Matrix Algebra for Quantum Chemistry

EMANUEL H. RUBENSSON

Doctoral Thesis in Theoretical Chemistry
Stockholm, Sweden 2008

Matrix Algebra for Quantum Chemistry
Doctoral Thesis

TRITA-BIO-Report 2008:23
ISBN 978-91-7415-160-2
ISSN 1654-2312
Printed by Universitetsservice US AB,
Stockholm, Sweden 2008
Typeset in LaTeX by the author.

Abstract

This thesis concerns methods of reduced complexity for electronic structure calculations. When quantum chemistry methods are applied to large systems, it is important to optimally use computer resources and only store data and perform operations that contribute to the overall accuracy. At the same time, precarious approximations could jeopardize the reliability of the whole calculation. In this thesis, the self–consistent field method is seen as a sequence of rotations of the occupied subspace. Errors coming from computational approximations are characterized as erroneous rotations of this subspace. This viewpoint is optimal in the sense that the occupied subspace uniquely defines the electron density. Errors should be measured by their impact on the overall accuracy instead of by their constituent parts. With this point of view, a mathematical framework for control of errors in Hartree–Fock/Kohn–Sham calculations is proposed. A unifying framework is of particular importance when computational approximations are introduced to efficiently handle large systems.

An important operation in Hartree–Fock/Kohn–Sham calculations is the calculation of the density matrix for a given Fock/Kohn–Sham matrix. In this thesis, density matrix purification is used to compute the density matrix with time and memory usage increasing only linearly with system size. The forward error of purification is analyzed and schemes to control the forward error are proposed. The presented purification methods are coupled with effective methods to compute interior eigenvalues of the Fock/Kohn–Sham matrix also proposed in this thesis. New methods for inverse factorizations of Hermitian positive definite matrices that can be used for congruence transformations of the Fock/Kohn–Sham and density matrices are suggested as well.

Most of the methods above have been implemented in the Ergo quantum chemistry program. This program uses a hierarchic sparse matrix library, also presented in this thesis, which is parallelized for shared memory computer architectures. It is demonstrated that the Ergo program is able to perform linear scaling Hartree–Fock calculations.

List of papers

PAPER 1. Rotations of occupied invariant subspaces in self–consistent field calculations,
Emanuel H. Rubensson, Elias Rudberg, and Paweł Sałek,
J. Math. Phys. **49**, 032103 (2008).

PAPER 2. Density matrix purification with rigorous error control,
Emanuel H. Rubensson, Elias Rudberg, and Paweł Sałek,
J. Chem. Phys. **128**, 074106 (2008).

PAPER 3. Computation of interior eigenvalues in electronic structure calculations facilitated by density matrix purification,
Emanuel H. Rubensson and Sara Zahedi,
J. Chem. Phys. **128**, 176101 (2008).

PAPER 4. Recursive inverse factorization,
Emanuel H. Rubensson, Nicolas Bock, Erik Holmström, and Anders M. N. Niklasson,
J. Chem. Phys. **128**, 104105 (2008).

PAPER 5. Truncation of small matrix elements based on the Euclidean norm for blocked data structures,
Emanuel H. Rubensson, Elias Rudberg, and Paweł Sałek,
J. Comput. Chem. **00**, 000–000 (2008).

PAPER 6. A hierarchic sparse matrix data structure for large–scale Hartree–Fock/Kohn–Sham calculations,
Emanuel H. Rubensson, Elias Rudberg, and Paweł Sałek,
J. Comput. Chem. **28**, 2531–2537 (2007).

PAPER 7. Hartree–Fock calculations with linearly scaling memory usage,
Elias Rudberg, Emanuel H. Rubensson, and Paweł Sałek,
J. Chem. Phys. **128**, 184106 (2008).

Comments on my contribution In the papers where I am first author, I have been driving the project from idea to publication. For these papers, I have also handled correspondence with journals. I assisted in the preparation of the manuscript for Paper 7 and developed and implemented some of the methods for which benchmarks were presented. In order to keep this thesis concise and focused on the key contributions, I have left some of my related publications outside the thesis. These publications can be found in Refs. 1–4. Further comments on the included papers can be found in Section 7.1 of the introductory chapters.

Acknowledgements

I gratefully acknowledge the supervision of this thesis by Paweł Sałek. Thank you for your belief in my ability to contribute to the field of theoretical chemistry. A special thanks to Elias Rudberg who has been a close collaborator in many of my projects. Thanks also to Hans Ågren for admitting me to PhD studies at the Department of Theoretical Chemistry at KTH.

Part of the work presented in this thesis has been carried out at the University of Southern Denmark (SDU), Odense, under the supervision of Hans Jørgen Aagaard Jensen. I am grateful for the hospitality that has been shown to me by Hans Jørgen, Hans Aage and Jette Nielsen, and Emmanuel Fromager during my stays in Odense.

Yet another part of the thesis work has been carried out during visits to the Los Alamos National Laboratory (LANL), New Mexico, USA. I am grateful to Anders Niklasson for inviting me to work with him in Los Alamos and for support also from Danielsson's foundation and from Pieter Swart and the Los Alamos mathematical modeling and analysis student program. Many thanks to Anders Niklasson, Nicolas Bock, Erik Holmström, and Matt Challacombe for fruitful collaborations and for sharing many reviving moments at the International Ten Bar Science Café under restless attendance by its eminent barista Travis Peery.

Thanks to all the people in the theoretical chemistry groups at KTH and SDU and the T-1 group at LANL for contributing to three nice working places. Thanks also to Sara Zahedi, Elias Rudberg, Peter Hammar, Paweł Sałek, and Ke–Yan Lian for valuable comments during the preparation of this thesis.

Finally, I would like to thank Sara Zahedi for joining me in the work on finding interior eigenpairs and for always being enthusiastic and encouraging about my work. The warmest thanks goes also to my supportive parents.

This research has been supported by the Sixth Framework Programme Marie Curie Research Training Network under contract number MRTN-CT-2003-506842 and the NorFA network in natural sciences "Quantum Modeling of Molecular Materials".

Contents

Part I

Introductory chapters

Chapter 1

Introduction

When the underlying physical laws of chemistry were established with the advent of quantum mechanics in the 1920's, researchers soon realized that the equations that come out of these laws are very complicated and computationally demanding to solve. Since then, theoretical chemists and physicists have come up with approximations that result in simpler equations and reduced computational demands. Many approximations exist today that can be applied in various combinations. These approximations can be roughly divided into two classes: model approximations and computational approximations. Model approximations provide simplified equations that describe the system under study under hypothetical conditions. Computational approximations simplify the solution and reduces the required computational effort for a given set of equations. Among model approximations, the Hartree–Fock and Kohn–Sham density functional theory methods allow for quantum mechanical treatment of relatively large systems.

The methods that traditionally have been used to solve the Hartree–Fock and Kohn–Sham equations require a computational effort that increases cubically with system size. This means that if the system size is doubled, the time needed to solve the equations is eight times longer. By the use of computational approximations, however, the complexity can, for many kinds of molecular systems, be reduced to linear. These approximations should ideally deliver trustworthy results with least possible use of computational resources. Coulomb interactions can be evaluated with computational resources proportional to the system size using the Fast Multipole Method (FMM). Linear scaling methods for the evaluation of Hartree–Fock exchange have also been presented. In case of Kohn–Sham density functional theory, the exchange–correlation contribution can be evaluated linearly as well. Have you successfully dealt with those parts? Then, the rest is matrix algebra which is the focus of this thesis.

1.1 Outline of thesis

This thesis consists of seven introductory chapters and seven papers. The introductory chapters are intended to create an interest for the included papers. The order of the papers more or less follows the disposition of the introductory chapters. In Chapter 2, the so–called self–consistent field method is described as a sequence of rotations of the occupied subspace. Chapter 2 introduces the key ideas of Paper 1 where a mathematical framework for control of approximations in self–consistent field calculations is presented. In Paper 2, this framework is used to control the forward error of density matrix purification. This thesis does to a large extent revolve around density matrix purification and Paper 2. In Chapter 3, density matrix purification as well as some alternative methods to construct the density matrix are discussed. Chapter 4 is an introduction to Paper 3 and discusses how interior eigenpairs of the so–called Fock/Kohn–Sham matrix can be efficiently computed. Chapter 5 concerns inverse factorizations that can be used for congruence transformations of the generalized eigenvalue problem. Methods for such inverse factorizations are proposed in Papers 4 and 6. In Chapter 6, sparse matrix data structures and methods to select small matrix elements for removal are discussed. This is an introduction to Papers 5 and 6. The introductory chapters end with some final remarks in Chapter 7. In the last paper, Paper 7, the overall performance of density matrix purification, as implemented using results of several of the previous papers, is demonstrated along with benchmarks of Coulomb and exchange matrix evaluations. The implementations used to evaluate the Coulomb and exchange matrices are described by Elias Rudberg in Ref. 5. All these implementations, of purification and Coulomb and exchange matrix evaluations, are part of the Ergo quantum chemistry program.[6] This program is able to perform Hartree–Fock and Kohn–Sham density functional theory calculations with a time and memory usage that increase only linearly with system size.

1.2 Notation

In the introductory chapters, the following notation will be used.

I will let n be the number of basis functions. The $n \times n$ *Fock/Kohn–Sham matrix* will be denoted by F_S where S is the *basis set overlap matrix*. If an orthonormal basis set is used, the Fock/Kohn–Sham matrix will be denoted by F as well. In this case, S is equal to identity and $F_S = F$. In the case of a non–orthonormal basis set, a matrix F can be obtained via a congruence transformation of F_S. Similarly, the symbols D and D_S will be used for the *density matrix*. I will let n_{occ} be the dimension of the so–called occupied

subspace.

The largest eigenvalue of F that belongs to the occupied part of the eigenspectrum will be referred to as the *Highest Occupied Molecular Orbital (HOMO) eigenvalue*. The smallest eigenvalue of F that belongs to the virtual part of the eigenspectrum will be referred to as the *Lowest Unoccupied Molecular Orbital (LUMO) eigenvalue*. The gap between the occupied and virtual parts of the eigenspectrum will be referred to as the *band gap* or the *HOMO–LUMO gap*. I will use the symbol μ for any value in but not outside the HOMO–LUMO gap. The value μ is often referred to as the *chemical potential*.

I will let $P_{\mathcal{X}}$ denote the matrix for orthogonal projection onto the subspace $\mathcal{X}$. Furthermore, $\|A\|_F$ will denote the *Frobenius norm* of A;

$$\|A\|_F = \sqrt{\sum_{ij} A_{ij}^2}, \tag{1.1}$$

and $\|A\|_2$ will denote the *Euclidean norm* of A;

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2. \tag{1.2}$$

The occupied subspace

In Hartree–Fock and Kohn–Sham calculations, the electron density is usually expanded in a set of n basis functions $\{\phi_i(r)\}$. In the following, a vector notation

$$\Phi^T(r) = [\phi_1(r)\ \phi_2(r)\ \ldots\ \phi_n(r)] \tag{2.1}$$

is used for the set of basis functions. The basis functions are often built up by combinations of polynomials and Gaussian functions centered at the nuclei of the molecule;

$$\phi(r) = p(r - r_0) \sum_i \beta_i e^{-\alpha_i (r-r_0)^2}. \tag{2.2}$$

Here r_0 is the center of a nucleus and p is a polynomial. These basis sets, which are usually referred to as Gaussian basis sets, are extensively discussed in Ref. 7. For simplicity, I will in the following assume that the basis set is orthonormal and return to the situation with a non–orthonormal basis set in Chapter 5.

For molecular systems with non–vanishing band gap, the electron density is uniquely defined by the so–called occupied subspace. Let $Q_{\mathcal{D}}$ be a $n \times n_{\text{occ}}$ matrix whose columns form an orthonormal basis for the occupied subspace $\mathcal{D}$ of dimension n_{occ}. The electron density $\rho(r)$ is given by

$$\rho(r) = \Phi^T(r) Q_{\mathcal{D}} Q_{\mathcal{D}}^T \Phi(r). \tag{2.3}$$

Therefore, given a basis set Φ, a search for the electron density is a search for the correct occupied subspace. The matrix for orthogonal projection onto the occupied subspace $D = Q_{\mathcal{D}} Q_{\mathcal{D}}^T$ is usually referred to as the density matrix.

A subspace $\mathcal{X}$ is an *invariant subspace* of a matrix A if and only if

$$\{Ax : x \in \mathcal{X}\} \subset \mathcal{X}. \tag{2.4}$$

The occupied subspace $\mathcal{D}$ is an invariant subspace of the density matrix D. Both the so–called Fock and Kohn–Sham matrices can be computed from the

density matrix, see Appendix A. Assume that for a given density matrix D with occupied invariant subspace $\mathcal{D}$, we compute a Fock/Kohn–Sham matrix F (according to Eq. (A.1) with $F_S = F$). If D is a solution to the Hartree–Fock/Kohn–Sham problem, then $\mathcal{D}$ is an invariant subspace of F as well. This is an immediate consequence of the Hartree–Fock/Kohn–Sham equations.[8] In other words, the electron density represented by D generates a field that, together with the external field, is self–consistent. Therefore, the method used to compute the density matrix D is usually referred to as the *self–consistent field method*. The occupied subspace being an invariant subspace of both F and D is important for the understanding of the self–consistent field method and the impact of computational approximations.

2.1 Rotations of the occupied subspace

As mentioned above, the Hartree–Fock/Kohn–Sham problem is usually solved using the self–consistent field method. In its simplest form, the self–consistent field method is a fixed point iteration where the two steps 1) construction of the Fock/Kohn–Sham matrix for a given electron density ($D \to F$) and 2) calculation of the density matrix for the resulting potential ($F \to D$) are repeated until convergence:

$$D_1 \longrightarrow F_2 \longrightarrow D_2 \longrightarrow F_3 \longrightarrow D_3 \longrightarrow \cdots . \tag{2.5}$$

Here, the Fock/Kohn–Sham matrix F_{i+1} is calculated from D_i according to Eq. (A.1). The density matrix D_i is calculated from the occupied invariant subspace of F_i, usually the subspace that corresponds to its n_{occ} smallest eigenvalues. Therefore, F_i and D_i share the same occupied invariant subspace. The $F \to D$ step will be further discussed in the next chapter. The $D \to F$ step is discussed by Rudberg in Ref. 5.

The desired fixed point, the self–consistent field solution, often has a rather small region of attraction. Usually, however, some convergence enhancing schemes are used to accelerate and hopefully even ensure convergence, see Refs. 9 and 10 for recent reviews. Examples include damping,[11,12] level shifting,[13] and so–called Direct Inversion in the Iterative Subspace (DIIS).[14,15]

The self–consistent field method can be seen as a sequence of rotations of the occupied subspace:

$$\mathcal{D}_1 \longrightarrow \mathcal{D}_2 \longrightarrow \mathcal{D}_3 \longrightarrow \cdots , \tag{2.6}$$

where $\mathcal{D}_i$ is the occupied invariant subspace of F_i and D_i. A rotation occurs every time a new Fock/Kohn–Sham matrix is constructed. In the calculation of the density matrix, no rotation would occur if exact arithmetics were used.

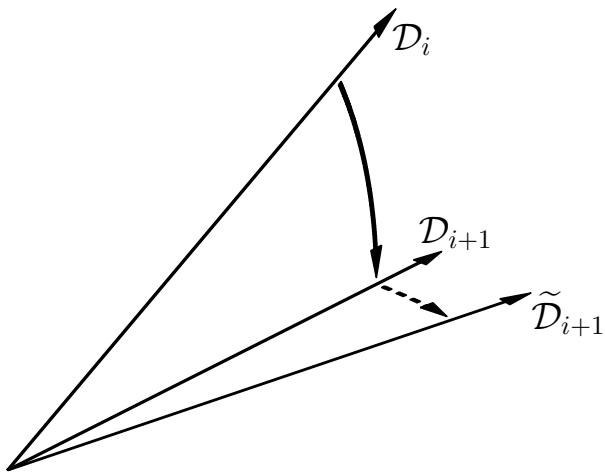


Figure 2.1: When computational approximations are used, erroneous rotations (⇢) distort the desired rotation (⟶) so that a perturbed subspace $\widetilde{\mathcal{D}}_{i+1}$ is obtained instead of the $\mathcal{D}_{i+1}$ subspace.

2.2 Erroneous rotations

In practice, both the $D \to F$ and $F \to D$ steps are carried out approximately in order to reduce the computational effort. Computational approximations such as Cauchy–Schwarz screening of integrals[8] and truncation of small matrix elements are frequently used. Consequently, a distorted subspace $\widetilde{\mathcal{D}}_{i+1}$ is obtained instead of $\mathcal{D}_{i+1}$ when F_{i+1} and D_{i+1} are computed from D_i. That is, an erroneous rotation

$$\mathcal{D}_{i+1} \dashrightarrow \widetilde{\mathcal{D}}_{i+1} \tag{2.7}$$

happens together with the desired

$$\mathcal{D}_i \to \mathcal{D}_{i+1} \tag{2.8}$$

rotation, see Figure 2.1. Note that in exact arithmetics, $\mathcal{D}_{i+1}$ is the occupied invariant subspace of both F_{i+1} and D_{i+1}. The approximate $\widetilde{\mathcal{D}}_{i+1}$ is the occupied invariant subspace of D_{i+1} only, but contains erroneous rotations coming from both the $D_i \to F_{i+1}$ and $F_{i+1} \to D_{i+1}$ steps.

How should computational approximations be measured and controlled? For the self–consistent field convergence, it is the accuracy of the occupied subspace that matters. The erroneous rotations of the occupied subspace should, intuitively, be small compared to the desired self–consistent field rotations. In practice, however, errors are often measured and controlled by their constituent parts rather than by their impact on the occupied subspace. For example, approximate evaluations of Coulomb and exchange interactions are usually performed by screening contributions to matrix elements that are below some predefined threshold value. This threshold value is often selected by trial and error or practical experience. To be able to select threshold values in a more systematic way we need relations between the erroneous rotations and the way we control approximations. In particular, when a rotation or difference between two subspaces is measured, one has to be careful so that rotations within the subspace do not take any part of the measure. For example, if a set of spanning vectors is used to represent the subspace, vectors within the set may rotate without changing the subspace.

A mathematical framework for the relation between approximations and erroneous rotations in self–consistent field calculations is presented in Paper 1. This framework is based on matrix perturbation theory from Refs. 16 and 17 with the key ingredient being canonical angles between invariant subspaces.

Chapter 3

Density matrix construction

One of the key operations in Hartree–Fock/Kohn–Sham calculations is to construct the density matrix D for a given Fock/Kohn–Sham matrix F, the $F \to D$ step of the previous chapter. Using an orthonormal basis set, the density matrix D is the matrix for projection onto the occupied invariant subspace $\mathcal{X}$ of F. This subspace is spanned by the eigenvectors of F that correspond to the n_{occ} smallest eigenvalues. Therefore, it is possible to construct the density matrix via a diagonalization of F:

$$FC_{\mathcal{X}} = C_{\mathcal{X}}\Lambda_{\mathcal{X}} \quad \Longrightarrow \quad D = C_{\mathcal{X}}C_{\mathcal{X}}^{T}. \tag{3.1}$$

Here $C_{\mathcal{X}}$ is a $n \times n_{\text{occ}}$ matrix that contains the eigenvectors that span $\mathcal{X}$ and $\Lambda_{\mathcal{X}}$ is a diagonal $n_{\text{occ}} \times n_{\text{occ}}$ matrix with the corresponding eigenvalues. The time needed to perform this operation generally grows cubically with system size. Also, standard diagonalization schemes usually make much effort to obtain accurate eigenvectors. Here, we want an accurate representation of the occupied invariant subspace but do not care about individual eigenvectors; any orthonormal basis $Q_{\mathcal{X}}$ for $\mathcal{X}$ would suffice. Several methods to obtain such a basis without direct diagonalization have been proposed, often based on Krylov subspace iterations.[18,19]

In a linear scaling method, $Q_{\mathcal{X}}$ would need to be sparse. Many possible choices of $Q_{\mathcal{X}}$ exist of which most are likely to be dense. The most difficult part for any method to efficiently construct a representation of the occupied subspace with $\mathcal{O}(n)$ memory and time usage is to enforce sparsity while controlling errors. In this chapter I will consider methods that employ the density matrix to represent the occupied subspace. Together with careful approaches to enforce sparsity, it is at least with some of these methods possible to control the occupied subspace error.

The aim of solving the eigenvalue problem in Eq. (3.1) is not always to obtain a representation of the occupied subspace. Therefore methods have been developed that avoid the full solution of Eq. (3.1) but that do not return

the density matrix. Examples include methods to estimate the distribution of eigenvalues[20] and methods to compute a subset of physically relevant eigenvectors, usually in a window around the band gap.[21,22] In this thesis I focus on self–consistent field calculations where a representation of the whole occupied invariant subspace is needed to compute a new Fock/Kohn–Sham matrix.

Methods of current interest can be divided into two classes: energy minimization and polynomial expansion methods.

3.1 Energy minimization

The correct density matrix D minimizes

$$\mathrm{Tr}[DF] \tag{3.2}$$

under the constraints $D = D^2$ (hereinafter *the idempotency condition*) and $\mathrm{Tr}[D] = n_{\mathrm{occ}}$ (hereinafter *the trace condition*).[23] The idea of energy minimization methods is to find a functional based on Eq. (3.2), somehow taking the constraints into account, and apply some minimization scheme such as the conjugate gradient method or Newton's method.[24]

3.1.1 First attempts

Li, Nunes, and Vanderbilt suggested to handle the idempotency condition by replacing the density matrix in Eq. (3.2) with its so–called McWeeny–purified version, and the trace condition by shifting the Fock/Kohn–Sham matrix with the chemical potential μ.[25] This results in the functional

$$\Omega_{\mathrm{LNV}}(D) = \mathrm{Tr}[(3D^2 - 2D^3)(F - \mu I)]. \tag{3.3}$$

Later Millam and Scuseria eliminated the need to know the chemical potential by a slight modification of the LNV functional;[26]

$$\Omega_{\mathrm{MS}}(D) = \mathrm{Tr}[(3D^2 - 2D^3)F] + \mu(\mathrm{Tr}[D] - n_{\mathrm{occ}}). \tag{3.4}$$

In this method the initial density matrix is chosen to have correct trace. The trace condition is then satisfied throughout the minimization by choosing μ so that the trace of the gradient $\nabla\Omega_{\mathrm{MS}}(D)$ is equal to zero in each step. Variants of these methods were presented by Daw[27] and Challacombe.[28]

Provided that the idempotency condition is satisfied, both the functionals above take properly the trace condition into account. The problem, however, is that the desired solution is not a global minimum because of the way the idempotency condition is handled. It is easy to construct a matrix that gives a lower functional value than the desired density matrix. As a consequence one has to be careful during the optimization and make sure not to leave the stable region.

3.1.2 Parametrized minimization

The idempotency problem can be avoided by use of an exponential parametrization of the density matrix. Given an approximate density matrix D_i that fulfills the trace and idempotency conditions, a new refined density matrix that fulfills the conditions as well, can be expressed in terms of an antisymmetric matrix X;[29]

$$D_{i+1} = e^X D_i e^{-X}. \tag{3.5}$$

Once again we modify Eq. (3.2) by inserting our expression for the density matrix and obtain the functional

$$\Omega(X) = \mathrm{Tr}[F e^X D_i e^{-X}]. \tag{3.6}$$

After improving X according to the used optimization scheme, for example by taking a step in the gradient direction, an improved density matrix D_{i+1} is given by Eq. (3.5). Eq. (3.5) can be evaluated using the so–called Campbell–Baker–Hausdorff expansion;[7]

$$e^X D_i e^{-X} = D_i + [X, D_i] + \frac{1}{2!}[X,[X,D_i]] + \frac{1}{3!}[X,[X,[X,D_i]]] + \ldots \tag{3.7}$$

where

$$[A, B] = AB - BA. \tag{3.8}$$

Exponential parametrization was used in Refs. 30–32. This is a conceptually appealing way of dealing with the idempotency and trace conditions. There are however some difficulties that need to be addressed when the exponential parametrization is applied. For example, it seems difficult to know how many terms of the expansion (3.7) to use. If too few terms are included, the conservation of idempotency could be lost. For this reason, an ad–hoc restriction of the step length, i.e. the magnitude of X, was used in Ref. 32. If many terms are included, idempotency could anyway be lost due to accumulating errors in the recursive evaluation. Furthermore, it is unclear how sparse the matrix X is and whether that sparsity has to be enforced by truncation of small matrix elements, and what impact that would have on accuracy and convergence. Possibly, some way to compute the exponential of a matrix can be adapted for evaluation of Eq. (3.5).[33]

3.2 Polynomial expansions

The density matrix can be defined using the step function

$$D = f(F) = \theta(\mu I - F) \tag{3.9}$$

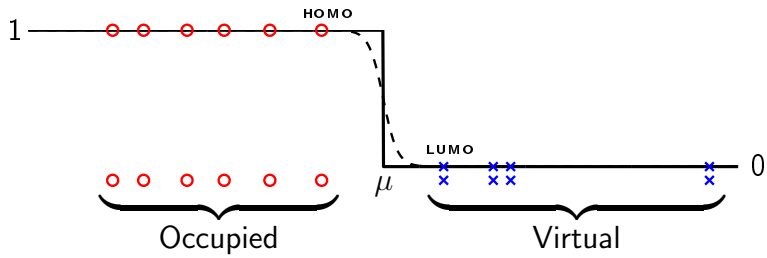


Figure 3.1: Schematic picture illustrating mapping by the step function $f(x)$ (solid line) of eigenvalues corresponding to occupied and virtual subspaces respectively. A function $p(x)$ (dashed line) that varies smoothly between 0 and 1 may be used to approximate $f(x)$. The only requirement on p is that it maps all eigenvalues to their desired values of 0 and 1.

where

$$\theta(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{otherwise} \end{cases} \tag{3.10}$$

is the Heaviside step function. By applying the step function $f(x)$, eigenvalues corresponding to the occupied and virtual invariant subspaces of F are mapped to 1 and 0 respectively. At first impression the discontinuity at μ may discourage any attempt to approximate this function by a polynomial expansion. However, in cases when there is a gap between the occupied and virtual parts of the eigenspectrum, the density matrix can be accurately computed without high resolution of the step; a polynomial $p(x)$ that varies smoothly between 0 and 1 in the gap may be used,[34,35] see Figure 3.1. I will discuss two different ways to construct a polynomial $p(x)$: Chebyshev expansion and iterative density matrix purification.

3.2.1 Chebyshev expansion

The Heaviside step function can be approximated by a truncated Chebyshev expansion. Straightforward application of the Chebyshev expansion leads, however, to Gibbs oscillations and errors that spread over the entire interval, see Figure 3.2. A possible remedy is to replace the Heaviside function with some function that varies smoothly from 0 to 1 in the HOMO–LUMO gap.[35–38] One choice is the Fermi–Dirac function

$$\frac{1}{1 + e^{-\beta(\mu - F)}} \tag{3.11}$$

where β depends on the HOMO–LUMO gap; the smaller the HOMO–LUMO gap is, the larger β is needed. In this way the oscillations are reduced. How-

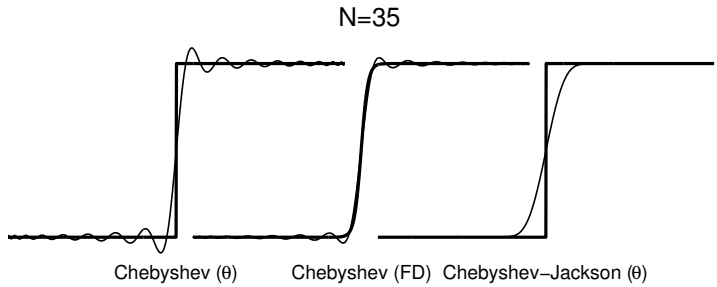


Figure 3.2: Chebyshev approximations of the Heaviside step function using $N = 35$ Chebyshev polynomials. Left: Chebyshev expansion of the Heaviside step function. Center: Chebyshev expansion of the Fermi–Dirac function with $\beta = 50$. Right: Chebyshev expansion of the Heaviside step function using the Jackson damping factors.

ever, the convergence towards 0 and 1 is anyway slow and purification, discussed in the following section, is often needed to get the eigenvalues sufficiently close to their desired values of 0 and 1.[38]

An alternative way to reduce the oscillations is to use some Gibbs damping factors in the Chebyshev expansion.[39] In the rightmost function depicted in Figure 3.2 the so–called Jackson kernel[39] has been used to reduce the oscillations. While the oscillations have been successfully damped, the slope at the inflexion point is smaller when the Jackson kernel is used.

3.2.2 Density matrix purification

Another way of constructing a polynomial $p(x)$ to approximate the step function, as depicted in Figure 3.1, is to iteratively apply low–order polynomials that push the eigenvalues towards 0 and 1 until convergence. Usually this is done by an initial linear transformation f_0 that maps the eigenvalues of F into the [0, 1] interval, followed by application of a sequence of polynomials $f_i,\ i = 1, 2, \ldots$ with fixed points at 0 and 1:

$$\begin{array}{rcl} X_0 & = & f_0(F) \\ X_i & = & f_i(X_{i-1}) \end{array} . \tag{3.12}$$

Already in 1956, McWeeny suggested to use the polynomial $x^2(3 - 2x)$ to refine matrices that are roughly idempotent.[23] During the last decade researchers have realized that purification transforms similar to the McWeeny polynomial can be used to improve the computational complexity of the en-

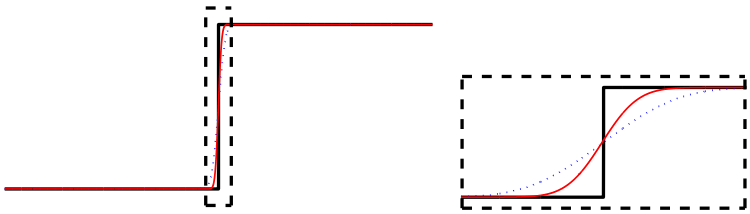
tire $F \rightarrow D$ step. In 1998, Palser and Manolopoulos presented two purification algorithms that both fit into the general scheme of Eq. (3.12).[40] In the first one, which they refer to as *grand–canonical purification*, f_0 is such that the occupied and virtual parts of the eigenspectrum of X_0 end up in the [0.5, 1] and [0, 0.5] intervals respectively. After that, the McWeeny polynomial is used to push eigenvalues to their desired values. Because of the fixed point at 0.5, the chemical potential μ is conserved throughout the iterations. Palser and Manolopoulos also propose *canonical purification* which does not require knowledge of the chemical potential for the initial transformation. This method conserves the trace instead of the chemical potential; the initial transformation makes sure that X_0 has the correct trace instead of correct chemical potential. After that, the polynomials are chosen such as to conserve the trace while pushing eigenvalues towards 0 and 1.

Subsequent to the work by Palser and Manolopoulos, a number of purification algorithms have been proposed.[41–46] Most of these proposals focus on finding polynomials that optimize the convergence of eigenvalues. Significant improvements were also made in this respect. The main difficulty with purification lies, however, in preserving the occupied subspace throughout the iterations when truncations are applied to maintain sparsity. One of the key contributions of this thesis is a solution of this issue which is presented in Paper 2.

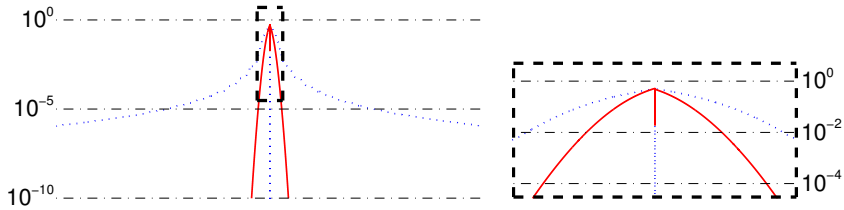
In Paper 2 we use the so–called trace–correcting purification polynomials x^2 and $2x - x^2$ suggested by Niklasson.[41] Using these low order polynomials is advantageous because only a single matrix multiply is needed in each iteration. This simplifies error control and reduces the need to store intermediate matrices.

3.2.3 Polynomial evaluation

The Chebyshev expansion and density matrix purification methods construct in different ways a high order polynomial that approximates the step function in Eq. (3.9). A polynomial like the Chebyshev expansion can be efficiently evaluated using the method proposed by Paterson and Stockmeyer.[47] With this method, a polynomial of order 25 can, for example, be evaluated with 8 matrix–matrix multiplications. Although this represents a significant improvement compared to straightforward evaluation, iterative construction of polynomials such as the one used in density matrix purification is much more efficient. By recursive application of low order polynomials, the polynomial degree increases exponentially with the number of matrix–matrix multiplications. By repeated application of x^2 one can for example generate a polynomial of order 2^m with only m multiplications. The difference between the two methods is illustrated in Figure 3.3.



(a) A step function $f(x)$ (Solid black lines) approximated by a Chebyshev–Jackson expansion $p_{\mathrm{CJ}}(x)$ (dotted blue lines) and purification $p_{\mathrm{P}}(x)$ (solid red lines). The right figure shows a closeup of the step.



(b) Absolute errors ($|f(x)-p_{\mathrm{X}}(x)|$) of the two step function approximations, Chebyshev–Jackson (X = CJ, dotted blue lines) and purification (X = P, solid red lines), depicted in Panel (a).

Figure 3.3: Comparison of two different ways to approximate a step function: Chebyshev expansion and purification methods. The methods have been allowed to use 20 matrix–matrix multiplications each. This gave a polynomial degree of 121 for the Chebyshev expansion and $2^{20} = 1048576$ for the purification method. The solid red lines show the result for purification. The purifying polynomials x^2 and $2x-x^2$ have been applied. The dotted blue lines show the result for a Chebyshev–Jackson approximation of the step function using the Paterson–Stockmeyer polynomial evaluation method.

3.3 Accuracy

Accuracy refers to the closeness of a computed solution to the exact solution of the problem under consideration. The accuracy of a solution to a problem does not only depend on the algorithm used to solve the problem but also on the problem itself.

3.3.1 Conditioning

A problem is said to be well–conditioned if its solution is insensitive to perturbations in the input data. In other words, for a well–conditioned problem, a small change in the input results in a small change in the exact solution. A *condition number* associated with a problem is a measure on the conditioning of the problem. A problem with low condition number is well–conditioned whereas a problem with high condition number is ill–conditioned. For the problem of computing a new density matrix for a given Fock matrix, the condition number can be defined as

$$\kappa_F = \lim_{h\to 0} \sup_{A:\|A\|_2=\|F\|_2} \frac{\|D(F+hA) - D(F)\|_2}{h}. \tag{3.13}$$

Here, I used the notation $D(F)$ for the exact density matrix corresponding to a given Fock/Kohn–Sham matrix F. In Paper 2 we show that

$$\kappa_F = \frac{\|F\|_2}{\xi} \tag{3.14}$$

where ξ is the size of the HOMO–LUMO gap. It is well–known that problems with small HOMO–LUMO gaps, arising for example from metallic systems, are difficult to solve. Eq. (3.14) gives a mathematical explanation to these difficulties; as $\xi \to 0$, $\kappa_F \to \infty$. If the HOMO–LUMO gap vanishes, the problem does not even have a unique solution. Because of Eq. (3.14), I expect all density matrix construction methods to run into difficulties as ξ decreases. The difficulties may just become manifest in different ways. Density matrix purification, for example, will for small gaps require more iterations to converge and tighter threshold values to conserve the occupied subspace. This is thoroughly discussed in Paper 2. In energy minimization methods small gaps are likely to result in shallow minima which will make the convergence of for example the conjugate gradient method more sensitive to perturbations. In some cases, however, a small HOMO–LUMO gap should be regarded as a sign of a modeling problem for the studied molecular system. For example, incorrect protonation can lead to small gaps and difficulties to converge the whole self–consistent field procedure.[48]

3.3.2 Forward error analysis

Error analysis can be carried out in different ways. The natural way is to analyze the difference between the computed result and the exact solution. This difference is sometimes referred to as the *forward error*. The *backward error* is the smallest possible change in input for which the exact solution is equal to the approximate solution of the original problem. In other words, the backward error is obtained by analyzing which problem the algorithm actually solved. Here, I shall focus on the forward error.

Let D denote the exact density matrix corresponding to the Fock/Kohn–Sham matrix F and let $\widetilde{D}$ denote the approximate matrix obtained by the applied algorithm. The forward error can be defined as

$$\varepsilon = \|\widetilde{D} - D\|_2. \tag{3.15}$$

In Paper 2, the forward error of density matrix purification is analyzed and schemes to control the forward error are proposed. A key in the analysis in Paper 2 is to distinguish between errors in the occupied invariant subspace $\widetilde{\mathcal{X}}$ of $\widetilde{D}$ and errors in eigenvalues. The forward error is separated into two parts;

$$\underbrace{\|\widetilde{D} - D\|_2}_{\equiv\varepsilon} \leq \underbrace{\|\widetilde{D} - P_{\widetilde{\mathcal{X}}}\|_2}_{\equiv\varepsilon^\lambda} + \underbrace{\|P_{\widetilde{\mathcal{X}}} - D\|_2}_{\equiv\varepsilon^\Theta}. \tag{3.16}$$

Here, the first norm on the right hand side, ε^λ, measures only deviations of the eigenvalues of $\widetilde{D}$ from 0 and 1, and the second norm on the right hand side, ε^Θ, measures only errors in the occupied subspace; recall that $D = P_{\mathcal{X}}$. We discuss in Paper 2 that in density matrix purification, ε^λ is expected to be large in the early iterations and decrease as the polynomial expansion approaches a step function. The subspace error, ε^Θ, is small in the early iterations but grows as the purification proceeds. We also show in Paper 2 that the distinction provided by Eq. (3.16) gives a natural convergence criterion; as soon as the forward error is dominated by ε^Θ, it is time to stop the purification process.

The separation of the forward error given by Eq. (3.16) is likely to be useful also for error and convergence analysis in other density matrix methods. In particular, the parameterized energy minimization is likely to benefit from such an analysis. In this class of methods, the subspace error is expected to decrease over the iterations whereas the error in eigenvalues is expected to be small. The parameterized energy minimization is in a way opposite to density matrix purification: In purification, one tries to move the eigenvalues to their desired values while conserving the occupied subspace. In parameterized energy minimization, one tries to rotate the occupied subspace while conserving the eigenvalues.

Chapter 4

Calculation of interior eigenpairs

The schemes for error control in density matrix purification, presented in Paper 2 and discussed in the previous chapter, require knowledge of the HOMO–LUMO gap. The reason for this can be understood by the condition number in Eq. (3.14); the smaller HOMO–LUMO gap, the larger condition number and the tighter computational threshold values are needed to achieve a certain level of accuracy. However, it is not straightforward to obtain information about the HOMO–LUMO gap because the HOMO and LUMO eigenvalues are interior eigenvalues, see for example Figure 3.1.

It is well known that most iterative methods designed to solve large sparse eigenvalue problems provide rapid convergence to well–separated extremal eigenvalues.[49] Unfortunately, these methods need to take many steps to compute approximations to interior eigenvalues, if they converge at all. Therefore, spectral transformations that move desired interior eigenvalues to the ends of the eigenspectrum are frequently employed.

4.1 Spectral transformations

The most frequently used spectral transformation is the *shift–and–invert* transformation, see for example Ref. 50. An iterative method for eigenvalues is applied to

$$(F - \sigma I)^{-1} \tag{4.1}$$

instead of to F. Here, σ should be chosen so that desired eigenvalues lie close to σ. The eigenvalues $\{\lambda_i\}$ of F are related to the eigenvalues $\{\nu_i\}$ of $(F - \sigma I)^{-1}$ by

$$\nu_i = \frac{1}{\lambda_i - \sigma}. \tag{4.2}$$

This transformation transforms desired eigenvalues close to σ to well separated extremal eigenvalues, see Figure 4.1(a). A drawback of this approach is that

the usual computational kernel matrix–vector multiplication is replaced by relatively expensive solutions of linear systems of the type $(F - \sigma I)x = b$.

A spectral transformation that avoids the solution of linear systems is the *shift–and–square* transformation.[21] The iterative eigenvalue method is applied to

$$(F - \sigma I)^2. \tag{4.3}$$

Although the desired eigenvalues are transformed to extremal eigenvalues, a drawback is that the eigenspectrum is quadratically compressed around the desired eigenvalues, see Figure 4.1(a).

In electronic structure calculations, eigenpairs of interest are usually located around the HOMO–LUMO gap. Recently, Xiang and coworkers realized that the density matrix can be used to project undesired parts of the eigenspectrum out of the way so that eigenvalues close to the HOMO–LUMO gap become extremal and more easily can be computed.[51] In this *shift–and–project* method, the largest eigenvalues of

$$D(F - \sigma_1 I) \tag{4.4}$$

end the smallest eigenvalues of

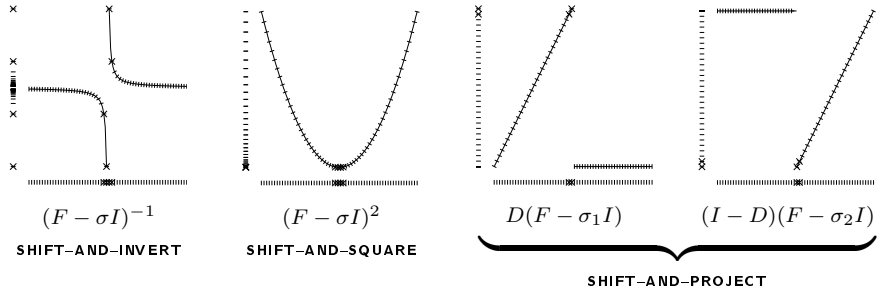
$$(I - D)(F - \sigma_2 I) \tag{4.5}$$

are computed, see Figure 4.1(a). Here, σ_1 and σ_2 are lower and upper bounds of the eigenspectrum. I recall that D is the matrix for orthogonal projection onto the occupied subspace and that $I - D$ is the complementary projection matrix that projects onto the virtual subspace. The shift–and–project method nicely avoids the solution of linear systems and the transformations do not affect the separation of eigenvalues. However, utilizing the ability of purification to give good separation between interior eigenvalues, we can do even better.

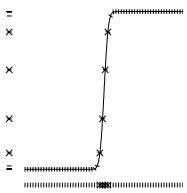
4.2 Utilizing density matrix purification

Purification has previously been used for interior eigenvalue calculations to obtain proper shift–values for the shift–and–transform methods[2] and to compute the density matrix, explicitly occurring in the shift–and–project method.[51]

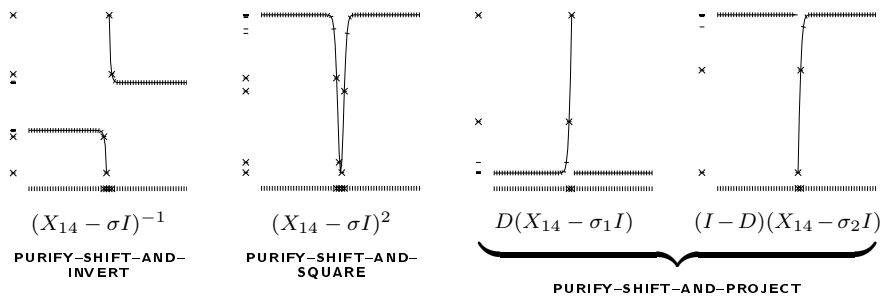
In Paper 3 we show that it is advantageous to apply the shift–and–transform methods, described in the previous section, to some intermediate purification matrix rather than to the Fock/Kohn–Sham matrix. This was motivated by the observation that interior eigenvalues of intermediate purification matrices are well–separated, see Figure 4.1(b). Figure 4.1(c) shows the eigenvalue filters obtained by application of purification followed by the shift–and–transform methods of the previous section. The purify–shift–and–transform



(a) Eigenvalue mapping by three shift–and–transform techniques applied directly to a matrix F with equidistant eigenvalues.



(b) Eigenvalue mapping by fourteen density matrix purification iterations applied to a matrix with equidistant eigenvalues.



(c) Eigenvalue mapping by the three purify–shift–and–transform techniques applied to a matrix with equidistant eigenvalues. The matrix X_{14} is the result of fourteen purification iterations applied to the matrix with equidistant eigenvalues, see Panel (b). After purification, the three shift–and–transform techniques are applied. Compare with Panel (a).

Figure 4.1: Comparison of the shift–and–transform techniques with and without utilization of purification. Four desired interior eigenvalues are marked with crosses.

methods all give very good separation between eigenvalues. Indeed, for the cases presented in Paper 3, the Lanczos method on average finds an eigenvalue more often than every second iteration when purification is utilized. As a comparison, when the shift–and–square and shift–and–project methods are used without utilizing purification, the Lanczos method on average finds an eigenvalue more seldom than every ninth iteration.

A drawback of the purify–shift–and–project method is that the density matrix is needed so that interior eigenvalues cannot be calculated on–the–fly during the purification procedure. For this reason, we use the purify–shift–and–square method for the calculation of the HOMO and LUMO eigenvalues in Paper 2.

Chapter 5

Inverse factorizations

In the previous chapters, an orthonormal basis set was assumed. In practice, one often works with non–orthonormal basis sets and the eigenvalue problem in Eq. (3.1) takes the generalized form

$$F_S C_{\mathcal{Y}} = S C_{\mathcal{Y}} \Lambda_{\mathcal{Y}} \quad \Longrightarrow \quad D_S = C_{\mathcal{Y}} C_{\mathcal{Y}}^T. \tag{5.1}$$

Here, S is the symmetric positive definite basis set overlap matrix which is equal to identity for orthonormal basis sets. A non–orthonormal basis set makes it more complicated to find a representation of the occupied subspace. There are essentially two ways to deal with non–orthonormal basis sets: 1) One can directly search for the eigenspace $\mathcal{Y}$ of $(F_S,\ S)$ instead of the invariant subspace $\mathcal{X}$ of F.[31, 52–54] 2) One can transform the generalized problem to standard form by the means of a congruence transformation, solve the problem in standard form, and transform back. Here, I will focus on the latter approach.

5.1 Congruence transformations

In order to transform Eq. (5.1) to the form of Eq. (3.1), we need an inverse factor Z of S such that

$$Z^T S Z = I. \tag{5.2}$$

Eq. (5.1) is then related to Eq. (3.1) by

$$\begin{aligned} Z C_{\mathcal{X}} &= C_{\mathcal{Y}} && (5.3)\\ F &= Z^T F_S Z && (5.4)\\ \Lambda_{\mathcal{X}} &= \Lambda_{\mathcal{Y}}. && (5.5) \end{aligned}$$

If, for example, density matrix purification is used to obtain the density matrix, the computational procedure looks like Algorithm 1. In a self–consistent

Algorithm 1 Congruence transformations and purification

1: $F = Z^T F_S Z$
2: **Purification** (See Section 3.2.2.)
$$\begin{aligned} X_0 &= f_0(F) \\ X_i &= f_i(X_{i-1}), \quad i = 1, 2, \ldots, m \\ D &= X_m \end{aligned}$$
3: $D_S = ZDZ^T$

field calculation, the inverse factor can be computed before entering the self–consistent field loop, once and for all, since the basis set does not change throughout the calculation. For this reason, the calculation of the inverse factor usually constitutes a small part of the entire calculation time, at least for system sizes that are within reach today. Ill–conditioned overlap matrices make it more difficult to find inverse factors. In such cases it might be preferable to not compute any inverse factor but directly search for the eigenspace $\mathcal{Y}$ of $(F_S,\ S)$ instead. However, an ill–conditioned overlap matrix can often be regarded as an indication of a basis set problem.

5.2 Inverse factors

How do we find an inverse factor? To begin with, the inverse factor will have to be sparse in order to be useful in linear scaling calculations. The purpose of applying the congruence transformation in the first place is to be able to use methods that can take advantage of sparsity in F and D. If Z would be significantly denser than F and D, some of the performance gained by using methods like density matrix purification would be lost in the inverse factorization and the congruence transformation. Fortunately, at least some of the possible choices of inverse factors appear to be sparse. For example, in Paper 7 we use the inverse Cholesky factor in Hartree–Fock calculations on water droplets and polypeptide helices. The storage requirements for the inverse Cholesky factors were in those calculations no worse than the storage requirement for the Fock and density matrices. Linear scaling calculations of the inverse square root $S^{-1/2}$ have been presented as well, by Jansík and coworkers.[55]

In the following, I will discuss algorithms to construct inverse Cholesky factors and an algorithm that finds an inverse factor that does not bear a name.

5.2.1 Inverse Cholesky calculation

One possible choice of Z is the inverse Cholesky factor which is an upper triangular matrix such that $S^{-1} = ZZ^T$. This factor was for example used by Millam and Scuseria together with their energy minimization method to calculate the density matrix.[26] Later, Challacombe suggested[28] that the so–called AINV algorithm by Benzi and coworkers[56] could be used to obtain the inverse Cholesky factor. This algorithm was originally developed to produce preconditioners for the solution of linear systems. Several variants of the AINV algorithm exist.[57,58] In Paper 6 we suggest a recursive variant of the blocked AINV algorithm[58] for hierarchic data structures. The AINV algorithms rapidly calculate inverse factors for systems with tens of thousands of basis functions. However, without truncation of small matrix elements during the procedure, these inverse factorizations often require much memory. Whereas many elements of the final result are often negligible, it is not straightforward to remove the small elements during the procedure because of interdependencies between the columns of Z occurring in the computational procedure. These interdependencies also make the algorithm difficult to parallelize.

5.2.2 Iterative refinement and recursive inverse factorization

If an approximate inverse factor Z_0 is known, iterative refinement can be used to systematically reduce the factorization error

$$\|Z_0^T S Z_0 - I\|_2 \tag{5.6}$$

until it is within desired accuracy.[59] In this algorithm, polynomials in the inverse factorization error matrix

$$\delta_0 = Z_0^T S Z_0 - I \tag{5.7}$$

are used to refine the inverse factor, see Paper 4. Since this algorithm uses only matrix–matrix multiplications, it is potentially easier to parallelize than inverse Cholesky algorithms. Truncation of small matrix elements is less worrisome since we do not have any error accumulation between the iterations. A sufficient condition for the iterative refinement to converge is that

$$\|\delta_0\|_2 < 1. \tag{5.8}$$

One way to obtain an initial approximate inverse factor Z_0 is to drop small matrix elements during the inverse Cholesky calculation of the previous section. It is however difficult to know what the drop tolerance should be to ensure that inequality (5.8) is fulfilled. In Paper 4, we present recursive inverse factorization which is a way to recursively apply iterative refinements

to obtain approximate inverse factors that fulfill inequality (5.8). The key of that paper is a theorem presented here in a slightly less general form:

Theorem 1. *Let $S = S_0 + \Delta S$ be a symmetric positive definite matrix partitioned as*

$$S_0 = \begin{pmatrix} A & 0 \\ 0 & C \end{pmatrix} \qquad \Delta S = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix} \tag{5.9}$$

and let $Z_A^T A Z_A = I$ and $Z_C^T C Z_C = I$ be inverse factorizations of A and C. Furthermore, let

$$Z_0 = \begin{pmatrix} Z_A & 0 \\ 0 & Z_C \end{pmatrix}. \tag{5.10}$$

Then, $\|Z_0^T S Z_0 - I\|_2 < 1$.

The proof of this theorem, presented in Paper 4, is based on some results by Haynsworth on the Schur complement.[60] Based on Theorem 1, we construct in Paper 4 a recursive algorithm to compute inverse factors of symmetric positive definite matrices. By the theorem above, it is clear that we are ready to start the iterative refinement procedure as soon as we have inverse factors Z_A and Z_C of the A and C matrices. This immediately suggests a recursive procedure where A and C are decomposed in the same way as S and so forth. This recursive procedure can continue all the way down to single matrix elements for which inverse factorizations are trivial. A more efficient approach would however be to let the decomposition recursively continue until the submatrices are small enough so that for example some inverse Cholesky algorithm can be used to compute the required inverse factors. Then iterative refinement is used to recursively assemble the inverse factor of S.

While the theorem guarantees that Eq. (5.8) is always fulfilled, the convergence speeds of the iterative refinements depend on the decomposition of S. In Paper 4 we show that recent advances in network theory can be used to obtain improved decompositions.

Chapter 6

Sparse matrix representations

Sparsity in matrices and efficient access to nonzero matrix elements are imperative for the efficiency of the algorithms discussed in the previous chapters. In particular, these algorithms depend on fast matrix–matrix multiplication.

Each element in the Fock/Kohn–Sham matrix F, the overlap matrix S, and the density matrix D corresponds to two basis functions centered at two atom centers of the molecule, see Figure 6.1. The magnitude of a matrix element generally depends on the distance between the two atom centers; if the distance is large, the corresponding matrix element is likely to be of small magnitude. If small matrix elements are removed, we can take advantage of the matrix sparsity that appears for extensive molecular systems. However, the removal of small matrix elements is an approximation which can introduce significant errors, if not done carefully. As discussed in Chapter 2, errors incurred by approximations can be seen as erroneous rotations of the occupied subspace.

Two questions should be addressed when sparse matrix representations are used in self–consistent field calculations: 1) How to remove small matrix elements while being in control of the occupied subspace? 2) How to store and operate on only nonzero matrix elements with high performance?

6.1 How to select small matrix elements for removal

We wish to remove small matrix elements that contribute little to the overall accuracy. More precisely, we would like to remove elements in such a way that we are in control of errors in the occupied subspace. Removal of small matrix elements (herein often referred to as *truncation*) can be seen as a perturbation E of the original matrix X so that the matrix after truncation

$$\widetilde{X} = X + E. \tag{6.1}$$

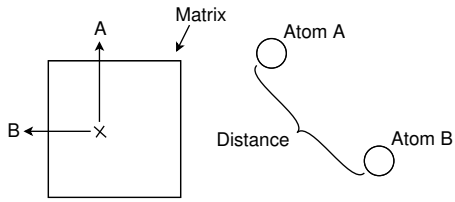


Figure 6.1: Each matrix element corresponds to two basis function centers, usually located at atom centers. Hence, each element is associated with a distance between two atoms.

It is not obvious when a matrix element should be considered to be negligible. One popular approach has been to remove all elements that correspond to an interatomic distance larger than some predefined cutoff radius.[25,31,32,37,38,40,61] If matrix elements are grouped into submatrices, each submatrix corresponds to two groups of atoms. In this case, the submatrix is dropped if the shortest distance between the two groups is greater than the predefined cutoff radius. Another approach to remove small elements is to drop all matrix elements below some predefined threshold value.[26,30,62] If elements are grouped into submatrices, a submatrix is dropped when its norm is below the threshold.[28,42] Unfortunately, a direct relation between threshold value and accuracy in the occupied subspace has not been known for any of these methods.

In Paper 1 we show that controlling the occupied subspace amounts to controlling some unitary–invariant norm of the error matrix and knowing the HOMO–LUMO gap. In Paper 5 we argue that the Euclidean norm is a suitable choice of unitary–invariant norm and propose Euclidean norm based truncation schemes. In these methods, small matrix elements are removed while ensuring that

$$\|E\|_2 \leq \tau \tag{6.2}$$

where the threshold value τ should be chosen based on the HOMO–LUMO gap and the desired accuracy in the occupied subspace.

Previously, we have presented truncation schemes based on matrix norms that can be calculated directly from the matrix elements.[1] In these schemes, norms or squared norms of submatrix blocks are computed and placed in ascending order in a vector. Truncation is executed by removing submatrices corresponding to the values in the vector. In case of the Frobenius norm, for example, the squared Frobenius norms of all nonzero submatrices are placed in a vector v. Removing all submatrices corresponding to elements in v from

v_1 to v_i gives an error matrix with norm

$$\|E\|_F = \sqrt{\sum_{j=1}^{i} v_j}. \tag{6.3}$$

It is thus trivial to find the index i for which as many submatrices as possible are removed while $\|E\|_F \leq \tau$.

The Euclidean norm, however, cannot be trivially calculated from the matrix elements and some more sophistication is needed in this case. In Paper 5 we take the Frobenius norm based truncation as a starting point and use a bisection scheme to find a proper index for which Eq. (6.2) is satisfied. In this bisection scheme, Euclidean norms of error matrix candidates are evaluated by calculation of the largest magnitude eigenvalues of the error matrices using the Lanczos method. We exploit certain Ritz value convergence patterns to reduce the total number of Lanczos iterations. Euclidean norm based truncation is an important part of the density matrix purification algorithm presented in Paper 2. Truncation based on unitary–invariant norms should, however, be an important part of any code that enforces sparsity and implements the framework of Paper 1.

6.2 How to store and access only nonzero elements

After removal of small matrix elements according to the previous section, we hopefully have a matrix with many zero matrix elements. The best possible data structure for a sparse matrix depends on the number of nonzero matrix elements and the nonzero pattern of the matrix. If the matrix has very few nonzero elements (in the order of ten nonzero elements per row) that are scattered all over the matrix, it is reasonable to use some data structure which stores and addresses element by element. If the matrix has many nonzero elements per row that lie close to each other, it can be advantageous to use a blocked data structure. Often, data locality can be improved by permuting the rows and columns of the matrix.

6.2.1 Permutations for data locality

The matrices that occur in Hartree–Fock and Kohn–Sham calculations with Gaussian basis sets often have thousands of nonzero elements per row. The matrices are to be regarded as semi–sparse rather than sparse. For this reason, a blocked sparse data structure is usually employed where nonzero matrix elements are grouped into submatrix blocks. The use of a blocked data structure can significantly increase the performance of matrix operations. However,

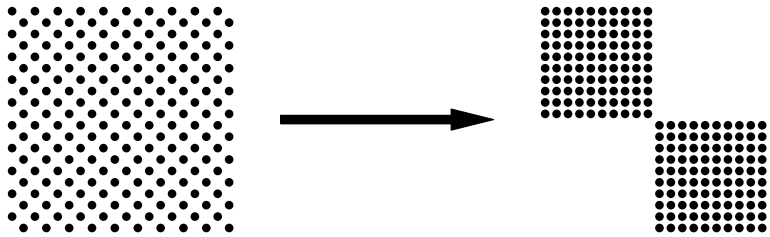


Figure 6.2: This figure illustrates how permutations of matrix rows and columns can result in improved data locality. Each dot corresponds to a nonzero matrix element. The left matrix has perfect non–locality in the sense that no nonzero matrix element is an immediate neighbor of another nonzero matrix element. In the right matrix the data locality has been dramatically improved by a permutation of the rows and columns of the left matrix.

grouping nonzero matrix elements into submatrices without loosing sparsity is only possible if the nonzero elements are neighbors. In Figure 6.2, the nonzero patterns of two matrices with 50% nonzero elements each are depicted. The figure shows that for the same sparsity level, the data locality can be quite different. In this case, however, the left matrix can be transformed to the right matrix by a permutation of its rows and columns.

The distance–magnitude relation described earlier in this chapter can be used to achieve data locality in the overlap, Fock/Kohn–Sham, and density matrices. Challacombe grouped basis functions belonging to the same atom and formed in this way *atom blocks*.[28] Later, it was recognized that the performance in submatrix operations, using standard dense matrix linear algebra libraries,[63–66] could be improved if basis functions belonging to several neighboring atoms were grouped into larger *multi–atom blocks*.[61] With atom or multi–atom blocks, the block size cannot be chosen at will. In Paper 6, however, we use a uniform block size. This means that basis functions centered on the same atom are not restricted to be in the same block. Using a uniform block size makes it easier to tune the block size with respect to the performance of a dense matrix library. Alternatively, one can tune small dense matrix operations for a single given block size.

Approaches that are not explicitly based on the distance–magnitude relation also exist. In Paper 4, for example, we suggest that the matrix can be seen as the connection matrix for a weighted network. This allows for network theory to be used to find community divisions of the network which correspond to a permutation and block partitioning of the matrix.

6.2.2 Data structures

In dense matrix data structures, the locations of and access patterns to matrix elements are completely known at compile time. In sparse matrix data structures, the locations of nonzero matrix elements are usually not known until runtime. Therefore, a sparse data structure needs to contain information about the locations of the nonzero matrix elements in the matrix and in computer memory. One of the most commonly used sparse matrix data structures is the compressed sparse column data structure,[67,68] used for example in the technical computing program Matlab.[69]

The compressed sparse column representation of a matrix is given by three one–dimensional arrays. Two arrays list the row indices and the numerical values of all nonzero matrix entries (in column–wise order). The third array lists column pointers where the i:th element is the index, in the two other arrays, of the first nonzero element of the i:th column of the matrix. For example, consider the matrix

$$A = \begin{pmatrix} 7 & & -3 & \\ & 8 & & 5 \\ & & 1 & \\ & -2 & & 6 \end{pmatrix}$$

A compressed sparse column representation of A is given by

$$\begin{array}{rccccccc} \textit{column pointers} = & 0 & 1 & & 3 & & 5 & & 7 \\ & \downarrow & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ \textit{row indices} = & 0 & 1 & 3 & 0 & 2 & 1 & 3 & \\ \textit{numerical values} = & 7 & 8 & -2 & -3 & 1 & 5 & 6 & \end{array}$$

In addition to the memory needed to store the numerical values, this data structure requires memory for the integer arrays used to keep track of the numerical values. When this data structure is used for semi–sparse matrices with many nonzero elements per row, the addressing overhead is unnecessarily large. Therefore, Hartree–Fock/Kohn–Sham codes often use blocked compressed sparse column data structures to reduce the overhead and take advantage of the performance of dense matrix libraries for submatrix–submatrix operations.[61,70,71]

In the blocked compressed sparse column data structure, the two indexing arrays are used to address entire submatrix blocks instead of single matrix elements. This is advantageous if matrix elements can be reorganized so that nonzero elements end up in some submatrices and zero elements end up in others as discussed previously. With our example matrix A this can be done by switching the second and the third rows and columns and block the matrix into four 2×2–submatrices. Then, a blocked compressed sparse column

representation of the reordered matrix is given by

$$\begin{array}{rccc} \textit{column pointers} = & 0 & 1 & 2 \\ & \downarrow & \downarrow & \downarrow \\ \textit{row indices} = & 0 & 1 & \\ \textit{submatrices} = & \begin{pmatrix} 7 & -3 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 8 & 5 \\ -2 & 6 \end{pmatrix} & \end{array}$$

We did not, however, manage to squeeze out all zero elements. When a blocked data structure is applied, there is often a trade off between squeezing out zero elements and reducing the addressing overhead.

Because of the indirect way of addressing the matrix elements, the compressed sparse column data structure is rather tiresome to deal with. Algorithms for this data structure take time to develop and often become complicated. However, when matrix elements are grouped into submatrices, the addressing overhead problem that we try to solve with the compressed sparse column data structure becomes less pronounced. For large enough submatrix blocks, most work is performed at the block–block level, making the overhead negligible. This allows for the use of simpler data structures.

In Paper 6 we suggest the use of a hierarchic sparse matrix data structure. The main advantage with this data structure is that new algorithms are easier to develop and implement. Another advantage is that matrix symmetry can easily be exploited, giving a speedup close to 2 for many operations. Hierarchic data structures have previously been used to reduce cache misses in dense matrix operations, see for example Ref. 72. Whereas an improved cache hit rate may be a positive side effect with the data structure proposed in Paper 6, our main motivation for using a hierarchy is to utilize sparsity in an efficient and transparent manner.

6.3 Performance

In Paper 7, the performance of the Ergo quantum chemistry program is demonstrated. In Figure 6.3, I have reproduced two of the performance graphs of density matrix purification from that paper. Density matrix purification is implemented as described in Paper 2 with ingredients from Papers 3, 5, and 6. The symmetric matrix square operation $S = \alpha T^2 + \beta S$ of Paper 6 is of particular importance for the performance. The figure shows that our implementation accomplishes linear scaling in time and memory.

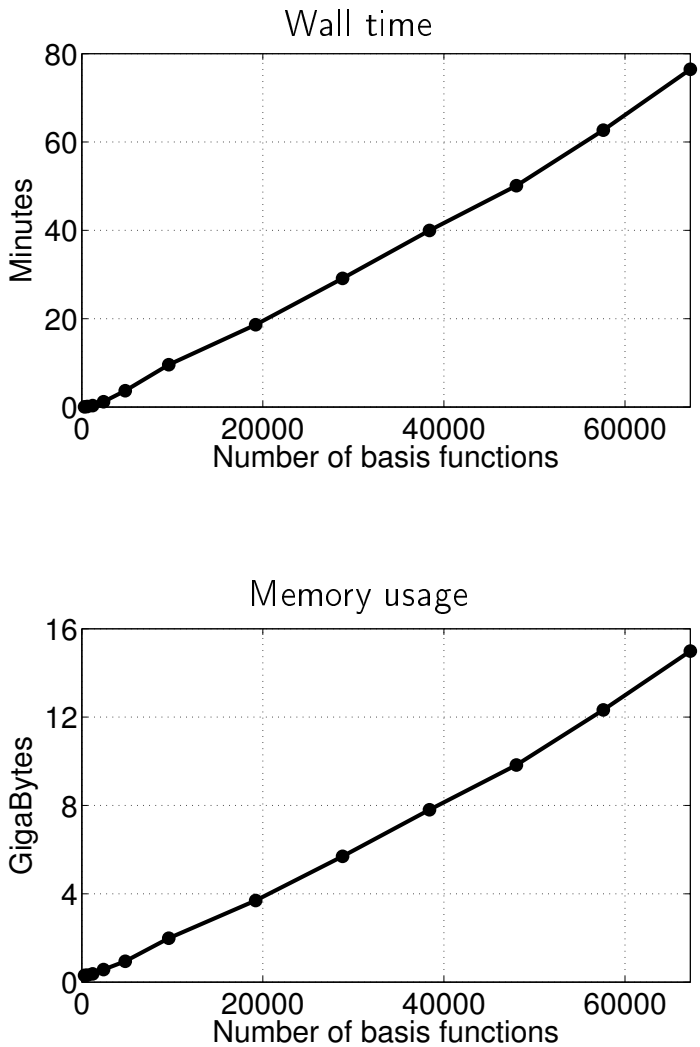


Figure 6.3: Timings and memory usage for density matrix purification in Hartree–Fock/3–21G calculations on glutamine–alanine helix systems of varying size. The data, which is taken from Figure 2 in Paper 7, includes congruence transformations.

Chapter 7

Final remarks

The introductory chapters were intended to create an interest for the included papers. After reading the previous chapters, you should have a good feeling for what the key ideas of the included papers are. If you would like to know how these ideas have been realized, learn more about the details, and see more results I encourage you to read the papers. However, before I end the introductory chapters I will give some further comments on each paper. In particular, I will write something about the way the work progressed. I will also shortly discuss possible future work that is immediately connected to the results of this thesis.

7.1 Further comments on included papers

Paper 1. Rotations of occupied invariant subspaces in self–consistent field calculations

In this paper, a mathematical framework for control of errors in self–consistent field calculations is proposed. The key idea is to measure errors coming from computational approximations by their impact on the occupied subspace rather than by their constituent parts. The general framework of this paper can be used in the handling of all kinds of computational approximations occurring in Hartree–Fock/Kohn–Sham calculations. Therefore, provided that developers of electronic structure codes appreciate and use the presented ideas, this paper could become the most important among the included papers.

Paper 2. Density matrix purification with rigorous error control

In this paper, the framework of Paper 1 is used to derive schemes to control the forward error of density matrix purification. Before the ideas of this paper were developed, we investigated the behavior of errors using truncation of small matrix elements based on the 1-norm.[1] We found it

quite difficult to explain the behavior of errors and even more difficult to rigorously control the forward error. At that time however, I was not yet aware of the possibility of relating unitary–invariant matrix norms to rotations of invariant subspaces. It turned out that truncation based on matrix norms, which was proposed in Ref. 1, could be used to control the error in the occupied subspace. The breakthrough came when I stumbled on some nice results on perturbations of invariant subspaces by Davis and Kahan[16] which are directly applicable to forward error control in density matrix purification.

PAPER 3. Computation of interior eigenvalues in electronic structure calculations facilitated by density matrix purification

The work presented in this paper was carried out as part of a PhD course in numerical linear algebra given by Axel Ruhe at the numerical analysis department at KTH. This work was motivated by the need to calculate the HOMO and LUMO eigenvalues in the density matrix purification algorithm of Paper 2. However, the developed schemes may also be used to compute a set of physically interesting eigenpairs in a window around the band gap.

PAPER 4. Recursive inverse factorization

Most of the work presented in this paper was carried out at the Los Alamos National Laboratory during the summer of 2007. Anders had an idea that an inverse factor of some block diagonal part of the matrix would be a sufficiently good starting guess so that iterative refinement could be used to find an inverse factor of the whole matrix. When I started to investigate this, I observed that *binary* principal submatrix decompositions seemed to give factorization errors smaller than one whereas other principal submatrix decompositions did not. This observation instigated the attempt to prove that binary decompositions always result in factorization errors smaller than one.

PAPER 5. Truncation of small matrix elements based on the Euclidean norm for blocked data structures

Already when working on my master project, I was not satisfied with the way sparsity was enforced in large–scale electronic structure calculations. Instead of using cutoff radius– or element magnitude–based truncation I started to control the error by the 1–norm of the entire matrix. Although using some matrix norm felt like the right thing to do, at that time I did not have particularly illuminating arguments for this. Later it turned out that if the error is controlled by some unitary–invariant norm such as the Frobenius or Euclidean norm, errors in the occupied subspace can be controlled (this is explained in Paper 1). In

Paper 5 we present methods to remove small matrix elements based on the Euclidean norm. Euclidean norm based truncation is one of the most important operations in our density matrix purification method presented in Paper 2.

PAPER 6. A hierarchic sparse matrix data structure for large–scale Hartree–Fock/Kohn–Sham calculations

My first sparse matrix library implementation used a blocked compressed sparse column data structure and was similar to existing implementations whose descriptions could be found in the literature. When this library was developed, I realized soon that it was quite inconvenient to access the submatrices and each matrix operation that needed to be implemented required a lot of thinking and the code was becoming increasingly difficult to maintain. Therefore I started to consider alternatives which resulted in the hierarchic data structure described in Paper 6 and used in the Ergo program. Using this data structure, I was able to implement many different matrix operations with little effort but high performance. The implemented operations include multiplications of symmetric, triangular, and on–the–fly transposed matrices as well as inverse Cholesky factorization. For several critical operations, the use of symmetry has reduced the computational time and memory usage with nearly 50%.

PAPER 7. Hartree–Fock calculations with linearly scaling memory usage

In this paper, we present a complete set of methods for Hartree–Fock calculations that require computational resources, CPU time and memory, that increase only linearly with system size. The hierarchic data structure of Paper 6 is extensively used for representation and manipulation of sparse matrices. The performance of the density matrix purification algorithm of Paper 2 is demonstrated along with benchmarks of Coulomb and exchange matrix evaluations. The schemes for calculation of interior eigenvalues and truncation of small matrix elements of Papers 3 and 5 were used as well.

7.2 Future outlook

Many of the methods that are presented in this thesis are used in the Ergo quantum chemistry program. Below follow a few ideas of what should or could be done in the future, both in the Ergo program and for method development in general.

- The framework of Paper 1 has not yet been fully implemented in the Ergo program. So far, density matrix purification is the only part that uses the framework. We are about to publish some schemes for error control in Coulomb and Hartree–Fock exchange matrix evaluations.[4] These schemes are based on runtime numerical error estimates. Whereas these methods represent a significant improvement compared to ad–hoc selection of threshold values, error control in integral evaluations based on analytical error bounds remains an open challenge. The schemes of Ref. 4 are likely to be useful also in evaluations of exchange correlation contributions in density functional theory, but this remains to be investigated. Finally, all these pieces should be put together so that threshold values for all approximations can be directly related to the desired accuracy in the occupied subspace at each stage of the self–consistent field convergence.

- The Ergo program has been parallelized for shared memory architectures. The program should also be parallelized for distributed memory architectures so that available computer resources can be fully utilized.

- As already discussed in Chapter 3, the idea to consider the occupied subspace could be useful for error analysis also of energy minimization methods that apply the exponential parameterization. A rigorous error analysis and appropriate schemes to control errors could make it possible to adequately compare these methods with density matrix purification.

Appendix A

Definitions

The Fock/Kohn–Sham matrix consists of one–electron H_1 and two–electron contributions $F_{2\text{el}}$;

$$F_S = H_1 + F_{2\text{el}}. \tag{A.1}$$

The one electron matrix depends only on the basis set and positions of the nuclei;

$$H_1 = -\int_{\mathbb{R}^3} \Phi(r) \frac{\nabla^2 \Phi^T(r)}{2} \mathrm{d}r - \int_{\mathbb{R}^3} \Phi(r) \sum_A \frac{Z_A}{|r_A - r|} \Phi^T(r) \mathrm{d}r. \tag{A.2}$$

Here Z_A and r_A is the charge and position of atom A. In the Hartree–Fock method, $F_{2\text{el}} = J + K$, in the Kohn–Sham method $F_{2\text{el}} = J + F_{\text{xc}}$, and in hybrid methods $F_{2\text{el}} = J + \alpha K + F_{\text{xc}}$ for some scalar α. These matrices can all be formulated in terms of the density matrix: the Coulomb matrix

$$J = 2 \iint_{\mathbb{R}^6} \frac{\Phi(r_1)\Phi^T(r_2) D_S \Phi(r_2)\Phi^T(r_1)}{|r_1 - r_2|} \mathrm{d}r_2 \mathrm{d}r_1, \tag{A.3}$$

the Hartree–Fock exchange matrix

$$K = \iint_{\mathbb{R}^6} \frac{\Phi(r_1)\Phi^T(r_2) D_S \Phi(r_1)\Phi^T(r_2)}{|r_1 - r_2|} \mathrm{d}r_2 \mathrm{d}r_1, \tag{A.4}$$

and the Kohn–Sham exchange correlation matrix

$$F_{\text{xc}} = \int_{\mathbb{R}^3} \Phi(r) \left. \frac{\partial \mathcal{F}}{\partial \rho} \right|_{\rho=\rho(r)} \Phi^T(r) \mathrm{d}r \tag{A.5}$$

where $\int \mathcal{F}(\rho) \mathrm{d}r$ is the energy functional which here for simplicity is assumed to be only density dependent. The overlap matrix

$$S = \int_{\mathbb{R}^3} \Phi(r) \Phi^T(r) \mathrm{d}r. \tag{A.6}$$

References

[1] Emanuel H. Rubensson and Paweł Sałek, *Systematic sparse matrix error control for linear scaling electronic structure calculations*, J. Comput. Chem. **26**, 1628–1637 (2005).

[2] Emanuel H. Rubensson and Hans Jørgen Aa. Jensen, *Determination of the chemical potential and HOMO/LUMO orbitals in density purification methods*, Chem. Phys. Lett. **432**, 591–594 (2006).

[3] Emanuel H. Rubensson, Elias Rudberg, and Paweł Sałek, *Sparse matrix algebra for quantum modeling of large systems*, Proceedings of PARA'06, Springer LNCS **4699**, 90–99 (2007).

[4] Elias Rudberg, Emanuel Rubensson, and Paweł Sałek, *Automatic selection of integral thresholds by extrapolation in Coulomb and exchange matrix constructions*, manuscript to be published.

[5] Elias Rudberg, *Quantum chemistry for large systems*, Ph.D. thesis, Department of Theoretical Chemistry, Royal Institute of Technology, Stockholm (2007).

[6] Elias Rudberg, Emanuel H. Rubensson, and Paweł Sałek, *Ergo (version 1.6); a quantum chemistry program for large scale self–consistent field calculations*, (2007).

[7] Trygve Helgaker, Poul Jørgensen, and Jeppe Olsen, *Molecular electronic-structure theory*, Wiley, Chichester (2000).

[8] Frank Jensen, *Introduction to computational chemistry*, Wiley, Chichester (1999).

[9] Konstantin N. Kudin and Gustavo E. Scuseria, *Converging self-consistent field equations in quantum chemistry - recent achievements and remaining challenges*, Math. Model. Num. Anal. **41**, 281–296 (2007).

[10] Lea Thøgersen, *Optimization of densities in Hartree-Fock and density-functional theory, atomic orbital based response theory, and benchmarking for radicals*, Ph.D. thesis, Department of Chemistry, University of Aarhus, Aarhus (2005).

[11] Michael C. Zerner and Michael Hehenberger, *A dynamical damping scheme for converging molecular scf calculations*, Chem. Phys. Lett. **62**, 550–554 (1979).

[12] Eric Cancès and Claude Le Bris, *Can we outperform the DIIS approach for electronic structure calculations*, Int. J. Quantum Chem. **79**, 82–90 (2000).

[13] V. R. Saunders and I. H. Hillier, *A "level-shifting" method for converging closed shell Hartree-Fock wave functions*, Int. J. Quantum Chem. **7**, 699–705 (1973).

[14] P. Pulay, *Convergence acceleration of iterative sequences. The case of SCF iteration*, Chem. Phys. Lett. **73**, 393 (1980).

[15] P. Pulay, *Improved SCF convergence acceleration*, J. Comput. Chem. **3**, 556 (1982).

[16] Chandler Davis and W. M. Kahan, *The rotation of eigenvectors by a perturbation. III*, SIAM J. Numer. Anal. **7**, 1–46 (1970).

[17] G. W. Stewart and Ji-guang Sun, *Matrix perturbation theory*, Academic Press, San Diego (1990).

[18] Otto F. Sankey, David A. Drabold, and Andrew Gibson, *Projected random vectors and the recursion method in the electronic-structure problem*, Phys. Rev. B **50**, 1376–1381 (1994).

[19] C. Bekas, E. Kokiopoulou, and Yousef Saad, *Computation of large invariant subspaces using polynomial filtered Lanczos iterations with applications in density functional theory*, SIAM J. Matrix Anal. Appl. **30**, 397–418 (2008).

[20] David A. Drabold and Otto F. Sankey, *Maximum entropy approach for linear scaling in the electronic structure problem*, Phys. Rev. Lett. **70**, 3631–3634 (1993).

[21] Lin-Wang Wang and Alex Zunger, *Solving Schrödinger's equation around a desired energy: Application to silicon quantum dots*, J. Chem. Phys. **100**, 2394–2397 (1994).

[22] Bin Gao, Jun Jiang, Kai Liu, Ziyu Wu, Wei Lu, and Yi Luo, *An efficient first-principle approach for electronic structures calculations of nanomaterials*, J. Comput. Chem. **29**, 434–444 (2007).

[23] R. McWeeny, *The density matrix in self-consistent field theory. I. Iterative construction of the density matrix*, Proc. R. Soc. London Ser. A **235**, 496–509 (1956).

[24] Michael T. Heath, *Scientific computing: An introductory survey*, McGraw-Hill, Singapore (1997).

[25] X.-P. Li, R. W. Nunes, and David Vanderbilt, *Density-matrix electronic-structure method with linear system-size scaling*, Phys. Rev. B **47**, 10891–10894 (1993).

[26] John M. Millam and Gustavo E. Scuseria, *Linear scaling conjugate gradient density matrix search as an alternative to diagonalization for first principles electronic structure calculations*, J. Chem. Phys. **106**, 5569–5577 (1997).

[27] Murray S. Daw, *Model for energetics of solids based on the density matrix*, Phys. Rev. B **47**, 10895–10898 (1993).

[28] Matt Challacombe, *A simplified density matrix minimization for linear scaling self-consistent field theory*, J. Chem. Phys. **110**, 2332–2342 (1999).

[29] Trygve Helgaker, Helena Larsen, Jeppe Olsen, and Poul Jørgensen, *Direct optimization of the AO density matrix in Hartree-Fock and Kohn-Sham theories*, Chem. Phys. Lett. **327**, 397–403 (2000).

[30] Anthony Dyan, Pierre Dubot, and Pierre Cenedese, *Solving the SCF problem in molecular orbital calculations through a sequence of quadratic programming: Extension to large systems*, Phys. Rev. B **72**, 125104 (2005).

[31] Helena Larsen, Jeppe Olsen, Poul Jørgensen, and Trygve Helgaker, *Direct optimization of the atomic-orbital density matrix using the conjugate-gradient method with a multilevel preconditioner*, J. Chem. Phys. **115**, 9685–9697 (2001).

[32] Yihan Shao, Chandra Saravanan, Martin Head-Gordon, and Christopher A. White, *Curvy steps for density matrix-based energy minimization: Application to large-scale self-consistent-field calculations*, J. Chem. Phys. **118**, 6144–6151 (2003).

[33] Cleve Moler and Charles Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Rev. **45**, 3 (2003).

[34] S. Goedecker and L. Colombo, *Efficient linear scaling algorithm for tight-binding molecular dynamics*, Phys. Rev. Lett. **73**, 122–125 (1994).

[35] S. Goedecker and M. Teter, *Tight-binding electronic-structure calculations and tight-binding molecular dynamics with localized orbitals*, Phys. Rev. B **51**, 9455–9464 (1995).

[36] Roi Baer and Martin Head-Gordon, *Chebyshev expansion methods for electronic structure calculations on large molecular systems*, J. Chem. Phys. **107**, 10003–10013 (1997).

[37] Kevin R. Bates, Andrew D. Daniels, and Gustavo E. Scuseria, *Comparison of conjugate gradient density matrix search and Chebyshev expansion methods for avoiding diagonalization in large-scale electronic structure calculations*, J. Chem. Phys. **109**, 3308–3312 (1998).

[38] WanZhen Liang, Chandra Saravanan, Yihan Shao, Roi Baer, Alexis T. Bell, and Martin Head-Gordon, *Improved Fermi operator expansion methods for fast electronic structure calculations*, J. Chem. Phys. **119**, 4117–4125 (2003).

[39] R. N. Silver, H. Roeder, A. F. Voter, and J. D. Kress, *Kernel polynomial approximations for densities of states and spectral functions*, J. Comput. Phys. **124**, 115–130 (1996).

[40] Adam H. R. Palser and David E. Manolopoulos, *Canonical purification of the density matrix in electronic-structure theory*, Phys. Rev. B **58**, 12704–12711 (1998).

[41] Anders M. N. Niklasson, *Expansion algorithm for the density matrix*, Phys. Rev. B **66**, 155115 (2002).

[42] Anders M. N. Niklasson, C. J. Tymczak, and Matt Challacombe, *Trace resetting density matrix purification in $\mathcal{O}(N)$ self-consistent-field theory*, J. Chem. Phys. **118**, 8611–8620 (2003).

[43] A. Holas, *Transforms for idempotency purification of density matrices in linear-scaling electronic-structure calculations*, Chem. Phys. Lett. **340**, 552–558 (2001).

[44] David A. Mazziotti, *Towards idempotent reduced density matrices via particle-hole duality: McWeeny's purification and beyond*, Phys. Rev. E **68**, 066701 (2003).

[45] H. J. Xiang, W. Z. Liang, Jinlong Yang, J. G. Hou, and Qingshi Zhu, *Spin-unrestricted linear-scaling electronic structure theory and its application to magnetic carbon-doped boron nitride nanotubes*, J. Chem. Phys. **123**, 124105 (2005).

[46] Ramiro Pino and Gustavo E. Scuseria, *Purification of the first-order density matrix using steepest descent and Newton-Raphson methods*, Chem. Phys. Lett. **360**, 117–122 (2002).

[47] M. S. Paterson and L. Stockmeyer, *On the number of nonscalar multiplcations necessary to evaluate matrix polynomials*, SIAM J. Comput. **2**, 60–66 (1973).

[48] Elias Rudberg, private communication.

[49] Zhaojun Bai, James Demmel, Jack Dongarra, Axel Ruhe, and Henk van der Vorst (eds.), *Templates for the solution of algebraic eigenvalue problems*, SIAM, Philadelphia (2000).

[50] Thomas Ericsson and Axel Ruhe, *The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems*, Math. Comput. **35**, 1251–1268 (1980).

[51] H. J. Xiang, Jinlong Yang, J. G. Hou, and Qingshi Zhu, *Linear scaling calculation of band edge states and doped semiconductors*, J. Chem. Phys. **126**, 244707 (2007).

[52] Ahmed H. Sameh and John A. Wisniewski, *A trace minimization algorithm for the generalized eigenvalue problem*, SIAM J. Numer. Anal. **19**, 1243–1259 (1982).

[53] R. W. Nunes and David Vanderbilt, *Generalization of the density-matrix method to a nonorthogonal basis*, Phys. Rev. B **50**, 17611–17614 (1994).

[54] Anders M. N. Niklasson, Valéry Weber, and Matt Challacombe, *Nonorthogonal density-matrix perturbation theory*, J. Chem. Phys. **123**, 044107 (2005).

[55] Branislav Jansík, Stinne Høst, Paul Jørgensen, and Jeppe Olsen, *Linear-scaling symmetric square-root decomposition of the ovelap matrix*, J. Chem. Phys. **126**, 124104 (2007).

[56] M. Benzi, C. D. Meyer, and M. Tuma, *A sparse approximate inverse preconditioner for the conjugate gradient method*, SIAM J. Sci. Comput. **17**, 1135–1149 (1996).

[57] M. Benzi, J. K. Cullum, and M. Tuma, *Robust approximate inverse preconditioning for the conjugate gradient method*, SIAM J. Sci. Comput. **22**, 1318–1332 (2000).

[58] M. Benzi, R. Kouhia, and M. Tuma, *Stabilized and block approximate inverse preconditioners for problems in solid and structural mechanics*, Comput. Meth. Appl. Mech. Eng. **190**, 6533–6554 (2001).

[59] A. M. N. Niklasson, *Iterative refinement method for the approximate factorization of a matrix inverse*, Phys. Rev. B **70**, 193102 (2004).

[60] Emilie V. Haynsworth, *Determination of the inertia of a partitioned Hermitian matrix*, Lin. Alg. Appl. **1**, 73–81 (1968).

[61] Chandra Saravanan, Yihan Shao, Roi Baer, Philip N. Ross, and Matrin Head-Gordon, *Sparse matrix multiplications for linear-scaling electronic structure calculations in an atom-centered basis set using multiatom blocks*, J. Comput. Chem. **24**, 618–622 (2003).

[62] P. E. Maslen, C. Ochsenfeld, C. A. White, M. S. Lee, and M. Head-Gordon, *Locality and sparsity of ab initio one-particle density matrices and localized orbitals*, J. Phys. Chem. A **102**, 2215–2222 (1998).

[63] *GotoBLAS*, `http://www.tacc.utexas.edu/resources/software/#blas`.

[64] *Automatically Tuned Linear Algebra Software (ATLAS)*, `http://math-atlas.sourceforge.net/`.

[65] *Intel Math Kernel Library (Intel MKL)*, `http://www.intel.com/cd/software/products/asmo-na/eng/307757.htm`.

[66] *AMD Core Math Library (ACML)*, `http://developer.amd.com/cpu/libraries/acml/Pages/default.aspx`.

[67] Fred G. Gustavson, *Two fast algorithms for sparse matrices: Multiplication and permuted transposition*, ACM Trans. Math. Softw. **4**, 250–269 (1978).

[68] S. Pissanetsky, *Sparse matrix technology*, Academic Press, New York (1984).

[69] J. R. Gilbert, C. Moler, and R. Schreiber, *Sparse matrices in MATLAB: Design and implementation*, SIAM J. Matrix Anal. Appl. **13**, 333–356 (1992).

[70] Matt Challacombe, *A general parallel sparse-blocked matrix multiply for linear scaling SCF theory*, Comp. Phys. Commun. **128**, 93 (2000).

[71] Jörg Kussmann and Christian Ochsenfeld, *Linear-scaling method for calculating nuclear magnetic resonance chemical shifts using gauge-including atomic orbitals within Hartree-Fock and density-functional theory*, J. Chem. Phys. **127**, 054103 (2007).

[72] E. Elmroth, F. Gustavson, I. Jonsson, and B. Kågström, *Recursive blocked algorithms and hybrid data structures for dense matrix library software*, SIAM Rev. **46**, 3–45 (2004).