INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
RICE UNIVERSITY

Conjugate Gradient Density Matrix Search: A Linear Scaling Alternative to Diagonalization

by

John M. Millam

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

Gustavo Scuseria, Chairman
Professor of Chemistry

John Hutchinson
Associate Professor of Chemistry

Peter Nordlander
Associate Professor of Physics

Houston Texas
May, 1997
Abstract

Conjugate Gradient Density Matrix Search: A Linear Scaling Alternative to Diagonalization

by

John M. Millam

Advances in the computation of the Coulomb, exchange, and correlation contributions to Gaussian-based Hartree-Fock and density functional theory Hamiltonians have demonstrated near-linear scaling with molecular size for these steps. These advances leave the $O(N^3)$ diagonalization bottleneck as the rate determining step for very large systems. In this work, a conjugate gradient density matrix search (CG-DMS) method has been successfully extended and computationally implemented for use with first principles calculations. A Cholesky decomposition of the overlap matrix and its inverse, which can be formed in near linear time for sparse systems, is used to transform to and back from an orthonormal basis. Linear scaling of CPU time for the density matrix search and crossover of CPU time with diagonalization is demonstrated for polyglycine chains containing up to 493 atoms and water clusters up to 900 atoms.
Acknowledgments

Dr. Gustavo Scuseria is acknowledged for the contributions of his experience, expertise, and patience to the development of this work. Andrew Livelsberger and Dr. Chunhui Xu are acknowledged for their contributions to the development and refinement of the CG-DMS method. John Burant and Dr. Eric Stratmann are acknowledged for their contributions in modifying the Fock matrix building routines.

Michael and Lois Millam are acknowledged for their constant support and encouragement, which has helped me in so many ways.
# Table of Contents

Chapter 1: Introduction .................................................. 1
Chapter 2: Alternatives to Diagonalization ......................... 4
Chapter 3: Transformation to an Orthonormal Basis ............... 6
Chapter 4: Conjugate Gradient Density Matrix Search .......... 9
Chapter 5: Matrix Sparsity ............................................ 15
Chapter 6: Results and Discussion .................................. 24
Chapter 7: Conclusions ................................................ 36
References .............................................................. 37
List of Figures

Figure 1. Matrix sparsity for polyglycine chains. 21
Figure 2. Matrix sparsity for water clusters. 22
Figure 3. Matrix sparsity for water clusters as a function of neglect threshold. 23
Figure 4. CPU timings for polyglycine chains. 32
Figure 5. Memory scaling for polyglycine chains. 33
Figure 6. CPU timings for water clusters. 34
Figure 7. Parallel performance. 35
List of Tables

Table 1. Matrix sparsity for water clusters as a function of neglect threshold. 18

Table 2. Effect of neglect threshold on matrix sparsity, CPU time, and accuracy. 26

Table 3. Effect of the SCF convergence criterion on convergence and accuracy. 26

Table 4. Comparison of different basis sets using CG-DMS. 28

Table 5. Comparison of CPU times between diagonalization and CG-DMS for polyglycine chains. 29

Table 6. Comparison of CPU times between diagonalization and CG-DMS for water clusters. 31
Chapter 1

Introduction

The goal of applying \textit{ab initio} quality methods to large systems of interest, such as proteins, bio-molecules, and nanostructures, is inherently limited by how the computational cost and requirements scale with the size of the system. Until recently, the rate limiting step for Gaussian-based Hartree-Fock (HF) and density functional theory (DFT) calculations was the computation of two-electron integrals. Using direct methods with integral screening, the cost of forming the Coulomb matrix can be limited to $\mathcal{O}(N^2)$ for large systems [1, 2, 3], where $N$ is the number of basis functions. Recent developments, however, have been able to reduce this scaling to $\mathcal{O}(N)$ for large systems: using Fast Multipole Methods (FMM) [4, 5] and tree codes [6], the Coulomb part of the effective Hamiltonian matrix has been reduced to near-linear scaling.

In addition to solving for the Coulomb matrix, DFT calculations require an exchange-correlation integration. This integration consists of four evaluations: quadrature weights, basis functions, electron density, and the exchange-correlation functional. Using spatial cutoffs, the quadrature weights [7] can be made to scale as $\mathcal{O}(N_a)$, where $N_a$ is the number of atoms [8]. Evaluation of the basis functions, electron density, and the exchange-correlation functional have also been shown to
scale linearly by introducing microbatches of grid points for which only a local subset of elements need to be computed [8].

For HF, computation of the exchange matrix also poses a challenge to achieving linear scaling. Using standard density matrix screening leaves $\mathcal{O}(N^2)$ two-electron integrals in the large molecule limit [9]. The "order-N exchange" (ONX) method [9] uses an integral screening approach that explicitly assumes an exponential decay in the electron density at large distances. This approach is justified for electrically insulating systems where the exchange interaction is well localized [10]. Alternatively, if one uses the FMM method for the Coulomb matrix, one can obtain almost all of the exchange energy using the "near field exchange" (NFX) method [11] with only a slight modification of the FMM procedure. In NFX, we use the near field two-electron integrals to form the near field exchange matrix using the standard exchange definition. Since the exchange interaction for insulators is localized, neglecting the exchange interaction in the FMM far field introduces a very small error [11].

These advances leave the diagonalization step as the time dominant step in the large molecule limit for first principles HF and DFT calculations. The diagonalization step consists of four parts: the similarity transformation of the Hamiltonian from the atomic orbital (AO) basis to an orthonormal basis, the diagonalization procedure, the back transformation of the eigenvectors to the AO basis, and the formation of the density from the eigenvectors. Because the diagonalization step has a very small prefactor, this $\mathcal{O}(N^3)$ procedure does not dominate the overall time scaling until very large systems [1, 2, 3]. However, a far more stringent limitation for
large systems, is diagonalization's $O(N^2)$ memory requirement. Therefore, replacing diagonalization with a method that avoids this CPU and memory bottleneck is a prerequisite to achieving linear scaling for any \textit{ab initio} method.
Chapter 2

Alternatives to Diagonalization

An obvious candidate for an alternative to standard diagonalization is Lanczos and related methods, which have shown great success in computing eigenvectors of large matrices [12]. The Lanczos method can be extended to compute multiple eigenvectors, but its CPU scaling is $\mathcal{O}(Nk^2)$ and memory requirement is $\mathcal{O}(Nk) + \mathcal{O}(k^2)$, where $k$ is the number of required eigenvectors [13, 14]. In quantum chemical calculations, the density matrix construction requires all eigenvectors corresponding to occupied orbitals. Since the number of occupied orbitals, $k$, scales with the size of the system, linear scaling cannot be achieved with this method.

In recent years, there has been great interest in the condensed matter physics community in development of $\mathcal{O}(N)$ methods which avoid diagonalization. It should be pointed out that in plane-wave methods where large numbers of basis functions are required, diagonalization greatly affects the computational scaling [15]. Alternatives to diagonalization have also been proposed for use with localized basis functions. Of particular interest to us are methods that are based on direct energy minimization. Direct minimization involves either searching for a set of localized molecular orbitals (LMO) [16, 17, 18, 19, 20, 21, 22, 23] or directly for the density matrix [24, 25, 26, 27, 28, 29, 30]. Other $\mathcal{O}(N)$ methods have also been proposed [31, 32, 33, 34, 35]. In this work, we report our implementation of a direct energy
minimization method for non-orthogonal bases based on a density matrix search (DMS). A similar method for orthogonal cases has already been implemented in our research group for tight-binding [29] and semiempirical [30] cases.

The DMS minimizes an energy functional of the density matrix subject to $N_e$-representability. $N_e$-representability for the density matrix, $P$, can be expressed by two constraints: the correct number of electrons, $\text{Tr}(PS) = N_e$, and idempotency, $PSP = P$. where $S$ is the overlap matrix and $N_e$ is the number of electrons. At convergence, the density matrix and the Fock matrix, $F$, must commute, i.e., $SPF = FPS$. We require the DMS energy functional to have a minimum which satisfies these three criteria. Li, Nunes, and Vanderbilt (LNV) [24] proposed a functional that incorporated the McWeeny purification transformation [36] to maintain idempotency during minimization. The McWeeny purification transformation (for an orthonormal basis) is given by

$$\hat{P} = 3P^2 - 2P^3,$$

and has the property that if $P$ is near idempotent, then the purified density matrix, $\hat{P}$, is closer to exact idempotency than the original density matrix [24, 36]. If the density matrix is already idempotent, the purification transformation returns the original density matrix.
Chapter 3

Transformation to an Orthonormal Basis

The original LNV method [24] was developed for tight binding (TB) models which usually assume an orthogonal basis [29, 37]. *Ab initio* methods using localized Gaussian basis sets require a modification to handle the non-orthogonal basis. Two main approaches may be used for this purpose. The first is to solve for the density matrix directly in the AO basis, which involves modifying the energy functional used by DMS to include the overlap matrix. This method has been described in the literature [38] and is not investigated here. The approach used in this work is to transform both the density and Fock matrices into an orthonormal basis, solving for the new density matrix using an orthonormal DMS, and then back transforming to the AO basis.

A large number of choices for a transformation to an orthonormal space are possible. The Löwdin transformation [39] is a natural choice because the transformed basis is closest to the original basis in a least-squares sense. The transformed density (\( P_{\text{ortho}} \)) and Fock (\( F_{\text{ortho}} \)) matrices are expressed as

\[
P_{\text{ortho}} = S^{1/2} P_{AO} S^{1/2}; \quad F_{\text{ortho}} = S^{-1/2} F_{AO} S^{-1/2}, \quad (3.1)
\]
where $S$ is the overlap matrix. However, the fundamental drawback of this choice is that it requires a diagonalization to form the needed transformation matrices, $S^{1/2}$ and $S^{-1/2}$.

Alternatively, we propose in this study a transformation to an orthonormal basis obtained from the Cholesky decomposition of the overlap matrix. The Cholesky decomposition partitions a positive definite matrix into a lower triangular matrix times an upper triangular matrix [40]. For symmetric positive definite matrices such as the overlap matrix, the lower triangular matrix is just the transpose of the upper triangular matrix. The decomposition of the overlap matrix can be expressed as

$$S = U^T U,$$  \hspace{1cm} (3.2)

where $U$ is an upper triangular matrix. The Cholesky transformation to an orthonormal basis can be expressed as

$$P_{\text{ortho}} = U P_{\text{AO}} U^T; \quad F_{\text{ortho}} = U^{-T} F_{\text{AO}} U^{-1}.$$  \hspace{1cm} (3.3)

The corresponding transformation of the orthonormal density to the AO basis is given by

$$P_{\text{AO}} = U^{-1} P_{\text{ortho}} U^{-T}.$$  \hspace{1cm} (3.4)

In test calculations, DMS using the Löwdin and Cholesky transformations gave comparable rates of convergence and similar matrix sparsities.

The computational scaling of the Cholesky decomposition [41] is given by

$$\text{cost} \propto \sum_{i=1}^{N} r_i^U (r_i^U + 3)/2,$$  \hspace{1cm} (3.5)
where $r_i^U$ is the number of elements in the $i^{th}$ row of $U$ and $N$ is the number of basis functions. If the $r_i^U$'s remain approximately constant with system size, then the computational cost goes up linearly because of the dependence on the sum. As the decomposition proceeds, fill-in may occur. If excess fill-in occurs, the sparsity of $U$ will be lost and correspondingly linear scaling will be lost because the $r_i^U$'s will not be approximately constant. For symmetric positive definite matrices, the pivot order is arbitrary and an optimal choice for the ordering will minimize fill-in. The optimal pivot ordering is an NP-complete problem but many good heuristic algorithms exist for choosing near optimal sparsity preserving pivot orderings, such as Cuthill-McKee, Reverse Cuthill-McKee, Nested Dissection, and Approximate Minimum Degree [41, 42].
Chapter 4

Conjugate Gradient Density Matrix Search

The DMS energy functional can be minimized using standard off-the-shelf optimization algorithms. As LNV [24] suggested, we have chosen a conjugate gradient DMS (CG-DMS) approach, since conjugate gradient only requires the value of the function and its gradient. Hessian based methods, such as Newton-Raphson optimization, can not be straightforwardly applied to the DMS since they require the computation, storage, and inversion of the exact Hessian matrix, which is formally $O(N^4)$ with $O(N^2)$ significant elements for large systems.

Conjugate Gradient (CG) minimization generates a series of “non-interfering” search directions along which the energy functional is minimized. For an energy functional $\Omega(P)$, of the density matrix $P$, the CG minimization search can be written algorithmically as follows [40]

\[
G_0 = H_0 = -\nabla \Omega(P_0)
\]

do $i = 1$, maximum iterations
\[
P_{i+1} = P_{i} + \lambda_{i}H_{i}
\]
\[
G_{i+1} = -\nabla \Omega(P_{i+1})
\]
\[
H_{i+1} = G_{i+1} + \gamma_{i} H_{i}
\]
enddo
where

$$\gamma_i = \frac{(G_{i+1} - G_i) \cdot G_{i+1}}{G_i \cdot G_i}.$$  \hfill (4.2)

$P_0$ is an initial guess for the density matrix, $G$ is the negative of the gradient, $H$ is the CG search direction, and $\lambda_i$ is the step length to the nearest minima. Minimization occurs as a series of one-dimensional line searches, $P_{i+1} = P_i + \lambda_i H_i$, along the CG search direction, $H_i$. The line search step length, $\lambda_i$, is chosen to correspond to a minimum in the energy functional along the search direction.

The energy functional used in this work for an orthonormal basis is a slight modification of the LNV [24] functional:

$$\Omega(P) = \text{Tr}(\hat{P}F) + \mu(\text{Tr}(P) - N_e)$$  \hfill (4.3)

$$\nabla \Omega(P) = 3PF + 3FP - 2P^2F - 2FP F - 2F^2P + \mu I$$  \hfill (4.4)

The first term in the functional is closely related to the true electronic energy but uses the McWeeny purification transformation, $\hat{P}$ [Eq. (2.1)], in place of the density matrix. The energy functional has a cubic dependence on the density matrix, $P$, due to the use of $\hat{P}$. For an idempotent or nearly idempotent density matrix, any perturbation that does not approximately maintain idempotency will be penalized by a cubic increase in the functional value. Thus, approximate idempotency can be maintained throughout the CG-DMS. Idempotency is not preserved exactly after each line search [see Eq. (4.1)], but can be restored by the repeated use of the purification transformation.

The second term in the functional, Eq. (4.3), is a Lagrangian constraint that enforces the correct number of electrons. The original LNV [24] functional form
used \( \hat{P} \) in place of \( P \) and chose \( \mu \) as the chemical potential. Doing so requires some update scheme for the chemical potential. The functional presented here is more general in that \( \mu \) can be computed explicitly at every step and the correct number of electrons is enforced exactly at every step. In this work, \( \mu \) is chosen such that the gradient of the functional, Eq. (4.4), is traceless:

\[
\mu = -\text{Tr}(3PF + 3FP - 2P^2F - 2FPF - 2FP^2)/N \tag{4.5}
\]

If the gradient and hence the search direction, \( H_i \), is traceless, then any finite step preserves the number of electrons.

\[
P_{i+1} = P_i + \lambda_i H_i \tag{4.6}
\]

\[
\text{Tr}(P_{i+1}) = \text{Tr}(P_i + \lambda_i H_i) = \text{Tr}(P_i) + \lambda_i \text{Tr}(H_i) = \text{Tr}(P_i) \tag{4.7}
\]

The initial density matrix, if necessary, is scaled by a constant, such that the trace of the density matrix yields the correct number of electrons. Doing so means that the Lagrangian constraint term will never enter the functional value and will only affect the gradient. This constraint term is trivially extendable to other functional forms and to the non-orthonormal case.

Examining the gradient expression, Eq. (4.3), we see that the minimum of the functional (when the gradient is zero) properly satisfies the DMS requirements. For any density matrix that has the correct trace and is idempotent, \( \mu \) goes to zero and the gradient simplifies to

\[
\nabla \Omega(P) = PF + FP - 2FPF. \tag{4.8}
\]
Hence, the gradient goes to zero if and only if the density matrix satisfies \( N_e \)-representability and commutation of \( P \) with \( F \).

The CG-DMS search can be improved by using preconditioning [40]. In preconditioned CG, the \( G \) search matrix [see Eq. (4.1)] is redefined to be

\[
G_{i+1} = -\mathcal{H}^{-1} \nabla \Omega(P_{i+1}),
\]

where \( \mathcal{H}^{-1} \) is an approximate inverse Hessian. The coefficient, \( \gamma_i \), in Eq. (4.2) must also be modified. The formula for the exact Hessian is known but forming the exact inverse Hessian is very expensive. For the purpose of this work, only diagonal preconditioning is used. For diagonal preconditioning, the Hessian is approximated by its diagonal elements only, which is trivial to store, compute, and invert

\[
G_{i+1} = -D^{-1} \nabla \Omega(P_{i+1}),
\]

\[
D_{\mu\nu\rho\sigma} = (3 - 2P_{\mu\rho})F_{\nu\sigma} + (3 - 2P_{\nu\sigma})F_{\mu\rho} - 4(PF)_{\mu\rho} - 4(PF)_{\nu\sigma}.
\]

When preconditioning is used, the definition of the Lagrangian, \( \mu \), must be readjusted such that the preconditioned gradient, \( G \), is traceless.

CG-DMS involves performing one-dimensional line searches to minimize the functional along each given search direction. Since the functional has a cubic dependence on the density matrix, the functional value along the line search can be exactly expressed as a cubic polynomial, \( \Omega(\lambda) \) [26], hence the line search can be carried out exactly. The line search density matrix is expressed as

\[
P_{i+1} = P_i + \lambda_i H_i,
\]
where $P_i$ is the initial density matrix, $\lambda_i$ is the step length, and $H_i$ is the search direction. Substituting the line search density matrix into the energy functional, we can expand the functional and reduce it to a third order polynomial in $\lambda_i$

$$\Omega(\lambda_i) = a + b\lambda_i + c\lambda_i^2 + d\lambda_i^3; \quad (4.13)$$

where

$$
\begin{align*}
a &= \Omega(P_i), \\
b &= \text{Tr}(H_i \nabla \Omega(P_i)), \\
c &= 3\text{Tr}(H_i^2 F) - 2\text{Tr}(P_i H_i F) - 2\text{Tr}(H_i P_i H_i F) - 2\text{Tr}(H_i^2 P_i F), \\
d &= -2\text{Tr}(H_i^3 F).
\end{align*}
$$

Taking the derivative with respect to $\lambda_i$ and setting it equal to zero gives us a quadratic equation, which can be straightforwardly evaluated

$$\Omega'(\lambda_i) = b + 2c\lambda_i + 3d\lambda_i^2 = 0. \quad (4.15)$$

This quadratic equation yields two roots, and the root corresponding to a minimum is chosen. In our tests, the case where there is no minima was not encountered, i.e., the case when there are complex roots.

Since the line search does not explicitly preserve idempotency, care must be taken when updating the density matrix. If the step size is too large, use of the purification transformation may converge the density matrix to idempotency slowly or not at all. To prevent this from occurring, a maximum step size is enforced. When the computed step length is greater than the maximum allowed step size, the step
length is reduced to the maximum allowed value and the CG search direction is reset to be the negative gradient. The search direction is reset because the reduced step no longer satisfies the requirements for CG but this does not significantly affect the convergence. A maximum step size of .75 was effective in all the cases studied here.
Chapter 5

Matrix Sparsity

CG-DMS alone does not yield linear scaling. The evaluation of the function, its gradient, and the one-dimensional line search all require matrix multiplication and other matrix operations which scale as $N^2$ and $N^3$. In large molecular systems, we can reduce the cost of the matrix operations by taking advantage of matrix sparsity, i.e., the fact that many elements of a matrix are zero or below a certain threshold and can be safely neglected. During standard diagonalization, the sparsity of the matrix is not preserved, so any advantage from matrix sparsity is generally lost. In CG-DMS, we reduce costs by only storing and computing significant matrix elements, which requires a special set of sparse matrix routines [41]. For sparse systems where the number of significant elements scales linearly with the size of the system, the cost of all sparse matrix operations will also scale linearly with the size of the system.

Two series of benchmarks are presented which show the feasibility of CG-DMS for large systems. A series of polyglycine chains that are constrained to be linear [43] were used as representative one-dimensional systems. Polyglycine chains ($C_{2n}$ $N_n$ $O_{n+1}$ $H_{3n+2}$, $n = 10 - 70$) containing 10 to 70 glycine units (73 to 493 atoms, respectively) were used in this work. As a representative three-dimensional system, we used water clusters from 30 to 300 water molecules [44]. We define the measure
of sparsity here to be the percentage of non-negligible elements in a given matrix. In most cases, the numerical threshold for determining which elements are negligible was chosen to be $10^{-6}$ a.u., and this choice will be discussed in more detail below. In the AO basis, there are three important matrices: the overlap, Fock, and density matrices. We see in Figures 1 and 2 that the overlap and Fock matrices for both polyglycine chains and water clusters using LDA/3-21G decay rapidly with increasing system size, and that the Fock matrix is only slightly less sparse than the overlap matrix. We note that the rapid decay of the Fock and overlap matrices is similar for both systems in spite of their difference in composition and dimensionality. The density matrix for both systems also shows a rapid decay although it is significantly less sparse than either the Fock or overlap matrices. For the first time, we show quantitatively that the number of surviving elements of the density matrix elements decays more slowly with system size than Fock matrix elements. At this point, it is difficult to a priori predict the sparsity of either the Fock or density matrices: the reader should be reminded that both matrices are diagonal (and hence sparse) in the canonical HF basis, which is generally delocalized.

Since the CG-DMS is carried out in the orthonormal basis formed by the Cholesky transformation, it is essential that we know the sparsity of the matrices in that basis. It turns out that in the transformed basis the Fock and density matrices share similar sparse matrix forms. Because of the similarity in matrix forms, it is computationally efficient to force both matrices to have the same form which is chosen here as the union of the forms of each matrix. Sparsity of the Fock ($F$) and density matrices ($P$) for the combined matrix form in the orthogonal
basis is also included in Figures 1 and 2, and in both cases it is comparable to the sparsity of the density matrix in the AO basis. We find that the transformation to the Cholesky orthonormal basis does not introduce additional elements in the density matrix. Transformation of the Fock matrix to the orthonormal basis does make it less sparse, but for all the cases studied in this work its sparsity is no worse than that of the density matrix in the AO basis.

Since the matrix sparsity is dependent upon the neglect threshold, it is important that we understand their relationship with each other. Calculations on water clusters were done with LDA/STO-3G for four different neglect thresholds: $10^{-4}$, $5 \times 10^{-5}$, $10^{-5}$, and $10^{-6}$ a.u. The percentage of surviving elements of the orthonormal Fock and density matrices is shown in Figure 3 and summarized in Table 1. All four curves differ dramatically in magnitude but show similar rapid decay with system size. Reducing the neglect threshold from $10^{-6}$ to $10^{-4}$ a.u., we see a dramatic change in the sparsity of $F$ and $P$: for 30 water molecules the number of surviving matrix elements is reduced from 93 to 39%, and for 300 water molecules from 25 to 5%. Clearly, the neglect threshold will have a considerable effect on the performance and memory requirements of CG-DMS. The effect of the neglect threshold on accuracy and CPU time will be discussed in detail below.

In this work, we have found that the most difficult task is how to take advantage of matrix sparsity properly, that is, how to decide which matrix elements to keep and which to eliminate. This must be done in an efficient way that maintains the desired level of accuracy in the final energy and preserves both CG and SCF convergence. Using established screening techniques, the sparse matrix form of the
Table 1  Percentage of surviving elements (orthonormal Fock and density matrices) for CG-DMS as a function of the neglect threshold for water clusters using LDA/STO-3G.

<table>
<thead>
<tr>
<th>Molecule</th>
<th>CG-DMS (neglect threshold in a.u.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$1 \times 10^{-4}$</td>
</tr>
<tr>
<td>(H$<em>2$O)$</em>{30}$</td>
<td>39.1</td>
</tr>
<tr>
<td>(H$<em>2$O)$</em>{60}$</td>
<td>24.8</td>
</tr>
<tr>
<td>(H$<em>2$O)$</em>{90}$</td>
<td>15.8</td>
</tr>
<tr>
<td>(H$<em>2$O)$</em>{120}$</td>
<td>13.9</td>
</tr>
<tr>
<td>(H$<em>2$O)$</em>{150}$</td>
<td>9.9</td>
</tr>
<tr>
<td>(H$<em>2$O)$</em>{300}$</td>
<td>4.9</td>
</tr>
</tbody>
</table>

AO Fock matrix can be directly determined each SCF cycle. There is, however, no analogous way to determine the sparse matrix form of the AO density matrix. Along these lines, two different approaches have been examined. The first approach is called “let it grow” (LIG), because each matrix is allowed to have its own form, which can grow or shrink every CG cycle. The form of the product of two matrices can be determined in linear time using well known sparse matrix routines [41]. The second approach is called “fixed form” (FF), where the forms of the matrices are determined before each set of CG cycles and are maintained throughout the CG cycles. In addition, the gradient, search direction, and other intermediate matrix products are assumed to have the same sparse matrix form as the density matrix.

LIG is the more intuitively appealing of the two methods. LIG in practice, however, suffers from several drawbacks. Since the size of each matrix is not known ahead of time, memory must either be allocated dynamically before each sparse
matrix operation or preassigned before CG with an additional memory padding to allow the matrices to grow. Each matrix product generates twice as many elements as are actually significant, so periodic screening of matrix product elements is required or the number of elements will grow and linear scaling will be lost. Using a thresholding criterion, the additional non-significant elements of the matrix product can be eliminated. The cost of determining the matrix form for every matrix operation and of computating extra non-significant elements for every matrix product makes LIG unnecessarily slow.

Using FF avoids many of the problems faced by LIG. Memory for CG-DMS is preassigned without any additional memory padding. More importantly, only one sparse matrix form needs to be stored instead of the 5-6 forms required by LIG. This allows FF to use roughly half the memory of LIG. FF, unlike LIG, avoids computing non-significant elements and avoids the need to determine the matrix form every step. In addition, it is possible to take advantage of the shared form to further reduce the computational cost. These differences lead to FF being 2-4 times faster than LIG per CG-DMS cycle. Taking advantage of FF, however, requires an intelligent choice of the fixed form. If the matrix form is too large, the additional memory and CPU requirements will eliminate the advantage of switching to FF. If the matrix form is too restrictive, accuracy and convergence may be severely compromised. Even more critical is the stability of purification transformations. If the matrix form is too restrictive, purification using FF may not be able to restore idempotency which would lead to poor convergence in the CG-DMS.
In this work, FF is used exclusively in the CG-DMS. The fixed form was chosen to be the union of the forms of the orthonormal density and Fock matrices. This choice maintains both accuracy and convergence without sacrificing sparsity. Fixed form using this choice performs well even for fairly large neglect thresholds and for density matrices that are far from convergence. If the initial guess density is very poor, the fixed form will cause it to grow every SCF cycle even though within CG-DMS the form is held constant. It is important to note at this point that in the orthonormal where the CG-DMS is carried out, these four matrices all share very similar sparsity patterns so the combined form is only slightly less sparse than that of the orthonormal density matrix. This similarity does not hold in the AO basis, so fixed form CG-DMS in that basis would require an unpractically large form.
Figure 1  Percentage of surviving matrix elements (larger than $10^{-6}$ a.u.) for polyglycine chains using LDA/3-21G.
Figure 2  Percentage of surviving matrix elements (larger than $10^{-6}$ a.u.) for water clusters using LDA/3-21G.
Figure 3  Percentage of surviving elements (orthonormal Fock and density matrices) as a function of four different neglect thresholds (a.u.) for LDA/STO-3G water clusters.
Chapter 6

Results and Discussion

Our computational code is interfaced to a development version of the GAUSSIAN suite of programs [45]. Both HF and DFT benchmark calculations were carried out. DFT calculations were done using the local density approximation (LDA). All timings are given for an IBM RS6000/370 workstation. The GAUSSIAN codes compute the overlap matrix, the Fock matrices, and the initial density matrix. The initial density matrix is taken from a projected INDO guess, which is the default for GAUSSIAN. In order to achieve truly $O(N)$ scaling, the diagonalization of the semiempirical Hamiltonian that determines the starting guess density matrix needs to be eliminated, and this work will be reported elsewhere [30]. All timings reported in this paper were obtained without taking advantage of the molecular point group symmetry.

We introduce a neglect threshold which is used to determine which matrix elements are to be considered negligible and hence are to be eliminated. We chose the same neglect threshold for all the matrices used by CG-DMS. Table 2 shows the effect of the neglect threshold on sparsity, on computational cost, and on the converged total energy of 20-glycine for HF and LDA/3-21G. We see that if we keep all the matrix elements (neglect threshold equals zero), CG-DMS exactly reproduces the results of diagonalization. Thresholds of $10^{-8}$ a.u. and smaller yield
nanohartree accuracy in the total energy. A neglect threshold of $10^{-6}$ a.u. typically yields microhartree accuracy.

CPU time, memory, and disk space are determined by the sparsity of the matrices, which in turn is a function of the neglect threshold and the molecule being studied. For 20-glycine, we see that the percentage of surviving elements is strongly dependent upon the neglect threshold. For even a small neglect threshold of $10^{-10}$ a.u., only 65% of the elements of the Fock and density matrices need to be computed and retained. As the neglect threshold is raised to $10^{-6}$ a.u. the number of significant elements is cut in half. Since the cost of sparse matrix routines is dependent upon the sparsity of the matrices, we also see a corresponding reduction in the computational cost. Reduction in the percentage of elements by a factor of 2 results in an approximate factor of 4 speed up. A net speed up of almost a factor of 7 is observed between the zero and the $10^{-6}$ a.u. neglect thresholds.

In GAUSSIAN [45], the standard "SCF=tight" option uses a SCF convergence threshold of $10^{-8}$ a.u. for the RMS density. For CG-DMS, if the goal is achieving microhartree error in the converged total energy, there is no benefit in converging the SCF procedure tighter than the energy error introduced by the neglect threshold. Table 3 shows the effect on the total converged energy for HF and DFT for different SCF convergence thresholds. For a neglect threshold of $10^{-6}$ a.u., we see that an SCF convergence threshold of $10^{-6}$ a.u. (and smaller) yields a negligible increase in the accuracy of the total energy as compared to using $10^{-8}$ a.u. In our tests, CG-DMS took no more SCF cycles than diagonalization even when using identical SCF convergence criteria and often took fewer cycles.
Table 2  Percentage of surviving elements (orthonormal Fock and density matrices), total CPU time (IBM RS6000/370) for CG-DMS, and differences in total energy between diagonalization and CG-DMS for 20-glycine with HF and LDA/3-21G as a function of the neglect threshold.

<table>
<thead>
<tr>
<th>Neglect Threshold (a.u.)</th>
<th>HF</th>
<th></th>
<th></th>
<th>LDA</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Surviving Elements (%)</td>
<td>CPU Time (h)</td>
<td>(\Delta E) ((\mu)hartree)</td>
<td>Surviving Elements (%)</td>
<td>CPU Time (h)</td>
<td>(\Delta E) ((\mu)hartree)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10(^{-5})</td>
<td>25.5</td>
<td>1.9</td>
<td>97.00</td>
<td>25.4</td>
<td>2.5</td>
<td>240.00</td>
<td></td>
</tr>
<tr>
<td>10(^{-6})</td>
<td>31.6</td>
<td>1.9</td>
<td>8.60</td>
<td>31.8</td>
<td>2.4</td>
<td>8.40</td>
<td></td>
</tr>
<tr>
<td>10(^{-7})</td>
<td>39.4</td>
<td>2.8</td>
<td>0.41</td>
<td>39.8</td>
<td>4.1</td>
<td>0.37</td>
<td></td>
</tr>
<tr>
<td>10(^{-8})</td>
<td>47.6</td>
<td>3.9</td>
<td>0.03</td>
<td>48.4</td>
<td>5.2</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>10(^{-9})</td>
<td>56.6</td>
<td>5.5</td>
<td>0.03</td>
<td>56.0</td>
<td>7.4</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>10(^{-10})</td>
<td>64.7</td>
<td>7.6</td>
<td>0.04</td>
<td>64.8</td>
<td>10.0</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>100.0</td>
<td>13.3</td>
<td>0.00</td>
<td>100.0</td>
<td>17.0</td>
<td>0.00</td>
<td></td>
</tr>
</tbody>
</table>

Table 3  Number of SCF cycles and differences in total energy between diagonalization and CG-DMS for 20-glycine with HF and LDA/3-21G as a function of SCF convergence criterion. A neglect threshold of 10\(^{-6}\) a.u. was used for the CG-DMS.

<table>
<thead>
<tr>
<th>SCF Convergence</th>
<th>HF</th>
<th></th>
<th></th>
<th>LDA</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SCF Cycles</td>
<td>(\Delta E) ((\mu)hartree)</td>
<td>SCF Cycles</td>
<td>(\Delta E) ((\mu)hartree)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10(^{-4})</td>
<td>6</td>
<td>3400.0</td>
<td>8</td>
<td>1350.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10(^{-5})</td>
<td>10</td>
<td>32.0</td>
<td>10</td>
<td>22.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10(^{-6})</td>
<td>13</td>
<td>8.7</td>
<td>13</td>
<td>8.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10(^{-7})</td>
<td>14</td>
<td>8.6</td>
<td>14</td>
<td>8.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10(^{-8})</td>
<td>14</td>
<td>8.6</td>
<td>14</td>
<td>8.4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Convergence in the CG-DMS is based on checking whether the measure of idempotency $[\text{Tr}(P^2) - N_e]$, the RMS of the gradient, and the maximum value of the gradient are below a specified threshold. CG-DMS convergence starts with very loose thresholds in the first SCF iteration which become progressively tighter each SCF iteration. This progressive convergence uses a threshold that is directly proportional to the RMS density difference from the previous SCF cycle. This convergence threshold is not allowed to drop below a user specified value ($5 \times 10^{-7}$ in this work). In addition to CG-DMS convergence checks, the CG step is allowed to terminate early and proceed to the next SCF step if the RMS or maximum gradient is shown to be increasing for two CG-DMS cycles in a row. Using these criteria, CG-DMS convergence usually requires between 4-7 CG-DMS iterations per SCF cycle.

CG-DMS is only mildly basis set dependent in terms of the rate of SCF convergence, CPU time, and sparsity. Table 4 compares the performance of both diagonalization and CG-DMS as a function of basis set. STO-3G gives a slightly better performance than the other basis sets, both converging in fewer SCF cycles and having a lower sparsity. The remaining basis sets (3-21G, 6-31G, and 6-31G**) performed similarly to one another and differed in sparsity by only a few percent. The ratio of CPU times between diagonalization and CG-DMS is roughly independent of basis set.

Number of SCF cycles and cumulative CPU timings for the diagonalization step and CG-DMS on a series of polyglycine chains for LDA/3-21G are given in Table 5. The CPU times are also plotted in Figure 4. In all our benchmark calculations, HF yielded results comparable to LDA so we expect similar results
Table 4  Number of SCF cycles, total CPU time (IBM RS6000/370), and percentage of surviving elements (orthonormal Fock and density matrices) using different basis sets for 20-polyglycine with LDA for diagonalization and CG-DMS. A neglect threshold of $10^{-6}$ a.u. and an SCF convergence threshold of $10^{-5}$ a.u. was used for CG-DMS.

<table>
<thead>
<tr>
<th>basis set</th>
<th>Diagonalization</th>
<th>CG-DMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SCF cycles</td>
<td>CPU Time (h)</td>
</tr>
<tr>
<td>STO-3G</td>
<td>11</td>
<td>0.05</td>
</tr>
<tr>
<td>3-21G</td>
<td>10</td>
<td>0.4</td>
</tr>
<tr>
<td>6-31G</td>
<td>11</td>
<td>0.5</td>
</tr>
<tr>
<td>6-31G**</td>
<td>10</td>
<td>4.9</td>
</tr>
</tbody>
</table>

for polyglycine chains using HF. Diagonalization scales cubically while near-linear scaling is observed for CG-DMS. The crossover in CPU time for diagonalization and CG-DMS occurs at 50-glycine. For 70-glycine (2,953 basis functions), we observe that CG-DMS is about 25% faster than diagonalization. The diagonalization step requires $O(N^2)$ memory whereas CG-DMS has a linear memory dependence. In our GAUSSIAN implementation CG-DMS currently takes up less memory than the diagonalization step for polyglycine chains 10-glycine and larger. The memory requirement for polyglycine chains shown in Figure 5 clearly shows linear scaling.

To improve the convergence in the CG-DMS iterations, we implemented a sparse matrix version of direct inversion of iterative subspace (DIIS-CG) [46, 47]. DIIS-CG updates the density matrix every CG cycle with the linear combination of previous density matrices which minimizes the error vector. This is in addition to SCF DIIS
Table 5  Number of SCF cycles, total CPU time (IBM RS6000/370) and percentage of surviving elements (orthonormal Fock and density matrices) for diagonalization and CG-DMS for polyglycine chains using LDA/3-21G. The last 4 chains for diagonalization are extrapolated. A neglect threshold of $10^{-6}$ a.u. and an SCF convergence threshold of $10^{-5}$ a.u. was used for CG-DMS.

<table>
<thead>
<tr>
<th>Molecule</th>
<th>Diagonalization</th>
<th>CG-DMS</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SCF cycles</td>
<td>CPU Time (h)</td>
<td>SCF cycles</td>
<td>CPU Time (h)</td>
</tr>
<tr>
<td>10-glycine</td>
<td>10</td>
<td>0.1</td>
<td>10</td>
<td>0.7</td>
</tr>
<tr>
<td>20-glycine</td>
<td>10</td>
<td>0.6</td>
<td>11</td>
<td>2.3</td>
</tr>
<tr>
<td>30-glycine</td>
<td>12</td>
<td>2.2</td>
<td>12</td>
<td>4.6</td>
</tr>
<tr>
<td>40-glycine</td>
<td>12</td>
<td>5.1</td>
<td>12</td>
<td>7.1</td>
</tr>
<tr>
<td>50-glycine</td>
<td>12</td>
<td>9.8</td>
<td>12</td>
<td>9.4</td>
</tr>
<tr>
<td>60-glycine</td>
<td>12</td>
<td>17.0</td>
<td>12</td>
<td>12.9</td>
</tr>
<tr>
<td>70-glycine</td>
<td>12</td>
<td>26.9</td>
<td>12</td>
<td>15.8</td>
</tr>
</tbody>
</table>
which is used every SCF cycle. DIIS-CG was found to give quicker and smoother convergence when used in the CG-DMS step.

As noted earlier, water clusters pose a greater challenge to the CG-DMS than polyglycine chains because they are more compact and hence their matrices are less sparse. Consequently, the CPU crossover between CG-DMS and diagonalization occurs much later than for polyglycine chains. In this work, calculations were carried out for water clusters containing up to 300 molecules using LDA/STO-3G with four different neglect thresholds: $10^{-4}$, $5 \times 10^{-5}$, $10^{-5}$, and $10^{-6}$ a.u. These thresholds were chosen to correspond to accuracies in the converged energy ranging from milihartree to microhartree. The percentage of surviving F and P matrix elements in the Cholesky orthonormal basis as a function of neglect threshold were discussed above (see Figure 3). CPU timings per SCF cycle for both diagonalization and CG-DMS are presented in Table 6 and Figure 6. CG-DMS calculations with $10^{-4}$ a.u. and $5 \times 10^{-5}$ a.u. neglect thresholds show near-linear behavior and crossover with diagonalization occurs at around 160 and 200 water molecules, respectively. The calculations carried out with neglect thresholds of $10^{-5}$ and $10^{-6}$ a.u., however, are clearly non-linear in the small molecule range but do scale significantly less than cubic for the clusters studied here. For these tighter neglect thresholds, the crossover in CPU time with diagonalization is estimated at 380 and 400 water molecules, respectively. Linear scaling is also expected at these cluster sizes.

There is another key advantage of using CG-DMS: parallelization. It is difficult to parallelize standard diagonalization in an efficient manner for many processors. On the other hand, the time dominant step for CG-DMS is sparse matrix multipli-
Table 6  CPU time per SCF cycle (in minutes of IBM RS6000/370) for diagonalization and CG-DMS using four different neglect thresholds for water clusters using LDA/STO-3G.

<table>
<thead>
<tr>
<th>Molecule</th>
<th>Diag.</th>
<th>CG-DMS (Neglect Threshold in a.u.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$1 \times 10^{-4}$</td>
</tr>
<tr>
<td>(H$<em>2$O)$</em>{30}$</td>
<td>0.0</td>
<td>0.3</td>
</tr>
<tr>
<td>(H$<em>2$O)$</em>{60}$</td>
<td>0.2</td>
<td>1.1</td>
</tr>
<tr>
<td>(H$<em>2$O)$</em>{90}$</td>
<td>0.6</td>
<td>2.4</td>
</tr>
<tr>
<td>(H$<em>2$O)$</em>{120}$</td>
<td>2.5</td>
<td>4.2</td>
</tr>
<tr>
<td>(H$<em>2$O)$</em>{150}$</td>
<td>4.9</td>
<td>5.2</td>
</tr>
<tr>
<td>(H$<em>2$O)$</em>{300}$</td>
<td>39.3</td>
<td>17.6</td>
</tr>
</tbody>
</table>

cation, which can be parallelized by dividing the product matrix into separate sets of rows, which are independently computed on separate nodes. Crossover in CPU with respect to diagonalization will be shifted to smaller systems as more processors are utilized. The effect of parallelization on CPU time for CG-DMS (not including the transformation to and from orthonormal) is given in Figure 7. CG-DMS with parallelization will make calculations possible that are not feasible with standard diagonalization.
Figure 4  Cumulative CPU timings for CG-DMS and diagonalization for polyglycine chains using LDA/3-21G. (Last four points in the diagonalization curve are extrapolated using $N^3$ scaling.)
Figure 5  Total memory requirement for CG-DMS and diagonalization for polyglycine chains using LDA/3-21G.
Figure 6  CPU timings per SCF cycle for CG-DMS and diagonalization for water clusters using LDA/STO-3G.
Figure 7  Parallel performance for CG-DMS on 20-glycine LDA/3-21G on a SGI power challenge.
Chapter 7

Conclusions

The CG-DMS approach is an alternative to diagonalization that scales nearly-linearly in both CPU time and memory for large sparse systems and can be straightforwardly parallelized. In this work, we propose a Cholesky decomposition of the overlap matrix to define a transformation to an orthogonal basis where the conjugate gradient density matrix search is performed. This step is necessary to deal with the non-orthogonal basis problem normally encountered in ab initio calculations. Furthermore, the sparsity of the Fock and density matrices in the orthogonal Cholesky basis is not worse than the sparsity of the density matrix in the AO basis. Linear scaling and CPU time crossover with diagonalization is demonstrated here for polyglycine chains and water clusters using HF and DFT methods. The accuracy of CG-DMS is dependent on the neglect threshold and can be adjusted to desired precision. CG-DMS performs equally well for all the basis sets studied here, including those containing polarization d functions. CG-DMS in conjunction with a linear scaling Fock matrix construction yields $\mathcal{O}(N)$ HF and DFT methods and make it possible to do accurate studies of bio-molecules and other nanostructures which previously have not been computationally accessible.
References


[43] The polyglycine chains are similar to those described in Refs. [31, 32, 33]. Coordinates for these molecules are available from us.

[44] The water clusters up to 150 water molecules are those used in Ref. [9]. We obtained the coordinates from the authors of Ref. [9]. Coordinates for the 300 water molecule cluster are available from us.

