Efficient computation of the exchange-correlation contribution in the density functional theory through multiresolution

Jing Kong,[a] Shawn T. Brown,[b] and Laszlo Fusti-Molnar
Q-Chem Inc., Pittsburgh, Pennsylvania 15213

A new algorithm is presented to improve the efficiency of the computation of exchange-correlation contributions, a major time-consuming step in a density functional theory (DFT) calculation. The new method, called multiresolution exchange correlation (mrXC), takes advantage of the variation in resolution among the Gaussian basis functions and shifts the calculation associated with low-resolution (smooth) basis function pairs to an even-spaced cubic grid. The cubic grid is much less dense in the vicinity of the nuclei than the atom-centered grid and the computation on the former is shown to be much more efficient than on the latter. MrXC does not alter the formalism of the current standard algorithm based on the atom-centered grid (ACG), but instead employs two fast and accurate transformations between the ACG and the cubic grid. Preliminary results with local density approximation have shown that mrXC yields three to five times improvement in efficiency with negligible error. The extension to DFT functionals with generalized gradient approximation is also briefly discussed. © *2006 American Institute of Physics*. [DOI: 10.1063/1.2173244]

I. INTRODUCTION

Density functional theory (DFT) is the most widely applied electronic structure method. It is the only tool that enables researchers in molecular sciences to carry out accurate *ab initio* quantum simulations with reasonable computational cost. There are three time-consuming steps in a DFT computation: the calculation of Coulomb interaction [and Hartree-Fock (HF) exchange if hybrid functionals are used], the determination of DFT exchange-correlation (XC) contribution, and the diagonalization of the Fock matrix. Numerous efforts have been made in the last decade to reduce the computational cost of the DFT calculation, since it is still quite computationally demanding for a system with a hundred atoms or more. The first linear-scaling Coulomb algorithm for DFT was proposed and implemented in the Q-CHEM program[1] in 1996 by White *et al.* with the continuous fast multipole method[2,3] (CFMM), a generalization of Greengard's fast multipole method[4] for continuously charged matter. CFMM for the first time enabled linear-scaling DFT calculation with generalized gradient approximation (GGA) functionals.[5–7] For the Coulomb interaction among charges that are near each other in which CFMM is not applicable, White and Head-Gordon[8] and Shao and Head-Gordon[9,10] developed the *J*-engine technique in Q-CHEM that reduced the computational cost of the conventional four-index integrals with large basis sets by several times. Fusti-Molnar and Kong[11] have recently implemented a Coulomb method in Q-CHEM called the Fourier transform Coulomb (FTC) method developed originally by Fusti-Molnar and Pulay,[12–14] which replaces the least efficient part of the previous Coulomb methods with an accurate numerical integration scheme that scales in $O(N^2)$

instead of $O(N^4)$ with the basis set size. It has been shown through a series of benchmark calculations that FTC provides manifold improvement in efficiency over the most efficient existing Coulomb scheme, i.e., the combination of CFMM and *J* engine. The combination of FTC, CFMM, and *J* engine represents the optimal solution to the Coulomb problem with Gaussian basis functions without loss of accuracy. Challacombe *et al.* also developed a method for fast coulomb calculation.[15] Algorithms have also been developed for fast computation of HF exchange with examples including the OnX method by Schwegler *et al.*[16] and the LinK method by Ochsenfeld *et al.*[17]

Advances in the Coulomb and HF exchange computation have made more pronounced the determination of the XC contribution. In particular, it has been shown[11] that the advances in the Coulomb computation have made the calculation of the XC contribution the most time-consuming step in a DFT calculation with GGA functionals. The computation time for the diagonalization of the Fock matrix is insignificant compared to that of the computation of either XC or Coulomb contributions for systems with up to several thousand basis functions, although the diagonalization scales cubically. For larger systems, several linear-scaling algorithms have been developed in recent years.[18–20] The focus of this paper will be the presentation of a new algorithm to speed up the XC part of the DFT calculation.

The standard implementation for evaluating the XC contribution was developed by Pople and co-workers in the early 1990s[21,22] with the development of the SG-1 grid.[23] Linear scaling was achieved by taking advantage of the negligible overlap among basis functions whose centers are far from each other. Since then, there have been relatively few efforts to improve the efficiency of the XC part. Stratmann *et al.*[24,25] changed the relative weights of Becke's weight scheme such that fewer effective neighboring atoms are found for each

---

[a] Electronic mail: jkong@q-chem.com
[b] Present address: Pittsburgh Supercomputing Center, Carnegie Mellon University, Pittsburgh, PA 15213.

atom. Combining it with some screening techniques, they have achieved significant gains in efficiency. Recently, Brown and Kong have developed and implemented a scheme utilizing the small differences between self-consistent-field (SCF) iterations and reducing the CPU time by a factor of two.[26]

Here, we propose a new scheme for the efficient evaluation of the XC contribution called the multiresolution exchange-correlation (mrXC) method. It is based on the well- known observation that a Gaussian function can be evaluated accurately on a numerical grid with sufficient resolution, and if a Gaussian is more diffuse (or smoother) than another, the resolution (or the density) of the grid for the former can be lower than that for the latter while retaining the same accuracy. Some preliminary results will be given to demonstrate the effectiveness of this method.

In practical DFT calculations, the forms of modern XC functionals are complicated and integrals over the functionals are generally evaluated numerically. A typical numerical quadrature for XC is composed of atom-centered grids (ACGs).[27] In such a grid, the density of the grid points, or the resolution of the points, is high near the nucleus because the electron density changes rapidly in this region and becomes increasingly lower away from the nucleus as the electron density changes slower. The grid density can vary by several orders of magnitude between the high and low density regions. In the current implementation, a basis function is evaluated at each grid point that it overlaps with, regardless of the grid density. While this is necessary for functions with high resolutions, it is unnecessary for smooth functions with low resolutions because such a function changes slowly in space. Instead, one can just evaluate the function values at an appropriate interval in the three-dimensional space (i.e., forming a cubic grid) and interpolate the values at the required points through those at the nearby vertices on the even-spaced cubic grid.

The feasibility of the interpolation was recently studied by Brown *et al.*[28] Divided-difference polynomial interpolation[29] was explored as a means of translating the electron density and its gradients from the cubic grid to points on the atom-centered grid. Aspects of accuracy, error control through the use of the grid density, were studied, and it is shown that majority of the effective basis function pairs (up to 80%) can be evaluated on the cubic grid accurately.

The mrXC scheme has the potential to accelerate the XC calculation significantly because the smooth basis functions are typically more expensive to evaluate under the current implementation due to their large spatial extent. Furthermore, there are more smooth basis functions than core basis functions, since most of the valence basis functions can be considered smooth. Using the cubic grid instead of the ACG also has a distinct technical advantage. It will be shown that Gaussian basis functions can be evaluated efficiently on a cubic grid by taking advantage of the uniformity of the points in the three dimensions and achieve much less cost per grid point than the ACG.

II. THEORY

To demonstrate the mrXC algorithm, we will first start with the simple local density approximation, i.e., the XC functional is a function of only the electron spin density, $f \equiv f(\rho_\alpha, \rho_\beta)$. Examples of such functionals include the Slater exchange functional[30] and the Vosko-Wilk-Nusair (VWN) correlation functional.[31]

To obtain the DFT energy, calculation of the XC contribution to the Fock matrix is necessary. The standard procedure to evaluate the XC contribution to the Fock matrix is the following:

(1) Generate the grid points for each atom and calculate the Becke relative weights.[27] The ACG in Q-CHEM is constructed with two parts: the radial part and the angular part. The radial part is typically handled by the Euler-Maclaurin scheme proposed by Murray *et al.*[32] or the Multix method by Chien and Gill,[33] and the angular component by Lebedev quadrature.[34–37]

(2) Evaluate the spin densities on the grid as follows:

$$\rho^\alpha(i) = \sum_{\mu\nu} P^\alpha_{\mu\nu} \phi_{\mu\nu}(i), \qquad (1)$$

where i is the index for the grid points and $\phi_{\mu\nu}$ is the product of two basis functions, i.e., $\phi_{\mu\nu}(i) = \phi_\mu(i)\phi_\nu(i)$. In general, $\phi_{\mu\nu}$ is a linear combination of Gaussian functions at different centers.

(3) Evaluate the functional derivatives, f^{ρ^α} and f^{ρ^β}, on the grid.

(4) Evaluate the XC matrix, the exchange-correlation contribution to the Fock matrix as follows:

$$F^\alpha_{\mu\nu} = \sum_i w(i) f^{\rho^\alpha}(i) \phi_{\mu\nu}(i), \qquad (2)$$

where $w(i)$ is the weight for point i. In practice, the weights are always combined with the functional derivatives, and so we will omit them to simplify discussions:

$$F^\alpha_{\mu\nu} = \sum_i f^{\rho^\alpha}(i) \phi_{\mu\nu}(i). \qquad (3)$$

Among these four steps, step (2) and (4) are the most time-consuming, because they include loops over basis function pairs and grid points. The computational cost of the two steps is equal in floating point operation counts, and scales quadratically with respect to the basis set size and asymptotically linear with respect to the molecular size. The reason for the linear scaling is that the Gaussian basis functions have limited spatial extent, and the effective number of significant function pairs has a constant limit for each grid point.

To reduce the computational cost of these two time-consuming steps, it is observed that a well-behaved function such as a Gaussian function has a finite resolution; meaning it can be represented accurately on an even-spaced grid (a cubic grid) with finite grid density. The value of the function at any point in the space (e.g., a point in the ACG) can be interpolated accurately using the values of the function at the grid points, especially for the purpose of integration; that is,

$$\phi_{\mu\nu}(i) = \sum_m C_{im}\phi_{\mu\nu}(m) \tag{4}$$

for certain function pairs. In the above equation, i is the index for the ACG, m is the index for the cubic grid, and C_{im} is the set of interpolation coefficients. We will call the function pairs that satisfy Eq. (4) *smooth* function pairs. The actual criterion for the smoothness of the Gaussian function will be discussed later.

It is then straightforward to obtain part of the density on the ACG through interpolation by combining Eqs. (1) and (4):

$$\rho^s(i) = \sum_{\mu\nu}\sum_m C_{im}P_{\mu\nu}\phi^s_{\mu\nu}(m) = \sum_m C_{im}\rho^s(m). \tag{5}$$

A superscript S (for smooth) is used to indicate the contribution of the smooth function pairs to the density, and a superscript C (for *compact*) for the part of the density that cannot be interpolated. The total density becomes

$$\rho(i) = \rho^S(i) + \rho^C(i), \tag{6}$$

where $\rho^c(i)$ is the density value on the ACG corresponding to the compact function pairs, which is evaluated explicitly at each grid point using Eq. (1). Please note that we have omitted the spin sign for simplicity, since the formulas for α and β spins are equivalent.

The above procedure introduces the additional cost of interpolation. This overhead can be kept low, however, if a local interpolation method is chosen: that is, the value at a point is only interpolated by the nearby points on the cubic grid, ensuring that the cost is only proportional to the number of ACG points, i.e., scales linearly. Once the total density at each ACG point is obtained, the functional derivative value can be evaluated at the point.

Similarly, the XC matrix elements that correspond to smooth function pairs can be formed on the cubic grid. This can be achieved by inserting Eq. (4) into Eq. (3):

$$F_{\mu\nu} = \sum_i\sum_m f^p(i)C_{im}\phi_{\mu\nu}(m) = \sum_m \tilde{f}^p(m)\phi_{\mu\nu}(m), \tag{7}$$

where

$$\tilde{f}^p(m) = \sum_i f^p(i)C_{im}. \tag{8}$$

It should be noted that although Eq. (7) is formally the same as Eq. (3), it is not simply a replacement of the original ACG with the cubic grid because $\tilde{f}^p(m)$ is not the value of the functional derivative on the latter. In fact, replacing $\tilde{f}^p(m)$ with the actual functional derivative value $f^p(m)$ will result in significant errors, for f^p cannot be accurately represented by the cubic grid since it is a function of the total electron density and contains compact basis function pairs that do not confirm Eq. (4).

Since the formulas presented here are formally the same as those for Coulomb evaluation with FTC,[11,14] further cost reduction can be achieved by combining the two schemes. Although evaluated on a cubic grid, FTC requires a grid density that is different from the one needed by the mrXC method because it uses Fourier functions as auxiliary func-

tions. Fourier functions are global in nature and, in general, need a lower grid density than that for the local interpolation functions used for mrXC. Because the cost of the calculation is directly proportional to the grid density, it is desirable to evaluate density and form the XC matrix on the coarse cubic grid (the cubic grid of less density). Therefore a fast transformation of values between two cubic grids (the coarse and the fine grids) of different grid density is needed, which comprises the second part of the mrXC algorithm.

This procedure is as follows:

(a) Evaluate Eq. (1) in the coarse cubic grid with the index n (called grid n henceforth). The fine grid has the index m and is termed grid m henceforth.

(b) Perform a fast Fourier transformation (FFT) of $\rho(n)$ to $\rho(k)$, where k is the index for the frequencies in the Fourier (or momentum) space.

(c) Expand the momentum space from $\max(k) = K^n_{max}$ to $\max(k) = K^m_{max}$. K^n_{max} and K^m_{max} decide the grid densities for the cubic grids n and m. Since grid m has a higher grid density, K^m_{max} is larger than K^n_{max}.

(d) Fourier back transform $\rho(k)$ to the denser cubic grid m to perform the interpolation of Eq. (5).

The procedure for forming the XC matrix is the reverse of the above.

(e) Fourier transform $\tilde{f}^p(m)$ to the momentum space $\tilde{f}^p(k)$.

(f) Remove the $\tilde{f}^p(k)$ with $k > K^n_{max}$, i.e., the plane-wave functions with frequencies higher than K^n_{max}.

(g) Fourier back transform the remaining $\tilde{f}^p(k)$ to grid n.

(h) Evaluate Eq. (7) on grid n. Steps (a) and (h) can be combined with FTC.

The proof of steps (a)–(d) is straightforward, if it is recognized that the smooth function pairs are assumed to be accurately expandable on grid n:

$$\phi_{\mu\nu}(n) = \sum_{k=-K^n_{max}}^{k=K^n_{max}} \phi_{\mu\nu}(k)\chi(k,r_n), \tag{9}$$

where $\chi(k,r_n)$ is a plane-wave function $[\chi(k,r_n) = e^{2\pi i k r_n}]$. Since K^m_{max} is larger than K^n_{max}, expanding $\phi_{\mu\nu}(n)$ to $\phi_{\mu\nu}(m)$ only involves setting $\phi_{\mu\nu}(k)$ to zero for any k larger than K^n_{max}:

$$\phi_{\mu\nu}(m) = \sum_{|k|\leq K^m_{max}} \phi_{\mu\nu}(k)\chi(k,r_m),$$

$$\phi_{\mu\nu}(k) = 0 \quad \text{for } |k| > K^n_{max}. \tag{10}$$

The electron density follows the same transformations since it is a linear combination of the function pairs.

The proof of steps (e)–(h) is less obvious and proceeds as follows. One starts by inserting Eq. (10) to Eq. (7):

TABLE I. The difference in the DFT energy (ΔE) caused by mrXC in comparison with the error of the SG-1 grid.

Molecule/Basis sets	Number of atoms	Number of basis functions	Error of SG-1 (μhartree/atom)	ΔE of mrXC (μhartree/atom)
Alanine/6-31G(df,pd)	13	209	1.6	0.01
Alanine/6-31G($2df,2pd$)	13	266	1.8	0.02
(Alanine)$_5$/6-31G(df,pd)	53	869	2.9	0.1
(Alanine)$_5$/6-31G($2df,2pd$)	53	1106	2.7	0.2

$$F_{\mu\nu} = \sum_{|k| \leqslant K_{max}^m} \phi_{\mu\nu}(k) \sum_m \widetilde{f^p}(m) \chi(k, r_m)$$

$$= \sum_{|k| \leqslant K_{max}^m} \phi_{\mu\nu}(k) \widetilde{f^p}(k) = \sum_{|k| \leqslant K_{max}^n} \phi_{\mu\nu}(k) \widetilde{f^p}(k), \qquad (11)$$

where $\widetilde{f^p}(k)$ is the Fourier transformation of $\widetilde{f^p}(m)$. The last "=" sign is the recognition that $\phi_{\mu\nu}(k)$ is zero for any k larger than K_{max}^n. By the Fourier transformation theory, the $\phi_{\mu\nu}(k)$ can be expressed in terms of $\phi_{\mu\nu}(n)$ as follows:

$$\phi_{\mu\nu}(k) = \sum_{n=1}^{N_n} \phi_{\mu\nu}(n) \chi^*(k, r_n), \qquad (12)$$

where N_n is the number of grid points of grid n. This equation can be inserted into Eq. (11) and $F_{\mu\nu}$ becomes

$$F_{\mu\nu} = \sum_{n=1}^{N_n} \phi_{\mu\nu}(r_n) \sum_{|k| \leqslant K_{max}^n} \widetilde{f^p}(k) \chi^*(k, r_n) = \sum_{n=1}^{N_n} \phi_{\mu\nu}(n) \widetilde{f^p}(n), \qquad (13)$$

where $\widetilde{f^p}(n)$ is the Fourier back transformation of $\widetilde{f^p}(k)$, corresponding to step 8 in the above procedure.

We will also show that cubic grids have the advantage in the computational cost per point compared to the ACG, in addition to being less dense near the nuclei. The points in a cubic grid are uniformly aligned in each direction in space, which allows for the sharing of common factors among the points. In addition, it allows fine screening of effective function pairs and efficient memory access. This advantage will be illustrated through some examples.

A. Smoothness and interpolation

As stated above, a smooth Gaussian function satisfies Eqs. (4) and (9). In practice, for a given interpolation scheme the smoothness of a Gaussian function depends on three parameters: the grid density of grid m, the exponent of the function, and the desired accuracy. The smoothness of a Gaussian function is primarily decided by its exponent and smoother Gaussians have smaller exponents. The divided-difference method[29] has been chosen as the local interpolation scheme [Eq. (4)], which uses polynomials as interpolation function. A detailed study of the relationship among the Gaussian exponent, the grid density, and the accuracy is being reported elsewhere[28] using the method. Based on that study, a cutoff value of 3 for the exponent is chosen for the function pairs that are considered smooth, with the grid density of the fine cubic grid being 6. This cutoff value will put

the majority of the basis function pairs in the smooth category and still yield accurate results. For Fourier interpolation in Eq. (9), a set of parameters have been established with FTC, depending on the grid density. For this study, the grid density for the coarse cubic grid (grid n) is 4, which is more than sufficient for the cutoff value of 3. A more elaborate screening scheme is also being developed to gain efficiency by moving more function pairs to the smooth part.[38]

Before showing some results, we should point out that the mrXC scheme described above is fundamentally different from adaptive grid methods, especially multiresolution analysis (MRA).[39] MRA is a branch of mathematics that shows that a numerical solution or analysis can be efficiently done through a grid that is adaptive to the varying resolution of a function in space, and has been applied to quantum chemistry.[40] In particular, Harrison *et al.*[41] and Yanai *et al.*[42,43] have developed a MRA-based numerical solution to HF and DFT equations that is efficient and free of basis set errors. Separately, Challacomb[44] has designed an adaptive Cartesian grid for XC computation and found that it was difficult to maintain a good accuracy with reasonable efficiency. In this study, we do not formally alter the ACG scheme that has been proven to work well. Rather we attempt to take advantage of the multiresolution nature of the basis functions and integrate the ACG and cubic grids that have predefined resolutions. The scheme also allows further reduction of cost by calculating part of the XC integrals together with the FTC portion of the Coulomb computation.

III. SOME EXAMPLES

A. Accuracy

To demonstrate the accuracy of the mrXC, we have calculated the DFT energies with the Slater functional[30] for two different molecules: alanine with 13 atoms, and the other a chain of five alanine units with 53 atoms. Two Pople-type basis sets are used, 6-31G(df,pd) and 6-31G($2df,2pd$).[45] Final SCF energies are reported with the convergence threshold being 10^{-8} of the root-mean-square (rms) DIIS error.[46] The quadrature for the XC numerical integration is SG-1,[21] a widely used efficient grid that yields good accuracy. The calculation results are listed in Table I.

Because numerical integration with quadrature grids such as SG-1 is an approximation, the accuracy of mrXC should be presented in the context of the errors of SG-1, which is a pruned quadrature ACG starting from 50 radial points and 194 angular points for each atom. To estimate the errors of SG-1 for the testing calculations, we have calculated the DFT energies with a very fine ACG grid with 99

TABLE II. The CPU timings of the electron density and XC matrix formations on the SG-1 quadrature grid with the original standard implementation and mrXC.

Molecule/Basis sets	Number of atoms	Number of basis functions	Original	mrXC	Speedup
(Alanine)$_5$/6-31G(df,pd)	53	869	798	265	3.01
(Alanine)$_5$/6-31G($2df$,$2pd$)	53	1106	1293	404	3.20
Taxol/6-31G(df,pd)	113	1925	922	235	3.92
Taxol/6-31G($2df$,$2pd$)	113	2450	1667	338	4.94

radial points and 590 angular points. The differences between SG-1 and the very fine grid are listed in Table I as the error of SG-1 without using mrXC. The next column lists the differences introduced by mrXC. It is apparent that mrXC does not change the accuracy of the calculations. SG-1 gives errors about a few microhartrees per atom and using mrXC only changes the errors by a few percent, which has little statistical significance.

B. Speedups of the DFT calculation

Two molecules have been selected to illustrate efficiency. The first one is (Alanine)$_5$, a chain of five alanine units. The second one is Taxol, a drug molecule with 113 atoms. Two basis sets are used: 6-31G(df,pd) and 6-31G($2df$,$2pd$). The computer is an AMD 2.1 MHz Athlon MP.

As elucidated in Sec. II, the explicit evaluation of density values and the elements of the XC matrix on the ACG count for almost all of the computational cost (ca. 97%) of the XC part in the current scheme. MrXC is designed to replace the explicit evaluation of smooth functions with the calculations on an even-spaced cubic grid. Table II lists the CPU time of the explicit evaluations and that of the mrXC scheme for the comparison of the efficiency of the two approaches for different combinations of molecules and basis sets. Because the XC matrix needs to be recalculated for each SCF cycles, the CPU times are the average of all the SCF cycles. The full density has been used for each SCF cycle so that the CPU time has little variation between cycles. Table II shows that using mrXC results in three to five times of speed up for these molecules of moderate size. Also the speedup increases when valence functions are added to the basis sets, since those valence functions have relatively small exponents and are therefore smooth.

As discussed in Sec. II the new scheme mrXC has four components for the electron density evaluation: (a) the density contribution from the compact function pairs are calculated explicitly on the ACG [Eq. (1)]; (b) the density values from smooth function pairs are calculated on the coarse cubic grid [step (a) in Sec. II]; (c) those values are transformed to the fine cubic grid through FFT [steps (b)–(d) in Sec. II]; and (d) they are then used to interpolate the density values of smooth function pairs on ACG through the local (divided-difference) interpolation [Eq. (5)]. The components for the computation of the XC matrix are essentially the same with the order of the steps reversed for the calculation with smooth function pairs. The CPU times of those components in the above benchmark calculations are listed in Table III,

with each entry as the sum of the CPU times of the same component for density evaluation and the formation of the XC matrix. For instance, the second column in the table contains the CPU times as the sums of those for density evaluation and the formation of XC matrix arisen from the compact function pairs.

It is at once apparent that the explicit calculation on the ACG still dominates the overall cost, which suggests room for further improvement of the algorithm by moving more functional pairs into the *smooth* category. Secondly, evaluations on the coarse cubic grid dominate the rest of the computational cost as predicted in Sec. II and it increases with the basis set size. It should be pointed out that this part of the calculation can be combined with FTC and therefore becomes "free" for closed-shell systems. On the other hand, the cost for Fourier transforms remains the same since it depends only on the shape of the molecule. The cost of the local interpolation is completely negligible.

C. Efficiency of cubic grid

The mrXC method takes advantage of the efficient electron density and the XC matrix calculations on the cubic grid as implemented in FTC.[11] To illustrate, Table IV lists the CPU times for evaluations of the density and XC matrix with smooth functions. It also lists the number of grid points with the ACG and the coarse cubic grid, respectively. The cubic grid is generated by using the maximum extent of the basis functions in each dimension of the molecule. As one can see, the overall calculation on the cubic grid is much more efficient (300–1200 times per point) than the ACG, even though the former has many times more grid points. This efficient implementation of the cubic grid computation is critical to the efficiency gain shown in Table II. There are at least two factors favoring the use of the cubic grid. One is that the points of an ACG concentrate towards the nuclei, and therefore find more significant basis function pairs, whereas most of the points in a cubic grid are far away from the nuclei. Secondly, the loop structure for the calculation with the basis

TABLE III. The detailed CPU timings for each mrXC component.

Molecule/Basis set	ACG	Smooth	Fourier	Interpolation
(Alanine)$_5$/6-31G(df,pd)	249	12	4	0.4
(Alanine)$_5$/6-31G($2df$,$2pd$)	381	19	4	0.4
Taxol/6-31G(df,pd)	167	46	21	0.5
Taxol/6-31G($2df$,$2pd$)	249	67	21	0.5

TABLE IV. Comparison of efficiency between ACG and the cubic grid for calculations with smooth Gaussian functions.

Molecule/Basis set	CPU time (in s) for the buildup of *smooth* density and XC matrix		Number of grid points in thousands	
	ACG	Cubic grid	ACG	Cubic grid
(Alanine)$_5$/6-31G(df,pd)	549	12	149	3894
(Alanine)$_5$/6-31G($2df,2pd$)	912	19	149	3894
Taxol/6-31G(df,pd)	755	46	317	6156
Taxol/6-31G($2df,2pd$)	1418	67	317	6156

functions can take advantage of the common factors along each direction of the uniform cubic grid, resulting in an efficient implementation.

IV. CONCLUSIONS

Presented here is a new algorithm to increase the efficiency of the numerical integration of the exchange-correlation potential in a DFT calculation. The method, called mrXC, does not alter the formulism of the current standard algorithm based on the atom-centered grid (ACG) nor the accuracy. The scheme involves calculating the XC matrix at two levels of resolution and shifts the calculation associated with smooth function pairs to the cubic grid that is much more efficient than the ACG. It uses FFT and local interpolations to make fast transformations among the grids. A preliminary implementation of it has yielded three to five times the efficiency gain for some test calculations with moderate-sized molecules and basis sets.

V. FUTURE WORK

The mrXC algorithm as it applies to local density approximation has been presented above. However, the most widely used functionals are GGA functionals,[5–7] which are functions of spin densities and their derivatives ($\nabla\rho$). With GGA functionals, the calculation of the XC matrix involves the evaluation of the derivatives of the basis function pairs $\nabla\phi_{\mu\nu}$. To handle them, the formulas in Sec. II could simply be extended to obtain $\nabla\rho$ on the coarse cubic grid, and then make three Fourier transformations for each derivative of the three coordinates x, y, and z to the fine cubic grid. A simpler and more cost-effective way is to recognize that the derivatives of a function pair have a simple form when expanded in Fourier functions:

$$\nabla\phi_{\mu\nu}(r) = \sum_k \bar{k}\phi_{\mu\nu}(k)\chi(k,r). \tag{14}$$

Then, the density derivatives become

$$\nabla\rho(r) = \sum_k \rho(k)\bar{k}\chi(k,r). \tag{15}$$

Equation (15) shows that obtaining the derivatives of the electron density requires only three Fourier transformations from the Fourier space to the fine cubic grid, and there is no need of calculating the derivatives on the coarse cubic grid. It has been shown[28] that the local interpolation is equally

applicable to $\nabla\rho$ as it is to ρ. In addition to making the implementation simpler, it is also efficient as the computational cost for FFT is much smaller than the evaluation on the coarse cubic grid as shown in Table III. For the formation of the XC matrix, the evaluation of the terms such as $\nabla\rho\cdot\nabla\phi_{\mu\nu}$ is required and can be done in momentum space as

$$\nabla\rho(r) \cdot \nabla\phi_{\mu\nu}(r) = \sum_k \nabla\rho(k) \cdot \bar{k}\phi_{\mu\nu}(k), \tag{16}$$

which again requires only three Fourier transformations from the fine cubic grid to the Fourier space and a trivial dot product for each point in the latter. The details of the algorithm and its implementation will be reported in a subsequent paper. It is also possible to make more than two levels of resolution and other fine tunings of the algorithm.

ACKNOWLEDGMENTS

[1] J. Kong, C. A. White, A. I. Krylov *et al.*, J. Comput. Chem. **21**, 1532 (2000).
[2] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chem. Phys. Lett. **230**, 8 (1994).
[3] C. A. White and M. Head-Gordon, J. Chem. Phys. **105**, 5061 (1996).
[4] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems* (MIT, London, 1987).
[5] A. D. Becke, Phys. Rev. A **38**, 3098 (1988).
[6] C. Lee, W. Yang, and R. G. Parr, Phys. Rev. B **37**, 785 (1988).
[7] B. Miehlich, A. Savin, H. Stoll, and H. Preuss, Chem. Phys. Lett. **157**, 6787 (1993).
[8] C. A. White and M. Head-Gordon, J. Chem. Phys. **104**, 2620 (1996).
[9] Y. Shao and M. Head-Gordon, Chem. Phys. Lett. **323**, 425 (2000).
[10] Y. Shao and M. Head-Gordon, J. Chem. Phys. **114**, 6572 (2001).
[11] L. Fusti-Molnar and J. Kong, J. Chem. Phys. **122**, 074108 (2005).
[12] L. Fusti-Molnar and P. Pulay, J. Chem. Phys. **117**, 7827 (2002).
[13] L. Fusti-Molnar and P. Pulay, J. Chem. Phys. **116**, 7795 (2002).
[14] L. Fusti-Molnar, J. Chem. Phys. **119**, 11080 (2003).
[15] M. Challacombe, E. Schwegler, and J. Almlöf, J. Chem. Phys. **104**, 4685 (1996).
[16] E. Schwegler, M. Challacombe, and M. Head-Gordon, J. Chem. Phys. **106**, 9708 (1996).
[17] C. Ochsenfeld, C. A. White, and M. Head-Gordon, J. Chem. Phys. **109**, 1663 (1998).
[18] S. Goedecker, Rev. Mod. Phys. **71**, 1085 (1999).
[19] Y. Shao, C. Saravanan, M. Head-Gordon, and C. A. White, J. Chem. Phys. **118**, 6144 (2003).

[20] X. Li, J. M. Millam, G. E. Scuseria, M. J. Frisch, and H. B. Schlegel, J. Chem. Phys. **119**, 7651 (2003).

[21] P. M. W. Gill, B. G. Johnson, J. A. Pople, and M. J. Frisch, Chem. Phys. Lett. **197**, 499 (1992).

[22] B. G. Johnson, P. M. W. Gill, and J. A. Pople, J. Chem. Phys. **98**, 5612 (1993).

[23] P. M. W. Gill, B. G. Johnson, and J. A. Pople, Chem. Phys. Lett. **209**, 506 (1993).

[24] R. E. Stratmann, J. C. Burant, G. E. Scuseira, and M. J. Frisch, J. Chem. Phys. **106**, 10175 (1997).

[25] R. E. Stratmann, G. E. Scuseria, and M. J. Frisch, Chem. Phys. Lett. **257**, 213 (1996).

[26] S. T. Brown and J. Kong, Chem. Phys. Lett. **408**, 395 (2005).

[27] A. D. Becke, J. Chem. Phys. **88**, 2547 (1988).

[28] S. T. Brown, L. Fusti-Molnar, and J. Kong, Chem. Phys. Lett. **418**, 490 (2006).

[29] C. de Boor, *A Practical Guide to Splines* (Springer-Verlag, New York, 2001).

[30] P. A. M. Dirac, Proc. Cambridge Philos. Soc. **26**, 376 (1930).

[31] S. H. Vosko, L. Wilk, and M. Nursair, Can. J. Phys. **58**, 1200 (1980).

[32] C. W. Murray, N. C. Handy, and G. J. Laming, Mol. Phys. **4**, 997 (1993).

[33] S.-H. Chien and P. M. W. Gill, J. Chem. Phys. (submitted).

[34] V. I. Lebedev, Zh. Vychisl. Mat. Mat. Fiz. **15**, 48 (1975).

[35] V. I. Lebedev, Zh. Vychisl. Mat. Mat. Fiz. **16**, 293 (1976).

[36] V. I. Lebedev, Sib. Math. J. **18**, 132 (1977).

[37] V. I. Lebedev and A. L. Skorokhodov, Dokl. Math. **45**, 587 (1992).

[38] L. Fusti-Molnar, J. Kong, and S. T. Brown (unpublished).

[39] D. F. Walnut, *An Introduction to Wavelet Analysis* (Birkhäser, Boston, 2004).

[40] S. Nagy and J. Pipek, Int. J. Quantum Chem. **84**, 523 (2001).

[41] R. J. Harrison, G. I. Fann, T. Yanai, Z. Gan, and G. Beylkin, J. Chem. Phys. **121**, 11587 (2004).

[42] T. Yanai, G. I. Fann, Z. T. Gan, R. J. Harrison, and G. Beylkin, J. Chem. Phys. **121**, 6680 (2004).

[43] T. Yanai, R. J. Harrison, and N. C. Handy, Mol. Phys. **103**, 413 (2005).

[44] M. Challacombe, J. Chem. Phys. **113**, 10037 (2000).

[45] P. C. Hariharan and J. A. Pople, Theor. Chim. Acta **28**, 213 (1973); M. M. Francl, W. J. Petro, W. J. Hehre, J. S. Binkley, M. S. Gordon, D. J. DeFree, and J. A. Pople, J. Chem. Phys. **77**, 3654 (1982); V. A. Rassolov, M. A. Ratner, J. A. Pople, P. C. Redfern, and L. A. Curtiss, J. Comput. Chem. **22**, 976984 (2001).

[46] P. Pulay, Chem. Phys. Lett. **73**, 393 (1980).