

# Fast and accurate Coulomb calculation with Gaussian functions

László Füsti-Molnár and Jing Kong

*Q-CHEM Inc., Pittsburgh, Pennsylvania 15213*

(Received 8 September 2004; accepted 23 November 2004; published online 9 February 2005)

Coulomb interaction is one of the major time-consuming components in a density functional theory (DFT) calculation. In the last decade, dramatic progresses have been made to improve the efficiency of Coulomb calculation, including continuous fast multipole method (CFMM) and J-engine method, all developed first inside Q-Chem. The most recent development is the advent of Fourier transform Coulomb method developed by Fusti-Molnar and Pulay, and an improved version of the method has been recently implemented in Q-Chem. It 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 size. The result is a much smaller slope in the linear scaling with respect to the molecular size and we will demonstrate through a series of benchmark calculations that it speeds up the calculation of Coulomb energy by several folds over the efficient existing code, i.e., the combination of CFMM and J-engine, without loss of accuracy. Furthermore, we will show that it is complementary to the latter and together the three methods offer the best performance for Coulomb part of DFT calculations, making the DFT calculations affordable for very large systems involving thousands of basis functions. © 2005 American Institute of Physics. [DOI: 10.1063/1.1849168]

## I. INTRODUCTION

Density function 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* simulations with reasonable computational cost. The computation of Coulomb interaction is one of the major components in a DFT calculation, and very often the most time consuming one with generalized gradient approximation (GGA) functionals.<sup>1-3</sup> In this paper, we will present how we have recently reduced the computational cost of the Coulomb interaction by several folds. The methods discussed here apply to calculations using Gaussian functions as basis, the dominant type of basis functions in electronic structure calculations. Due to the linear scaling nature of the DFT calculation, the increase of computational efficiency will be directly translated into the increase of the size of the molecule that can be calculated with the same amount of CPU time.

More than a decade ago before the application of DFT became the mainstream, Coulomb interaction was calculated together with the exact exchange as part of the *ab initio* Hartree-Fock (HF) formalism. The calculation involves the computation of four-center integrals of Gaussians and scales as  $O(N^4)$ , where  $N$  is the number of basis functions. Those integrals are also called electron repulsion integrals (ERIs). When the size of the molecule increases, the cost scales as  $O(N^2)$ , or more precisely  $O(V^2)$ ,  $V$  being the volume of the molecule, when the overlap of two distant Gaussian basis functions is negligible.<sup>4,5</sup> After the advent of more accurate functionals such as Becke's exchange<sup>1</sup> and other GGA functionals, the need for faster evaluation of Coulomb interaction rose since the HF exchange was absent from the formalism. For a DFT calculation with GGA functionals, there are three time-consuming steps. Among them, the numerical integra-

tion of DFT exchange-correlation (XC) can be implemented in linear-scaling fashion, i.e., in  $O(V)$ , using Becke's weighting scheme.<sup>6,7</sup> The CPU time for the diagonalization of the Fock matrix is insignificant compared with that of the computation of either XC or Coulomb for systems with up to several thousand basis functions, although it scales cubically. For larger systems, several linear-scaling algorithms have been developed in recent years.<sup>8-10</sup> Thus Coulomb part became the bottleneck. The first linear-scaling Coulomb algorithm for DFT was proposed and implemented in Q-Chem program<sup>11</sup> in 1996 by White *et al.* whose continues fast multipole method<sup>12-15</sup> (CFMM) was a generalization of Green-gard's fast multipole method<sup>16</sup> for continuously charged matter. CFMM calculates the well-separated charges in  $O(V)$ , and leaves the calculation of the interactions among near-field charges to the conventional ERI methods. To speed up the calculation of near field Coulomb energy, White, Shao, and Head-Gordon developed the J-engine technique<sup>17-19</sup> in Q-Chem that reduced the ERI cost by moving some computationally intensive operations involving nonzero angular momentum basis functions from the quartic to the quadratic loops. The speed-up can be several times with large basis sets. The combination of CFMM and J-engine offers to date the fastest calculation of total Coulomb energy in Gaussian basis that can reproduce the result of the conventional four-center integration at arbitrary precision.

We have recently implemented a Coulomb technology called the Fourier transform Coulomb method (FTC) in Q-Chem, first introduced by Fusti-Molnar and Pulay.<sup>20</sup> It replaces the most expensive part of the Coulomb interaction—the contribution from ERIs involving three or four Gaussian basis functions with small exponents—with an accurate numerical evaluation, using plane waves as the auxiliary

basis.<sup>21</sup> The computational cost of FTC scales in  $O(N^2)$  with respect to the basis function size and in  $O(V \ln V)$  with respect to the size of the molecule. A detailed study for the accuracy and the scaling properties of FTC has been published recently,<sup>22</sup> as well as an essential localization technique for the compact functions (Gaussians with large exponents). More than 17 times of computational efficiency has been gained for the largest calculation (alanine<sub>15</sub> molecule with 3660 basis functions<sup>22,23</sup>) without loss of accuracy. The cost of plane wave part itself was more than two orders of magnitudes smaller than that of the ERI evaluation parts it replaced.

In the above FTC benchmark studies, the rest of the Coulomb interaction was calculated with the conventional ERI techniques, which dominated the cost of the calculation. That cost can be further reduced by the CFMM and J-engine techniques. The reason is that certain types of integrals—the ERIs that involve two or more compact basis functions—cannot be efficiently replaced by plane wave procedures. On the other hand, most of those integrals are for interactions between well-separated charge distributions since the compact functions are well localized in space. The ERI calculation involving those integrals can be replaced with CFMM. The remaining near-field ERIs can be treated effectively with J-engine technique. We will demonstrate that the three methods are complimentary to each other, especially between FTC and CFMM, and the combination of the three leads to a unique and efficient solution of the Coulomb problem.

## II. FTC THEORY AND IMPLEMENTATION

The conventional analytical calculation of Coulomb matrix as part of Fock matrix can be expressed as a summation over four-center repulsion integrals or ERIs of basis functions:

$$J_{\mu\nu} = \sum_{\lambda,\gamma} P_{\lambda,\gamma} \langle \mu\nu | \lambda\gamma \rangle \\ = \sum_{\lambda,\gamma} \int P_{\lambda,\gamma} \phi_{\lambda}(\bar{r}_1) \phi_{\gamma}(\bar{r}_1) \frac{1}{r_{12}} \phi_{\mu}(\bar{r}_2) \phi_{\nu}(\bar{r}_2) d\bar{r}_1 d\bar{r}_2.$$

Its computational cost scales formally in  $O(N^4)$ . FTC method replaces this step by one  $O(V \ln V)$  and two  $O(N^2)$  steps. The  $O(N^2)$  steps also scale in  $O(V)$  asymptotically.

The first step in FTC is to build the electron density on an even spaced cubic grid,

$$\rho(\bar{r}_i) = \sum_{\mu\nu} P_{\mu\nu} \phi_{\mu}(\bar{r}_i) \phi_{\nu}(\bar{r}_i), \quad (2)$$

where  $P_{\mu,\nu}$  is the density matrix element belonging to the  $\phi_{\mu}(\bar{r}_i), \phi_{\nu}(\bar{r}_i)$  basis function pairs. This step scales in  $O(N^2)$  and  $O(V)$  asymptotically because a basis function only overlaps with other basis functions within a certain radius.

The second step is to transform the density values on the grid to the momentum space with fast Fourier transformation, and form the potential  $\nu(\bar{k})$  using the following diagonal operator:<sup>21,24</sup>

$$\nu(\bar{k}) = [1 - \cos(|\bar{k}|D)]/k^2, \quad (3)$$

where  $D$  is the maximum possible distance between the two interacting electrons in the given molecule. It is determined by the molecular structure and the basis set. This potential is then transformed back to the position space to form the Coulomb potential  $\nu(\bar{r}_i)$  on the cubic grid. This step scales as  $O(V \ln V)$ .

The third step is to calculate the Coulomb matrix by numerical integration using  $\nu(\bar{r}_i)$  obtained in the previous step:

$$J_{\mu,\nu} = \sum_i \nu(r_i) \phi_{\mu}(r_i) \phi_{\nu}(r_i) dV. \quad (4)$$

It scales the same as the first step.

Due to its numerical nature, the above integration procedure is not applicable accurately to all the basis functions. In a straightforward application of above procedure, the basis function pair  $\phi_{\mu}\phi_{\nu}$ , which is also a Gaussian when  $\phi_{\mu}$  and  $\phi_{\nu}$  are single Gaussians (Gaussian primitives), needs to be expandable in the given plane wave space defined by the grid density. The expandability is determined by the exponent and the angular momentum of the Gaussian. To identify the ERIs that could be replaced by FTC, **the basis functions were classified into two categories in the original implementation of FTC:<sup>20,22,23</sup> diffuse (denoted as “D”) and compact (denoted as “C”).** The use of the term “diffuse” had a different meaning from that of the diffuse function(s) in a Gaussian basis set. In the latter case, a diffuse basis function was a Gaussian with an extra-small exponent. In the context of FTC, the term diffuse was used to refer to the functions that were expandable in the given plane wave space, versus the  $C$  type functions that were not expandable. Note that the number of expandable basis functions ( $D$ -type functions) is usually much larger than the number of diffuse functions in a basis set. For the given grid, a certain limit  $\eta_{LIMIT}(L)$  was determined for the exponent for a given basis function, or rather the basis function shell, where  $L$  was the angular momentum. A shell refers to the spherical part of a set of basis functions that share the same exponent and angular momentum. A shell was classified as  $D$  type if its exponent was smaller or equal to  $\eta_{LIMIT}(L)$ . Otherwise, it is of  $C$  type. **In this implementation, we introduce a classification that is based on the shell pairs (corresponding to basis function pairs). The limit  $\eta_{LIMIT}(L^2)$  is dependent on the combined angular momentum  $L^2$ . We will consider the basis function pairs associated with the shell-pair expandable in a plane wave space if the sum of the exponents of the two shells is smaller or equal to  $\eta_{LIMIT}(L^2)$ , i.e.,**

$$\eta_1 + \eta_2 \leq \eta_{LIMIT}(L^2), \quad (5)$$

where  $\eta_1$  and  $\eta_2$  are the exponents of the two shells. This consideration is more truthful to Eqs. (2)–(4) since the density is directly dependent on the function pairs. The function pairs meeting the requirement of Eq. (5) are classified as  $DD$  type. FTC can replace the calculation of the ERIs of  $\langle DD | DD \rangle$  type, which are the most expensive part of the Coulomb calculation due to their spatial extension and the number of basis functions associated with them.

In addition, FTC can also replace the ERIs of  $\langle CD|DD \rangle$  type. A shell-pair is of  $CD$  type if  $\eta_1 + \eta_2 \geq \eta_{LIMIT}(L^2)$  but  $\eta_1$  or  $\eta_2$  is smaller than or equal to  $\eta_{LIMIT}(L^2)/2$ . The  $\langle CD|DD \rangle$  type integrals are the second most expensive to calculate after those of  $\langle DD|DD \rangle$  type. Special localization technique is used for the filtered core functions in order to calculate this part efficiently and accurately with plane waves.<sup>22</sup> The shell-pairs that are neither  $DD$  nor  $CD$  are called  $CC$  type. **This scheme of classification removes the need of uncontracting the basis set, which was an inconvenient necessity in the earlier implementation<sup>20,22,23</sup> and classify slightly more integrals to the part of smooth plane waves.** We have also significantly improved the loop structure and the data storage over the original design and achieved an efficiency gain of two to three times for the plane wave part.

It is very difficult for FTC to handle the remaining types of integrals, namely,  $\langle CC|DD \rangle$ ,  $\langle CD|CD \rangle$ ,  $\langle CC|CD \rangle$ ,  $\langle CC|CD \rangle$ , and the cost of those ERIs still dominates while the relative cost of the plane wave part is almost negligible.<sup>22,23</sup> Fortunately, methods already exist in Q-Chem to handle them. Those ERIs contain two or more compact functions, most of which can be handled effectively with CFMM, especially those that have compact functions on both sides of the operator, namely,  $\langle CD|CD \rangle$ ,  $\langle CC|CD \rangle$ , and  $\langle CC|CD \rangle$ . Function pairs containing compact basis functions are well separated in space beyond a few bonds, and majority of the Coulomb interactions among those function pairs can be calculated using CFMM when the molecule contains more than a dozen atoms. The remaining ERIs belong to the near field, and the more expensive ones among them involve basis functions with nonzero angular momentums, such as  $p$ ,  $d$ , and  $f$  functions. Integrals involving those functions can be treated efficiently with J-engine.<sup>18,19</sup> CFMM and J-engine have already been implemented in Q-Chem.<sup>11</sup> It is straightforward to modify the code for CFMM and J-engine to avoid computing integrals involving certain types of shell pairs since the original implementations are based on the loops over shell pairs.

We shall note that FTC, as CFMM and J-engine, is accurate and capable of reproducing the result of the conventional ERI algorithms to arbitrary precision, in contrast to other fast but approximate Coulomb methods developed in recent years, such as Gaussian-augmented plane-wave (GAPW) method<sup>25</sup> and density-fitting method.<sup>26</sup> To illustrate this, we have made side-by-side comparisons of accuracy between the FTC of this implementation and the latter two in Tables I and II. One can see from Table I that the error of GAPW program, which is widely used in solid state physics, is about two to three orders of magnitude larger than that of FTC with the same plane wave basis set (same energy cutoff). The accuracy of FTC is also compared with that of a density-fitting scheme<sup>26</sup> in Table II, which also has errors in the range of several hundred microhartrees for small molecules (the errors in relative energy are smaller), whereas the error of FTC is a few microhartrees or less with the default set of parameters. When the grid density and the cutoff thresholds are set tighter, the error of FTC becomes at least an order of magnitude less, which shows that the error of FTC is controllable. In the benchmark calculations in the

TABLE I. Accuracy comparison between the current implementation of the FTC method and the GAPW method. (All errors are the total energy differences of the final SCF between the given method and analytical integral evaluation scheme using standard Gaussian basis sets. 6-31G\* basis set were used. The GAPW results are obtained from Ref. 25. The SCF convergence criterion was  $10^{-7}$  in Brillouin condition.)

	Errors using GAPW <sup>a</sup> / $\mu E_h$	Errors using FTC/ $\mu E_h$ <sup>b</sup>	Errors using FTC/ $\mu E_h$ <sup>c</sup> (with tighter thresholds)
SF <sub>6</sub>	2670.0	6.14	0.15
CO <sub>2</sub>	-730.0	1.33	0.00
H <sub>2</sub> O	-220.0	0.37	0.01

<sup>a</sup>200 Ry energy cutoff was used.

<sup>b</sup> $4.5a_0^{-1}$  grid density (about 200 Ry) was used for fair comparison with GAPW results. The integral threshold was  $10^{-12}$  in both the ERI and FTC calculations.

<sup>c</sup> $5.0a_0^{-1}$  grid density (about 247 Ry) was used with tighter thresholds in the localization of the core functions (Ref. 22) and in the screening processes. The integral threshold was  $10^{-12}$  and the SCF convergence criterion was  $10^{-7}$  in Brillouin condition.

following section, we have used the default set of parameters with the grid density of  $3.75 a_0^{-1}$  (or 139 Ry) and  $\eta_{LIMIT}(L^2)$  being  $\approx 5 a_0^{-2}$ . The error for this setup is less than  $10^{-6}$  hartree per atom, which is less than the error in the numerical evaluation of DFT exchange-correlation part and sufficient for most applications. **We should note that the total error using  $O(N^2)$  algorithms such as FTC and CFMM increases more slowly with respect to the increase of molecular size than that of the conventional techniques that generate the explicit ERIs because the number of intermediates of the former, in  $O(N^2)$ , is much smaller than that of the latter in  $O(N^4)$ . On computers with fixed precision, less number of operations yields less error in general.**

### III. RESULTS AND DISCUSSIONS

The effects of FTC and that of the combination of all three algorithms—FTC, CFMM, and J-engine—will be shown in terms of the scaling behaviors with respect the size of the molecule and the size of the basis set. A series of three-dimensional carbon clusters (diamonds) are chosen to demonstrate the scaling in system size with the largest cluster having 366 carbon atoms. The C–C bond lengths were 1.52 Å. 6-311G( $d,p$ ) and 6-311G( $df,pd$ ) basis sets<sup>27</sup> are

TABLE II. Accuracy comparison of the current implementation of the FTC method with a recent version of the density fitting scheme. (All errors are total energy differences between the given method and analytical integral evaluation scheme using cc-pVDZ Gaussian basis set.)

	Errors using density fitting / $\mu E_h$ <sup>a</sup>	Errors using FTC/ $\mu E_h$ (with default thresholds) <sup>b</sup>	Errors using FTC/ $\mu E_h$ (with tighter thresholds) <sup>c</sup>
C <sub>2</sub> H <sub>2</sub>	-103.0	0.04	-0.02
C <sub>6</sub> H <sub>6</sub>	-534.0	3.41	-0.26
Glycine (Neutral form)	-550.0	3.52	0.13

<sup>a</sup>Results are obtained from Ref. 26.

<sup>b</sup> $3.75a_0^{-1}$  grid density (about 140 Ry) was used. The integral threshold was  $10^{-12}$  and the SCF convergence criterion was  $10^{-7}$  in Brillouin condition.

<sup>c</sup>See footnote <sup>c</sup> in Table I.



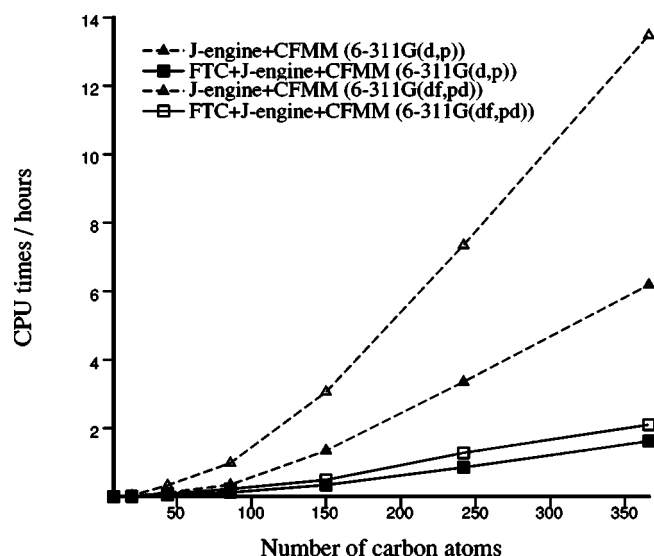


FIG. 1. The reduction of the Coulomb cost by FTC for carbon clusters of different sizes and with two different basis sets.

used to demonstrate the scaling with respect to the size of the basis set. The larger basis set has one additional  $f$  shell on each carbon atom compared to the basis set 6-311G( $d,p$ ). Each Coulomb CPU time has been taken from the second cycle of the self-consistent-field (SCF) iteration and Becke–Lee–Yang–Parr (BLYP) (Refs. 1–3) exchange-correlation functional was used. All the calculations have been done on an IBM Power4 P-680.

Figure 1 depicts the effect of adding FTC to the Coulomb calculation in Q-Chem, which has already been very efficient with the combination of CFMM and J-engine. There are four lines in the figure. The top two dashed lines show the CPU times versus molecular size without FTC for the two different basis sets, and the bottom two solid lines are the CPU times with FTC added. The corresponding numbers are listed in Tables III and IV. One can see immediately that FTC has provided a gain of efficiency for all the carbon clusters except the smallest  $C_8$ , and by several folds as the size of the molecule increases. It speeds up the Coulomb

calculation by about four times for 6-311G( $d,p$ ) and six times for the larger 6-311G( $df,pd$ ) basis for the systems equal to or larger than  $C_{150}$ . For the largest job of  $C_{366}$  with 6-311G( $df,pd$ ) basis set, the Coulomb CPU time for one SCF cycle has gone from 13.5 h to 2.1 h at 9150 basis functions. For smaller jobs, the CPU times are in the range of minutes instead of hours. We note here that the second cycle of the SCF is usually most expensive because the density matrix was nearly full. The average CPU time for an SCF cycle is about 30%–50% less when the difference density matrix is used in later cycles to explore some negligible differences of density matrix between SCF cycles. The job throughput time can be further decreased through parallelization, which is under development in our group.

One can also see that all four lines show linear scaling after the molecule is larger than  $C_{150}$ . The linear scaling of the dashed lines is due to CFMM, and that of the solid lines due to the combination of CFMM and FTC, both of which are linear scaling algorithms. The ones with FTC, however, have a much smaller slope than those without. Furthermore, the calculations with FTC show much less increase of CPU time when a larger basis set is used. Adding one  $f$  shell to each carbon atom amounts to the addition of 2562 basis functions, which has resulted in an increase of CPU time by 2.3 times without FTC, and by a mere 30% with FTC. The better performance of FTC is due to the fact that FTC scales in  $O(N^2)$  with respect to the basis set size, whereas J-engine still scales like  $O(N^4)$ , although it has a much smaller pre-factor than conventional ERI methods.

Before the advent of FTC, CFMM and J-engine have provided a great deal of gain of efficiency for Coulomb calculations over the conventional methods. It would be interesting to see how the two work together with FTC, especially since it has been shown that the cost of the conventional ERIs still dominates with the presence of FTC.<sup>22,23</sup> Our results in Table IV and Fig. 2 demonstrate in details how these techniques work separately and together. Figure 2 shows the effect of the combination of all three methods and the corresponding numbers are listed in Table IV. There are four lines in the figure. The top one is FTC with conventional ERI

TABLE III. Computational timings in minutes for diamond clusters using 6-311G( $d,p$ ) basis set. ( $C_1$  molecular symmetries and superposition of atomic densities guess were used in all calculations. All CPU times belong to the second iteration and the threshold for ERIs was  $10^{-10}$ . All Coulomb and XC timings are considerably smaller for the later iterations as the SCF calculation is converging. The calculations were performed on an IBM P-680 processor. A  $3.75a_0^{-1}$  grid density (about 140 Ry) and a  $10^{-12}$  threshold were used in the FTC calculations.)

Number of atoms	Number of basis functions	FTC+				
		J-engine+CFMM	J-engine+CFMM	FTC part	Diagonalization <sup>a</sup>	XC <sup>b</sup> part
8	144	0.037	0.085	0.063	0.001	0.055
20	360	0.602	0.405	0.190	0.009	0.613
44	792	6.933	2.800	0.620	0.100	4.13
86	1548	20.867	7.217	1.833	0.850	15.57
150	2700	80.817	20.433	4.400	5.883	58.93
242	4356	201.233	51.417	10.483	22.417	96.43
366	6588	371.267	97.083	20.867	77.817	175.98

<sup>a</sup>Standard lapack routine dsyev was used.

<sup>b</sup>Using SG-1 grid (Ref. 28).

TABLE IV. Computational timings (see footnotes in Table III) in minutes for diamond clusters using 6-311G(*df, pd*) basis set.

Number of atoms	Number of basis functions	J-engine+ CFMM	J-engine+ CFMM	FTC+ conventional integrals	FTC+ CFMM	FTC+ J-engine	FTC part	Diagonalization	XC part
8	200	0.135	0.156	0.144	0.144	0.156	0.083	0.002	0.105
20	500	1.98	0.783	1.30	1.30	0.783	0.233	0.025	1.02
44	1100	19.85	4.85	12.48	12.48	4.85	0.817	0.232	5.82
86	2150	59.13	13.37	96.12	21.98	24.18	2.22	2.43	28.13
150	3750	183.43	29.47	351.6	50.63	87.52	5.35	13.22	70.08
242	6050	440.80	76.90	877.3	119.1	255.75	12.40	57.07	140.60
366	9150	808.77	126.07	<sup>a</sup>	<sup>b</sup>	<sup>b</sup>	23.55	184.97	272.58

<sup>a</sup>The calculation is absent because of CPU time limit.

<sup>b</sup>Unnecessary since the corresponding number in column 5 is missing.

algorithms,<sup>4,5</sup> the second line is the combination of FTC and J-engine, the third line reflects the addition of CFMM to the top line, i.e., combination of FTC and CFMM with the conventional methods for the near field, and the bottom line the combination of all three methods. One can see that the combination of FTC with either CFMM or J-engine has yielded significant speed-up of calculations. For the job of  $C_{242}$  at 6-311G(*df, pd*), CFMM has sped up the calculation by 7.4 times, and J-engine 3.4 times. CFMM and J-engine together have provided an efficiency gain of 11.5 times. The reason for the more efficiency gain by CFMM is that it and FTC handle different types of ERIs. As we have explained in the last section, CFMM replaces most of the ERIs that have at least two compact functions when the molecule is large enough, whereas FTC does those that have one compact function at most. On the other hand, J-engine has the most efficiency for those that involve highest angular momentums, most of which are also *D*-type functions. FTC is more efficient than J-engine at handling those ERIs containing three or four *D*-type functions due its better scaling property. Still J-engine provides an efficient route to calculate those ERIs that cannot be covered by FTC and CFMM, majority of which have functions with nonzero angular momentums beyond the simplest molecules and basis sets. Furthermore, J-engine becomes more significant when the molecule is not large enough to provide sufficient separation distances for CFMM. One can see from Table II that CFMM plays no role for the structures up to  $C_{44}$ , whereas J-engine provides up to 2.6 times speed-up with the presence of FTC.

Also listed in the last columns of Tables III and IV are the CPU times of the XC numerical integration and the diagonalization of Fock matrix. In addition to Coulomb, the XC integration and the diagonalization of Fock matrix are the two other major computational components of a DFT calculation with GGA functionals. The diagonalization step has received a great deal of attention for improvement in the last decade<sup>8-10</sup> since it is of a formally scaling of  $O(N^3)$ , whereas the other two scale linearly. Those so-called  $O(N)$  methods, however, have not been applied widely in practice so far, mostly because the Coulomb step has dominated the cost for the application studies using DFT. The cost of XC evaluation is asymptotically linear [ $O(V)$ ] and has been lower than the Coulomb part. It now becomes the most ex-

pensive part for our benchmark calculations. Our group is now working on reducing the cost of XC and the preliminary results show that some parts of XC can be reduced significantly through combination with FTC, due to the numerical nature of the latter.

#### IV. CONCLUSION

FTC is the latest of a string of developments for efficient Coulomb calculations in recent years. It replaces the least efficient part of the previous Coulomb methods—the integrals involving well expandable functions within a predefined plane wave space—with an accurate numerical integration scheme that is far more efficient. The result is a much smaller slope in the linear scaling with respect to the molecular size and it has been demonstrated through a series of benchmark calculations that it speeds up the Coulomb calculation by several folds over the efficient existing Coulomb code, i.e., the combination of CFMM and J-engine. Furthermore, we have shown that it is complementary to the latter and together the three methods offer the best performance for

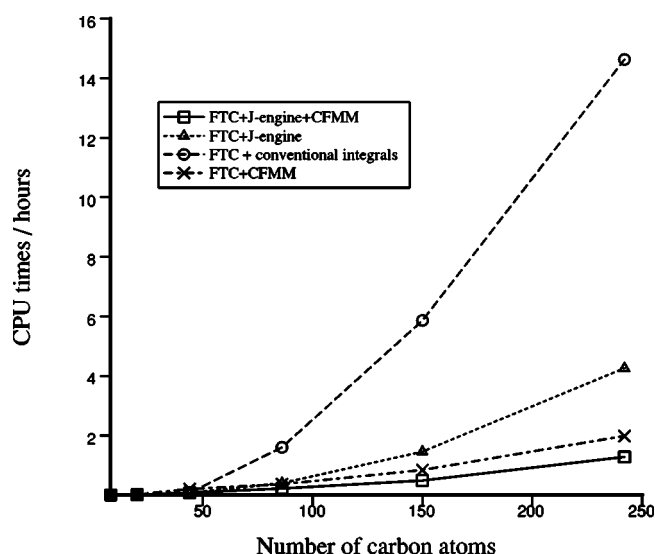


FIG. 2. Effect of the CFMM and the J-engine techniques in combination with FTC on the Coulomb computation for carbon clusters with different sizes and with 6-311G(*df, pd*) basis set.

Coulomb part of DFT calculations with no loss of accuracy, making the DFT calculations affordable for very large systems involving thousands of basis functions.

## ACKNOWLEDGMENTS

The authors wish to thank Dr. Shawn T. Brown and Dr. Yihan Shao for helpful discussions and technical assistance, and Alen Ford for providing a program for generating the structures of the carbon clusters. L.F.-M. wishes to thank Professor Peter Pulay for earlier discussion.

- <sup>1</sup>A. D. Becke, Phys. Rev. A **38**, 3098 (1988).
- <sup>2</sup>C. Lee, W. Yang, and R. G. Parr, Phys. Rev. B **37**, 785 (1988).
- <sup>3</sup>B. Miehlich, A. Savin, H. Stoll, and H. Preuss, Chem. Phys. Lett. **157**, 6787 (1993).
- <sup>4</sup>T. R. Adams, R. D. Adamson, and P. M. W. Gill, J. Chem. Phys. **107**, 124 (1997).
- <sup>5</sup>M. Head-Gordon and J. A. Pople, J. Chem. Phys. **89**, 5777 (1988).
- <sup>6</sup>A. D. Becke, J. Chem. Phys. **88**, 2547 (1988).
- <sup>7</sup>E. Stratmann, G. E. Scuseria, and M. J. Frisch, Chem. Phys. Lett. **257**, 213 (1996).
- <sup>8</sup>S. Goedecker, Rev. Mod. Phys. **71**, 1085 (1999).
- <sup>9</sup>Y. Shao, C. Saravanan, M. Head-Gordon, and C. A. White, J. Chem. Phys. **118**, 6144 (2003).
- <sup>10</sup>X. Li, J. M. Millam, G. E. Scuseria, M. J. Frisch, and H. B. Schlegel, J. Chem. Phys. **119**, 7651 (2003).
- <sup>11</sup>J. Kong, C. A. White, A. I. Krylov *et al.*, J. Comput. Chem. **21**, 1532 (2000).
- <sup>12</sup>C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chem. Phys. Lett. **230**, 8 (1994).
- <sup>13</sup>C. A. White and M. Head-Gordon, J. Chem. Phys. **105**, 5061 (1996).
- <sup>14</sup>C. A. White and M. Head-Gordon, Chem. Phys. Lett. **257**, 647 (1996).
- <sup>15</sup>C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chem. Phys. Lett. **253**, 268 (1996).
- <sup>16</sup>L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems* (MIT, London, 1987).
- <sup>17</sup>C. A. White and M. Head-Gordon, J. Chem. Phys. **104**, 2620 (1996).
- <sup>18</sup>Y. Shao and M. Head-Gordon, Chem. Phys. Lett. **323**, 425 (2000).
- <sup>19</sup>Y. Shao and M. Head-Gordon, J. Chem. Phys. **114**, 6572 (2001).
- <sup>20</sup>L. Fusti Monar and P. Pulay, J. Chem. Phys. **117**, 7827 (2002).
- <sup>21</sup>L. Fusti Monar and P. Pulay, J. Chem. Phys. **116**, 7795 (2002).
- <sup>22</sup>L. Fusti Monar, J. Chem. Phys. **119**, 11080 (2003).
- <sup>23</sup>L. Fusti Monar and P. Pulay, J. Mol. Struct.: THEOCHEM **666**, 25 (2003).
- <sup>24</sup>E. L. Pollock and J. Glosli, Comput. Phys. Commun. **95**, 93 (1996).
- <sup>25</sup>M. Krack and M. Parinello, Phys. Chem. Chem. Phys. **2**, 2105 (2000).
- <sup>26</sup>F. R. Manby, P. J. Knowles, and A. W. Lloïd, J. Chem. Phys. **115**, 9144 (2001).
- <sup>27</sup>R. Krishnan, J. S. Binkley, and J. A. Pople, J. Chem. Phys. **72**, 650 (1980).
- <sup>28</sup>P. M. W. Gill, B. G. Johnson, and J. A. Pople, Chem. Phys. Lett. **209**, 506 (1993).