

# Interpolation density values on a cartesian grid: Improving the efficiency of Lebedev based numerical integration in Kohn–Sham density functional algorithms

Shawn T. Brown<sup>1</sup>, László Füsti-Molnár, Jing Kong<sup>\*</sup>

*Q-Chem, Inc., 5001 Baum Blvd., Suite 690, Pittsburgh, PA 15213, USA*

Received 30 September 2005; in final form 21 October 2005

Available online 28 November 2005

## Abstract

Most modern Kohn–Sham density functional theory algorithms utilize atom-centered numerical quadrature techniques for integration. To take advantage of the Fourier Transform Coulomb method in the Q-CHEM package, which utilizes an evenly-spaced Cartesian grid to perform highly efficient numerical integration, divided difference interpolation is explored as a means of translating the electron density and its gradients from the Cartesian grid to atom-centered grid points. Aspects of accuracy, error control through the use of the grid density, and efficiency estimations are explored and the method is shown to provide an accurate means to link the FTC method and numerical DFT integration.

© 2005 Elsevier B.V. All rights reserved.

## 1. Introduction

It is well known that Kohn–Sham density functional theory (KS-DFT) is by far the most widely applied method in the toolkit of quantum chemical calculations. Many efforts have been made to improve the efficiency of KS-DFT algorithms in an attempt to ever increase the applicability of the method to larger molecular systems [1–5].

Recently, Füsti-Molnár et al. [6–8] has put forth the Fourier Transform Coulomb (FTC) method and implemented it in conjunction with the J-Engine and Continuous Fast Multipole (CFMM) Method in the Q-CHEM [9] computational chemistry package [1]. The method utilizes a Cartesian based grid to perform a numerical integration procedure for certain basis function pairs, taking advantage of the fact that many of the pairs are sufficiently

smooth to be described in this manner. There are two advantages to the FTC method that could be applicable to KS-DFT integration techniques: the division of Gaussian shell pairs into ‘smooth’ and ‘sharp’ categories based on their combined exponent and other more sophisticated methods, and an incredibly fast algorithm for computing the density of a molecule on a Cartesian grid. It is also important to note that if the density computed for the determination of the Coulomb integrals is used in the KS-DFT procedure, great increases in efficiency could readily be gained. While most modern KS-DFT algorithms involve solving the so-called exchange-correlation functional through numerical integration over atom-centered grids, combining concepts from FTC could provide large improvements in efficiency.

Using Cartesian grids for KS-DFT integration has been attempted before by Challacombe [10] with the HiCu method. The numerical integration in this technique used an adaptive cartesian grid and it was found that a large number of points were needed near the atomic centers to achieve decent accuracy. So it would be advantageous to try developing a method that uses a mixture of the atom-centered and the Cartesian grid.

<sup>\*</sup> Corresponding author.

*E-mail addresses:* [stbrown@psc.edu](mailto:stbrown@psc.edu) (S.T. Brown), [jkong@q-chem.com](mailto:jkong@q-chem.com) (J. Kong).

<sup>1</sup> Present address: Pittsburgh Supercomputing Center, Carnegie Mellon University, Pittsburgh, PA 15239, USA.

A possible road to the combination of Cartesian and atom-centered grids is to translate information between the two. To be more relevant to the current problem, it is possible to take the electronic density already determined by the FTC procedure for the same shell pairs to approximate the density at points on the atom-centered grid. Three-dimensional polynomial interpolation over evenly-spaced sampling points provides a framework for translation. It is also important to be able to handle other values of the density, such as gradients, so that the entire spectrum of functionals are available. Albeit in a different context, a similar approach is used by Lee et al. [11] to perform a real-space multigrid integration procedure in solid-state KS-DFT to facilitate the use of a coarse grid.

Further complications arise in determining the exchange-correlation ‘Fock’ matrix elements, for the total density is needed and therefore direct mixing of the Cartesian and atom-centered grids is difficult. This problem is addressed in recent work by Kong et al. [12] unveiling the multiresolution exchange-correlation (mrXC) method which solves this difficulty using a similar approach as is taken in this work.

To assess the viability of such an idea, the accuracy of three-dimensional interpolation as it applies to approximating the electron density of molecular systems in a finite basis is explored. In order to get a handle on the ranges of exponents in which interpolation is accurate preliminary study of its use in one-dimension for Gaussian basis functions is undertaken. Then actual molecular densities and their gradients will be determined with three-dimensional polynomial interpolation with an eye toward accuracy and the sheer number of shell pairs that can be considered.

## 2. Theory and implementation

### 2.1. Computing electron density in KS-DFT

The computation of the electron density and its gradient in a finite basis has been described elsewhere [13] and will be reviewed briefly here to facilitate the following discussion. The molecular exchange-correlation energy is determined by integrating the functional of the density and its variables (e.g. gradients)

$$E^{\text{XC}} = \int f(\rho_\alpha, \rho_\beta, \gamma_{\alpha\alpha}, \gamma_{\beta\beta}, \gamma_{\alpha\beta}) \mathrm{d}r, \quad (1)$$

$$\gamma_{\alpha\alpha} = |\nabla\rho_\alpha|^2, \quad \gamma_{\alpha\beta} = \nabla\rho_\alpha \cdot \nabla\rho_\beta. \quad (2)$$

In practical KS-DFT computations, the electron density is expanded in a finite set of basis functions so that:

$$\rho_\alpha = \sum_{\mu\nu} P_{\mu\nu}^\alpha \phi_\mu \phi_\nu, \quad (3)$$

$$\nabla\rho_\alpha = \sum_{\mu\nu} P_{\mu\nu}^\alpha \nabla(\phi_\mu \phi_\nu), \quad (4)$$

with similar formulas for  $\rho_\beta$ .  $P_{\mu\nu}^\alpha$  is an element from the spin density matrix, which translates the finite basis functions to the so-called Kohn–Sham orbitals that are determined by a self-consistent procedure.

The exact form of the functional used in the integral from Eq. (1) is at present unknown and several approximate forms have been proposed [14–17]. These functionals are of such a complicated nature that generally analytic integration is not practical or even possible, and therefore numerical grid techniques are employed. The most common method, and the one currently used in the Q-CHEM package [9], is a decomposition of the molecular integral to several atom-centered integrals using some weighting scheme based on atomic loci [18]. The atomic integrals are then evaluated by separation into radial and angular portions, where the radial part can be approximated via several types of quadrature [5,19,20] and the angular part is handled with Lebedev spherical methods [21–24]. So in effect, several atom-centered quadrature grids are defined for the molecule and the value of the functional is sampled at each of these grid points.

It is apparent that in order for this numerical quadrature to proceed, the electron density and its gradients must be determined at each point on the grid. Once the individual basis function values are sampled at given grid point, the formulas for the density and its gradient at that point are easily shown to be:

$$\rho_\alpha(r_{ig}) = \sum_{\mu\nu} P_{\mu\nu}^\alpha \phi_\mu(r_{ig}) \phi_\nu(r_{ig}), \quad (5)$$

$$\nabla\rho_\alpha(r_{ig}) = \sum_{\mu\nu} P_{\mu\nu}^\alpha \nabla(\phi_\mu(r_{ig}) \phi_\nu(r_{ig})). \quad (6)$$

Computing these values on an atom-centered numerical grid is carried out essentially by computing the value of the basis functions at grid point and then contacting these values with the density matrix elements. Since the density is expanded in terms of atom-centered functions, these types of numerical grids do an excellent job for a relatively modest number of points [4]. In terms of algorithm efficiency though, it is difficult to make this scheme efficient. In contrast, an evenly-spaced Cartesian grid, while consisting of far greater number of integration points, can be made highly efficient because each point lies along the same Cartesian coordinates as several others. So the cubature can be written into an algorithm in such a way as to reuse the  $x$ ,  $y$ , and  $z$  coordinates for several density values.

In order to compute the value of the Coulomb energy for a molecular system, the FTC method needs to compute the electron density on an evenly-spaced Cartesian grid, and can be formulated to efficiently determine the gradient of the density on the same grid. It would certainly be advantageous to reuse these values to increase the efficiency of the atom-centered algorithm. In order to accomplish this, a method must be developed that translates the information from the Cartesian grid.

### 2.2. Divided difference interpolation

The translation of values from evenly-spaced Cartesian grid points to the atom-centered grid is facilitated through

the use of interpolation. The electron density (and its gradients) are viewed as a three-dimensional function, Cartesian values can be thought of as series ‘knots’ to be used in determining three-dimensional interpolating polynomials that can approximate the values within its boundaries. (i.e. within the box formed by the Cartesian points) Since the points from the atom-centered numerical grid are contained within, interpolation from the sampling of the density on the Cartesian grid can be used.

There are several interpolation schemes that can be used for this purpose. The requirements on a method for the current purposes would be: reasonable accuracy so as not to destroy the accuracy of the overall numerical integration; uses only the value of the functions at the interpolation ‘knots’ so that analytic gradients of the functions need not be evaluated; use evenly-spaced knots since our interpolating points come from such a grid; understandable error control that can be manipulated through one or two parameters, and the algorithm for interpolation should be very efficient. It was determined that divided difference polynomial interpolation was the technique that best fits these requirements.

Polynomial interpolation is an approximate method that represents a function through a finite order polynomial obtained through sampling the function at known points [25]. Divided difference interpolation uses a polynomial derived from a numerical Taylor expansion about an initial coordinate.

$$f(x) \approx f[x_0] + (x - x_0) * f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] + \dots (x - x_0) \dots (x - x_{n-1})f[x_0, \dots, x_n], \quad (7)$$

with

$$f[x_0, x_1, \dots, x_n] = \frac{f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}]}{x_n - x_0}. \quad (8)$$

This expression can be computed via regression in a very efficient fashion with the generalization to three dimensions being easy to derive and implement from there.

This interpolation method fits all of the criteria mentioned above: it can be very accurately applied to Gaussian-like functions; only the functions need to be evaluated at each interpolation knot; it can be formulated with evenly-spaced samplings to coincide with the FTC method; it will be shown that the error can be controlled by one parameter; and it can be computed very efficiently.

A word about error control, the form of the error from divided difference interpolation is well known.

$$E(x) = \frac{1}{(n+1)!} f^{(n+1)}(\eta) \pi(x), \quad (9)$$

where  $n$  is the order of interpolation,  $f^{(n+1)}(\eta)$  is the  $n+1$  derivative of the function interpolated over an interval  $\eta$  inside the interval  $x_0$  to  $x_n$ , and  $\pi$  is a product function of differences in the distance between knots:

$$\pi(x) = (x - x_0)(x - x_1) \dots (x - x_n). \quad (10)$$

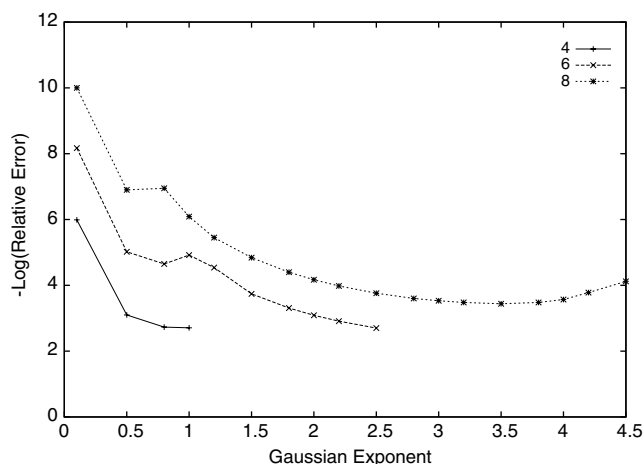
This shows that there are three factors that control the error in this interpolation scheme: the shape of the derivative of the function, the order of interpolation, and the distance between interpolation points. The shape of a Gaussian function and its derivatives will depend mostly on the exponent that is used. Fig. 1 shows the error in interpolating one-dimensional s-, p-, and d-type Gaussian functions with varying exponents and interpolation orders. The figures show that for a large range of exponents, using interpolation is effective and that as angular momentum is added (which in turns mimics the derivatives of the functions) the error profile changes little. Given the similarity in the shapes of higher derivative functions, there is no reason to believe that this will not be the case for even higher angular momenta. It also shows that the range at which a given interpolation is accurate increases with the order of the interpolation, as expected. This would indicate that scaling a threshold parameter for screening the shell pairs for interpolation should be scaled as well. So the other two factors can be controlled through the use of one parameter. Since decreasing the distance between ‘knots’ and increasing the order will achieve the best accuracy for the number of Cartesian points, the grid density will be used. The grid density describes the number of Cartesian points along the unit cell of the grid, and is ideal for use as the error control parameter.

### 2.3. Implementation

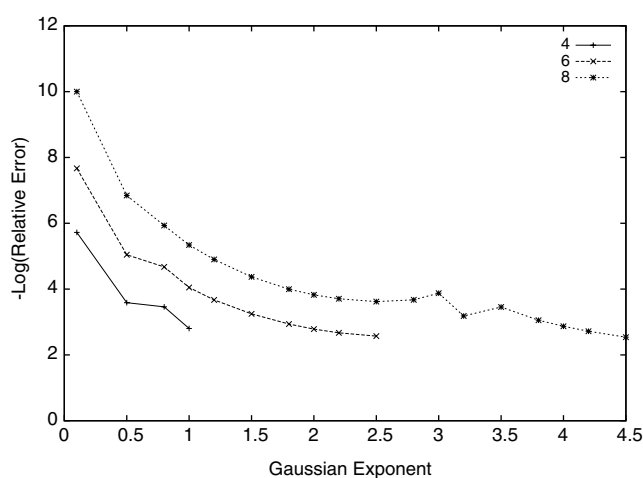
A prototype code for interpolating three-dimensional electron densities was implemented in the Q-CHEM 2.0 quantum chemistry program suite. A modified version of the existing KS-DFT code was made in order to yield meaningful results for testing this new method.

Since the divided-difference interpolation cannot be used to describe Gaussian functions with large exponents (see Fig. 1) it is necessary for us to partition the shell pair lists. Since much work has been done on the use of the Cartesian grid for integration through the FTC method by Füsti-Molnár et al. [6], we have adopted the same exponent cutoff parameter used there scaled to the proper grid density (for all cases in this study, the thresholds were 2.29, 5.15, 9.16 for grid densities of 4, 6, and 8, respectively). Only contracted basis functions were screened, i.e. the primitive Gaussian functions with the largest exponent is used. Examination of the numerical statistics of the data from Fig. 1 bears out the validity of this approach, for it can be seen that increasing the order of the interpolation does indeed increase the applicability of interpolation in approximating Gaussian functions. Shell pairs that can be handled via interpolation (i.e. with the produce of the Gaussian exponents below the prescribed threshold) will be termed ‘smooth’, while those that are not will be ‘sharp’. The total density and its gradient can be expressed then as a sum of values determined with the shell pairs in each region.

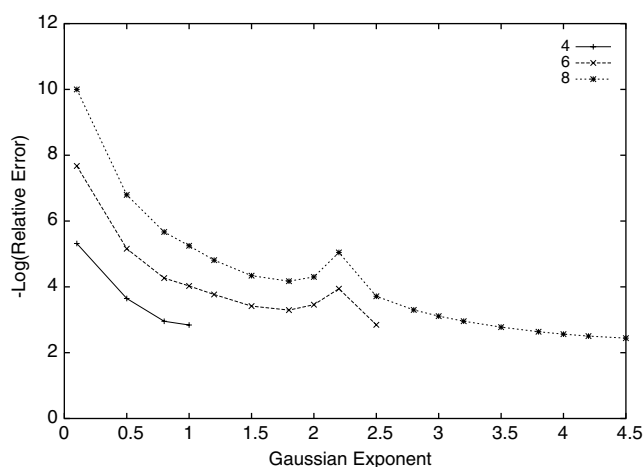
The basic algorithm consists of first computing the positions of all of the points on the atom-centered grid,



(a) s-type function



(b) p-type function



(c) d-type function

Fig. 1. The negative log of the relative error due to approximation of one-dimensional Gaussian functions of varying angular momentum as a function of the exponents with different orders of interpolation.

computing to which Cartesian box the points belong, and then interpolating the electron density at each point for the ‘smooth’ shell pairs. These values are then added to the ‘sharp’ density values from Q-Chem’s original KS-

DFT algorithm and the rest of the procedure continues as before.

### 3. Results and discussion

#### 3.1. Cyclohexane and taxol

To show the accuracy and effectiveness of the interpolation scheme in varying situations, benchmark computations were performed on cyclohexane and taxol molecules with varying basis sets using the Becke88 exchange functional [26]. This functional was chosen for its ease of implementation and to display that both the density and its gradient are handled with this new procedure. The results of these benchmarks are displayed in Table 1. There are several observations of note presented here.

Just a note on the measures of accuracy used in the benchmarks. The absolute value of the relative error in the electron count (i.e. the trace of the density matrix) is used as the main gauge to the accuracy. The most relevant difference to examine here is that between the electron count of the original algorithm and the new interpolation based one, which shows the accuracy of approximating the values on the atom-centered grid. The overall accuracy of the integration procedure would be determined by the difference between the total number of electrons and the value computed on the overall numerical grid. To say the interpolation procedure is an accurate approximation would be to say that it reproduces the density on the molecular grid, or at least at a lower error than the overall numerical integration. To show the accuracy of the gradients of the density, the relative error in the functional values on the molecular grid are used.

First, error control is indeed achieved through the use of the grid density. Even at grid density of  $4 a_0^{-1}$ , the relative error in the electron count is shown to be significantly less than relative error in the actual numerical integration procedure. As the atomic-centered grid size is increased, it appears as the interpolation procedure remains more accurate as well. There is a slight decrease in accuracy when the angular momentum of the basis set is increased which reflects the increased number of shell pairs handled via interpolation, but it should be noted that once again the error is at least one to two orders of magnitude smaller than the integration procedure itself. Increasing the size of the molecule causes a slightly better performance. The functional values computed with the density and the gradient of the density are displayed to show that both values are very effectively represented via our new interpolation scheme.

In the second column of Table 1 is the percentage of the values interpolated in the new procedure. This number is comprised of the number of basis function pairs multiplied by the number of grid points. This number is significant, because it roughly shows the ideal speed-up that this procedure can exhibit. With measurements of between 46% and 80% of the values being interpolated, it is quite clear the

Table 1  
Statistics for computing the electron density with divided difference of varying orders for cyclohexane and taxol with the Becke88 functional

Grid density	Percentage of shell pairs interpolated	Electron count	Relative error in electron count	Functional value	Relative error in functional energy
Cyclohexane (12 atoms, 48 electrons) Basis: 6-311G* (150 functions)					
SG-1 Grid (49344 points)					
4	64%	48.001220	1.9E–6	–32.923230	4.8E–7
6	64%	48.001127	6.4E–8	–32.932313	2.5E–8
8	64%	48.001130	6.0E–10	–32.923214	1.0E–10
No interp		48.001130		–32.923214	
Cyclohexane (12 atoms, 48 electrons) Basis: 6-31G(2df, 2pd) (354 functions)					
SG-1 Grid (49,344 points)					
4	46%	48.001242	1.5E–6	–33.114880	2.2E–6
6	80%	48.001184	2.9E–7	–33.114803	1.4E–7
8	80%	48.001170	3.1E–8	–33.114808	4.2E–8
No interp		48.001170		–33.114808	
75 Radial/302 angular grid (308,040 points)					
4	46%	48.000003	1.5E–6	–33.114608	2.1E–6
6	80%	47.999940	2.1E–7	–33.114531	2.3E–7
8	80%	47.999929	2.1E–8	–33.114540	1.6E–8
No interp		47.999930		–33.114539	
Taxol (113 atoms, 452 electrons) Basis: 6-31G* (1,032 functions)					
SG-1 Grid (311,328 points)					
4	76%	452.021189	1.7E–7	–364.502952	3.2E–7
6	76%	452.021108	1.1E–8	–364.502822	3.6E–8
8	76%	452.021111	4.4E–9	–364.502835	2.0E–10
No interp		452.021113		–364.502835	

The electron count is defined as the trace of the electron density matrix and is used here as a guide to the accuracy in integrating the density over the grid. Also included is the relative error in the integrated functional energy to show that the interpolated density values do not seriously effect the validity of the numerical integration procedure.

savings possible in such an approach. The percentage of values interpolated increases with grid density only in one case and is due to the fairly loose separation of the exponents in a given basis set. This would seem to indicate that a single exponent cutoff can be used for all orders of interpolation. As expected, with increased basis set size, a rise in the percentage of values in which interpolation is applicable is observed because more basis functions of low exponents are included.

### 3.2. Other examples

Table 2 contains results from benchmarking the method on several systems of varying sizes, functionals and basis sets to examine the effects of interpolation on the convergence of some *real-world* examples. These computations were all run until the convergence in the DIIS error vector reached below  $10^{-5}$ . Displayed is the relative error in the

final energy and electron count, and as the table shows, the method is very accurate across the range of systems. Granted, this is not a fully representative set of molecular systems, but it yields reason to believe the method is accurate enough to be applied in types of molecules in which KS-DFT is widely applicable.

Also in the table is the percentage of values that are interpolated (in this case the percentage is for the last iteration, but the percentage is fairly constant throughout the computation). Through the range of systems, there is an excellent coverage by the interpolation methods. Also shown is the percentage of time taken to compute the density and its gradient on the molecular grid for the ‘sharp’ shell pairs. This timing reflects the ideal speed-up we can expect from this procedure. If the functionals required only the density on the grid (i.e. local functionals) then given the fact that we have these values given from the FTC methods during Coulomb integrations, this ideal would more or less

Table 2  
Relative errors in the final energies of some representative molecules with various basis functions and functionals

Molecule/method	Relative error in converged final energy (percent basis function pairs interpolated)		
Grid density	4	6	8
Glutamine 6-311G/PW91	6.2E–7 (41%) <b>48%</b>	1.1E–8 (41%) <b>48%</b>	3.0E–10 (41%) <b>48%</b>
Alanine15 6-311G(df,pd)/BLYP	4.1E–7 (74%) <b>15%</b>	5.9E–9 (76%) <b>14%</b>	7.7E–10 (76%) <b>14%</b>
Diamond (86 atoms) 6-311(df, pd)/EDF1	8.7E–8(71%) <b>13%</b>	5.1E–9 (71%) <b>13%</b>	1.1E–10 (71%) <b>13%</b>

In parentheses is given the percentage of basis function pairs handled via interpolation and the number in bold are the percentage of time to compute the electron density and its gradient from the ‘sharp’ basis function pairs.

be reached. However, for gradient corrected functionals, such as the ones displayed here, the speed-up will be less than ideal, for there must be some method devised to represent the gradient of the density on the evenly-spaced Cartesian grid in a very efficient manner. Research is currently underway to achieve this goal. Additional improvement can be achieved by using more sophisticated techniques to judge the smoothness (expandability) of the shell pairs. Such a technique is already implemented in the Coulomb part of Q-CHEMS FTC package and shows a significant amount of additional shell pairs can be classified as ‘smooth’. Nonetheless, the current results clearly show that interpolation can be used to approximate both types of values very accurately in modern KS-DFT implementations.

#### 4. Concluding remarks

The use of interpolation for approximating density and gradient values on the atomic-centered numerical integration grid in KS-DFT yields some very promising results. The percentage of function pairs that can be treated by this method as ‘smooth’ is well over 50% and in some cases as high as 80%, and can be further increased. The method also displays good accuracy and error control through the use of the grid density. It is definitely a good candidate for full implementation in a modern KS-DFT algorithm and currently work has been done for local functionals [12] in the Q-CHEM quantum chemistry package to make this a reality.

#### Acknowledgments

This work was funded by the NIH SBIR program. The authors thank Drs. Yihan Shao and Emil Proynov for their helpful support and discussions. The authors also thank Dr. Thomas Furlani and the Center for Computational Research of the University of New York at Buffalo for computing and technical resources. S.T.B. also thank Prof.

Henry F. Schaefer III and his research group for helpful discussions. L.F.M. thank Prof. Peter Pulay for his many helpful initial discussions.

#### References

- [1] L. Füsti-Molnár, J. Kong, *J. Chem. Phys.* 122 (2005) 074108.
- [2] G.E. Scuseria, *J. Phys. Chem.* 103 (1999) 4782.
- [3] S.T. Brown, J. Kong, *Chem. Phys. Lett.* 408 (2005) 395.
- [4] P.M.W. Gill, B.G. Johnson, J.A. Pople, *Chem. Phys. Lett.* 209 (1993) 506.
- [5] P.M.W. Gill, S. Chien, *J. Comput. Chem* 24 (2003) 732.
- [6] L. Füsti-Molnár, *J. Chem. Phys.* 119 (2003) 11080.
- [7] L. Füsti-Molnár, P. Pulay, *J. Chem. Phys.* 116 (2002) 7795.
- [8] L. Füsti-Molnár, P. Pulay, *J. Chem. Phys.* 117 (2002) 7827.
- [9] J. Kong, C.A. White, A.I. Krylov, C.D. Sherrill, R.D. Adamson, T.R. Furlani, M.S. Lee, A.M. Lee, S.R. Gwaltney, T.R. Adams, C. Ochsenfeld, A.T.B. Gilbert, G.S. Kedziora, V.A. Rassolov, D.R. Maurice, N. Nair, Y. Shao, N.A. Besley, P.E. Maslen, J.P. Dombroski, H. Daschel, W. Zhang, P.P. Korambath, J. Baker, E.F.C. Byrd, T.V. Voorhis, M. Oumi, S. Hirata, C.-P. Hsu, N. Ishikawa, J. Florian, A. Warshel, B.G. Johnson, P.M.W. Gill, M. Head-Gordon, J.A. Pople, *J. Comput. Chem.* 21 (2000) 1532.
- [10] M. Challacombe, *J. Chem. Phys.* 113 (2000) 10037.
- [11] I.-H. Lee, Y.-H. Kim, R.M. Martin, *Phys. Rev. B* 61 (2000) 4397.
- [12] J. Kong, S.T. Brown, L. Füsti-Molnár, *J. Phys. Chem.*, submitted.
- [13] P.M.W. Gill, B.G. Johnson, J.A. Pople, M.J. Frisch, *Chem. Phys. Lett.* 197 (1992) 499.
- [14] B.G. Johnson, P.M.W. Gill, J.A. Pople, *J. Chem. Phys.* 98 (1993) 5612.
- [15] A.D. Becke, *J. Chem. Phys.* 98 (1993) 1372.
- [16] P.J. Stephens, F.J. Devlin, C.F. Chabalowski, M.J. Frisch, *J. Phys. Chem.* 98 (1994) 11623.
- [17] A.D. Becke, *J. Chem. Phys.* 112 (2000) 4020.
- [18] A.D. Becke, *J. Chem. Phys.* 88 (1988) 2547.
- [19] C.W. Murray, N.C. Handy, G.J. Laming, *Mol. Phys.* 4 (1993) 997.
- [20] O. Truetler, R. Ahlrichs, *J. Chem. Phys.* 102 (1994) 346.
- [21] V.I. Lebedev, Z. Vychisl, *Mat. Mat. Fiz.* 15 (1975) 48.
- [22] V.I. Lebedev, Z. Vychisl, *Mat. Mat. Fiz.* 16 (1976) 293.
- [23] V.I. Lebedev, *Sib. Mat. Zh.* 18 (1977) 132.
- [24] V.I. Lebedev, A.L. Skorokhodov, *Russ. Acad. Sci. Dokl. Math.* 45 (1992) 587.
- [25] C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, New York, 2001.
- [26] A.D. Becke, *Phys. Rev. A* 38 (1988) 3098.