



HAL
open science

Incremental solver for Orbital-Free Density Functional Theory

François Rouse, Stephane Redon

► **To cite this version:**

François Rouse, Stephane Redon. Incremental solver for Orbital-Free Density Functional Theory. Journal of Computational Chemistry, 2019, 40 (23), pp.2013-2027. 10.1002/jcc.25854 . hal-02135603

HAL Id: hal-02135603

<https://inria.hal.science/hal-02135603v1>

Submitted on 21 May 2019

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

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

Incremental solver for Orbital-Free Density Functional Theory

François Rousse^{*}, Stéphane Redon[†]

May 14, 2019

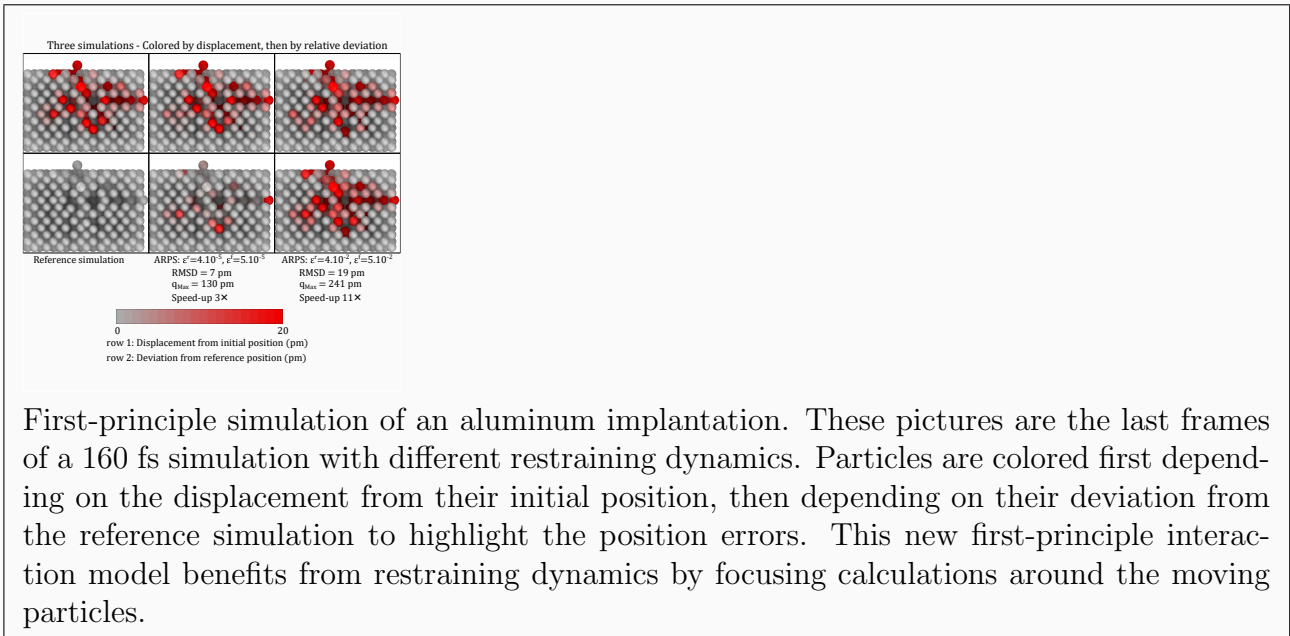
Abstract

First-principle calculations are still a challenge since they require a great amount of computational time. In this paper, we introduce a new algorithm to perform Orbital-Free Density Functional Theory (OF-DFT) calculations. Our new algorithm focuses computational efforts on important parts of the particle system, which, in the context of Adaptively Restrained Particle Simulations (ARPS) allows us to accelerate particle simulations.

Keywords: Ab-initio molecular dynamic, Orbital-Free DFT, Real-Space Finite-Differences, Adaptively Restrained Particle Simulations, Incremental algorithm ■

^{*}Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France

[†]Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France



INTRODUCTION

Molecular simulation has become a major part of biology, crystallography and chemistry over the last decades being able for example to predict protein structures¹, crystal tensor constraints² or energy barriers in chemical reactions³. This ‘in silico’ chemistry either uses empirical force fields parametrized to fit experiments or aims at modeling molecules through Schrödinger equation, the first-principle methods, to explain and so predict all chemistry at once. Quantum physics unfortunately miss mathematical methods to solve Schrödinger equation, it can be solved analytically for cases with one particle in specific electrical fields, but analytic resolution becomes impossible with more than two interacting particles and computational methods have to be used⁴. One of the most popular ones is the Density Functional Theory (DFT), with in particular the KS-DFT variant proposed by Kohn and Sham⁵.

The KS-DFT success comes from its good trade-off between computational cost and accuracy. The method is able to model moderately large systems –up to several hundreds of atoms– with correct energies, bonds and molecular structures. Nonetheless, computational costs become too high to model bigger systems –thousands of atoms– or for interactive first-principle simulations. This is because the computation of non-interacting kinetic energy requires orbitals which involves a search space of dimension $3n$, n being the number of orbitals considered.

A much faster first-principle method is the Orbital-Free Density Functional Theory. OF-DFT is based on KS-DFT but does not use orbitals to compute the non-interacting kinetic energy, using only the electron density². This simple dependence increases greatly computational speed. On the other hand, accuracy and portability are greatly decreased, for example molecules and covalently bonded systems are not properly modeled⁶. OF-DFT has nonetheless shown the ability to model metallic bonds like in aluminum^{2,7,8}.

Even with OF-DFT, the computational cost is still too high for being used on large dynamic systems. Hence, we propose here to accelerate OF-DFT simulations by combining the electron density calculation with Adaptively Restrained Particle Simulations (ARPS). ARPS modify the system’s dynamic by freezing the movement of slowest particles⁹ which reduces

the time needed to update forces in a simulation. ARPS have been originally developed for pairwise force fields like Lennard-Jones interactions¹⁰, but has already been applied to first-principles simulations^{11,12} and more recently, to the computation of long range interactions¹³. A parallel version has also been developed¹⁴ and implemented in LAMMPS¹⁵. An implementation of the method is available on the SAMSON software platform at <http://www.samson-connect.net> with several interaction models.

In section 1, we first explain ARPS, briefly recall the main ideas and equations behind OF-DFT, then describe our new version of OF-DFT and how it can be combined with ARPS by focusing the calculations on specific parts of the computational domain. In section 2 we validate our method by comparing the energies we compute with those obtained from another OF-DFT implementation (PROFESS), and measure the speedup achieved by our method with two examples.

METHODOLOGY

Before we describe our method, we first introduce the molecular dynamics in which simulations will take place, ARPS, then recall the main ideas and equations behind the OF-DFT.

Adaptively Restrained Particle Simulations

Adaptively Restrained Particle Simulations (ARPS) freeze the slowest particle movements so that the number of moved particles at each time step decreases.

Briefly, ARPS manage the behavior of particles by using their kinetic energy K and 2 thresholds: ε^r , the fully-restrained threshold, and ε^f , the full-dynamics threshold. If $K \leq \varepsilon^r$ the particle is inactive and its movement is totally restrained, if $K \geq \varepsilon^f$ the particle moves freely. Between those two thresholds, when $\varepsilon^r < K < \varepsilon^f$, a C^2 function permits a smooth transition between the two states (fully restrained and not restrained). For details and mathematical explanations, we refer the reader to^{9,10}. The consequences on macroscopic quantities of this restrained adaptive dynamic have been studied in^{16,17}.

For pairwise interaction models like Lennard-Jones potential, ARPS make it possible to update fewer forces. For others interaction models, incremental methods that compare

the differences in positions between two consecutive time steps and adapt the computation of forces in consequence can benefit from a reduction of the number of active (not fully restrained) particles. Several incremental methods have been developed to compute first-principle interactions^{11,12} or electrostatic interactions¹³. Other simulation methods that do not update the positions of all particles, like Monte-Carlo Simulations¹⁸, also benefit from incremental computation of forces and energies.

We have designed our new algorithm for OF-DFT calculations to benefit from such simulations.

Orbital-Free Density Functional Theory

In theory, the computation of the electronic structure of a molecular system is done by finding eigenvalues of the time-independent Schrödinger equation: $H\psi = E\psi$, with H the system's Hamiltonian, E its energy and ψ the wave-function. This equation can be solved analytically only for the simplest of systems – one particle in some specific electric potentials – and quickly becomes impossible numerically even for small systems – around 100 particles^{19,20} –. The first possible simplification is the Born-Oppenheimer approximation that differentiates nuclei and electrons behaviors. As the formers are thousands times heavier than the seconds, they are considered fixed while the electronic structure relaxes to its stable state. In the Schrödinger equation, E becomes the electronic energy, ψ the many-body electronic wave-function and H the Hamiltonian of electrons immersed in the electrostatic potential created by fixed nuclei. Another typical approximation is the Hohenberg-Kohn theorems²¹ on which DFT rely. The first one states that one can write the energy of an electron cloud E as a functional of its electron density ρ , the second one that the fundamental state of this electron cloud is the one minimizing this functional.

System energy

From the aforementioned simplification and theorems, the total energy E_{Tot} of a molecular system appears as the sum of the repulsive energy E_{ii} between nuclei in position \mathbf{R} and the electrons energy E within the potential created by the nuclei:

$$E_{Tot}[\rho, \mathbf{R}] = E[\rho, \mathbf{R}] + E_{ii}[\mathbf{R}] \quad (1)$$

where $E_{ii}[\mathbf{R}]$ is defined for a system of M atoms of charge $(Q_I)_{I \in [1, M]}$ such as:

$$E_{ii}[\mathbf{R}] = \frac{1}{2} \sum_{I=1}^M \sum_{\substack{J=1 \\ J \neq I}}^M \frac{Q_I Q_J}{|\mathbf{R}_I - \mathbf{R}_J|} \quad (2)$$

As we are working under the Born-Oppenheimer approximation, the nuclei are considered fixed while the electrons move. Thus, a dynamical simulation or a structural relaxation consists in iterating the following steps:

1. Computation of ρ and $E[\rho, \mathbf{R}]$ with fixed nucleus positions \mathbf{R} .
2. Computation of forces applied on nuclei.
3. Update of nuclei positions \mathbf{R} .

In practice, most of the computational effort is spent on the estimation of ρ during step 1, which is evaluated through the resolution of the constrained optimization problem detailed below.

Constrained optimization problem

The computation of ρ is done through Hohenberg-Kohn second theorem: for every electron density $\rho(\vec{r})$ of Q_e electrons ($\int \rho(\mathbf{r}) d\mathbf{r} = Q_e$), $E[\rho] \geq E_{ground}$, E_{ground} being the ground state energy. Thus, we find the ground state electron density ρ^* by minimizing the functional $E[\rho]$ with respect to two constraints, the quantity of electrons is fixed: $\int \rho(\mathbf{r}) d\mathbf{r} = Q_e$ and ρ is a positive variable: $\forall \mathbf{r}, \rho(\mathbf{r}) \geq 0$. ρ^* is the result of an optimization process:

$$\begin{cases} \rho^* = \operatorname{argmin}(E[\rho]) \\ \int \rho^*(\mathbf{r}) d\mathbf{r} = Q_e \\ \forall \mathbf{r}, \rho^*(\mathbf{r}) \geq 0 \end{cases} \quad (3)$$

To evaluate $E[\rho]$ and its derivative with respect to the variable ρ , several schemes are available. Among them we find the variants of Kohn-Sham DFT, with different functions

basis like LCAO, GTO, STO⁵ or PAW^{22,23} and Orbital-Free DFT. Here, we propose to use OF-DFT because of its computational efficiency and because it relies on local computations that are convenient for the incremental scheme we want to apply.

OF-DFT Energy functional

Kohn-Sham-DFT requires the use of wave-functions to evaluate the electronic kinetic energy²⁴. This usually involves a process of orbital orthogonalization to compute Kohn-Sham orbitals⁵, which causes 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. To describe the OF-DFT energy functionals and their derivatives, we use the names and notations from the Carter group². 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 as

$$E[\rho, \mathbf{R}] = T_s[\rho] + E_H[\rho] + E_{ext}[\rho, \mathbf{R}] + E_{XC}[\rho] \quad (4)$$

with T_s the electronic kinetic energy (KEDF), E_H the Hartree electronic repulsion energy, E_{ext} the interaction energy between electrons and the ambient electrostatic potential, and E_{XC} the exchange-correlation energy.

Electrostatic Energy

The first energy term driving the electron density is its interaction with the external electrostatic potential V_{ext} .

$$E_{ext}[\rho] = - \int V_{ext}(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} \quad (5)$$

where $V_{ext}(\mathbf{r})$ is the potential created by all nuclei indexed by $I \in \llbracket 1, M \rrbracket$, at positions \mathbf{R}_I . It may also contain the contribution due to an eventual external electrostatic field. However, in the rest of this paper, we will assume that E_{ext} only concerns the interaction between nuclei and electrons.

Hartree Electronic Repulsion Energy

The second term is the electronic interaction which pushes away the electrons from one another. In 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:

$$E_H[\rho] = \frac{1}{2} \int \int \frac{\rho(\mathbf{r}')\rho(\mathbf{r})}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' d\mathbf{r} \quad (6)$$

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

$$V_H(\mathbf{r}) = \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' \quad (7)$$

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

$$E_H[\rho] = \frac{1}{2} \int V_H(\mathbf{r})\rho(\mathbf{r}) d\mathbf{r} \quad (8)$$

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 orbitals as input, the OF-KEDF term is a functional of the electron density ρ^2 . A simple way to evaluate KEDF is 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 gas²⁵:

$$T_{TF}[\rho] = C_{TF} \int \rho^{5/3}(\mathbf{r}) d\mathbf{r} \quad (9)$$

with $C_{TF} = \frac{3}{10}(3\pi^2)^{2/3}$

- The von Weizsäcker functional T_{vW} , exact for a single orbital system²⁶:

$$T_{vW}[\rho] = \int \sqrt{\rho(\mathbf{r})} \left(-\frac{1}{2} \nabla^2 \right) \sqrt{\rho(\mathbf{r})} d\mathbf{r} \quad (10)$$

The classic scheme is a TF functional corrected by a vW functional, the $TF - \lambda vW$ model of the form:

$$T_s = T_{TF} + \lambda T_{vW} \quad (11)$$

A more precise way to evaluate OF-DFT KEDF was proposed by Wang and Teter²⁷, they added a non-local third term to enforce the kinetic energy to respect the linear response of an

electron gas²⁸. Other non-local third terms have been developed²⁹, among which the Wang-Govind-Carter KE^{30,31} for a better stability of the optimization algorithm and the Huang-Carter KE³² for better results on molecules, all based on the theoretical linear response of the electron gas. The $TF - \lambda vW$ functional lacks of precision, missing the shell structures and the exponential decay of densities². Nonetheless, here we will keep the $TF - \lambda vW$ KEDF. First because of the difficulties to calculate non-local KEDF on real-space with free boundary conditions, we tried to adapt the method proposed here^{33,34} but the speed and stability of our benchmarks were greatly reduced. Then because PROFESS predicts similar structure for aluminum with WGC KE and TF-vW KE: a FCC crystal with shifts of 0.07 Å for the lattice constant and 0.5 eV/atom for the bulk energy, and FCC aluminum will be our main system for benchmarks.

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.

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:

$$E_{XC}[\rho] = T[\rho] - T_s[\rho] + E_{ee}[\rho] - E_H[\rho] \quad (12)$$

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 Local Density Approximation (LDA) to evaluate E_{XC} . More precise approximations exist, for example Generalized Gradient Approximations (GGA) that uses local gradient of electron density³⁵, but in OF-DFT the main cause of error comes from the kinetic energy and the pseudo-potentials, not from the E_{XC} so we keep the LDA. Thus, E_{XC} can be rewritten as:

$$E_{XC}[\rho] = \int [\varepsilon_X(\rho(\mathbf{r})) + \varepsilon_C(\rho(\mathbf{r}))]\rho(\mathbf{r})d\mathbf{r} \quad (13)$$

where

$$\varepsilon_X(\mathbf{r}) = -\frac{3}{4} \left(\frac{3}{\pi} \rho(\mathbf{r}) \right)^{\frac{1}{3}} \quad (14)$$

$$\varepsilon_C(\mathbf{r}) = \begin{cases} a \ln(r_s) + b + cr_s \ln(r_s) + dr_s & , r_s < 1 \\ \frac{\gamma}{1 + \beta_1 \sqrt{r_s} + \beta_2 r_s} & , r_s \geq 1 \end{cases} \quad (15)$$

r_s is the Wigner–Seitz radius: $r_s(\rho) = \left(\frac{3}{4\pi\rho} \right)^{\frac{1}{3}}$. Different implementations of OF-DFT computation have been tested, either in Fourier space² or in real-space with finite elements³³ or with finite differences⁸. Fastest implementations, like PROFESS, use Fourier space. This 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 proposed a new Real-Space and Finite-Differences (RS-FD) OF-DFT method that is presented below.

Incremental OF-DFT

We describe here our method to speed up OF-DFT simulations. As the aim of our implementation is to be incremental in order to take full advantage of Adaptively Restrained Dynamics, we choose a Real-Space Finite-Differences (RSFD) scheme to keep the computations local. We first explain our RSFD implementation then describe how we update the electron density in a dynamical simulation.

Numerical implementation

Real Space, Finite Difference scheme

The first step is to choose 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 this Fourier space OF-DFT functional calculations scale linearly with the space size, the global cost of an energy evaluation comes from the Fourier transform required to go into Fourier space. With the Fast Fourier Transform method (FFT) this cost is limited to $[O(N \ln(N))]$. Here, we do not use Fourier space since it involves basis functions

having a global impact whereas we want to use local methods allowing to take advantage of locally restrained particles. Hence we use a real space scheme. Real space methods also present other advantages: they are well adapted to isolated system, can use arbitrary boundary conditions and make computations easy to parallelize^{34,36}. Thus every variables, ρ and V_{ext} among them, are computed on a discrete cell Ω 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 Ω is large enough to encompass all the molecular system's electrons. Those nodes are indexed by integers $i \in \llbracket 1, N \rrbracket$ with $N = N_x N_y N_z$ the total number of nodes. The functionals $\rho(\mathbf{r})$, $\chi(\mathbf{r})$, $V_{ext}(\mathbf{r})$ 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 position inside node i and $\mathbf{r}(i)$ its center.

Augmented Lagrangian method

The second 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 $\rho = e^x$ or $\rho = \chi^4$ were tested and the conclusion was that $\rho = \chi^2$ is the most stable one², for the von Weizsäcker functional does not diverge for small values of ρ . For coherence, 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 method³⁷, like in⁸. We define the Augmented Lagrangian as

$$\mathcal{L}_A[\chi, \lambda, \mu] = E[\chi] + \frac{\mu}{2} c_\Omega^2[\chi, Q_e] - \lambda c_\Omega[\chi, Q_e] \quad (16)$$

with $c_\Omega[\chi, Q_e] = \int_\Omega \chi^2(\mathbf{r}) d\mathbf{r} - Q_e$ the equality function, 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,

$$\nabla_\chi \mathcal{L}_A[\chi, \lambda, \mu] = 0 \text{ and } Q_e = 0 \quad (17)$$

which leads to:

$$\nabla_\chi E[\chi] = \lambda \nabla_\chi c_\Omega[\chi, Q_e] = 2\lambda\chi \quad (18)$$

and

$$\lambda = \frac{1}{2Q_e} \int_{\Omega} \chi \nabla_{\chi} E[\chi] d\mathbf{r} \quad (19)$$

We consider the energy minimum is reached when the Augmented Lagrangian derivative is below a tolerance threshold δ , ie. when $|\nabla_{\chi} \mathcal{L}_A[\chi, \lambda, \mu]| < \delta$. At each optimization increment, the lagrangian multiplier is computed with formula 19, 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 $\rho_m = \text{argmin}(E[\rho])$

- 1 Initialization of the cell Ω
 - 2 Computation of the external potential V_{ext} on Ω
 - 3 **while** $|\nabla_{\chi} \mathcal{L}_A[\chi, \lambda, \mu]| > \delta$ **do**
 - 4 $V_H(\mathbf{r}) \leftarrow \int_{\Omega} \frac{\chi_k^2(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'$: update of the Hartree Potential
 - 5 $\lambda = \frac{1}{2Q_e} \int_{\Omega} \chi(\mathbf{r}) \frac{\delta E[\chi]}{\delta \chi} d\mathbf{r}$: update of the Lagrangian multiplier
 - 6 $\partial \chi_k \leftarrow \nabla_{\chi} \mathcal{L}_A[\chi, \lambda, \mu]$: compute the Lagrangian gradient
 - 7 $\chi_{k+1} \leftarrow \text{abs}(\chi_k - \gamma \partial \chi_k)$: perform a steepest descent step and take the
 absolute value of the result
 - 8 **end**
-

Discrete version of OF-DFT

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

$$E_{ext}[\chi] = \sum_{i \in \llbracket 1, N \rrbracket} -V_{ext}(i) \chi^2(i) v_h \quad (20)$$

$$E_H[\chi] = \frac{1}{2} \sum_{i \in \llbracket 1, N \rrbracket} V_H(i) \chi^2(i) v_h \quad (21)$$

$$T_{TF}[\chi] = C_{TF} \sum_{i \in \llbracket 1, N \rrbracket} \chi^{10/3}(i) v_h \quad (22)$$

$$T_{vW}[\chi] = \sum_{i \in \llbracket 1, N \rrbracket} \chi(i) \left(-\frac{1}{2} \nabla^2 \right) \chi(i) v_h \quad (23)$$

$$E_{XC}[\chi] = \sum_{i \in \llbracket 1, N \rrbracket} \chi^2(i) (\varepsilon_X(\chi^2(i)) + \varepsilon_C(\chi^2(i))) v_h \quad (24)$$

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

$$\nabla_\chi E_{ext}(i) = 2V_{ext}(i) \chi(i) \quad (25)$$

$$\nabla_\chi E_H(i) = 2V_H(i) \chi(i) \quad (26)$$

$$\nabla_\chi T_{TF}(i) = C_{TF} \frac{10}{3} \chi^{7/3}(i) \quad (27)$$

$$\nabla_\chi T_{vW}(i) = -\nabla^2 \chi(i) \quad (28)$$

$$\nabla_\chi E_{XC}(i) = \mu_X(\chi(i)) + \mu_C(\chi(i)) \quad (29)$$

with

$$\mu_X(\chi(i)) = - \left(\frac{3}{\pi} \chi^2(i) \right)^{\frac{1}{3}} \quad (30)$$

$$\mu_C(\chi(i)) = \begin{cases} (b - \frac{1}{3}a) + a \ln(r_s) + \frac{1}{3}(2d - c)r_s + \frac{2}{3}cr_s \ln(r_s) & , r_s < 1 \\ \frac{\gamma + \frac{7}{6}\gamma\beta_1\sqrt{r_s} + \frac{4}{3}\gamma\beta_2r_s}{(1 + \beta_1\sqrt{r_s} + \beta_2r_s)^2} & , r_s \geq 1 \end{cases} \quad (31)$$

$$r_s(i) = \left(\frac{3}{4\pi\chi^2(i)} \right)^{\frac{1}{3}}. \quad (32)$$

Pseudo-potentials

Since core electrons are useless in the binding process between atoms, and since their distribution is hard to depict because very sharp, we consider ρ accounts only for the valence electrons, and V_{ext} accounts for the potential created by the ions composed by nuclei and their

core electrons: the pseudo-potential. To model the electrostatic field created by these ions, we have chosen the Bulk-derived Local Pseudo-potentials (BLPSs) computed by the Carter Group^{6,38–40} for they have been especially derived for OF-DFT. BLPSs are spherical and local potentials, meaning that for a particle P at position \mathbf{R}_P , the electrical potential created at \mathbf{r} is written $V_P(\mathbf{r}) = V_{BLPS}(|\mathbf{r} - \mathbf{R}_P|)$. This simplification does not change the form of OF-DFT functionals. We approximate the potential value on node i , $V_{ext}(i) = \int_{\mathbf{r} \in n(i)} V_{ext}(\mathbf{r}) d\mathbf{r}$, with a Gauss-Legendre quadrature of degree three⁴¹.

Hartree potential

The Hartree Potential V_H is not directly derived from its definition, but from the corresponding Poisson equation. Hence, equation 7 becomes

$$4\pi\rho = -\Delta V_H \tag{33}$$

with free boundary condition:

$$V_H(\mathbf{r}) \rightarrow 0 \text{ when } |\mathbf{r}| \rightarrow +\infty \tag{34}$$

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 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.

Kinetic Energy

Several λ are proposed and justified in literature^{42–44}. 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 to⁸.

Algorithm 2: Electronic Hartree Potential Computation on Ω

Input: The Electronic Density ρ on Ω

Output: The Electronic Potential V_H on Ω

- 1 Inject Electronic Density in Ω_2
 - 2 Inject Electronic Density in Ω_3
 - 3 Create Coulombic Boundary Conditions on Ω_3 's border: $\forall \mathbf{r} \in \partial\Omega_3, V_H(\mathbf{r}) = \frac{Q_0}{|\mathbf{r}-\mathbf{r}_0|}$
 - 4 Solve V_H on Ω_3
 - 5 Extract V_H boundary conditions for Ω_2 from Ω_3
 - 6 Solve V_H on Ω_2
 - 7 Extract V_H boundary conditions for Ω from Ω_2
 - 8 Solve V_H on Ω
-

Exchange-Correlation Potential

To compute our LDA XC Potential, we choose the parameters used by Perdew and Zunger⁴⁵: $a = 0.0311$, $b = -0.048$, $c = 0.002$, $d = -0.0116$, $\gamma = -0.1423$, $\beta_1 = 1.0529$ and $\beta_2 = 0.3334$.

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

Incremental update of the electron density

Incremental update pipeline

We consider a molecular system of M particles indexed by $I \in \llbracket 1, M \rrbracket$, 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 molecular 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:

$$V_{ext,t_1}(\mathbf{r}) = V_{ext,t_0}(\mathbf{r}) - V_{BLPS}(|\mathbf{r} - \mathbf{R}_i|) + V_{BLPS}(|\mathbf{r} - \mathbf{R}_i + \mathbf{d}|) \quad (35)$$

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

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.

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 potential:

$$\forall \mathbf{r} \mid |\mathbf{r} - \mathbf{R}_i| > r_\omega : V_{ext}(\mathbf{r}) \simeq \frac{Q_i}{|\mathbf{r} - \mathbf{R}_i|} \text{ and } V_H(\mathbf{r}) \simeq -\frac{Q_i}{|\mathbf{r} - \mathbf{R}_i|} \quad (36)$$

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:

$$\forall \mathbf{r} \mid |\mathbf{r} - \mathbf{R}_i| > r_\omega : V_{ext,t_1}(\mathbf{r}) + V_{H,t_1}(\mathbf{r}) \simeq V_{ext,t_0}(\mathbf{r}) + V_{H,t_0}(\mathbf{r}) \quad (37)$$

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:

1. ω , the loose density sub-area on which everything is updated, close from the moved particle.
2. the complementary of ω , $\Omega \setminus \omega$, on which the electron density is frozen and we update only the electric potentials V_H and V_{ext} .

To discriminate between those two areas, we use r_ω and call it the loose density sub-area radius, 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:

$$\omega = \{\mathbf{r} \mid \mathbf{r} \in \mathbb{B}(\mathbf{R}_i, r_\omega)\} \quad (38)$$

$$\partial\omega = \{\mathbf{r} \mid \mathbf{r} \in \mathbb{S}(\mathbf{R}_i, r_\omega)\} \quad (39)$$

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 :

$$w = \left\{ \mathbf{r} \mid \mathbf{r} \in \bigcup_{I \in [1, P]} \mathbb{B}(\mathbf{R}_I, r_\omega) \right\} \quad (40)$$

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

$$V_{H,t_1}(\mathbf{r}) = V_{H,t_0}(\mathbf{r}) - V_H^{app}(|\mathbf{r} - \mathbf{R}_I|) + V_H^{app}(|\mathbf{r} - \mathbf{R}_I + \mathbf{d}|) \quad (41)$$

With V_{H,t_1} 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:

$$V_H^{app}(r) = \begin{cases} -\frac{Q_I}{r} & \text{for } r > r_H \\ -\frac{Q_I}{r_H} & \text{for } r \leq r_H \end{cases} \quad (42)$$

A small r_H , for example 2 Bohrs 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,t_1} 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 .
- $C_\omega = \{\chi : \mathbb{R}^3 \rightarrow \mathbb{R} \mid \int_\omega \chi^2(\mathbf{r}) d\mathbf{r} = Q_\omega\}$ 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 : $i \in \omega_d \iff \mathbf{r}(i) \in \omega$

To describe fully the incremental update: algorithm 3.

Algorithm 3: Incremental OF-DFT update

Input: p atoms moved, the electron density of the previous step

Output: an electron density with minimized energy at the current step

```

1 Update of the external potential  $V_{ext}$  on  $\Omega$ ;
2 Rough update of the Hartree potential  $V_H$  on  $\Omega$ ;
  /* Restrained Update                                     */
3 while  $|\nabla_\chi \mathcal{L}_A[\chi, \lambda, \mu]|_\omega > \delta_\omega$  do
4    $V_H(\mathbf{r})|_\omega \leftarrow \int_\omega \frac{\chi_k^2(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'$  : update the Hartree Potential on  $\omega$  with Dirichlet
   boundary conditions on  $\partial\omega$ ;
5    $\lambda_\omega \leftarrow \frac{1}{2Q_\omega} \int_\omega \chi(\mathbf{r}) \frac{\delta E[\chi]}{\delta \lambda} d\mathbf{r}$  : update the local Lagrangian multiplier ;
6    $\partial\chi_k|_\omega \leftarrow \nabla_\chi \mathcal{L}_A[\chi, \lambda_\omega, \mu]|_\omega$  : compute the Lagrangian gradient on  $\omega$ ;
7    $\chi_{k+1}|_\omega \leftarrow abs(\chi_k|_\omega - a\partial\chi_k|_\omega)$  : compute a new increment and project on  $\chi \geq 0$  on
    $\omega$ ;
8 end
  /* Global Update                                       */
9 while  $|\nabla_\chi \mathcal{L}_A[\chi, \lambda, \mu]| > \delta$  do
10   $V_H(\mathbf{r}) \leftarrow \int_\Omega \frac{\chi_k^2(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'$  : update the Hartree Potential on  $\Omega$  ;
11   $\lambda \leftarrow \frac{1}{2Q} \int_\Omega \chi(\mathbf{r}) \frac{\delta E[\chi]}{\delta \lambda} d\mathbf{r}$  : update the local Lagrangian multiplier ;
12   $\partial\chi_k \leftarrow \nabla_\chi \mathcal{L}_A[\chi, \lambda_\omega, \mu]$  : compute the Lagrangian gradient on  $\Omega$ ;
13   $\chi_{k+1} \leftarrow abs(\chi_k - a\partial\chi_k)$  : compute a new increment and project on  $\chi \geq 0$  on  $\Omega$  ;
14 end

```

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 $|\omega_d|$. At

equal size the performance is decreased, but we aim at system where ω is around 10% the size of Ω .

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.

RESULTS

In this section, we validate our OF-DFT implementation by comparing its predictions with those of a Fourier space implementation: PROFESS². Most of the tests are inspired by other implementations, a Real-space Finite-Differences one⁸ and Real-Space Finite-Element one^{33,46}. Different Aluminum clusters energy are computed and compared with the one found by PROFESS. Depending on the benchmark, 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 $\varepsilon_b = (E_{Tot} - \sum E_j)/M$ with E_j the energy of the isolated 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 $TF - \lambda vW$ kinetic energy with $\lambda = 0.2$ and the BLPS given by PROFESS. Energy values found with PROFESS might be different in other works^{8,34,46,47} for they use Goodwin-Needs-Heine (GNH) pseudo-potentials instead of BLPS.

Validation

Before measuring any computational time, we need to ensure our implementation gives proper results. To do so we compute the electron densities and energies of different Aluminum compounds like dimer Al_2 , trimer Al_3 and Face-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.

Parameterization

We define the computational domain to encompass all electrons: a box that contains all nuclei, plus a margin. As one can see in figure 2, if we take a thin node size $h = 0.2$ Bohrs and a high order (10^{th}) Laplacian operator, our OF-DFT implementation computes an electron cloud of a single aluminum ion that reaches 10^{-10} electrons per cubic Bohrs at 12.2 Bohrs from the nucleus. For safety we choose a margin of 15 Bohrs.

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 first compute the atomic energy of a $5 \times 5 \times 5$ FCC Aluminum cluster of lattice parameter $a = 8.0$ Bohrs for different h and n and compare it to the atomic energy found with PROFESS. Errors are presented in figure 3. As one can see, the atomic energies errors computed with a 6^{th} or 10^{th} order Laplacian are nearly the same and significantly smaller than with a 2^{nd} order Laplacian. We choose the 6^{th} order Laplacian for the rest of the work, it is as precise as a 10^{th} order Laplacian, and about 20 % faster. To measure the effect of the node size h on the electrons energy precision, we have computed the hydrogen's electron energy and the Aluminum electrons energy for different node sizes, and with a 6^{th} order Laplacian. The electron's hydrogen is interesting first because its energy is known : $E_{el} = -0.5$ Hartree, and because it is alone and on a single orbital, hence it does not undergo Hartree repulsion and the vW kinetic energy functional is exact. So with an adaptation of the functionals, OF-DFT can retrieve the theoretical energy. For the aluminum atom electrons energy, we have kept the OF-DFT functionals as described before.

Figure 4 presents the errors, in log scale, of the energy computed. On the left for the hydrogen with an energy reference of $E_{el} = -0.5$ Hartree, the theoretical value, and on the right the aluminum atom, with an energy reference of $E_{el} = -2.104$ Hartree, the value computed with PROFESS. We observe first the dependence of error with respect to the discretization refinement, then that we can achieve arbitrary precision by choosing a proper grid node size. The choice of a grid node size for the future benchmarks and experiments is a trade-off between precision and efficiency. We have chosen $h = 0.5$ Bohr and a 6^{th} order Laplacian because the computational times are reasonable (an hour for medium size

systems, 1000 atoms) and they are common values in RS-FD implementations⁸. From figure 4, we predict errors around one percent for electronic energies. The $h = 0.5$ Bohr node size corresponds roughly to the 600 eV energy cut in Fourier space. For other tests involving more computations, bigger step sizes will also be considered ($h = 0.7$ Bohr).

Precision of the method

From the parameters proposed earlier, we show that our implementation computes correct energies for aluminum systems. We use the same methodology as the one proposed in⁸: 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 and equilibrium bond lengths of aluminum dimer and trimer. The results are presented in table 1. With the parameters used, we have a relative error below 2% for the binding energy and around 1% for bond lengths.

We then compute the equilibrium lattice of different aluminum clusters: $1 \times 1 \times 1$, $3 \times 3 \times 3$ and $5 \times 5 \times 5$ faces centered cubic (FCC) crystals of respectively $M = 14$, 172 and 666 atoms. Their binding energy is computed for different lattices between $a = 7.2$ Bohrs and $a = 8.4$ Bohrs and the minimum is retrieved with a cubic regression. The binding energies and the equilibrium bond lengths are compared in table 2, we also find a relative error below 2% for the binding energies and below 1% for the equilibrium lattice.

Finally, we derive the bulk cohesive energy ε_{coh} of FCC Aluminum with a lattice parameter $a = 8$ Bohrs from the binding energies of finite crystals of increasing size ($M = 14, 172, 365, 666, 1099, 1688$). 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 ε_b can be approximated as:

$$\varepsilon_b = \varepsilon_{coh} + a_{sides}M^{-\frac{1}{3}} + a_{edge}M^{-\frac{2}{3}} + a_{corner}M^{-1} \quad (43)$$

In figure 5 we show $\varepsilon_b(M^{-\frac{1}{3}})$ and its linear extrapolation. With our implementation, we find $\varepsilon_{coh} = -2.437$ eV/atom, just like the value given by PROFESS for a periodic FCC crystal: $\varepsilon_{coh} = -2.437$ eV/atom.

Our implementation, with a grid node size of $h = 0.5$ Bohr, computes energies with errors below 2% and bond lengths with errors below 1% compared to PROFESS. Those errors also decrease with the refinement of h .

Now we investigate the computational time of an the electron density calculation and update.

Performance

We evaluate here the performance of our method. First we stop on the Hartree potential computation because the method is a new mix between two classical solutions. Then we show and explain the speedup achieved by our OF-DFT method in dynamical simulation.

Hartree Potential computation

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 algorithms⁴⁸ coupled with a conjugate-gradient relaxation. As explained in⁴⁹, a full multi-grid method has a complexity in $O(N)$, a conjugate-gradient (CG) typically in $O(N^{\frac{5}{4}})$, and a method coupling both would have a complexity in between. On figure 6, 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 . The convergence condition here is $|e_r|_2 < 10^{-8}$. When using one core, we see appear 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^{50,51} 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 deserves to be tested

and compared with ours, for example the work⁵² is certainly more efficient and hence could be the next step of this work, maybe with an adaptation for small electron density increments.

Speedup of the incremental update

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 Bohrs and a node size of 0.7 Bohrs then move p atoms, randomly chosen, of 0.3 \AA 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_{at}^0 . During the incremental update, we pause computations between the restrained update and the global one to record the atomic energy E_{at}^2 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_{at}^2 to E_{at}^0 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 7 shows the evolution of relative errors $\Delta E_{at}^2 = \left| \frac{E_{at}^0 - E_{at}^2}{E_{at}^0} \right|$ when the loose density sub-area radius r_ω increases. At $r_\omega = 4$ Bohrs the errors reach 0.5% and for $r_\omega \geq 8$ Bohrs 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 8 Bohrs corresponds roughly to the distance from the nucleus at which the atom's effect on the global electron density vanishes.

In figures 8 and 9 are drawn the speedups $\frac{t_0}{t_2}$ with respect to r_ω and to the relative volume of the loose density sub-area used for the incremental update $v = \frac{|\omega|}{|\Omega|}$, $|\omega|$ and $|\Omega|$ being the volumes of respectively the loose density sub-area and of the whole 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 8. A proper

measure to estimate the update speed is the relative volume of ω : $v = \frac{|\omega|}{|\Omega|}$. 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 9 the data from our simulations and the linear regression 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 figure 10, the dots are the five tests means and the vertical bars their standard deviations. For $r_\omega \leq 6$ Bohrs, the speedup of the restrained update, figure 8, is lost because, as the errors were important, figure 7, their correction takes more time. 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 $r_\omega = 8$ Bohrs or $r_\omega = 10$ Bohrs depending on the number of atoms moved. We here have a maximum 20-fold speedup.

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 sub-area 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.

Aluminum implantation

In this last section we demonstrate the use of incremental density update for molecular simulations.

We simulate an atom implantation on an aluminum plate, we aim to be representative of doping by ion implantation⁵³: 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 \AA above the plate and thrown vertically toward it at 40 pm/fs . We chose a grid node size $h = 1 \text{ Bohr}$

and margins of 15 Bohrs 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 $r_\omega = 10$ Bohrs. 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 openMP⁵⁴.

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 $\varepsilon^f = 5.10^{-2}$ 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. Five frames of the implantation with $\varepsilon^f = 5.10^{-3}$ eV are in figure 11. They are colored first with their displacement then with their deviation from the reference simulation.

In figure 12, we drew the representative energies of each implantation: on top the kinetic and below the 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 ($\varepsilon^f \leq 0.005$ 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 $O(pN)$ with a small pre-factor, the local density update, scaling in $O(|\omega|) \sim O(p)$ 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 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 13 with respect to the simulation time, and in figure 14 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.

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 bi-cores and a Linux System, with a similar grid 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 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 twenty-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.

CONCLUSIONS

This paper proposes a new variant of the OF-DFT method. It relies on computations in real space, allowing reevaluations of the electron density on specific fractions of the space. Hence, this method can be combined with restrained dynamics methods for accelerating simulations. We have conducted dynamical simulations of two thousands atoms based on OF-DFT model in less than 5 hours. Although the speedup is promising, the method presented still misses a proper computation in real space of kinetic energies that exhibit better accuracy. The best choice would be a non-local KEDF like Wang-Teter or Wang-

Govind-Carter functionals that can have accuracies close to KS-DFT in specific cases⁶, we have tried a Density-Independent non-local KEDF with the method explained in^{33,34} but the stability and speed of our implementation were greatly decreased so we chose to stay with a $TF - \lambda vW$ KEDF. Another KEDF that keeps computations local has been tested in⁵⁵, it seems more adapted for real-space and incremental implementations of OF-DFT and is among the next steps for our solver.

ACKNOWLEDGMENTS

We gratefully acknowledge funding from the European Research Council through the ERC Starting Grant n. 307629.

References

1. E. Neveu, D. Ritchie, P. Popov, and S. Grudinin, “PEPSI-Dock: a detailed data-driven protein–protein interaction potential accelerated by polar Fourier correlation,” *Bioinformatics*, vol. 32, pp. i693–i701, Aug. 2016.
2. G. S. Ho, V. L. Lignères, and E. A. Carter, “Introducing profess: A new program for orbital-free density functional theory calculations,” *Computer Physics Communications*, vol. 179, pp. 839–854, dec 2008.
3. A. K. Rappé, C. J. Casewit, K. Colwell, W. A. Goddard III, and W. Skiff, “Uff, a full periodic table force field for molecular mechanics and molecular dynamics simulations,” *Journal of the American chemical society*, vol. 114, no. 25, pp. 10024–10035, 1992.
4. H. Nakashima and H. 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*, vol. 127, no. 22, p. 224104, 2007.
5. D. Marx and J. Hutter, *Ab initio molecular dynamics: basic theory and advanced methods*. Cambridge: Cambridge University Press, 2009.
6. J. Xia, C. Huang, I. Shin, and E. A. Carter, “Can orbital-free density functional theory simulate molecules?,” *The Journal of chemical physics*, vol. 136, no. 8, p. 084102, 2012.
7. G. S. Ho and E. A. Carter, “Mechanical response of aluminum nanowires via orbital-free density functional theory,” *Journal of Computational and Theoretical Nanoscience*, vol. 6, no. 6, pp. 1236–1246, 2009.
8. P. Suryanarayana and D. Phanish, “Augmented lagrangian formulation of orbital-free density functional theory,” *Journal of Computational Physics*, vol. 275, pp. 524–538, oct 2014.
9. S. Artemova and S. Redon, “Adaptively Restrained Particle Simulations,” *Physical Review Letters*, vol. 109, pp. 190201:1–5, Nov. 2012.

10. S. Artemova, *Adaptive algorithms for molecular simulation*. PhD thesis, Université de Grenoble, 2012.
11. M. Bosson, C. Richard, A. Plet, S. Grudinin, and S. Redon, “Interactive quantum chemistry: A divide-and-conquer ASED-MO method,” *Journal of Computational Chemistry*, vol. 33, pp. 779–790, Mar. 2012.
12. M. Bosson, S. Grudinin, and S. Redon, “Block-Adaptive Quantum Mechanics: An Adaptive Divide-and-Conquer Approach to Interactive Quantum Chemistry,” *Journal of Computational Chemistry*, vol. 34, pp. 492–504, Mar. 2013.
13. S. P. A. Edoth and S. Redon, “Incremental update of electrostatic interactions in adaptively restrained particle simulations,” *Journal of Computational Chemistry*, vol. 39, pp. 1455–1469, Apr. 2018.
14. K. K. Singh, D. F. Marin, and S. Redon, “Parallel Adaptively Restrained Molecular Dynamics,” in *2017 International Conference on High Performance Computing & Simulation (HPCS)*, (Genova, Italy), pp. 308 – 314, IEEE, July 2017.
15. K. K. Singh and S. Redon, “Adaptively Restrained Molecular Dynamics in LAMMPS,” *Modelling and Simulation in Materials Science and Engineering*, vol. 25, p. 055013, June 2017.
16. S. Redon, G. Stoltz, and Z. Trstanova, “Error analysis of modified langevin dynamics,” *Journal of Statistical Physics*, vol. 164, no. 4, pp. 735–771, 2016.
17. Z. Trstanova, *Mathematical and Algorithmic Analysis of Modified Langevin Dynamics*. Theses, Université Grenoble Alpes, Nov. 2016.
18. D. Frenkel and B. Smit, “Chapter 3 - monte carlo simulations,” in *Understanding Molecular Simulation (Second Edition)* (D. Frenkel and B. Smit, eds.), pp. 23 – 61, San Diego: Academic Press, second edition ed., 2002.
19. M. Schütz and H.-J. Werner, “Low-order scaling local electron correlation methods. iv. linear scaling local coupled-cluster (lccsd),” *The Journal of Chemical Physics*, vol. 114, no. 2, p. 661, 2001.

20. M. Schütz, “A new, fast, semi-direct implementation of linear scaling local coupled cluster theory,” *Phys. Chem. Chem. Phys.*, vol. 4, no. 16, pp. 3941–3947, 2002.
21. P. Hohenberg and W. Kohn, “Inhomogeneous electron gas,” *Physical Review*, vol. 136, pp. B864–B871, nov 1964.
22. P. E. Blöchl, “Projector augmented-wave method,” *Physical review B*, vol. 50, no. 24, p. 17953, 1994.
23. P. E. Blöchl, C. J. Först, and J. Schimpl, “Projector augmented wave method: ab initio molecular dynamics with full wave functions,” *Bulletin of Materials Science*, vol. 26, no. 1, pp. 33–41, 2003.
24. W. Kohn and L. J. Sham, “Self-consistent equations including exchange and correlation effects,” *Physical Review*, vol. 140, pp. A1133–A1138, nov 1965.
25. L. H. Thomas, “The calculation of atomic fields,” *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 23, p. 542, jan 1927.
26. C. v. Weizsäcker, “Zur theorie der kernmassen,” *Zeitschrift für Physik A Hadrons and Nuclei*, vol. 96, no. 7, pp. 431–458, 1935.
27. L.-W. Wang and M. P. Teter, “Kinetic-energy functional of the electron density,” *Physical Review B*, vol. 45, pp. 13196–13220, jun 1992.
28. J. Lindhard, “On the properties of a gas of charged particles,” *Kgl. Danske Videnskab. Selskab Mat.-Fys. Medd.*, vol. 28, 1954.
29. M. Foley and P. A. Madden, “Further orbital-free kinetic-energy functionals for ab initio molecular dynamics,” *Physical Review B*, vol. 53, no. 16, p. 10589, 1996.
30. Y. A. Wang, N. Govind, and E. A. Carter, “Orbital-free kinetic-energy functionals for the nearly free electron gas,” *Physical Review B*, vol. 58, no. 20, p. 13465, 1998.
31. Y. A. Wang, N. Govind, and E. A. Carter, “Orbital-free kinetic-energy density functionals with a density-dependent kernel,” *Physical Review B*, vol. 60, no. 24, p. 16350, 1999.

32. C. Huang and E. A. Carter, “Toward an orbital-free density functional theory of transition metals based on an electron density decomposition,” *Physical Review B*, vol. 85, no. 4, p. 045126, 2012.
33. V. Gavini, J. Knap, K. Bhattacharya, and M. Ortiz, “Non-periodic finite-element formulation of orbital-free density functional theory,” *Journal of the Mechanics and Physics of Solids*, vol. 55, no. 4, pp. 669–696, 2007.
34. S. Ghosh and P. Suryanarayana, “Higher-order finite-difference formulation of periodic orbital-free density functional theory,” *Journal of Computational Physics*, vol. 307, pp. 634–652, 2016.
35. P. Ziesche, S. Kurth, and J. P. Perdew, “Density functionals from lda to gga,” *Computational materials science*, vol. 11, no. 2, pp. 122–127, 1998.
36. X. Shao, Q. Xu, S. Wang, J. Lv, Y. Wang, and Y. Ma, “Large-scale ab initio simulations for periodic system,” *Computer Physics Communications*, vol. 233, pp. 78–83, 2018.
37. S. Wright and J. Nocedal, “Numerical optimization,” *Springer Science*, vol. 35, no. 67-68, p. 7, 1999.
38. C. Huang and E. A. Carter, “Transferable local pseudopotentials for magnesium, aluminum and silicon,” *Physical Chemistry Chemical Physics*, vol. 10, no. 47, p. 7109, 2008.
39. C. Huang and E. A. Carter, “Nonlocal orbital-free kinetic energy density functional for semiconductors,” *Phys. Rev. B*, vol. 81, p. 045206, Jan 2010.
40. B. Zhou, Y. A. Wang, and E. A. Carter, “Transferable local pseudopotentials derived via inversion of the kohn-sham equations in a bulk environment,” *Physical Review B*, vol. 69, no. 12, p. 125109, 2004.
41. P. Viot, “Méthode d’analyse numérique..” Lecture, Sept. 2006.
42. W. Yang, “Gradient correction in thomas-fermi theory,” *Physical Review A*, vol. 34, no. 6, p. 4575, 1986.

43. R. G. Parr and Y. Weitao, *Density-functional theory of atoms and molecules*, vol. 16. USA: Oxford university press, 1994.
44. N. Govind, J. Wang, and H. Guo, “Total-energy calculations using a gradient-expanded kinetic-energy functional,” *Physical Review B*, vol. 50, pp. 11175–11178, oct 1994.
45. J. P. Perdew and A. Zunger, “Self-interaction correction to density-functional approximations for many-electron systems,” *Physical Review B*, vol. 23, pp. 5048–5079, may 1981.
46. P. Motamarri, M. Iyer, J. Knap, and V. Gavini, “Higher-order adaptive finite-element methods for orbital-free density functional theory,” *Journal of Computational Physics*, vol. 231, no. 20, pp. 6596–6621, 2012.
47. N. Choly and E. Kaxiras, “Kinetic energy density functionals for non-periodic systems,” *Solid State Communications*, vol. 121, pp. 281–286, feb 2002.
48. T. Guillet and R. Teyssier, “A simple multigrid scheme for solving the poisson equation with arbitrary domain boundaries,” *Journal of Computational Physics*, vol. 230, no. 12, pp. 4756–4771, 2011.
49. D. Braess, “On the combination of the multigrid method and conjugate gradients,” in *Multigrid Methods II* (W. Hackbusch and U. Trottenberg, eds.), (Berlin, Heidelberg), pp. 52–64, Springer Berlin Heidelberg, 1986.
50. C. Temperton, “Direct methods for the solution of the discrete poisson equation: some comparisons,” *Journal of Computational Physics*, vol. 31, no. 1, pp. 1–20, 1979.
51. W. Gander and G. H. Golub, “Cyclic reduction—history and applications,” *Scientific computing (Hong Kong, 1997)*, no. 1, pp. 73–85, 1997.
52. L. Genovese, T. Deutsch, A. Neelov, S. Goedecker, and G. Beylkin, “Efficient solution of poisson’s equation with free boundary conditions,” *The Journal of chemical physics*, vol. 125, no. 7, p. 074105, 2006.

53. J. F. Gibbons, “Ion implantation in semiconductors—part ii: Damage production and annealing,” *Proceedings of the IEEE*, vol. 60, no. 9, pp. 1062–1096, 1972.
54. L. Dagum and R. Menon, “Openmp: an industry standard api for shared-memory programming,” *IEEE Computational Science and Engineering*, vol. 5, pp. 46–55, Jan 1998.
55. A. Borgoo, J. A. Green, and D. J. Tozer, “Molecular binding in post-kohn–sham orbital-free dft,” *Journal of chemical theory and computation*, vol. 10, no. 12, pp. 5338–5345, 2014.

Figure 1: 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 $100 \times 100 \times 100$.

Figure 2: Radial electron density of a single Aluminum ion and its decimal logarithm. The density vanishes below 10^{-10} e/Bohr³ for $r = 12.2$ Bohr.

Figure 3: Atomic Energy Error, log scale, of a 666 atoms Aluminum cluster with respect to the spatial discretization h and the Laplacian order used for the computation of system's kinetic energy and Hartree potential. The reference $E_{at,0} = -2.15866$ eV/atom is the atomic energy computed with PROFESS. We observe a net jump in precision between the 2nd and 6th order Laplacian, respectively the circles and the triangles, hence the choice of a 6th order Laplacian for the method.

Figure 4: Relative error, log scale, for the electronic energy calculation of a single atom with respect to the grid node size h . On the left for the hydrogen atom with $E_{el,0} = -0.5$ Hartree, the theoretical electronic energy, and with OF-DFT functionals adapted to a one-electron system. On the right for the aluminum atom with $E_{el,0} = -2.104$ Hartree, the value computed with PROFESS. Our method achieves arbitrary precision with the proper discretization, and with the aluminum atom, a one per cent precision for $h = 0.5$ Bohr.

Figure 5: Binding Energy of Aluminum FCC cluster of different sizes M , with respect to $M^{-\frac{1}{3}}$, and the linear extrapolation to $M^{-\frac{1}{3}} = 0$ that gives the cohesive energy of a periodic FCC Aluminum crystal. We find the same cohesive energy than PROFESS ($\varepsilon_b = -2.437$ eV/atom).

Figure 6: Wall-clock times of the Poisson solver used to compute Hartree potential. We compute the potential created by the density guess used for one Aluminum atom, a Gaussian function, at different grid sizes N . On one core, the complexity follows the one of Conjugate-Gradient method, in $O(N^{\frac{5}{4}})$. On several cores the computational times exhibit an apparent linear complexity surely due to the reduction of parallel computation overhead contribution.

Figure 7: Relative error of a restrained update with respect to the loose density sub-area radius r_ω , in Angström, and in gray the radial electron density of a lone aluminum atom.

Figure 8: Speedup, log scale, brought by a restrained update with respect to the loose density sub-area radius r_ω , the colors are for the number of moved atoms p .

Figure 9: Speedup, log scale, brought by a restrained update with respect to the updated sub-area's volume ratio, $v = \frac{|\omega|}{|\Omega|}$, the colors are the number of moved atoms 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 10: 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 moved atoms, 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.

Figure 11: Five frames of the aluminum implantation simulation, with restraining parameters $(\varepsilon^f, \varepsilon^r) = (0.005, 0.004)$ eV. The first lines is colored with the displacement of particles from their initial position, the second with the deviation from the reference position, $(\varepsilon^f = \varepsilon^r = 0)$ eV. The whole simulation took five hours on a 20-cores computer.

Figure 12: 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.

Figure 13: Time required to update electron density at a particular 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.

Figure 14: 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.

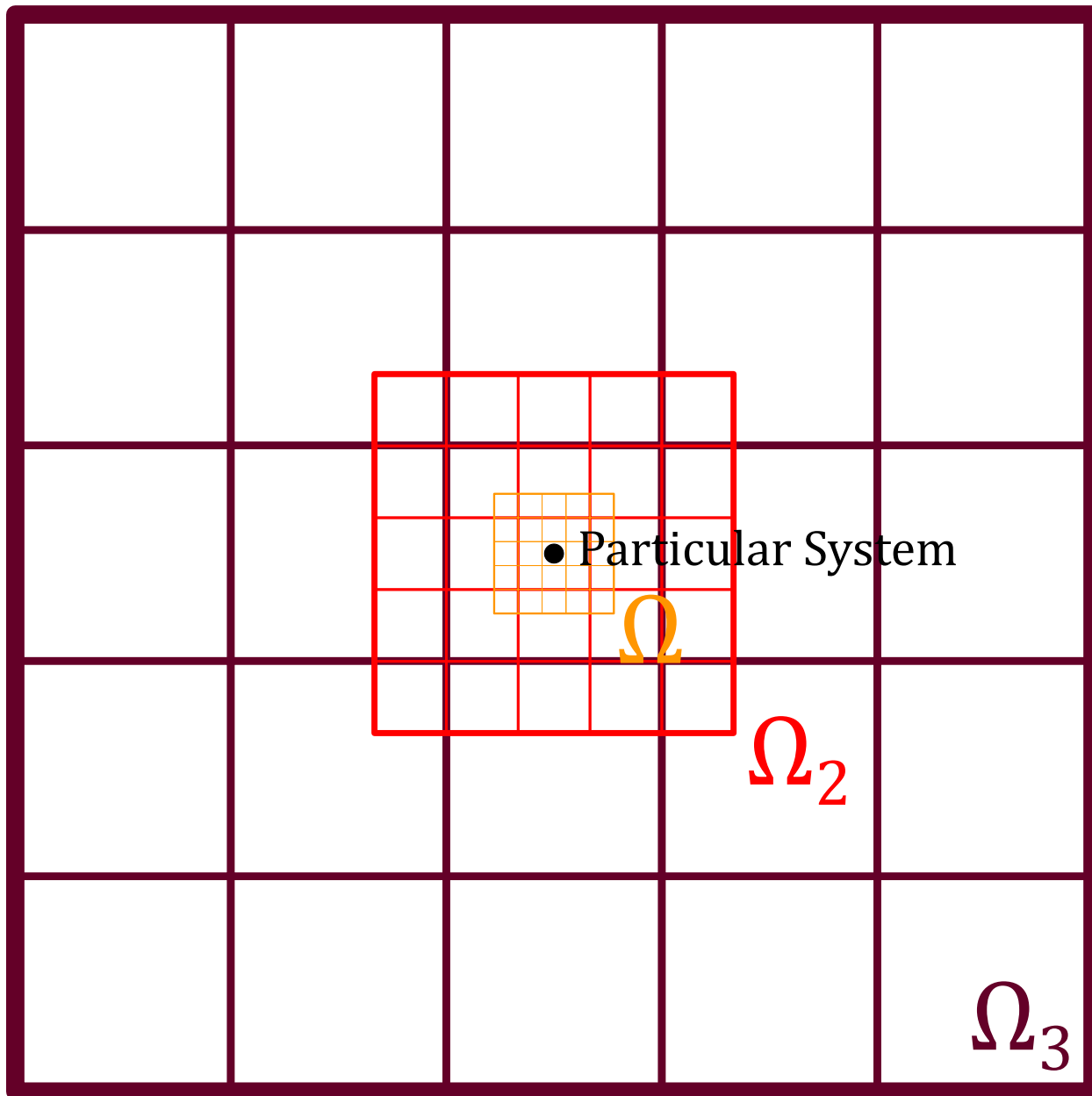


Figure 1
François Rousse and Stéphane
Redon
J. Comput. Chem.

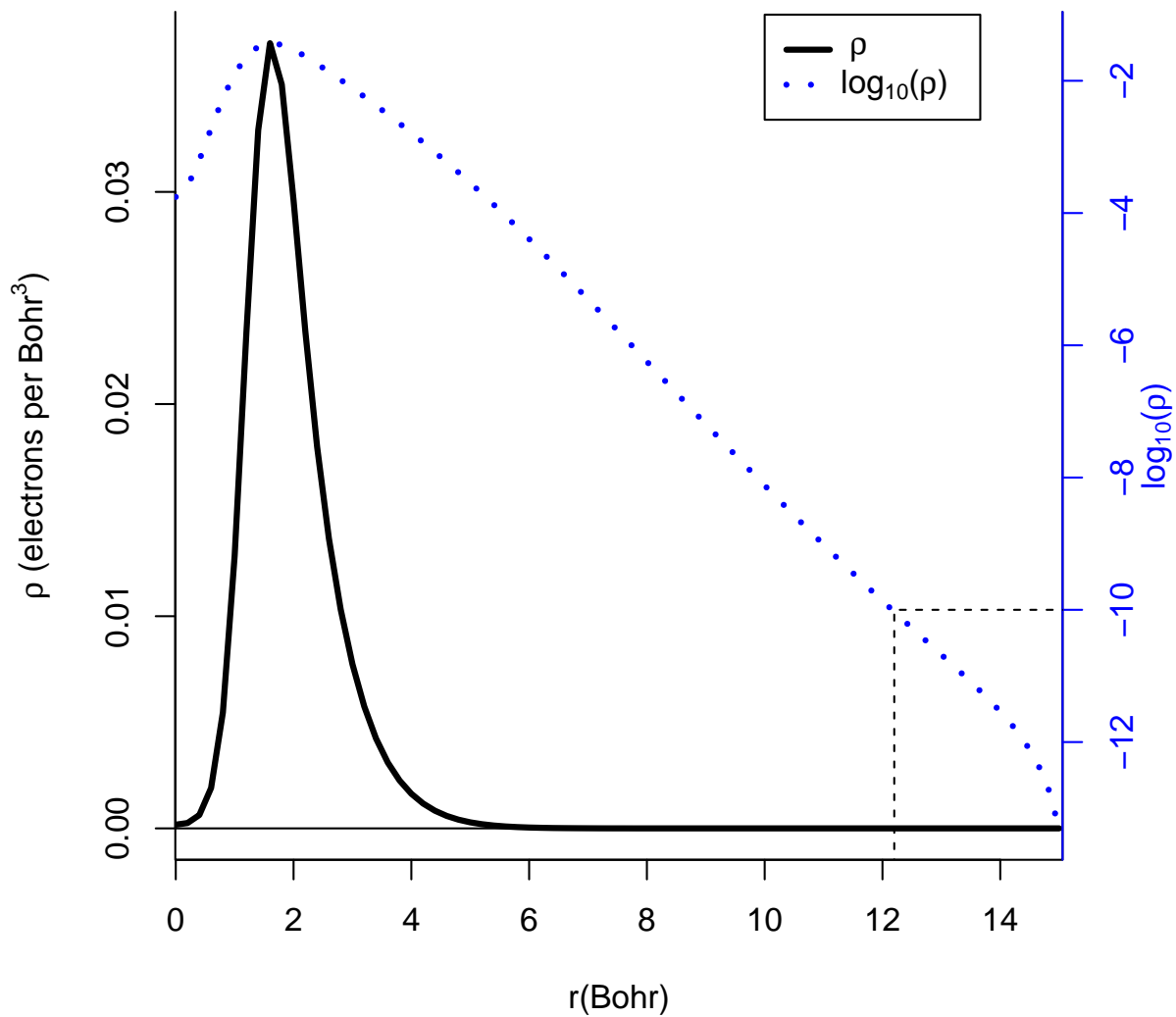


Figure 2
François Rouse and Stéphane
Redon
J. Comput. Chem.

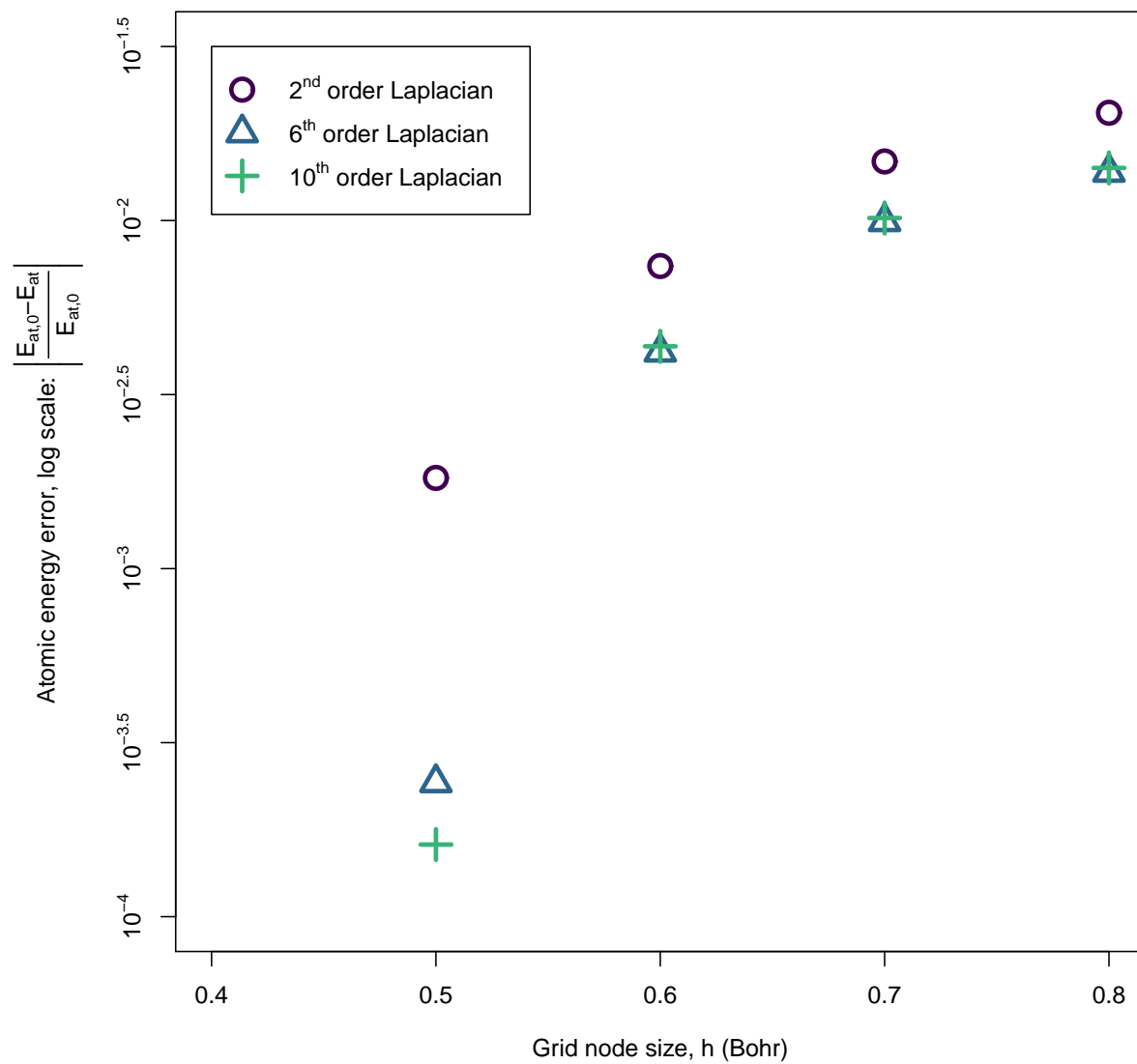


Figure 3
François Rouse and Stéphane Redon
J. Comput. Chem.

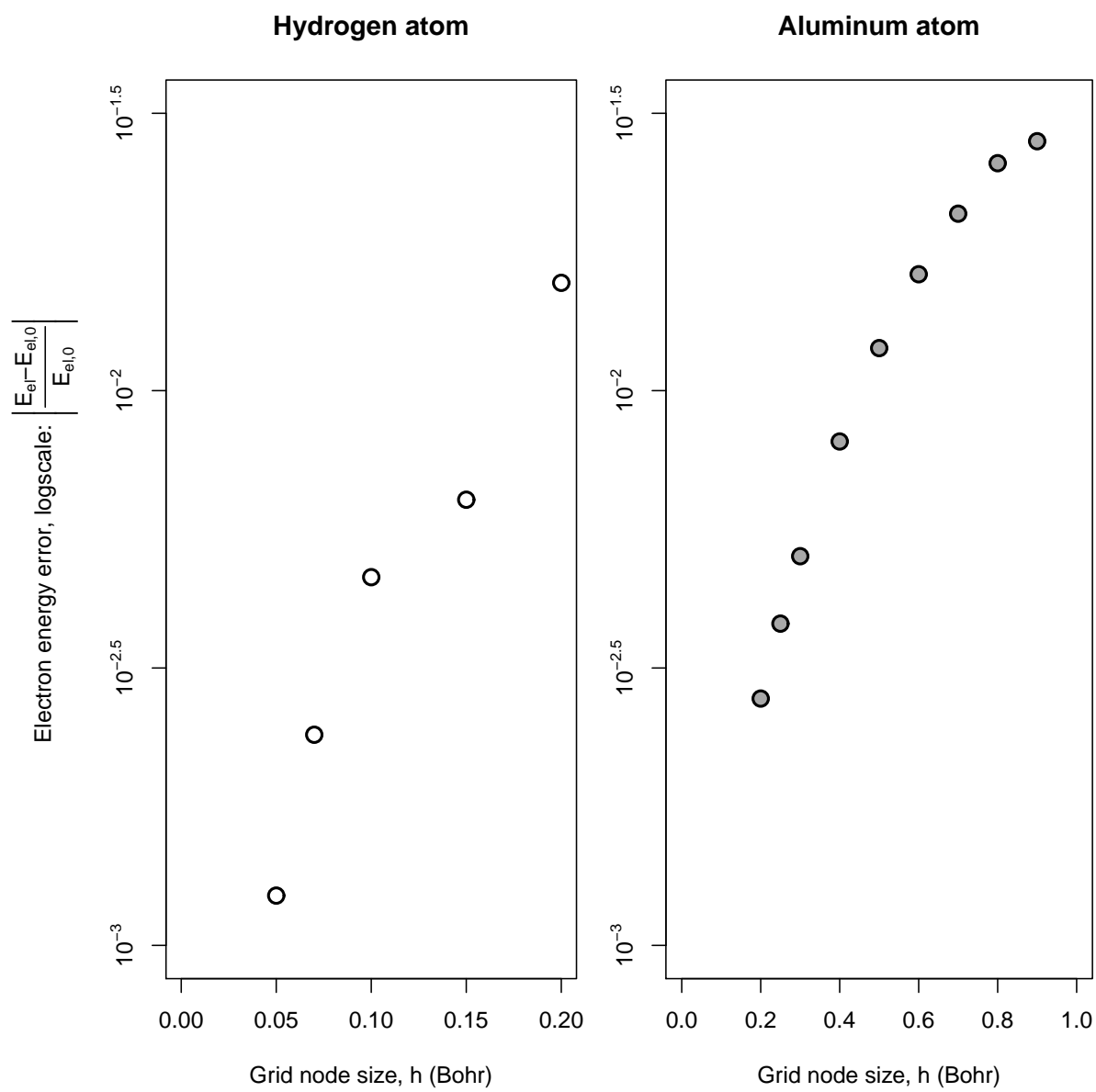


Figure 4
 François Rouse and Stéphane Redon
 J. Comput. Chem.

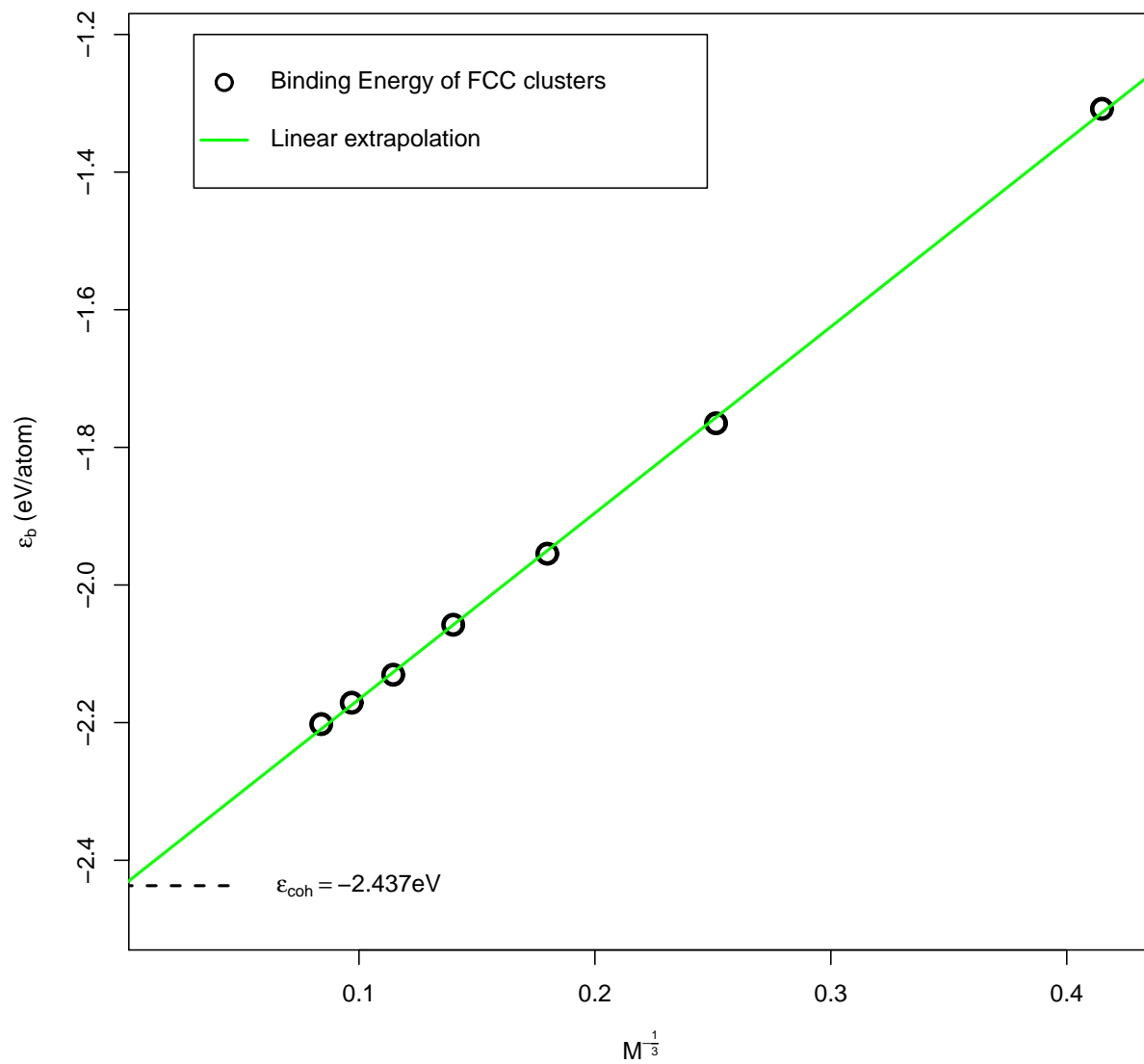


Figure 5
François Rouse and Stéphane
Redon
J. Comput. Chem.

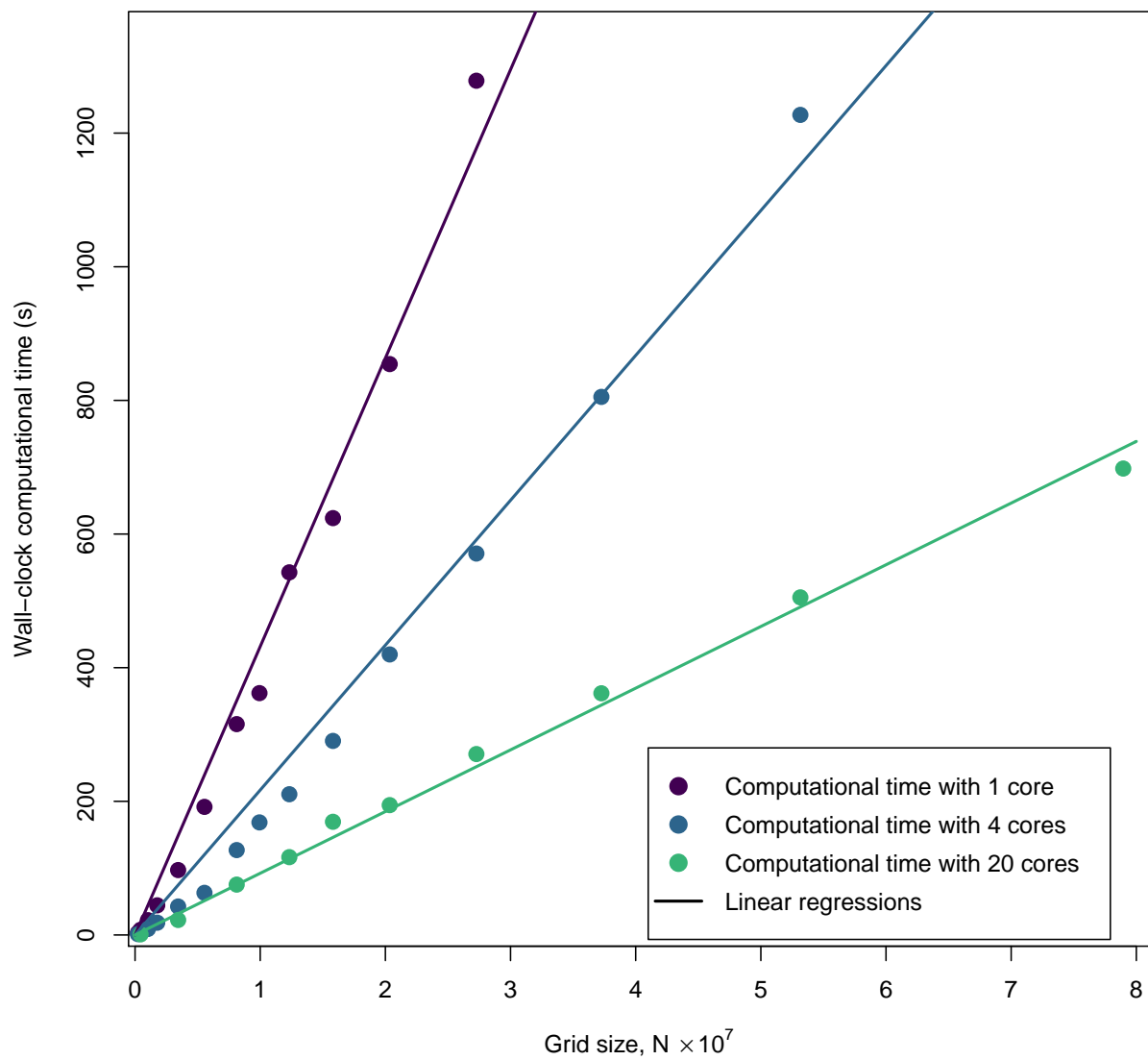


Figure 6
 François Rouse and Stéphane Redon
 J. Comput. Chem.

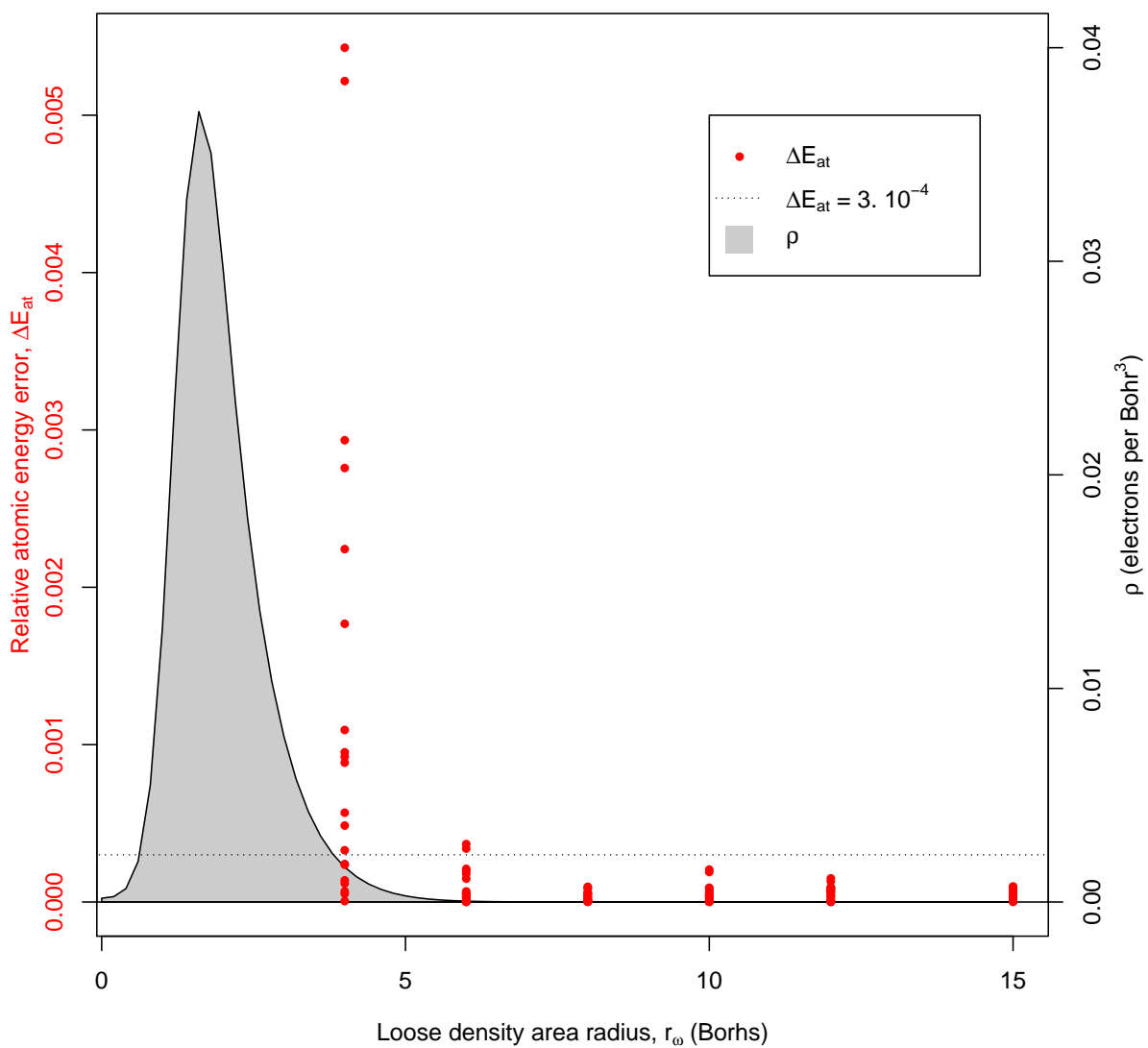


Figure 7
 François Rouse and Stéphane
 Redon
 J. Comput. Chem.

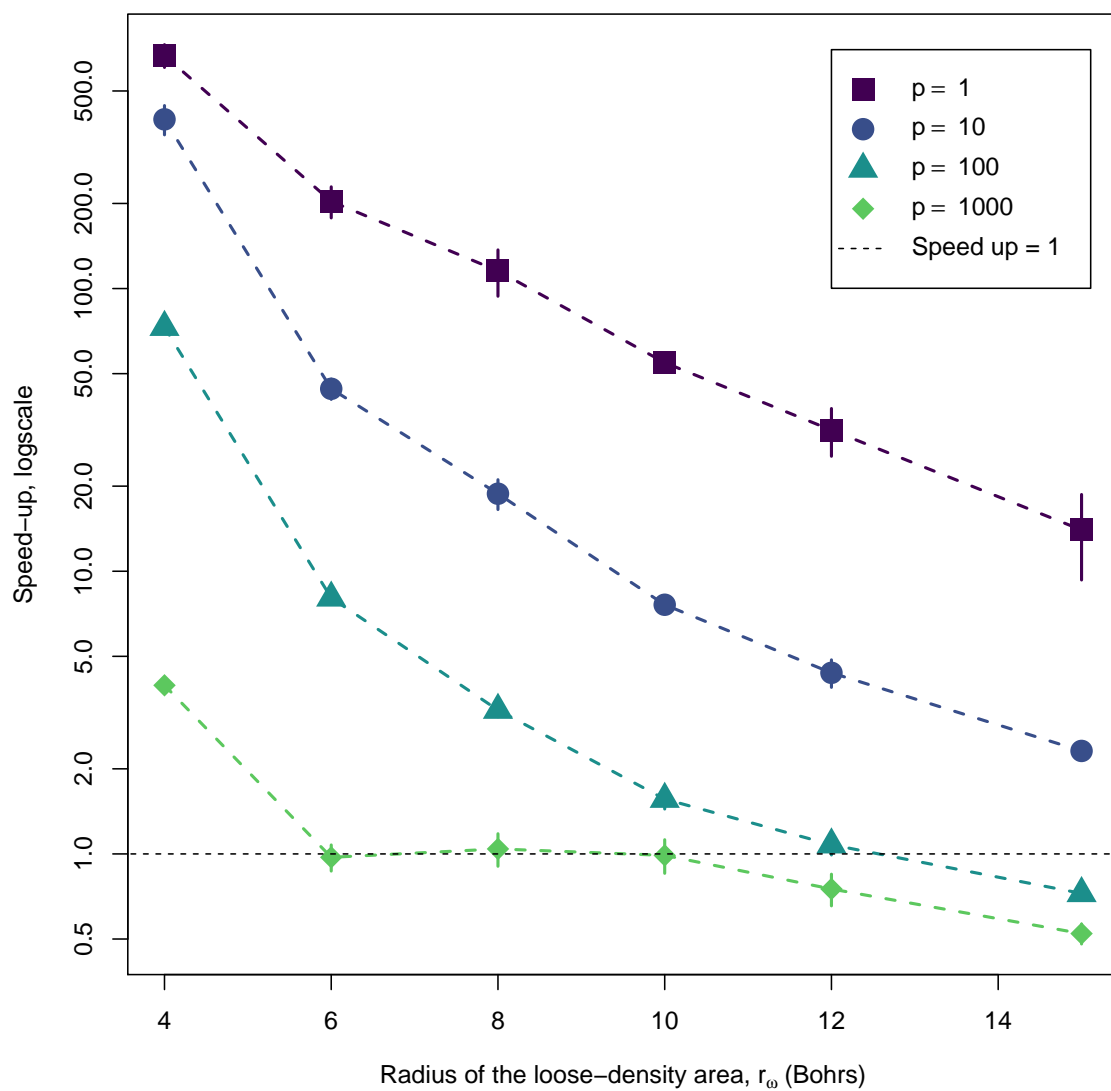


Figure 8
 François Rouse and Stéphane Redon
 J. Comput. Chem.

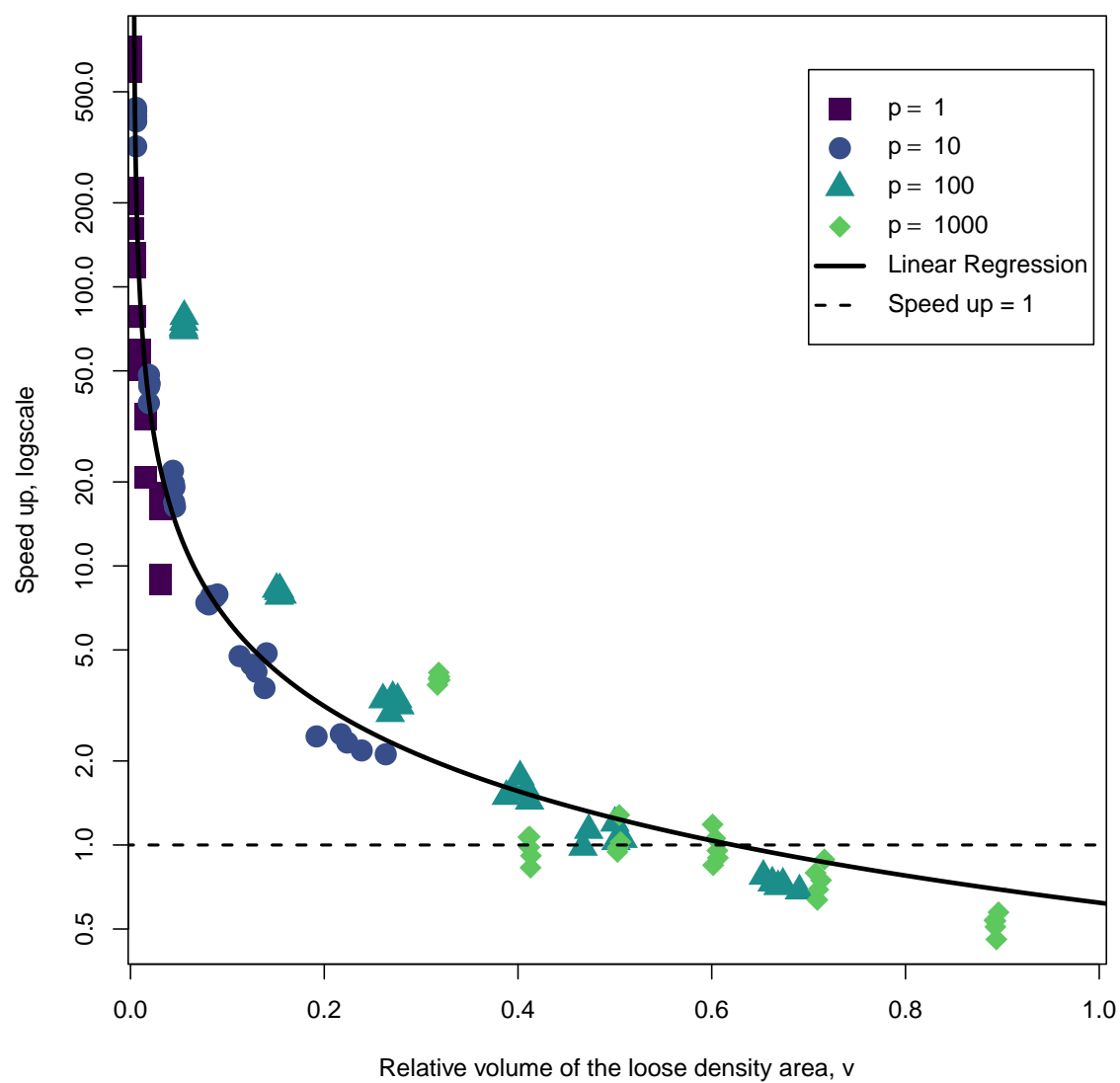


Figure 9
 François Rouse and Stéphane Redon
 J. Comput. Chem.

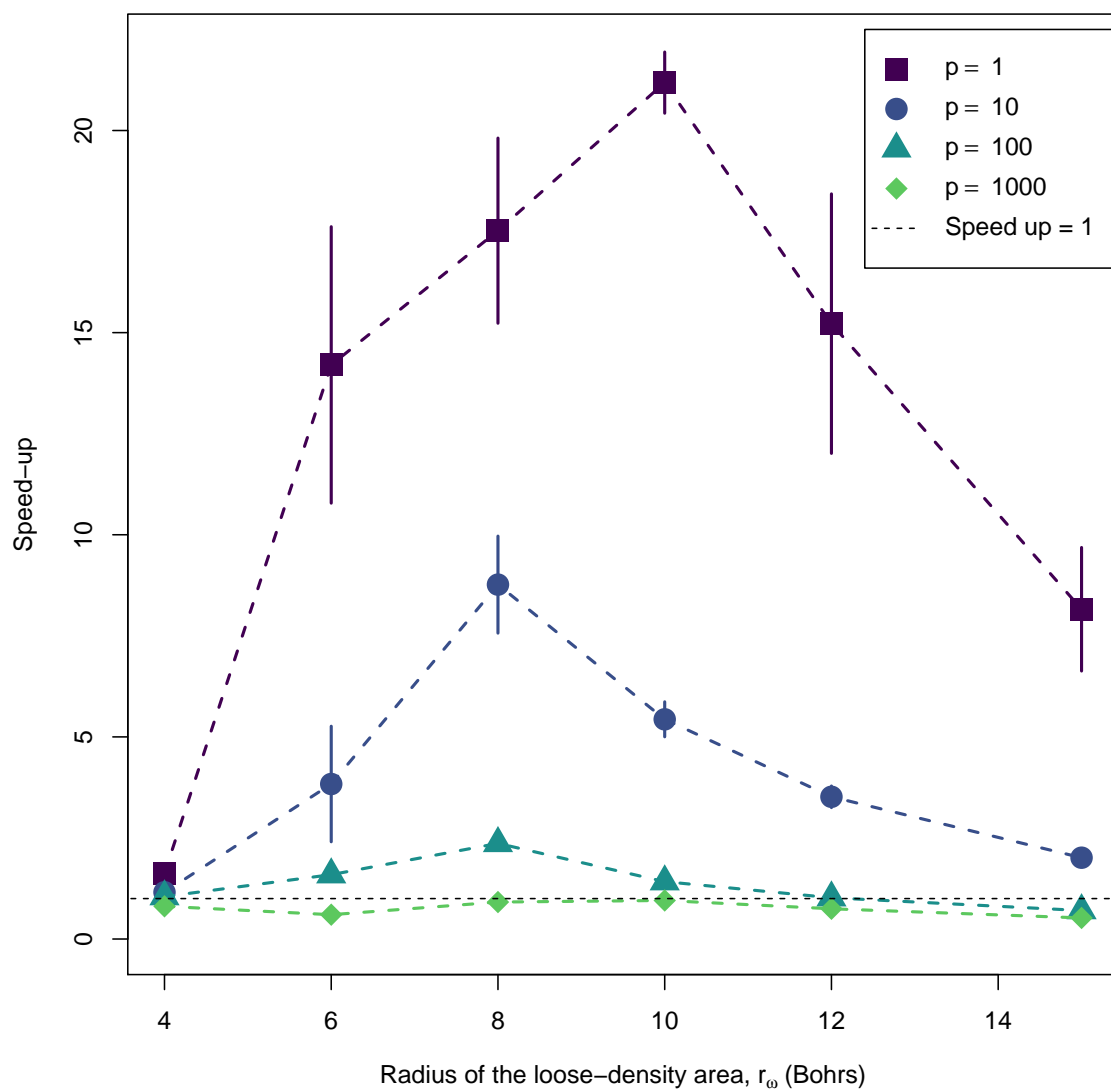


Figure 10
François Rouse and Stéphane Redon
J. Comput. Chem.

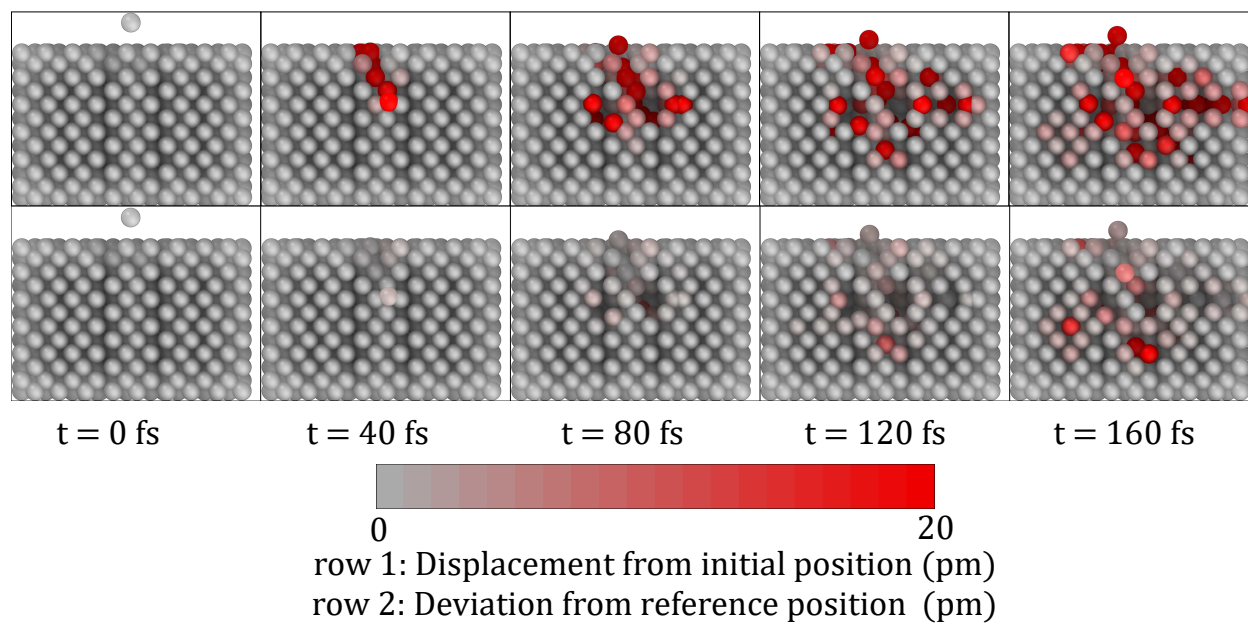


Figure 11
 François Rousse and Stéphane
 Redon
 J. Comput. Chem.

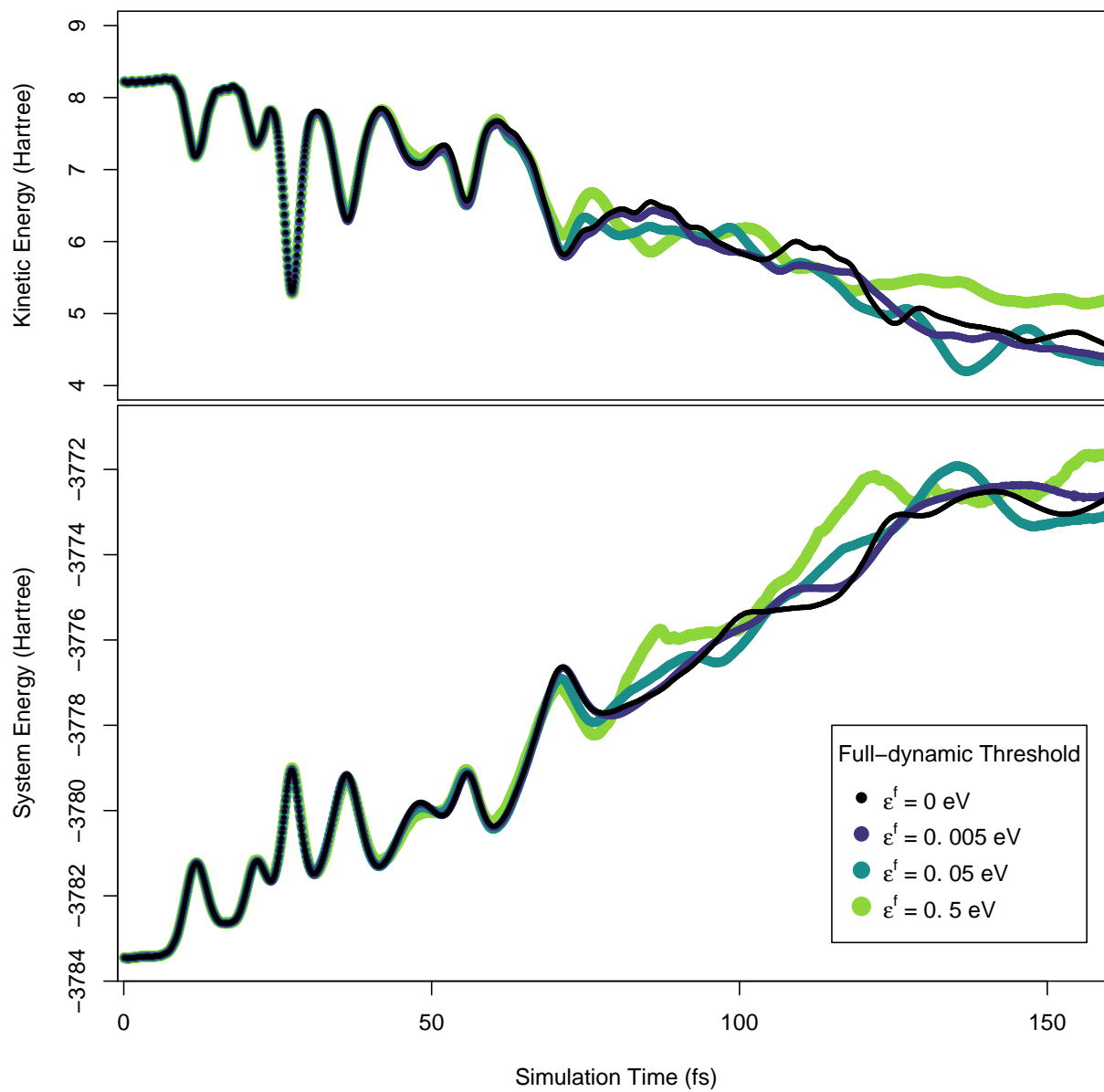


Figure 12
 François Rouse and Stéphane
 Redon
 J. Comput. Chem.

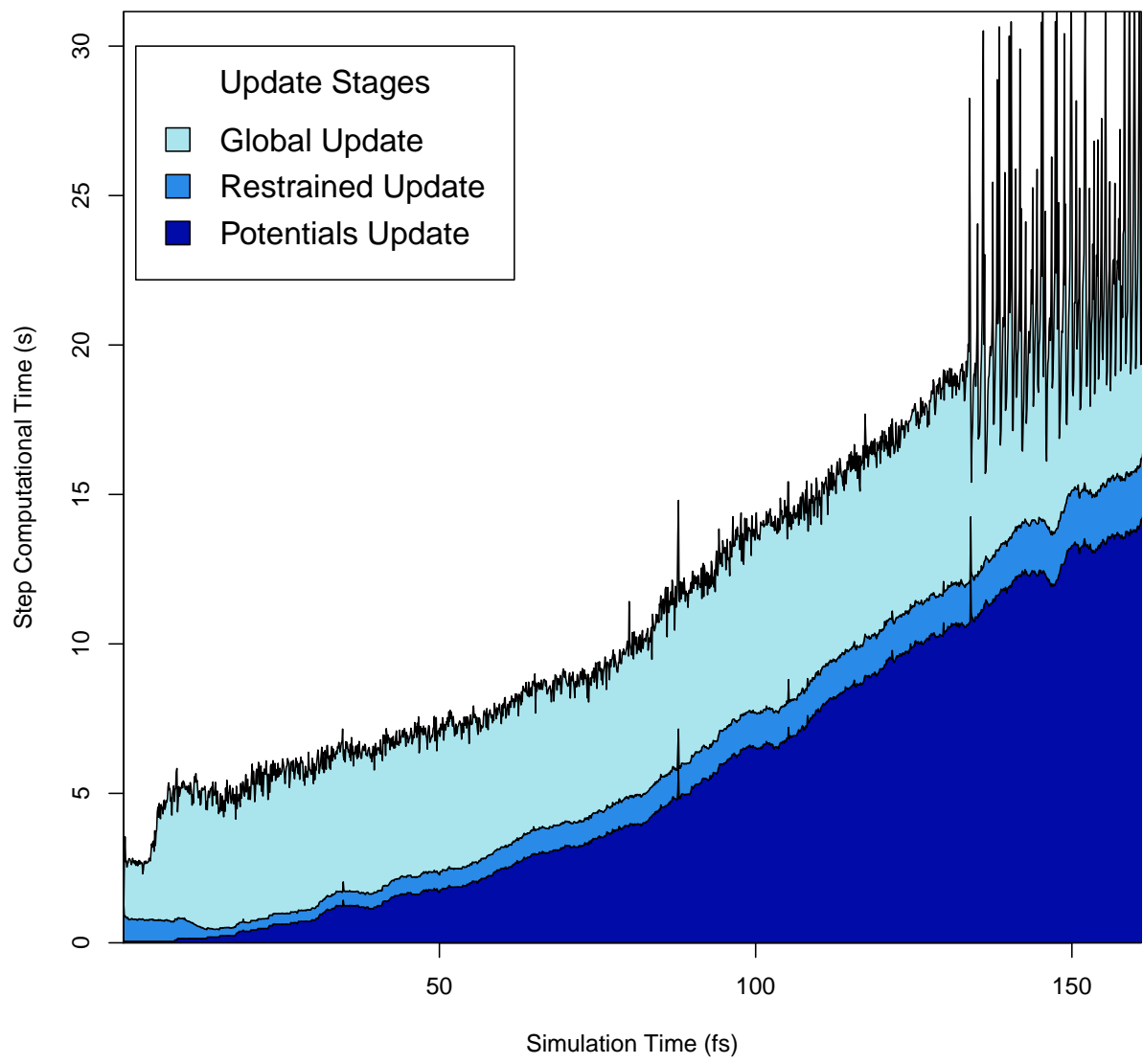


Figure 13
François Rouse and Stéphane Redon
J. Comput. Chem.

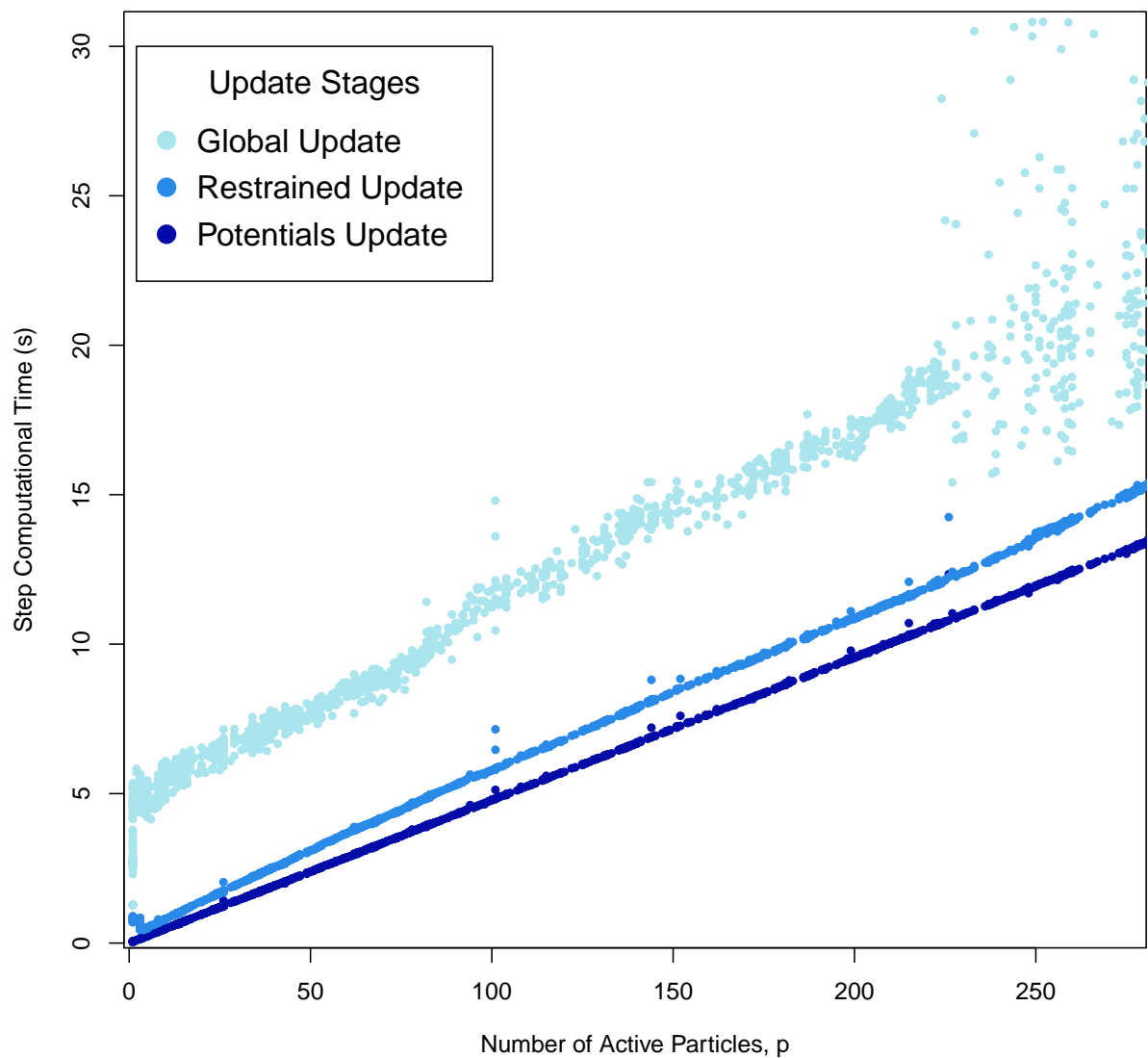


Figure 14
François Rouse and Stéphane Redon
J. Comput. Chem.

M	$\varepsilon_b(\text{eV/at})$		error(%)	$a_e(\text{Bohr})$		error(%)
	RS-FD	PROFESS		RS-FD	PROFESS	
Al ₂	-0.382	-0.384	0.52	5.09	5.07	0.39
Al ₃	-0.644	-0.649	0.77	5.25	5.18	1.3

Table 1: Binding Energy ε_b and equilibrium bond length R_e for Al_2 and Al_3

M	$\varepsilon_b(\text{eV/at})$		error(%)	$a_e(\text{Bohr})$		error(%)
	RS-FD	PROFESS		RS-FD	PROFESS	
14	-1.326	-1.353	2.0	7.61	7.554	0.74
172	-1.949	-1.984	1.8	7.77	7.767	0.04
666	-2.158	-2.160	0.09	7.86	7.814	0.59

Table 2: Binding energy (ε_b) and equilibrium lattice constant (a_e) for FCC clusters of M Aluminum atoms