
Real Space DFT by Locally Optimal Block Preconditioned Conjugate Gradient Method

Vincent Michaud-Rioux
Center for the Physics of Materials
Department of Physics
McGill University
Montreal, Quebec
2012

A Thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Science

CONTENTS

Abstract	vii
Résumé	ix
Statement of Originality	xi
Acknowledgments	xii
1 Introduction	1
2 Theory	5
2.1 Density Functional Theory	5
2.2 Kohn-Sham Self-Consistent Field Equations	11
2.3 Exchange-Correlation Energy Functionals	14
2.4 Summary	16
3 Real Space Pseudopotential Method	18
3.1 Real space and conventional methods	18
3.2 Real space pseudopotential formulation of the Kohn-Sham equations	22
3.3 Generation of the ionic potential	25
3.4 Summary	29
4 Numerical Methods	30
4.1 Discretization of the Kohn-Sham equations	30
4.1.1 Finite Difference Laplacian	31
4.1.2 Discretization of the ionic potential	38
4.1.3 Generation of the Hartree potential	39
4.1.4 Generation of the exchange-correlation potential	42
4.1.5 Hamiltonian Representation	43
4.2 Eigensolvers	46
4.2.1 Power Method	47
4.2.2 Inverse Iteration	49

4.2.3	Orthogonal Subspace Iteration	51
4.2.4	Chebyshev Filtered Orthogonal Iteration	53
4.2.5	Lanczos Algorithm	55
4.2.6	LOBPCG	58
4.3	Mixing techniques	62
4.4	Summary	64
5	Applications	66
5.1	MatRcal Input	66
5.2	High-order finite-differencing	69
5.3	HOMO-LUMO Gaps	72
5.4	Summary	78
6	Conclusion	80
	References	83

LIST OF FIGURES

4.1	Residual norm Δ against step size h : the test problem is the evaluation of the second derivative of e^x at $x = 0$	36
4.2	Residual norm Δ against step size h : the test problem is the evaluation of the Laplacian of $f(x, y, z) = \sin(x) + \sin(2y) + \sin(3z)$ on a Cartesian grid	37
4.3	Sparsity pattern of the Hamiltonian matrix of the unit cell of a Si(111)- 7×7 surface in real space. The matrix on the left shows the sparsity pattern of a $\mathcal{O}(h^{12})$ -Laplacian. The matrix itself represent the sum of the kinetic energy operator and all local potential energy operators. The matrices on the right represent the non-local pseudopotential energy operator in separable form (i.e. as a product of two low rank matrices). The number of non-zero entries of the Laplacian matrix scales linearly with the number of grid points. The number of non-zero entries in the non-local pseudopotential matrices is proportional to the number of atoms.	44
4.4	Sparsity pattern of the Hamiltonian matrix of the unit cell of a Si(111)- 7×7 surface in Fourier space. The number of non-zero entries of the Laplacian matrix scales linearly with the number of grid points. The number of non-zero entries in the non-local pseudopotential matrices is proportional to the number of atoms times the number of grid points.	45
4.5	Residual vector norm of the 5 lowest KS eigenstates of a molecule against the number of iterations during LOBPCG. The red curves represent the residual vector norm when LOBPCG is used without preconditioning. The blue curves represent the residual vector norm when LOBPCG is used with ILU preconditioning. . . .	61

5.1	Error on the total KS energy of a Benzene molecule with respect to the number of iterations for different Laplacians. Laplacian orders range from $\mathcal{O}(h^2)$ to $\mathcal{O}(h^{16})$	70
5.2	Error on the total KS energy of a SiH_4 molecule with respect to the number of iterations.	77

LIST OF TABLES

5.1	Comparison of “HOMO-LUMO” gaps (Hartree) by LDA using Gaussian (NIST CCCBDB database) and MatRcal. The right-most column shows the time it took MatRcal to reach convergence (i.e. $\delta V_{eff} < 10^{-4} V_{eff}$).	73
-----	--	----

Abstract

In condensed matter physics, atomistic first principle calculations are often necessary to achieve a microscopic understanding of the observed experimental phenomena and to make quantitative predictions of physical properties. In practice, atomic scale systems have irregularities (e.g. surface roughness) or defects (e.g. substitutional atoms or vacancies) that are too strong to be ignored or treated as small perturbations.

In this thesis, we report the development of a real space DFT code for studying atomic scale systems from first principles. Our code, named MatRcal, which stands for “Matlab-based real space calculator”, is developed in the technical computing language Matlab. The physics is described by density functional theory [1]. The method itself is based on projecting the Kohn-Sham Hamiltonian on a uniform Cartesian grid [2]. High-order finite-differencing is used to discretize the Laplacian operator [3]. The potential due to the atomic nuclei is approximated with *ab initio* pseudopotentials. The pseudopotentials are generated following the procedure proposed by Troullier and Martins [4]. We use the fully separable form introduced by Kleinman and Bylander [5]. We argue that the method is simpler and yet has many advantages compared with conventional spectral methods. We provide relevant mathematical techniques and implementation details. In particular, we present and compare different

eigensolvers used to diagonalize the Kohn-Sham Hamiltonian. We validate our software by comparing the HOMO-LUMO gaps of many organic and inorganic molecules obtained using our method with those obtained with the commercial code Gaussian. Our results are in excellent agreement. Our method gains in computational speed and algorithm parallelism, and its power in handling real space boundary conditions will be a major advantage for future applications in nanoelectronic device modelling.

Résumé

En physique de la matière condensée, les calculs numériques sont souvent nécessaires pour parvenir à comprendre les phénomènes microscopiques observés lors d’expériences ou à prédire quantitativement des propriétés physiques. En pratique, les systèmes d’échelle atomique sont irréguliers (rugosité de surface) ou comportent des défauts (atomes de substitution ou lacunes), ce qui induit des effets trop sévères pour être ignorés ou traités comme des perturbations.

Dans cette thèse, nous présentons une méthode qui permet d’étudier des systèmes d’échelle atomique à partir des lois fondamentales de la physique. Notre logiciel, nommé MatRcal, qui signifie “Matlab-based real space calculator”, est développé dans le langage Matlab. La physique est décrite par la théorie de la fonctionnelle de la densité [1]. La méthode projette l’Hamiltonien de Kohn-Sham sur un maillage Cartésien uniforme [2]. Le calcul des différences finies est utilisé pour discrétiser l’opérateur Laplacien [3]. Le potentiel dû aux noyaux atomiques est approximé par des pseudopotentiels non-empiriques. Les pseudopotentiels sont générés en suivant la procédure proposée par Troullier et Martins [4]. Nous utilisons la forme séparable introduite par Kleinman et Bylander [5]. Nous soutenons que la méthode est plus simple et pourtant présente de nombreux avantages par rapport aux conventionnelles méthodes spectrales. Nous introduisons plusieurs techniques mathématiques pertinentes à notre étude

et certains détails d'implémentation. Entre autres, nous présentons et comparons plusieurs algorithmes de calcul de vecteurs propres utilisés pour diagonaliser l'Hamiltonien de Kohn-Sham. Nous validons notre méthode en comparant la largeur de bande interdite "HOMO-LUMO" de nombreuses molécules organiques et inorganiques prédite par notre méthode avec celle prédite par le logiciel commercial Gaussian. Notre méthode permet des gains en rapidité et en parallélisme, mais la possibilité de traiter des conditions limites non-périodiques sera le principal atout pour de futures simulations de dispositifs nanoélectroniques.

Statement of Originality

Real space density functional theory calculators have been abandoned long ago. Spectral methods have dominated the field of *ab initio* structure calculations ever since. As the topic reaches its full maturity, both its strengths and limitations become more and more apparent. In particular, the limiting system size remains under a thousand atoms. The complexity of the algorithms is such that the problem cannot be palliate merely by hardware developments. It is now obvious that algorithms different in nature are needed to apply DFT to larger systems found in experiments.

Real space techniques have recently regained some popularity. Due to decades of hardware, software and mathematical developments, many of the reasons they were not used in the first place have become irrelevant or simply disappeared. It is now possible to find examples were real space methods already outperform traditional techniques in the literature [6]. In [7], Chelikowsky *et al.* present a method for solving the Kohn-Sham equations in real space using the calculus of finite differences.

We walked in their steps and can now report the implementation of a real space DFT solver. In [7], they report the implementation of Chebyshev filtered subspace iteration (CFSI) as their eigensolver. We implemented and tested many eigensolvers including CFSI and the more recent locally optimal block preconditioned conjugate gradient algorithm (LOBPCG) [8]. We found LOBPCG to be generally superior to CFSI. We note that it may be due to the fact that we simulated systems with a relatively small number of electrons and that for large systems CFSI may dominate. This is due to the fact that LOBPCG requires more subspace orthogonalizations than CFSI and its performance depends more on the efficiency of our orthogonalization algorithm. Our software allows to use both eigensolvers in a single calculation which is optimal in all cases.

Acknowledgments

I owe my most sincere thanks to my supervisor Prof. Hong Guo who proposed me an interesting research project for my Master's Thesis. He guided me, provided constant support and precious advice throughout the project. We have had the chance to discuss many related and unrelated topics. This helped me construct my understanding of physics and I look forward the continuation of our collaboration.

I would also like to thank Yibin Hu, Erik Zhu, Lei Liu and Dongping Liu for their help with the testing and validation of the software.

I acknowledge the financial support from the National Science and Engineering Research Council of Canada.

1

Introduction

Since the invention of the transistor in 1947, the field of electronics has progressed considerably, to a point where devices are now ubiquitous. For fifty years, the performance of electronic devices has been driven by the miniaturization of their basic components. We are currently at the twilight of miniaturization, where the dimensions of electronics devices are slightly bigger than the atomic scale [9, 10, 11]. At that scale, materials can no longer be thought of as continuous media. The presence of a single impurity atom can have dramatic effects [12]. The location of the impurity is important as well. Moreover, surface effects can seldom be ignored as is the case for large chunks of material[13]. The smallness of the atomic scale renders the traditional statistical and thermodynamics approximations inapplicable. Furthermore, at that scale, classical physics principles break down and a new kind of physics prevails. Quantum mechanical phenomena are no longer negligible and often dominate. This is clearly the case in effectively 1-dimensional or 2-dimensional systems which arise when the small size of a device along one direction freezes a degree of freedom [14, 15].

To understand the science of nanoelectronic devices, it is critical to use *ab initio* theories which can include the quantum effects and incorporate the discreteness of atomic scale systems. Otherwise, one must rely on empirical laws obtained by fitting the experimental data, a process which is becoming prohibitively expensive and unreliable at the nanoscopic scale.

The most powerful method for first principles modelling in material physics is density functional theory (DFT)[1, 2]. The success of DFT is evidenced by the 1998 Nobel Prize awarded to its discoverer, Prof. Walter Kohn. In DFT, the n-body problem of interacting electrons is reduced to a non-linear problem of non-interacting electrons. The central work is to solve the Kohn-Sham (KS) equations which is a self-consistent eigenvalue problem. The potential seen by electrons depends on the eigenstates of the KS Hamiltonian, and hence the eigenstates and potentials of the KS equations must be solved self-consistently.

Even though electronics are reaching the nanoscale, most devices consists of too many atoms to be simulated using DFT. It is crucial to bridge the gap between theory and experiment. A lot of efforts are consecrated to the search of powerful computational techniques which can analyse real-world applications with DFT [6, 16, 17, 18, 19, 20, 21, 22]. In a typical DFT method, one expands the KS Hamiltonian in terms of a basis set such as plane waves [23] or atomic centered orbitals [24, 25], and the KS equation becomes a matrix equation in the space spanned by the basis functions. These spectral techniques have been and still are quite successful, yet the characteristic problem size remains limited

to a few hundred atoms. In the last decade, methods that can simulate larger systems using a real space approach have been developed [26, 27, 28, 29, 30, 31]. Evidence of good performance - by Chebyshev filtered subspace iteration (CFSI) - was reported by Chelikowsky *et al* in [7]. We found that the performance of the locally optimal block preconditioned conjugate gradient method (LOGPCG) introduced by Knyazev [8] generally exceeds that of CFSI for solving the KS equations. The two algorithms can also be used concurrently, each prevailing in its own domain. We developed software that can carry DFT calculations in real space based on their work.

Chapter 2 contains a review of density functional theory. The proofs of the fundamental theorems of DFT are presented. We introduce the Kohn-Sham equations. We also discuss the local density approximation of the exchange and correlation energy.

In chapter 3, we compare real space approaches and spectral methods for KS-DFT calculations. We formulate the Kohn-Sham eigenvalue problem in real space. We also introduce the pseudopotential approximation and illustrate the generation procedure.

In chapter 4, we expose many mathematical techniques and implementation details necessary to solve the Kohn-Sham equations efficiently. We discuss the discretization of the Kohn-Sham equations, the resolution of the Poisson equation and the diagonalization of the Kohn-Sham Hamiltonian. We also cite many mixing techniques for accelerating the convergence of the self-consistent

field iterations.

In chapter 5, a few applications are presented. We test our software by comparing the predicted HOMO-LUMO gaps against those obtained with Gaussian.

Finally, chapter 6 summarizes the thesis and presents future developments.

2.1 Density Functional Theory

The quantum many-body problem can be theoretically formulated in terms of wavefunctions. The information contained in a many-body wavefunction grows exponentially with the number of particles. This scaling makes the many-body wavefunction based formalism impractical for numerical simulations of real material systems because of the prohibitive memory and computation requirements. For practical materials physics applications, one is usually interested in a very limited subset of information contained in the many-body wavefunction, hence exactly computing the entire many-body wavefunction is rarely necessary. In density functional theory (DFT), the many-body problem is reformulated in terms of a much smaller mathematical object: the electronic density. The density is merely a function of space and hence much less information is needed to encode it. This is perfect for numerical simulations since the memory requirements scale linearly with the number of particles. We will provide two set of

proofs which are the cornerstones of DFT.

We first introduce the fundamental theorems of DFT that were proven by Honenberg and Kohn in 1964 [1]. Consider the following Hamiltonian

$$H = T + V + U \quad (2.1)$$

$$T = \frac{1}{2} \int d\mathbf{r} |\nabla \psi(\mathbf{r})|^2 \quad (2.2)$$

$$V = \int d\mathbf{r} \psi^*(\mathbf{r}) V(\mathbf{r}) \psi(\mathbf{r}) \quad (2.3)$$

$$U = \int d\mathbf{r} d\mathbf{r}' \psi^*(\mathbf{r}) \psi^*(\mathbf{r}') \frac{1}{|\mathbf{r} - \mathbf{r}'|} \psi(\mathbf{r}) \psi(\mathbf{r}') \quad (2.4)$$

where ψ is a field operator, T is the kinetic energy, V is some external potential and U is the coulomb interaction between the electrons. Clearly, the electronic density ρ is a functional of the ground-state wavefunction Ψ since it can be written as $\rho(\mathbf{r}) = \langle \Psi | \psi^*(\mathbf{r}) \psi(\mathbf{r}) | \Psi \rangle$. But the converse is also true. In order to prove that, Honenberg and Kohn showed that V is a functional of ρ . The proof is the following. Suppose that there exists a second ground state Ψ' associated with an external potential V' and that give rise to an electronic density $\rho(\mathbf{r})$, then

$$E' = \langle \Psi' | H' | \Psi' \rangle < \langle \Psi | H' | \Psi \rangle = \langle \Psi | H - V + V' | \Psi \rangle \quad (2.5)$$

$$E' < E + \langle \Psi | V' - V | \Psi \rangle \quad (2.6)$$

$$E' < E + \int d\mathbf{r} \rho(\mathbf{r}) (V'(\mathbf{r}) - V(\mathbf{r})) \quad (2.7)$$

By symmetry we have

$$E < E' + \int d\mathbf{r} \rho(\mathbf{r})(V(\mathbf{r}) - V'(\mathbf{r})) \quad (2.8)$$

The inequalities 2.7 and 2.8 are summed to provide the following inconsistency

$$E + E' < E' + E \quad (2.9)$$

We conclude that V is uniquely determined by ρ and hence that it is a functional of ρ . As the kinetic and Coulomb terms are also functionals of the density, the Hamiltonian is a functional of the density. We conclude that the ground-state wavefunction can be viewed a functional of the density as it can be computed from the Hamiltonian. It follows that all the ground-state properties can be obtained from the density. It is unfortunate that the proof is by contradiction and does not constitute a tool for practical calculations.

Hohenberg and Kohn also demonstrated (again by contradiction) the existence of a universal energy functional of the density. The total energy functional can be written as

$$E[\rho] = \int d\mathbf{r} V(\mathbf{r})\rho(\mathbf{r}) + F[\rho] \quad (2.10)$$

The functional $F[\rho]$ is not dependent on the external potential and it is valid for any number of particles. Note that no approximation has been made so far

and that makes DFT an exact theory. It would be a tremendously powerful one if $F[\rho]$ were known but it is not the case. Much efforts have been dedicated to the development of approximate functionals, some of which will be presented below.

In the last theorem, Hohenberg and Kohn assume that the density yields the correct ground-state among those wavefunctions that are the ground-state of some external potential V . We call such functions V-representable functions. It is not known whether a given wavefunction is the ground-state for some external potential and that is why we present another insightful derivation by Levy [32].

As above, let

$$H = T + V + U \quad (2.11)$$

where T is the kinetic energy, V is some external potential and U is the coulomb interaction. Define a universal functional of the electronic density as

$$F[\rho] = \min (\langle \Psi_\rho | T + U | \Psi_\rho \rangle) \quad (2.12)$$

where the minimum is taken over all N-representable wavefunctions $|\Psi_\rho\rangle$ yielding a density ρ . A N-representable wavefunction is simply an antisymmetric wavefunction of N particles.

We now show the fundamental theorems of DFT. Let us define $|\Psi_\rho^{\min}\rangle$ as a wavefunction that satisfies equation 2.12.

We show that

$$\int d\mathbf{r} V(\mathbf{r})\rho(\mathbf{r}) + F[\rho] \geq E_{gnd} \quad (2.13)$$

The proof goes as follows

$$\int d\mathbf{r} V(\mathbf{r})\rho(\mathbf{r}) + F[\rho] \quad (2.14)$$

$$= \int d\mathbf{r} V(\mathbf{r})\rho(\mathbf{r}) + \min (\langle \Psi_\rho | T + U | \Psi_\rho \rangle) \quad (2.15)$$

$$= \int d\mathbf{r} V(\mathbf{r})\rho(\mathbf{r}) + \langle \Psi_\rho^{\min} | T + U | \Psi_\rho^{\min} \rangle \quad (2.16)$$

$$= \langle \Psi_\rho^{\min} | T + V + U | \Psi_\rho^{\min} \rangle \geq E_{gnd} \quad (2.17)$$

by definition of E_{gnd} .

We then show that

$$\int d\mathbf{r} V(\mathbf{r})\rho_{gnd}(\mathbf{r}) + F[\rho_{gnd}] = E_{gnd} \quad (2.18)$$

The proof goes as follow

$$E_{gnd} \leq \langle \Psi_{\rho_{gnd}}^{\min} | T + V + U | \Psi_{\rho_{gnd}}^{\min} \rangle \quad (2.19)$$

$$\langle \Psi_{gnd} | T + V + U | \Psi_{gnd} \rangle \leq \langle \Psi_{\rho_{gnd}}^{\min} | T + V + U | \Psi_{\rho_{gnd}}^{\min} \rangle \quad (2.20)$$

$$\langle \Psi_{gnd} | T + V + U | \Psi_{gnd} \rangle \leq \langle \Psi_{\rho_{gnd}}^{\min} | T + V + U | \Psi_{\rho_{gnd}}^{\min} \rangle \quad (2.21)$$

$$\int d\mathbf{r} V(\mathbf{r}) \rho_{gnd}(\mathbf{r}) + \langle \Psi_{gnd} | T + U | \Psi_{gnd} \rangle \leq \int d\mathbf{r} V(\mathbf{r}) \rho_{gnd}(\mathbf{r}) + \langle \Psi_{\rho_{gnd}}^{\min} | T + U | \Psi_{\rho_{gnd}}^{\min} \rangle \quad (2.22)$$

$$\langle \Psi_{gnd} | T + U | \Psi_{gnd} \rangle \leq \langle \Psi_{\rho_{gnd}}^{\min} | T + U | \Psi_{\rho_{gnd}}^{\min} \rangle \quad (2.23)$$

By the definition of $|\Psi_{\rho_{gnd}}^{\min}\rangle$

$$\langle \Psi_{gnd} | T + U | \Psi_{gnd} \rangle \geq \langle \Psi_{\rho_{gnd}}^{\min} | T + U | \Psi_{\rho_{gnd}}^{\min} \rangle \quad (2.24)$$

So

$$\int d\mathbf{r} V(\mathbf{r}) \rho_{gnd}(\mathbf{r}) + F[\rho_{gnd}] = E_{gnd} \quad (2.25)$$

Note that this proof is not only valid for non-degenerate ground-states, an assumption that Honenberg and Kohn made in their original article. In practice one accounts for degeneracies by finding the wavefunctions one at a time, taking care to impose orthogonality with the previously found wavefunctions. It is tempting to simply declare the latter proof as superior to the original. In the present situation, the density ρ is obviously a good basic variable. It is the

starting point of the Levy's proof. For more complicated systems, such as magnetic systems, it is not so clear what the basic variables are and they must be determined by a HK type proof before a Levy's type proof can be undertaken [33].

2.2 Kohn-Sham Self-Consistent Field Equations

We mentioned that the proofs of the fundamental theorems of DFT are not constructive and hence they do not provide a method for solving practical problems. Kohn and Sham made interesting observations that offer us a way to use DFT for electronic structure calculations [2].

The energy of a system can be written as

$$E[\rho] = \int d\mathbf{r} V(\mathbf{r})\rho(\mathbf{r}) + F[\rho] \quad (2.26)$$

$$= \int d\mathbf{r} V(\mathbf{r})\rho(\mathbf{r}) + T_0[\rho] + \frac{1}{2} \int d\mathbf{r} d\mathbf{r}' \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + E_{xc}[\rho] \quad (2.27)$$

where $T_0[\rho]$ is the kinetic energy of non-interacting particles and $\frac{1}{2} \int d\mathbf{r} d\mathbf{r}' \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}$ is the classical Coulomb energy term. The exchange-correlation energy $E_{xc}[\rho]$ accounts for all the quantum many-body effects. Let the ground-state density be ρ_0 , therefore we can write $\rho = \rho_0 + \delta\rho$. Let us look at the variation in the

energy δE due to a variation $\delta\rho$.

$$\begin{aligned}
E[\rho] - E[\rho_0] &= \int d\mathbf{r} V(\mathbf{r}) (\rho(\mathbf{r}) - \rho_0(\mathbf{r})) + T_0[\rho] - T_0[\rho_0] \\
&\quad + \frac{1}{2} \int d\mathbf{r} d\mathbf{r}' \frac{(\rho_0(\mathbf{r}) + \delta\rho(\mathbf{r})) (\rho_0(\mathbf{r}') + \delta\rho(\mathbf{r}')) - \rho_0(\mathbf{r})\rho_0(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \\
&\quad + E_{xc}[\rho] - E_{xc}[\rho_0] \\
&= \int d\mathbf{r} V(\mathbf{r}) \delta\rho(\mathbf{r}) + T_0[\rho] - T_0[\rho_0] \\
&\quad + \int d\mathbf{r} d\mathbf{r}' \frac{\rho_0(\mathbf{r}') \delta\rho(\mathbf{r}) + \mathcal{O}(\delta\rho^2)}{|\mathbf{r} - \mathbf{r}'|} + E_{xc}[\rho] - E_{xc}[\rho_0]
\end{aligned} \tag{2.28}$$

We now divide through by $\delta\rho$

$$\frac{E[\rho] - E[\rho_0]}{\delta\rho} = \int d\mathbf{r} V(\mathbf{r}) + \frac{T_0[\rho] - T_0[\rho_0]}{\delta\rho} + \int d\mathbf{r} d\mathbf{r}' \frac{\rho_0(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + \frac{E_{xc}[\rho] - E_{xc}[\rho_0]}{\delta\rho} \tag{2.29}$$

$$\left. \frac{\delta E[\rho]}{\delta\rho} \right|_{\rho=\rho_0} = \int d\mathbf{r} \left(V(\mathbf{r}) + \frac{\delta t_0[\rho]}{\delta\rho} + \int d\mathbf{r}' \frac{\rho_0(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + \left. \frac{\delta \epsilon_{xc}[\rho]}{\delta\rho} \right|_{\rho=\rho_0} \right) \tag{2.30}$$

where t_0 is the kinetic energy per volume and ϵ_{xc} is the exchange correlation energy per volume. If ρ_0 minimizes the total energy then it must satisfy

$$\left. \frac{\delta E[\rho]}{\delta\rho} \right|_{\rho=\rho_0} = 0 \tag{2.31}$$

and hence

$$0 = \int d\mathbf{r} \left(V(\mathbf{r}) + \frac{\delta t_0[\rho]}{\delta\rho} + \int d\mathbf{r}' \frac{\rho_0(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + \left. \frac{\delta \epsilon_{xc}[\rho]}{\delta\rho} \right|_{\rho=\rho_0} \right) \tag{2.32}$$

Since $\delta\rho$ is arbitrary, the integrand must be zero and we get that

$$\frac{\delta t_0[\rho]}{\delta\rho} + V(\mathbf{r}) + \int d\mathbf{r}' \frac{\rho_0(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + \left. \frac{\delta\epsilon_{xc}[\rho]}{\delta\rho} \right|_{\rho=\rho_0} = 0 \quad (2.33)$$

Let

$$V_{eff} = \int d\mathbf{r}' \frac{\rho_0(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + \left. \frac{\delta\epsilon_{xc}[\rho]}{\delta\rho} \right|_{\rho=\rho_0} = V_H[\rho] + V_{xc}[\rho] \quad (2.34)$$

then

$$\frac{\delta t_0[\rho]}{\delta\rho} + V(\mathbf{r}) + V_{eff}(\mathbf{r}) = 0 \quad (2.35)$$

Equation 2.35 has the same form as the one obtained when applying DFT to a non-interacting electrons in a potential $V(\mathbf{r}) + V_{eff}(\mathbf{r})$. The electronic ground-state density is obtained by solving the single particle Schrödinger's equation

$$\left(-\frac{1}{2}\nabla^2 + V(\mathbf{r}) + V_{eff}(\mathbf{r}) \right) \psi_i = \epsilon_i \psi_i \quad (2.36)$$

self-consistently with

$$\rho(\mathbf{r}) = \sum_{i=1}^N \psi_i^*(\mathbf{r}) \psi_i(\mathbf{r}) \quad (2.37)$$

where N is the number of electrons. The Kohn-Sham wavefunctions and energies should not be thought of as physical quantities in general. The Kohn-Sham

equations are a mathematical reformulation of the initial problem which is to find the electronic density and not individual electronic states.

We reiterate the original problem which is to compute the electronic ground-state of atomic scale systems. Solving the (linear) Schrödinger equation for the many-body wavefunction is a formidable task. The problem can be reduced to non-interacting electrons moving in an effective potential using DFT. Such a simplification is accompanied with the issues associated with non-linear equations. The unknown universal energy functional of the DFT theorems translates into an unknown exchange-correlation potential in the Kohn-Sham equations. We will present a way to approximate ϵ_{xc} (and hence V_{xc}) below.

2.3 *Exchange-Correlation Energy Functionals*

The accuracy of the Kohn-Sham method depends greatly on the accuracy of the exchange-correlation functional. We present the local density approximation (LDA) which is the simplest and most widely used functional in DFT. The approximation is the following : the exchange-correlation energy at one point is approximated by the exchange-correlation energy of a uniform electron gas with the same density as the density at that point. For clarity, the last statement is rewritten as equation 2.38 where $\epsilon_{xc}(\rho)$ is the exchange correlation energy per particle for a uniform electron gas of density ρ .

$$E_{xc}[\rho] = \int d\mathbf{r} \rho(\mathbf{r}) \epsilon_{xc}(\rho(\mathbf{r})) \quad (2.38)$$

In general, the approximation is good for systems that have a slowly varying density. According to equation 2.34 we have

$$V_{xc} = \frac{\delta\rho(\mathbf{r})\epsilon_{xc}(\rho)}{\delta\rho(\mathbf{r})} \quad (2.39)$$

$$= \epsilon_{xc}(\rho) + \rho(\mathbf{r}) \frac{\delta\epsilon_{xc}(\rho)}{\delta\rho(\mathbf{r})} \quad (2.40)$$

The exchange energy of a uniform electron gas is known exactly. We thus split the known and unknown part of $\epsilon_{xc}(\rho)$ (and V_{xc} similarly) as

$$\epsilon_{xc}(\rho) = \epsilon_x(\rho) + \epsilon_c(\rho) \quad (2.41)$$

The exchange energy of a uniform electron gas is [34]

$$\epsilon_x[\rho] = -\frac{3}{4} \left(\frac{3}{\pi} \right)^{\frac{1}{3}} \rho^{\frac{4}{3}} \quad (2.42)$$

which gives an exchange potential

$$V_x = - \left(\frac{3\rho}{\pi} \right)^{\frac{1}{3}} \quad (2.43)$$

No analytical expression for the correlation energy of a uniform electron gas is available. People have found expressions for the low and high density asymptotic behaviours [35]. Subsequently, Ceperley and Alder calculated the correlation energy of a uniform electron gas using stochastic methods [36]. Many have proposed analytic expressions that give the correct limiting behaviours to

interpolate the data [37, 38, 39, 40]. These expressions are not really physically motivated but they grant us convenient forms to work with.

In principle, a universal density functional would use the information of the whole density function to evaluate the exchange-correlation energy at one point. At the moment, we have no means of designing such a functional. As previously pointed out, the exchange-correlation energy can be approximately known using a single value of the density. In order to go beyond LDA, it is natural to consider the density in a neighbourhood of the point at which the exchange-correlation energy is to be evaluated. The density is a multivariate analytic function and its behaviour in a neighbourhood of some point is determined by its derivatives at that point consequently. People have designed functionals which evaluate the exchange-correlation energy using the density, its gradient and higher order derivatives [41, 42, 43, 44]. Such functionals are said to be derived within the generalized gradient approximation (GGA).

2.4 Summary

In this section, we presented the DFT theoretical formalism which is the key to *ab initio* atomic scale computations. We reviewed the proofs of Hohenberg-Kohn and Levy of the fundamental theorems of DFT and mentioned the complementarity nature of their proofs. We derived the Kohn-Sham equations using the variational principle. We showed that minimizing that energy with respect to the electronic density was equivalent to solving a Schrödinger equation where all

the non classical effects are incorporated in an effective potential which is itself a functional of the density. This results in a set of equations which are solved self-consistently. We discussed the local density approximation and mentioned how to improve on it using the derivatives of the density. We concluded that the Kohn-Sham equations can be used to solve quantum many-body problems approximately. The accuracy of the results depends on the exchange-correlation functional.

3

Real Space Pseudopotential Method

Our principal objective is to compute the electronic ground-state of atomic scale systems. The problem is solved using the self-consistent field theoretic approach in the Kohn-Sham formulation introduced above. A number of techniques for solving the Kohn-Sham equations exist. In the mid 90's, Chelikowsky et al. published a series of papers arguing favourably about real space methods [29, 30, 31]. They use high-order finite-differences to discretize the equations. In the last decade, real space DFT has become more popular and more involved techniques such as finite elements have emerged [45, 6]. In this thesis, we will refer to finite-difference delta function sets as real space bases.

3.1 Real space and conventional methods

In a typical DFT method, one expands the wavefunctions in terms of a basis set and the KS equation becomes a matrix equation in the space spanned by

the basis functions. Formally, it is written as equation 3.2.

$$H|\Psi\rangle = E|\Psi\rangle \quad (3.1)$$

$$\left(\sum_{\alpha,\beta} |\phi_\alpha\rangle H_{\alpha,\beta} \langle\phi_\beta| \right) |\Psi\rangle = E \left(\sum_{\alpha,\beta} |\phi_\alpha\rangle S_{\alpha,\beta} \langle\phi_\beta| \right) |\Psi\rangle \quad (3.2)$$

where $|\phi_\alpha\rangle$ are planewaves, orthogonalized planewaves, atomic orbitals, gaussians, etc. Each basis set has its own strengths and handicaps. Planewaves form a complete basis that allows for arbitrary accuracy but a myriad must be used to describe the oscillatory (sometimes even kinky) behaviour of the wavefunctions near the atomic nuclei. Orthogonalized planewaves were constructed to deal with this issue at the cost of some complexity and accuracy. The atomic orbitals provide a way of discretizing the Schrödinger equation in a very dense form but the accuracy is hardly tunable. Gaussians are convenient to work with since they are localized both in real space and momentum space but they form a non-orthonormal basis. These methods have been used successfully to calculate the electronic structure of many condensed matter systems. Nevertheless, there is a need for techniques that can solve large systems both efficiently and accurately.

Real space bases have long been ignored because they are among the least efficient ways of discretizing a Hamiltonian, by which we mean that a huge number of basis functions are necessary. It is unfortunate as they enjoy many advantages compared to other representations. The precision can be adjusted

simply by refining the grid. The exchange-correlation potential is evaluated in real space so that going back and forth to some other representation is not required contrary to conventional bases. Evaluating the exchange-correlation potential is expeditious for planewaves because of the fast Fourier transform algorithms even though their lack of parallelizability is inconvenient for large systems. It is more problematic for other bases sets such as atomic orbitals for which the transforms can become more expensive than solving the Schrödinger equation. Similarly, the Hartree potential is also frequently computed in real space leading to the same limitations for conventional bases. It is much simpler to treat non-periodic boundary conditions such as Dirichlet in real space although periodic boundary conditions are easily handled as well. No artificial structures such as supercells have to be used in order to treat non-periodic systems. Real space bases are inherently local and lead to embarrassingly parallelizable algorithms which is not a minor advantage for solving large atomic scale systems. We point out the potential gains accomplished using a real space basis in solving the Schrödinger equation below.

The most intensive part in electronic structure calculations is usually solving the Schrödinger equation which is an eigenvalue problem. Contrary to most basis sets, real space bases are orthonormal so that the generalized eigenvalue problem is avoided. Numerical eigenvalue problems are solved iteratively. The most widely used method is the QR-algorithm which is arguably one of the most important algorithms of the 20th century [46]. The QR-algorithm is tailored to

find the whole eigenspectrum of an operator. Our problem is one of minimization of the energy and hence only the few smallest eigenpairs are needed. There are diagonalization procedures that can compute the eigenpairs at the lower end of the spectrum only. The complexity of these algorithms is proportional to the number of non-zero entries of the Hamiltonian matrix without regard to its size. A large number of grid points are necessary in order to get a description of an electronic structure problem that is more accurate than, say, linear combinations of atomic orbitals (LCAO). In fact, the matrices resulting from a real space discretization cannot be handled by the QR-algorithm because the eigenvectors are too numerous. For most systems, real space pseudopotential methods yield a very sparse Hamiltonian whereas bases such as LCAO yield a dense Hamiltonian matrix. As a result, even though the real space representation of the Hamiltonian is much larger than the LCAO representation, it does not necessarily contain more non-zeros entries whence the Schrödinger equation can be solved efficiently.

We do not want to leave the impression that real space methods necessarily outperform other existing methods. Small basis sets such as LCAO or gaussians can describe systems remarkably efficiently. They yield small systems of equations which are relatively easily dealt with. However, they are incomplete bases and some experience is required in order to reach high accuracy. Planewaves compete more directly with real space methods as they are a complete basis. Certain problems, typically relatively small problems with smooth potentials

and low electronic density, can be solved advantageously by planewaves codes. However, the highest energy states generally show a very oscillatory behaviour in order to stay orthogonal to the lowest energy states and then a large number of planewaves is required. This leads to formidably large eigenvalue problems, usually smaller but similar in size to the problems arising in real space. Real space methods on the other hand can handle complicated geometries rather straightforwardly. More importantly, the asymptotic cost of iterative diagonalization procedures is sometimes smaller in real space, which we shall illustrate in section 4.1.5. We present a real space approach to the problem via the finite-difference method below. While this method is much simpler to implement, it is easily parallelizable, and hence is a good candidate toward modelling large condensed matter systems.

3.2 *Real space pseudopotential formulation of the Kohn-Sham equations*

It is natural to express equations 2.36 and 2.37 in real space since it is often the way we think of them in the back of our minds. We use the Born-Oppenheimer approximation which says that the electronic and nuclear degrees of freedom can be uncoupled.

$$\Psi = \psi_{\text{electronic}} \otimes \psi_{\text{nuclear}} \tag{3.3}$$

Heuristically, we can consider the nuclei as non-dynamical as they are so much heavier than the electrons. This way, we can ignore the nuclear degrees of freedom and hence considerably reduce the problem we try to solve.

The external potential $V(\mathbf{r})$ is the potential generated by the atoms' nuclei. Each point of a real space grid corresponds to a position vector $\mathbf{r}$ so that there is nothing else to be done than to evaluate $V(\mathbf{r})$ at all those points. We now consider the effective potential $V_{eff}(\mathbf{r})$. One contribution is the Hartree potential defined as

$$V_H(\mathbf{r}) = \int d\mathbf{r}' \frac{\rho_0(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \quad (3.4)$$

Notice that $\frac{1}{|\mathbf{r} - \mathbf{r}'|}$ is in fact the Green function of the three-dimensional Poisson equation. We use this to express the integral equation 3.4 in differential form as follows

$$\nabla^2 V_H(r) = -4\pi\rho(r) \quad (3.5)$$

where the 4π appears because we are using natural units. After each update of the density 2.37, the Hartree potential is updated accordingly using equation 3.5. The remaining contribution to $V_{eff}(\mathbf{r})$ is the exchange-correlation potential $V_{xc}(\mathbf{r})$. As discussed above, $V_{xc}(\mathbf{r})$ is evaluated using some functional of the density. If we use LDA, $V_{xc}(\mathbf{r}) = f(\rho(\mathbf{r}))$ where f is simply a real function of one variable. This is done after each update of the density just like the Hartree

potential.

We summarize the problem as follows [47]:

1. Generate a discrete operator $-\frac{1}{2}\nabla^2$ for the kinetic energy. Generate an initial guess of the charge density ρ_0 and compute

$$V_{total}[\rho_0] = V_{ion} + V_H[\rho_0] + V_{XC}[\rho_0] \quad (3.6)$$

where V_{ion} is the ionic (external) potential, $V_H[\rho]$ is the Hartree potential and $V_{XC}[\rho]$ is the exchange-correlation potential.

2. Solve Schrödinger equation

$$\left[-\frac{1}{2}\nabla^2 + V_{total}(r) \right] \Psi_i(r) = E_i \Psi_i(r) \quad (3.7)$$

for $\Psi_i(r)$, $i = 1, 2, \dots, s$, where s is the number of occupied states.

3. Compute the new charge density

$$\rho_{n+1}(r) = \sum_{i=1}^s |\Psi_i(r)|^2 \quad (3.8)$$

4. Update the Hartree potential by solving the Poisson equation

$$\nabla^2 V_H(r) = -4\pi\rho_{n+1}(r) \quad (3.9)$$

5. Update V_{XC} and compute

$$\tilde{V}_{total}[\rho_{n+1}] = V_{ion} + V_H[\rho_{n+1}] + V_{XC}[\rho_{n+1}] \quad (3.10)$$

6. If $\|\tilde{V}_{total} - V_{total}\| < \epsilon$, where ϵ is some prescribed tolerance, then stop.

Else go to step 2.

Most of the discussion on the procedure we just described will be done in chapter 4. Before that, we conclude the present chapter by presenting the broad lines of the real space pseudopotential generation process.

3.3 *Generation of the ionic potential*

One contribution to the total potential is the ionic potential coming from the atomic nuclei. It is possible to use the Coulomb potential but this approach is far less effective than pseudopotential methods. In most cases, the core electrons are so tightly bound to the nucleus of the atoms that they play a negligible role in chemical and physical phenomena. We can thus consider them as frozen degrees of freedom (similar to the Born-Oppenheimer approximation) and only worry about the valence electrons. The core electrons effectively screen the singular potential due to the nucleus and the valence electrons perceive a smoother potential as a consequence. Various methods have been developed to generate these effective (approximate) potentials. We have misleadingly written $V_{ion}(r)$ thus far. We should have written $V_{ion}(r, r') = V_{local}(r) + V_{nl}(r, r')$ instead since

pseudopotentials are most accurate in non-local form. We generate them following the prescriptions by Troullier-Martins and Kleinman-Bylander [4, 5].

We present the pseudopotential generation procedure in broad strokes. A system consisting of a single atom and all its electrons is analysed using density functional theory. It is important to generate the pseudopotentials using the same exchange-correlation functional $V_{xc}[\rho]$ that will be used during further computations. Once the potential (or density) has converged, the Schrödinger equation is solved one last time in the radial direction to obtain the eigenstates $R_{nl}^{AE}(r)$, where the superscript AE indicates all-electron wavefunction.

$$\left(-\frac{1}{2r} \frac{d^2}{dr^2} r + \frac{l(l+1)}{2r^2} + V_l(r) + V_H(r) + V_{xc}(r) \right) R_{nl}^{AE}(r) = \epsilon_{nl} R_{nl}^{AE}(r) \quad (3.11)$$

We then proceed with the generation of the pseudoeigenstates. We impose the following form proposed by Kerker [48] for the pseudowavefunctions of interest (corresponding to the valence electrons, i.e. those with maximal n)

$$R_l^{PP}(r) = \begin{cases} R_l^{AE}(r) & r \geq r_c; \\ r^l e^{p(r)} & r \leq r_c \end{cases} \quad (3.12)$$

where the superscript PP indicates pseudopotential wavefunction, $R_l^{AE}(r)$ is an eigenstate with angular momentum l , $p(r)$ is a polynomial and r_c is the core radius (a parameter set by the user). $p(r)$ can be chosen in various ways and

the form proposed by Troullier and Martins [4]

$$p(r) = c_0 + c_2 r^2 + c_4 r^4 + c_6 r^6 + c_8 r^8 + c_{10} r^{10} + c_{12} r^{12} \quad (3.13)$$

is most popular for DFT computations.

The Schrödinger equation is then inverted to produce $V_l^{scr}(r)$ in terms of $R_l^{PP}(r)$ or $p(r)$. The coefficients are determined from the following seven conditions [4]:

1. The charge inside the core (within a radius r_c) is equivalent for R_l^{PP} and R_l^{AE} . Pseudopotentials satisfying this condition are said to be norm-conserving.
2. R_l^{PP} is $\mathcal{C}^4$ which yields 5 constraints since it must be continuous and 4 times differentiable at $r = r_c$.
3. The curvature of the screened pseudopotential vanishes at the origin.

Next follows the unscreening step which consists of removing the Hartree and exchange-correlation contributions to the pseudopotentials

$$V_l(r) = V_l^{scr}(r) - V_H(r) - V_{xc}(r) \quad (3.14)$$

The non-local form of the pseudopotential is then obtained for each angular component in two steps. First, the local part $V_{local}(r)$, common to all angular momenta, is generated. This step is to some extent arbitrary because the

unaccounted for effects are absorbed in the non-local part. There are however studies that suggest certain forms may be better than others. Then the non-local part is generated following Kleinman and Bylander [5]. Let

$$\delta V_l(r) = V_l(r) - V_{local}(r) \quad (3.15)$$

Then the Kleinman-Bylander projection functions are defined as

$$\chi_{lm}^{KB}(r) = \delta V_l(r) R_l^{PP}(r) Y_{lm}(\hat{r}) \quad (3.16)$$

and the non-local component of the pseudopotentials is defined as

$$V_{nl}(r, r') = \sum_{l=0}^{l_{max}} \sum_{m=-l}^l | \chi_{lm}^{KB} \rangle v_l \langle \chi_{lm}^{KB} | \quad (3.17)$$

where

$$v_l = \langle \chi_{lm}^{KB} | \delta V_l | \chi_{lm}^{KB} \rangle \quad (3.18)$$

It is preferable not to have to generate these pseudopotentials each time we start a calculation. The ability of a pseudopotential to be valid in various environments is called transferability. In principle, it is optimal to design a pseudopotential specific to each problem. But it requires extensive knowledge and good intuition, or it can be time consuming so that most people use transferable pseudopotentials. Transferability is mostly determined by the core radius

r_c which needs to be large enough. A larger r_c also yields a smoother pseudopotential which is to some extent an advantage in real space since a coarser grid can be used. This competes against the fact that we would like the pseudopotentials to be as short ranged as possible in order to limit our simulation box size.

3.4 *Summary*

In this section, we described the class of problems we are mostly interested in, namely the computation of the electronic ground state of atomic scale systems. We reviewed the spectral formulation of the Kohn-Sham equations and pointed out its limitations. Notably, it is a field that has reached maturity and the typical system size limit remains under a thousand atoms. We compared the advantages and disadvantages of spectral and real space methods. In real space, the major bottleneck is the computation of the eigenstates of the Kohn-Sham equations. The large number of basis functions necessitated by real space methods is alleviated by the sparsity of the Hamiltonian matrix. When only a subspace of the Hamiltonian needs to be diagonalized, the algorithmic complexity depends on the sparsity rather than the size. We provided a simple procedure for solving the Kohn-Sham equations. Finally, we outlined the real space pseudopotential generation procedure.

Numerical Methods

4.1 *Discretization of the Kohn-Sham equations*

In chapter 3 we presented an algorithm for solving the Kohn-Sham equations. We will discuss numerical and implementation details concerning each step below.

It is important to establish the boundary conditions for the partial differential equations. One of the purpose of MatRcal developed in this thesis is to study crystals in which Bloch's theorem guarantees that the eigenstates satisfy the following condition [49]

$$\psi(\mathbf{r}) = e^{-i\mathbf{k}\cdot\mathbf{R}}\psi(\mathbf{r} + \mathbf{R}) \quad (4.1)$$

where $\mathbf{R}$ is a lattice vector. Kohn-Sham equations are thus solved in a single unit cell with periodic boundary conditions since the wavefunction in one unit cell is related to the wavefunction in another unit cell by a phase shift. Periodicity also allows to study non-periodic atomic arrangements such as molecules. In this

case, the atoms are put at the center of a supercell which is large enough that molecules interact negligibly with their images. Of course, a supercell should be chosen as small as possible in order to reduce computational cost.

One of the advantages of real space bases is the ability to avoid using supercells. A lot of unnecessary grid points have to be introduced indeed. It may seem contradictory to use periodic boundary conditions in this case. Since we are still at a preliminary stage of development, we do not need a maximally efficient implementation and for our purposes periodic boundary conditions are appropriate. Furthermore, only minor changes are required in order to implement non-periodic boundary conditions.

4.1.1 *Finite Difference Laplacian*

Expressing the kinetic energy term $-\frac{1}{2}\nabla^2$ is not straightforward. The real space representation of the Laplacian is not diagonal contrary to planewaves representation.

Finite differencing is used to gain information about the derivatives of a function using only values of the function itself. The idea is to use the Taylor expansion of u about $\mathbf{x}_i$ to approximate u at neighbouring points. A weighted sum of neighbouring terms is then performed so as to yield an expression for $\frac{\partial u}{\partial x}(\mathbf{x}_i)$. For example, consider u on a one dimensional grid with a uniform mesh

of size h . Then

$$\begin{aligned}
 u(x_{i-1}) &= u(x_i) + \frac{\partial u}{\partial x}(x_i)(x_{i-1} - x_i) + \dots \\
 u(x_{i-1}) &= u(x_i) - h \frac{\partial u}{\partial x}(x_i) + \mathcal{O}(h^2) \\
 u(x_{i+1}) &= u(x_i) + \frac{\partial u}{\partial x}(x_i)(x_{i+1} - x_i) + \dots \\
 u(x_{i+1}) &= u(x_i) + h \frac{\partial u}{\partial x}(x_i) + \mathcal{O}(h^2)
 \end{aligned} \tag{4.2}$$

We observe

$$\frac{u(x_{i+1}) - u(x_{i-1}))}{2h} + \mathcal{O}(h^2) = \frac{\partial u}{\partial x}(x_i) \tag{4.3}$$

The vector $(-\frac{1}{2}, 0, \frac{1}{2})$, which consists of the coefficients in front of the function evaluations, is called a stencil. Here the second derivatives cancel and we are left with a term $h^2 \frac{\partial^3 u}{\partial x^3}(x_i)$ to leading order in h . The residual is thus $\mathcal{O}(h^2)$ and the stencil is said to be second order. It does not mean that the error is on the order of h^2 but that the error scales like h^2 . It is also possible that the expression result in a better scaling if $\frac{\partial^3 u}{\partial x^3}(x_i)$ is particularly small for instance.

There is a general procedure to determine stencils for more complex differential operators. A stencil is a vector describing the action of a differential operator on a function at a point as linear combination of the evaluations of the function at neighbouring points. Based on this, we write

$$\frac{\partial^k u}{\partial x^k}(x_j) = \sum_i \alpha_i u(x_i) \tag{4.4}$$

We Taylor expand $u(x)$ about x_j to evaluate u at each points x_i .

$$\begin{aligned} \sum_i^n \alpha_i u(x_i) &= u(x_j) \sum_i^n \alpha_i + u_x(x_j) \sum_i^n \alpha_i (x_i - x_j) + \\ &u_{xx}(x_j) \frac{1}{2} \sum_i^n \alpha_i (x_i - x_j)^2 + u_{xxx}(x_j) \frac{1}{3!} \sum_i^n \alpha_i (x_i - x_j)^3 + \dots \end{aligned} \quad (4.5)$$

We then demand the right combination of α_i 's to get an expression for a certain derivative. For example, if we want the k^{th} derivative, we demand

$$\begin{aligned} \sum_i^n \alpha_i &= 0 \\ \sum_i^n \alpha_i (x_i - x_j) &= 0 \\ \frac{1}{2} \sum_i^n \alpha_i (x_i - x_j)^2 &= 0 \\ &\vdots \\ \frac{1}{k!} \sum_i^n \alpha_i (x_i - x_j)^k &= 1 \\ &\vdots \end{aligned} \quad (4.6)$$

The α_i 's are found solving a Vandermonde system of equations where the right hand side is a vector of zeros except for the k^{th} entry which is equal to $k!$. For

example, we can obtain the 4th order second derivative solving

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ (-2h) & (-h) & 0 & h & (2h) \\ (-2h)^2 & (-h)^2 & 0 & h^2 & (2h)^2 \\ (-2h)^3 & (-h)^3 & 0 & h^3 & (2h)^3 \\ (-2h)^4 & (-h)^4 & 0 & h^4 & (2h)^4 \end{pmatrix} \begin{pmatrix} \alpha_{k-2} \\ \alpha_{k-1} \\ \alpha_k \\ \alpha_{k+1} \\ \alpha_{k+2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \\ 0 \\ 0 \end{pmatrix} \quad (4.7)$$

which yields

$$\begin{pmatrix} \alpha_{k-2} & \alpha_{k-1} & \alpha_k & \alpha_{k+1} & \alpha_{k+2} \end{pmatrix} = \frac{1}{12h^2} \begin{pmatrix} -1 & 16 & -30 & 16 & -1 \end{pmatrix} \quad (4.8)$$

We can thus write the second order derivative at x_i as

$$u_{xx}(x_i) = \frac{-u(x_{i-2}) + 16u(x_{i-1}) - 30u(x_i) + 16u(x_{i+1}) - u(x_{i+2}))}{12h^2} + \mathcal{O}(h^4) \quad (4.9)$$

Vandermonde matrices rapidly become ill-conditioned as they grow in size. Actually, their condition number typically grows exponentially with their size. For example, if a symmetric stencil on a uniform grid is sought, a system which is at most 17×17 (corresponding to an $\mathcal{O}(h^{16})$ stencil) can be inverted using double precision arithmetic. It is in fact the best case scenario since forward stencil and non-uniform grid yield worst-conditioned systems. We usually need a Laplacian of order $\mathcal{O}(h^{12})$ and symmetric stencils can be used everywhere with periodic boundary conditions. We are thus luckily close to the limit. Eventually

we will need to invert ill-conditioned systems and we shall use algorithms that have been developed from analytic expressions for the inverse of a Vandermonde matrix [50, 51].

In general, differentiability is an issue since we assume the existence of a derivative to solve for it implicitly. If a function is k -differentiable, then finite-differencing cannot provide information about the $k + 1^{th}$ derivative, nor can it give information more precise than $\mathcal{O}(h^k)$ about the lower derivative. We assume that our potentials are sufficiently well-behaved that we don't need to worry about the differentiability of the eigenstates. This is a reasonable assumption as we are using pseudopotentials.

No assumption was made concerning the uniformity of the grid although our examples are on uniform grid. In principle, one can use finite differences on a non-uniform grid. To evaluate a differential operator at a point, the method necessitate the evaluation of the function along the direction of the derivative. In principle, the point of evaluation do not need to coincide with the grid, but it is a complication since one then needs to interpolate between the grid points where the function is known for each evaluation. It is simpler to set grid points along the derivatives' direction. It is a major constraint since it forces the grid to be orthorhombic. In practice, one has to compute a different stencil for each point, which may become prohibitive for large systems. We thus use uniform grids in the present document.

We mention that our demonstration and examples were done in one di-

mension but the technique generalizes easily to multidimensional differential operators. When computing multidimensional stencils, cross-terms such as $\sum_i^n \alpha_i (x_i - x_j)(y_i - y_j)$ must be included in the Vandermonde system. Since we are ultimately interested in discretizing a Laplacian which has no mixed derivatives we don't need to go through this.

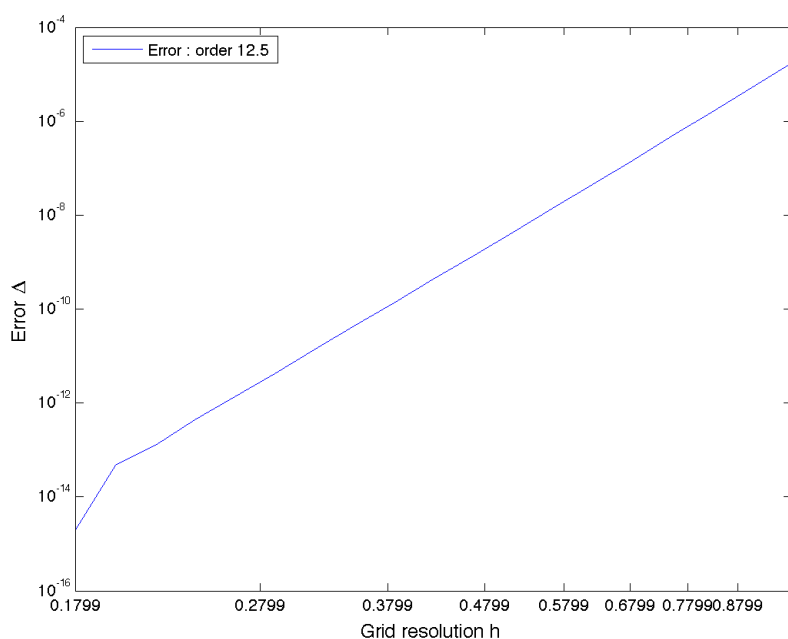


Figure 4.1: Residual norm Δ against step size h : the test problem is the evaluation of the second derivative of e^x at $x = 0$

A MATLAB function `stencil.m` was written. It returns an arbitrary order stencil of an arbitrary order derivative. The function file is tested by generating a stencil for a twelfth order second derivative. We test the stencil on the function e^x at $x = 0$. A convergence plot is shown in figure 4.1. The slope of the error $\Delta = e_{num}^x - 1$ against the step size is 12.5, thus confirming the expected

convergence rate to be order 12.

`stencil.m` only generates one dimensional stencils but we are interested in a 3-dimensional Laplacian. We use the symmetric nature of the multi-dimensional Laplacian to obtain it from the 1-dimensional Laplacian using the Kronecker tensor product recursively. Suppose that ∇_n^2 is a Laplacian in n dimension, then the Laplacian in $n + 1$ dimensions is obtained from $\nabla_{n+1}^2 = \nabla_n^2 \otimes I + I \otimes \nabla_1^2$, where I is the identity matrix.

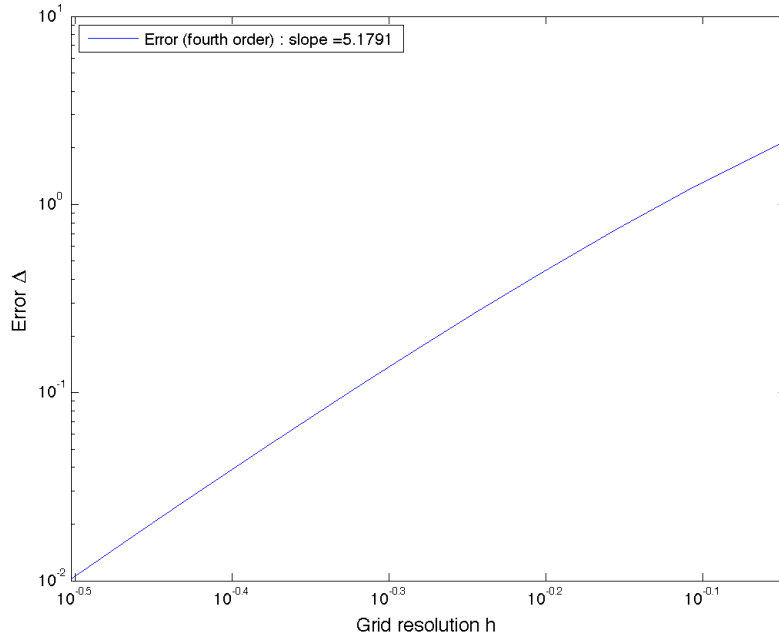


Figure 4.2: Residual norm Δ against step size h : the test problem is the evaluation of the Laplacian of $f(x, y, z) = \sin(x) + \sin(2y) + \sin(3z)$ on a Cartesian grid

The Laplacian is tested on a cubic grid $2\pi \times 2\pi \times 2\pi$ and the test function is $f(x, y, z) = \sin(x) + \sin(2y) + \sin(3z)$. A convergence plot is shown in figure 4.2.

The expected convergence rate is $\mathcal{O}(h^6)$ whereas the observed convergence rate is $\mathcal{O}(h^{5.1791})$. The plot shows that the slope increases as the step size h decreases indicating that the asymptotic convergence rate might indeed be $\mathcal{O}(h^6)$.

4.1.2 Discretization of the ionic potential

In the previous chapter, we described the pseudopotential generation procedure. We now discuss how to project the results on our real space grid.

In our real space basis, V_{local} takes the form of a real diagonal matrix. V_{local} is isotropic and hence we can write $V_{local}(\mathbf{r}) = V_{local}(r)$, that is V_{local} only depends on the magnitude of $\mathbf{r}$. We thus compute the distance r from each grid point to the center of the atomic nucleus and evaluate $V_{local}(r)$. We do not know the function for every r since we generated V_{local} on another grid and hence we need to interpolate the expression we obtained for $V_{local}(r)$. More than enough points are available so that any interpolation scheme is sufficiently accurate. Cubic interpolation, which is very precise in passing, is chosen because it is the fastest.

Computing $\sum_{l=0}^{l_{max}} \sum_{m=-l}^l |\chi_{lm}^{KB}\rangle v_l \langle \chi_{lm}^{KB}|$ (eq. 3.17) is the most time consuming part in the Hamiltonian generation because of the loops over over the quantum numbers l and m . A few considerations are required to make this step as efficient as possible. We know that some atoms do not have valence electrons with a given angular momentum. In those cases, the above procedure leads to small or vanishing non-local components $|\chi_{lm}^{KB}\rangle v_l \langle \chi_{lm}^{KB}|$. We can generally neglect them and generate the pseudopotentials for a specified set $\{l\}$. Note

that the radial part of χ_{lm}^{KB} only depends on l and that one should only interpolate $R_l^{PP}(r)$ once and not for each m . Furthermore, it is possible to rewrite

$$\sum_{m=\pm m_0} |\chi_{lm}^{KB}\rangle v_l \langle \chi_{lm}^{KB}| \text{ in the following way}$$

$$\sum_{m=\pm m_0} |\chi_{lm}^{KB}\rangle v_l \langle \chi_{lm}^{KB}| = 2v_l (\mathcal{R}(|\chi_{lm_0}^{KB}\rangle) \mathcal{R}(\langle \chi_{lm_0}^{KB}|) + \mathcal{I}(|\chi_{lm_0}^{KB}\rangle) \mathcal{I}(\langle \chi_{lm_0}^{KB}|))$$
(4.10)

This way, we half the loop over m and reduce the number of spherical harmonics that have to be generated. Another interesting consequence is that the non-local part of the pseudopotentials is no longer complex (the imaginary component was due to the spherical harmonics) which leads to a significant speed up since the linear algebra subroutines are performed on real numbers.

4.1.3 Generation of the Hartree potential

In section 2.2, we defined the Hartree potential as

$$V_H[\rho] = \int d\mathbf{r}' \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \quad (4.11)$$

V_H can thus be obtained by computing a three-fold integral for each grid point. Needless to say that it is not the most efficient way to evaluate V_H , especially when using high-level computer languages such as Matlab. We will show two efficient methods of computing V_H from the Poisson equation shortly, but we need to make some comments before.

It was mentioned that $V_{local}(r)$ is equal to the Coulomb potential beyond some radius r_c . Most pseudopotential generation software, including Nanobase [52], do not stop there. The local component is screened with the Hartree potential due to some charge distribution ρ_0 in such a way that $V_{local}(r)$ is strictly zero beyond some radius r_c . This has many fortunate implications including the locality of $V_{local}(r)$ in real space and the fast decay of the Kohn-Sham eigenstates away from the atoms. In the calculations, ρ_0 is taken as the initial guess and we simply update the Hartree potential using $\delta\rho = \rho - \rho_0$. For clarity, we write ρ instead of $\delta\rho$ in the present section.

As mentioned above, we can solve the differential equation 3.5 instead of equation 3.4. There are many ways of solving equation 3.5.

We already discussed the discretization of the operator ∇^2 . We shall thus use it to turn equation 3.5 into a linear matrix equation. The problem takes the form

$$Ax = b \tag{4.12}$$

where x is sought. It can be solved using a number of methods, direct (e.g. performing an LU decomposition) or iterative (e.g. preconditioned conjugate gradient). It turns out that this system is not always straightforward to solve when periodic boundary conditions are imposed because the matrix A becomes ill-conditioned.

As we are working with periodic boundary conditions, a simpler, accurate

and expeditious method is to solve equation 3.5 in Fourier space. Equation 3.5 is Fourier transformed to obtain

$$-k^2 \hat{V}_H(\mathbf{k}) = -4\pi \hat{\rho}(\mathbf{k}) \quad (4.13)$$

The Laplacian operator is diagonal in Fourier space and hence it is easily inverted. We simply compute

$$\hat{V}_H(\mathbf{k}) = 4\pi \frac{\hat{\rho}(\mathbf{k})}{k^2} \quad (4.14)$$

The inverse Fourier transform yields the answer $V_H(r)$. Fast Fourier transforms (FFT) are fast and show excellent scaling in terms of computational complexity, namely $\mathcal{O}(n \log(n))$ where n is the number of points transformed. Moreover, convergence is exponential such that a relatively coarse grid usually gives accurate results.

In section 3.1, we mentioned that the lack of parallelizability of FFT algorithms is one limitation of planewaves codes. In our implementation, the problem is diminished by taking advantage of the orthogonality of the grid. This characteristic allows us to Fourier transform quantities and operators along each dimension separately. For large systems, this is significantly faster than using 3D Fourier transforms and it may be affordable to compute the Hartree potential separately on each node. Our software also uses Matlab's biconjugate gradients stabilized (BICGSTAB) method to solve equation 3.5. We shall implement a

parallel version of BICGSTAB to handle systems for which the computation of the Hartree potential in Fourier space becomes computationally expensive.

4.1.4 *Generation of the exchange-correlation potential*

The non-classical effects are taken into consideration with the local density approximation (LDA) described above. We rewrite the exchange-correlation potential as

$$V_{xc} = V_x + V_c \quad (4.15)$$

where V_x is given by equation 2.43.

Perdew and Wang proposed a simple analytic expression for the correlation energy from we derive V_c [40]

$$\epsilon_c = -2A(1 + \alpha_1 r_s) \log \left(1 + \frac{1}{2A \left(\beta_1 r_s^{\frac{1}{2}} + \beta_2 r_s + \beta_3 r_s^{\frac{3}{2}} + \beta_4 r_s^{p+1} \right)} \right) \quad (4.16)$$

where

$$r_s = \left(\frac{3}{4\pi\rho} \right)^{\frac{1}{3}} \quad (4.17)$$

The correlation potential is found to be

$$V_c = \epsilon_c(\rho) + \rho(\mathbf{r}) \frac{\delta \epsilon_c(\rho)}{\delta \rho(\mathbf{r})} \quad (4.18)$$

$$= \epsilon_c(\rho) + \rho(\mathbf{r}) \frac{\delta \epsilon_c(r_s)}{\delta r_s} \frac{\delta r_s}{\delta \rho} \quad (4.19)$$

$$= \epsilon_c(\rho) - \frac{1}{3} \left(\frac{3}{4\pi\rho} \right)^{\frac{1}{3}} \frac{\delta \epsilon_c(r_s)}{\delta r_s} \frac{\delta r_s}{\delta \rho} \quad (4.20)$$

4.1.5 Hamiltonian Representation

We have shown that the Laplacian can be written as a non-local operator $\nabla^2(\mathbf{r}, \mathbf{r}')$. A fraction of the points surrounding a particular point is necessary to approximate the differential operator, which leads to a sparse matrix representation. All potentials, except the non-local part of the ionic potential, are local and take the form of a diagonal matrix. Those are added to the kinetic energy without changing the sparsity structure of the Hamiltonian matrix. The non-local contribution to the pseudopotentials is generally less sparse than the kinetic energy term. Fortunately, it can be written in real separable form. This form is not only more accurate, it provides a computationally efficient way of evaluating this part of the Hamiltonian. The Hamiltonian is thus real, sparse and symmetric. These are ideal conditions for diagonalization algorithms which are based on dense vector-sparse matrix multiplication.

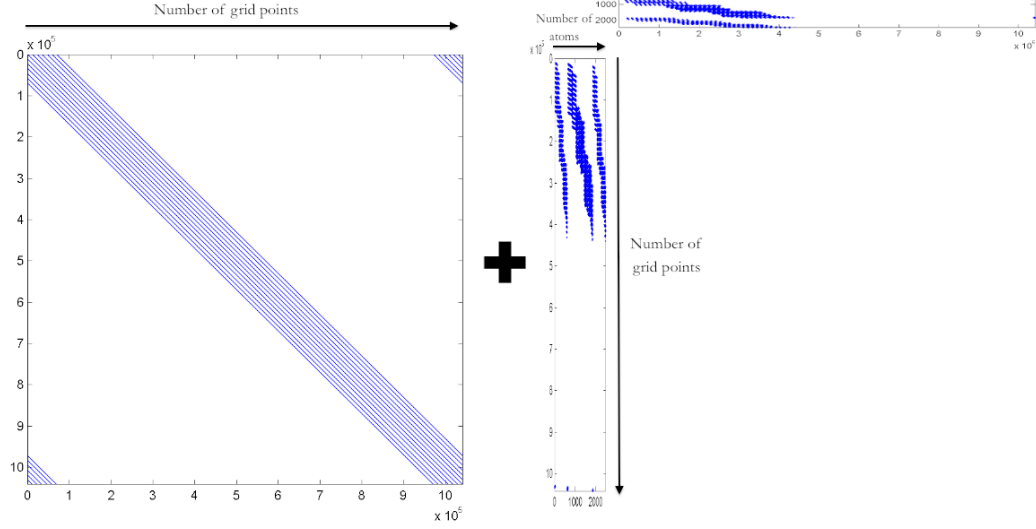


Figure 4.3: Sparsity pattern of the Hamiltonian matrix of the unit cell of a Si(111)- 7×7 surface in real space. The matrix on the left shows the sparsity pattern of a $\mathcal{O}(h^{12})$ -Laplacian. The matrix itself represent the sum of the kinetic energy operator and all local potential energy operators. The matrices on the right represent the non-local pseudopotential energy operator in separable form (i.e. as a product of two low rank matrices). The number of non-zero entries of the Laplacian matrix scales linearly with the number of grid points. The number of non-zero entries in the non-local pseudopotential matrices is proportional to the number of atoms.

We now develop on one of the closing remarks of section 3.1. The structure of the $\mathcal{O}(h^{12})$ -Laplacian of a Si(111)- 7×7 surface unit cell comprising 694 atoms is shown in figure 4.3. The matrix on the left represents the sum of the kinetic energy operator and the local potential energy operators. The matrices on the right represents the non-local pseudopotential energy operator in separable form (i.e. as a product of two low rank matrices). The non-local pseudopotential matrices have been stretched in the width direction because the sparsity pattern cannot be appreciated otherwise (the length to width ratio normally approaches

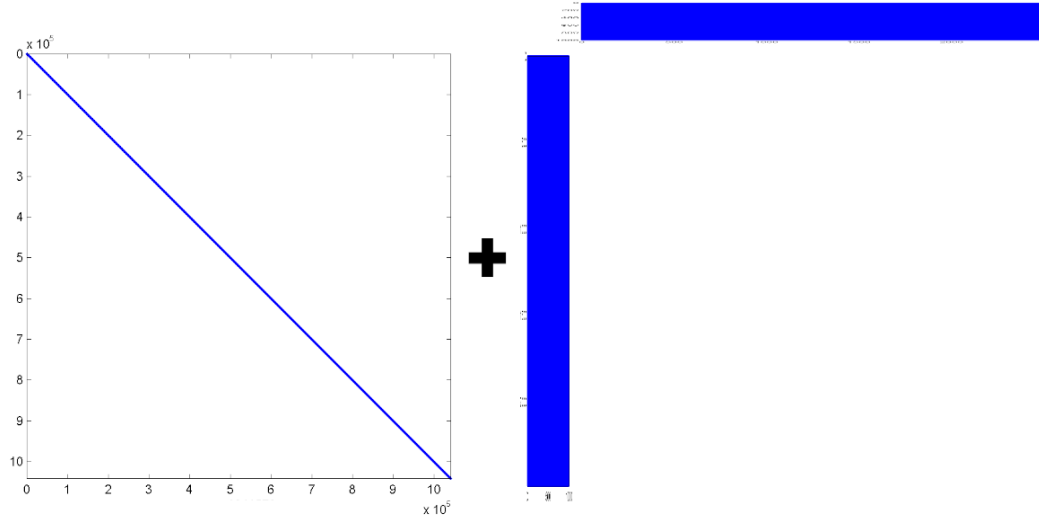


Figure 4.4: Sparsity pattern of the Hamiltonian matrix of the unit cell of a Si(111)- 7×7 surface in Fourier space. The number of non-zero entries of the Laplacian matrix scales linearly with the number of grid points. The number of non-zero entries in the non-local pseudopotential matrices is proportional to the number of atoms times the number of grid points.

1000:1). The Fourier space representation of the Hamiltonian is shown in figure 4.4. The Laplacian is now diagonal, but the non-local pseudopotentials are no longer sparse. This reflects the fact that the non-local pseudopotentials are very localized in real space and hence completely delocalized in Fourier space.

We indicated the scaling of the dimensions of each matrices in figure 4.3. The dimensions of the Laplacian and the length of the pseudopotential matrices are equal to the number of grid points. The width of the pseudopotential matrices is proportional to the number of atoms. We now analyze the behaviour of the sparsity of the Hamiltonian assuming a fixed resolution in both real space and Fourier space. In real space, the number of non-zero entries in the Laplacian scales linearly with the number of grid points and hence it scales linearly

with the number of atoms. The number of non-zero entries in the pseudopotential matrices also scales linearly with the number of atoms since additional grid points result in an equal number of supplemental zero entries. In Fourier space, the number of non-zero entries in the Laplacian scales linearly with the number of atoms but the number of non-zero entries in the pseudopotential matrices scales like the number of grid points times the number of atoms. At a fixed resolution, the number of non-zero entries in the pseudopotential matrices thus scales quadratically in Fourier space. The real space representation seems quite advantageous for large systems. As an example, for our Si(111)- 7×7 surface unit cell, the pseudopotential matrices occupy 25 MB of memory in real space whereas they occupy 20 GB of memory in Fourier space. As the system size increases, the numerous matrix-vector multiplications performed by the diagonalization procedures become considerably less costly in real space.

4.2 *Eigensolvers*

As previously mentioned, the bottleneck consists of solving the Schrödinger equation 3.7. We just showed that H takes the form of a real symmetric matrix. From the algorithmic point of view, the fact that H is Hermitian is very convenient [53].

The Hermitian eigenvalue problem is the simplest because the eigenvalues

are real. To see this consider a Hermitian matrix $A = A^\dagger$, then

$$Av = \lambda v = \lambda' v = A^\dagger v \quad (4.21)$$

$$\lambda = \lambda' \in \mathbb{R} \quad (4.22)$$

where $\dagger$ is used to denote the conjugate transpose of operators (matrices) and $'$ is used for scalars, vectors and block vectors. Heuristically, the simplification happens because it is easier to search along a line than to search a whole plane. Of course, nothing prevents us to go to higher dimensions before coming back where we know we will find our eigenvalues (i.e. we can still transit through $\mathbb{C}$). In addition, many algorithms require to find the left and right eigenvectors for general operators. They are of course the same for Hermitian operators which simplifies the algorithms and diminishes the computational cost.

The problem of electronic structure calculation is one of minimization of the energy and we are interested in finding the minimal eigenpairs consequently. In this section we present two algorithms that perform this task, namely the Chevyshev filtered subspace iteration (CFSI) and the locally optimal preconditioned conjugate gradient (LOBPCG) algorithms. A few algorithms necessary to CFSI will also be presented.

4.2.1 Power Method

We first present the power iteration algorithm. Although useless in practice, the algorithm illustrates the basic working principle of CFSI.

The eigenvectors of a Hermitian matrix A form a complete orthonormal basis

$$B = \{v_1, \dots, v_n\} \quad (4.23)$$

We can thus write a given vector as

$$y = \sum_i a_i v_i \quad (4.24)$$

If we apply the operator A^k to y we obtain

$$y = \sum_i \lambda_i^k a_i v_i \quad (4.25)$$

$$y = \lambda_n^k \sum_i \left(\frac{\lambda_i}{\lambda_n} \right)^k a_i v_i \quad (4.26)$$

where $\lambda_i < \lambda_{i+1}$ are the eigenvalues. From this expression, it is obvious that the eigenvector with the largest eigenvalue will eventually dominate and y will become numerically parallel to v_n . The algorithm works as follows :

Algorithm 1 Power Method [54]

```

generate an initial guess for the eigenvector  $y = y_0$ 
for  $k=1:niter$  do
     $v = \frac{y}{\|y\|}$ 
     $y = Av$ 
     $\theta = v'y$ 
    if  $\|y - \theta v\| \leq \epsilon$  then
        break
    end if
end for
set  $\lambda = \theta$  and  $x = v$ 

```

We make some comments on algorithm 1. In exact arithmetic, the algorithm would not work in the case were $y'_0 v_n = 0$. We do not worry about this possibility as no two vectors are perfectly orthogonal in finite precision. However, one should always use the best estimate at hand as a starting vector.

The rate of convergence is derived from the following observation. The largest eigenvectors v_n eventually wins, but it has to dwarf the second largest one v_{n-1} and hence the convergence rate is $\frac{\lambda_{n-1}}{\lambda_n}$.

4.2.2 Inverse Iteration

Any eigenpair can be computed by modifying the eigenspectrum of an operator. The spectrum must be changed in such a way that the eigenvalue of interest becomes the largest after transformation. The transform does not have to be bijective since once we have the eigenvectors we can recover the eigenvalues easily. This can be done in many ways but a most general technique is the inverse iteration algorithm which we briefly describe.

The inverse iteration algorithm is based on two observations. Let $Ax = \lambda x$ then

$$\begin{aligned} Ax - \sigma Ix &= \lambda x - \sigma Ix \\ (A - \sigma I)x &= (\lambda - \sigma)x \end{aligned} \tag{4.27}$$

The eigenvectors are unchanged by a shift of the eigenspectrum as mentioned above. We can thus perform power iteration on $A - \sigma I$ and subtract σ from the converged eigenvalue to obtain a different eigenpair of A . It is possible to get both ends of the spectrum by translating by the proper amount but it usually results in slow convergence rates. The second observation is that

$$\begin{aligned} Ax &= \lambda x \\ \lambda^{-1}x &= A^{-1}x \end{aligned} \tag{4.28}$$

in the case where A is invertible. The eigenvalues of A^{-1} are the inverses of the eigenvalues of A yet the eigenvectors are the same.

Using these two observations, we apply the power method to the operator $(A - \sigma I)^{-1}$ in order to obtain the eigenpair that is closest to σ . The closer σ is to an actual eigenvalue, the faster the convergence. We provide the following pseudo-code

Algorithm 2 Inverse Iteration [54]

```

generate an initial guess for the eigenvector  $y = y_0$ 
for  $k=1:niter$  do
     $v = \frac{y}{\|y\|}$ 
     $y = (A - \sigma I)^{-1} v$ 
     $\theta = v' y$ 
    if  $\|y - \theta v\| \leq \epsilon$  then
        break
    end if
end for
set  $\lambda = \sigma + \frac{1}{\theta}$  and  $x = v$ 

```

We point out that $(A - \sigma I)^{-1}$ need not be computed explicitly, a system $(A - \sigma I)y = v$ is solved instead. We suggest to perform a LU decomposition before entering the **for** loop from which y is then obtain by forward backward substitution.

The algorithm will yield the largest eigenvalue of $(A - \sigma I)^{-1}$, that is the one closest to σ . Following the analysis of the power method, the convergence rate is found to be $\frac{\lambda_k - \sigma}{\lambda_l - \sigma}$ where λ_k and λ_l are the closest and second closest eigenvalues to σ respectively. When a good approximation of λ_k is known, the eigenpair converges rapidly as the function $f(x) = (x - \lambda_k)^{-1}$ blows up. In general, one does not have a reasonable *ansatz* and hence the convergence can be quite slow.

4.2.3 Orthogonal Subspace Iteration

Every methods presented above work nicely when we seek a single non-degenerate eigenpair. Solving the Schrödinger equation requires finding many degenerate

eigenpairs most of the time. In this section, we present an extension of the power method called orthogonal subspace iteration. We present a pseudo-code of the algorithm

Algorithm 3 Orthogonal Iteration [54]

```

generate an initial guess for the orthonormal block vector  $V$ 
for  $k=1:niter$  do
     $Y = AV$ 
     $\Theta = V'Y$ 
    if  $\|Y - V\Theta\| \leq \epsilon$  then
        break
    end if
    orthonormalize  $Y$ 
end for
set  $\Lambda = \Theta$  and  $X = V$ 

```

The orthonormality can be maintained using any algorithm. Dividing Y by the Cholesky factor of $Y'Y$ is generally efficient. The orthogonality constraint allows us to find many eigenvectors at the same time.

Suppose that we perform the algorithm on a block vector of width m . Because of the orthogonality constraint, all eigenvectors compare to the largest eigenvector that is not in the subspace and hence the i^{th} eigenpair will converge at a rate $\frac{\lambda_{m+1}}{\lambda_i}$. It is thus natural to iterate on a subspace dimension larger than the number of sought eigenpairs to accelerate convergence. The optimal size of the subspace depends on the operator A in question, but 8 or 10 extra dimensions does it most of the time.

This method is much more efficient than the power method for many reasons. As we just saw, convergence is accelerated by choosing a subspace larger

than strictly necessary. The extra cost is the orthonormalization of the subspace. This is usually not computationally intensive since the dimension of the subspace is fixed and relatively small. We point out that $Y = AV$ could be replaced by $Y = A^s V$ advantageously as orthonormalization is not required at each step. But if s is too large the numerical rank of the blockvector Y might decrease and the algorithm will have to be restarted. Like most block methods degenerate eigenpairs are handled perfectly by this algorithm.

4.2.4 *Chebyshev Filtered Orthogonal Iteration*

Naturally, a better technique will combine the spectrum transformation and orthogonal iteration methods.

We should not lose sight that our ultimate goal is to compute the smallest eigenpairs of our Hamiltonian. We discuss the characteristics of a good spectrum transformation compatible with our goal. Simply put, a spectrum transformation is done by diagonalizing some function of the Hamiltonian. The function must be constructed from a finite number of multiplications and inversions because they are the matrix operations we can perform. Inverting the spectrum with respect to some value does not yield good results even for small subspaces because eigenvalues are either too much or not enough suppressed. We must concentrate on polynomials since any analytic function will have to be expressed in terms of its truncated power series. The transform should suppress all the eigenvalues but the smallest as much as possible in order to yield the best convergence rate.

The last statement is formulated as a mathematical problem to determine what is the best polynomial one can use. We distinguish between two regions of a polynomial :

1. the asymptotic region i.e. the region where the leading order term is dominant
2. the transition region i.e. the region between the two asymptotic regions

Suppose that we have estimates for the smallest and largest eigenvalue $\{\tilde{\lambda}_a, \tilde{\lambda}_b\}$ that we don't want, then we can stretch and translate the polynomial so that the transition region corresponds to the interval $[\tilde{\lambda}_a, \tilde{\lambda}_b]$. That leads us to the following question : What is the monic polynomial (with leading coefficient 1) that has the smallest infinity norm (achieves the smallest value in magnitude) on the interval $[-1, 1]$? It is easy to show that Chebyshev polynomials of the first kind $T_n(x)$ satisfy precisely the last statement. Chebyshev polynomials also have the convenient property that they can be defined recursively $C_{n+1}(x) = 2xC_n(x) - C_{n-1}(x)$.

We present the Chebyshev filtered subspace iteration (CFSI) algorithm

Algorithm 4 Chebyshev Filtered Subspace Iteration

```

generate an initial guess  $X$ 
estimate the smallest and largest unwanted eigenvalues  $\{\tilde{\lambda}_a, \tilde{\lambda}_b\}$ 
for  $k=1:niter$  do
    apply the Chebyshev spectral transform  $X = \text{Cfilter}(A, X, m, \tilde{\lambda}_a, \tilde{\lambda}_b)$ 
    orthonormalize  $X$ 
    project  $A$  in  $\text{span}(X)$   $H = X'AX$ 
    diagonalize  $H = S\Theta S'$ 
    update  $X = XS$ 
    compute the residual block vector  $R = AX - X\Theta$ 
    check convergence
end for

```

$\tilde{\lambda}_a$ is obtained by diagonalizing A in the subspace spanned by X . The Lanczos algorithm is used to find the largest eigenvalue. It will be presented in the next section.

This is a satisfying improvement compared to the power method algorithm. It is not the fastest in general, but it is reliable, easy to understand and code. We point out that Chebyshev-filtered orthogonal iteration has been used conclusively for DFT calculations by Zhou et al.[7].

4.2.5 Lanczos Algorithm

The Lanczos algorithm finds the eigenpairs of a Hermitian operator A by diagonalizing it in a Krylov subspace $\mathcal{K}_j(A, v) = \text{span}\{v, Av, A^2v, \dots, A^{j-1}v\}$. There is enough freedom to construct the basis of $\mathcal{K}_j(A, v)$ such that the projection of A is tridiagonal.

$$V_j'AV_j = T_j = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & & & \\ & & \ddots & \ddots & \\ & & \ddots & \alpha_{j-1} & \beta_{j-1} \\ & & & \beta_{j-1} & \alpha_j \end{pmatrix} \quad (4.29)$$

This is done using the recursion

$$AV_j = V_jT_j + re_j' \quad (4.30)$$

with the constraint $V_j'r = 0$. We deduce that

$$\begin{aligned} r_j &= Av_j - v_{j-1}\beta_{j-1} \\ \alpha_j &= v_j'r_j \\ \beta_j &= \|r_j - v_j\alpha_j\| \end{aligned} \quad (4.31)$$

As above, we compute $T_j = S\Theta S'$ and the approximate eigenvectors in $\mathcal{K}_j(A, v)$ are $x_j = V_jS_{:,j}$. The eigenpair θ_j, x_j is called a Ritz eigenpair. In principle, we will have

$$r_i = Ax_i - x_i\theta_i = AV_jS_{:,i} - V_jS_{:,i}\theta_i = (AV_j - V_jT_j)S_{:,i} = V_{:,j+1}\beta_jS_{j,i} \quad (4.32)$$

Here i labels the i^{th} eigenpair and j is the dimension of $\mathcal{K}_j(A, v)$. It follows that

$$\|r_i\| = \beta_j S_{j,i} \quad (4.33)$$

In practice, we cannot always maintain $\|V_{:,i}\| = 1$, and hence we may keep it when computing the norm of the residual to be safe.

We now present the Lanczos algorithm

Algorithm 5 Lanczos Algorithm

```

generate an initial guess for smallest eigenvector  $v = r$ 
set  $\beta_0 = \|r\|$ 
for  $k=1:niter$  do
     $v_j = \frac{r}{\beta_{j-1}}$ 
    orthonormalize  $V$ 
     $r = (A - \sigma I)^{-1} v_j$ 
     $r = r - v_{j-1}\beta_{j-1}$ 
     $\alpha_j = v_j' r$ 
     $r = r - v_j \alpha_j$ 
     $\beta_j = \|r\|$ 
    diagonalize  $T = S\Theta S'$ 
    update  $V = VS$ 
    test convergence
end for
set  $X = VS$ 

```

A good discussion of the convergence analysis for the extremal eigenpairs is found in [55]. It is not necessary to delve too much into the method since we only use it to compute an upper bound for the eigenvalues of the Hamiltonian. It takes 8 or 10 steps generally after which the residual is added to provide an upper bound.

4.2.6 LOBPCG

The Lanczos algorithm builds a basis for the Krylov subspace in which the operator A is tridiagonal. There are better subspaces in which A can be diagonalized. In particular, we do not worry about the form of the projected Hamiltonian since it is relatively small. With that in mind, we present the locally optimal block preconditioned conjugate gradient algorithm (LOBPCG) [8].

The idea is to iterate on a subspace of fixed dimension (contrary to Lanczos algorithm). At each step, the preconditioned residual is computed $R = T(AX_i - X_i\Lambda)$ where T is some preconditioner. A Rayleigh-Ritz process is then performed on $\text{span}\{X_i, X_{i-1}, R\}$ to obtain X_{i+1} . Some care is needed as X_i and X_{i-1} will become “parallel” as i increases. This is where the conjugate directions P_i come into play. They are defined as the directions in $\text{span}\{X_i, X_{i-1}\}$ that are orthogonal to $\text{span}\{X_i\}$. The algorithm goes as follows

Algorithm 6 LOBPCG Algorithm

```

generate an initial subspace  $X^1$ 
set conjugate subspace  $P^1 = 0$ 
for  $k=1:niter$  do
  compute residual  $R^k = AX^k - X^k\Lambda^k$ 
  test convergence
  preconditioning  $W^k = TR^k$ 
  make  $W^k \perp X^k$ 
  orthonormalize  $W^k$ 
  orthonormalize  $P^k$ 
  Rayleigh-Ritz on  $\text{Span} \{X^k, W^k, P^k\}$ 
  obtain  $x_j^{k+1} = \sum_i \alpha_i^k w_i^k + \tau_i^k x_i^k + \gamma_i^k p_i^k$  corresponding to the eigenvalues of
  interest
  set  $p_j^{k+1} = \sum_i \alpha_i^k w_i^k + \gamma_i^k p_i^k$ 
end for

```

We believe that this method is the best among those presented so far. Being a block method, it handles clustered or degenerate eigenpairs very well. It can take advantage of a good *ansatz* for eigenvectors. We can also play with the dimension of the iterative subspace to control convergence as in the orthogonal iteration although it is less necessary. Finally, even more freedom is provided as we can use preconditioning to accelerate convergence. Before illustrating this, we give a brief review of preconditioning.

Differential equations are ubiquitous in most scientific fields. Analytical solutions are rarely available so that they are solved numerically. In order to do so, differential equations are discretized and take the form of a linear system such as $Ax = b$. In some applications, the matrix A is such that it cannot be inverted in standard double precision arithmetic and the system is

said to be ill-conditioned. It is sometimes possible to palliate this by applying a preconditioner T to A . The system then becomes $(TA)x = (Tb)$ and with a good enough preconditioner the matrix (TA) will not be ill-conditioned anymore. A common choice of preconditioner is to use an approximate inverse of A but it is often preferable to use one that is more problem-motivated when possible.

We observed that preconditioning the residual is very rewarding. To illustrate that, we solve the Schrödinger equation with a smooth local potential without preconditioning first and then using an incomplete LU (ILU) preconditioner. The tolerance of the ILU factorization is set to infinity which means that the sparsity pattern of $L + U$ is the same as that of the original matrix. The factors obtained are not very accurate but the computational and memory costs are small. The residual vector norm for the 5 smallest eigenpairs is shown in figure 4.5. The number of iterations required for convergence is reduced by a factor of 2.5 or more due to ILU preconditioning. Given the small computational cost of the ILU preconditioning, the speed up is similar.

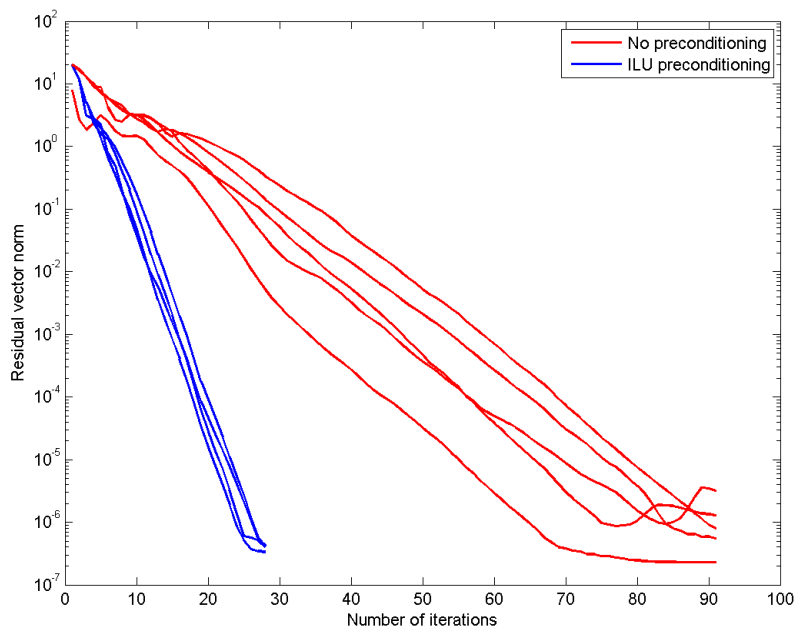


Figure 4.5: Residual vector norm of the 5 lowest KS eigenstates of a molecule against the number of iterations during LOBPCG. The red curves represent the residual vector norm when LOBPCG is used without preconditioning. The blue curves represent the residual vector norm when LOBPCG is used with ILU preconditioning.

When working with a more complex Hamiltonian such as the one shown in figure 4.3, we have additional choices of preconditioners. We can invert exactly or approximately some of the terms. In a planewave basis, it is common to precondition with the inverse of the kinetic energy. The fact that the kinetic energy accounts for a large part of the energy and that it is diagonal in that basis motivates this choice. In a real space basis, the cost of inverting the kinetic energy or the sum of the kinetic energy and the local potentials is similar. The later option is chosen because it is expected to be more accurate. We

thus perform a ILU factorization on $-\frac{1}{2}\nabla^2 + V_{local}(r) + V_H(r) + V_{XC}(r)$. It is then possible to include the effects of the non-local pseudopotential using the binomial inverse theorem [56]:

$$(\mathbf{A} + \mathbf{UBV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{UB}(\mathbf{B} + \mathbf{BVA}^{-1}\mathbf{UB})^{-1}\mathbf{BVA}^{-1} \quad (4.34)$$

Nevertheless, we observed that it is often preferable not to include the non-local pseudopotential in the preconditioner.

4.3 *Mixing techniques*

The Kohn-Sham equations 2.36 and 2.37 are a nonlinear eigenvalue problem and hence they must be solved self-consistently. We briefly reiterate the resolution process. We begin by solving the Schrödinger equation

$$\left(-\frac{1}{2}\nabla^2 + V_{total}(r, r')\right) \Psi(r) = E\Psi(r) \quad (4.35)$$

We compute the few eigenstates with the lowest energy. Next, the electronic density is calculated as

$$\rho(r) = \sum_{i=1}^s |\Psi_i(r)|^2 \quad (4.36)$$

The density allows us to update the Hartree and exchange-correlation potentials from which we derive a new total potential $V_{total}(r, r') = \tilde{V}_{total}(r, r')$. This finally gives us a new Schrödinger equation to diagonalize.

We shall view the sequence of potentials generated by the algorithm merely

as search directions. Updating the Hamiltonian in the way described above is likely to be counterproductive. In fact, there is no guarantee that the new potential will be closer to self-consistency than the last one. We need to reserve the right to go backward if self-consistency is lost. However, this cannot be done by computing each residual and keeping the potential that did the best. If the best were not the new potential, the update would again be one of the potentials already at our disposition and we would find ourselves stuck in a loop. A simple way to avoid this situation is to use linear mixing. A new search direction is computed from the old ones as

$$V_{in}^{n+1} = \alpha V_{out}^n + (1 - \alpha) V_{in}^n \quad (4.37)$$

where $0 \leq \alpha \leq 1$. The smaller α is, the more guaranteed but the slower the convergence is. It is thus tempting to use a large α even though we know it may not converge at all. A generalization where the last few directions are used follows naturally

$$V_{in}^{n+1} = \alpha V_{out}^n + \sum_{i=0} \beta_i V_{in}^{n-i} \quad (4.38)$$

where $\alpha + \sum_{i=0} \beta_i = 1$. There is not much improvement in doing that in general. It is possible to greatly accelerate convergence by allowing the β_i 's to change. We can use the information contained in the earlier search directions to derive the optimal search direction. It is necessary to have a term such as αV_{out}^n in

order to add new dimensions to the search.

Many procedures to compute the β_i 's exist. Anderson mixing has been used successfully [57, 6]. The mixing scheme [58] introduced by Pulay, often called direct inversion in the iterative subspace, is widely used as well [59, 60, 23]. We chose to use the Broyden mixer described by Srivastava in [61]. More details on Broyden's method are available in [62].

4.4 *Summary*

In this chapter, we introduced the method of finite differences. As presented, the method applies to non-uniform grids and can be used to generate non-symmetric stencils. We showed how to use the symmetry of the Laplacian to generate a N-dimensional Laplacian from a 1-dimensional Laplacian. Convergence tests have verified that both stencils and Laplacian have the predicted scaling behaviour. We described how the pseudopotentials are projected on the real space grid. We mentioned a few implementation details which may accelerate significantly the generation of the Hamiltonian. We introduced a few techniques for solving the Poisson equation which is solved to update the Hartree potential. We briefly introduced the form of Perdew and Wang for the exchange and correlation potential. The Hamiltonian matrix was shown to be real, sparse and symmetric. Eigensolvers adapted for this particular problem were presented consequently. A few simple algorithms such as the power method have been introduced to illustrate the basic working principles of CFSI. LOBPCG was also presented. We

showed the gain realized by using a ILU preconditioner in LOBPCG. Finally, we mentioned many potential mixing techniques necessary to the fast convergence of the self-consistent iterations.

Applications

5.1 *MatRcal Input*

In chapter 3 we presented a real space pseudopotential method for Kohn-Sham density functional theory calculations. In chapter 4 we presented the mathematical tools used to perform the procedure presented in chapter 3 along with some implementation details. In order to have a global view of the method, it is instructive to have a look at the input file of an H₂ molecule.

```
matRcal.path = 'C:/User/MATLAB/MatRcal/PotentialData/';

matRcal.xyz = ...
{
'H'      0      0      0.3815  1  []
'H'      0      0     -0.3815  1  []
};
matRcal.units.xyz = 'A';

matRcal.dimension = [20,20,20];
matRcal.units.dimension = 'Bohr';

matRcal.refinegrid = 1;
matRcal.refinegridpath = 'C:/User/MatRcal/SetupFiles/H2/H2_40.mat'
```

```

matRcal.resolution.present = [60,60,60];

matRcal.laplacian.accuracy = 12;

matRcal.eigensolver.CFilterDegree = 12;
matRcal.eigensolver.extraEigen = 2;

matRcal.eigensolver.tolerance = [1e-4,1e-4,1e-6];
matRcal.eigensolver.maxIteration = 30;
matRcal.eigensolver.precond.name = 'ilup';
matRcal.eigensolver.precond.setup.type = 'nofill';

matRcal.mixing.type = 'Broyden';
matRcal.mixing.maxhistory = 30;
matRcal.mixing.beta = 0.8;
matRcal.mixing.tolerance = 1e-4;

matRcal.maxSCloop = 50;

matRcal = MatRcal(matRcal);

```

The field `path` holds the location of the folder containing the pseudopotentials. They are generated and stored before the computation using the procedure described in section 3.3. The field `xyz` contains the specie, coordinates, number of valence electrons and the angular momenta of the Kleinman-Bylander projectors for each atom. As mentioned in section 4.1.2, depending on the atom specie and the chemical environment, some contributions to the non-local pseudopotential may be vanishingly small. The zero angular momentum component is included by default. One can then add [1,2,3] in the last column of `xyz` to include the p , d or f contributions to $V_{nl}(\mathbf{r}, \mathbf{r}')$. The field `units` contains the units in which the atom coordinates are expressed and the units of other

physical quantities such as the simulation box size. Atomic coordinates are often given in Angström and sometimes in nanometers, and hence they have to be converted to atomic units since it is simpler to work with. The size of the simulation box is specified in the field `dimension`. If vacuum is used to simulate non-periodic boundary conditions, it is often simpler to use the same units as the atom coordinates to get a sense of its size. The subfield `resolution.present` contains the number of points used along each direction. The total number of points is the product of the three vector entries. It is possible to use the results of a previous calculation to begin (or continue) a new one. This is achieved by setting the field `refinegrid` equal to 1. The path to the file containing the results of the previous calculation is set in `refinegridpath`. As the name of the field suggests, the results of a calculation performed on a coarse grid can be used to accelerate a calculation on a refined grid. It is also possible to save a calculation that has not terminated and continue it later without modifying the grid size. The field `laplacian.accuracy` contains the order of the Laplacian. The Laplacian generating function uses it together with the grid size and resolution to generate the Laplacian. This is all done ahead of the self-consistent loop.

The eigensolver parameters must then be determined. When using CFSI, the degree of the Chebyshev polynomial is assigned in `eigensolver.CFilterDegree`. A typical value lies between 6 and 12. The number of extra subspace dimensions is defined in `eigensolver.extraEigen`. It is customary to use between 2

and 10 extra dimensions. LOBPCG requires the user to stipulate the tolerance (it can be an interval) for the eigensolver in `eigensolver.tolerance` and the maximal number of iterations in `eigensolver.maxIteration`. The residual norm tolerance is usually taken to be equal or slightly below the target accuracy for the effective potential or the electronic density. The maximal number of iterations is quite variable but normally ranges from 20 to 50. If a preconditioner is used, the link to the preconditioning function is passed through `eigensolver.precond.name`. Additional information required by the preconditioner is added to the field `eigensolver.precond` if necessary. In the case of an ILU preconditioner, we can specify the tolerance for example.

A few mixing techniques were mentioned in section 4.3. The user can set all the necessary parameters in the field `mixing`. The name of the mixing function is assigned in the subfield `mixing.type`. In the case of Broyden mixing, we need to define the number of potentials to be mixed and the proportion of the new search direction compare with the norm of the potential. The convergence criterion for the self-consistent loop is set in the field `mixing.tolerance`.

The program is finally started by inputting the structure `matRcal` in the function `MatRcal`.

5.2 *High-order finite-differencing*

In section 4.1, we introduced the whole machinery of high-order finite-differencing and stencil computation. We justify this by looking at the effects of using low-

order Laplacians in the Kohn-Sham Hamiltonian.

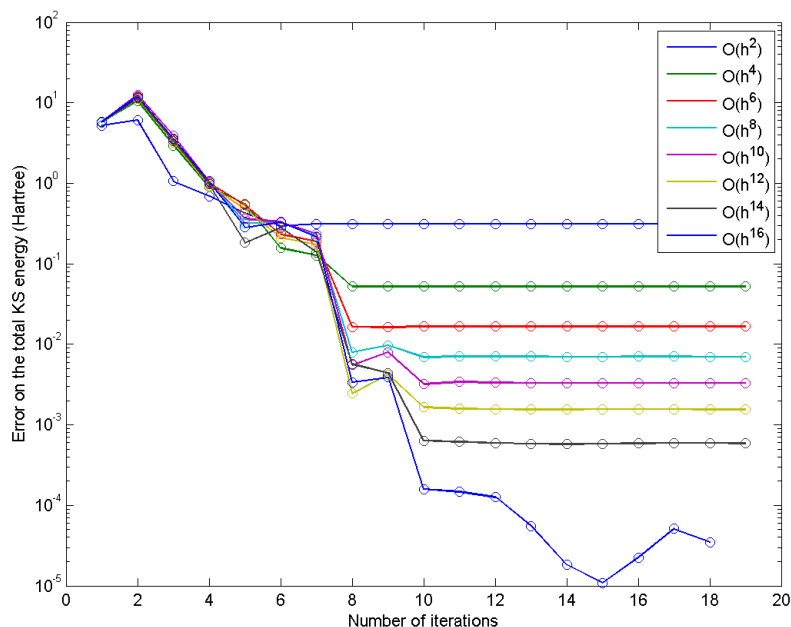


Figure 5.1: Error on the total KS energy of a Benzene molecule with respect to the number of iterations for different Laplacians. Laplacian orders range from $O(h^2)$ to $O(h^{16})$.

We compute the total KS energy of a benzene molecule using $O(h^2)$ Laplacians up to $O(h^{16})$ Laplacians. The total KS energy is defined as the sum of the energies of the occupied eigenstates of the KS Hamiltonian. It is not equal to the total energy of the system. As mentioned in section 2.2, the KS equations are merely an intermediary step in the search for the electronic ground state and the KS energies are expected to be non-physical. The error on the total KS energy against the number of iteration is displayed in figure 5.1. The error is defined as the difference between a given energy and that obtained at the last step

using the $\mathcal{O}(h^{16})$ Laplacian. We observe that the error in each computation (but the $\mathcal{O}(h^{16})$ computation) has plateaued after roughly 10 steps. This alone does not tell us much about the importance of using high-order finite-differencing. We also observe that the asymptotic error is perfectly correlated with the order of the Laplacians. This, together with the fact that high-order Laplacians are expected to be more accurate, strongly suggests that high-order Laplacians truly describe the kinetic energy better than low-order Laplacians for a fixed number of points. When performing a computation with a low-order Laplacian, we may converge to non-physical solutions even though nearly perfect self-consistency is achieved. This is because we find the electronic density that minimizes an energy functional which is not that of the system. There is no rule of thumb to know whether we have obtained a non-physical solution. In the present case, the spurious solution is a mathematical artefact due to a too low-order Laplacian. One way to check whether a solution is correct is thus to increase the order of the Laplacian before comparing the effective potentials or electronic densities of the two computations. This is normally not too computationally expensive since the converged potential and density can be used to begin with.

The order of the Laplacian is not the only important parameter. In order to obtain physical results, it is also important to have enough vacuum around non-periodic systems such as molecules. In order to confirm whether the molecules are sufficiently spaced, the dimensions of the simulation box are changed and the computation restarted. It is also important to have a sufficiently high grid reso-

lution to capture all the features of the pseudopotentials and the wavefunctions. Similarly, the grid resolution is increased until the results remain unchanged following a grid refinement. As mentioned above, one can always use the previous calculation as a starting point and the convergence is fast given that the solution was authentic. A concrete example of a non-physical solution will be discussed in the next section.

5.3 *HOMO-LUMO Gaps*

In order to validate that our program works correctly, we compute the energy gap between the highest occupied molecular orbital (HOMO) and the lowest unoccupied molecular orbital (LUMO) of different molecules. We then compare our results with the Computational Chemistry Comparison and Benchmark DataBase (CCCBDB) of the National Institute of Standards and Technology (NIST). CCCBDB comprises the HOMO-LUMO gap of a large number of molecules. Most molecules have been studied with many different density functionals and basis sets. We compare the results of MatRcal with those obtained with Gaussian using the basis sets 6-31G* and 6-31+G**, and a LDA functional. The program Gaussian can also perform a relaxation of the ionic structure. The atom coordinates obtained from 6-31G* and 6-31+G** are thus different in general. The atom coordinates obtained with 6-31G* were used by our program. The HOMO-LUMO gaps obtained with 6-31+G** are shown to give an order of magnitude for the discrepancy between different methods.

Molecule	HOMO-LUMO Gap (Hartree)			
	6-31G*	6-31+G**	MatRcal	Time (s)
CO ₂	0.313	0.310	0.319	85
CH ₄	0.433	0.354	0.433	18
C ₂ H ₄	0.218	0.209	0.219	105
C ₄ H ₁₀	0.333	0.276	0.318	124
C ₆ H ₆	0.194	0.190	0.194	66
C ₅ H ₅ N	0.148	0.146	0.145	47
C ₆ H ₅ NH ₂	0.147	0.143	0.142	23
C ₆ H ₅ NO ₂	0.119	0.118	0.122	97
CH ₃ COOH	0.200	0.198	0.189	178
CH ₂ C(CH ₃)CH ₃	0.202	0.193	0.201	282
SiH ₄	0.334	0.303	0.340	74
HCl	0.267	0.258	0.269	46

Table 5.1: Comparison of “HOMO-LUMO” gaps (Hartree) by LDA using Gaussian (NIST CCCBDB database) and MatRcal. The rightmost column shows the time it took MatRcal to reach convergence (i.e. $\delta V_{eff} < 10^{-4} V_{eff}$).

All results are displayed in table 5.1. We use a Laplacian of order 12 and 10 to 14 Bohr of vacuum along each direction. This means that the distance between the outermost atoms of each molecules is at least 10 to 14 Bohr. We typically use a 3 points per Bohr resolution along each direction. The results of 6-31G* and MatRcal generally agree up to a few percents. We think it is in excellent agreement considering that Gaussian uses (most probably) different pseudopotentials, a different expression for the exchange-correlation energy functional, a different basis set and different techniques throughout the resolution of the Kohn-Sham equations.

Experimental results for the HOMO-LUMO gaps are not reported as the Kohn-Sham DFT is notoriously bad at predicting gaps. This is especially true

when DFT is carried within the local density approximation which we are using. In DFT, energy gaps are computed as the energy difference between the lowest unoccupied KS eigenstate and the highest occupied KS eigenstate plus a contribution originating from the discontinuity of the exchange-correlation potential at integer particle numbers Δ_{xc} [63]. In the LDA, the exchange-correlation potential is continuous at integer particle numbers and hence $\Delta_{xc} = 0$, hence the gap misestimation.

The last column of table 5.1 shows the time it took for MatRcal to reach convergence. We cannot report the Gaussian timings as NIST does not provide them in the CCCBDB (possibly because the software license prevents it). The effective potential $V_{eff} = V_H + V_x + V_c$ is converged to one part in 10^{-4} . This convergence criterion is more than sufficient in order to stabilize the HOMO-LUMO gaps. The gap (and KS energies) usually converges well before the effective potential does indeed. All computations were done on a ThinkPad W520 equipped with an Intel i7-2720QM processor. LOBPCG with ILU preconditioning was used for every molecules. The timings found in the rightmost column of table 5.1 range from 20 seconds to 5 minutes.

From table 5.1, we notice that the timings are poorly correlated with the number of atoms or electrons. For example, it takes 85 seconds to compute the electronic structure of a CO₂ molecule and only 23 seconds to compute that of a C₆H₅NH₂ molecule. In our opinion, the most likely explanation is that the convergence time is greatly dependent on the simulation parameters, among

which :

1. the simulation box size;
2. the resolution of the grid;
3. the Laplacian accuracy;
4. the number of additional subspace dimensions;
5. the Chebyshev polynomials degree (CFSI);
6. the eigensolver tolerance (LOBPCG);
7. the maximal number of eigensolver iterations (LOBPCG);
8. the preconditioning technique (LOBPCG);
9. the mixing procedure parameters;
10. the convergence criterion.

The parameters listed above are related to the numerical methods used by the software. Unlike empirical parameters, they do not need to be finely tuned in order to obtain the correct physical quantities. Instead, they merely have to lie inside some (possibly unbounded) interval. For instance, the simulation box must be large enough to simulate a non-periodic arrangement, the grid resolution must be high enough to properly describe the potentials and the wavefunctions, the Laplacian order must be high enough to describe the kinetic

energy operator accurately, etc. Changing one parameter generally changes completely the convergence path (or the sequence of potentials, see subsection 4.3) and hence the convergence time changes likewise. This is true of iterative algorithms in general, but this behaviour is exacerbated by the non-linear nature of the Kohn-Sham equations. It is usually possible to adjust the parameters to get a faster convergence, but we did not do so for the computations reported in table 5.1. We do not want to mislead the reader into thinking that MatRcal is faster than it really is and it allows us to underline the importance of choosing the parameters carefully. One can get some instinct by practising on small molecules but in general it is impossible to know the optimal parameters in advance.

The convergence time also depends a lot on the initial electronic density *ansatz* ρ_0 . MatRcal uses the isolated atoms density by default. A large molecule whose equilibrium density resemble the isolated atoms density is expected to converge faster than a small molecule whose equilibrium density is very different from the isolated atoms density. One could possibly use first order perturbation theory to obtain a more educated guess for ρ_0 .

We complete the present section by showing that the convergence path is sometimes far from regular. As an example, we compute the electronic structure of a SiH_4 molecule and plot the fluctuation of the total Kohn-Sham energy as a function of the number of self-consistent cycles in figure 5.2. For this calculation, the CFSI algorithm was used instead of LOBPCG. After fifty iterations, the

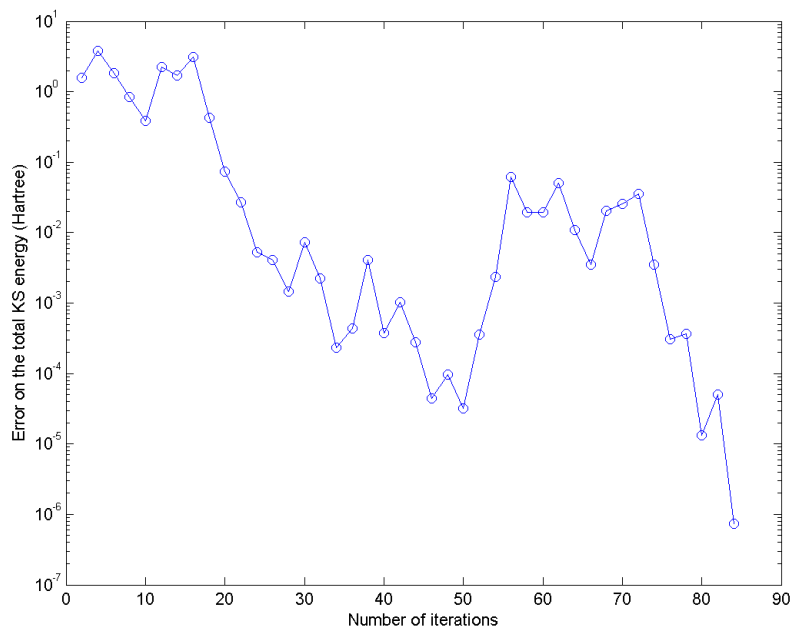


Figure 5.2: Error on the total KS energy of a SiH_4 molecule with respect to the number of iterations.

energy (and the effective potential) has stabilized and the energy fluctuations are smaller or equal to 10^{-4} Hartree giving the impression that convergence is achieved. The physical quantities obtained from this solution are incorrect. For instance, the HOMO-LUMO gap is 0.032 Hartree which is off by one order of magnitude. Taking a few more steps informs us that proper convergence was not achieved as the energy fluctuations swell to 10^{-1} Hartree. We interpret this false positive as a local minimum of the total energy functional $E[\rho]$. Recall that only a global minimum can be viewed as a physical solution according to the theorems of Hohenberg and Kohn (see section 2.1). The possibility that $E[\rho]$ has local minima is unpleasant since there is no guarantee that a

minimization scheme will get out of it and pursue a global minimization. In the present example, we do get out of the local minimum and converge to the correct solution. We should remark that such a problem never occurred when using LOBPCG which evidences further that the convergence time is quite dependent on the parameters or numerical methods.

5.4 *Summary*

In this chapter, we presented the input file for an H_2 molecule. This allowed us to provide an overview of the method and the relevant parameters. We made additional comments on practical issues such as the optimal degree of Chebyshev polynomial. We used the computation of the density of a benzene molecule to demonstrate the necessity of high-order finite-differencing. We showed that our software can predict accurately physical quantities such as HOMO-LUMO gaps. We emphasized that the non-linearity of the Kohn-Sham equations results in an unpredictable convergence path. The convergence path is further affected by the numerics. We displayed the convergence time for many molecules in table 5.1. The timings presented are, in a sense, upper bounds since no parameter optimization was done. For example, the order of the Laplacian and the resolution may have been higher than necessary and the size of the simulation box may have been larger than required to reach milliHartree precision. For such small systems, the convergence time depends principally on the number of self-consistent steps rather than the system size indeed. This suggests that

the full potential of the method is yet to be discovered, but a parallel implementation will be required because of the onerous memory requirements due to the wavefunctions. Incidentally, the parallelism should accelerate the computations appreciably. For reasons presented above and because the number of molecules that have been simulated is relatively small, it is venturesome to compare MatRcal against other codes but we think that its efficiency matches that of LCAO codes or Gaussian.

6

Conclusion

In this thesis, we have introduced a real space method for carrying *ab initio* calculations. We reviewed the fundamental results of density functional theory and the derivation of the Kohn-Sham equations. We then presented the local density approximation which is used by our software to estimate the exchange and correlation potentials. Our method is aimed at solving the KS equations for a given atomic scale system. It is based on high-order finite-differencing. We demonstrated that it is crucial to use high-order finite-differencing in order to accurately approximate the kinetic energy term in the KS equations with a tractable number of points. Pseudopotentials projected on a Cartesian grid are utilized to account for the potential energy due to the atomic nuclei and the core electrons. The advantages of their use is two-fold. The pseudopotentials being smoother than the Coulomb potential, a smaller number of point is necessary to discretize accurately the KS Hamiltonian. Moreover, since only the valence states are sought, the number of required eigenvectors is significantly reduced. The gain is even more impressive when we consider that the eigenpairs

at the interior of the spectrum typically converge much slower than extremal eigenpairs. We presented the CFSI and LOBPCG diagonalization algorithms in some details. We remark that MatRcal does not have adjustable physical parameters (at least directly). This was exemplified in an example input file. We provided examples of physical quantities obtained with our software. The calculated HOMO-LUMO gaps of many organic and inorganic molecules agree with those obtained with Gaussian and reported in the CCCBDB. We have thus created software that confirms the validity of the real space method introduced above. The software is able to compute the electronic ground state of atomic scale systems from first principles.

Nevertheless, much improvement remains to be done. A major amelioration shall be the implementation of a parallel version of our software. We mentioned that an important advantage of the method is that it is easily parallelizable yet it is not exploited. Effectively, the matrix-vector multiplications can easily be distributed among many processors which only need to communicate during the subspace orthonormalization process. From a DFT standpoint, LDA is the crudest approximation to the exchange correlation energy. We shall implement more sophisticated functionals of the density. The first advance should be the implementation of generalized gradient approximation based functionals. As the electronic density is readily available in real space, all derivatives of the density are easily obtained and GGA functionals should be relatively easy to implement. We will allow for the possibility to use non-periodic boundary conditions in the

future. A different Poisson solver will then be required. We already verified that the biconjugate gradient stabilized algorithm is fast and accurate for our purposes.

Even considering the advancements mentioned above, the bottleneck remains the diagonalization of the KS Hamiltonian. This is partly due to the large number of points used to represent the Hamiltonian. This number is artificially swollen by the vacuum set around non-periodic atomic structures. A way to cope with that would be to use less points in the vacuum regions. Even without the use of supercells, the number of points is larger than necessary since the regions with smooth features are described with the same density of points as the regions with sharp features. In order to decrease the time spent in the diagonalization process, we must use a non-uniform and non-Cartesian grid. We think that the method presented above should remain largely unchanged under such a transformation. The main modification happens in the generation of the Laplacian which would necessitate a combination of finite-differencing and interpolation schemes. Roughly speaking, the algorithm would have to visit each grid point and get an approximation for the Laplacian using the coordinates of the surrounding points. We think that using such a grid will significantly reduce the cost of solving the KS equations. In summary, the idea is to transfer a part of the computational cost from the self-consistent field iterations to the Hamiltonian generation procedure.

BIBLIOGRAPHY

- [1] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136(3B):B864–B871, Nov 1964.
- [2] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140(4A):A1133–A1138, Nov 1965.
- [3] H. Levy and F Lessman. *Finite difference equations, by H. Levy and F. Lessman*. Pitman London, 1959.
- [4] N. Troullier and José Luriaas Martins. Efficient pseudopotentials for plane-wave calculations. *Phys. Rev. B*, 43:1993–2006, Jan 1991.
- [5] Leonard Kleinman and D. M. Bylander. Efficacious form for model pseudopotentials. *Phys. Rev. Lett.*, 48:1425–1428, May 1982.
- [6] Lin Lin, Jianfeng Lu, Lexing Ying, and Weinan E. Adaptive local basis set for kohn-sham density functional theory in a discontinuous galerkin framework i: Total energy calculation. *J. Comput. Phys.*, 231(4):2140–2154, February 2012.

- [7] Yunkai Zhou, Yousef Saad, Murilo L. Tiago, and James R. Chelikowsky. Self-consistent-field calculations using chebyshev-filtered subspace iteration. *J. Comput. Phys.*, 219:172–184, November 2006.
- [8] Andrew V. Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM J. Sci. Comput*, 23:517–541, 2001.
- [9] Yu Huang, Xiangfeng Duan, Yi Cui, Lincoln J. Lauhon, Kyoung-Ha Kim, and Charles M. Lieber. Logic gates and computation from assembled nanowire building blocks. *Science*, 294(5545):1313–1317, 2001.
- [10] Wei Lu and Charles M. Lieber. Nanoelectronics from the bottom up. *Nat Mater*, 6(11):841–850, 11 2007.
- [11] D. Mijatovic, J. C. T. Eijkel, and A. van den Berg. Technologies for nanofluidic systems: top-down vs. bottom-up-a review. *Lab Chip*, 5:492–500, 2005.
- [12] Youqi Ke, Ke Xia, and Hong Guo. Disorder scattering in magnetic tunnel junctions: Theory of nonequilibrium vertex correction. *Phys. Rev. Lett.*, 100:166805, Apr 2008.
- [13] Youqi Ke, Ferdows Zahid, V. Timoshevskii, Ke Xia, D. Gall, and Hong Guo. Resistivity of thin cu films with surface roughness. *Phys. Rev. B*, 79:155406, Apr 2009.

- [14] Cheol-Hwan Park and Steven G. Louie. Making massless dirac fermions from a patterned two-dimensional electron gas. *Nano Letters*, 9(5):1793–1797, 2009. PMID: 19338276.
- [15] F D M Haldane. ‘Luttinger liquid theory’ of one-dimensional quantum fluids. I. Properties of the Luttinger model and their extension to the general 1D interacting spinless Fermi gas. *Journal of Physics C: Solid State Physics*, 14(19):2585–2609, July 1981.
- [16] WanZhen Liang, Chandra Saravanan, Yihan Shao, Roi Baer, Alexis T. Bell, and Martin Head-Gordon. Improved Fermi operator expansion methods for fast electronic structure calculations. *The Journal of Chemical Physics*, 119(8):4117, 2003.
- [17] Lin Lin, Chao Yang, Juan C. Meza, Jianfeng Lu, Lexing Ying, and Weinan E. SelInv—An Algorithm for Selected Inversion of a Sparse Symmetric Matrix. *ACM Transactions on Mathematical Software*, 37(4):1–19, February 2011.
- [18] WanZhen Liang, Roi Baer, Chandra Saravanan, Yihan Shao, Alexis T Bell, and Martin Head-Gordon. Fast methods for resumming matrix polynomials and Chebyshev matrix polynomials. *Journal of Computational Physics*, 194(2):575–587, March 2004.

- [19] Lin Lin, Jianfeng Lu, Roberto Car, and Weinan E. Multipole representation of the Fermi operator with application to the electronic structure analysis of metallic systems. *Physical Review B*, 79(11):1–7, March 2009.
- [20] S. Goedecker and M. Teter. Tight-binding electronic-structure calculations and tight-binding molecular dynamics with localized orbitals. *Phys. Rev. B*, 51:9455–9464, Apr 1995.
- [21] Paolo Giannozzi, Giuseppe Grosso, Saverio Moron, and Giuseppe Pastor Parravicini. The ordinary and matrix continued fractions in the theoretical analysis of hermitian and relaxation operators. *Applied Numerical Mathematics*, 4(1985):273–295, 1988.
- [22] Stefan Goedecker. Linear scaling electronic structure methods. *Rev. Mod. Phys.*, 71:1085–1123, Jul 1999.
- [23] G. Kresse and J. Furthmüller. Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set. *Phys. Rev. B*, 54:11169–11186, Oct 1996.
- [24] Jeremy Taylor, Hong Guo, and Jian Wang. *Ab initio* modeling of quantum transport properties of molecular electronic devices. *Phys. Rev. B*, 63:245407, Jun 2001.

- [25] Derek Waldron, Paul Haney, Brian Larade, Allan MacDonald, and Hong Guo. Nonlinear spin current and magnetoresistance of molecular tunnel junctions. *Phys. Rev. Lett.*, 96:166804, Apr 2006.
- [26] Thomas L. Beck. Real-space mesh techniques in density-functional theory. *Rev. Mod. Phys.*, 72:1041–1080, Oct 2000.
- [27] M. Alemany, Manish Jain, Leeor Kronik, and James Chelikowsky. Real-space pseudopotential method for computing the electronic properties of periodic systems. *Physical Review B*, 69(7):1–6, February 2004.
- [28] El Briggs, Dj Sullivan, and J Bernholc. Real-space multigrid-based approach to large-scale electronic structure calculations. *Physical review. B, Condensed matter*, 54(20):14362–14375, November 1996.
- [29] James R. Chelikowsky, N. Troullier, and Y. Saad. Finite-difference-pseudopotential method: Electronic structure calculations without a basis. *Phys. Rev. Lett.*, 72:1240–1243, Feb 1994.
- [30] James R. Chelikowsky, N. Troullier, K. Wu, and Y. Saad. Higher-order finite-difference pseudopotential method: An application to diatomic molecules. *Phys. Rev. B*, 50:11355–11364, Oct 1994.
- [31] Xiaodun Jing, N. Troullier, David Dean, N. Binggeli, James R. Chelikowsky, K. Wu, and Y. Saad. *Ab initio* molecular-dynamics simulations of

si clusters using the higher-order finite-difference-pseudopotential method.

Phys. Rev. B, 50:12234–12237, Oct 1994.

- [32] Mel Levy. Universal variational functionals of electron densities, first-order density matrices, and natural spin-orbitals and solution of the v-representability problem. *Proceedings of the National Academy of Sciences*, 76(12):6062–6065, 1979.
- [33] V. Sahni and X.-Y. Pan. On the Hohenberg-Kohn and Levy-Lieb Constrained Search Proofs of Density Functional Theory. In *APS Meeting Abstracts*, page 35009, February 2012.
- [34] R.G. Parr and WeiTao Yang. *Density-Functional Theory of Atoms and Molecules*. Oxford University Press, Oxford, 1989.
- [35] Murray Gell-Mann and Keith A. Brueckner. Correlation energy of an electron gas at high density. *Phys. Rev.*, 106:364–368, Apr 1957.
- [36] D. M. Ceperley and B. J. Alder. Ground state of the electron gas by a stochastic method. *Phys. Rev. Lett.*, 45:566–569, Aug 1980.
- [37] S. H. Vosko, L. Wilk, and M. Nusair. Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis. *Canadian Journal of Physics*, 58(8):1200–1211, 1980.

- [38] J. P. Perdew and Alex Zunger. Self-interaction correction to density-functional approximations for many-electron systems. *Phys. Rev. B*, 23:5048–5079, May 1981.
- [39] Lee A. Cole and J. P. Perdew. Calculated electron affinities of the elements. *Phys. Rev. A*, 25:1265–1271, Mar 1982.
- [40] John P. Perdew and Yue Wang. Accurate and simple analytic representation of the electron-gas correlation energy. *Phys. Rev. B*, 45:13244–13249, Jun 1992.
- [41] John P. Perdew and Wang Yue. Accurate and simple density functional for the electronic exchange energy: Generalized gradient approximation. *Phys. Rev. B*, 33:8800–8802, Jun 1986.
- [42] John P. Perdew, J. A. Chevary, S. H. Vosko, Koblar A. Jackson, Mark R. Pederson, D. J. Singh, and Carlos Fiolhais. Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation. *Phys. Rev. B*, 46:6671–6687, Sep 1992.
- [43] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, Oct 1996.
- [44] Zhigang Wu and R. E. Cohen. More accurate generalized gradient approximation for solids. *Phys. Rev. B*, 73:235116, Jun 2006.

- [45] J E Pask and P A Sterne. Finite element methods in ab initio electronic structure calculations. *Modelling and Simulation in Materials Science and Engineering*, 13(3):R71, 2005.
- [46] Jack Dongarra and Francis Sullivan. Guest editors' introduction: The top 10 algorithms. *Computing in Science and Engg.*, 2:22–23, January 2000.
- [47] Yunkai Zhou, Yousef Saad, Murilo L. Tiago, and James R. Chelikowsky. Parallel self-consistent-field calculations via chebyshev-filtered subspace acceleration. 2007.
- [48] G P Kerker. Non-singular atomic pseudopotentials for solid state applications. *Journal of Physics C: Solid State Physics*, 13(9):L189, 1980.
- [49] Neil W. Ashcroft and David N. Mermin. *Solid State Physics*. Thomson Learning, Toronto, 1 edition, January 1976.
- [50] N. Macon and A. Spitzbart. Inverses of vandermonde matrices. *The American Mathematical Monthly*, 65(2):pp. 95–100, 1958.
- [51] A. Eisinberg and G. Fedele. On the inversion of the vandermonde matrix. *Applied Mathematics and Computation*, 174(2):1384 – 1397, 2006.
- [52] NanoAcademic. Nanobase, 2012.
- [53] Beresford N. Parlett. *The symmetric eigenvalue problem*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998.

- [54] James Demmel, Jack Dongarra, Axel Ruhe, and Henk van der Vorst. *Templates for the solution of algebraic eigenvalue problems: a practical guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.
- [55] Ren cang Li and Ren cang Li. Sharpness in rates of convergence for cg and symmetric lanczos methods. Technical report, 2005.
- [56] Gilbert Strang. *Introduction to Linear Algebra, Third Edition*. Wellesley Cambridge Pr, March 2003.
- [57] M. Matsen. Fast and accurate scft calculations for periodic block-copolymer morphologies using the spectral method with anderson mixing. *The European Physical Journal E: Soft Matter and Biological Physics*, 30:361–369, 2009. 10.1140/epje/i2009-10534-3.
- [58] Péter Pulay. Convergence acceleration of iterative sequences. the case of scf iteration. *Chemical Physics Letters*, 73(2):393 – 398, 1980.
- [59] Pál Császár and Péter Pulay. Geometry optimization by direct inversion in the iterative subspace. *Journal of Molecular Structure*, 114(0):31 – 34, 1984.
- [60] Jürg Hutter, Hans Peter Lüthi, and Michele Parrinello. Electronic structure optimization in plane-wave-based density functional calculations by

- direct inversion in the iterative subspace. *Computational Materials Science*, 2(2):244 – 248, 1994.
- [61] G P Srivastava. Broyden’s method for self-consistent field convergence acceleration. *Journal of Physics A: Mathematical and General*, 17(6):L317, 1984.
- [62] D. D. Johnson. Modified broyden’s method for accelerating convergence in self-consistent calculations. *Phys. Rev. B*, 38:12807–12813, Dec 1988.
- [63] A. Seidl, A. Görling, P. Vogl, J. A. Majewski, and M. Levy. Generalized kohn-sham schemes and the band-gap problem. *Phys. Rev. B*, 53:3764–3774, Feb 1996.