Algorithmes incrémentaux pour la théorie de la fonctionnelle de la densité

François Rousse

To cite this version:

François Rousse. Algorithmes incrémentaux pour la théorie de la fonctionnelle de la densité. Modélisation et simulation. Université Grenoble - Alpes, 2019. Français. NNT : . tel-02435781v1

HAL Id: tel-02435781
https://hal.science/tel-02435781v1

Submitted on 11 Jan 2020 (v1), last revised 4 Mar 2020 (v2)

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA
COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : Mathématiques Appliquées

Arrêté ministériel : 25 mai 2016

Présentée par

François ROUSSE

Thèse dirigée par **Stephane REDON**, DR

préparée au sein du **Laboratoire Laboratoire Jean Kuntzmann**
dans **l'École Doctorale Mathématiques, Sciences et technologies de l'information, Informatique**

Algorithmes incrémentaux pour la théorie de la fonctionnelle de la densité sans orbitale

Incremental Algorithms for Orbital-Free Density Functional Theory

Thèse soutenue publiquement le **14 octobre 2019**,
devant le jury composé de :

Monsieur STEPHANE REDON
DIRECTEUR DE RECHERCHE, INRIA CENTRE DE GRENOBLE RHÔNE-ALPES, Directeur de thèse
Monsieur JEAN CLEROUIN
DIRECTEUR DE RECHERCHE, CEA/DAM ILE-DEFRANCE BRUYERES-LE-CHATEL, Rapporteur
Monsieur SCHNEIDER REINHOLD
PROFESSEUR, UNIVERSITE TECHNIQUE DE BERLIN ALLEMAGNE, Rapporteur
Monsieur JOHANNES DIETERICH
DOCTEUR-INGENIEUR, ADVANCED MICRO DEVICES A AUSTIN - USA, Examinateur
Monsieur PHILIPPE BLAISE
INGENIEUR DE RECHERCHE, CEA GRENOBLE, Examinateur
Monsieur FLORENT CALVO
DIRECTEUR DE RECHERCHE, CNRS DELEGATION ALPES, Président
Monsieur XAVIER BOUJU
DIRECTEUR DE RECHERCHE, CNRS DELEGATION OCCITANIE OUEST, Examinateur

Abstract

The ability to model molecular systems on a computer has become a crucial tool for chemists. Indeed molecular simulations have helped to understand and predict properties of nanoscopic world, and during the last decades have had large impact on domains like biology, electronic or materials development. Particle simulation is a classical method of molecular dynamic. In particle simulation, molecules are split into atoms, their inter-atomic interactions are computed, and their time trajectories are derived step by step. Unfortunately, inter-atomic interactions computation costs prevent large systems to be modeled in a reasonable time. In this context, our research team looks for new accurate and efficient molecular simulation models. One of our team's focus is the search and elimination of useless calculus in dynamical simulations. Hence has been proposed a new adaptively restrained dynamical model in which the slowest particles movement is frozen, computational time is saved if the interaction calculus method do not compute again interactions between static atoms. The team also developed several interaction models that benefit from a restrained dynamical model, they often updates interactions incrementally using the previous time step results and the knowledge of which particle have moved.

In the wake of our team's work, we propose in this thesis an incremental First-principles interaction models. Precisely, we have developed an incremental Orbital-Free Density Functional Theory method that benefits from an adaptively restrained dynamical model. The new OF-DFT model keeps computation in Real-Space, so can adaptively focus computations where they are necessary. The method is first proof-tested, then we show its ability to speed up computations when a majority of particle are static and with a restrained particle dynamic model. This work is a first step toward a combination of incremental First-principle interaction models and adaptively restrained particle dynamic models.

Résumé

L'informatique est devenue un outil incontournable de la chimie. En effet la capacité de simuler des molécules sur ordinateur a aidé à la compréhension du monde nanoscopic et à la prédiction de ses propriétés. La simulation moléculaire a eu ces dernières décennies un impact scientifique énorme en biologie, en électronique, en science des matériaux . . . La simulation de particules est une des méthodes classiques de dynamique moléculaire, les molécules y sont divisées en atomes, leurs interactions relatives calculées et leurs trajectoires déduites pas à pas. Malheureusement un calcul précis des interactions entre atomes demande énormément d'opérations et donc de temps, ce qui limite la portée de la simulation moléculaire à des systèmes de taille raisonnable. C'est dans ce contexte que notre équipe recherche de nouveaux modèles de simulation moléculaire rapide et précis. Un des angles de recherche est l'élimination des calculs inutiles des simulations. L'équipe a ainsi proposé un modèle de dynamique moléculaire dite restreinte de manière adaptative dans lequel le mouvement des particules les plus lentes est bloqué. Si la simulation ne recalcule pas les interactions inchangées entre atomes bloqués, le calcul des interactions est plus rapide. L'équipe a aussi développé plusieurs modèles d'interactions plus efficaces pour des modèles de dynamique restreinte de particules, ils mettent à jour les interactions de façon incrémentale en utilisant les résultats du pas de temps précédent et la liste des particules mobiles.

Dans le sillage des travaux de notre équipe de recherche, nous proposons dans cette thèse une méthode incrémentale pour calculer des interactions interatomique basées sur les modèles de Théorie de la Fonctionnelle de la Densité Sans Orbitale. La nouvelle méthode garde les calculs dans l'espace réel et peut ainsi concentrer les calculs où cela est nécessaire. Dans ce manuscrit nous vérifions cette méthode, puis nous évaluons les gains de vitesse lorsqu'une majorité de particule est bloquée, avec un modèle de dynamique restreinte. Ces travaux sont un pas vers la l'intégration de modèles d'interactions Premier-principes pour des modèles dynamiques restreint de manière adaptative.

List of publications

- François Rousse, Stéphane Redon 2019. Journal of Computational Chemistry, 2019. Incremental solver for orbital-free density functional theory
- Cyril Guedj Léonard Jaillet François Rousse Stéphane Redon 2018. Atomistic Modelling and Simulation of Transmission Electron Microscopy Images: Application to Intrinsic Defects of Graphene SIMULTECH 2018 , 11 pages
- Cyril Guedj Léonard Jaillet François Rousse Stéphane Redon 2018. Impact of Hydrogen on Graphene-based Materials: Atomistic Modeling and Simulation of HRSTEM Images AVS 65th International Symposium and Exhibition

Acknowledgements

I want to give my first thank to my PhD adviser Stéphane Redon, for the opportunity to work on this ambitious project, for all his scientific and work advise, for his patience and for his never-ending optimism. I also would like to thank Stéphane for gathering the NANO-D team, every moment with them have been incredible. Learning science with Stéphane and NANO-D members have been intense and fascinating.

I would like to express my deep gratitude to the jury members for their presence at my defense and for their questions during the very interesting discussions that followed. Defending in front of them has been deeply instructive. I extend this gratitude to the reviewers for carefully reading my thesis manuscript and for all their essential comments. I must acknowledge the European Research Council and INRIA for funding my thesis project.

I need to give a special thank to the hole NANO-D team. In particular to Léonard Jaillet for his precious help during my writings and his encouragement, Guillaumes Pagès and Alexandre Hoffmann for their scientific advise and their friendship, and Jocelyn Gaté for solving every single problem life has brought to the team's computers. Thank you to all the team members with whom I spent so many great moments: Sergei Grudinin, Sveta Artemova, Žofia Trstanova, Krishna Kant-Singh, Prince Edorh, Emilie Neveu, Silvia Dias Pinto, Khoa Nguyen, Karina Machado, Didier Devaurs, Dima Marin, Yassine Naimi, Masha Kadukova among others, and a special thank to Imma and Julie for their work and most importantly for their presence and their cheerfulness. My PhD has been longer than planned but I loved every moment spent with all of you.

I have spent the other part of my time in Grenoble finding friends that I cannot forget. So thank you Elodie, Thomas, Cédric, Elora, Aleksandra, Dharam, Romy, Lisa, Déborah, Romain, Erick and thank you Aurélien, Georgios and again, Julie and Žofia for those years among you, they have been a delight ! I must also thank my numerous flatmates, of course first of all the real greatest, Fabian, for the F&F is now eternal, then Nicolas and Clothilde, it has been too short but it has been perfect, and at last Marine, Victor, Samson and Guillaume from the world of art, long live La Daronette. A final thought for Quentin, François, Vianney and Florant, close friends that were away.

And at last a warm thank to my family, I know how lucky I am to have you.

Contents

List of Figures

List of Tables

Glossary

AILPS	Ab-Initio Local Pseudo-potential
AO	Atomic Orbital
ARPS	Adaptively Restrained Particle Simulations
ASED MO	Atom Superposition and Electron Delocalization Molecular Orbital
BLPS	Bulk-Derived Local Pseudo-potential
BO	Born-Oppenheimer
DFT	Density Functional Theory
EHMO	Extended Hückel Molecular Orbital
FCC	Face-Centered Cubic
FEG	Free Electron Gas
FFT	Fast Fourier Transform
GGA	Generalized Gradient Approximation
GNH	Goodwin-Needs-Heine
GTO	Gaussian-type Orbital
HC KEDF	Huang Carter KEDF
HF	Hartree-Fock
HREDF	Hartree Repulsion Energy Density Fonctional
KEDF	Kinetic Energy Functional
KS-DFT	Kohn-Sham Density Functional Theory
LCAO	Linear Combination of Atomic Orbitals
LDA	Local Density Approximation
LPS	Local Pseudo-potential
LSDA	Local Spin Density Approximation
MEP	Minimum Energy Path
MO	Molecular Orbital
MPI	Message Passing Interface
NEB	Nudge Elastic Band
OB-DFT	Orbital-Bound DFT
OF-DTF	Orbital-Free DFT
PAW	Projected Augmented Wavefunctions
PBC	Periodic Boundary Conditions
PES	Potential Energy Surface
PBE	Perdrew, Burke, Ernzerhof
PS	Pseudo-potential
PZ	Perdrew, Zunger
RS-FD	Real-Space Finite Differences
SCF	Self-Consistent Field
SE	Schrödinger Equation
STO	Slater Type Orbital
TDDFT	Time-Dependent DFT
TDSE	Time-Dependent Schrödinger Equation
TISE	Time-Independent Schrödinger Equation
TF-KEDF	Thomas-Fermi KEDF
vW-KEDF	von-Weizsacker KEDF
WT-KEDF	Wang-Teter KEDF
WGC-KEDF	Wang Govind Carter KEDF
XCEDF	Exchange-Correlation Energy Density Functional

1

Introduction

1.1 Molecular Simulation

With the arrival of computers and the development of computer sciences, chemistry, cosmology, meteorology and several other research domains have found a new tool to understand matter and life. By modeling and simulating "in silico" a system, whether a molecule, a galaxy or a wind flow, one could, with good models, predict its operation, its affinity with environments and its evolution in time without the difficulties of real experiments. Computer sciences have brought a new tool to understand domains of all scales, and in particular microscopic scale. Indeed, as small objects like molecules, proteins or crystals are hardly observable, the ability to model their shapes and simulate their reactions is crucial.

Modeling a molecular system can be done by numerous methods, we will focus on the most usual one: particle simulation. In particle simulation the molecular system is split into particles, often the atoms, the interactions between them are computed and a time evolution of the particle positions is derived from the interactions. Hence molecules and their reactions are seen on computer. The reality is that interactions between particles are often too complex to be calculated precisely and have to be approximated, so "in silico" molecules are only approximations of the real ones. Even approximated, the computation of interactions is the computational bottleneck of particle simulation, hence the research for accurate and efficient interaction models is the core of particle simulation.

The ideal interaction model would use exact forces derived from quantum physics.

Unfortunately the calculation of a molecule's electronic wave function is a complex problem. A family of methods was developed to approximate the result and have interactions computed from quantum physics, the first-principles methods. Density Functional Theory (DFT) is the most popular one but is still very slow compared to more simple pairwise interaction models. This thesis focuses on the development of a DFT method efficient enough to perform dynamical simulations.

We will introduce in this first chapter some principles of particle simulation. First why we can approximate atoms as ball-particles following Newton's law of movements, then how from particle interactions we simulate their trajectories, a rapid look of methods to compute particle interactions. At last we introduce some common methods to speed up simulations.

1.2 Laws of particles movement

1.2.1 Time-dependent Schrödinger equation

The precise simulation of particles – nuclei and electrons – movement starts with quantum mechanics. At a quantum level, a system of N particles is described by a state vector $\Psi(t)$, and the time evolution of this state vector is given by the time-dependent Schrödinger equation :

$$i\hbar\frac{\partial}{\partial t}\left|\Psi(t)\right\rangle = H\left|\Psi(t)\right\rangle \tag{1.1}$$

with $\hbar$ the reduced Planck constant $\hbar = \frac{h}{2\pi}$ and H the observable linked to the system's energy : the Hamiltonian operator. The most common way to represent a particles system state $\Psi(t)$ is with its position space wave function $\Psi(\mathbf{r}, t)$, a complex-valued function of two variables: the time t and the vector of particles positions $\mathbf{r}$. Other representations are possible like the momenta space wave function $\Psi(\mathbf{p}, t)$, $\mathbf{p}$ the vector of particles momenta, but we will keep the position space wave function. $\Psi(\mathbf{r}, t)$ contains all information on the particles system, the density probabilities of positions are given by the position observable $\langle\mathbf{r}|$, in position space it identifies with the identity function : $\rho(\mathbf{r}, t) = \langle\Psi(\mathbf{r}, t)|\mathbf{r}|\Psi(\mathbf{r}, t)\rangle = |\Psi(\mathbf{r}, t)|^2$, the density probabilities of momentum by the momentum observable $\langle\mathbf{p}|$: $\Psi(\mathbf{p}, t) = \langle\mathbf{p}|\Psi\rangle$ and the other features like spin, magnetic moment etc ... by their own observable.

The Schrödinger equation (SE) has proven itself to be impressively precise, being able to compute the shape of atomic orbitals(1), their energy with band separation or the energy of some particles with great precision, here the energy of the helium atom with a 40 digit precision(2). This precision comes with a cost, the Schrödinger equation can be analytically resolved only for the simplest systems, one particle in a simple electric potential. Errors appear if energies of particles are high enough to create relativistic behaviors, the relativistic Schrödinger equation (3) has to be used.

The quantum interactions between two particles make the Schrödinger equation impossible to solve, approximations and computers have to be used. Even with approximations, computations are too time consuming to hope having a large scale simulation, so we use a drastic approximation, heavy particles shall be considered as balls and behave as such, with the classical mechanics.

1.2.2 Classical limit

Before the work of Louis De Broglie on the wave-particle duality, atoms were considered as balls following Newton-like mechanics. The introduction of wave functions and quantum mechanics to describe the movement of particles has been a revolution in the understanding of matter, but has also made the calculations much more complex and so atomic simulations more difficult. Hence the classical limit approximation that aims to explain how and under which conditions classical mechanics can be recovered from quantum mechanics.

In contrary of a ball particle following Newton mechanics, a wave particle has no well defined position neither momentum, but distributed probabilities of positions and momenta, and the forces applied on the particle take into account this dispersion. The classical limit considers a model in which the reduced Planck constant $\hbar$ tends to zero, or at least in which the characteristic actions of the elements of the model are greater than $\hbar$ with several orders of magnitude. In this model the positions and momenta distributions tend to Dirac functions, so their dispersion tends to zero. For a particle in position $\mathbf{q}$:

$$|\Psi(\mathbf{r})|^2 \xrightarrow[\hbar\to 0]{} \delta(\mathbf{r}-\mathbf{q}) \tag{1.2}$$

with δ the Dirac delta function (4). With the positions and momenta being well defined, Erhenfest theorem states that Newtonian mechanics can be applied (5).

But that is the ideal approximation, and the emergence of classical mechanics from Schrödinger equation is still not entirely solved (6). Indeed, even with small $\hbar$, nothing prevents wave particles from spreading with time.

It is nonetheless possible to recover Newton laws of movement without having narrow wave packets with Bohmian method. Let us consider a wave particle defined in the polar form $\Psi = Re^{\frac{i}{\hbar}S}$ and a Hamiltonian expanded: $H = -\frac{\hbar^2}{2m}\nabla^2 + V$. The Schrödinger equation 1.1 is now:

$$i\hbar\frac{\partial \Psi}{\partial t} = -\frac{\hbar^2}{2m}\nabla^2\Psi + V\Psi \tag{1.3}$$

With $\rho = R^2$ the particle density, S the action in the Lagrangian equation(7) and so ∇S the particle momentum, the imaginary part gives the continuity equation:

$$\frac{\partial R^2}{\partial t} + \nabla\left[\frac{\nabla S}{m}R^2\right] = 0 \tag{1.4}$$

and the real part a modified Hamilton-Jacobi equation:

$$\frac{\partial S}{\partial t} + \frac{(\nabla S)^2}{2m} + V - \frac{\hbar^2}{2m}\frac{\nabla^2 R}{R} = 0 \tag{1.5}$$

In which the three first terms are the classical Hamilton-Jacobi equation and the last one is an additional term called the quantum potential $U = -\frac{\hbar^2}{2m}\frac{\nabla^2 R}{R}$. The quantum potential can be seen as a measure of the shift between the classical and quantum mechanics, and as $U \xrightarrow[\hbar\to 0]{} 0$, the wave particle follows the Newtonian mechanics when $\hbar \to 0$. Bohmian method concludes only on the Newton mechanics approximation and not on the dispersion of wave packets.

Actually, the dispersion of positions is not negligible, for example an aluminum atom at room temperature ($T = 300K$) has a characteristic wavelength of roughly $\lambda = 0.3$ Å for an inter-atomic distance of 3 Å. At atomic scale, the ball particle model is not obviously supported by quantum theory. Nonetheless the model is wildly used because of its simplicity and the satisfactory results it can give in practice. First-principles models rely at the same time on a wave function model with quantum mechanics and a ball particle model with Hamiltonian mechanics.

1.3 Particle Dynamics

1.3.1 Hamiltonian mechanics

In the Hamiltonian formalism, particles are characterized by their mass, positions and momenta. Let us take N particles of masses $(m_1, m_2, \ldots, m_N)$, $\mathbf{q} = (\mathbf{q_1}, \ldots, \mathbf{q_N})$ the vector of all their positions and $\mathbf{p} = (\mathbf{p_1}, \ldots, \mathbf{p_N})$ the vector of all their momenta. The total energy of this system is given by its Hamiltonian H, function of $\mathbf{q}$ and $\mathbf{p}$, and the time evolution of the positions and momenta is by Hamilton's equations (8):

$$\begin{cases} \dot{\mathbf{q}}(t) & = \nabla_{\mathbf{p}} H(\mathbf{q}(t), \mathbf{p}(t)) \\ \dot{\mathbf{p}}(t) & = -\nabla_{\mathbf{q}} H(\mathbf{q}(t), \mathbf{p}(t)) \end{cases} \tag{1.6}$$

with $(\mathbf{q}(0), \mathbf{p}(0))$ the initial state.

In classical mechanics, the total energy is the sum of the system's kinetic energy K and potential energy V :

$$H(\mathbf{q}, \mathbf{p}) = K(\mathbf{q}, \mathbf{p}) + V(\mathbf{q}, \mathbf{p}) \tag{1.7}$$

And in its widely-used form, the classical Hamiltonian is separable(9) :

$$H(\mathbf{q}, \mathbf{p}) = K(\mathbf{p}) + V(\mathbf{q}) \tag{1.8}$$

The total kinetic energy is the sum of all particle's kinetic energy :

$$K(\mathbf{p}) = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} \tag{1.9}$$

which is written more formally :

$$K(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T \mathbf{M}^{-1} \mathbf{p} \tag{1.10}$$

with M the diagonal matrix of masses. In the next chapter 1.3.3 we will use another kinetic energy function to modify the movements of particles.

The potential energy $V(\mathbf{q})$ describes the interactions between particles. Depending on the physical model chosen for the simulation, $V(\mathbf{q})$ can take different forms. They vary from a sum of pairwise interactions to more global considerations, we describe the main kinds in chapter 1.4. From the above equations 1.6 and 1.8 and from Newton second law :

$$-\nabla_{\mathbf{q}}V(\mathbf{q}) = -\nabla_{\mathbf{q}}H(\mathbf{q},\mathbf{p}) = \dot{\mathbf{p}}(t) = \mathbf{f} \tag{1.11}$$

with $\mathbf{f} = (\mathbf{f_1}, \ldots, \mathbf{f_N})$ the vector of forces undergone by each particle. Hence the name of force-field for $V(\mathbf{p})$.

With a separable Hamiltonian, and classical kinetic energy, motion equations are simplified and we retrieve Newtonian mechanics:

$$\begin{cases} \dot{\mathbf{q}}(t) = & \nabla_{\mathbf{p}}K(\mathbf{p}(t)) = \mathbf{p}/m \\ \dot{\mathbf{p}}(t) = & -\nabla_{\mathbf{q}}V(\mathbf{q}(t)) = \mathbf{f} \end{cases} \tag{1.12}$$

1.3.2 Discretization of Dynamics

Except for simple motions, the equations of movement cannot be solved analytically, so the time continuum t and the continuous state trajectory $\big(\mathbf{q}(t), \mathbf{p}(t)\big)$ are discretized. Let us note the time step Δt, the continuous state trajectory $\big(\mathbf{q}(t), \mathbf{p}(t)\big)$ becomes a sequence of states $\big(\mathbf{q}^n, \mathbf{p}^n\big)$, approximating the states of $\big(\mathbf{q}(t), \mathbf{p}(t)\big)$ at times $n\Delta t$: $\big(\mathbf{q}^n, \mathbf{p}^n\big) \approx \big(\mathbf{q}(n\Delta t), \mathbf{p}(n\Delta t)\big)$. To create this sequence, we integrate those equations, the simplest integrator is the Euler one:

$$\begin{cases} \mathbf{q}^{n+1} = \nabla_{\mathbf{p}}H(\mathbf{q}^n, \mathbf{p}^n) \\ \mathbf{p}^{n+1} = -\nabla_{\mathbf{q}}H(\mathbf{q}^n, \mathbf{p}^n) \end{cases} \tag{1.13}$$

And one of the most common is the velocity Verlet integrator that has an error of second order with respect to the time step size ∇t:

$$\begin{cases} \mathbf{p}^{n+1/2} = \mathbf{p}^n - \Delta V(\mathbf{q}^n)\dfrac{\Delta t}{2} \\ \mathbf{q}^{n+1} = \mathbf{q}^n + \Delta K(\mathbf{p}^{n+1/2})\Delta t \\ \mathbf{p}^{n+1} = \mathbf{p}^{n+1/2} - \Delta V(\mathbf{q}^{n+1})\dfrac{\Delta t}{2} \end{cases} \tag{1.14}$$

This integrator has a second order error because it modifies the positions with the momenta between two time steps – $\mathbf{p}^{n+1/2}$ –, nevertheless, the integrator still needs only one computation of forces $\Delta V(\mathbf{q})$ per time step. The computation of forces being very often the bottleneck of simulation, this makes the velocity Verlet integrator commonly used. Other integrators of higher order can be found in the literature (10)

1.3.3 Adaptively Restrained Particle Simulation

The NANO-D team has recently developed a new particle dynamics designed to speed up simulations: ARPS for Adaptively Restrained Particles Simulations(11, 12). ARPS freeze the slowest particles movements so that the number of moved particles at each time step decreases. Precisely, ARPS modify the kinetic energy function of the Hamiltonian by a function that vanishes for the small values of momenta. If one refers to the integrators 1.13 and 1.14, or even to the Hamilton's equation of motion 1.6 with a separable Hamiltonian, one observes that if the kinetic energy of a particle is null on an interval, its kinetic energy gradient is null too, and the particle is motionless: $\dot{\mathbf{q}} = 0$. In the ARPS method, the Hamiltonian is modified with a new kinetic energy function:

$$H_{AR}(\mathbf{q}, \mathbf{p}) = K_{AR}(\mathbf{p}) + V(\mathbf{q}) \tag{1.15}$$

with

$$K_{AR}(\mathbf{p}) = \sum_{i=1}^{N} k(\mathbf{p}_i) \tag{1.16}$$

k manages the behavior of particles by using their kinetic energy K and 2 thresholds: ε^r, the fully-restrained threshold, and ε^f, the full-dynamic threshold. If $K \leq \varepsilon^r$ the particle is inactive and its movement is totally restrained, if $K \geq \varepsilon^f$ the particle is active and moves freely. Between those two thresholds, when $\varepsilon^r < K < \varepsilon^f$, a C^2 function, s, permits a smooth transition between the two states (active and inactive) 1.17.

$$k(\mathbf{p}_i) = \begin{cases} 0 & \text{if} \dfrac{\mathbf{p}_i^2}{2m_i} \leq \varepsilon^r \\ \dfrac{\mathbf{p}_i^2}{2m_i} & \text{if} \dfrac{\mathbf{p}_i^2}{2m_i} \geq \varepsilon^f \\ s\left(\dfrac{\mathbf{p}_i^2}{2m_i}\right) & \text{if} \dfrac{\mathbf{p}_i^2}{2m_i} \in [\varepsilon^r, \varepsilon^f] \end{cases} \tag{1.17}$$

For details and mathematical explanations, we refer the reader to (11, 12). The consequences on macroscopic quantities of this restrained adaptive dynamics have been studied in (13, 14).

The goal of freezing the movements of particles is to speed up the computation of $\Delta V(\mathbf{q})$, the bottleneck of particle simulations. If few particles have moved, $\mathbf{q}^{n+1} \approx \mathbf{q}^n$,

$\Delta V(\mathbf{q}^{n+1})$ could be easier to update from $\Delta V(\mathbf{q}^{n})$ than to compute from scratch. The NANO-D team has already modified several classical force fields to benefit from this new dynamical method. The scheme often consists in comparing the differences between two consecutive time steps and in adapting the computation of forces in consequence to avoid useless calculus.

For example, for pairwise interaction models like Lennard-Jones potential, ARPD allows to update fewer forces. For other interaction model, incremental methods, that compare the differences between two consecutive time steps and adapt the computation of forces in consequence, can benefit from a reduction of the number of active particles. The NANO-D team has developed several incremental methods to compute first-principles interactions(15, 16) or electrostatic interactions (17). Other simulation methods that do not update the positions of all particles, like Monte-Carlo simulations(7), also benefit from incremental computation of forces and energies.

1.4 Particle Interactions

Having determined how particles move, we have now to calculate what force makes them move. We have simplified here the interactions between particles in two distinct types: empirical interactions and first-principles interactions. Basically, empirical force-field are simple interaction schemes whose parameterization is based on the observation of matter macro-properties –top-down approach–, and first-principles force-field are based on quantum physics and the resolution, or approximation, of Schrödinger equation –bottom-up approach–. In practice a lot of force-fields, and most of used "first-principles" method, are in between those two types, using quantum physics to define the shape of their interaction functions and macro-properties to tune some of those functions parameters.

1.4.1 Empirical force fields

For empirical force-fields, the strong bindings called covalent bonds between atoms do not emerge from the interaction model, they have to be modeled explicitly. Hence this general kind of potential energy that splits apart the interactions created by bonds from the rest, the rest containing other inter-atomic interactions plus an external potential:

$$V(\mathbf{q}) = V_{bonds}(\mathbf{q}) + V_{IA}(\mathbf{q}) + V_{ext}(\mathbf{q}) \tag{1.18}$$

With more complex particles, other properties could be added, for example the polarization of water particles. We will stay brief for the introduction.

1.4.1.1 Liquids and Gases

For systems without bonds, like simply modeled gases and liquids, and without any external potential, we keep only inter-atomic interactions $V_{IA}(\mathbf{q})$ often being a sum of pairwise interactions that depending only on the distance between the two particles concerned:

$$V_{IA}(\mathbf{q}) = \sum_{1 \leq j \leq i}^{N} \mathrm{v}_{pot}(|\mathbf{q}_i - \mathbf{q}_j|) \tag{1.19}$$

With $\mathrm{v}_{pot}(r)$ a function of the distance between two particles $r = |\mathbf{q}_i - \mathbf{q}_j|$.

When particles are charged, the electric interaction is generally preponderant on all the others, and the interaction between two particles i and j of charges Q_i and Q_j is written

$$\mathrm{v}(r) = \frac{1}{4\pi\varepsilon_0} \frac{Q_i Q_j}{r} \tag{1.20}$$

If particles are not charged, weaker interactions become noticeable. Among them, van der Waals forces gather permanent and induced dipole interactions, they are attractive forces with a shorter characteristic interaction distance. A classical way to represent them simply is the Lennard-Jones potential (18). The LJ potential also adds a very-short range repulsive energy to prevent particles to be too close, an easy way to insert the Pauli exclusion. The potential describes the interaction between two particles $\mathrm{v}_{LJ}(r)$.

$$\mathrm{v}_{LJ}(r) = 4E_0 \left[\left(\frac{r}{d}\right)^{-12} - \left(\frac{r}{d}\right)^{-6} \right] \tag{1.21}$$

In which E_0 and d depend on the particle type, some formulas allow to calculate interactions between two different particles(18).

The time for direct computation of those potentials scale as $O(N^2)$ because of the double sum. The efficient method depends on the potential, for short range potentials, like Lennard-Jones, cell lists to store a particle neighbor are common. For the electrical force-field, the interactions are often solved by computing the electric potential on the whole domain by solving the associated Poisson equation. We will see that we can reduce again the computation times in dynamical simulations by using restrained dynamics.

1.4.1.2 Bonded systems

For systems with covalent bonds, present in domains like biology, crystallography, organic chemistry etc ..., covalent bonds have a capital contribution to energy, hence the very common "balls and sticks" molecular representations. To model non-reactive systems, in which no covalent bond breaks or appears, an easy way to model bindings is to associate to bonds an elastic energy. Lengths, angles and torsions are maintained around equilibrium values, l_0, θ_0 and ϕ_0, by an elastic potential. If $\mathbf{q}_{ij} = \mathbf{q}_i - \mathbf{q}_j$ is the vector from atom i to atom j and i, j, k and l are four consecutive and linked atoms:

$$V_l = k_l \left(|\mathbf{q}_{ij}| - l_0\right)^2 \tag{1.22}$$

$$V_a = k_a \left(\widehat{\mathbf{q}_{ji}, \mathbf{q}_{jk}} - \theta_0\right)^2 \tag{1.23}$$

$$V_t = k_t \left(\phi_{\mathbf{q}_{ji},\mathbf{q}_{kl}} - \phi_0\right)^2 \tag{1.24}$$

$\phi_{\mathbf{q}_{ji},\mathbf{q}_{kl}}$ the torsion angle between $\mathbf{q}_{ji}$ and $\mathbf{q}_{kl}$. For example, in crystallography, the Keating model (19) associates a crystal to the energy

$$V(\mathbf{q}) = \alpha \sum_{bonds} \left(\mathbf{q}_{ij}^2 - l_0^2\right)^2 + \beta \sum_{bond\ angles} \left(\mathbf{q}_{ij}.\mathbf{q}_{ik} + \frac{1}{3} l_0^2\right)^2 \tag{1.25}$$

A more general version can be found here (20).

1.4.1.3 Reactive force fields

But chemical reactions are mainly about creations and breaks of bonds and models and simulation have to deal with those. The Morse potential models elastic bonds that can break:

$$V(\mathbf{q}_{ij}) = D_e \left(1 - e^{-a(|\mathbf{q}_{ij}| - l_0)}\right)^2 \tag{1.26}$$

In which D_e is the binding energy and l_0 the equilibrium bond length. The potential shape is similar to the Lennard-Jones potential shape. The important differences are the binding energy involved (D_e) and the quadratic shape of the Morse potential around the equilibrium length that gives the energy functional the desired spring behavior.

1.4.1.4 Universal force-field

The Universal Force-Field, or UFF, gathered all those interactions. In the original paper(21), the systems energy is written

$$E = E_R + E_\theta + E_\phi + E_\omega + E_{vdW} + E_{el} \tag{1.27}$$

with

- E_R the bond stretch energy, a Morse potential.
- E_θ the bond angle energy.
- E_ϕ the bond torsion energy.
- E_ω the inversion term accounting for planar angles.
- E_{vdW} the van der Waals interactions energy, a Lennard-Jones potential.
- E_{el} the electrostatic interactions energy, a Coulomb repulsion term.

The formulas of those energy functionals may be different for computational efficiency. All those energies depend on the atoms involved, and UFF rests on long tables of atom-dependent coefficients, and not only the different elements of the periodic table, each element is sorted in several types. For example, in the 1992 version of UFF, carbon has four types, $\mathrm{C_3}$ for tetrahedral, $\mathrm{C_2}$ and $\mathrm{C_R}$ for triangular, $\mathrm{C_1}$ for planar conformations. All the coefficients are tuned to fit sets of molecules.

One also can have more accurate force-fields that are specialized in one kind of system, for example:

- GROMOS(22)(for Groningen Molecular Simulation), implemented in GROMACS(23), is specialized for protein and their ligands.

- Brenner force-field (24) is specialized in the simulation of hydrocarbon molecules, graphene, graphite and diamond.

The parameters of all those force-fields are tuned to fit experiments or theoretical results.

1.4.1.5 Learned Force-Field

A new kind of force-fields has emerged with the increase of computational power. Those force-fields parameters are learned automatically from sets of molecule examples and not fitted from experiments. Learning abilities have been enhanced with the new advances in machine learning and the increase in computational power. With neural networks, that method creates a pure empirical force-field, a black box whose coefficients are not understood, they have been learned (25, 26). As deep-learning methods require a lot of examples to learn to model a property, classical database of molecules are not sufficient, so the methods often use already existing and precise force-fields to create large datasets of examples on which to learn, the first-principles force-fields, slow and accurate, are the standard choice.

1.4.2 First-Principles Methods

The aim of "first-principles" methods is to simulate the physic of particles behind chemistry: the behavior of the electrons and nuclei that creates and breaks the bonds, shape the molecules and make them interact. Empirical methods are in general simple, and so fast and able to model large systems, but because they are tuned on existing set of molecules they might not be able to predict behaviors that are unknown, rare or just not "obvious". The angle of the ammonia molecule NH_3 is $109,5°$, see figure 1.1, whereas a spring model trying to maximize angles would have given a flat molecule with 120° angles. Empirical models do not predict those shapes, they have to be hard coded in the model, as they are in UFF (21). First-principles models, by starting from the physic behind those shapes, should be able to model everything, and predict unknown molecules shapes.

We will spend more time to develop how a first-principles force-field is built in the next chapter, when explaining the equations of Orbital-Free Density Functional Theory 2. Briefly, they use quantum physics to model chemical properties. The Schrödinger

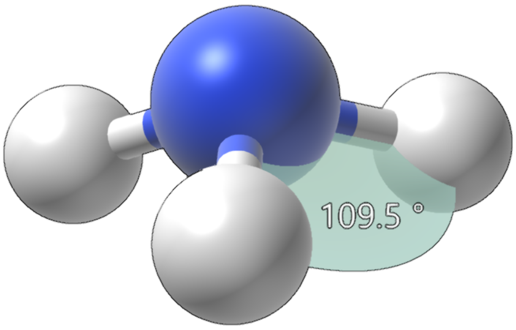

Figure 1.1: Ammonia molecule, NH_3 - For the NH_3 molecule, a simple spring model would have given a flat triangular molecule, with 120° bond angles. To model properly the ammonia molecule with an empirical force field, the equilibrium bond has to be hard coded in the force-field. Another option is a first-principles force-field that can model the reason for this angle: the non-binding electrons pair of nitrogen.

equation 1.1 that models how particles move in an electric field, and in particular how electrons move in the electric field created by nuclei is solved, or approximately solved. That allows to find where electrons rest to minimize a nuclei system's energy, and then deduce the shapes and dynamics of molecules. And to model the interaction between two atoms, the Schrödinger equation is solved for electrons in the electric potential created by the two nuclei. In figure 1.2, we drew the dihydrogen bond modeled with one first-principles method, the linear combination of atomic orbitals (LCAO). For first-principles methods, the energies and shape of molecules are determined ideally on the sole quantum physic.

But this perfect simulation method comes with a great computational cost. The calculus of a bond energy went from a simple function 1.26 to the resolution of the Schrödinger equation, an eigenvalue problem. That is why first-principles methods have kept focus on small systems. Moreover, SE is a complex equation to solve, mainly because the interaction between two electrons cannot be computed precisely in the general case. The equation can be solved analytically for systems with one electron because the interaction between electrons disappears, for example the hydrogen atomic orbitals 2.1 and their energy are solved analytically. For the electrons of the helium atom, computational methods are required (2). For multi-atomic systems, more approximations have to be used. A first one is to consider the solutions of the Schrödinger

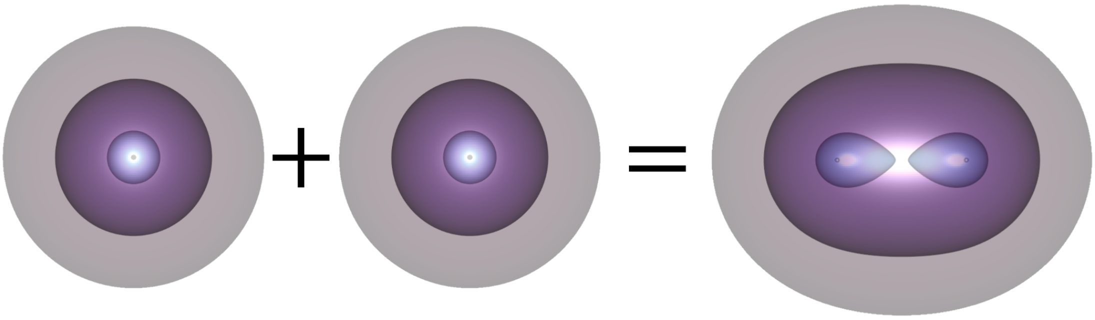

Figure 1.2: Electron density of the σ-bond of dihydrogen - To create a bond, two atoms share electrons. Here we have used a linear combination of atomic orbitals (LCAO) to model the binding molecular orbital: the $1s$-orbitals of each hydrogen, $|\psi_a\rangle$ and $|\psi_b\rangle$, merge into a σ-bond orbital $|\psi_\sigma\rangle = \alpha_a |\psi_a\rangle + \alpha_b |\psi_b\rangle$.

equation, or the molecular orbitals, are linear combinations of atomic orbitals (LCAO). LCAO are good approximations of molecular orbitals and are used in Hartree-Fock (HF) methods and even DFT as "Slater-type Orbitals" (STO), but still reduces the search space of solutions, and so the precision. Finally, as the reduction of molecular orbitals search-space with LCAO does not permit to model moderately large systems, the next approximation is more drastic: the Density Functional Theory (DFT) does not try to compute eigenvectors of SE, but instead computes directly the electron density corresponding to the useful eigenvectors. DFT reduces a problem of dimension $3N$, N the number of electrons in the system, to a problem of dimension 3, the electronic density function.

As this work is centered on computations for a DFT method, we will develop more explicitly first-principles methods in the next chapter.

1.5 Efficient molecular simulations

Many macroscopic properties of materials can only be explained with simulations involving a lot of particles, several thousands to billions. For example classical proteins contain thousands of atoms, and as their behavior depends on the solvent in which it evolves, often water, the solvent has to be simulated too (23). A living or pseudo-living organism like a virus contains at least hundreds of proteins. The size of simulation required to model those macroscopic objects become too much for a computer or even a super calculator to manage. To improve the speed of calculations, new algorithms have emerged. They often search for ways to reduce the number of dimensions to explore.

1.5.1 Periodic Boundary Conditions

To multiply artificially the number of particles simulated, the use of Periodic Boundary Conditions (PBC) is common. In PBC the computational domain is folded on itself: particles close to a border feel the interactions with particles on the other side through the border, see figure 1.3. PBC permits to model virtually infinite systems with a finite number of particles, besides some efficient function basis like plane-waves are periodic so PBC is well adapted for them.

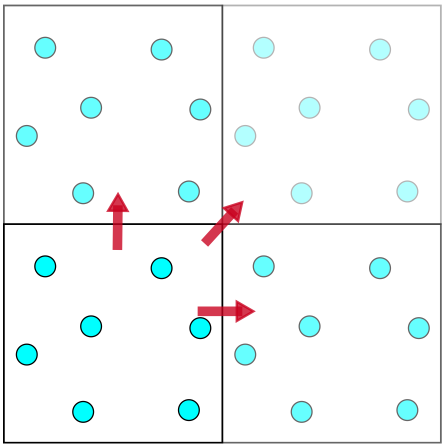

Figure 1.3: Periodic Boundary Conditions simulation - In Periodic Boundary Conditions (PBC), the computational domain is copied an infinity of times in every directions, so that with only a finite number of simulated particles, an infinite particle system is modeled.

1.5.2 Coarse-grained models

In a more general way, reducing the computational cost of algorithms often starts by identifying the dimensions of the search space the less useful and find representations of the system that do not include them, hence with less dimensions. For example in computational biology, atoms can be grouped by molecules or by molecule parts that become rigid objects. The coarse grains model restrains movement of particles and reduces the system's dimensions. For proteins, an all-atom model can be turned into a amino-acid model or even a chain model, see 1.4.

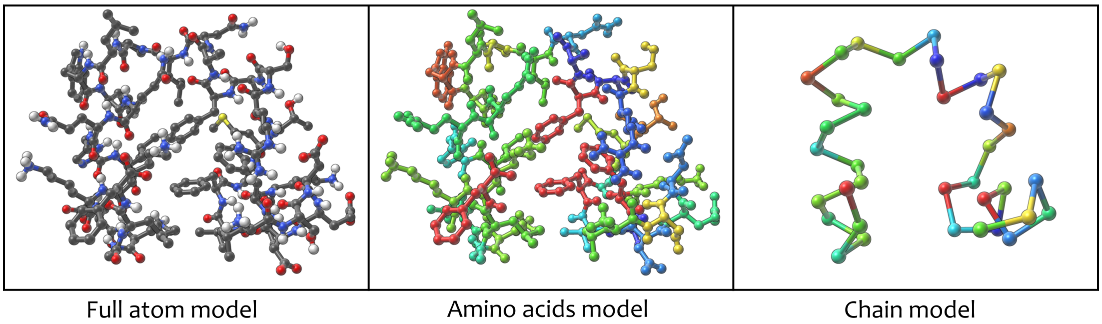

Figure 1.4: From all atom to chain model simulation - To reduce computation time of a protein simulation, the full protein is considered as a chain whose links are for example the alpha carbons. The interesting properties of amino acids like mass, hydrophobia, hydrogen binding abilities etc . . . are still represented, but are now properties of the chain links.

1.5.3 Restrained Dynamics and incremental algorithms

Adaptively Restrained Dynamics Simulations are in the same spirit, they aim at reducing the number of dimensions a dynamical model can explore at each time step. The movements of some particles are stopped, the space they explore at each time space is reduced. But to benefit from the reduction of movements, the interaction model of the simulation has to be adapted. We describe below several interaction models that do not compute the energy and forces from scratch at each time step, but that try to use a maximum of information from the precedent time step. We classified them here as ”incremental algorithms”.

1.5.3.1 Short-range interactions

For short-range interactions like Lennard-Jones or Morse potentials, the first classical improvement for an efficient simulation consists in computing interactions only between neighbor particles, other interactions being negligible. The neighbor grid algorithm, in which the domain is paved in cubic sub-domain with a side size corresponding to the length of the interaction limit in order to find easily the neighbor particles, is a famous method to manage the which interactions are to compute. Other and more specific methods to store or compute neighbor lists exist (27), the grid-based stays a very classic one.

In (11, 28) is shown that a restrained dynamic can increase again the speed of computation without losing simulation's features. As the interactions are pairwise and depends only on the inter-atomic distance $|\mathbf{q}_i - \mathbf{q}_j|$, if a pair of atoms does not move, the interaction between those two atoms stays the same at the next time step and does not need to be computed again. Freezing some particles movement decreases the number of forces to compute and reduce the quantity of computation required. More complex methods allows massively parallel simulations with ARPS (29, 30, 31).

1.5.3.2 Long-range interactions

Because of their long-range characteristic, in a charged particle system, all particles feel the interactions of all particles, so reducing the computation of forces to neighbors does not make sense. For efficiency reasons, the energy and the interactions of a long-range interactions model are solved through the corresponding Poisson equation. If we call ϕ the potential created by all charges, we have for punctual charges:

$$\phi(\mathbf{q}, x) = \frac{1}{4\pi\epsilon_0} \sum_{1 \leq j \leq N} \frac{Q_j}{|x - \mathbf{q}_j|} \tag{1.28}$$

that can be computed with its corresponding Poisson equation:

$$\mathit{\Delta}\phi(\mathbf{q}, x) = -\frac{1}{\epsilon_0} \sum_{1 \leq j \leq N} Q_j \delta_{\mathbf{q}_j}(x) \tag{1.29}$$

$\delta_{\mathbf{q}}(x)$ the Dirac distribution in $\mathbf{q}$. The total electrostatic energy can be written from 1.19

$$E(\mathbf{q}) = \frac{1}{2} \sum_{1 \leq i \leq N} Q_i \delta_{\mathbf{q}_i}(x) \phi(\mathbf{q}, x) \tag{1.30}$$

The Fourier transform allows, with the Fast Fourier Transform, to compute ϕ in an $\mathcal{O}(N \ln(N))$, N a measure of the system linked to the number of particles, and not in $\mathcal{O}(N^2)$. Multi-grid algorithms have the same scaling. In (17, 32), multi-grid algorithms have been modified to benefit from a restrained simulation.

1.5.3.3 First-Principle interactions

There is still no first-principles interaction model that has been adapted to benefit from a restrained dynamic. There is however a version of Atom Superposition and Electron Delocalization Molecular Orbital (ASED-MO) theory, that uses small LCAO functional basis and a simplified Hamiltonian, on the model of Extended Hückel Molecular Orbital (EHMO) has been developed to improve geometry relaxation by freezing particles movement. By dividing the molecular system in disjoint blocks, this version of ASED-MO updates only parts the electronic structure at each geometry optimization step(15, 16, 33).

Our work is in the continuation of those incremental algorithms that are modifications of classical interaction models to benefit from a restrained dynamic. We use this time a first-principles force-field: Orbital-Free Density Functional Theory.

1.6 SAMSON

In the wake of computational chemistry, hundreds of softwares and programs have emerged to design, model and simulate molecules. Most of them are specialized, either in crystallography, protein design, particle simulation or another domain. Yet we see nano-systems that cross those different domains, like a glucose detector made with the combination of proteins and a graphene sheet(34). From this finding and from the need for more efficient simulation methods, the NANO-D team of INRIA developed SAMSON, for Software for Adaptive Modelisation and Simulation of Nano System, a new software designed as an app-store: a simple, mandatory core, and tens of modules, called elements, all specialized in their own task (visualization, computation of interactions, computation of dynamics or all other properties), so that every type of molecular simulation could be made (protein design, crystallography, chemical reaction prediction, molecular dynamics, structural relaxation etc . . .). SAMSON also comes with a graphical interface and a software development kit to implement new elements. For those reasons and to benefit from the dynamical models already implemented (particle dynamics, energy relaxation etc . . .), the methods presented in this thesis have been implemented and tested in this software, and all the pictures are taken from it.

In order to be able to benefit from the work of other developers, molecular simulations are split in two steps

- Dynamic models that control particles movements
- Interaction models that compute the system's energy and the forces atoms undergo

So that a user can use a specific interaction models with dynamical models already existing or vice versa, and a developer can test its new dynamical model (respectively interaction model) with already existing interaction models (respectively dynamical model).

1.7 Contributions

The aim of this thesis is the development of algorithms that fasten the dynamical simulation of particle systems with a first-principles interaction model, Orbital-Free DFT. We chose to adapt OF-DFT computation methods in order to benefit from a restrained dynamical method like ARPS. That led to the contributions described below and in this manuscript.

- The development of a new interaction model has required the development of numerous tools to test the ideas and compare them to state of the art. Several of those tools were efficient enough to also serve other purposes. Indeed, all the ideas developed in this thesis have been tested with the help of SAMSON 1.6. The software is equipped with everything to model particle systems, several dynamical models to minimize energies or run dynamical simulations, the restrained dynamical models we intends to benefit from, proper visualization tools, and a SDK, allowing an easy implementation of new models. Nonetheless, SAMSON is still not equipped like a complete molecular simulation software and misses functionality we needed. So to create benchmarks easily, to display properties clearly or to measure them on the fly, we have developed several SAMSON elements.
  - An element to generate and manipulate crystals. The goal was at first only a way to generate easily Face-Centered Cubic (FCC) aluminum crystals, the main benchmark we have used. We have extended it first to be able to write and generate any crystal, then to read .cif format files. cif, for Crystallographic Information Files, is one classical format of crystal systems. The element is now on sale and has found clients.

  - Several elements in cooperation with our INRIA team, NANO-D, and a CEA research group for the study of defects in a graphene sheet that led to two publications (35, 36). Among them elements to evaluate and draw Radial Distribution Functions and Angle Distribution Functions during simulations and an element for the visualization of deformation in covalent crystal.
  - Other elements to help me testing algorithms and for the communication of my results. I have implemented a Restrained Energy Minimization scheme, a manual modification of particle speed or momenta, and a tool for the visualization of particle position shift, movement or velocity.

- The aim of this thesis is to link a first-principles method and adaptively restrained dynamics. To do so we had to develop another implementation of Orbital-Free Density Functional Theory that could benefit from a restrained dynamic. In Chapter 2 we recall the theory behind OF-DFT. We highlight the advantages of the method, the drawbacks still present and the efforts to address them. We then list several OF-DFT softwares we took inspiration from.

- We produce in Chapter 3 our new program of energy computation based on OF-DFT. We explain our choice for a Real-Space Finite-Differences method, our choice for the optimization algorithm then our choice for the pseudo-potential. At last, after evaluating the accuracy of our method by comparing the energy value computed with another implementation, plane-wave based, PROFESS. We then measure and compare the efficiency of our program with PROFESS.

- The Chapter 4 has been mostly published in a paper in Journal of Computational Chemistry. We make the link between the Adaptively Restrained Particle Dynamic and our OF-DFT method. To show the performance of the combination, we simulated a implantation of aluminum in a two thousands Aluminum atom plate arranged in FCC crystal, then use a saddle search algorithm, Nudge Elastic Band or NEB, to simulate a displacement of a defect in an Aluminum FCC crystal.

2

Orbital-Free Density Functional Theory

Density Functional Theory is a method to solve efficiently Schrödinger equation and compute particles systems properties with quantum mechanics. As DFT has a good trade-off between predictions and computational costs, the method has been widely used and a lot of variants have been developed using different basis sets: plane wave, Gaussian Type Orbitals, Slater Types Orbital, or others like PAW. Orbital-Free Density Functional Theory is one variant of DFT among others with a very low computational cost, hence the ability to simulate large clusters of atoms.

We describe in this chapter how Orbital-Free DFT emerges from the basics of quantum physic. We finish by a list of several methods and implementations of OF-DFT that were crucial for my work.

2.1 Schrödinger equation

We saw in section 1.2.1 that first-principles simulations start with the non relativistic time-dependent Schrödinger equation, here written in position space:

$$i\hbar\frac{\partial}{\partial t}\left|\Psi(\mathbf{r},t)\right\rangle = H\left|\Psi(\mathbf{r},t)\right\rangle \tag{2.1}$$

We consider here and for all the equations afterward only spinless particles because all the equations we need are the same. The Hamiltonian operator H is, like in classical

Hamiltonian mechanics, a measure of energy, and can be decomposed in a kinetic energy operator and a potential energy one :

$$H\left|\Psi(\mathbf{r},t)\right\rangle = -\frac{\hbar^2}{2m}\nabla^2\left|\Psi(\mathbf{r},t)\right\rangle + V(\mathbf{r},t)\left|\Psi(\mathbf{r},t)\right\rangle \tag{2.2}$$

As all the operators are linear, all linear combinations of solutions are also a solution. Thus we look at a basis of solutions, the eigenvectors of the operator H, or the solutions of the time-independent Schrödinger equation:

$$H\left|\Psi(\mathbf{r})\right\rangle = E\left|\Psi(\mathbf{r})\right\rangle \tag{2.3}$$

Here, $\mathbf{r}$ is not only a position vector, it also takes into account the spin of particles: $\mathbf{r} = \{\mathbf{r}_i\}$ with $\mathbf{r}_i = (x_i, y_i, z_i, s_i)$.

In simple cases, this equation can be solved. For one particle in a flat potential $V(\mathbf{q},t) = cste$, $\mathbf{q}$ a position in the 3D space, the solutions are traveling waves. For one particle in a Coulomb Potential $V(\mathbf{q},t) = \frac{Q}{|\mathbf{q}|}$, the eigenfunctions are the so called "hydrogen-like atomic orbitals", a combination of a radial function $R_{nl}(\mathbf{r})$ and spherical harmonics $Y_{l,m}(\theta,\varphi)$. They are, with the approximation of a fixed nucleus, the orbitals of the hydrogen atom. In this case the solutions are analytically solvable, we drew one in figure 2.1, the $3p$ hydrogen orbital.

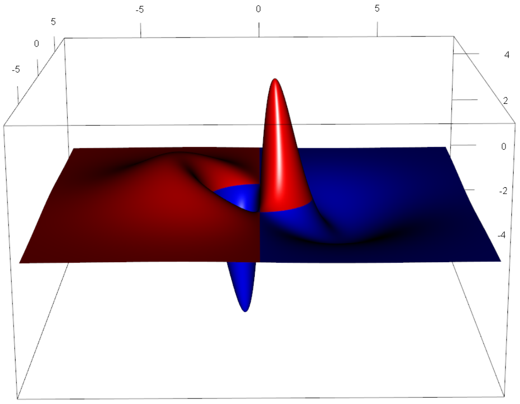

Figure 2.1: Real part of a $3p$ atomic orbital - Solving the Schrödinger equation with one electron in a Coulomb potential $V(\mathbf{r}) = \frac{1}{4\pi\varepsilon_0}\frac{Q}{\mathbf{r}}$ gives a set of wave functions of increasing energies.

Those solutions are more than toy models for they are a start to function basis of several DFT methods.

2.2 Born-Oppenheimer dynamics

To find more solutions, we will need several approximations. Let us go back to the time-dependent Schrödinger equation 2.1. We first split the particles depending on their mass: the nuclei on one side, electrons on the other.

We will now consider a particle system composed of N electrons, of mass m_e, charge e and positions $\mathbf{r} = \{\mathbf{r_i}\}_i$ and of M nuclei, of masses M_I, charges $Q_I e$ and positions $\mathbf{R} = \{\mathbf{R_I}\}_I$

$$H\Psi(\mathbf{r}, \mathbf{R}, t) = E\Psi(\mathbf{r}, \mathbf{R}, t) \tag{2.4}$$

In the international System of Unit, SI, and without any external electrostatic field, the Hamiltonian can be written:

$$H = -\sum_I \frac{\hbar^2}{2M_I}\nabla_I^2 - \sum_i \frac{\hbar^2}{2m_e}\nabla_i^2 + \frac{1}{4\pi\varepsilon_0}\sum_{i<j}\frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} - \frac{1}{4\pi\varepsilon_0}\sum_{I,i}\frac{e^2 Z_I}{|\mathbf{R}_I - \mathbf{r}_i|} + \frac{1}{4\pi\varepsilon_0}\sum_{I<J}\frac{e^2 Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} \tag{2.5}$$

with ε_0 the vacuum permittivity and the five terms being respectively :

1. the kinetic energy of the nuclei
2. the kinetic energy of the electrons
3. the electric interaction between electrons
4. the electric interaction between nuclei and electrons
5. the electric interaction between nuclei

And from here we simplify the computation of particles movement by decoupling the interactions undergone by electrons and nuclei.

- First we introduce the electronic Hamiltonian H_e, a clamped version of the total Hamiltonian:

$$H_e = -\sum_i \frac{h^2}{2m_e}\nabla_i^2 + \frac{1}{4\pi\varepsilon_0}\sum_{i<j}\frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} - \frac{1}{4\pi\varepsilon_0}\sum_{I,i}\frac{e^2 Z_I}{|\mathbf{R}_I - \mathbf{r}_i|} + \frac{1}{4\pi\varepsilon_0}\sum_{I<J}\frac{e^2 Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} \tag{2.6}$$

For each fixed nuclei positions, we can compute with H_e an electrons energy, electronic wave functions and electron movements.
The decoupling of nuclei and electron movement: due to the difference of mass between electrons and nuclei, we dissociate their movements, $\Psi(\mathbf{r}, \mathbf{R}, t) = \chi(\mathbf{R}, t)\psi(\mathbf{r}, t)$, $\chi(\mathbf{R}$ and $t)\psi(\mathbf{r}, t)$ respectively the nuclei wave function and electrons wave function. If nuclei are fixed, electrons move on their own, their movement dictated by the electronic Hamiltonian i the field of the nuclei:

$$\Psi(\mathbf{r}, \mathbf{R}, t) = \sum_{l=0}^{\infty} \chi_l(\mathbf{R}, t)\psi_l(\mathbf{r}, t) \tag{2.7}$$

- Then we use the adiabatic approximation(37): we consider nuclei move without interfering with the electrons quantum state. They are slow enough to let the electronic wave function relax without changing state. That means we can decouple the total wave function $\Psi(\mathbf{r}, \mathbf{R}, t)$ into the product of a nuclei wave function $\chi(\mathbf{R}, t)$ and an electron wave function $\psi(\mathbf{r}, t)$.

$$\Psi(\mathbf{r}, \mathbf{R}, t) = \chi(\mathbf{R}, t)\psi(\mathbf{r}, t) \tag{2.8}$$

- At last, we introduce the classical limit approximation(38) for the nuclei: we consider the positions and momentum wave functions of nuclei are close enough from Dirac distributions they can be seen as a ball-particles. Their position probability density can be written as a sum of Dirac distributions centered on $\{\mathbf{R}_I\}_{1..N}$, the nuclei positions:

$$\int \chi^*(\mathbf{R})\chi(\mathbf{R})d\mathbf{R} = \sum_{0}^{N} Q_I \delta_{\mathbf{R}_I} \tag{2.9}$$

  and hence can be considered as a set of point-positions $\{\mathbf{R}_I\}_{1..N}$. We have already shown 1.2.2 how heavy particles can be seen as ball-particles moving according to Newton's laws of mechanics.

With those approximations we manipulate systems composed of electrons wave functions that moves according to H_e and ball-particles nuclei that that moves according to classical mechanics. Several methods were proposed to mix the two dynamics.

Ehrenfest dynamics (38) propose to compute the movement of electrons with the time-dependent Schrödinger equation 2.1 and nuclei movement with the current electrons wave function:

$$\begin{cases} i\hbar\dfrac{\partial\psi}{\partial t} = H_e\psi \\ M_I\ddot{\mathbf{R}}_I = -\nabla_I \left\langle \varPsi \middle| H_e \middle| \varPsi \right\rangle \end{cases} \tag{2.10}$$

But this scheme requires scaling the time steps on the electrons movements and hence makes the nuclei movement computations slow. Car-Parrinello dynamics (38, 39) avoid the problem by using a modified version of Lagrangian mechanics. We will use Born-Oppenheimer (BO) dynamic, in which the time-independent Schrödinger equation (TISE) is solved at each nuclei time step. If we use the electrons ground-state $\varPsi_0$ for dynamic, BO dynamic is:

$$\begin{cases} E_0\varPsi_0 = H_e\varPsi_0 \\ M_I\ddot{\mathbf{R}}_I = -\nabla_I \left\langle \varPsi_0 \middle| H_e \middle| \varPsi_0 \right\rangle \end{cases} \tag{2.11}$$

If the ground-state is used more often, others electrons states can be used. This scheme allows the dynamic to use larger time steps, scaled on nuclei movement, but it requires efficient method to compute the solution of TISE. Once again, we refer to the book (38) for a complete description and explanation of the different approximations and dynamics we have cited.

2.3 Atomic units system

We open a small parenthesis to explain the unit system we will use for the rest of this work. When working with atomic physic, the International Unit System (SI) is not adapted to the orders of magnitude. Hence we have worked in Hartree atomic units. This system is based on those four physical constant:

- Electron mass m_e
- Electron charge e
- Coulomb constant $k_e = \frac{1}{4\pi\varepsilon_0}$
- Reduced Plank constant $\hbar = \frac{h}{2\pi}$

whose values are by definition one. The energy is measured in Hartree E_h and the distances in Bohr a_0:

- $1E_h = 27.2114$ eV
- $1a_0 = 0.529$ Å

Hence, in Hartree atomic units, the kinetic energy and electrical repulsion energy observables change:

$$-\frac{\hbar^2}{2m_e}\nabla^2 \quad \text{becomes} \quad -\frac{1}{2}\nabla^2 \tag{2.12}$$

$$\frac{e^2}{4\pi\varepsilon_0 r^2} \quad \text{becomes} \quad \frac{1}{r^2} \tag{2.13}$$

And the electronic Hamiltonian 2.6 becomes:

$$H_e = -\frac{1}{2}\sum_i \nabla_i^2 + \sum_{i<j}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} - \sum_{I,i}\frac{Z_I}{|\mathbf{R}_I - \mathbf{r}_i|} + \sum_{I<J}\frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} \tag{2.14}$$

From now, when writing about electrons energy and distances, we will use mostly Hartree atomic units, the energies will be in Hartree (E_h) and the distances in Bohr (a_e).

2.4 Hohenberg-Kohn theorems

All the Density Functional Theory is based on the Hohenberg-Kohn theorems(40). Those theorems transforms the problem of several interacting electrons into a problem of one electron density, reducing drastically the complexity, from a $3N$ dimensions to a 3 dimensions problem. The first theorem is a uniqueness property, the second a variational one.

First H-K theorem The first Hohenberg-Kohn theorem states that if N interacting electrons move in an external potential $V_{ext}(\mathbf{r})$, the ground-state energy is a unique functional of the density $\rho(\mathbf{r})$. Thus the ground state electron density is sufficient to construct the full Hamilton operator and hence to calculate - in principle - any ground state property of the system without the knowledge of the many-electrons wave function. This means that any ground state property can be expressed in terms of the ground state electron density $\rho(\mathbf{r})$. There is an energy functional E such as for the ground-state wave function Φ_0 and its corresponding electronic density ρ_0:

$$E[\rho_0] = \langle \Phi_0 | H_e | \Phi_0 \rangle \tag{2.15}$$

Second H-K theorem The ground state energy can be obtained by energy minimization: for an equal number of electrons, the functional E that delivers the ground state energy of the system, gives the lowest energy if and only if the input density is the true ground state density.

$$\forall \rho, E[\rho_0] \leq E[\rho] \tag{2.16}$$

Thus the challenge is no more to find a whole many-body wave function, but a one dimension function: the electron density. The whole Density Functional Theory rests on those two theorems.

That makes the Hohenberg-Kohn theorems crucial to electronic structure calculation, but does not solve the problem entirely .

2.5 Kohn-Sham DFT

Walter Kohn and Lu Jeu Sham have introduced a method to compute the ground state electron density based on the Hohenberg-Kohn theorems (41). To do so, they do not consider that the ground state density is the result of a N-electrons wave function, but of N one-electron wave functions that do not interact directly with each other.

2.5.1 Kohn-Sham energy functionals

The Kohn-Sham energy functional separates the computations into four energy functionals:

$$E_{KS}[\rho, \mathbf{R}] = E_{ext}[\rho, \mathbf{R}] + E_H[\rho] + E_{XC}[\rho] + T_s[\rho] \tag{2.17}$$

with ρ the electron density : a functional of the 3D space ($\rho(\mathbf{r})$ with $\mathbf{r}$ a 3D position) always positive and accounting for the probability position of N electrons :

$$\rho(\mathbf{r}) \geq 0 \tag{2.18}$$

$$\int \rho(\mathbf{r})d\mathbf{r} = N \tag{2.19}$$

The Kohn Sham functional replaces the Hamiltonian operator and the Hohenberg-Kohn theorems implies if $E_{KS}[\rho_0, \mathbf{R}] = \langle \Psi_0 | H | \Psi_0 \rangle$ for the electron ground state, ρ_0 is the electron density that minimizes $E_{KS}[\rho, \mathbf{R}]$.

2.5.1.1 Electrostatic Energy

The first energy term driving the electron density is its interaction with the external electrostatic potential V_{ext}. In most application, V_{ext} is only created by nuclei:

$$E_{ext}[\rho] = -\int V_{ext}(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} \tag{2.20}$$

where $V_{ext}(\mathbf{r})$ is the potential created by all nuclei indexed by $I \in [\![1, M]\!]$, at positions $\mathbf{R}_I$.

$$V_{ext}(\mathbf{r}) = -\sum_I \frac{Z_I}{|\mathbf{R}_I - \mathbf{r}|} \tag{2.21}$$

2.5.1.2 Hartree Electronic Repulsion Energy

Another term is the electronic interaction which pushes away the electrons from one another. In KS-DFT, a simple Coulomb repulsion is used, since all the quantum effects of electron-electron interactions are hidden in the Exchange-Correlation term. Hence, the electronic interaction $E_H[\rho]$ is defined such that:

$$E_H[\rho] = \frac{1}{2} \int \int \frac{\rho(\mathbf{r}')\rho(\mathbf{r})}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' d\mathbf{r} \tag{2.22}$$

Let us introduce the Hartree Potential V_H, representing the electric potential created by electrons:

$$V_H(\mathbf{r}) = \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' \tag{2.23}$$

Then, the Hartree Energy term can be rewritten as the interaction energy between the electrons and the electric potential they create:

$$E_H[\rho] = \frac{1}{2} \int V_H(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} \tag{2.24}$$

2.5.1.3 Kinetic Energy Density Functional

In KS-DFT, the exact kinetic energy of a N-electrons wave function is approximated by the exact kinetic energy of N one-electron non-interacting wave functions, plus a correction calculated in the Exchange-Correlation Energy. The numerical value of the non-interacting kinetic energy is not calculated directly from the density itself with $T_s[\rho]$, but by introducing a set of N one-electron wave functions orbitals :

$$T_s[\rho] = T_{KS}\left[\{\phi_i\}\right] = \sum_i^{occ} \left\langle \phi_i \left| -\frac{1}{2}\nabla^2 \right| \phi_i \right\rangle \tag{2.25}$$

The set of orthogonal orbitals $\{\phi_i\}_i$ is required to respect the electron density: Single-Slater determinant :

$$\rho(\mathbf{r}) = \sum_i \phi_i^*(\mathbf{r})\phi_i(\mathbf{r}) = \sum_i |\phi_i(\mathbf{r})|^2 \tag{2.26}$$

2.5.1.4 Exchange-Correlation Energy

The exchange correlation (XC) energy is the term that holds all the quantum effects not described in the other functionals. E_{XC} can be formally defined by:

$$E_{XC}[\rho] = T[\rho] - T_s[\rho] + E_{ee}[\rho] - E_H[\rho] \tag{2.27}$$

where $T[\rho]$ is the exact electronic kinetic energy and E_{ee} the exact electron-electron interaction energy. Unfortunately, the exact exchange correlation energy form is unknown, so we have to rely on approximations. In KS-DFT, the XC energy is the gathers all the approximations of the electron energy functional, if E_{XC} is exact, $E[\rho]$ is exact and the errors only would come from the basis functions used. That is the reason why a lot of research has been done on the development of accurate XC energy functional. A whole C++ library is dedicated to the exchange-correlation energy functionals, Libxc (42).

The more classical approximation for KS-DFT and so OF-DFT is the Local Density Approximation (LDA) that uses a local computation. E_{XC} is written as:

$$E_{XC}^{LDA}[\rho] = \int [\varepsilon_X(\rho(\mathbf{r})) + \varepsilon_C(\rho(\mathbf{r}))]\rho(\mathbf{r})d\mathbf{r} \tag{2.28}$$

where

$$\varepsilon_X(\mathbf{r}) = -\frac{3}{4}\left(\frac{3}{\pi}\rho(\mathbf{r})\right)^{\frac{1}{3}} \tag{2.29}$$

$$\varepsilon_C(\mathbf{r}) = \begin{cases} a\ln(r_s) + b + cr_s\ln(r_s) + dr_s, r_s < 1 \\ \dfrac{\gamma}{1+\beta_1\sqrt{r_s}+\beta_2 r_s}, r_s \geq 1 \end{cases} \tag{2.30}$$

$r_s(\rho(\mathbf{r})) = \left(\frac{3}{4\pi\rho}\right)^{\frac{1}{3}}$ is the Wigner–Seitz radius, the radius of the sphere containing one electron at density $\rho(\mathbf{r})$. This measure of electron density is a classical parameter in condensed matter physics. We choose a classical set of parameters used by Perdew and Zunger(43): $a = 0.0310907$, $b = -0.048$, $c = 0.002$, $d = -0.0116$, $\gamma = -0.1423$, $\beta_1 = 1.0529$ and $\beta_2 = 0.3334$.

More complex approximations have been developed, among them the Generalized Gradient Approximation (GGA). GGA is a semi local functional, meaning it uses the density value and its spatial derivatives. Those precise XC energy functional are also useful for OF-DFT for new OF-KEDF appear that can approximate very accurately the non-interacting KEDF of specific systems. We here describe one GGA functional, the PDE, for Perdew Burke and Ernzerhof (44), XC energy functional:

$$E_{XC}^{GGA}[\rho] = E_C^{GGA}[\rho] + E_X^{GGA}[\rho] \tag{2.31}$$

with

$$E_X^{GGA}[\rho] = \int [\varepsilon_X(\rho(\mathbf{r}))\rho(\mathbf{r})F_X d\mathbf{r} \tag{2.32}$$

$$\varepsilon_X(\rho(\mathbf{r})) = -\frac{3k_F}{4\pi} \tag{2.33}$$

$$k_F = (3\pi^2\rho(\mathbf{r}))^{\frac{1}{3}} \tag{2.34}$$

$$F_X = 1 + \kappa - \frac{\kappa}{1 + \frac{\mu s^2}{\kappa}} \tag{2.35}$$

$$s = \frac{|\nabla\rho(\mathbf{r})|}{2k_F\rho(\mathbf{r})} \tag{2.36}$$

$$\kappa = 0.804 \tag{2.37}$$

$$\mu = \frac{\beta\pi^2}{3} \tag{2.38}$$

$$\tag{2.39}$$

and

$$E_C^{GGA}[\rho] = \int [H + \varepsilon_C(\rho(\mathbf{r}))]\rho(\mathbf{r})d\mathbf{r} \tag{2.40}$$

$$\varepsilon_C = -2a(1 + \alpha_1 r_s)\ln\left[1 + \frac{1}{\zeta}\right] \tag{2.41}$$

$$\zeta = 2a(\beta_1 r_s^{\frac{1}{2}} + \beta_2 r_s + \beta_3 r_s^{\frac{3}{2}} + \beta_4 r_s^2) \tag{2.42}$$

$$r_s = \left[\frac{3}{4\pi\rho(\mathbf{r})}\right]^{\frac{1}{3}} \tag{2.43}$$

$$H = \gamma \ln\left[1 + \frac{\beta}{\gamma}t^2\left(\frac{1 + At^2}{1 + At^2 + A^2t^4}\right)\right] \tag{2.44}$$

$$A = \frac{\beta}{\gamma}[e^{\frac{-\varepsilon_C}{\gamma}} - 1]^{-1} \tag{2.45}$$

$$t = \frac{|\nabla\rho(\mathbf{r})|}{2k_s\rho(\mathbf{r})} \tag{2.46}$$

$$k_s = \sqrt{\frac{4k_F}{\pi}} \tag{2.47}$$

$$\gamma = \frac{1 - \ln 2}{\pi^2} \tag{2.48}$$

and all this with $\alpha_1 = 0.21370$, $\beta_1 = 7.5957$, $\beta_2 = 3.5876$, $\beta_3 = 1.6382$, $\beta_4 = 0.49294$ and $\beta = 0.066725$.

A lot of more XC energy functional exists, often more precise, because the accuracy of Kohn-Sham DFT depends on this functional, so many resources have been spend on developing a perfect XC functional. We can cite meta GGA E_{XC}^{mGGA} that uses second derivative of electron density $\nabla^2\rho$ and the local non-interacting kinetic energy $T_s(\mathbf{r})$, the exact-exchange functionals E_{XC}^{EXX}, and all the hybrid XC functional E_{XC}^{hGGA} that are linear combinations of the others. We refer again at Libxc, the C++ library dedicated to XC functionals for precisions (42).

2.5.2 Coupled Kohn-Sham equation

The introduction of non-interacting one-electron orbitals allows the equation to be modified into N coupled KS equations. $E_{KS}[\rho]$ becomes $E_{KS}\left[\{\phi_i\}_i\right]$ and the constraint on the number of electrons $\int \rho = N$ becomes constraint on the KS orbitals being orthonormal: $\langle\phi_i|\phi_j\rangle = \delta_{ij}$. Which leads to the Lagrangian:

$$\mathcal{L}_{KS}\left[\{\phi_i\}_i, \{\varepsilon_i\}_i\right] = E_{KS}\left[\{\phi_i\}_i\right] + \sum \lambda_{ij}\left(\delta_{ij} - \langle\phi_i|\phi_j\rangle\right) \tag{2.49}$$

The minimum with respect to the constraints is reached when

$$\begin{cases} \dfrac{\delta}{\delta\phi_i}\mathcal{L}_{KS}\left[\{\phi_i\}_i, \{\lambda_{ij}\}_{ij}\right] = 0 \\ \dfrac{\delta}{\delta\lambda_{ij}}\mathcal{L}_{KS}\left[\{\phi_i\}_i, \{\lambda_{ij}\}_{ij}\right] = 0 \end{cases} \tag{2.50}$$

$$\begin{cases} \dfrac{\delta}{\delta\phi_i}E_{KS}\left[\{\phi_i\}_i\right] = \displaystyle\sum_j \lambda_{ij}\phi_j \\ \langle\phi_i|\phi_j\rangle = \delta_{ij} \end{cases} \tag{2.51}$$

And if we write E_{KS} as an observable :

$$E_{KS}\left[\{\phi_i\}_i\right] = \langle\phi_i| - \frac{1}{2}\nabla^2 + V_{eff}(\mathbf{r})|\phi_i\rangle \tag{2.52}$$

Let us call V_{eff} the effective potential, the potential with whom all electrons interact:

$$V_{eff}(\mathbf{r}) = V_{ext} + V_H + V_{XC} = \frac{\delta}{\delta\rho}(E_{ext} + E_H + E_{XC}) \tag{2.53}$$

The coupled KS equations are:

$$\left[-\frac{1}{2}\nabla^2 + V_{eff}(\mathbf{r})\right]\phi_i(\mathbf{r}) = \varepsilon_i\phi_i(\mathbf{r}) \tag{2.54}$$

with the KS orbitals constrained to be orthogonal:

$$\langle \phi_i | \phi_j \rangle = \delta_{ij} \tag{2.55}$$

and satisfying the total electron density:

$$\rho(\mathbf{r}) = \sum_i |\phi_i(\mathbf{r})|^2 \tag{2.56}$$

In these equations, the orbitals interact with each other only through the effective potential V_{eff}, through the repulsion term V_H and the XC term that integrates the interacting part of the kinetic energy $T - T_s$.

2.5.3 Iterative resolution

The Hamiltonian in the Kohn-Sham equations is dependent of the electron density ρ, and so of the equations variables. That implies the equations cannot be resolved directly and an iterative process has to be used. A density is computed with a fixed Hamiltonian, the Hamiltonian is then modified to integrate the potential generated by the density, and a new density is computed from the new Hamiltonian, until the density is consistent with the potential used to compute it, see schema 2.2. In KS-DFT, the convergence of Self-Consistent Field (SCF) iterations is not guaranteed, explanations can be found here(45).

2.5.4 Basis functions

The last approximation is the basis in which the Kohn-Sham wave functions $\{\phi_i\}_i$ are numerically represented. The basis on which are written wave functions is the main criteria that distinguish KS-DFT methods and they are often named after their function basis. We name those basis functions $\{\chi(\mathbf{r})\}_i$:

$$\phi_i(\mathbf{r}) = \sum_k \alpha_{i,k} \chi_{i,k}(\mathbf{r}) \tag{2.57}$$

And we can cite among the most used basis for wave functions:

- *Hydrogen-like atomic orbitals* $\chi(\mathbf{r})_{nlm} = R_{nl}(r) Y_l^m(\theta, \phi)$, the orbitals of the hydrogen. $R_{nl}(r)$ the product of a polynomial of degree n and an exponential $e^{-\zeta r}$.
- *Slater type orbitals* $\chi(\mathbf{r})_{nlm} = r^{n-1} e^{-\zeta r} Y_l^m(\mathbf{r})$. They have the same exponential decrease than hydrogen-like orbitals, but their polynomial part is node-less.

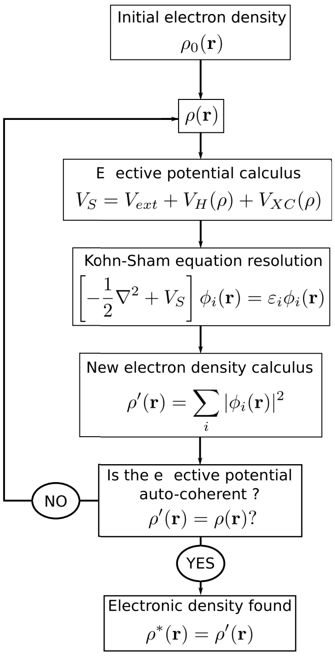

Figure 2.2: Scheme of iterative resolution of Kohn-Sham equations - The Hamiltonian and the electron density are computed alternately until convergence of ρ. If the effective potential field V_S gives the same electron density it was computed from, the field is considered self-coherent and the iterations stop.

- *Gaussian type orbitals* $\chi(\mathbf{r})_{nlm} = r^{n-1}e^{-\alpha r^2}Y_l^m(\mathbf{r})$ easier to compute than STO.

- *Plane wave basis* $\chi(\mathbf{r})_{\mathbf{k}} = e^{i\mathbf{k}.\mathbf{r}}$.

- *Bloch wave basis* $\chi(\mathbf{r})_{\mathbf{k}} = u(\mathbf{r})e^{i\mathbf{k}.\mathbf{r}}$, specialized for periodic systems, $u(\mathbf{r})$ has the same periodicity.

- *Wavelet basis* $\chi(\mathbf{r})_{n,m} = \frac{1}{\sqrt{a^{3m}}}\chi(\frac{\mathbf{r}-n\mathbf{b}}{a^m})$ with $\chi(t)$ the mother wavelet.

- *LAPW - LMTO* for Linear Augmented Planed Waves and Linear Muffin-Tin Orbitals (46).

- *PAW* for Projected Augmented Waves. A mix of several basis functions described above (47, 48).

They all have families of molecular systems on which they perform better, local basis-functions like STO or GTO are better suited for molecules, periodic functions like PAW or Bloch waves work better on systems with long-range interaction or periodic characteristic like crystals. PAW basis-function methods try to gather advantages of all the basis-functions families.

2.5.5 Orthogonalization of wave functions

In the iterative resolution of Kohn-Sham DFT 2.2, finding the KS orbitals is an eigenvalue problem. Even in for symmetric matrices, an eigenvalues computation generally scales in $O(n^3)$, and the process has to be repeated at each KS iteration. The computational cost of the kinetic energy functional of KS-DFT, that requires orthogonal orbitals and so an eigenvalue computation, restrains the KS-DFT to model large systems. To cope with this issue, the Orbital-Free DFT modifies the kinetic energy functional so that it depends only on the electron density. The computational time gain is considerable but the loss of accuracy is significant.

2.6 Orbital-Free DFT

2.6.1 OF-DFT Energy functional

Kohn-Sham-DFT requires the use of wave functions to evaluate the electronic kinetic energy (41). This usually involves a process of orbital orthogonalization to compute Kohn-Sham orbitals (38), which corresponds to a computational time scaling in $O(n^3)$, if n is the number of orbitals involved. On the contrary, OF-DFT scheme approximates $E[\rho]$ based only on electron density (49). To describe the OF-DFT energy functionals and their derivatives, we use the names and notations from the Carter group(50). For a system of N nuclei at positions $\mathbf{R} = \{\mathbf{R}_i\}_{i=1..N}$, the total electronic energy for OF-DFT is evaluated just as in KS-DFT:

$$E[\rho, \mathbf{R}] = T_s[\rho] + E_H[\rho] + E_{ext}[\rho, \mathbf{R}] + E_{XC}[\rho] \tag{2.58}$$

with

- E_{ext}, the electron interaction with the external electrostatic potential V_{ext}.
- E_H, the electrons repulsion energy or the interaction of electron with the electron potential they create V_H.
- T_s, the non-interacting electronic kinetic energy.
- E_{XC}, the exchange-correlation energy.

All those terms are describing the exact same contribution to the electron energy than in KS-DFT 2.5.1.

2.6.2 OF Kinetic Energy Density Functional

The KEDF term T_s is the one that distinguishes OF-DFT from KS-DFT. Whereas the KS-KEDF term takes wave functions ϕ as input, the OF-KEDF term is a functional of the electron density ρ alone(50). A simple way to evaluate KEDF it to use a linear combination of the two following terms:

- The Thomas-Fermi functional T_{TF}, which corresponds to the energy of a uniform electron gas (51):
$$T_{TF}[\rho] = C_{TF} \int \rho^{5/3}(\mathbf{r}) d\mathbf{r} \tag{2.59}$$
with $C_{TF} = \frac{3}{10}(3\pi^2)^{\frac{2}{3}}$

- The von Weizsäcker functional T_{vW}, exact for a single orbital system(52):
$$T_{vW}[\rho] = \int \sqrt{\rho(\mathbf{r})} \left(-\frac{1}{2}\nabla^2\right) \sqrt{\rho(\mathbf{r})} d\mathbf{r} \tag{2.60}$$

The classic scheme is a TF functional corrected by a vW functional, the $TF - \lambda vW$ model of the form:
$$T_s = T_{TF} + \lambda T_{vW} \tag{2.61}$$

The aim of the OF-KEDF functional is to approximate the KS-KEDF, which is exact in the case of non-interacting electrons. The rest of the kinetic energy is contained in the Exchange-Correlation term, common to KS-DFT and OF-DFT.

2.6.3 Further kinetic energy functional

We come back here on the most challenging part of OF-DFT: an accurate Kinetic method to compute kinetic energy from the electronic density only. We repeat KEDF is the difference between KS-DFT and OF-DFT and the part that explains the gain in computational speed and the loss in accuracy. The $TF - \lambda vW$ functional lacks of precision, missing the shell structures, the exponential decay of densities(50) and fast varying electronic densities like vacancies in metallic crystals(53). We describe here four ideas aiming to improve the kinetic energy functional.

2.6.3.1 Correction of the TF kinetic energy functional

We cite for notice a local correction developed to compute accurately the ground state kinetic energy of one atom (54). The functional is a $TF - \lambda vW$ kinetic with a correction for the Thomas-Fermi part and $\lambda = 1$, hence is called a full von Weizsäcker correction:

$$T_s[\rho] = \gamma(N, Z) T_{TF}[\rho] + T_{vW}[\rho] \tag{2.62}$$

Z the atomic number of the atom, N the electron number. Originally(54),

$$\gamma(N,Z) = 1 - \frac{C}{N^{\frac{1}{3}}} \tag{2.63}$$

We also find (49)

$$\gamma(N,Z) = \left(1 - \frac{2}{N}\right)\left(1 - \frac{C_1}{N^{\frac{1}{3}}} - \frac{C_2}{N^{\frac{2}{3}}}\right) \tag{2.64}$$

This functional gives precise energies for systems of one atom with $Z \in [\![2, 55]\!]$, but not with larger systems. The three next corrections are based on the addition of a third term to the $TF - \lambda vW$ model:

$$T_s[\rho] = T_{TF}[\rho] + \lambda T_{vW}[\rho] + T_3[\rho] \tag{2.65}$$

2.6.3.2 Gradient expansion of kinetic energy

The original correction is the theoretical correction of the Thomas-Fermi KEDF with respect to the density(55, 56). Considering ρ fluctuates around a mean and constant value ρ_0, the conventional gradient expansion (CGE) approximates the non-interaction kinetic energy as:

$$T_s[\rho] = T^0[\rho] + T^2[\rho] + T^4[\rho] + \dots \tag{2.66}$$

T^i the i^{th} order correction. The CGE finds $T^0 = T_{TF}$, the kinetic energy for a uniform electron gas and $T^2 = \frac{1}{9}T_{vW}$, confirming the $TF - \lambda vW$ KEDF and making a first proposition for the choice of λ. If $\lambda = \frac{1}{9}$ is the value that comes out the gradient expansion, it is not the only one and not always the best (50). Actually we mainly use $\lambda = 0.2$ in this work.

The higher terms diverge easily and do not improve significantly the accuracy of KEDF(56). Hence CGE is not a method of choice to work out accurate KEDF models and we have to turn to other approximations. The best methods consist in enforcing a characteristic of the non-interacting KEDF into the approximation.

2.6.3.3 Semi-local kinetic energies

A first idea is the addition of another semi-local term, id est based on spatial derivatives of ρ like the vW KEDF:

$$T_3[\rho] = T_{sl}[\rho, \nabla\rho, \nabla^2\rho, \dots] \tag{2.67}$$

The question remaining is how to choose the parameters of this third term. The results in (57, 58) show that focusing on other characteristics than only the absolute value of T_s can be beneficial. The research teams took into account the homogeneity of T_s under density scaling $-T_s$ is homogeneous of degree k if $T_s[\xi\rho] = \xi^k T_s[\rho]-$ to fit the third term's parameters. In (58) they concluded a functional of this form:

$$T_s[\rho] = T_{TF}[\rho] + c_1 T_{vW}[\rho] + c_2 \int \rho^{\frac{5}{3}} \left(\frac{|\nabla\rho|}{\rho^{\frac{4}{3}}} \right)^n d\mathbf{r} \tag{2.68}$$

with well chosen parameters c_1, c_2 and n bring improved the modeling binding for a series of small molecules. This improvement is done at small computational cost, the new functional is computed with a linear scaling with respect to the domain size, but it does not close the gap significantly between $TF - \lambda vW$ KEDF and the proper non-interacting kinetic energy of KS-DFT. A great description of the different classes of semi-local KEDF is done in (59) and a long list of semi-local KEDF is provided in the paper's supplementary.

2.6.3.4 Non-local kinetic energies

A more precise way to evaluate OF-DFT KEDF was proposed by Wang and Teter (60), they added a non-local third term to enforce the kinetic energy to respect the linear response of an electron gas (61). The response of a homogeneous non-interacting Fermi gas is known exactly(52), and gives the Fourier transform of the second functional derivative of an uniform gas of density ρ_0:

$$\hat{\mathfrak{F}} \left[\left. \frac{\delta^2 T_s[\rho]}{\delta\rho(\vec{\mathbf{r}})\delta\rho(\vec{\mathbf{r}}')} \right|_{\rho_0} \right] = -\frac{1}{\chi_{Lind}(q)} \tag{2.69}$$

$\hat{\mathfrak{F}}[f](\mathbf{k})$ being the Fourier transform of $f(\mathbf{r})$: $\hat{\mathfrak{F}}[f](\mathbf{k}) = \int f(\mathbf{r}) e^{-i\mathbf{kr}}$ and $\chi_{Lind}(q)$ the Lindhard susceptibility function:

$$\chi_{Lind}(\mathbf{q}) = -\frac{k_F}{\pi^2} \left[\frac{1}{2} + \frac{1-q^2}{4q} \ln \left| \frac{1+q}{1-q} \right| \right] \tag{2.70}$$

in which $\mathbf{q} = \frac{\mathbf{k}}{2k_F}$ is a dimensionless momentum, variable of the Fourier space, $q = |\mathbf{q}|$ its norm and $k_F = \left(3\pi^2\rho_0\right)^{\frac{1}{3}}$ is the Fermi wave vector, previously defined with ρ_0 the mean system's electron density.

The Wang–Teter (WT) KEDF enforces this linear response by adding a third term to the KEDF, the kernel energy T_K:

$$T_s[\rho] = T_{TF}[\rho] + T_{vW}[\rho] + T_K[\rho] \tag{2.71}$$

T_K having a general form:

$$\begin{aligned} T_K[\rho] &= \langle f(\rho(\mathbf{r}))|K(|\mathbf{r}-\mathbf{r}'|)|g(\rho(\mathbf{r}'))\rangle \\ &= \int f(\rho(\mathbf{r}))K(|\mathbf{r}-\mathbf{r}'|)g(\rho(\mathbf{r}'))d\mathbf{r}d\mathbf{r}' \end{aligned} \tag{2.72}$$

with $f(\rho(\mathbf{r}))$ and $g(\rho(\mathbf{r}))$ two arbitrary functions we choose to satisfies some limits of the exact non-interacting kinetic energy.

From equations 2.69 and 2.71 we have a condition on $T_K[\rho]$:

$$\hat{\mathfrak{F}}\left[\left.\frac{\delta^2 T_K[\rho]}{\delta\rho(\vec{\mathbf{r}})\delta\rho(\vec{\mathbf{r}}')}\right|_{\rho_0}\right] = -\frac{1}{\chi_{Lind}(q)} - \hat{\mathfrak{F}}\left[\left.\frac{\delta^2 T_{TF}[\rho]}{\delta\rho(\vec{\mathbf{r}})\delta\rho(\vec{\mathbf{r}}')}\right|_{\rho_0}\right] - \hat{\mathfrak{F}}\left[\left.\frac{\delta^2 T_{vW}[\rho]}{\delta\rho(\vec{\mathbf{r}})\delta\rho(\vec{\mathbf{r}}')}\right|_{\rho_0}\right] \tag{2.73}$$

If we define $\hat{K}(q)$

$$\hat{K}(q) = \frac{2k_F}{\pi^2} f'(\rho_0)g'(\rho_0)K(q) = \frac{k_F}{\pi^2}\hat{\mathfrak{F}}\left[\left.\frac{\delta^2 T_K[\rho]}{\delta\rho(\vec{\mathbf{r}})\delta\rho(\vec{\mathbf{r}}')}\right|_{\rho_0}\right] \tag{2.74}$$

with the linear responses of T_{TF} and T_{vW} given in (62):

$$\hat{\mathfrak{F}}\left[\left.\frac{\delta^2 T_{TF}[\rho]}{\delta\rho(\vec{\mathbf{r}})\delta\rho(\vec{\mathbf{r}}')}\right|_{\rho_0}\right] = \frac{\pi^2}{k_F} \tag{2.75}$$

$$\hat{\mathfrak{F}}\left[\left.\frac{\delta^2 T_{vW}[\rho]}{\delta\rho(\vec{\mathbf{r}})\delta\rho(\vec{\mathbf{r}}')}\right|_{\rho_0}\right] = \frac{\pi^2}{k_F}3q^2 \tag{2.76}$$

We come to:

$$\hat{K}(q) = -\frac{k_F}{\pi^2}\frac{1}{\chi_{Lind}(q)} - 1 - 3q^2 \tag{2.77}$$

Method name	α	β
Perrot(63)	1	1
Smargiassi-Maden(64)	$\frac{1}{2}$	$\frac{1}{2}$
Wang-Teter (60)	$\frac{5}{6}$	$\frac{5}{6}$
Wang-Govind-Carter(65)	$\frac{5+\sqrt{5}}{6}$	$\frac{5-\sqrt{5}}{6}$

Table 2.1: Parameters of different non-local kinetic energy functionals

For any choice of functions f and g. In most cases they are defined as $f[\rho] = \rho^{\alpha}$ and $g[\rho] = \rho^{\beta}$, α and β two parameters that depends on the kinetic energy functional chosen, table 2.1 cites some known parameter sets of kinetic functional.

K is dependent of the parameters chosen, so from now we note $K^{\alpha,\beta}$ and

$$T_K^{\alpha,\beta} = \int \rho(\mathbf{r})^{\alpha} K^{\alpha,\beta}(|\mathbf{r}-\mathbf{r}'|)\rho(\mathbf{r}')^{\beta} d\mathbf{r} d\mathbf{r}' \tag{2.78}$$

As described in (50), this functional is a convolution with the kernel $K^{\alpha,\beta}(r)$, and so can be solved efficiently in Fourier space:

$$T_K^{\alpha,\beta}[\rho] = \int \rho^{\alpha}(\mathbf{r}) \hat{\mathfrak{F}}^{-1}\left(\tilde{K}^{\alpha,\beta}(\mathbf{q})\rho^{\beta}(\mathbf{q})\right)(\mathbf{r}) d\mathbf{r} \tag{2.79}$$

In which the real-space form of $K^{\alpha,\beta}$ is not needed, only its Fourier-space form: $\tilde{K}^{\alpha,\beta}(\mathbf{q})$. $\hat{\mathfrak{F}}^{-1}$ is the reverse Fourier transform. We will see in the next chapter how $T_K^{\alpha,\beta}$ can approximately computed in real-space with the method developed (66, 67).

The main problem with those kernels is that they constraint the electron density to respect the Lindhard linear response only for one value of ρ, the mean ρ_0. However ρ varies on the computational domain, even the value of ρ_0 can vary depending on the computational domain's definition. This issue is addressed by using a Density-Dependent (DD) kernel, in equation 2.70, k_F is replaced by a geometric mean of the two Fermi vectors involved:

$$\xi_{\gamma}(\mathbf{r},\mathbf{r}') = \left(\frac{k_F^{\gamma}(\rho(\mathbf{r})) + k_F^{\gamma}(\rho(\mathbf{r}'))}{2}\right)^{\frac{1}{\gamma}} \tag{2.80}$$

with γ chosen depending on the system, see (50). $T_K^{\alpha,\beta}$ becomes $T_K^{\alpha,\beta,\gamma}$:

$$T_K^{\alpha,\beta,\gamma}[\rho] = \int \rho(\mathbf{r})^{\alpha} K^{\alpha,\beta,\gamma}(\xi_{\gamma}(\mathbf{r},\mathbf{r}'), |\mathbf{r}-\mathbf{r}'|)\rho(\mathbf{r}')^{\beta} d\mathbf{r} d\mathbf{r}' \tag{2.81}$$

Density Dependent kernel also corrects instabilities of Density Independent (DI) kernels for peaked densities(68). Nonetheless, it increases the complexity and is calculated by Taylor expansion around the density mean and the calculus of several Density Independent kernels (50). With a non-local kinetic energy functional using Density Dependent kernels, OF-DFT achieves accuracies very close to KS-DFT method for specific systems, but the computational cost can increase so much that the advantage of OF-DFT compared to KS-DFT, computational time, disappears(69). Several kernels were developed for different kind of systems, the Huang-Carter KEDF for semi-conductors (69, 70), here (71) for systems with covalent bonds but none has the portability of Kohn-Sham DFT.

For another overview of OF-DFT functionals, we refer to (49).

2.7 Existing methods

Several OF-DFT software have already been developed, each one of them showing different strengths either in the variability of energy functionals, computational speed, scalability or accuracy. The present work rests on those works, whether for the methods developed or for the benchmarks used to check the accuracy of computed energies. We present a quick description of some of them, sorted by their main basis used, Fourier space or real-space. We also present the solution found for the computation of non-local kinetic energy in real-space representations.

2.7.1 OF-DFT methods

2.7.1.1 Plane-wave method

PROFESS, for Princeton Orbital-Free Electronic Structure Software, is an open-source program for OF-DFT, available on this day on their Princeton Website: https://carter.princeton.edu/research/software/. The software and the work of the PROFESS team has been central in this work because it has been the reference for all measures of accuracy and computational time. They also are central in the whole field of OF-DFT for their research covers all the scale of computational chemistry:

- New functionals to improve the accuracy of OF-DFT, in particular a family of non-local kinetic energy, but also new local pseudo-potentials.

- New computational methods for efficient simulation.
- Demonstration of OF-DFT use and large-scale material simulations.

Moreover, as said above, their software is open source.

PROFESS is a plane-wave based code, so using periodic computational domain, and Fast Fourier Transform for every energy functional that could use one: Hartree repulsion energy, non-local KEDF, and the ion-ion interaction that, because of the periodic characteristic, becomes a major concern. The code can be parallelized. They use the classical C++ libraries for FFT (fftw3, Lapack), XC energy computation (Libxc) and parallelization (openmpi). The code provides several powerful non-local KEDF they developed, all the XC energy functions of the library Libxc, and the team provides also its own pseudo-potentials, specialized for bulk systems. Several dynamic schemes are available for minimizing system geometry, PROFESS gives the ability to minimize only the crystal cell or the atom position inside this cell, with the ability to restrain particles.

2.7.1.2 Real-Space methods

We present first two Real-Space Finite-Differences schemes.

- ATLAS is an OF-DFT software based on Real-Space Finite-Differences, developed in Jilin Univ (72). ATLAS computes electron density on a periodic domain, so uses Fast Fourier Transform (FFT) to efficiently compute Hartree Potential and non-local Kinetic energy (WT-KEDF, Density-Independent). They recently showed the ability of their program to be massively parallelized MPI (Message Passing Interface) and to process large-scale system: ”4 millions atoms on a 2048-cores server in one hour” (73).
- In the same style, a RS-FD OF-DFT code developed in the Georgia Institute of Technology (74). This one works with non-periodic domain and uses Poisson equation solver for the Hartree potential. All charges, ions and valence electrons, are here gathered in one neutral charge density function that eases the computation of total Coulomb energy. The KEDF is semi-local ($TF - \lambda vW$ KEDF). The optimization algorithm is an Augmented Lagrangian close to ours and the process is also parallelized but with MPI. Most of our benchmarks to verify the

validity of our methods are inspired by this work.

A new version (66) brought several improvements, the major one being the use of non-local KEDF on the model described by (67). On the contrary of the previous one, this version works with periodic conditions.

Then two Real-Space Finite-Elements schemes.

- In (75) is described a non-periodic scheme with local KEDF using the finite-elements mesh for multi-scale computations.
- Here (76) is a method non-periodic scheme with non-local KEDF. The non-local part is also computed with the method developed by (67). This work studies of the effect of the finite-elements mesh's order on the energy accuracy and algorithm convergence.

A important difference between those real-space method is whether or not their KEDF has a non-local part. Computation of non-local KEDF in real-space is an issue complex enough that some methods prefer to avoid it and stay with a pure local KEDF. We cite below two methods that solve this issue.

2.7.2 Non-local KEDF for real-space OF-DFT methods

Non-local KEDF are more accurate than local ones but bring a large computational cost surplus. Fourier space turns the convolutions like the non-local KEDF's one into a simple product, the computational cost is hence limited by the Fourier transform. Moreover, the Lindhard linear response is defined in Fourier space. For methods that stay in real-space, non-local KEDF becomes an issue. The simple solution is to use Fourier space but that imposes all the issues solved by staying in real-space – boundary conditions, parallelization, local computation etc ...–. Direct computation is not considered for computation costs.

- In (77), the Wang-Teter KEDF kernel is split in two part, a smooth one and a singular one. The singular part is solved with the Poisson solver of the Hartree potential, and the smooth one with a FFT, the Fourier representation being well suited for a smooth function. This method is a major improvement but keeps a Fourier-space part and is for now not extended to Density-Dependent KEDF kernel, more accurate.

- In (67), the Wang-Govind-Carter Density-Independent KEDF kernel is split into an infinite series of real-space compatible kernel then approximated at its four first terms. This method stays in real-space and is extended to the Wang-Govind-Carter Density-Dependent kernel. It is already used in (66) and (76). We describe the method in the next chapter and have used its Density-Independent version.

Fastest methods, like PROFESS, use Fourier space. That is because the computational bottleneck of real-space implementations are the resolutions of the non-local kinetic energy and of the Hartree potential that are faster to compute through Fourier space. However, Fourier space basis functions are not fit to benefit from a restrained dynamical model, thus, in our case, we chose a real-space method. A finite-elements method would not be adapted to moving particles because that would require to update the mesh at each time step. Hence we propose a new Real-Space and Finite-Differences (RS-FD) OF-DFT method that is presented below.

3

New implementation of Orbital-Free Density Functional Theory

We produce in this chapter our new OF-DFT code. Most of it has already been presented in (78). After the description of the Real-Space Finite-Differences (RS-FD) implementation, we assess the accuracy by comparison of computed electronic energies with our reference program, PROFESS. We then study the convergence characteristics of our program and measure on benchmarks the time taken to compute electronic structure.

3.1 Augmented Lagrangian

The first step is to tackle constraints. For the local constraint, we can compel $\rho \geq 0$ with a substitution, for example $\rho = \chi^2$. Other substitutions such as $\rho = e^\chi$ or $\rho = \chi^4$ were tested and the conclusion was that $\rho = \chi^2$ is the most stable one(50), for the von Weizsäcker functional does not diverge for small values of ρ. For consistency, we also impose χ to stay positive to prevent the von Weizsäcker functional from taking different values for the same density ρ. The global constraint $\int \rho = Q_e$ is handled with an Augmented Lagrangian method (79). Augmented Lagrangian has already been used for electron density computation (74), there is here a difference in the way the Lagrangian multiplier is updated, the formula is taken from (50). We define the

Augmented Lagrangian as

$$\mathcal{L}_A[\chi, \lambda, \mu] = E[\chi] + \frac{\mu}{2} c_{\Omega}^2[\chi, Q_e] - \lambda c_{\Omega}[\chi, Q_e] \tag{3.1}$$

with $c_{\Omega}[\chi, Q_e] = \int_{\Omega} \chi^2(\mathbf{r}) d\mathbf{r} - Q_e$ the equality functional, μ the penalization parameter and λ the Lagrangian multiplier. The evaluation of λ is not done incrementally with the classical Augmented Lagrangian formula for we have a more stable way to do it: when stabilized,

$$\nabla_{\chi} \mathcal{L}_A[\chi, \lambda, \mu] = 0 \text{ and } Q_e = 0 \tag{3.2}$$

which leads to:

$$\nabla_{\chi} E[\chi] = \lambda \nabla_{\chi} c_{\Omega}[\chi, Q_e] = 2\lambda\chi \tag{3.3}$$

and

$$\lambda = \frac{1}{2Q_e} \int_{\Omega} \chi \nabla_{\chi} E[\chi] d\mathbf{r} \tag{3.4}$$

We consider the energy minimum is reached when the Augmented Lagrangian derivative is below a tolerance threshold δ, when $|\nabla_{\chi} \mathcal{L}_A[\chi, \lambda, \mu]| < \delta$. At each optimization increment, the lagrangian multiplier is computed with formula 3.4, then a step is done in the steepest descent direction of the Augmented Lagrangian, algorithm 1.

Algorithm 1: Augmented Lagrangian algorithm for OF-DFT

```
Input: Positions of M atoms
Output: an electron density with minimized energy $\rho_m = argmin(E[\rho])$
Initialization of the cell $\Omega$
Computation of the external potential $V_{ext}$ on $\Omega$
while $|\nabla_{\chi} \mathcal{L}_A[\chi, \lambda, \mu]| > \delta$ do
    $V_H(\mathbf{r}) \leftarrow \int_{\Omega} \frac{\chi_k^2(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'$        : update of the Hartree Potential
    $\lambda = \frac{1}{2Q_e} \int_{\Omega} \chi(\mathbf{r}) \frac{\delta E[\chi]}{\delta \chi} d\mathbf{r}$        : update of the Lagrangian multiplier
    $\partial\chi_k \leftarrow \nabla_{\chi} \mathcal{L}_A[\chi, \lambda, \mu]$        : compute the Lagrangian gradient
    $\chi'_{k+1} \leftarrow abs(\chi_k - \gamma\partial\chi_k)$        : perform a steepest descent step and take the
      absolute value of the result
end
```

3.2 Real Space, Finite Difference implementation

3.2.1 Real Space scheme

The second step is the choice of the basis in which the calculations are performed. Fastest OF-DFT schemes use the Fourier space to evaluate a system's electronic energy and compute the electron density. Since in Fourier space OF-DFT functional calculations scale linearly with the grid size, the global cost of an energy evaluation comes from the Fourier transform. With the Fast Fourier Transform (FFT) this cost is limited to $[O(N \ln(N))$. Here, we do not use Fourier space since the basis functions have a global impact of the electron density whereas we want to use local methods allowing to take advantage of locally restrained particles. Hence we use a real space (RS) scheme which also presents other advantages: RS is well adapted to isolated system, can use arbitrary boundary conditions and makes computations easy to parallelize. Thus every variables, ρ and V_{ext} among them, are computed on a discrete cell $\mathit{\Omega}$ with cubic nodes of side size h and volume $v_h = h^3$. We call N_x, N_y and N_z the number of nodes on the x, y and z directions, they are chosen so that $\mathit{\Omega}$ is large enough to encompass all the molecular system's electrons. Those nodes are indexed by integers $i \in [\![1, N]\!]$ with $N = N_x N_y N_z$ the total number of nodes. The functions $\rho(\mathbf{r})$, $\chi(\mathbf{r})$, $V_{ext}(\mathbf{r})$ and $V_H(\mathbf{r})$ become $\rho(i)$, $\chi(i)$, $V_{ext}(i)$ and $V_H(i)$, the electron density, the external potential and the Hartree potential of node i. Similarly, $n(i)$ stands for the set of positions inside node i and $\mathbf{r}(i)$ its center.

3.2.2 Discrete version of OF-DFT

In real space, with finite differences and with the new variable χ, our energy functionals become:

$$E_{ext}[\chi] = \sum_{i \in [\![1,N]\!]} -V_{ext}(i)\chi^2(i)v_h \tag{3.5}$$

$$E_H[\chi] = \frac{1}{2} \sum_{i \in [\![1,N]\!]} V_H(i)\chi^2(i)v_h \tag{3.6}$$

$$T_{TF}[\chi] = C_{TF} \sum_{i \in [\![1,N]\!]} \chi^{10/3}(i)v_h \tag{3.7}$$

$$T_{vW}[\chi] = \sum_{i \in [\![1,N]\!]} \chi(i)\left(-\frac{1}{2}\nabla^2\right)\chi(i)v_h \tag{3.8}$$

$$E_{XC}[\chi] = \sum_{i \in [\![1,N]\!]} \chi^2(i)(\varepsilon_X(\chi^2(i) + \varepsilon_C(\chi^2(i)))v_h \tag{3.9}$$

and, $\forall i \in [\![1,N]\!]$, their derivatives with respect to χ can be written as:

$$\nabla_\chi E_{ext}(i) = 2V_{ext}(i)\chi(i) \tag{3.10}$$

$$\nabla_\chi E_H(i) = 2V_H(i)\chi(i) \tag{3.11}$$

$$\nabla_\chi T_{TF}(i) = C_{TF}\frac{10}{3}\chi^{7/3}(i) \tag{3.12}$$

$$\nabla_\chi T_{vW}(i) = -\nabla^2\chi(i) \tag{3.13}$$

$$\nabla_\chi E_{XC}(i) = \mu_X(\chi(i)) + \mu_C(\chi(i)) \tag{3.14}$$

with

$$\mu_X(\chi(i)) = -\left(\frac{3}{\pi}\chi^2(i)\right)^{\frac{1}{3}} \tag{3.15}$$

$$\mu_C(\chi(i)) = \begin{cases} (b - \frac{1}{3}a) + a\ln(r_s) + \frac{1}{3}(2d - c)r_s + \frac{2}{3}cr_s\ln(r_s) & , r_s < 1 \\ \frac{\gamma + \frac{7}{6}\gamma\beta_1\sqrt{r_s} + \frac{4}{3}\gamma\beta_2 r_s}{(1+\beta_1\sqrt{r_s}+\beta_2 r_s)^2} & , r_s \geq 1 \end{cases} \tag{3.16}$$

$$r_s(i) = \left(\frac{3}{4\pi\chi^2(i)}\right)^{\frac{1}{3}}. \tag{3.17}$$

3.2.3 Hartree potential

The Hartree Potential V_H is not directly derived from its definition, but from the corresponding Poisson equation. Hence, equation 2.23 becomes

$$4\pi\rho = -\varDelta V_H \tag{3.18}$$

with free boundary condition:

$$V_H(\mathbf{r}) \to 0 \text{ when } |\mathbf{r}| \to +\infty \tag{3.19}$$

The Poisson equation is solved with a conjugate gradient algorithm. To simulate the free boundary conditions, two others cells are used: Ω_2 and Ω_3. Ω_2 has the same center than the cell Ω, but has nodes three times larger with the same number of nodes and the same shape (same N_x, N_y and N_z), so is three times larger. Hence, a node of Ω_2 has the same size than 27 nodes of Ω. Ω_3 follows the same principle: same center and shape, and nodes three times larger than Ω_2. Thus we have three cells centered at the same position, each one encompassing the precedent like three Russian nesting dolls, see figure 3.1. V_H is first solved on the largest and coarsest cell Ω_3, with the boundary condition $V_H(\mathbf{r}) = \frac{Q_0}{|\mathbf{r}-\mathbf{r_0}|}$, Q_0 is the total valence electron charge and r_0 its barycenter. From this cell, the boundary conditions are extracted for the cell Ω_2 with a C1 interpolation and V_H is solved on Ω_2. The same procedure is done between Ω_2 and Ω, see algorithm 2.

Algorithm 2: Electronic Hartree Potential Computation on Ω

```
Input: The Electronic Density $\rho$ on $\Omega$
Output: The Electronic Potential $V_H$ on $\Omega$
Inject Electronic Density in $\Omega_2$
Inject Electronic Density in $\Omega_3$
Create Coulomb Boundary Conditions on $\Omega_3$'s border:
  $\forall \mathbf{r} \in \partial_{\Omega_3}, V_H(\mathbf{r}) = \frac{Q_0}{|\mathbf{r}-\mathbf{r_0}|}$
Solve $V_H$ on $\Omega_3$
Extract $V_H$ boundary conditions for $\Omega_2$ from $\Omega_3$
Solve $V_H$ on $\Omega_2$
Extract $V_H$ boundary conditions for $\Omega$ from $\Omega_2$
Solve $V_H$ on $\Omega$
```

3.2.4 Exchange-Correlation Potential

We have implemented in our code both the LDA functional with the Perdew and Zunger (43) set of parameters and the PDE XC functional (44) described higher 2.5.1.4. We

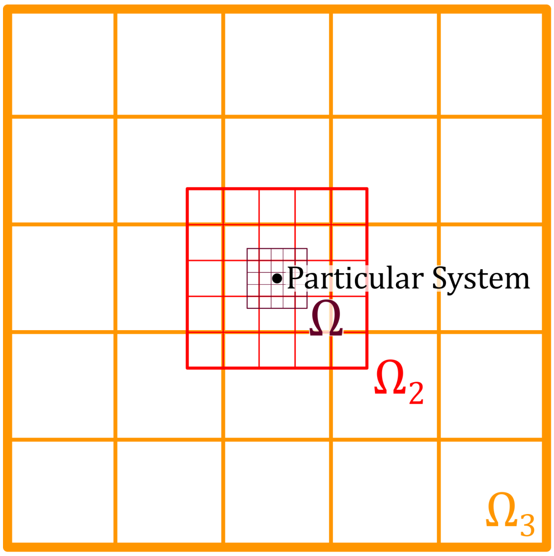

Figure 3.1: Hartree Potential Grids - Nesting of the three grid cells Ω, Ω_2 and Ω_3 that are used to compute the Hartree Potential on the thin cell Ω. Lengths are tripled from Ω to Ω_2 and from Ω_2 to Ω_3, but sizes stays equal, here $5 \times 5 \times 5$. The grids used to compute electron densities and their electric potentials are usually bigger for example $100 \times 100 \times 100$.

will mainly use the LDA in the benchmarks because in OF-DFT with a simple KEDF, the most significant parts of errors come from the pseudopotentials and the KEDF, so using a precise XC functional does not improve much the accuracy of the whole electron density calculus. In real space, both functionals are computed in a linear time.

3.2.5 Local Kinetic Energy

For the $TF - \lambda vW$ KEDF, several λ are proposed and justified in literature(80, 81, 82). We have taken $\lambda = 0.2$ for energies are more accurate on isolated molecular systems and it has already been chosen in another implementation we want to compare to(74). For all the benchmarks, we have used a local kinetic energy, the principal reason is the difficulties we have had to create a stable non-local KEDF in real space and that could be updated in a local and incremental way, which was the aim of the thesis. We describe nonetheless the implementation of the non-local kinetic energy in real-space.

3.2.6 Non-local Kinetic Energy

To compute a non-local kinetic energy functional is not an easy task in real space. As we want to stay in real space, we used the technique described in (67, 74) to implement a density dependent kernel WGC functional (65).

We remind the non local part of the kinetic energy :

$$T_K^{\alpha,\beta}[\rho] = \int \rho^\alpha(\mathbf{r}) K^{\alpha,\beta}(|\mathbf{r}-\mathbf{r}'|)\rho^\beta(\mathbf{r}')d\mathbf{r}d\mathbf{r}' \tag{3.20}$$

That we split :

$$T_K^{\alpha,\beta}[\rho] = \int \rho^\alpha(\mathbf{r}) \left(\int K^{\alpha,\beta}(|\mathbf{r}-\mathbf{r}'|)\rho^\beta(\mathbf{r}')d\mathbf{r}' \right) d\mathbf{r} \tag{3.21}$$

to isolate a kind of kinetic potential $V_K[\rho]$

$$V_K[\rho](\mathbf{r}) = \int K^{\alpha,\beta}(|\mathbf{r}-\mathbf{r}'|)\rho^\beta(\mathbf{r}')d\mathbf{r}' \tag{3.22}$$

that we can approximate in real space by decomposing its Fourier space transform. $\tilde{K}(q)$ can be written:

$$\tilde{K}(q) = \frac{N_2 q^2 + \cdots + N_{2m}q^{2m}}{D_0 + D_2 q^2 + \cdots + D_{2m}q^{2m}} \tag{3.23}$$

Because of the high order term of $\tilde{K}(q)$ 2.77 and because of its Taylor expansion in 0 and ∞. N_i and D_i are real coefficients. From there $\tilde{K}(q)$ can then be decomposed in partial fractions:

$$\tilde{K}(\mathbf{q}) = \sum_{j=1}^{m} \frac{P_j q^2}{q^2 + Q_j} \tag{3.24}$$

The convolution of $K(\mathbf{r})$ and $f(\mathbf{r})$ becomes in Fourier space a multiplication, so $V_K(\mathbf{q})$ in Fourier space is written:

$$V_K(\mathbf{q}) = \tilde{K}(q)f(\mathbf{q}) = \sum_{j=1}^{m} V_i(\mathbf{q}) \tag{3.25}$$

with

$$V_i(\mathbf{q}) = \frac{P_j q^2}{q^2 + Q_j} f(\mathbf{q}) \tag{3.26}$$

	$j = 1$	$j = 3$
P_j	$0.026696 + i0.145493$	$-0.826696 + i0.691930$
Q_j	$-0.818245 - i0.370856$	$0.343051 - i0.689646$

Table 3.1: Coefficients of the approximate partial fraction decomposition of the non-local KEDF kernel

$$\begin{gathered}
(q^2 + Q_j)V_i(\mathbf{q}) = P_j q^2 f(\mathbf{q}) \\
\left[-\frac{1}{(2k_F)^2}\nabla^2 + Q_j\right] V_j(\mathbf{r}) = -\frac{P_j}{(2k_F)^2}\nabla^2 f(\mathbf{r}) \\
\left[\nabla^2 - (2k_F)^2 Q_i\right] V_j(\mathbf{r}) = P_j \nabla^2 \rho^\beta(\mathbf{r})
\end{gathered} \tag{3.27}$$

Each $V_i(\mathbf{r})$ is the solution of a complex Helmholtz equation, the classical equation of stationary waves. We solved it with a bi-conjugate gradient solver.

Derivative Functional :

$$\frac{\delta T_{nl}^{\alpha,\beta}}{\delta \rho}(\mathbf{r}) = \alpha \rho^{\alpha-1}(\mathbf{r}) \int K^{\alpha,\beta}(|\mathbf{r} - \mathbf{r}'|)\rho^\beta(\mathbf{r}')d\mathbf{r}' + \beta \rho^{\beta-1}(\mathbf{r}) \int \rho^\alpha(\mathbf{r}')K^{\alpha,\beta}(|\mathbf{r} - \mathbf{r}'|)d\mathbf{r}' \tag{3.28}$$

with

$$\int K^{\alpha,\beta}(|\mathbf{r} - \mathbf{r}'|)\rho^\alpha(\mathbf{r}')d\mathbf{r}' = V^\alpha = \sum_{j=1}^{4} V_j^\alpha = 2 * \Re(V_1^\alpha + V_3^\alpha) \tag{3.29}$$

$$\int K^{\alpha,\beta}(|\mathbf{r} - \mathbf{r}'|)\rho^\beta(\mathbf{r}')d\mathbf{r}' = V^\beta = \sum_{j=1}^{4} V_j^\beta = 2 * \Re(V_1^\beta + V_3^\beta) \tag{3.30}$$

$$\tag{3.31}$$

So four complex tables to keep: V_1^α, V_3^α, V_1^β, V_3^β

$$\nabla_\chi E_{nl}^{\alpha,\beta}(i) = 2\chi(i)\left(\alpha\rho^{\alpha-1}(i) * 2 * \Re(V_1^\alpha(i) + V_3^\alpha(i)) + \beta\rho^{\beta-1}(i) * 2 * \Re(V_1^\beta(i) + V_3^\beta(i))\right) \tag{3.32}$$

3.3 Pseudo-potentials

Core electrons are not directly involved in the processes of atom binding. Hence, we may reduce our potential to a *pseudopotential*, where ρ accounts only for the valence electrons, and V_{ext} accounts for the potential created by the ions composed by nuclei and core electrons. This model is called the frozen core approximation, the effect of core electrons is present only in the shape of the pseudopotential that models the screening effect. Most of the pseudopotentials were developed for orbital-bounds methods like KS-DFT and so produce one potential per kind of electron to model, exactly per "angular momentum" l, they are non-local, and can be written:

$$V_{ps}(\mathbf{r}) = \sum_{l} V_l(r) \left| l \right\rangle \left\langle l \right| \tag{3.33}$$

with every angular momentum potential V_l being a radial function, hence the $V_l(r)$. In OF-DFT, because orbitals are not considered, the pseudopotential is local, and here is only a radial function:

$$V_{lps}(\mathbf{r}) = V(r) \tag{3.34}$$

And the total potential created by the ions is of course the sum of the ions potentials:

$$V_{ext}(\mathbf{r}) = \sum_{I} V_{lps}(|\mathbf{r} - \mathbf{R}_I|) \tag{3.35}$$

$\mathbf{R}_I$ the position if the I^{th} ion. We cite here three important local pseudopotentials (LPS).

3.3.1 Heine Abarenkov pseudopotentials

Heine-Abarenkov pseudopotentials are originally designed for spectroscopic energy levels(83, 84). In their local version, they are particularly simple for they are tuned with two parameters, a core radius R_m and a core potential A. HA potential is defined as:

$$V_{HA}(r) = \left\{ \begin{array}{cc} A & r < R_m \\ -\frac{Z_e}{r} & r > R_m \end{array} \right. \tag{3.36}$$

For aluminum, we took for example $Z_e = 3\ e$, the charge of the ion, $A = 0.11\ V$ and $R_m = 1.16\ a_e$. This pseudopotential is not continuous in $r = R_m$, see its radial representation 3.2, and so produce a significant dependence to ion position in the grid when used with a coarse grid mesh.

3.3.2 Goodwin-Needs-Heine pseudopotentials

A standard local pseudopotential is the Goodwin-Needs-Heine, or GNH, pseudopotential. GNH pseudopotentials are obtained from HA-PS written in the Fourier space and damped at high frequencies to remove the discontinuity (85). In the original paper, HA-PS in Fourier space is given as:

$$V_{HA}(q) = -\frac{4\pi}{\Omega q^2}\left[(Z - AR)\cos(Rq) + \frac{A}{q}\sin(qR)\right] \tag{3.37}$$

with q the Fourier space variable and Ω a normalization constant. And the damping function a steep sigmoid function with its step in q_c:

$$f(q) = e^{-\left(\frac{q}{q_c}\right)^6} \tag{3.38}$$

(85) gives those parameters for aluminum: $A = 0.1107$, $q_c = 3.5\ a_e^{-1}$ and $Z = 3\ E$. We find in (49) a real-space formulation:

$$V_{GNH}(r) = \frac{2}{\pi}\int_0^{\infty} \frac{\sin(ru)}{ru}\left((Z - AR)\cos(Ru) + \frac{A}{u}\sin(Ru)\right)e^{-(\frac{u}{R_c})^6}du \tag{3.39}$$

They also give parameter for Germanium and Arsenic element, but not for the entire periodic table. Indeed the accuracy of a PP family depends a lot on the element. For example, for Magnesium the "Madden PP" (86) and for Silicon the "Zhou PP" (87).

3.3.3 Bulk-derived Local Pseudopotentials, BLPS

To fill the lack of local and transferable pseudo-potential for OF-DFT, the Carter Group of Princeton computed another family of Pseudo-potential, especially derived for OF-DFT and for bulk systems (69, 88, 89). They are built from KS-DFT solutions on different crystal bulks (Simple Cubic, Bulk-Centered Cubic, Face-Centered Cubic, Hexagonal Close Packed or Diamond etc ...) and the inversion of KS equations to find the electric potential that can generate those solutions. They have shown that the bulk characteristics (bulk moduli, bulk equilibrium volumes, bulk equilibrium energies, vacancy energy ...) produced with their Bulk-Derived Local Pseudopotentials (BLPS) are closer to KS-DFT values than with previous state of the art PP.

We drew in 3.2 the three pseudopotential for aluminum described higher (HA, GNH and BLPS) plus the corresponding Coulomb potential. They all expose a whole in the center that models the screening effect.

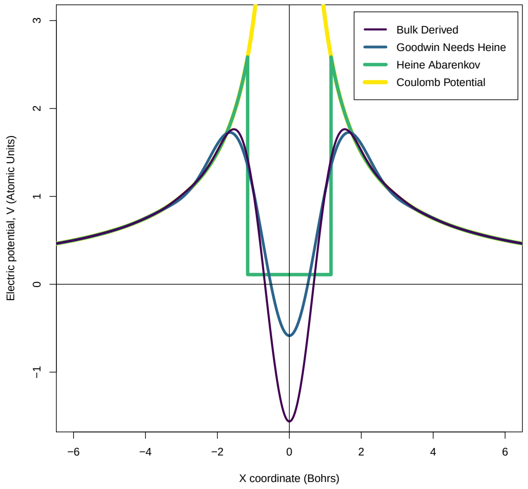

Figure 3.2: Pseudo-potentials for aluminum atoms - In purple the Bulk-Derived Local Pseudo-potential, in blue the Goodwin-Needs-Heine pseudopotential, in green the Heine-Abarenkov pseudopotential and in yellow the Coulomb potential of a ion of charge $Z = 3e$.

In this work we chose the Bulk-derived Local Pseudo-potentials (BLPS). Besides being more accurate, several elements are available and as they were computed with the same method they form a coherent set of pseudopotentials.

3.3.4 Gauss-Legendre quature

We now have to compute the potential on the computational space. On a grid cell, the potential value in one node $V(i)$ is in theory :

$$V(i) = \int_{\mathbf{r} \in n(i)} V(\mathbf{r}) d\mathbf{r} \tag{3.40}$$

We remind $n(i)$ is the set of positions inside node i. We use in practice the Gauss-Legendre quadrature to estimate $V(i)$. For a n-order Gauss-Legendre quadrature, if $\left\{\omega_i^{GL}\right\}_i$ are the n weight coefficients and $\left\{x_i^{GL}\right\}_i$ the positions of measure :

$$\int_{-1}^{1} f(t)dt = \sum_{i\in[1,n]} \omega_i^{GL} f(x_i^{GL}) \tag{3.41}$$

We recall in the appendices the one-dimension coefficients (position and weight) of the Gauss-Legendre quadrature(90, 91).

In three dimension in a cube of side size h and centered on $\mathbf{r}(i)$ the weights become

$$\omega_{i_x,i_y,i_z}^{GL} = \frac{1}{8}\omega_{i_x}^{GL}\omega_{i_y}^{GL}\omega_{i_z}^{GL}/8 \tag{3.42}$$

and the positions

$$\mathbf{r}_{i_x,i_y,i_z}^{GL}(i) = \mathbf{r}(i) + \frac{h}{2}x_{i_x}^{GL}\vec{e}_x + \frac{h}{2}x_{i_y}^{GL}\vec{e}_y + \frac{h}{2}x_{i_z}^{GL}\vec{e}_z \tag{3.43}$$

The ratio 8 in $\omega_{i_x,i_y,i_z}^{GL}$ calculus is here because the coefficients are tuned for the integration on the $[-1, 1]$ segment, so the corresponding cube is of volume 8.

$$\int_{\mathbf{r}\in n(i)} V(\mathbf{r})d\mathbf{r} = \sum_{(i_x,i_y,i_z)\in[1,n]^3} \omega_{i_x,i_y,i_z}^{GL} V(\mathbf{r}_{i_x,i_y,i_z}^{GL}(i)) \tag{3.44}$$

In figure 3.3 we have tested several the three pseudopotential with quadratures of different degree on a simple particle system, one atom, because it produces the greatest differences, less atoms means steeper potentials. We have chosen to compute V_{ext} with a Gauss-Legendre quadrature of degree three. For BLPS the precision does not improve after $n = 3$.

Now that we have methods to compute the four energies from the electron density, we can compute the electronic structure of our atomic system.

3.4 Results

In this section, we validate our OF-DFT implementation by comparing its predictions with those of a Fourier space implementation: PROFESS(50). Most of the tests are inspired by two other implementations, a Real-space Finite-Differences one(74) and a Real-Space Finite-Element one(75). Different aluminum clusters energy are compared with the one found by PROFESS, depending on the calculus, we might compare the electronic energy E, the total energy $E_{Tot} = E + E_{ii}$, the total energy per atom $E_{at} = E_{Tot}/M$ or the binding energy $\mathcal{E}_b = (E_{Tot} - \sum E_j)/M$ with E_j the energy of the isolated

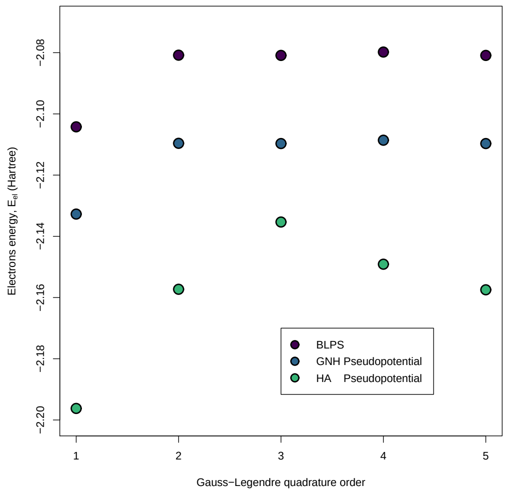

Figure 3.3: Effect of quadrature on electronic energy - For the three main local pseudopotentials (BLPS, GNH and HA), the effect of the order of the Gauss-Legendre quadrature on the electronic energy calculation of one aluminum atom. For the slowly-varying pseudopotentials, BLPS and GNH, a second order quadrature is sufficient to approximate their effect on a grid. On the contrary, HA pseudopotentials are harder to approximate because of its fast variations.

j^{th} atom. We then measure the method's performance and the speedup gained by an incremental update compared to a global update. At last we test our method on a simulation of aluminum implantation. For coherent comparisons, we have chosen TF-vW kinetic energy with $\lambda = 0.2$ and the BLPS given by PROFESS. The values found by PROFESS are slightly different in this work and in (74) for the pseudopotentials used are different: Goodwin-Needs-Heine (GNH) pseudopotentials instead of BLPS.

Before measuring any computational time, we need to ensure our implementation gives proper results. To do so we compute the electronic structures and energies of different aluminum compounds like dimer Al_2, trimer Al_3, tetramer Al_4 and Faces Centered Cubic (FCC) crystals of different sizes and lattice constants. First, we study the dependencies of parameters like the margin size of the computational domain, the spatial discretization and the Laplacian order used for the computation of Hartree potential and vW kinetic energy. This will allow to choose a proper set of parameters.

3.4.1 Parameterization

We want a computational domain that contains all the electrons. We build the domain as a box surrounding all the ions, plus a margin that needs to be defined large enough to encompass all the electrons. We choose to consider 10^{-10} electrons per cubic Bohrs as a null electron density. As one can see in figure 3.4, if we take a thin node size $h = 0.2a_e$ and a high order (10^{th}) Laplacian operator, our OF-DFT implementation computes an electron cloud of a single aluminum ion that vanishes at $12.2a_e$ from the nucleus. From such a result, we set a margin of $15a_e$ for all our tests, assuming this margin is sufficient to do not degrade the quality of the results. As simple examples, the domain for a single atom is a cube with $30a_e$ edges, and the one for a $6 \times 6 \times 6$ FCC bulk with $8a_e$ as lattice parameter is a cube with $5 \times 8 + 2 \times 15 = 70a_e$ edges.

Then we choose a node size h and a Laplacian derivative order n. Here, again, the goal is to find the best trade-offs between precision and efficiency. To do so, we compute the binding energy of a $5 \times 5 \times 5$ FCC aluminum cluster of lattice parameter $a = 8.0a_e$ for different h and n. Results are presented in figure 3.5. As one can see, the binding energies computed with a 2^{nd} order Laplacian are largely dependent on h. However, this dependency disappears when a Laplacian of order 6 or higher, is used. To choose a proper node size, we look at the energy per atom of our aluminum bulk, computed with a 6^{th} order Laplacian and different node sizes: figure 3.6 presents the

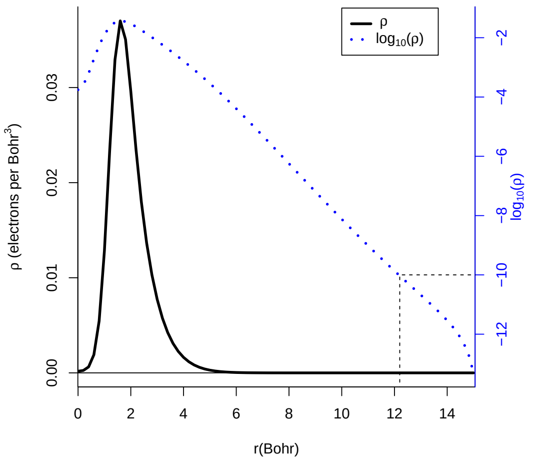

Figure 3.4: Radial electron density - Electron density distribution around a single aluminum ion and its decimal logarithm. The density vanishes below 10^{-10} e/Bohr^3 for $r = 12.2a_e$.

results. We observe that the relative error falls below 2% at $h = 0.5a_e$, although E_{at} largely depends on h even for small h. Since having a smaller h would drastically increase the computational costs, we consider $h = 0.5a_e$ as value sufficient to obtain a satisfying accuracy. Hence, for the validation of our method, computations will use a 6^{th} order Laplacian and a node size $h = 0.5a_e$, as in (74). For other tests involving more computations, bigger step sizes will also considered ($h = 0.7a_e$ and $h = 1a_e$).

3.4.2 Evaluation of accuracy

From the parameters proposed earlier, we show that our implementation computes correct energies. We first try to isolate one source of error, the discretization of the computational domain, with the example of the hydrogen atom. Then we work on aluminum clusters to compare our implementation with others.

3.4.2.1 Hydrogen atom

The hydrogen atom presents a particular case. First because the exact solution is known. Second we can retrieve the exact solution with OF-DFT by using only the external potential energy and the von Weizsäcker kinetic energy. Indeed, the hydrogen's

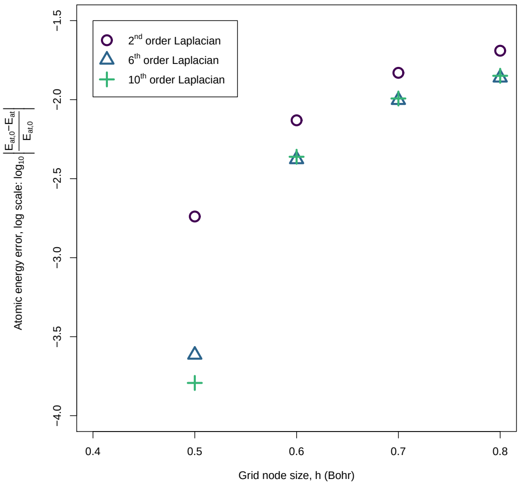

Figure 3.5: Binding Energy of a 666 atoms aluminum cluster - $\mathcal{E}_b$ for $M = 666$ with respect to the spatial discretization h and the Laplacian order used for the computation of system's kinetic energy and Hartree potential. With a 2^{nd} order Laplacian, the purple circle, the binding energy varies too much with h, hence the choice of a $6th$ order Laplacian for the method.

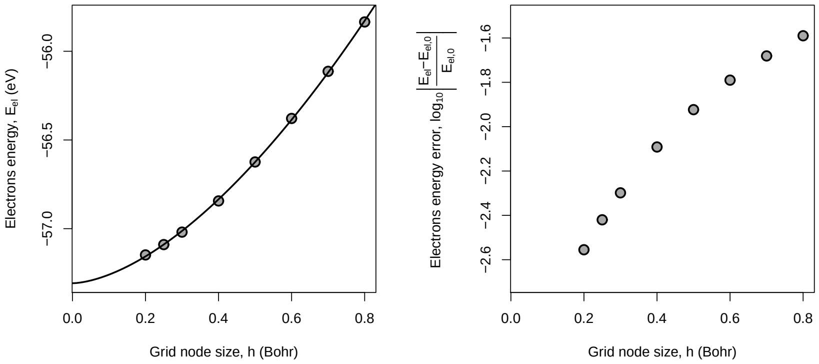

Figure 3.6: Electronic energy of a single aluminum atom - On the left, E_{el} for $M = 1$ with respect to the grid node size h and a polynomial regression $E_{el} = E_{el,C} + h^p$ that gives an asymptotic value of $E_{el,C} = -57.308$. On the right the log error compared to $E_{el,C}$.

electron is a one-particle system, hence the electrons interaction energy is null, and as we have a system with only one orbital, the von Weizsäcker kinetic energy is exact 2.6.2. In the OF-DFT functionals, the Exchange-Correlation energy is the correction of the electrons kinetic energy and electrons interaction energy approximations 2.5.1.4, is this system in which the electron interaction is null and the kinetic energy exact, the Exchange-Correlation energy is useless.

The hydrogen's electron is in an electric potential $V(\mathbf{r}) = -\frac{1}{|\mathbf{r}|}$. The time-independent Schrödinger equation gives the exact form of the energy ground state, the $1s$ hydrogen orbital ($n = l = m = 0$):

$$\psi(\mathbf{r}) = \frac{1}{\sqrt{\pi}} e^{-|\mathbf{r}|} \tag{3.45}$$

And the values of the potential and kinetic parts of the electron energy:

$$\langle\psi|V|\psi\rangle = -\frac{1}{\pi}\int \frac{1}{|\mathbf{r}|} e^{-2|\mathbf{r}|} d\mathbf{r} \qquad = -1 \tag{3.46}$$

$$\langle\psi| -\frac{1}{2}\nabla^2|\psi\rangle = -\frac{1}{2\pi}\int e^{-|\mathbf{r}|}\nabla^2 e^{-|\mathbf{r}|} d\mathbf{r} \qquad = 0.5 \tag{3.47}$$

And so :

$$\langle\psi|H|\psi\rangle = -0.5 \tag{3.48}$$

All the energy values are in Hartree.

For the OF-DFT, as we saw the Hartree repulsion was null, the Kinetic energy perfectly described by T_{vW} and so the XC energy useless, we can simplify the OF-DFT energy equation 2.58 to:

$$E_{el}[\rho, \mathbf{R}] = E_{ext}[\rho, \mathbf{R}] + T_s[\rho] \tag{3.49}$$

with this time

$$T_s[\rho] = T[\rho] = T_{vW}[\rho] \tag{3.50}$$

Using our variable, the square-root of the electron density, the exact solution is the same:

$$\chi(\mathbf{r}) = \frac{1}{\sqrt{\pi}} e^{-|\mathbf{r}|} \tag{3.51}$$

We computed the $1s$ orbital using OF-DFT with a computational domain of side size 10 a_e centered around the hydrogen atom, a 4^{th} Gauss-Legendre quadrature to avoid the singularity in 0 and different node grid node sizes to link the discretization to the accuracy.

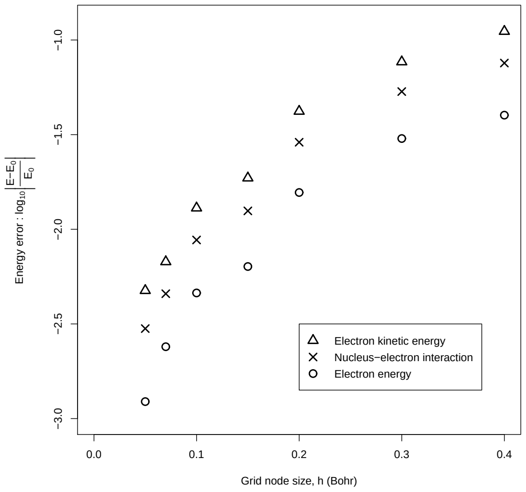

Figure 3.7: Errors of electron's hydrogen energies - The electron's energy are computed with different grid node sizes and compared to their theoretical values: $E_{ext} = -1$, $T = 0.5$ and $E_{el} = -0.5$, in Hartree. As the functionals used are exact, errors are here only due to the computational domain characteristics, size and discretization.

In Figure 3.7 we plot the relative errors of external potential energy, kinetic energy and total electron energy for several grid node sizes. As our method over-evaluates the kinetic energy and under-evaluates the nucleus-electron interaction under-evaluated, the total energy relative error is smaller than its two components. We can achieve arbitrary precision by reducing the grid node size.

Figure 3.7 clarifies that the errors are condensed around the singularity, where the external potential is steep. For systems with fast varying potentials, among them all-electrons systems with Coulomb potentials, adaptive grids like Finite-Elements methods (76) are more adapted because the mesh can be refined where the external potential is the steepest. Unfortunately, in a dynamical simulation this mesh has to be updated at each time step to follow nuclei. That is a reason why we have chosen a fixed mesh.

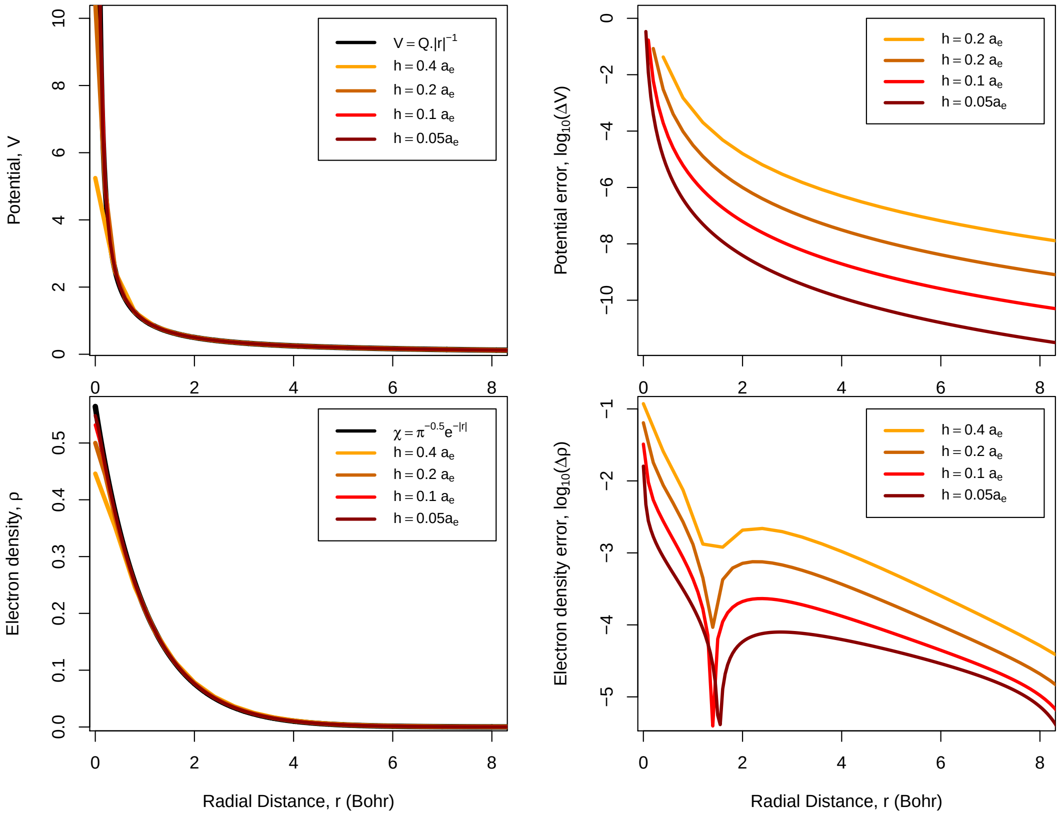

Figure 3.8: Electric potential and Electron density of the hydrogen atom - On the left the nucleus electric potential and the electron density, the theoretical values and the computed values with different discretizations. On the right the errors in log scale, of the computed values compared with the theoretical ones. For the computed values, we used a computational domain of side 20 a_e and grid node sizes of $h = 0.4, 0.2, 0.1, 0.05\ a_e$.

3.4.2.2 Aluminum clusters

We use the same methodology as the one proposed in (74): The energies and equilibrium bond lengths of different aluminum clusters are computed and compared with the ones found with PROFESS.

First, we compute binding energies of aluminum dimers, trimers and tetramers, arranged respectively in line, triangle and tetrahedron, with different bond lengths, figure 3.9 exposes the energy curves. We have derived from those curves the equilibrium binding energy and the equilibrium bond length of those three compounds that we can compare with the results found by PROFESS. The results are presented in Table 3.2. With the parameters used, we have a relative error below 2% for the binding energy and around 1% for the bond length.

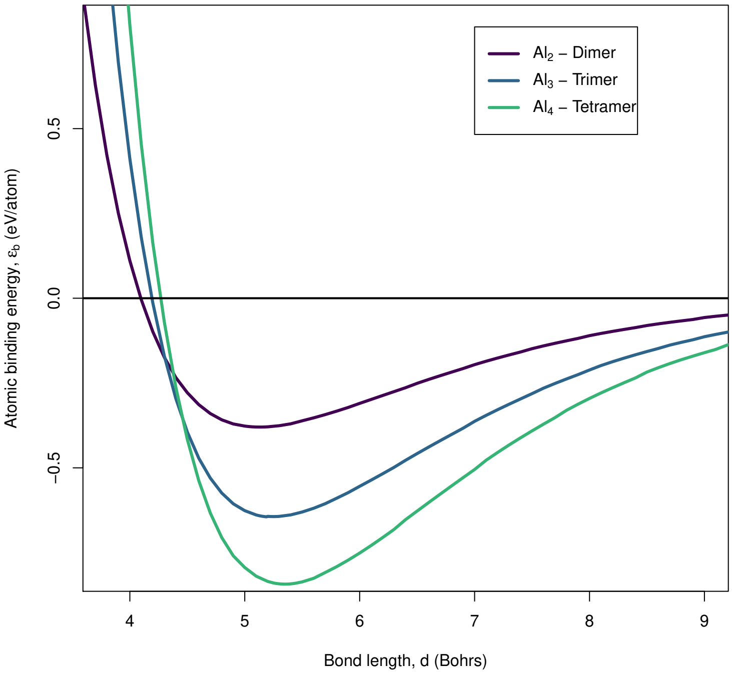

Figure 3.9: Binding Energies of small aluminum clusters - Binding Energy of the three smallest aluminum clusters (Al_2, Al_3 and Al_4), with respect to the bond length.

To show the density of the electron cloud that our method computes, we provide in figure 3.10 the heat map of the electron density of an aluminum dimer at equilibrium

Cluster	**ε_b(eV/at)**		**error(%)**	**R_e(Bohr)**		**error(%)**
	RS-FD	**PROFESS**		**RS-FD**	**PROFESS**	
Al_2	-0.382	-0.384	0.52	5.09	5.06	0.39
Al_3	-0.644	-0.649	0.77	5.25	5.18	1.3
Al_4	-0.843	-0.851	0.94	5.35	5.27	1.5

Table 3.2: Binding Energy ε_b and equilibrium bond length R_e for Al_2, Al_3 and Al_4

($d = 5.09\ a_e$).

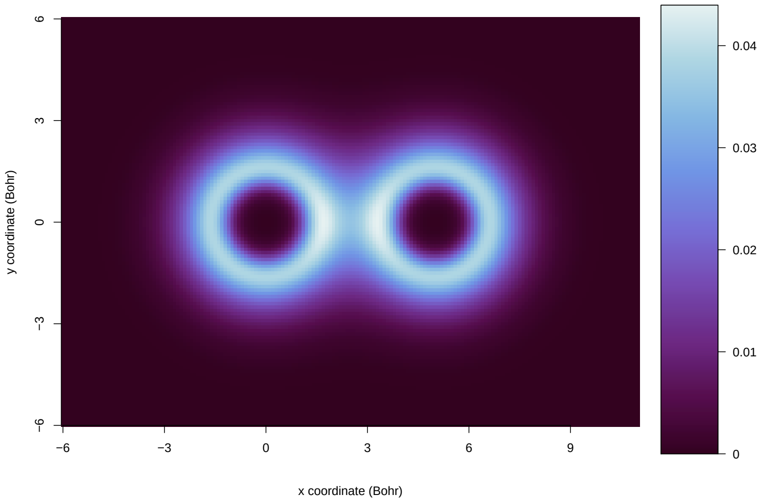

Figure 3.10: Aluminum dimer electron density - Electron density of an aluminum dimer. The scale is in electron per cubic bohr

We then compute binding energies of bigger aluminum clusters: $m \times m \times m$ faces centered cubic (FCC) crystals with m being between one and seven, so with a number of atoms of $M = 14$, 63, 172, 365, 666, 1099 and 1688 atoms. Their binding energy is computed for seven lattices between $a = 7.2\ a_e$ and $a = 8.4\ a_e$, figure 3.11 exposes the energy curves of the biggest crystals with their cubic regression. We have used this cubic regression to derived the crystals equilibrium binding energy and equilibrium bond length and have compared them with PROFESS results in table 3.3, we find a relative error of about 1% for the binding energies and for the equilibrium lattice.

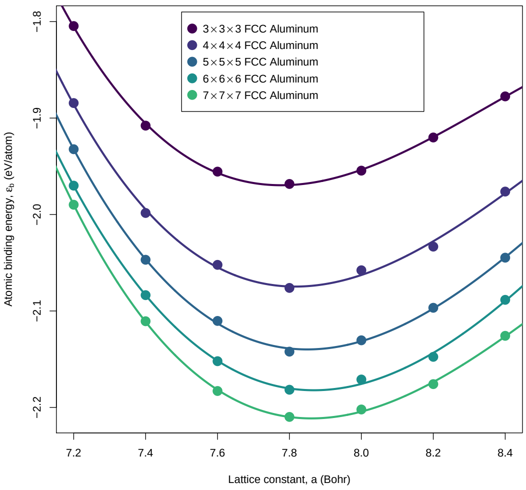

Figure 3.11: Binding Energies of aluminum crystals - Binding Energy of different aluminum FCC crystals of different sizes, so different atomic number M, with respect to the lattice constant. We also have drawn their cubic polynomial interpolation.

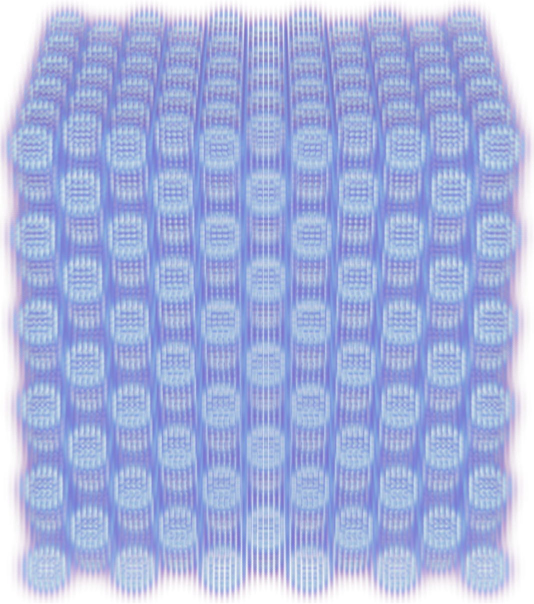

Figure 3.12: Electron density of an aluminum cluster - 3D representation of the electron density computed for a $M = 666$ FCC aluminum crystal of lattice $a = 8\ a_e$ with a grid node size $h = 0.5\ a_e$.

M	ε_b**(eV/at)**		**error(%)**	a_e**(Bohr)**		**error(%)**
	RS-FD	**PROFESS**		**RS-FD**	**PROFESS**	
14	-1.344	-1.357	0.96	7.616	7.528	1.17
172	-1.970	-1.988	0.91	7.777	7.706	0.92
365	-2.075	-2.092	0.81	7.816	7.733	1.07
666	-2.140	-2.157	0.79	7.849	7.750	1.28
1099	-2.182	-2.202	0.91	7.868	7.762	1.37
1688	-2.211	-2.236	1.12	7.861	7.771	1.16

Table 3.3: Binding energy (ε_b) and equilibrium lattice constant (a_e) for FCC clusters of M aluminum atoms

3.4.2.3 Aluminum bulk energy

Finally, we have derived the bulk cohesive energy $\mathcal{E}_{coh}$ of FCC aluminum with a lattice parameters $a = 8\ a_e$ from the binding energies of finite crystals of increasing size (M = 14, 63, 172, 666, 1099, 1688, and 2457). The binding energy of a finite crystal equals the binding energy of the full periodic crystal – the bulk cohesive energy – plus shifts due to the atoms on the sides, the edges, and the corners of the crystal. Hence, the binding energy $\mathcal{E}_b$ can be approximated as:

$$\mathcal{E}_b = \mathcal{E}_{coh} + a_{sides} M^{-\frac{1}{3}} + a_{edge} M^{-\frac{2}{3}} + a_{corner} M^{-1} \tag{3.52}$$

In figure 3.13 we show $\mathcal{E}_b(M^{-\frac{1}{3}})$ and its linear extrapolation. With our implementation, we find $\mathcal{E}_{coh} = -2.437$ eV/atom, like the value given by PROFESS for a periodic FCC crystal.

Our implementation computes structures predicted by OF-DFT properly, with errors around 1% for energies and bond lengths. Now we investigate the speed the electron density is computed and updated.

3.4.3 Non-local Kinetic Energy

We have implemented a Density-Dependent WGC kinetic energy with the method described in (67, 76) and also in the above section 3.2.6. As explained in (68), non-local kinetic energy functionals with a Density-Dependent kernel create instabilities for systems with high variations of electron density ρ because of the differences between the reference density used for the kernel ρ_0 and the local densities $\rho(\mathbf{r})$. We have

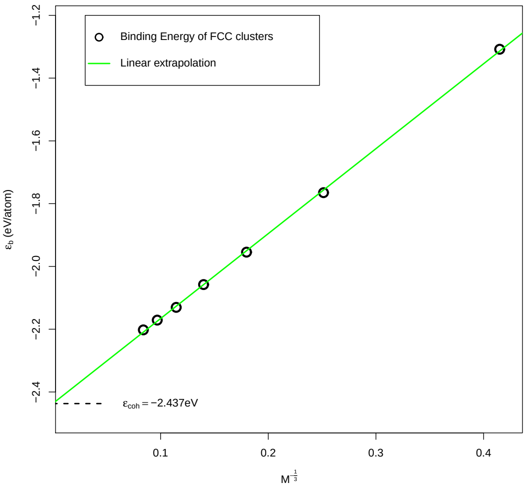

Figure 3.13: Extrapolation of Bulk Cohesive Energy - Binding Energy of aluminum FCC cluster of different sizes M, with respect to $M^{-\frac{1}{3}}$, and the extrapolation with a linear regression to $M^{-\frac{1}{3}} = 0$, the periodic FCC crystal.

experienced these instabilities in our method and have not pushed to implement a Density-Independent kernel. A Density-Independent kernel in Real-Space would have had a computational cost several times larger than in Fourier space like for PROFESS and even in the case of restrained dynamic we did find solutions to reduce computation significantly.

Nonetheless, to illustrate the shape of the WGC Density-Dependent kernel, we have drawn in figure 3.14 the shape of the two parts used to create a kind of potential. We take the equation 3.20, the kernel kinetic energy is described as

$$T_K^{\alpha,\beta}[\rho] = \int \rho^\alpha(\mathbf{r}) V^\alpha(\mathbf{r}) d\mathbf{r} \tag{3.53}$$

and as described in the former chapter, $V^\alpha(\mathbf{r})$ is the sum of solvable potential:

$$V^\alpha(\mathbf{r}) = \sum_1^\infty V_i^\alpha \tag{3.54}$$

Each V_i^α the solution of a Helmholtz equation. The four first terms are sufficient to describe V^α, and because V^α is purely real, each part of the sum has its exact conjugate in the sum, we have $V_1^\alpha = V_2^\alpha$ and $V_3^\alpha = V_4^\alpha$. Figure 3.14 shows a cross section of the two potentials V_1^α and V_3^α computed for a thin Gaussian electron density.

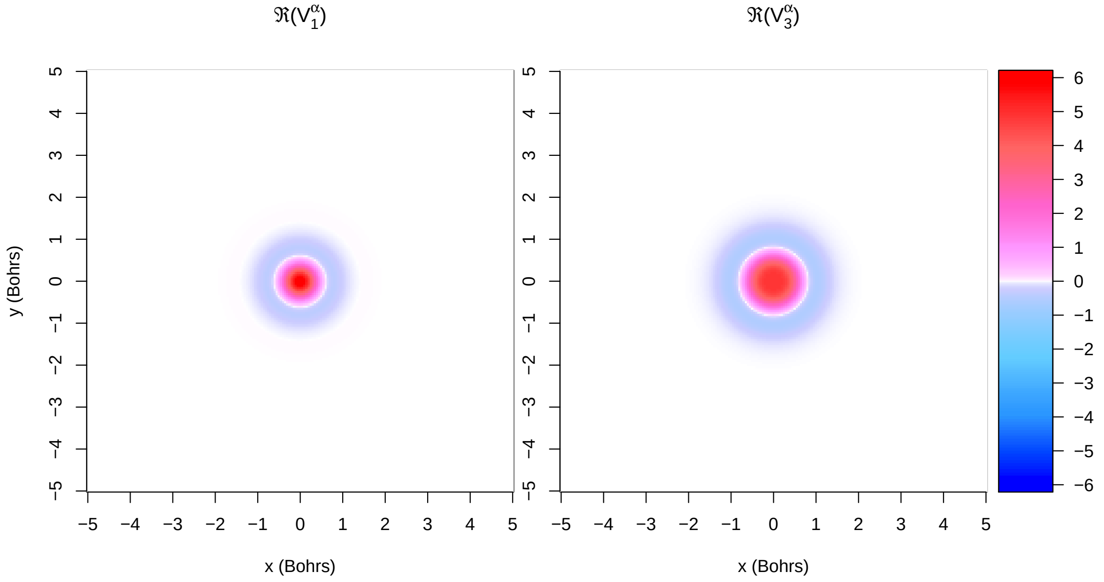

Figure 3.14: Two parts of the non-local kinetic potentials created by a thin Gaussian electron density. - To compute the kernel energy in real space, a kinetic potential V^{α} is computed. To calculate it in Real Space, we approximate V^{α} by an infinite sum of potentials that can each be solved using Helmholtz equations, a Partial Differential Equation usually used for wave equations. The four first terms of the infinite sum are sufficient to model properly the kinetic potential, and only two need to be solved. We have drawn their cross section here.

3.5 Convergence characteristics

Well parameterized, the Augmented Lagrangian produces good convergence characteristics. We show in Figure 3.15 the evolutions of the Lagrangian norm, energy and Lagrangian multiplier for the electron density computation of an aluminum atom. In this figure, the converged values are for the energy $E_{at,c} = -2.0809$ Hartree and for the Lagrangian multiplier $\lambda_c = 0.106457$. The straight lines in an log scale figure exposes that the algorithm has an exponential convergence.

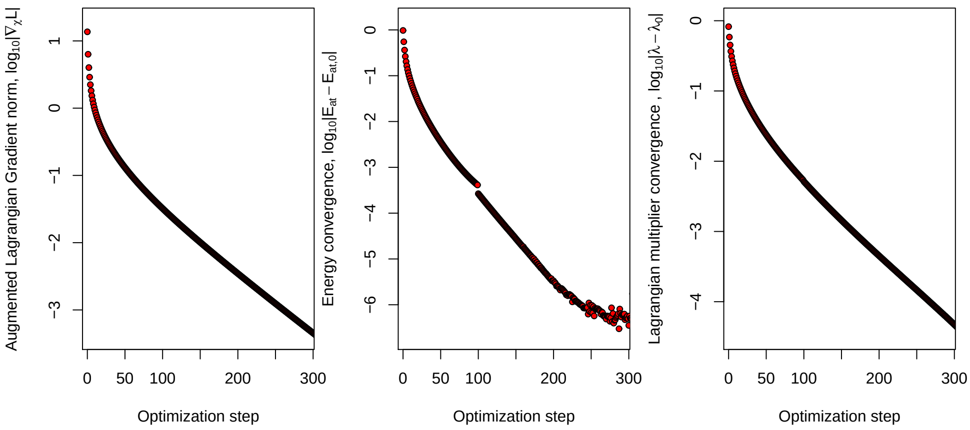

Figure 3.15: Convergence of Augmented Lagrangian Algorithm for one aluminum atom - On the left, $|\nabla_\chi \mathcal{L}_A[\chi, \lambda, \mu]|$ with respect to the optimization step, the convergence criteria is based on this values. In the middle the corresponding atomic energy $E_{at}[\chi]$, and on the right the corresponding Lagrangian multiplier λ.

For an aluminum FCC cluster of $M = 666$ atoms, convergence is the same, but slower. Figure 3.16 shows the 500 first steps of the electron density computation for this system. Here, $E_{at,c} = -2.1592$ Hartree and $\lambda_c = 0.117584$.

3.6 Computation time

We made series of measurements to present characteristic computational times of our OF-DFT method. Real-Space methods are not the fastest (74) for OF-DFT, but it has the advantage to be easily and efficiently parallelized (66, 73). We first measure computational times with respect to the number of cores used and the domain size,

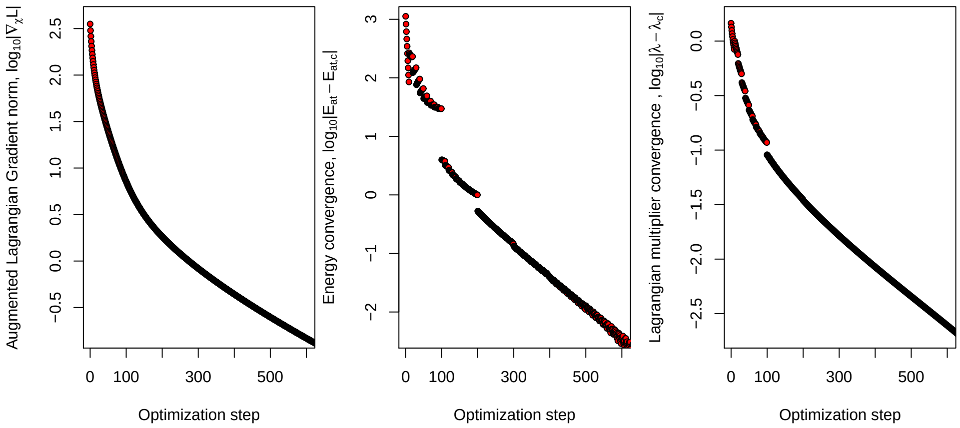

Figure 3.16: Convergence of Augmented Lagrangian Algorithm for an aluminum FCC cluster - On the left, $|\nabla_\chi \mathcal{L}_A[\chi, \lambda, \mu]|$ with respect to the optimization step, the convergence criteria is based on this values. In the middle the corresponding atomic energy $E_{at}[\chi]$, and on the right the corresponding Lagrangian multiplier λ.

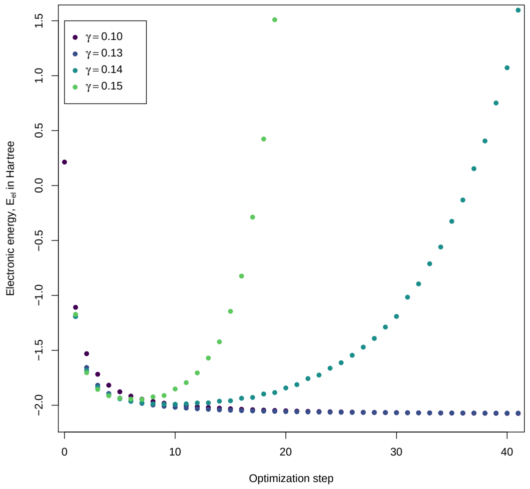

Figure 3.17: Augmented Lagrangian - Convergence of Energy - Electronic Energy $E_{el}[\chi]$ for several coefficient steps γ during the forty first steps of the electronic density computation of one aluminum atom. The converged value here is $E_{el,c} = -2.0809$ Hartree.

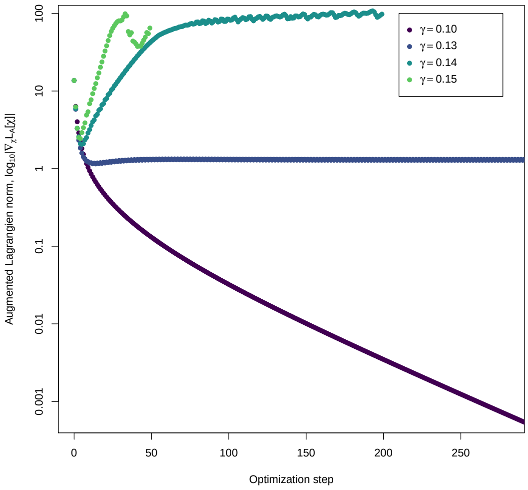

Figure 3.18: Augmented Lagrangian - Convergence of Lagrangian norm - Gradient norm of the Augmented Lagrangian $|\mathcal{L}_A[\chi, \lambda, \mu]|$ for several coefficient steps γ during the forty first steps of the electronic density computation of one aluminum atom. The convergence criteria is based on these values: the electronic density is considered converged when $|\nabla_\chi \mathcal{L}_A[\chi, \lambda, \mu]| \leq \delta$

then measures with respect to only the computational domain size and with a fixed machine: 20 Inter Xeon E5-2680 2.80 GHz bi-core processors.

3.6.1 Effect of parallelization

The computation of energies are parallelized. We tested to make cores work on different energy calculations (core 1 on E_{ext}, core 2 on T_s etc ...) but if the system is big enough (more than 10.000 nodes), to make all cores work on energy functionals one after the other is faster. Moreover, it is much more adapted to the computation of functionals with different complexity. Indeed, depending on the system size, the proportion of computational time dedicated to each energy evaluation varies, and the prediction of the optimal number of cores to associate with each energy functional is hazardous.

We have used the interface openMP(92) and have split statically the domain in p (number of cores available) compact sub-domains of same size to compute each energy.

3.6.1.1 Hartree Potential computation time

To be able to simulate free boundary conditions, and because we have decided to stay in real space, we compute the Hartree potential with a method inspired by adaptive multi-grid algorithms(93) coupled with a conjugate-gradient relaxation. As explained in (94), a full multi-grid method has a complexity in $\mathcal{O}(N)$, a conjugate-gradient (CG) typically in $\mathcal{O}(N^{\frac{5}{4}})$, and a method coupling both would have a complexity in between. In figure 3.19, we have drawn the computational time of the resolution of the Poisson equation of the same density, a Gaussian curve, with different grid precision $N = n^3$. The convergence condition here is $|e_r|_2 < 10^{-8}$. When using one core, we observe the quasi linear complexity of CG method. For more cores, the trend looks more linear because parallel computation overhead contribution is reduced with the size of our domain. The computation time does not scale like efficient Poisson solvers, FFT or block-cycle reduction(95, 96) but those methods are not suited to handle efficiently free boundary conditions.

Though our pseudo multi-grid - Conjugate-Gradient method does not provide the best performance, it allows to solve the free boundary conditions Poisson equation in an incremental and local way, with an electron density represented directly on a mesh, with a correct scalability with the domain size and a good scalability with the number of cores used. Faster ways exist to solve the free boundary condition Poisson equation that deserve to be tested and compared with ours, for example the work(97) is certainly more efficient and hence could be the next step of this work, maybe with an adaptation for small electron density increments.

3.6.1.2 Aluminum electron density computation time

We use a single aluminum atom, the simplest of benchmarks, to evaluate how parallelization affects the calculation times. We compute the atom's electron density in a computational domain of 30 a_e side size and different grid sizes to increase the number of grid nodes with our RS-FD code and with PROFESS. For each computation, we use one, four or twenty cores. The wall-clock times are drawn in figure 3.20, on the left our RS-FD program and on the right PROFESS, the scales are not the same.

For our RS-FD program, we use the same optimization step constant γ to focus on the effect of parallelization. With a bigger optimization step constant, the computation

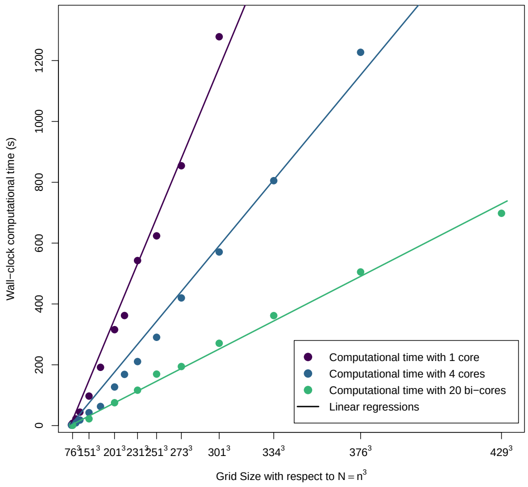

Figure 3.19: Wall clock time of Hartree Potential computation - Depending to the number of nodes of the grid and the number of cores used.

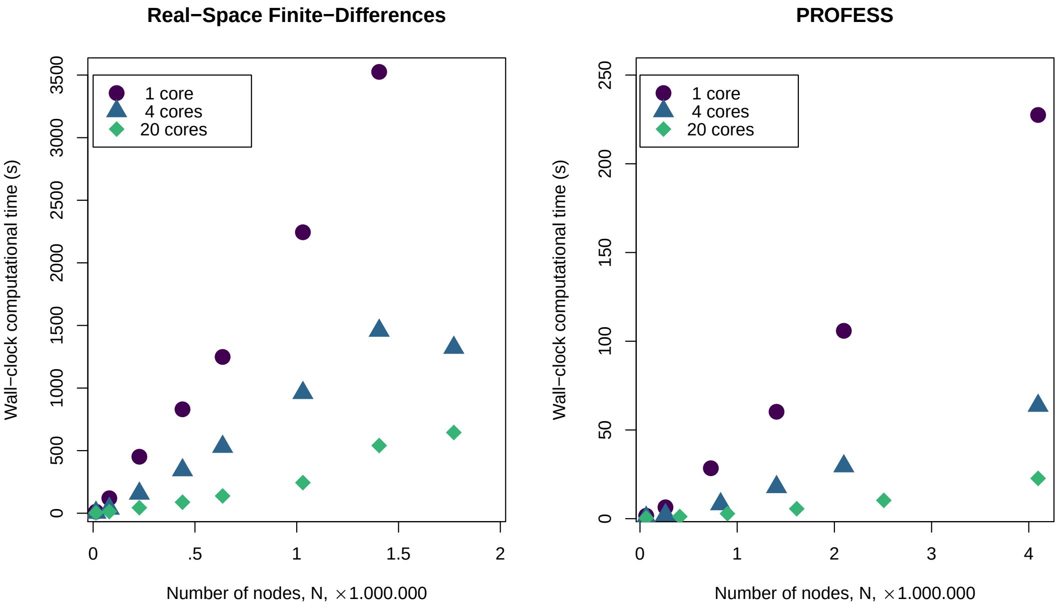

Figure 3.20: Wall clock time of electronic density computation of an single aluminum atom - On the left for our Real-Space Finite-Differences implementation and on on the right for PROFESS, a plane-wave based OF-DFT software, depending to the number of nodes of the grid and the number of cores used.

is faster for small grids, but grow unstable on big grids. With the optimization process fixed, the computation time scales linearly, using four cores divides the computational time by three on average, and using twenty cores by nine on average.

For PROFESS calculations, the discretization is determined by the plane-wave energy cutoff, E_{cut}. We used cutoffs of $E_{cut} = 200$, 600, 1200 , 1800 , 2400 and 3800 eV, from left to right on figure 3.20. They are not always vertically aligned because PROFESS can increase the number of nodes in some directions to use efficiently a parallel FFT.

Figure 3.20 also shows that our program, like other Real-Space implementations(74), does not compare with plane-wave implementation like PROFESS in term of computational time from simple initial density guesses.

3.6.2 Dependence on the domain size

The scalings of computation time with respect to each parameter taken separately are quasi linear, but we usually have to vary several parameters at the same time in practical modeling. When the number of atoms to model increases, the size of the domain increases too, and so the number of nodes. We have only studied the scaling between the number of nodes and the computation time. There is also a dependency with the number of atoms because an increase of the number of atoms leads to an equal increase of the number of electrons and so an increase of the average values of electric potential they create, which makes the Hartree Potential longer to compute and the optimization algorithm less stable. A growth of the computational domain with the corresponding number of atoms is hence worse what we observe when only one parameters were growing, and tends to a sub quadratic scaling. We draw in figure 3.21 the scaling of the computational time with respect to this "useful" growth of the domain size.

3.6.3 Computational time analysis

As Real-Space and Plane-Wave methods compute energy functionals differently, they do not use the same portion of resources for each part of the optimization process. In figure 3.22 we show how computational time is divided between energy functionals for our Real-Space method and PROFESS, and depending on the Kinetic energy used.

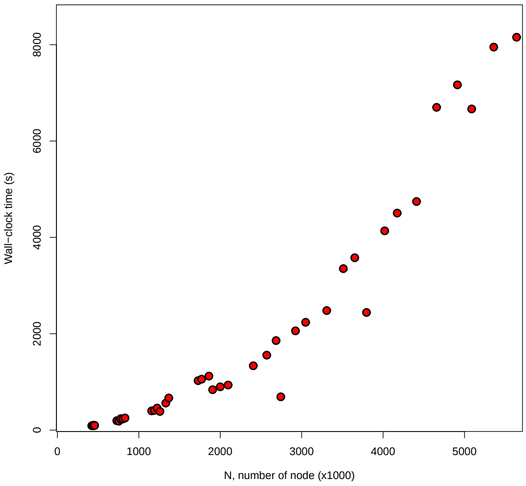

Figure 3.21: Wall clock time of electronic density computation - depending to the number of nodes of the grid when the grid is expended because the number of atoms increases. Due to the addition of several causes, increase of the number of electrons and increase of the number of nodes, the quasi-linear scaling is degraded and tends to a sub quadratic scaling.

The analysis has been done on a small system but is relevant whatever the system size(50).

For our Real-Space method, when using a semi-local kinetic energy, the $TF - \lambda vW$ kinetic energy, the computational bottleneck is the Hartree potential computation. Indeed, around 95 % of computational time is spend in the Poisson solver, like another RS-DF code (74). In comparison with a plane-wave method, for PROFESS the Hartree potential takes around 10 % of the computation time for the electronic energy (50). Using a Density-Independent Kernel for a non-local, the proportion of time spent on KEDF rise from less than 1 % to around 70 % for our RS method, and from 35 to 65 % for PROFESS.

The conclusion is that for Real-Space implementations and without non-local KEDF term, most of the computational time is lost for the resolution of the Hartree potential. We see in the next chapter how to break the computational bottleneck in dynamical simulations.

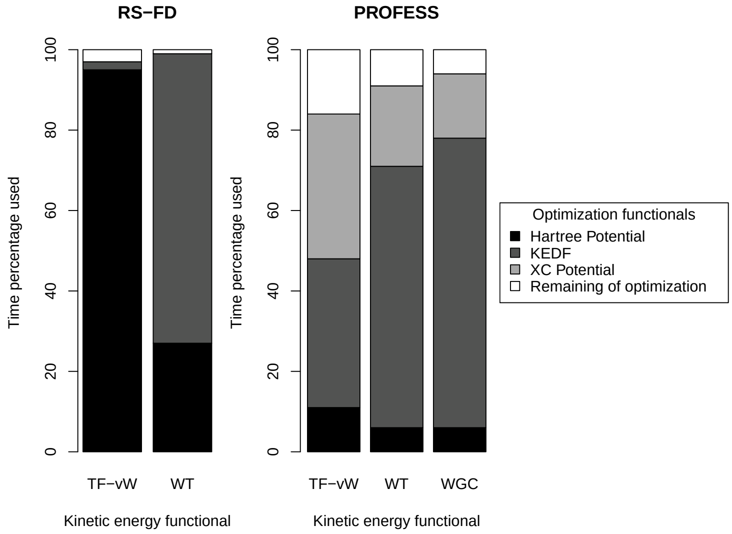

Figure 3.22: Use of computational time for OF-DFT - On the left for our RS-FD method, on the right for the reference plane-wave method PROFESS. For our RS-FD method, the XC potential computation is too fast to appear. The rest of optimization includes the external energy, the optimization algorithm and the projections on the constraints. Let us keep in mind those are percentages and not absolute times, $TF - \lambda vW$ KEDF are several times faster than WT and WGC KEDF and PROFESS is several times faster than our RS-DF method.

4

Incremental Orbital Free DFT

We saw that to compute from scratch an OF-DFT energy, reciprocal-space methods like PROFESS are more precise and efficient than finite-differences real-space ones. The purpose of our method is not to compute efficiently a solution from scratch, but to update one from a previous time step in a dynamical simulation. A real-space implementation allows to focus computations on specific parts of the domain, so in the case of restrained dynamical simulations, in which only the active particle moves at each time step, our method can update the electron density only where it matters: around the active particles. We show in this part that the computation time of our method scales almost as the number of particle moved, then we demonstrate the use and efficiency of such a method on a restrained dynamical simulation: an aluminum implantation.

4.1 Global Update

We consider a system of M particles indexed by $I \in [\![1, M]\!]$, at positions $\mathbf{R}_I$, of which we know ρ_0 the electron density at time t_0 given by OF-DFT. Lets imagine that from this system, the position of particle i changes by a small displacement $\mathbf{d}$. The simple approach to update the electron density is to modify the external potential V_{ext} created by this new set of positions:

$$V_{ext,t_1}(\mathbf{r}) = V_{ext,t_0}(\mathbf{r}) - V_{BLPS}(|\mathbf{r} - \mathbf{R}_i|) + V_{BLPS}(|\mathbf{r} - \mathbf{R}_i + \mathbf{d}|) \tag{4.1}$$

then optimize again the electron density on the whole space Ω, computing the Hartree potential V_H on Ω at each optimization increment. To sum up the classical density update:

1. update of external potential $V_{ext}(\mathbf{r})$
2. update of electron density ρ on the whole cell Ω

4.2 Incremental Update

4.2.1 Incremental update pipeline

To speed up the computation, instead of computing ρ directly on all Ω, we first compute an approximation of ρ only on a relevant part of Ω, a restrained area around the particle i called ω, then compute ρ on all Ω to erase the errors. Hence the incremental pipeline:

1. update of external potential $V_{ext}(\mathbf{r})$
2. restrained update of electron density on ω, a part of Ω
3. update of electron density on the whole cell Ω

We explain the new step, the restrained update, below.

4.2.2 Restrained Update

To perform the second step, the restrained update, we take into account that beyond a distance r_ω from the particle, the displacement is not felt and electron density does not change. In our model, this feature comes from the shapes of the two main potentials: for $|\mathbf{r}| \geq r_\omega$, V_{ext} and V_H are Coulomb potentials:

$$\forall \mathbf{r} \mid |\mathbf{r} - \mathbf{R}_i| > r_\omega : V_{ext}(\mathbf{r}) \simeq \frac{Q_i}{|\mathbf{r} - \mathbf{R}_i|} \text{ and } V_H(\mathbf{r}) \simeq -\frac{Q_i}{|\mathbf{r} - \mathbf{R}_i|} \tag{4.2}$$

with Q_i the ionic charge, the amount of valence electrons, of the particle i, and so the total electric potential is unchanged at the next time step:

$$\forall \mathbf{r} \mid |\mathbf{r} - \mathbf{R}_i| > r_\omega : V_{ext,t_1}(\mathbf{r}) + V_{H,t_1}(\mathbf{r}) \simeq V_{ext,t_0}(\mathbf{r}) + V_{H,t_0}(\mathbf{r}) \tag{4.3}$$

and the electron density stays the same. Hence, to update the electron density with a minimum of computation, the space Ω is split in two complementary areas, see figure 4.1:

1. ω, the loose density sub-area on which everything is updated, close from the moved particle.

2. the complementary of ω, $\Omega \backslash \omega$, on which the electron density is frozen and we update only the electric potentials V_H and V_{ext}.

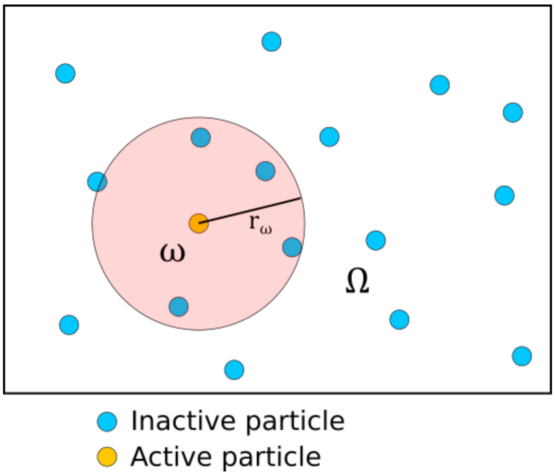

Figure 4.1: Creation of ω, the loose-density area - The loose density area is the union of all balls of radius r_ω centered around all active particles. This area is computed at each time step.

To discriminate between those two areas, we take our loose sub-area radius r_ω, and define ω as a ball of radius r_ω centered on the particle i and $\partial\omega$ as the border between the two areas:

$$\omega = \{\mathbf{r} \mid \mathbf{r} \in \mathbb{B}(\mathbf{R}_i, r_\omega)\} \tag{4.4}$$

$$\partial\omega = \{\mathbf{r} \mid \mathbf{r} \in \mathbb{S}(\mathbf{R}_i, r_\omega)\} \tag{4.5}$$

If, instead of one, P particles have moved at time t_1, the sub-area ω is the union of balls of radius r_ω centered on the moved particles, and $\partial\omega$ its frontier in Ω:

$$w = \left\{ \mathbf{r} \mid \mathbf{r} \in \bigcup_{I \in [\![1,P]\!]} \mathbb{B}(\mathbf{R}_I, r_\omega) \right\} \tag{4.6}$$

Before we update ρ on ω, the Hartree potential is updated approximately on the whole space. To do so, we take an approximation $V_H^{app}(r)$ so that we can have

$$V_{H,t_1}(\mathbf{r}) = V_{H,t_0}(\mathbf{r}) - V_H^{app}(|\mathbf{r} - \mathbf{R}_I|) + V_H^{app}(|\mathbf{r} - \mathbf{R}_I + \mathbf{d}|) \tag{4.7}$$

With $V_{H,t1}$ an approximation of the Hartree potential at time t_1. At long distance, for $|\mathbf{r} - \mathbf{R}_I| \geq r_\omega$, Hartree potential must equal the Coulomb potential. We chose the simplest V_H^{app}: a "chopped Coulomb potential", cut at a distance r_H from the center:

$$V_H^{app}(r) = \begin{cases} -\dfrac{Q_I}{r} & \text{for} \quad r > r_H \\ -\dfrac{Q_I}{r_H} & \text{for} \quad r \leq r_H \end{cases} \tag{4.8}$$

A small r_H, for example $2a_e$ like the characteristic size of the electron cloud, would be a better approximation of the final result with a fully relaxed electron density, but letting the Hartree potential follow the actual relaxing electron density is faster. Hence we choose a r_H slightly smaller than r_ω, so $V_H^{app}(r) = -\frac{Q_I}{r}$ for $r \geq r_\omega$. Thus, $V_{H,t1}$ is a precise approximation of V_H outside ω, and a wrong one inside.

Once V_{ext} and V_H have been respectively fully and partially updated, ρ and V_H are computed inside ω with Dirichlet boundary conditions on $\partial\omega$.

We call

- $Q_\omega = \int_\omega \rho^*(\mathbf{r}) d\mathbf{r}$ the electron quantity inside ω.
- $C_\omega = \{\chi : \mathbb{R}^3 \to \mathbb{R} \mid \int_\omega \chi^2(\mathbf{r}) d\mathbf{r} = Q_\omega\}$ the space of functions having Q_ω electrons inside ω
- ω_d the discrete version of ω, the set of indexes of nodes inside ω: $i \in \omega_d \iff \mathbf{r}(i) \in \omega$
- $\Gamma_\omega = \{\chi : \mathbb{N} \to \mathbb{R} \mid \sum_{i \in \omega_d} \chi^2(i) v_h = Q_\omega\}$ the discrete version of C_ω

To describe fully the incremental update: algorithm 3.

To update $V_H(\mathbf{r})|_\omega$ the Conjugate-Gradient would have required to work with tables of size N so we use the Gauss-Seidel method that allows to work with a table of size

Algorithm 3: Incremental OF-DFT update

```
Input: $p$ atoms moved, the electron density at the previous step
Output: an electron density with minimized energy at the current step
Update of the external potential $V_{ext}$ on $\Omega$;
Rough update of the Hartree potential $V_{ext}$ on $\Omega$;
/* Restrained Update                                                   */
while $|\nabla_\chi \mathcal{L}_A[\chi, \lambda, \mu]|_\omega| > \delta_\omega$ do
    $V_H(\mathbf{r})|_\omega \leftarrow \int_\omega \frac{\chi_k^2(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'$ : update the Hartree Potential on $\omega$ with border conditions on $\partial\omega$;
    $\lambda_\omega \leftarrow \frac{1}{2Q_\omega} \int_\omega \chi(\mathbf{r}) \frac{\delta E[\chi]}{\delta\chi} d\mathbf{r}$ : update the local Lagrangian multiplier ;
    $\partial\chi_k|_\omega \leftarrow \nabla_\chi \mathcal{L}_A[\chi, \lambda_\omega, \mu]|_\omega$ : compute the Lagrangian gradient on $\omega$;
    $\chi'_{k+1}|_\omega \leftarrow abs(\chi_k|_\omega - a\partial\chi_k|_\omega)$ : compute a new increment and project on $\chi \geq 0$ on $\omega$;
end
/* Global Update                                                       */
while $|\nabla_\chi \mathcal{L}_A[\chi, \lambda, \mu]| > \delta$ do
    $V_H(\mathbf{r}) \leftarrow \int_\Omega \frac{\chi_k^2(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'$ : update the Hartree Potential on $\Omega$ ;
    $\lambda \leftarrow \frac{1}{2Q} \int_\Omega \chi(\mathbf{r}) \frac{\delta E[\chi]}{\delta\chi} d\mathbf{r}$ : update the local Lagrangian multiplier ;
    $\partial\chi_k \leftarrow \nabla_\chi \mathcal{L}_A[\chi, \lambda_\omega, \mu]|_\omega$ : compute the Lagrangian gradient on $\Omega$;
    $\chi'_{k+1} \leftarrow abs(\chi_k - a\partial\chi_k)$ : compute a new increment and project on $\chi \geq 0$ on $\Omega$ ;
end
```

$|\mathit{\Gamma}_\omega|$. At equal size, the performance is decreased, but we aim at system where $|\omega|$ is much smaller than $\mathit{\Omega}$.

We have a loose density sub-area ω whose volume depends on particles moved – their number p and their positions $\mathbf{R}_I$ – and the radius r_ω. One of the goals of the following section will be to show that the computational time to update the electron density on ω increases with the sub-area's volume and the number of particles moved.

4.2.3 Computation time

This section is divided in two parts. We first look at the efficiency of the second step of the incremental update, the restrained update. We then measure the speedup gained with the entire incremental update.

For this, we first compute the electronic structure of a $6 \times 6 \times 6$ FCC aluminum crystal ($M = 1099$) with a margin of 15 a_e and a node size of 0.7 a_e then move p atoms, randomly chosen, of 0.3 Å in a random direction. Thus we have a "shaken" crystal whose electron density needs to be updated. Two different updates are performed: a global one done in a time t_0, and an incremental one in a time t_1. Both gives an atomic energy E^0_{at}. During the incremental update, we pause computations between the restrained update and the global one to record the atomic energy E^2_{at} and the computational time t_2. For this experiment $p \in \{1, 10, 100, 1000\}$ and $r_\omega \in \{4, 6, 8, 10, 12, 15\}$, in Bohrs, and for each combination of p and r_ω is tested five times, with different atoms moved, to have statistically significant measures of the errors and computational times.

We first look at the restrained update errors and update time differences, we compare E^2_{at} to E^0_{at} and t_2 to t_0. After the second step, ρ has been updated only inside ω and V_H has been approximately updated outside ω.

Figure 4.2 shows the evolution of relative errors $\mathit{\Delta} E^2_{at} = \frac{E^0_{at} - E^2_{at}}{E^0_{at}}$ when the loose density area radius r_ω increases. At $r_\omega = 4a_e$ the errors reach 0.5% and for $r_\omega \geq 8a_e$ they do not exceed 0.03%. The radial distribution of a lone aluminum atom's electron cloud is also drawn to emphasize that this limit of $8a_e$ corresponds roughly to the distance from the nucleus at which the atom's effect on the global electron density vanishes.

In figures 4.3 and 4.4 are drawn the speedups $\frac{t_0}{t_2}$ with respect to r_ω and to the relative volume of the loose density area used for the incremental update $v = \frac{|\omega|}{|\mathit{\Omega}|}$, $|\omega|$ and $|\mathit{\Omega}|$ being the volumes of respectively the loose density area and of the whole

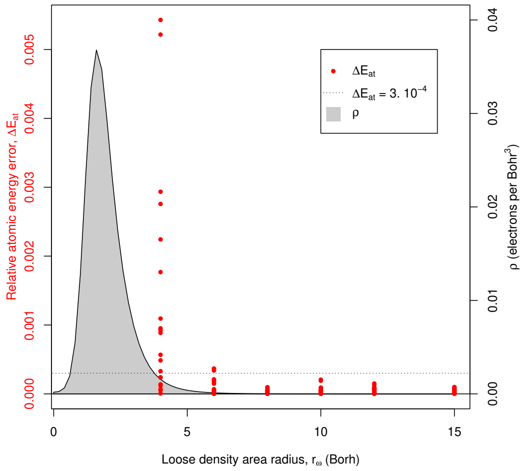

Figure 4.2: Errors of the restrained update - Relative error of a restrained update with respect to the loose density area radius r_ω, in Angström, and in grey the radial electron density of a lone aluminum atom.

area. On this example, we have a maximum 700-fold speedup for $(p, r_\omega) = (1, 4)$, then the speedup decreases as p or r_ω increases, until descending below a speedup of one: figure 4.3. A proper measure to estimate the update speed is the relative volume of the loose area: $v = \frac{|\omega|}{|\Omega|}$. The linear regression between the relative restrained update time $\frac{t_2}{t_0}$ and the relative loose area volume v gives $\frac{t_2}{t_0} = 2.01 \times v - 0.05$ with a correlation coefficient of $R^2 = 0.942$, showing the relative computational time is directly linked to the volume of the updated area. We have drawn in figure 4.4 the data from our simulations and the linear regression of the relation between speedup and v. For a relative volume above 0.5, a speedup cannot be guaranteed anymore.

We now study the computational time of the total incremental update. Speedup is represented in 4.5, the dots are the five tests means and the vertical bars their standard deviations. For $r_\omega \leq 6\ a_e$, the speedup of the restrained update, figure 4.4, is lost because, as the errors were important, figure 4.3, the correction is long. For $r_\omega \geq 10$ only few optimization increments are required to correct errors and we find almost the same speedup than with only the restrained update. Hence these bell-shaped curves with maxima for $r_\omega = 8\ a_e$ or $r_\omega = 10\ a_e$ depending on the number of atoms moved. We here have a maximum 20-fold speedup.

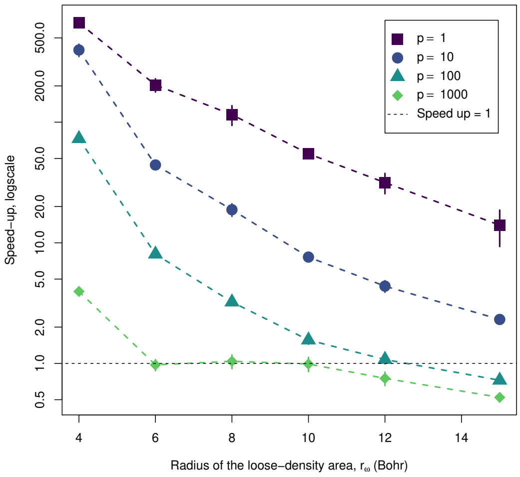

Figure 4.3: Restrained update speedup with respect to lose sub-area radius - Speedup, log scale, brought by a restrained update with respect to the loose density area radius r_ω, the colors are the number of atoms moved p.

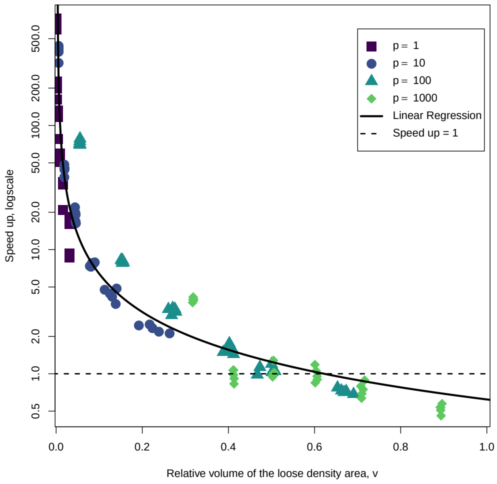

Figure 4.4: Restrained update speedup with respect to updated volume - Speedup, log scale, brought by a restrained update with respect to the updated area's volume ratio, $v = \frac{|\omega|}{|\Omega|}$, the colors are the number of atoms moved p. Also drawn in black the linear regression of the inverse of the speedup and the $Speedup \geq 1$ line, below which the restrained update is already slower than the global update. The time gained by the restrained update, compared to the global update, is inversely proportional to the volume updated.

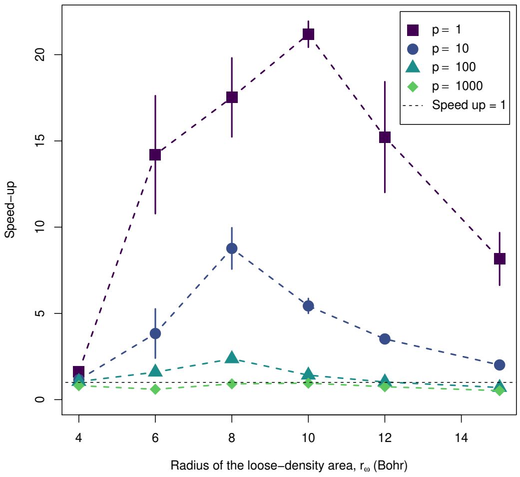

Figure 4.5: Incremental Update Speedup - Speedup brought by an incremental update – the three steps – compared to the global update, with respect to the loose density area radius, r_ω, in color, the number of atoms moved, p. The points are the means of the five measures, and the vertical bar their standard deviations. Also drawn the line below which the incremental update is slower than the global update.

By splitting the update in three stages – potentials update, restrained update then global update – we break the long process of moving the electron density increment by increment and updating the corresponding Hartree potential on the whole computational domain at each increment. The restrained update gives a proper approximation of ρ with a correct loose-density radius, and is fast when the domain is small, so when few particles have moved. The global update corrects efficiently the small errors left by the restrained update.

4.3 Aluminum impact simulation

In this section we demonstrate the efficiency of the incremental density update with a restrained dynamical model for particle simulations.

We simulate an atom implantation on an aluminum plate, we aim to be representative of doping by ion implantation (98): atoms are thrown toward a plate, cross several layers before being stopped and disrupt the crystal organization on their paths. The plate is an aluminum FCC crystal of 1944 atoms ($9 \times 6 \times 9$) and the impactor an aluminum atom placed 5 Å above the plate and thrown vertically toward it at 40 pm/fs. We chose a grid node size $h = 1$ a_e and margins of 15 a_e around the complex plate-impactor. Before the simulation starts, the plate geometry is relaxed to be at equilibrium. The implantation simulation is run with time steps $\partial t = 0.1$ fs and four sets of restraining parameters (ε^r and ε^f) to pinpoint their effect on the accuracy and speed of the simulation. To check if the simulation diverged from the reference, we follow the system's kinetic and OF-DFT energies during the whole implantation. During the simulation, the electron density is updated by an incremental OF-DFT algorithm with $r_\omega = 10$. The computer used for the simulation is a 20 Intel Xeon E5-2680 2.80 GHz bi-core processors, 32 GB of RAM, Windows 10 64-bits OS. Every computational step of the method has been parallelized with openMP(92).

We want restraining parameters to freeze efficiently the plate at the beginning of the simulation, for speedup, and to free all particles when the kinetic energy of the impactor will have dissipated into the plate, for accuracy. As the impactor initial kinetic energy is around 230 eV and we have a bit less than 2000 atoms, we count 0.1 eV per atom if the energy is equitably shared. We will test full-dynamics threshold of $\varepsilon^f = 5.10^{-1}$ eV, $\varepsilon^f = 5.10^{-2}$ eV, $\varepsilon^f = 5.10^{-3}$ eV and $\varepsilon^f = 5.10^{-5}$ eV with restrained-dynamic threshold $\varepsilon^r = 0.8\varepsilon^f$. We compare energies and computational times of those three runs with the reference, a full-dynamic simulation: $\varepsilon^r = \varepsilon^f = 0$ eV.

In figure 4.6 are shown five frames of the full-dynamic simulation, with $\varepsilon^f = \varepsilon^r = 0$ eV . They are colored with their displacement from the initial position, and the system is cut in the middle to make the relevant displacements appear. We see the propagation of the impact on five layers before the particle is stopped and its energy dissipated into other particles. That is the reference simulation, slow and "exact" in the sense that no error comes from a restrained dynamic. On full-dynamic and with our twenty cores computer, this simulation takes about two and a half days, 1600 steps of 130 seconds each, mainly for the update of electric potentials.

We then drew those five same frames but with restrained simulations, figure 4.7. From top to $\varepsilon^f = 5.10^{-1}$, $\varepsilon^f = 5.10^{-2}$, $\varepsilon^f = 5.10^{-3}$ and $\varepsilon^f = 5.10^{-5}$ eV, they are colored with their deviation from the reference position, the position of the same particle in the full-dynamic simulation. We see the increase of errors when the restrained pa-

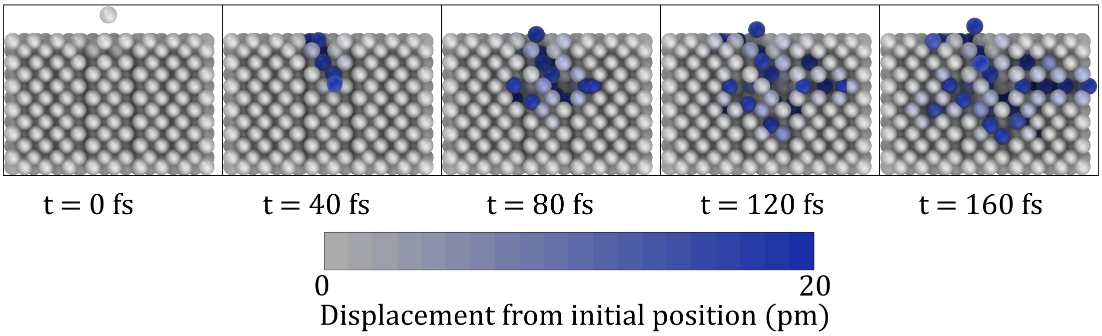

Figure 4.6: Aluminum Implantation Reference Simulation - Five frames of the aluminum implantation simulation without restrained dynamics. The particles are colored depending on their displacement from their initial position.

rameters become to big. For the restring parameters $\varepsilon^f = 5.10^{-3}$ the whole simulation took five hours, so a 10-fold speed up compared to the reference.

To clarify that the deviation increases with bigger restraining parameters, we display the maximum deviation Δq_{Max} and the root mean square deviation between the restrained simulations and the reference one. Figure 4.8 exposes the evolution of those measure of deviations during the simulations for five sets of restraining parameters, the four whose deviation were drawn in figure 4.7, plus a fifth one, smaller $(\varepsilon^f, \varepsilon^r) = (5.10^{-7}, 4.10^{-7})$ eV. The figure reveals below $(\varepsilon^f, \varepsilon^r) = (5.10^{-5}, 4.10^{-5})$ eV the deviations stagnates and we cannot recover the reference simulation by decreasing the restrains.

In figure 4.9, we drew the representative energies of four simulations, the full-dynamic one and three restrained-dynamic ones: on top the kinetic and below OF-DFT energy curves. We observe the energy curves shift away from the reference sooner when the restraining parameters are higher. Those curves show we can simulate properly this implantation with a restrained dynamic and incremental density updates, as long as restraining parameters are small enough (around $\varepsilon^f \leq 0.005$ eV).

As we saw in the previous sections, the update is divided in three stages: the update of the ions potential and the approximate update of the Hartree potential, scaling in $O(pN)$, the local density update, scaling in $O(|\omega|)$ and the global update scaling in $O(N)$. We hence record very fast step updates at the start of the simulation when p and $|\omega|$ are still small, and slower step updates when the simulation goes on, as more

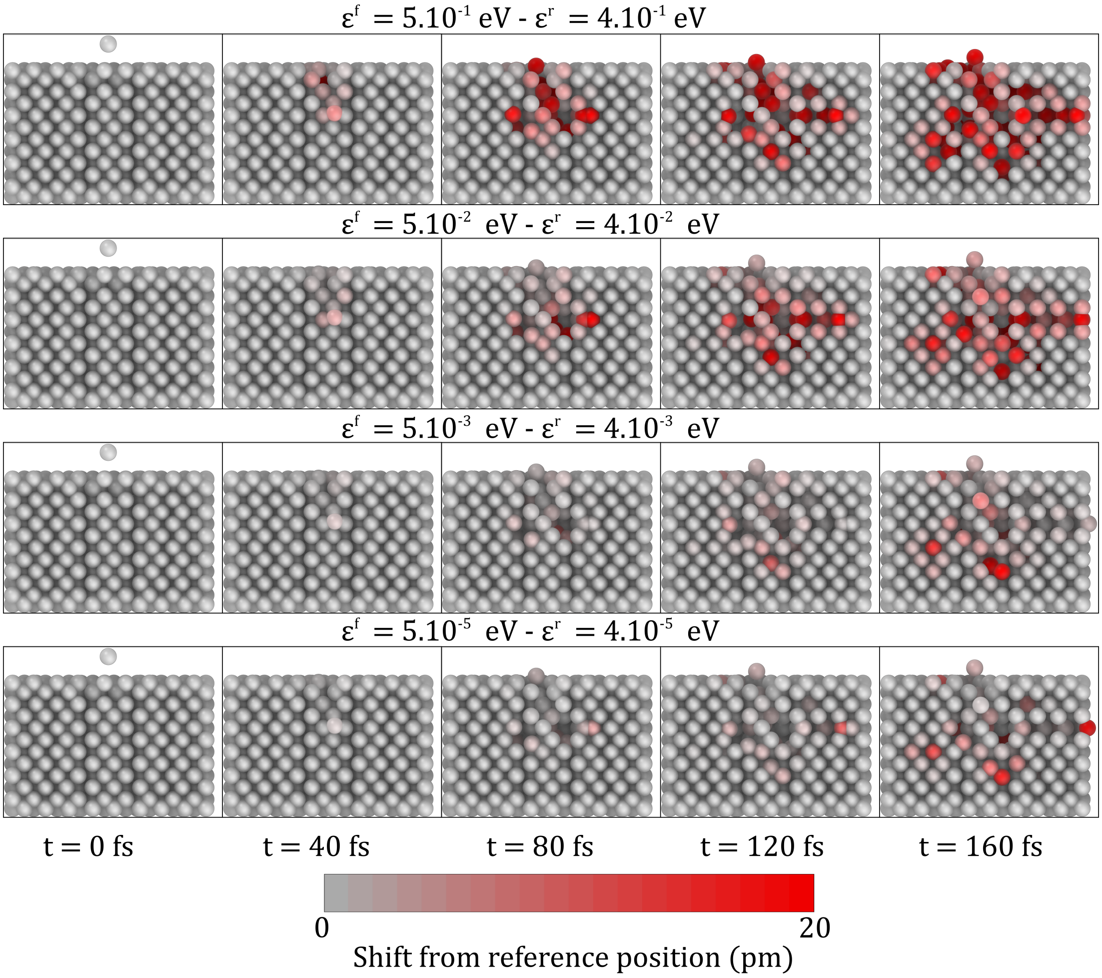

Figure 4.7: Aluminum Implantation Restrained Simulations - Five frames of the aluminum implantation simulation, with different restraining parameters and colored depending on the position shift compared to the reference. We have used, from top to bottom, $(\varepsilon^f, \varepsilon^r) = (0.5, 0.4)$ eV, $(\varepsilon^f, \varepsilon^r) = (0.05, 0.04)$ eV, $(\varepsilon^f, \varepsilon^r) = (0.005, 0.004)$ eV and $(\varepsilon^f, \varepsilon^r) = (5.10^{-5}, 4.10^{-5})$ eV.

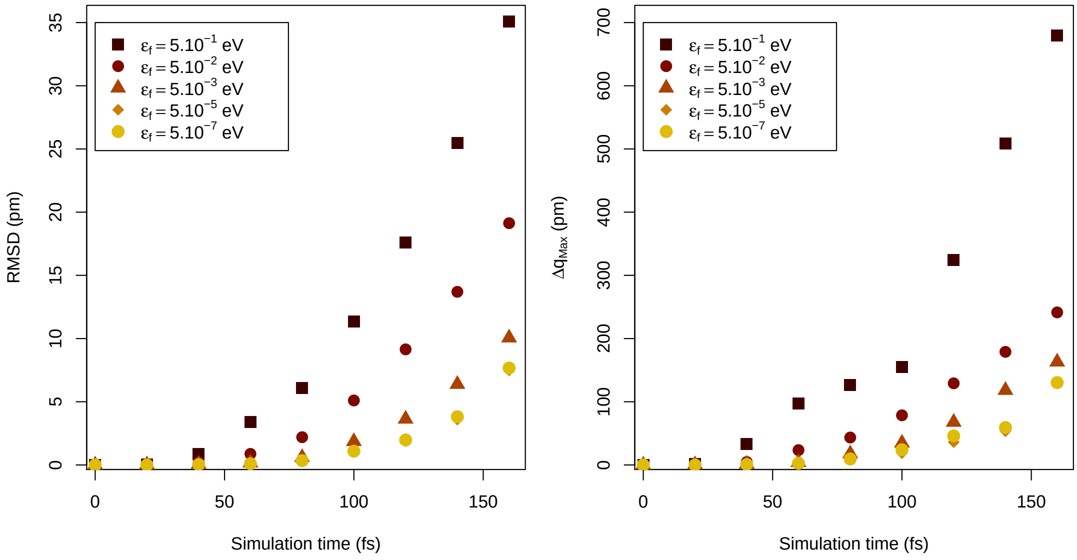

Figure 4.8: Deviations of a Restrained Simulation - The errors brought by a restrained dynamic simulations with OF-DFT, with different restraining parameters. On the left the RMSD between particles of the restrained simulation and the reference simulation, on the right the maximum shift between particles of restrained simulation and the reference simulation. We have drawn, the errors for restraining parameters $(\varepsilon^f, \varepsilon^r) = (0.5, 0.4)$ eV, $(\varepsilon^f, \varepsilon^r) = (0.05, 0.04)$ eV, $(\varepsilon^f, \varepsilon^r) = (0.005, 0.004)$, eV$(\varepsilon^f, \varepsilon^r) = (5.10^{-5}, 4.10^{-5})$ eV and $(\varepsilon^f, \varepsilon^r) = (5.10^{-7}, 4.10^{-7})$ eV.

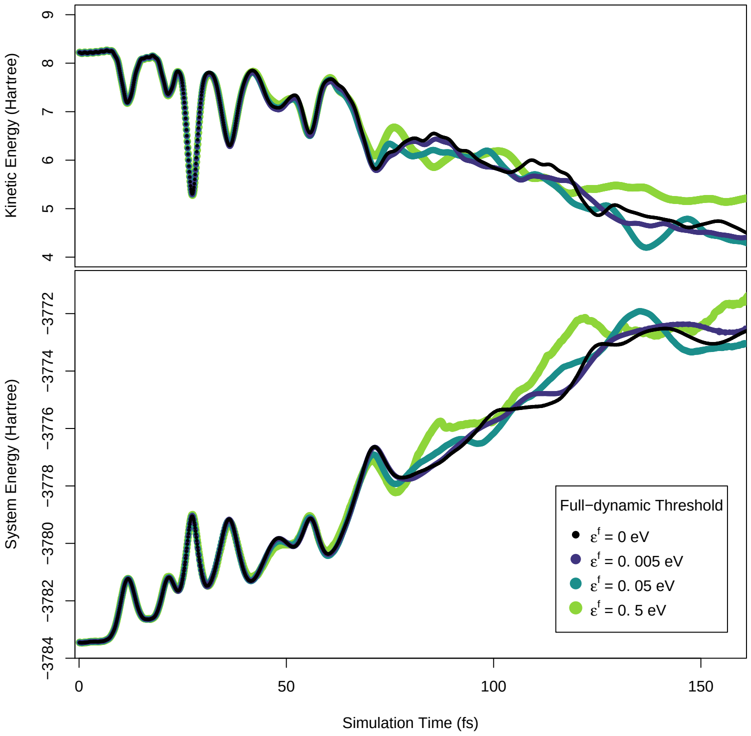

Figure 4.9: Aluminum Implantation Simulation Energies - Binding and Kinetic energy curves recording the first 160 fs of the implantation simulation with different restraining parameters. On top are drawn the kinetic energies and below the system OF-DFT energies. The black curves are the reference ones in which all particles are active. The red, blue and green ones have restraining parameters increasing and so proportions of active particle decreasing. We observe the energy curves splitting away from the reference sooner for the bigger restraining parameters $(\varepsilon^f, \varepsilon^r) = (0.05, 0.04)$ eV .

particles get disrupted and become active. The computational times of each time step have been recorded for the simulation with $\varepsilon^f = 0.005$ eV. They are drawn in figure 4.10 with respect to the simulation time, and in figure 4.11 with respect to the number of active particles. The simulation starts with computational time step of five seconds, with $\partial t = 0.1$ fs, it takes one minute to compute one femtosecond of simulation. At the end of the simulation, $t = 150$ fs or $p \geq 200$, computational time rose at 20 seconds per time step. For this example with a full-dynamic simulation ($\varepsilon^f = 0$ eV) a time step takes around 120 seconds, including 90 seconds for the update of potentials, and around 20 minutes without using the incremental update.

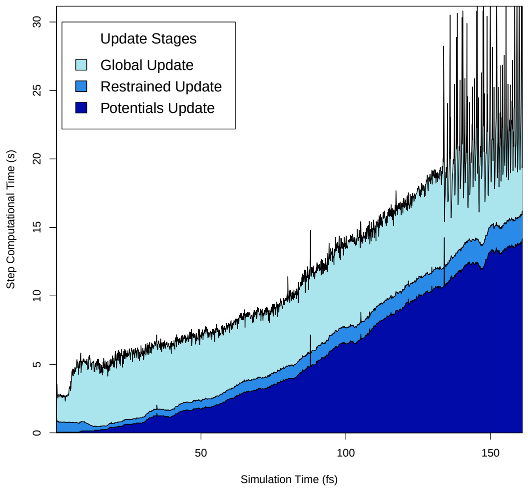

Figure 4.10: Evolution of Update Time with the advancement of simulation - Time required to update electron density at a particle simulation time for the implantation simulation with restraining parameters $(\varepsilon^f, \varepsilon^r) = (0.005, 0.004)$ eV. The computational times are decomposed in its three stages: update of potentials, restrained density update, and global density update. We see the update time increases as the simulation goes on for the number of active particles increases.

The overall update time achieved with this method is similar with what PROFESS does with a plane-wave method. On the same computer using the 20 cores and a Linux System, with a similar node size, here a kinetic energy cutoff of 200 eV, a similar convergence criteria and the same OF-DFT functional, the update of electron density of the first simulation, when only one particle has moved, step takes five seconds and

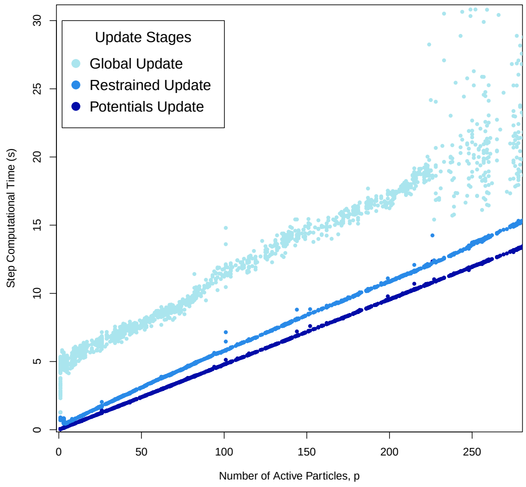

Figure 4.11: Evolution of Update Time with the number of active particles - Time required to update electron density depending on the number of active particles. The computational times are decomposed in its three stages: update of potentials, restrained density update, and global density update. We see the update time increases as the number of active particles increases.

a full update, when all particles have moved, takes 15 seconds. It is important to first take into account PROFESS relies on powerful C++ libraries like FFTW, LAPACK and Libxc that increase computational speed, then to consider the dependency of the speedup on the number of active particle: from the reference simulation $\varepsilon^f = 0$ eV, we achieve a five-fold speedup when less than 20% of particles are active, and a 20-fold speedup when less than 1% are.

By using our incremental update method and a restrained dynamic, we achieved a computational speed comparable to PROFESS with the same restrained dynamics.

4.4 Minimum Energy Path

At last, we show we can use our OF-DFT method to compute Minimum Energy Paths (MEP). Finding the MEP between two states in an energy landscape can be used in chemistry for example to study the enthalpy energy of a reaction, its energy barrier and its intermediate states. We compute here the displacement of a defect in an aluminum cluster. The energy landscape will be our OF-DFT solver with its incremental method, and the MEP will be computed with the Nudged Elastic Band (NEB) method (99).

NEB is an improvement of the Plain Elastic Band (PEB) method. In PEB, intermediate states are generated between the two extreme states, often the simplest ones along the strait line, then their geometry is relaxed with the energy function chosen (here OF-DFT) and with an elastic potential that prevents the successive states from getting away from each other. The extreme states are fixed and serve as anchors to the band of intermediate states.

If we call $\{\mathbf{R}_i\}_i$ the $P+1$ states, with $\mathbf{R}_0$ the initial state and $\mathbf{R}_P$ the final one, and $V(\mathbf{R})$ the energy landscape function, the PEB method minimizes the objective function S_{PEB} defined as:

$$S_{PEB}(\mathbf{R}_1,\dots,\mathbf{R}_{P-1}) = \sum_{i=0}^{P} V(\mathbf{R}_i) + \sum_{i=1}^{P} \frac{Pk}{2}(\mathbf{R}_i - \mathbf{R}_{i-1})^2 \tag{4.9}$$

k the spring constant, the P in the spring tension formula permits to compute paths that do not depends on the number of intermediate states. As explained in (99), this method has the disadvantages to cut inside the turns of the real MEP and, if there is a saddle point, to move away the intermediate states from it. The Nudged Elastic Band method corrects those two problems by projecting the two forces on different direction. The elastic force on the state i, $\mathbf{F}_i^s$ is projected on the direction of the band so that the path does not cut the turns. The energy functional force, $-\nabla E_{Tot}\mathbf{R}_i$, is projected on the space orthonormal to the band to prevent the states to move away from saddle points. At each optimization step, the force used is:

$$\mathbf{F}_i = -\nabla V(\mathbf{R}_i)|_\perp + \mathbf{F}_i^s|_\parallel \tag{4.10}$$

In this experiment, the landscape energy function is $E_{Tot}(\mathbf{R})$, the energy function that gives the OF-DFT ground-state energy of the particle system $\mathbf{R}$.

We study the displacement of a defect in the [110] direction of a FCC aluminum cluster. In a FCC crystal, [110] neighbors are the closest neighbors. The aluminum cluster is a $7 \times 7 \times 7$ FCC aluminum crystal, $M = 1098$ of lattice constant $a = 7.8\ a_e$. For OF-DFT computations, the domain has grid nodes of size $h = 0.7\ a_e$ and a margin of $12\ a_e$, a bit reduced because the surface atoms are not relevant here.

We create the initial state $\mathbf{R}_0$ with the defect in its initial position, freeze the movement of surface atoms to simulate an infinite crystal and relax the inside geometry.

We do the same with the final state $\mathbf{R}_P$ with the defect in its final position. We then create 8 intermediate states as linear combination of the extreme states, so $P = 9$. Those states relaxed with the formula 4.10 and a steepest descent method. We took for this experiment a spring constant of $k = 10\ nN/pm$.

Figure 4.12 shows the result of the optimization. The image has been cut orthogonally to the [111] direction to see the particle's movement and the shift of its neighbors. We can see the hexagonal packing of a FCC crystal in the [111] direction. The images are colored with respect to the deviation between their initial position, the linear combination, and the final one, once the position stabilized. The particles on the way of the moving particle are the one most affected.

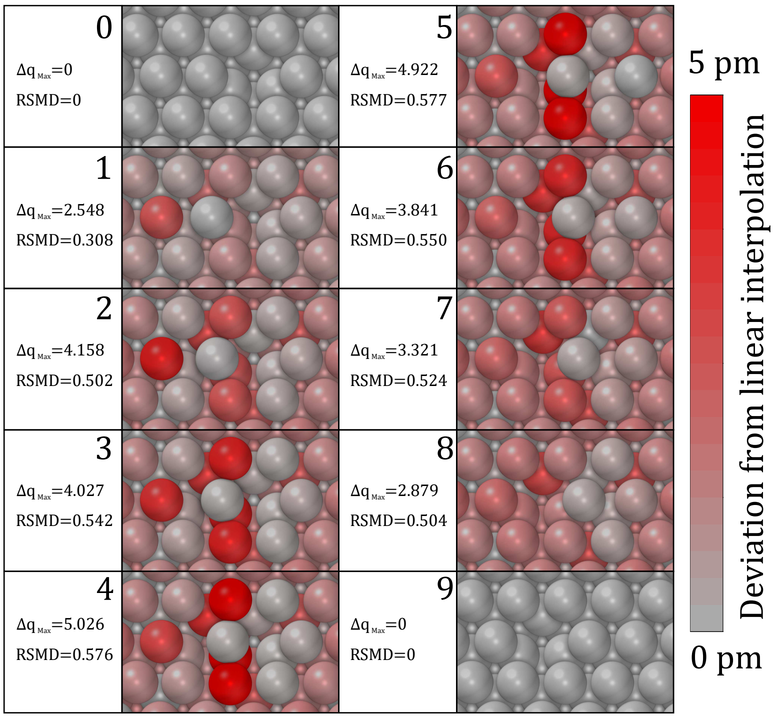

Figure 4.12: MEP of a defect displacement in aluminum FCC - The view is a [111] cut of a FCC crystal with a defect displacement. The particles are colored depending on the deviation from their initial position hat is a linear combination of the two extreme states. The extreme states, 0 and 9, are not updated so are in gray.

This experiment is another possible use of our incremental RS-FD OF-DFT solver. As the defect displacement involved a small area of a large crystal, computation costs can be reduced by focusing the update of electron density on relevant parts of the domain. We have shown that with a restrained dynamic and an incremental scheme for OF-DFT, we can achieve computational efficiency comparable with plane-wave methods, the current state of the art in OF-DFT.

5

Conclusion

5.1 Incremental RS-DF OF-DFT

We have demonstrated in this work that the computational speed of OF-DFT simulations can be improved by using an adaptively restrained dynamical model. The OF-DFT method chosen, RS-FD, is still several order of magnitude slower than plane-wave OF-DFT methods, but has other advantages related to the adaptability of a RS-DF computational domain: choice of boundary conditions, computational domain shape, easy parallelization etc... Now we have shown that RS-DF OF-DFT can be as fast as plane-wave OF-DFT with a modified dynamic and a system that make restrained dynamics efficient. As we were exposed to the limitations of a free boundary conditions RS-DF scheme, no efficient non-local KEDF and slow Hartree potential, we have outperformed plane-wave only on few systems examples. Other improvements are required to make RS OF-DFT a reference.

5.2 Perspectives

We cite here several ideas to complete or improve RS OF-DFT methods we did not have time to test.

Better KEDF in Real Space, non periodic system Without a proper Kinetic energy functional, the prediction abilities of OF-DFT are greatly reduced compared to KS-DFT, covalent bonds are not simulated so OF-DFT is limited to model materials with metallic bonds. KE functionals that exhibits better properties already exist, we have seen some of them in chapter 2.6.2 and have exposed the difficulties brought by implementing them in a RS-DE scheme in chapter 3.2.6. The solution could come from accurate semi-local KEDF. A recent machine-learned KEDF functionals (59) surpasses in accuracy all the other semi-local KEDF and improves their ability to model valence and bonding electrons. Another semi-local KEDF (100) is tuned to equals in accuracy some non-local KEDF on specific elements. Non-local KEDF accuracy is in general still unmatched by semi-local KEDF, but for RS-schemes the computational cost of a semi-local functionals is so small compared to non-local ones that they might be the solution of efficient KEDF in OF-DFT.

More optimization algorithms The optimization method we have used, an augmented Lagrangian backed by an efficient update of the Lagrangian multiplier, is reasonably fast, see section 3.6. Nonetheless, the update of the whole Lagrangian $\mathcal{L}_A[\chi, \lambda, \mu]$ is done with a steepest descent method and has to be improved. The speed up the optimization process will not be solved only by using a more efficient algorithm, but first by improving the stability of the RS-FD energy functionals. Indeed, because of the incremental methods we have used for computing the Hartree potential and the non-local kinetic energy, and because the convergence threshold of those methods can not be null, one electron density does not correspond to exactly one energy and one energy derivative, and the solution found depends on the initial guess of the optimization process. All the efficient minimization algorithms we have tested had difficulties to converge with a "noisy" OF-DFT energy functional.

Multilevel Grid A popular alternative to Fourier-space to compute electric repulsion, electrons or nuclei repulsion, is the multi-grid approach. The same method could be tested with OF-DFT electron density calculations. The multi-grid methods use several grids to represent a function, a density or a potential, to model but with different thinness. Usually the first grid is a very coarse representation of each grid is twice as thin as its predecessor. On one hand there is a gain in speed because the function to compute is first approximated on the first grid, the coarser one, then the approximation is passed on the second grid, twice as thin, to have a result more precise. On a second hand, it allows to update the function incrementally, either by selecting where the update will focus of by having an clever update. In section 3.2.3, we have done something alike to compute the Hartree potential, with three grids having nodes three times as thin as its predecessor, but with the same number of nodes, the aim was to model a volume model large enough to have monopole boundary conditions.

Most interesting particle systems are not homogeneous, their electron density is made of areas with fast-varying density, where high precision is required, and slow-varying density. One could think of having an adaptive grid: thin where density variations are most important, coarser elsewhere.

Fast computation of nuclei repulsion energy In this model, the computation of the nuclei repulsion energy $E_{II} = \sum_{I<J} \frac{Q_I Q_J}{|\mathbf{r}_I - \mathbf{r}_I|}$ is done by brute-force computation. The complexity of a brute-force computation of E_{II} is $\mathcal{O}(N^2)$ so that will become an issue in very large model or in periodic computational domains, but the focus of this thesis was the OF-DFT part and we did not reach a number of atoms sufficient for the nuclei repulsion energy to become the bottleneck of the computation efforts. Nonetheless, several efficient algorithms can be found to compute efficiently E_{II}. Some use Fourier transform, others multi-grid methods and could fit with the second point of the perspectives.

Wavelet scheme Another basis we aimed to try is a wavelet (101) basis. It would have some of the advantages of a multi-grid method (adaptivity of the grid and speed) and would focus the space function of search on relevant space. Like a plane-wave basis that limits the scope of represented functions and hence allows a good representation of smooth functions with fewer basis vectors, wavelets would model electronic densities with the same accuracy but less basis vector. A Daubechies wavelet basis (102) is already used for KS-DFT in the BigDFT code (103).

5.3 Deep learning in ab-initio simulations

We have seen that some parts of the DFT functional, here(59) the KEDF of OF-DFT, can be improved by machine learning. In a more general way, deep learning methods have reached ab-initio simulations. Neural Networks can now predict with precision ground state energies of an electron in various electric potentials (104) or help to solve Kohn-Sham equations (105). A emerging method is the use of DFT results as data set to train Neural Networks to model energy potentials (106) or directly forces(26). Neural networks have produced incredible results in the recent years, from beating humans at games humans were though unbeatable (107, 108) to solving the long last standing problem of protein folding (109), there is little doubt they will produce results in OF-DFT and in quantum physic modeling and simulations.

6

Appendices

6.1 Gauss Legendre quadrature coefficients

To approximate the value of integrals, in particular to compute electrical potential on a grid, we have used the Gauss-Legendre quadrature. The approximation of the integral of $f(t)$ on the segment $[-1,1]$ with a n^{th} degree Gaussian quadrature is expressed as

$$\int_{-1}^{1} f(t)dt = \sum_{i\in[1,n]} \omega_i^n f(x_i^n) \tag{6.1}$$

with $\{x_i^n\}_i$ the nodes and $\{\omega_i^n\}_i$ the weights of the quadrature.

For the Gauss-Legendre quadrature, the nodes are the roots of n^{th} Legendre polynomial $P_n(x)$, and the weights are given by :

$$\omega_i^n = \frac{2}{(1-x_{n,i}^2)[P'_n(x_{n,i})]^2} \tag{6.2}$$

We recall one formula of the n^{th} Legendre polynomial, the Rodrigues' formula :

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}(x^2-1)^n \tag{6.3}$$

For the first coefficients, among them those of the 3^{rd} order quadrature, we refer to table 6.1.

6.2 Laplace's Spherical Harmonic

In section 2.5.4 we have evoked the Spherical Harmonics basis function, noted $Y_{lm}(\theta,\phi)$. They are a set of orthonormal functions defined on the 3D sphere :

number of points n	position, x_i	weight ω_i
1	0	2
2	$\pm\sqrt{\frac{1}{3}}$	1
3	0 $\pm\sqrt{\frac{3}{5}}$	$\frac{8}{9}$ $\frac{5}{9}$
4	$\pm\sqrt{\frac{3}{7}-\frac{2}{7}\sqrt{\frac{6}{5}}}$ $\pm\sqrt{\frac{3}{7}+\frac{2}{7}\sqrt{\frac{6}{5}}}$	$\frac{18+\sqrt{30}}{36}$ $\frac{18-\sqrt{30}}{36}$
5	0 $\pm\frac{1}{3}\sqrt{5-2\sqrt{\frac{10}{7}}}$ $\pm\frac{1}{3}\sqrt{5+2\sqrt{\frac{10}{7}}}$	$\frac{128}{225}$ $\frac{322+13\sqrt{70}}{900}$ $\frac{322-13\sqrt{70}}{900}$

Table 6.1: Coefficients of low-order Gauss Legendre quadratures

$$\theta \in [0, \pi] \tag{6.4}$$

$$\phi \in [0, 2\pi] \tag{6.5}$$

$$\langle Y_{lm} | Y_{l'm'} \rangle = \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} Y_{lm} Y_{l'm'} \sin\theta d\theta d\phi = \delta_{ll'} \delta_{mm'} \tag{6.6}$$

They appear as the non-radial part of the solution of Laplacian equation in Spherical coordinates :

$$\Delta f(r, \theta, \phi) = 0 \tag{6.7}$$

and in the solutions of the Schrödinger equation in a Coulomb potential $V(\mathbf{r} = -\frac{Q}{r})$, the so-called "hydrogen-like atomic orbitals". For the details on the calculus of Hydrogen orbitals: (110).

Laplace's Spherical Harmonics can be written with two integers m and l, $m \in [\![-l, l]\!]$

$$Y_{lm}(\theta, \phi) = N_{lm} e^{im\phi} P_l^m(\cos(\theta)) \tag{6.8}$$

m/l	0	1	2
-2			$\frac{1}{4}\sqrt{\frac{15}{2\pi}}\sin^2\theta e^{-i2\phi}$
-1		$\frac{1}{2}\sqrt{\frac{3}{2\pi}}\sin\theta e^{-i\phi}$	$\frac{1}{2}\sqrt{\frac{1}{\pi}}\sin\theta\cos\theta e^{-i\phi}$
0	$\frac{1}{2}\sqrt{\frac{1}{\pi}}$	$\frac{1}{2}\sqrt{\frac{3}{\pi}}\cos\theta$	$\frac{1}{4}\sqrt{\frac{5}{\pi}}(3\cos^2\theta - 1)$
1		$-\frac{1}{2}\sqrt{\frac{3}{2\pi}}\sin\theta e^{i\phi}$	$-\frac{1}{2}\sqrt{\frac{1}{\pi}}\sin\theta\cos\theta e^{i\phi}$
2			$\frac{1}{4}\sqrt{\frac{15}{2\pi}}\sin^2\theta e^{i2\phi}$

Table 6.2: First Laplace's Spherical Harmonics

with N_{lm} a normalization constant and $P_l^m(x)$ the associated Legendre Polynomial, a more general form of Legendre polynomials $P_l(x)$ seen above 6.3. In particular, we have $P_l^0(x) = P_l(x)$.

We write here the associated Legendre polynomial Rodrigues' formula, and note that is not a proper polynomial when m is odd :

$$P_l^m(x) = \frac{(-1)^m}{2^l l!}(1-x^2)^{\frac{m}{2}}\frac{d^l}{dx^l}(x^2-1)^l \tag{6.9}$$

In table 6.2 are written in full the First Spherical Harmonics ($l \leq 2$).

References

[1] David J Griffiths and Darrell F Schroeter. *Introduction to quantum mechanics.* Cambridge University Press, 2018. 3

[2] Hiroyuki Nakashima and Hiroshi Nakatsuji. **Solving the Schrödinger equation for helium atom and its isoelectronic ions with the free iterative complement interaction (ICI) method**. *The Journal of chemical physics*, **127**(22):224104, 2007. 3, 13

[3] Markus Reiher and Alexander Wolf. *Relativistic quantum chemistry: the fundamental theory of molecular science.* John Wiley & Sons, 2014. 3

[4] Paul AM Dirac. **The Lagrangian in quantum mechanics**. In *Feynman's Thesis—A New Approach To Quantum Theory*, pages 111–119. World Scientific, 2005. 4

[5] Vidal Alonso, Salvatore De Vincenzo, and Luis González-Díaz. **Ehrenfest's theorem and Bohm's quantum potential in a "one-dimensional box"**. *Physics Letters A*, **287**(1-2):23–30, 2001. 4

[6] Valia Allori and Nino Zanghì. **On the classical limit of quantum mechanics**. *Foundations of Physics*, **39**(1):20–32, 2009. 4

[7] Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*, **1**. Elsevier, 2001. 4, 8

[8] Vladimir Igorevich Arnol'd. *Mathematical methods of classical mechanics*, **60**. Springer Science & Business Media, 2013. 5

[9] Herbert Goldstein, Charles Poole, and John Safko. **Classical mechanics**, 2002. 5

[10] Etienne Forest and Ronald D Ruth. **Fourth-order symplectic integration**. *Physica D: Nonlinear Phenomena*, **43**(1):105–117, 1990. 6

[11] Svetlana Artemova and Stephane Redon. **Adaptively Restrained Particle Simulations**. *Physical Review Letters*, **109**(19):190201:1–5, November 2012. 7, 17

[12] Svetlana Artemova. *Adaptive algorithms for molecular simulation.* Theses, Université de Grenoble, May 2012. 7

[13] Stephane Redon, Gabriel Stoltz, and Zofia Trstanova. **Error Analysis of Modified Langevin Dynamics**. *Journal of Statistical Physics*, **164**(4):735–771, August 2016. 7

[14] Zofia Trstanova. *Mathematical and Algorithmic Analysis of Modified Langevin Dynamics.* Theses, Université Grenoble Alpes, November 2016. 7

[15] Maël Bosson, Caroline Richard, Antoine Plet, Sergei Grudinin, and Stephane Redon. **Interactive quantum chemistry: A divide-and-conquer ASED-MO method**. *Journal of Computational Chemistry*, **33**(7):779–790, March 2012. 8, 18

[16] Maël Bosson, Sergei Grudinin, and Stephane Redon. **Block-Adaptive Quantum Mechanics: An Adaptive Divide-and-Conquer Approach to Interactive Quantum Chemistry**. *Journal of Computational Chemistry*, **34**(6):492–504, March 2013. 8, 18

[17] Semeho Prince A. Edorh and Stephane Redon. **Incremental update of electrostatic interactions in adaptively restrained particle simulations**. *Journal of Computational Chemistry*, pages 1–42, April 2018. 8, 17

[18] Joseph O Hirschfelder, Charles F Curtiss, Robert Byron Bird, and Maria Goeppert Mayer. *Molecular theory of gases and liquids*, **165**. Wiley New York, 1964. 9

[19] PN Keating. **Effect of invariance requirements on the elastic strain energy of crystals with application to the diamond structure**. *Physical Review*, **145**(2):637, 1966. 10

[20] Jan Skov Pedersen. **Surface relaxation by the keating model: A comparison with ab-initio calculations aind x-ray diffraction experiments**. *Surface science*, **210**(1-2):238–250, 1989. 10

[21] Anthony K Rappé, Carla J Casewit, KS Colwell, William A Goddard III, and WM Skiff. **UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations**. *Journal of the American chemical society*, **114**(25):10024–10035, 1992. 11, 12

[22] Maria M Reif, Philippe H Huünenberger, and Chris Oostenbrink. **New interaction parameters for charged amino acid side chains in the GROMOS force field**. *Journal of chemical theory and computation*, **8**(10):3705–3723, 2012. 11

[23] David Van Der Spoel, Erik Lindahl, Berk Hess, Gerrit Groenhof, Alan E Mark, and Herman JC Berendsen. **GROMACS: fast, flexible, and free**. *Journal of computational chemistry*, **26**(16):1701–1718, 2005. 11, 14

[24] Donald W Brenner. **Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films**. *Physical review B*, **42**(15):9458, 1990. 12

[25] Venkatesh Botu, Rohit Batra, James Chapman, and Rampi Ramprasad. **Machine learning force fields: Construction, validation, and outlook**. *The Journal of Physical Chemistry C*, **121**(1):511–522, 2016. 12

[26] Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, and Klaus-Robert Müller. **Machine learning of accurate energy-conserving molecular force fields**. *Science advances*, **3**(5):e1603015, 2017. 12, 104

REFERENCES

[27] Svetlana Artemova, Sergei Grudinin, and Stephane Redon. **A comparison of neighbor search algorithms for large rigid molecules**. *Journal of Computational Chemistry*, **32**(13):2865–2877, October 2011. 16

[28] Pierre-Luc Manteaux, François Faure, Stephane Redon, and Marie-Paule Cani. **Exploring the Use of Adaptively Restrained Particles for Graphics Simulations**. In *VRIPHYS 2013 - 10th Workshop on Virtual Reality Interaction and Physical Simulation*, VRIPHYS 2013 - 10th Workshop on Virtual Reality Interaction and Physical Simulation, pages 17–24, Lille, France, November 2013. Eurographics Association. 17

[29] Krishna Kant Singh, Dmitriy F. Marin, and Stephane Redon. **Parallel Adaptively Restrained Molecular Dynamics**. In *2017 International Conference on High Performance Computing & Simulation (HPCS)*, pages 308 – 314, Genova, Italy, July 2017. IEEE. 17

[30] Krishna Kant Singh and Stephane Redon. **Single-pass Incremental Force Updates for Adaptively Restrained Molecular Dynamics**. *Journal of Computational Chemistry*, **39**(8):412–423, March 2018. 17

[31] Krishna Kant Singh and Stephane Redon. **Adaptively Restrained Molecular Dynamics in LAMMPS**. *Modelling and Simulation in Materials Science and Engineering*, **25**(5):055013, June 2017. 17

[32] Semeho Edorh. *Incremental Algorithm for long range interactions*. Theses, Université Grenoble Alpes, October 2018. 17

[33] Maël Bosson. *Adaptive algorithms for computational chemistry and interactive modeling*. Theses, Université de Grenoble, October 2012. 18

[34] Sowmya Viswanathan, Tharangattu N Narayanan, Kiana Aran, Kathryn D Fink, Jacobo Paredes, Pulickel M Ajayan, Slawomir Filipek, Przemyslaw Miszta, H Cumhur Tekin, Fatih Inci, et al. **Graphene–protein field effect biosensors: glucose sensing**. *Materials Today*, **18**(9):513–522, 2015. 18

[35] Cyril Guedj, Léonard Jaillet, François Rousse, and Stéphane Redon. **Atomistic Modelling and Simulation of Transmission Electron Microscopy Images: Application to Intrinsic Defects of Graphene**. In *8th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, 2018. 20

[36] Cyril Guedj, Léonard Jaillet, François Rousse, and Stephane Redon. **Impact of Hydrogen on Graphene-based Materials: Atomistic Modeling and Simulation of HRSTEM Images**. In *AVS 65th International Symposium & Exhibition*, 2018. 20

[37] Joseph E Avron and Alexander Elgart. **Adiabatic theorem without a gap condition**. *Communications in mathematical physics*, **203**(2):445–463, 1999. 24

[38] Dominik Marx and Jürg Hutter. *Ab initio molecular dynamics: basic theory and advanced methods*. Cambridge University Press, 2009. 24, 25, 36

[39] Douglas A Gibson, Irina V Ionova, and Emily A Carter. **A comparison of Car—Parrinello and Born—Oppenheimer generalized valence bond molecular dynamics**. *Chemical physics letters*, **240**(4):261–267, 1995. 25

[40] P. Hohenberg and W. Kohn. **Inhomogeneous Electron Gas**. *Physical Review*, **136**(3B):B864–B871, nov 1964. 26

[41] W. Kohn and L. J. Sham. **Self-Consistent Equations Including Exchange and Correlation Effects**. *Physical Review*, **140**(4A):A1133–A1138, nov 1965. 27, 36

[42] Miguel AL Marques, Micael JT Oliveira, and Tobias Burnus. **Libxc: A library of exchange and correlation functionals for density functional theory**. *Computer Physics Communications*, **183**(10):2272–2281, 2012. 29, 32

[43] J. P. Perdew and Alex Zunger. **Self-interaction correction to density-functional approximations for many-electron systems**. *Physical Review B*, **23**(10):5048–5079, may 1981. 30, 51

[44] John P Perdew, John A Chevary, Sy H Vosko, Koblar A Jackson, Mark R Pederson, Dig J Singh, and Carlos Fiolhais. **Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation**. *Physical Review B*, **46**(11):6671, 1992. 30, 51

[45] Xin Liu, Xiao Wang, Zaiwen Wen, and Yaxiang Yuan. **On the convergence of the self-consistent field iteration in Kohn–Sham density functional theory**. *SIAM Journal on Matrix Analysis and Applications*, **35**(2):546–558, 2014. 33

[46] Hugues Dreyssé. *Electronic Structure and Physical Properties of Solids*. Springer, 2000. 35

[47] Peter E Blöchl. **Projector augmented-wave method**. *Physical review B*, **50**(24):17953, 1994. 35

[48] Peter E Blöchl, Clemens J Först, and Johannes Schimpl. **Projector augmented wave method: ab initio molecular dynamics with full wave functions**. *Bulletin of Materials Science*, **26**(1):33–41, 2003. 35

[49] Huajie Chen, Aihui Zhou, et al. **Orbital-free density functional theory for molecular structure calculations**. *Numer. Math. Theor. Meth. Appl*, **1**(1), 2008. 36, 38, 42, 56

[50] Gregory S. Ho, Vincent L. Lignères, and Emily A. Carter. **Introducing PROFESS: A new program for orbital-free density functional theory calculations**. *Computer Physics Communications*, **179**(11):839–854, dec 2008. 36, 37, 38, 41, 42, 47, 58, 78

[51] L. H. Thomas. **The calculation of atomic fields**. *Mathematical Proceedings of the Cambridge Philosophical Society*, **23**(05):542, jan 1927. 37

[52] CF v Weizsäcker. **Zur theorie der kernmassen**. *Zeitschrift für Physik A Hadrons and Nuclei*, **96**(7):431–458, 1935. 37, 39

[53] Balachandran Radhakrishnan and Vikram Gavini. **Effect of cell size on the energetics of vacancies in aluminum studied via orbital-free density functional theory**. *Physical Review B*, **82**(9):094117, 2010. 37

[54] Prabhat K Acharya, Libero J Bartolotti, Stephen B Sears, and Robert G Parr. **An atomic kinetic energy functional with full Weizsacker correction**. *Proceedings of the National Academy of Sciences*, **77**(12):6978–6982, 1980. 37

[55] DA Kirzhnits. **Quantum corrections to the Thomas-Fermi equation**. *Soviet Phys. JETP*, **5**, 1957. 38

[56] Sidney Yip. *Handbook of materials modeling*. Springer Science & Business Media, 2007. 38

[57] Alex Borgoo and David J Tozer. **Density scaling of noninteracting kinetic energy functionals**. *Journal of chemical theory and computation*, **9**(5):2250–2255, 2013. 39

[58] Alex Borgoo, James A Green, and David J Tozer. **Molecular binding in post-Kohn–Sham orbital-free DFT**. *Journal of chemical theory and computation*, **10**(12):5338–5345, 2014. 39

[59] Junji Seino, Ryo Kageyama, Mikito Fujinami, Yasuhiro Ikabata, and Hiromi Nakai. **Semi-local machine-learned kinetic energy density functional with third-order gradients of electron density**. *The Journal of chemical physics*, **148**(24):241705, 2018. 39, 102, 103

[60] Lin-Wang Wang and Michael P. Teter. **Kinetic-energy functional of the electron density**. *Physical Review B*, **45**(23):13196–13220, jun 1992. 39, 41

[61] Jens Lindhard. **On the properties of a gas of charged particles**. *Kgl. Danske Videnskab. Selskab Mat.-Fys. Medd.*, **28**, 1954. 39

[62] Steven David Schwartz. *Theoretical methods in condensed phase chemistry*, **5**. Springer Science & Business Media, 2002. 40

[63] F Perrot. **Hydrogen-hydrogen interaction in an electron gas**. *Journal of Physics: Condensed Matter*, **6**(2):431, 1994. 41

[64] Enrico Smargiassi and Paul A Madden. **Orbital-free kinetic-energy functionals for first-principles molecular dynamics**. *Physical Review B*, **49**(8):5220, 1994. 41

[65] Yan Alexander Wang, Niranjan Govind, and Emily A Carter. **Orbital-free kinetic-energy functionals for the nearly free electron gas**. *Physical Review B*, **58**(20):13465, 1998. 41, 53

[66] Swarnava Ghosh and Phanish Suryanarayana. **Higher-order finite-difference formulation of periodic Orbital-free Density Functional Theory**. *Journal of Computational Physics*, **307**:634–652, 2016. 41, 44, 45, 72

[67] Nicholas Choly and Efthimios Kaxiras. **Kinetic energy density functionals for non-periodic systems**. *Solid state communications*, **121**(5):281–286, 2002. 41, 44, 45, 53, 69

[68] Xavier Blanc and E Cances. **Nonlinear instability of density-independent orbital-free kinetic-energy functionals**. *The Journal of chemical physics*, **122**(21):214106, 2005. 42, 69

[69] Chen Huang and Emily A. Carter. **Nonlocal orbital-free kinetic energy density functional for semiconductors**. *Physical Review B*, **81**(4), jan 2010. 42, 56

[70] Ilgyou Shin and Emily A Carter. **Enhanced von Weizsäcker Wang-Govind-Carter kinetic energy density functional for semiconductors**. *The Journal of chemical physics*, **140**(18):18A531, 2014. 42

[71] Junchao Xia and Emily A Carter. **Density-decomposed orbital-free density functional theory for covalently bonded molecules and materials**. *Physical Review B*, **86**(23):235109, 2012. 42

[72] Wenhui Mi, Xuecheng Shao, Chuanxun Su, Yuanyuan Zhou, Shoutao Zhang, Quan Li, Hui Wang, Lijun Zhang, Maosheng Miao, Yanchao Wang, et al. **ATLAS: A real-space finite-difference implementation of orbital-free density functional theory**. *Computer Physics Communications*, **200**:87–95, 2016. 43

[73] Xuecheng Shao, Qiang Xu, Sheng Wang, Jian Lv, Yanchao Wang, and Yanming Ma. **Large-scale ab initio simulations for periodic system**. *Computer Physics Communications*, **233**:78–83, 2018. 43, 72

[74] Phanish Suryanarayana and Deepa Phanish. **Augmented Lagrangian formulation of orbital-free density functional theory**. *Journal of Computational Physics*, **275**:524–538, oct 2014. 43, 47, 52, 53, 58, 60, 61, 66, 72, 77, 78

[75] Vikram Gavini, Jaroslaw Knap, Kaushik Bhattacharya, and Michael Ortiz. **Non-periodic finite-element formulation of orbital-free density functional theory**. *Journal of the Mechanics and Physics of Solids*, **55**(4):669–696, 2007. 44, 58

[76] Phani Motamarri, Mrinal Iyer, Jaroslaw Knap, and Vikram Gavini. **Higher-order adaptive finite-element methods for orbital-free density functional theory**. *Journal of Computational Physics*, **231**(20):6596–6621, 2012. 44, 45, 64, 69

[77] Carlos J Garcia-Cervera. **An efficient real space method for orbital-free density-functional theory**. *Commun. Comput. Phys*, **2**(2):334–357, 2007. 44

[78] François Rousse and Stéphane Redon. **Incremental solver for orbital-free density functional theory**. *Journal of Computational Chemistry*, **0**(0), 2019. 47

[79] Stephen Wright and Jorge Nocedal. **Numerical optimization**. *Springer Science*, **35**(67-68):7, 1999. 47

[80] Weitao Yang. **Gradient correction in Thomas-Fermi theory**. *Physical Review A*, **34**(6):4575, 1986. 52

[81] Robert G Parr and Yang Weitao. *Density-functional theory of atoms and molecules*, **16**. Oxford university press, 1994. 52

[82] Niranjan Govind, Jian Wang, and Hong Guo. **Total-energy calculations using a gradient-expanded kinetic-energy functional**. *Physical Review B*, **50**(15):11175–11178, oct 1994. 52

[83] V Heine and I V Abarenkov. **A new method for the electronic structure of metals**. *Philosophical Magazine*, **9**(99):451–465, 1964. 55

[84] IV Abarenkov and V Heine. **The model potential for positive ions**. *Philosophical Magazine*, **12**(117):529–537, 1965. 55

[85] L Goodwin, RJ Needs, and Volker Heine. **A pseudopotential total energy study of impurity-promoted intergranular embrittlement**. *Journal of Physics: Condensed Matter*, **2**(2):351, 1990. 56

[86] S Watson, B. J Jesson, E. A Carter, and P. A Madden. **Ab initiopseudopotentials for orbital-free density functionals**. *Europhysics Letters (EPL)*, **41**(1):37–42, jan 1998. 56

[87] Baojing Zhou, Yan Alexander Wang, and Emily A. Carter. **Transferable local pseudopotentials derived via inversion of the Kohn-Sham equations in a bulk environment**. *Physical Review B*, **69**(12), mar 2004. 56

[88] Chen Huang and Emily A. Carter. **Transferable local pseudopotentials for magnesium, aluminum and silicon**. *Physical Chemistry Chemical Physics*, **10**(47):7109, 2008. 56

[89] Junchao Xia, Chen Huang, Ilgyou Shin, and Emily A Carter. **Can orbital-free density functional theory simulate molecules?** *The Journal of chemical physics*, **136**(8):084102, 2012. 56

[90] Gene H Golub and John H Welsch. **Calculation of Gauss quadrature rules**. *Mathematics of computation*, **23**(106):221–230, 1969. 58

[91] Pascal Viot. **Méthode d'analyse numérique.** Lecture, September 2006. 58

[92] L. Dagum and R. Menon. **OpenMP: an industry standard API for shared-memory programming**. *IEEE Computational Science and Engineering*, **5**(1):46–55, Jan 1998. 74, 90

[93] Thomas Guillet and Romain Teyssier. **A simple multigrid scheme for solving the Poisson equation with arbitrary domain boundaries**. *Journal of Computational Physics*, **230**(12):4756–4771, 2011. 75

[94] Dietrich Braess. **On the combination of the multigrid method and conjugate gradients**. In *Multigrid methods II*, pages 52–64. Springer, 1986. 75

[95] Clive Temperton. **Direct methods for the solution of the discrete Poisson equation: some comparisons**. *Journal of Computational Physics*, **31**(1):1–20, 1979. 75

[96] Walter Gander and Gene H Golub. **Cyclic reduction—history and applications**. *Scientific computing (Hong Kong, 1997)*, pages 73–85, 1997. 75

[97] Luigi Genovese, Thierry Deutsch, Alexey Neelov, Stefan Goedecker, and Gregory Beylkin. **Efficient solution of Poisson's equation with free boundary conditions**. *The Journal of chemical physics*, **125**(7):074105, 2006. 75

[98] James F Gibbons. **Ion implantation in semiconductors—Part II: Damage production and annealing**. *Proceedings of the IEEE*, **60**(9):1062–1096, 1972. 89

[99] Hannes Jónsson, Greg Mills, and Karsten W Jacobsen. **Nudged elastic band method for finding minimum energy paths of transitions**. In *Classical and quantum dynamics in condensed phase simulations*, pages 385–404. World Scientific, 1998. 97

[100] Lucian A Constantin, Eduardo Fabiano, and Fabio Della Sala. **Semilocal Pauli–Gaussian kinetic functionals for orbital-free density functional theory calculations of solids**. *The journal of physical chemistry letters*, **9**(15):4385–4390, 2018. 102

[101] Amara Graps. **An introduction to wavelets**. *IEEE computational science and engineering*, **2**(2):50–61, 1995. 103

[102] Albert Cohen, Ingrid Daubechies, and J-C Feauveau. **Biorthogonal bases of compactly supported wavelets**. *Communications on pure and applied mathematics*, **45**(5):485–560, 1992. 103

[103] Luigi Genovese, Brice Videau, Matthieu Ospici, Thierry Deutsch, Stefan Goedecker, and Jean-François Méhaut. **Daubechies wavelets for high performance electronic structure calculations: The BigDFT project**. *Comptes Rendus Mécanique*, **339**(2-3):149–164, 2011. 103

[104] Kyle Mills, Michael Spanner, and Isaac Tamblyn. **Deep learning and the Schrödinger equation**. *Physical Review A*, **96**(4):042113, 2017. 103

[105] Felix Brockherde, Leslie Vogt, Li Li, Mark E Tuckerman, Kieron Burke, and Klaus-Robert Müller. **Bypassing the Kohn-Sham equations with machine learning**. *Nature communications*, **8**(1):872, 2017. 103

[106] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. **ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost**. *Chemical science*, **8**(4):3192–3203, 2017. 104

[107] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. **Mastering the game of go without human knowledge**. *Nature*, **550**(7676):354, 2017. 104

[108] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojciech M. Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Yuhuai Wu, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver

Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. **AlphaStar: Mastering the Real-Time Strategy Game StarCraft II**. `https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/`, 2019. 104

[109] R Evans, J Jumper, J Kirkpatrick, L Sifre, TFG Green, C Qin, A Zidek, A Nelson, A Bridgland, H Penedones, et al. **De novo structure prediction with deeplearning based scoring**. *Annu Rev Biochem*, **77**:363–382, 2018. 104

[110] Gerald Teschl. **Mathematical methods in quantum mechanics**. *Graduate Studies in Mathematics*, **99**, 2009. 106