

# On the equivalence of LIST and DIIS methods for convergence acceleration

Alejandro J. Garza<sup>1</sup> and Gustavo E. Scuseria<sup>2</sup>

<sup>1</sup>Department of Chemistry, Rice University, Houston, Texas 77251-1892, USA

<sup>2</sup>Department of Chemistry and Department of Physics and Astronomy, Rice University, Houston, Texas 77251-1892, USA and Chemistry Department, Faculty of Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia

(Received 26 November 2014; accepted 16 April 2015; published online 27 April 2015)

Self-consistent field extrapolation methods play a pivotal role in quantum chemistry and electronic structure theory. We, here, demonstrate the mathematical equivalence between the recently proposed family of LIST methods [Wang *et al.*, *J. Chem. Phys.* **134**, 241103 (2011); Y. K. Chen and Y. A. Wang, *J. Chem. Theory Comput.* **7**, 3045 (2011)] and the general form of Pulay's DIIS [Chem. Phys. Lett. **73**, 393 (1980); *J. Comput. Chem.* **3**, 556 (1982)] with specific error vectors. Our results also explain the differences in performance among the various LIST methods. © 2015 AIP Publishing LLC. [<http://dx.doi.org/10.1063/1.4919283>]

## INTRODUCTION

The importance of self-consistent field (SCF) extrapolation in Hartree–Fock and Kohn–Sham calculations cannot be overstated: without efficient techniques like DIIS (direct inversion of the iterative subspace),<sup>1,2</sup> much of the current body of work in computational quantum chemistry and related fields would not be practical. These techniques have permeated the quantum chemistry field and are ubiquitous in most (if not all) software packages. DIIS has found its way to many other applications like extrapolation of coupled cluster solutions<sup>3</sup> and convergence acceleration in other computational science areas, e.g., the popular GMRES method of Saad and Schultz<sup>4</sup> is equivalent to DIIS for linear problems.<sup>5</sup>

Since Pulay's seminal paper,<sup>1</sup> there has been an enormous amount of work to refine, improve, and complement DIIS<sup>6–10</sup> but the core of the technique remains the same, as originally formulated. More recently, there have been claims in the literature<sup>11,12</sup> that a family of recently proposed LIST (linear expansion shooting techniques) methods supersedes the performance of DIIS. We, here, prove that the LIST methods are mathematically equivalent to the general form of DIIS<sup>2</sup> (with specific error vectors) and are not superior to the commonly used commutator-DIIS (CDIIS), as shown numerically in our previous work.<sup>13</sup> In short, both CDIIS and LIST work by minimizing an approximate error function, but the vanishing of the error function of the former is necessary and sufficient condition for convergence whereas this is (in general) not true for the latter.

## DIIS

We begin by writing DIIS in its general formulation as first outlined by Pulay<sup>1,2</sup> (see below). An improved density matrix  $\tilde{D}_k$  is constructed from a linear combination of the previously iterated density matrices

$$\tilde{D}_k = \sum_{j=1}^k c_j D_j. \quad (1)$$

The coefficients in the above equation are determined as

$$\{c_j\} = \arg \min \left\{ \left\| \sum_{j=1}^k c_j e_j \right\|, \sum_{j=1}^k c_j = 1 \right\}, \quad (2)$$

where  $e_j$  is an approximate error function associated with  $D_j$ , and  $\|\cdot\|$  denotes a suitable norm. That is, DIIS consists of creating a  $\tilde{D}_k$  which minimizes (in the least squares sense)  $e_j$  constrained to  $\text{Tr}(\tilde{D}_k) = N$ , where  $N$  is the number of electrons.

It is worth mentioning that DIIS is often a synonym for the widely used CDIIS, which utilizes a specific error vector  $e_j = [F_j, D_j]$ , where  $F$  is the Fock matrix. However, the formulation of DIIS that we provide above is more general, but it should still be attributed to Pulay:<sup>2</sup> “The DIIS method can be briefly recapitulated as follows: In each SCF step, construct an error vector  $e_i$ , here  $i$  is the step index... It is then possible to find a linear combination of consecutive parameter vectors  $p = \sum_i c_i p_i$  so that the corresponding error vector  $\sum_i c_i e_i$  approximates the zero vector in the least-squares sense.” The flexibility on the choice of  $e_j$  has allowed the application of DIIS to accelerate the convergence of methods other than Hartree–Fock and Kohn–Sham DFT such as coupled cluster,<sup>3</sup> multireference wavefunctions,<sup>14</sup> plane-wave-based density functionals,<sup>15</sup> and geometry optimization algorithms as in, e.g., the popular gradient-DIIS<sup>16–18</sup> (GDIIS). We also note that Rohwedder and Schneider<sup>5</sup> have analyzed the DIIS method assuming a general, approximate, residual-like term for the error function.

There are three LIST methods: LISTi,<sup>11</sup> LISTd,<sup>11</sup> and LISTb.<sup>12</sup> We first show how LISTi is equivalent to the general DIIS described above.

## LISTi

Let  $D_i^{\text{in}}$  and  $D_i^{\text{out}}$  denote input and output density matrices from the diagonalization of the Fock matrix  $F_i^{\text{in}}$ . The equations to solve for the LISTi coefficients  $\{c_j\}$  can be written as

(Ref. 11, Eq. (13))

$$\sum_j c_j g_{ij} = \sum_j c_j \text{Tr}((D_j^{\text{out}} - D_j^{\text{in}})(F_i^{\text{out}} - F_i^{\text{in}})) = 0, \quad (3)$$

$\forall i$ , restricted to  $\sum_j c_j = 1$ . Assuming summation over repeated indices, Eq. (3) is equivalent to  $|c_j g_{ij}| = 0$ ,  $\forall i$ , where  $|\cdot|$  denotes absolute value. It is, therefore, also equivalent to  $\max\{|c_j g_{ij}|, i \in I\} = 0$  with  $I = \{1, 2, \dots, k\}$ , where  $k$  is the dimension of the iterative subspace. Thus, the set  $\{c_j\}$  can be expressed as

$$\{c_j\} = \arg \min \left\{ \max \left\{ \left| \sum_j c_j g_{ij} \right|, i \in I \right\}, \sum_j c_j = 1 \right\}. \quad (4)$$

Noting that the infinity norm of a vector  $\vec{v}$  of length  $k$  is defined as  $\|\vec{v}\|_\infty = \max\{|v_1|, \dots, |v_k|\}$ , Eq. (4) becomes

$$\{c_j\} = \arg \min \left\{ \left\| \sum_j c_j g_{ij} \right\|_\infty, \sum_j c_j = 1 \right\}. \quad (5)$$

Furthermore, if we let  $x$  be an index such that

$$\left| \sum_j c_j g_{xj} \right| = \left\| \sum_j c_j g_{ij} \right\|_\infty, \quad (6)$$

then we see that LISTi can be formulated exactly in the form of Eq. (2) with  $e_j = g_{xj}$  and  $\|\cdot\| = |\cdot|$ , or in terms of the vector  $e_j = g_{ij}$  and  $\|\cdot\| = \|\cdot\|_\infty$ . Note that other norms could also be used because the norm of a vector  $\vec{v}$  is zero *if and only if*  $\vec{v}$  is the zero vector.

In deriving the above equivalence, we have not made use of the explicit form of the matrix  $g_{ij}$ . This has the corollary that any SCF convergence acceleration technique that consists of solving a linear system of the form of Eq. (3) (i.e.,  $\sum_j c_j b_{ij} = 0, \forall i, \sum_j c_j = 1$ ) to obtain extrapolation coefficients for Eq. (1) corresponds to the general DIIS minimization problem of Eq. (2). As we discuss below, LISTd and LISTb can also be written in this form.

## LISTd

We proceed now to outline the relationship between DIIS and LISTd. These results also clarify the reason for the poor convergence acceleration of LISTd.<sup>11-13</sup> In brief, the better convergence properties of LISTb and LISTi as compared to LISTd can be understood in terms of the properties of the error function being minimized; a SCF solution minimizes the error function in LISTb and LISTi, but not necessarily in LISTd.

The equations to be solved for the LISTd coefficients  $\{c_j\}$  are (Ref. 11, Eq. (10))

$$\sum_j c_j a_{ij} = \sum_j c_j \left[ E_i - E + \text{Tr}((D_j^{\text{out}} - D_i^{\text{out}})\Delta F_i) \right] = 0, \quad (7)$$

$\forall i$ , with  $\sum_j c_j = 1$ , where  $\Delta F_i = F_i^{\text{out}} - F_i^{\text{in}}$ , and  $E$  is the current best estimate for the energy. From the results of the ‘‘LISTi’’ section, it is already clear that Eq. (7) fits in the general

framework of Eq. (2). We can also express  $\{c_j\}$  as

$$\{c_j\} = \arg \min \left\{ \max \left\{ \left| \sum_j c_j a_{ij} \right|, i \in I \right\}, \sum_j c_j = 1 \right\}, \quad (8)$$

where we could also have used the infinity norm as in Eq. (5). Thus, we can view LISTd as minimizing the approximate error function  $e(\mathbf{c}) = \max\{|\sum_j c_j a_{ij}|, i \in I\}$  or, equivalently in terms of the extrapolated  $\tilde{D}$ ,

$$e(\tilde{D}) = \max\{|E_i - E + \text{Tr}((\tilde{D} - D_i^{\text{out}})\Delta F_i)|, i \in I\}. \quad (9)$$

Let now  $E$  be the energy for the converged density matrix  $D$ . It is easy to see that  $e(D) \neq 0$  in general because the matrices in the iterative subspace  $D_i$  do not need (and are, in fact, not expected) to be converged. Hence, by formulating LISTd as an error minimization problem, we see that the choice for the approximate error function is a poor one, since a converged density matrix can have nonzero error—and therefore, it does not satisfy the LISTd equations. This explains the unsatisfactory convergence acceleration of LISTd.

## LISTb

The LISTb equations are given by the transpose of the LISTd matrix (Ref. 12, Eq. (14))

$$\sum_j c_j a'_{ij} = \sum_j c_j \left[ E_j - E + \text{Tr}((D_i^{\text{out}} - D_j^{\text{out}})\Delta F_j) \right] = 0, \quad (10)$$

$\forall i$ , with  $\sum_j c_j = 1$ . Again, the results in the LISTi section imply that Eq. (10) corresponds to the general DIIS minimization problem of Eq. (2). Analogously to Eq. (8), we can write

$$\{c_j\} = \arg \min \left\{ \max \left\{ \left| \sum_j c_j a'_{ij} \right|, i \in I \right\}, \sum_j c_j = 1 \right\}, \quad (11)$$

and the error function being minimized can be written as  $e(\{D_j\}, \mathbf{c}) = \max\{|a_i(\{D_j\}, \mathbf{c})|, i \in I\}$ , with

$$a_i(\{D_j\}, \mathbf{c}) = \sum_j c_j \left[ E_j - E + \text{Tr}((D_i^{\text{out}} - D_j^{\text{out}})\Delta F_j) \right]. \quad (12)$$

Because of the transposition, the error function can no longer be written in terms of the extrapolated matrix  $\tilde{D}$ . However, we can consider the error function for a set of matrices  $\{D_j\}$  for which  $D_k = D$  is a converged density matrix. When  $E$  is the converged energy, then it is straightforward to see that, for a set of coefficients  $\mathbf{c}$  with  $c_j = 1$  if  $j = k$  and  $c_j = 0$  otherwise,  $e(\{D_j\}, \mathbf{c}) = e(D, 1) = 0$ . Thus, a converged density matrix minimizes the LISTb error, satisfying Eq. (10). In consequence, LISTb can correctly select  $D_k = D$  as an SCF solution from the set  $\{D_j\}$ , whereas LISTd (in general) does not do this. The improved convergence acceleration of LISTb as compared to LISTd can therefore be attributed to the minimization of a more suitable error function (which is in fact the idea behind the DIIS procedure). The same applies to LISTi: since  $D_j^{\text{out}} - D_j^{\text{in}}$  must be the zero matrix for a converged density, it is straightforward to see that the correct solution  $c_j = \delta_{jk}$  is a minimizer of Eq. (4).

TABLE I. Energies (in Hartrees) of the SCF solutions afforded by CDIIS, LIST, and stability analysis for various systems. The LIST data were taken from Ref. 12. The geometries are the same as those used in Refs. 11–13.

| System                 | Level of theory | $E_{\text{CDIIS}}$ | $E_{\text{LIST}}^a$ | $E_{\text{Stable}}$ | $E_{\text{Stable}} - E_{\text{CDIIS}}$ | $E_{\text{Stable}} - E_{\text{LIST}}$ | $E_{\text{LIST}} - E_{\text{CDIIS}}$ |
|------------------------|-----------------|--------------------|---------------------|---------------------|--|---------------------------------------|--------------------------------------|
| SiH <sub>4</sub>       | SVWN5/6-31G*    | -290.457 82        | -290.457 70         | -290.457 82         | 0.0×10 <sup>0</sup>                    | -1.2×10 <sup>-4</sup>                 | -1.2×10 <sup>-4</sup>                |
| [Cd(Im)] <sup>2+</sup> | B3LYP/3-21G     | -5667.008 71       | -5667.008 72        | -5667.009 41        | -7.0×10 <sup>-4</sup>                  | -6.9×10 <sup>-4</sup>                 | -1.3×10 <sup>-5</sup>                |
| Ru <sub>4</sub> (CO)   | B3LYP/LANL2DZ   | -488.697 27        | -488.710 67         | -488.717 65         | -2.0×10 <sup>-2</sup>                  | -7.0×10 <sup>-3</sup>                 | -1.3×10 <sup>-2</sup>                |
| UF <sub>4</sub>        | B3LYP/LANL2DZ   | -451.211 84        | -451.231 19         | -451.240 06         | -2.8×10 <sup>-2</sup>                  | -8.9×10 <sup>-3</sup>                 | -1.9×10 <sup>-2</sup>                |

<sup>a</sup>LISTi and LISTb converge to the same solutions.

Previously,<sup>12</sup> the better convergence properties of LISTb, as compared to LISTd, were attributed to an alleviation of the linear dependency problem of LISTd by the transposition. This assertion is incorrect; the ill-conditioning of the LISTd matrix  $\mathbf{A}$  is determined by its condition number, which is the ratio between the largest and smallest singular values of  $\mathbf{A}$ . The singular value decomposition of  $\mathbf{A}$  is  $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ , whereas for LISTb  $\mathbf{A}^T = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T$ . The transposition, therefore, does not alleviate the linear dependency of  $\mathbf{A}$  in any way. An argument based on Cramer's rule was also given in Ref. 12, which pointed out that LISTd would tend to yield coefficients of large magnitude and opposite signs when close to convergence. As is common knowledge in numerical analysis, subtracting two large numbers of opposite signs is unwise because of the possibility of catastrophic cancellation. However, this explanation is still unsatisfactory because the magnitudes of the coefficients would need to be extremely large—comparable to the inverse of machine precision—and LISTd performs poorly even when far from convergence.<sup>11,12</sup> The argument for the improved acceleration of LISTb over LISTd based on the minimization of approximate error functions seems, therefore, much more plausible than the aforementioned explanations and simultaneously clarifies why LISTi is also better than LISTd.

## DISCUSSION

The most widely utilized version of DIIS employs the commutator  $[F_i, D_i]$  as error vector since  $[F, D] = 0$  is a necessary and sufficient condition for a SCF solution.<sup>2</sup> This specific variant is commonly known as commutator-DIIS or CDIIS. We have seen here that LIST methods can be formulated as DIIS in the general framework outlined originally by Pulay.<sup>1</sup> Considering this equivalence between LIST and DIIS, and how well-established the latter is, it is unsurprising that LIST can provide convergence acceleration. However, of the three different flavors of LIST, just two (LISTi and LISTb) work properly since only these minimize a suitable error function. More specifically, LISTi and LISTb minimize errors associated with necessary (albeit not sufficient) conditions for convergence, whereas LISTd minimizes a function which is not related to necessary or sufficient conditions for convergence. An alternative analysis of the error functions minimized by LIST, also showing the relationship to DIIS with  $e_j = \Delta F_j$  and that the LIST coefficients are independent of the energy terms in Eqs. (7) and (10), is given in the Appendix.

Based largely on the poor performance of CDIIS for the singlets of SiH<sub>4</sub> (with a broken bond), [Cd(Im)]<sup>2+</sup>, Ru<sub>4</sub>(CO)

and UF<sub>4</sub>, the authors in Refs. 11 and 12 concluded that LIST methods were superior to CDIIS. For these systems, CDIIS appeared to be trapped in states higher in energy than the LIST solutions. However, no stability analysis was carried out to verify whether the LIST solutions were high-energy states too. Table I compares the energies reported for LIST<sup>12</sup> with those from our calculations in *Gaussian*<sup>19</sup> using CDIIS [SCF=(CDIIS, NoDamp) keyword] and RHF → RHF stability analysis [Stable=(RRHF, Opt)]. All our CDIIS calculations use the default iterative subspace of 20 vectors, the Harris guess,<sup>20</sup> and tight convergence criteria. Cartesian  $d$  and higher functions (6D, 10F) were used to compare the energies from *Gaussian* with those of NWChem<sup>21</sup> (the package used in Refs. 11 and 12) for a given basis. We also employed the Integral=Ultrafine keyword and, based on calculations with different grid sizes, we estimate the error due to grid size in our comparisons with the data from Ref. 12 to be about 0.1 mHartree. Based on these considerations, it appears like LIST and CDIIS converge to the same solutions for SiH<sub>4</sub> and [Cd(Im)]<sup>2+</sup>. For Ru<sub>4</sub>(CO) and UF<sub>4</sub>, LIST converges to lower energy states than DIIS; however, the data obtained from stability analysis  $E_{\text{Stable}}$  reveal that the LIST solutions are high-energy states too. We also carried out LIST calculations starting from the Harris guess—atomic densities were used as initial guess in Refs. 11 and 12—with our own implementation of LIST in *Gaussian*; for SiH<sub>4</sub> and [Cd(Im)]<sup>2+</sup>, these converge to the same solutions as CDIIS; for Ru<sub>4</sub>(CO) and UF<sub>4</sub>, LIST converges to energies of -488.710 67 and -451.230 12 Hartrees, respectively. Hence, it seems that LIST has the same problem as CDIIS for these systems—both converge to high-energy states—in agreement with what could be expected from the equivalences between LIST and DIIS derived above. We also note that we did not find any convincing indication of LIST superiority in our previous numerical studies.<sup>13</sup>

It is also germane to point out that—for SiH<sub>4</sub>, Ru<sub>4</sub>(CO), and UF<sub>4</sub>—wavefunctions much lower in energy can be found if one allows the initial guess to break spin symmetry or via RHF → UHF stability analysis. The symmetry breaking is an indication of static correlation. The symmetry-adapted single Slater determinant approximation therefore breaks down, which is most likely the reason for the many unphysical solutions that can be found in these systems. Convergence to these high-energy states is a problem that neither LIST nor CDIIS can truly fix because this depends crucially on the initial guess. Even using more robust (and expensive) algorithms, such as Backskey's quadratically convergent SCF procedure,<sup>22</sup> we were not able to find the lower energy solutions obtained via stability analysis in Table I starting from standard initial guesses.

Another problem shared by LIST and DIIS is that they may result in oscillations.<sup>11–13</sup> This behavior arises because minimization of an approximate error function does not force convergence. A way to solve this issue is to use methods that ensure a decrease in the energy at every iteration, as this guarantees convergence to a local minimum.<sup>6,10</sup> This is the motivation behind the optimal damping algorithm<sup>6</sup> and its generalization in EDIIS.<sup>7</sup> These techniques exploit the fact that, because of the *aufbau* principle, all local energy minima for an idempotent density matrix are in the convex set  $\tilde{\mathcal{P}}_N = \{\tilde{D} \in \mathcal{M}_S(N_b), \tilde{D}^2 \leq \tilde{D}, \text{Tr}(\tilde{D}) = N\}$  ( $N_b$  is the number of basis functions and  $\mathcal{M}_S(N_b)$  is the set of Hermitian matrices of dimension  $N_b$ ). Thanks to this property, EDIIS can reduce the energy at every iteration without a significant increase in cost over DIIS via an interpolation of previously iterated density matrices—interpolation, rather than extrapolation, is necessary to ensure  $\tilde{D}_k \in \tilde{\mathcal{P}}_N$ . The interpolation has the side effect of making convergence slower as compared to DIIS, and thus EDIIS is often combined with DIIS to make the algorithm faster than the former and more robust than the latter.<sup>7,13</sup> Basically, EDIIS is used when far from convergence (as judged by the DIIS error) to bring the density matrix near the convergence region, where DIIS is most efficient. Because of this improved robustness, a combination of EDIIS and DIIS has been used as the default option for SCF convergence in the *Gaussian* suite of programs for many years.<sup>19</sup>

## CONCLUSIONS

The LIST methods can all be formulated as extrapolation techniques that minimize an approximate error function associated with a density matrix in the iterative subspace, and thus fall within the general scheme of DIIS first described by Pulay. This formulation also explains why LISTd has such a poor performance—the DIIS error minimized is not a suitable one. The other LIST methods were derived from LISTd; however, they introduce approximations which lead to better error functions and thus have better convergence properties. Nevertheless, because of the equivalences shown here, they share the same problems and are not better than the commonly used CDIIS. From a formal perspective, CDIIS appears to be more desirable than LIST as  $[F, D] = 0$  is a necessary and sufficient condition for a SCF solution, whereas LISTi and LISTb minimize error functions corresponding to necessary (but not sufficient) conditions for convergence. Furthermore, any SCF convergence extrapolation technique (Eq. (1)) that consists of solving a linear system of the form  $\sum_j c_j b_{ij} = 0, \forall i$  restricted to  $\sum c_j = 1$ , to obtain the expansion coefficients is in fact a form of DIIS. Finally, we have also presented numerical data demonstrating that the numerical evidence used to argue about the superiority of LIST over CDIIS does not really indicate that LIST is performing better than CDIIS. Claims in the literature that LIST outperforms CDIIS, EDIIS, and their combinations are, therefore, inaccurate.

## ACKNOWLEDGMENTS

This work was supported as part of the Center for the Computational Design of Functional Layered Materials,

an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences under Award No. DE-SC0012575. G.E.S. is a Welch Foundation chair (C-0036).

## APPENDIX: ALTERNATIVE ANALYSIS OF LIST

Here, we demonstrate the close relationship between LISTi, LISTb, and DIIS with  $e_j = \Delta F_j$ . Consider the LISTb equations

$$\underbrace{\begin{pmatrix} \hat{\mathbf{A}} & \mathbf{1}^T \\ \mathbf{1} & 0 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} \mathbf{c} \\ \lambda \end{pmatrix}}_{\mathbf{x}} = \underbrace{\begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}}_{\mathbf{b}}, \quad (\text{A1})$$

where  $\mathbf{1} = (1, 1, \dots, 1)$ ,  $\lambda$  is a Lagrange multiplier, and the elements of  $\hat{\mathbf{A}} \in \mathbb{R}^{k \times k}$  are the  $a'_{ij}$  in Eq. (10). By Cramer's rule, the coefficient  $c_m$  is equal to  $\det(\mathbf{A}_m)/\det(\mathbf{A})$ , where  $\mathbf{A}_m$  is a matrix formed by replacing the  $m$ th column of  $\mathbf{A}$  by the column vector  $\mathbf{b}$ . Let us define a matrix  $\mathbf{B}$  that replaces the  $a'_{ij}$  elements in  $\mathbf{A}$  by  $b_{ij} = \text{Tr}(D_i^{\text{out}} \Delta F_j)$ . We note that the matrix  $\mathbf{B}$  can be constructed by adding  $[E - E_j + \text{Tr}(D_j^{\text{out}} \Delta F_j)] \times (\mathbf{1}, 0)^T$  to the  $j$ th column of  $\mathbf{A}$  for every  $j$  from one to  $k$ . Because adding a scalar multiple of one column to another column leaves the determinant unchanged, we see that  $\det(\mathbf{A}) = \det(\mathbf{B})$  and, for  $m \in [1, k]$ ,  $\det(\mathbf{A}_m) = \det(\mathbf{B}_m)$ , where  $\mathbf{B}_m$  is defined analogously to  $\mathbf{A}_m$ . It follows that solving  $\mathbf{A}\mathbf{x} = \mathbf{b}$  yields the same extrapolation coefficients as those obtained from the solution to  $\mathbf{B}\mathbf{x}' = \mathbf{b}$ , with  $\mathbf{x}' = (\mathbf{c}, \lambda)^T$ . Let us denote  $\Delta \tilde{F} = \sum_j c_j \Delta F_j$  and  $\Delta \tilde{D} = \sum_j c_j \Delta D_j$ . The equations to satisfy for LISTb, LISTi, and DIIS with  $e_j = \Delta F_j$  can be rewritten as

$$\text{Tr}(\Delta \tilde{F} D_i^{\text{out}}) = 0, \quad \forall i \quad (\text{LISTb}), \quad (\text{A2})$$

$$\text{Tr}(\Delta \tilde{D} \Delta F_i) = 0, \quad \forall i \quad (\text{LISTi}), \quad (\text{A3})$$

$$\text{Tr}(\Delta \tilde{F} \Delta F_i) = 0, \quad \forall i \quad (\text{DIIS}). \quad (\text{A4})$$

Hence, the reason for which LISTb and LISTi can provide convergence acceleration is because they indirectly minimize  $\Delta \tilde{F}$  (or  $\Delta \tilde{D}$ ), as DIIS with  $e_j = \Delta F_j$  would do. Note that the role of the energy in the original LISTb equations (Eq. (10)) is superfluous. Likewise, making an analysis of LISTd (Eq. (7)) analogous to the one above, one obtains  $\text{Tr}(\tilde{D} \Delta F_i) = 0, \forall i$ , which does not make use of the energy neither. Additionally, this expression is unrelated to necessary or sufficient conditions for convergence, so this is an alternative way of illustrating the reason for the poor performance of LISTd.

<sup>1</sup>P. Pulay, *Chem. Phys. Lett.* **73**, 393 (1980).

<sup>2</sup>P. Pulay, *J. Comput. Chem.* **3**, 556 (1982).

<sup>3</sup>G. E. Scuseria, T. J. Lee, and H. F. Schaefer III, *Chem. Phys. Lett.* **130**, 236 (1986).

<sup>4</sup>Y. Saad and M. H. Schultz, *SIAM J. Sci. Stat. Comput.* **7**, 856 (1986).

<sup>5</sup>T. Rohwedder and R. Schneider, *J. Math. Chem.* **49**, 1889 (2011).

<sup>6</sup>E. Cancès and C. Le Bris, *Int. J. Quantum Chem.* **79**, 82 (2000).

<sup>7</sup>K. N. Kudin, G. E. Scuseria, and E. Cancès, *J. Chem. Phys.* **116**, 8255 (2002).

<sup>8</sup>X. Hu and W. Yang, *J. Chem. Phys.* **132**, 054109 (2010).

<sup>9</sup>S. Høst, J. Olsen, B. Jansik, L. Thøgersen, P. Jørgensen, and T. Helgaker, *J. Chem. Phys.* **129**, 124106 (2008).

<sup>10</sup>E. Cancès, M. Defranceschi, W. Kutzelnigg, C. Le Bris, and Y. Maday, in *Handbook of Numerical Analysis: Special Volume: Computational Chemistry*, edited by P. G. Ciarlet and C. Le Bris (Elsevier, Amsterdam, 2003), Vol. X, pp. 116–130 and references therein.

- <sup>11</sup>Y. A. Wang, C. Y. Yam, Y. K. Chen, and G. Chen, *J. Chem. Phys.* **134**, 241103 (2011).
- <sup>12</sup>Y. K. Chen and Y. A. Wang, *J. Chem. Theory Comput.* **7**, 3045 (2011).
- <sup>13</sup>A. J. Garza and G. E. Scuseria, *J. Chem. Phys.* **137**, 054110 (2012).
- <sup>14</sup>T. P. Hamilton and P. Pulay, *J. Chem. Phys.* **84**, 5728 (1986).
- <sup>15</sup>J. Hutter, H. P. Lüthi, and M. Parrinello, *Comput. Mater. Sci.* **2**, 244 (1994).
- <sup>16</sup>P. Császár and P. Pulay, *J. Mol. Struct.* **114**, 31 (1984).
- <sup>17</sup>Ö. Farkas and H. B. Schlegel, *Phys. Chem. Chem. Phys.* **4**, 11 (2002).
- <sup>18</sup>X. Li and M. J. Frisch, *J. Chem. Theory Comput.* **2**, 835 (2006).
- <sup>19</sup>M. J. Frisch, G. W. Trucks, H. B. Schlegel *et al.*, GAUSSIAN 09, Revision B.1, Gaussian, Inc., Wallingford CT, 2009.
- <sup>20</sup>J. Harris, *Phys. Rev. B* **31**, 1770 (1985).
- <sup>21</sup>E. J. Bylaska *et al.*, NWChem, version 5.0, Pacific Northwest National Laboratory, Richland, WA, 2006.
- <sup>22</sup>G. B. Bacskay, *Chem. Phys.* **61**, 385 (1981).