Journal of Computational Physics **157,** 404–416 (2000)
doi:10.1006/jcph.1999.6388, available online at http://www.idealibrary.com on IDEAL®

Exploring Sparsity in Three-Dimensional Integration for Density-Functional Calculations

L. Lou

Wavefunction, Inc., 18401 von Karman, Suite 370, Irvine, California 92612
E-mail: liang@wavefun.com

Received March 23, 1999; revised July 27, 1999

The cost of a density-functional calculation with three-dimensional integration remains to be order N^3, although a large portion of the integration grid may have negligible effects on the generation of a matrix element, due to rapid decay of atom-centered basis functions with distance. This type of integration sparsity is exploited by prescreening for insignificant contributions based on a direct estimate of their magnitudes. The result is a substantial reduction in cost without sacrificing numerical precision. Timing on compact molecules shows that a near order N^2 scaling with system size can be obtained for $N \gtrsim 300$ basis functions. The overhead of prescreening is moderate and may be characterized by a break-even size $N \approx 200$.

Key Words: prescreening algorithms; fast three-dimensional integration; numerical basis functions; local density approximations; atomic clusters.

1. INTRODUCTION

The solutions of a polyatom in the local density approximations (LDA) of the density-functional theory [1] can be obtained by numerically integrating the one-particle equations over a three-dimensional grid [2–7]. The difficulties caused by nuclear cusps are overcome by properly partitioning the space around each nucleus, e.g., using primitive wedges to fill the atomic spheres and interstitial regions [8–11], or sampling the space multiple times with a set of overlapping spheres and a weight-partitioning function [5, 12]. As the components of the equations are discretized on the grid [2], identical arithmetic operations will result at each grid point, which can be performed with algorithms of extreme simplicity. For example, a vectorized algorithm [13] performing quadratures to build a matrix element may contain just a few dozens of lines of code and will be rather straightforward to upgrade to parallel platforms [14, 15] as the discretization has provided a natural data partition for distributions. In addition to the arithmetic simplicity, performing three-dimensional

0021-9991/00 $35.00

integrations removes the restrictions on the form of atomic orbital basis functions making it possible to employ orbital types which are more saturated but infeasible for analytical treatments. Examples of such orbital types include the Slater type orbitals [3, 16, 17] and numerical atomic orbitals [4, 5, 18, 19] (as exact LDA solutions of atoms and ions). In these orbital forms, certain physical properties of an atomic wave function, such as the nuclear cusps, an exponentially decaying tail, and nodal positions resembling atomic shell structure, would be preserved. Due to the lack of proper analytical expression or efficient algorithms for analytical integration, however, these function forms have to be expressed and processed numerically with three-dimensional integration.

Although solving a polyatomic problem by performing numerical integration has significant advantages, the cost to solution remains order N^3 in scaling with problem size, compared to a performance between N^2 and N^3 of methods [20–23] based on analytical integration with prescreening. The purpose of this work is to show that the performance of the three-dimensional integration for the same polyatomic problems can be brought in line with the best analytical methods as is measured by the scaling power in cost growth with problem size. Recently, several algorithms have been developed with asymptotically order N performance [24–29]. However, the range of their applications is rather limited. Among the main restrictions are large system size, low system dimensionality, and spatially tight basis functions [30]. For problems outside this domain, performance can decrease drastically, particularly when accuracy is in demand. The constraints on the present search for replacement algorithm are that it should maintain a low error bound while providing substantial speedup over the existing high performance order N^3 algorithms [13] for practical use.

There are two bottlenecks in the three-dimensional integration solutions for polyatomic problems within the LDA framework. One is the assembly of matrix elements for transforming from a grid representation to an atomic orbital representation, and the other is the mapping of electron density onto a real space grid from the orbital space through a matrix-to-scalar reduction at every grid point. The two procedures both involve adding up numerous tiny values whose magnitudes are largely determined by distance to the centers contributing atomic orbital basis functions. Due to the fast decay of the tail of these functions, the integration is sparse. Exploiting this sparsity can lead to a substantial reduction in cost for the problems to be solved.

In this paper, a simple algorithm to perform the required prescreening is presented. The algorithm scales as order N^2 over a wide size range starting from about 300 basis functions and breaks even with a fast reference order N^3 algorithm at 200 basis functions showing a rather moderate overhead. The errors caused by truncating small values are comparable or below the intrinsic noise of the numerical grid.

2. RATE-LIMITING PROCEDURES

Using an atomic orbital basis for the expansion of one-electron wave functions

$$\psi_i(\mathbf{r}) = \sum_\mu c_{i\mu}\chi_\mu(\mathbf{r}), \tag{1}$$

the Kohn–Sham one-electron equations [1]

$$H\psi_i(\mathbf{r}) = \epsilon_i\psi_i(\mathbf{r}) \tag{2}$$

can be rewritten in terms of a set of linear equations

$$\sum_i c_{i\nu}[H_{\mu\nu} - \epsilon_i S_{\mu\nu}] = 0, \tag{3}$$

where ϵ_i are the one-electron energies, S the overlap matrix, and H the Hamiltonian matrix,

$$S_{\mu\nu} = \int d^3r\, \chi_\mu(\mathbf{r})\chi_\nu(\mathbf{r}), \tag{4}$$

$$H_{\mu\nu} = \int d^3r\, \chi_\mu(\mathbf{r})H(\mathbf{r})\chi_\nu(\mathbf{r}). \tag{5}$$

H may be further divided according to dependence on electron density ρ

$$H = T + V[\rho], \tag{6}$$

where T is the kinetic energy and V the effective potential consisting of an electrostatic part V_s and an exchange-correlation part μ_{xc}. For basis functions χ of a general type, all matrix elements must be evaluated via numerical integrations on a three-dimensional grid [2],

$$\Lambda_{\mu\nu} \approx \sum_k \omega_k \chi_\mu(\mathbf{r}_k)\Lambda(\mathbf{r}_k)\chi_\nu(\mathbf{r}_k), \tag{7}$$

where the quadrature weights ω, basis functions χ, and operator Λ (which equals $T \equiv -\frac{1}{2}\nabla^2$ and V for H, or equals 1 for S) are defined on every grid point $\mathbf{r}_k$ [2]. When all the ingredients are available, the assembly of the matrix Λ requires order $N_p N^2$ operations, where N_p is the number of grid points and N the number of basis functions. The actual costs are quite different for the matrices T, S, and V within a complete self-consistent calculation. For S and T, it needs to be done only once at the beginning of the iterations, whereas for V, the computations are repeated each iteration with updated ρ. The cost for generating a basis function at a grid point is nearly a constant, which gives a total cost for all χ on the entire grid in the order of $N_p N$. The calculation of the weights requires negligible time.

In general, N_p is not a linear function of N but dependent on complex variables such as stoichiometry and geometry. However, with the restrictions that the systems are built with repetitive units, the relation can become linear and the cost of a matrix assembly becomes $\propto N^3$.

The density ρ is constructed by reducing a matrix to a scalar at each grid point and is an order N^3 procedure,

$$\rho(\mathbf{r}_k) = 2\sum_i |\psi_i(\mathbf{r}_k)|^2 = 2\sum_i \left|\sum_\mu c_{i\mu}\chi_\mu(\mathbf{r}_k)\right|^2. \tag{8}$$

This expression has an operation count of $\sim 2N_p N_{occ} N$, where N_{occ} is the number of occupied one-electron states and is usually several times smaller than N. However, since it contains coefficients c in nonlinear forms, Eq. (8) cannot be used efficiently for prescreening. The preferred linear dependence of c can be obtained by rearranging the terms in Eq. (8),

$$\rho(\mathbf{r}_k) = \sum_{\mu,\nu} P_{\mu\nu}\chi_\mu(\mathbf{r}_k)\chi_\nu(\mathbf{r}_k), \tag{9}$$

where $P_{\mu\nu}$ is an element of the density matrix

$$P_{\mu\nu} = 2\sum_i c_{i\mu} c_{i\nu}. \tag{10}$$

As expected, this expression has a higher operation count, $\sim N_p N^2$, increased by a factor of $N/2N_{occ}$ from that of Eq. (8).

The electrostatic potential V_s may be obtained by solving Poisson's equation with a multipole expansion for the density ρ [5, 31]. The cost for a multipole expansion solution of V_s is of order N^2. The exchange-correlation potential μ_{xc} can be evaluated with an order N^2 cost using fitted densities on the grid. However, the prefactor depends strongly on whether $\nabla\rho$ is to be evaluated. To avoid the high cost of evaluating $\nabla\rho$ dependent terms in μ_{xc}, a post-SCF treatment is adopted where the gradient corrections are included as a perturbation to the self-consistent LDA solutions [32–34]. Since the cost of evaluating V on the grid can be maintained low, the order N^3 matrix assembly and density construction will be the dominant procedures in the range of accessible sizes.

3. PRESCREENING ALGORITHMS

On each grid point, the right side of Eq. (7) is a matrix $\chi_\mu \omega \Lambda \chi_\nu$. There are many ways to determine the sparsity in these "micro" matrix elements. With rank-1 updates, it is possible to form a sparse matrix by eliminating small elements in the basis function vector χ, before it is used to expand ψ in Eq. (1) and build the matrix, e.g., by a spherical cutoff of the tail of the basis functions [26]. Since the cost of the prescreening tests becomes low for large N, asymptotic linear scaling can be achieved [26, 27] . However, this comes at the expense of the performance in the intermediate size range. In this size range, the vector χ will have much fewer insignificant elements resulting in a drastic decrease in the number of identifiable small matrix elements. The missing small values are those with a small product value $\chi_\mu \chi_\nu$ but moderate values for both χ_μ and χ_ν. With lesser sparsity to exploit, the efficiency of these algorithms decreases.

To account for this type of sparsity, the matrix elements must be examined more closely. Since the workload per element is small, simplifying approximations are needed to reduce the overhead. For atom-centered local basis functions, it is natural to partition the matrix with atomic centers. This will allow for several matrix elements to be examined once together with the spatial variation of the basis functions estimated using an "envelope" function. The prescreening can be made with the following test

$$\eta_a(\mathbf{r}_k)|\omega_k \Lambda(\mathbf{r}_k)|\,\eta_b(\mathbf{r}_k) < \epsilon^\Lambda, \tag{11}$$

where η_a is defined as the maximum, at a grid point $\mathbf{r}_k$, of all χ_μ with centers at the same nucleus a (i.e., $\mathbf{R}_\mu = \mathbf{R}_a$),

$$\eta_a(\mathbf{r}) \equiv \eta(|\mathbf{r} - \mathbf{R}_a|) = \max_{\mu\in a}\{|\chi_\mu(\mathbf{r}; \mathbf{R}_\mu)|\}, \tag{12}$$

and ϵ^Λ is the threshold which provides a control of both the truncation error and prescreening efficiency. When the inequality in Eq. (11) is held, the contributions to the matrix elements $\Lambda_{\mu\nu}$ from grid point $\mathbf{r}_k$ in the specified block ($\mu \in a$, $\nu \in b$) are discarded. The algorithm implementing this matrix sparseness test is shown in Fig. 1.

```
loop over k (grid points):
    store intermediate reuslts:
        t_μ ← χ_μ(k) * ω(k) * Λ^op(k), for all μ
    loop over a and b ≥ a (atom blocks):
        if η_a(k) * | ω(k) * Λ^op(k) | * η_b(k) ≥ ε^Λ, then
            for μ ∈ a and ν ∈ b, do
                Λ_μν ← Λ_μν + t_μ * χ_ν(k)
            end do
        end if
    end loop
end loop
```

FIG. 1. Algorithm implementing the prescreening test Eq. (11) for matrix update. See text.

Hydrogen atoms usually have much fewer basis functions than heavier ones such as carbon and silicon, and therefore smaller block size in the partitioned matrix. For a typical medium sized basis with double split valence [35], the block size is only 2 by 2, which is too small to justify the cost of the test to be made. A remedy is to fuse the hydrogen block into a neighboring block owned by another atom. This is convenient since a hydrogen atom is almost always bonded to its nearest neighbor heavier atom. Only rarely a hydrogen atom is in a non-bonding or bridging position, in which cases the corresponding matrix blocks should be left unfused.

The envelope η needs to be recalculated after the fusion. To keep the cost low, the center of the hydrogen basis functions is temporarily shifted to the center of the heavier atom $\mathbf{R}_a$. As a result, an error will be produced by miscalculating the distance from hydrogen center to a grid point. However, this error tends to be small due to the compensating effects of one χ in the product $\chi(\mathbf{R}_a)\omega\Lambda\chi(\mathbf{R}_b)$ being in greater distance to the grid point than the other χ. Only when the two χ are centered on the same atom ($\mathbf{R}_b = \mathbf{R}_a$) will this error become large. In this case, a tightened threshold can be used to reduce the errors.

In a similar way, the density construction process, Eq. (9), can be facilitated by exploiting the sparseness on an integration grid.

$$\eta_a(\mathbf{r}_k)\bar{P}_{ab}\eta_b(\mathbf{r}_k) < \epsilon^P, \tag{13}$$

where η are defined in Eq. (12), $\bar{P}_{ab}$ is the maximum of a block of $P_{\mu\nu}$ with the atomic centers a and b,

$$\bar{P}_{ab} = \max_{\mu\in a,\nu\in b}\{|P_{\mu\nu}|\}, \tag{14}$$

and ϵ^P is the threshold. The implementation is similar to that for matrix update, Fig. 1. However, the magnitudes of the "micro" contributions $\chi_\mu P_{\mu\nu}\chi_\nu$ can vary more wildly on a given grid point due to the presence of $P_{\mu\nu}$ which is index dependent.

4. PERFORMANCE ANALYSIS

Two sets of sample molecules with compact structure, polycyclopentadienes $(C_5H_6)_{m+1}$ ($m = 1$–8) and helicines $C_{4n+2}H_{2n+4}$ ($3 \le n \le 33$), see Fig. 2, are used for the demonstration

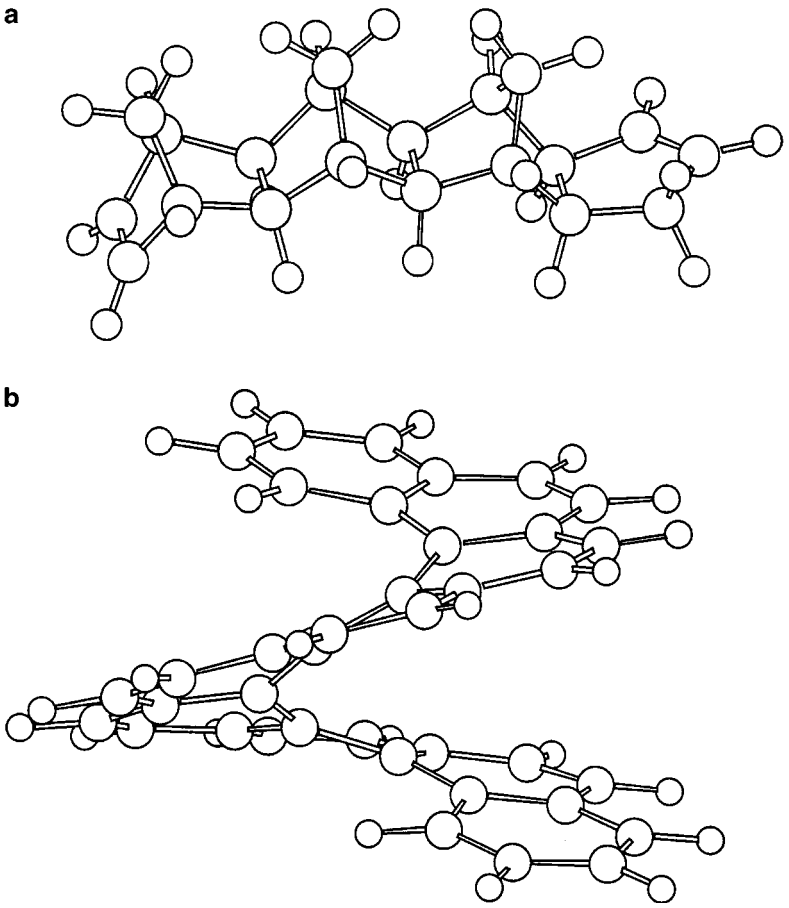

FIG. 2. Schematic view of sample molecules: (a) tetra-polycyclopentadiene and (b) [9]helicine. Large circles, carbon atoms; small circles, hydrogen atoms.

of the performance of the new prescreening algorithms. Chainlike molecules which tend to have low scaling power behavior are avoided. A double valence plus single polarization (on non-hydrogen atoms) basis is used which allocates 14 numerical atomic orbital functions (three s, six p, and five d-types) for a carbon atom and two s-type orbitals for a hydrogen atom. The integration grid is generated as spheres around each atom with the grid size approximately proportional to the basis set size, namely, $N_p \approx 540N$ for polycyclopentadienes and $430N$ for helicines. No symmetry is used for the helicines. The timing is made on MIPS R10000 processors with a 250 MHz clock speed. For large molecules, multiple processors are used with properly assigned parallel speedup coefficients for converting to uniprocessor time.

4.1. *Speedups*

Figure 3 shows the speedup of prescreening at different level of sparsity. The original order N^3 algorithms, which are optimized for matrix operations [13, 36], are used as reference. The degree of sparsity, measured as the fraction of the total "micro" contributions being discarded, increases with iteration as a result of updating upon results from the previous

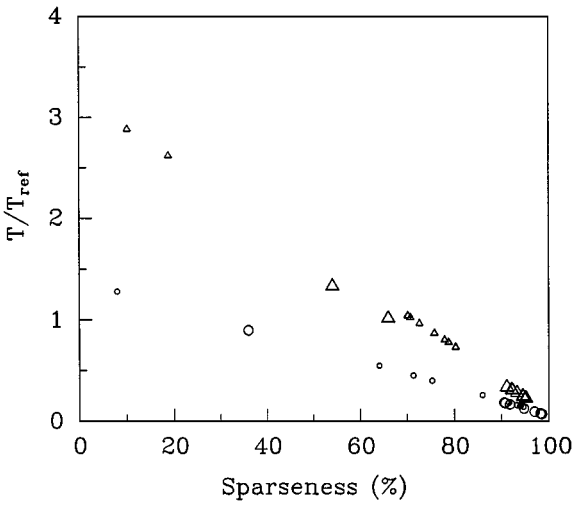

FIG. 3. Time ratio over the reference algorithms with increasing discard rate. Circles, matrix assembly; triangles, density construction. Small symbols (circles and triangles), [9]helicine; large symbols, [24]helicine.

iterations (differential update) [20]. Some performance characters can be seen here. First, the speed gain is rather small in the first iterations due to overhead. Even worse is the high point (70% discard rate) for the density construction to break even. This is partly due to the use of a higher operation count expression (by a factor of $N/2N_{occ} = 2.3$ here) and partly due to the overhead that can be attributed to the prescreening tests. The intersect of the T/T_{ref} curves at the zero discard rate gives a measure of the total overhead. This overhead is characteristic of the prescreening in three-dimensional integration. For four-center Gaussian integrals in conventional analytical integration, the sparsity comes from contributions of four different centers [20], compared to just two here in the three-dimensional integration.

Figure 4 shows the scaling laws of the prescreening algorithm for the helicines. For better illustrations, a properly normalized time T/N^2 is used. For comparison, the time for the

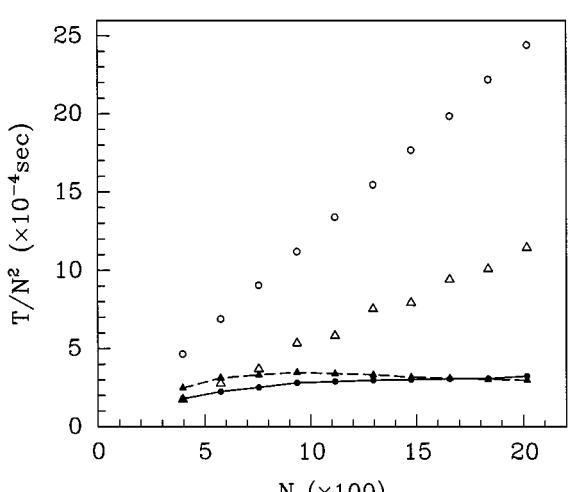

FIG. 4. Normalized time (T/N^2) vs N for helicines. Circles are for matrix assembly and triangles for density construction. Filled symbols (circles and triangles) are for precreening algorithm and open symbols for reference algorithm. Data are averaged over 10 iterations (using improved initial guess for density input, see text).

reference order N^3 algorithm is also included. A near N^2 scaling for the new algorithm is clearly seen. The scaling behavior of the polycyclopentadienes is essentially the same, except that these molecules span a smaller size range ($N \approx 200$–800) and the corresponding curve has a slightly negative slope. This difference is not surprising. The helicines are electronically more delocalized due to the six-membered carbon rings and their density matrix elements are expected to converge more slowly. Examinations of the distribution of $P_{\mu\nu}$ with separation $|\mathbf{R}_\mu - \mathbf{R}_\nu|$ for the helicines revealed that there is a significant population of nonzero elements separated by a few bond lengths upon convergence.

The asymptotic scaling for the density construction shown in Fig. 4 is achieved with the help of another prescreening upon the existing one. The added test is upon a block of grid points [27] for a cluster of atoms. This is possible since the $P_{\mu\nu}$ in the product $\chi_\mu P_{\mu\nu} \chi_\mu$ goes off quickly with interatomic distance after exceeding a critical length [25, 37]. For the helicines, the fraction of the total discard accounted for by this "macro" test increases readily from below 1/10 for [15]helicine ($N = 1116$) to 1/4 for [33]helicine ($N = 2016$). For matrix assembly, no significant improvements are observed, probably due to the sensitivity of the potential V on spatial locations. Rearranging grid points is too costly for the current self-consistent procedure which solves Poisson's equation with multipole component expansion on spherical grids built around each nucleus [5].

4.2. *Thresholds and Errors*

The thresholds ϵ in the prescreening tests Eqs. (11) and (13) need to be adjusted for use with differential update,

$$\epsilon^X = \min\left\{ d_X \langle \Delta\rho \rangle^{\gamma_X}, \bar{\epsilon}^X \right\}. \tag{15}$$

Here, $X = \Lambda$ or P while d, γ, and $\bar{\epsilon}$ are constant parameters. $\langle \Delta\rho \rangle$ is the root-mean-square difference of densities between two successive iterations, which decreases with iteration. $\bar{\epsilon}$ is used as a safeguard for $\langle \Delta\rho \rangle$ which in occasions can be too large to use. The values listed in Table I are chosen empirically; with these threshold values, the errors in the calculated total energy can be kept below 10^{-5} a.u. for all the sample molecules used in this section.

Figure 5 shows the distributions of $\chi_\mu \omega V \chi_\nu$ and $\chi_\mu P_{\mu\nu} \chi_\nu$ with respect to their magnitude. The center of the distribution shifts towards the left side for smaller magnitudes as the convergence increases. However, in the early stages of an iteration procedure, convergence can be poor and a significant proportion of the distribution would remain in relatively large

TABLE I
Parameters Used in Eq. (15) to Determine the Prescreening Thresholds ϵ in Eqs. (11) and (13)

X^a	d	γ	$\bar{\epsilon}^b$
T	2×10^{-11}	0	2×10^{-11}
S	1×10^{-10}	0	1×10^{-10}
V	4×10^{-9}	0.5	4×10^{-11}
P	2×10^{-6}	0.75	2×10^{-8}

[a] For one-center contributions, i.e., $\mathbf{R}_b = \mathbf{R}_a$, the ϵ is multiplied by 1/50, see text in Section 3.

[b] For the first build of $V_{\mu\nu}$ and ρ, the threshold is reduced to $\epsilon = 0.001\bar{\epsilon}$.

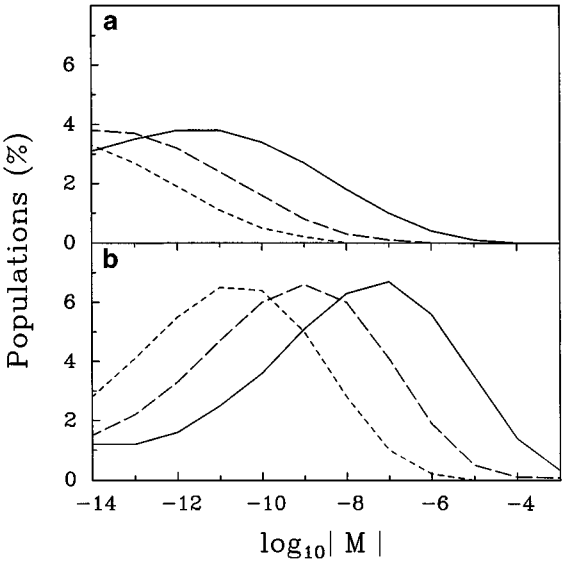

FIG. 5. Distributions of "micro" contributions M with iterations for [12]helicine. (a) $M = \chi_\mu \omega V \chi_\nu$ and (b) $M = \chi_\mu P_{\mu\nu} \chi_\nu$. The solid, long dashed, and short dashed lines represent the first, middle, and last iteration, respectively.

magnitude, particularly for contributions to the density, as it is shown here in Fig. 5. To improve the overall performance of the prescreening algorithm, it is crucial to supply the iteration procedure with a sufficiently converged density matrix. Such an input density can be obtained by iterating over on a coarser integration grid. Figure 6 shows its effects on the convergence of the total energy and density for a [12]helicine molecule.

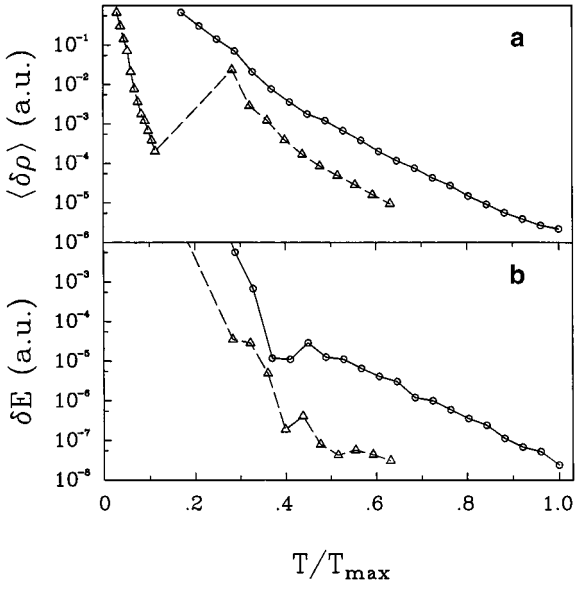

FIG. 6. The effects of the improved initial density matrix on the convergence of (a) density and (b) total energy, for a [12]helicine molecule. δE is the deviation of total energy at the current iteration from the converged value (data obtained without prescreening).

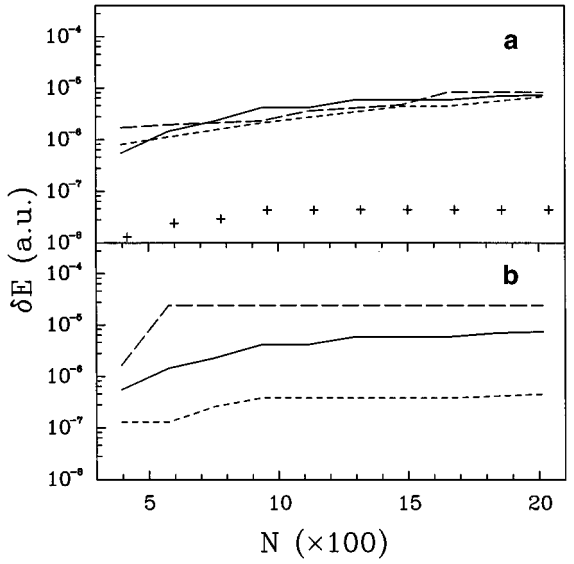

FIG. 7. Truncation errors in total energy. (a) Prescreening turned on for matrix assembly (long dashed), for density construction (short dashed), and for both (solid line). (b) Errors in total energy at different tolerance levels: using thresholds as given in Table I (solid line), scaled by 50 (long dashed), and scaled by 0.1 (short dashed). The crosses represent the errors per atom. The alternations in the errors actually observed are removed by connecting data points piecewise with a strictly uphill manner for increasing N. The resulting curve represents an upper bound of the errors.

Figure 7 shows the errors in total energy caused by prescreening. In the upper penal, partial errors arising from truncating in matrix update and in density construction separately are shown. They are in comparable magnitude. In the lower penal, the error in total energy δE is plotted as a function of N for thresholds being set at three different levels, with one using the values given in Table I and the other two scaled by 0.1 and 50. The errors are well bound. For the default thresholds, δE increases by approximately an order of magnitude from $N \approx 200$ to 2000, with its maximum less than 0.05×10^{-6} a.u. per atom. In comparison, the error intrinsic to the quadrature rules is much larger in maginitude, in the range between $0.1–10 \times 10^{-6}$ a.u. per atom, depending on the details of the grid used [8–12].

Errors can occur from the miscalculation of distances for the fused matrix blocks corresponding to a group consisting of a carbon atom and hydrogens attached to it (Section 3). This error can be estimated by simply restoring to one block (and one η) for each atom. Surprisingly, the results show no significant differences between the two ways of defining η in Eq. (12). This suggests that the mislocation error could be made sufficiently small to be of comparable magnitude of the "normal" truncation errors.

4.3. *Overall*

Figure 8 shows the time for a complete self-consistent calculation with prescreening threshold set at the above three levels for the helicines. The T/N^2 curves show an overall performance of order N^2 starting from $N \approx 500–600$. With the more "insulator" like and more openly structured polycyclopentadienes, the oneset of N^2 performance reduces to $N \approx 300$.

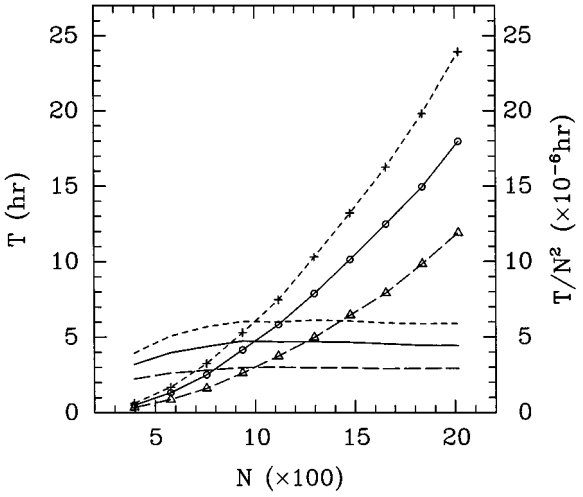

FIG. 8. Comparison of performance at different thresholding levels, using thresholds in Table I scaled by 1 (solid), 50 (long dashed), and 0.1 (short dashed). To the left, time of a complete self-consistent calculation for helicines (lines with symbols). To the right, time divided by N^2.

The cost of a complete self-consistent calculation with prescreening breaks even with the reference at a size $N \approx 200$. At such a small size, the matrix operations on the integration grid are no longer rate determinant and the difference in performance begins to vanish.

5. CONCLUSION REMARKS

A simple prescreening algorithm is presented with an effect to significantly reduce the cost of the density functional calculations with three-dimensional integrations. By a direct estimate of the spatial variation of the atomic orbital basis functions, a tight control on the truncation errors can be obtained while maintaining an order N^2 scaling over a wide size range. In the following, the limitations and some potential applications of the algorithm are discussed.

Molecular symmetry has been used to accelerate integrations on three-dimensional grid [3, 11, 14, 19, 37]. With a point group symmetry of order g, both the number of non-vanishing matrix elements and the grid size can be reduced by a factor of $\sim g$, and the computational cost will be reduced by the same proportion. However, adapting atom-centered basis functions to molecular symmetry will destroy the spatial locality of these basis functions, making them unbound to any atomic center. The prescreening schemes presented in Section 3 will fail. The present algorithms are incompatible with the use of symmetry adapted atomic orbital basis.

For polyatomic systems with low-lying electronic states competing for the ground state, such as in a transition metal cluster, small perturbations arising from cutoff errors in the early stages of an iteration procedure may change the course of convergence at later stages completely. The use of differential update for density matrix increases the chance for instability. The reduced magnitude of $\Delta P_{\mu\nu}$ as a result of differentiation encourages aggressive cutoffs where it should not, which causes divergence. This problem is expected to occur in methods performing density matrix weighed prescreening upon differentially updated quantities. The present algorithms are not suited for systems where one-electron occupancies near the Fermi level must be altered to achieve convergence [3, 38].

In the three-dimensional integration approaches based on divide and conquer, low power scaling performance is usually achieved with reducing the size of the partitioned subsystems at the expense of accuracy [28, 40, 38]. The cost for solving the subsystems, which is in the intermediate size range, is still dominated by the order N^3 integrations. Increasing the subsystem size, hence the accuracy, will be limited by this cost. Exploiting the sparsity in the subsystems can improve the overall scaling and allow larger "buffer" regions to be used for better error control.

The electrostatic potentials V_s can be obtained either through a projection [5] or a fitting [3, 41]. Either way, an atom-centered basis is required. When a fitting is used and it minimizes errors in the fitted density [3], a system of linear equations will be produced with the matrix elements in a form similar to that in Eq. (7). Since the fitting requires a large basis, the cost for generating this matrix can lead other order N^3 integration procedures. On the other hand, since the density is more localized than a molecular orbital basis function, greater sparsity can be expected for the integration.

ACKNOWLEDGMENTS

The author thanks Dr. Warren Hehre for critical discussions. The benchmark was performed on an Origin 2000 computer at Navel Research Laboratory, Washington, DC.

REFERENCES

1. W. Kohn and L. J. Sham, Self-consistent equations including exchange and correlation effects, *Phys. Rev. A* **140**, 1133 (1965).
2. D. E. Ellis and G. S. Painter, Discrete variational method for the energy-band problem with general crystal potential, *Phys. Rev. B* **2**, 2887 (1970).
3. E. J. Baerends, D. E. Ellis, and P. Pos, Self-consistent molecular Hartree-Fock-Slater calculations. I. The computational procedure, *Chem. Phys.* **2**, 41 (1973).
4. A. Rosen and D. E. Ellis, Relativistic molecular calculations in the Dirac–Slater model, *J. Chem. Phys.* **62**, 3039 (1975).
5. B. Delley, An all-electron numerical method for solving the local density functional for polyatomic molecules, *J. Chem. Phys.* **92**, 508 (1990).
6. A. D. Becke and R. M. Dickson, Numerical solution of Schrödinger's equation in polyatomic molecules, *J. Chem. Phys.* **92**, 3610 (1990).
7. W. Yang, Direct calculation of electron density in density-functional theory, *Phys. Rev. Lett.* **66**, 1438 (1991).
8. F. W. Averill and G. S. Painter, Pseudospherical integration scheme for electronic-structure calculations, *Phys. Rev. B* **39**, 8115 (1989).
9. P. M. Boerrigter, G. te Velde, and E. J. Baerends, Three-dimensional numerical integration for electronic structure calculations, *Int. J. Quantum Chem.* **33**, 87 (1988).
10. G. te Velde and E. L. Baerebds, Numerical integration for polyatomic systems, *J. Comput. Phys.* **99**, 84 (1992).
11. M. R. Perderson and K. A. Jackson, Variational mesh for quantum-mechanical simulations, *Phys. Rev. B* **41**, 7453 (1990).
12. A. D. Becke, A multicenter numerical integration scheme for polyatomic molecules, *J. Chem. Phys.* **88**, 2547 (1988).
13. W. Ravenek, Vectorization of quantum chemical programs: The HFS–LCAO package, in *Algorithms and Applications on Vector and Parallel Computers*, edited by H. J. J. te Riele, T. J. Dekker, and H. A. van de Vorst (Elsevier, Amsterdam, 1987), p. 1759.
14. L. Lou, P. Nordlander, and R. E. Smalley, Fullerene nanotubes in electric fields, *Phys. Rev. B* **52**, 1429 (1995).
15. Y. S. Li, M. C. Wrinn, J. M. Newsam, and M. P. Sears, Parallel implementation of a mesh-based density functional electronic structure code, *J. Comput. Chem.* **16**, 226 (1995).

16. L. Fan and T. Ziegler, Optimization of molecular structures by self-consistent and non-local density-functional theory, *J. Chem. Phys.* **95**, 7401 (1991).
17. A. Rosa, A. W. Ehlers, E. J. Baerends, J. G. Snijders, and G. te Velde, Basis set effects in density functional calculations on the metal–ligand and metal–metal bonds of $Cr(CO)_5$–CO and $(CO)_5Mn$–$Mn(CO)_5$, *J. Phys. Chem.* **100**, 5690 (1996).
18. F. W. Averill and D. E. Ellis, An efficient numerical multicenter basis set for molecular orbital calculations: Application to $FeCl_4$, *J. Chem. Phys.* **59**, 6412 (1973).
19. B. Delley, D. E. Ellis, A. J. Freeman, E. J. B. Rends, and D. Post, Binding energy and electronic structure of small copper particles, *Phys. Rev. B* **27**, 2183 (1983).
20. J. Almlöf, K. Faegri, Jr., and K. Korsell, Principles for a direct SCF approach to LCAO–MO *ab-initio* calculations, *J. Comput. Chem.* **3**, 385 (1982).
21. M. Häser and R. Ahlrichs, Improvements on the direct SCF method, *J. Comput. Chem.* **10**, 104 (1989).
22. P. M. W. Gill, B. G. Johnson, and J. A. Pople, A simple yet powerful upper bound for coulomb integrals, *Chem. Phys. Lett.* **217**, 65 (1994).
23. C. A. White and M. Head-Gordon, A *J* matrix engine for density functional theory calculations, *J. Chem. Phys.* **104**, 2620 (1996).
24. M. C. Strain, G. E. Scuseria, and M. J. Frisch, Achieving linear scaling for the electronic quantum coulomb problem, *Science* **271**, 51 (1996).
25. E. Schwegler and M. Challacombe, Linear scaling computation of the Hartree–Fock exchange matrix, *J. Chem. Phys.* **105**, 2726 (1996).
26. J. M. Perez-Jorda and W. Yang, An algorithm for 3D numerical integraion that scales linearly with the size of the molecules, *Chem. Phys. Lett.* **241**, 469 (1995).
27. R. E. Stratmann, G. E. Scuseria, and M. J. Trisch, Achieving linear scaling in exchange-correlation density functional quadratures, *Chem. Phys. Lett.* **257**, 213 (1996).
28. R. T. Gallant and A. St-Amant, Linear scaling for the charge density fitting procedure of the linear combination of Gaussian-type orbitals density functional methods, *Chem. Phys. Lett.* **256**, 569 (1996).
29. S. K. Goh and A. St-Amant, Using a fitted electronic density to improve the efficiency of a linear combination of Gaussian-type orbital calculation, *Chem. Phys. Lett.* **264**, 9 (1997).
30. C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Linear scaling density functional calculations via the continuous fast multipole method, *Chem. Phys. Lett.* **253**, 268 (1996).
31. B. Delley and D. E. Ellis, Efficient and accurate expansion methods for molecules in local density models, *J. Chem. Phys.* **76**, 1949 (1982).
32. L. Fan and T. Ziegler, The influence of self-consistency on nonlocal density functional calculations, *J. Chem. Phys.* **94**, 6057 (1991).
33. J. Andzelm and E. Wimmer, Density functional Gaussian-type-orbital approach to molecular geometries, vibrations, and reactions, *J. Chem. Phys.* **96**, 1280 (1992).
34. A. D. Becke, Density-functional thermochemistry. II. The effect of the exchange-only gradient correction, *J. Chem. Phys.* **96**, 2155 (1992).
35. W. J. Hehre, L. Radom, P. v. R. Schleyer, and J. A. Pople, *Ab Initio Molecular Orbital Theory* (Wiley, New York, 1986).
36. J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers* (SIAM, Philadelphia, 1991).
37. W. Kohn, Density functional theory for systems of very many atoms, *J. Int. Quantum Chem.* **56**, 229 (1995).
38. O. Warschkow, J. Dyke, and D. E. Ellis, A divide-and-conquer implementation of the discrete variational DFT method for large molecular and solid systems, *J. Comput. Phys.* **143**, 70 (1998).
39. B. I. Dunlap, Symmetry and Degeneracy in X_α and density functional theory, in *Ab Initio Methods in Quantum Chemistry, II*, edited by K. P. Lawley (Willey, New York, 1987), p. 287.
40. W. Yang and T. S. Lee, A density-matrix divide-and-conquer approach for electronic structure calculations of large molecules, *J. Chem. Phys.* **103**, 5674 (1995).
41. B. I. Dunap, J. W. D. Connolly, and J. R. Sabin, On some approximations in applications of X_α theory, *J. Chem. Phys.* **71**, 3396 (1979).