Density functional theory

and

nuclear quantum effects

Lin Lin

A Dissertation

Presented to the Faculty

of Princeton University

in Candidacy for the Degree

of Doctor of Philosophy

Recommended for Acceptance

by the Program in

Applied and Computational Mathematics

Advisers: Roberto Car and Weinan E

September, 2011

Abstract

This dissertation consists of two independent parts: density functional theory (Part I), and nuclear quantum effects (Part II).

Kohn-Sham density functional theory (KSDFT) is by far the most widely used electronic structure theory in condensed matter systems. The computational time of KSDFT increases rapidly with respect to the number of electrons in the system, which hinders its practical application to systems of large size. The central quantities in KSDFT are the electron density and the electron energy, which can be fully characterized by the diagonal elements and the nearest off-diagonal elements of the single particle density matrix. However, methods that are currently available require the calculation of the full density matrix. This procedure becomes highly inefficient for systems of large size. Part I of this dissertation develops a new method for solving KSDFT, which directly targets at the calculation of the diagonal and the nearest off-diagonal elements of the single particle density matrix. The new method is developed under the framework of Fermi operator expansion. The new method achieves the optimal expansion cost in the operator level. The electron density and the electron energy is then evaluated from a serires of Green's functions by a new fast algorithm developed in this dissertation. This dissertation also develops a novel method for discretizing the Hamiltonian of the system that achieves high accuracy with a very small number of basis functions. Combining all these components together, we obtain a unified, accurate, efficient method to solve KSDFT for insulating and metallic systems.

Nuclear quantum effects play an important role in a large variety of hydrogen bonded systems such as water and ice due to the small mass of protons (the nuclei of hydrogen atoms). The equilibrium proton dynamics is reflected in the quantum momentum distribution and is the focus of intense research. The standard open path integral formalism for computing the quantum momentum distribution requires the

calculation of quantum momentum distribution for one particle at a time, which is an inefficient process especially when the potential energy surface is generated from Kohn-Sham density functional theory. The information of the proton dynamics is reflected in the momentum distribution in a highly averaged way and the interpretation of the momentum distribution can involve significant amount of ambiguity. Part II of this dissertation develops the displaced path integral formalism which allows the computation of quantum momentum distribution for all particles at the same time and therefore greatly enhances the computational efficiency. Part II of this dissertation unambiguously interprets the quantum momentum distribution in two representative systems: ice Ih and high pressure ice. For ice Ih in which the potential is quasi-harmonic, this disseration clarifies the previously unclear relation between anisotropic and anharmonic effects in shaping the momentum distribution by analyzing the 3D proton momentum distribution and the associated vibrational dynamics. For high pressure ice in which the potential is strongly anharmonic and proton tunneling occurs, this dissertation assesses the important role of proton correlation effects by means of spectral decomposition of the single particle density matrix. The concerted proton tunneling process directly observed and quantified in this study reduces significantly the number of ionized configurations, and avoids the ionization catastrophe predicted by the mean field theory, which was used in previous studies to interpret the path integral simulation results.

Acknowledgements

I would like to thank my thesis advisers Professor Roberto Car and Professor Weinan E. Without their guidance and encouragement, I cannot imagine how I would overcome the problems and difficulties in my research. They have taught me how to ask questions, how to solve problems, and how to think as an applied mathematician and as a computational scientist. Their thoughts have deeply influenced me in the past four years.

I would like to thank my collaborators: Professor Carla Andreani, Professor Weiguo Gao, Dr. Juan Meza, Dr. Joseph Morrone, Professor Michele Parrinello, Dr. Antonino Pietropaolo, Dr. Amit Samanta, Dr. Roberto Senesi, and Dr. Chao Yang. Special thanks are given to Dr. Jianfeng Lu and Professor Lexing Ying for their numerous support and the fruitful work and discussions we had together. Without their help, my achievements would not be possible.

I would also like to thank the tremendous support and encouragement I received from other professors at Princeton University, especially from Professor Robert Calderbank and Professor Ingrid Daubechies.

Last but not least, my wife Dongxu Lu holds all my gratitude for her patience and for the love that she gives to me every day in my life. This thesis is dedicated to her, together with my beloved parents, my mother Xiaolan Liu and my father Chao Lin.

Contents

List of Figures

List of Tables

Part I

Density functional theory

Chapter 1

Introduction

1.1 Overview

The arrangement of the electrons characterizes the microscopic structure of molecules and systems in condensed phases in chemistry, biology, and material science. The arrangement of the electrons is described by the electronic structure theory. The various forms of the electronic structure theory differ by orders of magnitude from each other in terms of accuracy and efficiency. Among the different formalisms of the electronic structure theory, Kohn-Sham density functional theory (KSDFT) achieves the best compromise between accuracy and efficiency, and is by far the most widely used electronic structure theory. Nonetheless, the computational cost of KSDFT still increases rapidly with respect to the number of electrons in the system, which hinders the application of KSDFT to systems of large size. Reducing the computational cost of KSDFT requires combined knowledge of mathematics, physics and computer science. Part I of this dissertation explores the mathematical properties of KSDFT, and develops an accurate and efficient algorithm for applying KSDFT to systems of large scale.

The scale of the system, *i.e.* the number of the electrons is a crucial parameter

in many applications in chemistry, biology, and material science. It is desirable to have the same number of electrons in the numerical simulation as that in the real system. However, even a water droplet contains more than 10^{20} electrons. This overwhelmingly large magnitude is out of the scope of any of the existing simulation technique, and samples of smaller size has to be used in practice. The small sample size introduces a systematic error to the numerical simulation, called the size effect, which is non-negligible in many applications. For example, the diffusion constant of a polymer chain is underestimated by $2 \sim 3$ times due to the size effect [79]; Crack propagation [1] involves tens of thousands to millions of atoms and electrons by its nature, and cannot be observed with samples of smaller sizes. Systems with millions of electrons are usually treated by classical mechanics with empirical potential. The empirical potential energy surfaces have achieved success in describing a large class of phenomena provided that the empirical parameters are carefully optimized. On the other hand, the potential energy surfaces directly generated from quantum mechanics, such as from KSDFT, have the advantage that the computational result depends only on a small number of universal parameters, including atomic species, atomic positions, and a few parameters in the energy functional which do not depend on the specific setup of the system. As a result, simulations with potential energy surfaces generated from KSDFT are called "first principle" simulations, and are capable of treating bond-forming, bond-breaking, cracking, and other complicated chemical and mechanical processes without extra tuning of the parameters. It is thus desirable to directly apply KSDFT to study systems consisting millions of electrons, but this is far beyond the current capability of the KSDFT. In the standard methods for solving KSDFT, the computational cost increases as $\mathcal{O}(N^3)$ where N is the number of electrons in the system. Although the standard algorithm has been highly optimized in the past two decades to reduce the computational cost in practice [94,145,182,256], the cubic scaling still limits the application of KSDFT to systems with at most tens

of thousands of electrons.

Various efforts have been devoted to reduce the cubic scaling of KSDFT in the past two decades. The major breakthrough is achieved by the algorithms with linear scaling, *i.e.* $\mathcal{O}(N)$ scaling [88, 89, 91, 98, 99, 101, 102, 154, 184, 193, 229, 252]. Such linear scaling algorithms relies on the "nearsightedness" property of the electrons [142, 212], which means that the density perturbation induced by a local change in the external potential dies off exponentially with respect to the distance from the place where the perturbation was applied. According to DFT the ground state energy is a functional of the density, then the effect of a local perturbation on the density is also local because of nearsightedness, and the energy should not have a very long-range dependence on the density. The nearsightedness property allows one to divide the entire system into many pieces. The size of each piece has fixed size, and the total number of pieces is proportional to the number of electrons in the entire system. The computational cost for solving KSDFT in each piece is fixed, and the total computational cost is therefore proportional to the number of electrons in the entire system. Therefore, $\mathcal{O}(N)$ scaling is achieved.

The nearsightedness property is not valid for all systems, but is only valid for a class of materials called insulating systems, including sulfur, glass, paper, large organic molecules such as DNA and protein, most of the common salts and oxides, to name a few. The nearsightedness property is violated in metallic systems, namely the density perturbation induced by a local change in the external potential exhibits algebraic and oscillatory decay (called the Friedel oscillation [87]) with respect to the distance from the place where the perturbation was applied. Two thirds of the elements in the periodic table can directly form metallic systems, such as aluminum, lithium, copper, and iron. Non-metallic elements can also form metallic systems, *e.g.* graphene and carbon nanotube which only consists of carbon. Due to the violation of the nearsightedness property, the $\mathcal{O}(N)$ scaling algorithm is not directly

applicable to metallic systems. The nearsightedness property can be recovered by introducing an artificial finite temperature in the system, and the decay rate of the density perturbation induced by a local change in the external potential becomes exponential again with the exponent depending on the artificial temperature [100, 140].

However, it is not easy to take into account the nearsightedness in practical algorithms. First, the nearsightedness is not so precisely defined in practice, and particularly the nearsightedness decay length is difficult to be predicted in advance. Second, although the accuracy of the linear scaling methods can be systematically improved by increasing the size of each piece (usually characterized by a truncation radius R), the truncation radius R can still be quite large if high numerical accuracy is to be achieved. This is especially the case for metallic system where the truncation radius depends explicitly on the artificial temperature. The magnitude of the artificial temperature should be chosen carefully in order to balance the accuracy and the computational cost. Third, it is not a trivial step to implement the nearsightedness if the nearsightedness is imposed as a constraint on the Kohn-Sham orbitals or the Kohn-Sham single particle density matrix. As a result, the $\mathcal{O}(N^3)$ scaling methods are still the best uniform solution for applying KSDFT to insulating systems and to metallic systems. The important question remains open: whether it is possible to improve the $\mathcal{O}(N^3)$ scaling uniformly for all the systems without encountering the difficulties in nearsightedness algorithms?

This open question is positively answered in Part I of this dissertation. Instead of using the nearsightedness property, this dissertation focuses on the mathematical properties of KSDFT that are uniformly valid for insulating systems and metallic systems, at low and at high temperature. As a result, this dissertation develops algorithms with improved computational scaling over $\mathcal{O}(N^3)$ scaling methods for all systems. To be more specific, the amount of improvement depends on the dimension of the system. The computational cost of the present algorithm is $\mathcal{O}(N)$ for one

dimensional systems, $\mathcal{O}(N^{1.5})$ for two-dimensional systems, and $\mathcal{O}(N^2)$ for three-dimensional systems. Furthermore, the present algorithm can be combined with the nearsightedness property, and achieves $\mathcal{O}(N)$ scaling at all dimensions for insulating systems and for metallic system at high temperature.

This chapter provides the minimum amount of prerequisite knowledge for Part I of this dissertation. The rest of this chapter is organized as follows: Section 1.2 briefly introduces the quantum many body problem, and the electronic structure problem with the Bohr-Oppenheimer approximation, followed by Section 1.3 for the basic components of the Kohn-Sham density functional theory. The pseudopotential framework for KSDFT is introduced in Section 1.4. In Section 1.5 the mathematical properties of the KSDFT which are essential for the new method developed in Part I of this dissertation are discussed. Section 1.6 reviews the existing methods and the most widely used software packages for solving KSDFT. Finally Section 1.7 outlines the various components of the new method that will be discussed in detail in the rest of the Chapters in Part I.

1.2 Quantum many body problem and electronic structure

The microscopic properties of electrons in chemistry, biology and material science are accurately described by the many body Hamiltonian of the Schrödinger equation. The many body Hamiltonian associated with a system with N_{nuc} atoms and N electrons is

$$H = \sum_{I=1}^{N_{\text{nuc}}} \frac{P_I^2}{2M_I} + \sum_{i=1}^{N} \frac{p_i^2}{2} + V(R_1, \ldots, R_{N_{\text{nuc}}}, x_1, \ldots, x_N). \tag{1.1}$$

Atomic units are used throughout this dissertation. Namely, without further specification, the unit of energy is Hartree, the unit of length is Bohr, the unit of mass is the

electron mass m_e, the unit of charge is the electron charge e, and the Planck constant $\hbar$ equals to 1. Moreover, M_I is the mass of the I-th nucleus, R_I is the position of the I-th nucleus, and x_i is the position of the i-th electron. The momentum operator of the nucleus and the electron are denoted by P_I, p_i as $p_i = -i\nabla_{x_i}$, $P_I = -i\nabla_{R_I}$. Spin is neglected at the moment. V is the interaction energy between the nuclei and the electrons, given by

$$V(R_1, \ldots, R_{N_{\text{nuc}}}, x_1, \ldots, x_N) = \frac{1}{2}\sum_{I \neq J} \frac{Z_I Z_J}{|R_I - R_J|} + \frac{1}{2}\sum_{i \neq j} \frac{1}{|x_i - x_j|} - \sum_{i,I} \frac{Z_I}{|x_i - R_I|}. \tag{1.2}$$

The charge of the I-th nucleus is Z_I. The three terms in Eq. (1.2) represent the Coulomb interactions among nuclei-nuclei, electron-electron and nuclei-electron, respectively.

The many body Hamiltonian (1.1) contains all the information of systems, but all the information does not have equal importance in practice. In many cases, the important information is only contained in the most stable state of the system. This most stable state is characterized by the *ground state* of the many body Hamiltonian, *i.e.* the smallest eigenvalue and eigenvector of the many body Hamiltonian.

$$H\Psi(R_1, \ldots, R_{N_{\text{nuc}}}; x_1, \ldots, x_N) = E\Psi(R_1, \ldots, R_{N_{\text{nuc}}}; x_1, \ldots, x_N). \tag{1.3}$$

Eq. (1.3) is referred to as the *quantum many body problem*. E is called the ground state energy of the many body system, and Ψ is called the ground state wavefunction. Ψ should satisfy certain symmetry condition determined by the statistics of both electrons and nuclei. Especially, Ψ is an antisymmetric function with respect to the coordinates for the electrons $(x_1, \cdots, x_N)$. Ψ changes sign if any pair of coordinates

x_i and x_j are interchanged:

$$\Psi(R_1, \ldots, R_{N_{\rm nuc}}; x_1, \ldots, x_i, \ldots, x_j, \ldots, x_N) \\ = -\Psi(R_1, \ldots, R_{N_{\rm nuc}}; x_1, \ldots, x_j, \ldots, x_i, \ldots, x_N), \quad 1 \leq i < j \leq N. \quad (1.4)$$

The many body problem (1.3) can be analytically solved for a hydrogen atom which contains only one nucleus and one electron. The solution of (1.3) is already much more complicated in an H_2 molecule which contains two electrons and two nuclei. The solution becomes generally intractable for more than 20 particles even with the help of numerical methods and increasingly more powerful computers. The reason for this enormous complexity is that the dimension of the ground state wavefunction is $3(N_{\rm nuc} + N)$. Even if each spatial coordinate is discretized by 10 points, $10^{3(N_{\rm nuc}+N)}$ real numbers will be immediately required just to record one state of the system. Although the quantum many body problem is in principle an exact theory, it exhibits exponential complexity and is intractable without further theoretical approximation. The enormous complexity of the quantum many body problem was well summarized by Dirac in 1929 [73]: "The fundamental laws necessary to the mathematical treatment of large parts of physics and the whole of chemistry are thus fully known, and the difficult lies only in the fact that application of these laws leads to equations that are too complex to be solved."

The first step to reduce the complexity of the quantum many body problem is the Born-Oppenheimer approximation [38], which separates the complexity due to the electrons and that due to the nuclei. The mass of the electron is more than a thousand times smaller than the mass of the nuclei of the lightest element in the periodic table, *i.e.* hydrogen. The Born-Oppenheimer approximation recognizes that the electrons should therefore move much faster than the nuclei, and the state of the electrons is "slaved" to the motion of nuclei. More specifically, for fixed nuclei

positions $(R_1, \ldots, R_{N_{\text{nuc}}})$, the state of the electrons is described by the ground state of the many body Hamiltonian of the electrons:

$$H_e = -\frac{1}{2}\sum_i \Delta_{x_i} + \sum_i V_{\text{ext}}(x_i) + V_{ee}(x_1, \ldots, x_N). \tag{1.5}$$

The nuclei-electron interaction V_{ext} and the electron-electron interaction V_{ee} are defined as

$$V_{\text{ext}}(x) = -\sum_I \frac{Z_I}{|x - R_I|}, \quad V_{ee}(x_1, \ldots, x_N) = \frac{1}{2}\sum_{i \neq j} \frac{1}{|x_i - x_j|}. \tag{1.6}$$

Compared to Eq. (1.2), the nuclei-nuclei interaction is excluded from H_e, since the nuclei-nuclei interaction

$$V_{nn}(R_1, \ldots, R_{N_{\text{nuc}}}) = \frac{1}{2}\sum_{I \neq J} \frac{Z_I Z_J}{|R_I - R_J|}, \tag{1.7}$$

is a constant term for fixed nuclei positions $(R_1, \ldots, R_{N_{\text{nuc}}})$.

The ground state of the many body Hamiltonian of the electrons is defined by

$$H_e \Psi_e(r_1, \ldots, r_N; R_1, \ldots, R_{N_{\text{nuc}}}) = E_e(R_1, \ldots, R_{N_{\text{nuc}}}) \Psi_e(r_1, \ldots, r_N; R_1, \ldots, R_{N_{\text{nuc}}}), \tag{1.8}$$

E_e is called the ground state energy of the electrons. Ψ_e is called the ground state wavefunction of the electrons, and is an antisymmetric function.

The ground state energy $E_e(R_1, \ldots, R_{N_{\text{nuc}}})$ has important physical meaning. The ground state energy together with the nuclei-nuclei interaction V_{nn} form the effective inter-atomic potential $V_{\text{eff}}(R_1, \ldots, R_{N_{\text{nuc}}}) = E_e(R_1, \ldots, R_{N_{\text{nuc}}}) + V_{nn}(R_1, \ldots, R_{N_{\text{nuc}}})$. This inter-atomic potential is completely determined by the atomic species and the atomic positions, and has great advantage over the classical inter-atomic potential characterized by empirical parameters. Therefore, the ground state energy E_e

carries most of the information of the arrangements of the electrons. Calculating $E_e(R_1, \ldots, R_{N_{\text{nuc}}})$ from fixed nuclei positions $(R_1, \ldots, R_{N_{\text{nuc}}})$ is called the *electronic structure problem*.

After solving the electronic structure problem, the motions of the nuclei can be approximated by classical mechanics

$$M_I \ddot{R}_i = -\frac{\partial V_{\text{eff}}(R_1, \ldots, R_{N_{\text{nuc}}})}{\partial R}. \tag{1.9}$$

The approximation (1.9) can be improved by more sophisticated techniques such as path integrals formulation [83], which will be discussed in detail in Part II of this dissertation.

From now on we focus on the electronic structure problem, and we drop the subscript e in the ground state energy E_e and in the ground state wavefunction Ψ_e without causing ambiguity.

1.3 Kohn-Sham Density functional theory (KSDFT)

Born-Oppenheimer approximation reduces the quantum many body problem (1.3) to the electronic structure problem. The electronic structure problem still exhibits exponential complexity with respect to the number of electrons N, and it is necessary to make further approximations. Compared to the common acceptance of the Born-Oppenheimer approximation, less agreement is achieved on the approximation of the electronic structure problem. Various electronic structure theories with different accuracy and efficiency have been proposed, including Hartree-Fock [152, 159], configuration interaction [237], coupled cluster [63], Møller-Plesset perturbation theory [187], quantum Monte Carlo [51, 85], and density functional theory [125, 143], to name a few. After decades of development, density functional theory is commonly found to achieve the best compromise between accuracy and efficiency, and has be-

come the most widely used electronic structure theory.

The foundation of the density functional theory is the Hohenberg-Kohn theorem [125]. Hohenberg-Kohn theorem proves that the ground state electron energy E is uniquely determined by the electron density ρ up to a shift of a trivial constant. This dependence is denoted by the density functional $E[\rho]$. Given the N-body wave function Ψ, the electron density is defined as

$$\rho(x) = N \int |\Psi(x, x_2, \ldots, x_N)|^2 \, \mathrm{d}x_2 \cdots \mathrm{d}x_N, \tag{1.10}$$

$\rho(x)$ represents the probability of finding any of the N electrons at point x. The electron density $\rho(x)$ is a function of three coordinates rather than $3N$ coordinates. Therefore density functional theory remarkably reduces the complexity of the electronic structure problem.

If the exact form of the energy functional $E[\rho]$ is known, the ground state energy can be readily obtained by a minimization procedure over a three-dimensional function ρ with respect to the energy functional $E[\rho]$. However, Hohenberg-Kohn theorem only claims the existence of such energy functional without predicting the full detail of its actual form. Furthermore, the energy functional carefully chosen for one system can fail drastically for another system in principle.

The ground-breaking work is provided by Kohn and Sham [143]. Kohn and Sham approximated the energy functional of interacting electrons by an energy functional of non-interacting electrons together with a correction term. The energy functional of non-interacting electrons can be written analytically and contributes to most part of the ground state energy. The remaining correction term, which is called exchange-correlation functional, remains unknown but is relatively easy to be approximated roughly.

The Kohn-Sham density functional theory can be *formally* written down as fol-

lows [143]. The rigorous derivation, however, should follow the Levy-Lieb approach of constrained minimization [158]. First, the ground state energy of an interacting inhomogeneous system can be written as [125]

$$E[\rho] = \int V_{\text{ext}}(x)\rho(x)\,\mathrm{d}x + \frac{1}{2}\iint \frac{\rho(x)\rho(y)}{|x-y|}\,\mathrm{d}x\,\mathrm{d}y + G[\rho], \tag{1.11}$$

where the first term (V_{ext}) characterizes the nuclei-electron interaction and the second term gives the electron-electron interaction. $G[\rho]$ is a universal functional of the electron density. The Kohn-Sham density functional theory then approximates $G[\rho]$ as

$$G[\rho] \equiv E_{\text{K}}[\rho] + E_{\text{xc}}[\rho], \tag{1.12}$$

where $E_{\text{K}}[\rho]$ is the kinetic energy of N non-interacting electrons. $E_{\text{xc}}[\rho]$ is defined to be the exchange-correlation energy, which takes into account all the remaining ground state energies that are not represented by the previous terms. The many body wavefunction of N non-interacting electrons takes the form of the Slater determinant

$$\Psi(x_1, \cdots, x_N) = \frac{1}{\sqrt{N!}} \det \begin{pmatrix} \psi_1(x_1) & \cdots & \psi_1(x_N) \\ \vdots & \ddots & \vdots \\ \psi_N(x_1) & \cdots & \psi_N(x_N) \end{pmatrix}, \tag{1.13}$$

where the three-dimensional wavefunctions ψ_i are called the *electron orbitals*. The electron orbitals satisfy the orthonormal condition

$$\int \psi_i(x)^* \psi_j(x)\,\mathrm{d}x = \delta_{ij}. \tag{1.14}$$

The electron density is reconstructed from the electron orbitals according to the relation

$$\rho(x) = \sum_{i=1}^{N} |\psi_i(x)|^2. \tag{1.15}$$

The kinetic energy for the non-interacting electrons is

$$E_{\mathrm{K}}\left[\{\psi_i\}_{i=1}^{N}\right] = \frac{1}{2}\sum_{i=1}^{N}\int |\nabla\psi_i|^2\,\mathrm{d}x. \tag{1.16}$$

As a result, the Kohn-Sham energy functional is given by

$$E_{\mathrm{KS}}\left[\{\psi_i\}\right] = E_{\mathrm{K}}\left[\{\psi_i\}\right] + \int V_{\mathrm{ext}}(x)\rho(x)\,\mathrm{d}x + \frac{1}{2}\iint \frac{\rho(x)\rho(y)}{|x-y|}\,\mathrm{d}x\,\mathrm{d}y + E_{\mathrm{xc}}[\rho]. \tag{1.17}$$

To find the ground state energy, the energy functional (1.17) should be optimized over all possible electron orbitals $\{\psi_i\}_{i=1}^{N}$, and hence over all possible electron density ρ satisfying $\int \rho(x)\,\mathrm{d}x = N$. On the other hand, if ρ minimizes Kohn-Sham energy (1.17), the corresponding electron orbitals are also determined by the minimization procedure. Therefore the ground state energy of the Kohn-Sham density functional theory depends only on the electron density ρ. This issue will become clearer in Section 1.5.

The choice of the exchange-correlation functional remains unknown. Fortunately, it turned out that even the crudest approximation of exchange-correlation functional, namely the local density approximation (LDA) [52, 206] is often surprisingly accurate for systems with slowly varying charge densities. For example, the bond lengths and bond angles can be predicted by LDA within a few percent for many systems. More sophisticated exchange-correlation functionals such as generalized gradient approximation (GGA) [22, 149, 204], and hybrid exchange-correlation functionals [23, 205] further extend the applicability of KSDFT to a large class of molecules and systems in condensed phase. Without much loss of generality, in the following we will use the LDA form for exchange-correlation functional, *i.e.*

$$E_{\mathrm{xc}}[\rho] = \int \epsilon_{\mathrm{xc}}[\rho(x)]\,\mathrm{d}x. \tag{1.18}$$

To sum up, the Kohn-Sham density functional theory solves the following minimization problem over the electron orbitals $\{\psi_i\}$.

$$\begin{aligned} E_{\mathrm{KS}} = \min_{\{\psi_i\}} & \frac{1}{2}\sum_{i=1}^{N}\int |\nabla\psi_i|^2\,\mathrm{d}x + \int V_{\mathrm{ext}}(x)\rho(x)\,\mathrm{d}x + \frac{1}{2}\iint \frac{\rho(x)\rho(y)}{|x-y|}\,\mathrm{d}x\,\mathrm{d}y + \int \epsilon_{\mathrm{xc}}[\rho(x)]\,\mathrm{d}x, \\ s.t. \quad & \int \psi_i(x)\psi_j(x)\,\mathrm{d}x = \delta_{ij}, \quad i,j = 1,\ldots,N, \\ & \rho(x) = \sum_{i=1}^{N}|\psi_i(x)|^2. \end{aligned} \tag{1.19}$$

Here $\rho(x)$ is a function of $\{\psi_i\}$, and the minimization problem (1.19) is a nonlinear optimization problem. Eq. (1.19) can be solved directly using nonlinear optimization techniques [6, 25, 145, 207, 244, 250]. However, in practice it is more popular to solve the Euler-Lagrange equation associated to (1.19), which is called the self-consistent iteration. The self-consistent iteration method is used in this dissertation, and the basic procedure is summarized as follows.

The Euler-Lagrange equation corresponding to the minimization problem (1.19) is

$$\begin{aligned} & \left(-\frac{1}{2}\Delta + V_{\mathrm{eff}}[\rho]\right)\psi_i = \sum_{j=1}^{N}\psi_j\lambda_{ji}, \\ s.t. \quad & \int \psi_i(x)\psi_j(x)\,\mathrm{d}x = \delta_{ij}, \quad i,j = 1,\ldots,N, \\ & \rho(x) = \sum_{i=1}^{N}|\psi_i(x)|^2. \end{aligned} \tag{1.20}$$

We denote the effective potential by $V_{\mathrm{eff}}[\rho]$:

$$V_{\mathrm{eff}}[\rho](x) = V_{\mathrm{ext}}(x) + \int \frac{\rho(y)}{|x-y|}\,\mathrm{d}y + \epsilon'_{\mathrm{xc}}[\rho(x)]. \tag{1.21}$$

$\{\lambda_{ji}\}$ are the Lagrange multipliers corresponding to the orthonormal constraints of

the electron orbitals. Eq. (1.20) is invariant under unitary transformations of the electron orbitals. As a result Eq. (1.20) can be simplified as

$$\begin{aligned} &\left(-\frac{1}{2}\Delta + V_{\text{eff}}[\rho]\right)\psi_i = \psi_i \epsilon_i, \\ s.t. \quad &\int \psi_i(x)\psi_j(x)\,\mathrm{d}x = \delta_{ij}, \quad i,j = 1,\ldots,N, \\ &\rho(x) = \sum_{i=1}^{N} |\psi_i(x)|^2. \end{aligned} \tag{1.22}$$

In order to minimize (1.19) only the lowest N eigenvalues and eigenvectors are to be computed. The lowest N eigenvalues $\{\epsilon_i\}_{i=1}^N$ are called the occupied Kohn-Sham eigenvalues, and the corresponding lowest N eigenvectors $\{\psi_i\}_{i=1}^N$ are called the occupied Kohn-Sham orbitals. The minimization problem (1.19) is nonlinear, and as a result the eigenvalue problem (1.22) is a nonlinear eigenvalue problem.

The Euler-Lagrange equation (1.22) can be solved by fixing the electron density $\rho = \rho_{\text{in}}$ in the potential energy term $V_{\text{eff}}[\rho]$. Then the Kohn-Sham Hamiltonian $H = -\frac{1}{2}\Delta + V_{\text{eff}}[\rho]$ is a fixed linear operator. The corresponding lowest N eigenvalues and eigenvectors can therefore be computed by a standard linear eigenvalue procedure such as ARPACK [151]. The consequence of fixing the electron density in the Kohn-Sham Hamiltonian is that the output electron density ρ_{out} given by Eq. (1.15) does not necessarily match the input electron density ρ_{in}. In such case, a new density ρ is generated based on ρ_{in} and ρ_{out}. This new density ρ is used as the new input density for the eigenvalue problem (1.22). This procedure is repeated until $\rho_{\text{in}} = \rho_{\text{out}}$. Since the self-consistent electron density is obtained iteratively, this procedure is called the self-consistent iteration.

When the self-consistent electron density ρ is obtained, the ground state electron energy can be calculated from the Kohn-Sham energies ϵ_i and the electron density ρ

according to:

$$E_{KS} = \sum_{i=1}^{N} \epsilon_i - \frac{1}{2} \iint \frac{\rho(x)\rho(y)}{|x-y|} \, \mathrm{d}x \, \mathrm{d}y + \int \epsilon_{\mathrm{xc}}[\rho(x)] \, \mathrm{d}x - \int \epsilon'_{\mathrm{xc}}[\rho(x)]\rho(x) \, \mathrm{d}x. \quad (1.23)$$

1.4 KSDFT: pseudopotential framework

So far KSDFT is introduced as an all electron theory. Namely all the electrons are taken into account in the calculation. In quantum chemistry, the electrons are divided into two categories: core electrons and valence electrons. For most of the systems, only the valence electrons participate in the interstitial bonding region and in the chemical reactions, and the core electrons do not participate in the chemical reactions. However, the electron orbitals of the core electrons are highly non-smooth and the treatment of the core electrons requires a large number of basis functions per atom or a fine mesh for numerical integration. Therefore it is desirable to remove the core electrons and represent the core electrons effectively in the potential energy surface. This idea is achieved in the *pseudopotential framework* [241, 245]. The pseudopotential framework only involves valence electrons. The number of basis functions per atom to describe the valence electrons is also much smaller than that in the all electron framework, and there is no singularity in the electronic wavefunctions. Pseudopotential framework will be used throughout this dissertation to describe the Kohn-Sham Hamiltonian unless otherwise specified. We remark that similar results can be achieved by projected augmented wavefunctions method (PAW) [34]. The extensions to other frameworks such as PAW and the all-electron framework will be the work in future.

In the past three decades, a vast number of types of pseudopotentials have been developed. The most widely used pseudopotentials include the norm-conserving pseudopotential [120, 241], the dual-space pseudopotential [104, 121] and the ultrasoft pseu-

dopotential [245]. For a more detailed discussion and comparison of the pseudopotential theories, we refer the readers to the review article [59]. The Troullier-Martins pseudopotential [241] is one of the most popular norm-conserving pseudopotential. In what follows, the Troullier-Martins pseudopotential will be used for illustration purpose.

The Kohn-Sham energy functional in the Troullier-Martins pseudopotential framework is given by:

$$E_{\mathcal{K}}(\{\psi_i\}) = \frac{1}{2}\sum_{i=1}^{N}\int |\nabla\psi_i(x)|^2\,\mathrm{d}x + \int V_{\mathrm{ext}}(x)\rho(x)\,\mathrm{d}x + \sum_{\ell}\gamma_\ell\sum_{i=1}^{N}|\int b_\ell^*(x)\psi_i(x)\,\mathrm{d}x|^2 + \frac{1}{2}\iint\frac{\rho(x)\rho(y)}{|x-y|}\,\mathrm{d}x\,\mathrm{d}y + \int \epsilon_{\mathrm{xc}}[\rho(x)]\,\mathrm{d}x, \quad (1.24)$$

In (1.24), we have taken the Kleinman-Bylander form of the pseudopotential [138]. For each ℓ, b_ℓ is a function supported locally in the real space around the position of one of the atoms, and $\gamma_\ell = \pm 1$.

The Kohn-Sham equation, or the Euler-Lagrange equation associated with (1.24) reads

$$H_{\mathrm{eff}}[\rho]\psi_i = (-\tfrac{1}{2}\Delta + V_{\mathrm{eff}}[\rho] + \sum_{\ell}\gamma_\ell|b_\ell\rangle\langle b_\ell|)\psi_i = \epsilon_i\psi_i, \quad (1.25)$$

where the effective one-body potential V_{eff} is given by

$$V_{\mathrm{eff}}[\rho](x) = V_{\mathrm{ext}}(x) + \int\frac{\rho(y)}{|x-y|}\,\mathrm{d}y + \epsilon'_{\mathrm{xc}}[\rho(x)]. \quad (1.26)$$

After obtaining the self-consistent electron density, the total energy of the system can be expressed using the eigenvalues $\{\epsilon_i\}$ and ρ

$$E_{\mathrm{KS}} = \sum_{i=1}^{N}\epsilon_i - \frac{1}{2}\iint\frac{\rho(x)\rho(y)}{|x-y|}\,\mathrm{d}x\,\mathrm{d}y + \int\epsilon_{\mathrm{xc}}[\rho(x)]\,\mathrm{d}x - \int\epsilon'_{\mathrm{xc}}[\rho(x)]\rho(x)\,\mathrm{d}x. \quad (1.27)$$

In each step of the self-consistent iteration, we find $\widetilde{\rho}$ from a given effective po-

tential V_{eff}

$$\widetilde{\rho}(x) = \sum_{i=1}^{N} |\psi_i|^2(x), \tag{1.28}$$

where the $\{\psi_i\}$'s are the first N eigenfunctions of H_{eff}.

$$H_{\text{eff}}\psi_i = (-\tfrac{1}{2}\Delta + V_{\text{eff}} + \sum_{\ell} \gamma_\ell |b_\ell\rangle\langle b_\ell|)\psi_i = \epsilon_i \psi_i. \tag{1.29}$$

The $\{\psi_i\}$'s also minimize the variational problem

$$E_{\text{eff}}(\{\psi_i\}) = \frac{1}{2}\sum_{i=1}^{N} \int |\nabla \psi_i(x)|^2 \, \mathrm{d}x + \int V_{\text{eff}}(x)\rho(x) \, \mathrm{d}x + \sum_{\ell} \gamma_\ell \sum_{i=1}^{N} |\langle b_\ell, \psi_i\rangle|^2, \tag{1.30}$$

with the orthonormality constraints $\langle \psi_i, \psi_j \rangle = \delta_{ij}$.

1.5 Mathematical properties of KSDFT

In the self-consistent iteration framework for KSDFT, there are two major steps:

1. Given an input electron density ρ_{in}, calculate the output electron density ρ_{out}. This is done by solving the Kohn-Sham energies $\{\epsilon_i\}$ and the Kohn-Sham electron orbitals $\{\psi_i\}$ of $H[\rho_{\text{in}}]$.

2. Form a new input electron density from ρ_{in} and ρ_{out}.

In the standard method for solving KSDFT, the complexity of step 1 is $\mathcal{O}(N^3)$, and the complexity of step 2 is $\mathcal{O}(N)$. Therefore step 1 dominates the computational cost for solving KSDFT, and is the major bottleneck in order to reduce the complexity.

Step 1 essentially defines a map from ρ_{in} to ρ_{out}, which is referred to as the *Kohn-Sham map*. Step 1 involves a diagonalization process of $H[\rho_{\text{in}}]$ which is a nonlinear process. Therefore the Kohn-Sham map is a nonlinear map. Kohn-Sham map contains all the information of step 1. The mathematical properties of the

Kohn-Sham map are essential in order to achieve an accurate and efficient method for solving KSDFT.

In order to study the mathematical properties of the Kohn-Sham map, it is desirable to have the explicit form of the Kohn-Sham map, rather than the implicit form as defined in step 1. The explicit form of the Kohn-Sham map is as follows. For simplicity we assume the temperature is zero. The Hamiltonian matrix is denoted by $H \equiv H[\rho_{\text{in}}]$ which is discretized into a $N_t \times N_t$ matrix. $\{\epsilon_i\}_{i=1}^{N_t}$ and $\{\psi_i\}_{i=1}^{N_t}$ are all the eigenvalues and eigenvectors of the Hamiltonian matrix H. The output electron density can be rewritten in an alternative form:

$$\begin{aligned}\rho_{\text{out}}(x) &= \sum_{i=1}^{N} |\psi_i(x)|^2 \\ &= \begin{pmatrix} \psi_1(x) & \cdots & \psi_{N_t}(x) \end{pmatrix} \begin{pmatrix} \chi(\varepsilon_1 - \mu) & & \\ & \ddots & \\ & & \chi(\varepsilon_{N_t} - \mu) \end{pmatrix} \begin{pmatrix} \psi_1(x) \\ \vdots \\ \psi_{N_t}(x) \end{pmatrix}\end{aligned} \tag{1.31}$$

Here $\chi(x)$ is the Heaviside function that satisfies

$$\chi(x) = \begin{cases} 1, & x \leq 0, \\ 0, & x > 0. \end{cases} \tag{1.32}$$

μ is called the chemical potential. For a discretized system, μ is chosen to be in the range $(\epsilon_N, \epsilon_{N+1})$ as long as $\epsilon_{N+1} > \epsilon_N$. Eq. (1.31) can be written in a more compact form using the notation of matrix function:

$$\rho_{\text{out}}(x) = \left[\chi\left(H[\rho_{\text{in}}] - \mu I\right)\right]_{x,x} \equiv \text{diag}\ \ \chi(H[\rho_{\text{in}}] - \mu I). \tag{1.33}$$

Here χ is a matrix function and I is the identity matrix of size $N_t \times N_t$. Eq. (1.33) clearly shows that the Kohn-Sham map is nothing but the diagonal elements of the

matrix Heaviside function $\chi(H[\rho_{\text{in}}] - \mu I)$.

However, the value of the Heaviside function χ is either 0 or 1 on the spectrum of the Hamiltonian matrix. The Heaviside function is a not a smooth function, and the matrix Heaviside function is not well-defined for all systems. Here the important characteristic quantity is $\epsilon_{N+1} - \epsilon_N$, which is referred to as the *energy gap* of the system. It can be shown that as the number of electrons $N \to \infty$, the energy gap is always finite for insulating systems, and becomes zero for metallic systems [12]. As a result, the matrix Heaviside function is only well defined for insulating system, and is ill-defined for metallic systems.

The flaw of the matrix Heaviside function can be amended by a more generalized function called the matrix Fermi-Dirac function, which takes into account the finite temperature effect [186]

$$\rho = \text{diag } \frac{1}{1 + \exp(\beta(H - \mu))} \equiv \text{diag } f(H) \tag{1.34}$$

Fermi-Dirac function is closely related to the Heaviside function: If β is finite, the Fermi-Dirac function is a smooth function across the spectrum of the Hamiltonian H, and is well-defined regardless of the value of the energy gap. When $\beta \to \infty$, Fermi-Dirac function converges to the Heaviside function (see Fig. 1.1). The physical meaning of β is the inverse of the temperature of the system, and $\beta \to \infty$ implies that the temperature is zero. Therefore the matrix Heaviside function is also called the *zero temperature limit of the Fermi-Dirac function*.

The ground state energy E_{KS} can be written in terms of $f(H)$ as well. For insulating systems, we have

$$\sum_{i=1}^{N} \epsilon_i = \text{Tr}\left[H\chi(H - \mu I)\right], \tag{1.35}$$

and this relation can be directly generalized to both insulating systems and metallic systems as $\text{Tr}\left[Hf(H)\right]$. Thus, the matrix function $f(H)$ is of central importance

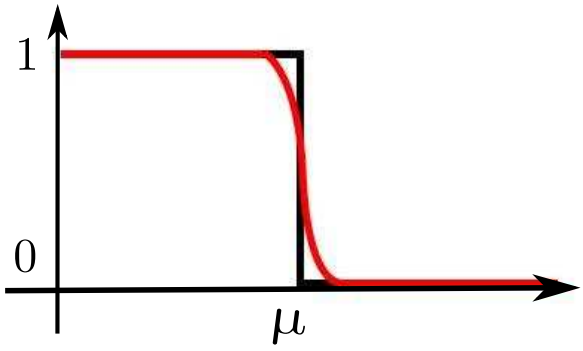


Figure 1.1: Heaviside function (black line) and Fermi-Dirac function at finite temperature (red line).

in KSDFT, and is referred to as the *density matrix* of the system. KSDFT can be written explicitly only using density matrix:

$$\rho = \text{diag}\ \ f(H[\rho]), \tag{1.36}$$

$$E_{KS} = \text{Tr}\left[H[\rho]f(H[\rho])\right] - \frac{1}{2}\iint \frac{\rho(x)\rho(y)}{|x-y|}\,\mathrm{d}x\,\mathrm{d}y + \int \epsilon_{\text{xc}}[\rho(x)]\,\mathrm{d}x - \int \epsilon'_{\text{xc}}[\rho(x)]\rho(x)\,\mathrm{d}x. \tag{1.37}$$

The density matrix $f(H)$ is a $N_t \times N_t$ matrix. Eq. (1.37) shows the remarkable property that not all the elements of the density matrix are required in KSDFT. Electron density ρ only requires the diagonal elements of the density matrix. The Hamiltonian matrix H contains the Laplacian operator, and the nearest off-diagonal elements of the density matrix are needed to calculate $\text{Tr}[Hf(H)]$ in the ground state energy. In summary, KSDFT *only requires the diagonal elements and the nearest off-diagonal elements of the density matrix.* This property is essential in order to achieve an accurate and efficient method for solving KSDFT.

Although this mathematical property has been observed for a long time [101], it is not at all reflected in the existing methods for KSDFT calculation. The direct reason is that it is not straightforward to calculate the diagonal and nearest off-

diagonal elements of a complicated matrix function. Before introducing the new method developed in this dissertation that directly calculates the diagonal and nearest off-diagonal elements of the density matrix, we first discuss the existing methods for solving KSDFT.

1.6 Existing methods and software packages for solving KSDFT

Most of the existing methods for solving KSDFT can be categorized into two types: cubic scaling methods and linear scaling methods. Within each category there are a large number of software packages available. In this section we mainly discuss two types of algorithms in the pseudopotential framework. To facilitate readers who are unfamiliar with this subject, a short list of the capabilities for the most versatile software packages is given at the end of the discussion. The URL address of each software package is provided for readers who are interested in further details. Finally, for completeness of the discussion we also mention some software packages for all-electron calculations.

1.6.1 Cubic scaling methods

Cubic scaling method is implemented in most of the popular software packages for KS-DFT calculation. The cubic scaling methods include the direct diagonalization methods such as the Davidson method [69], the conjugate gradient method (CG) [239], and the direct inversion in the iterative subspace (DIIS) method [214]. Other variants that also fall into this category include the Car-Parrinello method [49] and the Chebyshev filtering method [256], to name a few. The variants of the diagonalization methods result in different preconstant in front of the asymptotic cubic scaling. However, the orthogonalization step is inevitable in order to obtain the electron density and the

ground state energy, and leaves the cubic scaling unchanged.

Take the diagonalization method for instance, the lowest $\widetilde{N}$ eigenvalues where $\widetilde{N} = N$ for insulating system or $\widetilde{N} > N$ for metallic system (to include the finite temperature effect) are directly computed. The Kohn-Sham map is then evaluated as

$$\rho(x) = \sum_{i=1}^{\widetilde{N}} \frac{1}{1 + e^{\beta(\epsilon_i - \mu)}} |\psi_i(x)|^2. \tag{1.38}$$

Since the diagonalization method constructs all the occupied electron orbitals $\{\psi_i\}$ explicitly, the entire density matrix $f(H)$ is essentially constructed. It does not take advantage of the mathematical property that only the diagonal elements and nearest off-diagonal elements of the density matrix are needed in KSDFT calculation. The diagonalization method involves an orthogonalization step of the subspace $\Psi = [\psi_1, \cdots, \psi_{\widetilde{N}}]$. In the discrete case, the length of each vector ψ_i is proportional to N and the total number of electrons is N. Therefore the orthogonalization step scales as $\mathcal{O}(N^3)$ with respect to the number of electrons in the system, and the computational cost of KSDFT becomes very high for large number of electrons.

Below are some representative software packages for electronic structure calculation using cubic scaling methods:

- ABINIT: Diagonalization method with planewave basis functions.

  `http://www.abinit.org/`

- BigDFT: Diagonalization method with a two-level wavelet basis functions.

  `http://inac.cea.fr/L\_Sim/BigDFT/`

- CASTEP: Diagonalization method with planewave basis functions.

  `http://www.castep.org/`

- CP2K: Diagonalization method with mixed Gaussian and planewave basis functions.

http://cp2k.berlios.de/

- CPMD: Diagonalization method as well as Car-Parrinello method with planewave basis functions.

  http://www.cpmd.org/

- OPENMX (Open source package for material explorer): Diagonalization method with planewave basis functions and numerical atomic orbitals.

  http://www.openmx-square.org/

- PASRSEC (Pseudopotential Algorithm for Real-Space Electronic Calculations): Diagonalization method and Chebyshev filtering method with finite difference discretization.

  http://parsec.ices.utexas.edu/index.html

- Quantum ESPRESSO: Diagonalization method as well as Car-Parrinello method with planewave basis functions.

  http://www.quantum-espresso.org/

- VASP (Vienna Ab-initio Simulation Package): Diagonalization method with planewave basis functions.

  http://cms.mpi.univie.ac.at/vasp/

1.6.2 Linear scaling methods

The major breakthrough that reduces the $\mathcal{O}(N^3)$ in the past two decades is the linear scaling methods. The linear scaling methods use the nearsightedness property, which means that the density perturbation induced by a local change in the external potential decays off exponentially with respect to the distance from the place where the perturbation was applied, and also that the off-diagonal elements of the density

matrix decay exponentially [142, 212]. The nearsightedness property is valid for insulating systems and metallic systems at finite temperature. The nearsightedness property is not valid for metallic systems at zero temperature due to the well-known Friedel oscillation [87]. Due to the fast decay of the density matrix along the off-diagonal direction, the density matrix can be truncated beyond a certain range along the off-diagonal direction for insulating systems. Various methods have been proposed based on different perspectives of the nearsightedness property (for a detailed review, see [101]). We review briefly some representative linear scaling methods as below. The linear scaling methods are mainly divided into two classes.

The first class of linear scaling algorithms are based on the localization of electron orbitals and subspaces of electron orbitals. In the orbital minimization approach (OM) [88, 184], the truncation of the electron orbitals is imposed by adding an additional confining potential to the Hamiltonian. Orbital minimization approach can have multiple minima [136]. The orbital minimization method can be combined with the localization procedure (OML) [89] to eliminate the multiple minima problem. The localized subspace iteration method (LSI) [91] localizes the subspace consisting several electron orbitals, and obtains the optimal truncation radius.

The second class of linear scaling algorithms are based on the localization of the density matrix directly. In the divide-and-conquer method (D&C) [251], the electron density is divided into a set of loosely coupled subsystems. Each subsystem is solved separately by standard diagonalization methods and linear scaling is achieved. The density matrix minimization method (DMM) [154, 229] achieves linear scaling by directly truncating the density matrix beyond a predetermined truncation radius, with the help of the McWeeny purification transformation [185]. The density matrix is then optimized using nonlinear conjugate gradient method. The Fermi operator expansion method (FOE) [19, 99] expands the Fermi-Dirac matrix function into simple matrix functions that can be directly evaluated without diagonalization of the

Hamiltonian. These simple matrix functions can be polynomials or rational functions of the Hamiltonian matrix. Each simple matrix function is only evaluated within the truncation range of the density matrix along the off-diagonal direction, and the FOE method achieves linear scaling.

Several widely used linear scaling methods for the electronic structure calculation of insulating systems include:

- CONQUEST: b-spline basis functions and Pseudo-atomic orbitals (PAO). Linear scaling is achieved by McWeeny's purification method [185].

  `http://hamlin.phys.ucl.ac.uk/NewCQWeb/bin/view`

- ONETEP (Order-N Electronic Total Energy Package): Non-orthogonal generalized Wannier functions (NGWF). Linear scaling is achieved by density kernel optimization method which is a variant of the density matrix minimization method [154].

  `http://www2.tcm.phy.cam.ac.uk/onetep/`

- SIESTA (Spanish Initiative for Electronic Simulations with Thousands of Atoms): Numerical atomic orbitals. Linear scaling is achieved by orbital minimization method [184].

  `http://www.icmab.es/siesta/`

1.6.3 All-electron methods

As opposed to the pseudopotential framework which only involves valence electrons, all electron methods treat the core electrons and the valence electrons on the same footing. All electron methods can avoid the numerical error caused by the limited transferability in the pseudopotentials, but the computational cost of the all-electron methods is generally significantly larger than that of the pseudopotential methods.

The increased computational cost is mainly due to the fact that the electron orbitals are non-smooth at the positions of the nuclei (satisfying the cusp condition [134]) and are much more oscillatory around the positions of the nuclei. As a result, analytic or semi-analytic forms of basis functions are generally used in the all-electron calculations, such as Slater-type orbitals (STO), Gaussian-type orbitals (GTO) and numerical atomic orbitals (NAO). Several widely used software packages for all-electron calculation include:

- ADF (Amsterdam Density Functional): Diagonalization methods with Slater-type orbitals (STO).

  `http://www.scm.com/`

- GAUSSIAN: Diagonalization methods with Gaussian-type orbitals (GTO).

  `http://www.gaussian.com/`

- FHI-aims: Diagonalization methods with numerical atomic orbitals (NAO).

  `https://aimsclub.fhi-berlin.mpg.de/`

- WIEN2k: Diagonalization methods with full-potential (linearized) augmented plane-wave (FP-LAPW) plus local orbitals (LO) basis functions.

  `http://www.wien2k.at/`

1.7 Unified, accurate and efficient method for solving KSDFT

Section 1.6 summarizes the most widely used numerical schemes for solving KSDFT. Cubic scaling methods involve the orthogonalization step which is intrinsically of $\mathcal{O}(N^3)$ scaling and is difficult to be improved in general. Linear scaling methods uses the nearsightedness property for insulating systems and metallic systems at finite

temperature. However, the performance of the linear scaling methods relies crucially on the truncation radius. The truncation radius can be quite large if high numerical accuracy is to be achieved, especially for metallic system where the truncation radius depends explicitly on the artificial temperature. Therefore linear scaling algorithms typically exhibit advantage only for systems with a very large number of electrons [101]. Moreover, linear scaling methods still construct the entire density matrix, and they do not aim at calculating the diagonal elements and nearest off-diagonal elements of the density matrix directly. In order to avoid the difficulties in the linear scaling methods, it is desirable to design a method that does not use nearsightedness, and that calculates the diagonal elements and nearest off-diagonal elements of the density matrix directly. Part I of this dissertation achieves this goal under the framework of Fermi operator expansion (FOE).

FOE expands the Fermi-Dirac matrix function into simple matrix functions. Each simple matrix function is calculated directly without diagonalization process, and thus FOE does not involve the orthogonalization step. In calculating simple matrix functions, FOE does not necessarily require the nearsightedness property. As a result FOE has none of the previously mentioned drawbacks. The new method is accurate, efficient, and is applicable to both insulating and metallic systems at low or at high temperature. First we discuss the basic procedure of FOE.

FOE expands the Fermi-Dirac matrix function $f(H)$ into simple matrix functions $\{f_i(H)\}$, *i.e.*

$$f(H) \approx \sum_{i=1}^{P} f_i(H), \tag{1.39}$$

and the electron density and the ground state energy can be calculated as

$$\rho \approx \sum_{i=1}^{P} \text{diag } \ f_i(H[\rho]), \tag{1.40}$$

$$\begin{aligned} E_{KS} \approx & \sum_{i=1}^{P} \text{Tr}\left[H[\rho] f_i(H[\rho])\right] - \frac{1}{2} \iint \frac{\rho(x)\rho(y)}{|x-y|} \,\mathrm{d}x \,\mathrm{d}y \\ & + \int \epsilon_{\text{xc}}[\rho(x)] \,\mathrm{d}x - \int \epsilon'_{\text{xc}}[\rho(x)]\rho(x) \,\mathrm{d}x. \end{aligned} \tag{1.41}$$

Therefore under the framework of FOE, only the diagonal elements and nearest off-diagonal elements of each matrix function $f_i(H)$ are to be calculated. The calculation of $f_i(H)$ for different i are independent from each other.

Each simple matrix function $f_i(H)$ should take certain form in order to be evaluated directly without diagonalization. To be more specific, $f_i(H)$ can only be polynomial matrix function or rational matrix function. FOE based on both the polynomial expansion [16, 99, 102, 155, 156] and the rational expansion [19, 103, 144, 160, 164, 199, 227] have been developed. FOE based on the polynomial expansion requires calculating the powers of H. FOE based on the rational expansion requires calculating the inverse of H. Both operations are $\mathcal{O}(N^3)$ without further simplification, and FOE does not exhibit advantage over diagonalization methods for metallic systems. As a result, FOE is only mentioned sporadically in literature for certain classes of systems [144, 227].

Part I of this dissertation develops FOE to be an accurate and efficient method for solving KSDFT in all systems. The new method achieves uniform improvement over the $\mathcal{O}(N^3)$ method for systems under all dimensions. The asymptotic scaling of the new method is $\mathcal{O}(N)$ for one dimensional system, $\mathcal{O}(N^{1.5})$ for two-dimensional system, and $\mathcal{O}(N^2)$ for three-dimensional system. Furthermore, the new method can be combined with the nearsightedness property to achieve $\mathcal{O}(N)$ scaling at all dimensions for insulating systems and for metallic systems at high temperature.

We also expect that the new method should not only exhibit improved asymptotic scaling but also have a relatively small preconstant. To this end it is necessary to systematically study all the phases of FOE. In this dissertation, the complete flowchart of FOE is divided into four phases (see Fig. 1.2): Discretize the Hamiltonian operator H into a matrix of finite size (discretization); Represent the Fermi-Dirac matrix function $f(H)$ into appropriate simple matrix functions $\{f_i(H)\}$ (representation); Evaluate the diagonal and nearest off-diagonal elements of each $\{f_i(H)\}$ (evaluation); Self-consistent iteration (iteration). Part I of this dissertation develops a novel scheme, named the adaptive local basis functions for the discretization step. The adaptive local basis functions achieve high accuracy (below 10^{-3} Hartree/atom) with a very small number of basis functions. This dissertation presents the optimal strategy for the representation step, which represents the Fermi-Dirac operator in terms of a simple rational expansion called the pole expansion. This dissertation further develops a fast algorithm for evaluating the diagonal and nearest off-diagonal elements of each rational function, called the selected inversion algorithm. The computational scaling of the selected inversion algorithm to evaluate each rational function is $\mathcal{O}(N)$ for one dimensional systems, $\mathcal{O}(N^{1.5})$ for two-dimensional systems, and $\mathcal{O}(N^2)$ for three-dimensional systems. Self-consistent iteration is an important component in the KSDFT calculation. However, the self-consistent iteration does not cause the $\mathcal{O}(N^3)$ scaling problem and is a relatively separate issue. The self-consistent iteration is not discussed in this dissertation, but will be studied in the future work.

The rest of Part I of this dissertation is organized as follows. Chapter 2 discusses the discretization technique for KSDFT, and introduces the novel adaptive local basis functions. Chapter 3 discusses various representation methods of the Fermi-Dirac operator, and presents the optimal strategy for representing the Fermi-Dirac operator in terms of rational expansion. Chapter 4 introduces a new methodology named selective inversion for evaluating the diagonal elements and nearest off-diagonal ele-

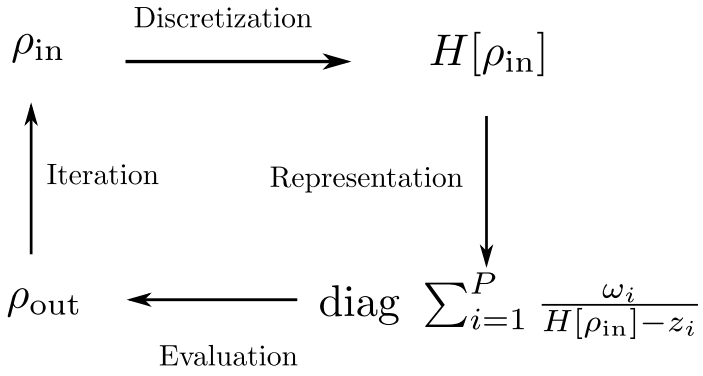


Figure 1.2: Flowchart of the unified, accurate and efficient method developed in this dissertation for solving KSDFT.

ments of each simple matrix function. The work of selected inversion also inspired us developing a fast algorithm for the construction of hierarchical matrices [161]. This is illustrated in Chapter 5. Finally, Chapter 6 concludes Part I of the dissertation with open questions and further work.

Chapter 2

Discretization of the Hamiltonian matrix: adaptive local basis functions

2.1 Introduction

In order to study the electronic structure using KSDFT with numerical methods, the first step is to discretize the Kohn-Sham Hamiltonian into a matrix of finite size. If space is uniformly discretized, the Kohn-Sham Hamiltonian generally requires a basis set with a large number of degrees of freedom per atom. For most chemical systems, the kinetic energy cutoff typically ranges from 15Ry to 90Ry for standard planewave discretization in the norm-conserving pseudopotential framework [241], which amounts to about $500 \sim 5000$ basis functions per atom. The required number of basis functions per atom is even larger for uniform discretization methods other than planewaves, such as the finite difference method [2, 58] and the finite element method [201, 202, 242].

The large number of basis functions per atom originates from the rapid oscillation

of the Kohn-Sham electron orbitals. The Kohn-Sham orbitals oscillate rapidly around the nuclei and become smooth in the interstitial region of the nuclei. Physical intuition suggests that the rapid oscillations around the nuclei are inert to changes in the environment. A significant part of the rapid oscillations can already be captured by the orbitals associated with isolated atoms. These orbitals are called atomic orbitals. Numerical methods based on atomic orbitals or similar ideas have been designed based on this observation [14, 35, 71, 81, 132, 135, 141, 198]. Environmental effect is not built into the atomic orbitals directly, but can only be approximated by fine tuning the adjustable parameters in these atomic orbitals. The values of the adjustable parameters therefore vary among different chemical elements, and sometimes vary among the different ambient environment of atoms. The quality of the atomic orbitals is difficult to be improved systematically, but relies heavily on the knowledge of the underlying chemical system.

Atomic orbitals and uniform discretization methods can be combined, as in the mixed basis methods [4, 34, 230, 236]. The quality of the basis functions can therefore be systematically improved. However, fine tuning the adjustable parameters is still necessary due to the absence of the environmental effect in the basis functions, and in certain circumstances the number of basis functions per atom is still large.

In this chapter, we propose a novel discretization method to build the environmental effects into the basis set to achieve further dimension reduction of the basis set. The basis functions are constructed adaptively and seamlessly from the atomic configuration in local domains, called elements. The basis functions are discontinuous at the boundary of the elements, and they form the basis set used in the discontinuous Galerkin (DG) framework. The discontinuous Galerkin framework has been widely used in numerical solutions of partial differential equations (PDE) for more than four decades, see for example [7, 8, 15, 64, 66, 247] and the references therein. One of the main advantages of the DG method is its flexibility in the choice of the basis func-

tions. The flexibility of the DG framework allows us to employ these discontinuous basis functions to approximate the continuous Kohn-Sham orbitals, and allows us to achieve high accuracy (below 10^{-3} Hartree/atom) in the total energy calculation with the number of basis functions per atom close to the minimum possible number of basis functions for the electronic structure calculation, namely the number of basis functions used by the tight binding method [81, 231]. Our method is implemented in parallel with a rather general data communication framework, and the current implementation is able to calculate the total energy for systems consisting thousands of atoms.

The novel discretization scheme developed in this chapter can be applied to both the diagonalization methods and the Fermi operator expansion method that are going to be discussed in Chapter 3 and Chapter 4. To simplify the discussion the diagonalization method will be used in the current chapter.

The idea of constructing basis functions adaptively from the local environment has also been explored in other circumstances in numerical analysis such as reduced basis method [48, 62, 174, 175] and multi-scale discontinuous Galerkin method [246, 254, 255] for solving PDE. In the current context, we apply the DG algorithm to solve eigenvalue problems with oscillatory eigenfunctions, and the basis functions are constructed by solving auxiliary local problems numerically.

This chapter is organized as follows. Section 2.2 introduces the discontinuous Galerkin framework for Kohn-Sham density functional theory. The construction of the adaptive local basis functions is introduced in Section 2.3. Section 2.4 discusses implementation issues in more detail. The performance of our method is reported in Section 2.5, followed by the discussion and conclusion in Section 2.6. Materials in this chapter have been presented in [165].

2.2 Discontinuous Galerkin framework for Kohn-Sham density functional theory

The discontinuous Galerkin (DG) methods have been developed for different types of partial differential equations [7, 8, 15, 64, 66, 247]. One of the main advantages of the DG method is its flexibility in the choice of the approximation space, as the DG method does not require the continuity condition of the basis functions across the interfaces of the elements. This flexibility is important for constructing effective discretization schemes for Kohn-Sham density functional theory.

We present in the following a DG method for the evaluation of the electron density. Among the different formalisms in the DG framework, we will use the interior penalty method [7, 15]. Other DG methods, such as the local DG method (LDG) can be employed as well [65]. The interior penalty method naturally generalizes the variational principle (1.30).

We denote by Ω the computational domain with the periodic boundary condition. Ω is also referred to as the global domain in the following discussion. Bloch boundary conditions can be taken into account as well without essential modification. Let $\mathcal{T}$ be a collection of quasi-uniform rectangular partitions of Ω:

$$\mathcal{T} = \{E_1, E_2, \cdots, E_M\}, \tag{2.1}$$

and $\mathcal{S}$ be the collection of surfaces that correspond to $\mathcal{T}$. Each E_k is called an element of Ω. For a typical choice of partitions used in practice, the elements are chosen to be of the same size. For example, for a crystalline material, elements can be chosen as integer multiples of the conventional cell of the underlying lattice. As a result, unlike the usual finite element analysis, the element size will remain the same. [1]

[1] In the language of finite element method, we will not use the h-refinement.

In the following discussion, we use extensively the inner products defined as below

$$\langle v, w\rangle_E = \int_E v^*(x)w(x)\,\mathrm{d}x, \tag{2.2}$$

$$\langle \boldsymbol{v}, \boldsymbol{w}\rangle_S = \int_S \boldsymbol{v}^*(x)\cdot\boldsymbol{w}(x)\,\mathrm{d}s(x), \tag{2.3}$$

$$\langle v, w\rangle_{\mathcal{T}} = \sum_{i=1}^{M}\langle v, w\rangle_{E_i}, \tag{2.4}$$

$$\langle \boldsymbol{v}, \boldsymbol{w}\rangle_{\mathcal{S}} = \sum_{S\in\mathcal{S}}\langle \boldsymbol{v}, \boldsymbol{w}\rangle_S. \tag{2.5}$$

In the interior penalty method, the discretized energy functional corresponding to (1.30) is given by

$$E_{\mathrm{DG}}(\{\psi_i\}) = \frac{1}{2}\sum_{i=1}^{N}\langle\nabla\psi_i, \nabla\psi_i\rangle_{\mathcal{T}} - \sum_{i=1}^{N}\langle\{\{\nabla\psi_i\}\}, [\![\psi_i]\!]\rangle_{\mathcal{S}} + \langle V_{\mathrm{eff}}, \rho\rangle_{\mathcal{T}} + \frac{\alpha}{h}\sum_{i=1}^{N}\langle[\![\psi_i]\!], [\![\psi_i]\!]\rangle_{\mathcal{S}} + \sum_{\ell}\gamma_\ell\sum_{i=1}^{N}|\langle b_\ell, \psi_i\rangle_{\mathcal{T}}|^2. \tag{2.6}$$

Here the last term comes from the non-local terms in Eq. (1.30), and $\{\{\,\cdot\,\}\}$ and $[\![\,\cdot\,]\!]$ are the average and the jump operators across surfaces, defined as follows. For $S \in \mathcal{S}^\circ$ the set of interior surfaces, we assume S is shared by elements K_1 and K_2. Denote by n_1 and n_2 the unit normal vectors on S pointing exterior to K_1 and K_2, respectively. With $u_i = u|_{\partial K_i}$, $i = 1, 2$, we set

$$[\![u]\!] = u_1 n_1 + u_2 n_2 \quad \text{on } S. \tag{2.7}$$

For $S \in \mathcal{S}^\partial$ where $\mathcal{S}^\partial$ is the union of the surfaces on the boundary, we set

$$[\![u]\!] = un \quad \text{on } S, \tag{2.8}$$

where n is the outward unit normal vector. For vector-valued function q, we define

$$\{\{q\}\} = \tfrac{1}{2}(q_1 + q_2) \quad \text{on } S \in \mathcal{S}^\circ, \tag{2.9}$$

where $q_i = q|_{\partial K_i}$, and

$$\{\{q\}\} = q \quad \text{on } S \in \mathcal{S}^\partial. \tag{2.10}$$

Note that in the current context $\mathcal{S} = \mathcal{S}^\circ$ since we assume periodic boundary condition for the computational domain, and every surface is an interior surface. The constant α in (2.6) is a positive penalty parameter, which penalizes the jumps of functions across element surfaces to guarantee stability. The choice of α will be further discussed in Section 2.5.

Assume that we have chosen for each element E_k a set of basis functions $\{\varphi_{k,j}\}_{j=1}^{J_k}$, where J_k is the total number of basis functions in E_k. We extend each $\varphi_{k,j}$ to the whole computational domain Ω by setting it to be 0 on the complement set of E_k. Define the function space $\mathcal{V}$ as

$$\mathcal{V} = \text{span}\{\varphi_{k,j},\, E_k \in \mathcal{T},\, j = 1, \cdots, J_k\}. \tag{2.11}$$

We minimize (2.6) for $\{\psi_i\} \subset \mathcal{V}$. The energy functional (2.6) in the approximation space $\mathcal{V}$ leads to the following eigenvalue problem for $\{\psi_i\}_{i=1}^N$. For any $v \in \mathcal{V}$,

$$\begin{aligned}\frac{1}{2}\langle \nabla v, \nabla \psi_i\rangle_{\mathcal{T}} - \frac{1}{2}\langle [\![v]\!], \{\{\nabla \psi_i\}\}\rangle_{\mathcal{S}} - \frac{1}{2}\langle \{\{\nabla v\}\}, [\![\psi_i]\!]\rangle_{\mathcal{S}} + \frac{\alpha}{h}\langle [\![v]\!], [\![\psi_i]\!]\rangle_{\mathcal{S}} \\ + \langle v, V_{\text{eff}}\psi_i\rangle_{\mathcal{T}} + \sum_\ell \gamma_\ell \langle v, b_\ell\rangle_{\mathcal{T}}\langle b_\ell, \psi_i\rangle_{\mathcal{T}} = \lambda_i \langle v, \psi_i\rangle_{\mathcal{T}}.\end{aligned} \tag{2.12}$$

Setting $v = \varphi_{k',j'}$ and

$$\psi_i = \sum_{E_k \in \mathcal{T}} \sum_{j=1}^{J_k} c_{i;k,j}\varphi_{k,j}, \tag{2.13}$$

we arrive at the following linear system

$$\sum_{k,j}\Big(\frac{1}{2}\langle\nabla \boldsymbol{f}_{k',j'},\nabla \boldsymbol{f}_{k,j}\rangle_{\mathcal{T}}-\frac{1}{2}\langle[\![\boldsymbol{f}_{k',j'}]\!],\{\{\nabla \boldsymbol{f}_{k,j}\}\}\rangle_{\mathcal{S}}$$
$$-\frac{1}{2}\langle\{\{\nabla \boldsymbol{f}_{k',j'}\}\},[\![\boldsymbol{f}_{k,j}]\!]\rangle_{\mathcal{S}}+\frac{\alpha}{h}\langle[\![\boldsymbol{f}_{k',j'}]\!],[\![\boldsymbol{f}_{k,j}]\!]\rangle_{\mathcal{S}}+\langle \boldsymbol{f}_{k',j'},V_{\text{eff}}\boldsymbol{f}_{k,j}\rangle_{\mathcal{T}}$$
$$+\sum_{\ell}\gamma_\ell\langle \boldsymbol{f}_{k',j'},b_\ell\rangle_{\mathcal{T}}\langle b_\ell,\boldsymbol{f}_{k,j}\rangle_{\mathcal{T}}\Big)c_{i;k,j}=\lambda_i\sum_{k,j}\langle \boldsymbol{f}_{k',j'},\boldsymbol{f}_{k,j}\rangle c_{i;k,j}. \quad (2.14)$$

We define A to be the matrix with entries given by the expression in the parentheses, B to be the matrix with entries $\langle \boldsymbol{f}_{k',j'},\boldsymbol{f}_{k,j}\rangle$, and c_i to be the vector with components $(c_{i;k,j})_{k,j}$. We have the following simple form of generalized eigenvalue problem

$$Ac_i=\lambda_i Bc_i$$

for $i=1,2,\ldots,N$. Following the standard terminologies in the finite element method, we call A the (DG) stiffness matrix, and B the (DG) mass matrix. In the special case when the DG mass matrix B is equal to the identity matrix, we have a standard eigenvalue problem $Ac_i=\lambda_i c_i$. Once $\{c_i\}$ are available, the electron density is calculated by

$$\widetilde{\rho}=\sum_{i=1}^{N}\sum_{E_k\in\mathcal{T}}|\sum_{j=1}^{J_k}c_{i;k,j}\varphi_{k,j}|^2. \quad (2.15)$$

2.3 Basis functions adapted to the local environment

The proposed framework in the previous section is valid for any choice of basis functions. To improve the efficiency of the algorithm, it is desirable to use less number of basis functions while maintaining the same accuracy. In order to achieve this goal, the choice of the functions $\{\varphi_{k,j}\}$ is important. In this section, we discuss a way to

construct the basis functions $\{\varphi_{k,j}\}$ that are adapted to the local environment.

The starting point is the observation as follows. The Kohn-Sham orbitals $\{\psi_i\}$ exhibit singularities around the nuclei. In an all electron calculation, the nuclei charge density is the summation of delta functions located at the positions of the nuclei (or numerical delta function after discretization) and the Kohn-Sham orbitals have cusp points at the positions of the atoms. In the pseudopotential framework which involves only valence electrons, one can still see that the Kohn-Sham orbitals and the electron density are much more oscillatory near the atom cores than in the interstitial region, as illustrated in Fig. 2.1. In the setting of the real space method or the planewave method, in order to resolve the Kohn-Sham orbitals around the atom cores where the derivatives of Kohn-Sham orbitals become large, one has to use a uniform fine mesh. Therefore, the number of mesh points becomes huge even for a small system. This makes the electronic structure calculation expensive.

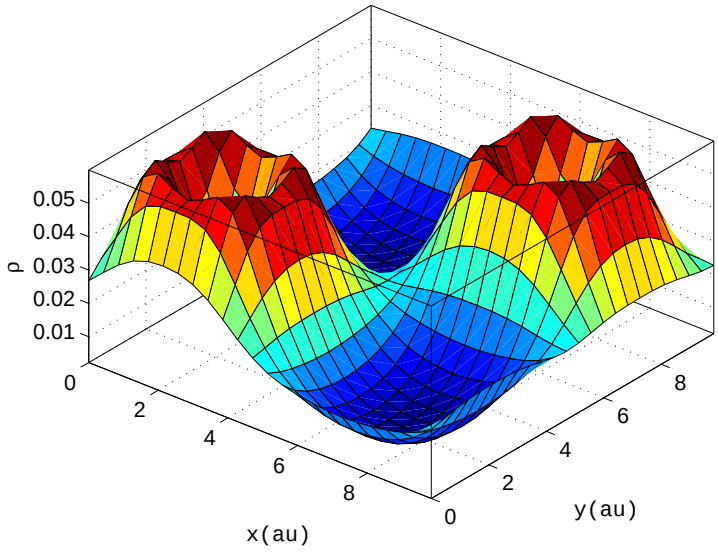


Figure 2.1: Electron density on a (001) slice of a mono-crystalline silicon system passing through two Si atoms. The two Si atoms are located at $(2.57, 2.57)$ au and at $(7.70, 7.70)$ au in this plane, respectively. The electron density shows oscillatory behavior near the nuclei of Si atoms and becomes smooth in the interstitial region.

In order to reduce the cost, we note that the Kohn-Sham orbitals are smooth away from the atoms and the uniform fine discretization is not efficient enough. A natural idea would be to use adaptive mesh refinement techniques, which is just started to be explored in electronic structure calculations [68, 82].

Our approach builds the oscillatory behavior of the Kohn-Sham orbitals near the atom cores directly into the basis functions. Hence, a small number of basis functions are enough to characterize the Kohn-Sham orbitals. This idea is not entirely new. For example, the philosophy of pseudopotential techniques is quite similar, though the reduction is done at the analytic level. On the side of numerical methods, the current idea is closely related to atomic orbital basis and numerical atomic orbitals [35, 81].

The main difference from the previous approaches is that instead of predetermining basis functions based on the information from isolated atoms, our approach builds the information from the local environment into the basis functions as well. Thanks to the flexibility of the discontinuous Galerkin framework, this can be done in a seamless and systematic way. The basis functions form a complete basis set in the global domain Ω. The basis set is therefore efficient, and at the same time the accuracy can be improved systematically. This is an important difference between this approach and the previous methods along the same line.

The basis functions $\{\varphi_{k,j}\}$ are determined as follows. Given the partition $\mathcal{T}$ and the effective potential V_{eff}, let us focus on the construction of $\{\varphi_{k,j}\}$, $j = 1, \cdots, J_k$ for one element $E_k \in \mathcal{T}$. As discussed above, our approach is to adapt $\{\varphi_{k,j}\}$ to the local environment in E_k.

For each element E_k, we take a region $Q_k \supset E_k$. Q_k is called the extended element associated with the element E_k. The set $Q_k \backslash E_k$ is called the buffer area. We assume that Q_k extends symmetrically along the $\pm x(y, z)$ directions from the boundary of E_k. The length of the buffer area extended beyond the boundary of E_k along the $\pm x(y, z)$ direction is called the "buffer size along the $x(y, z)$ direction". We restrict the effective

Hamiltonian on Q_k by assuming the periodic boundary condition on ∂Q_k and denote by H_{eff,Q_k} the restricted Hamiltonian. H_{eff,Q_k} is discretized and diagonalized, and the corresponding eigenfunctions are denoted by $\{\widetilde{\varphi}_{k,j}\}$, indexed in increasing order of the associated eigenvalues. We restrict the first J_k eigenfunctions $\{\widetilde{\varphi}_{k,j}\}$ from Q_k to E_k, denoted by $\{\varphi_{k,j}\}$. Each $\varphi_{k,j}$ is therefore defined locally on E_k. As discussed before we extend each $\varphi_{k,j}$ to the global domain Ω by setting the value to be 0 on the complement of E_k. The resulting functions, still denoted by $\{\varphi_{k,j}\}$ are called the adaptive local basis functions. Numerical result suggests that we can take very small J_k to achieve high accuracy.

The reason why we choose the periodic boundary condition on Q_k for the restriction H_{eff,Q_k} is twofold. On one hand, the periodic boundary condition captures better the bulk behavior of the system (than the Dirichlet boundary condition for example); On the other hand, the periodic boundary condition makes the solution of H_{eff,Q_k} more easily adapted to existing DFT algorithms and packages, as most of them can treat periodic boundary conditions. Other choices such as the Neumann boundary condition are possible, and the optimal choice of boundary conditions remains to be an open question.

The basis functions constructed from the buffer region well capture the local singular behavior of Kohn-Sham orbitals near the nuclei. Hence, the approximation space formed by $\{\varphi_{k,j}\}$ gives an efficient and accurate discretization to the problem, as will be illustrated by numerical examples in Section 2.5. Note that the $\{\widetilde{\varphi}_{k,j}\}$'s are the eigenfunctions of the self-adjoint operator H_{eff,Q_k} on Q_k, and therefore form a complete basis set on Q_k when $J_k \to \infty$. This implies that after restriction, the functions $\{\varphi_{k,j}\}$ also form a complete basis set on E_k as $J_k \to \infty$. The accuracy can therefore be systematically improved in the electronic structure calculation.

Eq. (2.14) proposes a generalized eigenvalue problem. From numerical point of view it would be more efficient if we can choose $\{\varphi_{k,j}\}$ such that the DG mass matrix

is an identity matrix and that Eq. (2.14) becomes a standard eigenvalue problem. Moreover, as J_k increases, the basis functions $\{\varphi_{k,j}\}$ can be degenerate or nearly degenerate, which increases the condition number of the DG stiffness matrix. Both problems can be solved at the same time by applying a singular value decomposition (SVD) filtering step, resulting in an orthonormal basis set $\{\varphi_{k,j}\}$:

1. For each k, form a matrix $M_k = (\varphi_{k,1}, \varphi_{k,2}, \cdots, \varphi_{k,J_k})$ with $\varphi_{k,j}$;

2. Calculate SVD decomposition $UDV^* = M_k$,

$$D = \text{diag } (\lambda_{k,1}, \lambda_{k,2}, \cdots, \lambda_{k,J_k}),$$

   where $\lambda_{k,j}$ are singular values of M_k ordered decreasingly in magnitude;

3. For a threshold δ, find $\widetilde{J}_k$ such that $|\lambda_{k,\widetilde{J}_k}| > \delta$ and $|\lambda_{k,\widetilde{J}_k+1}| < \delta$ ($\widetilde{J}_k = J_k$ if all singular values are larger than the threshold). Take U_j be the j-th column of U, $j = 1, \cdots, \widetilde{J}_k$;

4. Set $J_k \leftarrow \widetilde{J}_k$ and $\varphi_{k,j} \leftarrow U_{k,j}$ for $j = 1, \cdots, \widetilde{J}_k$.

Remark 1. *Although the threshold δ can avoid numerical degeneracy of the basis functions, the numerical degeneracy is not observed for the cases studied in section 2.5. In other words, $J_k = \widetilde{J}_k$.*

After constructing the basis functions $\{\varphi_{k,j}\}$, we then apply the discontinuous Galerkin framework to solve the $\{\psi_i\}$ and hence ρ corresponding to H_{eff}. The overall algorithm can be summarized as follows:

1. Set $n = 0$, let $\mathcal{T}$ be a partition of Ω into elements, and ρ_0 be an initial trial electron density;

2. Form the effective potential $V_{\text{eff}}[\rho_n]$ and the effective Hamiltonian $H_{\text{eff}}[\rho_n]$;

3. For each element $E_k \in \mathcal{T}$, calculate the eigenfunctions $\{\varphi_{k,j}\}, j = 1, \cdots, J_k$ corresponding to the Hamiltonian H_{eff,Q_k} on the extended element Q_k, and obtain the orthonormal adaptive local basis functions $\{\varphi_{k,j}\}$.

4. Solve (2.14) to obtain the coefficients $\{c_{i;k,j}\}$ for the Kohn-Sham orbitals and reconstruct the electron density $\widetilde{\rho}$ by (2.15);

5. Mixing step: Determine ρ_{n+1} from ρ_n and $\widetilde{\rho}$. If $\|\rho_n - \widetilde{\rho}\| \leq \delta$, stop; otherwise, go to step (2) with $n \leftarrow n + 1$.

We remark that due to the flexibility of the DG framework one can supplement the functions $\{\varphi_{k,j}\}$ constructed above by other functions in E_k, such as local polynomials in E_k, Gaussian functions restricted to E_k, and other effective basis functions based on physical and chemical intuition. From practical point of view, we find that the adaptive basis set constructed above already achieves satisfactory performance.

2.4 Implementation details

This section explains the implementation details for the adaptive local basis functions. This section is mostly written for the readers who are less familiar with the DG implementation.

2.4.1 Grids and interpolation

The adaptive local basis functions involve three types of domains: the global domain Ω, the extended elements $\{Q_k\}$, and the elements $\{E_k\}$. Quantities defined on these domains are discretized with different types of grids.

- On Ω, the quantities such as ρ and V_{eff} are discretized with a uniform Cartesian grid with a spacing fine enough to capture the singularities and oscillations in these quantities.

- The grid on Q_k is simply the restriction of the uniform grid of Ω on Q_k. This is due to the consideration that all quantities on Q_k are treated as periodic and hence a uniform grid is the natural choice.

- The grid on E_k is a three-dimensional Cartesian Legendre-Gauss-Lobatto (LGL) grid in order to accurately carry out the operations of the basis functions $\{\varphi_{k,j}\}$ such as numerical integration and trace operator for each element E_k.

Transferring various quantities between these three grids requires the following interpolation operators.

- Ω to Q_k. This is used when we restrict the density ρ_n and the effective potential V_{eff} to the extended element Q_k. Since the grid on Q_k is the restriction of the grid on Ω, this interpolation operator simply copies the required values.

- Q_k to E_k. This is used when one restricts $\{\widetilde{\varphi}_{k,j}\}$ and their derivatives to E_k. As the grid on Q_k is uniform, the interpolation is done by Fourier transform. Due to the fact that both grids are Cartesian, the interpolation can be carried out dimension by dimension, which greatly improves the efficiency.

- E_k to Ω. This is used when one assembles the Kohn-Sham orbitals $\{\psi_i\}$ from the coefficients $\{c_{i;k,j}\}$ of the elements. The interpolation from the LGL grid to the uniform grid is done by Lagrange interpolation, again carried out dimension by dimension. Averaging is performed for the grid points of Ω shared by multiple elements.

The non-local pseudopotentials are used both in solving $\{\widetilde{\varphi}_{k,j}\}$ on each Q_k and in the numerical integration step on the LGL grid of each E_k. In our implementation, the non-local pseudopotentials are directly generated in real space on Q_k and on E_k without further interpolation between the grids.

2.4.2 Implementation of the discontinuous Galerkin method

We use planewaves in each extended element Q_k to discretize the local effective Hamiltonian H_{eff,Q_k} and the LOBPCG algorithm [139] with the preconditioner proposed in [239] to diagonalize the discretized Hamiltonian. The resulting eigenfunctions $\{\widetilde{\varphi}_{k,j}\}_{j=1}^{J_k}$ of H_{eff,Q_k} are restricted to E_k and interpolated onto its LGL grid. Within the SVD filtering step, the inner product that we adopt is the discrete weighted ℓ_2 product with the LGL weights inside E_k. The main advantage of the SVD filtering step is that the discontinuous Galerkin method results in a standard eigenvalue problem.

The assembly of the DG stiffness matrix follows (2.14) strictly and consists of the following steps.

- For the first term $\frac{1}{2}\langle \nabla \boldsymbol{f}_{k',j'}, \nabla \boldsymbol{f}_{k,j} \rangle_{\mathcal{T}}$ and the fifth term $\langle \boldsymbol{f}_{k',j'}, V_{\text{eff}} \boldsymbol{f}_{k,j} \rangle_{\mathcal{T}}$, their contributions are non-zero only when $k = k'$ since otherwise two basis functions have disjoint support. Hence, for each fixed k, we compute $\langle \nabla \boldsymbol{f}_{k,j'}, \nabla \boldsymbol{f}_{k,j} \rangle_{E_k}$ and $\langle \boldsymbol{f}_{k,j'}, V_{\text{eff}} \boldsymbol{f}_{k,j} \rangle_{E_k}$. The integration is done numerically using the LGL grid on E_k. Part of the stiffness matrix corresponding to these two terms clearly has a block diagonal form.

- For the second, third, and fourth terms of (2.14), one needs to restrict basis functions and their derivatives to element faces. As the one-dimensional LGL grid contains the endpoints of its defining interval, this is done simply by restricting the values of the three-dimensional LGL grid to the element faces. One then calculates these three terms using numerical integration on these resulting two-dimensional LGL grids. Since the integral is non-zero only when E_k and $E_{k'}$ are the same element or share a common face, part of the stiffness matrix corresponding to these three terms is again sparse.

- The last term of (2.14) is $\sum_{\ell} \gamma_{\ell} \langle \boldsymbol{f}_{k',j'}, b_{\ell} \rangle_{\mathcal{T}} \langle b_{\ell}, \boldsymbol{f}_{k,j} \rangle_{\mathcal{T}}$. The integration is again

approximated using the LGL grids of the elements. Notice that the contribution is non-zero iff $\boldsymbol{f}_{k',j'}$ and $\boldsymbol{f}_{k,j}$ overlap with the support of a common b_ℓ. Since each b_ℓ is localized around a fixed atom, $\boldsymbol{f}_{k,j}$ and $\boldsymbol{f}_{k',j'}$ need to be sufficiently close for this term to be non-zero. As a result, part of the stiffness matrix corresponding to this last term is also sparse.

Though the DG stiffness matrix A is sparse, this property is not yet exploited in the current implementation. The eigenvalues and eigenvectors of the DG stiffness matrix are calculated using the pdsyevd routine of ScaLAPACK by treating it as a dense matrix. We plan to replace it with more sophisticated solvers that leverage the sparsity of A in future.

2.4.3 Parallelization

Our algorithm is fully implemented for the message-passing environment. To simplify the discussion, we assume that the number of processors is equal to the number of elements. It is then convenient to index the processors $\{P_k\}$ with the same index k used for the elements. In the more general setting where the number of elements is larger than the number of processors, each processor takes a couple of elements and the following discussion will apply with only minor modification. Each processor P_k locally stores the basis functions $\{\boldsymbol{f}_{k,j}\}$ for $j = 1, 2, \ldots, J_k$ and the unknowns $\{c_{i;k,j}\}$ for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, J_k$. We further partition the non-local pseudopotentials $\{b_\ell\}$ by assigning b_ℓ to the processor P_k if the atom associated to b_ℓ is located in the element E_k.

The eigenfunctions of the local Hamiltonian H_{eff,Q_k} are calculated on each processor P_k. In order to build the local Hamiltonian H_{eff,Q_k}, the processor P_k needs to access all the non-local pseudopotentials of which the associated atoms are located in Q_k. This can be achieved by communication among E_k and its nearby elements. Once these pseudopotentials are available locally, the eigenfunctions of H_{eff,Q_k} are

computed in parallel without any extra communication between the processors.

The parallel implementation for assembling the DG stiffness matrix is more complicated:

- For the calculation of the first and the fifth terms of the DG stiffness matrix A in Eq. (2.14), each processor P_k performs numerical integration on E_k. Since the local basis functions $\{\boldsymbol{f}_{k,j}\}$ are only non-zero on E_k, this step is carried out fully in parallel.

- To calculate the second, third, and fourth terms, each processor P_k computes the surface integrals restricted to the *left, front,* and *bottom* faces of E_k. This requires the basis functions of the left, front, and bottom neighboring elements.

- To calculate the sixth term, each processor P_k computes the parts associated with the non-local pseudopotentials $\{b_\ell\}$ located on P_k. This requires the access to the basis functions of all elements that overlap with b_ℓ.

To summarize, each processor P_k needs to access the basis functions from its neighboring elements and from the elements that overlap with the support set of the non-local pseudopotentials located on the elements associated with P_k. Due to the locality of the non-local pseudopotentials, these elements are geometrically close to P_k. Since the size of the elements is generally equal to or larger than one unit cell, the support set of the non-local pseudopotentials are generally within the range of the neighboring elements. Therefore, the number of the non-local basis functions required by P_k is bounded by a small constant times the typical number of the basis functions in an element.

The use of the pdsyevd routine of ScaLAPACK for solving the eigenvalue problem (2.14) results in another source of communication. ScaLAPACK requires A to be stored in its block cyclic form. The block cyclic form is quite different from the form of which the DG stiffness matrix is assembled (as mentioned above). As a result, one

needs to redistribute A into this block cyclic form before calling pdsyevd and then redistribute the eigenfunctions afterwards.

In order to support these two sources of data communication, we have implemented a rather general communication framework that only requires the programmer to specify the desired non-local data. This framework then automatically fetches the data from the processors that store them locally. The actual communication is mostly done using asynchronous communication routines MPI_Isend and MPI_Irecv.

2.5 Numerical examples

In order to illustrate how our method works in practice, we present numerical results for the ground state electronic structure calculation, using sodium (Na) and silicon (Si) as characteristic examples for metallic and insulating systems, respectively. We find that high accuracy (below 10^{-3} Hartree/atom) is achieved by using only a small number of adaptive local basis functions for one, two, and three dimensional systems under a variety of conditions. Because of the small number of basis functions per atom, our DG algorithm already shows significant reduction in computational time for a small system with 128 Na atoms. We demonstrate that the current implementation is able to solve systems with thousands of atoms, and that the algorithm has a potential to be applied to much larger systems with a more advanced implementation.

This section is organized as follows: section 2.5.1 introduces the setup of the test systems and the criterion for the quantification of the error. Section 2.5.2 discusses the simplest case with the mono-crystalline quasi-1D system, followed by the discussion on the disordered quasi-1D system in section 2.5.3. We illustrate in section 2.5.4 the performance of the adaptive local basis functions under the DG framework for the quasi-2D and bulk 3D systems. We discuss how to choose the penalty parameter α in section 2.5.5. Finally we demonstrate the computational performance of our

implementation of the DG method in section 2.5.6.

2.5.1 Setup

We use the local density approximation (LDA) [52, 206] for the exchange-correlation functional and the Troullier-Martins pseudopotential [241]. More sophisticated pseudopotentials and exchange-correlation functionals can also be used without changing the structure of the implementation. All quantities are reported in atomic units (au). All calculations are carried out on the Ranger system maintained at Texas Advanced Computing Center (TACC) under NSF TeraGrid program. Ranger is a distributed-memory parallel system with 3,936 16-way SMP compute nodes and a total of 15,744 AMD Opteron quad-cores processors. Each compute node has a theoretical peak performance of 9.2 gigaflops per second (Gflops) per core, and has 32 gigabyte (GB) of memory (2 GB per core). InfiniBand technology is used for the interconnection between all nodes that ensures high data communication performance.

Fig. 2.2 (a) and (b) illustrate one unit cell of the crystalline Na and Si system, respectively. Na has a body centered cubic (bcc) unit cell, with 2 atoms per cell and a lattice constant of 7.994 au. Si has a diamond cubic unit cell, with 8 atoms per cell and a lattice constant of 10.261 au. Fig. 2.2 (c) shows a quasi-1D Na system with 4 unit cells extended along the z direction. The global domain is partitioned into 4 elements $\{E_k\}_{k=1}^4$ with one unit cell per element. The red area represents one of the elements E_2, and the corresponding extended element Q_2 consists of both the red area and the blue area. We recall that the buffer size along the $x(y, z)$ direction refers to the length of the buffer area extended beyond the boundary of the element E_k along the $x(y, z)$ direction. We use the number of unit cells as the unit of the buffer size. Fig. 2.2 shows the case with the buffer size of 1.0 (unit cell) along the z direction, and 0.0 along the x and y directions. The application of the adaptive local basis functions is not restricted to the study of mono-crystalline systems, and the

potential function in each extended element does not necessarily satisfy the periodic boundary condition. Mono-crystalline systems will be studied first to illustrate how the adaptive local basis functions work in practice. Disordered system, as well as cases with fractional buffer size will also be studied below.

To facilitate the comparison between different systems and parameters, we measure in all the examples the error of the total energy per atom. To be more specific, we obtain first the self-consistent electron density and the corresponding total energy in the global domain Ω as in Eq. (1.27). This self-consistent electron density is used as the input electron density to construct the Hamiltonian H_{eff}. The output electron density is then evaluated using the DG method, and the corresponding total energy given by the DG method is compared to that calculated in the global domain. Comparing the error of the total energy in a single evaluation step allows us to assess the numerical error even when the number of the basis functions is not sufficient. For the case with sufficient number of the basis functions, it is found that the error of the total energy per atom with additional mixing steps is consistent with the error in the single evaluation step. Here we set the target accuracy to be 10^{-3} au per atom. The real space grid size in the global domain and in the extended elements is set to be 0.50 au and 0.43 au for Na and Si, respectively. This grid size guarantees that the uncertainty in the total energy in the global domain Ω is below the target accuracy. The number of LGL grids inside each element is 16 and 24 along all the three dimensions for Na and Si, respectively, which ensures the accuracy of numerical integration.

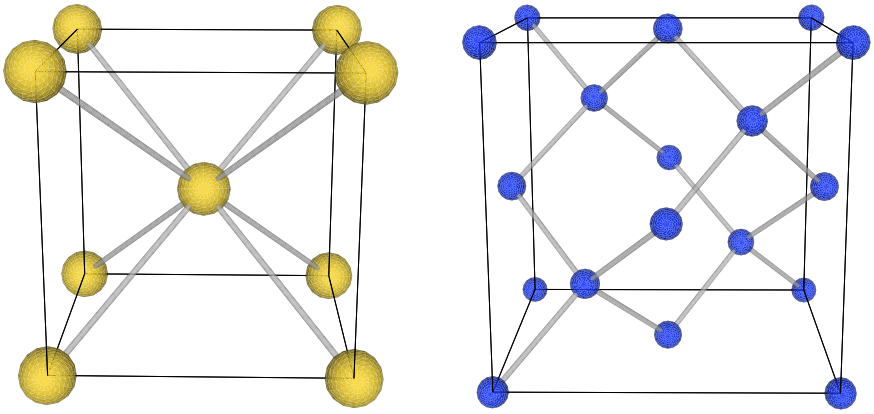
We remarked in the end of section 2.3 that the DG framework is very flexible and can incorporate not only the adaptive local basis functions but also other basis functions such as local polynomials. In practice we find that the adaptive local basis functions are computationally more efficient than polynomials. Therefore in the following discussion only adaptive local functions will be used in the basis set.

The number of adaptive local functions per atom is also referred to as the degrees of freedom (DOF) per atom.

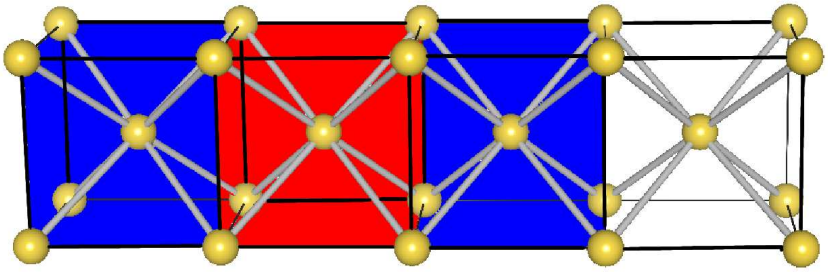
2.5.2 Periodic Quasi-1D system

Fig. 2.3 (a) shows the error of the total energy per atom with respect to different buffer sizes and different numbers of basis functions per atom (DOF per atom), for the quasi-1D periodic sodium system in Fig. 2.2 (c). The element size is fixed to be one unit cell. The penalty parameter α is 20. The error decreases systematically with respect to the increase of the buffer size. The target accuracy 10^{-3} au is plotted as the black dashed horizontal line. For a small buffer size of 0.25 (red triangle with solid line) the target accuracy is not yet reached with 20 basis functions per atom. For a buffer size of 0.50 (black diamond with solid line) only 5 basis functions per atom is needed to reach the target accuracy. For a larger buffer size of 0.75 (blue star with solid line), the error is already far below the target accuracy with merely 2 basis functions per atom. The potential function in the extended element does not satisfy the periodic boundary condition along the z direction in the case with the buffer size of 0.75, but the numerical results indicate that this violation does not much affect the quality of the resulting basis functions in each element.

Similar behavior of the error is also found in the silicon system. Fig. 2.3 (b) shows the error of the total energy per atom for the quasi-1D periodic silicon system with four unit cells extended along the z direction, and with element size being one unit cell. For a buffer size of 0.25 (red triangle with solid line) the number of basis functions per atom needed to reach the target accuracy is more than 12. For a buffer size of 0.50 (black diamond with solid line) and 0.75 (blue star with solid line) the DOF per atom needed to reach the target accuracy is 6 and 5, respectively. Physical intuition suggests that the minimum number of basis functions is 4, which reflects one $2s$ and three $2p$ atomic orbitals. $20 \sim 40$ number of basis functions per atom



(a) (b)

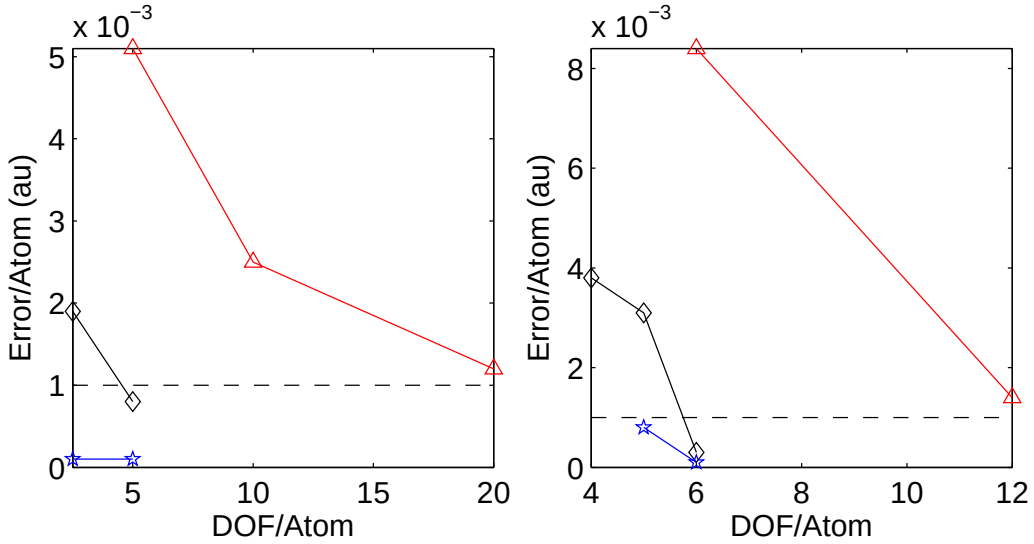


(c)

Figure 2.2: (a) The unit cell for Na. (b) The unit cell for Si. (c) A quasi-1D Na system with 4 unit cells extended along the z direction. The red area represents one of the elements E_2. The corresponding extended element Q_2 consists of both the red area and the blue area. The buffer size is 1.0 unit cell along the z direction, and is 0.0 along the x and y directions.

is generally required to achieve good accuracy if Gaussian type orbitals or numerical atomic orbitals are to be used [35]. Therefore for the quasi-1D system, our algorithm achieves nearly the optimal performance in terms of the number of basis functions per atom.

The behavior of the error found above does not depend on the length of the quasi-1D system. Fig. 2.4 compares the error of the total energy per atom of the quasi-1D mono-crystalline sodium system with respect to the length of the global domain (in the unit of unit cell numbers), for 3 DOF per atom (blue diamond with dashed line),



(a) (b)

Figure 2.3: (a) The error of the total energy per atom (the y axis) for a periodic quasi-1D sodium system consisting of 4 unit cells, with respect to the number of adaptive local basis functions per atom (the x axis). The buffer sizes are chosen to be 0.25 (red triangle with solid line), 0.50 (black diamond with solid line), and 0.75 (blue star with solid line). (b) The error of the total energy per atom for a periodic quasi-1D silicon system consisting of 4 unit cells, with respect to the number of adaptive local basis functions per atom (the x axis). The legend is the same as in (a). The black dashed horizontal line refers to the target accuracy which is 10^{-3} au per atom.

and 5 DOF per atom (red triangle with solid line), respectively. The element size is fixed to be one unit cell. The buffer size is 0.50, and penalty parameter $\alpha = 20$. The error exhibits stable behavior with respect to the length of the global domain.

2.5.3 Quasi-1D system with random perturbation

The application of the adaptive local basis functions is not restricted to the mono-crystalline systems. It can also be applied to disordered system as well. To elucidate this fact we add a random perturbation uniformly distributed between $[-0.1, 0.1]$ au to each Cartesian component of the atomic positions of the quasi-1D sodium system and silicon system studied above. The global domain is kept the same, and so is

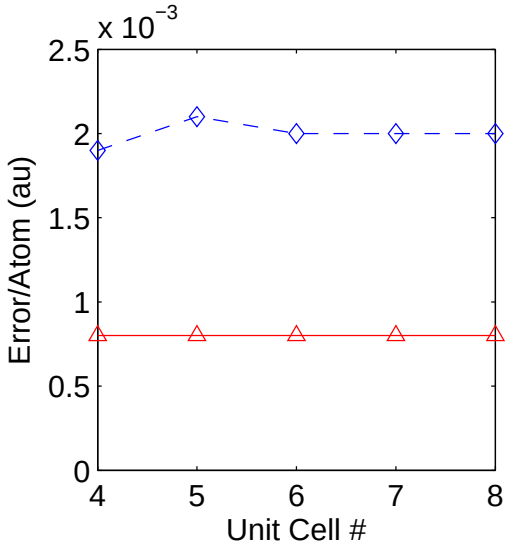


Figure 2.4: The error of the total energy per atom for a quasi-1D sodium system with respect to the length of the global domain along the z direction in Ω . The buffer size is fixed to be 0.50. We present the results with 3 basis functions per atom (blue diamond with dashed line) and 5 basis functions per atom (red triangle with solid line), respectively.

the partition of the elements and the corresponding extended elements. Fig. 2.5 illustrates the error of the total energy per atom with the disordered sodium system (red diamond with solid line) and the disordered silicon system (blue diamond with dashed line), respectively. The buffer size is 0.50 and the penalty parameter $\alpha = 20$. 4 and 6 DOF per atom is needed to reach the target accuracy for Na and Si, respectively. The number of the basis functions is comparable to that presented in Fig. 2.3.

2.5.4 Quasi-2D and 3D Bulk system

Now we study the dimension dependence of the behavior of the error. Our implementation of the DG method is also able to calculate the total energy for the quasi-2D and bulk 3D systems. Fig. 2.6 (a) shows the behavior of the error for a quasi-2D sodium system with the buffer size of 0.50 (red triangle with solid line) and of 1.00 (blue triangle with dashed line), respectively. Fig. 2.6 (b) shows the behavior of the

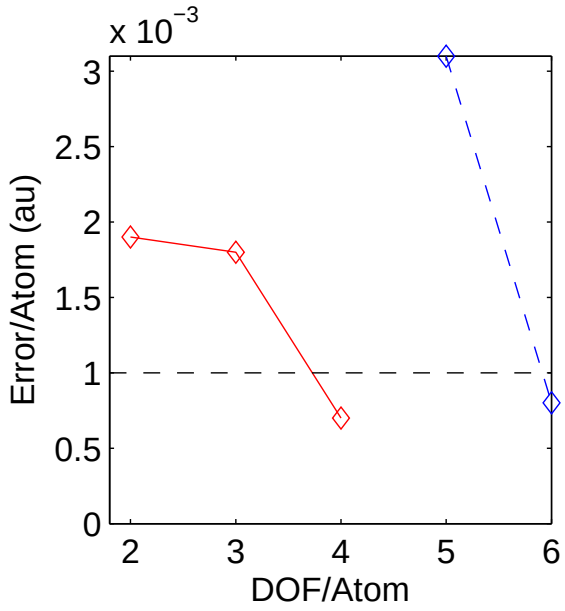
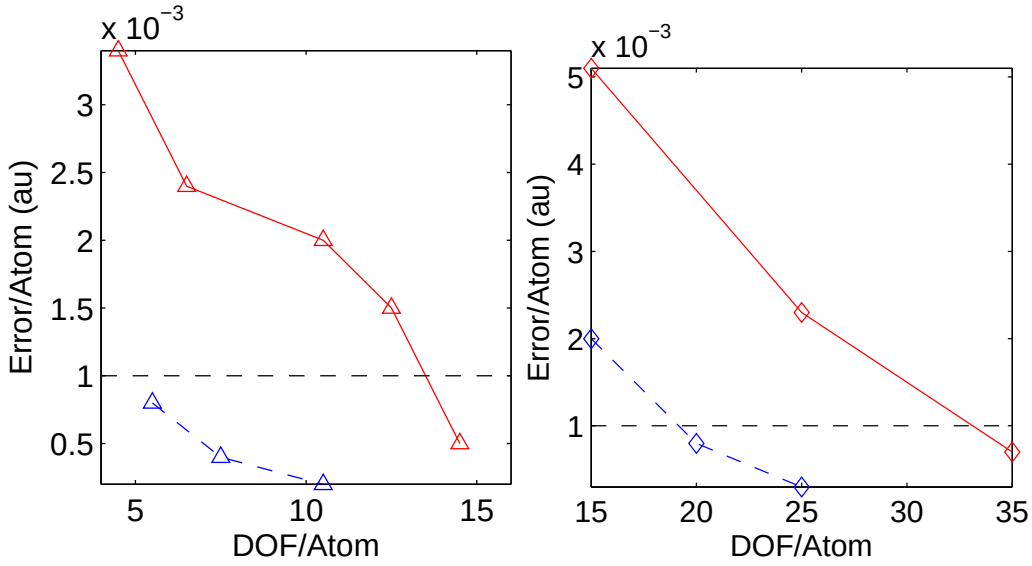


Figure 2.5: The error of the total energy per atom (the y axis) with respect to the number of basis functions per atom (the x axis), for a disordered quasi-1D sodium system (red diamond with solid line) and a disordered quasi-1D silicon system (blue diamond with dashed line). The buffer size is fixed to be 0.50. The black dashed horizontal line refers to the target accuracy which is 10^{-3} au per atom.

error for 3D bulk sodium system using the buffer size of 0.50 (red diamond with solid line) and 1.00 (blue diamond with dashed line), respectively. The buffer area extends beyond the element only along the y and z directions in the quasi-2D case, and the buffer area extends along all the three directions in the bulk 3D case. With increased dimensionality, the number of sodium atoms in each element remains the same, but the number of sodium atoms in the extended element increases with the volume of the buffer area. For example, the numbers of the sodium atoms in the extended element with a buffer size of 1.00 are $4, 18, 54$ for quasi-1D, quasi-2D and 3D bulk systems, respectively. The increased number of atoms in the extended elements leads to more eigenfunctions in the extended elements, and therefore more basis functions per atom in the elements. For a buffer size of 0.50, 15 and 35 basis functions per atom are required to reach target accuracy for the quasi-2D and bulk 3D sodium systems, respectively. By increasing the buffer size to 1.00, the required DOF per atom decreases

to 5 and 20 for the quasi-2D and bulk 3D sodium systems, respectively.



(a) (b)

Figure 2.6: (a) The error of the total energy per atom (the y axis) for a quasi-2D sodium system with respect to the number of basis functions per atom (the x axis). The buffer size is chosen to be 0.50 (red triangle with solid line), and 1.00 (blue triangle with dashed line), respectively. (b) The error of the total energy per atom for a bulk 3D sodium system (the y axis) with respect to the number of basis functions per atom (the x axis). The buffer size is chosen to be 0.50 (red diamond with solid line), and 1.00 (blue diamond with dashed line), respectively. The black dashed horizontal line refers to the target accuracy which is 10^{-3} au per atom.

2.5.5 The penalty parameter

The interior penalty formulation of the discontinuous Galerkin method contains an important parameter α for stability reason. $\alpha = 20$ has been applied uniformly to all the examples studied above. Fig. 2.7 shows the α-dependence of the error of the total energy per atom for the quasi-1D sodium system (red triangle with solid line) and the quasi-1D silicon system (blue diamond with dashed line), respectively. The buffer size is 0.50, and the DOF per atom used is 5 and 6 for sodium and silicon, respectively. There exists a threshold value of α for both sodium and silicon, and in this case the

threshold value of α is slightly below 20. The error increases dramatically if α is below this threshold value, since the inter-element continuity of the Kohn-Sham orbitals is not effectively enforced by the penalty term. After passing this threshold value the error increases much slower, but is still visible especially for very large value of α. This is because the penalty term is included in the variational formulation (2.14) and therefore is also reflected in the eigenvalues and eigenvectors. The rate of increase for the error can be system dependent, and in this case the rate of increase for the error in the silicon system is larger than that in the sodium system. Fig. 2.7 indicates that the penalty parameter α plays an important role in the stability of the algorithm, but the particular choice of the value of α is not crucial. The algorithm is stable with respect to a large range of α values.

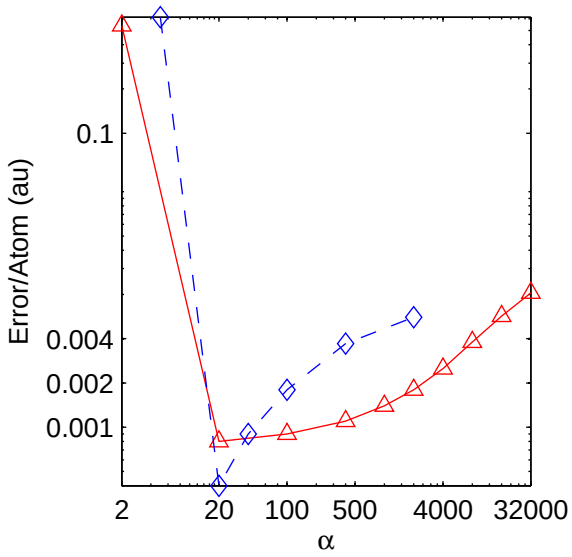


Figure 2.7: The error of the total energy per atom (the y axis) with respect to the penalty parameter α (the x axis), for a quasi-1D sodium system (red triangle with solid line) and a quasi-1D silicon system (blue diamond with dashed line). The number of basis functions per atom for sodium and silicon is 5 and 6, respectively. The buffer size is fixed to be 0.50.

2.5.6 Computational efficiency

The small number of the adaptive basis functions per atom can lead to significant saving of the computational time. We illustrate the efficiency of our algorithm using a bulk 3D mono-crystalline sodium system with the buffer size of 1.00 and with 20 basis functions per atom. Fig. 2.6 suggests that this choice of the parameters leads to the target accuracy. The size of the global domain Ω ranges from $4 \times 4 \times 4$ unit cells with 128 Na atoms to $12 \times 12 \times 12$ unit cells with 3456 atoms. Each element is chosen to be one unit cell. The number of processors used is proportional to the number of unit cells, and $1,728$ processors are used in the problem with $12 \times 12 \times 12$ unit cells. We compare the wall clock time for a single evaluation step of the electron density with a typical number of 10 LOBPCG iterations for solving the adaptive basis functions in the extended elements. Fig. 2.8 compares the wall clock time for solving the adaptive basis functions in the extended elements (blue diamond with dashed line), for solving the DG eigenvalue problem using ScaLAPACK (red triangle with solid line), and for the overhead in the DG method (black circle with dot dashed line). Since both the size of the extended elements and the number of basis functions per atom are fixed, the computational time for solving the adaptive basis functions does not depend on the global domain size. The overhead in the DG method includes SVD filtering of the basis functions, numerical integration, and data communication. All numerical integrations are localized inside each element and its neighboring elements. Our implementation ensures that the data communication is restricted to be within nearest neighboring elements. Therefore the time for the overhead increases mildly with respect to the global system size. For system size smaller than $1,000$ atoms, solving the adaptive local basis functions in the extended elements is more time consuming than the DG eigensolver. The complexity of the DG eigensolver scales cubically with respect to global system size, and starts to dominate the cost of computational time for system size larger than $1,000$ atoms. Since the number of processors is proportional to

the number of elements, the ideal wall clock time for the DG solver scales quadratically with respect to the number of atoms. This quadratic scaling is illustrated by the slope of the small red triangle in Fig. 2.8. Numerical result shows that up to $3,456$ atoms, the performance of ScaLAPACK is still in good correspondence with respect to the ideal scaling. In this case the matrix size of the DG Hamiltonian matrix is $69,120$.

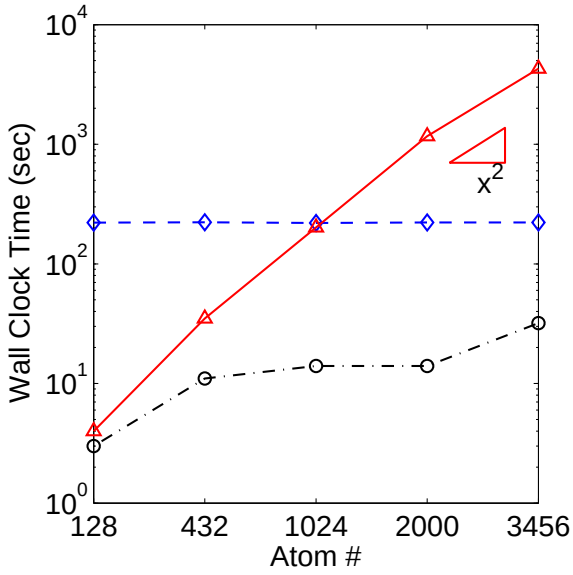


Figure 2.8: The wall clock time for solving the adaptive local basis functions in the extended elements (blue diamond with dashed line), for solving the DG eigenvalue problem using ScaLAPACK (red triangle with solid line), and for the overhead in the DG formalism (black circle with dot dashed line). The x axis is the number of atoms for different bulk 3D sodium systems. The slope of the small red triangle illustrates the ideal quadratic scaling (x^2) for the wall clock time cost for the DG eigenvalue solver in parallel.

The efficiency due to the dimension reduction of the adaptive basis functions can be illustrated by the comparison between the cost of the computational time of the LOBPCG eigensolver directly in the global domain with a planewave basis set (Global) , and that of the DG eigenvalue problem with the adaptive basis functions (DG), as reported in Table 2.1. The global solver uses a typical number of 10 LOBPCG iteration steps. On a single processor, the global solver costs $2,235$ sec for the bulk 3D sodium system with 128 atoms, and $53,395$ sec for the bulk 3D sodium

system with 432 atoms. By assuming that the global solver can be ideally parallelized, the third column of Table 2.1 reports the computational time measured on a single processor divided by the number of processors used in the corresponding DG eigensolver. The fourth column reports the wall clock time for the DG eigensolver executed in parallel. We remark that the computational time for solving the adaptive local basis functions is not taken into account, since we are comparing the saving of the computational time due to the dimension reduction of the basis functions. It is found that the saving of the computational time is already significant even when the system size is relatively small.

Atom#	Proc#	Global (sec)	DG (sec)
128	64	35	4
432	216	248	35

Table 2.1: The comparison of the cost of the computational time using the planewave discretization (the LOBPCG solver directly applied in the global domain) and that using the adaptive local basis functions (the DG eigenvalue solver using ScaLAPACK). The systems under study are the bulk 3D sodium system with $4 \times 4 \times 4$ unit cells (128 Na atoms), and with $6 \times 6 \times 6$ unit cells (432 Na atoms), respectively.

2.6 Conclusion

In this chapter we proposed the adaptive local basis method for discretizing the Kohn-Sham Hamiltonian operator. We demonstrated that the adaptive local basis functions are efficient for calculating the total energy and electron density, and can reach high accuracy (below 10^{-3} Hartree/atom) with complexity comparable to tight binding method. The adaptive local basis functions are discontinuous in the global domain, and the continuous Kohn-Sham orbitals and electron density are reconstructed from these discontinuous basis functions using the discontinuous Galerkin (DG) framework. The environmental effect is automatically built into the basis functions, thanks to the flexibility provided by the DG framework.

In order to generalize the current framework to the force calculation and further to the geometry optimization and the *ab initio* molecular dynamics simulation, the adaptive local basis functions and their derivatives with respect to the positions of the atoms (called Pulay force [213]) should be both accessible. Our preliminary result suggests that the effect of the Pulay force can be systematically reduced. This generalization will be studied in detail in the near future.

The current implementation of the DG method is already able to perform the total energy calculation for systems consisting of thousands of atoms. We are aware of the fact that calculations of this range is already achievable with several existing software packages using plane wave basis functions with iterative methods. However, the performance of the DG method with adaptive local basis functions can be improved by taking into account the block sparsity of the DG stiffness matrix. Furthermore, the local nature of the adaptive basis functions allows us to incorporate the recently developed pole expansion and selected inversion type fast algorithms [164, 169, 170] into the DG framework. The capability of the resulting algorithm is expected to be greatly enhanced compared to the current implementation. This is also within our scope in the near future.

Chapter 3

Representation of the Fermi operator: Pole expansion

3.1 Introduction

In this chapter we study the decomposition of the Fermi operator, which represents the finite temperature density matrix $\boldsymbol{\rho}$:

$$\boldsymbol{\rho} = f(\boldsymbol{H}) = \frac{2}{1+\exp(\beta(\boldsymbol{H}-\mu))} = 1 - \tanh\Big(\frac{\beta}{2}(\boldsymbol{H}-\mu)\Big), \tag{3.1}$$

where Here tanh is the hyperbolic tangent function. As opposed to the formulation (1.34) in Chapter 1, we add the factor 2 in the numerator accounts for the spin degeneracy of the electrons.

The Fermi operator is a complicated matrix-valued function, and cannot be directly computed without further simplification. The Fermi operator expansion framework expands the Fermi operator into a series of simple functions

$$f(\boldsymbol{H}) \approx \sum_{i=1}^{P} f_i(\boldsymbol{H}). \tag{3.2}$$

Each simple function $f_i(\boldsymbol{H})$ is a polynomial or a rational function of $\boldsymbol{H}$, and can be calculated without diagonalization. The number of simple functions (P) to approximate the Fermi operator reflects the **representation cost** of Fermi operator. In practice it is desirable to have the representation cost P as small as possible.

The representation cost of the Fermi operator is a function of $\beta\Delta E$ (for metallic system at finite temperature) or $\Delta E/E_g$ (for insulating systems) [101], where ΔE is the spectral width of the discretized Hamiltonian matrix, and E_g is the spectrum gap of the Hamiltonian around the chemical potential. If polynomials are used to expand the Fermi operator for the metallic system with $E_g = 0$, the representation cost scales as $\mathcal{O}(\beta\Delta E)$. Therefore the number of polynomials P can be thousands or more if the temperature is low or the spectrum width ΔE is large. The representation cost can be reduced by means of rational functions to $\mathcal{O}(\beta\Delta E)^{1/2}$ [53, 199]. However, the optimal representation cost remains unclear.

This dissertation develops two novel strategies, the multipole expansion and the pole expansion, to reduce the computational cost. Both the multipole expansion and the pole expansion techniques reduce the representation cost of Fermi operator down to logarithmic scaling $\mathcal{O}\Big(\ln(\beta\Delta E)\Big)$. Numerical examples show that the logarithmic scaling enables accurate and efficient representation of the Fermi operator even for $\beta\Delta E$ being in the order of millions. Since the scaling of the representation cost is the same in the multipole expansion and in the pole expansion, the difference between the two methods is in the preconstant. Numerical example shows that the preconstant in the pole expansion is smaller than that in the multipole expansion, and therefore the pole expansion is computationally more efficient. Pole expansion will be used in designing accurate and efficient algorithms for the evaluation of the electron density in Chapter 4.

The rest of this chapter is organized as follows. Section 3.2 introduces the multipole expansion, the pole expansion is described in Section 3.3. The relation between

the multipole expansion and the pole expansion is discussed in Section 3.4, followed by the conclusion given in Section 3.5. The multipole expansion and the pole expansion use different mathematical techniques and are both interesting from numerical analysis point of view. Readers who are more interested in the main flow of the new method for solving KSDFT can directly read Section 3.3 and then go to Chapter 4. Materials in this chapter have been presented in [160, 164].

3.2 Multipole expansion

3.2.1 Formulation

The multipole expansion of the Fermi operator starts from the Matsubara representation [176]

$$\boldsymbol{\rho} = 1 - 4\Re \sum_{l=1}^{\infty} \frac{1}{\beta(\boldsymbol{H} - \mu) - (2l-1)\pi i}. \tag{3.3}$$

The summation in (3.3) can be seen as a summation of residues contributed from the poles $\{(2l-1)\pi i\}$, with l a positive integer, on the imaginary axis. This suggests to look for a multipole expansion of the contributions from the poles, as done in the fast multipole method (FMM) [107]. To do so, we use a dyadic decomposition of the poles, in which the n-th group contains terms from $l = 2^{n-1}$ to $l = 2^n - 1$, for a total of 2^{n-1} terms (see Figure 3.1 for illustration). We decompose the summation in Eq.(3.3) accordingly, with $x = \beta(\boldsymbol{H} - \mu)$ for simplicity

$$\sum_{l=1}^{\infty} \frac{1}{x - (2l-1)\pi i} = \sum_{n=1}^{\infty} \sum_{l=2^{n-1}}^{2^n-1} \frac{1}{x - (2l-1)\pi i} = \sum_{n=1}^{\infty} S_n. \tag{3.4}$$

The basic idea is to combine the simple poles into a set of multipoles at $l = l_n$, where l_n is taken as the midpoint of the interval $[2^{n-1}, 2^n - 1]$

$$l_n = \frac{3 \cdot 2^{n-1} - 1}{2}. \tag{3.5}$$

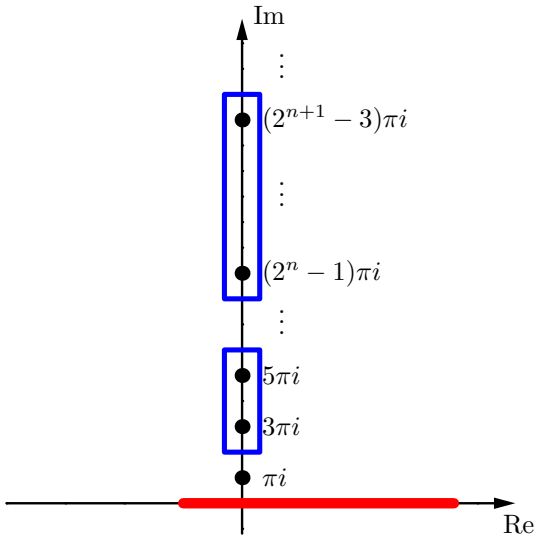


Figure 3.1: Illustration of the pole decomposition (3.12). From 2^n to $2^{n+1}-1$ poles are grouped together as shown in the figure. The spectrum is indicated by the red line on the real axis.

Then the S_n term in the above equation can be written as

$$\begin{aligned} S_n &= \sum_{l=2^{n-1}}^{2^n-1} \frac{1}{x-(2l_n-1)\pi i-2(l-l_n)\pi i} \\ &= \sum_{l=2^{n-1}}^{2^n-1} \frac{1}{x-(2l_n-1)\pi i} \sum_{\nu=0}^{\infty} \Big(\frac{2(l-l_n)\pi i}{x-(2l_n-1)\pi i}\Big)^{\nu} \\ &= \sum_{l=2^{n-1}}^{2^n-1} \frac{1}{x-(2l_n-1)\pi i} \sum_{\nu=0}^{P-1} \Big(\frac{2(l-l_n)\pi i}{x-(2l_n-1)\pi i}\Big)^{\nu} \\ &\qquad + \sum_{l=2^{n-1}}^{2^n-1} \frac{1}{x-(2l-1)\pi i} \Big(\frac{2(l-l_n)\pi i}{x-(2l_n-1)\pi i}\Big)^{P} \end{aligned} \tag{3.6}$$

In deriving Eq. (3.6) we used the result for the summation of a geometric series. Using the fact that x is real, the second term in Eq. (3.6) can be bounded by

$$\sum_{l=2^{n-1}}^{2^n-1} \Big|\frac{1}{x-(2l-1)\pi i}\Big| \Big|\frac{2(l-l_n)\pi i}{x-(2l_n-1)\pi i}\Big|^P \le \sum_{l=2^{n-1}}^{2^n-1} \frac{1}{|(2l-1)\pi|} \Big|\frac{2(l-l_n)}{2l_n-1}\Big|^P \le \frac{1}{2\pi}\frac{1}{3^P} \tag{3.7}$$

Therefore, we can approximate the sum S_n by the first P terms, and the error decays

exponentially with P:

$$\left| S_n(x) - \sum_{l=2^{n-1}}^{2^n-1} \frac{1}{x-(2l_n-1)\pi i} \sum_{\nu=0}^{P-1} \Big(\frac{2(l-l_n)\pi i}{x-(2l_n-1)\pi i} \Big)^\nu \right| \le \frac{1}{2\pi} \frac{1}{3^P}, \tag{3.8}$$

uniformly in x. The overall philosophy here is similar to the fast multipole method [107]: Given a preset error tolerance, one selects P, the number of terms to retain in S_n, according to Eq. (3.8).

Interestingly, the remainder of the summation in Eq. (3.3) from $l = m$ to ∞ has an explicit expression

$$\Re \sum_{l=m}^{\infty} \frac{1}{2x-(2l-1)i\pi} = \frac{1}{2\pi} \Im \psi \left(m - \frac{1}{2} + \frac{i}{\pi} x \right), \tag{3.9}$$

where ψ is the digamma function $\psi(z) = \Gamma'(z)/\Gamma(z)$. It is well known [131] that the digamma function has the following asymptotic expansion

$$\psi(z) \sim \ln(z) - \frac{1}{2z} - \frac{1}{12z^2} + \mathcal{O}\Big(\frac{1}{z^4}\Big), \quad |\arg z| \le \pi \text{ and } |z| \to \infty. \tag{3.10}$$

Therefore,

$$\begin{aligned} \Im \psi \left(m - \frac{1}{2} + \frac{i}{\pi} x \right) &\sim \Im \ln \left(m - \frac{1}{2} - \frac{i}{\pi} x \right) + \mathcal{O}\Big(\frac{1}{m^2}\Big) \\ &= \arctan \left(\frac{2x}{(2m-1)\pi} \right) + \mathcal{O}\Big(\frac{1}{m^2}\Big), \quad m \to \infty. \end{aligned} \tag{3.11}$$

Figure 3.2 shows that the asymptotic approximation (3.11) is already rather accurate when $m = 10$.

Eq. (3.11) also shows the effectiveness of the multipole representation from the viewpoint of traditional polynomial approximations. At zero temperature, the Fermi-Dirac function is a step function that cannot be accurately approximated by any finite order polynomial. At finite but low temperature, it is a continuous function with a

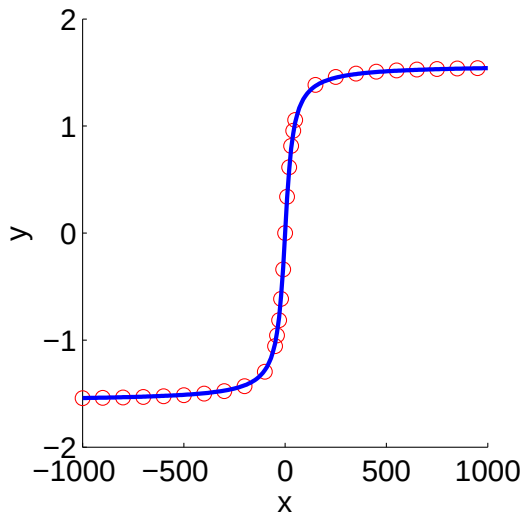


Figure 3.2: The function $\Im\psi\left(m-\frac{1}{2}+\frac{i}{\pi}x\right)$(red circle), *i.e.* the remainder of the pole expansion in Eq. (3.12) is compared with the function $\arctan\left(\frac{2x}{(2m-1)\pi}\right)$ (blue solid line) for $m=10$

very large derivative at $x=0$, *i.e.* when the energy equals the chemical potential μ. The magnitude of this derivative becomes smaller and, correspondingly, the Fermi function becomes smoother as the temperature is raised. One can use the value of the derivative of the Fermi function at $x=0$ to measure the difficulty of an FOE. After eliminating the first m terms in the expansion, Eq. (3.11) shows that asymptotically the derivative is multiplied by the factor $\frac{2}{(2m-1)\pi}$, which is equivalent to a rescaling of the temperature by the same factor. In particular, if we explicitly include the first 2^N terms in the multipole representation of the Fermi operator, we are left with a remainder which is well approximated by Eq. (3.11), so that, effectively, the difficulty is reduced by a factor 2^N. As a matter of fact standard polynomials approximations, such as the Chebyshev expansion, can be used to efficiently represent the remainder in Eq. (3.9) even at very low temperature.

In summary, we arrive at the following multipole representation for the Fermi

operator

$$\boldsymbol{\rho} = 1 - 4\Re \sum_{n=1}^{N} \sum_{l=2^{n-1}}^{2^n-1} \frac{1}{\beta(\boldsymbol{H}-\mu) - (2l_n-1)\pi i} \sum_{\nu=0}^{P-1} \Big(\frac{2(l-l_n)\pi i}{\beta(\boldsymbol{H}-\mu) - (2l_n-1)\pi i} \Big)^{\nu} - \frac{2}{\pi} \Im\psi \left(2^N - \frac{1}{2} + \frac{i}{2\pi}\beta(\boldsymbol{H}-\mu) \right) + \mathcal{O}(N/3^P). \quad (3.12)$$

The multipole part is evaluated directly as discussed below, and the remainder is evaluated with the standard polynomial method.

3.2.2 Numerical calculation and error analysis

To show the power of the multipole expansion, we discuss a possible algorithm to compute the Fermi operator in electronic structure calculations and present a detailed analysis of its cost in terms of $\beta\Delta\epsilon$. Given the Hamiltonian matrix $\boldsymbol{H}$, it is straightforward to compute the density matrix $\boldsymbol{\rho}$ from the multipole expansion if we can calculate the Green's functions $\boldsymbol{B}_{l_n} = [\beta(\boldsymbol{H}-\mu) - (2l_n-1)\pi i]^{-1}$ for different n.

A possible way to calculate the inverse matrices is by the Newton-Schulz iteration. For any non-degenerate matrix $\boldsymbol{A}$, the Newton-Schulz iteration computes the inverse $\boldsymbol{B} = \boldsymbol{A}^{-1}$ as

$$\boldsymbol{B}_{k+1} = 2\boldsymbol{B}_k - \boldsymbol{B}_k\boldsymbol{A}\boldsymbol{B}_k. \quad (3.13)$$

The iteration error is measured by the spectral radius, *i.e.* the eigenvalue of largest magnitude, of the matrix $\boldsymbol{I} - \boldsymbol{A}\boldsymbol{B}_k$ where $\boldsymbol{I}$ is the identity matrix. In the following we denote the spectral radius of the matrix $\boldsymbol{A}$ by $\sigma(\boldsymbol{A})$. Then the spectral radius at the k-th step of the Newton-Schulz iteration is $\boldsymbol{R}_k = \boldsymbol{I} - \boldsymbol{A}\boldsymbol{B}_k$ and

$$\sigma(\boldsymbol{R}_{k+1}) = \sigma(\boldsymbol{R}_k)^2 = \sigma(\boldsymbol{R}_0)^{2^{k+1}}. \quad (3.14)$$

Thus the Newton-Schulz iteration has quadratic convergence. With a proper choice

of the initial guess (see [53]), the number of iterations required to converge is bounded by a constant, and this constant depends only on the target accuracy.

The remainder, *i.e.* the term associated to the digamma function in Eq. (3.12), can be evaluated by standard polynomial approximations such as the Chebyshev expansion. The order of Chebyshev polynomials needed for a given target accuracy is proportional to $\beta\Delta\epsilon/2^{N+1}$ (see [16, Appendix]).

Except for the error coming from the truncated multipole representation, the main source of error in applications comes from the numerical approximation of the Green's functions $\boldsymbol{B}_{l_n}$. To understand the impact of this numerical error on the representation of the Fermi operator, let us rewrite

$$\boldsymbol{S}_n = \sum_{l=2^{n-1}}^{2^n-1} \boldsymbol{B}_{l_n} \sum_{\nu=0}^{P-1} (-2(l-l_n)\pi i \boldsymbol{B}_{l_n})^\nu = \sum_{\nu=0}^{P-1} \boldsymbol{B}_{l_n}^{\nu+1} \sum_{l=2^{n-1}}^{2^n-1} (-2(l-l_n)\pi i)^\nu.$$

The factor $\sum_l (-2(l-l_n)\pi i)^\nu$ is large, but we can control the total error in S_n in terms of the spectral radius $\sigma(\boldsymbol{B}_{l_n} - \widehat{\boldsymbol{B}}_{l_n})$. Here $\widehat{\boldsymbol{B}}_{l_n}$ is the numerical estimate of $\boldsymbol{B}_{l_n}$.

The error is bounded by

$$\sigma(\widehat{\boldsymbol{S}}_n - \boldsymbol{S}_n) \leq \sum_{\nu=0}^{P-1} 2^{n-1}(2^{n-1}\pi)^\nu \sigma\big(\boldsymbol{B}^{\nu+1} - \widehat{\boldsymbol{B}}^{\nu+1}\big) \leq \sum_{\nu=0}^{P-1} (2^{n-1}\pi)^{\nu+1} \sigma\big(\boldsymbol{B}^{\nu+1} - \widehat{\boldsymbol{B}}^{\nu+1}\big), \tag{3.15}$$

where we have omitted the subscript l_n in $\boldsymbol{B}_{l_n}$ and in $\widehat{\boldsymbol{B}}_{l_n}$. In what follows the quantity $\sum_{\nu=0}^{P-1}(2^{n-1}\pi)^{\nu+1}\sigma\big(\boldsymbol{B}^{\nu+1} - \widehat{\boldsymbol{B}}^{\nu+1}\big)$ will be denoted by e_P. Then we have

$$\begin{aligned} e_P &= \sum_{\nu=0}^{P-1} (2^{n-1}\pi)^{\nu+1} \sigma\big((\boldsymbol{B}^\nu - \widehat{\boldsymbol{B}}^\nu)\boldsymbol{B} + (\widehat{\boldsymbol{B}}^\nu - \boldsymbol{B}^\nu)(\boldsymbol{B} - \widehat{\boldsymbol{B}}) + \boldsymbol{B}^\nu(\boldsymbol{B} - \widehat{\boldsymbol{B}})\big) \\ &\leq \sum_{\nu=1}^{P-1} (2^{n-1}\pi)^{\nu+1} \big(\sigma(\boldsymbol{B}) + \sigma(\boldsymbol{B} - \widehat{\boldsymbol{B}})\big)\sigma(\boldsymbol{B}^\nu - \widehat{\boldsymbol{B}}^\nu) + \sum_{\nu=0}^{P-1} (2^{n-1}\pi)^{\nu+1} \sigma(\boldsymbol{B})^\nu \sigma(\boldsymbol{B} - \widehat{\boldsymbol{B}}). \end{aligned} \tag{3.16}$$

Here we took into account the fact that the $\nu = 0$ term in the first summation is equal to zero, and have used the properties $\sigma(\boldsymbol{A}+\boldsymbol{B}) \leq \sigma(\boldsymbol{A}) + \sigma(\boldsymbol{B})$, and $\sigma(\boldsymbol{AB}) \leq$

$\sigma(\boldsymbol{A})\sigma(\boldsymbol{B})$, respectively.

Noting that $2^{n-1}\pi\sigma(\boldsymbol{B}_{l_n}) \leq 1/3$ and changing ν to $\nu+1$ in the first summation, we can rewrite e_P as

$$\begin{aligned} e_P &\leq \big(\frac{1}{3} + 2^{n-1}\pi\sigma(\boldsymbol{B}-\widehat{\boldsymbol{B}})\big)\sum_{\nu=0}^{P-2}(2^{n-1}\pi)^{\nu+1}\sigma(\boldsymbol{B}^{\nu+1}-\widehat{\boldsymbol{B}}^{\nu+1}) + \sum_{\nu=0}^{P-1}\frac{1}{3^\nu}2^{n-1}\pi\sigma(\boldsymbol{B}-\widehat{\boldsymbol{B}}) \\ &\leq (\frac{1}{3} + 2^{n-1}\pi\sigma(\boldsymbol{B}-\widehat{\boldsymbol{B}}))e_{P-1} + \frac{3}{2}2^{n-1}\pi\sigma(\boldsymbol{B}-\widehat{\boldsymbol{B}}) \\ &= (\frac{1}{3}+e_1)e_{P-1} + \frac{3}{2}e_1. \end{aligned} \tag{3.17}$$

In the last equality, we used the fact that $e_1 = 2^{n-1}\pi\sigma(\boldsymbol{B}-\widehat{\boldsymbol{B}})$. Therefore, the error e_P satisfies the following recursion formula

$$e_P + \frac{3e_1/2}{e_1-2/3} \leq (\frac{1}{3}+e_1)\left(e_1 + \frac{3e_1/2}{e_{P-1}-2/3}\right) \leq (\frac{1}{3}+e_1)^{P-1}\left(e_1+\frac{3e_1/2}{e_1-2/3}\right). \tag{3.18}$$

Taking $e_1 \leq \frac{2}{3}$, we have

$$e_P \leq e_1 = 2^{n-1}\pi\sigma(\boldsymbol{B}-\widehat{\boldsymbol{B}}). \tag{3.19}$$

Therefore, using Eq. (3.14) we find that the number k of Newton-Schulz iterations must be bounded as dictated by the following inequality in order for the error $\sigma(\widehat{\boldsymbol{S}}_n - \boldsymbol{S}_n)$ to be $\leq 10^{-D}/N$.

$$2^{n-1}\sigma(\boldsymbol{R}_0)^{2^k} \leq \frac{10^{-D}}{N}. \tag{3.20}$$

Here we have used the fact that $\sigma(\boldsymbol{B}_{l_n}) \leq 1/\pi$ for any n. Each Newton-Schulz iteration requires two matrix by matrix multiplications, and the number of matrix by matrix multiplications needed in the Newton-Schulz iteration for $\boldsymbol{B}_{l_n}$ with $n < N$ is bounded by

$$2\log_2\left(\frac{D\log_2 10 + N + \log_2 N}{-\log_2\sigma(\boldsymbol{R}_0)}\right). \tag{3.21}$$

To obtain a target accuracy $\sigma(\boldsymbol{\rho}-\widehat{\boldsymbol{\rho}}) \leq 10^{-D}$ for a numerical estimate $\widehat{\boldsymbol{\rho}}$ of the

density matrix, taking into account the operational cost of calculating the remainder and the direct multipole summation in the FOE, the number of matrix by matrix multiplications n_{MM} is bounded by

$$n_{\mathrm{MM}} \leq 2N \log_2 N + C_1 N + C_2 2^{-N-1} \beta \Delta\epsilon. \tag{3.22}$$

Here we used the property: $\log_2(x+y) \leq \log_2 x + \log_2 y$ when $x, y \geq 2$, and defined the constant C_1 as follows:

$$C_1 = \frac{2}{N} \sum_{n=1}^{N} \log_2 \left(\frac{D \log_2 10 + \log_2 N}{-\log_2 \sigma\big((\boldsymbol{R}_0)_{l_n}\big)} \right). \tag{3.23}$$

The dependence on $2^{-N-1}\beta\Delta\epsilon$ in the last term on the right hand side of (3.22) comes from Chebyshev expansion used to calculate the remainder. From numerical calculations on model systems, the constant C_1 and C_2 will be shown to be rather small. Finally, choosing $N \propto \ln(\beta\Delta\epsilon)$, we obtain

$$n_{\mathrm{MM}} \propto (\ln \beta\Delta\epsilon) \cdot (\ln \ln \beta\Delta\epsilon) \tag{3.24}$$

with a small prefactor.

3.2.3 Numerical examples

We illustrate the algorithm in three simple cases. The first is an off-lattice one dimensional model defined in a supercell with periodic boundary conditions. In this example, we discretize the Hamiltonian with the finite difference method, resulting in a very broad spectrum with a width of about 2000eV, and we choose a temperature as low as 32K. In the second example we consider a nearest neighbor tight binding Hamiltonian in a three dimensional simple cubic lattice and set the temperature to 100K. In the third example we consider a three dimensional Anderson model with

random on-site energy on a simple cubic lattice at 100K.

One dimensional model with large spectral width

In this example, a one dimensional crystal is described by a periodic supercell with 10 atoms, evenly spaced. We take the distance between adjacent atoms to be $a = 5.29$Å. The one-particle Hamiltonian is given by

$$\boldsymbol{H} = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + V. \tag{3.25}$$

The potential V is given by a sum of Gaussians centered at the atoms with width $\sigma = 1.32$Å and depth $V_0 = 13.6$eV. The kinetic energy is discretized using a simple 3-point finite difference formula, resulting in a Hamiltonian $\boldsymbol{H}$ with a discrete eigenvalue spectrum with lower and upper eigenvalues equal to $\epsilon_- = 6.76$eV and $\epsilon_+ = 1959$eV, respectively. Various temperatures from 1024K to 32K were tried. Figure 3.3 reports the linear-log graph of n_{MM}, the number of matrix by matrix multiplications needed to evaluate the density matrix using our FOE, versus $\beta\Delta\epsilon$, with $\beta\Delta\epsilon$ plotted in a logarithmic scale. The logarithmic dependence can be clearly seen. The prefactor of the logarithmic dependence is rather small: when $\beta\Delta\epsilon$ is doubled, a number of additional matrix multiplications equal to 17 is required to achieve two-digit accuracy ($D = 2$), a number equal to 19 is needed for $D = 4$, and a number equal to 21 is needed for $D = 6$, respectively. The observed D-dependence of the number of matrix multiplications agrees well with the prediction in (3.22).

In order to assess the validity of the criterion for the number of matrix multiplications given in Eq. (23), we report in Table 3.1 the calculated relative energy error and relative density error, respectively, at different temperatures, when the number of matrix multiplications is bounded as in formula (23) using different values for D. The relative energy error, $\Delta\epsilon_{\mathrm{rel}}$, measures the accuracy in the calculation of the total

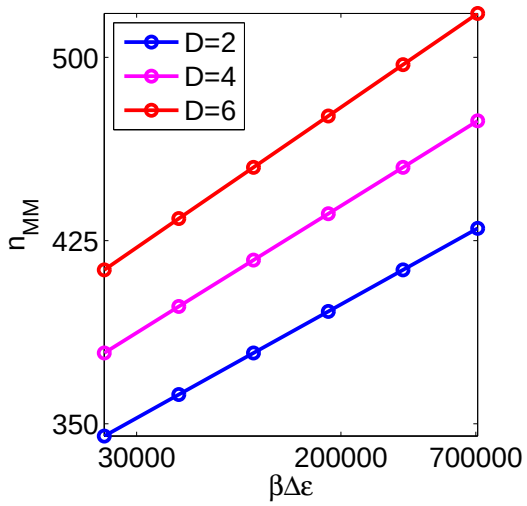


Figure 3.3: Linear-log plot of the number of matrix matrix multiplications n_{MM} versus $\beta\Delta\epsilon$. n_{MM} depends logarithmically on $\beta\Delta\epsilon$ with a small constant prefactor.

electronic energy corresponding to the supercell $E = \mathrm{Tr}(\boldsymbol{\rho}\boldsymbol{H})$. It is defined as

$$\Delta\epsilon_{\mathrm{rel}} = \frac{|\hat{E} - E|}{|E|}. \tag{3.26}$$

Similarly the relative L^1 error in the density function in real space is defined as

$$\Delta\rho_{\mathrm{rel}} = \frac{\mathrm{Tr}\,|\hat{\boldsymbol{\rho}} - \boldsymbol{\rho}|}{\mathrm{Tr}\,\boldsymbol{\rho}}. \tag{3.27}$$

Because $\mathrm{Tr}\,\boldsymbol{\rho} = N_{\mathrm{e}}$, where N_{e} is the total number of electrons in the supercell, $\Delta\rho_{\mathrm{rel}}$ is the same as the L^1 density error per electron. Table 3.1 shows that for all the values of $\beta\Delta\epsilon$, our algorithm gives a numerical accuracy that is even better than the target accuracy D. This is not surprising because our theoretical analysis was based on the most conservative error estimates.

Periodic three dimensional tight-binding model

In this example we consider a periodic three dimensional single-band tight-binding Hamiltonian in a simple cubic lattice. The Hamiltonian, which can be viewed as the discretized form of a free-particle Hamiltonian, is given in second quantized notation

T	$\beta\Delta\epsilon$	$\Delta\epsilon_{\rm rel}$		
		$D=2$	$D=4$	$D=6$
$1024K$	2.22×10^4	1.64×10^{-3}	5.98×10^{-6}	3.31×10^{-8}
$512K$	4.44×10^4	1.73×10^{-3}	6.49×10^{-6}	3.70×10^{-8}
$256K$	8.89×10^4	1.78×10^{-3}	6.83×10^{-6}	3.96×10^{-8}
$128K$	1.78×10^5	1.74×10^{-3}	6.55×10^{-6}	3.75×10^{-8}
$64K$	3.56×10^5	1.75×10^{-3}	6.62×10^{-6}	3.80×10^{-8}
$32K$	7.12×10^5	1.76×10^{-3}	6.66×10^{-6}	3.82×10^{-8}
T	$\beta\Delta\epsilon$	$\Delta\rho_{\rm rel}$		
		$D=2$	$D=4$	$D=6$
$1024K$	2.22×10^4	4.21×10^{-4}	2.23×10^{-6}	1.50×10^{-8}
$512K$	4.44×10^4	4.63×10^{-4}	2.52×10^{-6}	1.74×10^{-8}
$256K$	8.89×10^4	4.77×10^{-4}	2.62×10^{-6}	1.81×10^{-8}
$128K$	1.78×10^5	5.04×10^{-4}	2.80×10^{-6}	1.95×10^{-8}
$64K$	3.56×10^5	4.92×10^{-4}	2.70×10^{-6}	1.86×10^{-8}
$32K$	7.12×10^5	4.84×10^{-4}	2.64×10^{-6}	1.80×10^{-8}

Table 3.1: One dimensional Hamiltonian model with large spectral gap. Relative energy error $\Delta\epsilon_{\rm rel}$ and relative L^1 density error $\Delta\rho_{\rm rel}$ for a large range of values of $\beta\Delta\epsilon$ and several values of D.

by:

$$\boldsymbol{H} = -t \sum_{<i,j>} c_i^+ c_j, \tag{3.28}$$

where the sum includes the nearest neighbors only. Choosing a value of 2.27eV for the hopping parameter t the band extrema occur at $\epsilon_+ = 13.606$eV, and at $\epsilon_- = -13.606$eV, respectively. In the numerical calculation we consider a periodically repeated supercell with 1000 sites and chose a value of $100K$ for the temperature. Table 3.2 shows the dependence of $n_{\rm MM}$, $\Delta\epsilon_{\rm rel}$, and $\Delta\rho_{\rm rel}$ on the chemical potential μ, for different D choices. Compared to the previous one dimensional example in which $\beta\Delta\epsilon$ was as large as 7.12×10^5, here $\beta\Delta\epsilon = 1600$ due to the much smaller spectral width of the tight-binding Hamiltonian. When $\mu = 0$ the chemical potential lies exactly in the middle of the spectrum. This symmetry leads to a relative error as low as 10^{-19} for the density function.

μ	$D=4$			$D=8$		
	n_{MM}	$\Delta\epsilon_{\mathrm{rel}}$	$\Delta\rho_{\mathrm{rel}}$	n_{MM}	$\Delta\epsilon_{\mathrm{rel}}$	$\Delta\rho_{\mathrm{rel}}$
-10.88eV	320	4.09×10^{-9}	2.31×10^{-10}	376	2.27×10^{-13}	2.37×10^{-14}
-5.44eV	308	1.48×10^{-9}	3.15×10^{-11}	356	4.77×10^{-13}	2.52×10^{-15}
0.00eV	305	1.55×10^{-9}	6.26×10^{-19}	357	2.98×10^{-15}	6.26×10^{-19}
5.44eV	308	1.45×10^{-8}	1.34×10^{-12}	356	5.36×10^{-13}	1.07×10^{-16}
10.88eV	320	1.69×10^{-8}	1.78×10^{-13}	376	1.09×10^{-12}	1.80×10^{-17}

Table 3.2: Three dimensional periodic tight binding model. Number of matrix matrix multiplications n_{MM}, relative energy error $\Delta\epsilon_{\mathrm{rel}}$, and relative L^1 density error $\Delta\rho_{\mathrm{rel}}$. For $\mu = 0$, the algorithm achieves machine accuracy for the absolute error of the density function as a consequence of symmetry.

Three dimensional disordered Anderson model

In this example we consider an Anderson model with on-site disorder on a simple cubic lattice. The Hamiltonian is given by

$$\boldsymbol{H} = -t \sum_{<i,j>} c_i^+ c_j + \sum_i \epsilon_i c_i^+ c_i. \tag{3.29}$$

This Hamiltonian contains random on-site energies ϵ_i uniformly distributed in the interval $[-1.13\text{eV}, 1.13\text{eV}]$, and we use the same hopping parameter t as in the previous (ordered) example. In the numerical calculation we consider, as before, a supercell with 1000 sites with periodic boundary conditions, and choose again a temperature of 100K. In one realization of disorder corresponding to a particular set of random on-site energies, the spectrum has extrema at $\epsilon_+ = 13.619$eV and at $\epsilon_- = -13.676$eV. The effect of disorder on the density function is remarkable: while in the periodic tight-binding case the density was uniform, having the same constant value at all the lattice sites, now the density is a random function in the lattice sites within the supercell. Table 3.3 reports for the disordered model the same data that were reported in Table 3.2 for the ordered model. We see that the accuracy of our numerical FOE is the same in the two cases, irrespective of disorder. The only difference is that the super convergence due to symmetry for $\mu = 0$ no longer exists in the disordered case.

μ	$D=4$			$D=8$		
	n_{MM}	$\Delta\epsilon_{\mathrm{rel}}$	$\Delta\rho_{\mathrm{rel}}$	n_{MM}	$\Delta\epsilon_{\mathrm{rel}}$	$\Delta\rho_{\mathrm{rel}}$
-10.88eV	320	5.16×10^{-9}	1.72×10^{-10}	376	3.16×10^{-13}	2.59×10^{-14}
-5.44eV	308	4.75×10^{-9}	2.43×10^{-11}	356	3.71×10^{-13}	1.48×10^{-15}
0.00eV	305	8.08×10^{-10}	9.50×10^{-13}	357	1.76×10^{-14}	2.39×10^{-17}
5.44eV	308	1.01×10^{-8}	1.22×10^{-12}	356	3.57×10^{-13}	8.05×10^{-17}
10.88eV	320	1.30×10^{-8}	1.56×10^{-13}	376	9.56×10^{-13}	1.83×10^{-17}

Table 3.3: Three dimensional Anderson model with on-site disorder. Number of matrix matrix multiplications n_{MM}, relative energy error $\Delta\epsilon_{\mathrm{rel}}$, and relative L^1 density error $\Delta\rho_{\mathrm{rel}}$.

3.3 Pole expansion

3.3.1 Pole expansion: basic idea

Efficient representation of the Fermi-Dirac function can be achieved alternatively by using the discretized contour integral in the complex plane:

$$\begin{aligned} f(x) =& \frac{1}{2\pi i}\int_{\Gamma} f(z)(z-x)^{-1}\,\mathrm{d}z, \\ \approx& \sum_{i=1}^{P}\frac{\omega_i}{x-z_i}, \quad x\in\mathbb{R}, z_i\in\mathbb{Z}, \omega_i\in\mathbb{Z}. \end{aligned} \tag{3.30}$$

Here $\{z_i\}$ are the quadrature points on the complex contour Γ, and $\{\omega_i\}$ are the quadrature weights. Each point z_i is a single-pole on the complex plane, and Eq. (3.30) is referred to as the *pole expansion* in the following discussion. The advantage of the pole expansion is that when substituting $\boldsymbol{H}$ for x in Eq. (3.30), each term $\frac{\omega_i}{\boldsymbol{H}-z_i}$ only involves matrix-inversion but not matrix-matrix multiplication as in the multipole expansion. As shall be seen in Chapter 4, pole expansion (3.30) allows the development of accurate and efficient algorithm for solving KSDFT. The representation cost of the pole expansion developed in this section scales as $\mathcal{O}(\log(\beta\Delta E))$. The mathematical technique used in the pole expansion originates from the idea in [118].

Let us first briefly recall the main idea of [118]. Consider a function f that is

analytic in $\mathcal{C}\backslash(-\infty, 0]$ and an operator $\boldsymbol{A}$ with spectrum in $[m, M] \subset \mathbb{R}^+$, one wants to evaluate $f(\boldsymbol{A})$ using a rational expansion of f by discretizing the contour integral

$$f(\boldsymbol{A}) = \frac{1}{2\pi i} \int_\Gamma f(z)(z - \boldsymbol{A})^{-1} \, \mathrm{d}z. \tag{3.31}$$

The innovative technique in [118] was to construct a conformal map that maps the stripe $S = [-K, K] \times [0, K']$ to the upper half (denoted as Ω^+) of the domain $\Omega = \mathcal{C}\backslash\big((-\infty, 0] \cup [m, M]\big)$. This special map from $t \in S$ to $z \in \Omega^+$ is given by

$$z = \sqrt{mM}\Big(\frac{k^{-1} + u}{k^{-1} - u}\Big), \quad u = \mathrm{sn}(t) = \mathrm{sn}(t|k), \quad k = \frac{\sqrt{M/m} - 1}{\sqrt{M/m} + 1}. \tag{3.32}$$

Here $\mathrm{sn}(t)$ is one of the Jacobi elliptic functions and the numbers K and K' are complete elliptic integrals whose values are given by the condition that the map is from S to Ω^+.

Applying the trapezoidal rule with Q equally spaced points in $(-K + iK'/2, K + iK'/2)$,

$$t_j = -K + \frac{iK'}{2} + 2\frac{(j - \frac{1}{2})K}{Q}, \quad 1 \le j \le Q, \tag{3.33}$$

we get the quadrature rule (denote $z_j = z(t_j)$)

$$f_Q(\boldsymbol{A}) = \frac{-4K\sqrt{mM}}{\pi Q k} \Im \sum_{j=1}^{Q} \frac{f(z_j)(z_j - \boldsymbol{A})^{-1} \, \mathrm{cn}(t_j) \, \mathrm{dn}(t_j)}{(k^{-1} - \mathrm{sn}(t_j))^2}. \tag{3.34}$$

Here cn and dn are the other two Jacobi elliptic functions in standard notation and the factor $\mathrm{cn}(t_j) \, \mathrm{dn}(t_j)(k^{-1} - \mathrm{sn}(t_j))^{-2}\sqrt{mM}/k$ comes from the Jacobian of the function $z(t)$.

It is proved in [118] that the convergence is exponential in the number of quadrature points Q and the exponent deteriorates only logarithmically as $M/m \to \infty$:

$$\|f(\boldsymbol{A}) - f_Q(\boldsymbol{A})\| = \mathcal{O}(e^{-\pi^2 Q/(\log(M/m)+3)}). \tag{3.35}$$

To adapt the idea to our setting with the Fermi-Dirac function or the hyperbolic tangent function, we face with two differences: First, the tanh function has singularities on the imaginary axis. Second, the operator we are considering, $\beta(\boldsymbol{H}-\mu)$, has spectrum on both the negative and positive axis.

3.3.2 Gapped case: insulating system

We first consider the case when the Hamiltonian $\boldsymbol{H}$ has a gap in its spectrum around the chemical potential μ, such that $\mathrm{dist}(\mu, \sigma(\boldsymbol{H})) = E_g > 0$. Physically, this will be the case when the system is an insulator.

Let us consider $f(z) = \tanh(\frac{\beta}{2} z^{1/2})$ acting on the operator $\boldsymbol{A} = (\boldsymbol{H}-\mu)^2$. Now, $f(z)$ has singularities only on $(-\infty, 0]$ and the spectrum of $\boldsymbol{A}$ is contained in $[E_g^2, E_M^2]$, where

$$E_M = \max_{E\in\sigma(\boldsymbol{H})} |E-\mu|.$$

We note that obviously $E_M \leq \Delta E$. Hence we are back in the same scenario as considered in [118] except that we need to take care of different branches of the square root function when we apply the quadrature rule.

More specifically, we construct the contour and quadrature points z_j in the z-plane using parameters $m = E_g^2$ and $M = E_M^2$. Denote $g(\xi) = \tanh(\beta\xi/2)$, $\xi_j^{\pm} = \pm z_j^{1/2}$, and $\boldsymbol{B} = \boldsymbol{H} - \mu$. The quadrature rule is then given by

$$g_Q(\boldsymbol{B}) = \frac{-2K\sqrt{mM}}{\pi Q k} \Im \left(\sum_{j=1}^{Q} \frac{g(\xi_j^+)(\xi_j^+ - \boldsymbol{B})^{-1} \,\mathrm{cn}(t_j)\,\mathrm{dn}(t_j)}{\xi_j^+ (k^{-1} - \mathrm{sn}(t_j))^2} + \sum_{j=1}^{Q} \frac{g(\xi_j^-)(\xi_j^- - \boldsymbol{B})^{-1} \,\mathrm{cn}(t_j)\,\mathrm{dn}(t_j)}{\xi_j^- (k^{-1} - \mathrm{sn}(t_j))^2} \right), \quad (3.36)$$

where the factors $\xi_j^{\pm}$ in the denominator come from the Jacobian of the map from z to ξ. The number of poles to be inverted is $N_{\mathrm{pole}} = 2Q$. After applying (3.35), we

have a similar error estimate for $g(\boldsymbol{B})$

$$\|g(\boldsymbol{B}) - g_Q(\boldsymbol{B})\| = \mathcal{O}(e^{-\pi^2 Q/(2\log(E_M/E_g)+3)}). \tag{3.37}$$

In Fig. 3.4, a typical configuration of the quadrature points is shown. The x-axis is taken to be $E - \mu$. We see that in this case the contour consists of two loops, one around the spectrum below the chemical potential and the other around the spectrum above the chemical potential.

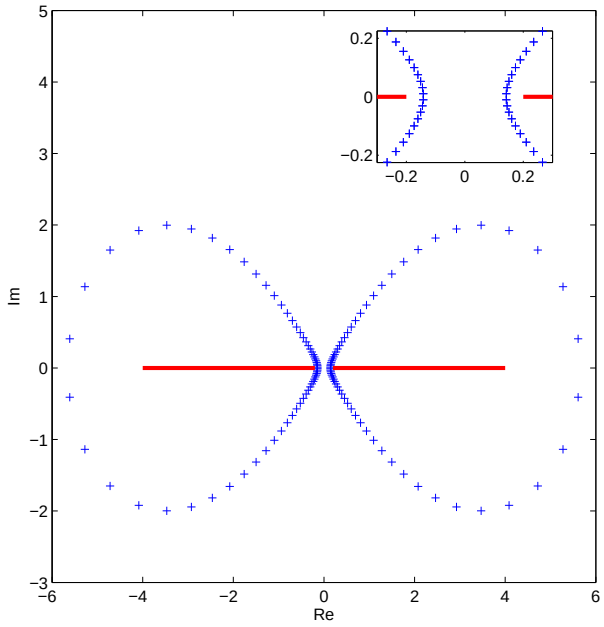


Figure 3.4: A typical configuration of the poles on a two-loop contour. $Q = 30$, $E_g = 0.2$, $E_M = 4$ and $\beta = 1000$. The red line indicates the spectrum. The inset shows the poles close to the origin. The x-axis is $E - \mu$ with E the eigenvalue of $\boldsymbol{H}$. The poles with negative imaginary parts are not explicitly calculated.

Note further that as the temperature goes to zero, the Fermi-Dirac function converges to the step function:

$$\eta(\xi) = \begin{cases} 2, & \xi \leq 0, \\ 0, & \xi > 0. \end{cases} \tag{3.38}$$

Therefore, the contribution of the quadrature points ξ_j^+ on the right half plane ($\Re\xi_j^+ >$

0) is negligible when β is large. In particular, for the case of zero temperature, one may choose only the quadrature points on the left half plane. The quadrature formula we obtain then becomes

$$\eta_Q(\boldsymbol{B}) = \frac{-4K\sqrt{mM}}{\pi Q k} \Im\left(\sum_{j=1}^{Q} \frac{(\xi_j^- - \boldsymbol{B})^{-1} \operatorname{cn}(t_j) \operatorname{dn}(t_j)}{\xi_j^- (k^{-1} - \operatorname{sn}(t_j))^2} \right). \tag{3.39}$$

The number of poles to be inverted is then $N_{\text{pole}} = Q$.

We show in Fig. 3.5 a typical configuration of the set of quadrature points. Only one loop is required compared with Fig. 3.4.

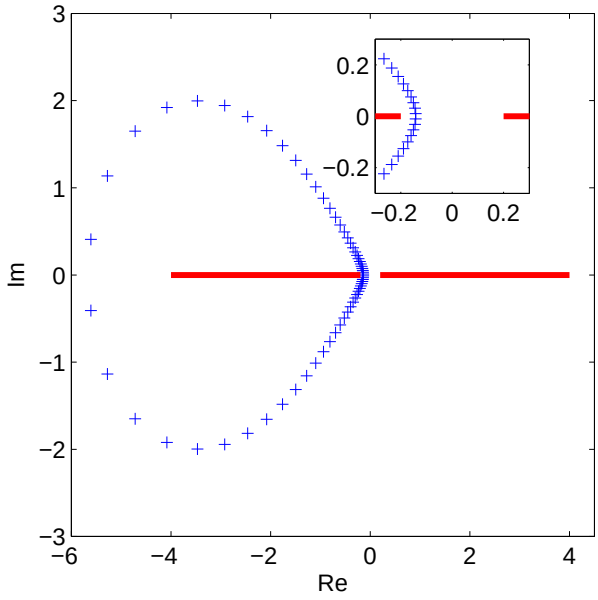


Figure 3.5: A typical configuration of the poles for zero temperature ($\beta = \infty$). $Q = 30$, $E_g = 0.2$ and $E_M = 4$. The red line indicates the spectrum. The inset zooms into the poles that is close to the origin. The x-axis is $E - \mu$ with E the eigenvalue of $\boldsymbol{H}$. The poles with negative imaginary parts are not explicitly calculated.

3.3.3 Gapless case: metallic system

The more challenging case is when the spectrum of $\boldsymbol{H}$ does not have a gap, *i.e.* $E_g = 0$. Physically, this corresponds to the case of metallic systems. In this case, the

construction discussed in the last subsection does not work.

To overcome this problem, we note that the hyperbolic tangent function $\tanh(\frac{\beta}{2}z)$ is analytic except at poles $(2l-1)\pi/\beta i,\ l \in \mathbb{Z}$ on the imaginary axis. Therefore, we could construct a contour around the whole spectrum of $\boldsymbol{H}$ which passes through the imaginary axis on the upper half plane between the origin and $\pi/\beta i$ and also on the lower half plane between the origin and $-\pi/\beta i$. Thus, we will have a dumbbell shaped contour as shown in Fig. 3.6.

To be more specific, let us first construct the contour and quadrature points z_j in the z-plane as in the last subsection using parameters $m = \pi^2/\beta^2$ and $M = E_M^2 + \pi^2/\beta^2$. Denote $\xi_j^{\pm} = \pm(z_j - \pi^2/\beta^2)^{1/2}$, $g = \tanh(\beta\xi/2)$ and $\boldsymbol{B} = \boldsymbol{H} - \mu$. The quadrature rule takes the following form

$$g_Q(\boldsymbol{B}) = \frac{-2K\sqrt{mM}}{\pi Q k}\Im\left(\sum_{j=1}^{Q}\frac{g(\xi_j^+)(\xi_j^+ - \boldsymbol{B})^{-1}\,\mathrm{cn}(t_j)\,\mathrm{dn}(t_j)}{\xi_j^+(k^{-1}-\mathrm{sn}(t_j))^2} + \sum_{j=1}^{Q}\frac{g(\xi_j^-)(\xi_j^- - \boldsymbol{B})^{-1}\,\mathrm{cn}(t_j)\,\mathrm{dn}(t_j)}{\xi_j^-(k^{-1}-\mathrm{sn}(t_j))^2}\right). \quad (3.40)$$

When apply the quadrature formula, the number of poles to be inverted is $N_{\mathrm{pole}} = 2Q$. Fig. 3.6 shows a typical configuration of quadrature points for $Q = 30$. The map $\xi(z) = (z - \pi^2/\beta^2)^{1/2}$ maps the circle in the z-plane to a dumbbell-shaped contour (put two branches together).

Actually, what is done could be understood as follows. Similar to [118], we have constructed a map from the rectangular domain $[-3K, K] \times [0, K']$ to the upper half of the domain

$$U = \{z \mid \Im z \geq 0\}\backslash\big([-E_M, E_M] \cup i[\pi/\beta, \infty)\big).$$

The map is carried out in three steps, shown in Fig. 3.7. The first two steps use the original map constructed in [118], however with extended domain of definition. First,

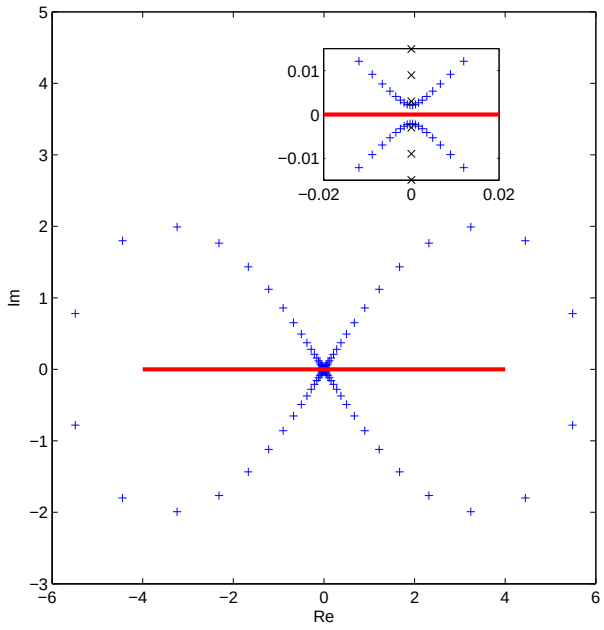


Figure 3.6: A typical configuration of the poles on a dumbbell-shaped contour. $Q = 30$, $E_g = 0$, $E_M = 4$ and $\beta = 1000$. The inset zooms into the part close to the origin. The red line indicates the spectrum. The black crosses indicate the positions of the poles of tanh function on the imaginary axis. The poles with negative imaginary parts are not explicitly calculated.

the Jacobi elliptic function

$$u = \mathrm{sn}(t) = \mathrm{sn}(t|k), \quad k = \frac{\sqrt{M/m} - 1}{\sqrt{M/m} + 1} \tag{3.41}$$

maps the rectangular domain to the complex plane, with the ends mapping to $[1, k^{-1}]$ and the middle vertical line $-K + i[0, K']$ to $[-k^{-1}, -1]$. Then, the Möbius transformation

$$z = \sqrt{mM}\left(\frac{k^{-1} + u}{k^{-1} - u}\right) \tag{3.42}$$

maps the complex plane to itself in such a way that $[-k^{-1}, -1]$ and $[1, k^{-1}]$ are mapped to $[0, m]$ and $[M, \infty]$, respectively. Finally, the shifted square root function

$$\xi = (z - m)^{1/2} \tag{3.43}$$

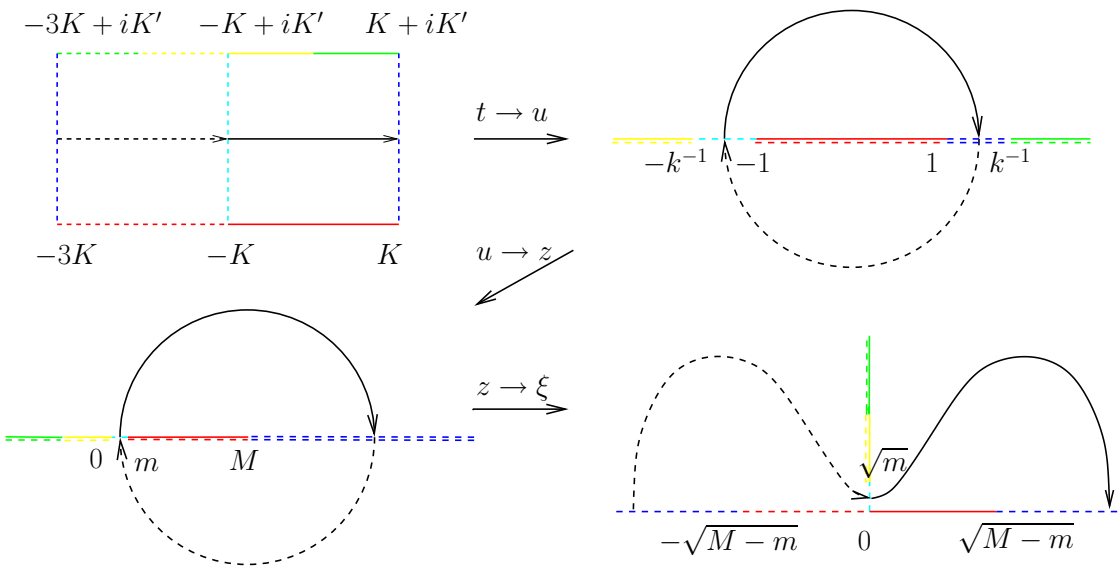


Figure 3.7: The map from the rectangular domain $[-3K, K] \times [0, K']$ to the upper-half of the domain U. The map is constructed in three steps: $t \to u \to z \to \xi$. The boundaries are shown in various colors and line styles.

maps the complex plane to the upper-half plane (we choose the branch of the square root such that the lower-half plane is mapped to the second quadrant and the upper-half plane is mapped to the first quadrant), in such a way that $[0, m]$ is sent to $i[0, \sqrt{m}]$ and $[M, \infty)$ is sent to $(-\infty, -\sqrt{M-m}] \cup [\sqrt{M-m}, \infty)$. The map can be extended to a map from $[-7K, K] \times [0, K']$ to the whole U, in this case, the z-plane becomes a double-covered Riemann surface with branch point at m.

Since the function g is analytic in the domain U, the composite function $g(t) = g(\xi(z(u(t))))$ is analytic in the stripe in the t-plane, and therefore, the trapezoidal rule converges exponentially fast. Using a similar analysis that leads to (3.35), it can be shown that

$$\|g(\boldsymbol{B}) - g_Q(\boldsymbol{B})\| = \mathcal{O}(e^{-CQ/\log(\beta E_M)}), \tag{3.44}$$

where C is a constant.

We remark that the construction proposed in this subsection also applies to the gapped case. In practice, if the temperature is high (so that β is small) or the gap

around the chemical potential is small (in particular, for gapless system), the contour passing through the imaginary axis will be favorable; otherwise, the construction in the last subsection will be more efficient.

3.3.4 Numerical examples

We test the pole expansion described above using a two dimensional nearest neighbor tight binding model for the Hamiltonian. The matrix components of the Hamiltonian can be written as (in atomic units),

$$H_{i'j';ij} = \begin{cases} 2 + V_{ij}, & i' = i, j' = j, \\ -1/2 + V_{ij}, & i' = i \pm 1, j' = j \text{ or } i' = i, j' = j \pm 1. \end{cases} \tag{3.45}$$

The on-site potential energy V_{ij} is chosen to be a uniform random number between 0 and 10^{-3}. The domain size is 32×32 with periodic boundary condition. The chemical potential will be specified later. The accuracy is measured by the L^1 error of the electronic density profile per electron

$$\Delta\rho_{\text{rel}} = \frac{\text{Tr}\,|\widehat{P} - P|}{N_{\text{Electron}}}. \tag{3.46}$$

Contour integral representation: gapped case

The error of the contour integral representation is determined by N_{pole}. At finite temperature $N_{\text{pole}} = 2Q$, while at zero temperature $N_{\text{pole}} = Q$, with Q being the quadrature points on one loop of the contour. The performance of the algorithm is studied by the minimum number of N_{pole} such that $\Delta\rho_{\text{rel}}$ (the L^1 error in the electronic density per electron) is smaller than 10^{-6}. For a given temperature, the chemical potential μ is set to satisfy

$$\text{Tr}\, P = N_{\text{Electron}}. \tag{3.47}$$

$\beta\Delta E$	N_{pole}	$\Delta\rho_{\text{rel}}$
$4,208$	40	5.68×10^{-7}
$8,416$	44	3.86×10^{-7}
$16,832$	44	3.60×10^{-7}
$33,664$	44	3.55×10^{-7}
$67,328$	44	3.57×10^{-7}
$134,656$	44	3.47×10^{-7}
$269,312$	44	3.55×10^{-7}

Table 3.4: N_{pole} and L^1 error of electronic density per electron with respect to various $\beta\Delta E$. The energy gap $E_g \approx 0.01$. The contour integral representation for gapped system at finite temperature is used for the calculation. The performance of the algorithm depends weakly on $\beta\Delta E$.

In our setup the energy gap $E_g \approx 0.01$ Hartree $= 0.27$ eV and $E_M \approx 4$ Hartree. Therefore, this system can be regarded as a crude model for semiconductor with a small energy gap. The number of N_{pole} and the error $\Delta\rho_{\text{rel}}$ are shown in Table 3.4 with respect to $\beta\Delta E$ ranging between $4,000$ and up to $270,000$. Because of the existence of the finite energy gap, the performance is essentially independent of $\beta\Delta E$, as is clearly shown in Table 3.4.

When the temperature is low and therefore when β is large, as discussed before the finite temperature result is well approximated by the zero temperature Fermi operator, *i.e.* , the matrix sign function. In such case the quadrature formula is given by (3.39). Only the contour that encircles the spectrum lower than chemical potential is calculated, and $N_{\text{pole}} = Q$.

In order to study the dependence of $\Delta\rho_{\text{rel}}$ on the number of poles N_{pole}, we tune artificially the chemical potential to reduce the energy gap to 10^{-6} Hartree. Fig. 3.8 shows the exponential decay of $\Delta\rho_{\text{rel}}$ with respect to N_{pole}. For example, in order to reach the 10^{-6} error criterion, $N_{\text{pole}} \approx 50$ is sufficient. The increase in N_{pole} is very small compared to the large decrease of energy gap and this is consistent the logarithmic dependence of N_{pole} on E_g given by (3.37).

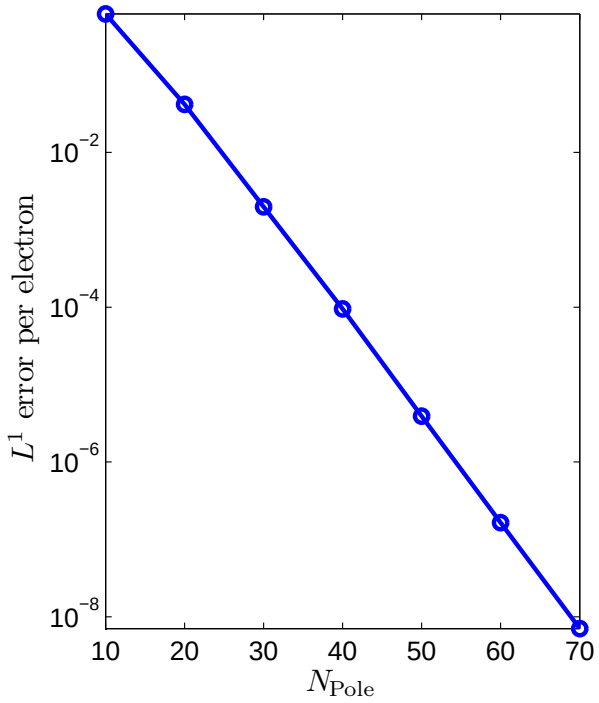


Figure 3.8: The lin-log plot of the L^1 error of electronic density per electron with respect to N_{pole}. The energy gap $E_g \approx 10^{-6}$. The contour integral representation for gapped system at zero-temperature is used for calculation.

Contour integral representation: gapless case

For gapless systems such as metallic systems, our quadrature formula in (3.40) exploits the effective gap on the imaginary axis due to finite temperature. In the following results the chemical potential is set artificially so that $E_g = 0$. $E_M \approx 4$ Hartree and the error criterion is still 10^{-6} as in the gapped case. Table 3.5 reports the number of poles N_{pole} and the error $\Delta\rho_{\mathrm{rel}}$ with respect to $\beta\Delta E$ ranging from $4,000$ up to 4 million. These results are further summarized in Fig. 3.9 to show the logarithmic dependence of N_{pole} on $\beta\Delta E$, as predicted in the analysis of (3.44).

$\beta\Delta E$	N_{pole}	$\Delta\rho_{\mathrm{rel}}$
$4,208$	58	1.90×10^{-7}
$8,416$	62	5.32×10^{-7}
$16,832$	66	8.28×10^{-7}
$33,664$	72	3.55×10^{-7}
$67,328$	76	3.46×10^{-7}
$134,656$	80	1.69×10^{-7}
$269,312$	84	8.89×10^{-8}
$538,624$	88	7.09×10^{-8}
$1,077,248$	88	8.94×10^{-7}
$2,154,496$	88	4.25×10^{-7}
$4,308,992$	92	3.43×10^{-7}

Table 3.5: N_{pole} and L^1 error of electronic density per electron with respect to various $\beta\Delta E$. $E_g = 0$. The contour integral representation for gapless system is used for the calculation.

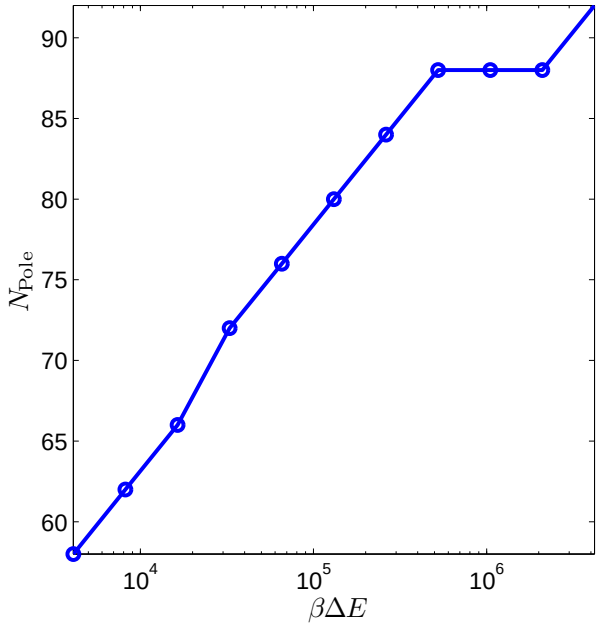


Figure 3.9: Log-lin plot of N_{pole} with respect to $\beta\Delta E$. The contour integral representation for gapless system is used for the calculation.

3.4 Discussion

Compared to the pole expansion, the multipole expansion has a disadvantage that the inverse matrices should be multiplied in order to form multipoles. This makes it difficult to apply the fast algorithms that will be introduced in Chapter 4. On the other hand, it is also possible to find an expansion similar to (3.12) that uses only simple poles. As we mentioned earlier, the key idea in deriving (3.12) is to combine the poles in each group together to form multipoles as the distance between them and the real axis is large. However, if instead we want an expansion that involves only simple poles, it is natural to revisit the variants of FMM that only use simple poles, for example, the version introduced in [253]. The basic idea there is to use a set of equivalent charges on a circle surrounding the poles in each group to reproduce the effective potential away from these poles.

Specifically, take the group of poles from $l = 2^{n-1}$ to $l = 2^n - 1$ for example. Consider a circle B_n with center $c_n = (3 \cdot 2^{n-1} - 2)\pi i$ and radius $r_n = 2^{n-1}\pi$. It is clear that the circle B_n encloses the poles considered. Take P equally spaced points $\{x_{n,k}\}_{k=1}^{P}$ on the circle B_n. Next, one needs to place equivalent charges $\{\rho_{n,k}\}_{k=1}^{P}$ at these points such that the potential produced by these equivalent charges match with the potential produced by the poles inside B_n away from the circle. This can be done in several ways, for example, by matching the multipole expansion, by discretizing the potential on B_n generated by the poles, and so on. Here we follow the approach used in [253].

We simply take a bigger concentric circle $\mathcal{B}_n$ outside B_n with radius $R_n = 2^n\pi$ and match the potential generated on $\mathcal{B}_n$ by the poles and by the equivalent charges on B_n. For this purpose, we solve for $\rho_{n,k}$ the equations

$$\sum_{k=1}^{P} \frac{\rho_{n,k}}{y - x_{n,k}} = \sum_{l=2^{n-1}}^{2^n-1} \frac{1}{y - (2l-1)\pi i}, \quad y \in \mathcal{B}_n. \tag{3.48}$$

Regularization techniques such as Tikhonov regularization are required here since this is a first-kind Fredholm equation.

One can also prove that similar to the original version of the multipole representation, the error in the potential produced by the equivalent charges decay exponentially in P, the details can be found in [253]. Putting these all together, we can write down the following expansion of the Fermi-Dirac function

$$\boldsymbol{\rho} = 1 - 4\Re \sum_{n=1}^{N_G} \sum_{k=1}^{P} \frac{\rho_{n,k}}{\beta(\boldsymbol{H} - \mu) - x_{n,k}} - \frac{2}{\pi} \Im \psi \left(M_{\text{pole}} + \frac{1}{2} + \frac{i}{2\pi} \beta(\boldsymbol{H} - \mu) \right) + \mathcal{O}(N_G / 3^P). \quad (3.49)$$

The number of poles that are effectively represented in the original Matsubara representation is still $M_{\text{pole}} = 2^{N_G} - 1$. $N_{\text{pole}} = N_G P$ simple poles are now to be calculated in practice.

The tail part can be approximated using a Chebyshev polynomial expansion. Similar to the analysis in [160], it can be shown that the complexity of the expansion is $\mathcal{O}(\log \beta \Delta E)$. As we pointed out earlier, the advantage of (3.49) over (3.12) is that only simple poles are involved in the formula. This is useful when combined with fast algorithms for extracting the diagonal of an inverse matrix.

Note that in (3.12) and (3.49), for $2^{n-1} < P$ there would be no savings if we use P terms in the expansion. They are written in this form just for simplicity. In practice the first P simple poles will be calculated separately and the multipole expansion will be used starting from the $(P+1)$-th term and the starting level is $n = \log_2 P + 1$. We show in Fig. 3.10 a typical configuration of the set of poles in the multipole representation type algorithm.

The approach (3.49) based on the multipole representation has three parts of error: the finite-term multipole expansion, the finite-term Chebyshev expansion for the tail part, and the truncated matrix-matrix multiplication in the Chebyshev expansion.

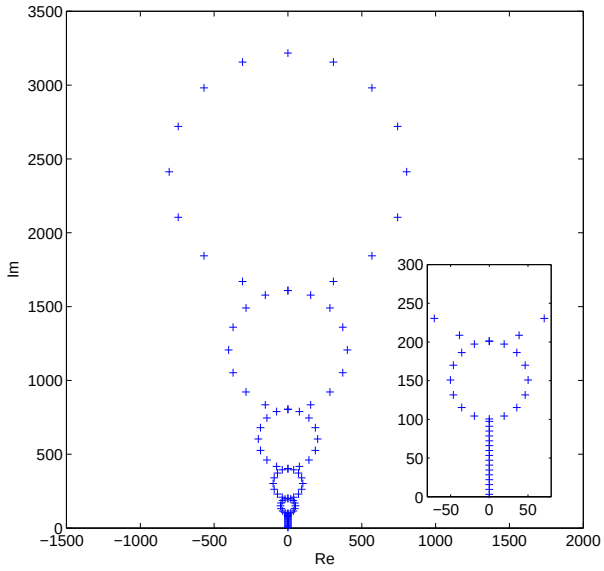


Figure 3.10: A typical configuration of the poles in the multipole representation type algorithm. $M_{\text{pole}} = 512$ and $P = 16$ is used in this figure. The poles with negative imaginary parts are not explicitly shown. The inset shows the first few poles. The first 16 poles are calculated separately and the starting level is $n = 5$.

The error from the multipole expansion is well controlled by P in (3.49). When $P = 16$, $1/3^P \sim \mathcal{O}(10^{-8})$. The number of groups N_G is usually no more than 20, and therefore the error introduced by multipole expansion is around $\mathcal{O}(10^{-7})$, which is much less than the error criterion 10^{-6}.

The number of terms in the Chebyshev expansion for the tail part N_{Cheb} is $\mathcal{O}\big(\frac{\beta\Delta E}{M_{\text{pole}}}\big)$, with M_{pole} being the number of poles that are excluded in the tail part in the pole expansion. The truncation radius for the tail part is $\mathcal{O}\big(\exp(-C\frac{\beta\Delta E}{M_{\text{pole}}})\big)$. In order to reach a fixed target accuracy, we set M_{pole} to be proportional to $\beta\Delta E$. Due to the fact that $M_{\text{pole}} \approx 2^{N_G} \approx 2^{N_{\text{pole}}/P}$, this requires N_{pole} to grow logarithmically with respect to $\beta\Delta E$.

The target accuracy for the Chebyshev expansion is set to be 10^{-7} and the truncation radius for the tail is set to be 4 for the metallic system under consideration. For $\beta\Delta E = 4208$, M_{pole} is set to be 512 so that the error is smaller than 10^{-6}. For other cases, M_{pole} scales linearly with $\beta\Delta E$. The lin-log plot in Fig. 3.11 shows the

logarithmic dependence of N_{pole} with respect to $\beta\Delta E$. For more detailed results, Table 3.6 measures M_{pole}, N_{pole}, N_{Cheb}, and $\Delta\rho_{\text{rel}}$ for $\beta\Delta E$ ranging from 4000 up to 1 million. For all cases, N_{Cheb} is kept as a small constant. Note that the truncation radius is always set to be a small number 4, and this indicates the tail part is extremely localized in the multipole representation due to the effectively raised temperature.

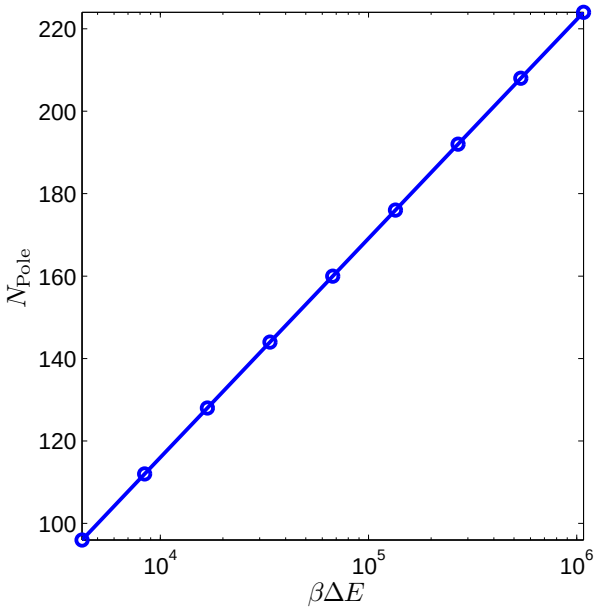


Figure 3.11: log-lin plot of N_{pole} with respect to $\beta\Delta E$. The multipole representation is used for the calculation.

Table 3.6 indicates that the error exhibits some slight growth. We believe that it comes from the growth of the number of groups in the multipole representation (3.49) and also the extra $\log\log$ dependence on $\beta\Delta E$. When compared with the results reported in Table 3.5, we see that for the current application to electronic structure, the pole expansion outperforms the multipole representation in terms of both the accuracy and the number of poles used.

$\beta\Delta E$	M_{pole}	N_{pole}	N_{Cheb}	$\Delta\rho_{\text{rel}}$
$4,208$	512	96	22	4.61×10^{-7}
$8,416$	$1,024$	112	22	4.76×10^{-7}
$16,832$	$2,048$	128	22	4.84×10^{-7}
$33,664$	$4,096$	144	22	4.88×10^{-7}
$67,328$	$8,192$	160	22	4.90×10^{-7}
$134,656$	$16,384$	176	22	4.90×10^{-7}
$269,312$	$32,768$	192	22	6.98×10^{-7}
$538,624$	$65,536$	208	22	3.20×10^{-6}
$1,077,248$	$131,072$	224	22	7.60×10^{-6}

Table 3.6: The number of poles calculated N_{pole}, the order of Chebyshev expansion for the tail part N_{Cheb}, and the L^1 error of electronic density per electron with respect to various $\beta\Delta E$. The number of poles excluded in the tail part M_{pole} is chosen to be proportional to $\beta\Delta E$.

3.5 Conclusion

We have developed the multipole expansion and pole expansion techniques to expand the Fermi operator into simple functions in section 3.2 and in section 3.3, respectively. The two techniques originate from different mathematical observations. The multipole expansion is similar to the fast multipole methods (FMM) which groups the poles together using Taylor expansion. The pole expansion constructs the optimal Cauchy contour integral and the optimal rule for numerical integration. Both techniques achieve the optimal representation cost, *i.e.* the complexity is $\mathcal{O}(\log \beta\Delta E)$.

From practical point of view, the pole expansion is more advantageous. After the detailed comparison in Section 3.4, we find that the preconstant of the pole expansion is smaller. The pole expansion only requires the inversion of matrices, which enables to apply the fast algorithms that will be introduced in Chapter 4.

Chapter 4

Evaluation of the Fermi operator: Selected inversion

4.1 Introduction

The pole expansion developed in Chapter 3 expands the Fermi operator $f(H)$ into simple rational functions

$$f(H) \approx \sum_{i=1}^{P} \frac{\omega_i}{H - z_i I}. \tag{4.1}$$

The question remains to obtain all the diagonal elements and the nearest off-diagonal elements of $(H - z_i)^{-1}$. Clearly, if the diagonal elements and the nearest off-diagonal elements are extracted after directly inverting the matrix, the computational cost is $\mathcal{O}(N^3)$, and there is no advantage in using the Fermi operator expansion method compared to the standard diagonalization method.

There are two possible ways to reduce the computational complexity of calculating the diagonal elements and the nearest off-diagonal elements of an inverse matrix. The first way is to compress the matrix $H - z_i I$ and invert the compressed matrix directly with lower computational cost. The common techniques that fall into this category include fast multiple method [107, 253], hierarchical matrix [36, 114], fast

wavelet transform [31, 33], discrete symbol calculus [72] (**Update reference here**), to name a few. The second way is to avoid the full inverse, but to calculate the diagonal elements, and as few as possible related elements of the inverse matrix. We will discuss briefly the reason why the first way, *i.e.* the existing matrix compression techniques do not apply to the electronic structure calculation. We will then introduce the second way which is able to calculate the diagonal elements of the inverse matrix accurately with reduced computational cost, which shall be referred to later as the *selected inversion* technique.

For a typical electronic structure calculation with the domain taken to be a periodic box $[0, n]^d$ (d is the dimension) after normalization, the potential function $V(x)$ generally oscillates on the $O(1)$ scale, and μ is often on the order of $O(n^d)$. As a result, the operator $(H - \mu I)$ is far from being positive definite. In many computations, the Hamiltonian is sampled with a constant number of points per unit length. Therefore, the discretization of $H - z_i I$, denoted by A, is a matrix of dimension $N = \mathcal{O}(n^d)$.

For the case when A is a positive definite matrix, several ingenious approaches have been developed to represent and manipulate the inverse matrix of A, denoted by G efficiently. One strategy is to represent A and G using multi-resolution basis like wavelets [31, 33]. It is well known that for positive definite A, the wavelet basis offers an asymptotically optimal sparse representation for both A and $G = A^{-1}$. Together with the Newton-Schulz iteration for inverting a matrix, it gives rise to a linear scaling algorithm for calculating the inverse matrix G from A. In one dimension, assuming that we use L levels of wavelet coefficients, and if we truncate the matrix elements that correspond to the wavelets which are centered at locations with distance larger than R, then the cost of matrix-matrix multiplication is roughly $\mathcal{O}(R^2 L^3 N)$. In $2D$, a naive extension using the tensor product structure will lead to a complexity of $\mathcal{O}(R^4 L^6 N)$. This has linear scaling, but the prefactor is rather large: Consider a moderate situation with $R = 10$ and $L = 8$, $R^4 L^6$ is on the order of

10^9. This argument is rather crude, but it does reveal a paradoxical situation with the wavelet representation: Although in principle linear scaling algorithms can be derived using wavelets, they are not practical in $2D$ and $3D$, unless much more sophisticated techniques are developed to reduce the prefactor.

Another candidate for positive definite A is to use the hierarchical matrices [36, 114]. The main observation is that the off-diagonal blocks of the matrix G are numerically low-rank and thus can be approximated hierarchically using low rank factorizations. The cost of multiplying and inverting hierarchical matrices scales as $\mathcal{O}(N)$. Therefore, by either combining with the Newton-Schulz iteration, or directly inverting the hierarchical matrices with block LU decomposition, one obtains a linear scaling algorithm.

Both of these two approaches are quite successful for A being positive definite. Unfortunately as we pointed out earlier, for the application to electronic structure analysis, our matrix A is far from being positive definite. In fact, the matrix elements of G are highly oscillatory due to the shift of chemical potential in the Hamiltonian. Consequently, the inverse matrix G does not have an efficient representation in either the wavelet basis or the hierarchical matrix framework. The same argument applies to other fast algorithms designed for elliptic operators, such as fast multiple method [107, 253] and discrete symbol calculus [72].

The selected inversion algorithm developed in this chapter follows the orthogonal direction to the matrix compression approach. The selected inversion method does not calculate all the elements of the inverse matrix G, but only the diagonal elements and the nearest off-diagonal elements of the inverse matrix, and as few as possible other related elements. The selected inversion algorithm is able to accurately compute the diagonal elements and the nearest off-diagonal elements of G with $\mathcal{O}(N)$ complexity for quasi-1D systems, $\mathcal{O}(N^{1.5})$ complexity for quasi-2D systems, and $\mathcal{O}(N^2)$ complexity for 3D bulk systems. The selected inversion method is applicable if the

Hamiltonian operator is discretized by localized basis functions, such as finite difference, finite element, spectral element, and adaptive local basis functions developed in Chapter 2.

This chapter is organized as following. Section 4.2 introduces the basic idea of the selected inversion and illustrates how selected inversion reduces the computational cost for extracting the diagonals and the nearest off-diagonal elements for general symmetric matrices. Section 4.3 introduces the SelInv software which is a sequential algorithm for performing the selected inversion for general symmetric matrices. A parallel selected inversion for structured 2D matrices is developed in Section 4.4. The discussion and future work is summarized in Section 4.5. Materials in this chapter have been presented in [162, 169, 170].

4.2 Selected inversion: Basic idea

4.2.1 Dense matrix

An obvious way to obtain selected components of A^{-1} is to compute A^{-1} first and then simply pull out the needed entries. The standard approach for computing A^{-1} is to first decompose A as

$$A = LDL^T, \tag{4.2}$$

where L is a unit lower triangular matrix and D is a diagonal or a block-diagonal matrix. Equation (4.2) is often known as the LDL^T factorization of A. For positive definite matrices, D can always be kept as a diagonal matrix. For general symmetric matrices, a block LDL^T factorization that allows 2×2 block pivots [45, 46] or partial pivoting [96] may be used to achieve numerical stability in the factorization. Given such a factorization, one can obtain $A^{-1} = (x_1, x_2, \ldots, x_n)$ by solving a number of

triangular systems

$$Ly = e_j, \quad Dw = y, \quad L^T x_j = w, \tag{4.3}$$

for $j = 1, 2, \ldots, n$, where e_j is the j-th column of the identity matrix I. The computational cost of such algorithm is generally $\mathcal{O}(n^3)$, with n being the dimension of A. However, when A is sparse, we can exploit the sparsity structure of L and e_j to reduce the complexity of computing selected components of A^{-1}. We will examine this type of algorithm, which we will refer to as *direct inversion*, further in Section 4.3.3 when we compare the performance of direct inversion with that of our new fast algorithm.

The selected inversion algorithm which is summarized below also perform an LDL^T factorization of A first. However, the algorithm does not require solving (4.3) directly. Before we present this algorithm, it will be helpful to first review the major operations involved in the LDL^T factorization of A.

Let

$$A = \begin{pmatrix} \alpha & b^T \\ b & \hat{A} \end{pmatrix}, \tag{4.4}$$

be a nonsingular symmetric matrix. The first step of an LDL^T factorization produces a decomposition of A that can be expressed by

$$A = \begin{pmatrix} 1 & \\ \ell & I \end{pmatrix} \begin{pmatrix} \alpha & \\ & \hat{A} - bb^T/\alpha \end{pmatrix} \begin{pmatrix} 1 & \ell^T \\ & I \end{pmatrix},$$

where α is often referred to as a pivot, $\ell = b/\alpha$ and $S = \hat{A} - bb^T/\alpha$ is known as the *Schur complement*. The same type of decomposition can be applied recursively to the Schur complement S until its dimension becomes 1. The product of lower triangular

matrices produced from the recursive procedure, which all have the form

$$\begin{pmatrix} I & & \\ & 1 & \\ & \ell^{(i)} & I \end{pmatrix},$$

where $\ell^{(1)} = \ell = b/\alpha$, yields the final L factor. At this last step the matrix in the middle becomes diagonal, which is the D matrix.

To simplify our discussion, we assume here that all pivots produced in the LDL^T factorization are sufficiently large so that no row or column permutation (pivoting) is needed during the factorization. The discussion can be readily generalized if D contains 2×2 blocks.

The key observation is that A^{-1} can be expressed by

$$A^{-1} = \begin{pmatrix} \alpha^{-1} + \ell^T S^{-1} \ell & -\ell^T S^{-1} \\ -S^{-1}\ell & S^{-1} \end{pmatrix}. \tag{4.5}$$

This expression suggests that once α and ℓ are known, the task of computing A^{-1} can be reduced to that of computing S^{-1}.

Because a sequence of Schur complements is produced recursively in the LDL^T factorization of A, the computation of A^{-1} can be organized in a recursive fashion too. Clearly, the reciprocal of the last entry of D is the (n, n)-th entry of A^{-1}. Starting from this entry, which is also the 1×1 Schur complement produced in the $(n-1)$-th step of the LDL^T factorization procedure, we can construct the inverse of the 2×2 Schur complement produced at the $(n-2)$-th step of the factorization procedure, using the recipe given by (4.5). This 2×2 matrix is the trailing 2×2 block of A^{-1}. As we proceed from the lower right corner of L and D towards their upper left corner, more and more elements of A^{-1} are recovered. The complete procedure can be easily described by a MATLAB script shown in Algorithm 1.

Algorithm 1 A MATLAB script for computing the inverse of a dense matrix A given its LDL^T factorization.

Input: A unit triangular matrix L and a diagonal matrix D such that $A = LDL^T$;
Output: The inverse of A denoted by `Ainv`.

```
Ainv(n,n) = 1/D(n,n);
for j = n-1:-1:1
  Ainv(j+1:n,j) = -Ainv(j+1:n,j+1:n)*L(j+1:n,j);
  Ainv(j,j+1:n) = Ainv(j+1:n,j)';
  Ainv(j,j)     = 1/D(j,j) - L(j+1:n,j)'*Ainv(j+1:n,j);
end;
```

For the purpose of clarity, we use a separate array `Ainv` in Algorithm 1 to store the computed A^{-1}. In practice, A^{-1} can be computed in place. That is, we can overwrite the array used to store L and D with the lower triangular and diagonal part of A^{-1} incrementally.

4.2.2 Sparse matrix

It is not difficult to observe that if A is a dense matrix, the complexity of Algorithm 1 is $\mathcal{O}(n^3)$ because a matrix vector multiplication involving a $j \times j$ dense matrix is performed at the j-th iteration of this procedure, and $(n-1)$ iterations are required to fully recover A^{-1}. Therefore, when A is dense, this procedure does not offer any advantage over the standard way of computing A^{-1}. Furthermore, all elements of A^{-1} are needed and computed. No computational cost can be saved if we just want to extract selected elements (e.g., the diagonal elements) of A^{-1}.

However, when A is sparse, a tremendous amount of savings can be achieved if we are only interested in the diagonal components of A^{-1}. If the vector ℓ in (4.5) is sparse, computing $\ell^T S^{-1} \ell$ does not require all elements of S^{-1} to be obtained in advance. Only those elements that appear in the rows and columns corresponding to the nonzero rows of ℓ are required.

Therefore, to compute the diagonal elements of A^{-1}, we can simply modify the

procedure shown in Algorithm 1 so that at each iteration we only compute selected elements of A^{-1} that will be needed by subsequent iterations of this procedure. It turns out that the elements that need to be computed are completely determined by the nonzero structure of the lower triangular factor L. To be more specific, at the j-th step of the selected inversion process, we compute $(A^{-1})_{i,j}$ for all i such that $L_{i,j} \neq 0$. Therefore, our algorithm for computing the diagonal of A^{-1} can be easily illustrated by a MATLAB script (which is not the most efficient implementation) shown in Algorithm 2.

Algorithm 2 A MATLAB script for computing selected matrix elements of A^{-1} for a sparse symmetric matrix A.

Input: A unit triangular matrix L and a diagonal matrix D such that $A = LDL^T$;
Output: Selected elements of A^{-1} denoted by `Ainv`, i.e. the elements $(A^{-1})_{i,j}$ such that $L_{i,j} \neq 0$.

```
Ainv(n,n) = 1/D(n,n);
for j = n-1:-1:1
   % find the row indices of the nonzero elements in
   % the j-th column of L
   inz = j + find(L(j+1:n,j)~=0);
   Ainv(inz,j) = -Ainv(inz,inz)*L(inz,j);
   Ainv(j,inz) =  Ainv(inz,j)';
   Ainv(j,j)   =  1/D(j,j) - Ainv(j,inz)*L(inz,j);
end;
```

To see why this type of selected inversion is sufficient, we only need to examine the nonzero structure of the k-th column of L for all $k < j$ since such a nonzero structure tells us which rows and columns of the trailing sub-block of A^{-1} are needed to complete the calculation of the (k, k)-th entry of A^{-1}. In particular, we would like to find out which elements in the j-th column of A^{-1} are required for computing $A^{-1}_{i,k}$ for any $k < j$ and $i \geq j$.

Clearly, when $L_{j,k} = 0$, the j-th column of A^{-1} is not needed for computing the k-th column of A^{-1}. Therefore, we only need to examine columns k of L such that $L_{j,k} \neq 0$. A perhaps not so obvious but critical observation is that for these columns,

(b) The elimination tree.

Figure 4.1: The lower triangular factor L of a sparse 10×10 matrix A and the corresponding elimination tree.

If $L_{j,k} \neq 0$ and $k < j$, then the node k is a descendant of j in the elimination tree. An example of the elimination tree of a matrix A and its L factor are shown in Figure 4.1. Such a tree can be used to clearly describe the dependency among different columns in a sparse LDL^T factorization of A. In particular, it is not too difficult to show that constructing the j-th column of L requires contributions from

descendants of j that have a nonzero matrix element in the j-th row [172].

Similarly, we may also use the elimination tree to describe which selected elements within the trailing sub-block A^{-1} are required in order to obtain the (j, j)-th element of A^{-1}. In particular, it is not difficult to show that the selected elements must belong to the rows and columns of A^{-1} that are among the ancestors of j.

4.3 SelInv – An algorithm for selected inversion of a sparse symmetric matrix

4.3.1 Block Algorithms and Supernodes

The selected inversion procedure described in Algorithm 1 and its sparse version can be modified to allow a block of rows and columns to be modified simultaneously. A block algorithm can be described in terms a block factorization of A. For example, if A is partitioned as

$$A = \begin{pmatrix} A_{11} & B_{21}^T \\ B_{21} & A_{22} \end{pmatrix},$$

its block LDL^T factorization has the form

$$A = \begin{pmatrix} I & \\ L_{21} & I \end{pmatrix} \begin{pmatrix} A_{11} & \\ & A_{22} - B_{21}A_{11}^{-1}B_{21}^T \end{pmatrix} \begin{pmatrix} I & L_{21}^T \\ & I \end{pmatrix}, \tag{4.6}$$

where $L_{21} = B_{21}A_{11}^{-1}$ and $S = A_{22} - B_{21}A_{11}^{-1}B_{21}^T$ is the Schur complement. The corresponding block version of (4.5) can be expressed by

$$A^{-1} = \begin{pmatrix} A_{11}^{-1} + L_{21}^T S^{-1} L_{21} & -L_{21}^T S^{-1} \\ -S^{-1}L_{21} & S^{-1} \end{pmatrix}.$$

There are at least three advantages of using a block algorithm:

1. It allows us to use level 3 BLAS (Basic Linear Algebra Subroutine) to develop an efficient implementation by exploiting the memory hierarchy in modern microprocessors.

2. When applied to sparse matrices, it tends to reduce the amount of indirect addressing overhead.

3. It allows 2×2 block pivots that can be used to overcome numerical instabilities that may arise when A is indefinite.

When A is sparse, the columns of A and L can be partitioned into *supernodes*. A supernode is a maximal set of contiguous columns $\{j, j+1, ..., j+s\}$ of L such that they have the same nonzero structure below the $(j+s)$-th row and the lower triangular part of $L_{j:j+s,j:j+s}$ is completely dense. An example of a supernode partition of the lower triangular factor L associated with a 49×49 sparse matrix A is shown in Figure 4.2. The definition of a supernode can be relaxed to include columns whose nonzero structures are nearly identical with adjacent columns. However, we will not be concerned with such an extension in this chapter. We will use upper case script letters such as $\mathcal{J}$ to denote a supernode. Following the convention introduced in [195], we will interpret $\mathcal{J}$ either as a supernode index or a set of column indices contained in that supernode depending on the context.

We should note here that the supernode partition of A or L is completely based on the nonzero structure of A. Although it is desirable to create supernodes that contain all 2×2 block pivots priori to numerical factorization of A, this is generally difficult to do for sparse matrices. When the size of a supernode is larger than 1, we can still use 2×2 block pivots within this supernode to improve numerical stability of the LDL^T factorization. This type of strategy is often used in multifrontal solvers [11, 77].

We denote the set of row indices associated with the nonzero rows below the di-

Figure 4.2: A supernode partition of L.

agonal block of the $\mathcal{J}$-th supernode by $S_{\mathcal{J}}$. These row indices are further partitioned into $n_{\mathcal{J}}$ disjoint subsets $\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_{n_{\mathcal{J}}}$ such that $\mathcal{I}_i$ contains a maximal set of contiguous row indices and $\mathcal{I}_i \subset \mathcal{K}$ for some supernode $\mathcal{K} > \mathcal{J}$. Here $\mathcal{K} > \mathcal{J}$ means $k > j$ for all $k \in \mathcal{K}$ and $j \in \mathcal{J}$. In Figure 4.3, we show how the nonzero rows associated with one of the supernodes (the 26-th supernode which begins at column 27) are partitioned. The purpose of the partition is to create dense submatrices of L that can be easily accessed and manipulated. The reason we impose the constraint $\mathcal{I}_i \subset \mathcal{K}$, which is normally not required in the LDL^T factorization of A, will become clear in Section 4.3.2. We should also note that, under this partitioning scheme, it is possible that $\mathcal{I}_i$ and $\mathcal{I}_j$ belong to the same supernode even if $i \neq j$.

The use of supernodes leads to a necessary but straightforward modification of the elimination tree. All nodes associated with columns within the same supernode are collapsed into a single node. The modified elimination tree describes the dependency among different supernodes in a supernode LDL^T factorization of A (see [195, 221]). Such dependency also defines the order by which selected blocks of A^{-1} are computed.

Using the notion of supernodes, we can modify the selected inversion process de-

scribed by the MATLAB script shown in Algorithm 2 to make it more efficient. If columns of L can be partitioned into n_{sup} supernodes, a supernode based block selected inversion algorithm can be described by the pseudocode shown in Algorithm 3.

Algorithm 3 A supernode-based algorithm for computing the selected elements of A^{-1}.

Input: (1) The supernode partition of columns of A: $\{1, 2, ..., n_{sup}\}$;
(2) A supernode LDL^T factorization of A;
Output: Selected elements of A^{-1}, i.e. $(A^{-1})_{i,j}$ such that $L_{i,j} \neq 0$.

1: Compute $A^{-1}_{n_{sup},n_{sup}} = D^{-1}_{n_{sup},n_{sup}}$;
2: **for** $\mathcal{J} = n_{sup} - 1, n_{sup} - 2, ..., 1$ **do**
3: Identify the nonzero rows in the $\mathcal{J}$-th supernode $S_{\mathcal{J}}$;
4: Perform $Y = A^{-1}_{S_{\mathcal{J}},S_{\mathcal{J}}} L_{S_{\mathcal{J}},\mathcal{J}}$;
5: Calculate $A^{-1}_{\mathcal{J},\mathcal{J}} = D^{-1}_{\mathcal{J},\mathcal{J}} + Y^T L_{S_{\mathcal{J}},\mathcal{J}}$;
6: Set $A^{-1}_{S_{\mathcal{J}},\mathcal{J}} \leftarrow -Y$;
7: **end for**

4.3.2 Implementation details

We now describe some of the implementation details that allow the selected inversion process described schematically in Algorithm 3 to be carried out in an efficient manner.

We assume a supernode LDL^T factorization has been performed using, for example, an efficient left-looking algorithm described in [195, 221]. Such an algorithm typically stores the nonzero elements of L in a contiguous array using the compressed column format [76]. This array will be overwritten by the selected elements of A^{-1}. The row indices associated with the nonzero rows of each supernode are stored in a separate integer array. Several additional integer arrays are used to mark the supernode partition and column offsets.

As we illustrated in Algorithm 3, the selected inversion process proceeds backwards from the last supernode n_{sup} towards the first supernode. For all supernodes $\mathcal{J} < n_{sup}$,

we need to perform a matrix-matrix multiplication of the form

$$Y = (A^{-1})_{S_{\mathcal{J}},S_{\mathcal{J}}} L_{S_{\mathcal{J}},\mathcal{J}}, \tag{4.7}$$

where $\mathcal{J}$ serves the dual purposes of being a supernode index and an index set that contains all column indices belonging to the $\mathcal{J}$-th supernode, and $S_{\mathcal{J}}$ denotes the set of row indices associated with nonzero rows within the $\mathcal{J}$-th supernode of L.

Recall that the row indices contained in $S_{\mathcal{J}}$ are partitioned into a number of disjoint subsets $\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_{n_{\mathcal{J}}}$ such that $\mathcal{I}_i \subset \mathcal{K}$ for some supernode $\mathcal{K} > \mathcal{J}$. Such a partition corresponds to a row partition of the dense matrix block associated with the $\mathcal{J}$-th supernode into $n_{\mathcal{J}}$ submatrices. The same partition is applied to the rows and columns of the submatrix $(A^{-1})_{S_{\mathcal{J}},S_{\mathcal{J}}}$ except that this submatrix is not stored in a contiguous array. For example, the nonzero row indices of the 26-th supernode in Figure 4.2, which consists of columns 27, 28 and 29, can be partitioned as

$$S_{26} = \{30\} \cup \{40, 41\} \cup \{43, 44, 45\}.$$

This partition as well as the corresponding partition of the blocks in the trailing A^{-1} that are used in (4.7) is highlighted in Figure 4.3.

We carry out the matrix-matrix multiplication (4.7) by using Algorithm 4. The outer loop (line 2) of this algorithm goes through each block column of $(A^{-1})_{S_{\mathcal{J}},S_{\mathcal{J}}}$ indexed by $\mathcal{I}_j \in S_{\mathcal{J}}$, and accumulates $(A^{-1})_{*,\mathcal{I}_j} L_{\mathcal{I}_j,\mathcal{J}}$ in the dense matrix Y stored in a contiguous work array. The inner loop of this algorithm, which starts from line 6, simply goes through the nonzero blocks of $(A^{-1})_{*,\mathcal{I}_j}$ to perform $(A^{-1})_{\mathcal{I}_i,\mathcal{I}_j} L_{\mathcal{I}_j,\mathcal{J}}$, $i = j + 1, ..., n_{\mathcal{J}}$, one block at a time. Because A^{-1} is symmetric, we store only the selected nonzero elements in the lower triangular part of the matrix (except the diagonal blocks in which both the upper and lower triangular parts of the matrix are stored in a full dense matrix.) Hence, our implementation of (4.7) also computes the

Figure 4.3: The partition of the nonzero rows in S_{26} and the matrix elements needed in $A^{-1}_{30:49,30:49}$ for the computation of $A^{-1}_{30:49,30:49}L_{30:39,27:29}$.

contribution of $(A^{-1})^T_{*,\mathcal{I}_j}L_{\mathcal{I}_i,\mathcal{J}}$ to Y as the $\mathcal{I}_j$-th block column of A^{-1} is accessed (step 10) in the inner loop of Algorithm 4.

An efficient implementation of Algorithm 4 requires each sub-block of $A^{-1}_{S_\mathcal{J},S_\mathcal{J}}$ (within the storage allocated for L) to be identified quickly and the product of $(A^{-1})_{\mathcal{I}_i,\mathcal{I}_j}$ and $L_{\mathcal{I}_j,\mathcal{J}}$, as well as the product of $[(A^{-1})_{\mathcal{I}_i,\mathcal{I}_j}]^T$ and $L_{\mathcal{I}_i,\mathcal{J}}$, to be placed at appropriate locations in the Y array. To achieve these goals, we use an integer array `indmap` with n entries to record the relative row positions of the first row of $\mathcal{I}_i$ in Y, for $i = 2, 3, ..., n_\mathcal{J}$. (The relative positions of all other nonzero rows can be easily calculated once the relative row position of the first row of $\mathcal{I}_i$ is determined, because the row numbers in $\mathcal{I}_i$ are contiguous.) To be specific, all the entries of `indmap` are initialized to be zero. If k is an element in $\mathcal{I}_i$ (all elements in $\mathcal{I}_i$ are sorted in an ascending order), then `indmap[k]` is set to be the relative distance of row k from the last row of the diagonal block of the $\mathcal{J}$-th supernode in L. For example, in Figure 4.3, the leftmost supernode S_{26}, which contains columns $27, 28, 29$, contains 6 nonzero rows below its diagonal block. The nonzero entries of the `indmap` array for

Algorithm 4 Compute $Y = (A^{-1})_{S_\mathcal{J},S_\mathcal{J}} L_{S_\mathcal{J},\mathcal{J}}$ needed in Step 4 of Algorithm 3.

Input: (1) The $\mathcal{J}$-th supernode of L, $L_{S_\mathcal{J},\mathcal{J}}$, where $S_\mathcal{J}$ contains the indices of the nonzero rows in $\mathcal{J}$. The index set $S_\mathcal{J}$ is partitioned into disjoint $n_\mathcal{J}$ subsets of contiguous indices, i.e. $S_\mathcal{J} = \{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_{n_\mathcal{J}}\}$;
(2) The nonzero elements of A^{-1} that have been computed previously. These elements are stored in $L_{S_\mathcal{K},\mathcal{K}}$ for all $\mathcal{K} > \mathcal{J}$;

Output: $Y = (A^{-1})_{S_\mathcal{J},S_\mathcal{J}} L_{S_\mathcal{J},\mathcal{J}}$;

1: Construct an `indmap` array for nonzero rows in the $\mathcal{J}$-th supernode;
2: **for** $j = 1, 2, ..., n_\mathcal{J}$ **do**
3: Identify the supernode $\mathcal{K}$ that contains $\mathcal{I}_j$;
4: Let $\mathcal{R}_1 =$ `indmap`$(\mathcal{I}_j)$;
5: Calculate $Y_{\mathcal{R}_1,*} \leftarrow Y_{\mathcal{R}_1,*} + (A^{-1})_{\mathcal{I}_j,\mathcal{I}_j} L_{\mathcal{I}_j,\mathcal{J}}$;
6: **for** $i = j+1, j+2, ...n_\mathcal{J}$ **do**
7: Use `indmap` to find the first nonzero row in the $\mathcal{K}$-th supernode that belongs to $\mathcal{I}_i$ so that $(A^{-1})_{\mathcal{I}_i,\mathcal{I}_j}$ can be located;
8: Let $\mathcal{R}_2 =$ `indmap`$(\mathcal{I}_i)$;
9: Calculate $Y_{\mathcal{R}_2,*} \leftarrow Y_{\mathcal{R}_2,*} + (A^{-1})_{\mathcal{I}_i,\mathcal{I}_j} L_{\mathcal{I}_j,\mathcal{J}}$;
10: Calculate $Y_{\mathcal{R}_1,*} \leftarrow Y_{\mathcal{R}_1,*} + [(A^{-1})_{\mathcal{I}_i,\mathcal{I}_j}]^T L_{\mathcal{I}_i,\mathcal{J}}$;
11: **end for**
12: **end for**
13: Reset the nonzero entries of `indmap` to zero;

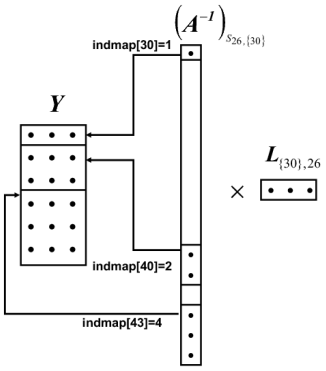
S_{26} are

```
indmap[30] = 1,
indmap[40] = 2,
indmap[41] = 3,
indmap[43] = 4,
indmap[44] = 5,
indmap[45] = 6.
```

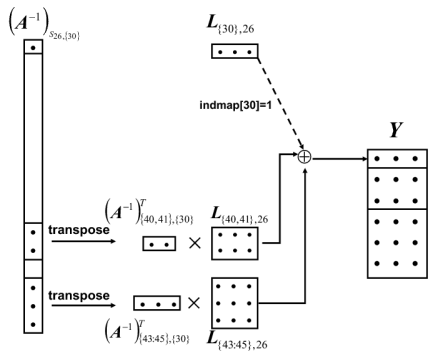
These entries of the `indmap` array are reset to zeros once the calculation of (4.7) is completed for each $\mathcal{J}$. A similar indirect addressing scheme is used in [195] for gathering the contributions from the descendants of the $\mathcal{J}$-th supernode that have already been updated in the previous steps of a left-looking supernodal LDL^T factorization. Our use of indirect addressing collects contributions from the ancestors of the $\mathcal{J}$-th supernode as $(A^{-1})_{S_{\mathcal{J}},\mathcal{J}}$ is being updated.

Once the `indmap` array is properly set up, the sub-block searching process indicated in line 7 of the pseudocode shown in Algorithm 4 goes through the row indices k of the nonzero rows of the $\mathcal{K}$-th supernode (that contains $\mathcal{I}_j$) until a nonzero `indmap[k]` is found (step 7). A separate pointer p to the floating point array allocated for L is incremented at the same time. When a nonzero `indmap[k]` is found, the position in the floating point array pointed by p gives the location of $(A^{-1})_{\mathcal{I}_i,\mathcal{I}_j}$ required in line 9 of the special matrix-matrix multiplication procedure shown in Algorithm 4. Meanwhile, the value of `indmap[k]` gives the location of the target work array Y at which the product of $(A^{-1})_{\mathcal{I}_i,\mathcal{I}_j}$ and $L_{\mathcal{I}_j,\mathcal{J}}$ is accumulated. After lines 9 and 10 are executed in the inner loop of Algorithm 4, the remaining nonzero rows in the $\mathcal{K}$-th supernode are examined until the next desired sub-block in the $\mathcal{K}$-th supernode of A^{-1} is found or until all nonzero rows within this supernode have been examined. Figure 4.4 shows of how the `indmap` array is used to place the product of $(A^{-1})_{\mathcal{I}_i,\{30\}}$ and $L_{\{30\},26}$ as well as the product of $(A^{-1})^T_{\mathcal{I}_i,\{30\}}$ and $L_{\mathcal{I}_i,26}$ in the Y array at lines 9

and 10 of Algorithm 4 for the example problem given in Figure 4.3.



(a) The use of `indmap` in Step 9.



(b) The use of `indmap` in Step 10. The dashed arrow indicates that $L_{\{30\},26}$ provides only the index for the row of Y that is modified.

Figure 4.4: A schematic drawing that illustrates how `indmap` is used in Steps 9 and 10 in the first outer iteration of Algorithm 4 for $\mathcal{J} = 26$ in the example given in Figure 4.3.

Before we copy Y to the appropriate location in the array that stores the $\mathcal{J}$-th supernode of L, we need to compute the diagonal block of A^{-1} within this supernode by the following update:

$$(A^{-1})_{\mathcal{J},\mathcal{J}} = (A^{-1})_{\mathcal{J},\mathcal{J}} + Y^T L_{S_{\mathcal{J}},\mathcal{J}},$$

where $(A^{-1})_{\mathcal{J},\mathcal{J}}$, which is stored in the diagonal block of the storage allocated for L, contains the inverse of the diagonal block $D_{\mathcal{J},\mathcal{J}}$ (which may contain 2×2 pivots) produced by the supernode LDL^T factorization before the update is performed.

4.3.3 Performance

Here we report the performance of our selected inversion algorithm SelInv. Our performance analysis is carried out on the Franklin Cray XT4 supercomputing system

maintained at NERSC[1]. Each compute node consists of a 2.3 GHz single socket quad-core AMD Opteron processor (Budapest) with a theoretical peak performance of 9.2 GFlops/sec per core (4 flops/cycle if using SSE128 instructions). Each core has 2 GB of memory. Our test problems are taken from the Harwell-Boeing Test Collection [76] and the University of Florida Matrix Collection[70]. These matrices are widely used benchmark problems for sparse direct methods. The names of these matrices as well as some of their characteristics are listed in Table 4.1 and 4.2. All matrices are real and symmetric. The multiple minimum degree (MMD) matrix reordering strategy [171] is used to minimize the amount of nonzero fills in L. We used the supernodal left-looking algorithm and code provided by the authors of [195] to perform the LDL^T factorization of A. Table 4.3 gives the performance result in terms of computational time as well as floating point operations per second (flops) for both the factorization and the selected inversion algorithms respectively. We also report the average flops measured on-the-fly using PAPI [44]. The dimension of the matrices we tested ranges from $2,000$ to 1.5 million, and the number of nonzero elements in the L factor ranges from 0.1 million to 0.2 billion. For the largest problem G3_circuit, the overall computation takes only 350 seconds. Among these problems, the best performance is obtained with the problem pwtk. For this particular problem, the factorization part attains 26% of the peak performance of the machine, and the selected inversion part attains 68% of the peak flops. The average (of the factorization and inversion) flops ratio is 46%. The flops performance is directly related to the supernode size distribution due to the reordering strategy. For pwtk, 90% of the supernodes have sizes larger than 5. By contrast, the dimension of parabolic_fem is more than twice the dimension of pwtk, but 81% of the supernodes contain only one column. Consequently, SelInv cannot take full advantage of level 3 BLAS when it is used to solve this problem. As a result, its performance is worse on this problem than on pwtk.

[1] http://www.nersc.gov/

To demonstrate how much we can gain by using the selected inversion algorithm, we compare the timing statistics of the selected inversion algorithm with that of the *direct inversion* algorithm mentioned. In our implementation of the direct inversion algorithm, we compute the diagonal elements of A^{-1} using $e_j^T A^{-1} e_j = (L^{-1}e_j)^T D^{-1}(L^{-1}e_j)$, where e_j is the j-th column of the identity matrix. When computing $y = L^{-1}e_j$ (via solving $Ly = e_j$), we modify only the nonzero entries of y. The positions of these entries can be predicted by the traversal of a directed graph constructed from the nonzero structure of L [95]. This approach reduces the number of floating point operations significantly compared to a naive approach that does not take into account the sparsity of L or e_j. However, it still has a higher asymptotic complexity compared to the selected inversion algorithm we presented earlier. This can be seen from the following example in which A is a discretized Laplacian operator obtained from applying a five-point stencil on an $m \times m$ grid in 2D where $m = n^{1/2}$. Assuming A is ordered by nested dissection [92] so that the last m columns of A corresponds to nodes in the largest separator, we can see that solving $Ly = e_j$, for $j = n - m + 1, ..., n$, would require a total of $\mathcal{O}(m^2) = \mathcal{O}(n)$ operations because the lower triangular part of $L_{n-m+1:n,n-m+1:n}$ is completely dense. Because these columns belong to a supernode that is at the root of the elimination tree, they are all reachable from node j on the directed graph constructed from solving $Ly = e_j$ for $j = 1, 2, ..., n - m$. Consequently, the overall complexity for solving $Ly = e_j$ for $j = 1, 2, ..., n$ is $\mathcal{O}((n - m)n + n) = \mathcal{O}(n^2)$. This is higher than the $\mathcal{O}(n^{3/2})$ complexity associated with selected inversion. Similarly, if A is a discretized Laplacian operator obtained from applying a seven-point stencil on an $m \times m \times m$ grid in 3D where $m = n^{1/3}$, the complexity of direct inversion becomes $\mathcal{O}(n^{7/3})$ because the largest separator contains $n^{2/3}$ columns, whereas the complexity of selected inversion is $\mathcal{O}(n^2)$.

Although it is difficult to perform such analysis for a general sparse matrix, similar

problem	description
bcsstk14	Roof of the Omni Coliseum, Atlanta.
bcsstk24	Calgary Olympic Saddledome arena.
bcsstk28	Solid element model, linear statics.
bcsstk18	R.E. Ginna Nuclear Power Station.
bodyy6	NASA, Alex Pothen.
crystm03	FEM crystal free vibration mass matrix.
wathen120	Gould,Higham,Scott: matrix from Andy Wathen, Oxford Univ.
thermal1	Unstructured FEM, steady state thermal problem, Dani Schmid, Univ. Oslo.
shipsec1	DNV-Ex 4 : Ship section/detail from production run-1999-01-17.
pwtk	Pressurized wind tunnel, stiffness matrix.
parabolic_fem	Diffusion-convection reaction, constant homogeneous diffusion.
tmt_sym	Symmetric electromagnetic problem, David Isaak, Computational_EM_Works.
ecology2	Circuitscape: circuit theory applied to animal/gene flow, B. McRae, UCSB.
G3_circuit	Circuit simulation problem, Ufuk Okuyucu, AMD, Inc.

Table 4.1: Test problems

difference in complexity should hold. To provide a more concrete comparison, we list the timing measurements for both approaches in Table 4.4 as well as the speedup factor. The speedup factor is defined by the time for selected inversion divided by the time for direct inversion. In this comparison, selected inversion refers to the second part of SelInv, i.e., the time for LDL^T factorization is not counted since factorization is used in both algorithms. We also terminate the direct inversion algorithm if its running time is larger than 3 hours. We observe that for the smallest problem bcsstk14, the speedup factor is already 13. For larger problems, the speedup can be several hundreds or more.

problem	n	$\|A\|$	$\|L\|$
bcsstk14	1,806	32,630	112,267
bcsstk24	3,562	81,736	278,922
bcsstk28	4,410	111,717	346,864
bcsstk18	11,948	80,519	662,725
bodyy6	19,366	77,057	670,812
crystm03	24,696	304,233	3,762,633
wathen120	36,441	301,101	2,624,133
thermal1	82,654	328,556	2,690,654
shipsec1	140,874	3,977,139	40,019,943
pwtk	217,918	5,926,171	56,409,307
parabolic_fem	525,825	2,100,225	34,923,113
tmt_sym	726,713	2,903,837	41,296,329
ecology2	999,999	2,997,995	38,516,672
G3_circuit	1,585,478	4,623,152	197,137,253

Table 4.2: Characteristic of the test problems

problem	factorization time (sec)	factorization flops (G/sec)	selected inversion time (sec)	selected inversion flops (G/sec)	average flops (G/sec)
bcsstk14	0.007	1.43	0.010	2.12	1.85
bcsstk24	0.019	1.75	0.020	3.65	2.71
bcsstk28	0.023	1.63	0.024	3.46	2.54
bcsstk18	0.080	1.80	0.235	1.54	1.60
bodyy6	0.044	1.49	0.090	1.68	1.61
crystm03	0.452	2.26	0.779	2.95	2.70
wathen120	0.251	2.12	0.344	3.47	2.90
thermal1	0.205	1.53	0.443	1.66	1.62
shipsec1	18.48	2.38	17.66	5.45	3.88
pwtk	16.43	2.48	14.55	6.28	4.26
parabolic_fem	6.649	2.34	20.06	1.91	2.02
tmt_sym	10.64	2.35	13.98	4.02	3.30
ecology2	6.789	2.32	16.04	2.35	2.34
G3_circuit	136.5	2.24	218.7	3.27	2.88

Table 4.3: The time cost, and flops result for factorization and selected inversion process respectively. The last column reports the average flops reached by SelInv.

problem	selected inversion time	direct inversion time	speedup
bcsstk14	0.01 sec	0.13 sec	13
bcsstk24	0.02 sec	0.58 sec	29
bcsstk28	0.02 sec	0.88 sec	44
bcsstk18	0.24 sec	5.73 sec	24
bodyy6	0.09 sec	5.37 sec	60
crystm03	0.78 sec	26.89 sec	34
wathen120	0.34 sec	48.34 sec	142
thermal1	0.44 sec	95.06 sec	216
shipsec1	17.66 sec	3346 sec	192
pwtk	14.55 sec	5135 sec	353
parabolic_fem	20.06 sec	7054 sec	352
tmt_sym	13.98 sec	$>$ 3 hours	$>$ 772
ecology2	16.04 sec	$>$ 3 hours	$>$ 673
G3_circuit	218.7 sec	$>$ 3 hours	$>$ 49

Table 4.4: Timing comparison between selected inversion and direct inversion. The speedup factor is defined by the direct inversion time divided by the selected inversion time.

4.3.4 Application to electronic structure calculation of aluminum

Here we show how SelInv can be applied to electronic structure calculations based on the pole expansion introduced in Chapter 3. We need to compute the diagonal of the inverse of a number of complex symmetric (non-Hermitian) matrices $H-(z_i+\mu)I$ ($i = 1, 2, ..., P$). A fast implementation of the SelInv algorithm described in Section 4.3.2 can be used to perform this calculation efficiently.

The example we consider here is a quasi-2D aluminum system with a periodic boundary condition. For simplicity, we only use a local pseudopotential, *i.e.* $V_{\text{pse}}(\boldsymbol{r})$ is a diagonal matrix. The Laplacian operator Δ is discretized using a second-order seven-point stencil. A room temperature of 300K (which defines the value of β) is used. The aluminum system has a face centered cubic (FCC) crystal structure. We include 5 unit cells along both x and y directions, and 1 unit cell along the z direction

in our computational domain. Each unit cell is cubic with a lattice constant of 4.05Å. Therefore, there are altogether 100 aluminum atoms and 300 valence electrons in the experiment. The position of each aluminum atom is perturbed from its original position in the crystal by a random displacement around 10^{-3}Å so that no point group symmetry is assumed in our calculation. The grid size for discretization is set to 0.21Å. The resulting Hamiltonian matrix size is $159,048$.

We compare the density evaluation performed by both PARSEC and the pole expansion technique. In PARSEC, the invariant subspace associated with the smallest 310 eigenvalues is computed using ARPACK [151]. Each self-consistent iteration step takes $2,490$ seconds. In the pole expansion approach, we use 60 poles, which gives a comparable relative error in electron density on the order of 10^{-5} (in L_1 norm.) The MMD reordering scheme is used to reduce the amount of fill in the LDL^T factorization. In addition to using the selected inversion algorithm to evaluate each term, an extra level of coarse grained parallelism can be utilized by assigning each pole to a different processor. The evaluation of each term takes roughly $1,950$ seconds. Therefore, the total amount of time required to evaluate the electron density for each self-consistent iteration step on a single core is $1,950 \times 60$ seconds. As a result, the performance of the selected inversion based pole expansion approach is only comparable to the invariant subspace computation approach used in PARSEC if the extra level of coarse grained parallelism is used.

A 3D isosurface plot of the electron density as well as the electron density plot restricted on the $z = 0$ plane are shown in Figure 4.5.

We also remark that the efficiency of selected inversion can be further improved for this particular problem. One of the factors that has prevented the SelInv from achieving even higher performance for this problem is that most of the supernodes produced from the MMD ordering of H contains only 1 column even though many of these supernodes have similar (but not identical) nonzero structures. Consequently,

Figure 4.5: (a)3D isosurface plot of the electron density together with the electron density restricted to $z = 0$ plane. (b) The electron density restricted to $z = 0$ plane.

both the factorization and inversion are dominated by level 1 BLAS operations. Further performance gain is likely to be achieved if we relax the definition of a supernode and treat some of the zeros in L as nonzero elements. This approach has been demonstrated to be extremely helpful in [9].

4.4 Parallel selected inversion algorithm

4.4.1 Algorithmic and implementation

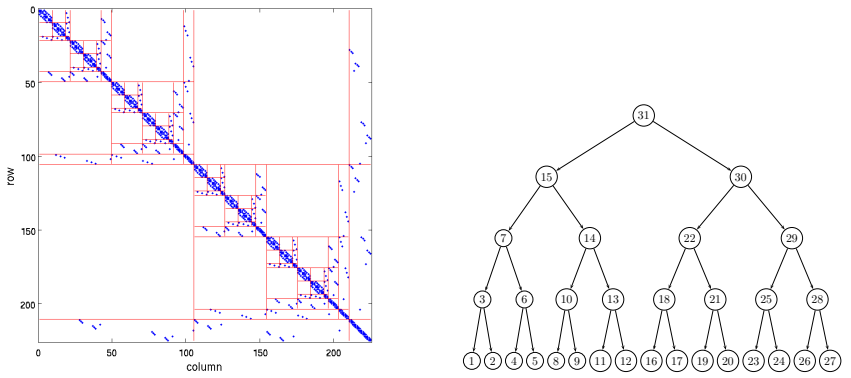
In this subsection, we present the algorithmic and implementation of a parallel procedure for selected inversion. Our algorithm is quite general as long as a block LDL^T factorization is available. We make use of the elimination tree and other structure information that can be generated during a preprocessing step that involves both matrix reordering and symbolic factorization. For illustration purpose, we use a 2D Laplacian with nearest neighbor interaction, where the nearest neighbor is defined in terms of a standard 5-point stencil, as an example in this section. However, the techniques we describe here are applicable to other higher order stencils for both 2D and 3D systems and to irregular problems obtained from, e.g., a finite element discretization. Although we have developed an efficient parallel implementation of a

supernodal LDL^T factorization for 2D problems, we will focus our discussion on the selected inversion procedure only.

The Sequential Algorithm Before we present the sequential algorithms for the selected inversion process, we need to introduce some notations and terminologies commonly used in the sparse matrix literature. We use the technique of *nested dissection* [92] to reorder and partition the sparse matrix A. The reordered matrix has a sparsity structure similar to that shown in Fig. 4.6a. For 2D problems defined on a rectangular grid, nested dissection corresponds to a recursive partition of the 2D grid into a number of subdomains with a predefined minimal size. In the example shown in Fig. 4.7a, this minimal size is 3×3. Each subdomain is separated from other subdomains by *separators* that are defined in a hierarchical or recursive fashion. The largest separator is defined to be a set of grid points that divides the entire 2D grid into two subgrids of approximately equal sizes. Smaller separators can be constructed recursively within each subgrid. These separators are represented as rectangular oval boxes in Fig. 4.7a and are labeled in post order in Fig. 4.7b. The separators and minimal subdomains can be further organized in a tree structure shown in Fig. 4.6b. This tree is sometimes called a separator tree, which is also the elimination tree associated with a block LDL^T factorization of the reordered and partitioned matrix A. Each leaf node of the tree corresponds to a minimal subdomain. Other nodes of the tree correspond to separators defined at different levels of the partition. For general symmetric sparse matrices, separators and leaf nodes can be obtained from the analysis of the adjacency graph associated with the nonzero structure of A [133].

We will denote a set of row or column indices associated with each node in the separator tree by an uppercase typewriter typeface letter such as `I`. Each one of these nodes corresponds to a diagonal block in the block diagonal matrix D produced from the block LDL^T factorization. A subset of columns in `I` may have a similar

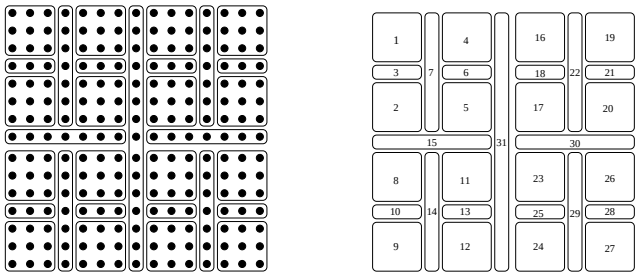
nonzero structure below the diagonal block. These columns can be grouped together to form what is known as a *supernode* or a *relaxed supernode*. (See [77] for a more precise definition of a supernode, and [9] for the definition of a relaxed supernode.) Sparse direct methods often take advantage of the presence of supernodes or relaxed supernodes in the reordered matrix A to reduce the amount of indirect addressing. Because the nonzero matrix element within a supernode can be stored as a dense matrix, we can take full advantage of `BLAS3` when working with supernodes.



(a) The reordered and partitioned matrix A

(b) The separator (or elimination) tree in postorder

Figure 4.6: The separator tree associated with the nested dissection of the 15×15 grid shown in Fig. 4.7a can also be viewed as the elimination tree associated with a block LDL^T factorization of the 2D Laplacian defined on that grid.



(a) Nested dissection partition and separators

(b) The ordering of separators and subdomains.

Figure 4.7: The nested dissection of a 15×15 grid and the ordering of separators and subdomains associated with this partition.

Once the separator tree and the block LDL^T factorization of A become available,

we can use the pseudocode shown in Algorithm 5 to perform block selected inversion. As we can see from this pseudocode that `Ainv(J,K)` is calculated if and only if `L(J,K)` is a nonzero block. Such a calculation makes use of previously calculated blocks `Ainv(J,I)`, where both `J` and `I` are ancestors of the node `K`.

Algorithm 5 A selected inversion algorithm for a sparse symmetric matrix A given its block LDL^T factorization $A = LDL^T$.

```
for K = {separator tree nodes arranged in reverse post
order} do
  for J ∈ {ancestors of K} do
    Ainv(J,K) ← 0;
    for I ∈ {ancestors of K} do
      Ainv(J,K) ← Ainv(J,K) − Ainv(J,I) ∗ L(I,K);
    end for
    Ainv(K,J) ← Ainv(J,K)^T;
  end for
  Ainv(K, K) ← D(K,K)^{-1};
  for J ∈ {ancestors of K} do
    Ainv(K,K) ← Ainv(K,K) − L(J,K)^T ∗ Ainv(J,K);
  end for
end for
```

The pseudocode in Algorithm 5 treats the matrix block `L(J,I)` as if it is a dense matrix. As we can see from Fig. 4.6a, this is clearly not the case. In order to carry out the matrix-matrix multiplication efficiently, we must take advantage of these sparsity structures. In particular, we should not store the zero rows and columns in `L(I,K)`. Moreover, during the calculation of `Ainv(J,K)`, selected rows and columns of `Ainv(J,I)` must be extracted before the submatrix associated with these rows and columns are multiplied with the corresponding nonzero rows and columns of `L(I,K)`. We place the extracted rows and columns in a `Buffer` array in Algorithm 6. The `Buffer` array is then multiplied with the corresponding nonzero columns of `L(I,K)`. As a result, the product of the nonzero rows and columns of these matrices will have a smaller dimension. We will call the multiplication of the nonzero rows and columns of `Buffer` and `L(I,K)` a *restricted* matrix-matrix multiplication, and denote it by $\otimes$. The row and column indices associated with the needed rows and columns of `Ainv(J,I)`

are called *absolute indices*. These indices can be predetermined by a *symbolic analysis* procedure, and they are retrieved by calling the `GetAbsIndex` function in Algorithm 6 that shows how restricted multiplication is used in the selected inversion process.

Algorithm 6 Selected inversion of `A` with restricted matrix-matrix multiplication given its block LDL^T factorization.

```
subroutine SeqSelInverse
for K = {separator tree nodes arranged in reverse post order} do
  for J ∈ {ancestors of K} do
    Ainv(J,K) ← 0;
    for I ∈ {ancestors of K} do
      [IA,JA] ← GetAbsIndex(L,K,I,J);
      Buffer ← selected rows and columns of Ainv(J,I) starting at (IA, JA);
      Ainv(J,K) ← Ainv(J,K) − Buffer ⊗ L(I,K);
    end for
    Ainv(K,J) ← transpose(Ainv(J,K));
  end for
  Ainv(K,K) ← D(K,K)^{-1};
  for J ∈ {ancestors of K} do
    Ainv(K,K) ← Ainv(K,K) − L(J,K)^T ⊗ Ainv(J,K);
  end for
end for
return Ainv;
end subroutine
```

Parallelization The sequential algorithm described above is very efficient for problems that can be stored on a single processor. For example, we have used the algorithm to compute the diagonal of a discretized Kohn-Sham Hamiltonian defined on a 2047×2047 grid. The entire computation, which involves more than 4 million degrees, took less than 2 minutes on an AMD Opteron processor.

For larger problems that we would like to solve in electronic structure calculation, the limited amount of memory on a single processor makes it difficult to store the L and D factors in-core. Furthermore, because the complexity of the computation is $\mathcal{O}(n^{3/2})$ in 2D [163], the CPU time required to complete a calculation on a single processor will eventually become excessively long.

Thus, it is desirable to modify the sequential algorithm so that the selected inversion process can be performed in parallel on multiple processors. The parallel algorithm we describe below focuses on distributed memory machines that do not share a common pool of memory.

Task parallelism and data distribution The elimination tree associated with the block LDL^T factorization of the reordered A (using nested dissection) provides natural guidance for parallelizing the factorization calculation. It can thus be viewed also as a *parallel task tree*. The same task tree can be used for carrying out selected inversion.

We divide the computational work among different branches of the tree. A *branch* of the tree is defined to be a path from the root to a node `K` at a given level ℓ as well as the entire subtree rooted at `K`. The choice of ℓ depends on the number of processors available. For a perfectly balanced tree, our parallel algorithm requires the number of processors p to be a power of two, and ℓ is set to $\log_2(p) + 1$. Fig. 4.8a illustrates the parallel task tree in the case of 4 processors.

In terms of tree node to processor mapping, each node at level ℓ or below is assigned to a unique processor. Above level ℓ, each node is shared by multiple processors. The amount of sharing is hierarchical, and depends on the level at which the node resides. For a perfectly balanced tree, a level-k node is shared by $2^{\ell-k}$ processors. We will use `procmap(J)` in the following discussion to denote the set of processors assigned to node `J`. Each processor is labeled by an integer processor identification (id) number between 0 and $p-1$. This processor id is known to each processor as `mypid`. In section 4.4.2, we show that this simple parallelization strategy leads to good load balance for a 2D Hamiltonian defined on a rectangular domain and discretized with a five point stencil. For irregular computational domain or non-uniform mesh partitioning strategy, more complicated task-to-processor mapping algorithms

should be used [210] to take into account the structure of the separator tree. It may also be necessary to perform task scheduling on-the-fly [3].

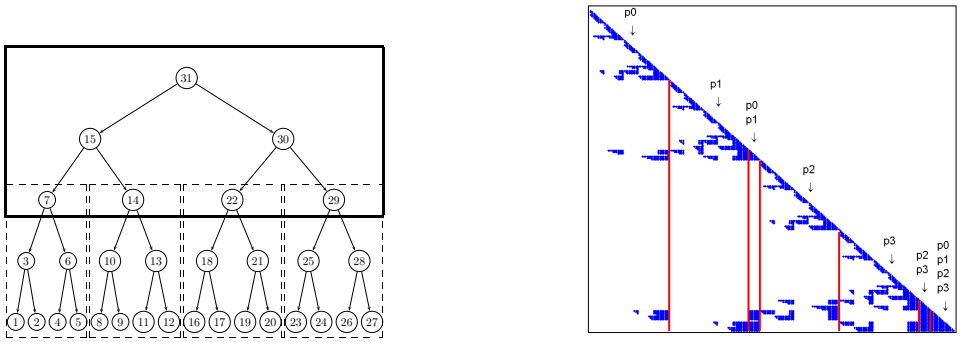
The data distribution scheme used for selected inversion is compatible with that used for LDL^T factorization. We should emphasize that the matrix `D(J,J)` in our implementation of the block LDL^T factorization is not necessarily diagonal. Again, we do not store the entire submatrix `L(I,J)`, but only the nonzero subblock within this submatrix as well as the starting location of the nonzero subblock.

In our parallel LDL^T factorization computation, the `L(I,J)` and `D(J,J)` submatrices associated with any `J` in an aggregated leaf node are stored on a single processor p to which the aggregated leaf node is assigned. These matrices are computed using a sequential sparse LDL^T factorization algorithm on this processor. Furthermore, this computation is done independently from that of other processors.

When `J` is an ancestor of an aggregated leaf node, computing `L(I,J)` and `D(J,J)` requires the participation of all processors that are assigned to this node `procmap(J)`. As a result, it is natural to divide the nonzero subblock in `L(I,J)` and `D(J,J)` into smaller submatrices, and distribute them among all processors that belong to `procmap(J)`. Fig. 4.8b illustrates how the columns of the L factor are partitioned and distributed among 4 processors.

Distributing these smaller submatrices among different processors is also necessary for overcoming the memory limitation imposed by a single processor. For example, for a 2D Hamiltonian defined on a $16,383 \times 16,383$ grid, the dimension of `D(J,J)` is $16,383$ for the root node `J`. This matrix is completely dense, hence contains $16,383^2$ matrix elements. If each element is stored in double precision, the total amount of memory required to store `D(J,J)` alone is roughly 2.1 gigabytes (GB). As we will see in section 4.4.2, the distribution scheme we use in our parallel algorithm leads to only a mild increase of memory usage per processor as we increase the problem size and the number of processors in proportion.

To achieve a good load balanced, we use a 2D block cyclic mapping consistent with that used by `ScaLAPACK` to distribute the nonzero blocks of `L(I,J)` and `D(J,J)` for any `J` that is an ancestor of an aggregated leaf node. In our parallel selected inversion algorithm, the distributed nonzero blocks of `L(I,J)` and `D(J,J)` are overwritten by the corresponding nonzero blocks of `Ainv(I,J)` and `Ainv(J,J)`.



(a) Parallel task tree

(b) Columns of the L factor are partitioned and distributed among different processors.

Figure 4.8: Task parallelism expressed in terms of parallel task tree and corresponding matrix to processor mapping.

Parallel selected inversion algorithm Once the task-to-processor mapping and the initial data distribution is established, the parallelization of the selected inversion process can be described in terms of operations performed on different branches of the parallel task tree simultaneously by different processors. As illustrated in the subroutine `ParSelInverse` in Algorithm 8, each processor moves from the root of the task tree down towards to an aggregated leaf node along a particular branch identified by `mybranch`. At each node `K`, it first computes `Ainv(J,K)` for ancestors `J` of `K` that satisfy $\texttt{L}(\texttt{J},\texttt{K}) \neq 0$. This calculation is followed by the computation of the diagonal block `Ainv(K,K)`. These two operations are accomplished by the subroutine `ParExtract` shown in the left column of Algorithm 8. Before moving one step further along `mybranch`, all processors belonging to `procmap(K)` perform some additional data redistribution by calling the subroutine `ParRestrict` listed in the right

column of Algorithm 8, so that selected components of `Ainv(J,K)` that will be needed in the subsequent calculation of `Ainv(J,I)` for all descendents `I` of `K` are placed at appropriate locations in a `Buffer` array created for each child of `K`. This step is essential for reducing synchronization overhead and will be discussed in detail later. After `ParRestrict` is called, no communication is required between the processors assigned to different children of `K`. Finally, when each processor reaches an aggregated leaf node `K`, it calls the sequential selected inversion subroutine `SeqSelInverse` (Algorithm 6) to compute `Ainv(J,I)` for all descendents `I` of `K`. No inter-processor communication is required from this point on.

Algorithm 7 Parallel algorithm from extracting selected elements of the inverse of a symmetric matrix A.

subroutine `ParSelInverse`
`K` $\leftarrow$ `root`;
while (`K` is not an aggregated leaf node) **do**
  Update `Ainv(K,J)` for all `J = ancestor(K)` by calling `ParExtract(K);`
  Update Buffer by calling `ParRestrict(K);`
  `K` $\leftarrow$ `child(K)` along `mybranch`;
end while
Call Sequential algorithm to obtain `Ainv` at the leaf node;
return `Ainv`;
end subroutine

Avoiding synchronization bottleneck Avoiding synchronization bottleneck is the key to achieving scalable performance in selected inversion. Synchronization is needed in selected inversion as each processor proceeds from the root of the parallel task tree to an aggregated leaf node because the submatrix `L(I,K)` required at a particular node `K` of the parallel task tree is distributed in a block cyclic fashion among a larger group of processors that are mapped to the ancestors of `K`. Some of these processors will not participate in the computation of `Ainv(K,K)`. Therefore, data redistribution is required to move the required matrix elements from this larger group of processors to the set of processors in `procmap(K)`.

We use the `ScaLAPACK` subroutine `PDGEMR2D` to perform such a data redistribution.

Algorithm 8 Parallel implementation of selected inversion of A given its block LDL^T factorization.

```
subroutine ParExtract(K)
for J ∈ {ancestors of K} do
  Ainv(J,K) ← 0;
  for I ∈ {ancestors of K}
  do
    Ainv(J,K) ← Ainv(J,K)−
      Buffer(J,I) ⊗ L(I,K);
  end for
  Ainv(K,J) ←
  Ainv(J,K)^T;
end for
Ainv(K,K) ← D(K,K);
for J ∈ {ancestors of K} do
  Ainv(K,K) ← Ainv(K,K)−
    L(J,K)^T ⊗ Ainv(J,K);
end for
return Ainv;
end subroutine
```

```
subroutine ParRestrict(K)
if (K is the root) then
  Buffer ← D(K,K);
end if
for C ∈ {children of K} do
  for all I,J ∈ {ancestors of K} do
    if L(J,C) ≠ 0 and L(I,C) ≠ 0)
    then
      [IR,JR] ←
      GetRelIndex(C,K,I,J);
      Restrict Buffer(J,I) to a sub-
      matrix starting at (IR, JR).
    end if
  end for
end for
return Buffer;
end subroutine
```

When PDGEMR2D is called to redistribute data from a larger processor group A to a smaller processor group B that is contained in A, all processors in A are *blocked*, meaning that no processor in A can proceed with its own computational work until the data redistribution initiated by processors in B is completed. This blocking feature of PDGEMR2D, while necessary for ensuring data redistribution is done in a coherent fashion, creates a potential synchronization bottleneck.

To be specific, when the selected nonzero rows and columns in Ainv(J,I) (Algorithm 8) are to be extracted from a large number of processors in procmap(I) and redistributed among a subset of processors in procmap(K), a direct extraction and redistribution via the use of PDGEMR2D will block all processors in procmap(I). If K is several levels away from I, a communication bottleneck that involves all processors in procmap(I) is created. This bottleneck makes the computation of Ainv(J,K) a sequential process for all descendants K of I that are at the same level.

The strategy we use to overcome this synchronization bottleneck is to place se-

lected nonzero elements of `Ainv(J,I)` that would be needed for subsequent calculations in a `Buffer` array. Selected subblocks of the `Buffer` array will be passed further to the descendants of `I` as each processor moves down the parallel task tree. The task of extracting necessary data and placing it in `Buffer` is performed by the subroutine `ParRestrict` shown in Algorithm 8. At a particular node `I`, the `ParRestrict` call is made simultaneously by all processors in `procmap(I)`, and the `Buffer` array is distributed among processors assigned to each child of `I` so that the multiplication of the nonzero subblocks of `Ainv(J,I)` and `L(J,K)` can be carried out in parallel (by `pdgemm`). Because this distributed `Buffer` array contains all information that would be needed by descendants of `K`, no more direct reference to `Ainv(J,I)` is required for any ancestor `I` of `K` from this point on. As a result, no communication is performed between processors that are assigned to different children of `I` once `ParRestrict` is called at node `I`.

As each processor moves down the parallel task tree within the `while` loop of the subroutine `ParSelInverse` in Algorithm 7, the amount of data extracted from the `Buffer` array by the `ParRestrict` subroutine becomes smaller and smaller. The new extracted data is distributed among a smaller number of processors also. Each call to `ParRestrict(I)` requires a synchronization of all processors in `procmap(I)`, hence incurring some synchronization overhead. This overhead becomes smaller as each processor gets closer to an aggregated leaf node because each `ParRestrict` call is then performed within a small group of processors. When an aggregated leaf node is reached, all selected nonzero rows and columns of `Ainv(J,I)` required in subsequent computation are available in the `Buffer` array allocated on each processor. As a result, no communication is required among different processors from this point on.

Since the desired data in the `Buffer` array is passed level by level from a parent to its children, we only need to know the relative positions of the subblocks needed by a child within the `Buffer` array owned by its parent. Such positions can be recorded

by *relative indices* that are obtained by the subroutine `GetRelIndex` in Algorithm 8. They are used for data extraction in `ParRestrict`. The use of relative indices is not necessary when each process reaches a leaf node at which the sequential selected inversion subroutine `SeqSelInverse` is called.

4.4.2 Performance of the parallel selected inversion algorithm

In this subsection, we report the performance of our implementation of the selected inversion algorithm for a discretized 2D Kohn-Sham Hamiltonian H using five-point stencil with a zero shift, which we will refer to as **PSelInv** in the following. The nested dissection procedure stops when the dimension of the subdomain is 3×3. We analyze the performance statistics by examining several aspects of the implementation that affect the efficiency of the computation and communication. Our performance analysis is carried out on the Franklin system maintained at National Energy Research Scientific Computing (NERSC) Center. Franklin is a distributed-memory parallel system with 9,660 compute nodes. Each compute node consists of a 2.3 GHz single socket quad-core AMD Opteron processor (Budapest) with a theoretical peak performance of 9.2 gigaflops per second (Gflops) per core. Each compute node has 8 gigabyte (GB) of memory (2 GB per core). Each compute node is connected to a dedicated SeaStar2 router through Hypertransport with a 3D torus topology that ensures high performance, low-latency communication for MPI. The floating point calculation is done in 64-bit double precision. We use 32-bit integers to keep index and size information.

Our implementation of the selective inversion achieves very high single processor performance. In particular, when the grid size reaches $2,047$, we are able to reach 67% (6.16/9.2) of the peak performance of a single Franklin core.

Here we will mainly focus on the parallel performance of our algorithm and implementation. Our objective for developing a parallel selected inversion algorithm is

to enable us and other researchers to study the electronic structure of large quantum mechanical systems when a vast amount of computational resource is available. Therefore, our parallelization is aimed at achieving a good *weak scaling*. Weak scaling refers to a performance model similar to that used by Gustafson [112]. In such a model, performance is measured by how quickly the wall clock time increases as both the problem size and the number of processors involved in the computation increase. Because the complexity of the factorization and selected inversion procedures is $\mathcal{O}(n^{3/2})$, where n is the matrix dimension and m is the number of grids in one dimension. We will simply call m the *grid size* in the following. Clearly $n = m^2$. We also expect that, in an ideal scenario, the wall-clock time should increase by a factor of two when the grid size doubles and the number of processor quadruples.

In addition to using `MPI_Wtime()` calls to measure the wall clock time consumed by different components of our code, we also use the Integrated Performance Monitoring (IPM) tool [228], the `CrayPat` performance analysis tool [129] as well as `PAPI` [44] to measure various performance characteristics of our implementation.

Single Processor Performance We first report the performance of selected inversion algorithm when it is executed on a single processor. The single processor performance is measured in terms of the CPU time and the floating point operations performed per second (flops). Table 4.5 lists the performance characteristic of single processor calculations for Hamiltonians defined on square grids with different sizes. We choose the grid size m to be $m = 2^\ell - 1$ for some integer $\ell > 1$ so that a perfectly balanced elimination tree is produced from a nested dissection of the computational domain.

The largest problem we can solve on a single processor contains $2,047 \times 2,047$ grid points. The dimension of the corresponding matrix is over 4 million. The memory requirement for solving problems defined on a larger grid (with $\ell > 11$) exceeds what

grid size	matrix dimension	symbolic	factorization	inversion	total	Gflops
127	16,129	0.01	0.04	0.03	0.08	1.29
255	65,025	0.05	0.21	0.17	0.43	2.57
511	261,121	0.22	1.18	1.03	2.43	3.89
1023	1,046,529	0.93	7.29	6.76	15.0	5.12
2047	4,190,209	4.21	48.8	47.3	100.3	6.15

Table 4.5: Single processor performance

is available on a single node of the Franklin system. Thus they can only be solved in parallel using multiple processors.

We can clearly see from Table 4.5 that the symbolic analysis of the LDL^T factorization takes a small fraction of the total time, especially when the problem is sufficiently large. The selected inversion calculation (after a block LDL^T factorization has been performed) takes slightly less time to complete than that required by the factorization. The total CPU time listed in the 6th column of the table confirms the $\mathcal{O}(n^{3/2})$ complexity.

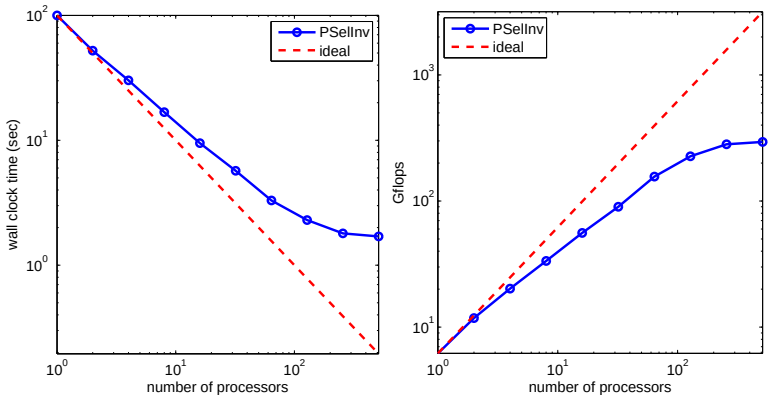
We also observe that a high flops rate is achieved for larger problems. In particular, when the grid size reaches $2,047$, we achieve 67% (6.16/9.2) of peak performance of the machine. This is due to the fact that as the problem size increases, the overall computation is dominated by computation performed on the dense matrix blocks associated with large supernodes. Therefore the performance approaches that of dense matrix-matrix multiplications.

Parallel Scalability We report the performance of our implementation when it is executed on multiple processors. Our primary interest is in the weak scaling of the parallel computation with respect to an increasing problem size and an increasing number of processors. The strong scaling of our implementation for a problem of fixed size is described in Table 4.6. We report the wall clock time (in seconds) for selected inversion of A^{-1} defined on a $2,047 \times 2,047$ grid. In the third column of the

table, we report also the speedup factor defined as $\tau = t_1/t_{n_p}$, where t_{n_p} is the wall clock time recorded for an n_p-processor run.

n_p	wall clock time	speedup factor	Gflops
1	100.1	1.0	6.2
2	52.2	1.9	11.8
4	30.2	3.3	20.2
8	16.8	6.0	33.5
16	9.5	10.5	55.9
32	5.7	17.6	90.0
64	3.3	30.3	156.2
128	2.3	42.3	226.4
256	1.8	55.6	281.7
512	1.7	58.9	294.2

Table 4.6: The scalability of parallel computation used to obtain A^{-1} for A of a fixed size ($n = 2047 \times 2047$.)



(a) (b)

Figure 4.9: Log-log plot of total wall clock time and total Gflops with respect to number of processors, compared with ideal scaling. The grid size is fixed at 2047×2047.

Figure 4.9 compares the performance of our algorithm, called PSelInv, with ideal scaling in terms of total wall clock time and total Gflops. As we can clearly see from Table 4.6 and Figure 4.9, for problem of this fixed size, deviation from the ideal speedup begins to show up when the computation is performed in parallel on 4

processors. The performance barely improves in a 512-processor run compared to the 256-processor run. Beyond that point, communication overhead starts to dominate. We will discuss the sources of communication overhead in the next few subsections.

In terms of weak scaling, PSelInv performs quite well with up to 4,096 processors for problems defined on a $65,535 \times 65,535$ grid (with corresponding matrix dimension around 4.3 billion). In Table 4.7, we report the wall clock time recorded for several runs on problems defined on square grids of different sizes. To measure weak scaling, we start with a problem defined on a $1,023 \times 1,023$ grid, which is solved on a single processor. When we double the grid size, we increase the number of processors by a factor of 4. In an ideal scenario in which communication overhead is small, we should expect to see a factor of two increase in wall clock time every time we double the grid size and quadruple the number of processors used in the computation. Such prediction is based on the $\mathcal{O}(m^3)$ complexity of the computation. In practice, the presence of communication overhead will lead to a larger amount of increase in total wall clock time. Hence, if we use $t(m, n_p)$ to denote the total wall clock time used in an n_p-processor calculation for a problem defined on a square grid with grid size m, we expect the weak scaling ratio defined by $\tau(m, n_p) = t(m/2, n_p/4)/t(m, n_p)$, which we show in the last column of Table 4.7, to be larger than two. However, as we can see from this table, deviation of $\tau(m, n_p)$ from the ideal ratio of two is quite modest even when the number of processors used in the computation reaches $4,096$.

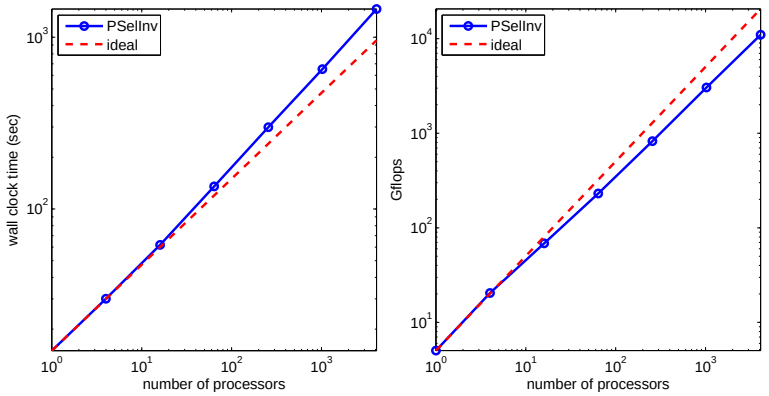
A closer examination of the performance associated with different components of our implementation reveals that our parallel symbolic analysis takes a nearly constant amount of time that is a tiny fraction of the overall wall clock time for all configurations of problem size and number of processors. This highly scalable performance is primarily due to the fact that most of the symbolic analysis performed by each processor is carried out within an aggregated leaf node that is completely independent from other leaf nodes.

Table 4.7 shows that the performance of our block selected inversion subroutine achieves nearly ideal weak scaling up to $4,096$ processors. The scaling of flops and wall clock time can be better viewed in Fig. 4.10, in which the code performance is compared to ideal performance using a log-log plot. We should point out that the performance of our implementation of the parallel LDL^T factorization is comparable to that achieved by the state-of-art sparse matrix software packages such as MUMPS [3] on relatively small 2D problem used in our experiment even though our factorization includes the additional computation of using the ScaLAPACK subroutines `pdgetri` to invert the diagonal blocks of D. (We have not been able to use MUMPS to factor problems that are discretized with 8191×8191 or more grid points.) From Table 4.7, we can also see that the selected inversion time is significantly less than that associated with factorization when the problem size becomes sufficiently large. This is due primarily to the fact that selected inversion involves less amount of indirect addressing, and almost all float point operations involved in block selected inversion are dense matrix-matrix multiplications.

grid size	n_p	symbolic time	factorization time	inversion time	total time	weak scaling ratio
1,023	1	0.92	7.29	6.77	14.99	–
2,047	4	1.77	14.44	13.82	30.04	2.00
4,095	16	1.82	34.26	25.39	61.82	2.05
8,191	64	1.91	86.35	47.07	135.34	2.18
16,383	256	1.98	207.51	89.91	299.41	2.21
32,767	1024	2.08	474.94	174.57	651.59	2.17
65,535	4096	2.40	1109.09	348.13	1459.62	2.24

Table 4.7: The scalability of parallel computation used to obtain A^{-1} for A for increasing system sizes. The largest grid size is $65,535 \times 65,535$ and corresponding matrix size is approximately 4.3 billion.

Load Balance To have a better understanding of the parallel performance of our code, let us now examine how well the computational load is balanced among different processors. Although we try to maintain a good load balance by distributing the



(a) (b)

Figure 4.10: Log-log plot of total wall clock time and total Gflops with respect to number of processors, compared with ideal scaling. The grid size starts from 1023×1023, and is proportional to the number of processors.

nonzero elements in `L(I,J)` and `D(J,J)` as evenly as possible among processors in `procmap(J)`, such a data distribution strategy alone is not enough to achieve perfect load balance as we will see below.

One way to measure load balance is to examine the flops performed by each processor. We collected such statistics by using PAPI [44]. Fig. 4.11 shows the overall flop counts measured on each processor for a 16-processor run of the selected inversion for A defined on a $4,095 \times 4,095$ grid. There is clearly some variation in operation counts among the 16 processors. Such variation contributes to idle time that shows up in the communication profile of the run, which we will report in the next subsection. Such variation can be explained by how the separator tree nodes are order and its relationship with the 2D grid topology.

Communication Overhead A comprehensive measurement of the communication cost can be collected using the `IPM` tool. Table 4.8 shows the overall communication cost increases moderately as we double the problem size and quadruple the number of processors at the same time.

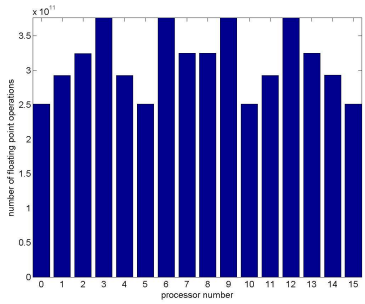


Figure 4.11: The number of flops performed on each processor for the selected inversion of A^{-1} defined on a $4,095 \times 4,095$ grid.

grid size	n_p	communication (%)
1,023	1	0
2,047	4	2.46
4,095	16	11.14
8,191	64	20.41
16,383	256	28.43
32,767	1024	34.46
65,535	4096	40.80

Table 4.8: Communication cost as a percentage of the total wall clock time.

As we discussed earlier, the communication cost can be attributed to the following three factors:

1. Idle wait time. This is the amount of time a processor spends waiting for other processors to complete their work before proceeding beyond a synchronization point.

2. Communication volume. This is the amount of data transfered among different processors.

3. Communication latency. This factor pertains to the startup cost for sending a single message. The latency cost is proportional to the total number of messages communicated among different processors.

The communication profile provided by `IPM` shows that `MPI_Barrier` calls are the largest contributor to the communication overhead. An example of such a profile obtained from a 16-processor run on a $4,095 \times 4,095$ grid is shown in Fig. 4.12. In this particular case, `MPI_Barrier` represents more than 50% of all communication cost. The amount of idle time the code spent in this MPI function is roughly 6.3% of the overall wall clock time.

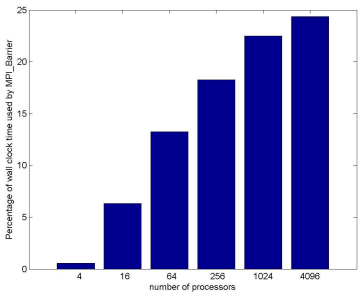
The `MPI_Barrier` and `BLACS_Barrier` (which shows up in the performance profile as `MPI_Barrier`) functions are used in several places in our code. In particular, the barrier functions are used in the selected inversion process to ensure relative indices are properly computed by each processor before selected rows and columns of the matrix block associated with a higher level node are redistributed to its descendants. The idle wait time spent in these barrier function calls is due to the variation of computational loads. Using the call graph provided by `CrayPat`, we examined the total amount of wall clock time spent in these `MPI_Barrier` calls. For the 16-processor run (on the $4,095 \times 4,095$ grid), this measured time is roughly 2.6 seconds, or 56% of all idle time spent in `MPI_Barrier` calls. The rest of the `MPI_Barrier` calls are

[name]	[time]	[calls]	<%mpi>	<%wall>
MPI_Barrier	67.7351	960	52.21	6.32
MPI_Recv	30.4719	55599	23.49	2.84
MPI_Reduce	16.6104	18260	12.80	1.55
MPI_Send	7.86273	25865	6.06	0.73
MPI_Bcast	5.86476	100408	4.52	0.55
MPI_Allreduce	0.842473	320	0.65	0.08
MPI_Isend	0.261145	29734	0.20	0.02
MPI_Testall	0.0563367	33515	0.04	0.01
MPI_Sendrecv	0.0225533	1808	0.02	0.00
MPI_Allgather	0.00237397	16	0.00	0.00
MPI_Comm_rank	8.93647e-05	656	0.00	0.00
MPI_Comm_size	1.33585e-05	32	0.00	0.00

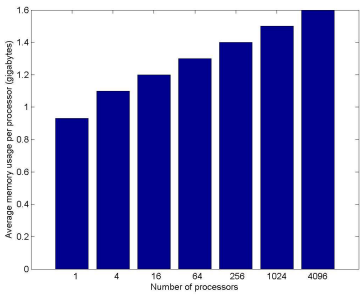
Figure 4.12: Communication profile for a 16-processor run on a $4,095 \times 4,095$ grid.

made in ScaLAPACK matrix-matrix multiplication routine `pdgemm`, dense matrix factorization and inversion routines `pdgetrf` and `pdgetri`, respectively.

Fig. 4.13a shows that the percentage of wall clock time spent in `MPI_Barrier` increases moderately as more processors are used to solve larger problems. Such increase is due primarily to the increase in the length of the critical path in both the elimination tree and in the dense linear algebra calculations performed on each separator.



(a) The percentage of time spent in `MPI_Barrier` as a function of n_p (and the corresponding grid size m).



(b) The average memory usage per processor as a function of n_p and m.

Figure 4.13: Communication overhead and memory usage profile

In addition to the idle wait time spent in `MPI_Barrier`, communication overhead

is also affected by the volume of data transfered among different processors and how frequent these transfers occur. It is not difficult to show that the total volume of communication should be proportional to the number of nonzeros in L and independent from the number of processors used. Fig. 4.12 shows that the total amount of wall clock time spent in MPI data transfer functions `MPI_Send`, `MPI_Recv`, `MPI_ISend`, `MPI_Reduce`, `MPI_Bcast` and `MPI_Allreduce` etc. is less than 5% of the overall wall clock time for a 16-processor run on a $4,095 \times 4,095$ grid. Some of the time spent in `MPI_Recv` and collective communication functions such as `MPI_Reduce` and `MPI_Bcast` corresponds to idle wait time that are not accounted for in `MPI_Barrier`. Thus, the actual amount of time spent in data transfer is much less than 5% of the total wall clock time. This observation provides an indirect measurement of the relatively low communication volume produced in our calculation.

In terms of the latency cost, we can see from Fig. 4.12 that the total number of MPI related function calls made by all processors is roughly 258,000 (obtained by adding up the call numbers in the third column). Therefore, the total number of messages sent and received per processor is roughly 16,125. The latency for sending one message on Franklin is roughly 8 microsecond. Hence, the total latency cost for this particular run is estimated to be roughly 0.13 seconds, a tiny fraction of the overall wall clock time. Therefore, latency does not contribute much to communication overhead.

Memory Consumption In addition to maintaining good load balance among different processors, the data-to-processor mapping scheme also ensures that the memory usage per core only increases logarithmically with respect to the matrix dimension in the context of weak scaling. This estimation is based on the observation that when the grid size is increased by a factor of two, the dimension of the extra blocks associated with `L` and `D` to are proportional to the grid size, and the total amount of extra memory requirement is proportional to the square of the grid size. Since the number

of processors is increased by a factor of four, the extra memory requirement stays fixed regardless of the grid size. This logarithmic dependence is clear from Fig. 4.13b, where the average memory cost per core with respect to number of processors is shown. The x-axis is plotted in logarithmic scale.

4.4.3 Application to electronic structure calculation of 2D rectangular quantum dots

We now show how the parallel selected inversion algorithm can be used to speed up electronic structure calculations. The example we use here is a 2D electron quantum dot confined in a rectangular domain, a model investigated in [216] with the local density approximation (LDA) for the 2D exchange-correlation functional [13]. This model is also provided in the test suite of the Octopus software [50], which we use for comparison.

We calculate the electron density using the pole expansion introduced in section 3.3. In this example, the Laplacian operator Δ is discretized using a five-point stencil. The electron temperature is set to be 300K. The area of the quantum dot is L^2. In a two-electron dot, setting $L = 1.66$Å and discretizing the 2D domain with 31×31 grid points yields an total energy error that is less than 0.002Ha. When the number of electrons becomes larger, we increase the area of the dot in proportion so that the average electron density is fixed. A typical density profile with 32 electrons is shown in Fi. 4.14. In this case, the quantum dot behaves like a metallic system with a tiny energy gap around 0.08eV.

We compare the density evaluation performed by both Octopus and the pole expansion technique. In Octopus, the invariant subspace associated with the smallest $n_e/2 + n_{\mathrm{h}}$ smallest eigenvalues of H is computed using a conjugate gradient (CG) like algorithm, where n_e is the number of electrons in the quantum dot and n_{h} is the number of extra states for finite temperature calculation. The value of n_{h} depends

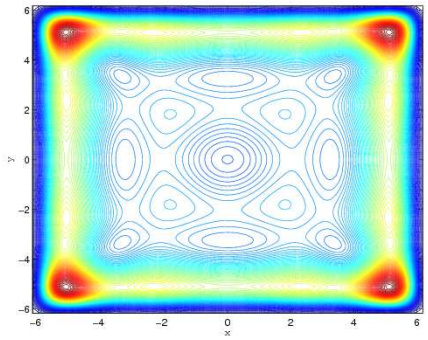


Figure 4.14: A contour plot of the density profile of a quantum dot with 32 electrons.

on the system size and temperature. For example, in the case of 32 electrons 4 extra states are necessary for the electronic structure calculation at 300K. In the pole expansion approach, we use 80 poles, which in general could give a relative error in electron density on the order of 10^{-7} (in L_1 norm) [164].

In addition to using the parallel algorithm to evaluate each term, an extra level of coarse grained parallelism can be achieved by assigning each pole to a different group of processors.

In Table 4.9, we compare the efficiency of the pole expansion technique for the quantum dot density calculation performed with the standard eigenvalue calculation approach implemented in Octopus. The maximum number of CG iterations for computing each eigenvalue in Octopus is set to the default value of 25. We label the pole expansion-based approach that uses the algorithm and implementation as PCSelInv, where the letter C stands for complex. The factor 80 in the last column of Table 4.9 accounts for 80 poles. When a massive number of processors are available, this pole number factor will easily result in a factor of 80 reduction in wall clock time for the PCSelInv calculation, whereas such a perfect reduction in wall clock time cannot be easily obtained in Octopus.

We observe that for quantum dots that contain a few electrons, the standard density evaluation approach implemented in Octopus is faster than the pole expansion

n_e(#Electrons)	Grid	#proc	Octopus time(s)	PCSelInv time(s)
2	31	1	< 0.01	0.01×80
8	63	1	0.03	0.06×80
32	127	1	0.78	0.03×80
128	255	1	26.32	1.72×80
		4	10.79	0.59×80
512	511	1	1091.04	9.76×80
		4	529.30	3.16×80
		16	131.96	1.16×80
2048	1023	1	out of memory	60.08×80
		4	out of memory	19.04×80
		16	7167.98	5.60×80
		64	1819.39	2.84×80

Table 4.9: Timing comparison of electron density evaluation between Octopus and PCSelInv for systems of different sizes. The multiplication by 80 in the last column accounts for the use of 80 pole.

approach. However, when the number of electrons becomes sufficiently large, the advantage of the pole expansion approach using the algorithms presented to compute diag $[H-(z_i+\mu)I]^{-1}$ becomes quite evident. This is because the computation cost associated with the eigenvalue calculation in Octopus is dominated by the computation performed to maintain mutual orthogonality among different eigenvectors when the number of electrons in the quantum dot is large. The complexity of this computation alone is $\mathcal{O}(n^3)$, whereas the overall complexity of the pole-based approach is $\mathcal{O}(n^{3/2})$. The crossover point in our experiment appears to be 512 electrons. For a quantum dot that contains 2048 electrons, PCSelInv is eight times faster than Octopus.

4.5 Conclusion

This chapter has developed the selected inversion algorithm for extracting the diagonal elements and the nearest off-diagonal elements of symmetric matrices. The selected inversion algorithm is numerically exact. The computational complexity of

the selected inversion algorithm is $\mathcal{O}(N)$ for quasi-1D systems, $\mathcal{O}(N^{1.5})$ for quasi-2D systems and $\mathcal{O}(N^2)$ for 3D systems, where N is the number of electrons in the system. Combined with the pole expansion developed in Chapter 3, the selected inversion algorithm can be used for accurate and efficient calculation of solving KSDFT. The asymptotic computational scaling is universally improved compared to the cubic scaling for systems of all dimensions.

We have developed a sequential selected inversion algorithm, called SelInv, for solving the diagonal and nearest off-diagonal elements of a general symmetric matrix. SelInv is applied to a variety of benchmark problems with dimension as large as 1.5 million. SelInv is already two orders of magnitude faster than the direct inversion method for moderately large matrices. We have developed a parallel selected inversion algorithm, called PSelInv, for solving the diagonal and nearest off-diagonal elements of structured 2D matrices. PSelInv is able to solve problems defined on a $65,535 \times 65,535$ grid with 4.3 billion degrees of freedom on $4,096$ processors, and exhibits an excellent weak scaling property.

The selected inversion algorithm have also been applied to study the electronic structure of quantum dots and aluminum. The quantum dots is a two-dimensional system, and the new parallel selected inversion algorithm shows significant advantage over the diagonalization method. Aluminum is a three-dimensional system, and SelInv is only marginally superior to the diagonalization method. The difference in the performance is twofold. First, the asymptotic scaling of the selected inversion algorithm is $\mathcal{O}(N^{1.5})$ for two dimensional system, and $\mathcal{O}(N^2)$ for three dimensional system, due to the difference in the sparsity of the Cholesky factor of the Hamiltonian matrix. The computational time of the selected inversion algorithm is expected to be more expensive in two dimensional case than that in the three dimensional case. Second, the selected inversion algorithm has a larger preconstant than that in the diagonalization method. The preconstant is largely determined by the number of basis

functions per atom, denoted by c. It can be shown that after taking into account the pre-constant, the computational complexity of the diagonalization method is $\mathcal{O}(cN^3)$, while the computational complexity of the selected inversion algorithm is $\mathcal{O}(c^3N)$ for one dimensional system, $\mathcal{O}(c^3N^{1.5})$ for two dimensional system, and $\mathcal{O}(c^3N^2)$ for the three dimensional case. Take the three dimensional system for example, asymptotically the crossover between the selected inversion method and the diagonalization method happens at $N = c^2$. In the scenario studied in Section 4.3 for aluminum system, finite difference method is used for the discretization of the Hamiltonian matrix. It is well known that the finite difference discretization leads to a large number of basis functions per atom. It is desirable to use a discretization scheme that leads to small basis functions per atom. The adaptive local basis functions method developed in Chapter 2 results in a small number of basis functions per atom, and is therefore the natural choice of basis functions in the future work.

Another direction for future work is the parallel selected inversion algorithm for general symmetric matrices. The selected inversion algorithm contains three phases: symbolic analysis; LDL^T factorization; selected inversion. The symbolic analysis can be done in parallel relatively easily for 2D Hamiltonians discretized on a rectangular domain by finite difference. For problems defined on irregular grids (e.g., problems that are discretized by finite elements or some other techniques), a general parallel symbolic analysis based on graph partitioning [133, 210, 257, 257] should be used. The LDL^T factorization can in principle be performed by any of the existing sparse matrix solvers [3, 10, 110, 111, 195, 215, 221, 223]. The selected inversion based on supernodes can be generalized to the parallel version using the same method as in SelInv, once the parallel symbolic analysis is available. This is an area of research we are currently pursuing.

Chapter 5

Fast construction of $\mathcal{H}$ matrix

5.1 Introduction

In this chapter, we consider the following problem: Assume that an unknown symmetric matrix G has the structure of a hierarchical matrix ($\mathcal{H}$-matrix) [37, 115, 116], that is, certain off-diagonal blocks of G are low-rank or approximately low-rank (see the definitions in Sections 5.1.3 and 5.2.2). The task is to construct G efficiently only from a "black box" matrix-vector multiplication subroutine (which shall be referred to as matvec in the following). In a slightly more general setting when G is not symmetric, the task is to construct G from "black box" matrix-vector multiplication subroutines of both G and G^{T}. In this work, we focus on the case of a symmetric matrix G. The proposed algorithm can be extended to the non-symmetric case in a straightforward way.

This work is inspired from the work of selected inversion developed in Chapter 4, and this chapter is relatively independent. Readers who are focusing on the main flow of this dissertation can skip this chapter and directly go to the conclusion of Part I in Chapter 6. Materials in this chapter have been presented in [161].

5.1.1 Motivation and applications

Our motivation is mainly for the situation that G is given as the Green's function of an elliptic equation. In this case, it is proved that G is an $\mathcal{H}$-matrix under mild regularity assumptions [20]. For elliptic equations, methods like preconditioned conjugate gradient, geometric and algebraic multigrid methods, sparse direct methods provide application of the matrix G on vectors. The algorithm proposed in this work then provides an efficient way to construct the matrix G explicitly in the $\mathcal{H}$-matrix form.

Once we obtain the matrix G as an $\mathcal{H}$-matrix, it is possible to apply G on vectors efficiently, since the application of an $\mathcal{H}$-matrix on a vector is linear scaling. Of course, for elliptic equations, it might be more efficient to use available fast solvers directly to solve the equation, especially if only a few right hand sides are to be solved. However, sometimes, it would be advantageous to obtain G since it is then possible to further compress G according to the structure of the data (the vectors that G will be acting on), for example as in numerical homogenization [80]. Another scenario is that the data has special structure like sparsity in the choice of basis, the application of the resulting compressed matrix will be more efficient than the "black box" elliptic solver.

Let us remark that, in the case of elliptic equations, it is also possible to use the $\mathcal{H}$-matrix algebra to invert the direct matrix (which is an $\mathcal{H}$-matrix in *e.g.* finite element discretization). Our method, on the other hand, provides an efficient alternative algorithm when a fast matrix-vector multiplication is readily available, and is able to compute the inverse of an $\mathcal{H}$-matrix of dimension $n \times n$ with $\mathcal{O}(\log n)$ matrix-vector multiplications. We also remark that the preconstant in front of the $\mathcal{O}(\log n)$ scaling can be large, and this may hinder the application of the current version of the algorithm in many scenarios. However, from a computational point of view, what is probably more attractive is that our algorithm facilitates a parallelized construction of the $\mathcal{H}$-matrix, while the direct inversion has a sequential nature [115].

As another motivation, the purpose of the algorithm is to recover the matrix via

a "black box" matrix-vector multiplication subroutine. A general question of this kind will be that under which assumptions of the matrix, one can recover the matrix efficiently by matrix-vector multiplications. If the unknown matrix is low-rank, the recently developed randomized singular value decomposition algorithms [119, 157, 248] provide an efficient way to obtain the low-rank approximation through application of the matrix on random vectors. Low-rank matrices play an important role in many applications. However, the assumption is too strong in many cases that the whole matrix is low-rank. Since the class of $\mathcal{H}$-matrices is a natural generalization of the one of low-rank matrices, the proposed algorithm can be viewed as a further step in this direction.

5.1.2 Randomized singular value decomposition algorithm

A repeatedly leveraged tool in the proposed algorithm is the randomized singular value decomposition algorithm for computing a low rank approximation of a given numerically low-rank matrix. This has been an active research topic in the past several years with vast literature. For the purpose of this work, we have adopted the algorithm developed in [157], although other variants of this algorithm with similar ideas can also be used here. For a given matrix A that is numerically low-rank, this algorithm goes as following to compute a rank-r factorization.

Algorithm 9 Construct a low-rank approximation $A \approx U_1 M U_2^{\mathrm{T}}$ for rank r

1: Choose a Gaussian random matrix $R_1 \in \mathbb{R}^{n\times(r+c)}$ where c is a small constant;
2: Form AR_1 and apply SVD to AR_1. The first r left singular vectors give U_1;
3: Choose a Gaussian random matrix $R_2 \in \mathbb{R}^{n\times(r+c)}$;
4: Form $R_2^{\mathrm{T}} A$ and apply SVD to $A^{\mathrm{T}} R_2$. The first r left singular vectors give U_2;
5: $M = (R_2^{\mathrm{T}} U_1)^{\dagger}[R_2^{\mathrm{T}}(AR_1)](U_2^{\mathrm{T}} R_1)^{\dagger}$, where $B^{\dagger}$ denotes the Moore-Penrose pseudoinverse of matrix B [105, pp. 257–258].

The accuracy of this algorithm and its variants has been studied thoroughly by

several groups. If the matrix 2-norm is used to measure the error, it is well-known that the best rank-r approximation is provided by the singular value decomposition (SVD). When the singular values of A decay rapidly, it has been shown that Algorithm 9 results in almost optimal factorizations with an overwhelming probability [119]. As Algorithm 9 is to be used frequently in our algorithm, we analyze briefly its complexity step by step. The generation of random numbers is quite efficient, therefore in practice one may ignore the cost of steps 1 and 3. Step 2 takes $(r+c)$ matvec of matrix A and $\mathcal{O}(n(r+c)^2)$ steps for applying the SVD algorithms on an $n \times (r+c)$ matrix. The cost of step 4 is the same as the one of step 2. Step 5 involves the computation of $R_2^{\mathrm{T}}(AR_1)$, which takes $\mathcal{O}(n(r+c)^2)$ steps as we have already computed AR_1 in step 2. Once $R_2^{\mathrm{T}}(AR_1)$ is ready, the computation of M takes additional $\mathcal{O}((r+c)^3)$ steps. Therefore, the total complexity of Algorithm 9 is $\mathcal{O}(r+c)$ matvecs plus $\mathcal{O}(n(r+c)^2)$ extra steps.

5.1.3 Top-down construction of $\mathcal{H}$-matrix

We illustrate the core idea of our algorithm using a simple one-dimensional example. The algorithm of constructing a hierarchical matrix G is a top-down pass. We assume throughout the article that G is symmetric.

For clarity, we will first consider a one dimension example. The details of the algorithm in two dimensions will be given in Section 2. We assume that a symmetric matrix G has a hierarchical low-rank structure corresponding to a hierarchical dyadic decomposition of the domain. The matrix G is of dimension $n \times n$ with $n = 2^{L_M}$ for an integer L_M. Denote the set for all indices as $\mathcal{I}_{0;1}$, where the former subscript indicates the level and the latter is the index for blocks in each level. At the first level, the set is partitioned into $\mathcal{I}_{1;1}$ and $\mathcal{I}_{1;2}$, with the assumption that $G(\mathcal{I}_{1;1}, \mathcal{I}_{1;2})$ and $G(\mathcal{I}_{1;2}, \mathcal{I}_{1;1})$ are numerically low-rank, say of rank r for a prescribed error tolerance ε. At level l, each block $\mathcal{I}_{l-1;i}$ on the above level is dyadically decomposed into two

blocks $\mathcal{I}_{l;2i-1}$ and $\mathcal{I}_{l;2i}$ with the assumption that $G(\mathcal{I}_{l;2i-1}, \mathcal{I}_{l;2i})$ and $G(\mathcal{I}_{l;2i}, \mathcal{I}_{l;2i-1})$ are also numerically low-rank (with the same rank r for the tolerance ε). Clearly, at level l, we have in total 2^l off-diagonal low-rank blocks. We stop at level L_M, for which the block $\mathcal{I}_{L_M,i}$ only has one index $\{i\}$. For simplicity of notation, we will abbreviate $G(\mathcal{I}_{l;i}, \mathcal{I}_{l;j})$ by $G_{l;ij}$. We remark that the assumption that off-diagonal blocks are low-rank matrices may not hold for general elliptic operators in higher dimensions. However, this assumption simplifies the introduction of the concept of our algorithm. More realistic case will be discussed in detail in Sections 5.2.3 and 5.2.4.

The overarching strategy of our approach is to peel off the off-diagonal blocks level by level and simultaneously construct their low-rank approximations. On the first level, $G_{1;12}$ is numerically low-rank. In order to use the randomized SVD algorithm for $G_{1;12}$, we need to know the product of $G_{1;12}$ and also $G_{1;12}^{\mathrm{T}} = G_{1;21}$ with a collection of random vectors. This can be done by observing that

$$\begin{pmatrix} G_{1;11} & G_{1;12} \\ G_{1;21} & G_{1;22} \end{pmatrix} \begin{pmatrix} R_{1;1} \\ 0 \end{pmatrix} = \begin{pmatrix} G_{1;11}R_{1;1} \\ G_{1;21}R_{1;1} \end{pmatrix}, \tag{5.1}$$

$$\begin{pmatrix} G_{1;11} & G_{1;12} \\ G_{1;21} & G_{1;22} \end{pmatrix} \begin{pmatrix} 0 \\ R_{1;2} \end{pmatrix} = \begin{pmatrix} G_{1;12}R_{1;2} \\ G_{1;22}R_{1;2} \end{pmatrix}, \tag{5.2}$$

where $R_{1;1}$ and $R_{1;2}$ are random matrices of dimension $n/2 \times (r + c)$. We obtain $(G_{1;21}R_{1;1})^{\mathrm{T}} = R_{1;1}^{\mathrm{T}} G_{1;12}$ by restricting the right hand side of Eq. (5.1) to $\mathcal{I}_{1;2}$ and obtain $G_{1;12}R_{1;2}$ by restricting the right hand side of Eq. (5.2) to $\mathcal{I}_{1;1}$, respectively. The low-rank approximation using Algorithm 9 results in

$$G_{1;12} \approx \widehat{G}_{1;12} = U_{1;12} M_{1;12} U_{1;21}^{\mathrm{T}}. \tag{5.3}$$

$U_{1;12}$ and $U_{1;21}$ are $n/2 \times r$ matrices and $M_{1;12}$ is an $r \times r$ matrix. Due to the fact that G is symmetric, a low-rank approximation of $G_{1;21}$ is obtained as the transpose

of $G_{1;12}$.

Now on the second level, the matrix G has the form

$$\begin{pmatrix} G_{2;11} & G_{2;12} & & \\ G_{2;21} & G_{2;22} & G_{1;12} & \\ & & G_{2;33} & G_{2;34} \\ G_{1;21} & & G_{2;43} & G_{2;44} \end{pmatrix}.$$

The submatrices $G_{2;12}$, $G_{2;21}$, $G_{2;34}$, and $G_{2;43}$ are numerically low-rank, to obtain their low-rank approximations by the randomized SVD algorithm. Similar to the first level, we could apply G on random matrices of the form like $(R_{2;1}, 0, 0, 0)^{\mathrm{T}}$. This will require $4(r+c)$ number of matrix-vector multiplications. However, this is not optimal: Since we already know the interaction between $\mathcal{I}_{1;1}$ and $\mathcal{I}_{1;2}$, we could combine the calculations together to reduce the number of matrix-vector multiplications needed. Observe that

$$\begin{pmatrix} G_{2;11} & G_{2;12} & & \\ G_{2;21} & G_{2;22} & G_{1;12} & \\ & & G_{2;33} & G_{2;34} \\ G_{1;21} & & G_{2;43} & G_{2;44} \end{pmatrix} \begin{pmatrix} R_{2;1} \\ 0 \\ R_{2;3} \\ 0 \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} G_{2;11}R_{2;1} \\ G_{2;21}R_{2;1} \end{pmatrix} + G_{1;12} \begin{pmatrix} R_{2;3} \\ 0 \end{pmatrix} \\ \begin{pmatrix} G_{2;33}R_{2;3} \\ G_{2;43}R_{2;3} \end{pmatrix} + G_{1;21} \begin{pmatrix} R_{2;1} \\ 0 \end{pmatrix} \end{pmatrix}. \quad (5.4)$$

Denote

$$\widehat{G}^{(1)} = \begin{pmatrix} 0 & \widehat{G}_{1;12} \\ \widehat{G}_{1;21} & 0 \end{pmatrix} \quad (5.5)$$

with $\widehat{G}_{1;12}$ and $\widehat{G}_{1;21}$ the low-rank approximations we constructed on the first level,

then

$$\widehat{G}^{(1)} \begin{pmatrix} R_{2;1} \\ 0 \\ R_{2;3} \\ 0 \end{pmatrix} = \begin{pmatrix} \widehat{G}_{1;12} \begin{pmatrix} R_{2;3} \\ 0 \end{pmatrix} \\ \widehat{G}_{1;21} \begin{pmatrix} R_{2;1} \\ 0 \end{pmatrix} \end{pmatrix}. \tag{5.6}$$

Therefore,

$$(G - \widehat{G}^{(1)}) \begin{pmatrix} R_{2;1} \\ 0 \\ R_{2;3} \\ 0 \end{pmatrix} \approx \begin{pmatrix} G_{2;11} R_{2;1} \\ G_{2;21} R_{2;1} \\ G_{2;33} R_{2;3} \\ G_{2;43} R_{2;3} \end{pmatrix}, \tag{5.7}$$

so that we simultaneously obtain $(G_{2;21}R_{2;1})^{\mathrm{T}} = R_{2;1}^{\mathrm{T}} G_{2;12}$ and $(G_{2;43}R_{2;3})^{\mathrm{T}} = R_{2;3}^{\mathrm{T}} G_{2;34}$. Similarly, applying G on $(0, R_{2;2}, 0, R_{2;4})^{\mathrm{T}}$ provides $G_{2;12}R_{2;2}$ and $G_{2;34}R_{2;4}$. We can then obtain the following low-rank approximations by invoking Algorithm 9.

$$\begin{aligned} G_{2;12} &\approx \widehat{G}_{2;12} = U_{2;12} M_{2;12} U_{2;21}^{\mathrm{T}}, \\ G_{2;34} &\approx \widehat{G}_{2;34} = U_{2;34} M_{2;34} U_{2;43}^{\mathrm{T}}. \end{aligned} \tag{5.8}$$

The low-rank approximations of $G_{2;21}$ and $G_{2;43}$ are again given by the transposes of the above formulas.

Similarly, on the third level, the matrix G has the form

$$\begin{pmatrix} G_{3;11} & G_{3;12} & & & & & & \\ G_{3;21} & G_{3;22} & & & & & & \\ & & G_{3;33} & G_{3;34} & & & & \\ & & G_{3;43} & G_{3;44} & & & & \\ & & & & G_{3;55} & G_{3;56} & & \\ & & & & G_{3;65} & G_{3;66} & & \\ & & & & & & G_{3;77} & G_{3;78} \\ & & & & & & G_{3;87} & G_{3;88} \end{pmatrix} \begin{matrix} & & & & & & & \\ & G_{2;12} & & & & & & \\ & & & & & & & \\ G_{2;21} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{matrix} \begin{matrix} & & & & & & & \\ & & & & & & & \\ & & G_{1;12} & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{matrix} \begin{matrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & G_{2;34} & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{matrix}, \tag{5.9}$$

$$\begin{matrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & G_{1;21} & & & & & & \\ & & & & & & & \\ & & & & & & & \end{matrix} \begin{matrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & G_{2;43} & & & \\ & & & & & & & \end{matrix}$$

and define

$$\widehat{G}^{(2)} = \begin{pmatrix} 0 & \widehat{G}_{2;12} & & & & & & \\ \widehat{G}_{2;21} & 0 & & & & & & \\ & & 0 & & & & & \\ & & & 0 & & & & \\ & & & & 0 & \widehat{G}_{2;34} & & \\ & & & & \widehat{G}_{2;43} & 0 & & \\ & & & & & & 0 & \\ & & & & & & & 0 \end{pmatrix}. \tag{5.10}$$

We could simultaneously obtain the product of $G_{3;12}$, $G_{3;34}$, $G_{3;56}$ and $G_{3;78}$ with random vectors by applying the matrix G with random vectors of the form

$$(R_{3;1}^{\mathrm{T}}, 0, R_{3;3}^{\mathrm{T}}, 0, R_{3;5}^{\mathrm{T}}, 0, R_{3;7}^{\mathrm{T}}, 0)^{\mathrm{T}},$$

then subtract the product of $\widehat{G}^{(1)} + \widehat{G}^{(2)}$ with the same vectors. Again invoking Algorithm 9 provides us the low-rank approximations of these off-diagonal blocks.

The algorithm continues in the same fashion for higher levels. The combined random tests lead to a constant number of matvec at each level. As there are $\log(n)$ levels in total, the total number of matrix-vector multiplications scales logarithmically.

When the block size on a level becomes smaller than the given criteria (for example, the numerical rank r used in the construction), one could switch to a deterministic

way to get the off-diagonal blocks. In particular, we stop at a level L ($L < L_M$) such that each $\mathcal{I}_{L;i}$ contains about r entries. Now only the elements in the diagonal blocks $G_{L,ii}$ need to be determined. This can be completed by applying G to the matrix

$$(I, I, \ldots, I)^{\mathrm{T}},$$

where I is the identity matrix whose dimension is equal to the number of indices in $\mathcal{I}_{L;i}$.

Let us summarize the structure of our algorithm. From the top level to the bottom level, we peel off the numerically low-rank off-diagonal blocks using the randomized SVD algorithm. The matrix-vector multiplications required by the randomized SVD algorithms are computed effectively by *combining* several random tests into one using the zero pattern of the *remaining* matrix. In this way, we get an efficient algorithm for constructing the hierarchical representation for the matrix G.

5.1.4 Related works

Our algorithm is built on top of the framework of the $\mathcal{H}$-matrices proposed by Hackbusch and his collaborators [20, 37, 115]. The definitions of the $\mathcal{H}$-matrices will be summarized in Section 5.2. In a nutshell, the $\mathcal{H}$-matrix framework is an operational matrix algebra for efficiently representing, applying, and manipulating discretizations of operators from elliptic partial differential equations. Though we have known how to represent and apply these matrices for quite some time [108], it is the contribution of the $\mathcal{H}$-matrix framework that enables one to manipulate them in a general and coherent way. A closely related matrix algebra is also developed in a more numerical-linear-algebraic viewpoint under the name *hierarchical semiseparable matrices* by Chandrasekaran, Gu, and others [56, 57]. Here, we will follow the notations of the $\mathcal{H}$-matrices as our main motivations are from numerical solutions of elliptic

PDEs.

A basic assumption of our algorithm is the existence of a fast matrix-vector multiplication subroutine. The most common case is when G is the inverse of the stiffness matrix H of a general elliptic operator. Since H is often sparse, much effort has been devoted to computing $u = Gf$ by solving the linear system $Hu = f$. Many ingenious algorithms have been developed for this purpose in the past forty years. Commonly-seen examples include multifrontal algorithms [78, 92], geometric multigrids [41, 43, 115], algebraic multigrids (AMG) [42], domain decompositions methods [232, 240], wavelet-based fast algorithms [32] and preconditioned conjugate gradient algorithms (PCG) [28], to name a few. Very recently, both Chandrasekaran *et al* [55] and Martinsson [179] have combined the idea of the multifrontal algorithms with the $\mathcal{H}$-matrices to obtain highly efficiently direct solvers for $Hu = f$. Another common case for which a fast matrix-vector multiplication subroutine is available comes from the boundary integral equations where G is often a discretization of a Green's function restricted to a domain boundary. Fast algorithms developed for this case include the famous fast multipole method [108], the panel clustering method [117], and others. All these fast algorithms mentioned above can be used as the "black box" algorithm for our method.

As shown in the previous section, our algorithm relies heavily on the randomized singular value decomposition algorithm for constructing the factorizations of the off-diagonal blocks. This topic has been a highly active research area in the past several years and many different algorithms have been proposed in the literature. Here, for our purpose, we have adopted the algorithm described in [157, 248]. In a related but slightly different problem, the goal is to find low-rank approximations $A = CUR$ where C contains a subset of columns of A and R contains a subset of rows. Papers devoted to this task include [74, 75, 106, 177]. In our setting, since we assume no direct access of entries of the matrix A but only its impact through matrix-vector

multiplications, the algorithm proposed by [157] is the most relevant choice. An excellent recent review of this fast growing field can be found in [119].

In a recent paper [178], Martinsson considered also the problem of constructing the $\mathcal{H}$-matrix representation of a matrix, but he assumed that one can access arbitrary entries of the matrix besides the fast matrix-vector multiplication subroutine. Under this extra assumption, he showed that one can construct the $\mathcal{H}^2$ representation of the matrix with $\mathcal{O}(1)$ matrix-vector multiplications and accesses of $\mathcal{O}(n)$ matrix entries. However, in many situations including the case of G being the inverse of the stiffness matrix of an elliptic differential operator, accessing entries of G is by no means a trivial task. Comparing with Martinsson's work, our algorithm only assumes the existence of a fast matrix-vector multiplication subroutine, and hence is more general.

As we mentioned earlier, one motivation for computing G explicitly is to further compress the matrix G. The most common example in the literature of numerical analysis is the process of numerical homogenization or upscaling [80]. Here the matrix G is often again the inverse of the stiffness matrix H of an elliptic partial differential operator. When H contains information from all scales, the standard homogenization techniques fail. Recently, Owhadi and Zhang [197] proposed an elegant method that, under the assumption that the Cordes condition is satisfied, upscales a general H in divergence form using metric transformation. Computationally, their approach involves d solves of form $Hu = f$ with d being the dimension of the problem. On the other hand, if G is computed using our algorithm, one can obtain the upscaled operator by inverting a low-passed and down-sampled version of G. Complexity-wise, our algorithm is more costly since it requires $\mathcal{O}(\log n)$ solves of $Hu = f$. However, since our approach makes no analytic assumptions about H, it is expected to be more general.

5.2 Algorithm

We now present the details of our algorithm in two dimensions. In addition to a top-down construction using the peeling idea presented in the introduction, the complexity will be further reduced using the $\mathcal{H}^2$ property of the matrix [37, 116]. The extension to three dimensions is straightforward.

In two dimensions, a more conservative partition of the domain is required to guarantee the low-rankness of the matrix blocks. We will start with discussion of this new geometric setup. Then we will recall the notion of hierarchical matrices and related algorithms in Section 5.2.2. The algorithm to construct an $\mathcal{H}^2$ representation for a matrix using matrix-vector multiplications will be presented in Sections 5.2.3 and 5.2.4. Finally, variants of the algorithm for constructing the $\mathcal{H}^1$ and uniform $\mathcal{H}^1$ representations will be described in Section 5.2.5.

5.2.1 Geometric setup and notations

Let us consider an operator G defined on a 2D domain $[0,1)^2$ with periodic boundary condition. We discretize the problem using an $n = N \times N$ uniform grid with N being a power of 2: $N = 2^{L_M}$. Denote the set of all grid points as

$$\mathcal{I}_0 = \{(k_1/N, k_2/N) \mid k_1, k_2 \in \mathbb{N},\, 0 \le k_1, k_2 < N\} \tag{5.11}$$

and partition the domain hierarchically into $L+1$ levels ($L < L_M$). On each level l ($0 \le l \le L$), we have $2^l \times 2^l$ boxes denoted by $\mathcal{I}_{l;ij} = [(i-1)/2^l, i/2^l) \times [(j-1)/2^l, j/2^l)$ for $1 \le i, j \le 2^l$. The symbol $\mathcal{I}_{l;ij}$ will also be used to denote the grid points that lies in the box $\mathcal{I}_{l;ij}$. The meaning should be clear from the context. We will also use $\mathcal{I}_l$(or $\mathcal{J}_l$) to denote a general box on certain level l. The subscript l will be omitted, when the level is clear from the context. For a given box $\mathcal{I}_l$ for $l \ge 1$, we call a box $\mathcal{J}_{l-1}$ on level $l-1$ its parent if $\mathcal{I}_l \subset \mathcal{J}_{l-1}$. Naturally, $\mathcal{I}_l$ is called a child of $\mathcal{J}_{l-1}$. It is

clear that each box except those on level L will have four children boxes.

For any box $\mathcal{I}$ on level l, it covers $N/2^l \times N/2^l$ grid points. The last level L can be chosen so that the leaf box has a constant number of points in it (*i.e.* the difference $L_M - L$ is kept to be a constant when N increases).

For simplicity of presentation, we will start the method from level 3. It is also possible to start from level 2. Level 2 needs to be treated specially, as for level 3. We define the following notations for a box $\mathcal{I}$ on level l ($l \geq 3$):

$\mathsf{NL}(\mathcal{I})$ Neighbor list of box $\mathcal{I}$. This list contains the boxes on level l that are adjacent to $\mathcal{I}$ and also $\mathcal{I}$ itself. There are 9 boxes in the list for each $\mathcal{I}$.

$\mathsf{IL}(\mathcal{I})$ Interaction list of box $\mathcal{I}$. When $l = 3$, this list contains all the boxes on level 3 minus the set of boxes in $\mathsf{NL}(\mathcal{I})$. There are 55 boxes in total. When $l > 3$, this list contains all the boxes on level l that are children of boxes in $\mathsf{NL}(\mathcal{P})$ with $\mathcal{P}$ being $\mathcal{I}$'s parent minus the set of boxes in $\mathsf{NL}(\mathcal{I})$. There are 27 such boxes.

Notice that these two lists determine two symmetric relationship: $\mathcal{J} \in \mathsf{NL}(\mathcal{I})$ if and only if $\mathcal{I} \in \mathsf{NL}(\mathcal{J})$ and $\mathcal{J} \in \mathsf{IL}(\mathcal{I})$ if and only if $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$. Figs. 5.1 and 5.2 illustrate the computational domain and the lists for $l = 3$ and $l = 4$, respectively.

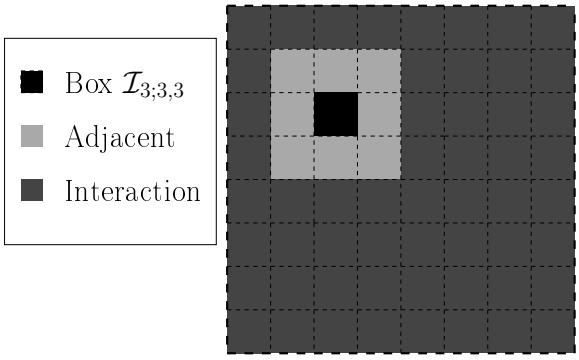


Figure 5.1: Illustration of the computational domain at level 3. $\mathcal{I}_{3;3,3}$ is the black box. The neighbor list $\mathsf{NL}(\mathcal{I}_{3;3,3})$ consists of 8 adjacent light gray boxes and the black box itself, and the interaction list $\mathsf{IL}(\mathcal{I}_{3;3,3})$ consists of the 55 dark gray boxes.

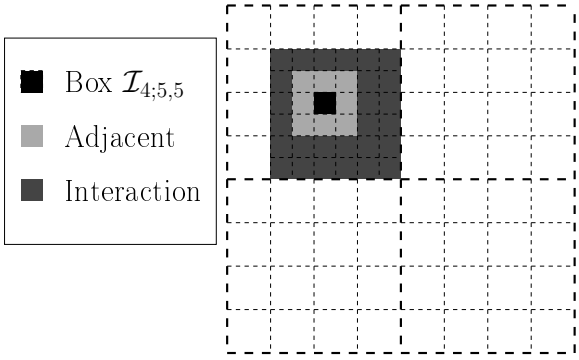


Figure 5.2: Illustration of the computational domain at level 4. $\mathcal{I}_{4;5,5}$ is the black box. The neighbor list $\mathsf{NL}(\mathcal{I}_{4;5,5})$ consists of 8 adjacent light gray boxes and the black box itself, and the interaction list $\mathsf{IL}(\mathcal{I}_{4;5,5})$ consists of the 27 dark gray boxes.

For a vector f defined on the $N \times N$ grid $\mathcal{I}_0$, we define $f(\mathcal{I})$ to be the restriction of f to grid points $\mathcal{I}$. For a matrix $G \in \mathbb{R}^{N^2 \times N^2}$ that represents a linear map from $\mathcal{I}_0$ to itself, we define $G(\mathcal{I}, \mathcal{J})$ to be the restriction of G on $\mathcal{I} \times \mathcal{J}$.

A matrix $G \in \mathbb{R}^{N^2 \times N^2}$ has the following decomposition

$$G = G^{(3)} + G^{(4)} + \cdots + G^{(L)} + D^{(L)}. \tag{5.12}$$

Here, for each l, $G^{(l)}$ incorporates the interaction on level l between a box with its interaction list. More precisely, $G^{(l)}$ has a $2^{2l} \times 2^{2l}$ block structure:

$$G^{(l)}(\mathcal{I}, \mathcal{J}) = \begin{cases} G(\mathcal{I}, \mathcal{J}), & \mathcal{I} \in \mathsf{IL}(\mathcal{J}) \text{ (eq. } \mathcal{J} \in \mathsf{IL}(\mathcal{I})); \\ 0, & \text{otherwise} \end{cases}$$

with $\mathcal{I}$ and $\mathcal{J}$ both on level l. The matrix $D^{(L)}$ includes the interactions between adjacent boxes at level L:

$$D^{(L)}(\mathcal{I}, \mathcal{J}) = \begin{cases} G(\mathcal{I}, \mathcal{J}), & \mathcal{I} \in \mathsf{NL}(\mathcal{J}) \text{ (eq. } \mathcal{J} \in \mathsf{NL}(\mathcal{I})); \\ 0, & \text{otherwise} \end{cases}$$

with $\mathcal{I}$ and $\mathcal{J}$ both on level L. To show that (5.12) is true, it suffices to prove that for any two boxes $\mathcal{I}$ and $\mathcal{J}$ on level L, the right hand side gives $G(\mathcal{I}, \mathcal{J})$. In the case that $\mathcal{I} \in \mathsf{NL}(\mathcal{J})$, this is obvious. Otherwise, it is clear that we can find a level l, and boxes $\mathcal{I}'$ and $\mathcal{J}'$ on level l, such that $\mathcal{I}' \in \mathsf{IL}(\mathcal{J}')$, $\mathcal{I} \subset \mathcal{I}'$ and $\mathcal{J} \subset \mathcal{J}'$, and hence $G(\mathcal{I}, \mathcal{J})$ is given through $G(\mathcal{I}', \mathcal{J}')$. Throughout the text, we will use $\|A\|_2$ to denote the matrix 2-norm of matrix A.

5.2.2 Hierarchical matrix

Our algorithm works with the so-called hierarchical matrices. We recall in this subsection some basic properties of this type of matrices and also some related algorithms. For simplicity of notations and representation, we will only work with symmetric matrices. For a more detailed introduction of the hierarchical matrices and their applications in fast algorithms, we refer the readers to [115, 116].

$\mathcal{H}^1$ matrices

Definition 5.2.1. *G is a (symmetric) $\mathcal{H}^1$-matrix if for any $\varepsilon > 0$, there exists $r(\varepsilon) \lesssim \log(\varepsilon^{-1})$ such that for any pair $(\mathcal{I}, \mathcal{J})$ with $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$, there exist orthogonal matrices $U_{\mathcal{I}\mathcal{J}}$ and $U_{\mathcal{J}\mathcal{I}}$ with $r(\varepsilon)$ columns and matrix $M_{\mathcal{I}\mathcal{J}} \in \mathbb{R}^{r(\varepsilon)\times r(\varepsilon)}$ such that*

$$\|G(\mathcal{I}, \mathcal{J}) - U_{\mathcal{I}\mathcal{J}} M_{\mathcal{I}\mathcal{J}} U_{\mathcal{J}\mathcal{I}}^{\mathrm{T}}\|_2 \leq \varepsilon \|G(\mathcal{I}, \mathcal{J})\|_2. \tag{5.13}$$

The main advantage of the $\mathcal{H}^1$ matrix is that the application of such matrix on a vector can be efficiently evaluated: Within error $\mathcal{O}(\varepsilon)$, one can use $\widehat{G}(\mathcal{I}, \mathcal{J}) = U_{\mathcal{I}\mathcal{J}} M_{\mathcal{I}\mathcal{J}} U_{\mathcal{J}\mathcal{I}}^{\mathrm{T}}$, which is low-rank, instead of the original block $G(\mathcal{I}, \mathcal{J})$. The algorithm is described in Algorithm 10. It is standard that the complexity of the matrix-vector multiplication for an $\mathcal{H}^1$ matrix is $\mathcal{O}(N^2 \log N)$ [115].

Algorithm 10 Application of a $\mathcal{H}^1$-matrix G on a vector f.

```
1: u = 0;
2: for l = 3 to L do
3:    for I on level l do
4:       for J ∈ IL(I) do
5:          u(I) = u(I) + U_IJ(M_IJ(U_JI^T f(J)));
6:       end for
7:    end for
8: end for
9: for I on level L do
10:   for J ∈ NL(I) do
11:      u(I) = u(I) + G(I, J)f(J);
12:   end for
13: end for
```

Uniform $\mathcal{H}^1$ matrix

Definition 5.2.2. *G is a (symmetric) uniform $\mathcal{H}^1$-matrix if for any $\varepsilon > 0$, there exists $r_U(\varepsilon) \lesssim \log(\varepsilon^{-1})$ such that for each box $\mathcal{I}$, there exists an orthogonal matrix $U_\mathcal{I}$ with $r_U(\varepsilon)$ columns such that for any pair $(\mathcal{I}, \mathcal{J})$ with $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$*

$$\|G(\mathcal{I}, \mathcal{J}) - U_\mathcal{I} N_{\mathcal{I}\mathcal{J}} U_\mathcal{J}^{\mathrm{T}}\|_2 \leq \varepsilon \|G(\mathcal{I}, \mathcal{J})\|_2 \tag{5.14}$$

with $N_{\mathcal{I}\mathcal{J}} \in \mathbb{R}^{r_U(\varepsilon) \times r_U(\varepsilon)}$.

The application of a uniform $\mathcal{H}^1$ matrix to a vector is described in Algorithm 11. The complexity of the algorithm is still $\mathcal{O}(N^2 \log N)$. However, the prefactor is much better as each $U_\mathcal{I}$ is applied only once. The speedup over Algorithm 10 is roughly $27r(\varepsilon)/r_U(\varepsilon)$ [115].

$\mathcal{H}^2$ matrices

Definition 5.2.3. *G is an $\mathcal{H}^2$ matrix if*

- *it is a uniform $\mathcal{H}^1$ matrix;*

Algorithm 11 Application of a uniform $\mathcal{H}^1$-matrix G on a vector f

```
1: u = 0;
2: for l = 3 to L do
3:    for J on level l do
4:       f̃_J = U_J^T f(J);
5:    end for
6: end for
7: for l = 3 to L do
8:    for I on level l do
9:       ũ_I = 0;
10:      for J ∈ IL(I) do
11:         ũ_I = ũ_I + N_IJ f̃_J;
12:      end for
13:   end for
14: end for
15: for l = 3 to L do
16:    for I on level l do
17:       u(I) = u(I) + U_I ũ_I;
18:    end for
19: end for
20: for I on level L do
21:    for J ∈ NL(I) do
22:       u(I) = u(I) + G(I, J) f(J);
23:    end for
24: end for
```

- *Suppose that $\mathcal{C}$ is any child of a box $\mathcal{I}$, then*

$$\|U_{\mathcal{I}}(\mathcal{C},:) - U_{\mathcal{C}} T_{\mathcal{C}\mathcal{I}}\|_2 \lesssim \varepsilon, \tag{5.15}$$

for some matrix $T_{\mathcal{C}\mathcal{I}} \in \mathbb{R}^{r_U(\varepsilon)\times r_U(\varepsilon)}$.

The application of an $\mathcal{H}^2$ matrix to a vector is described in Algorithm 12 and it has a complexity of $\mathcal{O}(N^2)$, Notice that, compared with $\mathcal{H}^1$ matrix, the logarithmic factor is reduced [116].

Remark 2. *Applying an $\mathcal{H}^2$ matrix to a vector can indeed be viewed as the matrix form of the fast multipole method (FMM)* [108]*. One recognizes in Algorithm 12 that the second top-level* **for** *loop corresponds to the M2M (multipole expansion to multipole expansion) translations of the FMM; the third top-level* **for** *loop is the M2L (multipole expansion to local expansion) translations; and the fourth top-level* **for** *loop is the L2L (local expansion to local expansion) translations.*

In the algorithm to be introduced, we will also need to apply a partial matrix $G^{(3)} + G^{(4)} + \cdots + G^{(L')}$ for some $L' \leq L$ to a vector f. This amounts to a variant of Algorithm 12, described in Algorithm 13.

Algorithm 12 Application of a $\mathcal{H}^2$-matrix G on a vector f

```
1: u = 0;
2: for J on level L do
3:    f̃_J = U_J^T f(J);
4: end for
5: for l = L − 1 down to 3 do
6:    for J on level l do
7:       f̃_J = 0;
8:       for each child C of J do
9:          f̃_J = f̃_J + T_{CJ}^T f̃_C;
10:      end for
11:   end for
12: end for
13: for l = 3 to L do
14:   for I on level l do
15:      ũ_I = 0;
16:      for J ∈ IL(I) do
17:         ũ_I = ũ_I + N_{IJ} f̃_J;
18:      end for
19:   end for
20: end for
18: for l = 3 to L − 1 do
19:   for I on level l do
20:      for each child C of I do
21:         ũ_C = ũ_C + T_{CI} ũ_I;
22:      end for
23:   end for
24: end for
25: for I on level L do
26:   u(I) = U_I ũ_I;
27: end for
28: for I on level L do
29:   for J ∈ NL(I) do
30:      u(I) = u(I) + G(I, J) f(J);
31:   end for
32: end for
```

5.2.3 Peeling algorithm: outline and preparation

We assume that G is a symmetric $\mathcal{H}^2$ matrix and that there exists a fast matrix-vector subroutine for applying G to any vector f as a "black box". The goal is to construct an $\mathcal{H}^2$ representation of the matrix G using only a small number of test vectors.

The basic strategy is a top-down construction: For each level $l = 3, \ldots, L$, assume that an $\mathcal{H}^2$ representation for $G^{(3)} + \cdots + G^{(l-1)}$ is given, we construct $G^{(l)}$ by the following three steps:

1. *Peeling.* Construct an $\mathcal{H}^1$ representation for $G^{(l)}$ using the peeling idea and the $\mathcal{H}^2$ representation for $G^{(3)} + \cdots + G^{(l-1)}$.

2. *Uniformization.* Construct a uniform $\mathcal{H}^1$ representation for $G^{(l)}$ from its $\mathcal{H}^1$ representation.

3. *Projection.* Construct an $\mathcal{H}^2$ representation for $G^{(3)} + \cdots + G^{(l)}$.

Algorithm 13 Application of a partial $\mathcal{H}^2$-matrix $G^{(3)} + \cdots + G^{(L')}$ on a vector f

$u = 0$;
for $\mathcal{J}$ on level L' **do**
  $\widetilde{f}_{\mathcal{J}} = U_{\mathcal{J}}^{\mathrm{T}} f(\mathcal{J})$;
end for
for $l = L' - 1$ down to 3 **do**
  for $\mathcal{J}$ on level l **do**
    $\widetilde{f}_{\mathcal{J}} = 0$;
    for each child $\mathcal{C}$ of $\mathcal{J}$ **do**
      $\widetilde{f}_{\mathcal{J}} = \widetilde{f}_{\mathcal{J}} + T_{\mathcal{C}\mathcal{J}}^{\mathrm{T}} \widetilde{f}_{\mathcal{C}}$;
    end for
  end for
end for
for $l = 3$ to L' **do**
  for $\mathcal{I}$ on level l **do**
    $\widetilde{u}_{\mathcal{I}} = 0$;
    for $\mathcal{J} \in \mathsf{IL}(\mathcal{I})$ **do**
      $\widetilde{u}_{\mathcal{I}} = \widetilde{u}_{\mathcal{I}} + N_{\mathcal{I}\mathcal{J}} \widetilde{f}_{\mathcal{J}}$;
    end for
  end for
end for
for $l = 3$ to $L' - 1$ **do**
  for $\mathcal{I}$ on level l **do**
    for each child $\mathcal{C}$ of $\mathcal{I}$ **do**
      $\widetilde{u}_{\mathcal{C}} = \widetilde{u}_{\mathcal{C}} + T_{\mathcal{C}\mathcal{I}} \widetilde{u}_{\mathcal{I}}$;
    end for
  end for
end for
for $\mathcal{I}$ on level L' **do**
  $u(\mathcal{I}) = U_{\mathcal{I}} \widetilde{u}_{\mathcal{I}}$;
end for

The names of these steps will be made clear in the following discussion. Variants of the algorithm that only construct an $\mathcal{H}^1$ representation (a uniform $\mathcal{H}^1$ representation, respectively) of the matrix G can be obtained by only doing the peeling step (the peeling and uniformization steps, respectively). These variants will be discussed in Section 5.2.5.

After we have the $\mathcal{H}^2$ representation for $G^{(3)} + \cdots + G^{(L)}$, we use the peeling idea again to extract the diagonal part $D^{(L)}$. We call this whole process the *peeling* algorithm.

Before detailing the peeling algorithm, we mention two procedures that serve as essential components of our algorithm. The first procedure concerns with the uniformization step, in which one needs to get a uniform $\mathcal{H}^1$ representation for $G^{(l)}$ from its $\mathcal{H}^1$ representation, *i.e.*, from $\widehat{G}(\mathcal{I}, \mathcal{J}) = U_{\mathcal{I}\mathcal{J}} M_{\mathcal{I}\mathcal{J}} U_{\mathcal{J}\mathcal{I}}^{\mathrm{T}}$ to $\widehat{G}(\mathcal{I}, \mathcal{J}) = U_{\mathcal{I}} N_{\mathcal{I}\mathcal{J}} U_{\mathcal{J}}^{\mathrm{T}}$, for all pairs of boxes $(\mathcal{I}, \mathcal{J})$ with $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$. To this end, what we need to do is to

find the column space of

$$[U_{\mathcal{I}\mathcal{J}_1} M_{\mathcal{I}\mathcal{J}_1} \mid U_{\mathcal{I}\mathcal{J}_2} M_{\mathcal{I}\mathcal{J}_2} \mid \cdots \mid U_{\mathcal{I}\mathcal{J}_t} M_{\mathcal{I}\mathcal{J}_t}], \tag{5.16}$$

where $\mathcal{J}_j$ are the boxes in $\mathsf{IL}(\mathcal{I})$ and $t = |\mathsf{IL}(\mathcal{I})|$. Notice that we weight the singular vectors U by M, so that the singular vectors corresponding to larger singular values will be more significant. This column space can be found by the usual SVD algorithm or a more effective randomized version presented in Algorithm 14. The important left singular vectors are denoted by $U_{\mathcal{I}}$, and the diagonal matrix formed by the singular values associated with $U_{\mathcal{I}}$ is denoted by $S_{\mathcal{I}}$.

Algorithm 14 Construct a uniform $\mathcal{H}^1$ representation of G from the $\mathcal{H}^1$ representation at a level l

1: **for** each box $\mathcal{I}$ on level l **do**
2: Generate a Gaussian random matrix $R \in \mathbb{R}^{(r(\varepsilon)\times t)\times(r_U(\varepsilon)+c)}$;
3: Form product $[U_{\mathcal{I}\mathcal{J}_1} M_{\mathcal{I}\mathcal{J}_1} \mid \cdots \mid U_{\mathcal{I}\mathcal{J}_t} M_{\mathcal{I}\mathcal{J}_t}]R$ and apply SVD to it. The first $r_U(\varepsilon)$ left singular vectors give $U_{\mathcal{I}}$, and the corresponding singular values give a diagonal matrix $S_{\mathcal{I}}$;
4: **for** $\mathcal{J}_j \in \mathsf{IL}(\mathcal{I})$ **do**
5: $I_{\mathcal{I}\mathcal{J}_j} = U_{\mathcal{I}}^{\mathrm{T}} U_{\mathcal{I}\mathcal{J}_j}$;
6: **end for**
7: **end for**
8: **for** each pair $(\mathcal{I}, \mathcal{J})$ on level l with $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$ **do**
9: $N_{\mathcal{I}\mathcal{J}} = I_{\mathcal{I}\mathcal{J}} M_{\mathcal{I}\mathcal{J}} I_{\mathcal{J}\mathcal{I}}^{\mathrm{T}}$;
10: **end for**

Complexity analysis: For a box $\mathcal{I}$ on level l, the number of grid points in $\mathcal{I}$ is $(N/2^l)^2$. Therefore, $U_{\mathcal{I}\mathcal{J}_j}$ are all of size $(N/2^l)^2 \times r(\varepsilon)$ and $M_{\mathcal{I}\mathcal{J}}$ are of size $r(\varepsilon) \times r(\varepsilon)$. Forming the product $[U_{\mathcal{I}\mathcal{J}_1} M_{\mathcal{I}\mathcal{J}_1} \mid \cdots \mid U_{\mathcal{I}\mathcal{J}_t} M_{\mathcal{I}\mathcal{J}_t}]R$ takes $\mathcal{O}((N/2^l)^2 r(\varepsilon)(r_U(\varepsilon) + c))$ steps and SVD takes $\mathcal{O}((N/2^l)^2 (r_U(\varepsilon) + c)^2)$ steps. As there are 2^{2l} boxes on level l, the overall cost of Algorithm 14 is $\mathcal{O}(N^2 (r_U(\varepsilon) + c)^2) = \mathcal{O}(N^2)$. One may also apply to $[U_{\mathcal{I}\mathcal{J}_1} M_{\mathcal{I}\mathcal{J}_1} \mid \cdots \mid U_{\mathcal{I}\mathcal{J}_t} M_{\mathcal{I}\mathcal{J}_t}]$ the deterministic SVD algorithm, which has the same order of complexity but with a prefactor about $27 r(\varepsilon)/(r_U(\varepsilon) + c)$ times larger.

The second procedure is concerned with the projection step of the above list, in

which one constructs an $\mathcal{H}^2$ representation for $G^{(3)} + \cdots G^{(l)}$. Here, we are given the $\mathcal{H}^2$ representation for $G^{(3)} + \cdots + G^{(l-1)}$ along with the uniform $\mathcal{H}^1$ representation for $G^{(l)}$ and the goal is to compute the transfer matrix $T_{\mathcal{C}\mathcal{I}}$ for a box $\mathcal{I}$ on level $l-1$ and its child $\mathcal{C}$ on level l such that

$$\|U_{\mathcal{I}}(\mathcal{C},:) - U_{\mathcal{C}} T_{\mathcal{C}\mathcal{I}}\|_2 \lesssim \varepsilon.$$

In fact, the existing $U_{\mathcal{C}}$ matrix of the uniform $\mathcal{H}^1$ representation may not be rich enough to contain the columns of $U_{\mathcal{I}}(\mathcal{C},:)$ in its span. Therefore, one needs to update the content of $U_{\mathcal{C}}$ as well. To do that, we perform a singular value decomposition for the combined matrix

$$[U_{\mathcal{I}}(\mathcal{C},:)S_{\mathcal{I}} \mid U_{\mathcal{C}} S_{\mathcal{C}}]$$

and define a matrix $V_{\mathcal{C}}$ to contain $r_U(\varepsilon)$ left singular vectors. Again $U_{\mathcal{I}}, U_{\mathcal{C}}$ should be weighted by the corresponding singular values. The transfer matrix $T_{\mathcal{C}\mathcal{I}}$ is then given by

$$T_{\mathcal{C}\mathcal{I}} = V_{\mathcal{C}}^{\mathrm{T}} U_{\mathcal{I}}(\mathcal{C},:)$$

and the new $U_{\mathcal{C}}$ is set to be equal to $V_{\mathcal{C}}$. Since $U_{\mathcal{C}}$ has been changed, the matrices $N_{\mathcal{C}\mathcal{D}}$ for $\mathcal{D} \in \mathsf{IL}(\mathcal{C})$ and also the corresponding singular values $S_{\mathcal{C}}$ need to be updated as well. The details are listed in Algorithm 15.

Complexity analysis: The main computational task of Algorithm 15 is again the SVD part. For a box $\mathcal{C}$ on level l, the number of grid points in $\mathcal{I}$ is $(N/2^l)^2$. Therefore, the combined matrix $[U_{\mathcal{I}}(\mathcal{C},:)S_{\mathcal{I}} \mid U_{\mathcal{C}} S_{\mathcal{C}}]$ is of size $(N/2^l)^2 \times 2r_U(\varepsilon)$. The SVD computation clearly takes $\mathcal{O}((N/2^l)^2 r_U(\varepsilon)^2) = \mathcal{O}((N/2^l)^2)$ steps. Taking into the consideration that there are 2^{2l} boxes on level l gives rise to an $\mathcal{O}(N^2)$ estimate for the cost of Algorithm 15.

Algorithm 15 Construct an $\mathcal{H}^2$ representation of G from the uniform $\mathcal{H}^1$ representation at level l

for each box $\mathcal{I}$ on level $l-1$ **do**
  for each child $\mathcal{C}$ of $\mathcal{I}$ **do**
    Form matrix $[U_{\mathcal{I}}(\mathcal{C},:)S_{\mathcal{I}} \mid U_{\mathcal{C}}S_{\mathcal{C}}]$ and apply SVD to it. The first $r_U(\varepsilon)$ left singular vectors give $V_{\mathcal{C}}$, and the corresponding singular values give a diagonal matrix $W_{\mathcal{C}}$;
    $K_{\mathcal{C}} = V_{\mathcal{C}}^{\mathrm{T}} U_{\mathcal{C}}$;
    $T_{\mathcal{C}\mathcal{I}} = V_{\mathcal{C}}^{\mathrm{T}} U_{\mathcal{I}}(\mathcal{C},:)$;
    $U_{\mathcal{C}} = V_{\mathcal{C}}$;
    $S_{\mathcal{C}} = W_{\mathcal{C}}$;
  end for
end for
for each pair $(\mathcal{C},\mathcal{D})$ on level l with $\mathcal{C} \in \mathsf{IL}(\mathcal{D})$ **do**
  $N_{\mathcal{C}\mathcal{D}} = K_{\mathcal{C}} N_{\mathcal{C}\mathcal{D}} K_{\mathcal{D}}^{\mathrm{T}}$;
end for

5.2.4 Peeling algorithm: details

With the above preparation, we are now ready to describe the peeling algorithm in detail at different levels, starting from level 3. At each level, we follow exactly the three steps listed at the beginning of Section 5.2.3.

Level 3

First in the peeling step, we construct the $\mathcal{H}^1$ representation for $G^{(3)}$. For each pair $(\mathcal{I},\mathcal{J})$ on level 3 such that $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$, we will invoke randomized SVD Algorithm 9 to construct the low rank approximation of $G_{\mathcal{I},\mathcal{J}}$. However, in order to apply the algorithm we need to compute $G(\mathcal{I},\mathcal{J})R_{\mathcal{J}}$ and $R_{\mathcal{I}}^{\mathrm{T}} G(\mathcal{I},\mathcal{J})$, where $R_{\mathcal{I}}$ and $R_{\mathcal{J}}$ are random matrices with $r(\varepsilon)+c$ columns. For each box $\mathcal{J}$ on level 3, we construct a matrix R of size $N^2 \times (r(\varepsilon)+c)$ such that

$$R(\mathcal{J},:) = R_{\mathcal{J}} \quad \text{and} \quad R(\mathcal{I}_0 \backslash \mathcal{J},:) = 0.$$

Computing GR using $r(\varepsilon)+c$ matvecs and restricting the result to grid points $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$ gives $G(\mathcal{I},\mathcal{J})R_{\mathcal{J}}$ for each $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$.

After repeating these steps for all boxes on level 3, we hold for any pair $(\mathcal{I}, \mathcal{J})$ with $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$ the following data:

$$G(\mathcal{I}, \mathcal{J})R_{\mathcal{J}} \quad \text{and} \quad R_{\mathcal{I}}^{\mathrm{T}} G(\mathcal{I}, \mathcal{J}) = (G(\mathcal{J}, \mathcal{I})R_{\mathcal{I}})^{\mathrm{T}}.$$

Now, applying Algorithm 9 to them gives the low-rank approximation

$$\widehat{G}(\mathcal{I}, \mathcal{J}) = U_{\mathcal{I}\mathcal{J}} M_{\mathcal{I}\mathcal{J}} U_{\mathcal{J}\mathcal{I}}^{\mathrm{T}}. \tag{5.17}$$

In the uniformization step, in order to get the uniform $\mathcal{H}^1$ representation for $G^{(3)}$, we simply apply Algorithm 14 to the boxes on level 3 to get the approximations

$$\widehat{G}(\mathcal{I}, \mathcal{J}) = U_{\mathcal{I}} N_{\mathcal{I}\mathcal{J}} U_{\mathcal{J}}^{\mathrm{T}}. \tag{5.18}$$

Finally in the projection step, since we only have 1 level now (level 3), we have already the $\mathcal{H}^2$ representation for $G^{(3)}$.

Complexity analysis: The dominant computation is the construction of the $\mathcal{H}^1$ representation for $G^{(3)}$. This requires $r(\varepsilon)+c$ matvecs for each box $\mathcal{I}$ on level 3. Since there are in total 64 boxes at this level, the total cost is $64(r(\varepsilon) + c)$ matvecs. From the complexity analysis in Section 5.2.3, the computation for the second and third steps cost an extra $\mathcal{O}(N^2)$ steps.

Level 4

First in the peeling step, in order to construct the $\mathcal{H}^1$ representation for $G^{(4)}$, we need to compute the matrices $G(\mathcal{I}, \mathcal{J})R_{\mathcal{J}}$ and $R_{\mathcal{I}}^{\mathrm{T}} G(\mathcal{I}, \mathcal{J})$ for each pair $(\mathcal{I}, \mathcal{J})$ on level 4 with $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$. Here $R_{\mathcal{I}}$ and $R_{\mathcal{J}}$ are again random matrices with $r(\varepsilon) + c$ columns.

One approach is to follow exactly what we did for level 3: Fix a box $\mathcal{J}$ at this

level, construct R of size $N^2 \times (r(\varepsilon)+c)$ such that

$$R(\mathcal{J},:) = R_{\mathcal{J}} \quad \text{and} \quad R(\mathcal{I}_0 \backslash \mathcal{J},:) = 0.$$

Next apply $G - G^{(3)}$ to R, by subtracting GR and $G^{(3)}R$. The former is computed using $r(\varepsilon)+c$ matvecs and the latter is done by Algorithm 13. Finally, restrict the result to grid points $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$ gives $G(\mathcal{I},\mathcal{J})R_{\mathcal{J}}$ for each $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$.

However, we have observed in the simple one-dimensional example in Section 5.1.3 that random tests can be combined together as in Eq. (5.6) and (5.7). We shall detail this observation in the more general situation here as following. Observe that $G - G^{(3)} = G^{(4)} + D^{(4)}$, and $G^{(4)}(\mathcal{J},\mathcal{I})$ and $D^{(4)}(\mathcal{J},\mathcal{I})$ for boxes $\mathcal{I}$ and $\mathcal{J}$ on level 4 is only nonzero if $\mathcal{I} \in \mathsf{NL}(\mathcal{J}) \cup \mathsf{IL}(\mathcal{J})$. Therefore, $(G - G^{(3)})R$ for R coming from $\mathcal{J}$ can only be nonzero in $\mathsf{NL}(\mathcal{P})$ with $\mathcal{P}$ being $\mathcal{J}$'s parent. The rest is automatically zero (up to error ε as $G^{(3)}$ is approximated by its $\mathcal{H}^2$ representation). Therefore, we can combine the calculation of different boxes as long as their non-zero regions do not overlap.

More precisely, we introduce the following sets $\mathcal{S}_{pq}$ for $1 \le p, q \le 8$ with

$$\mathcal{S}_{pq} = \{\mathcal{J}_{4;ij} \mid i \equiv p \,(\mathrm{mod}\ 8),\ j \equiv q \,(\mathrm{mod}\ 8)\}. \tag{5.19}$$

There are 64 sets in total, each consisting of four boxes. Fig. 5.3 illustrates one such set at level 4. For each set $\mathcal{S}_{pq}$, first construct R with

$$R(\mathcal{J},:) = \begin{cases} R_{\mathcal{J}}, & \mathcal{J} \in \mathcal{S}_{pq}; \\ 0, & \text{otherwise.} \end{cases}$$

Then, we apply $G - G^{(3)}$ to R, by subtracting GR and $G^{(3)}R$. The former is computed using $r(\varepsilon)+c$ matvecs and the latter is done by Algorithm 13. For each $\mathcal{J} \in \mathcal{S}_{pq}$,

restricting the result to $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$ gives $G(\mathcal{I}, \mathcal{J})R_{\mathcal{J}}$. Repeating this computation for all sets $\mathcal{S}_{pq}$ then provides us with the following data:

$$G(\mathcal{I}, \mathcal{J})R_{\mathcal{J}} \quad \text{and} \quad R_{\mathcal{I}}^{\mathrm{T}} G(\mathcal{I}, \mathcal{J}) = (G(\mathcal{J}, \mathcal{I})R_{\mathcal{I}})^{\mathrm{T}},$$

for each pair $(\mathcal{I}, \mathcal{J})$ with $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$. Applying Algorithm 9 to them gives the required low-rank approximations

$$\widehat{G}(\mathcal{I}, \mathcal{J}) = U_{\mathcal{I}\mathcal{J}} M_{\mathcal{I}\mathcal{J}} U_{\mathcal{J}\mathcal{I}}^{\mathrm{T}} \tag{5.20}$$

with $U_{\mathcal{I}\mathcal{J}}$ orthogonal.

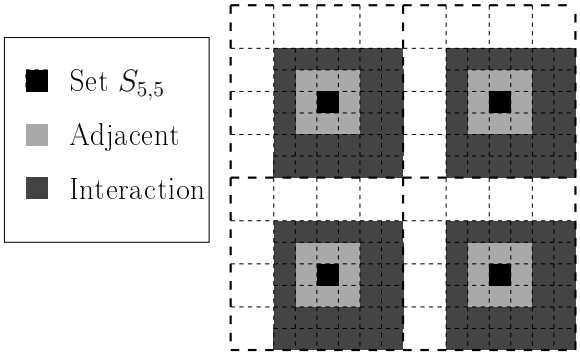


Figure 5.3: Illustration of the set S_{55} at level 4. This set consists of four black boxes $\{\mathcal{I}_{4;5,5}, \mathcal{I}_{4;13,5}, \mathcal{I}_{4;5,13}, \mathcal{I}_{4;13,13}\}$. The light gray boxes around each black box are in the neighbor list and the dark gray boxes in the interaction list.

Next in the uniformization step, the task is to construct the uniform $\mathcal{H}^1$ representation of $G^{(4)}$. Similar to the computation at level 3, we simply apply Algorithm 14 to the boxes on level 4 to get

$$\widehat{G}(\mathcal{I}, \mathcal{J}) = U_{\mathcal{I}} N_{\mathcal{I}\mathcal{J}} U_{\mathcal{J}}^{\mathrm{T}}. \tag{5.21}$$

Finally in the projection step, to get $\mathcal{H}^2$ representation for $G^{(3)} + G^{(4)}$, we invoke Algorithm 15 at level 4. Once it is done, we hold the transfer matrices $T_{\mathcal{C}\mathcal{I}}$ between any $\mathcal{I}$ on level 3 and each of its children $\mathcal{C}$, along with the updated uniform $\mathcal{H}^1$-matrix

representation of $G^{(4)}$.

Complexity analysis: The dominant computation is again the construction of $\mathcal{H}^1$ representation for $G^{(4)}$. For each group $\mathcal{S}_{pq}$, we apply G to $r(\varepsilon)+c$ vectors and apply $G^{(3)}$ to $r(\varepsilon)+c$ vectors. The latter takes $\mathcal{O}(N^2)$ steps for each application. Since there are 64 sets in total, this computation takes $64(r(\varepsilon)+c)$ matvecs and $\mathcal{O}(N^2)$ extra steps.

Level l

First in the peeling step, to construct the $\mathcal{H}^1$ representation for $G^{(l)}$, we follow the discussion of level 4. Define 64 sets $\mathcal{S}_{pq}$ for $1 \le p, q \le 8$ with

$$\mathcal{S}_{pq} = \{\mathcal{J}_{l;ij} \mid i \equiv p \,(\mathrm{mod}\ 8),\ j \equiv q \,(\mathrm{mod}\ 8)\}. \tag{5.22}$$

Each set contains exactly $2^l/8 \times 2^l/8$ boxes. For each set $\mathcal{S}_{pq}$, construct R with

$$R(\mathcal{J},:) = \begin{cases} R_{\mathcal{J}}, & \mathcal{J} \in \mathcal{S}_{pq}; \\ 0, & \text{otherwise.} \end{cases}$$

Next, apply $G - [G^{(3)} + \cdots + G^{(l-1)}]$ to R, by subtracting GR and $[G^{(3)} + \cdots + G^{(l-1)}]R$. The former is again computed using $r(\varepsilon)+c$ matvecs and the latter is done by Algorithm 13 using the $\mathcal{H}^2$ representation of $G^{(3)}+\cdots+G^{(l-1)}$. For each $\mathcal{J} \in \mathcal{S}_{pq}$, restricting the result to $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$ gives $G(\mathcal{I},\mathcal{J})R_{\mathcal{J}}$. Repeating this computation for all sets $\mathcal{S}_{pq}$ gives the following data for any pair $(\mathcal{I},\mathcal{J})$ with $\mathcal{I} \in \mathsf{IL}(\mathcal{J})$

$$G(\mathcal{I},\mathcal{J})R_{\mathcal{J}} \quad \text{and} \quad R_{\mathcal{I}}^{\mathrm{T}} G(\mathcal{I},\mathcal{J}) = (G(\mathcal{J},\mathcal{I})R_{\mathcal{I}})^{\mathrm{T}}.$$

Now applying Algorithm 9 to them gives the low-rank approximation

$$\widehat{G}(\mathcal{I},\mathcal{J}) = U_{\mathcal{I}\mathcal{J}} M_{\mathcal{I}\mathcal{J}} U_{\mathcal{J}\mathcal{I}}^{\mathrm{T}} \tag{5.23}$$

with $U_{\mathcal{I}\mathcal{J}}$ orthogonal.

Similar to the computation at level 4, the uniformization step that constructs the uniform $\mathcal{H}^1$ representation of $G^{(l)}$ simply by Algorithm 14 to the boxes on level l. The result gives the approximation

$$\widehat{G}(\mathcal{I},\mathcal{J}) = U_{\mathcal{I}} N_{\mathcal{I}\mathcal{J}} U_{\mathcal{J}}^{\mathrm{T}}. \tag{5.24}$$

Finally in the projection step, one needs to compute an $\mathcal{H}^2$ representation for $G^{(3)} + \cdots + G^{(l)}$. To this end, we apply Algorithm 15 to level l.

The complexity analysis at level l follows exactly the one of level 4. Since we still have exactly 64 sets $\mathcal{S}_{pq}$, the computation again takes $64(r(\varepsilon)+c)$ matvecs along with $\mathcal{O}(N^2)$ extra steps.

These three steps (peeling, uniformization, and projection) are repeated for each level until we reach level L. At this point, we hold the $\mathcal{H}^2$ representation for $G^{(3)} + \cdots G^{(L)}$.

Computation of $D^{(L)}$

Finally we construct of the diagonal part

$$D^{(L)} = G - (G^{(3)} + \cdots + G^{(L)}). \tag{5.25}$$

More specifically, for each box $\mathcal{J}$ on level L, we need to compute $G(\mathcal{I},\mathcal{J})$ for $\mathcal{I} \in \mathsf{NL}(\mathcal{J})$.

Define a matrix E of size $N^2 \times (N/2^L)^2$ (recall that the box $\mathcal{J}$ on level L covers

$(N/2^L)^2$ grid points) by

$$E(\mathcal{J},:) = I \quad \text{and} \quad E(\mathcal{I}_0 \backslash \mathcal{J},:) = 0,$$

where I is the $(N/2^L)^2 \times (N/2^L)^2$ identity matrix. Applying $G - (G^{(3)} + \cdots + G^{(L)})$ to E and restricting the results to $\mathcal{I} \in \mathsf{NL}(\mathcal{J})$ gives $G(\mathcal{I}, \mathcal{J})$ for $\mathcal{I} \in \mathsf{NL}(\mathcal{J})$. However, we can do better as $(G - (G^{(3)} + \cdots + G^{(L)}))E$ is only non-zero in $\mathsf{NL}(\mathcal{J})$. Hence, one can combine computation of different boxes $\mathcal{J}$ as long as $\mathsf{NL}(\mathcal{J})$ do not overlap.

To do this, define the following $4 \times 4 = 16$ sets $\mathcal{S}_{pq}$, $1 \le p, q \le 4$

$$\mathcal{S}_{pq} = \{\mathcal{J}_{L,ij} \mid i \equiv p \,(\mathrm{mod}\ 4),\ j \equiv q \,(\mathrm{mod}\ 4)\}.$$

For each set $\mathcal{S}_{pq}$, construct matrix E by

$$E(\mathcal{J},:) = \begin{cases} I, & \mathcal{J} \in \mathcal{S}_{pq}; \\ 0, & \text{otherwise.} \end{cases}$$

Next, apply $G - (G^{(3)} + \cdots + G^{(L)})$ to E. For each $\mathcal{J} \in \mathcal{S}_{pq}$, restricting the result to $\mathcal{I} \in \mathsf{NL}(\mathcal{J})$ gives $G(\mathcal{I}, \mathcal{J})I = G(\mathcal{I}, \mathcal{J})$. Repeating this computation for all 16 sets $\mathcal{S}_{pq}$ gives the diagonal part $D^{(L)}$.

Complexity analysis: The dominant computation is for each group $\mathcal{S}_{pq}$ apply G and $G^{(3)} + \cdots + G^{(L)}$ to E, the former takes $(N/2^L)^2$ matvecs and the latter takes $\mathcal{O}((N/2^L)^2 N^2)$ extra steps. Recall by the choice of L, $N/2^L$ is a constant. Therefore, the total cost for 16 sets is $16(N/2^L)^2 = \mathcal{O}(1)$ matvecs and $\mathcal{O}(N^2)$ extra steps.

Let us now summarize the complexity of the whole peeling algorithm. From the above discussion, it is clear that at each level the algorithm spends $64(r(\varepsilon)+c) = \mathcal{O}(1)$ matvecs and $\mathcal{O}(N^2)$ extra steps. As there are $\mathcal{O}(\log N)$ levels, the overall cost of the peeling algorithm is equal to $\mathcal{O}(\log N)$ matvecs plus $\mathcal{O}(N^2 \log N)$ steps.

It is a natural concern that whether the error from low-rank decompositions on top levels accumulates in the peeling steps. As observed from numerical examples in Section 5.3, it does not seem to be a problem at least for the examples considered. We do not have a proof for this though.

5.2.5 Peeling algorithm: variants

In this section, we discuss two variants of the peeling algorithm. Let us recall that the above algorithm performs the following three steps at each level l.

1. *Peeling.* Construct an $\mathcal{H}^1$ representation for $G^{(l)}$ using the peeling idea and the $\mathcal{H}^2$ representation for $G^{(3)} + \cdots + G^{(l-1)}$.

2. *Uniformization.* Construct a uniform $\mathcal{H}^1$ representation for $G^{(l)}$ from its $\mathcal{H}^1$ representation.

3. *Projection.* Construct an $\mathcal{H}^2$ representation for $G^{(3)} + \cdots + G^{(l)}$.

As this algorithm constructs the $\mathcal{H}^2$ representation of the matrix G, we also refer to it more specifically as the $\mathcal{H}^2$ version of the peeling algorithm. In what follows, we list two simpler versions that are useful in practice

- the $\mathcal{H}^1$ version, and

- the uniform $\mathcal{H}^1$ version.

In the $\mathcal{H}^1$ version, we only perform the peeling step at each level. Since this version constructs only the $\mathcal{H}^1$ representation, it will use the $\mathcal{H}^1$ representation of $G^{(3)} + \cdots + G^{(l)}$ in the computation of $(G^{(3)} + \cdots + G^{(l)})R$ within the peeling step at level $l + 1$.

In the uniform $\mathcal{H}^1$ version, we perform the peeling step and the uniformization step at each level. This will give us instead the uniform $\mathcal{H}^1$ version of the matrix.

Accordingly, one needs to use the uniform $\mathcal{H}^1$ representation of $G^{(3)} + \cdots + G^{(l)}$ in the computation of $(G^{(3)} + \cdots + G^{(l)})R$ within the peeling step at level $l + 1$.

These two simplified versions are of practical value since there are matrices that are in the $\mathcal{H}^1$ or the uniform $\mathcal{H}^1$ class but not the $\mathcal{H}^2$ class. A simple calculation shows that these two simplified versions still take $\mathcal{O}(\log N)$ matvecs but requires $\mathcal{O}(N^2 \log^2 N)$ extra steps. Clearly, the number of extra steps is $\log N$ times more expensive than the one of the $\mathcal{H}^2$ version. However, if the fast matrix-vector multiplication subroutine itself takes $\mathcal{O}(N^2 \log N)$ steps per application, using the $\mathcal{H}^1$ or the uniform $\mathcal{H}^1$ version does not change the overall asymptotic complexity.

Between these two simplified versions, the uniform $\mathcal{H}^1$ version requires the uniformization step, while the $\mathcal{H}^1$ version does not. This seems to suggest that the uniform $\mathcal{H}^1$ version is more expensive. However, because (1) our algorithm also utilizes the partially constructed representations for the calculation at future levels and (2) the uniform $\mathcal{H}^1$ representation is much faster to apply, the construction of the uniform $\mathcal{H}^1$ version turns out to be much faster. Moreover, since the uniform $\mathcal{H}^1$ representation stores one $U_{\mathcal{I}}$ matrix for each box $\mathcal{I}$ while the $\mathcal{H}^1$ version stores about 27 of them, the uniform $\mathcal{H}^1$ is much more memory-efficient, which is very important for problems in higher dimensions.

5.3 Numerical results

We study the performance of the hierarchical matrix construction algorithm for the inverse of a discretized elliptic operator. The computational domain is a two dimensional square $[0, 1)^2$ with periodic boundary condition, discretized as an $N \times N$ equispaced grid. We first consider the operator $H = -\Delta + V$ with Δ being the discretized Laplacian operator and the potential being $V(i, j) = 1 + W(i, j), \quad i, j = 1, \ldots, N$. For all (i, j), $W(i, j)$ are independent random numbers uniformly distributed in $[0, 1]$.

The potential function V is chosen to have this strong randomness in order to show that the existence of $\mathcal{H}$-matrix representation of the Green's function depends weakly on the smoothness of the potential. The inverse matrix of H is denoted by G. The algorithms are implemented using MATLAB. All numerical tests are carried out on a single-CPU machine.

We analyze the performance statistics by examining both the cost and the accuracy of our algorithm. The cost factors include the time cost and the memory cost. While the memory cost is mainly determined by how the matrix G is compressed and does not depend much on the particular implementation, the time cost depends heavily on the performance of matvec subroutine. Therefore, we report both the wall clock time consumption of the algorithm and the number of calls to the matvec subroutine. The matvec subroutine used here is a nested dissection reordered block Gauss elimination method [92]. For an $N \times N$ discretization of the computational domain, this matvec subroutine has a computational cost of $\mathcal{O}(N^2 \log N)$ steps.

Table 5.1 summarizes the matvec number, and the time cost per degree of freedom (DOF) for the $\mathcal{H}^1$, the uniform $\mathcal{H}^1$ and the $\mathcal{H}^2$ representations of the peeling algorithm. The time cost per DOF is defined by the total time cost divided by the number of grid points N^2. For the $\mathcal{H}^1$ and the uniform $\mathcal{H}^1$ versions, the error criterion ε in Eq. (5.13), Eq. (5.14) and Eq. (5.15) are all set to be 10^{-6}.

The number of calls to the matvec subroutine is the same in all three cases (as the peeling step is the same for all cases) and is reported in the third column of Table 5.1. It is confirmed that the number of calls to matvec increases logarithmically with respect to N if the domain size at level L, i.e. $2^{L_M - L}$, is fixed as a constant. For a fixed N, the time cost is not monotonic with respect to L. When L is too small the computational cost of $D^{(L)}$ becomes dominant. When L is too large, the application of the partial representation $G^{(3)} + \ldots + G^{(L)}$ to a vector becomes expensive. From the perspective of time cost, there is an optimal L_{opt} for a fixed N. We find that this

optimal level number is the same for $\mathcal{H}^1$, uniform $\mathcal{H}^1$ and $\mathcal{H}^2$ algorithms. Table 5.1 shows that $L_{\text{opt}} = 4$ for $N = 32, 64, 128$, $L_{\text{opt}} = 5$ for $N = 256$, and $L_{\text{opt}} = 6$ for $N = 512$. This suggests that for large N, the optimal performance is achieved when the size of boxes on the final level L is 8×8. In other words, $L = L_M - 3$.

The memory cost per DOF for the $\mathcal{H}^1$, the uniform $\mathcal{H}^1$ and the $\mathcal{H}^2$ algorithms is reported in Table 5.2. The memory cost is estimated by summing the sizes of low-rank approximations as well as the size of $D^{(L)}$. For a fixed N, the memory cost generally decreases as L increases. This is because as L increases, an increasing part of the original dense matrix is represented using low-rank approximations.

Both Table 5.1 and Table 5.2 indicate that uniform $\mathcal{H}^1$ algorithm is significantly more advantageous than $\mathcal{H}^1$ algorithm, while the $\mathcal{H}^2$ algorithm leads to a further improvement over the uniform $\mathcal{H}^1$ algorithm especially for large N. This fact can be better seen from Fig. 5.4 where the time and memory cost per DOF for $N = 32, 64, 128, 256, 512$ with optimal level number L_{opt} are shown. We remark that since the number of calls to the matvec subroutine are the same in all cases, the time cost difference comes solely from the efficiency difference of the low rank matrix-vector multiplication subroutines.

We measure the accuracy for the $\mathcal{H}^1$, the uniform $\mathcal{H}^1$ and the $\mathcal{H}^2$ representations of G with its actual value using the operator norm (2-norm) of the error matrix. Here, the 2-norm of a matrix is numerically estimated by power method [105] using several random initial guesses. We report both absolute and relative errors. According to Table 5.3, the errors are well controlled with respect to both increasing N and L, in spite of the more aggressive matrix compression strategy in the uniform $\mathcal{H}^1$ and the $\mathcal{H}^2$ representations. Moreover, for each box $\mathcal{I}$, the rank $r_U(\varepsilon)$ of the uniform $\mathcal{H}^1$ representation is only slightly larger than the rank $r(\varepsilon)$ of the $\mathcal{H}^1$ representation. This can be seen from Table 5.4. Here the average rank for a level l is estimated by averaging the values of $r_U(\varepsilon)$ (or $r(\varepsilon)$) for all low-rank approximations at level l. Note

that the rank of the $\mathcal{H}^2$ representation is comparable to or even lower than the rank in the uniform $\mathcal{H}^1$ representation. This is partially due to different weighting choices in the uniformization step and $\mathcal{H}^2$ construction step.

N	L	matvec number	$\mathcal{H}^1$ time per DOF (s)	Uniform $\mathcal{H}^1$ time per DOF (s)	$\mathcal{H}^2$ time per DOF (s)
32	4	3161	0.0106	0.0080	0.0084
64	4	3376	0.0051	0.0033	0.0033
64	5	4471	0.0150	0.0102	0.0106
128	4	4116	0.0050	0.0025	0.0024
128	5	4639	0.0080	0.0045	0.0045
128	6	5730	0.0189	0.0122	0.0125
256	4	7169	0.015	0.0054	0.0050
256	5	5407	0.010	0.0035	0.0033
256	6	5952	0.013	0.0058	0.0057
256	7	7021	0.025	0.0152	0.0154
512	5	8439	0.025	0.0070	0.0063
512	6	6708	0.018	0.0050	0.0044
512	7	7201	0.022	0.0079	0.0072

Table 5.1: matvec numbers and time cost per degree of freedom (DOF) for the $\mathcal{H}^1$, the uniform $\mathcal{H}^1$ and the $\mathcal{H}^2$ representations with different grid point per dimension N and low rank compression level L. The matvec numbers are by definition the same in the three algorithms.

N	L	$\mathcal{H}^1$ memory per DOF (MB)	Uniform $\mathcal{H}^1$ memory per DOF (MB)	$\mathcal{H}^2$ memory per DOF (MB)
32	4	0.0038	0.0024	0.0024
64	4	0.0043	0.0027	0.0026
64	5	0.0051	0.0027	0.0026
128	4	0.0075	0.0051	0.0049
128	5	0.0056	0.0029	0.0027
128	6	0.0063	0.0029	0.0027
256	4	0.0206	0.0180	0.0177
256	5	0.0087	0.0052	0.0049
256	6	0.0067	0.0030	0.0027
256	7	0.0074	0.0030	0.0027
512	5	0.0218	0.0181	0.0177
512	6	0.0099	0.0053	0.0049
512	7	0.0079	0.0031	0.0027

Table 5.2: Memory cost per degree of freedom (DOF) for the $\mathcal{H}^1$, the uniform $\mathcal{H}^1$ and the $\mathcal{H}^2$ versions with different grid point per dimension N and low rank compression level L.

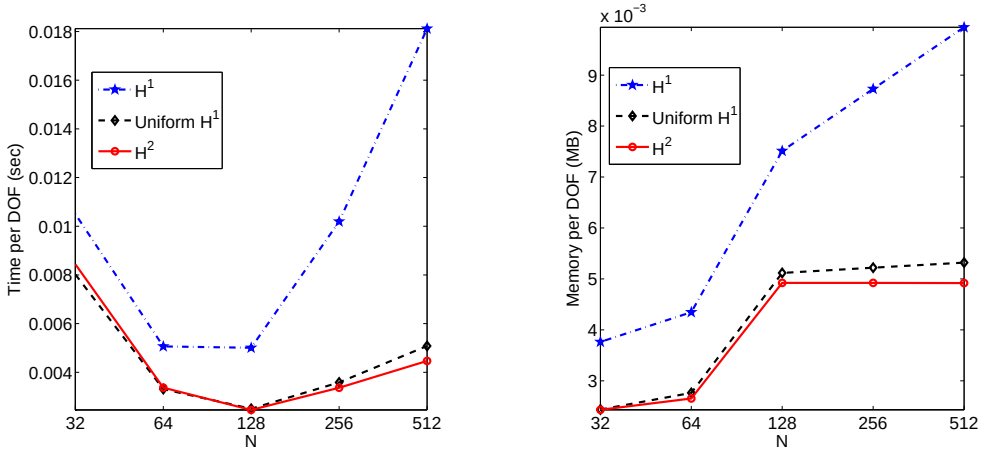


Figure 5.4: Comparison of the time and memory costs for the $\mathcal{H}^1$, the uniform $\mathcal{H}^1$ and the $\mathcal{H}^2$ versions with optimal level L_{opt} for $N = 32, 64, 128, 256, 512$. The x-axis (N) is set to be in logarithmic scale.

N	L	$\mathcal{H}^1$		Uniform $\mathcal{H}^1$		$\mathcal{H}^2$	
		Absolute error	Relative error	Absolute error	Relative error	Absolute error	Relative error
32	4	2.16e-07	3.22e-07	2.22e-07	3.31e-07	2.20e-07	3.28e-07
64	4	2.10e-07	3.15e-07	2.31e-07	3.47e-07	2.31e-07	3.46e-07
64	5	1.96e-07	2.95e-07	2.07e-07	3.12e-07	2.07e-07	3.11e-07
128	4	2.16e-07	3.25e-07	2.26e-07	3.39e-07	2.24e-07	3.37e-07
128	5	2.60e-07	3.90e-07	2.68e-07	4.03e-07	2.67e-07	4.02e-07
128	6	2.01e-07	3.01e-07	2.09e-07	3.13e-07	2.08e-07	3.11e-07
256	4	1.78e-07	2.66e-07	1.95e-07	2.92e-07	2.31e-07	3.46e-07
256	5	2.11e-07	3.16e-07	2.26e-07	3.39e-07	2.27e-07	3.40e-07
256	6	2.75e-07	4.12e-07	2.78e-07	4.18e-07	2.30e-07	3.45e-07
256	7	1.93e-07	2.89e-07	2.05e-07	3.08e-07	2.24e-07	3.36e-07
512	5	2.23e-07	3.35e-07	2.33e-07	3.50e-07	1.42e-07	2.13e-07
512	6	2.06e-07	3.09e-07	2.17e-07	3.26e-07	2.03e-07	3.05e-07
512	7	2.67e-07	4.01e-07	2.74e-07	4.11e-07	2.43e-07	3.65e-07

Table 5.3: Absolute and relative 2-norm errors for the $\mathcal{H}^1$, the uniform $\mathcal{H}^1$ and the $\mathcal{H}^2$ algorithms with different grid point per dimension N and low rank compression level L. The 2-norm is estimated using power method.

l	$\mathcal{H}^1$ average rank	Uniform $\mathcal{H}^1$ average rank	$\mathcal{H}^2$ average rank
4	6	13	13
5	6	13	11
6	6	12	9

Table 5.4: Comparison of the average rank at different levels between the $\mathcal{H}^1$, the uniform $\mathcal{H}^1$, and the $\mathcal{H}^2$ algorithms, for $N = 256$.

The peeling algorithm for the construction of hierarchical matrix can be applied as well to general elliptic operators in divergence form $H = -\nabla \cdot (a(\boldsymbol{r})\nabla) + V(\boldsymbol{r})$. The computational domain, the grids are the same as the example above, and five-point discretization is used for the differential operator. The media is assumed to be high contrast: $a(i,j) = 1 + U(i,j)$, with $U(i,j)$ being independent random numbers uniformly distributed in $[0,1]$. The potential functions under consideration are (1) $V(i,j) = 10^{-3}W(i,j)$; (2) $V(i,j) = 10^{-6}W(i,j)$. $W(i,j)$ are independent random numbers uniformly distributed in $[0,1]$ and are independent of $U(i,j)$. We test the $\mathcal{H}^2$ version for $N = 64$, $L = 4$, with the compression criterion $\varepsilon = 10^{-6}$. The number of matvec is comparable to that reported in Table 5.1. The resulting L^2 absolute and relative error of the Green's function are reported in Table 5.5. The results indicate that the algorithms work well in these cases, despite the fact that the off-diagonal elements of the Green's function have a slower decay than the first example. We also remark that the small relative error for case (2) is due to the large 2-norm of H^{-1} when V is small.

Potential	matvec	Absolute error	Relative error
$V(i,j) = 10^{-3}W(i,j)$	4420	5.91e-04	2.97e-07
$V(i,j) = 10^{-6}W(i,j)$	4420	3.60e-03	1.81e-09

Table 5.5: The number of matvec, and the absolute and relative 2-norm errors for the $\mathcal{H}^2$ representation of the matrix $(-\nabla \cdot (a\nabla) + V)^{-1}$ with $N = 64, L = 4$ and two choice of potential function V. The 2-norm is estimated using power method.

5.4 Conclusion

In this work, we present a novel algorithm for constructing a hierarchical matrix from its matrix-vector multiplication. One of the main motivations is the construction of the inverse matrix of the stiffness matrix of an elliptic differential operator. The proposed algorithm utilizes randomized singular value decomposition of low-rank

matrices. The off-diagonal blocks of the hierarchical matrix are computed through a top-down peeling process. This algorithm is efficient. For an $n \times n$ matrix, it uses only $\mathcal{O}(\log n)$ matrix-vector multiplications plus $\mathcal{O}(n \log n)$ additional steps. The algorithm is also friendly to parallelization. The resulting hierarchical matrix representation can be used as a faster algorithm for matrix-vector multiplications, as well as for numerical homogenization or upscaling.

The performance of our algorithm is tested using two 2D elliptic operators. The $\mathcal{H}^1$, the uniform $\mathcal{H}^1$ and the $\mathcal{H}^2$ versions of the proposed algorithms are implemented. Numerical results show that our implementations are efficient and accurate and that the uniform $\mathcal{H}^1$ representation is significantly more advantageous over $\mathcal{H}^1$ representation in terms of both the time cost and the memory cost, and $\mathcal{H}^2$ representation leads to further improvement for large N.

Although the algorithms presented require only $\mathcal{O}(\log n)$ matvecs, the actual number of matvecs can be quite large (for example, several thousands for the example in Section 5.3). Therefore, the algorithms presented here might not be the right choice for many applications. However, for computational problems in which one needs to invert the same system with a huge of unknowns or for homogenization problems where analytic approaches do not apply, our algorithm does provide an effective alternative.

The current implementation depends explicitly on the geometric partition of the rectangular domain. However, the idea of our algorithm can be applied to general settings. For problems with unstructured grid, the only modification is to partition the unstructured grid with a quadtree structure and the algorithms essentially require no change. For discretizations of the boundary integral operators, the size of an interaction list is typically much smaller as many boxes contain no boundary points. Therefore, it is possible to design a more effective combination strategy with small number of matvecs. These algorithms can also be extended to the 3D cases in a straightforward way, however, we expect the constant to grow significantly. All these

cases will be considered in the future.

Chapter 6

Conclusion of Part I

Part I of this dissertation has developed a novel method for solving KSDFT which is uniformly applicable to both insulating systems and metallic systems, at low temperature and at high temperature. This method is accurate and efficient, and the key element of the new method is that it focuses explicitly on the diagonal elements and the nearest off-diagonal elements that are needed to calculate the electron density and the ground state electron energy.

The new method is developed under the framework of Fermi operator expansion. The Fermi operator expansion method expands the Fermi operator $f(H)$ into simple functions. The contribution of each simple function to the electron density, and the ground state electron energy can be calculated directly without diagonalization. The Fermi operator expansion method includes four phases:

1. Discretization: discretization of the Hamiltonian operator;

2. Representation: representation of the Fermi operator into simple functions;

3. Evaluation: evaluation of the electron density and the ground state electron energy based on each simple function;

4. Iteration: self-consistent iteration of the electron density.

Part I of this dissertation has developed accurate and efficient methods for the discretization, the representation and the evaluation phases of Fermi operator expansion for solving KSDFT. The adaptive local basis method developed in Chapter 2 can be highly accurate with complexity comparable to tight binding method, *i.e.* the minimum possible number of basis functions per atom to discretize the Hamiltonian operator. The pole expansion developed in Chapter 3 achieves the optimal representation cost of the Fermi operator. The complexity of the pole expansion is $\mathcal{O}(\log \beta \Delta E)$. The selected inversion algorithm accurately calculates the electron density and the ground state energy and achieves lower computational cost than the standard diagonalization method uniformly for all dimensions. The complexity of the selected inversion is $\mathcal{O}(N)$ for one dimensional system, $\mathcal{O}(N^{1.5})$ for two dimensional system and $\mathcal{O}(N^2)$ for three dimensional systems.

In order to combine all the new methods developed in Part I of this dissertation into a practical software for electronic structure calculation, a large amount of work remains to be done.

The ground state electron energy calculated by the adaptive local basis functions can reach high accuracy with a small number of basis functions. This property is ideal for the selected inversion technique. By taking into account the block sparsity of the DG stiffness matrix, the pole expansion method and the selected inversion method can be combined to calculated the electron density and the ground state electron energy in the DG framework. The capability of the resulting algorithm is expected to be greatly enhanced compared to the current implementation as in either Chapter 2 or Chapter 4. To this end the selected inversion algorithm should be implemented in parallel as well. The parallelization of the selected inversion algorithm for general matrices has been addressed in the conclusion section of Chapter 4.

Besides the capability of the calculation of the ground state electron energy, another important aspect of an electronic structure software is *ab initio* molecular dy-

namics simulation. In order to perform ab initio molecular dynamics simulation, the derivatives of the basis functions with respect to the positions of the atoms (known as the Pulay force [213]) have to be systematically studied. This work is currently in progress.

The self-consistent iteration is a relative separate issue, since the self-consistent iteration does not directly contribute to the cubic scaling in the diagonalization method. However, it is not clear yet how to control the number of iterations for general large system, especially for general metallic systems. The self-consistent iteration will be systematically studied in future.

Part II

Nuclear quantum effects

Chapter 7

Introduction

Hydrogen bonded systems, including water and ice, are ubiquitous on earth. Hydrogen bond glues particles in soft matters, and the nature of the dynamics of protons (the nuclei of hydrogen atoms) plays a critical role in the behavior of biological systems [24] and chemical systems [130]. The vast majority of the numerical simulations for hydrogen bonded systems treat protons as classical particles. However, the behavior of protons is significantly affected by quantum mechanics even at ambient temperature due to the small nuclear mass. The isotopic effect in water [67], the quantum ferroelectric behavior of KH_2PO_4 (KDP) [219], and the formation of ice phases under high pressure [26], are just a few of the relevant phenomena where the nuclear quantum effects play a crucial role. Therefore, investigating the impact of nuclear quantum effects on molecular properties and equilibrium proton dynamics in hydrogen bond systems is the focus of intense research. The proton dynamics is reflected in the momentum distribution of protons due to the non-commutative relation between the momentum operator and the position operator in quantum mechanics. The proton momentum distribution can be computed from numerical simulation by means of path integral formalism [51, 83, 183, 243] and can be measured directly from Deep Inelastic Neutron Scattering (DINS) experiment [5, 218, 219].

The path integral formalism describes the equilibrium dynamics of nuclei by mapping a quantum system consisting of N particles to a equivalent classical system consisting of NP particles (in the cases discussed here $P \sim 30$). Such mapping is exact in the $P \to \infty$ limit if exchange effects among atoms can be neglected. The potential energy surface is then evaluated for P times compared to a classical simulation. If the potential energy surface is generated using first principle methods such as Kohn-Sham density functional theory discussed in Part I of this dissertation, the Kohn-Sham orbitals are also evaluated for P times. Given the high computational complexity in the evaluation of the potential energy surface, the computation of quantum momentum distribution is a demanding task, and efficiency is a crucial issue. However, the open path integral formalism can only evaluate the quantum momentum distribution for one particle at a time, even if all the particles share the same environment and are essentially equivalent with each other. One can choose to evaluate the momentum distribution for multiple particles at the same time, but the accuracy has to be sacrificed in a way that is difficult to control *a priori*.

Part II of this dissertation develops the novel displaced path integral formalism which converts the problem of calculating quantum momentum distribution into a problem of calculating free energy differences. The displaced path integral formalism can therefore be combined with a large pool of free energy calculation techniques to improve the computational efficiency. This dissertation demonstrates that when combined with free energy perturbation method, the quantum momentum distributions for all particles can be computed at the same time with a standard closed path integral simulation. The resulting formulation is shown to be more efficient than the open path integral formalism when applied to a force-field water system. Furthermore, in the displaced path integral formalism, the end-to-end distribution, *i.e.* the Fourier transform of the momentum distribution factorizes into a free particle part and an environmental part. This factorization facilitates the interpretation of the quantum

momentum distribution, since only the environmental part contains the information of the potential energy surface. The displaced path integral formalism also gives rise to a novel semiclassical analysis of the quantum momentum distribution, and provides a new kinetic energy estimator.

After obtaining the quantum momentum distribution, it remains a difficult task to extract the information of proton dynamics from the momentum distribution. The proton dynamics is governed by the potential energy surface of the system. For a system consisting of N atoms, the potential energy surface is a $3N$ dimensional function, while the momentum distribution is only a 3 dimensional quantity. In principle there are infinite potential energy surfaces corresponding to the same momentum distribution. The interpretation of the momentum distribution can therefore be highly ambiguous and controversial. This dissertation resolves such ambiguity in the interpretation of the proton momentum distribution using two representative examples as follows.

Recent DINS studies have observed a secondary feature in the tail of the spherically averaged distribution in confined water [90], and a large excess kinetic energy in supercooled water [208]. Such features were attributed to quantum tunneling between the two wells of an effective anharmonic 1D potential. However, anisotropy can mimic features of a spherical distribution that one might associate to anharmonicity in a 1D model [233]. Even in a much simpler system such as monocrystalline ice Ih, the relative importance of anisotropy and anharmonicity remains unclear. Most of the current DINS experiments can only be performed on polycrystalline samples, where only the spherically averaged momentum distribution could be measured. Most of the path integral molecular dynamics simulations only report the spherical momentum distribution but not the full 3D momentum distribution and the effect of the anisotropicity. The unknown details of the full 3D momentum distribution due to the spherical averaging operation severely increases the difficulty for the interpretation.

Part II of this dissertation clarifies the relation between anisotropy and anharmonicity in ice Ih by analyzing the 3D proton momentum distribution obtained from the *ab initio* path integral simulation. The proton momentum distribution is found to be well described by an anisotropic Gaussian distribution originating from a quasi-harmonic model for the potential of the mean force of protons. Anisotropy stemming from the molecular orientations in the crystal has clearly a larger effect in shaping the momentum distribution than anharmonicity. The large effect of molecular anisotropy implies that it is not possible to unambiguously attribute features of the spherically averaged distribution to anharmonicity. Part II of this dissertation reveals the direct relation between the principal frequencies of the quantum momentum distribution (*i.e.* the eigenvalues of the covariance matrix of the momentum distribution) and the phonon spectrum in the vibrational dynamics. The full path integral simulation result is to a large extent in agreement with the vibrational dynamics, which supports the quasi-harmonic form for the potential of the mean force. The remaining deviation between the path integral simulation and the vibrational dynamics is mainly visible along the hydrogen bond direction, indicating the anharmonic effect along the bond.

The proton dynamics becomes more challenging in the presence of proton tunneling which is beyond the quasi-harmonic regime. Proton tunneling is important in phase transitions such as the ferroelectric to paraelectric transition in KDP and the sequence of transitions leading to hydrogen bond symmetrization in high pressure ice. In the case of high pressure ice, at large inter-oxygen distance such as $d_{OO} \sim 2.53$Å typically of ice VIII, the system is characterized by asymmetric hydrogen bonds and satisfies the ice rule [29, 203], which means that on the four hydrogen bonds connecting an oxygen to its neighboring oxygens, two protons are near the central oxygen and two are near the neighboring oxygens, as is required to keep the water molecules intact. As the inter-oxygen distance decreases to $d_{OO} \sim 2.45$Å typically of ice VII, the protons become delocalized along the hydrogen bonds, accompanied by the ap-

pearance of ionized configurations such as H_3O^+ and OH^- or H_4O^{++} and O^{--} which locally break the ice rule. The standard picture to interpret path integral studies was based on mean field theory [26, 27], and proton correlation effects were not taken into consideration. The mean field theory overestimates the number of ionized configurations which is called the ionization catastrophe [235].

Part II of this dissertation unambiguously assesses the important role of proton correlation effects in high pressure ice by means of spectral decomposition of the single particle density matrix, which contains the information of the momentum distribution as well as that of the spatial distribution. The correlation effects among protons manifests themselves in the concerted proton tunneling process which is directly observed in the simulation by the study of the centroid of the paths in imaginary time. The total energy cost of a concerted proton tunneling process is lower than that of a complete ionization catastrophe predicted by the mean field theory. Concerted proton tunneling reduces the number of ionized configurations and the local charge neutrality is partially restored. Finally, this dissertation demonstrates that the correlated character of proton dynamics can be described in terms of an ensemble of potentials of the mean force, which provides a more accurate description of the hydrogen bond symmetrization transitions than that predicted by a single potential of the mean force in the mean field theory.

Part II of this dissertation is organized as follows: Chapter 8 gives a short introduction on the quantum momentum distribution, and develops the displaced path integral formalism. Chapter 9 discusses the relative importance of anisotropy and anharmonicity for the proton momentum distribution in ice Ih, and its relation to the vibrational dynamics. This discussion is followed in Chapter 10 for the theoretical interpretation of a recently performed DINS experiment on the proton momentum distribution in ice Ih. The correlation effect in hydrogen bonded systems with proton tunneling is illustrated for ice under high pressure in Chapter 11. The conclusion of

Part II of this dissertation is given in Chapter 12.

Chapter 8

Displaced path integral formalism

8.1 Introduction

The momentum distribution of quantum particles conveys unique information of the potential energy surface of the system and is of great interest in practice. The momentum distribution can be measured with Deep Inelastic Neutron Scattering experiments (DINS), and calculated with computer simulation. In this chapter, we discuss the computational methods for the quantum momentum distribution. To simplify the notation, we first discuss the momentum distribution for a single particle under an external potential as follows. The momentum distribution $n(\boldsymbol{p})$ is expressed in terms of the single particle density matrix ρ as

$$\begin{aligned} n(\boldsymbol{p}) =& \langle \boldsymbol{p} \left| \rho \right| \boldsymbol{p} \rangle \\ =& \int d\boldsymbol{r} d\boldsymbol{r}' \langle \boldsymbol{p} | \boldsymbol{r} \rangle \left\langle \boldsymbol{r} \left| \rho \right| \boldsymbol{r}' \right\rangle \langle \boldsymbol{r}' | \boldsymbol{p} \rangle \\ =& \frac{1}{(2\pi\hbar)^3} \int d\boldsymbol{r} d\boldsymbol{r}' e^{\frac{i}{\hbar}\boldsymbol{p}\cdot(\boldsymbol{r}-\boldsymbol{r}')} \rho(\boldsymbol{r}, \boldsymbol{r}') \\ =& \frac{1}{(2\pi\hbar)^3} \int d\boldsymbol{x}\ e^{\frac{i}{\hbar}\boldsymbol{p}\cdot\boldsymbol{x}} \widetilde{n}(\boldsymbol{x}) \end{aligned} \tag{8.1}$$

The end-to-end distribution $\widetilde{n}(\boldsymbol{x})$ defined in the last equality is the Fourier transform of the momentum distribution. The end-to-end distribution characterizes the information along the off-diagonal elements of the density matrix:

$$\widetilde{n}(\boldsymbol{x}) = \langle \delta(\boldsymbol{r} - \boldsymbol{r}' = \boldsymbol{x}) \rangle = \int d\boldsymbol{r} d\boldsymbol{r}' \delta(\boldsymbol{r} - \boldsymbol{r}' = \boldsymbol{x}) \rho(\boldsymbol{r}, \boldsymbol{r}'). \tag{8.2}$$

In a condensed system particles move in a high dimensional space and statistical sampling is the only viable computational strategy for calculating the momentum distribution. The statistical sampling is usually done using the Feynman path representation [83]. The path integral discretization of the density matrix maps the quantum system onto a set of P replicas ("beads") that obey classical physics, thereby allowing one to utilize the machinery of computational classical statistical mechanics, namely Monte Carlo and molecular dynamics strategies. The discretized density matrix takes the form

$$\rho(\boldsymbol{r}, \boldsymbol{r}') = \lim_{P \to \infty} \int_{\substack{\boldsymbol{r}_1 = \boldsymbol{r} \\ \boldsymbol{r}_{P+1} = \boldsymbol{r}'}} \mathrm{d}r_2 \ldots \mathrm{d}r_P \, e^{-\beta U_{\text{eff}}} \tag{8.3}$$

$$U_{\text{eff}} = \sum_{i=1}^{P} \frac{mP}{2\hbar^2 \beta^2} |\boldsymbol{r}_i - \boldsymbol{r}_{i+1}|^2 + \frac{V(\boldsymbol{r}_1) + V(\boldsymbol{r}_{p+1})}{2P} + \sum_{i=2}^{P} \frac{V(\boldsymbol{r}_i)}{P} \tag{8.4}$$

In the limit $P \to \infty$, the density matrix may be written in the form of continuous path integral as [51]

$$\rho(\boldsymbol{r}, \boldsymbol{r}') = \int_{\boldsymbol{r}(\beta\hbar) = \boldsymbol{r}', \boldsymbol{r}(0) = \boldsymbol{r}} \mathfrak{D}\boldsymbol{r}(\tau) e^{-\frac{1}{\hbar} \int_0^{\beta\hbar} d\tau \left(\frac{m\dot{\boldsymbol{r}}^2(\tau)}{2} + V[\boldsymbol{r}(\tau)] \right)}. \tag{8.5}$$

Eq. (8.4) and (8.5) are introduced under the single particle picture but can be easily extended to multi-particle systems. If the system consists of M distinguishable par-

ticles, the momentum distribution corresponding to the l-th particle can be obtained by sampling the single particle density matrix corresponding to the l-th particle:

$$\rho_l(\boldsymbol{r},\boldsymbol{r}') = \int_{\boldsymbol{R}(\beta\hbar)=\boldsymbol{R}',\boldsymbol{R}(0)=\boldsymbol{R}} \mathfrak{D}\boldsymbol{R}(\tau) e^{-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \, \frac{m\dot{\boldsymbol{R}}^2(\tau)}{2} + V[\boldsymbol{R}(\tau)]}. \tag{8.6}$$

Here we have used the compact notation

$$\boldsymbol{R}(\tau) = \big(\boldsymbol{r}_1(\tau), \ldots, \boldsymbol{r}_{l-1}(\tau), \boldsymbol{r}_l(\tau), \boldsymbol{r}_{l+1}(\tau), \ldots, \boldsymbol{r}_M(\tau)\big),$$

and

$$\boldsymbol{R} = \big(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{l-1}, \boldsymbol{r}, \boldsymbol{r}_{l+1}, \ldots, \boldsymbol{r}_M\big),$$
$$\boldsymbol{R}' = \big(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{l-1}, \boldsymbol{r}', \boldsymbol{r}_{l+1}, \ldots, \boldsymbol{r}_M\big).$$

Note that only the path $r_l(\tau)$ is "open" with the condition that $\boldsymbol{r}_l(0) = r, \boldsymbol{r}_l(\beta\hbar) = r'$. All other paths $\boldsymbol{r}_k(\tau), k \neq l$ are "closed" with the condition that $\boldsymbol{r}_k(0) = \boldsymbol{r}_k(\beta\hbar)$.

From Eq. (8.6) we can understand the challenge in computing the momentum distribution in a condensed system. In order to calculate the momentum distribution of the l-th particle, it is required to open the path for the l-th particle with all other particles being represented by closed paths. However, in a bulk material with a large number of particles of the same species, the momentum distribution can only be garnered for one particle at a time. This leads to a very inefficient sampling process. It has been shown that if the paths of multiple particles are "opened" and these paths are sufficiently far apart from each other, the impact upon the resultant distribution is negligible [192]. In general, this strategy requires one to balance two contradictory requirements. On one hand the number of open paths has to be large enough to obtain good statistics, while on the other hand it cannot be too large as the sampling will become incorrect.

This dissertation develops a novel method, called the displaced path integral for-

malism that is more efficient in sampling the momentum distribution. In the displaced path integral formalism, the momentum distribution can be garnered from a post-processing step using the trajectory of a standard closed path integral simulation. Numerical examples using one dimensional model potentials as well as a three-dimensional water system indicate that the new method is accurate and efficient. The displaced path integral formalism is also conceptually advantageous. The end-to-end distribution factorizes into a free particle part and an environmental part. The information of the potential energy surface is completely contained in the environmental part, which facilitates the interpretation of the simulation and experimental result. Furthermore, the displaced path integral formalism allows a novel semiclassical analysis of the quantum momentum distribution. This semiclassical analysis is in parallel to the Feynman-Hibbs analysis of the closed path integral formulation [83], and is shown to be applicable to a large class of quantum systems. The displaced path integral formalism also provides a new kinetic energy estimator for quantum particles.

This chapter is organized as follows: Section 8.2 derives the displaced path integral formulation, and illustrates its performance in different regimes using a one dimensional example. Section 8.3 applies the displaced path integral formalism to water system using a force field model, and shows that the new method is accurate and efficient in many body systems. The discussion is followed in Section 8.4 which introduces a new quantity, called the mean force, for interpreting the quantum momentum distribution. The displaced path integral formalism also serves as a new tool for the semiclassical analysis of the quantum momentum distribution, and we discuss this in Section 8.5. In Section 8.6, the displaced path formalism also provides a new kinetic energy estimator. Finally, the displaced path integral formalism is introduced for distinguishable particles, but can also be extended to indistinguishable particles. Section 8.7 generalizes the displaced path integral formalism to bosonic systems, fol-

lowed by the conclusion of this chapter in Section 8.8. Part of the materials in this chapter have been presented in [166].

8.2 Displaced path integral formalism

To simplify the notation, we introduce the displaced path formalism in this section for a single particle under an external potential. The end-to-end distribution $\widetilde{n}(\boldsymbol{x})$ is the Fourier transform of the momentum distribution

$$\widetilde{n}(\boldsymbol{x}) = \mathrm{Tr}[e^{-\frac{i}{\hbar}\boldsymbol{p}\cdot\boldsymbol{x}}e^{-\beta H}]/Z \equiv \frac{Z(\boldsymbol{x})}{Z}, \tag{8.7}$$

and $\widetilde{n}(\boldsymbol{x})$ can be expressed in the open path integral formulation [51]

$$\begin{aligned}\widetilde{n}(\boldsymbol{x}) =& \frac{1}{Z}\int d\boldsymbol{r}d\boldsymbol{r}'\delta\left(\boldsymbol{r}-\boldsymbol{r}'-\boldsymbol{x}\right)\rho\left(\boldsymbol{r},\boldsymbol{r}'\right) \\ =& \frac{\int_{\boldsymbol{r}(0)-\boldsymbol{r}(\beta\hbar)=\boldsymbol{x}}\mathfrak{D}\boldsymbol{r}(\tau)e^{-\frac{1}{\hbar}\int_0^{\beta\hbar}d\tau\left(\frac{m\dot{\boldsymbol{r}}^2(\tau)}{2}+V[\boldsymbol{r}(\tau)]\right)}}{\int_{\boldsymbol{r}(\beta\hbar)=\boldsymbol{r}(0)}\mathfrak{D}\boldsymbol{r}(\tau)e^{-\frac{1}{\hbar}\int_0^{\beta\hbar}d\tau\left(\frac{m\dot{\boldsymbol{r}}^2(\tau)}{2}+V[\boldsymbol{r}(\tau)]\right)}}.\end{aligned} \tag{8.8}$$

As illustrated in Section 8.1, in a many particle system, Eq. (8.8) requires to open the path for only one particle with all other particles being represented by closed paths. Therefore one particle is marked as special among all the particles in the system, even if all the particles are embedded in the same ambient environment, and the momentum distribution can only be calculated one particle at a time. It is desirable to find an alternative algorithm that can calculate the momentum distribution of all particles at the same time. This objective essentially requires that all the paths have to remain closed during the simulation.

One possible way to evaluate the end-to-end distribution from closed path integral formalism is the perturbation method. In the discrete setup as in Eq. (8.3) and (8.4), one can perturb the end point of the path $\boldsymbol{r}' = \boldsymbol{r}_{P+1}$ by a small amount

away from the starting point of the path $\boldsymbol{r} = \boldsymbol{r}_1$, and the end-to-end distribution is calculated by sampling a corresponding estimator along the closed path integral trajectory. However, as one refines the discretization along the imaginary time and increases the number of beads P to infinity, the variance of this estimator must also goes to infinity. Therefore, the simple perturbation method is problematic and does not have a well-defined continuous limit. This resembles the scenario of the primitive kinetic energy estimator where the variance of the estimator of the quantum kinetic energy goes to infinity when $P \to \infty$ [124].

The displaced path formalism developed in this dissertation solves the problem of infinite variance mentioned above by means of a simple transform. The open path $\boldsymbol{r}(\tau)$ is converted to a closed path $\widetilde{\boldsymbol{r}}(\tau)$ by applying a $\boldsymbol{x}$-dependent linear transformation in the path space:

$$\boldsymbol{r}(\tau) = \widetilde{\boldsymbol{r}}(\tau) + y(\tau)\boldsymbol{x}, \tag{8.9}$$

Here $y(\tau) = C - \frac{\tau}{\beta\hbar}$ and C is an arbitrary constant. Then the numerator of (8.8) becomes

$$\begin{aligned} &\int_{\boldsymbol{r}(0)-\boldsymbol{r}(\beta\hbar)=\boldsymbol{x}} \mathfrak{D}\boldsymbol{r}(\tau) e^{-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \left(\frac{m\dot{\boldsymbol{r}}^2(\tau)}{2} + V[\boldsymbol{r}(\tau)]\right)} \\ =& e^{-\frac{m\boldsymbol{x}^2}{2\beta\hbar^2}} \int_{\widetilde{\boldsymbol{r}}(\beta\hbar)=\widetilde{\boldsymbol{r}}(0)} \mathfrak{D}\widetilde{\boldsymbol{r}}(\tau) e^{-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \left(\frac{m\dot{\widetilde{\boldsymbol{r}}}^2(\tau)}{2} + V[\widetilde{\boldsymbol{r}}(\tau)+y(\tau)\boldsymbol{x}]\right)}. \end{aligned} \tag{8.10}$$

In Eq. (8.10), the term $e^{-\frac{m\boldsymbol{x}^2}{2\beta\hbar^2}}$ corresponds to the exact end-to-end distribution for a free particle system. This term comes naturally from the derivative of $y(\tau)$. The choice of the constant C influences the variance of estimators for the end-to-end distribution. It is found that the lowest variance is achieved when $C = 1/2$, since this choice has the smallest displacement from the closed path configuration. Therefore under the displaced path integral formalism, the end-to-end distribution is

represented as

$$\widetilde{n}(\boldsymbol{x}) = \widetilde{n}_0(\boldsymbol{x}) \frac{\int \mathfrak{D}\boldsymbol{r}(\tau) \exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \ \left(\frac{m\dot{\boldsymbol{r}}^2(\tau)}{2} + V[\boldsymbol{r}(\tau) + y(\tau)\boldsymbol{x}]\right)\right)}{\int \mathfrak{D}\boldsymbol{r}(\tau) \exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \ \left(\frac{m\dot{\boldsymbol{r}}^2(\tau)}{2} + V[\boldsymbol{r}(\tau)]\right)\right)}, \tag{8.11}$$

where $\widetilde{n}_0(\boldsymbol{x}) = e^{-\frac{m\boldsymbol{x}^2}{2\beta\hbar^2}}$ is the end-to-end distribution corresponding to a free particle.

Computational advantages arise from the explicit factorization of $\widetilde{n}_0(\boldsymbol{x})$ and the remaining part, called the "environmental part". It follows from Eq. (8.11) that, having put $Z(\boldsymbol{0}) = Z$, we can write $\frac{\widetilde{n}(\boldsymbol{x})}{\widetilde{n}_0(\boldsymbol{x})} = \frac{Z(\boldsymbol{x})}{Z(\boldsymbol{0})}$ as a ratio between two partition functions. To calculate this ratio or its logarithm, called the "potential of the mean force" or "excess free energy" $U(\boldsymbol{x}) = -\ln\frac{Z(\boldsymbol{x})}{Z(\boldsymbol{0})}$, one can apply all the methods available in standard statistical mechanics, and open path integral method [51] becomes one of the many choices. Below we introduce two methods that are more efficient than the open path integral method. The first method combines the displaced path integral formalism with the free energy perturbation method [258], and the second method combines the displaced path integral formalism with thermodynamic integration method [137].

The free energy perturbation method calculates the end-to-end distribution $\widetilde{n}(\boldsymbol{x})$ by perturbation from the closed path configuration:

$$\begin{aligned}
\widetilde{n}(\boldsymbol{x}) =& \frac{Z(\boldsymbol{x})}{Z(0)} \\
=& \widetilde{n}_0(\boldsymbol{x})\widetilde{n}_V(\boldsymbol{x}) \\
=& \widetilde{n}_0(\boldsymbol{x}) \frac{\int \mathfrak{D}\boldsymbol{r}(\tau)\mathcal{N}(\boldsymbol{x};0) \exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \ \left(\frac{m\dot{\boldsymbol{r}}^2(\tau)}{2} + V[\boldsymbol{r}(\tau)]\right)\right)}{\int \mathfrak{D}\boldsymbol{r}(\tau) \exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \ \left(\frac{m\dot{\boldsymbol{r}}^2(\tau)}{2} + V[\boldsymbol{r}(\tau)]\right)\right)} \\
\equiv& \widetilde{n}_0(\boldsymbol{x})\langle\mathcal{N}(\boldsymbol{x};0)\rangle_0,
\end{aligned} \tag{8.12}$$

The estimator for the environmental part of the end-to-end distribution is denoted

by

$$\mathcal{N}(\boldsymbol{x};0) = e^{-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\ (V[\boldsymbol{r}(\tau)+y(\tau)\boldsymbol{x}]-V[\boldsymbol{r}(\tau)])}. \tag{8.13}$$

The free energy perturbation method only requires a standard closed path integral simulation, and the end-to-end distribution can be calculated using a post-processing step with the estimator in Eq. (8.13), which is the difference of the potential energy between the displaced path configuration and the closed path configuration. The variance of the estimator (8.13) is finite, and is small in many systems as will be shown later. The potential of the mean force is calculated as the logarithm of the environmental part of the end-to-end distribution

$$U(\boldsymbol{x}) = -\ln \langle \mathcal{N}(\boldsymbol{x};0) \rangle_{\mathbf{0}}\,. \tag{8.14}$$

where the average is evaluated using the closed path distribution $Z(\mathbf{0})$.

Now we apply the free energy perturbation method to study the momentum distribution of a single particle in an external double well potential

$$V = \frac{m\omega^2}{2}x^2 + Ae^{-(x/b)^2} \tag{8.15}$$

with $\omega = 0.0050$, $b = 0.25$ and $A = 0.012$ in atomic unit. The particle is assumed to have proton mass $m = 1836$. This potential mimics the potential of the mean force along the hydrogen bond direction in tunneling ice [190]. The barrier of this potential is 2400K. 32 beads are used for discretizing the path along the imaginary time and the length of the trajectory is $30ps$. The end-to-end distribution obtained from the displaced path formalism is compared with that from the open path integral method (Figure 8.1). The exact end-to-end distribution obtained by directly diagonalizing the Hamiltonian is also included for comparison. The results obtained from both the open path integral method and the displaced path method are consistent with the exact

result, while the result in the open path integral method contains more noise than that in the displaced path method. The potential of the mean force corresponding to the double well model is also compared between the two methods (Fig. 8.2). Despite the strong anharmonic feature in the potential energy surface, the displaced path method is able to accurately calculate the potential of the mean force.

The displaced path method is already more efficient than the open path integral method for a single particle. The reason is that the end-to-end distribution is a smooth function, and the end-to-end distribution can be well approximated by its values at a few points along the x axis. The displaced path formalism exactly calculates the end-to-end distribution on a certain prescribed set of end-to-end distance x. On the other hand, the open path integral samples the values of the end-to-end distribution at all points. As a result the statistics on each point is less than that in the displaced path formulation.

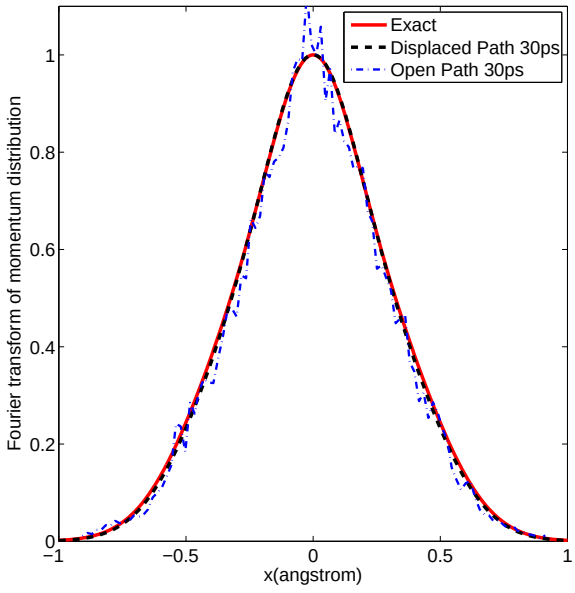


Figure 8.1: The end-to-end distribution of a particle in a double well potential at 300K obtained from exact diagonalization (red solid line), from displaced path method (black dashed line), and from the open path integral method (blue dot dashed line).

The natural factorization of the free particle part and the environmental part in

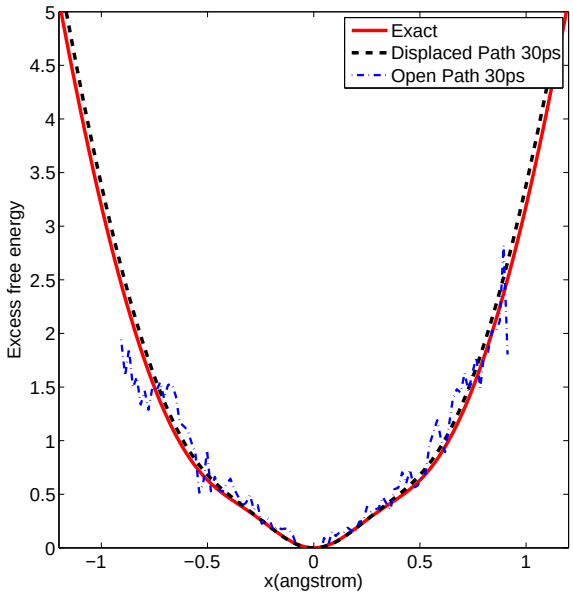


Figure 8.2: The potential of the mean force of a particle in a double well potential at 300K. Red solid line: exact result. Black dashed line: displaced path formulation with 30ps data. Blue dot dashed line: open path simulation with 30ps data. The potential of the mean force is in the unit of k_BT.

the end-to-end distribution also facilitates the visualization of the quantum effect in the momentum distribution. Since the quantum effect is only included in the environmental part, we can define the environmental part of the momentum distribution as

$$\widetilde{n}(p) = \frac{1}{2\pi\hbar}\int e^{\frac{i}{\hbar}px}\widetilde{n}(x), \tag{8.16}$$

The momentum distribution $n(p)$ and the environmental part of the momentum distribution $\widetilde{n}(p)$ are compared in Fig. 8.3. The quantum effect is only indicated as the extended tail in the momentum distribution, while the quantum effect is much more amplified in the environmental part of the momentum distribution, which exhibits a node point around 4Å^{-1}. This existence of the node in the environmental part of the momentum distribution indicates strong anharmonicity in the potential energy surface.

The free energy perturbation method gives accurate description of the end-to-end

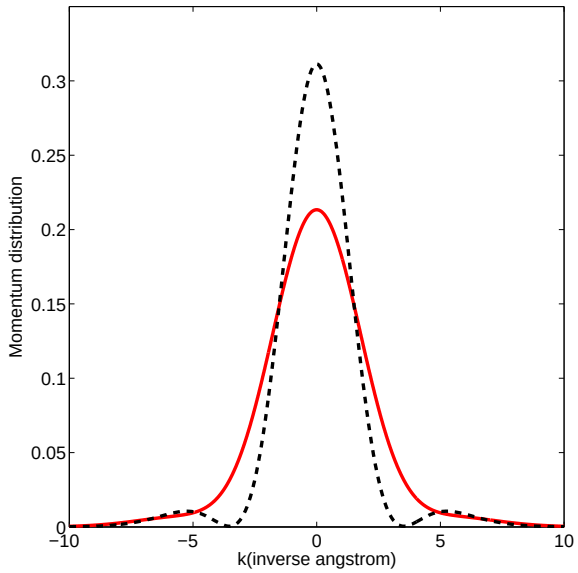


Figure 8.3: Red solid line: momentum distribution $n(p)$. Black dashed line: environmental part of the momentum distribution $\widetilde{n}(p)$, where the quantum effect is amplified.

distribution of the double well system even when the quantum effect is relatively large (Fig. 8.3). However, in some cases the perturbation method can be less accurate. This occurs when the system is at low temperature and β becomes large. As the variance of the estimator (8.13) increases with β, the free energy perturbation method becomes increasingly inefficient. To illustrate this case, we use the double well potential (8.15) again, but at a different temperature as 100K. The path integral is discretized with 64 beads and the length of the trajectory is 300ps. The potential of the mean force, and the momentum distribution are calculated using the displaced path formulation (Fig. 8.4 and Fig. 8.5, respectively). Compared to the exact potential of the mean force and the momentum distribution using diagonalization method, we find that the accuracy of the displaced path estimator is reduced when the system is at lower temperature. The statistical accuracy can be improved with a longer trajectory, however, it is desirable to design more efficient methods to calculate the momentum distribution in this case.

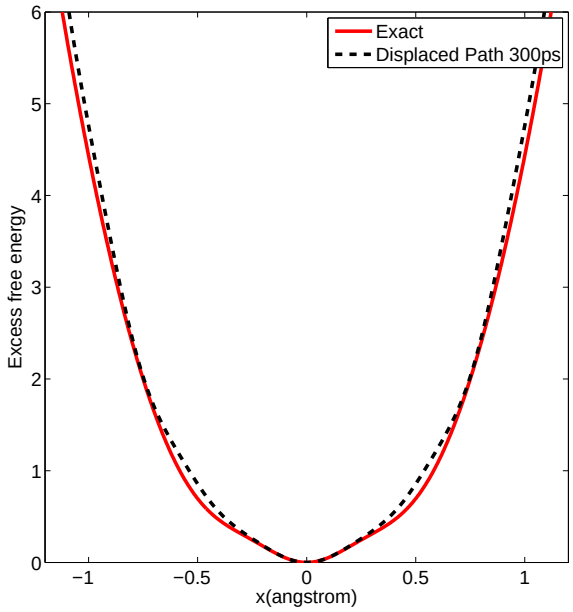


Figure 8.4: The potential of the mean force of a particle in a double well potential at 100K, obtained from the exact diagonalization method (red solid line), and from the displaced path method (black dashed line). The unit of the potential of the mean force is $k_B T$.

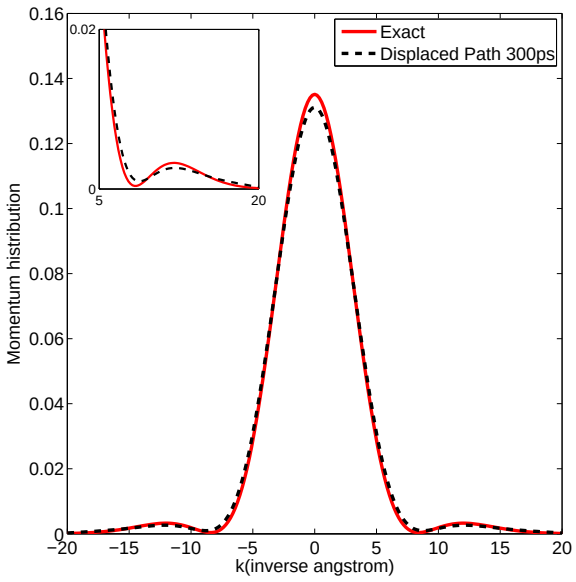


Figure 8.5: The momentum distribution of a particle in a double well potential at 100K. Red solid line: exact result. Black dashed line: displaced path formula with $300ps$ data. An inset with the same legend is included to describe difference in the second shoulder.

For fixed trajectory length in simulation, the accuracy of the free energy perturbation method is determined by the variance of the estimator $\mathcal{N}(x;0)$, which takes the form

$$\begin{aligned}&\langle \mathcal{N}(x;0)^2\rangle_0 - \langle \mathcal{N}(x;0)\rangle_0^2 \\ =&\langle e^{-\frac{2}{\hbar}\int_0^{\beta\hbar} d\tau\, V[r(\tau)+y(\tau;x)]-V[r(\tau)]}\rangle_0 - \langle e^{-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\, V[r(\tau)+y(\tau;x)]-V[r(\tau)]}\rangle_0^2.\end{aligned} \tag{8.17}$$

The variance of $\mathcal{N}(x;0)$ at different x is computed for the double well potential at 300K (left panel in Fig. 8.6) and at 100K (right panel in Fig. 8.6), respectively. The variance of the estimator at 100K is about 25 times bigger than that at 300K. For a general estimator A, the sampling statistical error $\sigma^2(A)$ can be expressed as [86, Appendix D]

$$\sigma^2(A) \approx \frac{2t_A^c}{t}(\langle A^2\rangle - \langle A\rangle^2). \tag{8.18}$$

Here t_A^c is the characteristic decay time of the correlation function of A. Therefore the sampling statistical error at 100K is much larger that than at 300K using a trajectory of the same length, and the free energy perturbation method is not very efficient at 100K.

The discussion above indicates that the free energy perturbation method is mostly suited for studying the momentum distribution of quantum systems at intermediate temperature. If the system is at low temperature, or the potential energy surface of the quantum system has a large fluctuation, the estimator in the free energy perturbation method will have a large variance and the statistical error is difficult to reduce. To overcome this problem, we note that the variance of $\mathcal{N}(x;0)$ is not uniform along x direction (see the right panel of Fig. 8.6). The variance is small when x is small ($|x| < 0.1$Å) or large ($|x| > 1.0$Å). The largest variance occurs at intermediate displacement $|x| \approx 0.5$Å. The variance can be reduced by inserting an intermediate

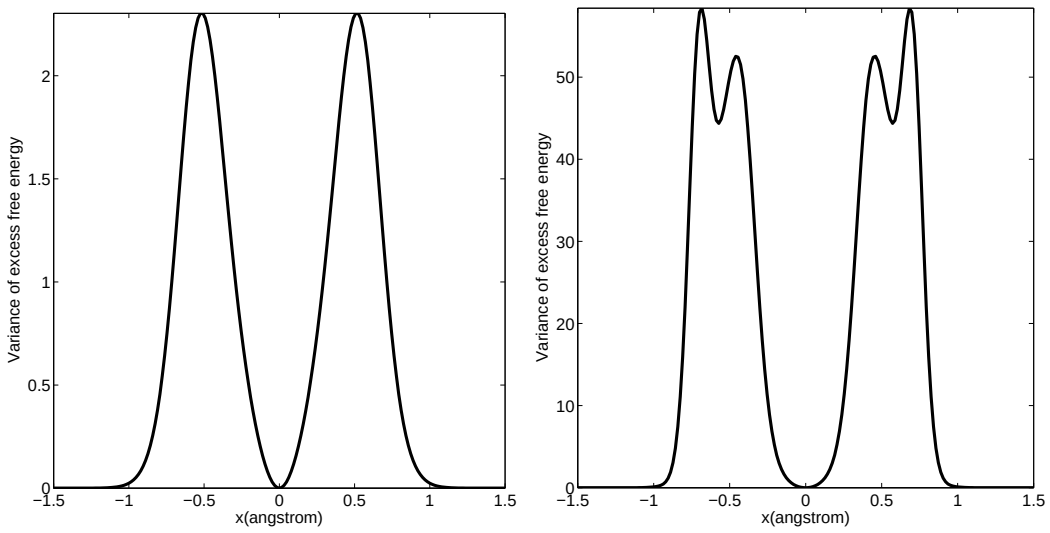


Figure 8.6: The variance of $\mathcal{N}(x;0)$ for double well model at 300K (left panel) and at 100K (right panel).

point x' in the free energy perturbation method as

$$\begin{aligned}\widetilde{n}_V(x) =& \frac{Z_V(x)}{Z_V(0)} = \frac{Z_V(x)}{Z_V(x')}\frac{Z_V(x')}{Z_V(0)} \\ =& \frac{\int \mathfrak{D}r(\tau)\exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\ \frac{m\dot{r}^2(\tau)}{2} + V[r(\tau)+y(\tau;x)]\right)}{\int \mathfrak{D}r(\tau)\exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\ \frac{m\dot{r}^2(\tau)}{2} + V[r(\tau)+y(\tau;x')]\right)}\times \\ & \frac{\int \mathfrak{D}r(\tau)\exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\ \frac{m\dot{r}^2(\tau)}{2} + V[r(\tau)+y(\tau;x')]\right)}{\int \mathfrak{D}r(\tau)\exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\ \frac{m\dot{r}^2(\tau)}{2} + V[r(\tau)]\right)} \\ =& \langle \mathcal{N}(x;x')\rangle_{x'}\langle \mathcal{N}(x';0)\rangle_0\end{aligned} \tag{8.19}$$

The last equality in Eq. (8.19) also defines a new quantity $\mathcal{N}(x;x')$, which is the estimator of the ratio of the end-to-end distribution at x to the end-to-end distribution at x'. When $x'=0$, $\mathcal{N}(x;0)$ is the end-to-end distribution and Eq. (8.19) is the same as Eq. (8.13).

Instead of computing $\mathcal{N}(x;0)$ for all x, Eq. (8.19) first computes $\mathcal{N}(x;0)$ for $0 < x \le x'$ and then calculates $\mathcal{N}(x;x')$ for $x > x'$. The variance of the estimator is therefore reduced. Eq. (8.19) can be applied recursively by injecting multiple inter-

mediate points x'. In practice a few intermediate points is already able to accurately calculate the momentum distribution. Take the double well potential at $100K$ for instance. We apply Eq. (8.19) with intermediate points $x' = 0, 0.11, 0.26, 0.42, 0.53$Å, respectively, with the sampling trajectory of 60ps at each intermediate points with the path integral discretized by 64 beads. The total length of the trajectory is still 300ps. The accuracy of the momentum distribution and the potential of the mean force is greatly improved (see Fig. 8.7 and Fig. 8.8). The variance of the estimator $\mathcal{N}(x, x')$ is more than 25 times smaller than that of the free energy perturbation estimator (see Fig. 8.9). The discontinuity in Fig. 8.9 indicates the positions of the intermediate points.

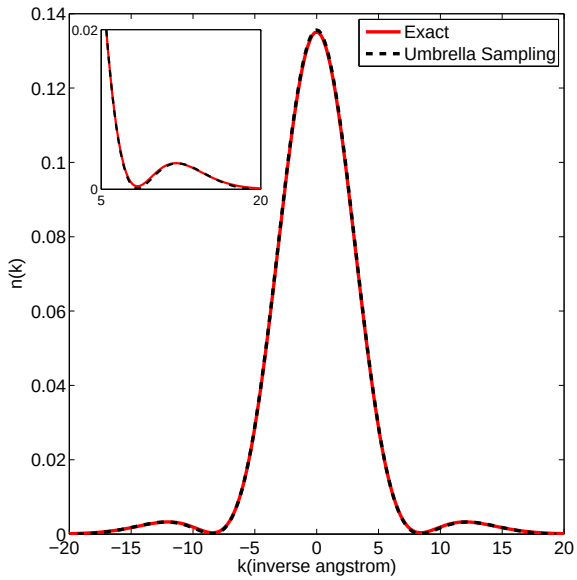


Figure 8.7: The momentum distribution of a particle in a double well potential at 100K using Eq. (8.19). Red solid line: exact result obtained by diagonalization of the Hamiltonian matrix. Black dashed line: displaced path formula (8.19). An inset with the same legend is included for better illustration of the tail of the momentum distribution.

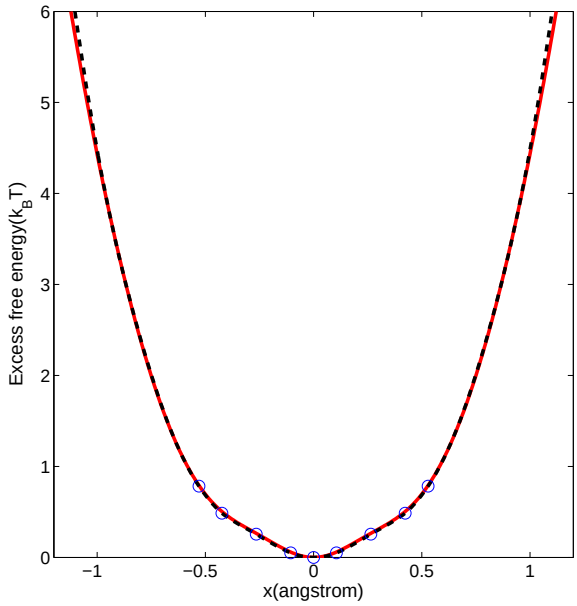


Figure 8.8: The potential of the mean force of a particle in a double well potential at 100K. Red solid line: exact result. Black dashed line: Displaced path formula (8.19). The potential of the mean force is in the unit of k_BT.

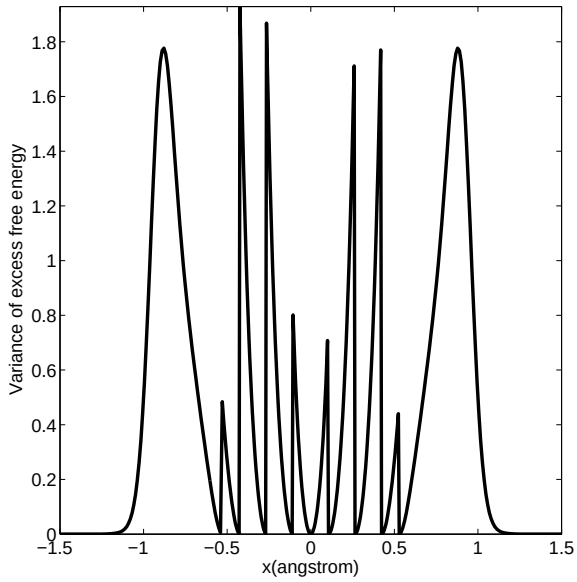


Figure 8.9: The variance for estimating the end-to-end distribution for 100K double well model using Eq. (8.19). The discontinuity indicates the intermediate points to enhance the sampling efficiency.

We have demonstrated that the free energy perturbation method is very effective even at low temperature by inserting several intermediate points x'. It is natural to ask what is the continuous limit of Eq. (8.19) by inserting infinite number of intermediate points. This continuous limit is the thermodynamic integration formula in statistical mechanics [137], which calculates the potential of the mean force by integrating its gradient $\boldsymbol{F}(\boldsymbol{x}')$:

$$U(\boldsymbol{x}) = \int_0^{\boldsymbol{x}} d\boldsymbol{x}' \cdot \boldsymbol{F}(\boldsymbol{x}'). \tag{8.20}$$

The derivative quantity $\boldsymbol{F}(\boldsymbol{x}')$, called the mean force, can be sampled directly from the path integral simulation according to the intermediate partition function $Z(\boldsymbol{x}')$

$$\boldsymbol{F}(\boldsymbol{x}') = \left\langle \frac{1}{\hbar} \int_0^{\beta\hbar} d\tau \ \nabla_{\boldsymbol{r}} V[\boldsymbol{r}(\tau) + y(\tau)\boldsymbol{x}']y(\tau) \right\rangle_{\boldsymbol{x}'} \tag{8.21}$$

We remark that both Eq. (8.19) and the thermodynamic integration formula (8.21) require inserting intermediate points x' which leads to open paths. However, Eq. (8.19) and Eq. (8.21) are more accurate methods for calculating the momentum distribution, and can be applied in a fully controlled way. The computation of the potential of mean force has analogies with non-Boltzmann sampling methods. Modern techniques for enhanced sampling, such as metadynamics [128] and well-tempered metadynamics [18] can be applied to improve the statistical accuracy. This work is currently in progress.

8.3 Application of the displaced path integral formalism to water

The displaced path formalism is introduced in Section 8.2 under the single particle picture, and it can be readily generalized to distinguishable many particle systems.

Following the notation of Eq. (8.6), the end-to-end distribution of the l-th particle of a many body system takes the form under the displaced path formalism as

$$\widetilde{n}(\boldsymbol{x}) = \frac{1}{Z}\int_{\boldsymbol{R}(\beta\hbar)=\boldsymbol{R}(0)} \mathfrak{D}\boldsymbol{R}(\tau) e^{-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\, \frac{m\dot{\boldsymbol{R}}^2(\tau)}{2} + V[\boldsymbol{R}(\tau)+Y_l(\tau)\boldsymbol{x}]}. \tag{8.22}$$

with the compact notation

$$\boldsymbol{R}(\tau) = \big(\boldsymbol{r}_1(\tau), \ldots, \boldsymbol{r}_{l-1}(\tau), \boldsymbol{r}_l(\tau), \boldsymbol{r}_{l+1}(\tau), \ldots, \boldsymbol{r}_M(\tau)\big), \tag{8.23}$$

and $Y_l(\tau)$ is a M-dimensional function with the only nonzero entry at the l-th component as

$$Y_l(\tau) = \big(0, \ldots, y(\tau), \ldots, 0\big). \tag{8.24}$$

Again we find that displaced path formalism does not introduce any special particle among all the particles and the formulation is exact. We can sample the momentum distribution using all particles in the system, and thus can greatly enhance the sampling efficiency. The displaced path algorithm is tested with a flexible model for water [173]. The simulation box contains 32 water molecules. The temperature is set to be 296K. Both protons and oxygens are treated by quantum mechanics, and are represented by 64 classical beads. The end-to-end distribution takes the spherical averaged form in water. The quantum effect for protons in water at room temperature is relatively small [191], which allows us to use free energy perturbation (8.14) and compare the results with open path integral simulation [192]. In the latter case, in principle one proton path should be opened and all other paths should be closed as discussed in Section 8.1. However, the resulting statistics would be poor. In order to boost statistics one proton path per water molecule was opened, as it was found that this approximation leads to a negligible error in the momentum distribution due to the relatively weak interaction between protons belonging to different water

molecules [192]. The displaced path integral formulation allows one to compute the end-to-end distribution without opening any proton path, and therefore all the protons can be included in the calculation of the end-to-end distribution without any approximation. We show the end-to-end distribution calculated both from a 268 ps open path simulation and from a 12 ps displaced path simulation that utilizes the estimator given by Eq. (8.14) in Fig. 8.10 (a), and the comparison of the potential of mean force in Fig. 8.10 (b). In both simulations, the time step is 0.24 fs. Two consecutive steps contain highly correlated information, and the free energy perturbation estimator may be computed every 20 steps. Thus with only a small increase in computational overhead in comparison to an open path simulation of the same length, the displaced path formulation has a large gain in terms of sampling efficiency.

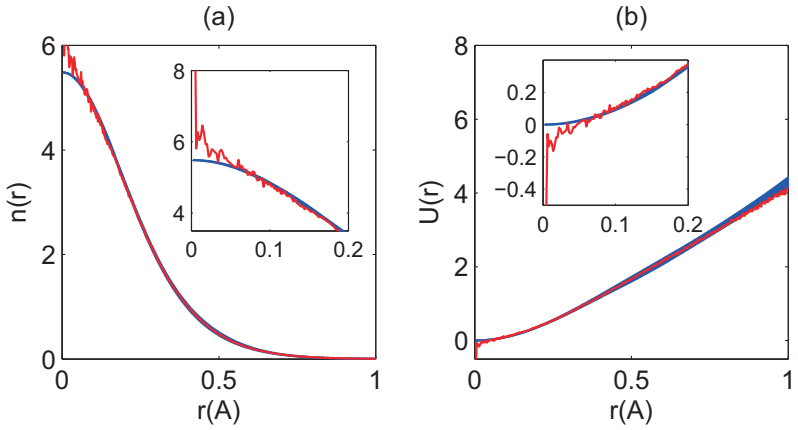


Figure 8.10: Comparison of (a) the end-to-end distribution and (b) the potential of mean force in SPC/F2 water. In both figures, the red line is computed by a 268ps open path integral simulation. The thick blue line is calculated using the displaced path estimator (8.14), with the thickness indicating the 95% confidence interval. The noise near $r = 0$ in both insets for open path simulation is due to the r^2 weight in the spherical integration, while the displaced path gives correct small r behavior by definition.

8.4 A new way of interpreting the momentum distribution

The thermodynamic integration approach given in Eq. (8.21) is not only computationally advantageous, but also provides one with the potential of mean force $U(\boldsymbol{x})$, and its gradient $\boldsymbol{F}(\boldsymbol{x})$. Both $U(\boldsymbol{x})$ and $\boldsymbol{F}(\boldsymbol{x})$ are key quantities for interpreting the physics underlying the momentum distribution $n(\boldsymbol{p})$. We first note that the kinetic energy K is given by $K = \frac{\hbar^2}{2m}\nabla \cdot \boldsymbol{F}(\boldsymbol{x})\Big|_{\boldsymbol{x}=\boldsymbol{0}} + \frac{3}{2\beta} \equiv K_V + \frac{3}{2\beta}$. Since $3/2\beta$ is the free particle contribution, the non-classical contribution is completely included in the excess kinetic energy term K_V, and is determined by the zero point curvature of $U(\boldsymbol{x})$. Secondly, if the momentum distribution of an individual particle is accessible (as is possible e.g. in simulations) and the underlying potential energy surface is harmonic, the end-to-end distribution follows a Gaussian distribution and the mean force is given by a straight line. Any deviation of $\hat{\boldsymbol{q}} \cdot \boldsymbol{F}(\boldsymbol{x})$ from linearity signals anharmonic behavior along the $\hat{\boldsymbol{q}}$ direction.

In experiments, the spherically averaged momentum distribution is accessible in liquids, and amorphous and polycrystalline solids, while the directional distribution is accessible in monocrystalline materials. The latter distribution provides more information about the underlying potential energy surface. However, in single crystals the total momentum distribution is the sum of the contributions of individual particles participating in bonds with different orientations. As a consequence the difference between directional and spherical momentum distribution is usually very small as shown in the top panel of Fig. 8.11. This figure is based on an anisotropic harmonic model with three distinct principal frequencies fitted from the *ab initio* path integral simulation for ice Ih [191]. The bottom panel of the same figure clearly shows that the distinction between the spherical and directional distributions is enhanced when comparing the mean forces. It is therefore of great interest to link directly the mean

force to the experimental data, *i.e.* to the Compton profile. The Compton profile is given by

$$J(\hat{\boldsymbol{q}}, y) = \int n(\boldsymbol{p})\delta(y - \boldsymbol{p}\cdot\hat{\boldsymbol{q}})d\boldsymbol{p}. \tag{8.25}$$

$\hat{\boldsymbol{q}}$ indicates the direction of the neutron detector [218].

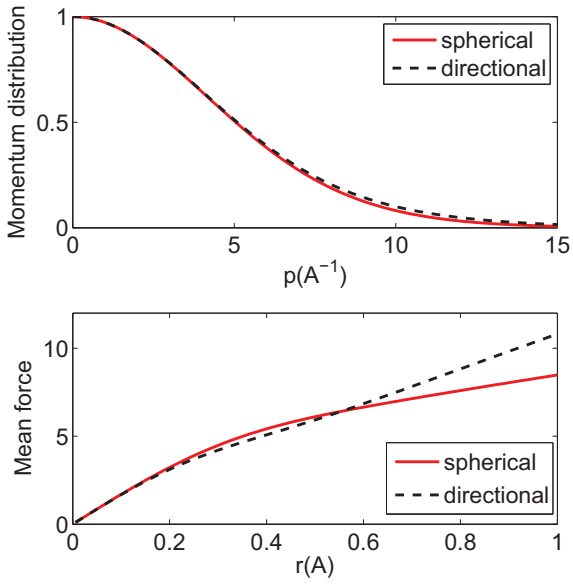


Figure 8.11: Top panel: the momentum distribution of the protons in ice Ih resulting from an anisotropic harmonic model (see text). Both the spherical and the directional distribution along the c-axis are shown. Bottom panel: the corresponding spherical and directional mean force projected along the c-axis. The curves are plotted as a function of the end-to-end distance. The mean force enhances the differences between spherical and directional distributions.

We define $x_\parallel = \boldsymbol{x}\cdot\hat{\boldsymbol{q}}$, and denote by $\boldsymbol{x}_\perp$ the $\boldsymbol{x}$ component orthogonal to $\hat{\boldsymbol{q}}$. Correspondingly $p_\parallel = \boldsymbol{p}\cdot\hat{\boldsymbol{q}}$, and $\boldsymbol{p}_\perp$ is the $\boldsymbol{p}$ component orthogonal to $\hat{\boldsymbol{q}}$. One has

$$\begin{aligned} J(\hat{\boldsymbol{q}}, y) &= \frac{1}{(2\pi\hbar)^3}\int d\boldsymbol{x}d\boldsymbol{p}\ \widetilde{n}(\boldsymbol{x})e^{\frac{i}{\hbar}\boldsymbol{p}\cdot\boldsymbol{x}}\delta(y - \boldsymbol{p}\cdot\hat{\boldsymbol{q}}) \\ &= \frac{1}{(2\pi\hbar)^3}\int dx_\parallel d\boldsymbol{x}_\perp dp_\parallel d\boldsymbol{p}_\perp\ \widetilde{n}(\boldsymbol{x})e^{\frac{i}{\hbar}x_\parallel p_\parallel + \frac{i}{\hbar}\boldsymbol{p}_\perp\cdot\boldsymbol{x}_\perp}\delta(y - p_\parallel) \\ &= \frac{1}{2\pi\hbar}\int dx_\parallel\ \widetilde{n}(x_\parallel\hat{\boldsymbol{q}})e^{\frac{i}{\hbar}x_\parallel y}. \end{aligned} \tag{8.26}$$

Given the end-to-end distribution can be expressed as

$$\widetilde{n}(\boldsymbol{x}) = e^{-\frac{m\boldsymbol{x}^2}{2\beta\hbar^2}} e^{-U(\boldsymbol{x})}, \tag{8.27}$$

the potential of mean force $U(\boldsymbol{x})$ can be obtained from the Compton profile as

$$U(x_\parallel \hat{\boldsymbol{q}}) = -\frac{m x_\parallel^2}{2\beta\hbar^2} - \ln \int dy\ J(\hat{\boldsymbol{q}}, y) e^{-\frac{i}{\hbar} x_\parallel y}. \tag{8.28}$$

The mean force $\boldsymbol{F}(\boldsymbol{x})$ is the gradient of $U(\boldsymbol{x})$. Taking into account that $J(\hat{\boldsymbol{q}}, y)$ is an even function of y one obtains

$$\hat{\boldsymbol{q}} \cdot \boldsymbol{F}(x_\parallel \hat{\boldsymbol{q}}) = -\frac{m x_\parallel}{\beta\hbar^2} + \frac{\int_0^\infty dy\ y \sin(x_\parallel y/\hbar) J(\hat{\boldsymbol{q}}, y)}{\hbar \int_0^\infty dy\ \cos(x_\parallel y/\hbar) J(\hat{\boldsymbol{q}}, y)}. \tag{8.29}$$

In the bottom panel of Fig. 8.11 the slope of the mean force, either spherical or directional, at $r = 0$ is equal to the excess kinetic energy K_V divided by the constant $\frac{\hbar^2}{2m}$. This is an exact result that originates from the symmetry property of ice Ih. In general the spherical and directional mean force can have different slopes at $r = 0$. The deviation of the spherical and directional forces from linearity at finite r results from the averaging process and is not a sign of anharmonicity. Thus in the interpretation of the experimental Compton profile, which results from the contribution of many particles, one must distinguish the case of an anisotropic harmonic potential energy surface from that of an anharmonic potential energy surface. To the best of our knowledge the procedure that is currently adopted to fit the experimental data [5, 208, 218] does not separate well anisotropic and anharmonic effects. We propose here an alternative approach in which the mean force is associated to the experimental Compton profile according to Eq. (8.29). The projections of the mean force along different directions are then fitted to an anisotropic harmonic model averaged as required by the crystal symmetry. Any systematic deviation from experiment

of the mean force originating from the harmonic contribution, can then be associated to anharmonicity and used to further refine the underlying model potential energy surface.

The framework introduced here may also be utilized to provide insight to the investigation of anharmonic systems. Consider for example a particle with the proton mass subject to a model double well 1D-potential. $V = \frac{m\omega^2}{2}z^2 + A\exp(-\frac{z^2}{2\xi^2})$ with $\omega = 1578$K, and $\xi = 0.094$Å. A characterizes the barrier height and is set to be $1263\text{K}, 3789\text{K}$, and 6315K, respectively. These parameters mimic different tunneling regimes for protons along a hydrogen bond [27, 190]. The temperature is set to be 30K. At this temperature the behavior of the systems is dominated by the ground-state, and the end-to-end distribution can be approximated by the overlap integral $\widetilde{n}(x) = \int dz\psi(z)\psi(z+x)$ where $\psi(z)$ is the ground-state wavefunction and $F(x) = -\frac{d}{dx}\ln\widetilde{n}(x)$. In Fig. 8.12 we can see how qualitatively different the mean force can be in the three cases. One goes from a fully monotonic behavior for $A = 1263$K which is a model for a low energy barrier hydrogen bond [26], to the strongly non monotonic mean forces for $A = 3789$K and $A = 6315$K where the tunneling states lie below the barrier height. Additionally, it is not very difficult to relate features of the mean force to the underlying effective potential.

It is also instructive to study $F(x)$ as a function of temperature when the higher states are mixed in the density matrix. This is done in Fig. 8.13 for the double well potential with $A = 3789$K. For temperatures in the $100 - 500$K range, the behavior is dominated by the two lowest eigenstates. The slope of $F(x)$ at small x, which is proportional to the excess kinetic energy K_V, shows little dependence on T. It can be shown with detailed analysis that this is a generic feature of two level tunneling systems. Other characters seen in Fig. 8.13 in the same range of temperatures, such as the more pronounced kink at intermediate x and the enhanced softening of the mean force at large x, derive from the odd symmetry of the first excited state contribution.

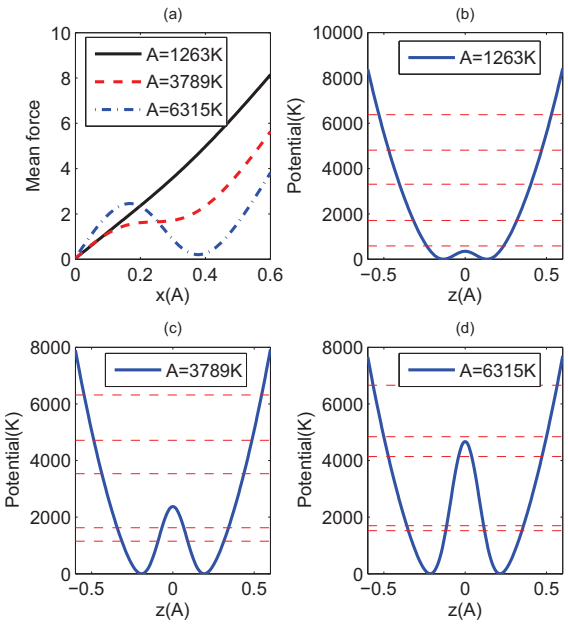


Figure 8.12: (a) The mean force corresponding to a double well model at $T = 30$K, for different barrier heights $A = 1263$K (black solid line), $A = 3789$K (red dashed line), and $A = 6315$K (blue dot-dashed line). (b) Potential energy surface for $A = 1263$K (blue solid line), and the first five energy levels (red dashed line). (c) (d) the same as (b), but with $A = 3789$K and $A = 6315$K respectively.

Eventually at higher T the kink in $F(x)$ disappears as the mean force progressively resumes linear behavior with a slope that tends to zero as high temperature classical limit is reached.

8.5 Semiclassical limit of displaced path integral formalism

So far the displaced path integral formalism can be applied to compute the momentum distribution of any quantum system as long as the exchange effect among the particles can be neglected. Furthermore, if the quantum effect is not strong and the quantum path only perturbs from its centroid by a small magnitude, the computational procedure can be further simplified. This is known as the semiclassical limit

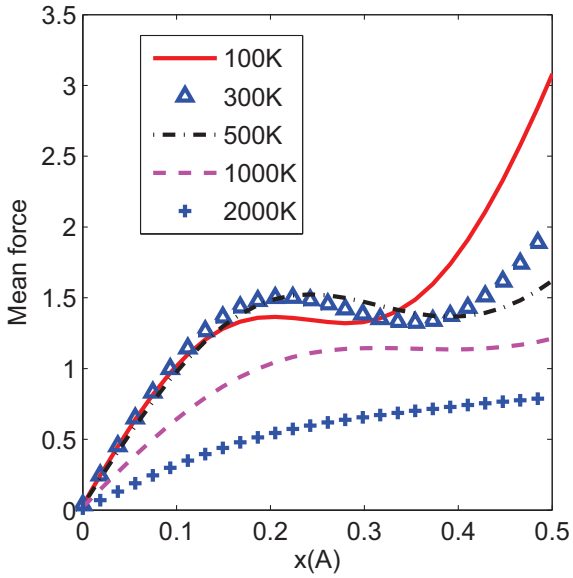


Figure 8.13: The mean force corresponding to a double well model at $A = 3789$K for different temperatures 100K (red solid line), 300K (blue triangle), 500K (black dot-dashed line), 1000K (magenta dashed line), and 2000K (blue cross).

of the path integral theory. The semiclassical limit has been well studied in terms of closed path integrals and position distribution [83]. However, little work has been done in the context of momentum distribution. One of the known result is that if the quantum effect is weak, the quantum effect on the momentum distribution can be described as an effective increase of temperature [21, 147]:

$$T_{\text{eff}} = T + \frac{\beta^2\hbar^2}{12mk_B}\langle (V'(\bar{x}))^2\rangle_{\text{cl}}. \tag{8.30}$$

Here $\langle\cdot\rangle_{\text{cl}}$ means the average is taken over the classical trajectory. Eq. (8.30) essentially approximates the quantum momentum distribution by an isotropic Gaussian distribution. The anisotropic effect is mixed and represented by a single parameter T_{eff}, leaving aside the anharmonic effect. In this section we derive a more accurate semiclassical limit for momentum distribution, which naturally generalizes the estimate by Feynman and Hibbs [83] to the case with the momentum distribution.

To facilitate the discussion, we first briefly review the derivation of the semiclassi-

cal limit for the closed path integral formalism. The semiclassical limit considers the case when $\beta\hbar$ is small, namely the system is at high temperature. Then each path $r(\tau)$ can be considered to be centered around its centroid

$$\bar{r} = \frac{1}{\beta\hbar}\int_0^{\beta\hbar} r(\tau)d\tau. \tag{8.31}$$

The partition function Z can be written as

$$Z = \int d\bar{r}\int dr_1 \int_{r_1}^{r_1} \widetilde{\mathfrak{D}}r(\tau)\exp\Big(-\frac{1}{\hbar}\int_0^{\beta\hbar}\frac{m\dot{r}^2}{2} + V(r)d\tau\Big). \tag{8.32}$$

Here the notation $\int d\bar{r}\int dr_1\int_{r_1}^{r_1}\widetilde{\mathfrak{D}}r(\tau)$ is another way of writing the unconstrained path integral $\int \mathfrak{D}r(\tau)$, except that the inside integration satisfies the constraint with fixed centroid

$$\frac{1}{\beta\hbar}\int_0^{\beta\hbar} r(\tau)d\tau = \bar{r}, \quad r(0) = r(\beta\hbar) = r_1. \tag{8.33}$$

In the semiclassical analysis, the internal degrees of freedom r_1 should be integrated out, and the partition function is only determined by the centroid $\bar{r}$. To this end we rewrite the partition function as

$$Z = \int d\bar{r}\left(\int dr_1\int_{r_1}^{r_1}\widetilde{\mathfrak{D}}r(\tau)P[r(\tau);\bar{r}]e^f\right)\cdot W(\bar{r}). \tag{8.34}$$

Here

$$f = -\frac{1}{\hbar}\int_0^{\beta\hbar} V[r(\tau)]d\tau. \tag{8.35}$$

$P[r(\tau),\bar{r}]$ is the probability measure corresponding to free particles

$$P[r(\tau),\bar{r}] = \frac{\exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar}\frac{m\dot{r}^2(\tau)}{2}d\tau\right)}{W(\bar{r})} \tag{8.36}$$

with an normalizing factor

$$W(\bar{r}) = \int dr_1 \int_{r_1}^{r_1} \widetilde{\mathfrak{D}} r(\tau) \exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} \frac{m\dot{r}^2(\tau)}{2} d\tau\right). \tag{8.37}$$

In order to integrate out the internal degrees of freedom, we apply the Jensen's inequality (*i.e.* the variational principle)

$$\langle e^f \rangle \geq e^{\langle f \rangle}, \tag{8.38}$$

with the average $\langle \cdot \rangle$ defined by

$$\langle A \rangle \equiv \int dr_1 \int_{r_1}^{r_1} \widetilde{\mathfrak{D}} r(\tau) P[r(\tau); \bar{r}] \cdot A. \tag{8.39}$$

The question remains to evaluate the average $\langle f \rangle$.

For fixed position of the centroid $\bar{r}$, we have

$$\begin{aligned} \langle f \rangle &= -\frac{1}{\hbar} \int dr_1 \int_{r_1}^{r_1} \widetilde{\mathfrak{D}} r(\tau) P[r(\tau); \bar{r}] \int_0^{\beta\hbar} dt\ V[r(t)] \\ &= -\frac{1}{\hbar} \int_0^{\beta\hbar} dt \int dr_1 \int_{r_1}^{r_1} \widetilde{\mathfrak{D}} r(\tau) P[r(\tau); \bar{r}] V[r(t)] \\ &\equiv -\frac{1}{\hbar} \int_0^{\beta\hbar} dt\ I(t; \bar{r}). \end{aligned} \tag{8.40}$$

Since each path can be seen as a $\beta\hbar$-length segment of a periodic path, the two sets $\{r(0)\}$ and $\{r(t)\}$ are exactly the same. Therefore $I(t; \bar{r})$ is independent of t, and we have

$$I(\bar{r}) \equiv I(0; \bar{r}) = \int dr_1 \int_{r_1}^{r_1} \widetilde{\mathfrak{D}} r(\tau) P[r(\tau); \bar{r}] V(r_1), \quad \langle f(\bar{r}) \rangle = -\beta I(\bar{r}). \tag{8.41}$$

$I(\bar{r})$ can be written as

$$I(\bar{r}) = \int dY \int_Y^Y \widetilde{\mathfrak{D}} z(\tau) P[z(\tau);\bar{r}] V[\bar{r}+Y] \tag{8.42}$$

with $z(\tau) = r(\tau) - \bar{r}, Y = r_1 - \bar{r}$, and we have used $P[r(\tau);\bar{r}] = P[z(\tau);\bar{r}]$. The new path $z(\tau)$ satisfies the constraint $\int_0^{\beta\hbar} d\tau z(\tau) = 0$. This constraint is linear in z, and it is convenient to rewrite the constraint using the Dirac-δ function as $\delta(\int_0^{\beta\hbar} z d\tau)$. This δ-function can be eliminated in the integral by means of its Fourier transform

$$\delta\left(\int_0^{\beta\hbar} z d\tau\right) = \int_{-\infty}^{\infty} dk \frac{\beta}{2\pi} \exp\left(ik \int_0^{\beta\hbar} z(\tau) d\tau\right), \tag{8.43}$$

and

$$\begin{aligned} W(\bar{r}) &= \int dk \frac{\beta}{2\pi} dr_1 \int_{r_1-\bar{r}}^{r_1-\bar{r}} \mathfrak{D} z(\tau) \exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} \frac{m\dot{z}^2(\tau) - ikz}{2} d\tau\right) \\ &= \frac{\beta}{2\pi} \int dr_1 \int dk \sqrt{\frac{m}{2\pi\beta\hbar^2}} \exp\left\{\frac{i}{\hbar}\left[\frac{1}{2}(ik)(-i\beta\hbar)2r_1 - \frac{(ik)^2(-i\beta\hbar)^3)}{24m}\right]\right\} \\ &= \frac{\beta}{2\pi}\sqrt{\frac{m}{2\pi\beta\hbar^2}} \int dr_1 e^{-6mr_1^2/\beta\hbar^2} \sqrt{\frac{24m\pi}{\beta^3\hbar^2}} \\ &= \sqrt{\frac{m}{2\pi\beta\hbar^2}}. \end{aligned} \tag{8.44}$$

Therefore $W(\bar{r})$ is independent of $\bar{r}$. This fact can be readily seen since $W(\bar{r})$ is the normalization factor for free particle system which is translational invariant. Similarly

$$I(\bar{r}) = \sqrt{\frac{6m}{\pi\beta\hbar^2}} \int dY \int_{-\infty}^{\infty} V(\bar{r}+Y) e^{-6Y^2 m/\beta\hbar^2}, \tag{8.45}$$

In summary, the partition function is expressed under the semiclassical limit as

$$Z \approx \sqrt{\frac{m}{2\pi\beta\hbar^2}} \int d\bar{r} e^{-\beta U(\bar{r})}, \tag{8.46}$$

with the effective classical potential as

$$U(\bar{r}) = \sqrt{\frac{6m}{\pi\beta\hbar^2}} \int_{-\infty}^{\infty} dY\, V(\bar{r}+Y) e^{-6Y^2 m/\beta\hbar^2}. \tag{8.47}$$

We see that the internal degrees of freedom of the quantum path have been integrated out, and the partition function is represented in terms of an effective classical potential acting on the centroid of the path. Eq. (8.47) indicates that in the semiclassical limit, the quantum effect mimics a smearing Gaussian kernel with mean-square spread $(\beta\hbar^2/12m)^{1/2}$.

The displaced path formalism evaluates the end-to-end distribution as an estimator based on closed path integrals. Therefore the same derivation above can be applied to study the semi-classical limit of the end-to-end distribution, and therefore the momentum distribution. The denominator of the end-to-end distribution is the partition function Z already calculated in Eq. (8.46). The numerator of the end-to-end distribution is

$$\begin{aligned}
\mathrm{Tr}[e^{-\frac{i}{\hbar}px} e^{-\beta H}] &= \exp\left(-\frac{mx^2}{2\beta\hbar^2}\right) \int d\bar{r} \int dr_1 \int_{r_1}^{r_1} \widetilde{\mathfrak{D}} r(\tau) \\
&\quad \exp\left\{-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \left(\frac{m}{2}\dot{r}^2(\tau) + V[r(\tau)+y(\tau)x]\right)\right\} \\
&= \exp\left(-\frac{mx^2}{2\beta\hbar^2}\right) \int d\bar{r} \int dr_1 \int_{r_1-\bar{r}}^{r_1-\bar{r}} \widetilde{\mathfrak{D}} z(\tau) \\
&\quad \exp\left\{-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \left(\frac{m}{2}\dot{z}^2(\tau) + V[\bar{r}+z(\tau)+y(\tau)x]\right)\right\}
\end{aligned} \tag{8.48}$$

with $y(\tau), z(\tau)$ defined the same as before.

Applying again the variational principle

$$\begin{aligned}
\mathrm{Tr}[e^{-\frac{i}{\hbar}px} e^{-\beta H}] \approx &\exp\left(-\frac{mx^2}{2\beta\hbar^2}\right) \int d\bar{r} \Big(\exp\Big\{ -\frac{1}{\hbar} \int dr_1 \int_{r_1-\bar{r}}^{r_1-\bar{r}} \widetilde{\mathfrak{D}} z(\tau) \\
&P[z(\tau);\bar{r}] \int_0^{\beta\hbar} dt V[z(t)+\bar{r}+y(t)x] \Big\}\Big) \cdot W(\bar{r})
\end{aligned} \tag{8.49}$$

and write $\langle f(\bar{r})\rangle$ as

$$\begin{aligned}\langle f(\bar{r})\rangle &= -\frac{1}{\hbar}\int dr_1 \int_{r_1-\bar{r}}^{r_1-\bar{r}} \widetilde{\mathfrak{D}} z(\tau) P[z(\tau);\bar{r}] \int_0^{\beta\hbar} dt\, V[z(t)+y(t)x+\bar{r}] \\ &= -\frac{1}{\hbar}\int_0^{\beta\hbar} dt \int dr_1 \int_{r_1-\bar{r}}^{r_1-\bar{r}} \widetilde{\mathfrak{D}} z(\tau) P[z(\tau);\bar{r}] V[z(t)+y(t)x+\bar{r}] \\ &\equiv -\frac{1}{\hbar}\int_0^{\beta\hbar} dt\; I(t;\bar{r}).\end{aligned} \tag{8.50}$$

Using the same technique in Eq. (8.43), we find that

$$\langle f(\bar{r})\rangle = -\frac{1}{\hbar}\int_0^{\beta\hbar} dt \int dY e^{-6mY^2/\beta\hbar^2} V(Y+y(t)x+\bar{r}) \tag{8.51}$$

Therefore the numerator of the end-to-end distribution is

$$\mathrm{Tr}[e^{-\frac{i}{\hbar}px} e^{-\beta H}] = e^{-\frac{mx^2}{2\beta\hbar^2}} \sqrt{\frac{m}{2\pi\beta\hbar^2}} \int d\bar{r} e^{-\beta U(\bar{r};x)} \tag{8.52}$$

with the effective classical potential $U(\bar{r};x)$ as

$$U(\bar{r};x) = \frac{1}{\beta\hbar}\int_0^{\beta\hbar} dt \int dY e^{-6mY^2/\beta\hbar^2} V(Y+y(t)x+\bar{r}). \tag{8.53}$$

The physical meaning of this effective potential is clear. It replaces to the closed path effective potential $U(\bar{r})$ by an average on the displaced path

$$U(\bar{r};x) = \frac{1}{\beta\hbar}\int_0^{\beta\hbar} dt\, U(\bar{r}+y(t)x). \tag{8.54}$$

To sum up, the end-to-end distribution can be rewritten in the semi-classical limit as

$$\widetilde{n}(x) \approx e^{-\frac{mx^2}{2\beta\hbar^2}} \frac{\int d\bar{r} e^{-\beta U(\bar{r};x)}}{\int d\bar{r} e^{-\beta U(\bar{r})}}, \tag{8.55}$$

and the displaced path estimator for the end-to-end distribution under the semiclas-

sical limit is

$$\mathcal{N}(x;0) \approx \exp\left\{-\beta[U(\bar{r};x) - U(\bar{r})]\right\}. \tag{8.56}$$

We use the double well model (8.15) again to illustrate the accuracy of the new estimator of the end-to-end distribution under semiclassical limit (8.56). The end-to-end distribution at 800K and 300K are shown in Fig. 8.14 and 8.15, respectively. The difference between the quantum and the classical end-to-end distribution is already visible at 800K. The free energy perturbation estimator (8.56) accurately reproduces the end-to-end distribution. On the other hand, the isotropic estimator (8.30) overshoots the quantum effect, and the deviation from the exact result is even larger than the deviation obtained from the classical Maxwell-Boltzmann distribution.

The performance of the semiclassical estimator (8.56) at 300K is more interesting. It has been reported that the double well model at 300K mimics the quantum effect of ice VII with proton tunneling [190]. The end-to-end distribution in this case strongly deviates from a Gaussian distribution. However, the semiclassical estimator (8.56) still gives a rather accurate description of the non-Gaussian end-to-end distribution. This example clearly indicates the new semiclassical estimator can be applied to study the quantum momentum distribution for a large class of systems.

Besides the conceptual advantage, the semiclassical estimator (8.56) has computational advantage when combined with the free energy perturbation method. Eq. (8.54) indicates that if the effective potential for the closed path integral simulation can be obtained efficiently, then $U(\bar{r}, x)$ is also readily obtained by means of Gauss-quadrature along the t direction with a small number of integration points. The number of integration points can be much less than the number of replicas P required in the standard path integral simulation. This is our work in progress.

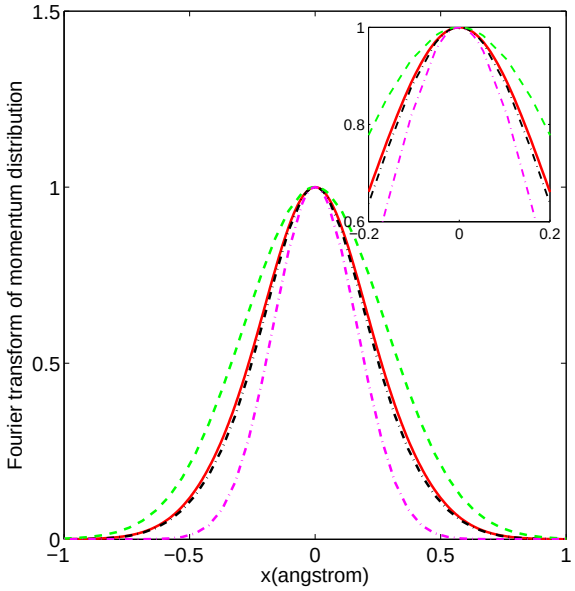


Figure 8.14: The end-to-end distribution corresponding to the double well potential at 800K. Red solid line: the exact result. Black dot dashed line: the result from the new semiclassical estimator 8.56. Magenta dot dashed line: the result from the isotropic estimator 8.30. Green dashed line: classical Maxwell-Boltzmann distribution.

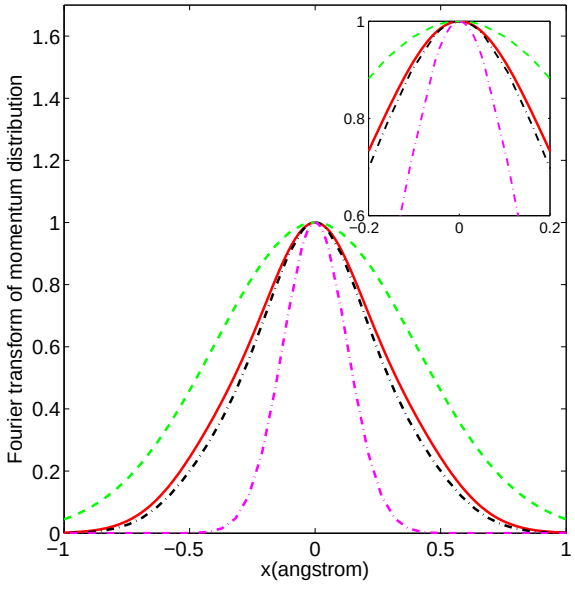


Figure 8.15: The end-to-end distribution corresponding to the double well potential at 300K. Red solid line: the exact result. Black dot dashed line: the result from the semiclassical estimator 8.56. Magenta dot dashed line: the result from the isotropic estimator 8.30. Green dashed line: classical Maxwell-Boltzmann distribution.

8.6 A new kinetic estimator

We have shown that the displaced path integral formalism is more advantageous compared to the open path integral method for calculating the momentum distribution. In this section we show that the displaced path integral formalism also provides an alternative way of calculating the kinetic energy of quantum particles.

In what follows the new estimator is introduced under single particle picture at one dimension. The generalization to the many particle case is straightforward. The kinetic energy is directly related to the curvature of the end-to-end distribution at $x = 0$ as

$$\langle K \rangle = \frac{1}{Z} \mathrm{Tr} \left[\frac{p^2}{2m} e^{-\beta H} \right] = -\frac{1}{Z} \frac{\hbar^2}{2m} \frac{d^2}{dx^2} \mathrm{Tr} \left[e^{-\frac{i}{\hbar} p x e^{-\beta H}} \right] \Big|_{x=0}. \tag{8.57}$$

The curvature of the end-to-end distribution can be represented using the displaced path integral formulation, which reads

$$\begin{aligned}
\frac{d^2}{dx^2}\widetilde{n}(x) =& \frac{1}{Z}\frac{d^2}{dx^2}\Big\{ \int \mathfrak{D}r(\tau) \exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \ \frac{m\dot{r}^2}{2} + V[r(\tau) + y(\tau)x] \right) \cdot \exp\left(\frac{-mx^2}{2\beta\hbar^2} \right) \Big\} \Big|_{x=0} \\
=& \frac{1}{Z}\frac{d}{dx}\Big\{ \int \mathfrak{D}r(\tau) \exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \ \frac{m\dot{r}^2}{2} + V[r(\tau) + y(\tau)x] \right) \cdot \exp\left(\frac{-mx^2}{2\beta\hbar^2} \right) \cdot \\
& \left[-\frac{mx}{\beta\hbar^2} - \frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \ V'[r(\tau) + y(\tau)x] y(\tau) \right] \Big\} \Big|_{x=0} \\
=& \frac{1}{Z} \int \mathfrak{D}r(\tau) \exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \ \frac{m\dot{r}^2}{2} + V[r(\tau)] \right) \cdot \\
& \left\{ -\frac{m}{\beta\hbar^2} + \left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \ V'[r(\tau)] y(\tau) \right)^2 - \frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \ V''[r(\tau)] y^2(\tau) \right\}
\end{aligned} \tag{8.58}$$

Eq. (8.58) involves the second derivative of the potential energy which is difficult to compute in practice. We would like to represent the kinetic energy estimator only using the first order derivative of the potential energy, which is the force on the atoms and is available in any molecular dynamics simulation and in most of the Monte Carlo simulations.

We define the average of a quantity A as

$$\langle A[r(\tau)]\rangle = \frac{1}{Z}\int \mathfrak{D}r(\tau)\exp\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\ \frac{m\dot{r}^2}{2} + V[r(\tau)]\right) A[r(\tau)] \tag{8.59}$$

then Eq. (8.58) can be rewritten as

$$\frac{d^2}{dx^2}\widetilde{n}(x) = -\frac{m}{\beta\hbar^2} + \langle\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\ V'[r(\tau)]y(\tau)\right)^2\rangle + \langle -\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\ V''[r(\tau)]y^2(\tau)\rangle \tag{8.60}$$

The first term of (8.60) is a constant coming from the free particle contribution, or the classical Maxwell-Boltzmann distribution. The second term can be rewritten as

$$\langle\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\ V'[r(\tau)]y(\tau)\right)^2\rangle = \frac{1}{\hbar^2}\langle\int_0^{\beta\hbar} d\tau\ \int_0^{\beta\hbar} dt V'[r(t)]V'[r(\tau)]y(\tau)y(t)\rangle \tag{8.61}$$

To further simplify the formulation, we note that we have used the convention that we open the path at $r(0)$. However, the path $r(\tau)$ can be "opened" at any imaginary time slice $r(s)$. Taking this symmetry into account, Eq. (8.61) equals to

$$\frac{1}{\beta\hbar^3}\langle\int_0^{\beta\hbar} ds\int_0^{\beta\hbar} d\tau\ \int_0^{\beta\hbar} dt V'[r(t+s)]V'[r(\tau+s)]y(\tau)y(t)\rangle \tag{8.62}$$

Let us define a correlation function between forces at imaginary time t and τ as

$$V^{(1)}_{\rm corr}(t-\tau) \equiv \frac{1}{\beta\hbar}\int_0^{\beta\hbar} ds\ V'[r(t+s)]V'[r(\tau+s)]. \tag{8.63}$$

$V^{(1)}_{\rm corr}(t-\tau)$ only depends on the difference along the imaginary time axis $u = t-\tau$. With this notation Eq. (8.62) becomes

$$\frac{1}{\hbar^2}\langle\int_0^{\beta\hbar} d\tau\int_0^{\beta\hbar} dt\ V^{(1)}_{\rm corr}(t-\tau)y(t)y(\tau)\rangle \tag{8.64}$$

After some further simplification, the second term of (8.60) becomes

$$\frac{\beta}{2\hbar}\langle\int_0^{\beta\hbar} du\; V_{\rm corr}^{(1)}(u)\left(y^2(u)-\frac{1}{12}\right)\rangle \tag{8.65}$$

The second derivative in the third term of Eq. (8.60) can be eliminated using the symmetry along the imaginary time axis as well. The kinetic energy estimator is invariant when substituting $y(\tau)x$ by $y(\tau)x+f(x)$, with $f(x)$ an arbitrary function of x. Therefore

$$\begin{aligned}\frac{d^2}{dx^2}\widetilde{n}(x) =& -\frac{m}{\beta\hbar^2}+\langle\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\; V'[r(\tau)]y(\tau)\right)^2\rangle+\langle-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\; V''[r(\tau)]y^2(\tau)\rangle\\ =& -\frac{m}{\beta\hbar^2}+\langle\left(-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\; V'[r(\tau)](y(\tau)+f'(0))\right)^2\rangle+\\ & \langle-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\; V''[r(\tau)](y(\tau)+f'(0))^2\rangle+\langle-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\; V'[r(\tau)]f''(0)\rangle.\end{aligned} \tag{8.66}$$

Here we assume $f'(0)$ and $f''(0)$ are arbitrary, but $f(0)=0$. All terms containing $f'(0)$ and $f''(0)$ must vanish, and we have

$$\langle\left[-\frac{1}{\hbar}d\tau\int_0^{\beta\hbar} V'[r(\tau)]\right]^2\rangle=\langle\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\; V''[r(\tau)]\rangle, \tag{8.67a}$$

$$\langle\left[-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\; V'[r(\tau)]y(\tau)\right]\cdot\left[-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\; V'[r(\tau)]\right]\rangle=\langle\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\; V''[r(\tau)]y(\tau)\rangle, \tag{8.67b}$$

$$\langle\int_0^{\beta\hbar} d\tau\; V'[r(\tau)]\rangle=0. \tag{8.67c}$$

We only need Eq. (8.67a) in order to simplify the third term in Eq. (8.58) as

follows

$$\begin{aligned}\langle\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\ V''[r(\tau)]y^2(\tau)\rangle &= \langle\frac{1}{\beta\hbar^2}\int_0^{\beta\hbar} ds\ \int_0^{\beta\hbar} d\tau\ V''(r(\tau+s))y^2(\tau)\rangle \\ &= \frac{1}{12\hbar}\langle\int_0^{\beta\hbar} d\tau\ V''(r(s))\rangle \\ &= \frac{1}{12}\langle\left[-\frac{1}{\hbar}\int_0^{\beta\hbar} V'[r(\tau)]\right]^2\rangle = \frac{\beta}{12\hbar}\langle\int_0^{\beta\hbar} du\ V_{\rm corr}^{(1)}(u)\rangle.\end{aligned} \tag{8.68}$$

Combining the results of (8.65) and (8.68), the kinetic energy finally takes a very simple form

$$\langle K\rangle = \frac{1}{2\beta} + \frac{\beta^2\hbar^2}{4m}\langle\frac{1}{\beta\hbar}\int_0^{\beta\hbar} du\ V_{\rm corr}^{(1)}(u)\left[\frac{u}{\beta\hbar}\left(1-\frac{u}{\beta\hbar}\right)\right]\rangle \tag{8.69}$$

The new kinetic energy estimator (8.69) shares similar properties with the well-known virial estimator [124]: The classical contribution and quantum contribution to the kinetic energy are naturally separated from each other, and the quantum part has a well defined continuous limit. Both methods only require the force along the trajectory and the computational costs are the same. Moreover, the new estimator (8.69) reveals that the quantum contribution only comes from the correlation of the forces at differential imaginary time slices. The weight $\frac{u}{\beta\hbar}\left(1-\frac{u}{\beta\hbar}\right)$ implies that the "self correlation" of the force $V_{\rm corr}^{(1)}(0) = V_{\rm corr}^{(1)}(\beta\hbar)$ does not contribute to quantum effect. The correlation function at other imaginary time slices contribute to the kinetic energy according to a positive parabolic weight function $\frac{u}{\beta\hbar}\left(1-\frac{u}{\beta\hbar}\right)$.

The performance of the new kinetic energy estimator (8.69) is compared with the virial estimator using three examples: harmonic oscillator at 300K; double well potential at 300K; double well potential at 100K.

For harmonic oscillator at 300K, the new kinetic energy estimator and the virial estimator are compared in Fig. 8.16. The average value of the kinetic energy estimated

by the new estimator is $(1.2819 \pm 0.0815) \times 10^{-3}$, and the variance is 1.610×10^{-3}. The average value of the kinetic energy estimated by the virial estimator is $(1.2632 \pm 0.0219) \times 10^{-3}$, and the variance is 4.318×10^{-4}. The exact kinetic energy is 1.2629×10^{-3}. The correlation of forces along imaginary axis in Fig. 8.17 which has a parabolic shape.

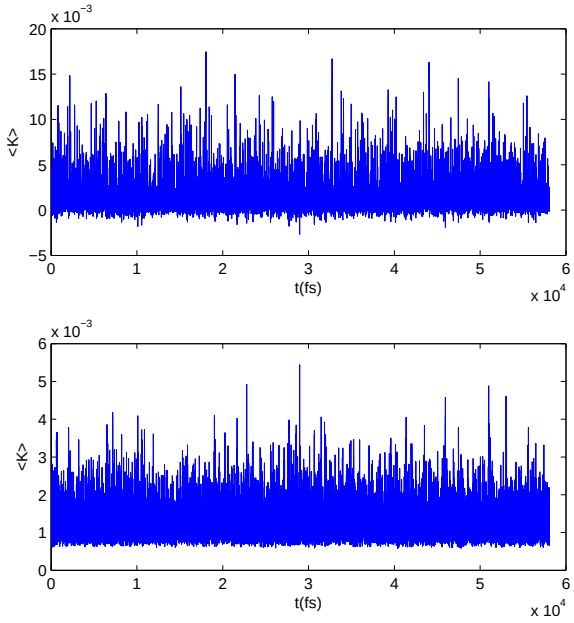


Figure 8.16: Comparison of the kinetic energy estimator based on displaced path formula (upper panel) and virial estimator (lower panel) for the harmonic potential at 300K.

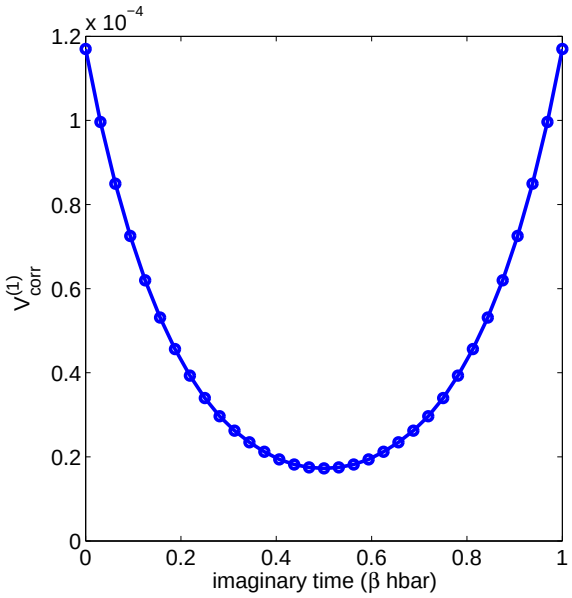


Figure 8.17: The correlation function $V^{(1)}_{\rm corr}(u)$ along the imaginary time axis for the harmonic potential at 300K.

The same comparison is performed for the double well potential at 300K (Fig. 8.18). The average value of the kinetic energy of the new estimator is $(1.3350\pm0.1036)\times10^{-3}$, and the variance is 2.895×10^{-3}. For the virial estimator, the average value is $(1.3281 \pm 0.0467) \times 10^{-3}$, and the variance is 9.225×10^{-4}. The exact kinetic energy is 1.3439×10^{-3}. The correlation of forces along imaginary axis in Fig. 8.19.

Finally we test the double well potential at 100K (Fig. 8.20). The average value of kinetic energy estimated by the new estimator is $(1.0055 \pm 0.4116) \times 10^{-3}$, and the variance is 8.135×10^{-3}. For the virial estimator, the average value is $(0.9974 \pm 0.0594)\times10^{-3}$, and the variance is 8.2938×10^{-4}. The exact kinetic energy is 1.0234×10^{-3}. The correlation of forces along imaginary axis in Fig. 8.21.

From the three examples above, we find that the new kinetic energy estimator is an unbiased method for sampling the kinetic energy of quantum particles. However, the variance of the new estimator is in general larger than that in the virial estimator.

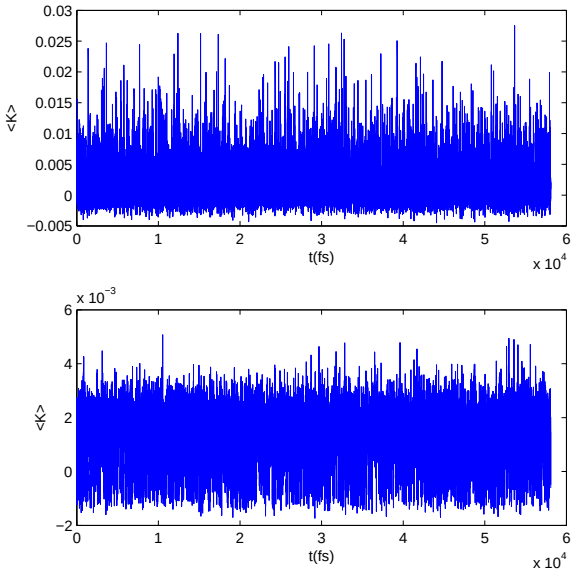


Figure 8.18: Comparison of the kinetic energy estimator based on displaced path formula (upper panel) and virial estimator (lower panel) for the double well at 300K.

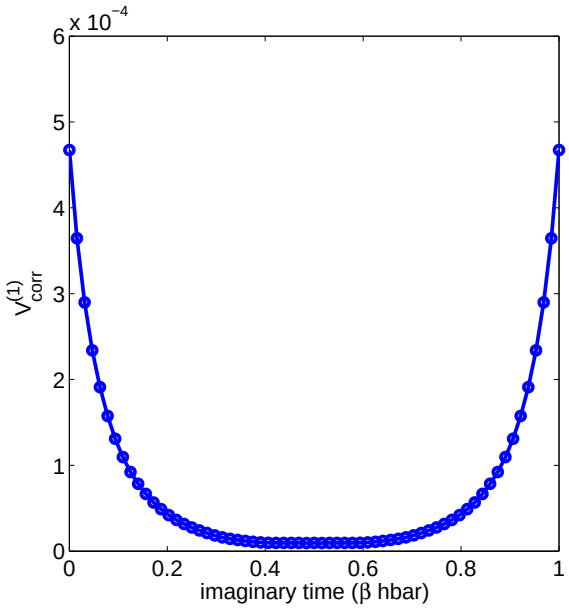


Figure 8.19: The correlation function $V_{\rm corr}^{(1)}(u)$ along the imaginary time axis for the double well potential at 300K.

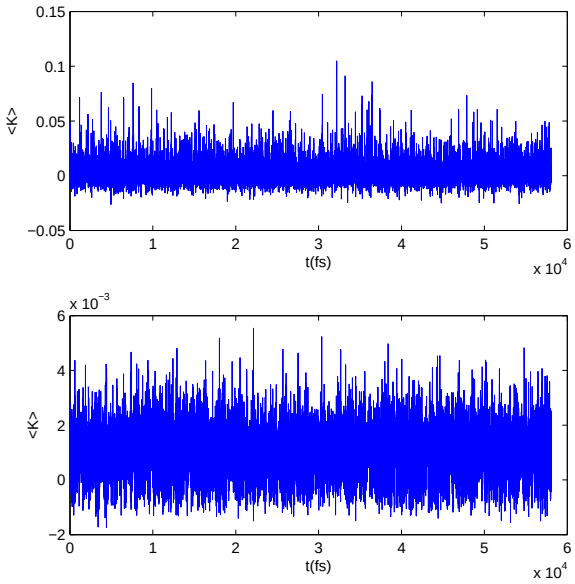


Figure 8.20: Comparison of the kinetic energy estimator based on displaced path formula (upper panel) and virial estimator (lower panel) for the double well at 100K.

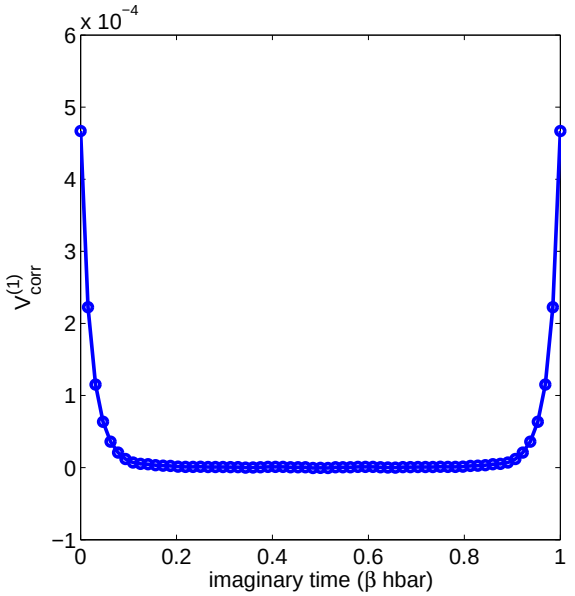


Figure 8.21: The correlation function $V_{\rm corr}^{(1)}(u)$ along the imaginary time axis for the double well potential at 100K.

8.7 Displaced path formalism for bosons

So far we have established the displaced path formalism for distinguishable quantum particles. The displaced path integral formalism has various advantages from both the conceptual and the computational point of view. In this section the displaced path formalism is generalized to bosons. The same formulation holds for fermions as well, but the resulting estimator will suffer from the sign problem and the statistical sampling becomes impractical. For an system consisting of N bosons, we write the collective variable $\boldsymbol{R}$ to denote the position of the particles $(\boldsymbol{r}_1, \cdots, \boldsymbol{r}_N)$. The density matrix of bosons is

$$\langle \boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_N | \rho_B | \boldsymbol{r}_1 + \boldsymbol{x}, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_N \rangle \equiv \langle \boldsymbol{R} | \rho_B | \boldsymbol{R} + X \rangle \tag{8.70}$$

Here the collective variable $X = (\boldsymbol{x}, 0, \cdots, 0)$.

The one particle density matrix for bosons ρ_B is defined by symmetrizing the one particle density matrix for distinguishable particles as [51]

$$\begin{aligned} \langle \boldsymbol{R} | \rho_B | \boldsymbol{R} + X \rangle =& \frac{1}{N!} \sum_T \langle \boldsymbol{R} | \rho | T[\boldsymbol{R} + X] \rangle \\ =& \frac{1}{N!} \sum_T \int_{\boldsymbol{R}(0)=\boldsymbol{R}, \boldsymbol{R}(\beta\hbar)=T[\boldsymbol{R}+X]} \mathfrak{D}\boldsymbol{R}(\tau) e^{-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau \, \frac{m\dot{\boldsymbol{R}}^2(\tau)}{2} + V[\boldsymbol{R}(\tau)]}. \end{aligned} \tag{8.71}$$

Here T represents any possible $N-$ permutation. The key step of generalizing displaced path formulation is that for each T, we separate the orbit of the particle 1 under the permutation T from the other degrees of freedom. The orbit of the particle 1 under the permutation is defined to be the collection of particles

$$A = \{1, T(1), T^2(1), \cdots, T^{n-1}(1)\} \tag{8.72}$$

such that $T^n(1) = 1$. Then we group all these particles using the collective variable

$$\boldsymbol{R}_A(\tau) = (\boldsymbol{r}_1(\tau), \cdots, \boldsymbol{r}_{T^{n-1}(1)}(\tau)), \tag{8.73}$$

and $\boldsymbol{R}_B(\tau)$ to be the collection of $\boldsymbol{r}_i(\tau)$ with i outside the orbit of the particle 1, we have

$$\begin{aligned}\langle \boldsymbol{R} |\rho| T[\boldsymbol{R}+X]\rangle = \int_{\boldsymbol{R}(0)=\boldsymbol{R},\boldsymbol{R}(\beta\hbar)=T[\boldsymbol{R}+X]} \mathfrak{D}\boldsymbol{R}_A(\tau)\mathfrak{D}\boldsymbol{R}_B(\tau) \\ e^{-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\, \frac{m\dot{\boldsymbol{R}}_A^2(\tau)}{2}+\frac{m\dot{\boldsymbol{R}}_B^2(\tau)}{2}+V[\boldsymbol{R}_A(\tau),\boldsymbol{R}_B(\tau)]}.\end{aligned} \tag{8.74}$$

Next we introduce the displaced path transformation with respect to permutation T as

$$\{Y_T(\tau)\}_i = \frac{i-1}{n} + \frac{\tau}{n\beta\hbar}, \quad i = 1, \cdots, n. \tag{8.75}$$

With this notation, Eq. (8.74) can be rewritten as

$$\begin{aligned}\langle \boldsymbol{R} |\rho| T[\boldsymbol{R}+X]\rangle =& e^{-\frac{m\boldsymbol{x}^2}{2n\beta\hbar^2}} \int_{\boldsymbol{R}(0)=\boldsymbol{R},\boldsymbol{R}(\beta\hbar)=T[\boldsymbol{R}]} \mathfrak{D}\boldsymbol{R}_A(\tau)\mathfrak{D}\boldsymbol{R}_B(\tau) \\ & e^{-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\, \frac{m\dot{\boldsymbol{R}}_A^2(\tau)}{2}+\frac{m\dot{\boldsymbol{R}}_B^2(\tau)}{2}+V[\boldsymbol{R}_A(\tau)+Y_T(\tau)\boldsymbol{x},\boldsymbol{R}_B(\tau)]}.\end{aligned} \tag{8.76}$$

Plugging Eq. (8.76) into Eq. (8.71), we have the displaced path formulation for bosons.

The displaced path formulation for bosons has the following properties:

1. The free particle contribution $e^{-\frac{m\boldsymbol{x}^2}{2n\beta\hbar^2}}$ factorizes inside each permutation T from the environmental contribution.

2. The free particle contribution $e^{-\frac{m\boldsymbol{x}^2}{2n\beta\hbar^2}}$ converges rapidly to 1 with respect to the size of the orbit. As a matter of fact, this directly indicates that if there is a nonzero probability for the infinitely long chain, the end-to-end distribution will have a non-vanishing value as $x \to \infty$, *i.e.* the off-diagonal long range order.

3. Perturbation method and thermodynamical integration method can still be ap-

plied as in distinguishable particle case.

8.8 Conclusion

In this chapter we have introduced the displaced path formalism for computing the end-to-end distribution, and therefore the momentum distribution of quantum particles. The new formalism has advantage both conceptually and computationally. From conceptual point of view, the free particle contribution and the environmental contribution factorize, and the potential energy surface is only reflected in the environmental part. We further derived the mean force, which is directly related to the potential energy surface, and can be used to analyze the Compton profile in the deep inelastic neutron scattering experiment.

From computational point of view, the displaced path formalism is more efficient than open path integral formulation. Numerical examples indicate that the advantage is already clear for one particle system. The advantage of the displaced path integral formulation is more prominent in the many particle case since the momentum distribution of all quantum particles can be evaluated using one single closed path integral simulation if the quantum effect is relatively small. In the situation of strong quantum effects, either thermodynamic integration technical or special enhanced sampling techniques must be used. This is our work in progress.

We have also established the semiclassical limit of the displaced path integral formulation. The semiclassical limit of the displaced path integral formulation provides a much more accurate description of the quantum effect than the isotropic model which regards the quantum effect on the momentum distribution as an effective increase of temperature.

The displaced path formulation gives a new kinetic energy estimator. The new kinetic energy estimator shares many similar properties with the virial estimator. The

classical and the quantum contribution to the kinetic energy are well separated from each other and the computational cost is the same as the virial estimator. In the new kinetic energy estimator, the quantum kinetic energy is determined by weighted average of the correlation of the force along the imaginary time axis. The new estimator does not explicitly depend on the path $r(\tau)$, as opposed to the virial formulation $\int_0^{\beta\hbar} d\tau\ r(\tau)V'[r(\tau)]$. Finally, we generalized the displaced path integral formalism to boson systems. The practical implementation of this formulation is our future work.

Other future work of the displaced path integral formulation can be its application in *ab initio* molecular dynamics simulation to sample the momentum distribution of quantum particles, particularly the directional momentum distribution for crystalline systems. In this case the displaced path integral formulation can focus on the important directions such as the hydrogen bond direction, and the performance should be much superior to that of the open path integral formulation. Besides the free energy perturbation method and the thermodynamic integration method, other sampling techniques are also in our scope in order to enhance the statistical sampling and to reduce the variance of the displaced path integral estimators.

Chapter 9

Momentum distribution, vibrational dynamics and the potential of mean force in ice

9.1 Introduction

Investigating the impact of hydrogen (H) bonding on molecular properties is the focus of intense research, but even behavior as fundamental as the equilibrium dynamics of the protons participating in H bonds remains poorly understood. Proton dynamics is reflected in the momentum distribution probed by deep inelastic neutron scattering (DINS) [5]. Recent DINS studies of H bonded systems have made striking observations, such as the presence of a secondary feature in the tail of the spherically averaged distribution in confined water [90], and estimates of a surprisingly large quantum kinetic energy of the proton in undercooled water [208, 209]. The secondary feature was attributed to quantum tunneling between the two wells of an anharmonic 1D potential [90]. It is not clear, however, to what extent the dynamics of an interacting many body system can be reduced to that of a single proton along a bond. For instance, it

has been pointed out that anisotropy can mimic features of a spherical distribution that one might associate to anharmonicity in a 1D model [233], and yet so far there is no conclusive study of this issue. To interpret experiments in confined and undercooled water, the unknown details of the molecular structure are a severe source of difficulty. However, even in the simpler case of ice Ih, it is not clear if the physics can be captured by simple model potentials, and how anharmonicity, anisotropy and structural disorder influence the momentum distribution.

In order to tackle these issues we consider the open path integral Car-Parrinello molecular dynamics (PICPMD) data for ice Ih that yielded the accurate spherical momentum distribution reported in a prior publication [191]. In this prior study, no attempt was made to relate the distribution to the equilibrium dynamics of the proton or to investigate the role of the environment in terms of a potential of mean force. In simulations this task is facilitated by access to the full 3D distribution, in contrast to experiments on polycrystalline samples, where only the spherically averaged distribution could be measured [5, 218]. In addition, crystalline symmetry allows the use of harmonic analysis to quantify the relation between the momentum distribution and vibrational dynamics, thereby elucidating the role of anharmonicity and disorder on the proton ground state.

We find that anisotropy stemming from the molecular orientations in the crystal has a larger effect on the momentum distribution than anharmonicity. The latter is effectively described within a quasi-harmonic model and is particularly important in the stretching motions, corroborating pump-probe laser experiments on the excited state dynamics of ice and water [17, 249]. This finding impacts the interpretation of infrared and x-ray spectroscopies, and regarding DINS experiments, the large effect of molecular anisotropy implies that it is not possible to unambiguously attribute to anharmonicity features of the spherically averaged distribution. Substantially more information, capable of disentangling anisotropy from anharmonicity, can be extracted

from the directional distribution, for which we now present the theoretical prediction for a realistic system.

This chapter is organized as follows. Section 9.2 introduces the quasi-harmonic potential of the mean force in ice Ih. Based on the quasi-harmonic potential of the mean force, we present a theoretical prediction of the directional momentum distribution for ice Ih. The principal frequencies in the quasi-harmonic potential is interpreted via the analysis of the vibrational dynamics in ice Ih in Section 9.3. The vibrational dynamics also reveals the existence of the anharmonicity along the hydrogen bonding direction, as well as the nuclear quantum effect of oxygens. The conclusion of this chapter is given in Section 9.4. The materials in this chapter have been presented in [167].

9.2 Momentum distribution and the potential of the mean force

The PICPMD simulation sampled the end-to-end distribution of the open Feynman paths of the protons [191], *i.e.* $\widetilde{\nu}(\boldsymbol{x}) = \frac{1}{N_p}\sum_i \widetilde{\nu}_i(\boldsymbol{x})$ where the sum runs over the N_p protons in the cell and the vector $\boldsymbol{x}$ points from one end of the path to the other. The momentum distribution $\nu(\boldsymbol{p})$ is the Fourier transform of $\widetilde{\nu}(\boldsymbol{x})$. For each distribution $\widetilde{\nu}_i(\boldsymbol{x})$ we compute the correlation matrix $C_{i,\alpha\beta} = \langle x_\alpha x_\beta \rangle$. Within the statistical errors of the simulation the eigenvalues $\{\sigma_k^2\}_{k=1}^3$ of C_i are the same for all the protons, while the associated eigenvectors $\{\boldsymbol{v}_{i,k}\}_{k=1}^3$ are proton specific directions related by crystalline symmetry to the directions of the other protons. This suggests an anisotropic Gaussian form for the end-to-end distribution: $\widetilde{\nu}_i(\boldsymbol{x}) \propto \exp\left(-\frac{1}{2}\boldsymbol{x}^T C_i^{-1}\boldsymbol{x}\right)$. Thus the momentum distribution is $\nu_i(\boldsymbol{p}) \propto \exp\left(-\frac{1}{2\hbar^2}\boldsymbol{p}^T C_i \boldsymbol{p}\right)$, implying that the corresponding potential of mean force has the *effective* harmonic form $V(\boldsymbol{r}) = \frac{M}{2}\boldsymbol{r}^T A_i \boldsymbol{r}$, where M and $\boldsymbol{r}$ denote the proton mass and position. A_i has eigenvalues ω_k^2 and shares with

C_i the eigenvectors, $\boldsymbol{v}_{i,k}$. The ω_k are related to the σ_k^2 by,

$$\frac{1}{\sigma_k^2} = \frac{M\omega_k}{2\hbar} \coth \frac{\hbar\omega_k}{2k_BT}, \tag{9.1}$$

and ω_k and $\boldsymbol{v}_{i,k}$ are denoted the principal frequencies and directions of proton i. Since the principal frequencies do not depend on i all the protons have equivalent local environments within the simulation error bars.

The hypothesis that the potential of mean force of the proton in ice Ih is quasi-harmonic can be verified directly by analyzing the quantile function [97] of the end-to-end distribution. For a one-dimensional probability distribution $p(x)$, the quantile function $Q(p)$ characterizes the inverse of the cumulative probability distribution $F(x)$

$$Q(p) = \inf\{x \in R : p \leq F(x)\}. \tag{9.2}$$

The quantile function can be used to compare two probability distributions by plotting the corresponding quantile functions against each other. This is called the quantile-quantile plot. The quantile-quantile plot between the end-to-end distribution collected from the PICPMD simulation along the hydrogen bond direction and the normal distribution that best fits this data is shown in the left panel of Fig. 9.1 alongside a plot of each distribution (right panel). The end-to-end distribution along the bond direction is very close to a normal distribution, thereby showing that the potential of mean force along this direction can be well described by a quasi-harmonic form. The quantile-quantile plot also exhibits small deviations at the tails indicating the presence of some degree of additional anharmonicity.

By averaging over the protons we obtain the frequencies $\bar{\omega}_k$ with error bars in the first row of Table 9.1. In terms of the $\bar{\sigma}_k^2$ the spherically averaged end-to-end

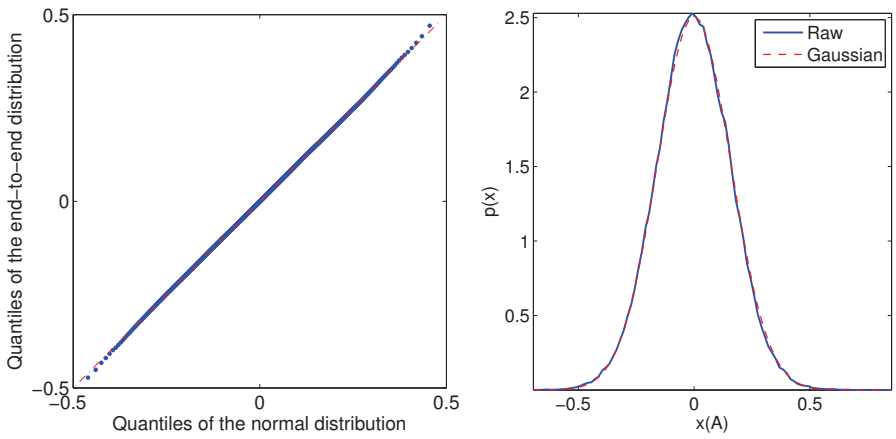


Figure 9.1: The Quantile-quantile plot between the end-to-end distribution along the bond direction and the normal distribution is depicted in the left panel. The distributions are shown in the right panel. The end-to-end distribution along the bond direction is very close to a normal distribution, but with small deviation at the tail. The quantile-quantile plot indicates that the potential of the mean force along the bond direction is well modeled by a quasi-harmonic potential.

distribution takes the form,

$$n(x) = \frac{1}{\sqrt{8\pi^3}\bar{\sigma}_1\bar{\sigma}_2\bar{\sigma}_3} \int_{|\boldsymbol{x}|=x} \mathrm{d}\Omega \; e^{-\frac{x_1^2}{2\bar{\sigma}_1^2}-\frac{x_2^2}{2\bar{\sigma}_2^2}-\frac{x_3^2}{2\bar{\sigma}_3^2}}. \tag{9.3}$$

Fig. 9.2(a) shows that this curve differs negligibly from the corresponding "raw" distribution extracted from the simulation, indicating that an effective harmonic model faithfully represents the spherically averaged data. Consistent with chemical intuition, the associated principal directions reflect the orientation of each water molecule in the crystal. The principal axes corresponding to the highest frequency are close to the oxygen-oxygen nearest neighbor directions, whereas the eigenvectors associated with the middle and lowest frequency correspond respectively to directions in and perpendicular to the HOH molecular plane.

The PICPMD principal frequencies differ from their harmonic counterparts (see Table 9.1). The latter were obtained with the phonon calculation discussed below. Thus the model that better represents the data is anisotropic and quasi-harmonic.

We can now resolve, in the case of ice, a major issue that troubled the interpretation of experiments [233] by quantifying the relative importance of anisotropy and anharmonicity. We depict in Fig. 9.2 (b) the spherical distributions corresponding to, respectively, the quasi-harmonic model (first row of Table 9.1), the harmonic model (second row of Table 9.1), and the isotropic model with frequency $\bar{\omega} = 1186\ \mathrm{cm}^{-1}$ that best fits the data. Anisotropy and anharmonicity are both significant, but anisotropy clearly has the larger effect. The isotropic model corresponds to a classical Maxwell-Boltzmann distribution with an effective temperature $\widetilde{T} = 869K$. In spite of $\widetilde{T}$ being significantly higher than the equilibrium temperature of the simulation ($T = 269K$), the isotropic model severely underestimates quantum effects, a finding that is also illustrated by a kinetic energy ($E_K = 111\mathrm{meV}$) approximately 30 percent smaller than the simulation value ($E_K = 143\mathrm{meV}$).

	$\bar{\omega}_1(\mathrm{cm}^{-1})$	$\bar{\omega}_2(\mathrm{cm}^{-1})$	$\bar{\omega}_3(\mathrm{cm}^{-1})$	$E_K(\mathrm{meV})$
PICPMD	2639 ± 60	1164 ± 25	775 ± 20	143 ± 2
Harmonic	3017.6 ± 8.2	1172.5 ± 8.9	870.3 ± 14.6	157.5 ± 0.3

Table 9.1: Average proton principal frequencies and kinetic energies obtained from PICPMD and phonon calculations. The error bars reflect statistical errors and physical effect of disorder in the PICMD and phonon data, respectively.

All the principal frequencies in Table 9.1 are well in excess of the equilibrium temperature, indicating that the dynamics of the proton is dominated by quantum zero-point motion. Dependence of the molecular orientations upon the crystalline framework originates anisotropies that reflect the symmetry of the environment in the momentum and end-to-end distributions. To study these effects we focus on the latter distribution, which factorizes into the product of a spherical free-particle contribution and an anisotropic environmental component $\widetilde{n}_V$, i.e. $\widetilde{n}(\boldsymbol{x}) \propto e^{-\frac{Mk_BT\boldsymbol{x}^2}{2\hbar^2}}\widetilde{n}_V(\boldsymbol{x})$ [166]. Rather than extracting $\widetilde{n}_V(\boldsymbol{x})$ directly from the PICPMD data, which would be affected by substantial noise, we reconstruct $\widetilde{n}_V(\boldsymbol{x})$ from the superposition of the individual proton contributions within the quasi-harmonic model. Here we use the

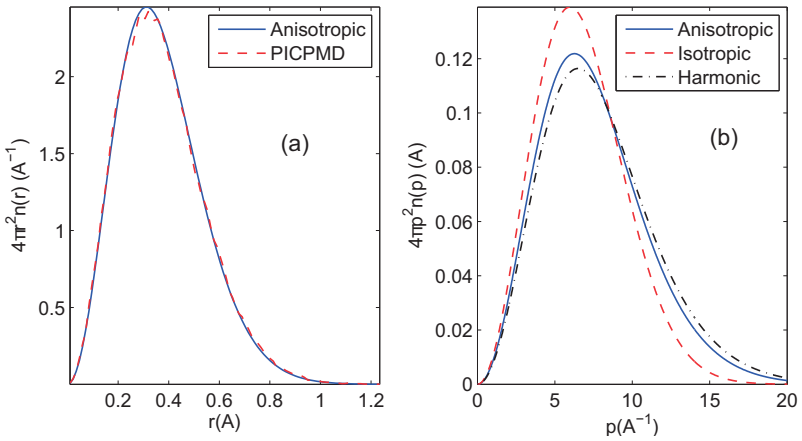


Figure 9.2: (a) The spherical end-to-end distribution directly collected from PICPMD data (red dashed line) compared with that reconstructed by the anisotropic fit (blue line). (b) Comparison of the spherical momentum distribution of the harmonic crystal (black dot-dashed line) with anisotropic (blue line) and isotropic (red dashed line) fits.

fact that there are 24 unique orientations of the molecules in the hexagonal ice crystal [122], and we also include the effects of proton disorder estimated below in the phonon calculation. Fig. 9.3 (a) depicts the log scale plot of one individual environmental end-to-end distribution projected on the basal plane of ice Ih. The elliptic shape of the contour comes directly from the quasi-harmonic model. Fig. 9.3 (b) illustrates the log scale plot of the superposition of all the environmental end-to-end distributions. The hexagonal shape of superpositioned distribution is a striking manifestation of quantum mechanics as in classical physics $\widetilde{n}_V(\boldsymbol{x})$ is equal to 1. While the distribution is spherical at the center, hexagonal character emerges at intermediate displacements and becomes pronounced in the tail of the distribution where blurring of the contour lines due to disorder can be detected. Experiments on ice Ih have only measured the spherical distribution [218] but it is likely that the full three dimensional distribution should become accessible in the future with improved instrumentation and preparation techniques. Directional momentum distributions have already been reported for materials such as KDP [219] and $Rb_3H(SO_4)_2$ [126]. It should be noted,

however, that the greatest sensitivity to anisotropy is in the exponential tail of the distribution, a finding indicating that substantial resolution may be necessary to experimentally disentangle anisotropy, anharmonicity and other environmental effects.

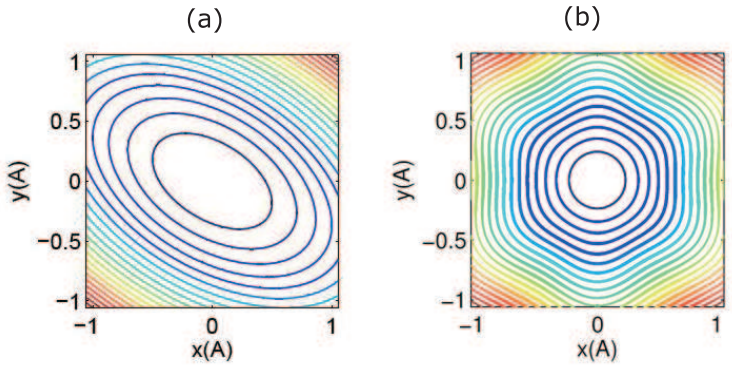


Figure 9.3: (a) "Environmental part" of the end-to-end distribution corresponding to one individual proton projected in the basal plane of ice Ih plotted in logarithmic scale. (b) "Environmental part" of the end-to-end distribution corresponding to the superposition of all protons projected in the basal plane of ice Ih plotted in logarithmic scale. The super positioned end-to-end distribution reflects the symmetry of the oxygen sub-lattice. The blurring of the contour lines reflects the disorder effect detected in the phonon calculation.

9.3 Vibrational dynamics

Now we discuss the relationship between the principal frequencies and the vibrational spectrum. The latter includes four main features experimentally: a stretching band centered at ≈ 3250 cm^{-1} [30], a bending band centered at ≈ 1650 cm^{-1} [238], a wide librational band between ≈ 400cm^{-1} and 1050cm^{-1} [30, 211] and a band of network modes below ≈ 400cm^{-1} [153]. These features are reproduced in the phonon spectrum of ice that we calculate by diagonalizing the dynamical matrix. This calculation is performed with Qbox [113] by adopting the same supercell, electronic structure parameters and disordered proton configuration of the PICPMD simulation [191]. The

dynamical matrix is calculated with a finite difference method (grid size of 0.0053Å). The resulting phonon density of states shown in Fig. 9.4 (a) agrees with experiment, and is consistent with previous calculations [189], which did not include proton disorder, indicating that such effects have a small influence on the spectrum. We indicate phonon frequencies and eigenvectors by ω_k^{ph} and $e_{i\alpha,k}$, respectively, where α are Cartesian components, $i,k = 1,\cdots,3N-3$, and N is the number of supercell atoms. In the quantum harmonic approximation the momentum distribution of particle i of mass M_i has the anisotropic Gaussian form $\nu_i(\boldsymbol{p}_i) \propto \exp\left(-\frac{1}{2}\boldsymbol{p}_i^T {C_i^{ph}}^{-1} \boldsymbol{p}_i\right)$ with correlation matrix [54],

$$C_{i,\alpha\beta}^{ph} = \langle p_{i,\alpha} p_{i,\beta} \rangle = \sum_k e_{i\alpha,k} e_{i\beta,k} \frac{M_i \hbar \omega_k^{ph}}{2} \coth\left(\frac{\hbar\omega_k^{ph}}{2k_B T}\right). \tag{9.4}$$

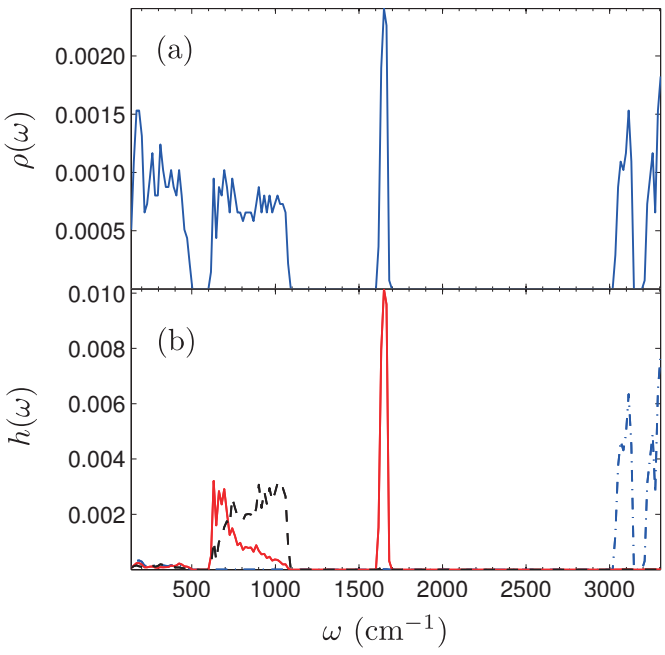


Figure 9.4: (a) Density of states of the phonon spectrum. (b) The population function for the principal axes corresponding to $\bar{\omega}_1$ (blue dot-dashed line), $\bar{\omega}_2$ (red solid line) and $\bar{\omega}_3$ (black dashed line). Network modes below 500cm^{-1} contribute non-negligibly to all principal frequencies.

As a consequence of disorder the eigenvalues of $C^{ph}_{i,\alpha\beta}$, depend on the proton index i. The harmonic average frequencies are reported in the second row of Table 9.1. The corresponding standard deviations originate almost entirely from ice disorder, being at least an order of magnitude larger than the numerical errors estimated from the small asymmetry of the calculated dynamical matrix. The statistical errors in the PICPMD simulation (Table 9.1) are on average a few times larger than the harmonic estimate of disorder, confirming that, within error bars, all proton environments are equivalent. We expect that longer runs combined with better estimators of the end-to-end distribution [166] should improve the statistical accuracy to the point that disorder effects could become measurable in future simulations.

The population function,

$$h(\omega^{ph}_k; l) = \frac{1}{N_p} \sum_{i=1}^{N_p} \left(\sum_{\alpha=1}^{3} v_{i\alpha,l} e_{i\alpha,k} \right)^2 , \tag{9.5}$$

gives the weight of the phonon k in the principal direction l and is depicted in Fig. 9.4 (b). It is found that $\bar{\omega}_1$ is 94% stretching, $\bar{\omega}_2$ is 47% bending and 48% libration, and $\bar{\omega}_3$ is 97% libration. Taking only stretching, bending, and libration into account, and using weights proportional to h we infer that $\bar{\omega}_1 \sim 3160\text{cm}^{-1}$, $\bar{\omega}_2 \sim 1210\text{cm}^{-1}$, and $\bar{\omega}_3 \sim 895\text{cm}^{-1}$. In comparison, the values in the second line of Table 9.1 are red-shifted by contributions from network modes ($6\%, 4\%$, and 3% to $\bar{\omega}_1, \bar{\omega}_2$, and $\bar{\omega}_3$, respectively), an intriguing result suggesting that fine details of the momentum distribution should reflect intermediate range order properties of the H bond network.

The behavior of the population function $h(\omega_k; l)$ can be explained by the behavior of a water monomer confined in an effective medium. Let us consider a free water monomer without the effective medium first. If the rotation mode is neglected, the free water monomer is confined in a 2D plane spanned by the two hydrogen bonds. We assume that the Hamiltonian can be completely characterized by the three vibra-

tional harmonic modes: symmetric stretching, asymmetric stretching and bending. To further simply the discussion, we assume the oxygen mass is infinite, *i.e.* the harmonic modes only involve the motion of protons. We also assume that the symmetric and asymmetric stretching frequencies are the same frequency ω_s, and we denote the bending frequency by ω_b. The Hamiltonian of a free water monomer is then

$$H(q,y) = \frac{1}{2}\sum_{i=1}^{3} q_i^2 + \frac{1}{2}\sum_{i=1}^{3} \omega_i^2 y_i^2. \tag{9.6}$$

The relation between the normal coordinates (q,y) and the Cartesian coordinates (p,x) is

$$\sqrt{m_i}x_i = \sum_j e_{ij}y_j, \quad \frac{1}{\sqrt{m_i}}p_i = \sum_j e_{ij}q_j. \tag{9.7}$$

Here e_{i1}, e_{i2}, e_{i3} are the eigenvectors corresponding to the frequencies $\omega_1 = \omega_s$, $\omega_2 = \omega_s$, $\omega_3 = \omega_b$.

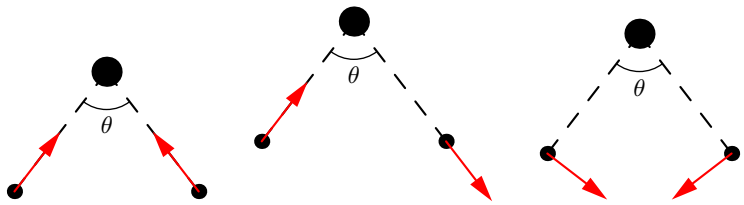


Figure 9.5: Normal modes for symmetric stretching (left), asymmetric stretching (middle) and bending modes (right). Big ball: oxygen. Small ball: hydrogen.

Since the motion of the oxygen is fixed, the eigenvectors can be directly read from

Fig. 9.5:

$$
\begin{aligned}
\{e_{i1}\} &= \frac{1}{\sqrt{2}} \begin{pmatrix} \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \\ -\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, \\
\{e_{i2}\} &= \frac{1}{\sqrt{2}} \begin{pmatrix} \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \\ \sin\frac{\theta}{2} & -\cos\frac{\theta}{2} \end{pmatrix}, \\
\{e_{i3}\} &= \frac{1}{\sqrt{2}} \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ -\cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \end{pmatrix}.
\end{aligned} \tag{9.8}
$$

The first line of $\{e_{ij}\}$ is the x, y components of the first proton, and the second line is the x, y components of the second proton. The 2-norm of each $\{e_{ij}\}_i$ is normalized to be 1. The two protons are identical, and we only calculate the momentum distribution for the first proton. The covariance matrix of the momentum distribution for the first proton is

$$
C_{\alpha\beta} = \sum_{i=1}^{3} e_{\alpha i} e_{\beta i} z_i, \quad \alpha, \beta = 1, 2, \tag{9.9}
$$

and the weight for harmonic frequency i is

$$
z_i = m\frac{\hbar\omega_i}{2} \coth \frac{\hbar\omega_i}{2k_B T}. \tag{9.10}
$$

For the free water monomer, we also write $z_1 = z_2 = z_s$, $z_3 = z_b$. Plug Eq. (9.8) into Eq. (9.9) we have

$$
\begin{aligned}
C_{11} &= \frac{1}{2}\left(2z_s \sin^2\frac{\theta}{2} + z_b \cos^2\frac{\theta}{2}\right), \\
C_{12} &= \frac{1}{2}\sin\frac{\theta}{2}\cos\frac{\theta}{2}(2z_s - z_b), \\
C_{22} &= \frac{1}{2}\left(2z_s \cos^2\frac{\theta}{2} + z_b \sin^2\frac{\theta}{2}\right).
\end{aligned} \tag{9.11}
$$

The two eigenvalues of C are

$$\sigma_1^2 = z_s, \qquad \sigma_2^2 = z_b/2, \tag{9.12}$$

and the eigenvector corresponding to σ_1^2 is along the OH bond direction, and that corresponding to σ_2^2 is along the orthogonal direction. We take $\omega_s = 3400\text{cm}^{-1}$, $\omega_b = 1500\text{cm}^{-1}$, then at room temperature $\frac{\hbar\omega_s}{2k_BT} = 8.75$, $\frac{\hbar\omega_b}{2k_BT} = 3.64$, and $\coth\frac{\hbar\omega_s}{2k_BT} = 1.000, \coth\frac{\hbar\omega_b}{2k_BT} = 1.001$, i.e. $z(\omega)$ is approximately a linear function with respect to ω if ω comes from the bulk part of the vibrational spectrum of water.

The second moments of the momentum distribution (9.12) should be interpreted in terms of the effective frequencies, and the two effective frequencies are approximately ω_s and $\omega_b/2$. The two principal directions are along and orthogonal to OH bond direction, respectively. We also note that this result is independent of the bond angle θ.

The analysis above can be readily generalized to the case that a water monomer confined in an effective medium. The Hamiltonian for a water monomer confined in the effective medium is

$$H(q,y) = \frac{1}{2}\sum_{i=1}^{6} q_i^2 + \frac{1}{2}\sum_{i=1}^{6} \omega_i^2 y_i^2. \tag{9.13}$$

The relation $\omega_1 = \omega_2 = \omega_s, \omega_3 = \omega_b$ still holds, and we have $\omega_4 = \omega_5 = \omega_6 = \omega_l$, where ω_l is the libration frequency. With fixed position of the oxygen, the 6 eigenvectors of

the dynamical matrix can also be written analytically, but in three dimension as

$$
\begin{aligned}
\{e_{i1}\} &= \frac{1}{\sqrt{2}} \begin{pmatrix} \sin\frac{\theta}{2} & \cos\frac{\theta}{2} & 0 \\ -\sin\frac{\theta}{2} & \cos\frac{\theta}{2} & 0 \end{pmatrix}, \\
\{e_{i2}\} &= \frac{1}{\sqrt{2}} \begin{pmatrix} \sin\frac{\theta}{2} & \cos\frac{\theta}{2} & 0 \\ \sin\frac{\theta}{2} & -\cos\frac{\theta}{2} & 0 \end{pmatrix}, \\
\{e_{i3}\} &= \frac{1}{\sqrt{2}} \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} & 0 \\ -\cos\frac{\theta}{2} & -\sin\frac{\theta}{2} & 0 \end{pmatrix}, \\
\{e_{i4}\} &= \frac{1}{\sqrt{2}} \begin{pmatrix} -\cos\frac{\theta}{2} & \sin\frac{\theta}{2} & 0 \\ -\cos\frac{\theta}{2} & -\sin\frac{\theta}{2} & 0 \end{pmatrix}, \\
\{e_{i5}\} &= \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \\
\{e_{i6}\} &= \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix}.
\end{aligned}
\tag{9.14}
$$

The covariance matrix of the momentum distribution of the first proton is given by

$$
C_{\alpha\beta} = \sum_{i=1}^{6} e_{\alpha i} e_{\beta i} z_i, \quad \alpha, \beta = 1, 2, 3. \tag{9.15}
$$

After some calculation, we find the three eigenvalues of C are

$$
\sigma_1^2 = z_s, \quad \sigma_2^2 = (z_b + z_l)/2, \quad \sigma_3^2 = z_l. \tag{9.16}
$$

The three eigenvectors are along the OH bond direction, orthogonal to OH bond direction but inside HOH plane, and perpendicular to the HOH plane, respectively. The three effectively frequencies are approximately $\omega_s, (\omega_b + \omega_l)/2, \omega_l$. This result is again independent of the bond angle θ, and is consistent with the free water monomer

result by assigning $\omega_l = 0$. This recovers the result of the population function in Fig. 9.4.

The potential energy surface is generated with the same protocol in path integral and phonon calculations. We thus attribute the difference between the average principal frequencies in the two rows of Table 9.1 to anharmonicity. This originates from quantum delocalization, present in the PICPMD simulation, which causes the proton to sample the potential over an extended range. Along the bond direction the proton spans from ≈ -0.2Å to $\approx +0.3$Å relative to the centroid of the path. This is much larger than the corresponding classical thermal spread ($\approx \pm 0.05$Å) indicating that quantum anharmonicity is essentially unaffected by temperature. The asymmetry of the quantal spread suggests that the first correction to the harmonic potential depends cubically on displacement.

In order to gain better insight on the anharmonicity in the bonding direction, we perform the following analysis. The potential energy surface is obtained by moving one proton along the hydrogen bond direction while the positions of all other atoms are fixed at their equilibrium positions. The resultant potential energy surface is depicted in Fig. 9.6, and the deviation from a harmonic potential can be readily seen. The ground state wavefunction $|\Psi^2|$ is also plotted in Fig. 9.6 in order to show the extent of the quantum delocalization of the proton (-0.2Å $\sim +0.3$Å). The potential energy surface about $x = 0$ is asymmetric, indicating a cubic dependence on displacement in the first anharmonic correction (black dashed line in Fig. 9.6). Higher order corrections set in at displacements larger than ≈ 0.3Å, which is clearly beyond the range of the ground state wavefunction. The harmonic frequency at the minimum of this potential is 3065cm^{-1}, close to the value of $\bar{\omega}_1$ garnered from the phonon calculation (see Table I in the manuscript). The size of the anharmonicity can be gauged upon comparison of this harmonic value with the effective frequency of 2847cm^{-1} obtained from the end-to-end distribution associated with the potential

in Fig. 9.6 at $T = 269$K. As expected, the anharmonicity lowers the value of the frequency and the shift is close to that between the PICPMD and the phonon derived results. The anharmonicity is a consequence of quantum delocalization which causes the proton to sample the potential energy surface over an extended range in the bond direction. It should be noted that the potential in Fig. 9.6 differs from the potential of mean force that the proton experiences in the simulation. We expect however that along the bond direction the two potentials should behave qualitatively similarly as suggested by the close similarity of their respective harmonic frequencies and anharmonic shifts. The delocalization along the bond that we find is comparable to the one observed in other ice phases with intact water molecules (see e.g. [26, 190]).

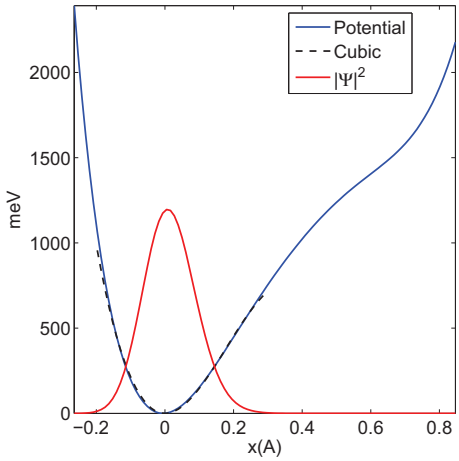


Figure 9.6: The potential energy surface of the proton in ice Ih along the bond direction (blue solid line), the cubic fitting potential (black dashed line) and the corresponding ground state wavefunction $|\Psi^2|$(red solid line).

The phonon calculation reported in the main text also yields an estimate of the quantum effects on the oxygen nuclei. The corresponding principal frequencies are $\bar{\omega}_1 = 640.1 \pm 16.6\text{cm}^{-1}$, $\bar{\omega}_2 = 585.1 \pm 9.8\text{cm}^{-1}$, and $\bar{\omega}_3 = 351.9 \pm 30.7\text{cm}^{-1}$. The frequencies mostly arise from the network modes, but are blue-shifted due to oxygen participation in stretching, bending and libration. The kinetic energy estimate for oxygen is 56.4 ± 0.4meV, and is approximately 35% in excess of the classical result

(35meV at 269K). The magnitude of this effect is consistent with earlier predictions [188] and with that found for fluorine nuclei in recent calculations on hydrogen flouride [146].

9.4 Conclusion

We find that to a large extent the momentum distribution in ice is a simple anisotropic Gaussian distribution. This does not mean, however, that ice behaves like a harmonic crystal as the principal frequencies of the distribution differ from those of a harmonic crystal. Anharmonicity, enhanced by H bonding, is appreciable in the libration dominated $\bar{\omega}_3$ and is particularly significant in the stretching dominated $\bar{\omega}_1$, in agreement with optical pump-probe experiments [17, 249]. The quantal character of the anharmonicity is consistent with the observed T-independence of the lifetime of excited stretching modes in ice [249]. Our findings have implications for the calculation of observables in ice, such as infrared spectra, which typically ignore quantum anharmonicity [60], and x-ray absorption spectra, which typically ignore quantum configurational disorder [61]. The approach presented here could be applied directly to the study of other crystalline H bonded systems, and is also an important step towards a better understanding of the proton momentum distribution in disordered H bonded systems such as water under different conditions. In such cases only the spherically averaged momentum distribution is accessible in experiment and simulation can provide essential microscopic information to supplement and interpret the experimental data. Finally, we remark that while the qualitative picture emerging from our calculations is robust, the path integral data have relatively large error bars and the quantitative details depend on the accuracy of the underlying Born Oppenheimer potential energy surface. The latter should reflect the known limitations of the GGA functional used in this study [109, 222] and comparisons with future high

resolution experiments should help to clarify this issue.

Chapter 10

Quantum proton in hexagonal ice: interpretation of a new experiment

10.1 Introduction

Although liquid and solid phases of water are the focus of a considerable number of experimental and theoretical investigations because of their biological and technological importance, several physical properties of water are not well understood. Progress in this area requires an accurate description of the proton motion in hydrogen bonded systems, something that has been difficult to measure directly. Recently new experimental and simulation techniques have been used to probe the quantum state of protons in water and ice by examining the proton momentum distribution, $n(p)$, which is determined almost entirely by quantum effects [5]. Experimentally, $n(p)$ can be directly measured by Deep Inelastic Neutron Scattering (DINS) [217, 219], where neutrons probe proton dynamics at high energy, $\hbar\omega$, and high momentum, $\hbar q$, transfers. As well as providing information on proton quantum dynamics, DINS is also sensitive to the proton's local environment, *i.e.* the potential of mean force experienced by the protons. In recent years, several DINS studies have addressed the

study of bulk water in stable liquid [200], solid [218], and supercooled liquid [208,209] phases. In parallel, novel simulation techniques have been employed to calculate the $n(p)$ using open path integral simulations [192] implemented with first principles molecular dynamics[49] within the Path Integral Car-Parrinello Molecular Dynamics (PICPMD) framework [183]. The path integral simulation has access to the three dimensional $n(\vec{p})$, and thus provides complementary information to the spherical averaged $n(p)$ obtained via DINS from isotropic ice samples. The calculated $n(p)$ in ice from Ref. [192] revealed both agreement and discrepancies with that measured in previous DINS measurements on ice at T=269 K by Reiter *et al.* [218]. In particular the calculated $n(p)$ failed to reproduce the tail of the experimental distribution.

This section reports new theoretical and experimental studies of the proton $n(p)$ in ice at T=269 K and T=271 K. At these temperatures, the momentum distribution in ice is due almost entirely to zero point motion, providing a sensitive probe of the proton's local environment. Here a quasi-harmonic description is expected to be valid, whereas in the supercooled liquid at 271 K, the large excess of proton mean kinetic energy was interpreted, in Ref. [208], in terms of possible anharmonicity in the potential energy surface.

This chapter is organized as follows. The setup of DINS experiment is introduced in Section 10.2. The analysis of the experimental data is discussed in Section 10.3, followed by a non-parametric approach of quantifying the uncertainty in the experimental data in Section 10.4. The conclusion of this chapter is given in Section 10.5. Materials in this chapter have been presented in [84].

10.2 DINS Experiment setup

Refined DINS measurements, using resonance-detector (RD) and foil-cycling techniques (FCT), provide remarkable improvements, with respect to existing measure-

ments on bulk ice [218], in both energy resolution (32 % narrowing of FWHM), and counting statistics, (*i.e.* achieving 1% error at the center of the overall Neutron Compton profile and of 15% at 1/15 of the peak height, respectively). Perhaps, more importantly, a much better separation between proton peaks and the peaks from heavier atoms in the sample and the container is achieved, thus eliminating any possible spurious distortion due to inaccurate subtraction of the O, Al contributions. This also considerably reduces the uncertainty in the determined kinetic energy, from 11% in the previous measurements [218], to $\simeq 1$ % in the present case. Moreover, the current resolution line shape has a finite variance, allowing us also to carry out non parametric determinations of kinetic energy as outlined below.

The DINS experiment was performed at the time of flight Vesuvio spectrometer (ISIS neutron source-UK) in the range 2.5 eV $\leq \hbar\omega_r \leq$ 30 eV. Scattered neutrons were detected by 64 scintillator detectors, located in the angular range $32.75^\circ \leq \vartheta \leq 72.5^\circ$. The sample was a 65×65×1 mm^3 slab of polycrystalline ice contained in an Al holder, equipped with Rh/Fe thermometers. At each scattering angle the energy of the scattered neutrons, E_1, is selected by using the RD and FCT by Au analyzers (E_1 =4897 meV), providing a resolution in y-space of approximately 2 Å^{-1} FWHM, and a complete removal of the sample-independent background. For each detector, the time of flight data were corrected for multiple scattering, heavy atom (O, Al) recoil signals, and residual gamma background. The time of flight spectra were then transformed into fixed-angle experimental NCP, $F_l(y,q) = [J_{IA}(y) + \Delta J_l(y,q)] \otimes R_l(y,q)$ where l refers to the angular position of the l-th detector. The set $F_l(y,q)$ is expressed in terms of l independent determinations of the longitudinal $n(p)$, $J_{IA}(y)$, and q-dependent deviations from the impulse approximation (Final State Effects), $\Delta J_l(y,q)$, broadened by the instrumental resolution function $R_l(y,q)$. Fixed-angle histograms of $F_l(y,q)$ have been binned in the range -30 Å$^{-1} \leq y \leq$ 30 Å^{-1} and then normalized.

The overall quality of the DINS spectra can be appreciated in Figure 10.1, which shows the angle-averaged $F_l(y,q)$, henceforth named $\bar{F}(y)$.

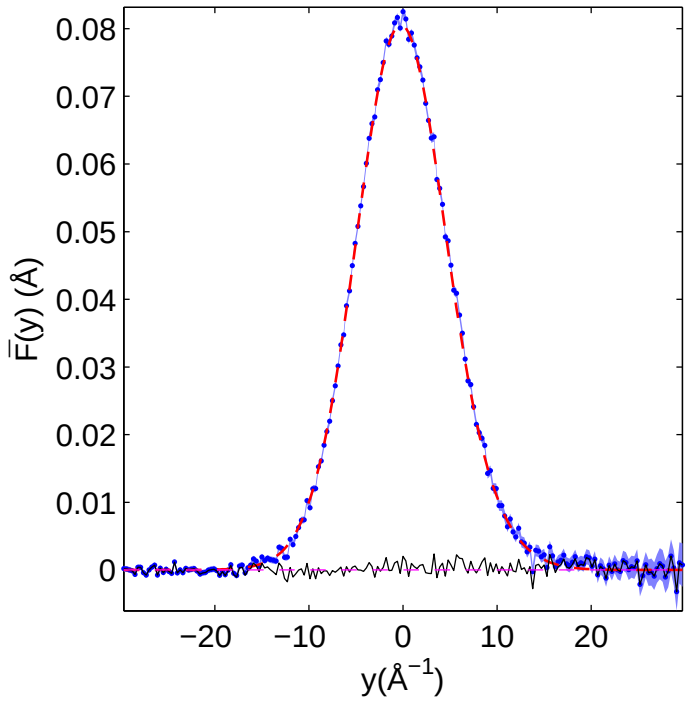


Figure 10.1: Experimental Neutron Compton Profile for ice at T = 271 K averaged over the whole set of the scattering angles ($\bar{F}(y)$=< $F_l(y,q)$ >$_l$) (blue dots with error bars). The angle-averaged best fit is reported as a red dashed line for the M1 model (see text for details). The fit residuals are reported as a black continuous line.

10.3 Data analysis by parametric estimation

DINS data were analyzed within the Impulse Approximation (IA), *i.e.* a neutron-single atom scattering process with conservation of momentum and kinetic energy. The recoil energy is: $\hbar\omega_r = \hbar^2 q^2/2M$, $\hbar\omega_r$, where M is the proton mass, and q is the wave vector transfer. The dynamical structure factor for an isotropic system is related to $n(p)$ by:

$$S(q,\omega) = \int n(p)\delta(\omega - \frac{\hbar q^2}{2M} - \frac{\mathbf{q}\cdot\mathbf{p}}{M})dp = \frac{M}{q}J_{IA}(y) \qquad (10.1)$$

where $y = \frac{M}{q}(\omega - \frac{\hbar q^2}{2M})$ and $J_{IA}(y)$ is the longitudinal $n(p)$. The single particle mean kinetic energy is: $\langle E_K \rangle = \frac{3\hbar^2}{2M} \int_{-\infty}^{\infty} y^2 J_{IA}(y) dy = \frac{3\hbar^2}{2M} \sigma^2$.

The prime objective of the present DINS experiment was to determine the $J_{IA}(y)$ line shape from the $F_l(y,q)$ spectra, thus determining $n(p)$ and $\langle E_K \rangle$. This has been accomplished via : 1) Determination of the kinetic energy $\langle E_K \rangle$ by numerical integration of $\bar{F}(y)y^2$; 2) Full analysis of the DINS line shape via simultaneous fitting of the individual $F_l(y,q)$ spectra with: a) a model-independent line shape; b) a three dimensional anisotropic Gaussian line shape derived from a quasi-harmonic model as suggested by a recent study of the PICPMD simulations for hexagonal ice [167]. As outlined in Ref. [234], the numerical integration of $\bar{F}(y)y^2$, provides a first-order estimate of σ^2 and $\langle E_K \rangle$: by integrating $\bar{F}(y)y^2$ and subtracting the variance of the angle-averaged resolution ($\sigma_R^2 = 0.98$ Å^{-2}) we obtain $\sigma^2 = 27.0 \pm 2.7$ Å^{-2} . Systematic uncertainties, due to the limited range of integration, and residual differences between angle-averaged and constant-q representations of $\bar{F}(y)$ are evaluated to be $\simeq 0.3$ Å^{-2}. Therefore $\sigma^2 = 27 \pm 3$ Å^{-2}, $\sigma = 5.2 \pm 0.3$ Å^{-1}, and $\langle E_K \rangle = 169 \pm 19$ meV. This determination can be used as a constraint for the variance of $n(p)$ in fitting the $F_l(y,q)$ data set. The DINS data were then analyzed using a model-independent form for $J_{IA}(y)$ [226], already used in previous work [5]:

$$J_{IA}(y) = \frac{e^{\frac{-y^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} \Big[1 + \sum_{n=2}^{\infty} \frac{a_n}{2^{2n} n!} H_{2n}\Big(\frac{y}{\sqrt{2}\sigma}\Big) \Big]. \tag{10.2}$$

where H_{2n} are the Hermite polynomials and a_n the corresponding coefficients. The $n(p)$ is expressed in terms of a Gaussian times a series of generalized Laguerre polynomials, $L_n^{\frac{1}{2}}(\frac{p^2}{2\sigma^2})$, with coefficients $(-1)^n a_n$. For finite q values, the deviation from the IA can be accounted for by additive corrections [5], $\Delta J(y,q) \simeq c_\Delta \frac{e^{\frac{-y^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} H_3(y/\sqrt{2}\sigma)/q$, with $c_\Delta \propto \langle \nabla^2 V \rangle$, where V is the effective proton potential [226]. The simultaneous fitting of the above line shape, to the whole set of $F_l(y,q)$ spectra, has been performed

via equation (2). This fit, referred to as M1 in the following, yielded $\sigma = 4.99 \pm 0.03$ Å^{-1}, $a_2 = 0.10 \pm 0.01$ and $a_{n>2} = 0$,$c_\Delta = 0.36 \pm 0.02$ and $\langle E_K \rangle = 156 \pm 2$ meV.

Eq. (10.2) has the most general form, but may not facilitate interpretation of the data. For example, if the momentum distribution corresponding to an anisotropic harmonic system $V(x, y, z) = \frac{m}{2}(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2)$ is to be measured, the harmonic frequencies ω_i cannot be directly reflected in Eq. (10.2). In that case one calculates the harmonic frequencies ω_i by measuring σ_i^2, *i.e.* the variance of the momentum distribution $n(\vec{p})$ along direction i by

$$\sigma_i^2 = \frac{m\hbar\omega_i}{2} \coth \frac{\beta\hbar\omega_i}{2}. \tag{10.3}$$

While it is only the spherically averaged momentum distribution that is accessible in the experiment, the PICPMD is able to access the three dimensional $n(\vec{p})$. As a result the harmonic frequencies along three directions can be calculated with relatively small error bars. The frequencies obtained from PICPMD are: $\omega_z = 2639 \pm 60$ cm^{-1}, $\omega_x = 1164 \pm 25$ cm^{-1}, $\omega_y = 775 \pm 20$ cm^{-1}. The interpretation of these effective frequencies deserves further comment. A careful analysis of the PICPMD result shows that the effective frequencies $\omega_{x,y,z}$ measured from proton $n(p)$ are closely related to the vibrational spectrum [167]. The experimentally measured vibrational spectrum of hexagonal ice concentrates at the stretching frequency (centered at ≈ 3200 cm^{-1}), bending frequency (centered at ≈ 1650 cm^{-1}) and libration frequency (broad spectrum centered at ≈ 900 cm^{-1}) respectively [188]. It is shown [167] that ω_z, ω_y and ω_x represent weighted averages of the stretching frequencies, librational frequencies and a mix of bending and librational frequencies respectively, with red-shifts due to network phonon modes with frequencies at and below 200 cm^{-1}. The PICPMD analysis indicates a clear connection between the quantum momentum distribution and the vibrational spectrum. It is also possible to extract the effective frequen-

cies from the experimental (spherically averaged) distribution directly by taking: $n(p) = \left\langle \frac{1}{\sqrt{8\pi^3}\sigma_z\sigma_x\sigma_y} \exp\left(-\frac{p_z^2}{2\sigma_z^2} - \frac{p_x^2}{2\sigma_x^2} - \frac{p_y^2}{2\sigma_y^2}\right) \right\rangle_\Omega$. The experimental NCP have been fitted using the above model, labeled M2, with σ_x,σ_y, σ_z as free parameters, and with Final State Effects as outlined above for M1. However, numerical results show that σ_x and σ_y tend to be degenerate, given the current data set, leaving the error bars on the effective frequencies poorly defined. Although some studies have used $\sigma_t = \sigma_x = \sigma_y$ as a parameter for transverse direction[218], this is not an accurate representation of the physics. Compromise between the numerical optimization and physical intuition was achieved by adding a weighting term in the least square fitting of the experimental Compton profiles to minimize the deviation between σ_i from the fitting and from the PICPMD analysis. The magnitude of the weighting term reflects the physical range of σ_i, or equivalently the physical range of the effective frequency ω_i. The estimated effective frequencies in M2 are $\omega_z = 2797\pm95$ cm^{-1}, $\omega_x = 1233\pm110$ cm^{-1}, $\omega_y = 922\pm81$ cm^{-1}. It is noted that ω_z is underestimated in PICPMD analysis by 200 cm^{-1} compared to ω_z in M2. This underestimation is likely due to large extent to the error in the exchange-correlation functional in the simulation based on density functional theory. The BLYP exchange-correlation functional [22, 149] used in the current simulation overestimates the hydrogen bond strength because of self-interaction error, and therefore softens the potential along the hydrogen bond direction. The radial momentum distribution $4\pi p^2 n(p)$ from M1 , M2 and PICPMD analysis are plotted in Figure. 10.2. PICPMD analysis results in a shorter tail than M1 and M2. The tail behavior is dominated by ω_z which is underestimated in PICPMD. The underestimation can also be confirmed from the kinetic energy: 156 ± 2 meV (M1), 154 ± 2 meV (M2) and 143 ± 2 meV (PICPMD).

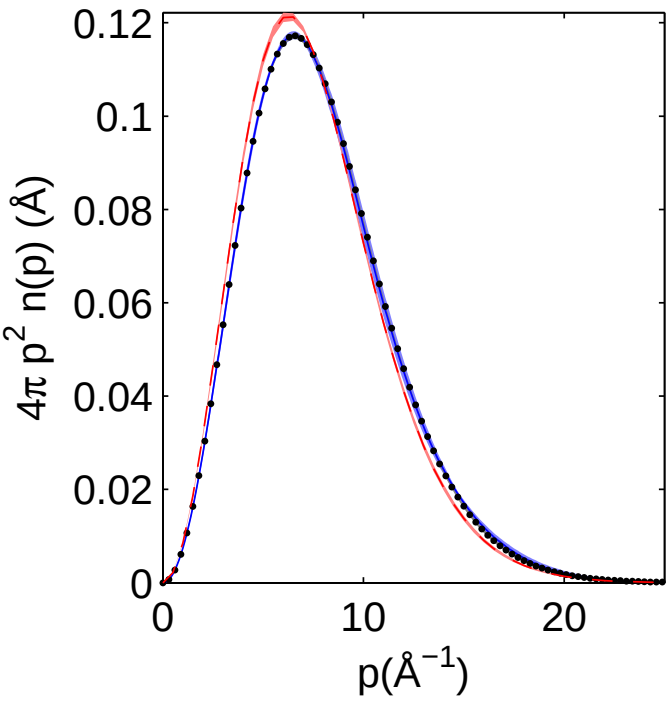


Figure 10.2: Experimental radial momentum distribution obtained using model M1 (blue solid line), M2 (black dots) and PICPMD (red dashed line) with error bars. Errors on the radial momentum distribution for M1 and M2 are determined from the uncertainty in the measured coefficients, through their correlation matrix calculated by the fitting program.

10.4 Nonparametric uncertainty quantification

The fact that σ_x and σ_y tend naturally to be degenerate in M2 also indicates that the spherical momentum distribution itself is not sensitive enough to distinguish all three anisotropic frequencies of the system. This confirms the recent theoretical work for hexagonal ice [166] in which a more sensitive quantity named mean force is proposed. The mean force is defined as $f(x) = (-\log n(x))' - \frac{mx}{\beta\hbar^2}$. $n(x)$ in the first term is the spherical end-to-end distribution, *i.e.* the Fourier transform of the spherical momentum distribution. The second term represents the free particle contribution which is irrelevant to the potential energy surface. The experimental NCP $\bar{F}(y)$ has been corrected for the Final State Effects $\Delta J(y,q)$ providing the "asymptotic" $\bar{F}_{IA}(y)$,

so that the mean force can be directly calculated by

$$f(x) = -\frac{mx}{\beta\hbar^2} + \frac{\int_0^\infty dy\ y \sin(xy/\hbar)\bar{F}_{IA}(y)}{\hbar \int_0^\infty dy\ \cos(xy/\hbar)\bar{F}_{IA}(y)}. \tag{10.4}$$

The mean force calculated using Eq. (10.4) (blue solid line), from the anisotropic Gaussian model M2 (black dots) and from PICPMD (red dashed line) are plotted in Fig. 10.3 with error bars. The three mean forces have good correspondence below 0.4

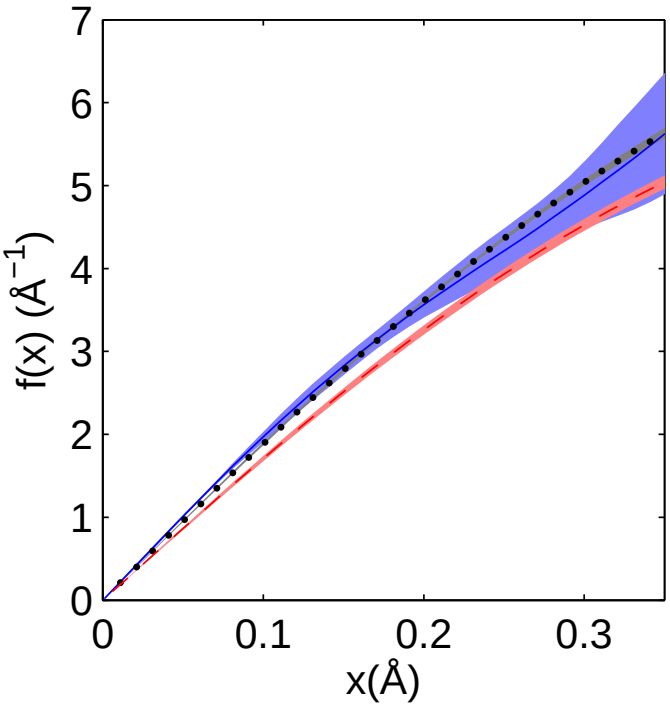


Figure 10.3: Mean force calculated directly from the experimental asymptotic Compton profile, $\bar{F}_{IA}(y)$ (blue solid line), M2 (black dots) and PICPMD analysis (red dashed line) with error bars.

Å, indicating that the proton quantum state in ice is well described by harmonic and anisotropic motion. Above 0.4 Å, the mean force calculated from the experimental Compton profile tends to diverge. The mean force is related to the derivative of the Fourier transform of the Compton profile, and therefore, at large x, is related to its highest frequency components, *i.e.* to the noise. The kinetic energy estimated from the mean force is 156 ± 9meV. The error bar of the kinetic energy calculated from

the mean force is larger than that obtained from M1 or M2, since the mean force is an non-parametric approach and is model independent. The uncertainty of the mean force indicates the accuracy of the experiment required to resolve the anisotropic frequencies accurately.

10.5 Conclusion

In conclusion, we have elucidated the connection between the proton momentum distribution and the underlying potential energy surface in ice. To a large extent, the physics of the PICPMD simulation is captured by a simple anisotropic Gaussian model. This does not mean, however, that ice behaves like a harmonic crystal as the effective frequencies of the distribution differ from those of a harmonic crystal. The anharmonicity is particularly important in the stretching dominated $\bar{\omega}_1$. The estimated anharmonic shift ($200 - 300\text{cm}^{-1}$) is large but consistent with previous estimates based on optical pump-probe experiments [17, 249]. We should remark, however, that while the qualitative picture emerging from our calculations is robust, the path integral data have relatively large statistical errors and the quantitative details depend on the accuracy of the underlying Born-Oppenheimer potential energy surface. The latter should reflect the known limitations of the GGA functional used in this study [109, 222] and comparisons with future high resolution experiments should help to clarify this issue. The cause of the anharmonicity is quantum delocalization of the protons over an extended range of the potential energy surface. This effect, already present in stretching modes in the gas phase, is substantially enhanced by H bonding. Interestingly, we also find non-negligible anharmonicity (100cm^{-1}) in $\bar{\omega}_3$, which is dominated by libration modes. Finally, the result that the momentum distribution is affected, albeit weakly, by network modes is intriguing as it suggests that fine details of the distribution should also reflect intermediate range order properties

of the H bond network.

This work illustrates how the theoretical and experimental determination of the momentum distribution on a benchmark system like polycrystalline ice can directly access the physical mechanisms describing the proton quantum state. This study can be further used to investigate the role of nuclear quantum effects in a variety of hydrogen bonded systems.

Chapter 11

Correlated tunneling in hydrogen bonds

11.1 Introduction

Proton tunneling plays an important role in phase transitions such as the ferroelectric to paraelectric transition in H bonded KDP or the sequence of transitions leading to H bond symmetrization in pressurized ice. These transitions exhibit large isotope effects that can only be explained by invoking quantum fluctuations. In this chapter we focus in particular on ice VIII, VII and X as these phases epitomize the H bond symmetrization transformation that takes place when the oxygen-oxygen nearest neighbor separation is progressively reduced by applying pressure, thereby mutating the ice crystal from a H-bonded molecular system to an heteroatomic system with covalent/ionic bonds. The lower pressure phase ice VIII is characterized by the usual asymmetric H bonds, similar to those found in ice Ih, the stable ice structure at standard temperature and pressure. In these phases each molecule is bonded to four neighboring molecules and the proton distribution in the lattice of H bonds satisfies the ice rules [29, 203]. A configuration satisfies the ice rules if on the four bonds

connecting an oxygen to its neighbors two hydrogens (protons) are near the central oxygen and two are near the neighboring oxygens, as is required to keep the water molecules intact. While disordered in ice Ih, the proton lattice is ordered in ice VIII engendering an antiferroelectric arrangement of molecular dipoles. Long range antiferroelectric order is lost in the phase occurring at intermediate pressure, ice VII, where the protons can be found with equal probability on the two bond sites. The transition from ice VIII to ice VII can be viewed as an order-disorder transition leading from an antiferroelectric to a paraelectric crystal. Lastly, in ice X, which is the phase at the highest pressure, H bond symmetrization is complete and the probability distribution of the proton along the bond is unimodal and centered at midbond.

The important role of quantum tunneling in the transformation from ice VIII to VII was first suggested by Stillinger and Schweizer in a series of remarkable papers based on an insightful but simplified model for the proton sublattice. In the first of these papers [235] the authors adopted a mean field approximation describing the transition in terms of independent protons tunneling coherently back and forth between the two off-center bond sites. In the subsequent papers [224, 225] they pointed out that, although qualitatively correct, the mean field description was not to be trusted quantitatively as it led to a complete ionization catastrophe accompanied by a large deviation of the ice rules. Correlations among the protons should oppose coherent tunneling, they noticed, in order to partially restore the ice rules and the charge neutrality of the water molecules. The crucial role of tunneling in the H bond symmetrization transitions was confirmed 15 years later in a cutting-edge *ab initio* simulation by Parrinello and collaborators [27]. This study adopted Feynman's path integral formulation of quantum statistical mechanics to sample the equilibrium configurations of the oxygen and hydrogen nuclei in a periodically repeated supercell containing 16 water molecules. In this approach the nuclei can take any position in space (an off-lattice model) and are treated quantum mechanically without recourse

to mean field or variational approximations as in Refs. [224, 225, 235]. Thermal effects are also included while Refs. [224, 225, 235] were limited to $T = 0$K. Finally, in Ref. [27] the internuclear interactions were generated from first-principles, *i.e.* they were obtained on the fly from the instantaneous ground state energy of the electrons, calculated accurately, albeit approximately, within electronic density functional theory (DFT). This approach avoids empirical parameterization of the interatomic force field, an important aspect in the context of the H bond symmetrization transformation in which the character of the interatomic bonds, the electron charge distribution, and the potential experienced by the protons are all expected to undergo considerable changes. Besides confirming that proton tunneling drives the transitions Ref. [27] reported a novel prediction, namely that zero point motion leads to H bond symmetrization before the potential experienced by the proton converts from a double to a single well. This finding suggested the following classification of H bonds: (i) at large inter-oxygen separations such as $d_{OO} \sim 2.78$Å, typical of ice Ih, or $d_{OO} \sim 2.53$Å, typical of ice VIII, standard H bonds are present in which the tunneling splitting is zero or negligibly small and the protons experience an effective asymmetric single well potential, (ii) at intermediate separations such as $d_{OO} \sim 2.45$Å, typical of ice VII, so called high barrier H bonds (HBHB) are present in which tunnel splitting is non negligible and larger than the thermal energy, and at least one of the split levels falls well below the top of the barrier between the two wells along a bond so that quantum tunneling originates a bimodal proton distribution, (iii) at shorter separations such as $d_{OO} \sim 2.31$Å within the stability range of ice X, so-called low barrier H bonds (LBHB) are present in which the potential remains double welled but the proton distribution is unimodal due to zero-point motion, (iv) at even shorter separations within the stability range of ice X, the potential becomes single welled as illustrated in Fig. 11.1.

The picture in Fig. 11.1 is suggestive but still rooted in mean field theory. Even

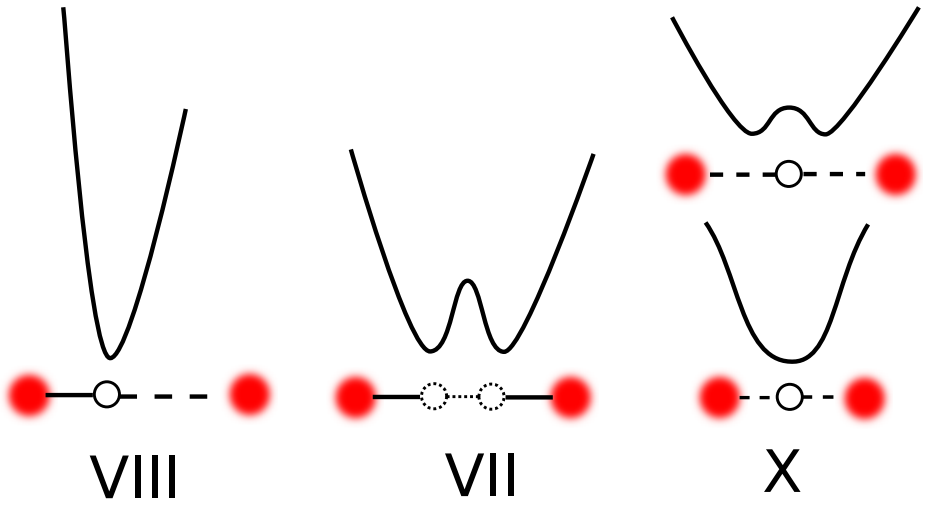


Figure 11.1: Cartoon depicting the understanding established in the literature. As pressure is increased the bond undergoes a transition from single welled (ice VIII) to a high-barrier (HBHB, ice VII) and then low-barrier (LBHB, ice X) double welled potentials until a unimodal form centered at the midpoint (highest pressure, ice X) persists.

though the path integral simulation included proton correlations consistent with the finite size of the periodic supercell and with the adopted DFT approximation for the Born-Oppenheimer potential energy surface, Ref. [27] did not consider the role of correlations in forming its picture. These could only be monitored to limited extent in a simulation that only accessed the particle density in space. The proton density distribution, $n(\vec{r})$, is the diagonal part of the corresponding single particle density matrix $\rho(\vec{r},\vec{r'})$. Access to diagonal and off-diagonal components of $\rho(\vec{r},\vec{r'})$ provide more direct information about correlations. Indeed, for a system in the ground state the density matrix is idempotent, i.e. $\rho^2 = \rho$, when correlations are absent, while deviations from idempotency, signal entanglement due to correlations. Recently, we performed *ab initio* path integral simulations of ice Ih [191], VIII, VII, and X [190], in which we employed the same methodology of Ref. [27] but sampled open in addition to closed Feynman paths, thus allowing access to the full single-particle density matrix $\rho(\vec{r},\vec{r'})$ of the protons. The so called end-to-end distribution of the Feynman paths $\widetilde{n}(\vec{x})$ is defined as $\int d\vec{r}d\vec{r'}\ \rho(\vec{r},\vec{r'})\delta(\vec{r}-\vec{r'}-\vec{x})$: it is the distribution of the distances

separating the two opposite ends of an open Feynman path. The Fourier transform of $\widetilde{n}(\vec{x})$ yields the momentum distribution $n(\vec{p})$ of a quantum particle. In the case of the protons this quantity can be directly compared to available DINS experiments. The proton momentum distribution was calculated in this way in Ref. [191] for liquid water and ice Ih, and in Ref. [190] for ice VIII, VII and X. Correlations among the protons, not considered in the cited publications, are the subject of this chapter. We limit the discussion to ice. We start by demonstrating that the basic physics can be unveiled by considering a reduced model in which only the projection of the density matrix along a bond, $\rho(x, x')$, is taken into account. This follows from statistical analysis showing that proton motions along the bond are uncorrelated from the perpendicular motions, and is consistent with simple intuition suggesting that correlations should play a role only when the displacements along the bond are so large as to break the ice rules.

In both ice Ih at T=269 K and in ice VIII at T=100K $\rho(x, x')$ is idempotent within the statistical errors of the simulation, indicating that the proton is approximately in a pure quantum state. Finding that these systems are ground state dominated is not surprising as stretching motions in common H bonds have frequencies around 3000cm^{-1} and the zero point energy is much larger than k_BT. Interestingly, in ice VII at $T = 100K$, a system with HBHB, large deviations from idempotency are found. Deviations from idempotency reduce significantly but are still visible in ice X at T=100K, when LBHB are present. The mixed state character of the density matrix $\rho(x, x')$ may be due to thermal and/or correlation effects, but our analysis shows that the latter dominate. Collective tunneling is a consequence of the ice rules that act to protect the integrity of the water molecules and reduce the likelihood of ionized configurations such as H_3O^+ and OH^-. In ice VII correlations favor concerted tunneling of the protons. Interestingly, correlation effects can also be detected in ice X when LBHB are present and individual water molecules can no longer be identified as

the proton density distribution on the bond has become symmetric and unimodal. In this case correlations manifest as charge density waves that reflect dipolar fluctuations on the bonds. In both cases, that of HBHB and that of LBHB, the requirement of local charge neutrality is the mechanism that constrains quantum fluctuations causing correlated dynamics.

Plots like the one in Fig. 11.1 are very useful to illustrate the H bond symmetrization transformation but are an oversimplification. To make the picture more quantitative it is convenient to construct the effective potential that simultaneously fits $n(x)$ and $\widetilde{n}(x)$, i.e. the positions and the end-to-end distribution of the proton along the bond. In ice Ih and VIII, which are effectively in a pure quantum state at the temperature of the simulation, position and end-to-end distributions convey the same information. In fact both distributions are simply expressed in terms of the real ground state wavefunction $\psi(x)$ of the proton along the bond: $n(x) = \psi(x)^2$ and $\widetilde{n}(x) = \int dy\ \psi(y+x)\psi(y)$. The effective bond potential for the proton provides a unique perspective on the anharmonicity of the proton environment. Anharmonicity is substantially larger in ice VIII than in ice Ih, as one would expect for a system close to the onset of tunneling fluctuations. In presence of tunneling we are unable to find a unique potential that simultaneously fits the position and end-to-end distributions. This signals that the proton is no longer in a pure state but is in an entangled state due to its correlations with the other protons. When this occurs the proton can only be described by a potential ensemble reflecting its mixed quantum state character. This analysis leads to a new picture of the H bond symmetrization transformation that is presented in this chapter.

It has been pointed out that proton dynamics hindered by ice rule correlations has similarities with the physics of strongly correlated electron systems near the Mott-Hubbard transition [194]. The analogy is even closer with frustrated magnetic systems, that precisely for that reason have been dubbed spin ices, *i.e.* materials in

which spin fluctuations are hindered by constraints equivalent to the ice rules [40]. Correlations in these systems are usually investigated within coarse grained models that deal with a restricted set of dynamical variables and typically neglect coupling to the lattice. We use instead a realistic model that includes all the degrees of freedom and treats thermal and quantum fluctuations on equal footing. As a downside we face severe limitations on the size of the simulation cell and could only perform few simulations at different lattice parameters. In spite of these limitations some general qualitative effects emerged from the simulations: (i) correlations are important whenever tunneling occurs; (ii) concerted ring tunneling, i.e. the only process allowed by ice rules, does occur but less frequently than less extended collective motions indicating some violation of the ice rules (iii) fleeting ionized configurations do occur but with substantially less charge separation frequency than the mean field prediction.

The chapter is organized as follows. In Section 11.2 we discuss the three high pressure ice systems under study and the simulation details, and then present the results of these computations in Section 11.3. In Section 11.4 we discuss the evidence supporting separability of the equilibrium dynamics along the bond from that in the plane perpendicular to the bond. We also report a singular value decomposition of the density matrix $\rho(x, x')$, yielding occupation numbers and eigenstates. Pure and mixed states of the proton are discussed in this context, and an analysis is presented to show that thermal effects are negligible compared to correlation effects in both HBHB and LBHB cases. In Section 11.5 the correlations are discussed with reference to local charge neutrality and population of ionized defects, for which simulation results and mean field values are compared. We also discuss in this section the statistics of the fluctuating H bond patterns along closed hexagonal rings in the ice VII simulation, showing substantial deviation from the mean field picture. In Section 11.6 we discuss how space and end-to-end distributions can be mimicked by an ensemble of effective bond potentials for the proton. This analysis leads to a new picture of the sequence of

H bond symmetrization transitions. Finally, in Section 11.7 we conclude with general considerations and comment on the consequences of the present findings in the more general context H bonded systems. Materials in this chapter have been presented in [168, 190].

11.2 Simulation details

Ice possesses a rich phase diagram. At ambient pressure and below 0°C, water is most stable in a hexagonal crystal structure. This is the form of ice for which the momentum distribution has been previously studied [47, 191]. However, under conditions of very high pressure, individual water molecules are arranged in inter penetrating cubic hydrogen bonded lattices. This arrangement forms an effective body centered cubic (BCC) lattice structure. The structure and the momentum distribution of ice Ih has been previously studied [191]. In this study, we will concentrate on three phases, Ice VII, VIII, and X. Ice VIII is proton ordered, exhibiting an anti-ferroelectric hydrogen bonding pattern. In comparison, Ice VII is proton disordered. Under higher pressures, the oxygen-oxygen distance reduces to the point where the proton's most stable position is equidistant between oxygen atoms and is located at the midpoint of the hydrogen bond axis. This "symmetric" form of ice is known as ice X [148, 150].

The work of Benoit and Marx [27, 181] has shown that by varying the lattice parameter (which changes the volume, and is equivalent to a change in pressure) of an Ice VIII cell one may, after a suitable equilibration period, generate Ice VII and Ice X. In the case of Ice VII, the system will tunnel through the barrier along the hydrogen bond axis, thereby disrupting the proton ordering in the system. At even smaller volumes, Ice X becomes thermodynamically favored. A schematic is provided in Figure 11.2 that illustrates these concepts.

Presently, we consider a $2 \times 2 \times 2$ BCC supercell containing 16 water molecules

at three different lattice constants. The lattice constants, as well as the corresponding molar volumes, pressures, and most probable oxygen-oxygen nearest neighbor distance are given in Table 11.1. The approximate pressures are garnered from the equation of state given by Hemley *et al.* [123].

The first principles open path methodology is employed in order to generate the trajectories. After an equilibration of 4 ps, each system is simulated for 75 ps, with the exception of System 2, which is sampled for 120 ps. A time step of 0.0725 fs is employed in all simulations. Each system is sampled at 100K. The temperature is controlled by means of massive Nose-Hoover chain thermostats [127, 180, 196]. Each path contains 32 replicas. The electronic states are evolved utilizing the Car-Parrinello methodology [49] with a fictitious mass of 340 atomic units. The electronic structure is described via the Kohn-Sham formulation of Density Functional Theory [143] where exchange and correlation effects are treated by the BLYP functional [22, 149]. The valence orbital are expanded in a plane wave basis set with a cutoff of 75 Rydberg. Troullier-Martins norm-conserving pseudopotentials [241] are utilized to model the valence effects of core electrons. The dynamical masses associated with the staging coordinates are set to be a factor of 4 larger than the staging masses.

Despite the small number of water molecules in the simulation cell, there are 2048 electron states (32 replicas $\times$ 16 molecules $\times$ 4 states per molecule) present in the system. This is a relatively large system by the standards of first principles simulation, and only state-of-the-art computational resources make possible the calculation of the relatively long trajectories and multiple systems reported in this study. All computations are performed on IBM Blue Gene/L hardware with the CPMD program, which has been optimized for this architecture.

System Number	Lattice Constant	Molar Volume	Approx. Pressure	d_{OO}^{mp}
1	2.67 Å	5.74 cm^3/mol	90 GPa	2.31Å
2	2.84 Å	6.90 cm^3/mol	45 GPa	2.45Å
3	2.94 Å	7.62 cm^3/mol	31 GPa	2.53Å

Table 11.1: Characteristic values that relay the size of each 16 molecule high pressure ice cell are given in the table above. The pressure is approximated from the equation of state given by Hemley *et al.* [123] The value of d_{OO}^{mp} is the most probable oxygen-oxygen distance between nearest neighbor, hydrogen bonded molecules.

11.3 Simulation results

The distributions in position and momentum space are computed in each system. Open paths are utilized for the computation of the momentum distribution, and closed paths are appropriate for the position distribution. Since our simulation contains both open and closed paths, we are able to use the closed paths for position distributions, and the open paths for the computation of the momentum distribution. The nature of this system in position space has already been elucidated in previous studies. Here we repeat this work in order to explore the relation between the position and momentum space distributions.

In Figure 11.3, the first peak of the oxygen-oxygen radial distribution function is shown. Shortening of the oxygen-oxygen distance is apparent as the molar volume is decreased. The position of the first peak of each distribution is reported in Table 11.1. Although there is roughly two-tenths of an angstrom difference between oxygen-oxygen distances of Systems 1 and 3, this has a dramatic impact upon the nature of the proton that is confined on the potential energy surface. It is this shortening that drives the phase transition between the forms of ice under study.

The position space distribution of the proton along the oxygen-oxygen hydrogen bond axis is illustrated by the oxygen-hydrogen radial distribution functions (Figure 11.4) and the probability distribution of the proton position along the hydrogen bond

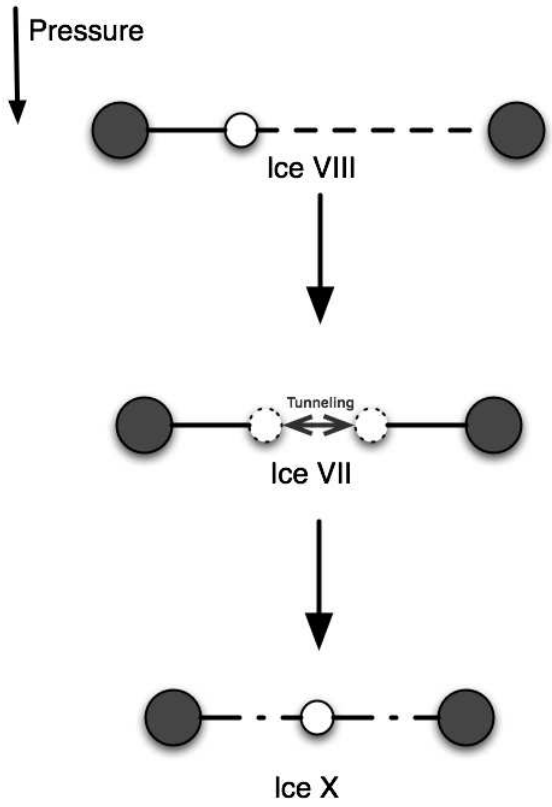


Figure 11.2: A schematic of the atoms involved in a single hydrogen bond in the three high pressure ice phases presently under study. The gray circles represent oxygen atoms and the white circles represent hydrogen. As the pressure upon the system increases the average oxygen-oxygen distance decreases, which has important consequences for the state of the proton. This may be covalently bonded (Ice VIII), tunnel between wells (Ice VII) or lie in a symmetric state between the oxygen atoms (Ice X).

axis (Figure 11.5). It can be seen that the proton in System 3 remains covalently bonded to its oxygen, although the covalent bond distribution is broader than in typical water phases. This system retains the Ice VIII structure. It can be seen in Figure 11.4 that the covalent bond and hydrogen bond peaks of the radial distribution function of System 1 merge. This broad single peak located at the midpoint between the two oxygen atoms is indicative of a symmetric hydrogen bond as found in the Ice X phase.

Evidence of quantum tunneling can be seen in System 2. The bimodal nature of

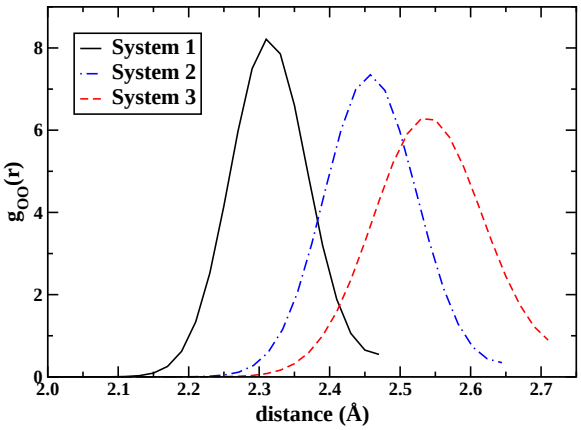


Figure 11.3: The first peak of the oxygen-oxygen radial distribution function in System 1 (solid curve), System 2 (dot-dashed curve) and System 3 (dashed curve). As one would expect, as the molar volume is decreased, the nearest neighbor oxygen-oxygen distance is as well.

the proton distribution in Figure 11.5, as well as the fact that one peak is near the covalently bonded peak of System 3 indicates that there are tunneling events from one well to another along the hydrogen bond axis. It was shown in the work of Benoit and Marx [27,181] that classical protons at this molar volume and temperature do not cross the barrier and remain trapped in a single well. This calculation showed that thermal hopping over the barrier is disfavored and quantum tunneling dominates. As noted in Section 11.2, the tunneling disrupts the anti-ferroelectric ordering and engenders the formation of Ice VII. We note that the bimodal distribution in Figure 11.5 is not perfectly symmetric. This may be caused by insufficient sampling or asymmetries that arise from correlated proton motions.

We note that the present distributions are somewhat more delocalized when compared with the work of Benoit and Marx [27, 181]. This is likely a result of the use of a larger number of replicas in the present computation. However there are many other differences in the details of the simulation that may impact this result, including trajectory length and the choice of exchange-correlation functional. Overall however, the description of the proton in position space along the hydrogen bond axis is in

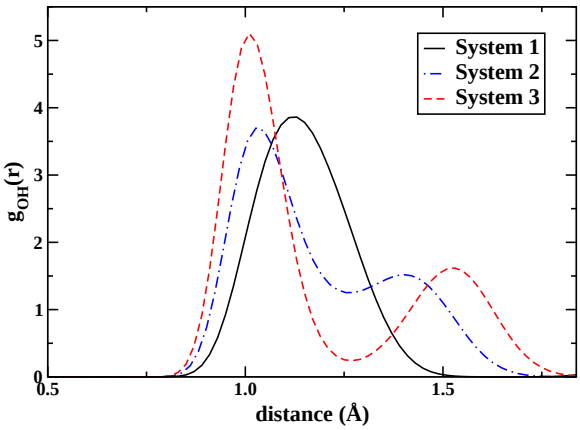


Figure 11.4: The oxygen-hydrogen radial distribution function in System 1 (solid curve), System 2 (dot-dashed curve) and System 3 (dashed curve). Whereas in System 3 there is a distinction between covalent and hydrogen bonding distances, the two peaks have merged in System 1.

good agreement with this and later work.

The momentum distribution is plotted along the oxygen-oxygen axis (Figure 11.6), as well along the two corresponding perpendicular axes (Figure 11.7). These are effective one-dimensional plots that are computed via the Fourier transform of the path end-to-end distance distribution along these directions. In Figure 11.7, one can view a trend that the momentum distributions in the directions perpendicular to the hydrogen bond broaden with decreasing system molar volume. This is consistent with the uncertainty principle given that as the protons become more confined in position space, the corresponding momentum distributions have a greater variance. Aside from this difference, there is little distinction between the systems under study in these directions when compared to the momentum distribution projected onto the hydrogen bonding axis (see Figure 11.6). This is a logical conclusion as the large qualitative differences in position space occur in the hydrogen bond direction (see Figure 11.5), as shown presently and in previous work on high pressure ice.

One may also note in Figure 11.7 that the distributions are similar along the two directions perpendicular to the hydrogen bond axis. This chemically intuitive result

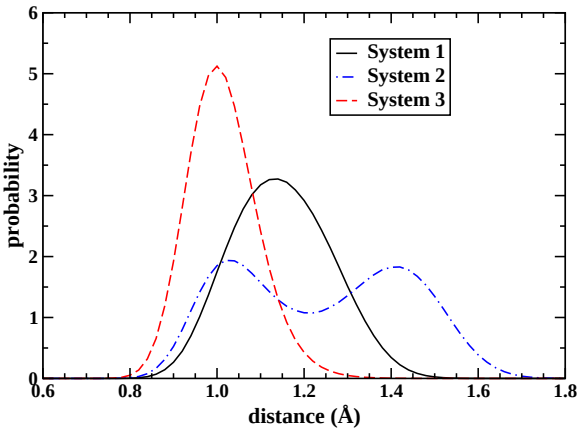


Figure 11.5: The distance distribution of the proton along the oxygen-oxygen direction in System 1 (solid curve), System 2 (dot-dashed curve) and System 3 (dashed curve). This direction is analogous to the hydrogen bonding axis. One may note that the distribution of System 2 is delocalized across two wells.

.

is in agreement with a previous study of the "shape" of the proton high pressure ice phases [26], where it was found that the position space distribution in the perpendicular directions were of similar extent. In addition, the proton's variance in the perpendicular directions was shown to decrease with increasing pressure , thereby providing complementa discussed above.

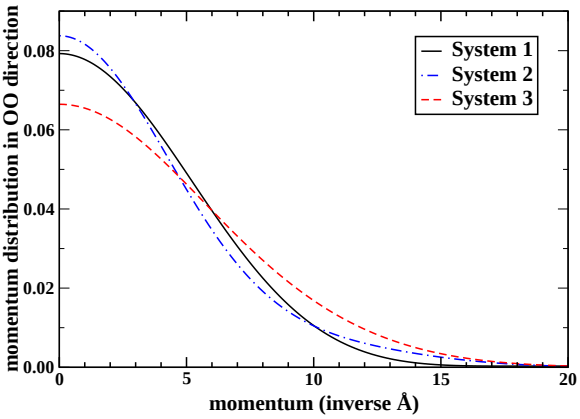


Figure 11.6: The proton momentum distribution in the oxygen-oxygen (OO) direction in System 1 (solid curve), System 2 (dot-dashed curve) and System 3 (dashed curve). It is in this orientation that the distinctions between phases occur.

The position space distributions show that System 1 contains symmetric hydrogen

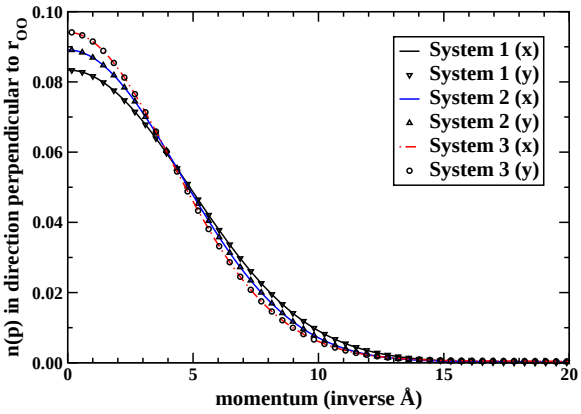


Figure 11.7: The proton momentum distribution perpendicular to the oxygen-oxygen direction (denoted "x") in System 1 (solid curve), System 2 (dot-dashed curve) and System 3 (dashed curve). Also plotted are the proton momentum distributions in the mutually orthogonal direction (denoted "y") in System 1 (triangles pointing downward), System 2 (triangles pointing upward) and System 3 (circles). The differences in widths of these curves indicates the relative pressure upon each system.

bonds, System 2 exhibits a bimodal proton distribution and in System 3, the protons are covalently bonded. In Figure 11.6, we present the momentum distributions in the hydrogen bonding direction. The covalently bonded System 3 possesses the narrowest position distribution (see Figure 11.5) and therefore the correspondingly broadest momentum distribution. The high-momentum tail of this distribution is dominated by the OH stretch (see Section 11.1). In the symmetric hydrogen bonded case (System 1), the more delocalized bond yields a narrower momentum distribution with a shortened tail. This signature in proton momentum distributions corresponds to a red-shift of the OH stretching frequency in stronger hydrogen bonded environments. This has been observed previously in the experimental [218] and simulation momentum distribution of liquid water and hexagonal ice [191] . The symmetric hydrogen bond may be considered the "strongest" class of hydrogen bonding. Such an interpretation is bourne out by experiments on symmetric hydrogen bonds observed in water confined in nanotubes [90] and $Rb_3H(SO_4)_2$ [126] that exhibit greatly narrowed momentum distributions with shortened tails.

The shape of the proton momentum distribution in the tunneling direction in System 2 lends to a more detailed description. It appears to have an anomalous shape when compared to the other distributions. Namely, it is narrow at low momentum, yet its tail behavior is similar to that of the covalently bonded System 3. This tail behavior is likely engendered by the localization in the covalently bonded well that is a component of the tunneling system. Therefore the highest frequency components of the system are similar to those exhibited in System 3. The narrowness exhibited in the low-momentum region is related to the overall delocalized nature of the proton.

11.4 Reduced longitudinal model

In this section we study how the correlations among the protons are affected by a changing environment. Our study is greatly facilitated by the fact that, within the statistical accuracy of the data, the density matrix factorizes into a longitudinal and a transverse component relative to the bond, *i.e.* $\rho(\vec{r},\vec{r'}) \sim \rho(x,x')\rho(\vec{b},\vec{b'})$. In other words position and end-to-end distributions along the bonding direction and in the plane orthogonal to it are mutually independent. The separation between bonding and transverse directions is valid in the sense of Spearman's rank correlation coefficient R.

Here we give a short introduction to the test of independence using Spearman's rank correlation coefficient and apply it to illustrate the separation of the potential energy surface along and perpendicular to the hydrogen bonding direction for Ice Ih, VIII, VII and X. Consider a series of observation points $(X_i, Y_i)_{i=1}^N$ coming from a two dimensional continuous random variable (X, Y). In this case X is the end-to-end distance along the hydrogen bonding direction and Y is the end-to-end vector projected on the plane orthogonal to the hydrogen bonding direction. If X and Y are independent, the effective potential experienced by the proton should separate along

hydrogen bonding direction and orthogonal direction. We sort $\{X_i\}_{i=1}^N$ in ascending order as $X_{S_1}, \ldots, X_{S_N}$, and S_i is called the rank of X_i. Similarly we define T_i to be the rank of Y_i. Spearman's rank correlation coefficient is defined to be the correlation between the pairs $(S_i, T_i)_{i=1}^N$ *i.e.*

$$R = \frac{\sum_{i=1}^N (S_i - \bar{S})(T_i - \bar{T})}{\sqrt{\sum_{i=1}^N (S_i - \bar{S})^2 \sum_{i=1}^N (T_i - \bar{T})^2}}, \tag{11.1}$$

with

$$\bar{S} = \bar{T} = \frac{N+1}{2}. \tag{11.2}$$

Spearman's rank correlation coefficient can be expressed in the more convenient form as

$$R = \frac{12}{N(N+1)(N-1)} \sum_{i=1}^N S_i T_i - 3\frac{N+1}{N-1}. \tag{11.3}$$

The advantage of Spearman's rank correlation coefficient is that it is based on the rank of X and Y rather than on the detailed behavior of these random variables. As a result, Spearman's R can be used as a test of independence without assuming that (X, Y) follows a Gaussian distribution. This is particularly useful in the present context because we know that anharmonicity exists at least along the hydrogen bonding direction, and the end-to-end distribution along the hydrogen bonding direction is not Gaussian. Furthermore, $R = \pm 1$ occurs only if X and Y are functionally dependent on each other. This dependence can be linear or non-linear. If Spearman's rank correlation coefficient is 0 then X and Y are independent.

The Spearman's R calculated for the end-to-end distances along and orthogonal to the hydrogen bonding direction in Ice Ih, VIII, VII and X is listed in Table 11.2. The Spearman's R is universally small across the different ice phases, strongly supporting factorization of the density matrix. In the case of ice Ih, factorization could be expected because in a recent work we have shown that the momentum distribution

has an effective quasi-harmonic (Gaussian) form with 3 principal frequencies associated to longitudinal and transverse motions, respectively, relative to the bond [167]. The three high pressure phases considered here are characterized by much stronger anharmonicity along the bonding direction, well beyond the quasi-harmonic model. It is interesting that even in these cases quantitative analysis shows that longitudinal-transverse decoupling holds. It means that anharmonicity remains confined in the longitudinal direction in accord with simple chemical intuition.

System	R
Ih	0.029
VIII	0.032
VII	0.027
X	0.025

Table 11.2: Spearman's rank correlation coefficient for the end-to-end vector distance along and orthogonal to the hydrogen bonding direction in ice Ih, VIII, VII and X.

Let $\rho(x, x') = \langle x \,|\rho|\, x' \rangle$ be the normalized longitudinal density matrix, *i.e.* $\mathrm{Tr}[\rho] = 1$. If $P(i)$ denotes the eigenvalues of ρ and $|\phi(i)\rangle$ are the corresponding eigenvectors, one has:

$$\rho = \sum_i |\phi(i)\rangle P(i) \langle \phi(i)| \tag{11.4}$$

Here $P(i)$ is the equilibrium population (occupation probability) of the state $|\phi(i)\rangle$ and $\sum_i P(i) = 1$. The spectrum of the density matrix is very instructive. If only one eigenvalue, $P(1)$, is different from zero, the density matrix is idempotent *i.e.* $\rho^2 = \rho$ and the equilibrium ensemble of the proton along the bond is pure, corresponding to the quantum state $|\phi(1)\rangle$. If more than one eigenvalue is different from zero, the density matrix deviates from idempotency, *i.e.* $\rho^2 < \rho$ and the equilibrium ensemble of the proton is mixed.

We bin the PICPMD simulation data for the longitudinal density matrix with a spacing $\Delta x = 0.015$Å and the corresponding discretized density matrix is diagonal-

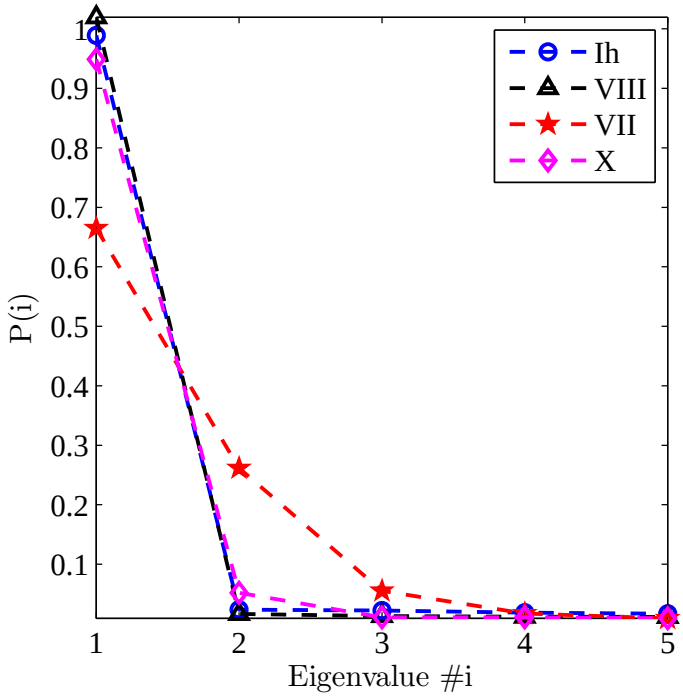


Figure 11.8: The first 5 eigenvalues of the longitudinal density matrix for ice Ih, VIII, VII and X. Within the accuracy of the simulation, $P(1) = 1$ for ice Ih and ice VIII, $P(1)$, $P(2)$, and $P(3)$ are different from zero for ice VII, $P(1)$ and $P(2)$ are different from zero for ice X. The ice Ih trajectory is shorter and the corresponding eigenvalues are affected by larger error bars than the three high pressure phases.

ized. The largest 5 eigenvalues of the longitudinal density matrix in ice Ih, VIII, VII and X are reported in Fig. 11.8. We find that both ice Ih at $T = 269$ K and ice VIII at $T = 100$ K have idempotent density matrix within the statistical errors of the simulation, indicating that the proton is in a pure quantum state. Thus, in these ices the proton motions along the bond are on average uncorrelated and the systems are ground state dominated at their respective temperatures. This is not surprising as the stretching motions in standard H bonds have typical frequencies around 3000cm^{-1} corresponding to a zero point energy much larger than $k_B T$. Interestingly, in the case of ice VII at $T = 100$ K, which is in the HBHB regime, we find large deviations from idempotency. Deviations from idempotency are significantly reduced but still visible in ice X, which is in the LBHB regime. Deviation from idempotency indicates ice VII

and X are not in a pure but in a mixed quantum state. Mixed state character of the longitudinal density matrix $\rho(x, x')$ may result from thermal or correlation effects or from a combination of both. Our analysis shows that correlations dominate in both ices at $T = 100$ K.

To infer the relative importance of thermal and correlation effects we adopt a two state model. This is a good approximation given the small value of the third eigenvalue, which is nonzero only in ice VII. In this ice we take $P(1) = .72$ and $P(2) = .28$ by renormalizing to 1 the sum of the first two eigenvalues in Fig. 11.8. We then consider an effective single particle 1D Hamiltonian that includes a kinetic energy term (with the proton mass) and a quartic double well potential term. We optimize the potential so that the 1D Hamiltonian has the two populated eigenvectors of the reduced density matrix as its two lowest energy states. The optimal potential is depicted in Fig. 11.9 and is given by

$$V(x) = 0.1100x^4 - 0.0475x^2 + 0.0051, \tag{11.5}$$

where we added a constant term to shift the minimum of the potential wells to 0 .

Fig. 11.10 compares the eigenvectors of the density matrix (labeled "Raw") and the eigenvectors of the quartic potential with optimal parameters (labeled "Fit"). The first two singular vectors of the density matrix are indeed very well approximated by the optimal potential.

The energies of the two lowest eigenvalues of the 1D Hamiltonian are given in Fig. 11.9. The corresponding tunneling splitting, 547K, is much larger than the equilibrium temperature of the simulation (100K), indicating that thermal effects are not the major cause of the mixed character of the equilibrium proton ensemble. [1] Notice that the barrier height ($1614K$) is well above the ground state energy

[1] This result is entirely consistent with a previous study [190] that position and end-to-end distributions of the proton in ice VII could be modeled by an effective potential only by invoking a temperature significantly higher than the temperature of the simulation.

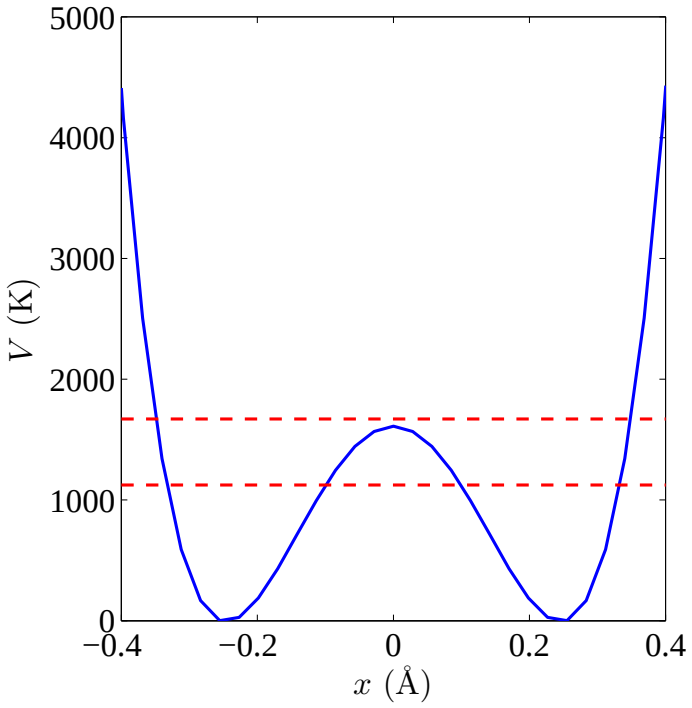


Figure 11.9: The optimized quartic double well potential that reproduces the lowest two states of the longitudinal density matrix. The horizontal dashed lines indicate the ground and the first excited state of this potential, respectively.

(1124K), confirming the HBHB character of ice VII. The analysis does not change appreciably if we include the third populated eigenstate of the density matrix: in that case the same procedure gives a tunnel splitting of 548K. The potential in Fig. 11.9 can be viewed as a mean field potential for the proton: it is qualitatively (and even quantitatively) very similar to the mean field potential that was suggested for a tunneling proton in Ref. [235] on the basis of phenomenological considerations. In the mean field approximation each proton moves in a self-consistent potential that includes the effect of the other protons in an average way. The fact that a system which is ground state dominated is not in the ground state of the mean field potential indicates that correlations, neglected in the mean field approximation, are important. These correlations reflect the ice rules that act to retain the integrity of the water molecules, *i.e.* the local charge neutrality, when the protons tunnel between the two

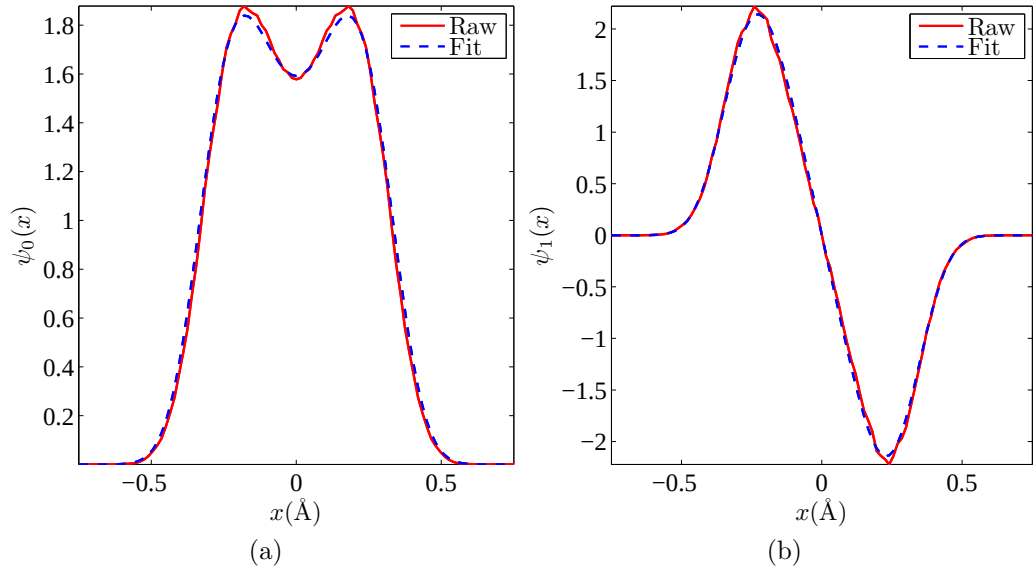


Figure 11.10: (a) The largest singular vector of the longitudinal density matrix (red solid line) and the ground state of the quartic potential in Eq. (11.5) (blue dashed line). (b) The second largest singular vector of the longitudinal density matrix (red solid line) and the first excited state of the quartic potential in Eq. (11.5) (blue dashed line).

sides of a bond.

The pure versus mixed state character of the equilibrium ensemble is well characterized in terms of the entanglement (or von Neumann) entropy. This is defined as $S = -\,\mathrm{Tr}[\rho \log \rho]$. Using the eigenvalues reported in Fig. 11.8 we find that S is essentially 0 in ice Ih and VIII, indicating pure state character. On the other hand, $S = 0.60$ in ice VII and $S = 0.20$ in ice X, indicating mixed state character in both cases but with a larger entanglement in ice VII than in ice X.

11.5 Proton correlations

Proton correlations originate from the requirement of local charge neutrality that favors intact water molecules over ionized configurations such as H_3O^+ and OH^-. These effects can be quantified by monitoring the deviation from local charge neu-

trality at each oxygen site in terms of the centroids of the Feynman paths for the protons. According to the position of the centroid along a bond we may assign the corresponding proton to either one of the two oxygens linked by the bond. We call the two corresponding proton sites N (near) and F (far), respectively, relative to one of the oxygens. Local charge neutrality demands that each oxygen has two N and two F protons on the four bonds with the neighboring oxygens: this is called the ice rule [29, 203]. In our simulation the rule is obeyed with probability $P = 1$ in ice VIII and ice Ih, but we find $P = 0.85$ in ice VII. This is to be compared with the mean field value $P_{MFA} = 0.375$ [235] in which protons occupy randomly the two sites. The large discrepancy between simulation and mean field theory, underlines the important role of correlations which oppose the formation of ionized species. The above discussion corresponds to a two-state model for the proton. We find, however, that a three-state model, in which the proton can occupy three distinct sites, N, F, and C(center) represents the simulation data more accurately. A path with a centroid near the bond center (C) is delocalized on the bond and the corresponding proton can be assigned with equal weight (1/2) to two oxygens. The boundaries of the respective states are defined according to regions of the centroid distribution projected along the bond. Although the precise definition of the boundaries is not unique the physics is consistent for reasonable choices. In this model local charge neutrality is satisfied not only by configurations like $NNFF$ but also by configurations such as $NFCC$ and $CCCC$, which would violate the strict ice rule. In the simulation we find that C states occur with approximately half the frequency of either N or F states. Local change neutrality is satisfied with $P = 0.75$ as compared with a mean field value $P_{MFA} = 0.23$, indicating that the fractional deviation of the simulation from mean field is larger in the three-state than in the two-state model. Furthermore, the distribution of charged species is far narrower in our results as is illustrated in Fig. 11.11. In particular we find that in $> 99\%$ of species that violated charge neutrality

there is a net charge of only ± 0.5, indicating that fully ionized species such as H_3O^+ are exceedingly rare.

One mechanism for proton tunneling is the creation of short-lived ionic configurations. This is in contrast to concerted proton tunneling along the closed water rings that comprise the crystal structure. In our simulation ice VIII and VII have BCC structure, i.e. we neglect a small tetragonal distortion in ice VIII. The BCC structure is made of two interpenetrating FCC (Ic) sublattices. Only 6-membered rings, i.e. the shortest closed loops of connected bonds, are contained in the 16-molecule simulation cell. There is one such ring per FCC sublattice. As elsewhere in this work, we adopt the three-state model discussed above. The convention presently used is that the near side is located clockwise to the left side of the bond. In ice VIII, the anti-ferroelectric arrangement of the two proton ordered sub lattices yields proton patterns with three consecutive N and three consecutive F states that are anti-correlated with respect to each other in the two rings. In ice VII, quantum fluctuations disorder the patterns but in the simulation there is still residual anti-correlation among the two sublattices. This result is due to the finite size of the cell and is in agreement with the findings of Ref. [27]. The probability of a ring configuration possessing a consecutive block of N, F, or C states is given in Fig. 11.12. This is contrasted to the distribution one would attain if the proton states along the bonds were randomly distributed with probabilities $P_N = P_F = 2P_C$, as predicted by a mean field model. We find that longer "blocks" are favored in the simulation in contrast to the random case. Notice that the very small size of the simulation cell prevents us from studying quantitatively the spatial extent of proton correlations, but Fig. 11.12 suggests that concerted jumps on rings longer than 6-fold should be present in the real crystal. At the same time the figure shows that the number of configurations that correspond to concerted ring tunneling i.e. $NNNNNN$, $FFFFFF$, or $CCCCCC$ comprise less than 10% of configurations. This finding indicates that mechanisms which violate local charge

neutrality do occur, albeit to a much smaller degree than in the mean field model and

cc

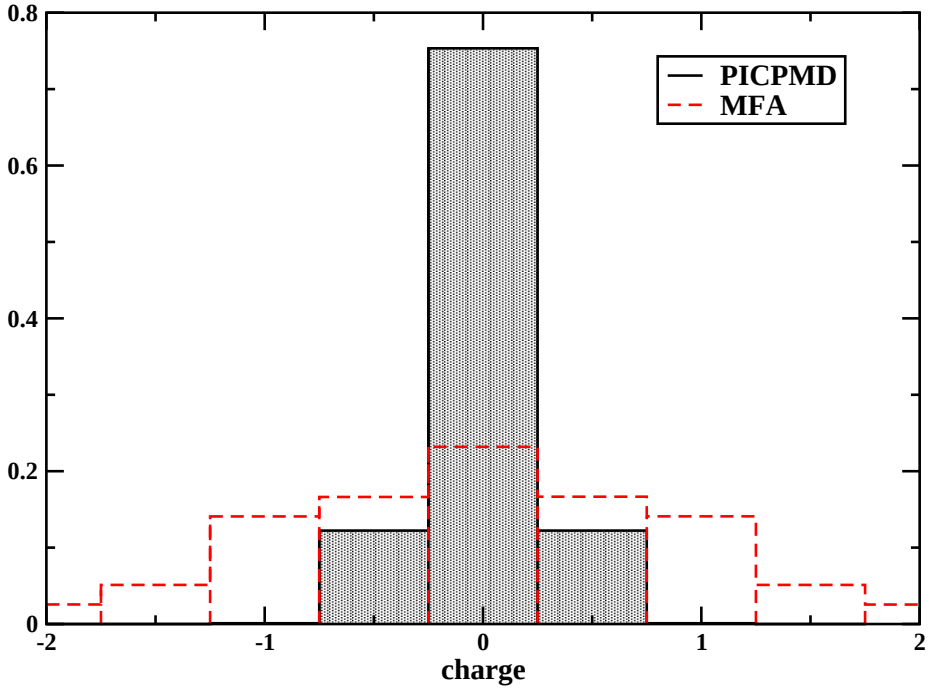


Figure 11.11: The distribution of local charge density in Ice VII according to the 3-state model discussed in the text (gray bars). This result can be seen to be in stark contrast to the randomly distributed set of charged species predicted by the mean field approximation (dashed, red bars).

11.6 Effective proton potential

The environmental changes occurring in the H bond symmetrization process are well illustrated by introducing the effective potential that, in the 1D Schroedinger equation for the proton dynamics along the bond, gives a ground state solution that simultaneously fits the position and end-to-end distributions. This is possible in ice Ih and VIII as these systems are ground state dominated.

Both ice Ih and VIII exhibit "typical" hydrogen bonding and we utilize the fol-

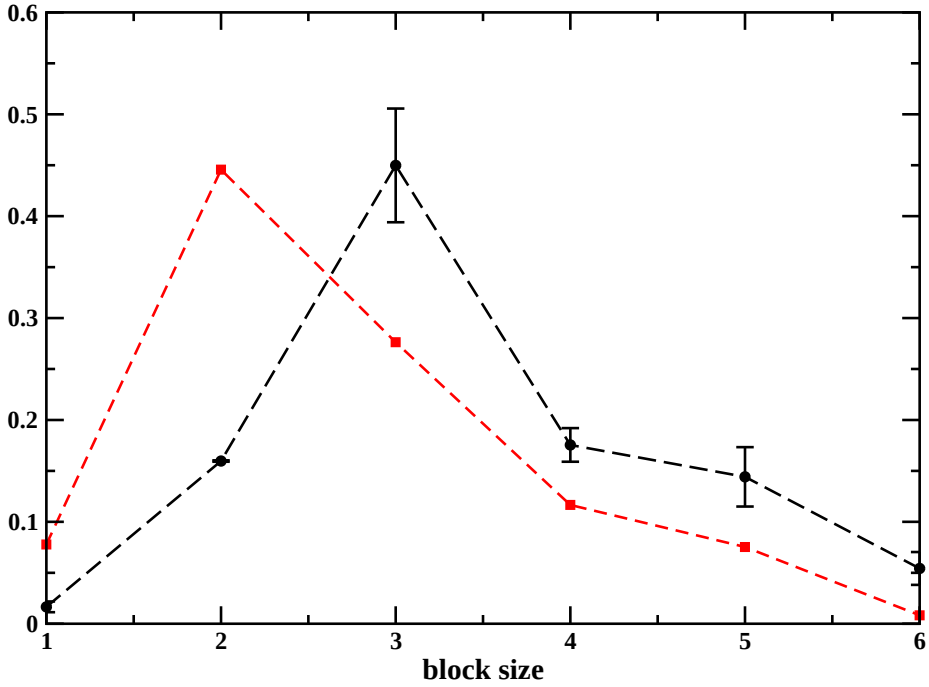


Figure 11.12: The probability of a ring configuration having a consecutive block of N, F, or C states of length L (black dashed line with circles and error bars). The red dashed line with squares is the resultant random distribution where the probability of drawing an N or F on a bond along the ring is twice that of drawing a C.

phase	a_2	a_3	x_0
Ih	15450	-30666	0.011
VIII	19143	-49764	0.016

Table 11.3: Parameters for the cubic potential in Eq. (11.6) for ice Ih and ice VIII. a_n is given in meV/Å^n and x_0 is given in Å.

lowing functional form for the effective potential:

$$V(x) \quad = \quad a_3(x - x_0)^3 + a_2(x - x_0)^2. \tag{11.6}$$

The parameters for the fits are given in Table 11.3. The corresponding potentials are given in Fig. 11.13 and the resulting position and end-to-end distributions are compared with the raw data in Fig. 11.14.

In ice Ih and VIII the proton is localized in a covalent well and the model potential

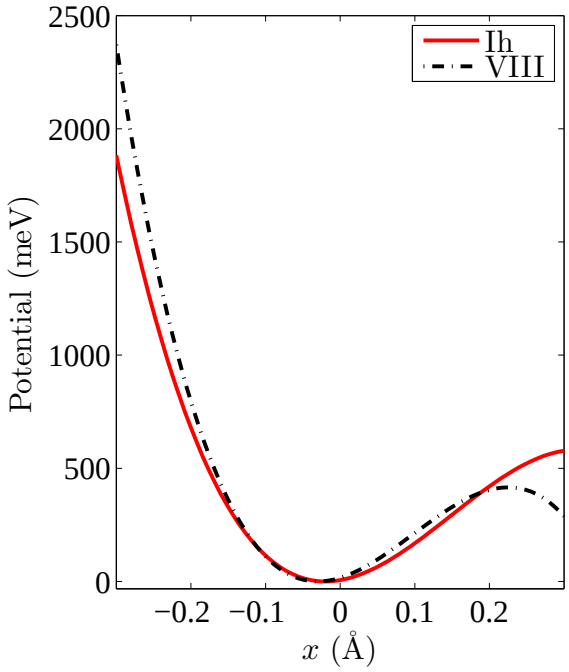


Figure 11.13: The effective cubic potential for ice Ih (red solid line) and ice VIII (black dashed line) along hydrogen bond direction.

includes harmonic confinement and an anharmonic cubic correction. One may notice that in ice VIII the spatial distribution is narrower near the maximum and at the same time spills out more towards the distant O. Overall, however, the proton is more confined in ice VIII than in ice Ih. The potential in Fig. 11.13 reflects this behavior and shows clearly greater anharmonicity (and skewness) in ice VIII than in ice Ih.

Interestingly, in ice VII no unique potential that simultaneously fits the position and end-to-end distributions can be found, even though the system remains ground state dominated. As explained in Sects. 3 and 4 this is because the proton is in an entangled state reflecting short range correlations with the other protons that originate from the requirement of local charge neutrality. The tunneling proton is well described in terms of a three-state model corresponding to three distinct positions of the centroid of the Feynman path along the bond: N, C, and F (see Sect. 4). This suggest that position and end-to-end distributions should be fitted using a mixed ensemble in which three distinct potentials are used to model the 3 states of the

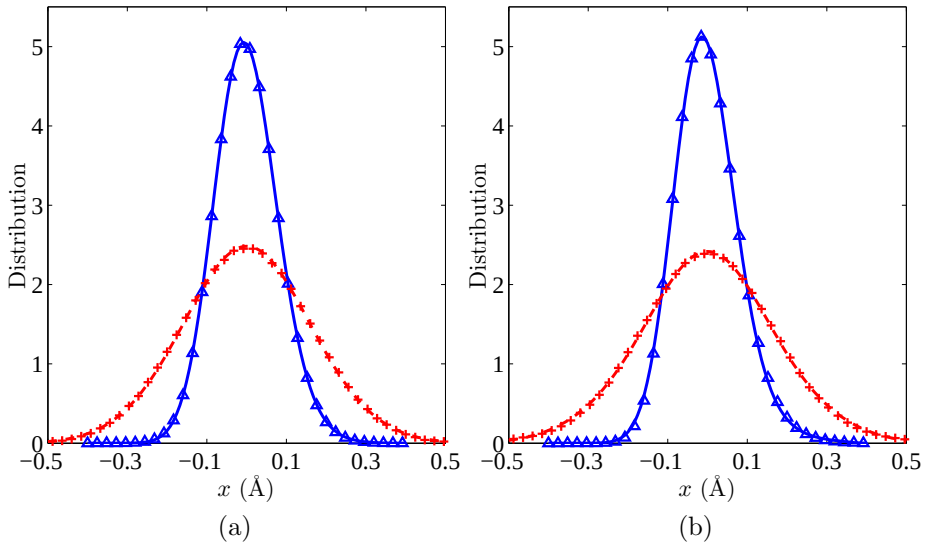


Figure 11.14: (a) The position distribution of ice Ih obtained from the PICPMD simulation (blue solid line) and that reconstructed from the cubic potential (blue triangle), together with the end-to-end distribution of ice Ih obtained from the PICPMD simulation (red dashed line) and that reconstructed from the cubic potential (red cross); (b) The position and the end-to-end distributions in ice VIII. The legend is the same as in (a).

proton. For the three potentials we adopt the following form:

$$\begin{aligned} V_N(x) &= a_4x^4 + a_2x^2 + a_1x, \\ V_C(x) &= a_4x^4 + a_2x^2, \\ V_F(x) &= a_4x^4 + a_2x^2 - a_1x. \end{aligned} \tag{11.7}$$

Each potential i generates a position and end-to-end distribution, respectively. The total position and end-to-end distributions are given by the ensemble average

$$\begin{aligned} n(x) &= \omega_N n_N(x) + \omega_C n_C(x) + \omega_F n_F(r), \\ \widetilde{n}(x) &= \omega_N \widetilde{n}_N(x) + \omega_C \widetilde{n}_C(x) + \omega_F \widetilde{n}_F(x). \end{aligned} \tag{11.8}$$

with weights given by $\omega_F = \mu, \omega_C = 1-2\mu, \omega_F = \mu$, respectively. Using a, b, c, μ as optimization parameters, both position and end-to-end distributions can be accurately

reproduced as shown in Fig. 11.15 (b). The three types of potentials are depicted in Fig. 11.16. It is found that both the position and the end-to-end distributions are fitted accurately with $\omega_N = \omega_F = 0.40$ and $\omega_C = 0.20$, which is consistent with the analysis in Sect. 4. Notice, however, that the three corresponding states of the proton overlap and are not orthogonal: thus the three weights $\omega_N, \omega_C, \omega_F$ are not equivalent to the eigenvalues of the density matrix discussed in Sect. 3. V_N, V_F are tilted towards one or the other side of the hydrogen bond by the linear term in Eq. 11.7. This term has the effect of lowering the potential when the proton is on the bond side that maintains charge neutrality at the nearest oxygen site. Thus V_N, V_F break inversion symmetry relative to the bond center penalizing ionized configurations. V_C is a double well potential, with the potential barrier lower than the zero-point motion energy.

It should be noted that a two-state model for the proton and the corresponding two-state potential ensemble is sufficient to capture the main qualitative features of the position and end-to-end distributions, but the fit is less accurate than the one provided by the adopted three-state model. There are physical reasons for this finding. Fractional charge fluctuations are allowed in the 3-state model while only integer charge fluctuations are possible within the 2-state model. Fractional charge fluctuations minimize the deviation from local charge neutrality (see Sect. 4). Moreover the 3-state model mimics an effect of the coupling of the proton with the lattice: when the proton is in C the bond length is on average slightly shorter than when it is in N or F. A similar correlation was already reported in Ref. [27]: it indicates that quantum fluctuations couple with the geometrical structure, an effect that is quite important in KDP where it leads to the so-called Ubbelohde effect upon isotopic substitution [220].

Mixed state character is not as prominent but still noticeable in our ice X sample. In this case the best fit is provided by the same potential ensemble (11.7) with the

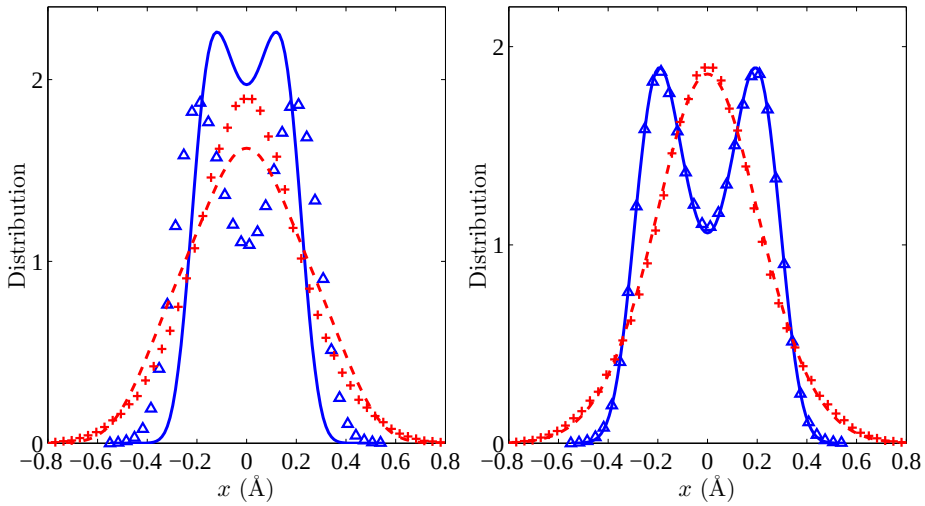


Figure 11.15: (a) The position distribution of ice VII obtained from the PICPMD simulation (blue solid line) and that reconstructed from a double well potential (blue triangle), together with the end-to-end distribution of ice VII obtained from the PICPMD simulation (red dashed line) and that reconstructed from the same double well potential (red cross). A unique potential cannot accurately fit position and end-to-end distributions of ice VII. (b) The position distribution of ice VII obtained from the PICPMD simulation (blue solid line) and that reconstructed from a three-state potential ensemble (blue triangle), together with the end-to-end distribution of ice VII obtained from the PICPMD simulation (red dashed line) and that reconstructed from the same three-state potential ensemble (red cross).

parameters given in the second row of Table 11.4. We find that position and the end-to-end distributions are fitted accurately with $\omega_N = \omega_F = 0.05$ and $\omega_C = 0.90$, as illustrated in Fig. 11.17. The proton in ice X is predominately in the C state, consistent with the LBHB character.

11.7 Conclusion

In this chapter we have presented an investigation of the position and momentum space distributions of the proton in tunneling and symmetric hydrogen bonded systems. Novel first principles open path integral molecular dynamics algorithms were utilized in order to compute the momentum distributions. Three phases of high pres-

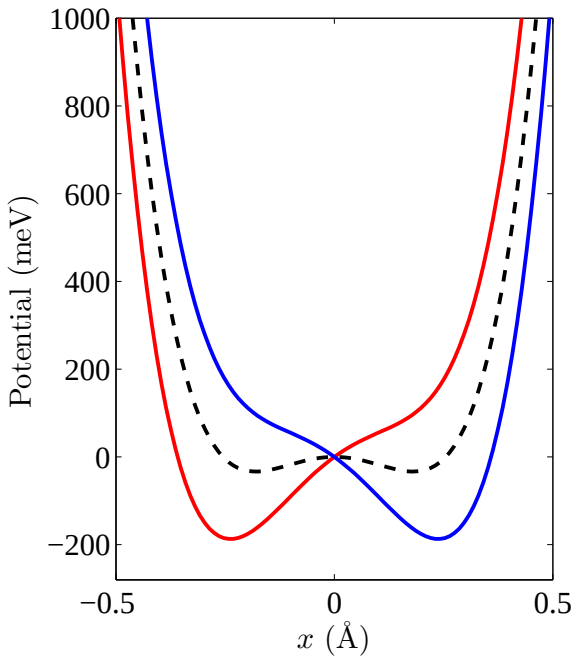


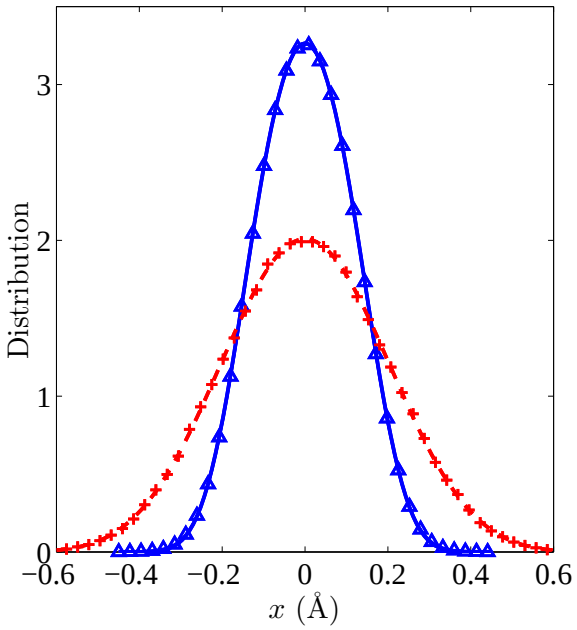
Figure 11.16: Three-state potential ensemble for ice VII. The two tilted potentials (black and red lines) have equal weights $\omega_N = \omega_F = 0.40$, the double well potential (dashed blue line) has weight $\omega_C = 0.20$.

phase	a_1	a_2	a_4
VII	725	-2038	31511
X	1645	-1263	8922

Table 11.4: Parameters for the three-state potential ensemble for ice VII and ice X. a_n is given in meV/Å^n.

sure ice were studied at 100K. Each phase typifies a qualitatively different state of the proton, covalently bonded (Ice VIII), tunneling (Ice VII), and equally shared between nearest-neighbor oxygens (Ice X).

We have quantified the role played by correlations in the proton disordering transition occurring when antiferroelectric ice VIII converts to paraelectric ice VII. At sufficiently low temperature this transition is driven mostly by quantum fluctuations that lead to tunneling and delocalization of the protons in the bonds that connect the oxygen vertices in the crystalline lattice. To analyze the PICPMD simulation data we used two concepts that are new in this context. We performed a spectral



(a)

Figure 11.17: The position distribution of ice X obtained from the PICPMD simulation (blue solid line) and that reconstructed from a three-state potential ensemble (blue triangle), together with the end-to-end distribution of ice VII obtained from the PICPMD simulation (red dashed line) and that reconstructed from the same three-state potential ensemble (red cross).

decomposition of the single particle density matrix, a property that is available in simulations that sample not only the spatial distribution of the quantum particles but also their momentum distribution, or equivalently the end-to-end distribution of the open Feynman paths. The spectral analysis of the density matrix allowed us to assess unambiguously the role of correlations by quantifying the entanglement of the proton state, its deviation from the prediction of mean field theory, and the characteristic energy scale of the entanglement, which turned out to be much larger than k_BT. Next, we monitored the centroids of the paths to study, in particular, concerted ring fluctuations of the centroids. This analysis allowed us to associate unambiguously proton correlations to local charge neutrality. The latter requirement generalizes the so-called ice rule due to Bernal, Fowler and Pauling [29, 203], which applies to coarse grained models with two proton sites on each bond.

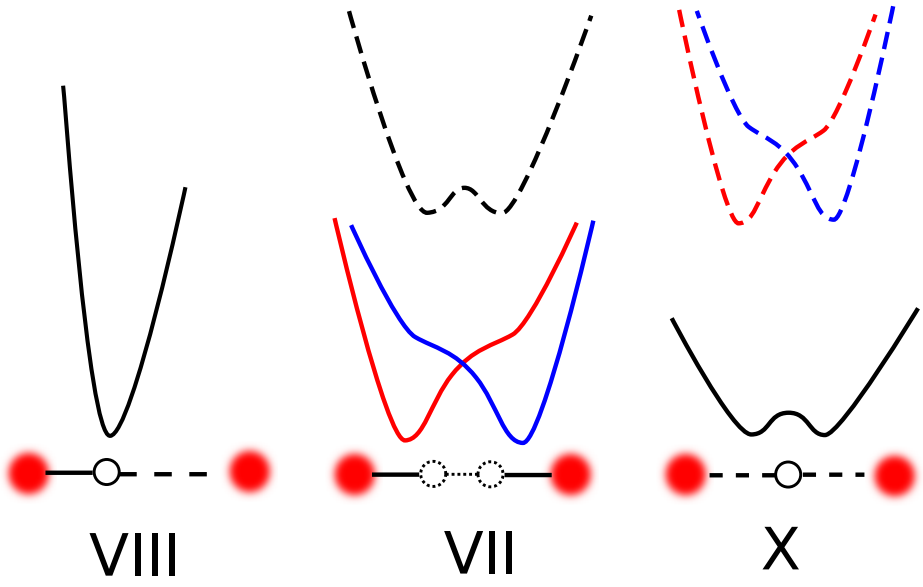


Figure 11.18: Classification of H bonds established in this chapter: The proton in ice VIII (and in ice Ih) is in a pure quantum state and experiences an asymmetric single well potential that keeps it localized on one side of the bond. The proton in ice VII (HBHB) and in ice X (LBHB) is in a mixed quantum state and experiences a potential ensemble that depends on its location on the bond. Dominant potentials are indicated by full lines and less dominant ones by dashed lines. The proton distribution is symmetric and bimodal in ice VII and symmetric and unimodal in ice X.

The standard picture used to interpret previous PICPMD studies of the H bond symmetrization transitions at high pressure was based on mean field theory and did not take into account the correlations present in the simulations. This picture is illustrated in Fig. 11.1: it assumes that in ice VII each proton tunnels coherently back and forth between two sites (N and F) on the opposite sides of a bond. This process, if random, would lead to a large number of ionized configurations, such as H_3O^+ and OH^- or H_4O^{++} and O^{--}, but this so-called ionization catastrophe [235] is penalized by the energy cost of dissociating the water molecules. To avoid this cost, concerted tunneling processes take place that reduce the number of ionized configurations. However, charged defects are not entirely suppressed as if only complete ring tunneling jumps were allowed. Because of correlations the state of the proton that at low enough temperature is essentially a pure quantum state in ice VIII, be-

comes a mixed quantum state in ice VII and to some extent as well in ice X, where charge fluctuations on the bonds are present and the system is in the so-called LBHB regime. The mixed state character can be described in terms of a potential ensemble, as illustrated in Fig. 11.18. The new picture provides a more accurate description of the H bond symmetrization transitions than Fig. 11.1.

Several questions remain open, such as whether or not concerted tunneling processes occur on rings longer than 6-fold with measurable frequency or how do collective fluctuations decay in space. Some answers to these questions may come from future simulations on larger cells. For instance, simulations on cells that are at least 8 times bigger than the present 16 molecule one would be feasible on modern supercomputers. These simulations could take advantage of algorithms like a newly proposed estimator of the end-to-end distribution [166] to improve accuracy and statistics. Other issues involve the entropy change between the ordered and the disordered phase. Overall it is important to understand the precise nature of the quantum many-body ground-state of the protons in the disordered phase and the nature of the corresponding excitation spectrum. Coarse grained models using a spin Hamiltonian to describe the proton system may be very useful in this respect and could benefit from input from realistic off-lattice simulations like the present one. For instance, the present simulation suggests that a spin 1 model should provide a more accurate representation of the protons in ice VII than a spin 1/2 model.

Finally, the present study has implications for other H bonded systems where proton tunneling occurs. Ring tunneling processes like those that we have observed here have been hypothesized to occur on ordered rings in ice Ih at low temperature to explain isotope effects in the quasi-elastic neutron peak [39] . In ice Ih the bond length is significantly longer than in ice VII and concerted tunneling should have a much lower frequency than in the present simulation. However, the system for which the present results has more direct implications is KDP, a H bonded molecular

crystal which is ferroelectric at very low temperature and undergoes a transition to a paraelectric phase driven by quantum fluctuations at $T_c = 121K$. The phosphate groups in KDP are in a local tetrahedral environment and are linked together by H bonds as the water molecules in ice. The ferro-to-para transition corresponds to disordering of the H sublattice. The processes that we find in each ferroelectrically ordered sublattice of ice VIII upon transition to ice VII at low temperature should have strong similarities with the quantum fluctuations that lead to the disordering transition in KDP.

Chapter 12

Conclusion of Part II

Part II of this dissertation introduces novel methodologies for the computation and the interpretation of the quantum momentum distribution in hydrogen bonded systems.

From computational point of view, the widely used open path integral formalism only allows the calculation of the quantum momentum distribution for one particle at a time. This constraint is removed by the displaced path integral formalism developed in Chapter 8, and the efficiency for calculating the momentum distribution is greatly improved. The classical contribution and the quantum contribution of the end-to-end distribution factorize in the displaced path integral formalism, which facilitates the interpretation of the momentum distribution. The displaced path integral formalism introduces a useful quantity called the mean force, which can be used as a non-parametric method to quantify the uncertainty in the Deep Inelastic Neutron Scattering experimental data. The displaced path integral formalism gives rise to a new semiclassical limit analysis of the quantum momentum distribution. Numerical result shows that this new semiclassical limit is more accurate than the isotropic semiclassical limit, and is able to capture the quantum momentum distribution quantitatively for a large class of systems. The displaced path integral formalism also provides a new kinetic energy estimator, and can be generalized to systems consisting

of indistinguishable particles. The practical implementation of the displaced path integral formalism for bosons will be the future work. Other directions within our scope are the application of the displaced path integral formalism to *ab initio* molecular dynamics simulation, as well as more efficient free energy sampling methods for the calculation of momentum distributions.

Part II of this dissertation also elucidates an important issue in the interpretation of the quantum momentum distributions, *i.e.* the relation between anisotropy and anharmonicity using ice Ih as an example in Chapter 9. This is achieved by a detailed analysis of the full 3D momentum distribution garnered from PICPMD simulation, as well as from the related vibrational dynamics. We find that to a large extent the momentum distribution in ice Ih is a simple anisotropic Gaussian distribution, and that the potential of the mean force can be well modeled by a quasi-harmonic model potential. The anisotropic principal frequencies in the potential of the mean force are the weighted average of the stretching, bending and the libration modes in the vibrational spectrum. Anharmonicity is particularly visible in the stretching mode, but is largely suppressed by anisotropy in the spherically averaged momentum distribution. This analysis is useful for the interpretation of the experimental data as illustrated in Chapter 10.

Finally Chapter 11 unambiguously assesses the importance of correlated proton tunneling in high pressure ice by means of spectral decomposition of the single particle density matrix. Concerted proton tunneling is directly observed and quantified in the simulation. Concerted proton tunneling reduces the number of ionized configurations compared to that in a complete ionization catastrophe, and thus partially restores the local charge neutrality. This dissertation demonstrates that the correlated character of proton dynamics can be described by an ensemble of potentials of the mean force, which is found to give accurate description of both the position and the momentum distribution of protons.

The analysis and methodology developed in this dissertation are quite general, and can be useful for further study of experimental and simulation results for the quantum momentum distribution in more complicated systems. To this end further work should be done. In order to apply the displaced path integral formalism to a larger class of systems in practice, modern technique to enhance the statistical sampling should be included. This work is currently in progress. The analysis of quantum effects in other challenging systems such as supercooled water and KDP etc. will also be studied in future.

Bibliography

[1] F.F. Abraham, D. Brodbeck, W.E. Rudge, and X. Xu, *A molecular dynamics investigation of rapid fracture mechanics*, J. Mech. Phys. Solids **45** (1997), 1595–1605.

[2] M. Alemany, M. Jain, L. Kronik, and J. Chelikowsky, *Real-space pseudopotential method for computing the electronic properties of periodic systems*, Phys. Rev. B **69** (2004), 075101.

[3] P. Amestoy, I. Duff, J.-Y. L'Excellent, and J. Koster, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM J. Matrix Anal. and Appl. **23** (2001), 15–41.

[4] O. Krogh Andersen, *Linear methods in band theory*, Phys. Rev. B **12** (1975), 3060–3083.

[5] C. Andreani, D. Colognesi, J. Mayers, G.F. Reiter, and R. Senesi, *Measurement of momentum distribution of light atoms and molecules in condensed matter systems using inelastic neutron scattering*, Adv. Phys. **54** (2005), 377–469.

[6] T.A. Arias, M.C. Payne, and J.D. Joannopoulos, *Ab initio molecular dynamics: Analytically continued energy functionals and insights into iterative solutions*, Phys. Rev. Lett. **69** (1992), 1077–1080.

[7] D. N. Arnold, *An interior penalty finite element method with discontinuous elements*, SIAM J. Numer. Anal. **19** (1982), 742 –760.

[8] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM J. Numer. Anal. **39** (2002), 1749.

[9] C. Ashcraft and R. Grimes, *The influence of relaxed supernode partitions on the multifrontal method*, ACM Trans. Math. Software **15** (1989), 291–309.

[10] _____, *SPOOLES: An object-oriented sparse matrix library*, Ninth SIAM conference on parallel processing, 1999.

[11] C. Ashcraft, R. G. Grimes, and J. G. Lewis, *Accurate symmetric indefinite linear equation solvers*, SIAM J. Matrix Anal. Appl. **20** (1998), 513–561.

[12] N. Ashcroft and N. Mermin, *Solid state physics*, Thomson Learning, Toronto, 1976.

[13] C. Attaccalite, S. Moroni, P. Gori-Giorgi, and G. Bachelet, *Correlation energy and spin polarization in the 2d electron gas*, Phys. Rev. Lett. **88** (2002), 256601.

[14] F. Averill and D. Ellis, *An efficient numerical multicenter basis set for molecular orbital calculations: application to FeCl4*, J. Chem. Phys. **59** (1973), 6412–6418.

[15] I. Babuška and M. Zlámal, *Nonconforming elements in the finite element method with penalty*, SIAM J. Numer. Anal. **10** (1973), 863 –875.

[16] R. Baer and M. Head-Gordon, *Chebyshev expansion methods for electronic structure calculations on large molecular systems*, J. Chem. Phys. **107** (1997), 10003.

[17] H. J. Bakker and H. K. Nienhuys, *Delocalization of protons in liquid water*, Science **297** (2002), 587–590.

[18] A. Barducci, G. Bussi, and M. Parrinello, *Well-tempered metadynamics: a smoothly converging and tunable free-energy method*, Phys. Rev. Lett. **100** (2008), 020603.

[19] S. Baroni and P. Giannozzi, *Towards very large-scale electronic-structure calculations*, Europhys. Lett. **17** (1992), no. 6, 547–552.

[20] M. Bebendorf and W. Hackbusch, *Existence of $\mathcal{H}$-matrix approximants to the inverse FE-matrix of elliptic operators with L^∞-coefficients*, Numer. Math. **95** (2003), 1–28.

[21] T.L. Beck, *Quantum Contributions to Free Energy Changes in Fluids*, Springer series in chemical physics, 2007, pp. 389.

[22] A.D. Becke, *Density-functional exchange-energy approximation with correct asymptotic behavior*, Phys. Rev. A **38** (1988), 3098–3100.

[23] ———, *Density functional thermochemistry. iii. the role of exact exchange*, J. Chem. Phys. **98** (1993), 5648–5652.

[24] R.P. Bell, *The proton in chemistry*, Cornell University Press, 1973.

[25] P. Bendt and A. Zunger, *New approach for solving the density-functional self-consistent-field problem*, Phys. Rev. B **26** (1982), 3114–3137.

[26] M. Benoit and D. Marx, *The shapes of protons in hydrogen bonds depend on the bond length*, ChemPhysChem **6** (2005), 1738.

[27] M. Benoit, D. Marx, and M. Parrinello, *Tunnelling and zero-point motion in high-pressure ice*, Nature **392** (1998), 258.

[28] M. Benzi, C.D. Meyer, and M. Tuma, *A sparse approximate inverse preconditioner for the conjugate gradient method*, SIAM J. Sci. Comput. **17** (1996), 1135–1149.

[29] J.D. Bernal and R.H. Fowler, *A theory of water and ionic solution, with particular reference to hydrogen and hydroxyl ions*, J. Chem. Phys **1** (1933), 515–548.

[30] J.E. Bertie and E. Whalley, *Infrared spectra of ices Ih and Ic in the range 4000 to 350 cm^{-1}*, J. Chem. Phys. **40** (1964), 1637.

[31] G. Beylkin, R. Coifman, and V. Rokhlin, *Fast wavelet transforms and numerical algorithms I*, Comm. Pure Appl. Math **44** (1991), 141–183.

[32] ———, *Fast wavelet transforms and numerical algorithms I*, Comm. Pure Appl. Math **44** (1991), no. 2, 141–183.

[33] G. Beylkin, N. Coult, and M.J. Mohlenkamp, *Fast spectral projection algorithms for density-matrix computations*, J. Comput. Phys. **152** (1999), 32 –54.

[34] P. E. Blöchl, *Projector augmented-wave method*, Phys. Rev. B **50** (1994), 17953.

[35] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, and M. Scheffler, *Ab initio molecular simulations with numeric atom-centered orbitals*, Comput. Phys. Commun. **180** (2009), 2175–2196.

[36] S. Börm, L. Grasedyck, and W. Hackbusch, *Hierarchical matrices*, 2006. Max-Planck-Institute Lecture Notes.

[37] ———, *Hierarchical matrices*, 2006. Max-Planck-Institute Lecture Notes.

[38] M. Born and R. Oppenheimer, *Zur quantentheorie der molekeln*, Ann. Phys. (Leipzig) **389** (1927), 457–484.

[39] L. E. Bove, S. Klotz, A. Paciaroni, and F. Sacchetti, *Anomalous proton dynamics in ice at low temperatures*, Phys. Rev. Lett. **103** (2009), 165901.

[40] S.T. Bramwell and M.J.P. Gingras, *Spin ice state in frustrated magnetic pyrochlore materials*, Science **294** (2001), 1495.

[41] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp. **31** (1977), 333–390.

[42] A. Brandt, S. McCormick, and J. Ruge, *Algebraic multigrid (AMG) for sparse matrix equations*, Sparsity and its applications, 1985, pp. 257–284.

[43] W.L. Briggs, V. E. Henson, and S. F. McCormick, *A multigrid tutorial*, Second, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2000.

[44] S. Browne, J. Dongarra, N. Garner, G. Ho, and P. Mucci, *A portable programming interface for performance evaluation on modern processors*, Int. J. High Perform. C. **14** (2000), 189.

[45] J. R. Bunch and L. Kaufman, *Some stable methods for calculating inertia and solving symmetric linear systems*, Math. Comp. (1977), 163–179.

[46] J. R. Bunch and B. N. Parlett, *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM J. Numer. Anal. **8** (1971), 639–655.

[47] C.J. Burnham, G.F. Reiter, T. Abdul-Redah, H. Reichert, and H. Dosch, *On the origin of the redshift of the oh stretch in ice ih: evidence from the momentum distribution of the protons and the infrared spectral density*, Phys. Chem. Chem. Phys. **8** (2006), 3966.

[48] E. Cancès, C. Le Bris, Y. Maday, N.C. Nguyen, A.T. Patera, and G.S.H. Pau, *Feasibility and competitiveness of a reduced basis approach for rapid electronic structure calculations in quantum chemistry*, High-dimensional partial differential equations in science and engineering, 2007, pp. 15–47.

[49] R. Car and M. Parrinello, *Unified approach for molecular dynamics and density-functional theory*, Phys. Rev. Lett. **55** (1985), 2471–2474.

[50] A. Castro, H. Appel, M. Oliveira, C. Rozzi, X. Andrade, F. Lorenzen, M. Marques, E. Gross, and A. Rubio, *Octopus:a tool for the application of time-dependent density functional theory*, Phys. Stat. Sol. B **243** (2006), 2465–2488.

[51] D.M. Ceperley, *Path integrals in the theory of condensed helium*, Rev. Mod. Phys. **67** (1995), 279.

[52] D.M. Ceperley and B.J. Alder, *Ground state of the electron gas by a stochastic method*, Phys. Rev. Lett. **45** (1980), 566–569.

[53] M. Ceriotti, T.D. Kühne, and M. Parrinello, *An efficient and accurate decomposition of the Fermi operator.*, J. Chem. Phys **129** (2008), 024707.

[54] Michele Ceriotti, Giacomo Miceli, Antonino Pietropaolo, Daniele Colognesi, Angeloclaudio Nale, Michele Catti, Marco Bernasconi, and Michele Parrinello, *Nuclear quantum effects in ab initio dynamics: Theory and experiments for lithium imide*, Phys. Rev. B **82** (2010), 174306.

[55] S. Chandrasekaran, M. Gu, X. S. Li, and J. Xia, *Superfast multifrontal method for structured linear systems of equations*, 2006. Technical Report, LBNL-62897.

[56] S. Chandrasekaran, M. Gu, and W. Lyons, *A fast adaptive solver for hierarchically semiseparable representations*, Calcolo **42** (2005), no. 3-4, 171–185.

[57] S. Chandrasekaran, M. Gu, and T. Pals, *A fast ULV decomposition solver for hierarchically semiseparable representations*, SIAM J. Matrix Anal. Appl. **28** (2006), no. 3, 603–622.

[58] J. Chelikowsky, N. Troullier, and Y. Saad, *Finite-difference-pseudopotential method: Electronic structure calculations without a basis*, Phys. Rev. Lett. **72** (1994), 1240–1243.

[59] J.R. Chelikowsky, *The pseudopotential-density functional method applied to nanostructures*, J. Phys. D: Appl. Phys. **33** (2000), R33.

[60] W. Chen, M. Sharma, R. Resta, G. Galli, and R. Car, *Role of dipolar correlations in the infrared spectra of water and ice*, Phys. Rev. B **77** (2008), 245114.

[61] W. Chen, X. Wu, and R. Car, *X-ray absorption signatures of the molecular environment in water and ice*, Phys. Rev. Lett. **105** (2010), 17802.

[62] Y. Chen, J.S. Hesthaven, Y. Maday, and J. Rodríguez, *Certified reduced basis methods and output bounds for the harmonic Maxwell's equations*, SIAM J. Sci. Comput. **32** (2010), 970–996.

[63] J. Čížek, *On the Correlation Problem in Atomic and Molecular Systems. Calculation of Wavefunction Components in Ursell-Type Expansion Using Quantum-Field Theoretical Methods*, J. Chem. Phys. **45** (1966), 4256.

[64] B. Cockburn, G.E. Karniadakis, and C.-W. Shu, *Discontinuous Galerkin methods: Theory, computation and applications*, Lecture Notes in Computational Science and Engineering, vol. 11, Springer-Verlag, Berlin, 2000.

[65] B. Cockburn and C.-W. Shu, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal. **35** (1998), 2440.

[66] ———, *Runge–Kutta discontinuous Galerkin methods for convection-dominated problems*, J. Sci. Comp. **16** (2001), 173–261.

[67] C.H. Collie, J.B. Hasted, and D.M. Ritson, *The dielectric properties of water and heavy water*, Proc. Phys. Soc. **60** (1948), 145.

[68] X. Dai, X. Gong, Z. Yang, D. Zhang, and A. Zhou, *Finite volume discretizations for eigenvalue problems with applications to electronic structure calculations*, Multiscale Model. Simul. **9** (2011), 208–240.

[69] E.R. Davidson, *The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices*, J. Comput. Phys. **17** (1975), 87.

[70] T. Davis, *University of Florida sparse matrix collection*, NA Digest **97** (1997), 7.

[71] B. Delley and D. Ellis, *Efficient and accurate expansion methods for molecules in local density models*, J. Chem. Phys. **76** (1982), 1949–1960.

[72] L. Demanet and L. Ying, *Discrete symbol calculus*, SIAM Rev. (2010).

[73] P. Dirac, *Quantum mechanics of many-electron systems*, Proc. R. Soc. A **123** (1929), 714–733.

[74] P. Drineas, R. Kannan, and M. W. Mahoney, *Fast Monte Carlo algorithms for matrices. II. Computing a low-rank approximation to a matrix*, SIAM J. Comput. **36** (2006), no. 1, 158–183.

[75] ———, *Fast Monte Carlo algorithms for matrices. III. Computing a compressed approximate matrix decomposition*, SIAM J. Comput. **36** (2006), no. 1, 184–206.

[76] I. Duff, R. Grimes, and J. Lewis, *User's guide for the Harwell-Boeing sparse matrix collection*, Research and Technology Division, Boeing Computer Services, Seattle, Washington, USA (1992).

[77] J.S. Duff and J.K. Reid, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software **9** (1983), 302–325.

[78] ———, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software **9** (1983), 302–325.

[79] B. Dünweg and K. Kremer, *Molecular dynamics simulation of a polymer chain in solution*, J. Chem. Phys. **99** (1993), 6983–6997.

[80] B. Engquist and O. Runborg, *Wavelet-based numerical homogenization with applications*, Multiscale and multiresolution methods, 2002, pp. 97–148.

[81] H. Eschrig, *Optimized LCAO method and the electronic structure of extended systems*, Akademie Verlag and Springer, Berlin, 1988.

[82] J.L. Fattebert, RD Hornung, and AM Wissink, *Finite element approach for density functional theory calculations on locally-refined meshes*, J. Comput. Phys. **223** (2007), 759–773.

[83] R.P. Feynman and A.R. Hibbs, *Quantum Mechanics and Path Integrals*, McGraw-Hill New York, 1965.

[84] D. Flammini, A. Pietropaolo, R. Senesi, C. Andreani, F. McBride, A. Hodgson, M. Adams, L. Lin, and R. Car, *Quantum proton in hexagonal ice: interpretation of a new experiment*, Phys. Rev. Lett., submitted (2011).

[85] W.M.C. Foulkes, L. Mitas, R.J. Needs, and G. Rajagopal, *Quantum Monte Carlo simulations of solids*, Rev. Mod. Phys. **73** (2001), 33.

[86] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, Academic Press, 2002.

[87] J. Friedel, *XIV. The distribution of electrons round impurities in monovalent metals*, Phil. Mag. **43** (1952), 153.

[88] G. Galli and M. Parrinello, *Large scale electronic structure calculations*, Phys. Rev. Lett. **69** (1992), 3547–3550.

[89] W. Gao and W. E, *Orbital minimization with localization*, Discret. Contin. Dyn. S. **23** (2009), 249–264.

[90] V. Garbuio, C. Andreani, S. Imberti, A. Pietropaolo, G.F. Reiter, R. Senesi, and M.A. Ricci, *Proton quantum coherence observed in water confined in silica nanopores*, J. Chem. Phys. **127** (2007), 154501.

[91] C. J. García-Cervera, Jianfeng Lu, Yulin Xuan, and Weinan E, *Linear-scaling subspace-iteration algorithm with optimally localized nonorthogonal wave functions for kohn-sham density functional theory*, Phys. Rev. B **79** (2009), 115110.

[92] A. George, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal. **10** (1973), 345–363.

[93] A. George and J. Liu, *Computer solution of large sparse positive definite systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[94] Paolo Giannozzi, Stefano Baroni, Nicola Bonini, Matteo Calandra, Roberto Car, Carlo Cavazzoni, Davide Ceresoli, Guido L Chiarotti, Matteo Cococcioni, Ismaila Dabo, Andrea Dal Corso, Stefano de Gironcoli, Stefano Fabris, Guido Fratesi, Ralph Gebauer, Uwe Gerstmann, Christos Gougoussis, Anton Kokalj, Michele Lazzeri, Layla Martin-Samos, Nicola Marzari, Francesco Mauri, Riccardo Mazzarello, Stefano Paolini, Alfredo Pasquarello, Lorenzo Paulatto, Carlo Sbraccia, Sandro Scandolo, Gabriele Sclauzero, Ari P Seitsonen, Alexander Smogunov, Paolo Umari, and Renata M Wentzcovitch, *Quantum espresso: a modular and open-source software project for quantum simulations of materials*, J. Phys.: Condens. Matter **21** (2009), 395502.

[95] J.R. Gilbert, *Predicting structure in sparse matrix computations*, SIAM J. Matrix Anal. Appl. **15** (1984), 62–79.

[96] J.R. Gilbert and T. Peierls, *Sparse partial pivoting in time proportional to arithmetic operations*, SIAM J. Sci. Stat. Comput. **9** (1986), 862–874.

[97] W. Gilchrist, *Statistical modelling with quantile functions*, CRC Press, 2000.

[98] MJ Gillan, DR Bowler, AS Torralba, and T. Miyazaki, *Order-N first-principles calculations with the conquest code*, Comput. Phys. Commun. **177** (2007), 14–18.

[99] S. Goedecker, *Integral representation of the fermi distribution and its applications in electronic-structure calculations*, Phys. Rev. B **48** (1993), 17573–17575.

[100] ———, *Decay properties of the finite-temperature density matrix in metals*, Phys. Rev. B **58** (1998), 3501.

[101] ———, *Linear scaling electronic structure methods*, Rev. Mod. Phys. **71** (1999), 1085–1123.

[102] S. Goedecker and L. Colombo, *Efficient linear scaling algorithm for tight-binding molecular dynamics*, Phys. Rev. Lett. **73** (1994), 122–125.

[103] S. Goedecker and M. Teter, *Tight-binding electronic-structure calculations and tight-binding molecular dynamics with localized orbitals*, Phys. Rev. B **51** (1995), 9455–9464.

[104] S. Goedecker, M. Teter, and J. Hutter, *Separable dual-space gaussian pseudopotentials*, Phys. Rev. B **54** (1996), 1703.

[105] G.H. Golub and C.F. Van Loan, *Matrix computations*, third, Johns Hopkins Univ. Press, Baltimore, 1996.

[106] S.A. Goreinov, E.E. Tyrtyshnikov, and N.L. Zamarashkin, *A theory of pseudoskeleton approximations*, Linear Algebra Appl. **261** (1997), 1–21.

[107] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys. **73** (1987), 325–348.

[108] ———, *A fast algorithm for particle simulations*, J. Comput. Phys. **73** (1987), 325–348.

[109] J. C. Grossman, E. Schwegler, E. W. Draeger, F. Gygi, and G. Galli, *Towards an assessment of the accuracy of density functional theory for first principles simulations of water*, J. Chem. Phys. **120** (2004), 300.

[110] A. Gupta, *WSMP: the Watson Sparse Matrix Package*, IBM Research Report **21886** (1997), 98462.

[111] A. Gupta, G. Karypis, and V. Kumar, *Highly scalable parallel algorithms for sparse matrix factorization*, IEEE Trans. Parallel Distrib. Syst. **8** (1997), 502–520.

[112] J. Gustafson, *Reevaluating Amdahl's law*, Comm. ACM **31** (1988), 532–533.

[113] F. Gygi, *Architecture of Qbox: A scalable first-principles molecular dynamics code*, IBM J. Res. Dev. **52** (2008), 137.

[114] W. Hackbusch, *A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices.*, Computing **62** (1999), 89–108.

[115] ———, *A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices.*, Computing **62** (1999), 89–108.

[116] W. Hackbusch, B. Khoromskij, and S. A. Sauter, *On $\mathcal{H}^2$-matrices*, Lectures on applied mathematics (Munich, 1999), 2000, pp. 9–29.

[117] W. Hackbusch and Z. P. Nowak, *On the fast matrix multiplication in the boundary element method by panel clustering*, Numer. Math. **54** (1989), 463–491.

[118] N. Hale, N. J. Higham, and L. N. Trefethen, *Computing A^α, $\log(A)$, and related matrix functions by contour integrals*, SIAM J. Numer. Anal. **46** (2008), no. 5, 2505–2523.

[119] N. Halko, P.G. Martinsson, and J.A. Tropp, *Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions*, 2009. preprint, arXiv:0909.4061.

[120] D. R. Hamann, M. Schlüter, and C. Chiang, *Norm-conserving pseudopotentials*, Phys. Rev. Lett. **43** (1979), 1494.

[121] C. Hartwigsen, S. Goedecker, and J. Hutter, *Relativistic separable dual-space gaussian pseudopotentials from h to rn*, Phys. Rev. B **58** (1998), 3641.

[122] J.A. Hayward and J.R. Reimers, *Unit cells for the simulation of hexagonal ice*, J. Chem. Phys. **106** (1997), 1518.

[123] R.J. Hemley, A.P. Jephcoat, H.K. Mao, C.S. Zha, L.W. Finger, and D.E. Cox, *Static compression of H_2O-ice to 128 GPa (1.28 Mbar)*, Nature **330** (1987), 737.

[124] M. Herman, E. Bruskin, and B. Berne, *On path integral monte carlo simulations*, J. Chem. Phys. **76** (1982), 5150–5155.

[125] P. Hohenberg and W. Kohn, *Inhomogeneous electron gas*, Phys. Rev. **136** (1964), B864–B871.

[126] D. Homouz, G.F. Reiter, J. Eckert, J. Mayers, and R. Blinc, *Measurement of the 3D born-oppenheimer potential of a proton in a hydrogen-bonded system via deep inelastic neutron scattering: The superprotonic conductor $Rb_3H(SO_4)_2$*, Phys. Rev. Lett. **98** (2007), 115502.

[127] W.G. Hoover, *Canonical dynamics: Equilibrium phase-space distributions*, Phys. Rev. A **31** (1985), 1695–1697.

[128] M. Iannuzzi, A. Laio, and M. Parrinello, *Efficient exploration of reactive potential energy surfaces using Car-Parrinello molecular dynamics*, Phys. Rev. Lett. **90** (2003), 238302.

[129] Cray Inc., *Using cray performance analysis tools*, Cray Inc., 2009.

[130] G.A. Jeffrey and W. Saenger, *Hydrogen bonding in biological structures*, Springer-Verlag Berlin, 1991.

[131] H. Jeffreys and B. S. Jeffreys, *Methods of mathematical physics*, Third, Cambridge, at the University Press, 1956.

[132] J. Junquera, O. Paz, D. Sanchez-Portal, and E. Artacho, *Numerical atomic orbitals for linear-scaling calculations*, Phys. Rev. B **64** (2001), 235111.

[133] G. Karypis and V. Kumar, *A parallel algorithm for multilevel graph partitioning and sparse matrix ordering*, J. Parallel Distrib. Comput. **48** (1998), 71–85.

[134] T. Kato, *On the eigenfunctions of many-particle systems in quantum mechanics*, Commun. Pure Appl. Math. **10** (1957), 151.

[135] S. Kenny, A. Horsfield, and H. Fujitani, *Transferable atomic-type orbital basis sets for solids*, Phys. Rev. B **62** (2000), 4899–4905.

[136] J. Kim, F. Mauri, and G. Galli, *Total-energy global optimizations using nonorthogonal localized orbitals*, Phys. Rev. B **52** (1995), 1640–1648.

[137] J.G. Kirkwood, *Statistical mechanics of fluid mixtures*, J. Chem. Phys. **3** (1935), 300.

[138] L. Kleinman and D.M. Bylander, *Efficacious form for model pseudopotentials*, Phys. Rev. Lett. **48** (1982), 1425–1428.

[139] A.V. Knyazev, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comp. **23** (2001), 517–541.

[140] E. Koch and S. Goedecker, *Locality properties and Wannier functions for interacting systems*, Solid State Commun. **119** (2001), 105.

[141] K. Koepernik and H. Eschrig, *Full-potential nonorthogonal local-orbital minimum-basis band-structure scheme*, Phys. Rev. B **59** (1999), 1743–1757.

[142] W. Kohn, *Density functional and density matrix method scaling linearly with the number of atoms*, Phys. Rev. Lett. **76** (1996), 3168–3171.

[143] W. Kohn and L. Sham, *Self-consistent equations including exchange and correlation effects*, Phys. Rev. **140** (1965), A1133–A1138.

[144] F.R. Krajewski and M. Parrinello, *Linear scaling electronic structure Monte Carlo method for metals*, Phys. Rev. B **75** (2007), 235108.

[145] G. Kresse and J. Furthmüller, *Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set*, Comput. Mater. Sci. **6** (1996), 15–50.

[146] M. Krzystyniak, *Nuclear momentum distribution in solid and liquid hf from ab initio calculation*, J. Chem. Phys. **133** (2010), 144505.

[147] L.D. Landau and E.M. Lifshitz, *Statistical Physics, Vol. 1*, Vol. 24, Oxford: Pergamon, 1980.

[148] C. Lee, D. Vanderbilt, K. Laasonen, R. Car, and M. Parrinello, *Ab initio studies on the structural and dynamical properties ofice*, Phys. Rev. B **47** (1993), 4863–4872.

[149] C. Lee, W. Yang, and R.G. Parr, *Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density*, Phys. Rev. B **37** (1988), 785–789.

[150] Changyol Lee, David Vanderbilt, Kari Laasonen, R. Car, and M. Parrinello, *Ab initio studies on high pressure phases of ice*, Phys. Rev. Lett. **69** (1992), 462–465.

[151] R. Lehoucq, D. Sorensen, and C. Yang, *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, SIAM, 1998.

[152] I.N. Levine, *Quantum Chemistry*, Englewood Cliffs, New Jersey: Prentice Hall, 1991.

[153] J. Li, *Inelastic neutron scattering studies of hydrogen bonding in ices*, J. Chem. Phys. **105** (1996), 6733.

[154] X.-P. Li, R. W. Nunes, and David Vanderbilt, *Density-matrix electronic-structure method with linear system-size scaling*, Phys. Rev. B **47** (1993), 10891–10894.

[155] C.and Shao Y. Liang W.and Saravanan, R. Baer, A. Bell, and M. Head-Gordon, *Improved fermi operator expansion methods for fast electronic structure calculations*, J. Chem. Phys. **119** (2003), 4117–4125.

[156] W. Liang, R. Baer, C. Saravanan, Y. Shao, A. Bell, and M. Head-Gordon, *Fast methods for resumming matrix polynomials and chebyshev matrix polynomials*, J. Comput. Phys. **194** (2004), 575 –587.

[157] E. Liberty, F. Woolfe, P.G. Martinsson, V. Rokhlin, and M. Tygert, *Randomized algorithms for the low-rank approximation of matrices*, Proc. Natl. Acad. Sci. USA **104** (2007), 20167–20172.

[158] E. H. Lieb, *Density functional for Coulomb systems*, Int J. Quantum Chem. **24** (1983), 243.

[159] E.H. Lieb and B. Simon, *The Hartree-Fock theory for Coulomb systems*, Commun. Math. Phys. **53** (1977), 185–194.

[160] L. Lin, J. Lu, R. Car, and W. E, *Multipole representation of the Fermi operator with application to the electronic structure analysis of metallic systems*, Phys. Rev. B **79** (2009), 115133.

[161] L. Lin, J. Lu, and L. Ying, *Fast construction of hierarchical matrix representation from matrix-vector multiplication*, J. Comput. Phys. **230** (2011), 4071.

[162] L. Lin, J. Lu, L. Ying, R. Car, and W. E, *Fast algorithm for extracting the diagonal of the inverse matrix with application to the electronic structure analysis of metallic systems*, Comm. Math. Sci. **7** (2009), 755–777.

[163] ———, *Fast algorithm for extracting the diagonal of the inverse matrix with application to the electronic structure analysis of metallic systems*, Comm. Math. Sci. **7** (2009), 755.

[164] L. Lin, J. Lu, L. Ying, and W. E, *Pole-based approximation of the Fermi-Dirac function*, Chinese Ann. Math. **30B** (2009), 729.

[165] ———, *Adaptive local basis set for Kohn-Sham density functional theory in a discontinuous Galerkin framework I: Total energy calculation*, J. Comput. Phys., submitted (2011).

[166] L. Lin, J. A. Morrone, R. Car, and M. Parrinello, *Displaced path integral formulation for the momentum distribution of quantum particles*, Phys. Rev. Lett. **105** (2010), 110602.

[167] ———, *Momentum distribution, vibrational dynamics, and the potential of mean force in ice*, Phys. Rev. B **83** (2011), 220302(R).

[168] L. Lin, J.A. Morrone, and R. Car, *Correlated tunneling in hydrogen bonds*, J. Stat. Phys., submitted (2011).

[169] L. Lin, C. Yang, J. Lu, L. Ying, and W. E, *A fast parallel algorithm for selected inversion of structured sparse matrices with application to 2D electronic structure calculations*, SIAM J. Sci. Comput. **33** (2011), 1329.

[170] L. Lin, C. Yang, J. Meza, J. Lu, L. Ying, and W. E, *SelInv – An algorithm for selected inversion of a sparse symmetric matrix*, ACM. Trans. Math. Software **37** (2010), 40.

[171] J. Liu, *Modification of the minimum degree algorithm by multiple elimination*, ACM Trans. Math. Software **11** (1985), 141–153.

[172] ———, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl. **11** (1990), 134.

[173] J. Lobaugh and Gregory A. Voth, *A quantum model for water: Equilibrium and dynamical properties*, J. Chem. Phys. **106** (1997), 2400–2410.

[174] Y. Maday, A.T. Patera, and G. Turinici, *Global a priori convergence theory for reduced-basis approximations of single-parameter symmetric coercive elliptic partial differential equations*, C. R. Acad. Sci. Paris, Ser. I **335** (2002), 289–294.

[175] Y. Maday and E.M. Rønquist, *A reduced-basis element method*, J. Sci. Comput. **17** (2002), 447–459.

[176] G.D. Mahan, *Many-particle Physics*, Plenum Pub Corp, 2000.

[177] M.W. Mahoney and P. Drineas, *CUR matrix decompositions for improved data analysis*, Proc. Natl. Acad. Sci. USA **106** (2009), 697–702.

[178] P.G. Martinsson, *Rapid factorization of structured matrices via randomized sampling*, 2008. preprint, arXiv:0806.2339.

[179] ———, *A fast direct solver for a class of elliptic partial differential equations*, Journal of Scientific Computing **38** (2009), 316–330.

[180] G.J. Martyna, M.L. Klein, and M.E. Tuckerman, *Nose-hoover chains: The canonical ensemble via continuous dynamics*, J. Chem. Phys. **97** (1992), 2635.

[181] D. Marx, *Proton transfer in ice*, Classical and quantum dynamics in condensed phase simulations, 1998, pp. 359.

[182] D. Marx and J. Hutter, *Ab initio molecular dynamics: theory and implementation*, Modern methods and algorithms of quantum chemistry **1** (2000), 301–449.

[183] D. Marx, M.E. Tuckerman, and G.J. Martyna, *Quantum Dynamics via adiabatic ab initio centroid molecular dynamics*, Comput. Phys. Commun. **118** (1999), 166–184.

[184] F. Mauri, G. Galli, and R. Car, *Orbital formulation for electronic-structure calculations with linear system-size scaling*, Phys. Rev. B **47** (1993), 9973–9976.

[185] R. McWeeny, *Some recent advances in density matrix theory*, Rev. Mod. Phys. **32** (1960), 335–369.

[186] N.D. Mermin, *Thermal properties of the inhomogeneous electron gas*, Phys. Rev. **137** (1965), A1441–A1443.

[187] C. Møller and M.S. Plesset, *Note on an approximation treatment for many-electron systems*, Phys. Rev. **46** (1934), 618.

[188] R. Moreh and D. Nemirovski, *On the proton kinetic energy in h_2o and in nanotube water*, J. Chem. Phys. **133** (2010), 084506.

[189] I. Morrison and S. Jenkins, *First principles lattice dynamics studies of the vibrational spectra of ice*, Physica B **263** (1999), 442–444.

[190] J. A. Morrone, L. Lin, and R. Car, *Tunneling and delocalization in hydrogen bonded systems: a study in position and momentum space*, J. Chem. Phys. **130** (2009), 204511.

[191] J.A. Morrone and R. Car, *Nuclear quantum effects in water*, Phys. Rev. Lett. **101** (2008), 017801.

[192] J.A. Morrone, V. Srinivasan, D. Sebastiani, and R. Car, *Proton momentum distribution in water: an open path integral molecular dynamics study*, J. Chem. Phys. **126** (2007), 234504.

[193] A.A. Mostofi, J.R. Yates, Y.S. Lee, I. Souza, D. Vanderbilt, and N. Marzari, *Wannier90: a tool for obtaining maximally-localised wannier functions*, Comput. Phys. Commun. **178** (2008), 685–699.

[194] A.H.C. Neto, P. Pujol, and E. Fradkin, *Ice: A strongly correlated proton system*, Phys. Rev. B **74** (2006), 024302.

[195] E. Ng and B. Peyton, *Block sparse Cholesky algorithms on advanced uniprocessor computers*, SIAM J. Sci. Comput. **14** (1993), 1034.

[196] Shuichi Nosé, *A unified formulation of the constant temperature molecular dynamics methods*, J. Chem. Phys. **81** (1984), 511–519.

[197] H. Owhadi and L. Zhang, *Metric-based upscaling*, Comm. Pure Appl. Math. **60** (2007), 675–723.

[198] T. Ozaki, *Variationally optimized atomic orbitals for large-scale electronic structures*, Phys. Rev. B **67** (2003), 155108.

[199] ———, *Continued fraction representation of the Fermi-Dirac function for large-scale electronic structure calculations*, Phys. Rev. B **75** (2007), 035123.

[200] C. Pantalei, A. Pietropaolo, R. Senesi, S. Imberti, C. Andreani, J. Mayers, C. Burnham, and G. Reiter, *Proton momentum distribution of liquid water from room temperature to the supercritical phase*, Phys. Rev. Lett. **100** (2008), 177801.

[201] J.E. Pask, B.M. Klein, C.Y. Fong, and P.A. Sterne, *Real-space local polynomial basis for solid-state electronic-structure calculations: A finite-element approach*, Phys. Rev. B **59** (1999), 12352–12358.

[202] J.E. Pask and P.A. Sterne, *Finite element methods in ab initio electronic structure calculations*, Modelling Simul. Mater. Sci. Eng. **13** (2005), R71.

[203] L. Pauling, *The structure and entropy of ice and of other crystals with some randomness of atomic arrangement*, J. Am. Chem. Soc. **57** (1935), 2680–2684.

[204] J.P. Perdew, K. Burke, and M. Ernzerhof, *Generalized gradient approximation made simple*, Phys. Rev. Lett. **77** (1996), 3865–3868.

[205] J.P. Perdew, M. Ernzerhof, and K. Burke, *Rationale for mixing exact exchange with density functional approximations*, J. Chem. Phys. **105** (1996), 9982–9985.

[206] J.P. Perdew and A. Zunger, *Self-interaction correction to density-functional approximations for many-electron systems*, Phys. Rev. B **23** (1981), 5048–5079.

[207] B.G. Pfrommer, J. Demmel, and H. Simon, *Unconstrained energy functionals for electronic structure calculations*, J. Comput. Phys. **150** (1999), 287–298.

[208] A. Pietropaolo, R. Senesi, C. Andreani, A Botti, M.A. Ricci, and F. Bruni, *Excess of proton mean kinetic energy in supercooled water*, Phys. Rev. Lett. **100** (2008), 127802.

[209] A. Pietropaolo, R. Senesi, C. Andreani, and J. Mayers, *Quantum effects in water: proton kinetic energy maxima in stable and supercooled liquid*, Braz. J. Phys. **39** (2009), 318–321.

[210] A. Pothen and C. Sun, *A mapping algorithm for parallel sparse Cholesky factorization*, SIAM J. Sci. Comput. **14** (1993), 1253–1253.

[211] H. Prask, H. Boutin, and S. Yip, *Frequency spectrum of hydrogenous molecular solids by Inelastic Neutron Scattering. Hexagonal H_2O ice*, J. Chem. Phys. **48** (1968), 3367.

[212] E. Prodan and W. Kohn, *Nearsightedness of electronic matter*, Proc. Natl. Acad. Sci. **102** (2005), 11635–11638.

[213] P. Pulay, *Ab initio calculation of force constants and equilibrium geometries in polyatomic molecules I. Theory*, Mol. Phys. **17** (1969), 197–204.

[214] ———, *Convergence acceleration of iterative sequences: The case of SCF iteration*, Chem. Phys. Lett. **73** (1980), 393–398.

[215] P. Raghavan, *DSCPACK: Domain-separator codes for solving sparse linear systems*, Technical Report Rep.CSE-02-004, 2002.

[216] E. Räsänen, H. Saarikoski, V. Stavrou, A. Harju, M. Puska, and R. Nieminen, *Electronic structure of rectangular quantum dots*, Phys. Rev. B **67** (2003), 235307.

[217] G. Reiter and R. Silver, *Measurement of interionic potentials in solids using deep-inelastic neutron scattering*, Phys. Rev. Lett. **54** (1985), 1047–1050.

[218] G.F. Reiter, J.C. Li, J. Mayers, T. Abdul-Redah, and P. Platzman, *The proton momentum distribution in water and ice*, Braz. J. Phys. **34** (2004), 142.

[219] G.F. Reiter, J. Mayers, and P. Platzman, *Direct observation of tunneling in kdp using neutron compton scattering*, Phys. Rev. Lett. **89** (2002), 135505.

[220] J.M. Robertson and AR Ubbelohde, *Structure and thermal properties associated with some hydrogen bonds in crystals. i. the isotope effect*, Proc. Roy. Soc. London Ser. A **170** (1939), 222.

[221] E. Rothberg and A. Gupta, *An efficient block-oriented approach to parallel sparse choleskyfactorization*, SIAM J. Sci. Comput. **15** (1994), 1413–1439.

[222] B. Santra, A. Michaelides, and M. Scheffler, *Coupled cluster benchmarks of water monomers and dimers extracted from density-functional theory liquid water: The importance of monomer deformations*, J. Chem. Phys. **131** (2009), 124509.

[223] O. Schenk and K. Gartner, *On fast factorization pivoting methods for symmetric indefinite systems*, Elec. Trans. Numer. Anal. **23** (2006), 158–179.

[224] K.S. Schweizer and F.H. Stillinger, *High pressure phase transitions and hydrogen-bond symmetry in ice polymorphs*, J. Chem. Phys. **80** (1984), 1230.

[225] ———, *Phase transitions induced by proton tunneling in hydrogen-bonded crystals. ground-state theory*, Phys. Rev. B **29** (1984), 350.

[226] V. F. Sears, *Scaling and final-state interactions in deep-inelastic neutron scattering*, Phys. Rev. B **30** (1984), 44–51.

[227] R. B. Sidje and Y. Saad, *Rational approximation to the Fermi-Dirac function with applications in density functional theory*, Technical Report umsi-2008-279, Minnesota Supercomputer Institute, University of Minnesota, 2008.

[228] D. Skinner and W. Kramer, *Understanding the causes of performance variability in HPC workloads*, IEEE International Symposium on Workload Characterization, IISWC05, 2005.

[229] C.K. Skylaris, P.D. Haynes, A.A. Mostofi, and M.C. Payne, *Introducing ONETEP: Linear-scaling density functional simulations on parallel computers*, J. Chem. Phys. **122** (2005), 084119.

[230] J. C. Slater, *Wave functions in a periodic potential*, Phys. Rev. **51** (1937), 846–851.

[231] J. C. Slater and G. F. Koster, *Simplified LCAO method for the periodic potential problem*, Phys. Rev. **94** (1954), 1498–1524.

[232] B. F. Smith, P. E. Bjørstad, and W. D. Gropp, *Domain decomposition*, Cambridge Univ. Press, Cambridge, 1996.

[233] A. K. Soper, *Comment on "excess of proton mean kinetic energy in supercooled water"*, Phys. Rev. Lett. **103** (2009), 069801.

[234] T. R. Sosnick, W. M. Snow, and P. E. Sokol, *Deep-inelastic neutron scattering from liquid ^4he*, Phys. Rev. B **41** (1990), 11185–11202.

[235] F.H. Stillinger and K.S. Schweizer, *Ice under pressure: transition to symmetrical hydrogen bonds*, J. Phys. Chem. **87** (1983), 4281.

[236] N. Sukumar and J.E. Pask, *Classical and enriched finite element formulations for Bloch-periodic boundary conditions*, Int. J. Numer. Meth. Engng. **77** (2009), 1121–1138.

[237] A. Szabo and N.S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, McGraw-Hill, New York, 1989.

[238] M.J. Taylor and E. Whalley, *Raman spectra of ices Ih, Ic, II, III, and V*, J. Chem. Phys. **40** (1964), 1660.

[239] M.P. Teter, M.C. Payne, and D.C. Allan, *Solution of Schrödinger's equation for large systems*, Phys. Rev. B **40** (1989), 12255.

[240] A. Toselli and O. Widlund, *Domain decomposition methods – algorithms and theory*, Springer Series in Computational Mathematics, vol. 34, Springer-Verlag, Berlin, 2005.

[241] N. Troullier and José Luriaas Martins, *Efficient pseudopotentials for plane-wave calculations*, Phys. Rev. B **43** (1991), 1993–2006.

[242] E. Tsuchida and M. Tsukada, *Electronic-structure calculations based on the finite-element method*, Phys. Rev. B **52** (1995), 5573–5578.

[243] M.E. Tuckerman, D. Marx, M.L. Klein, and M. Parrinello, *Efficient and general algorithms for path integral car-parrinello molecular dynamics*, J. Chem. Phys. **104** (1996), 5579.

[244] T. Van Voorhis and M. Head-Gordon, *A geometric approach to direct minimization*, Mol. Phys. **100** (2002), 1713–1721.

[245] David Vanderbilt, *Soft self-consistent pseudopotentials in a generalized eigenvalue formalism*, Phys. Rev. B **41** (1990), 7892–7895.

[246] W. Wang, J. Guzmán, and C.-W. Shu, *The multiscale discontinuous Galerkin method for solving a class of second order elliptic problems with rough coefficients*, Int. J. Numer. Anal. Model. **8** (2011), 28 –47.

[247] M.F. Wheeler, *An elliptic collocation-finite element method with interior penalties*, SIAM J. Numer. Anal. **15** (1978), 152–161.

[248] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert, *A fast randomized algorithm for the approximation of matrices*, Appl. Comput. Harmon. Anal. **25** (2008), 335–366.

[249] S. Woutersen, U. Emmerichs, H.K. Nienhuys, and H.J. Bakker, *Anomalous temperature dependence of vibrational lifetimes in water and ice*, Phys. Rev. Lett. **81** (1998), 1106–1109.

[250] C. Yang, J.C. Meza, and L.W. Wang, *A constrained optimization algorithm for total energy minimization in electronic structure calculations*, J. Comput. Phys. **217** (2006), 709–721.

[251] W. Yang, *Direct calculation of electron density in density-functional theory*, Phys. Rev. Lett. **66** (1991), 1438–1441.

[252] ———, *Electron density as the basic variable: a divide-and-conquer approach to the ab initio computation of large molecules*, J. Mol. Struct.: THEOCHEM **255** (1992), 461–479.

[253] L. Ying, G. Biros, and D. Zorin, *A kernel-independent adaptive fast multipole algorithm in two and three dimensions*, J. Comput. Phys. **196** (2004), 591 –6262.

[254] L. Yuan and C.-W. Shu, *Discontinuous Galerkin method based on non-polynomial approximation spaces*, J. Comput. Phys. **218** (2006), 295 –323.

[255] ———, *Discontinuous Galerkin method for a class of elliptic multi-scale problems*, Int. J. Numer. Methods Fluids **56** (2007), 1017 –1032.

[256] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky, *Self-consistent-field calculations using chebyshev-filtered subspace iteration*, J. Comput. Phys. **219** (2006), 172–184.

[257] E. Zmijewski and J. R. Gilbert, *A parallel algorithm for sparse symbolic Cholesky factorization on a multiprocessor*, Parallel Comput. **7** (1988), 199–210.

[258] R.W. Zwanzig, *High-Temperature Equation of State by a Perturbation Method. I. Nonpolar Gases*, J. Chem. Phys. **22** (1954), 1420.