Density matrix search using direct inversion in the iterative subspace as a linear scaling alternative to diagonalization in electronic structure calculations

Xiaosong Li
Department of Chemistry, Wayne State University, Detroit, Michigan 48202-3489

John M. Millam
Semichem, Shawnee Mission, Kansas 66222

Gustavo E. Scuseria
Department of Chemistry, Rice University, Houston, Texas 77005-1892

Michael J. Frisch
Gaussian Inc., North Haven, Connecticut 06473-1712

H. Bernhard Schlegel
Department of Chemistry, Wayne State University, Detroit, Michigan 48202-3489

(Received 11 June 2003; accepted 21 July 2003)

For electronic structure calculations on large systems, solving the self-consistent-field (SCF) equations is one of the key bottlenecks. Density matrix search methods provide an efficient linear scaling approach for circumventing the traditional $O(N^3)$ diagonalization procedure for solving the SCF equations. The conjugate gradient density matrix search (CG-DMS) method is a successful implementation of this approach. An alternative density matrix search method, QN–DMS, employs direct inversion in the iterative subspace using a quasi-Newton (QN) step to estimate the error vector. For linear polyglycine chains of 10–100 residues, the present approach scales linearly and is 30% faster than CG-DMS. For clusters of up to 300 water molecules, this method shows smoother and efficient convergence, and displays nearly linear scaling. © *2003 American Institute of Physics.* [DOI: 10.1063/1.1607961]

I. INTRODUCTION

Modeling and simulation methods are being applied to systems of ever increasing size, placing greater demands on computational hardware and software. Advances in computer technology continue to provide increasing processor speed and memory size at decreasing cost. Linear scaling algorithms for electronic structure calculations have greatly improved the outlook for applications of *ab initio* calculations on large systems. For biomolecules, polymers and nanostructures, molecular orbital calculations deal with thousands of basis functions, and any decrease in the formidable cost of the computations would be welcome. The rate limiting step in Hartree–Fock (HF) and density functional theory (DFT) calculations used to be the computation of two-electron integrals. Recently, the cost of forming the Coulomb matrix has been reduced to near-linear scaling for large systems by using the fast multipole method.[1–4] Both the ''order-N exchange'' (ONX) (Ref. 5) and ''near-field-exchange'' (NFX) (Ref. 6) methods overcome the asymptotic $O(N^2)$ bottleneck for the exchange matrix and achieve linear scaling in the large molecule limit. This leaves the convergence of the self-consistent-field (SCF) equations as the remaining obstacle in HF and DFT calculations on large systems (for leading reference on the SCF scaling, see Ref. 7).

In the traditional Roothaan SCF procedure, the density matrix is used to form the Fock matrix, and the Fock matrix is diagonalized to yield new molecular orbital coefficients and a new density matrix; the process is repeated until self-consistency is achieved. The direct inversion in the iterative subspace (DIIS) technique[8,9] greatly reduces the number of iterations required to reach self-consistency compared to the Roothaan SCF procedure. The idea of DIIS is to optimize the variables by minimizing the errors in the least-squares sense within a subspace. Pulay's approach uses the commutator (**FPS**−**SPF**) as the error vectors and interpolation or extrapolation of the Fock matrices (denoted here by C-DIIS). More recently, Kudin, Scuseria, and Cances developed the E-DIIS algorithm, which minimizes the energy by interpolation of the density within the iterative subspace.[10] This method provides enhanced stability in solving the SCF equations, but for large systems the $O(N^3)$ Fock matrix diagonalization remains the dominant time-consuming step. Furthermore, the procedure has an $O(N^2)$ memory requirement which is a stringent limitation for large systems.

Several categories of methods have been proposed in literatures as alternatives to diagonalization (for leading reference, see Ref. 11): Fermi operator expansion,[12–14] divide-and-conquer,[15–17] direct minimization of the energy with respect to the density (under the constraint that the density remains idempotent),[18,19] etc. Of various implementations and benchmarks,[20–34] conjugate gradient density matrix search (CG-DMS) (Ref. 19) using sparse matrix techniques, developed by Millam and Scuseria in 1997, appears to be one of the best.

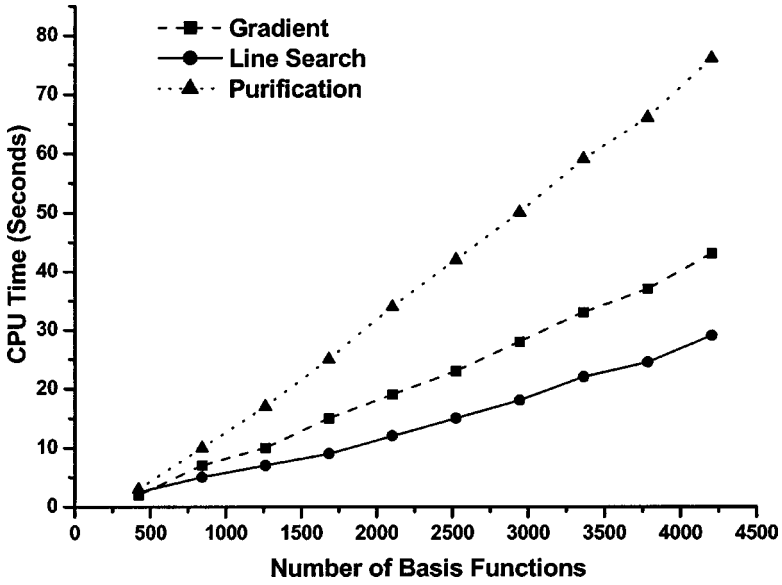

FIG. 1. Comparison of costs for the gradient, line search, and purification for polyglycine chains.

In the CG–DMS approach, the energy is minimized with respect to the density along a series of conjugate directions based on the Li–Nunes–Vanderbilt (LNV) (Ref. 18) functional using diagonally preconditioners and cubic line searches. The CG-DMS method scales linearly with system size and is cost effective in large systems. Overall $O(N)$ scaling in CPU time and memory is achieved for linear chains such as polyglycine. For three-dimensional systems like water clusters, CG-DMS shows linear scaling for individual SCF cycles. An improvement to CG-DMS uses Pulay's DIIS approach to update the density matrix every DMS cycle, giving quicker and smoother convergence.[19] In the current paper, the abbreviation CG-DMS denotes the conjugate gradient density matrix search method in conjunction with C-DIIS. Within an individual SCF cycle, a large fraction of the cost is for the conjugate search step. Thus, there may still be room to improve the efficiency and reduce the cost of SCF convergence for large systems.

For geometry optimization, the quasi-Newton and DIIS algorithms are more efficient than conjugate gradient methods.[35] In the current paper, we present a density matrix search method for SCF convergence that employs a DIIS approach using a quasi-Newton step as the error vector (QN-DIIS-DMS or QN-DMS for short). This improved method is compared to diagonalization and CG-DMS for a series of polyglycine chains and water clusters.

II. METHODOLOGY

In the density matrix search method,[19] each SCF cycle starts by forming the Fock matrix **F** using the density matrix **P** from the previous cycle (or the initial guess if it is the first entry). After the first step, a C-DIIS approach can be used to extrapolate or interpolate the Fock matrix. Instead of diagonalizing the Fock matrix, a series of density matrix search steps are employed. In QN-DMS, a set of quasi-Newton steps are taken and are combined using DIIS. If the convergence criteria are not fulfilled, the calculation proceeds to the next SCF cycle.

In the density matrix search, an estimate of the error in a minimization can be obtained from a quasi-Newton step:

$$\Delta \mathbf{x}_i = -\mathbf{H}^{-1}\mathbf{g}_i = \mathbf{e}_i , \tag{1}$$

TABLE I. Cumulative CPU time (hours) and number of SCF cycles for polyglycine chains at LSDA/3-21G with threshold of 10^{-8} a.u.

	Diagonalization		CG-DMS			QN-DMS			
Molecule	SCF cycles	CPU time	SCF cycles	Total CG iterations	CPU time	SCF Cycles	Total QN iterations	CPU time	Surviving elements[a]
10-glycine	13	0.08	8	32	0.14	7	24	0.08	67.9
20-glycine	13	0.33	8	32	0.48	8	26	0.33	54.3
30-glycine	16	0.87	9	36	0.90	8	26	0.55	38.5
40-glycine	16	1.64	9	36	1.14	8	26	0.75	29.8
50-glycine	16	2.95	9	36	1.61	8	26	1.01	24.9
60-glycine	16	4.96	9	36	1.96	8	26	1.23	21.2
70-glycine	17	7.82	9	36	2.25	9	28	1.55	18.3
80-glycine	19	11.85	9	36	2.63	9	28	1.84	16.2
90-glycine	21	18.84	9	36	2.95	9	28	2.11	14.5
100-glycine	17	19.78	9	36	3.30	9	28	2.36	13.1

[a]Percentage surviving elements of the union of **F**, **P**, **FP**+**PF**, and **PFP** at the begining of the SCF.

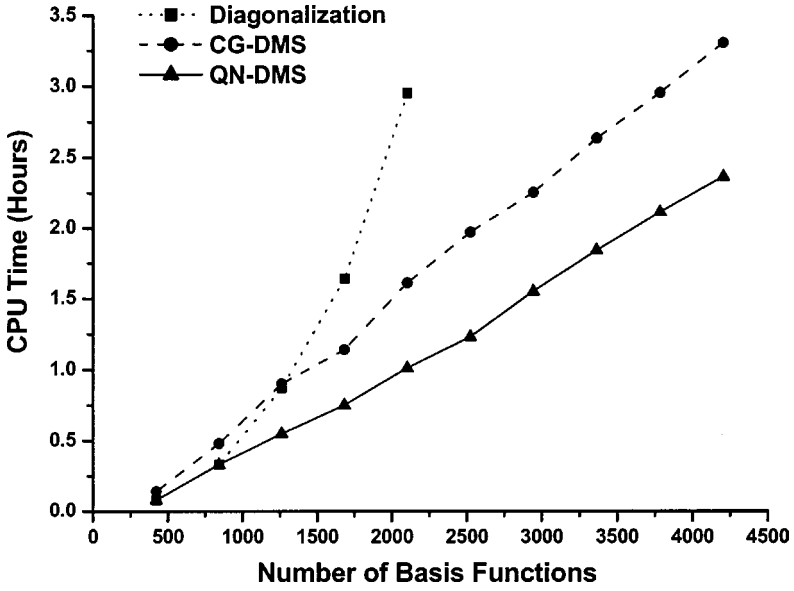

FIG. 2. Cumulative CPU time of the diagonalization, CG-DMS, and QN-DMS methods for polyglycine chains at LSDA/3-21G (threshold of 10^{-8} a.u.).

where $\mathbf{g}_i$ is the gradient and $\mathbf{H}$ is an approximate Hessian. In calculating the gradient of the energy with respect to the density, one must include the idempotency constraint. This can be done by expressing the energy in terms of the McWeeny purified density, $\tilde{\mathbf{P}}=3\mathbf{P}^2-2\mathbf{P}^3$. As in CG-DMS (Ref. 19) and the atom-centered basis functions density matrix propagation (ADMP) method for *ab initio* molecular dynamics,[36–38] the gradient is a modified LNV (Ref. 18) functional,

$$\mathbf{g}=dE(\tilde{\mathbf{P}})/d\mathbf{P}=3\mathbf{FP}+3\mathbf{PF}-2\mathbf{FP}^2-2\mathbf{PFP}-2\mathbf{P}^2\mathbf{F}, \quad (2)$$

where $\mathbf{F}$ and $\mathbf{P}$ are the Fock and density matrices, respectively, in an orthonormal basis. If the density matrix is nearly idempotent, the diagonal elements of the gradient matrix are very close to zero. Of the various choices for the transformation to an orthonormal space, Löwdin and Cholesky transformations are the most widely used. The transformation of the density and Fock matrices from the atomic orbital (AO) basis ($\mathbf{P}_{\mathrm{AO}}$ and $\mathbf{F}_{\mathrm{AO}}$) into the orthonormal basis can be expressed as

$$\mathbf{P}=\mathbf{U}\mathbf{P}_{\mathrm{AO}}\mathbf{U}^T, \quad \mathbf{F}=\mathbf{U}^{-T}\mathbf{F}_{\mathrm{AO}}\mathbf{U}^{-1}, \quad (3)$$

where $\mathbf{U}$ can be obtained by Cholesky decomposition ($\mathbf{S}=\mathbf{U}^T\mathbf{U}$, where $\mathbf{U}$ is upper triangular),[39] or $\mathbf{U}=\mathbf{S}^{1/2}$ for Löwdin orthonormalization ($\mathbf{S}$ is the overlap matrix). The Cholesky orthonormal basis is more favorable for sparse matrix manipulations since forming the Cholesky decomposition is $O(N)$ scaling if $\mathbf{S}$ is sparse, and the resulting transformation matrix $\mathbf{U}$ as well as the resulting $\mathbf{P}$ is also sparse.

In the CG-DMS method, a very efficient preconditioner can be constructed from the diagonal elements of the Hessian[19]:

$$\begin{aligned} H_{\mu\nu\mu\nu}&=d^2E(\tilde{\mathbf{P}})/dP_{\mu\nu}^2 \\ &=(3-2\mathbf{P}_{\mu\mu})\mathbf{F}_{\nu\nu}+(3-2\mathbf{P}_{\nu\nu})F_{\mu\mu} \\ &\quad -4(\mathbf{PF})_{\mu\mu}-4(\mathbf{PF})_{\nu\nu}. \end{aligned} \quad (4)$$

Since the storage and computing requirements for the diagonal Hessian are minimal, we choose to use it in estimating the error vector, Eq. (1), for the DIIS procedure. Within each DMS iteration, the Hessian is reevaluated using the fixed $\mathbf{F}$ and the new $\mathbf{P}$ from the previous DMS step, and subsequently error vectors used in the DIIS equation are reconstructed using the saved gradients and the new Hessian.

The DIIS method is employed to obtain a linear extrapolation and interpolation of the saved density matrices

$$\mathbf{P}^*=\Sigma c_i\mathbf{P}_i, \quad (5)$$

TABLE II. Cumulative CPU time (hours) and number of SCF cycles for polyglycine chains at LSDA/6-31G(d,p) with threshold of 10^{-8} a.u.

	Diagonalization		CG-DMS			QN-DMS			
Molecule	SCF cycles	CPU time	SCF cycles	Total CG iterations	CPU time	SCF Cycles	Total QN iterations	CPU time	Surviving elements[a]
10-glycine	12	0.30	8	32	0.63	8	31	0.46	75.7
20-glycine	12	1.21	9	36	2.52	8	31	1.54	54.6
30-glycine	16	3.94	9	36	3.96	8	31	2.46	38.9
40-glycine	16	8.48	9	36	5.10	8	32	3.20	29.7
50-glycine	15	14.4	9	36	6.82	9	34	4.76	25.1
60-glycine			9	36	8.76	9	34	5.97	21.4

[a]Percentage surviving elements of the union of $\mathbf{F}$, $\mathbf{P}$, $\mathbf{FP}+\mathbf{PF}$, and $\mathbf{PFP}$ at the begining of the SCF.

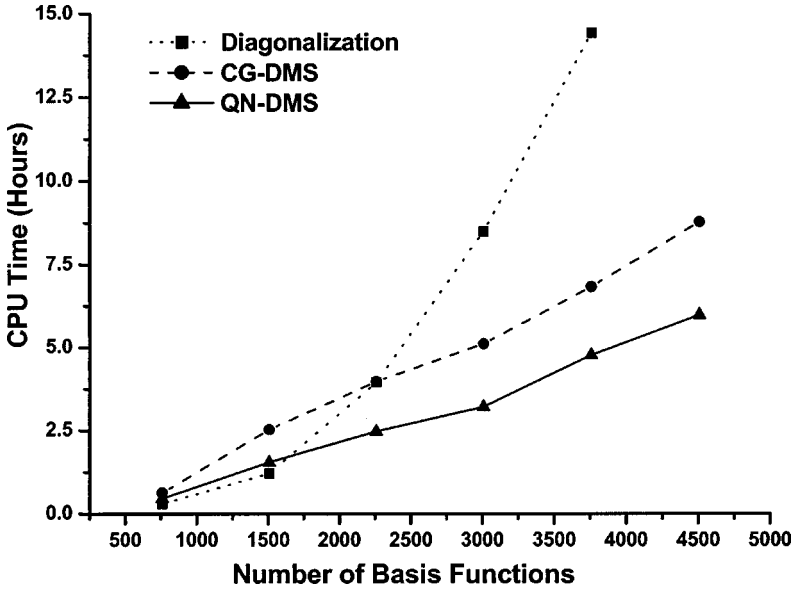

FIG. 3. Cumulative CPU time of the diagonalization, CG-DMS, and QN-DMS methods for polyglycine chains at LSDA/6-31G(d,p) (threshold of 10^{-8} a.u.).

where $\Sigma c_i = 1$, to minimize the length of the error vector. The error for $\mathbf{P}^*$ is the linear combination of the previous error vectors,

$$\mathbf{e}^* = \Sigma c_i \mathbf{e}_i = \Delta \mathbf{P}. \tag{6}$$

Since the error vectors are chosen as the quasi-Newton estimates of the change in the density, $\mathbf{e}_i$, the linear combination of the error vectors is also the predicted change in the density, $\mathbf{P}^*$. The coefficients c_i are obtained by minimizing $|\mathbf{e}_i|^2$ or $|\Delta \mathbf{P}|^2$ with the constraint $\Sigma c_i = 1$. This least-squares problem leads to the following equation[8,9]:

$$\begin{pmatrix} a_{1,1} & \cdots & a_{1,k} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{k,1} & \cdots & a_{k,k} & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_k \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \tag{7}$$

where $a_{i,j} = \mathrm{Tr}(e_i^T e_j)$ and λ is a Lagrangian multiplier. The dimension of $\mathbf{A}$ is small and the equation can be solved by any standard method. The new density matrix is then calculated using the expression

$$\mathbf{P}_{i+1} = \mathbf{P}^* + \Delta \mathbf{P} = \Sigma c_i \mathbf{P}_i + \Sigma c_i \mathbf{e}_i . \tag{8}$$

The new density matrix is scaled to enforce the constraint on the number of electrons, $\mathrm{Tr}(\mathbf{P}) = N_e$,

$$\lambda = N_e / \mathrm{Tr}(\mathbf{P}), \tag{9}$$

$$\mathbf{P}_{i+1} = \lambda \mathbf{P}_{i+1}. \tag{10}$$

Because the diagonal elements of the gradient are very close to zero, the scale factor λ normally falls in the range of 1.0 ± 10^{-7} and serves to control numerical noise. Several McWeeny purification transformations,[40] $\tilde{\mathbf{P}} = 3\mathbf{P}^2 - 2\mathbf{P}^3$, are applied to restore idempotency, $\mathbf{PP} = \mathbf{P}$. At the beginning of the next SCF cycle, the Fock matrix is recalculated and extrapolated using the C-DIIS method.

To yield linear scaling, sparse matrix technique must be used. The cost is reduced by storing and manipulating only the significant elements using a set of sparse matrix codes.[41] The threshold is chosen to be in the range of 10^{-5}–10^{-8} a.u., which has been shown to be cost effective while still yielding microhartree accuracy in the energy.[19] In every SCF cycle, the Fock and density matrices have the same form, which is chosen as the union of **F**, **P**, **PF**+**FP**, and **PFP** as in CG-DMS.

TABLE III. Percentage of surviving elements[a] of the matrix for water clusters at LSDA/STO-3G.

	Sparsity threshold (a.u.)			
Molecule	1×10^{-5}	1×10^{-6}	1×10^{-7}	1×10^{-8}
$(H_2O)_{30}$	66.4	87.9	98.1	99.8
$(H_2O)_{60}$	45.2	69.9	89.6	98.3
$(H_2O)_{90}$	35.1	57.4	79.7	94.2
$(H_2O)_{120}$	28.3	48.1	69.6	87.3
$(H_2O)_{150}$	22.4	39.4	61.2	80.3
$(H_2O)_{300}$	13.4	24.6	39.3	55.8

[a]Percentage surviving elements of the union of **F**, **P**, **FP**+**PF**, and **PFP** at the begining of the SCF.

III. BENCHMARKS AND DISCUSSION

The test cases are calculated using the development version of GAUSSIAN series of program[42] with the addition of QN-DMS algorithm presented here. All timing data are obtained on an Athlon 1800+MP workstation. For all methods considered here, an SCF convergence threshold of 10^{-4} a.u. for the root-mean-square (rms) density change, 10^{-2} a.u. for the maximum density change, and 10^{-4} a.u. for the energy change are used. In the local density matrix search procedures (both CG-DMS and QN-DMS), the idempotency and rms of the gradient are converged to thresholds of 10^{-12} and 10^{-4}, respectively. For the conventional SCF algorithm with diagonalization, the E-DIIS method was used.[10] As in previous papers,[19] polyglycine chains and water clusters are used as test cases.

In a CG-DMS iteration, calculations of the gradient, line search and purification are the most expensive components. As shown in Fig. 1 for LSDA/3-21G calculations on polyg-

TABLE IV. Cumulative CPU time (minutes) and number of SCF cycles for water clusters at LSDA/STO-3G.

			CG-DMS						QN-DMS					
	Diagonalization		1×10^{-5} a			1×10^{-8} a			1×10^{-5} a			1×10^{-8} a		
Molecule	SCF cycles	CPU time	SCF cycles	Total CG iter.	CPU time	SCF cycles	Total QN iter.	CPU time	SCF cycles	Total QN iter.	CPU time	SCF cycles	Total QN iter.	CPU time
$(H_2O)_{30}$	7	4.6	6	19	3.9	6	17	4.5	6	19	3.9	6	19	4.2
$(H_2O)_{60}$	8	17.1	6	24	13.3	6	17	18.5	6	19	13.0	6	19	16.1
$(H_2O)_{90}$	14	56.0	16	64	68.9	6	24	41.8	6	19	25.6	6	19	34.0
$(H_2O)_{120}$	7	48.1	10	40	68.2	7	22	71.1	6	19	42.2	6	19	56.7
$(H_2O)_{150}$	16	155.1	19	76	183.8	7	28	121.3	6	19	58.1	6	19	85.8
$(H_2O)_{200}$	13	233.2	7	28	129.0	7	22	180.7	6	18	94.0	6	18	140.8
$(H_2O)_{250}$	18	479.3	8	32	197.9	7	22	274.7	6	18	140.5	6	18	220.5
$(H_2O)_{300}$	19	693.5	7	28	235.8	7	21	364.3	6	19	196.0	6	19	318.2

[a]Threshold for sparse matrix manipulation, in a.u.

lycine chains, each of the components scales linear, which consequently result in overall linear scaling. QN-DMS uses the same modules to calculate the gradient and purify the density as the CG-DMS, but has the advantage of avoiding the expensive line search. Although the QN-DMS involves the approximate Hessian and the solution of the DIIS equations, the cost of these additional computations is trivial compared to the line search.

In a few cases, when the density is far away from convergence or when the surface is very anharmonic and some elements of the Hessian are very small or negative, a simple quasi-Newton step can be too large. Such situations result in a density that is no longer idempotent even after numerous McWeeny purifications. If any of the diagonal elements of the Hessian are too small or negative, the entire Hessian is shifted so that the smallest value is greater than 0.01 a.u. The stability can be improved further by controlling the step size using techniques such as the rational function optimization (RFO) (Refs. 43–45) or the trust radius model (TRM) (Refs. 35 and 46–50), which are frequently used in geometry optimization. Alternatively, we can use the CG-DMS method, which employs a cubic line search, to obtain an optimal step in the descent direction. This can be more stable than a simple quasi-Newton step outside the near-quadratic region. In our implementation, we switch to the CG-DMS method when the density matrix generated from QN-DMS is no longer idempotent. Among all the tests, this switch is required only in the first step for the 90 water cluster at LSDA/6-31G(d,p).

Table I compares the costs for SCF convergence via diagonalization, CG-DMS and QN-DMS for polyglycine chains at the LSDA/3-21G level of theory. In these systems, a cutoff threshold of 10^{-8} is chosen for the matrix elements, which leads to a very reasonable sparsity. Cumulative CPU timings for SCF convergence are plotted in Fig. 2. As might be anticipated, diagonalization becomes very expensive for large systems because of the $O(N^3)$ scaling with the number of basis functions. Both CG-DMS and QN-DMS methods achieve overall linear scaling and are more cost effective than diagonalization for large systems. However, QN-DMS requires fewer iterations in the density matrix search to reach convergence, an average of 3.3 iterations. By comparison,

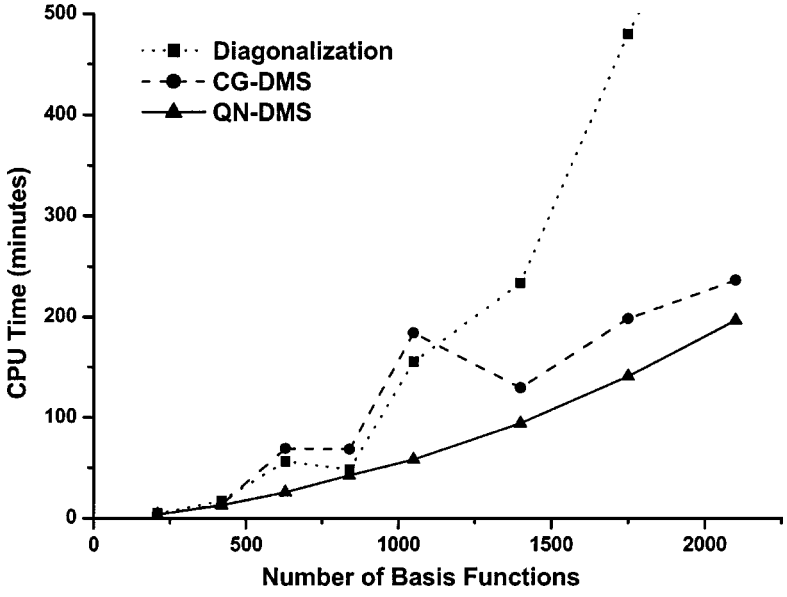

FIG. 4. Cumulative CPU time of the diagonalization, CG-DMS, and QN-DMS methods for water clusters at LSDA/STO-3G (threshold of 10^{-5} a.u.).

TABLE V. Absolute deviation in energy (kcal/mol) compared with SCF=TIGHT for water clusters at LSDA/STO-3G.

		CG-DMS		QN-DMS	
Molecule	Diagonalization	1×10^{-5} [a]	1×10^{-6} [a]	1×10^{-5} [a]	1×10^{-6} [a]
$(H_2O)_{30}$	0.01	0.03	0.03	0.02	0.03
$(H_2O)_{60}$	0.07	0.18	0.09	0.12	0.08
$(H_2O)_{90}$	0.93	0.41	0.11	0.03	0.03
$(H_2O)_{120}$	0.08	0.35	0.15	0.18	0.12
$(H_2O)_{150}$	2.27	0.33	0.12	0.07	0.01
$(H_2O)_{200}$	5.74	0.32	0.07	0.01	0.04
$(H_2O)_{250}$	0.54	0.40	0.01	0.03	0.01
$(H_2O)_{300}$	4.18	0.47	0.10	0.03	0.05
Ave. error	1.73	0.31	0.09	0.06	0.05

[a]Threshold for sparse matrix manipulation, in a.u.

the CG-DMS method does not often reach convergence within the maximum of 4 iterations allowed. Consequently, SCF convergence by QN-DMS requires on average one less SCF cycles in half of the cases.

Larger basis sets with polarization functions, such as 6-31G(d,p), have more coupling between the orbital coefficients and much less sparsity than STO-3G or 3-21G basis sets, which substantially increases the computational cost. Table II and Fig. 3 compare the three methods considered here for polyglycine chains at the LSDA/6-31G(d,p) level of theory. The cost of both QN-DMS and CG-DMS increase linearly with the chain length, and the ratio of the CPU times is very similar to that obtained at the LSDA/3-21G level of theory. The overall performance of QN-DMS is about 30% faster than CG-DMS on polyglycine chains. This conclusion holds for HF/3-21G as well as for the LSDA/3-21G and LSDA/6-31G(d,p) results presented here.

Three-dimensional water clusters are more challenging tests than linear polyglycine chains because they are more compact and hence the matrices are less sparse. Table III lists the percentage of surviving matrix elements at different sparsity thresholds for a set of water clusters. Since water clusters are much less sparse than polyglycine chains for a same threshold, the cutoff inevitably has a greater effect on the cost and scaling of the computation. Cumulative CPU timings and the number of SCF cycles for diagonalization, CG-DMS, and QN-DMS density matrix search with two different cutoff values are presented in Table IV for water clusters at the LSDA/STO-3G level of theory. The number of iterations required to reach self-consistency in density using diagonalization varies from 7 to 19, indicating a strong dependence of the SCF convergence on the size and configuration of the system. In contrast, QN-DMS shows almost no dependency on the structure of the system, and converges quickly and smoothly compared to the other two methods. Cumulative CPU times for the diagonalization, CG and QN-DMS with 10^{-5} cutoff threshold are plotted in Fig. 4. The cost of the diagonalization method increases dramatically with the system size. Because of differences in the number of SCF cycles needed to reach convergence, the average cost of CG-DMS is significantly greater than QN-DMS, varying from 18% to 215% for 90–300 water clusters. QN-DMS shows stable near-linear scaling for water clusters.

Some understanding of the difference in the convergence behavior of CG-DMS and QN-DMS can be deduced by comparing the SCF energies from the present DMS calculations

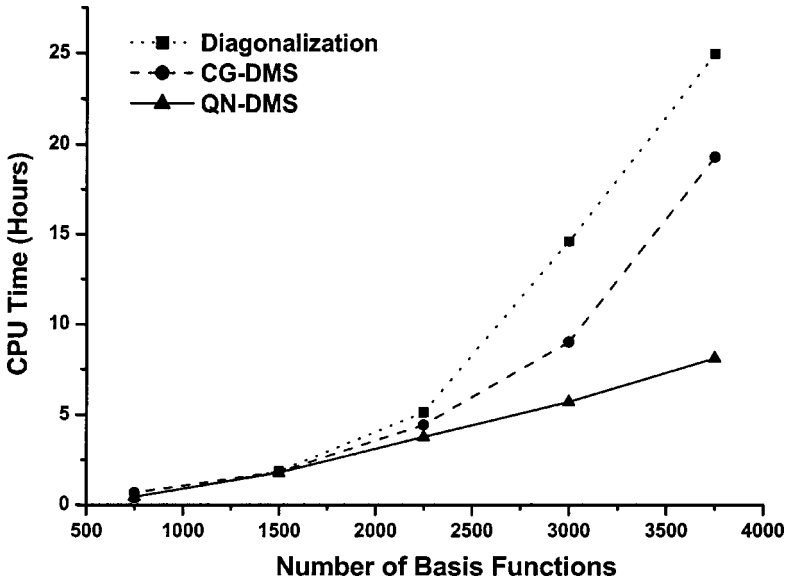

FIG. 5. Cumulative CPU time of the diagonalization, CG-DMS, and QN-DMS methods for water clusters at LSDA/6-31G(d,p) (threshold of 10^{-5} a.u.).

TABLE VI. Cumulative CPU time (hours) and number of SCF cycles for water clusters at LSDA/6-31G(d,p).

			CG-DMS						QN-DMS					
	Diagonalization		1×10^{-5}[a]			1×10^{-8}[a]			1×10^{-5}[a]			1×10^{-8}[a]		
Molecule	SCF cycles	CPU time	SCF cycles	Total CG iter.	CPU time	SCF cycles	Total QN iter.	CPU time	SCF cycles	Total QN iter.	CPU time	SCF cycles	Total QN iter.	CPU time
$(H_2O)_{30}$	11	0.40	10	40	0.68	8	31	0.99	8	22	0.44	7	24	0.68
$(H_2O)_{60}$	12	1.86	8	32	1.83	8	31	6.78	8	24	1.78	7	24	3.82
$(H_2O)_{90}$	14	5.13	10	40	4.43	8	32	15.6	9	27	3.77	8	27	10.1
$(H_2O)_{120}$	21	14.6	13	52	9.01	9	36	31.7	9	24	5.70	9	29	20.4
$(H_2O)_{150}$	21	25.0	20	80	19.3				9	26	8.11			

[a]Threshold for sparse matrix manipulation, in a.u.

to more tightly converged calculations using diagonalization and convergence of 10^{-8} on the rms density difference (Table V). With a sparsity threshold of 10^{-5}, the SCF energies obtained by QN-DMS are a factor of 5 closer to the accurate energy than CG-DMS. A smaller sparsity cutoff leads to a smoother energy landscape, and with a threshold of 10^{-6} or smaller, QN-DMS and CG-DMS both converge smoothly and accurately. This suggests that the difference in the behavior seen with a 10^{-5} cutoff can be attributed to the fact that DIIS is more efficient and robust than CG for difficult optimization problems. The DIIS approach can be thought of a multidimensional interpolation and extrapolation and is better able to handle stronger coupling and anharmonicity than CG methods with line searches.

Calculations on water clusters with the 6-31G(d,p) basis set are more demanding since the increased coupling can make SCF convergence more difficult and the lower degree of sparsity can delay the onset of linear scaling. As can be seen in Fig. 5, QN-DMS shows more nearly linear behavior than CG-DMS with a 10^{-5} sparsity cutoff. The data in Table VI indicates that QN-DMS requires substantially fewer SCF cycles and/or DMS search iterations and has a more uniform rate of convergence. Similar results are also obtained for tight SCF convergence criteria. For example, QN-DMS is ~40% faster than CG-DMS for LSDA/6-31G(d,p) calculation on $(H_2O)_{90}$ using 10^{-8} for convergence on the rms density changes.

IV. CONCLUSION

QN-DMS is another alternative to diagonalization that scales linearly in both CPU time and memory when used in conjunction with a linear scaling Fock matrix construction and sparse matrix manipulations. By employing direct inversion in the iterative subspace with a quasi-Newton method to estimate the error, QN-DMS avoids the cost of the line search and speeds up the SCF convergence compared to the CG-DMS. QN-DMS shows smooth and fast convergence and usually requires fewer SCF cycles than CG-DMS and diagonalization methods. Both density matrix search methods are far cheaper than the diagonalization for large systems. The speed of the convergence using QN-DMS is not strongly dependent on the size and configuration of the system. The overall cost of QN-DMS is about 30% less than CG-DMS for polyglycine chains. In three-dimensional water clusters, QN-DMS shows nearly linear scaling in overall cost with both the STO-3G and 6-31G(d,p) basis sets.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (Grant Nos. CHE 0131157, CHE 9982156, and CISE 9977815), and Gaussian Inc.

[1] L. Greengard and V. Rokhlin, J. Comput. Phys. **73**, 325 (1987).
[2] M. C. Strain, G. E. Scuseria, and M. J. Frisch, Science **271**, 51 (1996).
[3] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chem. Phys. Lett. **253**, 268 (1996).
[4] C. A. White and M. Head-Gordon, J. Chem. Phys. **101**, 6593 (1994).
[5] E. Schwegler and M. Challacombe, J. Chem. Phys. **105**, 2726 (1996).
[6] J. C. Burant, G. E. Scuseria, and M. J. Frisch, J. Chem. Phys. **105**, 8969 (1996).
[7] G. E. Scuseria, J. Phys. Chem. A **103**, 4782 (1999).
[8] P. Pulay, Chem. Phys. Lett. **73**, 393 (1980).
[9] P. Pulay, J. Comput. Chem. **3**, 556 (1982).
[10] K. N. Kudin, G. E. Scuseria, and E. Cances, J. Chem. Phys. **116**, 8255 (2002).
[11] S. Goedecker, Rev. Mod. Phys. **71**, 1085 (1999).
[12] S. Goedecker and L. Colombo, Phys. Rev. Lett. **73**, 122 (1994).
[13] S. Goedecker, J. Comput. Phys. **118**, 261 (1995).
[14] S. Goedecker and M. Teter, Phys. Rev. B **51**, 9455 (1995).
[15] W. T. Yang, Phys. Rev. Lett. **66**, 1438 (1991).
[16] W. T. Yang, J. Chem. Phys. **94**, 1208 (1991).
[17] Q. S. Zhao and W. T. Yang, J. Chem. Phys. **102**, 9598 (1995).
[18] X. P. Li, W. Nunes, and D. Vanderbilt, Phys. Rev. B **47**, 10891 (1993).
[19] J. M. Millam and G. E. Scuseria, J. Chem. Phys. **106**, 5569 (1997).
[20] D. R. Bowler and M. J. Gillan, Comput. Phys. Commun. **120**, 95 (1999).
[21] P. D. Haynes and M. C. Payne, Phys. Rev. B **59**, 12173 (1999).
[22] M. Challacombe, J. Chem. Phys. **110**, 2332 (1999).
[23] U. Stephan and D. A. Drabold, Phys. Rev. B **57**, 6391 (1998).
[24] W. Kohn, Phys. Rev. Lett. **76**, 3168 (1996).
[25] E. Hernandez, M. J. Gillan, and C. M. Goringe, Phys. Rev. B **53**, 7147 (1996).
[26] A. E. Carlsson, Phys. Rev. B **51**, 13935 (1995).
[27] A. H. Sameh and J. A. Wisniewski, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **19**, 1243 (1982).
[28] C. H. Xu and G. E. Scuseria, Chem. Phys. Lett. **262**, 219 (1996).
[29] A. D. Daniels, J. M. Millam, and G. E. Scuseria, J. Chem. Phys. **107**, 425 (1997).
[30] A. D. Daniels, J. M. Millam, and G. E. Scuseria, Abstr. Pap. - Am. Chem. Soc. **215**, 373 (1998).
[31] K. R. Bates, A. D. Daniels, and G. E. Scuseria, J. Chem. Phys. **109**, 3308 (1998).
[32] A. D. Daniels and G. E. Scuseria, J. Chem. Phys. **110**, 1321 (1999).
[33] A. D. Daniels and G. E. Scuseria, Phys. Chem. Chem. Phys. **2**, 2173 (2000).
[34] A. M. N. Niklasson, C. J. Tymczak, and M. Challacombe, J. Chem. Phys. **118**, 8611 (2003).
[35] R. Fletcher, *Practical Methods of Optimization* (Wiley, Chichester, 1981).

[36] H. B. Schlegel, J. M. Millam, S. S. Iyengar, G. A. Voth, A. D. Daniels, G. E. Scuseria, and M. J. Frisch, J. Chem. Phys. **114**, 9758 (2001).
[37] S. S. Iyengar, H. B. Schlegel, J. M. Millam, G. A. Voth, G. E. Scuseria, and M. J. Frisch, J. Chem. Phys. **115**, 10291 (2001).
[38] H. B. Schlegel, S. S. Iyengar, X. S. Li, J. M. Millam, G. A. Voth, G. E. Scuseria, and M. J. Frisch, J. Chem. Phys. **117**, 8694 (2002).
[39] G. H. Golub and C. F. van Loan, *Matrix Computations* (Johns Hopkins University Press, Baltimore, 1996).
[40] R. McWeeny, Rev. Mod. Phys. **32**, 335 (1960).
[41] S. Pissanetsky, *Sparse Matrix Technology* (Academic, New York, 1984).
[42] M. J. Frisch, G. W. Trucks, H. B. Schlegel *et al.*, computer code GAUSSIAN 01 (Development Version), Gaussian, Inc., Pittsburgh, 2002.
[43] A. Banerjee, N. Adams, J. Simons, and R. Shepard, J. Phys. Chem. **89**, 52 (1985).
[44] J. Simons and J. Nichols, Int. J. Quantum Chem. **24**, 263 (1990).
[45] O. Farkas and H. B. Schlegel, J. Chem. Phys. **111**, 10806 (1999).
[46] H. B. Schlegel, in *Modern Electronic Structure Theory*, edited by D. R. Yarkony (World Scientific, Singapore, 1995).
[47] H. B. Schlegel, in *Encyclopedia of Computational Chemistry* (Wiley, New York, 1998).
[48] *Non-linear Optimization*, edited by M. J. D. Powell (Academic, New York, 1982).
[49] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Non-linear Equations*. (Prentice-Hall, Englewood Cliffs, NJ, 1983).
[50] L. E. Scales, *Introduction to Non-linear Optimization* (Macmillam, Basingstoke (Springer-Verlag, New York, 1985).