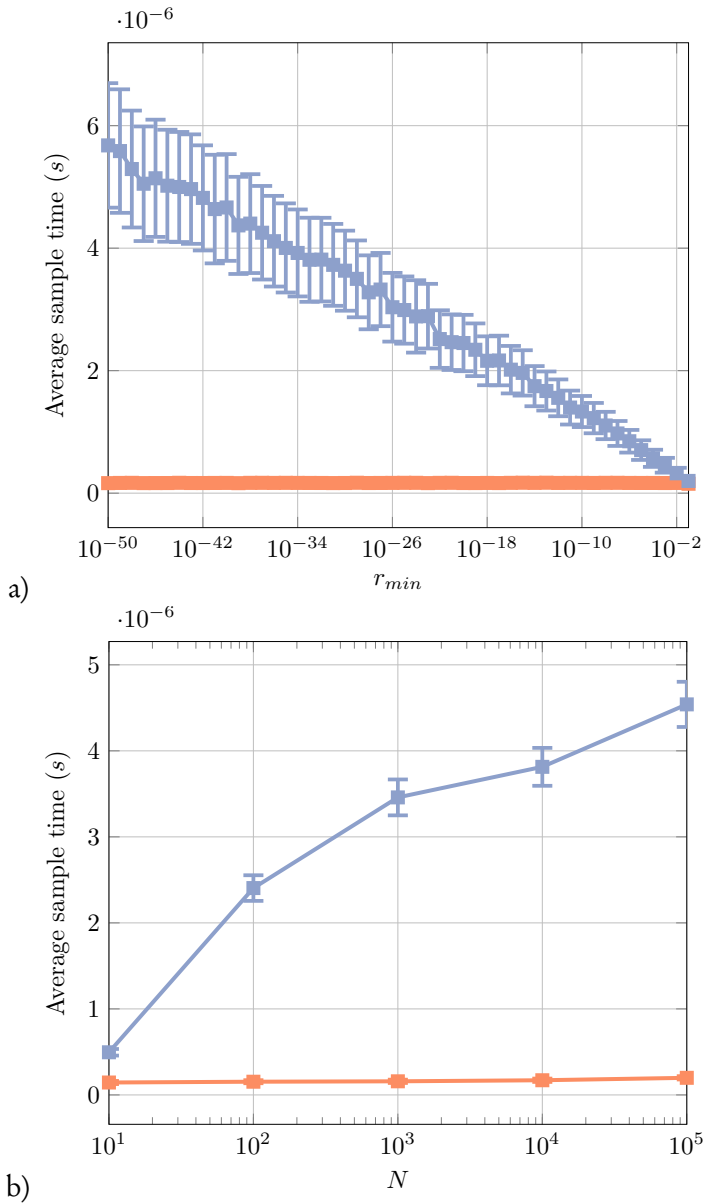


Figure 4.8: Average sample time from an Acceptance-Rejection structure for different values of the minimum rate r_{min} (a) and the number of events N (b). We show the uniform distribution (orange) and a decreasing distribution with $\beta < 2$ (light blue). The sample time clearly follows the expected logarithmic law with the ratio r_{max}/r_{min} for decreasing distributions and it is (relatively) constant for non-decreasing ones. We also note a correlation with N for decreasing distributions: as more events are added to the structure, the discrete probability distribution that we sample from gets closer to the underlying rate distribution, in particular for small values of rate, which affects the performance of the structure.

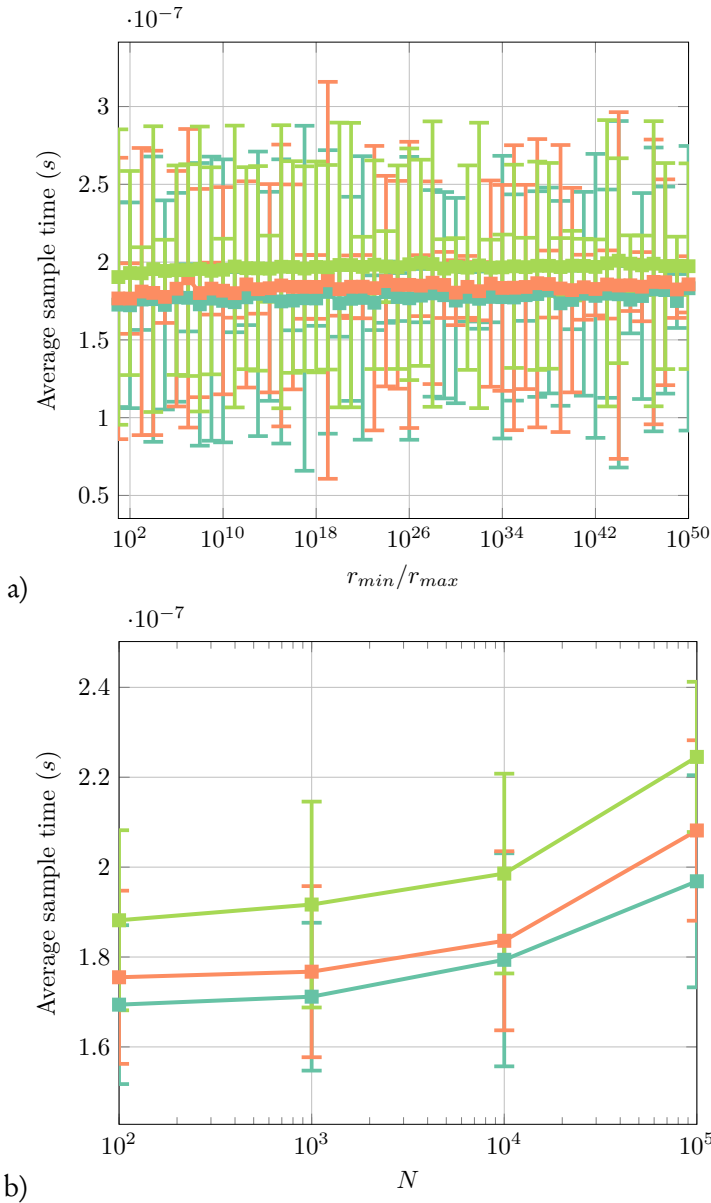


Figure 4.9: Average sample time from a Tree of Groups data structure for different values of r_{max}/r_{min} (a) and the number of events N (b). We show the increasing distribution (light blue), uniform distribution (orange) and a decreasing distribution with $\beta > 2$ (green). The expected degradation of performance for this data structure is so slow ($O(\log \log r_{max}/r_{min})$) that it appears constant with regards to r_{min} . We note that the sample time for the decreasing distribution grows with N for the previously described statistical effect and both grow for $N = 10^5$, which is when we start hitting the cache memory limit in this experiment.

Finally, the average sample time grows for all rate distributions once we reach $N = 10^5$, which is when we start hitting the cache memory limit in this experiment.

4.5.4 Cascade of Groups

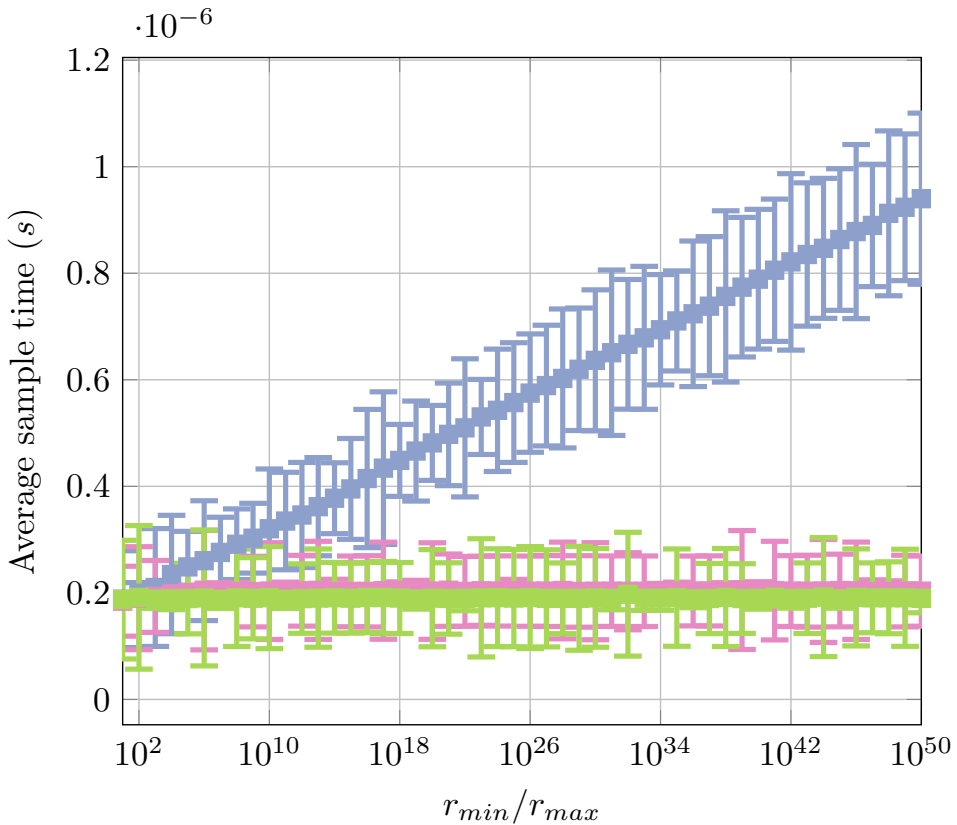


Figure 4.10: Average sample time from a Cascade of Groups data structure for different values of r_{max}/r_{min} . We show three decreasing distributions: $\beta < 2$ (blue), $\beta = 2$ (pink) and $\beta > 2$ (green). The experimental results confirms the constant time sampling for $\beta \geq 2$, which is a stronger results than our Theorem 5. The effect of N on the sampling time is negligible in comparison to the difference between rate distributions.

The Cascade of Group shows a stronger result we expected from Theorem 5. As we can see in Figure 4.10 it guarantees expected constant time sampling for rate distributions that decrease at least as fast as $1/r^2$. The effect of N on the sampling time is negligible in comparison to the difference between rate distributions.

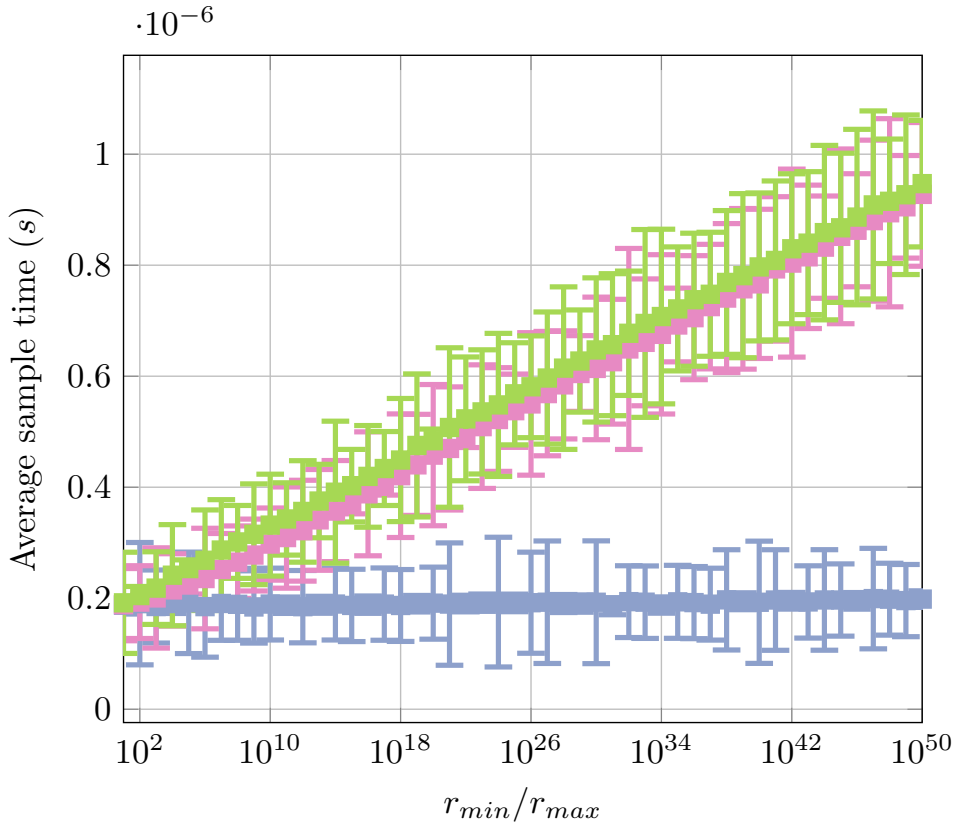


Figure 4.11: Average sample time from a Reverse Cascade of Groups data structure for different values of r_{max}/r_{min} . We show three decreasing distributions: $\beta < 2$ (blue), $\beta = 2$ (pink) and $\beta > 2$ (green). The result is, as expected, exactly the opposite of the Cascade of Groups: the sampling is performed in expected constant time for $\beta < 2$. The effect of N on the sampling time is negligible in comparison to the difference between rate distributions.

4.5.5 Reverse Cascade of Groups

Quite appropriately, the Reverse Cascade of Groups has the opposite result of the Cascade of Group; as we can see in Figure 4.11, the structure performs the sampling in expected constant time for rate distributions that decrease slower than $1/r^2$, which is in line with our Theorem 6. The effect of N on the sampling time is negligible in comparison to the difference between rate distributions.

4.5.6 Two Levels Acceptance-Rejection

We implemented the Two Levels Acceptance-Rejection that we previously described, setting $B = r_{\max}$ and $c = 2$. Theorem 7 tells us that the performance of this data structure is correlated with the amount of total rate inside it: if $R_{\text{tot}} > Bg$, both sampling and update should require constant time, regardless of the rate distribution. Events are therefore added to the data structure until this condition is satisfied, implying that N is no longer an experimental parameter in our control. We also note that the number of events required for this condition becomes rapidly large for (faster) decreasing distributions and bigger ratio ranges.

As we can see in Figure 4.12, this data structure does in fact guarantee constant time samples and updates, albeit with some variability, as long as the condition on the amount of rate is satisfied. Unfortunately, the number of events required for decreasing distributions quickly fill the cache memory and already for $r_{\max}/r_{\min} = 10^4$, in our specific implementation and system, we lose the expected constant time performance.

4.6 Discussion and outlook

In this work, we have presented two basic data structures for sampling from a discrete probability distribution, the Acceptance-Rejection method and the Complete Binary Tree, and used them as building blocks for some multi-level data structures for the dynamic case: the Tree of Groups, the Cascade of Groups and Two Levels Acceptance-Rejection.

We have proved, under our assumptions, constant time sampling and updates for different classes of rate distributions and a generic result that requires an assumption on the amount of rate in the structure. These results have been confirmed by our experimental analysis, which has also highlighted the practical advantages of the Tree of Groups when faced with real-life constraints and the downsides of the theoretically optimal Two Levels Acceptance-Rejection. Multilevel methods allowed us both to optimize the sampling to the particular conditions of the problem, and obtain significant general results.

While inspired by a practical application, our set of assumptions is arbitrary. Fur-

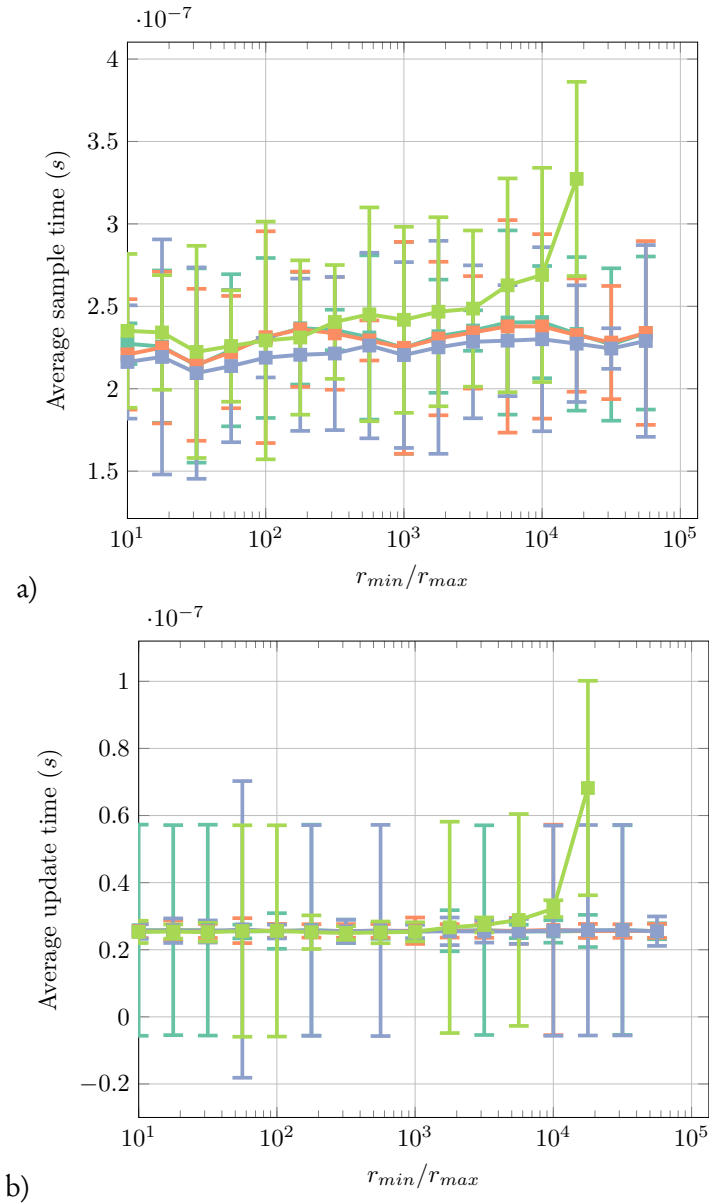


Figure 4.12: Average sample (a) and update (b) time from a Two Levels Acceptance-Rejection data structure for different values of r_{\max}/r_{\min} , constructed such that $R_{\text{tot}} > B g$. We show four distributions: an increasing distribution (light blue), a uniform distribution (orange), a decreasing distribution with $\beta < 2$ (blue) and one with $\beta > 2$ (green). Under its assumption, this method guarantees both constant time update and sampling, but we note that the number of events required to satisfy the condition grows with both the ratio r_{\max}/r_{\min} and faster decreasing distributions, to the point where the cache memory limit is clearly hit at around $r_{\max}/r_{\min} = 10^4$ for $\beta > 2$.

ther study is warranted for other sets of assumptions, both inspired by theoretical interest and realistic applications. For instance, we could make assumptions on the updates and assume that the rates are increased or decreased by a known constant quantity when updated while removing other assumptions.

CHAPTER 5

An Event-Based model for polycrystalline graphene

Abstract - The structural evolution of polycrystalline graphene can be simulated following the method of Wooten, Winer and Weaire, originally intended for simulations of amorphous silicon. Bond transpositions are randomly proposed and either accepted or rejected following the Metropolis probability. In a well-relaxed sample, almost all bond transpositions are rejected. The simulation grinds to a halt, effectively introducing a floor to the simulated energy.

In this chapter, we consider an Event-Based approach for simulating the evolution of polycrystalline graphene as a more performing alternative for well-relaxed samples, as its performance is not correlated to the state of the sample. Furthermore, the bulk of the calculations under this method lends itself to a straightforward parallel implementation.

Under the assumptions that the transition rates can be computed from just the neighborhood surrounding the relative bond transposition, two further techniques are introduced. In the Local Event-Based technique, the transition rates are determined by relaxing the degrees of freedom of only the atoms in the neighborhood of those involved in the bond transposition of interest; only transitions involving atoms in the neighborhood of those involved in an accepted bond transposition are recomputed. We further assume that the graphs described by the atoms in the neighborhood of a bond transposition determines its rate and introduce the Topological Event-Based technique which reduces the number of required computations by assigning the same rate to bond transpositions presenting the same local graph.

5.1 Introduction

Graphene is an allotrope of carbon, shaped like a honeycomb lattice. It presents a large set of exotic properties, both mechanical and electronic, and is a two-dimensional crystal [16–23]. Large samples experimentally produced are typically polycrystalline, with both intrinsic [40, 54, 55] and extrinsic [56] lattice defects. Structural defects are common in graphene [28], as it can easily host them thanks to the flexibility of the carbon atoms in hybridization and they can easily be frozen in the sample during the annealing process [29–31]. Study of these defects is warranted as they both have a significant detrimental effect on the properties expected from pristine graphene [33, 34], and they lead to new and different effects [35–38].

Polycrystalline graphene can be studied with continuous random network (CRN) models [48], following the method from Wooten, Winer, and Weaire (WWW), an explicit algorithm initially introduced to simulate samples of amorphous Si and Ge [46, 50]. In the WWW approach, a configuration consists of a list of the (three-dimensional) coordinates of all N atoms, coupled with an explicit list of the bonds between them. Atoms are allowed to occupy any point in the three-dimensional space; the only constraint is that each atom is always perfectly coordinated (in the case of graphene, all atoms have exactly three bonds), a fair assumption given that unsaturated carbon bonds are energetically very costly [28]. Molecular dynamics allows us to follow the actual dynamical evolution of the system, once a potential is defined, such as the empirical potential for polycrystalline graphene proposed by Jain et al. [48]:

$$E = \frac{3}{16} \frac{\alpha}{d^2} \sum_{i,j} (r_{ij}^2 - d^2)^2 + \frac{3}{8} \beta d^2 \sum_{j,i,k} \left(\theta_{j,i,k} - \frac{2\pi}{3} \right)^2 + \gamma \sum_{i,jkl} r_{i,jkl}^2 \quad (5.1)$$

with r_{ij} the length of the bond between atoms i and j , $\theta_{j,i,k}$ the angle centered on the atom i between the bonds with atoms j and k , and $r_{i,jkl}$ the distance between the atom i and the plane described by the three atoms j, k, l to which it is bounded. The ideal bond-length of graphene is $d = 1.420 \text{ \AA}$. The other parameters, extracted from DFT calculations [48], are $\alpha = 26.060 \text{ eV/\AA}^2$, $\beta = 5.511 \text{ eV/\AA}^2$ and $\gamma = 0.517 \text{ eV/\AA}^2$. Further terms can be added as need, for instance the substrate term described in Chapter 2 or the straining term described in Chapter 3.

The molecular dynamics algorithm can be altered in such a way to converge the system to a minimum faster than just integrating the equations of motion. For instance, FIRE (Fast Inertial Relaxation Engine) achieves significantly faster relaxation than ordinary molecular dynamics [59]. The system is frozen (i.e. all velocities are set to zero) if $\hat{v} \cdot \hat{F} < 0$, which means that the system is moving away from a minimum; this prevents it from oscillating around a minimum. The direction of

the velocity vector is also mixed at each step with that of the force vector, pointing the system slightly towards the minimum. Both the magnitude of the mixing and the size of the time steps are increased while the system is moving towards a minimum and sharply decreased when it overtakes it.

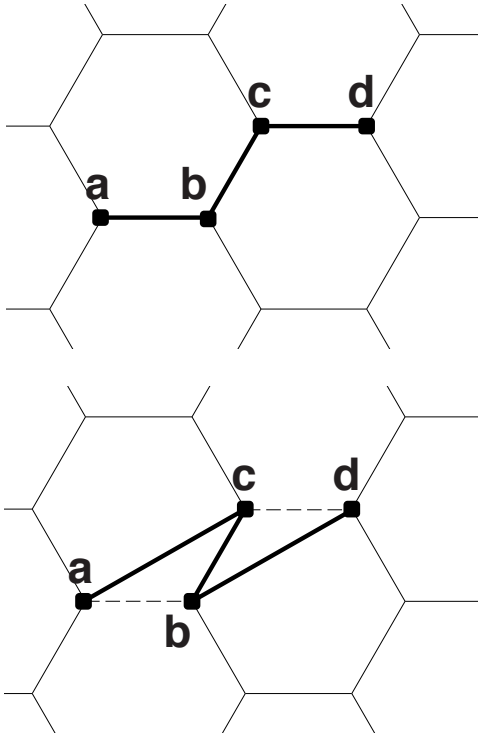


Figure 5.1: A bond transposition according to the Wooten, Winer, and Weaire (WWW) algorithm. Four consecutively connected atoms (a , b , c and d) are selected; two bonds (ab and cd) are switched, maintaining the sample perfectly coordinated.

Through this process, we are able to resolve individual atomic vibrations, which requires time-steps of the scale of a few femtoseconds. Structural changes (i.e. changes in the bond list), however, have a significantly larger characteristic time, which puts them beyond the reach of our computational capabilities, and would require a more complex potential, without an explicit bond list. In the WWW approach, the structure of the sample is instead allowed to evolve through discrete events denominated *bond transpositions* in which, as shown in Figure 5.1, two bonds are switched among four atoms which are consecutively connected (the first is neighbour to the second, etcetera) and distinct (i.e. the same atom does not appear twice in the sequence). A sequence of four atoms that satisfies these conditions identifies a bond transposition. It is equivalent to its reverse

$\{(a, b, c, d) \equiv \{d, c, b, a\}\}$, as both sequences break the same bonds $((a, b)$ and $(c, d))$, replacing them with (a, c) and (b, d) .

In amorphous silicon, a material similar to polycrystalline graphene, WWW bond transpositions have been shown to be the predominant mechanism of structural evolution of graphene [96]. While this confirmation is missing for polycrystalline graphene, it has been shown that realistic samples of graphene can be produced through just WWW bond transpositions, starting from a random seed [48, 53]. The difference in characteristic time allows us to decouple the two processes and study the structural changes in polycrystalline graphene with the Transition State Theory formalism.

5.1.1 Transition State Theory in graphene

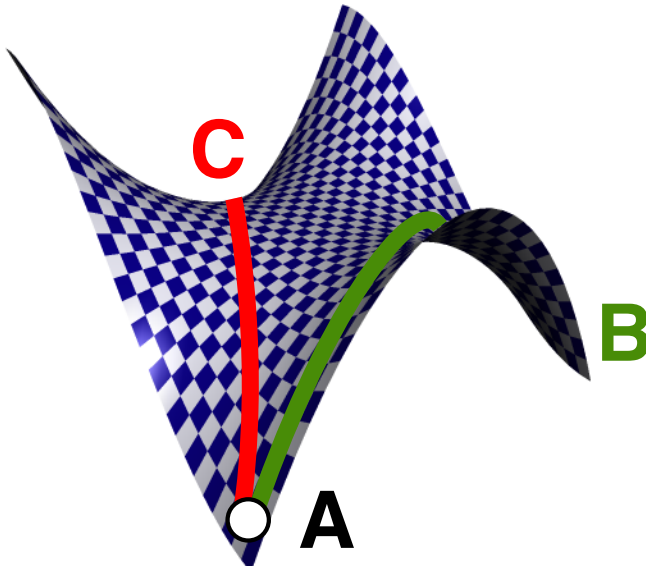


Figure 5.2: Schematic illustration of the Transition State Theory approach. In the configuration space, the system lies in the basin of the minimum A until it manages to cross one of the surrounding saddle points and falls into the basin of another minimum, either B or C depending on which saddle point it crosses, where it quickly settles before the sequence can begin again. According to TST, the transition rate between two states is determined by the activation energy, i.e. the energy difference between the current state and the saddle point between the two states.

In the configuration space, the system vibrates within the basin of a minimum and only occasionally jumps to another one, where it quickly settles around the minimum before the sequence can begin again [97]. In the Transition State Theory (TST) formalism, the transition between two different states A and B is a Poisson

process with a rate equal to the equilibrium flux through the dividing surface between A and B in the configuration space [97–100]. Close to the minimum, the potential energy is well described by a harmonic energy expansion and the rate of a transition from the state A to the state B can be written following Arrhenius' Law [98, 101]:

$$k_{A \rightarrow B} = \nu_0 \exp(-E_a/k_B T) = \nu_0 \exp\left(-\frac{E_{S_{AB}} - E_A}{k_B T}\right). \quad (5.2)$$

Here, E_a is the energy difference between the saddle point S_{AB} and the initial state A , T is the temperature of the system (which is considered at thermal equilibrium) and k_B is the Boltzmann's constant and ν_0 a constant with the dimension of a rate, which connects the time scale of the simulation to that of the real dynamics of the process.

At least the order of magnitude of the characteristic vibrational frequency ν_0 can be estimated from Raman spectrography. The most prominent modes in polycrystalline graphene are the G band, related to the covalent carbon-carbon bonds, and the defect-induced D band. They show a Raman shift (expressed in wavenumbers) of respectively 1582 and 1350 cm^{-1} , when excited by a laser with wavelength 514 nm [102]. The corresponding wavelength for both bands is around 550 nm, which suggests a characteristic vibrational frequency ν_0 in the order of 10^{14} Hz.

Once the saddle point is crossed, the system inevitably falls in the next basin, no recrossings are allowed. It quickly settles around its minimum: before the sequence can begin again, the system has lost memory of the trajectory that led it to that state. Under these assumptions, Eq (5.2) is the exact rate [97, 103] and detailed balance at equilibrium is preserved:

$$\frac{k_{A \rightarrow B}}{k_{B \rightarrow A}} = \frac{\exp(-(E_{S_{AB}} - E_A)/k_B T)}{\exp(-(E_{S_{AB}} - E_B)/k_B T)} = \frac{\exp(-E_B/k_B T)}{\exp(-E_A/k_B T)} = \frac{P_B^{\text{eq}}}{P_A^{\text{eq}}}. \quad (5.3)$$

In general, the energy of the saddle point has to be higher than that of both states involved in the transition. If we assume that it is always larger than their maximum of a fixed amount, i.e.

$$E_{S_{AB}} \approx \max\{E_A, E_B\} + \Delta E \quad (5.4)$$

for some constant energy ΔE , we can write the rate for a generic transition from a state A to a state B as:

$$k_{A \rightarrow B} = \nu_0 \exp(-\Delta E/k_B T) \min\left\{1, \exp\left(-\frac{E_B - E_A}{k_B T}\right)\right\}. \quad (5.5)$$

Under this strong, but sensible, assumption, a transition can be interpreted as a continuous Metropolis process [104–106] with rate

$$k_0 = \nu_0 \exp\left(-\frac{\Delta E}{k_B T}\right), \quad (5.6)$$

where a random transition to a state B is selected from an uniform distribution and it is accepted according to its Metropolis probability:

$$P_{\text{Met}}(A, B) = \min\left\{1, \exp\left(-\frac{E_B - E_A}{k_B T}\right)\right\}. \quad (5.7)$$

This expression can be further simplified if we are not interested in the relationship between simulated (Monte Carlo) and real-time by setting $k_0 = 1$ in some arbitrary unit, resulting in the transition rate being equal to the Metropolis probability in Monte Carlo time units.

5.1.2 Performance challenges

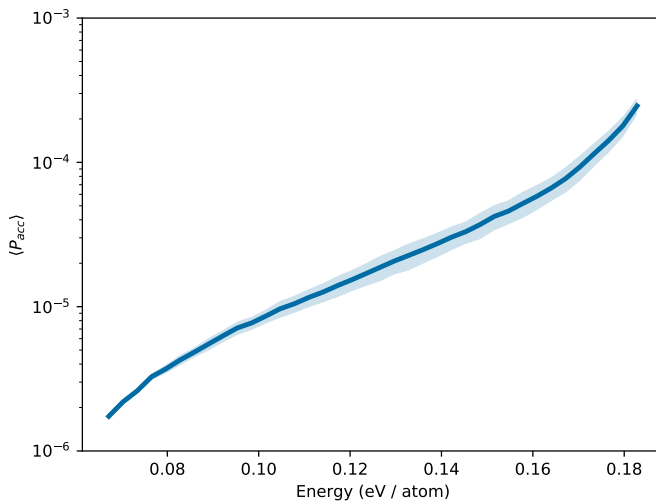


Figure 5.3: Average acceptance probability (i.e. the inverse of the average number of attempts required per accepted bond transposition) at different energy densities for the sample of 3200 atoms studied in Chapter 2. The shaded area shows one standard deviation from the average. An accepted bond transposition requires between 5'000 (at 0.18eV/atom) and almost 600'000 (at 0.07eV/atom) attempts on average.

The approach that we have discussed so far is quite powerful if a significant fraction of attempted bond transpositions is accepted. A sample that is far from equilibrium, such as the initial random seed, evolves quickly as most of the changes would

lower its energy significantly and are, therefore, (deterministically) accepted. Realistic samples are at, or very close to, equilibrium; most of the possible changes would increase its energy and it usually takes hundreds of attempts before a single bond transposition is accepted.

In Chapter 2 we have introduced a few techniques to accelerate the selection process, discarding bond transpositions earlier than before completing molecular dynamics on the entire sample. While the average time spent on a rejected bond transposition is significantly reduced, their number is unchanged. As a sample is relaxed to lower energies, the average acceptance probability reaches extremely small average acceptance probabilities, as shown in Figure 5.3, necessitating a large number of attempts in order to further evolve the sample. At very low energies, the simulation grinds to a halt, introducing an effective floor to the energy that can be reached. Furthermore, this approach does not easily lend itself to parallel computing. While the molecular dynamics itself can be parallelised [51], the evolution of the sample must proceed sequentially, as it would not be simple to reconcile different accepted transitions. A different approach is needed.

5.1.3 An Event-Based approach

In this chapter, we explore the possibility of replacing the acceptance-rejection algorithm with a sampling algorithm from a list of bond transpositions with precomputed rates. We introduce an assumption that limits the number of bond transpositions that are recomputed after each structural change to a constant and describe a method based on this assumption. We then introduce a stronger assumption that connects the rate of a possible bond transposition to the graph described by the atoms in its surroundings and describe a method based on this assumption that would be an efficient solution for the simulation of polycrystalline graphene at low energies.

5.2 Methods

In an Event-Based approach, the list of available events from the current state of the system together with their rate is explicitly maintained.

The event set can be generated iteratively, given the bond list and the rules of WWW bond transpositions. Under the assumption of Eq. (5.4), the rate of a bond transposition can be computed in a way similar to the acceptance probability in the acceptance-rejection approach. In a sample in a state A , relaxed with molecular dynamics to an energy minimum E_A , each possible bond transposition is attempted and the coordinates of the sample are allowed to relax with molecular dynamics to

an energy minimum E_B . The rate is then computed as:

$$k_{A \rightarrow B} = k_0 \min \left\{ 1, \exp \left(-\frac{E_B - E_A}{k_B T} \right) \right\}, \quad (5.8)$$

in Monte Carlo time units. The bond transposition is then reverted to the state A and the atoms restored to their initial positions. As a result of this approximation, all possible bond transpositions that reduce the energy of the system have the same rate, in the same way that in the WWW approach they have an equal probability of being selected and accepted.

On the same central bond, (b, c) in our example, insist four unique possible bond transpositions: $\{a, b, c, d\}$, $\{a', b, c, d\}$, $\{a, b, c, d'\}$ and $\{a', b, c, d'\}$. Since each atom is coupled to three other atoms, from each state there are $3/2 \times 4 \times N$ bond transpositions available, minus those that are prohibited because they would be labeled by a sequence with repeated atoms, a small number in a well-relaxed sample.

The sampling itself can be performed with one of the data structures discussed in Chapter 4. The important quantities for determining the best solution for our problem are the size of the event set, which as we have shown is $O(N)$, and the maximum and minimum rate allowed; the rate has a ceiling at $r_{\max} = 1$, in arbitrary Monte Carlo time units, while there is no theoretical floor r_{\min} . Nevertheless, it is hard for a well-relaxed sample to see an increase of energy for more than a few eVs in a single bond transpositions; we can place a (generous) higher bound to the energy increase at $\approx 10\text{eV}$, resulting in a rate ratio of $r_{\max}/r_{\min} \approx 10^{-17}$: if the rate of an energy-reducing event is of the order of the ns^{-1} , as it is the typical scale of such phenomenon, the rate of an event beyond the cutoff would occur, on average, every 1157 days (3.17 years). We can either consider these events practically forbidden and discard them or place them in a different group, adding a step to the sampling process: stochastically determine whether we are going to sample from the allowed ($r > r_{\min}$) or rare ($r < r_{\min}$) events.

It could be sensible, for a very small sample, to implement the event sampling as either a Complete Binary Tree (see Chapter 4, Section 4.3.2) with expected sample and update time $O(\log N)$. In any other case, due to the empirical limits of the range of rates, it is more appropriate to implement the event sampling as a Tree of Groups (see Chapter 4, Section 4.4.2) with expected sample time proportional to $O(\log \log r_{\max}/r_{\min})$ and constant update time.

The transition is then applied to the system and the simulated time is increased by a quantity [107]

$$\Delta t = -\frac{\ln(u)}{\sum_{i \in \mathbb{E}} r_i}, \quad (5.9)$$

with $u \in [0, 1]$ a random number. The coordinates of the system are then relaxed with molecular dynamics.

The system is now in a different state with a different topology. It is not possible to just take the previous event set and sample a new event from it. The list of available bond transpositions was computed from the previous bond list, which has now changed. The rate of a bond transposition was computed from the energy difference between the previous state and the state that would have been reached by applying it to the previous state.

Before sampling a transition, it is, therefore, necessary to again identify and compute the rate of all $6 \times N$ available events: a huge computational cost to pay for sampling a single event. This is especially clear when compared to the acceptance-rejection approach. Let us assume that a well-relaxed sample in a WWW simulation has a very small Metropolis acceptance probability P_{accept} ; the average number of attempts before a bond transposition is finally accepted is $\tau_W = (P_{\text{accept}})^{-1}$, each of them requiring a single relaxation with molecular dynamics. With an Event-Based simulation, no molecular dynamics is necessary to sample a transition; after a bond transposition is applied, though, rebuilding the event list requires $\tau_E \approx 6 \times N$ relaxations with molecular dynamics, one for each possible bond transposition, with N the number of atoms in the sample. For samples with more atoms than

$$N_{\text{th}} = \frac{\tau_W}{6} = \frac{1}{6 P_A} \quad (5.10)$$

such an Event-Based model will be on average slower. Even if one of every 10,000 attempted bond transpositions is accepted, a WWW model will be faster on average if the sample has at least 1667 atoms, half the size of the sample we have studied in Chapter 3. Such a model would be competitive only for either very small samples, where the long-range effects are barely, if at all, visible, or extremely well-relaxed samples at low temperatures.

The Event-Based approach naturally lends itself to a parallel implementation. Most of the computational time is spent in calculating transition rates: we can consider each of them a separate task, independent from each other, that is performed concurrently, while the sampling is still sequential. Nevertheless, for large samples, the performance gain from a parallel implementation would be insufficient to make a difference.

In this section, we describe two methods that would curb this cost and allow for an efficient Event-Based simulation of polycrystalline graphene, at least for reasonably-relaxed samples.

5.2.1 Local Event-Based technique

As we discussed in Chapter 2, most of the energy lost during the relaxation is localised in the neighborhood of the applied bond transposition (i.e. atoms up to a hopping distance l from those involved in it). After applying a possible bond transposition, we allow only the atoms in its neighborhood to relax with molecular dynamics to an energy minimum E_B^{loc} . Assuming a harmonic approximation close to the minimum, we can estimate the energy that the sample would reach if fully relaxed E_B as

$$\tilde{E}_B \approx E_B^{\text{loc}} - c_f |\hat{F}|^2 \quad (5.11)$$

with E_B^{loc} the energy of the system after local relaxation, c_f an empirically determined constant and \hat{F} the global force vector. Since we are not relaxing the entire sample, the computational time required is not dependent on the size of the system, but only on the constant l (see Table 5.1), resulting in a significant performance improvement for larger samples.

Events are sampled according to the approximated rate, obtained from the approximated relaxed energy:

$$\tilde{k}_{A \rightarrow B} = k_0 \min \left\{ 1, \exp \left(-\frac{\tilde{E}_B - E_A}{k_B T} \right) \right\}. \quad (5.12)$$

After selecting an event, we correct for the exact rate by allowing the entire sample to relax completely to the actual relaxed energy E_B . Then, the proposed structural change is accepted according to a probability

$$P_{A \rightarrow B} = \min \left\{ 1, \frac{k_{A \rightarrow B}}{\tilde{k}_{A \rightarrow B}} \right\}, \quad (5.13)$$

with $k_{A \rightarrow B}$ the exact rate. Meanwhile, the average ratio

$$R = \left\langle \frac{k_{A \rightarrow B}}{\tilde{k}_{A \rightarrow B}} \right\rangle \quad (5.14)$$

is monitored during the simulation, as a diagnostic. Only one global relaxation is required for each sampled bond transposition. This computational cost can be amortised by performing a global relaxation only after a certain number of samplings or, similarly to some of the techniques described in Chapter 2, when a certain threshold of $|\hat{F}|^2$ is reached, at the risk of possibly introducing artefacts. In these cases, if the average ratio R strays significantly from 1, the details of the amortisation should be reassessed.

| l | $\langle N_{\text{loc}} \rangle$ | P_A |
|-----|----------------------------------|-------|
| 2 | 28 | 0.5% |
| 3 | 53 | 0.3% |
| 4 | 90 | 0.2% |

Table 5.1: Average number of atoms involved in local relaxation $\langle N_{\text{loc}} \rangle$ for different local relaxation distances l (see Table 2.1). The number of relaxations required is approximately $6 \times \langle N_{\text{loc}} \rangle$, which is equivalent to the expected number of attempts required for an Early Decision simulation (see Chapter 2) with acceptance probability P_A . The Local Event-Based method becomes competitive for well-relaxed samples with an acceptance probability significantly below 1%, regardless to their size, without taking into account any performance gain from parallelisation of the computation of transition rates.

Under the assumptions of this method, atoms not belonging to the neighborhood of a transition have no effect on the rate of such transition. After a bond transposition has been sampled and applied to the sample, the transitions involving atoms belonging to its neighborhood are deleted and reconstructed iteratively, with their rates recomputed accordingly. The rates of transitions that have atoms in their neighborhood that are also in that of a sampled transition, without being involved in either, are not recomputed as the change is expected to be negligible and can be corrected through the acceptance process after a global relaxation.

If there are N_{loc} atoms in the neighborhood of a sampled transition, rebuilding the event list requires $\tau_{\text{loc}} = 6 \times N_{\text{loc}}$ global relaxations, regardless of the size of the sample. The performance of this method is constant with the size of the sample and is dependent only on the hopping radius of the neighborhood l . Nevertheless, the number of local relaxations required is quite large, even without accounting for some occasional rejected events. As we show in Table 5.1, for reasonable values of l the performance of this method is comparable to the Early Decision method from Chapter 2 when the acceptance probability is between 0.2 and 0.5%, ignoring the occasional global relaxations in both methods, and faster for more relaxed samples, without taking into account the performance gain from a parallel implementation of the computation of transition rates.

5.2.2 Topological Event-Based technique

We further assume that, if the neighborhoods of two proposed bond transpositions have the same topology, their transition rate is approximately the same. This is equivalent to assuming that, given two neighborhood-sized samples with the same topology, they will reach approximately the same energy through molecular dynamics. We expect this to be a reasonable assumption, at least when the sample

is not subjected to other processes in the meanwhile. For instance, if a stretching force is applied to the sample, it can access a different minimum with the same topology, as we showed in Chapter 3.

Under this technique, before computing the (approximated) rate of a transition according to Equation (5.12), we check whether we have already encountered the graph described by the bonds between the atoms in the neighborhood. Similarly to the k-ART technique [108], an event with local graph G is labelled by the canonical form $Canon(G)$; since the canonical forms of two graphs that are isomorphic are, by definition, identical, we can easily determine whether the rate for a transition isomorphic to the current one is already known. In that case, the local relaxation is not performed and the (approximated) transition rate is set equal to that of the transition with the same canonical form $Canon(G)$.

The problem to write a canonical form of a graph is computationally equally hard as determining whether two graphs are isomorphic. The computational complexity of these problems is still an open problem in computer science [109]. In particular, it has not been proven to be either NP-complete or in general solvable in polynomial time, while a polynomial-time solution is available in different special cases [110–113] and the best theoretical result for a generic graph runs in time $n^{p(\log n)}$ for some polynomial $p(X)$ [114]. Fortunately, efficient graph canonisation is implemented in publicly available tools such as nauty [115] and RDKit [116]. The catalogue can be efficiently implemented with hash tables [91, Section 11.2], which allow for both average constant time lookup and amortised constant time insertion. The space required to store it grows linearly with the number of elements in the library, which can lead to sub-optimal results in a real-life situation as the data structure becomes too big for the faster memory and it is moved to slower and slower memory.

Once all transition rates are known, either obtained through local relaxations or from graph isomorphisms, sampling is performed as usual; as in the previous technique, a single global relaxation is performed after applying the sampled bond transposition and its exact rate is computed. The proposed transition is then accepted according to Equation (5.13) in order to correct for the approximation. Only the transitions involving atoms in the neighborhood of an applied bond transposition are recomputed, as their local graph will have changed.

Careful consideration should be put in implementing this method for samples that are still very chaotic, i.e. not yet relaxed to a realistic configuration, as the number of unique local graphs will be larger, the performance cost of maintaining the catalogue of graphs larger and the gain from avoiding local relaxations smaller. Nevertheless, there is no clearly defined boundary as with the method presented

in the previous section, and, in most cases, this method should result in a significantly faster evolution of the sample than the first or the original, non Event-Based, methods.

Similarly to the previous technique, the computation of transition rates can be implemented as a series of concurrent processes, exploiting the performance of multiple processing units at the same time. It is also possible to amortise the global relaxation by performing it once a specified threshold is reached, especially for very large samples.

5.3 Conclusions

We initially set ourselves to design an Event-Based approach to simulating the evolution of polycrystalline graphene because the performance of the Metropolis-Based WWW approach slows significantly for well-relaxed samples, introducing a practical lower limit to the energy of a simulated sample. The approach we introduced in this Chapter does not suffer from this issue, as its performance is not correlated with the state of the sample, but only with the number of rates that are recomputed after each sampling while featuring the same dynamics. The bulk of the computational load under this approach lends itself to straightforward parallel implementation since the transition rates can be calculated concurrently, allowing more efficient use of the computational resources when multiple processing units are available.

We further introduced two techniques that improve the performance of this approach under the assumption that the transition rates can be computed only from the neighborhood surrounding the bond transposition, similarly to the approach we discussed in Chapter 2. The atoms in the neighborhood are relaxed with molecular dynamics in order to compute the transition rates in both methods. In the Topological Event-Based technique, if the graph describing the structure of the neighbourhood is isomorphic to one of an already computed transition rate, the relaxation is skipped and the transition rate is set equal to the already computed one. Both methods have an error correction mechanism, through an Acceptance-Rejection process.

Research on the details of the implementation, in particular the possibility of a parallel implementation, and experimental verification of the performance of the different techniques with different samples (i.e. chaotic or relaxed, large or small, etc.) are now warranted. We expect future work to be focused on this matter and on the study of larger, realistic samples of graphene that these techniques could allow.

It is important to mention that, while more simple and compatible with Metropolis dynamics, it is not necessary for the transition rates to follow Eq. (5.8) and the energy barriers to follow the approximation of Eq. (5.4). It is entirely possible to replace the potential with an explicit bond list with a potential that is also defined during a transition, explicitly compute the energy barrier during a transition and derive the transition rate from Arrhenius' Law, i.e. Equation (5.2). Such an approach, combined with the Topological Event-Based technique, could replicate the dynamics of the Kinetic Activation-Relaxation technique [108] with a significant performance boost from reducing the number of energy barriers that are estimated, which could prove itself useful in certain situations. Nevertheless, for most applications the Metropolis dynamics is sufficiently realistic, as we discussed in the Introduction of this Chapter.

Finally, although we considered the specific case of graphene, we expect that this approach will be applicable, and possibly quite effective, to the study of more amorphous materials, representing an interesting alternative to the established Wooten, Winer and Weaire approach.

Bibliography

- [1] G. M. A. Baretti, *The Italian Library. Containing an account of the Lives and Works of the Most Valuable Authors of Italy*. (A. Millar, London, 1757), p. 52.
- [2] D. Chung, "Review: Graphite", *J. Mater. Sci.* **37**, 1475 (2002).
- [3] B. C. Brodie, "On the atomic weight of graphite", *Phil. Trans. R. Soc.* **149**, 249 (1859).
- [4] P. Wallace, "The band theory of graphite", *Phys. Rev.* **71**, 622 (1947).
- [5] J. McClure, "Diamagnetism of graphite", *Phys. Rev.* **104**, 666 (1956).
- [6] J. Slonczewski and P. Weiss, "Band structure of graphite", *Phys. Rev.* **109**, 272 (1958).
- [7] A. Geim and K. Novoselov, "The rise of graphene", *Nature Mater* **6**, 183 (2007).
- [8] G. W. Semenoff, "Condensed-matter simulation of a three-dimensional anomaly", *Phys. Rev. Lett.* **53**, 2449 (1984).
- [9] E. Fradkin, "Critical behavior of disordered degenerate semiconductors. II. Spectrum and transport properties in mean-field theory", *Phys. Rev. B* **33**, 3263 (1986).
- [10] F. Haldane, "Model for a quantum hall effect without Landau levels: Condensed-matter realization of the "Parity anomaly"", *Phys. Rev. Lett.* **61**, 2015 (1988).
- [11] R. Peierls, "Quelques propriétés typiques des corps solides", in *Annales de l'Institut Henri Poincaré*, Vol. 5, 3 (1935), pp. 177–222.
- [12] L. D. Landau, "Zur Theorie der Phasenumwandlungen", *Phys. Z. Sowjetunion* **11**, 26 (1937).
- [13] N. Mermin, "Crystalline order in two dimensions", *Phys. Rev.* **176**, 250 (1968).
- [14] K. Novoselov, "Electric field effect in atomically thin carbon films", *Science* **306**, 666 (2004).
- [15] *The Nobel prize in Physics 2010*, (2010) <https://www.nobelprize.org/prizes/physics/2010/summary/> (visited on 06/02/2021).
- [16] A. Castro Neto et al., "The electronic properties of graphene", *Rev. Mod. Phys.* **81**, 109 (2009).
- [17] A. Geim, "Graphene: Status and prospects", *Science* **324**, 1530 (2009).
- [18] R. Nair et al., "Unimpeded permeation of water through helium-leak-tight graphene-based membranes", *Science* **335**, 442 (2012).
- [19] A. Smith et al., "Electromechanical piezoresistive sensing in suspended graphene membranes", *Nano Lett.* **13**, 3237 (2013).
- [20] R. J. Dolleman et al., "Graphene squeeze-film pressure sensors", *Nano Lett.* **16**, 568 (2015).
- [21] S. J. Cartamil-Bueno et al., "Colorimetry technique for scalable characterization of suspended graphene", *Nano Lett.* **16**, 6792 (2016).
- [22] C. Lee et al., "Measurement of the elastic properties and intrinsic strength of monolayer graphene", *Science* **321**, 385 (2008).

- [23] K. Z. Milowska, M. Wońska, and M. Wierzbowska, “Contrasting elastic properties of heavily b- and n-doped graphene with random impurity distributions including aggregates”, *J. Phys. Chem. C* **117**, 20229 (2013).
- [24] “Moving towards the market”, *Nat. Mater.* **18**, 519 (2019).
- [25] L. Lin, H. Peng, and Z. Liu, “Synthesis challenges for graphene industry”, *Nat. Mater.* **18**, 520 (2019).
- [26] Zion Market Research, *Graphene market (Mono-layer and bi-layer graphene, few layer graphene, graphene oxide and graphene nano platelets) for composites, energy storage, electronics and others applications: Global industry perspective, comprehensive analysis, and forecast*, tech. rep. (2016).
- [27] T. Foley et al., *Graphene Flagship Annual Report 2020*, tech. rep. (Graphene Flagship, European Commission, 2020).
- [28] F. Banhart, J. Kotakoski, and A. V. Krasheninnikov, “Structural defects in graphene”, *ACS Nano* **5**, 26 (2010).
- [29] A. Hashimoto et al., “Direct evidence for atomic defects in graphene layers”, *Nature* **430**, 870 (2004).
- [30] J. C. Meyer et al., “Direct imaging of lattice atoms and topological defects in graphene membranes”, *Nano Lett.* **8**, 3582 (2008).
- [31] J. Kotakoski et al., “From point defects in graphene to two-dimensional amorphous carbon”, *Phys. Rev. Lett.* **106**, 105505 (2011).
- [32] L. Vicarelli et al., “Controlling defects in graphene for optimizing the electrical properties of graphene nanodevices”, *ACS Nano* **9**, 3428 (2015).
- [33] X. Du et al., “Approaching ballistic transport in suspended graphene”, *Nature Nanotech* **3**, 491 (2008).
- [34] K. Bolotin et al., “Ultrahigh electron mobility in suspended graphene”, *Solid State Commun.* **146**, 351 (2008).
- [35] J. Lu et al., “Properties of strained structures and topological defects in graphene”, *ACS Nano* **7**, 8350 (2013).
- [36] L. Liu et al., “Defects in graphene: Generation, healing, and their effects on the properties of graphene: A review”, *J. Mater. Sci. Technol.* **31**, 599 (2015).
- [37] S. Güryel et al., “Effect of structural defects and chemical functionalisation on the intrinsic mechanical properties of graphene”, *Phys. Chem. Chem. Phys.* **15**, 659 (2013).
- [38] F. D’Ambrosio, V. Juričić, and G. T. Barkema, “Discontinuous evolution of the structure of stretching polycrystalline graphene”, *Phys. Rev. B* **100**, 161402 (2019).
- [39] A. Stone and D. Wales, “Theoretical studies of icosahedral c60 and some related species”, *Chem. Phys. Lett.* **128**, 501 (1986).
- [40] O. V. Yazyev and Y. P. Chen, “Polycrystalline graphene and other two-dimensional materials”, *Nature Nanotech* **9**, 755 (2014).
- [41] A. Fasolino, J. Los, and M. Katsnelson, “Intrinsic ripples in graphene”, *Nature Mater* **6**, 858 (2007).
- [42] T. Zhang, X. Li, and H. Gao, “Defects controlled wrinkling and topological design in graphene”, *J. Mech. Phys. Solids* **67**, 2 (2014).
- [43] O. Lehtinen et al., “Atomic scale study of the life cycle of a dislocation in graphene from birth to annihilation”, *Nat Commun* **4**, 2098 (2013).
- [44] J. H. Warner et al., “Rippling graphene at the nanoscale through dislocation addition”, *Nano Lett.* **13**, 4937 (2013).
- [45] W. Zachariasen, “The atomic arrangement in glass”, *J. Am. Chem. Soc.* **54**, 3841 (1932).

- [46] F. Wooten, K. Winer, and D. Weaire, “Computer generation of structural models of amorphous Si and Ge”, *Phys. Rev. Lett.* **54**, 1392 (1985).
- [47] G. T. Barkema and M. E. Newman, *Monte Carlo Methods in Statistical Physics* (Clarendon Press, Oxford, 1999).
- [48] S. K. Jain et al., “Strong long-range relaxations of structural defects in graphene simulated using a new semiempirical potential”, *J. Phys. Chem. C* **119**, 9646 (2015).
- [49] G. Voronoi, “Nouvelles applications des paramètres continus a la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les paralléloèdres primitifs.”, *Journal für die reine und angewandte Mathematik (Crelle’s Journal)* **1908**, 198 (1908).
- [50] F. Wooten and D. Weaire, “Modeling tetrahedrally bonded random networks by computer”, *Solid State Physics - Advances in Research and Applications* **40**, 1 (1987).
- [51] P. van Dieten, “Computer simulations of vibrations induced by the stretching of polycrystalline graphene”, Master’s thesis (Utrecht University, 2019).
- [52] G. Barkema and N. Mousseau, “High-quality continuous random networks”, *Phys. Rev. B* **62**, 4985 (2000).
- [53] F. D’Ambrosio, J. Barkema, and G. T. Barkema, “Efficient structural relaxation of polycrystalline graphene models”, *Nanomaterials* **11**, 1242 (2021).
- [54] H. I. Rasool et al., “Conserved atomic bonding sequences and strain organization of graphene grain boundaries”, *Nano Lett.* **14**, 7057 (2014).
- [55] Y. Tison et al., “Grain boundaries in graphene on SiC(0001) substrate”, *Nano Lett.* **14**, 6382 (2014).
- [56] P. T. Araujo, M. Terrones, and M. S. Dresselhaus, “Defects and impurities in graphene-like materials”, *Mater. Today* **15**, 98 (2012).
- [57] N. Metropolis et al., “Equation of state calculations by fast computing machines”, *J. Chem. Phys.* **21**, 1087 (1953).
- [58] W. Hastings, “Monte Carlo sampling methods using Markov chains and their applications”, *Biometrika* **57**, 97 (1970).
- [59] E. Bitzek et al., “Structural relaxation made simple”, *Phys. Rev. Lett.* **97**, 170201 (2006).
- [60] M. Matsumoto, A. Baba, and I. Ohmine, “Topological building blocks of hydrogen bond network in water”, *J. Chem. Phys.* **127**, 134504 (2007).
- [61] T. Witten, “Stress focusing in elastic sheets”, *Rev. Mod. Phys.* **79**, 643 (2007).
- [62] C. Si, Z. Sun, and F. Liu, “Strain engineering of graphene: A review”, *Nanoscale* **8**, 3207 (2016).
- [63] J. G. Kirkwood, “The skeletal modes of vibration of long chain molecules”, *J. Chem. Phys.* **7**, 506 (1939).
- [64] S. K. Jain, V. Juričić, and G. T. Barkema, “Probing crystallinity of graphene samples via the vibrational density of states”, *J. Phys. Chem. Lett.* **6**, 3897 (2015).
- [65] S. K. Jain, V. Juričić, and G. T. Barkema, “Structure of twisted and buckled bilayer graphene”, *2D Mater.* **4**, 015018 (2016).
- [66] S. K. Jain, V. Juričić, and G. T. Barkema, “Probing the shape of a graphene nanobubble”, *Phys. Chem. Chem. Phys.* **19**, 7465 (2017).
- [67] E. Huisman et al., “Semiflexible filamentous composites”, *Phys. Rev. Lett.* **105**, 1 (2010).
- [68] Y. Cao et al., “Correlated insulator behaviour at half-filling in magic-angle graphene superlattices”, *Nature* **556**, 80 (2018).
- [69] Y. Cao et al., “Unconventional superconductivity in magic-angle graphene superlattices”, *Nature* **556**, 43 (2018).
- [70] M. Yankowitz et al., “Tuning superconductivity in twisted bilayer graphene”, *Science* **363**, 1059 (2019).

- [71] X. Lu et al., “Superconductors, orbital magnets and correlated states in magic-angle bilayer graphene”, *Nature* **574**, 653 (2019).
- [72] J. Pei et al., “Producing air-stable monolayers of phosphorene and their defect engineering”, *Nat Commun* **7**, 10450 (2016).
- [73] J. Zhang et al., “Scalable growth of high-quality polycrystalline MoS₂ monolayers on SiO₂ with tunable grain sizes”, *ACS Nano* **8**, 6024 (2014).
- [74] J. Xie et al., “Stitching h-BN by atomic layer deposition of LiF as a stable interface for lithium metal anode”, *Sci. Adv.* **3**, eaao3170 (2017).
- [75] A. Geim and I. Grigorieva, “Van der Waals heterostructures”, *Nature* **499**, 419 (2013).
- [76] C. D. Santangelo, “Extreme mechanics: Self-folding origami”, *Annu. Rev. Condens. Matter Phys.* **8**, 165 (2017).
- [77] F. D’Ambrosio, H. L. Bodlaender, and G. T. Barkema, “Dynamic sampling from a discrete probability distribution with a known distribution of rates”, *Comput Stat*, 10.1007/s00180-021-01159-3 (2021).
- [78] M. Breeman et al., “Computer simulation of metal-on-metal epitaxy”, *Thin Solid Films* **272**, 195 (1996).
- [79] A. F. Voter, “Classically exact overlayer dynamics: Diffusion of rhodium clusters on rh(100)”, *Phys. Rev. B* **34**, 6819 (1986).
- [80] C. Wong and M. Easton, “An efficient method for weighted sampling without replacement”, *SIAM J. Comput.* **9**, 111 (1980).
- [81] P. M. Maurer, “Finite random variates using differential search trees”, *Simul. Series* **49**, 273 (2017).
- [82] A. Walker, “New fast method for generating discrete random numbers with arbitrary frequency distributions”, *Electron. Lett.* **10**, 127 (1974).
- [83] A. J. Walker, “An efficient method for generating discrete random variables with general distributions”, *ACM Trans. Math. Softw.* **3**, 253 (1977).
- [84] G. Marsaglia, “Generating discrete random variables in a computer”, *Commun. ACM* **6**, 37 (1963).
- [85] Y. Matias, J. S. Vitter, and W.-C. Ni, “Dynamic generation of discrete random variates”, *Theory Comput. Systems* **36**, 329 (2003).
- [86] B. van de Klundert, “Efficient Generation of Discrete Random Variates”, Master’s thesis (Utrecht University, 2019).
- [87] S. Rajasekaran and K. W. Ross, “Fast algorithms for generating discrete random variates with changing distributions”, *ACM Trans. Model. Comput. Simul.* **3**, 1 (1993).
- [88] T. Hagerup, K. Mehlhorn, and J. I. Munro, “Maintaining discrete probability distributions optimally”, in *Automata, Languages and Programming* (Springer Berlin Heidelberg, 1993), pp. 253–264.
- [89] D. T. Gillespie, “Exact stochastic simulation of coupled chemical reactions”, *J. Phys. Chem.* **81**, 2340 (1977).
- [90] A. Slepoy, A. P. Thompson, and S. J. Plimpton, “A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks”, *J. Chem. Phys.* **128**, 205101 (2008).
- [91] T. H. Cormen et al., *Introduction to Algorithms*, 2nd (MIT Press, 2001).
- [92] S. Edelkamp and S. Schrödl, “*Dictionary data structures”, in *Heuristic Search. Theory and Applications* (Elsevier, San Francisco, 2012), pp. 89–159.
- [93] P. E. Black, “Binary heap”, in *Dictionary of Algorithms and Data Structures* (NIST, 2019).

- [94] M. Loeve, “Centering at medians and symmetrization”, in *Probability Theory I* (Springer, 1977), p. 256.
- [95] B. Welford, “Note on a method for calculating corrected sums of squares and products”, *Technometrics* **4**, 419 (1962).
- [96] G. Barkema and N. Mousseau, “Identification of relaxation and diffusion mechanisms in amorphous silicon”, *Phys. Rev. Lett.* **81**, 1865 (1998).
- [97] A. F. Voter, F. Montalenti, and T. C. Germann, “Extending the time scale in atomistic simulation of materials”, *Annu. Rev. Mater. Res.* **32**, 321 (2002).
- [98] K. J. Laidler, “Theories of Reaction Rates”, in *Chemical Kinetics*, 2nd (Pearson, London, 1987), pp. 80–135.
- [99] H. Eyring, “The activated complex in chemical reactions”, *J. Chem. Phys.* **3**, 107 (1935).
- [100] D. G. Truhlar, B. C. Garrett, and S. J. Klippenstein, “Current status of transition-state theory”, *J. Phys. Chem.* **100**, 12771 (1996).
- [101] S. Arrhenius, “Über die dissociationswärme und den einfluss der temperatur auf den dissociationsgrad der elektrolyte”, *Zeitschrift für Physikalische Chemie* **4**, 96 (1889).
- [102] M. A. Pimenta et al., “Studying disorder in graphite-based systems by raman spectroscopy”, *Phys. Chem. Chem. Phys.* **9**, 1276 (2007).
- [103] G. H. Vineyard, “Frequency factors and isotope effects in solid state rate processes”, *J. Phys. Chem. Solids* **3**, 121 (1957).
- [104] N. Metropolis et al., “Equation of state calculations by fast computing machines”, *J. Chem. Phys.* **21**, 1087 (1953).
- [105] S. Chib and E. Greenberg, “Understanding the metropolis-Hastings algorithm”, *Amer. Statist.* **49**, 327 (1995).
- [106] D. D. Minh and D. L. P. Minh, “Understanding the Hastings algorithm”, *Communications in Statistics - Simulation and Computation* **44**, 332 (2014).
- [107] K. A. Fichthorn and W. Weinberg, “Theoretical foundations of dynamical Monte Carlo simulations”, *J. Chem. Phys.* **95**, 1090 (1991).
- [108] L. K. Béland et al., “Kinetic activation-relaxation technique”, *Phys. Rev. E* **84**, 046704 (2011).
- [109] M. Grohe and P. Schweitzer, “The graph isomorphism problem”, *Commun. ACM* **63**, 128 (2020).
- [110] P. Kelly, “A congruence theorem for trees”, *Pacific J. Math.* **7**, 961 (1957).
- [111] J. Hopcroft and J. Wong, “Linear time algorithm for isomorphism of planar graphs (Preliminary report)”, in *STOC '74: Proceedings of the sixth annual ACM Symposium on Theory of Computing* (1974), pp. 172–184.
- [112] G. S. Lueker and K. S. Booth, “A linear time algorithm for deciding interval graph isomorphism”, *J. ACM* **26**, 183 (1979).
- [113] C. J. Colbourn, “On testing isomorphism of permutation graphs”, *Networks* **11**, 13 (1981).
- [114] L. Babai, “Graph isomorphism in quasipolynomial time”, in *STOC '16: Proceedings of the forty-eighth annual ACM symposium on Theory of Computing* (2016), pp. 684–697.
- [115] B. D. McKay and A. Piperno, “Practical graph isomorphism, II”, *J. Symb. Comput.* **60**, 94 (2014).
- [116] N. Schneider, R. A. Sayle, and G. A. Landrum, “Get your atoms in order—An open-source implementation of a novel and robust molecular canonicalization algorithm”, *J. Chem. Inf. Model.* **55**, 2111 (2015).

Summary

The isolation of graphene, a discovery worthy of a Nobel Prize, has led to an explosion of research and industrial interest for this remarkable two-dimensional material, due to its unique thermal, mechanical, optical and electronic properties. Graphene can now be found in a plethora of different technologies, with new and innovative products expected to reach the market during this decade. There are still important issues with the production of graphene and structural defects are still common, especially intrinsic defects that can easily form due to rapid quenching from a high temperature or irradiation. Defects affect the material at a long-range scale, and can significantly alter its properties; in particular, it leads to buckling of the material in the third dimension. Computer simulations are an appropriate tool for the study of the complex behaviour of defected graphene, in particular polycrystalline graphene, but performance is critical and can be a limiting factor in the study of larger samples, especially in their production starting from a random seed.

In Chapter 2, we introduce and discuss two techniques and a tool for generating computer samples of polycrystalline graphene efficiently, a necessary step for the study of large-scale properties of graphene. Starting from a completely random sample to prevent biases, we evolve it towards a more physical, less disordered configuration. In the classical Wooten, Winer, and Weaire approach, the coordinates of the sample are relaxed with molecular dynamics after each attempted bond transposition and then the Metropolis acceptance probability is computed from the relaxed energy. In our methods, we initially relax only the coordinates of the atoms in the local neighborhood around those involved in the bond transposition; the relaxed energy is then tentatively computed from a harmonic approximation, based on empirical parameters. In the first method, called *early rejection*, moves can be rejected based on the approximated relaxed energy; otherwise, the sample is relaxed completely and the acceptance is based on the energy of the fully relaxed sample, preserving detailed balance. In the second method, called *early decision*, the acceptance or rejection is based exclusively on the estimated relaxed energy, with global relaxation triggered after a certain number of accepted structural changes. We test both methods on samples of 3,200, 10,024 and 20,000 atoms, showing a

speed-up of multiple orders of magnitudes, especially for larger samples. We finally introduce a *graphical manipulation tool* to easily remove unwanted artefacts, such as bond crossings, that are usually present even in well-relaxed samples that are generated from a random seed.

In Chapter 3, we study a sample of polycrystalline graphene under a stretching force, with a computer simulation. While graphene is a two-dimensional material, polycrystalline graphene has an inherent tendency to buckle, developing an out-of-plane, three-dimensional structure, which can be altered by applying a stretching force to the sample. In this chapter, we study the evolution of the buckling structure of polycrystalline graphene under an adiabatic change in the stretching force with a computer simulation, by continuously changing its amplitude and relaxing the sample after each change. We establish that the evolution of the three-dimensional structure of graphene under a stretching force happens in a non-linear fashion: tiny increases of the magnitude of the force can lead to significant displacements, similar to avalanches, which in turn can create vibrations in the material. We further show that this displacement exhibits a hysteretic behaviour, reaching a different energy minimum upon a cycle from low to high stress and back. These behaviours open up another direction in studying the dynamical elasticity of polycrystalline quasi-two-dimensional systems, and in particular the implications on their mechanical and thermal properties.

In Chapter 4, we study the problem of sampling from a dynamically changing discrete probability distribution, i.e. from a list of events with a given rate that can be updated at runtime. We assume that, due to knowledge of the underlying processes, some prior information is known on the distribution of the rates, in particular the maximum and minimum rate, and that the number of possible outcomes is arbitrarily large. We consider three basic data structures (the Acceptance-Rejection method, the Complete Binary Tree and the Alias method) and we show that they can be used as building blocks in multi-level data structures: events can be grouped according to their rate and different data structures are used to sample the group and then the event from the sampled group. We prove that for particular data structures the expected time of sampling and update is (amortized) constant when the rate distribution follows certain conditions or the number of possible events is sufficiently large. We describe a data structure that guarantees sub-logarithmic expected update and sample time in all cases. Experimental verification of all the presented results is also given, highlighting the limits given by the constraints of a real-life setting.

In Chapter 5, after discussing the theoretical background of the Metropolis process in the WWW approach, we set ourselves to design an Event-Based approach to simulating the evolution of polycrystalline graphene in well-relaxed samples.

The Acceptance-Rejection process is replaced by sampling a transition with one of the techniques described in Chapter 4, after computing the rates for all possible transitions. Following similar assumptions on locality to those of Chapter 2, two techniques are introduced in order to reduce the number of computations required after each applied transition. In the Local Event-Based technique, the rate is computed by relaxing only the neighborhood of the atoms involved in the bond transposition of interest; the rate of transitions not involving atoms in the neighborhood of an applied bond transposition is not recomputed. In the Topological Event-Based technique we further assume that the graph described by the atoms in the neighborhood of a bond transposition determines its rate. Before (re)computing a bond transposition, we check against already observed local graphs; if there is a match, the same rate is assigned, without any further computations. We finally discuss the consequences of this approach to future work.

Samenvatting

Het isoleren van grafeen, beloond met de Nobelprijs, heeft geleid tot een explosie aan onderzoek en industriële belangstelling voor dit opmerkelijke tweedimensionale materiaal, vanwege zijn unieke thermische, mechanische, optische en elektronische eigenschappen. Grafeen wordt nu gebruikt in een breed spectrum van technologieën, met nieuwe en innovatieve producten die naar verwachting in dit decennium de markt bereiken. Er zijn nog steeds grote uitdagingen bij de productie van grafeen, dat vaak rijk is aan defecten, vooral intrinsieke defecten die gemakkelijk gevormd worden bij snelle afkoeling of bestraling. Defecten beïnvloeden het materiaal over een grote afstand en kunnen de eigenschappen sterk veranderen, met name leiden ze tot kromtrekking in de derde dimensie. Computersimulaties zijn een geschikt tool voor de bestudering van het complexe gedrag van defectrijk grafeen, en in het bijzonder polykristallijn grafeen, maar de rekenefficiëntie kan van doorslaggevend belang zijn in het bestuderen van grotere samples, vooral als deze geproduceerd worden vanuit een willekeurige kiemstructuur.

In hoofdstuk 2 introduceren en bespreken we twee technieken en een tool voor het efficiënt genereren van computerstructuren van polykristallijn grafeen, een noodzakelijke stap voor de bestudering van de eigenschappen van grafeen op grotere schaal. We beginnen met een volledig willekeurige structuur die vrij is van voorkeursrichtingen, en evolueren deze naar een natuurlijke en meer geordende structuur. Hierbij volgen we de klassieke aanpak van Wooten, Winer en Weaire, waarin de atoomcoördinaten volgen uit moleculaire dynamica, volgend op bondtransposities die geaccepteerd of verworpen worden op grond van de geminimaliseerde energie, conform het Metropolis algoritme. In onze aanpak worden in eerste instantie tijdens de minimalisatie alleen de coördinaten van de atomen in de directe omgeving van de bondtranspositie aangepast; de energie na volledige minimalisatie wordt vervolgens geschat met een harmonische benadering, gebruikmakend van empirische parameters. In de eerste aanpak, die we early rejection noemen, kunnen voorgestelde bondtransposities verworpen worden op grond van deze geschatte energie; zo niet, dan volgt het definitief accepteren van de nieuwe structuur op grond van de volledig geminimaliseerde energie; dit behoudt detailed balance.

In de tweede aanpak, die we *early decision* noemen, volgt het al dan niet accepteren van de voorgestelde bondtranspositie puur op grond de geschatte energie, waarbij wel globale minimalisatie toegepast wordt na een zeker aantal geaccepteerde structuurveranderingen. Beide manieren van aanpak worden getest in simulaties van structuren met 3200, 10.024 en 20.000 atomen, en ze laten een efficiëntieverhoging zien met meerdere orden van grootte, vooral bij de grotere structuren. Ook introduceren we een grafische manipulatiETOOL waarmee ongewenste structurele artefacten eenvoudig opgelost kunnen worden, zoals elkaar kruisende atoomverbindingen die zelfs na lang gerelaxeerde structuren kunnen opduiken wanneer ze zijn ontwikkeld vanuit een volledig willekeurige structuur.

In hoofdstuk 3 bestuderen we polykristallijn grafeen onder uitrekking, in een computersimulatie. Waar grafeen een tweedimensionaal materiaal is, heeft polykristallijn grafeen een inherente neiging tot kromtrekken, met driedimensionale structuurvorming buiten het vlak, die verandert onder invloed van een strekkracht. In dit hoofdstuk bestuderen we het gedrag van de driedimensionale kromming onder invloed van een adiabatische verandering in de strekkracht, waarbij de structuur continu zich volledig aan kan passen na elke incrementele uitrekking. Wij laten zien dat de veranderingen in de driedimensionale structuur bij uitrekking niet-lineair zijn: een minuscule verandering in de strekkracht kan resulteren in significante verplaatsingen in de derde dimensie, analoog aan een lawine (*avalanche*), en deze verplaatsingen kunnen resulteren in trillingen in het materiaal. We laten ook zien dat deze verplaatsingen in de derde dimensie hysteresis vertonen, met een ander energie minimum na een cyclus van lage tot hoge strekkracht en terug. Deze effecten openen nieuwe mogelijkheden om de dynamische elasticiteit van polykristallijne tweedimensionale systemen te bestuderen, en dan met name hun mechanische en thermische eigenschappen.

In hoofdstuk 4 bestuderen we het bemonsteren van gebeurtenissen met een in de tijd veranderende discrete kansverdeling, i.e. het selecteren van één gebeurtenis uit een lijst van gebeurtenissen, waarbij de frequenties regelmatig aangepast moeten kunnen worden. We veronderstellen dat op grond van kennis van de onderliggende processen er enige voorkennis is over de verdeling van de frequenties, specifiek een minimale en maximale waarde voor de frequentie, en bovendien dat het aantal mogelijke uitkomsten willekeurig groot is. Drie basisstructuren worden beschouwd (de Acceptance-Rejection methode, de Complete Binary tree en de Alias methode), en we laten zien dat deze drie gebruikt kunnen worden als bouwstenen in multi-level datastructuren: gebeurtenissen kunnen gegroepeerd worden op basis van hun frequentie en verschillende datastructuren worden gebruikt om allereerst de groep te selecteren, en vervolgens een gebeurtenis binnen de groep. We bewijzen dat voor bepaalde datastructuren de verwachte selectietijd en aanpassing van de

frequenties (amortized) constant is als de frequentieverdeling voldoet aan bepaalde voorwaarden of het aantal mogelijke gebeurtenissen groot genoeg is. We presenteren een datastructuur die gegarandeerd sub-logaritmische tijd nodig heeft voor zowel een frequentieaanpassing als selectie. Deze resultaten worden experimenteel geverifieerd, waarbij ook de beperkingen in een realistische situatie getoond worden.

In hoofdstuk 5 wordt eerst de theoretische basis van het WWW algoritme met Metropolis acceptatie besproken, waarna toegewerkt wordt naar een Event-Based benadering van de simulatie van polykristallijn grafeen in energetisch geminimaliseerde structuren. De standaard aanpak waarin willekeurige bondtransposities voorgesteld en vervolgens meestal verworpen worden, wordt vervangen door een van de technieken die zijn beschreven in hoofdstuk 4, waarbij de frequenties voor alle mogelijke transitie worden berekend. Gebruik makend van de aanname van lokaliteit die ook in Hoofdstuk 2 gedaan is, worden twee technieken geïntroduceerd voor de reductie van het aantal rekenoperaties benodigd per bondtranspositie. In de Local Event-Based methode kunnen bij de berekening van de frequentie alleen de atomen in de buurt van de bondtranspositie bewegen, terwijl de coördinaten van de andere atomen vastgehouden worden. In de Topological Event-Based methode nemen we aan dat de topologie van de graaf gevormd door de atoombindingen rondom de bondtranspositie bepalend is voor de frequentie. Voorafgaand aan de (her)berekening van een bondtranspositie wordt opgezocht of eerder dezelfde graaf tegengekomen is; en als dat zo is, wordt de frequentie daarvan overgenomen. Het hoofdstuk wordt afgesloten met een bespreking van de gevolgen van deze beide methoden voor toekomstig werk.

Federico D'Ambrosio

Curriculum Vitae

Education

- 2016–2022 **Ph.D. in Computer Science**, *Utrecht University*.
Thesis: *Computer models for the study of graphene*.
Supervisors: Gerard T. Barkema and Hans L. Bodlaender.
- 2013–2016 **Master's degree in Physics**, *University of Padova, Italy*.
Thesis: *Temperature response of heat conducting models*.
Supervisor: Marco Baiesi.
- 2009–2013 **Bachelor's degree in Physics**, *University of Padova, Italy*.
Thesis: *Disappearance neutrino signal with the Nessie Spectrometers*.
Supervisor: Stefano Dusini.

Experience

- Oct. 2021–Present **Data Scientist (M&AR)**, *ABN Amro Bank N. V., Amsterdam*.
Interest rate risk modeling.
- Nov. 2016–May 2021 **PhD Candidate**, *Utrecht University, Utrecht*, Supervisors: Professors Gerard T. Barkema and Hans L. Bodlaender.
- Research on polycrystalline graphene with computational Monte Carlo models. Theoretical analysis of the underlying physical problems, development of custom simulations in C++, data analysis and data visualization in Python, scientific collaborative writing and scientific presentations.
 - Research on development of computational techniques to improve computational Monte Carlo models. Theoretical and experimental analysis of computational problems met during previous research, computational complexity analysis of possible solutions, validation of theoretical results with computer simulations on both real and simulated data, interdisciplinary scientific writing and scientific presentations.
 - Assistant teacher for the courses of the Department of Information and Computing Sciences (Image Processing, Graphics, Introductory Project and Game Physics).

- Sept. 2015 **Biophysics Internship**, *Fondazione Ricerca Biomedica Avanzata onlus - VIMM*, Padova, Italy.
Collection of experimental data of optical fluorescence microscopy, under supervision, to study spacial and temporal dynamics of calcium ions inside the cell, data analysis with MATLAB and custom simulations of the obtained data, implementation of a model to correctly reconstruct the concentration of cell calcium from fluorescence data, validation of the model with custom simulations, report writing and presentation of the results to the research group.

Journal publications

- **F. D'Ambrosio**, J. Barkema, G.T. Barkema, *Efficient structural relaxation of polycrystalline graphene models*. In Nanomaterials.
- **F. D'Ambrosio**, H.L. Bodlaender, G.T. Barkema, *Dynamic Sampling from a Discrete Probability Distribution with a Known Distribution of Rates*. In Computational Statistics.
- **F. D'Ambrosio**, V. Juričić, G.T. Barkema, *Discontinuous evolution of the structure of stretching polycrystalline graphene*. In Physical Review B.
- **F. D'Ambrosio**, M. Baiesi, *Thermal response of a Fermi–Pasta–Ulam chain with Andersen thermostats*. In The European Physical Journal B.

Academic presentations

- Mar. 2020 Oral Presentation at APS March Meeting 2020 (cancelled due to COVID-19)
- Oct. 2019 Oral Presentation at the Assembly of Advanced Materials Congress
- Jun. 2018 Poster Presentation at Highlights of Algorithms 2018
- Oct. 2016 Poster Presentation at Venice meeting on fluctuations in small complex systems III

Miscellaneous

- 2018–2019 **Faculty Council member**, *Faculty of Science, Utrecht University*, Utrecht, Netherlands.
Representative of the employees of the Department of Information and Computing Sciences in the students and employees representation council.
- 2017–2018 **Treasurer**, *Prout – PhD Network Utrecht*, Utrecht, Netherlands.
Treasurer of the PhD association of the University of Utrecht

- 2014 **Board member**, *ESU Padova*, Padova, Italy.
Students' representative in the board of the regional company that manages financial aid, accommodation and food service to the students of the University of Padova
- 2012–2016 **Department Council member**, *Department of Physics and Astronomy, University of Padova*, Padova, Italy.
Representative of the students of the Physics BSc and MSc courses in the Department Council.
- 2012–2014 **Board member**, *Studenti Per - Udu Padova*, Padova, Italy.
Board member of the Students' Union of the University of Padova

È andata così.

Thursday, April 21st 2022

Utrecht, Netherlands

