



Interactive physically-based structural modeling of hydrocarbon systems

Maël Bosson, Sergei Grudinin, Xavier Bouju, Stephane Redon

► To cite this version:

Maël Bosson, Sergei Grudinin, Xavier Bouju, Stephane Redon. Interactive physically-based structural modeling of hydrocarbon systems. *Journal of Computational Physics*, 2012, 231 (6), pp.2581-2598. 10.1016/j.jcp.2011.12.006 . hal-00755542

HAL Id: hal-00755542

<https://inria.hal.science/hal-00755542>

Submitted on 22 Nov 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Interactive physically-based structural modeling of hydrocarbon systems

Mael Bosson^a, Sergei Grudinin^a, Xavier Bouju^b, Stephane Redon^a

^a *NANO-D - INRIA Grenoble - Rhône-Alpes / CNRS Laboratoire Jean Kuntzmann, 655, avenue de l'Europe Montbonnot, 38334 Saint Ismier Cedex, France*

^b *CEMES - CNRS, 29, rue Jeanne-Marvig, BP 94347, 31055 Toulouse Cedex 4, France*

Abstract

Hydrocarbon systems have been intensively studied *via* numerical methods, including electronic structure computations, molecular dynamics and Monte Carlo simulations. Typically, these methods require an initial structural model (atomic positions and types, topology, etc.) that may be produced using scripts and/or modeling tools. For many systems, however, these building methods may be ineffective, as the user may have to specify the positions of numerous atoms while maintaining structural plausibility.

In this paper, we present an interactive *physically-based* modeling tool to construct structural models of hydrocarbon systems. As the user edits the geometry of the system, atomic positions are also influenced by the Brenner potential, a well-known bond-order reactive potential. In order to be able to interactively edit systems containing numerous atoms, we introduce a new *adaptive* simulation algorithm, as well as a novel algorithm to *incrementally* update the forces and the total potential energy based on the list of updated

Email addresses: `mael.bosson@inria.fr` (Mael Bosson),
`sergei.grudinin@inria.fr` (Sergei Grudinin), `xavier.bouju@cemes.fr` (Xavier Bouju), `stephane.redon@inria.fr` (Stephane Redon)

relative atomic positions. The computational cost of the adaptive simulation algorithm depends on user-defined error thresholds, and our potential update algorithm depends linearly with the number of updated bonds. This allows us to enable efficient physically-based editing, since the computational cost is decoupled from the number of atoms in the system. We show that our approach may be used to effectively build realistic models of hydrocarbon structures that would be difficult or impossible to produce using other tools.

Keywords: Interactive modeling, Adaptive algorithms, Hydrocarbon builder, Brenner potential

1. Introduction

Scientific and technological advances are increasingly enabling the design of objects at the molecular scale.[1, 2, 3, 4, 5] Similar to the digital prototyping tools that have been developed for macroscopic engineering, powerful tools are more and more in demand to study, design and prototype nano-objects. The complexity of the underlying physics makes it difficult to manually create and edit virtual nano-objects, though. Hence, if atomic positions are not guided by the digital prototyping tool to plausible configurations, most users will often create unrealistic, high-energy structures that need to be minimized prior to start simulation steps. The final structure might then be very different from the initial plan of the user. An effective way to help the user overcome this problem is to continuously minimize the system’s energy during editing.

Such *physically-based* interactive modeling is a challenging problem, though, since it requires a very low computational cost at each modeling step. Semi-

empirical electronic structure calculations and first principle calculations may not be appropriate for such a modeling because of their high computational cost. Moreover, most classical force fields essentially model valence bonds using a harmonic potential, thus preventing topology changes during simulation, which reduces their applicability in interactive builders. An effective approach consists in simulating bond formation and destruction with so-called *reactive* force fields, such as the ReaxFF[6] method or the Brenner potential. The Brenner potential is an empirical many-body potential-energy expression with a chemical bond description.[7, 8] It has been used to model *e.g.* carbon nanotube growth[9] or reconstructed diamond growth[10] through chemical vapor deposition, nanotube fracture,[11] buckminsterfullerene dynamics,[12] etc. The first proposed version of the Brenner potential had some limitations that were later removed.[13] Several other extensions or corrections have been proposed, *e.g.* to handle other atom types[14] or weak long-range interactions.[15, 16] Brenner *et al.* have described the use of this potential to simulate nano-engineering, which involves the creation and destruction of chemical bonds.[17] We show in this paper that this potential is also particularly adapted to *virtual nano-engineering*, *i.e.* interactive digital modeling of carbon and hydrocarbon structures.

We have recently proposed an approach to speed up a modeling time step in the framework of a classical force field, and demonstrated it on the CHARMM19 and CHARMM22 force fields.[18] The method relies on partial updates of neighbor lists and interaction potentials and forces, which are made possible by incremental updates of the molecular state description (precisely, only a subset of the torsional degrees of freedom are moved

at each time step). The other key feature is an efficient algorithm able to predict which degrees of freedom are the most important. We propose a similar approach in cartesian coordinates to dynamically focus the available power on the most active region of the system in editing. Precisely, the algorithm chooses a subset of mobile atoms and, taking advantage of the fact that Brenner’s reactive force-field function only depends on atomic relative positions, performs an incremental update of the Brenner potential. This enables interactivity and efficient guide for the user during the editing of a system.

The paper is organized as follows. An overview is proposed in Sect. 2. It provides a brief introduction to our interactive modeling methodology and to the Brenner potential. Sect. 3 analyzes the structure of the computations involved in updating the Brenner potential, which makes an incremental approach possible, and deduces an algorithm to incrementally update the total potential energy and interatomic forces. Sect. 4 describes the new adaptive approach. In Sect. 5 we present the applications and results. The computational cost of the incremental update is analyzed. A new minimizer is proposed as an application of the new adaptive approach. This leads to the interactive modeler that may be used for digital prototyping of hydrocarbon systems. Finally, Sect. 6 concludes the paper and discusses possible extensions and other applications of our work.

2. Overview

2.1. Interactive physically-based structural modeling

The main idea behind interactive physically-based structural modeling is to continuously minimize the system energy during editing to help the user build realistic structures. In practice, our builder alternates a time step of minimization with a time step of user action. In order to maintain interactive update rates even when the model contains numerous atoms, though, we propose to use an adaptive, partial update of the system’s state at each time step.

Precisely, the structure of interactive modeling algorithm time step consists in alternating two main steps:

1. User action step

- (a) let the user move / create / delete one or more atoms
- (b) incrementally update forces and the potential energy

2. Adaptive minimization step

- (a) choose a new set of mobile atoms
- (b) perform a minimization step with the corresponding set
- (c) incrementally update forces and the potential energy

When the user performs a local action on the system of atoms, the potential energy and interatomic forces are incrementally updated: thanks to the structure of the Brenner potential, only *some* forces need to be updated, and the total potential energy may be updated through recomputation of *some* terms only. This helps us to control the computational cost of a time step.

We describe in Sect. 3 a new algorithm to incrementally update energies and forces.

Once all forces and the potential energy are known, the adaptive minimization step chooses a set of “most mobile atoms” and updates the positions of these atoms only (since moving all atoms might prevent interactive rates when constructing large systems). The choice of the set of *active* degrees of freedom (i.e. the set of mobile atoms) is performed automatically at each time step, based on user-defined precision constraints (*e.g.* the total number of mobile atoms at each time step, or the maximum allowable error on the acceleration of the molecular system), as well as the current forces being applied on each atom. Controlling the number of mobile atoms, and thus the cost of the system’s update, allows us to decouple the cost of the time step from the number of atoms in the system, and helps us maintain interactive editing rates. We describe in Sect. 4 a new adaptive algorithm to determine the set of most mobile atoms.

Note that detecting neighboring atoms is independent of the underlying force-field as well, and we may use for example the adaptive neighbor list algorithm described in Rossi *et al.*[18] As a result, we assume in this paper that, when we need to update forces and energies at each time step, we are given the list of pairs of neighboring atoms whose relative position has changed due to user actions or to adaptive minimization.

2.2. Brenner potential

For completeness, we now introduce the second generation Brenner potential.[13] This potential belongs to the class of bond-order potentials, and thus has the ability to evolve smoothly between different bonding states. Shortly, it al-

allows the calculation of the interaction energy of chemically covalent bond parametrized by a bond-order function.

2.2.1. Potential expression

The Brenner potential has the particularity to be only focused on covalent bonds (no long-range interaction), *i.e.* the total potential energy is a sum over *interacting atoms* (two atoms are said to be interacting if the distance between them is smaller than 2 Å):

$$E_b = \sum_i \sum_{j>i} [V^R(r_{ij}) - b_{ij}V^A(r_{ij})], \quad (1)$$

where b_{ij} is the bond order. The functions V^R and V^A are respectively repulsive and attractive terms. A complete description of these terms and their parameters is provided in the original publication [13]. The complete analytic form of the function is recalled in appendix.

2.2.2. Forces

In order to proceed, we need to determine the force terms, *i.e.* the gradient of the potential, from the description above. Because the force applied on one atom only depends on relative positions, we can express it as a sum over the interacting pairs in which this atom is involved:

$$\frac{\partial V}{\partial x_i} = \sum_{j, (i,j) \in \mathcal{B}} \left(\frac{\partial V}{\partial r_{ji}} \right) \left(\frac{\partial r_{ji}}{\partial x_i} \right), \quad (2)$$

where r_{ji} is the distance between atoms i and j , and \mathcal{B} is a set of all the pairs of atoms involved in interaction:

$$\mathcal{B} = \{(i, j), r_{ij} < D_{ij}^{max}\}, \quad (3)$$

where D_{ij}^{max} depends on atom types.

For future use, we also define:

$$\mathcal{A} = \{(i, j, k), (i, j) \text{ and } (j, k) \in \mathcal{B}, i \neq j\} , \quad (4)$$

$$\mathcal{D} = \{(i, j, k, l), (i, j), (j, k) \text{ and } (k, l) \in \mathcal{B}, i \neq k \neq l\} , \quad (5)$$

and we decompose the potential as follows:

$$V = \sum_i \sum_{j>i} [V^R(r_{ij}) - b_{ij} V^A(r_{ij})] = \sum_{(i,j) \in \mathcal{B}, j>i} V_{ij}^B . \quad (6)$$

Then, a direct analysis of the influence of the relative position of every pair in interaction on the potential is sufficient to know the forces on each atom. Because of the bond order term, the relative position of atoms i and j can influence several V_{kl}^B terms.

Indeed, the total force on one atom can be split into three contributions.

$$\frac{\partial V}{\partial x_i} = \sum_{j, (i,j) \in \mathcal{B}} \left(\sum_{(k,l) \in \mathcal{B}} \frac{\partial V_{kl}^B}{\partial r_{ji}} \right) \left(\frac{\partial r_{ji}}{\partial x_i} \right) , \quad (7)$$

$$\sum_{(k,l) \in \mathcal{B}} \frac{\partial V_{kl}^B}{\partial r_{ji}} = \frac{\partial V_{ij}^B}{\partial r_{ji}} + \frac{\partial V_{ij}^A}{\partial r_{ji}} + \frac{\partial V_{ij}^D}{\partial r_{ji}} , \quad (8)$$

where V_{ji}^A is the sum of bond terms that involve either i or j :

$$V_{ij}^A = \sum_{k, (j,i,k) \in \mathcal{A}} V_{ik}^B + \sum_{k, (i,j,k) \in \mathcal{A}} V_{jk}^B , \quad (9)$$

and V_{ji}^D is the sum of bond terms that involve neighbors of i or j :

$$V_{ij}^D = \sum_{(k,l), (i,j,k,l) \in \mathcal{D}} V_{kl}^B + \sum_{(k,l), (j,i,k,l) \in \mathcal{D}} V_{kl}^B . \quad (10)$$

3. An incremental algorithm to update forces and potential

We may now deduce an algorithm able to incrementally update both the potential and atomic forces.

3.1. Influence of relative motions

In the previous section, we have shown that relative positions of interacting atoms may influence several terms in the potential. Figure 1 presents these dependencies in a schematic case. Adaptive modeling also requires us to know the dependency of *forces* on the relative position of two atoms, but this may be directly deduced from the previous analysis. Indeed, equation 7 shows that the force on atom i needs to be updated when either one of the three terms in the sum in equation 8 has been updated. These terms, in turn, need to be updated when the bond terms involved in equation 9 and equation 10 are updated. This dependency is illustrated in Figure 2. Figure 3 shows the complete dependency between the terms involved in both the potential and the atomic forces.

3.2. The algorithm

As noted in Sect. 2, the incremental collision detection algorithm provides us with a set of pairs of atoms whose relative position has changed since the last time step. Let $\mathcal{B}_{updated}$ denote this set. Note that this set may include pairs of newly interacting atoms, as well as pairs of atoms which are now separated by a distance longer than the force-field cutoff (2 Å in the case of the Brenner potential), but which were interacting at the previous time step.

At each time step, the algorithm starts with this information to update the sets \mathcal{B} , \mathcal{A} and \mathcal{D} . This update is incremental in the sense that the sets are

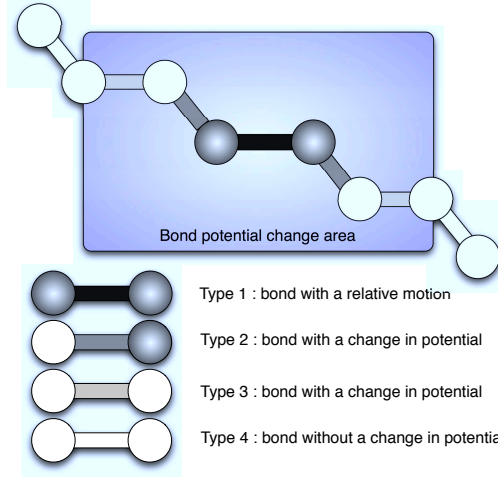


Figure 1: Dependency of the Brenner potential terms on the relative motion of atoms (see Sect. 3.1). Atoms belonging to “Type 1” bonds have a relative motion, which induces a modification in every term of the bond potential contribution. “Type 2” bonds are rigid, but with a bond order change due to its dependency on the number of neighbors, as well as on angles and on dihedrals. “Type 3” bonds are rigid, but with a bond order change caused by the change on the conjugate number N^{conj} . The relative motion of atoms in “Type 1” bonds does not impact the potential contribution of “Type 4” bonds.

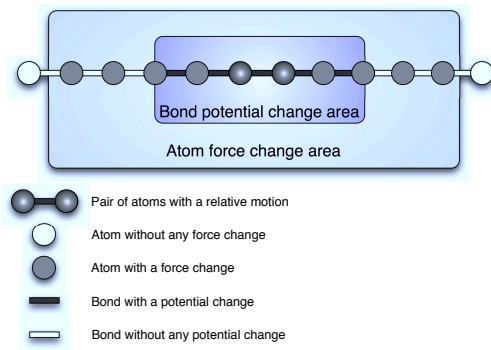


Figure 2: Dependency of atomic forces on relative positions updates (see Sect. 3.1).

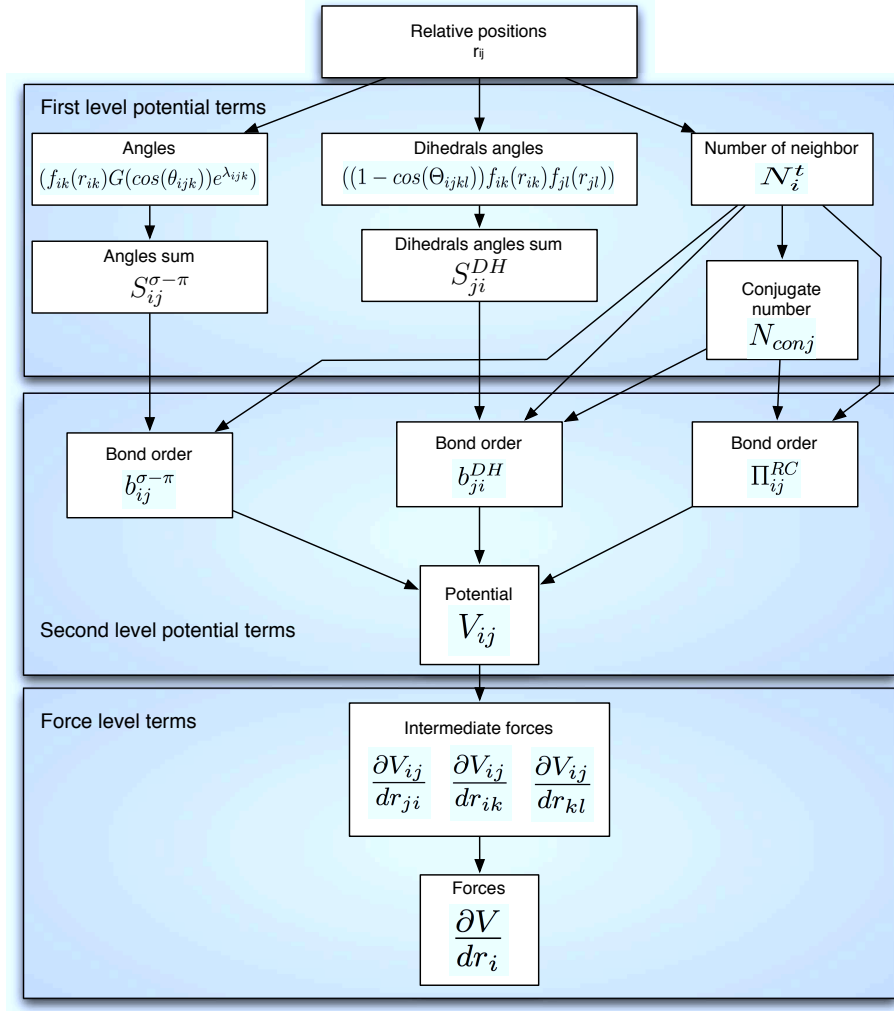


Figure 3: Scheme illustrating the Brenner potential terms dependency. The appropriate data structure for incremental update is deduced from this schematic representation of the terms dependency.

not rebuild from scratch, but elements are inserted or deleted. In practice, the algorithm stores all terms which contribute to the total potential and to the atomic forces. To achieve this, we introduce four sets of tuples:

$$I^{Atom} = \{I_i^{Atom} = (N_i^H, N_i^C, N_i^t, N_i^{aux}), i \in \mathcal{Atoms}\} ,$$

$$I^{Bond} = \{I_{ij}^{Bond} = (V_{ij}^R, V_{ij}^A, b_{ij}^{\sigma-\pi}, S_{ij}^{\sigma-\pi}, b_{ji}^{\sigma-\pi}, S_{ji}^{\sigma-\pi}, \Pi_{ij}^{RC}, b_{ji}^{DH}, S_{ji}^{DH}, N_{ij}^{conj}), \\ i < j, (i, j) \in \mathcal{B}\} ,$$

$$I^{Angle} = \{I_{ijk}^{Angle} = (f_{ik}(r_{ik})G(\cos(\theta_{ijk}))e^{\lambda_{ijk}}), i < k, (i, j, k) \in \mathcal{A}\} ,$$

$$I^{Dihedral} = \{I_{ijkl}^{Dihedral} = \{(1 - \cos(\Theta_{ijkl}))f_{ik}(r_{ik})f_{jl}(r_{jl})\}, i < l, (i, j, k, l) \in \mathcal{D}\} . \quad (11)$$

These sets contain the terms that need to be updated to compute the total potential for the adaptive simulation. They contain respectively the *information* attached to one, two, three or four atoms (please refer to the Appendix for a detailed description of the potential function).

Similarly, we define four more sets to allow for an incremental update of the forces, corresponding to the decomposition proposed in the previous section:

$$\begin{aligned} F^{atom} &= \left\{ F_i^{atom} = \frac{\partial V}{\partial r_i}, i \in \mathcal{Atoms} \right\} , \\ F^B &= \left\{ F_{ij}^B = \frac{\partial V_{ij}^B}{\partial r_{ji}}, i < j, (i, j) \in \mathcal{B} \right\} . \\ F^A &= \left\{ F_{ijk}^A = \frac{\partial V_{ij}^B}{\partial r_{ik}}, i < k, (i, j, k) \in \mathcal{A} \right\} , \\ F^D &= \left\{ F_{ijkl}^D = \frac{\partial V_{ij}^B}{\partial r_{kl}}, i < l, (i, j, k, l) \in \mathcal{D} \right\} . \end{aligned} \quad (12)$$

F^{atom} contains the forces acting on each atom, and is updated based on the intermediate force terms contained in the sets F^B , F^A , and F^D .

3.3. Pseudo-code

The structure of the algorithm is deduced from the hierarchical dependency presented in Figure 3. The algorithm works with auxiliary sets $\hat{\mathcal{A}}toms$, $\hat{\mathcal{B}}$, $\hat{\mathcal{A}}$ and $\hat{\mathcal{D}}$ containing respectively singletons, pairs, triplets and quadruplets of atomic indices which correspond to obsolete terms. The computation of the potential and forces is organized in four main steps:

1. Initialization.

Based on $\mathcal{B}_{updated}$, the output of the collision detection algorithm, incrementally update the sets \mathcal{B} , \mathcal{A} and \mathcal{D} as indicated in Sect. 3.2. Set $\hat{\mathcal{A}}toms$, $\hat{\mathcal{B}}$, $\hat{\mathcal{A}}$ and $\hat{\mathcal{D}}$ to \emptyset .

2. First level potential update.

Detect terms that have to be updated directly from the list of bonds with a relative motion, *i.e.* $\forall (i, j) \in \mathcal{B}_{updated}$:

- (a) insert (i, j) in $\hat{\mathcal{B}}$
- (b) insert i and j in $\hat{\mathcal{A}}toms$
- (c) $\forall k, (i, j, k) \in \mathcal{A}$, insert (i, j, k) in $\hat{\mathcal{A}}$
- (d) $\forall (k, l), (i, j, k, l) \in \mathcal{D}$, insert (i, j, k, l) in $\hat{\mathcal{D}}$
- (e) $\forall (k, l), (k, i, j, l) \in \mathcal{D}$, $k < l$, insert (k, i, j, l) in $\hat{\mathcal{D}}$

Update terms:

- (a) $\forall i \in \hat{\mathcal{A}}toms$, update the four corresponding values in I_i^{Atom}
- (b) $\forall (i, j) \in \hat{\mathcal{B}}$, update V_{ij}^R , V_{ij}^A and N_{ij}^{conj} in I_{ij}^{Bond}
- (c) $\forall (i, j, k) \in \hat{\mathcal{A}}$, update the angle-dependent value I_{ijk}^{Angle} and its influence in I_{ij}^{Bond} and I_{jk}^{Bond} (*i.e.* the terms $S_{ij}^{\sigma-\pi}$ and $S_{ji}^{\sigma-\pi}$)
- (d) $\forall (i, j, k, l) \in \hat{\mathcal{D}}$, update the dihedral-dependent value $I_{ijkl}^{Dihedral}$ and its influence in I_{jk}^{Bond} (*i.e.* the term S_{ji}^{DH})

3. Second level potential update.

Extend the list of potential terms that have to be updated to include bonds without a relative motion, but with a bond order change:

- (a) $\forall(i, j) \in \mathcal{B}_{updated}, \forall(k), (i, j, k) \in \mathcal{A}$, insert (j, k) in $\widehat{\mathcal{B}}$
- (b) $\forall(i, j) \in \mathcal{B}_{updated}, \forall(k, l), (i, j, k, l) \in \mathcal{D}$, insert (k, l) in $\widehat{\mathcal{B}}$

Update the total potential

- (a) $\forall(i, j) \in \widehat{\mathcal{B}}$, update all the bond order terms (*i.e.* $b_{ij}^{\sigma-\pi}$, $b_{ji}^{\sigma-\pi}$, Π_{ij}^{RC} and b_{ji}^{DH}). Update V_{ij}^B based on V_{ij}^R , V_{ij}^A and b_{ij} , and update its contribution to the total sum V .

4. Force update.

Extend invalid sets to invalid force terms:

- $\forall(i, j) \in \widehat{\mathcal{B}}$,
- (a) $\forall k, (i, j, k) \in \mathcal{A}$, insert (i, j, k) in $\widehat{\mathcal{A}}$
- (b) $\forall(k, l), (i, j, k, l) \in \mathcal{D}$, insert (i, j, k, l) in $\widehat{\mathcal{D}}$

Update forces:

- (a) $\forall(i, j) \in \widehat{\mathcal{B}}$, update the corresponding F^B terms and their influence in the sum defining F_i^{Atom} and F_j^{Atom}
- (b) $\forall(i, j, k) \in \widehat{\mathcal{A}}$, update the corresponding F^A terms and their influence in the sum defining F_i^{Atom} , F_j^{Atom} and F_k^{Atom}
- (c) $\forall(i, j, k, l) \in \widehat{\mathcal{D}}$, update the corresponding F^D terms and their influence in the sum defining F_i^{Atom} , F_j^{Atom} , F_k^{Atom} and F_l^{Atom}

We do not provide the details of the update of the F^B , F^A and F^D terms, as their expression is straightforward — it can be directly retrieved from the expression of the potential — but tedious.

Note on sums updates: one important point in our algorithm is how we update the sums involved in the Brenner potential and forces. Indeed, sums are never recomputed from scratch, but are incrementally updated. Precisely, when we update a term and have to update its contribution to a sum (*e.g.* one term V_{kl}^B has been recomputed, and the sum in Equation 10 has to be updated), we subtract the old value of the term from the sum, and add the new value to it. This allows us to update obsolete terms only. Numerical error may be increased by this approach, however, due to the accumulation of subtractions and additions. To avoid this issue, a periodic re-initialization may be performed every *e.g.* million steps, where energies and forces are computed for the whole molecular system. Periodic re-initialization is very rarely needed, though, since the sums involve only a limited number of terms, and the loss of precision due to one incremental sum update (one addition and one subtraction) is around machine precision. Thus, the computational cost is amortized over the whole interactive session.

3.4. *Extension of the Brenner potential*

For some applications, a notable limitation of the Brenner potential is the absence of long-range interactions. To overcome this problem, some extensions have been proposed.[19, 20, 21, 22, 16] The general idea is to add a non-bonded interaction potential term that is a sum of pairwise potential contributions:

$$E_{NB} = \frac{1}{2} \sum_{i,j} V_{NB}(r_{ij}). \quad (13)$$

The difficulty is to correctly treat the intermolecular interactions without biasing the intramolecular part. This correction has been done carefully in

e.g. the popular AIREBO potential [16]. In this potential, the long-range term is progressively activated and combined with the accurate short-range part of the Brenner potential.

Long-range interactions are particularly important when one deals with carbon structures above a surface. The adsorption of fullerenes and nanotubes has been extensively studied with a large range of surfaces, including metallic, semiconducting and insulating ones. Here, we focus on the graphite surface and graphene as well. A previous study was based on a self-consistent real-space scheme for calculating the van der Waals interaction energy between a fullerene molecule and a graphitic substrate with atomic surface corrugation.[23] By using linear response theory to evaluate the dipole-dipole interactions between the molecule and the substrate, the van der Waals energy was accurately computed and the optimized conformations of the adsorbed molecule were found.

For simplicity, we use the approach of Los and Fasolino [19], and we modulate the exp-6 van der Waals potential parameters:

$$V_{NB(r_{ij})} = b \exp(-c_0 r) - \epsilon \left(\frac{\sigma}{r} \right)^6 - V_{shift}, \quad (14)$$

to retrieve optimum adsorption energies close to the already calculated ones and the corresponding distances between the center of the molecule and the surface.[23] Precisely, with parameters $b = 3224.9$ eV, $c_0 = 3.5995$ Å⁻¹, $\epsilon = 0.01396$ eV, $\sigma = 3.44$ Å, the energy errors are 0.0162 eV and 0.0108 eV and the distance errors are 0.08 Å and 0.09 Å for the top and hollow configurations respectively. In this case, the two-body potential expression can be introduced efficiently in our adaptive approach thanks to the notion of partial forces.[18]

As illustrated in Figure 4, the preferred orientation of the fullerene is always with a six-membered ring parallel to the surface. The agreement with other calculations and experiments is very satisfactory.[24, 25, 26] This stacking is somehow similar to the adsorption of aromatic molecules on graphene, where π - π interactions dominate.[24, 25, 27, 28]

Once a confident set of parameters is extracted, one can easily deal with larger systems where long-range interaction are important. For instance, it is well known that large fullerenes and nanotubes are deformed when adsorbed on a surface.[29] One can handle such phenomena with this new parametrization. The approach we propose in this paper should also deserve attention to investigate structural properties of graphite [30] or graphene sheets in which one is able to design structures like ribbons or Y-shaped junctions for nano-electronics.

Finally, the interaction of a carbon nanotube with a graphene surface is an important topic in the context of ultimate atomic force microscopy (AFM).[31] Indeed, it has been shown that a single-walled carbon nanotube anchored at the apex of an AFM tip increases the sensitivity of the measured forces. By cycling approaches and retracts, one is able to extract physical information of the surface, such as energy dissipation.[32, 26, 33] Nevertheless, it is crucial to understand the mechanical behavior of the nanotube, by mixing strong deformations when it vertically approaches the surface with long-range van der Waals contributions when it is deposited on it. We believe that all these developments can be addressed by the on-the-fly calculations using the algorithm and the numerical tool described in this paper.

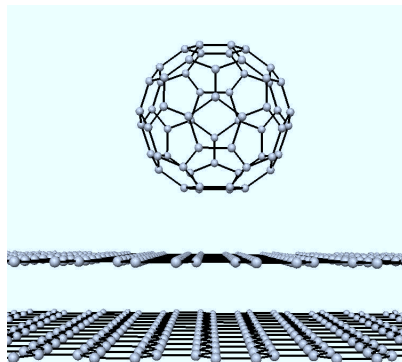


Figure 4: Buckminsterfullerene in interaction with a graphite surface thanks to the fitted pairwise van der Waals term. The fitting procedure has been performed to respect the fullerene-surface stable distance.

4. A new adaptive algorithm

The incremental algorithm presented in the previous section is able to take advantage of the freezing of groups of atoms. We now focus on the choice of the active degrees of freedom. An adaptive algorithm has been developed in the framework of internal coordinates and articulated bodies simulation [34]. Here, we extend this approach to cartesian coordinates in order to address the problem of providing interactive and efficient structure optimization. This is the essential feature that guides the user in our modeler.

The adaptive approach in cartesian coordinates is to decide for each atom if it might move or be frozen in space. Thus, the algorithm has to decide to activate or freeze an atom depending on the norm of the force applied to the atom. This decision is made by comparing the norm with a certain threshold value. This threshold value is defined either automatically based on the system state (*e.g.* average acceleration, maximum acceleration) or

can be simply predefined by the user. This cartesian adaptive approach is illustrated in Figure 5. In this paper, we implement the cartesian adaptive approach using the steepest descent minimization algorithm,[35] which serves to guide the model towards an optimized geometry. In practice, the adaptive minimizer does not use the components of the energy gradient (the partial derivatives) that are below a given threshold. This amounts to providing a perturbed minimization direction. This is detailed in Sect. 5 as an application of this new adaptive approach coupled with the incremental update of the Brenner potential.

An important feature in our algorithm is that the threshold criteria and the active atoms set are not computed from scratch. At the end of each point energy calculation a list of the atoms with force change and the corresponding new threshold are incrementally computed as explained in Sect. 3. In order to avoid the linear cost of determining the new set of active atoms, a binary tree is used to represent the system. Each leaf node represents an individual atom, while each internal node represents a set of atoms. Each leaf node stores the norm of the force applied to the corresponding atom. Each non-leaf node stores the maximum of the two force norms of its children, as illustrated in Figure 6. We use two tree passes in order to update tree nodes' values and to determine the new active atoms. In the first, bottom-up pass, force norms are updated in a sub-tree of the binary tree, starting from the leaves with modified norms, in $O(k^{old}(\log(\frac{n}{k^{old}}) + 1))$ times where k^{old} is the number of active atoms and n the total number of atoms. In the second, top-down pass, the new active atoms (i.e. the atoms with the force norms which are now the largest), are determined in $O(k^{new}(\log(\frac{n}{k^{new}}) + 1))$ times

where k^{new} is the new number of active atoms. This process is illustrated in Figure 6 as well.

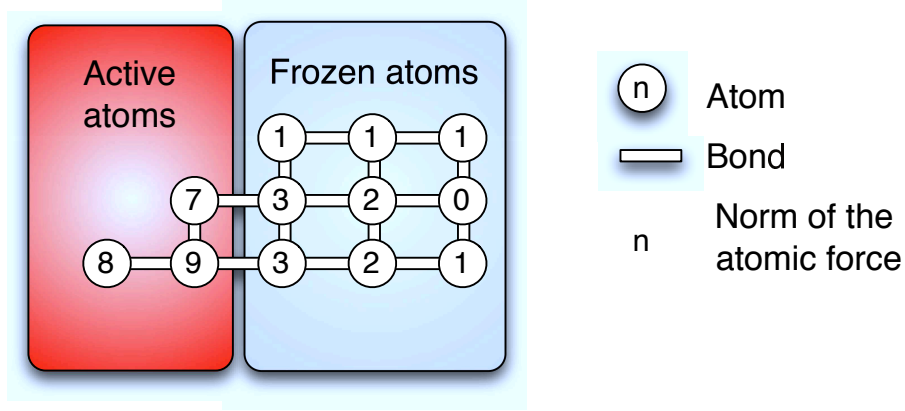


Figure 5: The cartesian adaptive approach chooses the most active atoms as mobile particles. The other atoms are frozen in space. This permits to allow fast local minimization with a small active region and an important frozen part. The mobile atoms are chosen automatically based on the information from atomic forces. In this schematic case, the user has arbitrarily decided to allow only three mobile atoms. As a result, the three leftmost atoms are activated because they have the largest forces.

5. Results and applications

5.1. Computational complexity of the Brenner potential incremental update

Our algorithm is linear with the size of the working sets $\hat{\mathcal{A}}_{atoms}$, $\hat{\mathcal{B}}$, $\hat{\mathcal{A}}$ and $\hat{\mathcal{D}}$. Based on the assumption that each atom may only have a bounded number of neighbors (which is true for realistic systems), these sets have a linear size in the number of bond updates $|\mathcal{B}_{updated}|$. Thus, the overall complexity is linear in the number of relative positions updates indexed in

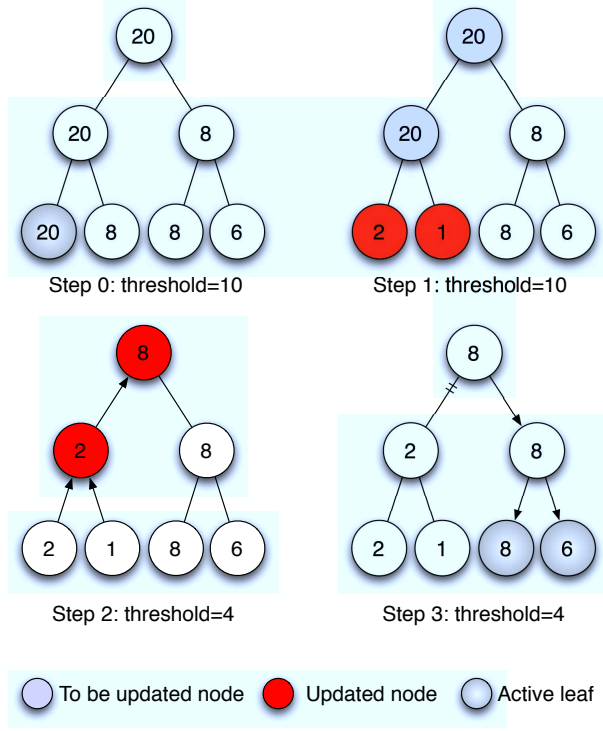


Figure 6: This figure illustrates the procedure to determine the active zone, when the threshold is automatically set to half the largest atomic force norm. The algorithm uses a binary tree to represent the atomic system, in which each leaf corresponds to one atom. In this example, the four leaves correspond, from left to right to atoms 1 to 4. Internal tree nodes represent *groups* of atoms. The value indicated in each leaf node is the norm of the force applied to its corresponding atom. For internal nodes, this value is the maximum of the norms of the forces applied to atoms in the corresponding group. In step 0, the threshold is automatically set to 10. As a result, only atom 1 moves. In step 1, the potential is incrementally updated, and the norms of the forces applied to atoms 1 and 2 are updated. In step 2, the values associated to the tree nodes are incrementally updated through a bottom-up pass that starts from the modified leaf nodes values. Because of this bottom-up update, the adaptive threshold becomes equal to 4. In step 3, the new active atoms are determined through a top-down pass, by visiting only the nodes that have a value larger than the adaptive threshold.

$B_{updated}$, since each relative position update has an impact on a bounded number of terms.

In practice, the complexity of the algorithm depends on the local topology of the system and the bond update pattern. To demonstrate this, we generated a carbon nanotube of 8000 atoms thanks to TubeGen,[36] and perturbed the positions of a set of N atoms, $1 < N < 8000$. This benchmark is presented in Figure 7 where the solid curves represent the computational cost of the forces and potential update. To illustrate the dependency of this cost on the activation pattern (*i.e.* the evolving set of active atoms), both random and continuous activations are compared. The curve with a continuous activation pattern shows a linear behavior. Indeed, continuous activation is equivalent to consider a nanotube system of length equal to the length of the active region plus a buffer zone. Thus, the linear behavior is a direct consequence of the linear dependency of the Brenner potential on the number of bonds. The computational cost of the random activation step is more important, since updated relative positions are isolated in space with a high probability, which results in a larger number of terms that have to be recomputed. In general, for any molecular system, there exists two constants C_{min} and C_{max} such that the total time T_U to update the potential and forces after updating N_b relative positions satisfies:

$$C_{min}N_b \leq T_U \leq C_{max}N_b. \quad (15)$$

Figure 7 shows the update time T_U as a function of the number of bonds with a relative motion.

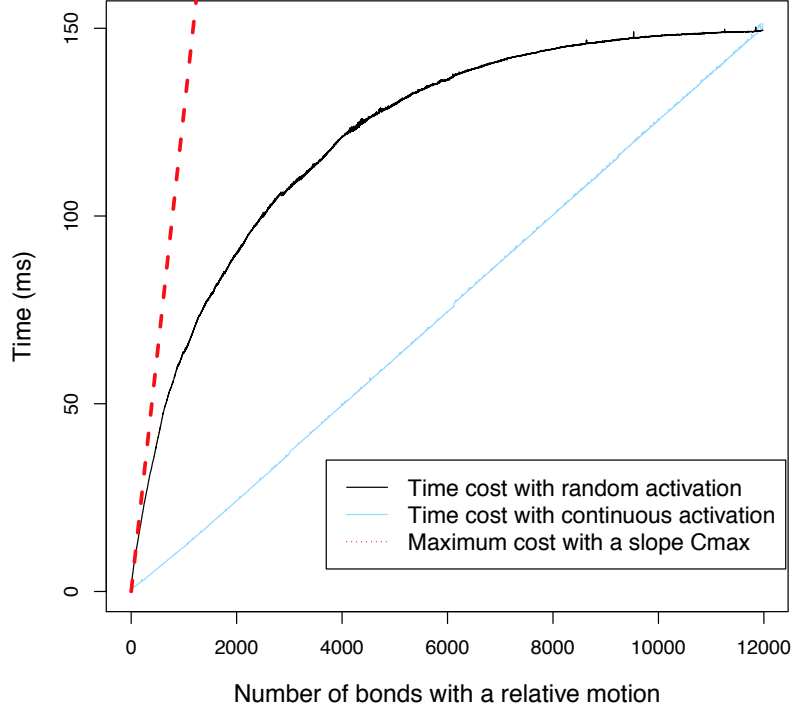


Figure 7: Forces and potential incremental update complexity of a carbon nanotube composed of 8000 atoms and 11980 bonds. In this benchmark, a number of pairs of atoms have a relative motion and the solid curves represent the time cost of the forces and potential update. The curve of the random activation pattern is steep at the beginning as the relative motions are isolated in space with a high probability and cause a maximum number of terms to recompute. Here, the slope is similar to the slope C_{max} . For a large number of relative motions, the curve becomes flat since a new relative motion does not add many new terms to recompute. Thus the marginal cost of a new relative motion is minimal. The curve with a continuous activation pattern illustrates the linear dependency of the Brenner potential on the number of bonds, since this continuous activation is equivalent to consider a nanotube system with an increasingly longer length. Its slope is also approximately equal to the slope of the minimal cost C_{min} .

5.2. *Efficient and interactive energy minimization of local deformation*

Energy minimization is a classical tool for structure analysis and has already been used with the Brenner potential.[37, 38, 39] We apply the *adaptive* algorithm presented in Sect. 4 with the incremental update to the steepest descent minimization. Classical minimization algorithms typically update the positions of all atoms at each iteration. As a result, all energies and forces are updated at each iteration. In many cases, however, the displacement of many atoms will be small, and will not contribute much to a decrease in potential energy. We thus propose to focus the computational resources by freezing in space the least mobile atoms at each iteration. As discussed in the previous section, our incremental algorithm is able to take advantage of the freezing of groups of atoms. As a result, computational resources will be automatically focused on the most mobile areas of the system. This approach is expected to speed up energy minimization of systems with non-uniform force distributions.

We demonstrate this approach on a locally deformed carbon nanotube with a few broken cycles, which should converge to the reference structure (Figure 8).

We minimize the deformed structure using an automatic, relative threshold (only the atoms that have an acceleration larger than half the maximum atom acceleration at the current time step will move). We stop the minimization procedure when the root-mean-square-deviation (RMSD)[40] between the structure being minimized and the reference one becomes smaller than 0.01 Å. The Brenner potential accuracy is in the order of 0.01 Å,[41] and thus a better convergence would not be significant. Figure 9 plots the RMSD as

a function of wall-clock time. The adaptive minimizer converges 3.5 times faster than the original minimizer to the RMSD of 0.1 Å and more than two times faster to the RMSD of 0.01 Å. Indeed, in a first period, the deformation is very local and an adaptive approach is very efficient. Then, when the force distribution is more uniform, the non-adaptive approach becomes faster due to the overhead of the adaptive algorithm. This is illustrated by the second curve in Figure 9, which shows the performance of the adaptive steepest descent when all degrees of freedom are active. In this case, detecting which terms have to be updated is useless and time consuming. In our implementation, the overhead is around 6 %. Figure 10 presents the snapshot of a video showing the computational benefits of the incremental approach in a similar situation of a nanotube locally deformed with a few broken cycles. The video is accessible on the web: [link](#).

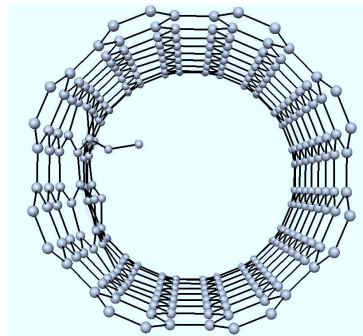
To understand why our algorithm can accelerate energy minimization, recall that the adaptive steepest descent algorithm does not follow the classical gradient direction on the energy surface but a close, truncated one, to reduce the cost of a minimization step. In fact, this is equivalent to looking for a more efficient direction in the sense of the ratio between the energy decrease and the effective computational cost of the position update. The classical steepest descent algorithm is based on the idea that the most efficient way to reduce the energy is to follow the steepest direction on the energy surface. However, in this argument, the update of the energy potential is seen as a black box and thus the algorithm does not consider the (potentially variable) update cost. We believe this update cost should be taken into account in order to choose the minimization direction, and that it might be beneficial to

try and maximize the energy decrease *per computational unit* at each time step. The overall benefit of such a strategy may depend on the general shape of the potential energy surface. Therefore, it is still difficult to predict the advantage of such an approach in the general case. We have shown that our adaptive approach allows us to finely tune between the update cost and the energy decrease, which speeds up the minimization when the force distribution is localized (when the force distribution is uniform, moving all atoms whose force is larger than half the maximum force will lead to activating *all* atoms, so that our truncated gradient will be equal to the actual gradient).

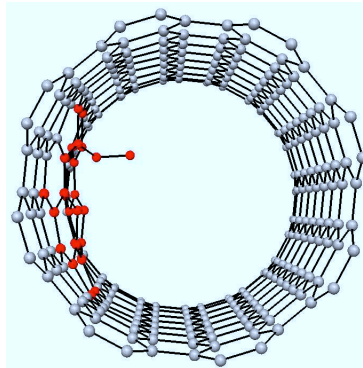
5.3. Interactive modeling

The classical approach to study hydrocarbon systems is typically to first build a structural model, and then to perform geometry optimization to prepare the model for simulation. Because of the complexity of the physics at this scale, however, this approach is not efficient in many cases. Indeed, when the user creates the initial structural model, he can easily produce high-energy, unstable structures, in particular for large systems. As a result, the geometry optimization step may bring on very different structures from the ones targeted by the user, so that the user would have to go back to the construction step. Overall, creating a complex structural model by decoupling the construction step from the geometry optimization step may be particularly ineffective.

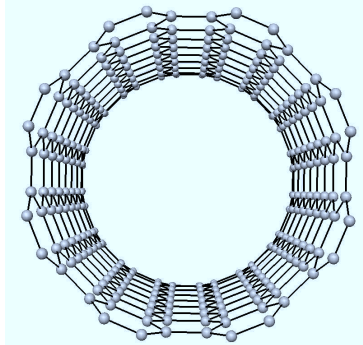
Thus, we believe an effective way to help the user to construct a structural model of an hydrocarbon system is to perform geometry optimization *during* construction. In practice, at each time step, we combine user actions with energy minimization to produce structures that have lower potential energies.



Locally deformed nanotube



Activation zone during the adaptive minimization



Reference nanotube structure

Figure 8: Different snapshots of an experiment where a carbon nanotube is being relaxed after a local deformation. The adaptive steepest descent minimizer automatically focuses the computational resources on the most mobile atoms. This leads to a more efficient minimization scheme in such a situation.

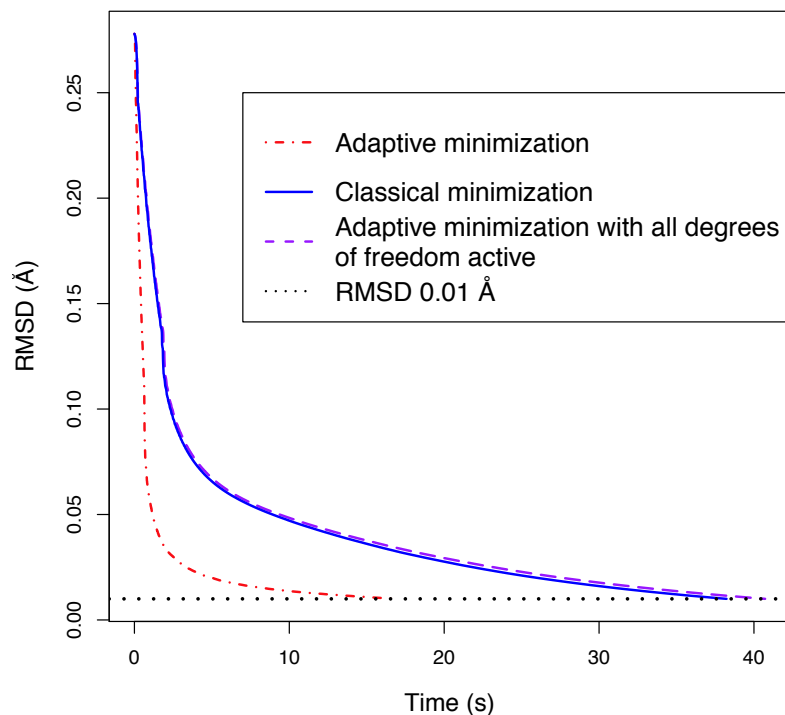


Figure 9: Performance of the adaptive minimizer in the case of a local deformation of a nanotube. The three curves show the performance of three different approaches to recover the stable structure with a RMSD of 0.01 \AA of the nanotube shown in Figure 8. The most efficient one is the adaptive steepest descent minimizer, which focuses the computational resources to the most mobile atoms. The two other methods follow the same path on the energy surface and are similar in performance. However, because of the computational overhead of the incremental update, the classical steepest descent method is about 6 % faster than the adaptive one when all atoms are activated.

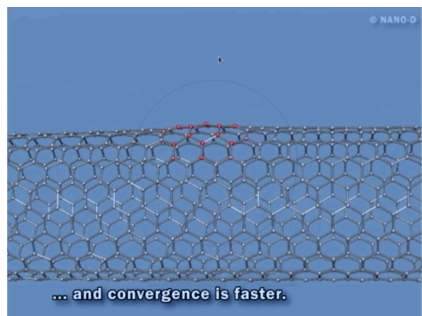


Figure 10: Snapshot of a video showing the computational benefits of the incremental approach when minimizing large structures that have been locally deformed.

This physically-based interactive modeling requires a very low computational cost at each simulation step. One possibility for the user is to act on a small area of the system, creating local deformations only. Another possibility is to perform deformations which require physics simulation in local areas only (such as when two large groups of atoms move rigidly and interact with each other through a few atoms only). The minimizer presented in the previous section has been designed for such cases, since the force distribution should reach a peak in a local area. Furthermore, the adaptive minimizer can guarantee interactivity independently of the size of the system (by directly controlling the threshold value and thus the number of allowed components in the truncated gradient — a choice specifically based on available computational resources). We also recall that the reactive Brenner potential is particularly adapted, since a fundamental step in structural modeling is the creation and destruction of chemical bonds. This makes coupling an adaptive minimization algorithm with an incremental reactive potential update algorithm an effective way to achieve interactive modeling.

Remark that a local deformation can have a local impact as well as a global impact. The locality of the active zone does not prevent the minimizer to produce delocalized deformation. Indeed in the spirit of the adaptive paradigm, the active region is automatically determined thanks to the information of the atomic forces and it might go through the whole system in a linear number of time steps when needed. An important benefit is that a local user action might lead to a non-local, coordinated displacement of a group of atoms (for example when a user pulls on an atom that belongs to a chain), which enables the user to induce large, physically-based deformations of the nanosystem. More generally, such an approach allows the user to perform even non-local editing (*e.g.* rotation and translation of entire groups of atoms), while ensuring that the underlying physics is taken into account.

We illustrate this possibility in Figure 14, where a complex, original nanosystem is built in a few minutes thanks to the interactivity of the physically-based modeler. In Step 0, the user loads two equilibrated graphene sheets in the modeling tool. In Step 1, the user connects both sheets (left part in Figure 14), simply by pulling on atoms. As the two sheets are being attached, the minimizer progressively induces a curvature in the structure, which actually helps the user to attach the sheets. In Step 2, the right part is connected, again by pulling on atoms. In the final step, the two remaining sides have been attached, and a few atoms has been removed from the corners to form a "nano-vesicle". A closed bilayer graphene has already been observed experimentally by high-resolution transmission electron microscopy [42]. This shows that the present numerical tool should be efficient for experimentalists to design and to build graphitic nanostructures.

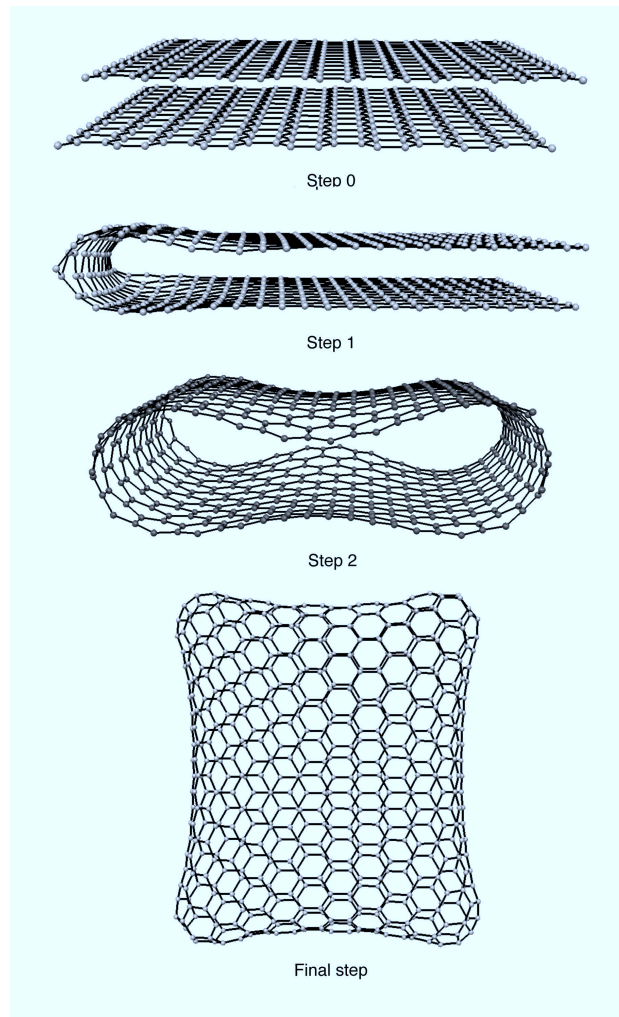


Figure 11: Different steps to prototype a "nano-vesicle" with the adaptive interactive modeler. Starting with two graphene sheets, the adaptive methodology enables the user to prototype a new structure by applying forces on atoms with the mouse. We demonstrate the efficiency of the modeler by the creation of this new original carbon structure.

Similarly, carbon nanotubes with a large variety of defects have already been observed experimentally by high-resolution transmission electron microscopy [43] and our methodology allows to intuitively prototype such complex geometries. Figure 11 shows how our modeler may be used to interactively close a nanotube. The user has chosen to add atoms and put them close together to create cycles of 5, 6 or 7 carbon atoms. The tool automatically induces the corresponding curvature on the system leading to the design of the apex of the nanotube. As discussed above, the process could have been done with a nanotube of any size as the computational cost of each minimization step only depends on the number of active atoms. We also present two videos to illustrate the efficiency of the interactive editor. Figure 12 presents a snapshot of a video in which a user designs the apex of a nanotube. Figure 13 presents a snapshot of a video in which the user adds structural defects to a carbon nanotube model. The videos are also accessible on the web: [link](#).

6. Conclusion

In this paper, we have presented two algorithms enabling digital prototyping of hydrocarbon structures. Interactivity is guaranteed by the adaptive minimization algorithm coupled with the incremental update algorithm of the Brenner potential. This allows us to focus computational resources on the regions that are the most affected by user actions. Furthermore, when user actions have a local impact, the adaptive approach appears to be an effective way to rapidly reach neighboring energy minima, which helps the user build realistic structures.

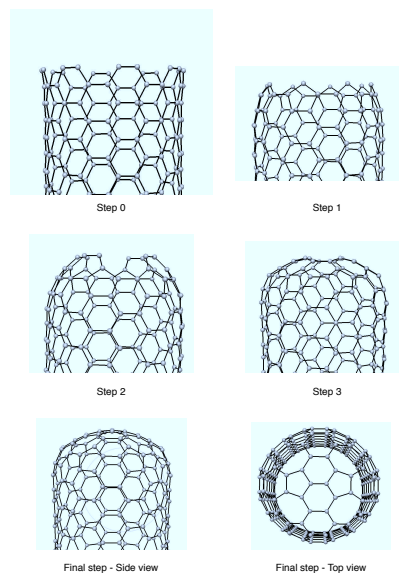


Figure 12: Snapshots of a nanotube capping process with the adaptive interactive modeler. A classical nanotube is loaded in the modeler and the closure is performed by applying forces on bordering atoms. The curvature at the tip appears automatically thanks to the adaptive minimization running while the user edits the structure's topology. Thanks to the adaptive methodology, this operation can be done in a few minutes.

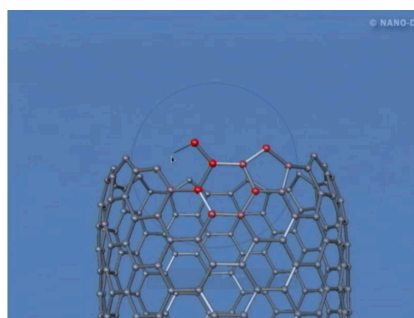


Figure 13: Snapshot of a video showing a user interactively prototyping the closure of a carbon nanotube.

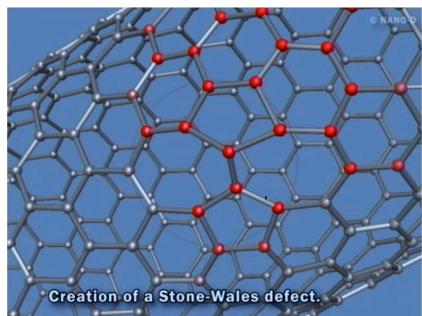


Figure 14: Snapshot of a video demonstrating how our algorithm may be used to interactively add structural defects to a carbon nanotube model.

The proposed modified steepest descent algorithm may allow for an important speed up when the energy gradient is non-uniform. This happens when the user stresses the system locally. This might happen in other contexts as well (e.g. docking problems). We would like to extend our approach in several directions. First we have mentioned that the overhead resulting from the incremental update and the marginal cost of a relative motion may make our approach slower than the classical one in some cases. This suggests the need for hybrid minimization algorithms which would be able to switch between the classical and the adaptive approach at runtime based on the distribution of forces in the system. Also, we would like to investigate whether our new cartesian adaptive approach to compute a truncated gradient could be useful in combination with other minimization algorithms. Finally we want to determine whether the cartesian adaptive approach may be combined with an adaptive internal degrees-of-freedom simulation method [18], i.e. group atoms into clusters which might deform while creating and breaking bonds, which should provide great functionality both for simulation

and digital prototyping.

Despite the computational efficiency of the Brenner potential, the cost of molecular dynamics or Monte Carlo simulations may still be too high when the size of the system increases, or when long simulations are required. A classical approach to address this problems is to use parallel computers.[44, 45, 46] We believe that we may extend our work in the future as an alternative answer. Indeed, our incremental algorithm will make it possible to recompute changing terms only, based on the list of moving atoms, and can thus be used in any Monte Carlo methods based on the Brenner potential [19, 47, 15]. We are also exploring the possibility of adaptive dynamic simulation, with non-zero temperature. Indeed, the proposed tool helps the user produce molecular models that have optimized geometries. However, these structures should thus be analyzed through simulations, to compute *e.g.* free energy barriers, statistical properties, etc.

Finally, another important challenge in interactive molecular modeling that we would now like to address is to extend the adaptive approach to semi-empirical quantum mechanical models, which would provide better accuracy as well as interactive electronic structure computation.

Software availability

The new algorithms will be made available through SAMSON (Software for Adaptive Modeling and Simulation of Nanosystems) [48], being developed in the NANO-D group .

Acknowledgement

We wish to thank Brigitte Bidégaray-Fesquet for careful reading of the manuscript and helpful remarks. This work was supported in part by the French National Research Agency (ANR) through the COSINUS program (project SAMSON ANR-08-COSINUS-015).

7. Appendix

7.1. Potential expression

Recall that the Brenner potential is a sum over the chemical bonds.

$$E_b = \sum_i \sum_{j>i} [V^R(r_{ij}) - b_{ij}V^A(r_{ij})] \quad (16)$$

where b_{ij} is the bond order. The functions V^R and V^A are respectively repulsive and attractive terms.

$$V^R(r) = f(r)(1 + Q)Ae^{-\alpha r} \quad (17)$$

$$V^A(r) = f(r) \sum_{n=1,3} B_n e^{-\beta_n r} \quad (18)$$

These two terms are limited to the nearest neighbors by a simple smooth cut-off function f . The values of Q , A and B_n have been determined for each atomic type.

$$f(r) = \begin{cases} 1 & r < D_{ij}^{min} \\ 1 + \cos\left(\frac{(r-D_{ij}^{min})}{(D_{ij}^{max}-D_{ij}^{min})}\right)/2 & D_{ij}^{min} < r < D_{ij}^{max} \\ 0 & r > D_{ij}^{max} \end{cases} \quad (19)$$

where D_{ij}^{max} and D_{ij}^{min} are type-dependent parameters, and r is the distance between atoms i and j . Originally, the parameters have been determined for carbon and hydrogen only [7]. Later they have been extended to Silicon [14, 49, 50, 51]. Bonds are defined dynamically *via* the f function. As a result, the potential has the ability to describe chemical reactions.

7.1.1. Bond order

The benefit of a reactive potential mainly depends on the bond order. In the Brenner potential case, we can decompose it into four terms depending on environment parameters including angles, dihedral angles and number of neighbors.

$$b_{ij} = \frac{1}{2}[b_{ij}^{\sigma-\pi} + b_{ji}^{\sigma-\pi}] + \Pi_{ij}^{RC} + b_{ji}^{DH} \quad (20)$$

7.1.2. Angle influence

The first term in Equation 20 is given as a sum over the angle θ_{ijk} in which the bond (i, j) is involved.

$$b_{ij}^{\sigma-\pi} = \left[1 + \sum_{k \neq i, j} f_{ik}(r_{ik}) G(\cos(\theta_{ijk})) e^{\lambda_{ijk}} + P_{ij}(N_i^C, N_i^H) \right]^{-1/2} \quad (21)$$

where G is a spline and P a bicubic spline. λ_{ijk} is a constant. N_i^C and N_i^H are respectively the number of carbon and hydrogen neighbors. The last numbers are continuous values as they are computed by a sum of the f function.

$$\begin{aligned} N_i^C &= \sum_{k \neq i, j}^{carbon \ atoms} f_{ik}(r_{ik}) \\ N_i^H &= \sum_{k \neq i, j}^{hydrogen \ atoms} f_{ik}(r_{ik}) \end{aligned} \quad (22)$$

For the description of our incremental update algorithm we define :

$$S_{ij}^{\sigma-\pi} = \sum_{k \neq i, j} f_{ik}(r_{ik}) G(\cos(\theta_{ijk})) e^{\lambda_{ijk}} \quad (23)$$

7.1.3. Radical energetics and π bonds influence

The third term in Equation 20 depends on whether the bond has a radical character and is part of a conjugate system.

$$\Pi_{ij}^{RC} = F_{ij}(N_i^t, N_j^t, N_{ij}^{conj}) \quad (24)$$

where F is a tricubic spline. N_i^t is the total number of neighbors of atom i . N_{ij}^{conj} determines whether the bond is part of a conjugate system from the information of the number of neighbors of each neighbor. This term is decomposed in parts that depend either on atom i , or on atom j :

$$N_{ij}^{conj} = 1 + N_i^{aux} + N_j^{aux} \quad (25)$$

7.1.4. Dihedral term

The last term in Equation. 20 is a sum over the dihedral angles Θ_{ijkl}

$$b_{ji}^{DH} = T_{ij}(N_i^t, N_j^t, N_{ij}^{conj}) \left[\sum_{k \neq i,j} \sum_{l \neq i,j} (1 - \cos(\Theta_{ijkl})) f_{ik}(r_{ik}) f_{jl}(r_{jl}) \right] \quad (26)$$

where T is a tricubic spline. This term favors a planar configuration in the case of double C-C bonds. For the description of our incremental update we define :

$$S_{ji}^{DH} = \sum_{k \neq i,j} \sum_{l \neq i,j} (1 - \cos(\Theta_{ijkl})) f_{ik}(r_{ik}) f_{jl}(r_{jl}) \quad (27)$$

For a complete description of the potential, please refer to [13].

References

- [1] F. Ample, S. Ami, C. Joachim, F. Thiemann, G. Rapenne, A Morse manipulator molecule for the modulation of metallic Shockley surface states, Chem. Phys. Lett. 434 (2007) 280–285.

- [2] H. Dietz, S. M. Douglas, W. M. Shih, Folding DNA into twisted and curved nanoscale shapes, *Science* 325 (2009) 725–730.
- [3] L. Grill, K.-H. Rieder, F. Moresco, G. Rapenne, S. Stojkovic, X. Bouju, C. Joachim, Rolling a single molecular wheel at the atomic scale, *Nature Nanotech.* 2 (2007) 95–98.
- [4] C. Joachim, H. Tang, F. Moresco, G. Rapenne, G. Meyer, The design of a nanoscale molecular barrow, *Nanotechnology* 13 (2002) 330–335.
- [5] Y. Shirai, A. J. Osgood, Y. Zhao, Y. Yao, L. Saudan, H. Yang, C. Yu-Hung, L. B. Alemany, T. Sasaki, J.-F. Morin, J. M. Guerrero, K. F. Kelly, J. M. Tour, Surface-rolling molecules, *J. Am. Chem. Soc* 128 (2006) 4854–4864.
- [6] A. C. T. Van Duin, S. Dasgupta, F. Lorant, W. A. Goddard III, ReaxFF: a reactive force field for hydrocarbons, *J. Phys. Chem. A* 105 (2001) 9396–9409.
- [7] D. W. Brenner, Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films, *Phys. Rev. B* 42 (1990) 9458–9471.
- [8] D. W. Brenner, The art and science of an analytic potential, *phys. stat. sol. (b)* 217 (2000) 23–40.
- [9] S. B. Sinnott, R. Andrews, D. Qian, A. M. Rao, Z. Mao, E. C. Dickey, F. Derbyshire, Model of carbon nanotube growth through chemical vapor deposition, *Chem. Phys. Lett.* 315 (1999) 25–30.

- [10] S. J. Harris, D. G. Goodwin, Growth on the reconstructed diamond (100) surface, *J. Phys. Chem.* 97 (1993) 23–28.
- [11] T. Belytschko, S. P. Xiao, G. C. Schatz, R. S. Ruoff, Atomistic simulations of nanotube fracture, *Phys. Rev. B* 65 (2002) 235430.
- [12] D. W. Brenner, J. A. Harrison, C. T. White, R. J. Colton, Molecular dynamics simulations of the nanometer-scale mechanical properties of compressed Buckminsterfullerene, *Thin Solid Films* 206 (1991) 220–223.
- [13] D. W. Brenner, O. A. Shenderova, J. A. Harrison, S. J. Stuart, B. Ni, S. B. Sinnott, A second-generation reactive empirical bond order (REBO) potential energy expression for hydrocarbons, *J. Phys.: Condens. Mat.* 14 (2002) 783–802.
- [14] A. J. Dyson, P. V. Smith, Extension of the Brenner empirical interatomic potential to C-Si-H systems, *Surf. Sci.* 355 (1996) 140–150.
- [15] J. H. Los, A. Fasolino, Intrinsic long-range bond-order potential for carbon: Performance in Monte Carlo simulations of graphitization, *Phys. Rev. B* 68 (2003) 024107.
- [16] S. J. Stuart, A. B. Tutein, J. A. Harrison, A reactive potential for hydrocarbons with intermolecular interactions, *J. Chem. Phys.* 112 (2000) 6472.
- [17] D. W. Brenner, S. B. Sinnott, J. A. Harrison, O. A. Shenderova, Simulated engineering of nanostructures, in: *In Fourth Foresight Conference on Molecular Nanotechnology*.

- [18] R. Rossi, M. Isorce, S. Morin, J. Flocard, K. Arumugam, S. Crouzy, M. Vivaudou, S. Redon, Adaptive torsion-angle quasi-statics: a general simulation method with applications to protein structure analysis and design, *Bioinformatics* 23 (2007) 408–417.
- [19] J. H. Los, A. Fasolino, Monte Carlo simulations of carbon-based structures based on an extended Brenner potential, *Comput. Phys. Commun.* 147 (2002) 178–181.
- [20] J. H. Los, L. M. Ghiringhelli, E. J. Meijer, A. Fasolino, Improved long-range reactive bond-order potential for carbon. I. Construction, *Phys. Rev. B* 72 (2005) 214102.
- [21] L. M. Ghiringhelli, J. H. Los, A. Fasolino, E. J. Meijer, Improved long-range reactive bond-order potential for carbon. II. Molecular simulation of liquid carbon, *Phys. Rev. B* 72 (2005) 214103.
- [22] L. J. Karssemeijer, A. Fasolino, Phonons of graphene and graphitic materials derived from the empirical potential LCBOPII, *Surf. Sci.* 605 (2011) 1611–1615.
- [23] P. A. Gravi, M. Devel, P. Lambin, X. Bouju, C. Girard, A. A. Lucas, Adsorption of C_{60} molecules, *J. Chem. Phys.* 53 (1996) 1622–1629.
- [24] Y. J. Dappe, M. A. Basanta, F. Flores, J. Ortega, Weak chemical interaction and van der Waals forces between graphene layers: A combined density functional and intermolecular perturbation theory approach, *Phys. Rev. B* 74 (2006) 205434.

- [25] Y. J. Dappe, J. Ortega, F. Flores, Intermolecular interaction in density functional theory: Application to carbon nanotubes and fullerenes, *Phys. Rev. B* 79 (2009) 165409.
- [26] M. Seydou, Y. J. Dappe, S. Marsaudon, X. Bouju, A.-M. Bonnot, J.-P. Aimé, Atomic force microscope measurements and LCAO- S^2 + vdW calculations of contact length between carbon nanotube and graphene surface, *Phys. Rev. B* 83 (2011) 045410.
- [27] Y. V. Shtogun, L. M. Woods, Many-body van der Waals interaction between graphitic nanostructures, *J. Phys. Chem. Lett.* 1 (2010) 1356–1362.
- [28] J. Björk, F. Hanke, C.-A. Palma, P. Samori, M. Cecchini, M. Persson, Adsorption of aromatic and anti-aromatic systems on graphene through π - π stacking, *J. Phys. Chem. Lett.* 1 (2010) 3407–3412.
- [29] T. Hertel, R. E. Walkup, P. Avouris, Deformation of carbon nanotubes by surface van der Waals forces, *Phys. Rev. B* 58 (1998) 13870–13873.
- [30] S. Lebègue, J. Harl, T. Gould, J. G. Ángyán, G. Kresse, J. F. Dobson, Cohesive properties and asymptotics of the dispersion interaction in graphite by the random phase approximation, *Phys. Rev. Lett.* 105 (2010) 196401.
- [31] N. R. Wilson, J. V. Macpherson, Carbon nanotube tips for atomic force microscopy, *Nature Nanotech.* 4 (2009) 483–491.
- [32] M. Seydou, S. Marsaudon, J. Buchoux, J.-P. Aimé, A.-M. Bonnot,

- Molecular mechanics investigations of carbon nanotube and graphene sheet interaction, *Phys. Rev. B* 80 (2009) 245421.
- [33] M. Delmas, M. Monthieux, T. Ondarçuhu, Contact angle hysteresis at the nanometer scale, *Phys. Rev. Lett.* 106 (2011) 136102.
 - [34] S. Redon, N. Galoppo, M. C. Lin, Adaptive dynamics of articulated bodies, *ACM Transactions on Graphics (TOG)* 24 (2005) 936945.
 - [35] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical recipes in FORTRAN: The art of scientific computing*, 2nd ed. (1992).
 - [36] J. T. Frey, D. J. Doren, Tubegen 3.3 (web-interface, <http://turin.nss.udel.edu/research/tubegenonline.html>), University of Delaware (2005).
 - [37] O. Hod, E. Rabani, R. Baer, Carbon nanotube closed-ring structures, *Phys. Rev. B* 67 (2003) 195408.
 - [38] Y. Huang, J. Wu, K. C. Hwang, Thickness of graphene and single-wall carbon nanotubes, *Phys. Rev. B* 74 (2006) 245413.
 - [39] C. D. Reddy, S. Rajendran, K. M. Liew, Equilibrium configuration and continuum elastic properties of finite sized graphene, *Nanotechnology* 17 (2006) 864–870.
 - [40] D. L. Theobald, Rapid calculation of RMSDs using a quaternion-based characteristic polynomial, *Acta Cryst. A* 61 (2005) 478–480.

- [41] A. V. Petukhov, A. Fasolino, Reconstructions of diamond (100) and (111) surfaces: Accuracy of the Brenner potential, *phys. stat. sol. (a)* 181 (2000) 109–114.
- [42] Z. Liu, K. Suenaga, P. J. F. Harris, S. Iijima, Open and Closed Edges of Graphene Layers, *Phys. Rev. Lett* 102 (2009) 015501.
- [43] T.W. Chamberlain, J.C. Meyer, J. Biskupek, J. Leschner, A. Santana, N.A. Besley, E. Bichoutskaia, U. Kaiser, A.N. Khlobystov, Reactions of the inner surface of carbon nanotubes and nanoprotrusion processes imaged at the atomic scale, *Nature Chemistry* 3 (2011) 732–737.
- [44] I. Rosenblum, J. Adler, S. Brandon, Multi-processor molecular dynamics using the Brenner potential: parallelization of an implicit multi-body potential, *Int. J. Mod. Phys. C* 10 (1999) 189–203.
- [45] A. Omeltchenko, J. Yu, R. K. Kalia, P. Vashishta, Crack front propagation and fracture in a graphite sheet: A molecular-dynamics study on parallel computers, *Phys. Rev. Lett.* 78 (1997) 2148–2151.
- [46] D. Srivastava, S. T. Barnard, Molecular dynamics simulation of large-scale carbon nanotubes on a shared-memory architecture, in: *Supercomputing '97: Proceedings of the 1997 ACM/IEEE conference on Supercomputing*, ACM, New York, NY, USA, 1997, pp. 1–10.
- [47] J. Pikunic, C. Clinard, N. Cohaut, K. E. Gubbins, J.-M. Guet, R. J.-M. Pellenq, I. Rannou, J.-N. Rouzaud, Structural modeling of porous

carbons: Constrained reverse Monte Carlo method, *Langmuir* 19 (2003) 8565–8582.

- [48] NANO-D, <http://nano-d.inrialpes.fr/> .
- [49] J.-Z. Que, M. W. Radny, P. V. Smith, A. J. Dyson, Application of the extended Brenner potential to the Si(111)7×7:H system I: cluster calculations, *Surf. Sci.* 444 (2000) 123–139.
- [50] J.-Z. Que, M. W. Radny, P. V. Smith, A. J. Dyson, Application of the extended Brenner potential to the Si(111)7×7:H system II: periodic calculations, *Surf. Sci.* 444 (2000) 140–155.
- [51] J. D. Schall, G. Gao, J. A. Harrison, Elastic constants of silicon materials calculated as a function of temperature using a parametrization of the second-generation reactive empirical bond-order potential, *Phys. Rev. B* 77 (2008) 115209.