Improved Fermi operator expansion methods for fast electronic structure calculations

WanZhen Liang
Department of Chemistry and Department of Chemical Engineering, University of California, Berkeley, California 94720

Chandra Saravanan and Yihan Shao
Department of Chemistry, University of California, Berkeley, California 94720

Roi Baer
Department of Physical Chemistry, the Hebrew University, Jerusalem 91904, Israel

Alexis T. Bell
Department of Chemical Engineering, University of California, Berkeley, California 94720

Martin Head-Gordon[a)]
Department of Chemistry, University of California, Berkeley, California 94720

(Received 22 April 2003; accepted 19 May 2003)

Linear scaling algorithms based on Fermi operator expansions (FOE) have been considered significantly slower than other alternative approaches in evaluating the density matrix in Kohn–Sham density functional theory, despite their attractive simplicity. In this work, two new improvements to the FOE method are introduced. First, novel fast summation methods are employed to evaluate a matrix polynomial or Chebyshev matrix polynomial with matrix multiplications totalling roughly twice the square root of the degree of the polynomial. Second, six different representations of the Fermi operators are compared to assess the smallest possible degree of polynomial expansion for a given target precision. The optimal choice appears to be the complementary error function. Together, these advances make the FOE method competitive with the best existing alternatives. © *2003 American Institute of Physics.*
[DOI: 10.1063/1.1590632]

I. INTRODUCTION

Polynomial expansions of matrix functions $f(\mathbf{X})$ have been widely applied in the context of linear scaling electronic structure algorithms for energies and forces[1–4] and for spectral properties like the density of states (DOS), as well as response functions and optical spectra.[5–12] In other areas of computational condensed matter physics and chemistry, one also frequently encounters matrix functions, such as the time-evolution operator $e^{-i\mathbf{H}t}$,[12,13] the partition function $e^{-\beta\mathbf{H}}$,[6,14,15] the Heaviside step function $\Theta(\mu-\mathbf{H})$, and the Green's function $1/(E-\mathbf{H})$.[5] In principle, these matrix functions can be exactly evaluated as

$$f(\mathbf{H})=Uf(\mathbf{h})U^{\dagger} \tag{1}$$

since a Hermitian matrix $\mathbf{H}$ can always be diagonalized by a unitary transformation, $U^{\dagger}\mathbf{H}U=\mathbf{h}$. Diagonalization of a matrix scales as $O(N^3)$, where N is the dimension of the matrix $\mathbf{H}$. It is undesirable to diagonalize the matrix when the system size is big, or in addition, the matrix $\mathbf{H}$ is very sparse and massively parallel computing is required. When a polynomial expansion of the matrix function is employed, one can exploit sparse matrix multiplications to evaluate the products entering the matrix polynomials. The resulting computational effort can, in principle, increase only linearly with the dimension of the matrix. Here we focus on the application of the matrix polynomial approximation in linear scaling electronic structure algorithms.

It is well established that molecular calculations with atom centered basis functions can be carried out in computational times that scale linearly with system size in the large system limit.[16–19] Linear scaling is achieved by using Gaussian orbitals in Kohn–Sham DFT calculations, variants of the fast multipole method for the Coulomb problem,[20–27] linear scaling numerical quadrature evaluation for purely local exchange-correlation potential[28–30] and various diagonalization-free methods to bypass the $O(N^3)$ Hamiltonian diagonalization bottleneck.[1–5,31–45] The physical basis of these linear scaling methods is utilization of spatial locality or the principle of ''the nearsightedness of equilibrium systems,''[46] which states that the properties of a certain observation region comprising one or a few atoms are only weakly influenced by factors that are spatially far away from the observation region.

One class of linear scaling algorithms for constructing the density matrix is based on a direct, polynomial representation of the density matrix $\hat{\rho}$ in terms of the Hamiltonian operator $\hat{\mathbf{H}}$ (or its representation in terms of Wannier functions) via the Heaviside step function or Fermi–Dirac function or the sign matrix function and take advantage of the

a)Author to whom correspondence should be addressed. Electronic mail: mhg@bastille.cchem.berkeley.edu

decay properties of $\hat{\mathbf{H}}$ (and overlap $\mathbf{S}$). To obtain linear scaling one has to cut off exponentially decaying quantities when they are small enough. In the following we review a few examples which employed matrix polynomials or Chebyshev matrix polynomials to obtain ρ.

A polynomial expansion of ρ was employed to derive a Lanczos method within the occupied subspace,[47] whereas Goedecker and Colombo[2] developed a projection method for the computation of the finite temperature total energies and forces with orthonormal local basis sets within the framework of tight-binding theory. The algorithm is based on the fact that the finite temperature density matrix can be written as the Fermi–Dirac function of the Hamiltonian, which can be expanded in terms of Chebyshev matrix polynomials (of the Hamiltonian). This yields a representation of the density as a function of the local Hamiltonian so that it can be evaluated on a computer. It was generalized to nonorthorgonal basis sets,[31] allowing the nonvariational computation of band-structure energies, density matrix, and forces for systems with nonvanishing gaps. Baer and Head-Gordon[48,49] further improved the method for electronic structure calculation on large molecular systems. A related approach has been proposed in which a kernel polynomial method was employed to find the smoothest Chebyshev approximation of the Heaviside step function subjected to a finite number of expansion terms.[1] The density of states (DOS) can also be expanded based on a kernel polynomial method.[1,5,6] The DOS and Fermi projection operator are approximated by an expansion of Chebyshev polynomials. This method proves to be valuable in the case of small energy gaps.

The exponential parameterization of ρ in the atomic orbital (AO) basis[3,4] utilizes unitary transformations of an initial idempotent density matrix and has been employed to optimize the density matrix and thus the energy. This approach had previously been used in the molecular orbital basis. In the AO basis, it has been shown to be an efficient linear scaling algorithm.[43] In this method, a new trial density matrix ρ is parameterized as a unitary transformation of an idempotent matrix from previous steps, i.e., $\rho = e^{-\Delta S}\rho_0 e^{S\Delta} = \Sigma_{n=0}^{\infty}[(-1)^n/n!]\hat{\Delta}^n\rho_0$. Here $\hat{\Delta}\rho_0 = [\Delta, \rho_0]_S$. This unitary transformation can be evaluated via matrix polynomials in Δ. Linear scaling can be achieved because the density matrix and Hamiltonian operator are both sparse in the AO representation.

Purification algorithms[33,37] have excellent performance, compared to other linear scaling methods. Palser and Manolopoulos[33] suggested an algorithm for expanding the single-particle density matrix in terms of the Hamiltonian operator. The method involves the purification of a specialized initial density matrix, which can be done either canonically (at a fixed electron count N) or grand canonically (at a fixed chemical potential μ). Niklasson[34] further developed the algorithm based on higher order purification polynomials with stationary end points in [0, 1], which can reduce the number of matrix multiplications required. A family of purification transforms was proposed by Holas.[50] The nth purification transformation is a polynomial in the initial density matrix.

As the above examples show, in several diagonalization-free methods, the quantities of interest are expanded into matrix polynomials or Chebyshev matrix polynomials, so that only matrix multiplications and additions are required. However, to obtain the density matrix by the above methods, one has to perform many matrix multiplications. In straightforward evaluation of a matrix polynomial of degree N, which may be a few hundreds or even thousands, one has to evaluate $N-1$ matrix multiplications. For example, in the Chebyshev Fermi–Dirac operator expansion (CFOE) method, one has to evaluate a few hundreds or even thousands of matrix multiplications in order to achieve an energy accuracy of about 10^{-5} a.u. at the Hartree–Fock (HF) or density functional theory (DFT) level. Thus, it is well established that the FOE is not an economic method for SCF density matrix evaluation in the traditional quantum chemistry framework compared to the other linear scaling methods. Detailed comparisons of computational times have been given in Refs. 51 and 52 with relatively low accuracy calculations using a semiempirical method for systems with a large highest occupied molecular orbital–lowest occupied molecular orbital (HOMO–LUMO) gaps. By contrast, the FOE method is viewed as effective in linear scaling tight-binding calculations where relatively lower order polynomials (~ 50) are required.[16] It also has the advantage that floating point operations can be cast as matrix–vector rather than matrix–matrix multiplications. This permits one to avoid generating intermediate matrices which are less sparse than the Hamiltonian. It also means one can avoid storing the entire density matrix.

In the context of application to linear scaling DFT or Hartree–Fock calculations, the problem is to find a rapidly convergent expansion or fast resumming methods which can minimize the number of matrix multiplications, since the cost of matrix multiplications completely dominates the cost of matrix additions in evaluating matrix or Chebyshev matrix polynomials. In Ref. 53, we presented fast effective resumming methods for matrix polynomials and Chebyshev matrix polynomials which significantly reduces the number of matrix multiplications. In this paper, we apply these algorithms to accelerate the FOE method. A significant reduction of computational time is obtained by this implementation. By employing the locality of the density matrix and Fock matrix and their matrix products, linear scaling is achieved. The resulting computational time is comparable with other efficient linear scaling methods, such as canonical purification (CP),[33] the curvy-step method,[43] and the sign matrix algorithm[44,45] for *ab initio* SCF calculations.

The paper is arranged as follows: In Sec. II, we briefly describe a fast algorithm[53] to resum matrix polynomials and Chebyshev matrix polynomials. A brief review of the FOE method is given in Sec. III. An optimal projection function is chosen to express the density operator for insulators. Results assessing the choice of representation for the Fermi operator and the performance of the fast summation method are given in Sec. IV. Finally, we conclude in Sec. V.

II. A FAST METHOD FOR RESUMMING MATRIX POLYNOMIALS AND CHEBYSHEV MATRIX POLYNOMIALS

A matrix polynomial of degree N is a function of the form,

$$f(\mathbf{X})=\sum_{i=0}^{N} a_i\mathbf{X}^i. \tag{2}$$

$a_0,a_1,\ldots, a_N$ are the coefficients. Alternatively, any function that is approximated by a power series, over the interval $[-1,1]$ can be approximated by a linear combination of Chebyshev polynomials,

$$f(\mathbf{X})=\sum_{i=0}^{N} c_i\mathbf{T}_i(\mathbf{X}), \tag{3}$$

where $\mathbf{X}$ is a matrix with eigenvalues in the interval $[-1,1]$, and $\mathbf{T}_i(\mathbf{X})$ is the Chebyshev polynomial of degree i. Conventionally, $\mathbf{X}^i$ and $\mathbf{T}_i(\mathbf{X})$ are obtained via the calculated $\mathbf{X}^{i-1}$ and $\mathbf{T}_{i-1}(\mathbf{X})$ and $\mathbf{T}_{i-2}(\mathbf{X})$ matrices, respectively. Thus, a total of $N-1$ matrix multiplications are required. Three fast methods for resumming matrix polynomials and Chebyshev matrix polynomials have been described in detail in Ref. 53. The central strategy of these algorithms is the hierarchical decomposition of matrix polynomials into many blocks and to reuse multiple intermediate quantities to reduce the matrix multiplications. We will employ the $\sqrt{N+1}\times\sqrt{N+1}$ scheme, i.e., algorithm III in Ref. 53 to resum Chebyshev polynomials. This algorithm for resumming matrix polynomials is similar to that first described in Ref. 54. A simple description of the algorithm is given here.

The algorithm requires three steps to be performed. At first we divide the polynomial of degree N into SP subpolynomials ($SP=\mathrm{int}(\sqrt{N+1})+\mathrm{res}$), i.e.,

$$\begin{aligned} f(\mathbf{X}) &= \sum_{m=0}^{N} a_m\mathbf{X}^m \\ &= \sum_{i=0}^{N_1-1} \bar{a}_i^0\mathbf{X}^i+\sum_{n=2}^{SP}\mathbf{X}^{(n-1)\times nl}\sum_{i=1}^{N_n}\bar{a}_i^{n-1}\mathbf{X}^i. \end{aligned} \tag{4}$$

Each subpolynomial except the first and last one includes nl terms ($N_n=nl=\mathrm{int}((N+1)/\sqrt{N+1})$, $1<n<SP$). The first one and last one include $nl+1$ and $N-(SP-1)\times nl$ terms, respectively. Here res$=1$ if $N>nl\times\mathrm{int}(\sqrt{N+1})$, otherwise res$=0$. Then in the second step we calculate and save the necessary $nl-1$ matrices $\mathbf{X}^2,\mathbf{X}^3,\ldots,\mathbf{X}^{nl}$ and write Eq. (4) in a more efficient way as

$$\begin{aligned} f(\mathbf{X})=\sum_{i=0}^{N_1-1}\bar{a}_i^0\mathbf{X}^i+\mathbf{X}^{nl}\Bigg\{\sum_{i=1}^{N_2}\bar{a}_i^1\mathbf{X}^i \\ +\mathbf{X}^{nl}\Bigg\{\sum_{i=1}^{N_3}\bar{a}_i^2\mathbf{X}^i+\cdots+\mathbf{X}^{nl}\Bigg\{\sum_{i=1}^{N_{SP}}\bar{a}_i^{SP-1}\mathbf{X}^i\Bigg\}\Bigg\}\Bigg\} \end{aligned} \tag{5}$$

to reduce the number of matrix multiplications. The third step is to add all terms in Eq. (5) together. The coefficients can be easily obtained from those of a_i as $\bar{a}_i^n=a_{i+n\times nl}$, $\bar{a}_i^n=\bar{a}_i^n$.

To resum a general polynomial of degree N, the algorithm requires a total number of matrix multiplications given by

$$\langle M\rangle=nl+SP-2. \tag{6}$$

For example, to resum a polynomial of degree 25 by this algorithm, we write the polynomial as

$$\begin{aligned} f(\mathbf{X}) &= \sum_{i=0}^{25} a_i\mathbf{X}^i \\ &= \sum_{i=0}^{5}\bar{a}_i^0\mathbf{X}^i+\mathbf{X}^5\sum_{i=1}^{5}\bar{a}_i^1\mathbf{X}^i+\mathbf{X}^{10}\sum_{i=1}^{5}\bar{a}_i^2\mathbf{X}^i+\mathbf{X}^{15}\sum_{i=1}^{5}\bar{a}_i^3\mathbf{X}^i+\mathbf{X}^{20}\sum_{i=1}^{5}\bar{a}_i^4\mathbf{X}^i \\ &= \sum_{i=0}^{5}\bar{a}_i^0\mathbf{X}^i+\mathbf{X}^5\Bigg\{\sum_{i=1}^{5}\bar{a}_i^1\mathbf{X}^i+\mathbf{X}^5\Bigg\{\sum_{i=1}^{5}\bar{a}_i^2\mathbf{X}^i+\mathbf{X}^5\Bigg\{\sum_{i=1}^{5}\bar{a}_i^3\mathbf{X}^i+\mathbf{X}^5\sum_{i=1}^{5}\bar{a}_i^4\mathbf{X}^i\Bigg\}\Bigg\}\Bigg\}. \end{aligned} \tag{7}$$

As Eq. (7) shows, we need to calculate a total number of 8 matrix multiplications, which includes forming 4 matrices X^2, X^3, X^4, X^5 and another 4 matrix multiplications which arise from the subdivision of the polynomial. The total of matrix multiplications required by our algorithm is substantially less than the 24 required by the conventional method. For a matrix polynomial of higher degree N, the number of matrix multiplications required by this algorithm tends to approximately $2\sqrt{N}$.

These algorithms for summing simple matrix polynomials can be easily adapted to accelerate the summation of other matrix series, such as Chebyshev matrix polynomials. Since the eigenvalues of $\mathbf{X}$ lie in the range $[-1,1]$ in Chebyshev polynomials, Chebyshev approximations are numerically very stable and are less susceptible to error accumulation than the equivalent power series in finite precision floating point arithmetic.[55] Thus, in the context of linear scaling electronic structure methods, Chebyshev matrix polynomials have been proposed as a more stable and efficient alternative to simple matrix polynomials.[2,48,49,56] Some differences arise from the recursion relations associated with Chebyshev matrix polynomials. The Chebyshev polynomial

of degree i, $\mathbf{T}_i$, is defined by the following recursion relations:[55]

$$\mathbf{T}_0(\mathbf{X})=\mathbf{I}, \quad \mathbf{T}_1(\mathbf{X})=\mathbf{X}, \quad \mathbf{T}_2(\mathbf{X})=2\mathbf{X}^2-\mathbf{I},$$

$$\mathbf{T}_{n+m}(\mathbf{X})=2\mathbf{T}_n(\mathbf{X})\mathbf{T}_m(\mathbf{X})-\mathbf{T}_{|n-m|}(\mathbf{X}). \tag{8}$$

Thus, the coefficients of the expansion and polynomials will be changed during the division. Equation (4) is changed to

$$\begin{aligned} f(\mathbf{X}) &= \sum_{n=1}^{SP} f_n(\mathbf{X}) \\ &= \sum_{i=0}^{N_1-1} \bar{c}_i^{\theta}\mathbf{T}_i(\mathbf{X}) + \sum_{n=2}^{SP} \mathbf{T}_{(n-1)\times nl}(\mathbf{X}) \sum_{i=1}^{N_n} \bar{c}_i^{n-1}\mathbf{T}_i(\mathbf{X}) \\ &= \sum_{i=0}^{N_1-1} \bar{\bar{c}}_i^{0}\mathbf{T}_i(\mathbf{X}) + \mathbf{T}_{nl}(\mathbf{X})\left\{ \sum_{i=1}^{N_2} \bar{\bar{c}}_i^{1}\mathbf{T}_i(\mathbf{X}) + \cdots \right. \\ &\quad \left. + \mathbf{T}_{nl}(\mathbf{X})\left\{ \sum_{i=1}^{N_{SP}} \bar{\bar{c}}_i^{SP-1}\mathbf{T}_i(\mathbf{X}) \right\} \right\}. \end{aligned} \tag{9}$$

Here N_n is the length of the nth subpolynomial. The coefficients change according to the following recurrences,

$$\bar{c}_i^n=\begin{cases} 2c_{(SP-1)\times nl+i} & n=SP-1 \text{ and } 1\leqslant i\leqslant N_{SP}; \\ 2c_{n\times nl+i}-\bar{c}_{nl-i}^{n+1} & 1\leqslant n<SP-1 \text{ and } 1\leqslant i<nl; \\ c_i-\frac{1}{2}\bar{c}_{nl-i}^{1} & n=0 \text{ and } 0\leqslant i<nl; \end{cases}$$

and $\bar{c}_{nl}^n=2c_{n\times nl+1}-\bar{c}_{nl}^{n+2}$ $(1<n<SP-2)$, $\bar{c}_{nl}^{\theta}=c_{nl}-\frac{1}{2}\bar{c}_{nl}^{2}$, and $\bar{c}_i^{SP-1}=0$ $(i>N_{SP})$. The coefficient $\{\bar{\bar{c}}_i^n\}$ can be calculated based on the relation,

$$\bar{\bar{c}}_i^n=\sum_{m\geqslant n}^{SP-1} \bar{c}_i^m A_{n,m}. \tag{10}$$

Here A is a $SP\times SP$ matrix, which is evaluated according to

$$A_{m,n}=\begin{cases} 2A_{m-1,m-1} & \text{if } m=n; \\ 2A_{m-1,n-1}-A_{m,n-2} & \text{if } m\neq n \text{ and } m\leqslant n-2,\ldots,0 \end{cases}$$

if $n\geqslant 3$, and $A_{0,0}=1$, $A_{1,1}=1$, $A_{2,2}=2$, $A_{0,2}=-1$, and all other elements of the matrix A are equal to 0. The number of matrix multiplications required to resum a Chebyshev polynomial is equal to that for resumming a matrix polynomial of the same degree.

III. CHEBYSHEV FERMI PROJECTION OPERATOR EXPANSION METHOD

We shall explore the use of the fast resummation algorithm described above to accelerate Chebyshev Fermi projection operator expansion (CFOE) method. We give a simple description of the method, including the question of what representation to use for the Fermi operator.

The one electron density operator is defined by

$$\hat{\rho}(\mathbf{r},\mathbf{r}')=\sum_i n_i\Phi_i(\mathbf{r})\Phi_i^*(\mathbf{r}'), \tag{11}$$

where n_i is an occupancy number, equal to 1 or 0. Since $\hat{\rho}$ is a projector onto the space spanned by the N_e (number of electrons) lowest energy orbitals, it is equivalent to a step function,

$$\hat{\rho}=\Theta(\mu-\hat{H}). \tag{12}$$

A Heaviside step function can be defined as any of the following limits:[57]

$$\Theta(\mu-\hat{H})=\tfrac{1}{2}\lim_{\beta\to\infty}\operatorname{erfc}(-\beta(\mu-\hat{H})), \tag{13}$$

$$=\lim_{\beta\to\infty}\frac{1}{1+e^{-\beta(\mu-\hat{H})}}, \tag{14}$$

$$=\lim_{\beta\to\infty}\left[\frac{1}{2}+\frac{1}{\pi}\tan^{-1}(\beta\pi(\mu-\hat{H}))\right], \tag{15}$$

$$=\lim_{\beta\to\infty}e^{-e^{-\beta(\mu-\hat{H})}}, \tag{16}$$

$$=\tfrac{1}{2}\lim_{\beta\to\infty}[1+\tanh(\beta(\mu-\hat{H}))], \tag{17}$$

$$=\lim_{\beta\to\infty}\begin{cases} \frac{1}{2}e^{\beta(\mu-\hat{H})} & \text{for } \hat{H}\geqslant\mu, \\ 1-\frac{1}{2}e^{-\beta(\mu-\hat{H})} & \text{for } \hat{H}\leqslant\mu. \end{cases} \tag{18}$$

Here $\operatorname{erfc}(x)$ is the complementary error function and μ is the chemical potential, which is defined by the number of electrons,

$$\operatorname{Tr}[\hat{\rho}]=N_e. \tag{19}$$

The elements of the density matrix ρ are defined as

$$\rho_{\alpha,\beta}=\langle\phi_\alpha|\hat{\rho}|\phi_\beta\rangle \tag{20}$$

in the basis set $\{\phi_\alpha\}$. Thus, we may express the density matrix at zero temperature as $\rho_{\alpha,\beta}=\langle\phi_\alpha|\Theta(\mu-\hat{H})|\phi_\beta\rangle$. However, for systems with nonvanishing HOMO–LUMO gap, a sufficiently large but finite value of β can be employed. After all, for an insulator, the only requirement on the electronic weight distribution is that it should be one in the valence-band region and zero in the conduction-band region. In the band gap, it can vary smoothly from one to zero.[58] Thus, the zero temperature density matrix may be accurately approximated by any of the following functions with a finite value of β for systems with nonvanishing HOMO–LUMO gaps:

$$\rho\approx\tfrac{1}{2}\operatorname{erfc}(-\beta(\mu-\mathbf{H})) \quad \text{(CEF)}, \tag{21}$$

$$\approx\frac{1}{1+e^{-\beta(\mu-\mathbf{H})}} \quad \text{(FD)}, \tag{22}$$

$$\approx\left[\frac{1}{2}+\frac{1}{\pi}\tan^{-1}(\beta\pi(\mu-\mathbf{H}))\right] \quad \text{(atan)}, \tag{23}$$

$$\approx e^{-e^{-\beta(\mu-\mathbf{H})}} \quad \text{(exp)}, \tag{24}$$

$$\approx\tfrac{1}{2}[1+\tanh(\beta(\mu-\mathbf{H}))] \quad \text{(tanh)}, \tag{25}$$

$$\approx\begin{cases} \frac{1}{2}e^{\beta(\mu-\mathbf{H})} & \text{for } H\geqslant\mu, \\ 1-\frac{1}{2}e^{-\beta(\mu-\mathbf{H})} & \text{for } H\leqslant\mu \quad \text{(two)}. \end{cases} \tag{26}$$

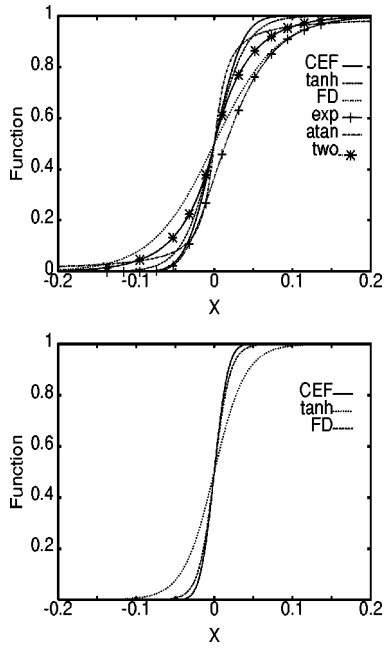

FIG. 1. A plot of the six different finite β approximations to the step function, contained in Eqs. (21)–(26). The functions are each plotted for a fixed β value of 25.0 (upper panel) and 50.0 (lower panel). The function which approaches 0 and 1 fastest will provide the most accurate representation for a given value of β.

We plot these functions with $\beta=25$ and 50 in Fig. 1. All functions have values in the range [0, 1] and approach 0 and 1 smoothly near μ. All functions except the expression of $e^{-e^{-\beta(\mu-\mathbf{H})}}$ are antisymmetric. The complementary error function approaches zero and one faster than other functions for a given β value. From Fig. 1, we also observe that as we increase of the value of β, the functions approach 0 and 1 faster.

We employ the FOE method to obtain the one-electron reduced density matrix at the *ab initio* HF/DFT levels, using Gaussian nonorthogonal atom-centered basis sets.[59] The FOE method has been generalized to nonorthogonal basis sets,[31,60] such that the overlap matrix S is involved in the expression of ρ. Alternatively we may keep the expression for ρ unchanged by transforming the AO Fock matrix $\mathbf{F_{AO}}$ and the AO density matrix $\boldsymbol{\rho}_{\mathbf{AO}}$ to an orthonormal basis using $\mathbf{H}=\mathbf{L}^{-1}\mathbf{H_{AO}}(\mathbf{L}^{-1})^T$ and $\rho=\mathbf{L}^T\rho_{\mathbf{AO}}\mathbf{L}$, where $\mathbf{L}\mathbf{L}^T=\mathbf{S}$, the overlap matrix. Usually, the transformation is obtained by Cholesky decomposition.[55,61,62]

The Fermi–Dirac function [Eq. (22)] has been used in Refs. 2, 47–49, 51, 52, and 61 to approximate the electronic weight distribution of metals at finite temperature or insulators at zero temperature. The density matrix ρ is approximated by a Fermi–Dirac function with a finite value of β, which is expanded as a finite order Chebyshev polynomial in the effective Hamiltonian matrix (Fock matrix),

$$\rho=\Sigma_{i=0}^{N}a_i(\beta_s,\mu_s)\mathbf{T}_i(\mathbf{H}_s). \tag{27}$$

Here $\mu_s=(\mu-\overline{E})/\Delta E$, $\beta_s=\beta\Delta E$, N is the degree of the polynomial and a_i are the expansion coefficients, which can be obtained by using the fast Fourier transform.[48] $\mathbf{H}_s$ is the scaled and shifted $\mathbf{H}$ matrix, defined so that its eigenvalues lie in the interval $[-1,1]$. This scale and shift is performed because Chebyshev polynomials are defined within this interval. If $\Delta E=E_{\max}-E_{\min}$ is the eigenvalue spread, and $\overline{E}=(E_{\max}+E_{\min})/2$ is the average of $E_{\max}$ and $E_{\min}$, then $\mathbf{H}_s$ is given by

$$\mathbf{H}_s=\frac{\mathbf{H}-\overline{E}}{\Delta E}. \tag{28}$$

$E_{\min}$ and $E_{\max}$ are the minimum and maximum eigenvalues of the Hamiltonian (Fock) matrix $\mathbf{H}$.

Actually all functions in Eqs. (21)–(26) can be expanded into finite order Chebyshev polynomials. Generally for a given value of β, functions which approach one and zero more slowly will require larger values of β to satisfy a desired accuracy for the electronic weight distribution. Thus they may need more polynomial terms to converge the expansion series after β is chosen to obtain a desired accuracy in the density matrix. For a metal, $\beta=1/kT$, and thus β is connected to a real temperature, while in insulators, $\beta\propto 1/\delta e$ and thus depends on the HOMO–LUMO gap, $\delta e=E_{\mathrm{LUMO}}-E_{\mathrm{HOMO}}$, of the system.[48] The value of β controls the proximity of the projection function to the true density matrix. At a specified precision 10^{-D}, β can be choosen according to the following equations:

$$F_{\mu,\beta_1}(E_{\mathrm{HOMO}})\geqslant 1-10^{-D}$$

and

$$F_{\mu,\beta_2}(E_{\mathrm{LUMO}})\leqslant 10^{-D}, \tag{29}$$

β takes the larger value of β_1,β_2. By choosing the smallest allowed value of β this way, it is still large enough to satisfy the specified precision of ρ. The values of β are different for different projection functions $F_{\mu,\beta}(F)$. For example, $\beta\simeq 2D\log_e^{10}/\delta\epsilon$, $\sqrt{2D\log_e^{10}}/\delta\epsilon$, and $D\log_e^{10}/\delta\epsilon$ for FD [Eq. (22)], CEF [Eq. (21)], and tanh [Eq. (25)] functions, respectively. In Sec. IV, we compare the properties of these functions and present the degree of polynomials needed for the different functions in order to satisfy the energy precision 10^{-5}. We will find that the length of the polynomial corresponding to the complementary error function is shorter than the others.

μ should be determined by enforcing the electron count of the system [see Eq. (19)]. The accuracy of μ controls the accuracy of the density matrix. Since T_n does not depend on μ, one may perform a calculation of several density matrices which correspond to different trial chemical potentials as Refs. 48, 51, and 52 have suggested. But it may not be best to save a lot of matrices. Alternatively, for insulators,

$$\mu=\frac{E_{\mathrm{HOMO}}+E_{\mathrm{LUMO}}}{2} \tag{30}$$

should be a good choice (in fact optimal for the antisymmetric functions). For metals, one may employ the finite difference approximation of Eq. (19) to find the correction $\Delta\mu$,

$$\Delta\mu=\frac{\Delta N_e}{\mathrm{Tr}\left(\Sigma_{n=0}^{N-1}\dfrac{\partial a_n}{\partial\mu}T_n(\mathbf{X})\right)}. \tag{31}$$

Then $\mu+\Delta\mu$ is employed for density matrix evaluation.[16] In this work, we adopt Eq. (30) for μ.

The parameters β_s, μ_s, and $\mathbf{H}_s$ depend on eigenvalues $E_{\min}$, $E_{\max}$, E_{HOMO}, and E_{LUMO} of $\mathbf{H}$. One has to calculate these quantities. In this work, these parameters are calculated by a linear-scaling sparse real symmetric matrix Lanczos algorithm.[63,64] The Lanczos algorithm (LA) is used to repeatedly improve an approximate solution until it reaches sufficient accuracy. The conventional LA is good enough for eigenvalues on the extreme edges of the spectrum but frequently is not very good for interior eigenvalues. This latter situation can be changed completely when the iteration is driven not by $\mathbf{H}$ itself but with a spectral filter, a specially designed function of $\mathbf{H}$, denoted $f(\mathbf{H})$. For example, the method termed *shift and invert*, uses the filter $f(\mathbf{H})=1/(E-\mathbf{H})$.[64] This method has the highly desirable effect of throwing the eigenvalues of $\mathbf{H}$ that lies near E to the extreme edges of the spectrum. In order to generate the next Lanczos vector Q_{j+1}, one has to calculate the vector-matrix multiplication $f(\mathbf{H})Q_j=V_j$. We follow the shift-and-invert strategy of Ericsson,[64] to solve the equation $(\mathbf{H}-E)V_j=Q_j$ for V_j instead of calculating $1/(E-\mathbf{H})$ explicitly. Obtaining these eigenvalues involves only matrix–vector products rather than matrix multiplications, so the cost of these steps is small compared to the subsequent solution for the density matrix and matrix diagonalization as Table II shows.

Once β and μ are computed, we then choose a degree of polynomial such that the sufficient precision is obtained. This degree, N, is obtained by stopping evaluation of the series when the Nth terms satisfy a stopping criterion, $|a_{i-1}|>|a_i|$ and $|a_i|<T_2$. T_2 should be two or three powers of ten smaller than 10^{-D}. For example, if one hopes to have precision 10^{-D} for the density matrix, then T_2 should be set to 10^{-D-2}.

IV. RESULTS

When the one-electron density matrix is expanded into Chebyshev polynomials of the Fock matrix, it can be easily obtained by sparse matrix multiplication and addition, since the density matrix and the Fock matrix in a Gaussian basis set have finite decay ranges. A blocked sparse matrix multiplication scheme is employed for our matrix multiplications.[65] In this scheme large nonzero submatrices are obtained by forming many-atom blocks. These blocks are obtained by a boxing scheme, where the system is spatially partitioned into many boxes with each box containing many atoms. This division of the system is the same as is used in some tree code methods.[26] While the fraction of negligible submatrices is lower than the actual elemental sparsity, the blocking scheme benefits from the use of highly-optimized level-3 basic linear algebra subroutines (BLAS) for large submatrix multiplications. These large-block multiplications also ultimately result in the reduction in CPU time for sparse matrix multiplications. When the density matrix is expressed as a Chebyshev polynomial of the Fock matrix, the $T_n(\mathbf{H})$ matrices are symmetric. Thus, we can employ symmetric sparse matrix multiplications and additions, which can reduce CPU time by roughly 30% compared to unsymmetric matrices.

TABLE I. The average values of β and degree of polynomials (N) and the number of matrix multiplications $\langle M\rangle$ (which does not include basis transformation and purification) per SCF cycles for linear alkane $C_{60}H_{122}$ at BLYP/STO-3G and BLYP/6-31G**. The density matrices are expanded into Chebyshev polynomials based on the complementary error function, Fermi–Dirac function, etc. The deviation of energy is obtained by comparing with the energy calculated by direct diagonalization.

Basis	Function	$\langle M\rangle$	N	β	Error
STO-3G	CEF	21	129	9.68	$1.2e^{-6}$
	tanh	31	269	21.33	$1.1e^{-6}$
	FD	31	269	42.69	$1.1e^{-6}$
	exp	39	412	42.69	$2.8e^{-6}$
	two	49	695	42.69	$2.3e^{-6}$
	atan	>62	>1024		
6-31G**	CEF	42	448	19.33	$5.5e^{-6}$
	FD	60	956	95.61	$9.3e^{-5}$

We have implemented the fast methods for resumming matrix and Chebyshev matrix polynomials into a development version of Q-CHEM.[66] In the following we present timings for our new algorithm to explore its performance relative to conventional diagonalization methods. All calculations are for the problem of evaluating the density matrices with a converged accuracy of 10^{-5}. All other parts of the calculations, such as Fock matrix building, are excluded from the timings. The calculations are for two classes of model systems: one-dimensional linear alkanes and two-dimensional water clusters. Two different basis sets (STO-3G and 6-31G**) are used to examine basis set effects on the performance of the algorithm. All timings were obtained using a development version of the Q-CHEM program package[66] on a 375 MHZ IBM Power-3 workstation (Model 270).

Here we are interested in evaluating the density matrix at zero temperature and for insulators. Thus, we can expand the density matrix using any of a series of functions [see Eqs. (21)–(26)] with finite values of β, since all these functions satisfy the requirements of the zero temperature electronic weight distribution for insulators. Table I shows the β values and degree of polynomial corresponding to different functions. The degree of polynomial for the complementary error function is shorter than that corresponding to other functions. In Fig. 2, we plot four different finite $\beta_s=\beta\Delta E$ approxima-

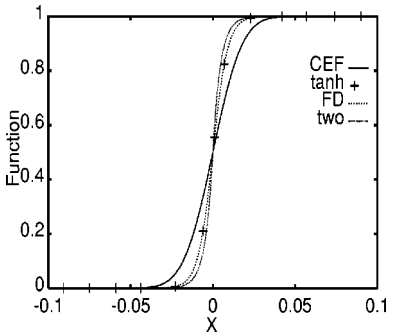

FIG. 2. A plot comparing four different finite β approximation to the step function, corresponding to the STO-3G data given in Table I for precision 10^{-5}. Each function type (CEF, tanh, FD, and two) has its own β value given in Table I. The greater smoothness of the CEF representation permits use of a polynomial representation of lower degree, as shown in Table I.

TABLE II. The average CPU time for each SCF cycle in obtaining the density matrix for a series of linear alkanes at BLYP/STO-3G and BLYP/6-31G** level. The geometry is ideal with the C–C bond length of 1.54 Å, C–H bond length of 1.10 Å, and C–C–C bond angle of 109.5°. The CPU time recorded is only for forming the density matrix. For the FOE method, the density matrix is expanded based on the CEF. The CPU time in the canonical purification (CP) method and the fixed trace sign matrix search (FTSMS) method are also shown for comparision. LA is the CPU times spent on each SCF cycles for the calculation of $E_{\max}$, $E_{\min}$, and HOMO–LUMO gap. The sparse block size included 10 carbon atoms in the STO-3G basis and 4 in 6-31G**. $\langle M \rangle$ is the total number of matrix multiplications needed per SCF cycle.

		$\langle M \rangle$			FOE (s)				
Molecule	Basis	FOE	CP	FTSMS	LA(s)	Total(s)	CP (s)	FTSMS (s)	Diag (s)
$C_{60}H_{122}$	STO-3G	27	33	33	0.25	1.27	1.18	1.21	1.35
$C_{120}H_{242}$	STO-3G	27	33	33	0.92	3.45	3.17	3.39	11.03
$C_{180}H_{362}$	STO-3G	27	33	33	1.51	5.96	5.34	5.46	40.12
$C_{240}H_{482}$	STO-3G	27	33	33	2.20	8.87	7.95	8.28	91.73
$C_{60}H_{122}$	6-31G**	47	65	65	2.85	110.43	89.20	101.51	73.70
$C_{120}H_{242}$	6-31G**	45	65	65	6.90	328.59	222.61	282.83	574.30

tions to the step function, corresponding to the STO-3G data given in Table I for precision 10^{-5}. CEF, tahn, FD and two all have their corresponding β value given in Table I. The greater smoothness of the CEF representation permits use of a polynomial representation of lower degree, as shown in Table I. The Fermi–Dirac function approaches one and zero more slowly than the CEF at same value of β. Thus, it requires a larger value of β to satisfy the target precision of the electronic weight distribution. More polynomial terms are required to converge the series expansion based on the Fermi–Dirac function, where the series in Eq. (27) is terminated with the ith term if $|a_{i-1}|>|a_i|$ and $|a_i|<T_2$.

The average degree of polynomial N and the average number of matrix multiplications $\langle M \rangle$ per SCF cycle obtained from a converged density matrix are also shown in Table I. $\langle M \rangle$ is obtained by averaging the number of matrix multiplications in each SCF cycle. We note the number of matrix multiplications, $\sim 2\sqrt{N}$, is far less than the length of polynomial N. A large increase in the polynomial length is noted when the basis set changes from STO-3G to 6-31G** for linear alkanes, since the Chebyshev polynomial length for FOE is proportional to $\Delta E/\delta\epsilon$.[48] As the basis set is extended, ΔE increases and $\delta\epsilon$ may also decrease so that N thus increases. For the target precision 10^{-D}, we use $D=\min(\max(3,n+1),5)$ on the nth SCF cycle. Then the threshhold T_2, which is employed to terminate the series, is set to $T_2=\max(10^{-7},10^{-D-3})$.

We use the CEF to approximate the density matrix in the following calculations since it requires the shortest polynomial length. Table II for the linear alkanes shows computational time versus the number of basis functions for the new algorithm and for the conventional diagonalization algorithm. These calculations are performed at the BLYP/STO-3G and BLYP/6-31G** levels. Comparing against the conventional diagonalization calculations, significant computational savings are observed with the new algorithm for large molecules. Direct evaluation of the matrix function requires all the eigenvalues and eigenvectors of the matrix **X**. To obtain all of them through a full diagonalization of the matrix requires a calculation of $O(N^3)$ complexity as Table II shows. Our algorithm can clearly avoid the bottleneck if **X** is sparse. A crossover is noted for systems whose number of carbons are about 60 and 120 for STO-3G and 6-31G**, respectively. The computational time asymptotically scales linearly with molecular size. The SCF cycles required in FOE method are exactly equal to that needed by the conventional diagonalization calculations. The CPU time in each SCF iteration is comparable with the canonical purification (CP) method and the fixed trace sign matrix search (FTSMS) approach.[45] (Here we do not employ the damping technique suggested in Ref. 45.) The total number of matrix multiplications required by CP and the fixed trace sign matrix search approach are the same. FOE requires fewer matrix multiplications than CP and FTSMS approaches. However, the FOE method requires many matrix extra additions as a result of the fast resummation method. About 1/4 of the CPU time for the resummation of the matrix polynomial is used for matrix additions when the matrices are very sparse (for example, the

TABLE III. The average CPU time for each SCF cycle in obtaining the density matrix is shown for water clusters at the HF/STO-3G and HF/6-31G** levels. The CPU time recorded is only for forming the density matrix. The density matrix is expanded based on the CEF. The sparse block matrix size included eight water molecules at the HF/STO-3G level and four water molecules at the HF/6-31G** level.

		$\langle M \rangle$		CPU time(s)			SCF cycle		
Molecule	Basis	CEF	CP	CEF	Diag	CP	CEF	Diag	CP
3×3	STO-3G	27	42	3.48	2.32	4.39	7	7	7
4×4	STO-3G	27	43	10.94	13.56	13.46	6	6	6
5×5	STO-3G	26	43	21.81	52.71	30.58	6	6	6
6×6	STO-3G	27	43	41.95	156.53	58.51	6	6	6
3×3	6-31G**	43	49	203.26	111.23	169.01	7	7	7
4×4	6-31G**	43	49	684.12	801.51	445.91	7	7	7

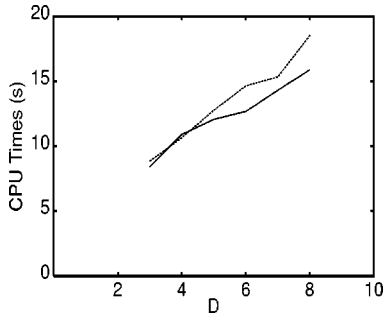

FIG. 3. The average CPU time for each SCF cycle in obtaining the density matrix is shown for a 4×4 water cluster at the HF/STO-3G level for different target precisions of 10^{-D}. The CPU time recorded is only for forming the density matrix. The density matrix is expanded based on the CEF. The sparse block matrix size included eight water molecules. The solid line is for the FOE method and the dashed line is for the CP method.

matrices of large alkanes at BLYP/STO-3G). The CPU time for matrix additions is trivial compared to that for matrix multiplications when matrices are dense (for example, the matrices of large alkanes at BLYP/6-31G** and water clusters at HF/6-31G**). In evaluating matrix multiplications and additions, we employ the symmetric properties of matrices and their matrix products in the FOE, CP, and the fixed trace sign matrix methods. Thus the CPU time of the CP method is different than Ref. 43. In the above calculations, we employed the cutoff lengths $L=19.56$ Å at BLYP/STO-3G and $L=55.89$ Å at BLYP/6-31G** levels, respectively. The cutoff length $L=19.56$ Å is enough for linear alkanes at BLYP/STO-3G levels since as we increase it from $L=19.56$ Å to $L=58.69$ Å, the deviation of energy remains unchanged in the FOE method as well as the CP method.

Next we check the computational time for a two-dimensional water cluster built from a unit cell containing eight water molecules. The average O–O separation is about 2.8 Å. The computational results are shown in Table III. The calculations are performed at the HF/STO-3G and HF/6-31G** levels. It is noted that a crossover of CPU time occurs around 4×4 water clusters with both basis sets. A significant CPU time saving is reached by our novel algorithm even for two-dimensional water clusters.

In the calculations of Tables I, II, and III, a target precision of 10^{-5} is employed. Next we vary the target precision of 10^{-D} to check the CPU time versus D for a two-dimensional 4×4 water cluster at the HF/STO-3G level, since the cutoff length and the termination criteria of the series depend on D. A plot is shown in Fig. 3. It is noted that the CPU times in FOE are less than CP. The difference between the CPU times in FOE and CP slightly increases as the target precision is increased. It shows that the FOE method is slightly better than the CP method with a small basis set when higher target precision is needed for the evaluation of the density matrix.

V. CONCLUSIONS

(1) Use of the fast summation methods for Chebyshev matrix polynomial of degree N reduces the number of matrix multiplications from $N-1$ to roughly $2\sqrt{N}$. This makes the Chebyshev polynomial expansion approach to linear scaling electronic structure calculations competitive with the best alternatives.

(2) Various representations for the smoothed step function have been investigated, and we find that using the complementary error function yields a significant improvement over the usual Fermi–Dirac function.

ACKNOWLEDGMENTS

The authors would like to acknowledge valuable discussions with Vladimir Mandelshtam. W.Z.L. particularly expresses her deep gratitude to Professor Arup K. Chakraborty and Mr. Baron Peters for many stimulating discussions relevant to this work. Financial support from BP (W.Z.L.) and the Israel–U.S. Binational Science Foundation (R.B. and M.H.G.) as well as partial support (M.H.G.) from the National Science Foundation (CHE-9981997) are gratefully acknowledged.

[1] R. N. Silver, H. Röder, A. F. Voter, and J. D. Kress, J. Comput. Phys. **124**, 115 (1996).
[2] S. Goedecker and L. Colombo, Phys. Rev. Lett. **73**, 122 (1994).
[3] T. Helgaker, P. Jørgensen, and J. Olsen, *Molecular Electronic-Structure Theory* (Wiley, Chichester, 2000).
[4] T. Helgaker, H. Larsen, J. Olsen, and P. Jørgensen, Chem. Phys. Lett. **327**, 397 (2000).
[5] A. F. Voter, J. D. Kress, and R. N. Silver, Phys. Rev. B **53**, 12733 (1996).
[6] R. N. Silver and H. Röder, Int. J. Mod. Phys. C **5**, 735 (1994).
[7] N. Furukawa, Y. Motome, and H. Nakata, Comput. Phys. Commun. **142**, 410 (2001).
[8] L. W. Wang, Phys. Rev. B **49**, 10154 (1994).
[9] D. Neuhauser, J. Chem. Phys. **93**, 2611 (1990).
[10] M. R. Wall and D. Neuhauser, J. Chem. Phys. **102**, 8011 (1995).
[11] G. A. Parker, W. Zhu, Y. Huang, D. K. Hoffman, and D. J. Kouri, Comput. Phys. Commun. **96**, 27 (1996).
[12] H. Tal-ezer and R. Kosloff, J. Chem. Phys. **81**, 3967 (1984).
[13] T. Iitaka, S. Nomura, H. Hirayama, X. Zhao, Y. Aoyagi, and T. Sugano, Phys. Rev. E **56**, 1222 (1997).
[14] R. Kosloff and H. Tal-ezer, Chem. Phys. Lett. **127**, 233 (1986).
[15] Y. Motome and N. Furukawa, Phys. Soc. Jpn. **68**, 3853 (1999).
[16] S. Goedecker, Rev. Mod. Phys. **71**, 1085 (1999).
[17] G. E. Scuseria, J. Phys. Chem. A **103**, 4782 (1999).
[18] W. Yang and J. M. Pérez-Jordá, in *Encyclopedia of Computational Chemistry*, edited by P. v. R. Schleyer (Wiley, New York, 1998), pp. 1496–1513.
[19] P. Ordejón, Comput. Mater. Sci. **12**, 157 (1998).
[20] L. Greengard and V. Rokhlin, J. Comput. Phys. **73**, 325 (1987).
[21] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems* (MIT Press, Cambridge, 1988).
[22] L. Greengard, Science **265**, 909 (1994).
[23] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chem. Phys. Lett. **230**, 8 (1994).
[24] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chem. Phys. Lett. **253**, 268 (1996).
[25] C. A. White and M. Head-Gordon, J. Chem. Phys. **105**, 5061 (1996).
[26] M. C. Strain, G. E. Scuseria, and M. J. Frisch, Science **271**, 51 (1996).
[27] W. Z. Liang, S. Yokojima, D. H. Zhou, and G. H. Chen, J. Phys. Chem. A **104**, 2445 (2000).
[28] B. G. Johnson, C. A. White, Q. Zhang, B. Chen, R. L. Graham, P. M. W. Gill, and M. Head-Gordon, in *Recent Developments in Density Functional Theory*, edited by J. M. Seminario (Elsevier Science, Amsterdam, 1996), Vol. 4, p. 441.
[29] R. E. Stratmann, G. E. Scuseria, and M. J. Frisch, Chem. Phys. Lett. **257**, 213 (1996).
[30] J. M. Pérez-Jordá and W. Yang, Chem. Phys. Lett. **241**, 469 (1995).
[31] U. Stephan and D. A. Drabold, Phys. Rev. B **57**, 6391 (1998).
[32] W. Yang, Phys. Rev. Lett. **66**, 1438 (1991); W. Yang and T. S. Lee, J. Chem. Phys. **103**, 5674 (1995); W. Yang, Phys. Rev. B **56**, 9294 (1997).
[33] A. H. R. Palser and D. Manolopoulos, Phys. Rev. B **58**, 12704 (1998).

[34] A. M. Niklasson, Phys. Rev. B **66**, 155115 (2002).
[35] F. Mauri, G. Galli, and R. Car, Phys. Rev. B **47**, 9973 (1993).
[36] P. Ordejón, D. A. Drabold, M. P. Grumbach, and R. M. Martin, Phys. Rev. B **48**, 14646 (1993).
[37] X. P. Li, R. W. Nunes, and D. Vanderbilt, Phys. Rev. B **47**, 10891 (1993).
[38] C. H. Xu and G. E. Scuseria, Chem. Phys. Lett. **262**, 219 (1996).
[39] A. D. Daniels, J. M. Millam, and G. E. Scuseria, J. Chem. Phys. **107**, 425 (1997).
[40] C. Ochsenfeld and M. Head-Gordon, Chem. Phys. Lett. **270**, 399 (1997).
[41] S. Yokojima, D. H. Zhou, and G. H. Chen, Chem. Phys. Lett. **302**, 495 (1999).
[42] M. Head-Gordon, Y. Shao, C. Saravanan, and C. A. White, Mol. Phys. **101**, 37 (2003).
[43] Y. Shao, C. Saravanan, M. Head-Gordon, and C. A. White, J. Chem. Phys. **118**, 6144 (2003).
[44] G. Beylkin, N. Coult, and M. Mohlenkamp, J. Comput. Phys. **152**, 32 (1999).
[45] K. Németh and G. E. Scuseria, J. Chem. Phys. **113**, 6035 (2000).
[46] W. Kohn, Chem. Phys. Lett. **208**, 167 (1993); Phys. Rev. Lett. **76**, 3168 (1996).
[47] O. F. Sankey, D. A. Drabold, and A. Gibson, Phys. Rev. B **50**, 1376 (1994).
[48] R. Baer and M. Head-Gordon, J. Chem. Phys. **107**, 10003 (1997).
[49] R. Baer and M. Head-Gordon, Phys. Rev. Lett. **79**, 3962 (1997).
[50] A. Holas, Chem. Phys. Lett. **340**, 552 (2001).
[51] K. R. Bates, A. D. Daniels, and G. E. Scuseria, J. Chem. Phys. **109**, 3308 (1998).
[52] A. D. Daniels and G. E. Scuseria, J. Chem. Phys. **110**, 1321 (1999).
[53] W. Z. Liang, R. Baer, C. Saravanan, Y. Shao, A. T. Bell, and M. Head-Gordon, J. Comput. Phys. (to be published).
[54] M. S. Paterson and L. J. Stockmeyer, SIAM J. Comp. **2**, 60 (1973).
[55] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortrain 77* (Cambridge University Press, New York, 1996).
[56] R. Kosloff, Annu. Rev. Phys. Chem. **45**, 145 (1994).
[57] (a) http://mathworld.wolfram.com/HeavisideStepFunction.html; (b) R. Bracewell, ''Heaviside's unit step function, $H(x)$,'' *The Fourier Transformation and Its Applications*, 3rd ed. (McGraw-Hill, New York, 1999), pp. 57–61; (c) J. Spanier and K. B. Oldham, in *An Atlas of Functions* (Hemisphere, Washington, D.C., 1987), Chap. 8, pp. 63–69.
[58] S. Goedecker and M. Teter, Phys. Rev. B **51**, 9455 (1995).
[59] E. R. Davidson and D. Feller, Chem. Rev. **86**, 681 (1986).
[60] S. Goedecker, J. Comput. Phys. **118**, 261 (1995).
[61] J. M. Millam and G. E. Scuseria, J. Chem. Phys. **106**, 5569 (1997).
[62] M. Challacombe, J. Chem. Phys. **110**, 2332 (1999).
[63] J. K. Cullum and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations Vol. II Programs*, 1985.
[64] T. Ericsson and A. Ruhe, Math. Comput. **35**, 1251 (1980).
[65] C. Saravanan, Y. Shao, R. Baer, P. N. Ross, and M. Head-Gordon, J. Comput. Chem. **24**, 618 (2003).
[66] J. Kong, C. A. White, A. I. Krylov *et al.*, J. Comput. Chem. **21**, 1532 (2000).