Communication: Linear-expansion shooting techniques for accelerating self-consistent field convergence

Yan Alexander Wang, Chi Yung Yam, Ya Kun Chen, and GuanHua Chen

Additional information on J. Chem. Phys.

Communication: Linear-expansion shooting techniques for accelerating self-consistent field convergence

Yan Alexander Wang,[1,a)] Chi Yung Yam,[2] Ya Kun Chen,[1] and GuanHua Chen[2,a)]

[1]*Department of Chemistry, University of British Columbia, Vancouver, BC V6T 1Z1, Canada*
[2]*Department of Chemistry, University of Hong Kong, Hong Kong*

Based on the corrected Hohenberg-Kohn-Sham total energy density functional [Y. A. Zhang and Y. A. Wang, J. Chem. Phys. **130**, 144116 (2009)], we have developed two linear-expansion shooting techniques (LIST)– direct LIST (LISTd) and indirect LIST (LISTi), to accelerate the convergence of self-consistent field (SCF) calculations. Case studies show that overall LISTi is the most robust and efficient algorithm for accelerating SCF convergence, whereas LISTd is advantageous in the early stage of an SCF process. More importantly, LISTi outperforms Pulay's direct inversion in the iterative subspace (DIIS) [P. Pulay, J. Comput. Chem. **3**, 556 (1982)] and its two recent improvements, energy-DIIS [K. N. Kudin, G. E. Scuseria, and E. Cancès, J. Chem. Phys. **116**, 8255 (2002)] and augmented Roothaan-Hall energy-DIIS [X. Hu and W. Yang, J. Chem. Phys. **132**, 054109 (2010)]. © *2011 American Institute of Physics*. [doi:10.1063/1.3609242]

Ever since the beginning of quantum theory, how to efficiently solve the Schrödinger equation for complex systems has been the major endeavor of theoreticians in both computational physics and chemistry communities.[1] With the invention of the Hartree-Fock method (HFM) and density-functional theory (DFT), the computational task has been shifted to identifying the converged solution of an iterative self-consistent field (SCF) system,[1–5] because the formulations require the exact solution (*before* it is known) in the construction of the Hamiltonian. Consequently, in an SCF procedure, an initial guess (i.e., the input) of the exact solution has to be utilized to obtain an approximate solution (i.e., the output). If the output solution differs from the input, the whole process must be repeated with better initial guesses until self-consistency is achieved, upon which the difference between the output and the input vanishes.

From a different perspective, in the field of differential equations, shooting method represents a mathematical scheme for solving a boundary value problem by changing it to the solution of an initial value problem.[6] In a broad sense, "boundary values" collectively put together a set of restraints (i.e., boundary *conditions*) imposed on the solutions of the differential equation, whereas "initial values" dictate specific values (i.e., initial *conditions*) of the unknown solutions at given fixed points in the domain of the solution.[6] For example, in a finite-well tunneling system, the continuity requirement of the wave function and its first derivatives (with respect to position) at the edges of the potential box constitutes the boundary values (conditions). In light of this conceptual lineup, all SCF convergence algorithms are indeed shooting methods with varied efficiencies: the Schrödinger-like equation is the differential equation under investigation, the self-consistency of the fully converged SCF solution is a bound-

ary value, and generating initial guesses is an initial value problem.

Despite its conceptual simplicity, an SCF system is often an ill-conditioned mathematical problem: many a time, less intelligent initial guesses do not lead the iterative process toward convergence.[1] Among many algorithms to speed up SCF convergence, Pulay's direct inversion in the iterative subspace (DIIS) method[2] has established itself as the most widely adopted, nearly universally applicable convergence accelerating scheme.[1] However, DIIS does fail sometimes, for which interest of designing more powerful SCF convergence *engine* has been feverishly pursued ever since.[1–5] Recently, the energy-DIIS (EDIIS) (Ref. 3) and the augmented Roothaan-Hall energy-DIIS (ADIIS) (Ref. 4) offered two most noticeable improvements over the traditional DIIS scheme.[2]

In the implementation of DIIS,[2] the residual error vectors $\{\mathbf{e}_i\}$ of a series of trial Fock matrices $\{\mathbf{F}_i\}$ are measured through the commutators between $\{\mathbf{F}_i\}$ and their associated density matrices $\{\mathbf{D}_i\}$: $\mathbf{e}_i = [\mathbf{F}_i, \mathbf{D}_i]$. After m iterations, a linear combination of previous Fock matrices,

$$\mathbf{F}_m^{\text{comb}} = \sum_{i=1}^{m} c_i \mathbf{F}_i \, , \qquad (1)$$

carries a residual error, $\mathbf{e}_m^{\text{comb}} = \sum_{i=1}^{m} c_i \mathbf{e}_i$, whose norm can be minimized subject to the normalization condition for the expansion coefficients $\{c_i\}$:

$$\min\left\{ \langle \mathbf{e}_m^{\text{comb}} | \mathbf{e}_m^{\text{comb}} \rangle, \ \sum_{i=1}^{m} c_i = 1 \right\} \, . \qquad (2)$$

The optimal values of $\{c_i\}$ are then used in Eq. (1) to generate the initial guess for the next iteration, and the above procedure is restarted all over again unless full convergence is attained. This numerical realization might contribute to the occasional failures of DIIS because the coefficients minimizing the norm of the error vector are not necessarily optimal either for the

a)Authors to whom correspondence should be addressed. Electronic addresses: yawang@chem.ubc.ca and ghc@everest.hku.hk.

expansion of the Fock matrices or for the convergence of the total energy. Moreover, the most important quantity, the total energy, plays no role in DIIS, which further deteriorates the performance of DIIS for many systems.[3–5]

For such reasons, based on the associated energies $\{E[\mathbf{D}_i]\}$ of the density matrices $\{\mathbf{D}_i\}$, EDIIS (Ref. 3) and ADIIS (Ref. 4) propose two distinct total energy expressions:

$$\mathcal{E}_{\text{EDIIS}} = \sum_{i=1}^m c_i E[\mathbf{D}_i] - \frac{1}{2} \sum_{i,j}^m c_i c_j \langle (\mathbf{D}_i - \mathbf{D}_j)(\mathbf{F}_i - \mathbf{F}_j) \rangle,$$
(3)

$$\mathcal{E}_{\text{ADIIS}} = E[\mathbf{D}_m] + 2 \sum_{i=1}^m c_i \langle (\mathbf{D}_i - \mathbf{D}_m)\mathbf{F}_m \rangle$$
$$+ \sum_{i,j}^m c_i c_j \langle (\mathbf{D}_i - \mathbf{D}_m)(\mathbf{F}_j - \mathbf{F}_m) \rangle.$$
(4)

Because Eqs. (3) and (4) originate from perturbation theory, they can only be incorporated into the following *interpolation* minimization procedure with non-negative $\{c_i\}$,[3,4]

$$\min \left\{ \mathcal{E}_{\text{EDIIS}} \text{ or } \mathcal{E}_{\text{ADIIS}}, \sum_{i=1}^m c_i = 1, c_i \geq 0 \right\},$$
(5)

which is essentially different from the general *extrapolation* minimization of DIIS in Eq. (2). To some degree of disappointment, the performance of EDIIS and ADIIS is *uneven*: for systems that DIIS really works, both EDIIS and ADIIS are less robust; for other systems that DIIS does not work well, both EDIIS and ADIIS are much more effective though.[3,4] Overall, ADIIS outshines EDIIS in almost every situation, despite that both EDIIS and ADIIS should be switched over to DIIS (i.e., EDIIS + DIIS and ADIIS + DIIS) to accomplish stability and smoothness near convergence.[3,4]

Such limited success of ADIIS and EDIIS motivated us to contrive a better SCF convergence engine, which will universally outperform all three DIIS, EDIIS, and ADIIS schemes (and their combos) for every system. To fulfill this goal, we begin with building our own energy expression from the corrected Hohenberg-Kohn-Sham functional (cHKS),[5]

$$E_i^{\text{cHKS}} = E[\mathbf{D}_i^{\text{out}}] + \langle (\mathbf{D}_m^{\text{b}} - \mathbf{D}_i^{\text{out}}) \Delta \mathbf{F}_i \rangle,$$
(6)

where $\Delta \mathbf{F}_i = (\mathbf{F}_i^{\text{out}} - \mathbf{F}_i^{\text{in}})/2$, the superscripts "out" and "in" betoken output and input, respectively, and $\mathbf{D}_m^{\text{b}}$ designates the *best* approximate to the fully converged $\mathbf{D}_0$ based on information available at the latest iteration m. Similar to Eq. (1), we introduce a linear expansion for $\mathbf{D}_m^{\text{b}}$ in terms of $\{\mathbf{D}_i\}$:

$$\mathbf{D}_m^{\text{b}} = \sum_{j=1}^m c_j \mathbf{D}_j,$$
(7)

with the exact contents of $\{\mathbf{D}_i\}$ to be decided later. After plugging Eq. (7) back into Eq. (6) and going through several steps of manipulations, we get our own total energy expression potentially suitable for a DIIS-like maneuver,

$$\mathcal{E}_i^{\text{cHKS}} = \sum_{j=1}^m c_j \left\{ E[\mathbf{D}_i^{\text{out}}] + \langle (\mathbf{D}_j - \mathbf{D}_i^{\text{out}}) \Delta \mathbf{F}_i \rangle \right\}.$$
(8)

In comparison to Eqs. (3) and (4), Eq. (8) is the most compact total energy expression accurate up to second order in density matrix residual error for both HFM and DFT.[5] Further inspection reveals the unfortunate fact: Eq. (8) is only *linear* in $\{c_i\}$, which immediately precludes a straightforward DIIS-type minimization in a least-squares sense. To meaningfully engage Eq. (8) in accelerating SCF convergence, we have proffered two new mechanisms based on the shooting technique in numerical analysis.[6] We thus name our own SCF convergence schemes "linear-expansion shooting techniques" (LIST), which are advanced hereafter.

Let us first present a *direct* approach to LIST (LISTd). It has been well established that Eq. (6) is accurate up to second order in density matrix residual error;[5] therefore, Eq. (8) should be very close to the fully converged energy for all iterations involved in the linear expansion, Eq. (7). This immediately suggests an accurate approximation of *shooting* the exact energy (up to second order in residual error),

$$\mathcal{E}_1^{\text{cHKS}} = \mathcal{E}_2^{\text{cHKS}} = \cdots = \mathcal{E}_m^{\text{cHKS}} = E[\mathbf{D}_m^{\text{b}}] \doteq E[\mathbf{D}_0],$$
(9)

leading to a matrix equation for $\{c_i\}$ and $E[\mathbf{D}_m^{\text{b}}]$, nearly identical to that of DIIS:[2]

$$\underbrace{\begin{pmatrix} 0 & -1 & -1 & \cdots & -1 \\ -1 & a_{11} & a_{12} & \cdots & a_{1m} \\ -1 & a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & a_{m1} & a_{m2} & \cdots & a_{mm} \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} E \\ c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix}}_{\mathbf{C}} = \underbrace{\begin{pmatrix} -1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}}_{\mathbf{O}},$$
(10)

where $a_{ij} = E[\mathbf{D}_i^{\text{out}}] + \langle (\mathbf{D}_j - \mathbf{D}_i^{\text{out}}) \Delta \mathbf{F}_i \rangle$. To avoid any damping effect slowing down SCF convergence in actual calculations, it is much more desirable to use $\{\mathbf{D}_i^{\text{out}}\}$, instead of $\{\mathbf{D}_i^{\text{in}}\}$, for $\{\mathbf{D}_i\}$ in Eq. (7). Unfortunately, a closer inspection of Eq. (10) unveils an unsettling linear-dependency issue: LISTd is very much likely to fail as it gets very close to self-consistency, where the density matrix residual error becomes very small. For instance, the last two columns of matrix $\mathbf{A}$ in Eq. (10) turn virtually identical whenever $\mathbf{D}_m^{\text{out}} \doteq \mathbf{D}_{m-1}^{\text{out}}$, yielding a nearly singular matrix $\mathbf{A}$.

To alleviate such a linear-dependency problem, we consider two alternative linear expansions, one based on $\{\mathbf{D}_i^{\text{in}}\}$ and the other, $\{\mathbf{D}_i^{\text{out}}\}$, along with their corresponding energies:

$$\widetilde{\mathbf{D}}_m^{\text{b}} = \sum_{i=1}^m \tilde{c}_i \mathbf{D}_i^{\text{in}} \implies \left\{ \widetilde{\mathcal{E}}_i^{\text{cHKS}} \right\},$$
(11)

$$\mathbf{D}_m^{\text{b}} = \sum_{i=1}^m c_i \mathbf{D}_i^{\text{out}} \implies \left\{ \mathcal{E}_i^{\text{cHKS}} \right\}.$$
(12)

Obviously, the closer the full convergence, i.e., $\mathbf{D}_i^{\text{in}} \doteq \mathbf{D}_i^{\text{out}}$, the better the approximations, $\tilde{c}_i = c_i$ and $\widetilde{\mathcal{E}}_i^{\text{cHKS}} = \mathcal{E}_i^{\text{cHKS}}$.
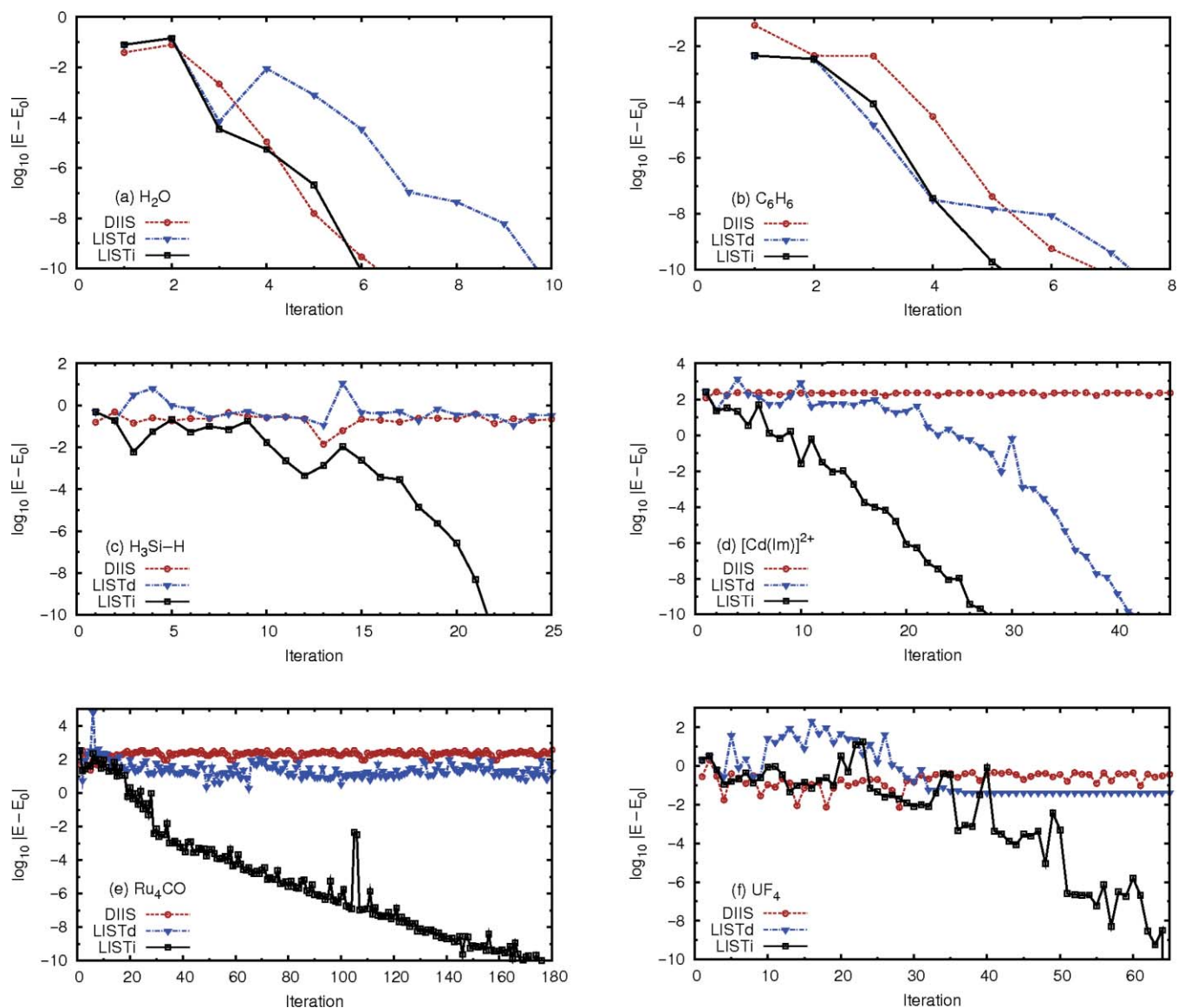
FIG. 1. Convergence of the total energy (in hartrees) for (a) water (H_2O), (b) benzene (C_6H_6), (c) silane (H_3Si–H) with a much extended Si–H bond, (d) cadmium(II)-imidazole complex $[Cd(Im)]^{2+}$, (e) tetranuclear ruthenium carbonyl cluster $Ru_4(CO)$, and (f) tetrahedral uranium fluoride (UF_4). For H_2O, the two H–O bond lengths and the H–O–H bond angle are 0.965 Å and 103.75°, respectively. The C=C and C–H bond lengths in C_6H_6 are 1.396 and 1.097 Å, respectively. For H_2O and C_6H_6, exchange and correlation functionals within local density approximation (LDA) (Refs. 8 and 9) were used with the 6-31G basis set. For H_3Si–H, the three regular and the elongated Si–H bond lengths are 1.47 and 4.00 Å, respectively; all the H–Si–H bond angles are 109.28°.[5] H_3Si–H was calculated at the LDA/6-31G* level of theory.[5] All the U–F bond lengths in UF_4 are 1.98 Å.[3] $[Cd(Im)]^{2+}$ was calculated at the B3LYP/3-21G level of theory[10,11] using the core Hamiltonian as the very first initial guess.[4] Both $Ru_4(CO)$ and UF_4 were calculated at B3LYP/LanL2DZ level with core electrons of Ru and U represented by the LanL2 pseudopotentials.[12] The geometries of $[Cd(Im)]^{2+}$, $Ru_4(CO)$, and UF_4 are the same as those used before.[3,4] Except for $[Cd(Im)]^{2+}$, atomic densities were utilized as the very first initial guesses for all other systems.

This fact immediately leads to

$$\sum_{j=1}^{m} c_j \langle (\mathbf{D}_j^{\text{out}} - \mathbf{D}_j^{\text{in}}) \Delta \mathbf{F}_i \rangle = 0 \qquad (13)$$

or to a matrix form strikingly similar to that of LISTd, Eq. (10),

$$\begin{pmatrix} 0 & -1 & -1 & \cdots & -1 \\ -1 & g_{11} & g_{12} & \cdots & g_{1m} \\ -1 & g_{21} & g_{22} & \cdots & g_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & g_{m1} & g_{m2} & \cdots & g_{mm} \end{pmatrix} \begin{pmatrix} 0 \\ c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad (14)$$

except $g_{ij} = \langle (\mathbf{D}_j^{\text{out}} - \mathbf{D}_j^{\text{in}}) \Delta \mathbf{F}_i \rangle$. It is worth noting that, without instituting a direct shooting within each linear expansion separately via Eq. (9), we just go through a distinct shooting scheme between two linear expansions, which can be best termed an *indirect* approach to LIST (LISTi) to differentiate it from the *direct* approach, LISTd.

We have implemented LISTd and LISTi into the NWChem code.[7] For LISTd, LISTi, and DIIS, it was found that only four output Fock matrices are adequately needed in the linear combination to generate the input Fock matrix for the next iteration. At each iteration of a LIST calculation, the cHKS functional was evaluated for the total energy.[5] All calculations were carried out without any other types of

SCF accelerating algorithms, and full convergence was defined as the energy difference between two consecutive iterations below 10^{-10} hartrees. To benchmark the effectiveness of our LIST methods, we have chosen six molecular systems of varied sizes and bonding complexity (Fig. 1): water and benzene are the two easy cases for DIIS to perform very well, whereas a silane molecule with a much extended Si–H bond,[5] a tetrahedral uranium fluoride,[3] a cadmium(II)-imidazole complex, and a tetranuclear ruthenium carbonyl cluster[4] are the four troublesome systems for DIIS to show no sign of convergence.

Surprisingly, Figs. 1(a) and 1(b) show that LISTd converges the SCF despite suffering from its own linear-dependence problem. However, the rate of convergence of LISTd can be much slower than LISTi, despite that LISTd performs better in the early stage of SCF process. LISTd is thus not recommendable as a general strategy to speed up SCF convergence. We also observe LISTi to be the most effective scheme, convincingly outperforming DIIS for almost all iterations from start to finish. This behavior is quite different from EDIIS and ADIIS, which are much inferior to DIIS for those systems that can be handled well by DIIS.[3,4]

To keep the comparability with other approaches for the four challenging cases, we have intentionally maintained the systems' structural and computational details virtually the same as before.[3–5] Figures 1(c)–1(f) exhibit the results for these four tough cases that DIIS simply does not show any sign of convergence at all for hundreds of iterations. Interestingly, LISTd fairs quite well for $[Cd(Im)]^{2+}$, reaching to full convergence within 42 iterations. Unfortunately, the other three systems prove to be extremely difficult for LISTd: either nonconvergent for both silane and $Ru_4(CO)$ or wrongfully convergent for UF_4. This again confirms our above assessment of LISTd: not to be entrusted for general usage. Most amazingly, in all four hard cases, LISTi accelerates SCF toward convergence with ease, rivaling the best performance of the ADIIS + DIIS scheme[4] and surpassing that of EDIIS and EDIIS + DIIS schemes.[3,4] Keep in mind that LISTi has achieved such an excellent performance all within a stand-alone, stable methodology without invoking DIIS at any stage of the SCF process. LISTi definitely has its own advantages over ADIIS + DIIS, or any other DIIS-type schemes.[2–4] In retrospect, built upon the most compact total energy expression accurate up to second order in residual error for both HFM and DFT, the formulation of LISTi ventures into both positive and negative domains of $\{c_i\}$ to explore the full extent of the linear-expansion space, which certainly contributes to the superiority of LISTi over DIIS, EDIIS, and ADIIS.

In conclusion, between the two different approaches to LIST, LISTd is mainly useful in the early stage of SCF process when the SCF is far away from full convergence, but is a less effective method not to be trusted for general applications. LISTi emerges as the most robust SCF convergence engine, better than previous DIIS and related EDIIS and ADIIS approaches.[2–4] Evidently, LISTi manifests its potential to become a universally applicable algorithm capable of accelerating SCF convergence for versatile systems.

[1] F. Jensen, *Introduction to Computational Chemistry*, 2nd Ed. (Wiley, West Sussex, 2007).

[2] P. Pulay, J. Comput. Chem. **3**, 556 (1982).

[3] K. N. Kudin, G. E. Scuseria, and E. Cancès, J. Chem. Phys. **116**, 8255 (2002).

[4] X. Hu and W. Yang, J. Chem. Phys. **132**, 054109 (2010).

[5] Y. A. Zhang and Y. A. Wang, J. Chem. Phys. **130**, 144116 (2009).

[6] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran 77* (Cambridge University Press, Cambridge, 1992).

[7] E. J. Bylaska *et al.*, NWCHEM, version 5.0, Pacific Northwest National Laboratory, Richland, WA, 2006.

[8] J. C. Slater and K. H. Johnson, Phys. Rev. B **5**, 844 (1972).

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

[10] A. D. Becke, J. Chem. Phys. **98**, 5648 (1993).

[11] C. Lee, W. Yang, and R. G. Parr, Phys. Rev. B **37**, 785 (1988).

[12] P. J. Hay and W. R. Wadt, J. Chem. Phys. **82**, 299 (1985).