Error Vector Choice in Direct Inversion in the Iterative Subspace Method

IRINA V. IONOVA and EMILY A. CARTER*

Department of Chemistry and Biochemistry, University of California, Los Angeles, California 90095-1569

Received 30 October 1995; accepted 23 February 1996

ABSTRACT

Based on Banach's principle, we formally obtain possible choices for an error vector in the direct inversion in the iterative subspace (DIIS) method. These choices not only include all previously proposed error vectors, but also a new type of error vector which is computationally efficient and applicable to much wider range of problems. The error vector analysis also reveals a strong connection between DIIS and damping, thus adding to understanding of the reasons behind DIIS's effect on convergence. We illustrate our conclusions with several examples. © 1996 by John Wiley & Sons, Inc.

Introduction

The direct inversion in the iterative subspace (DIIS) method introduced by Pulay in 1980[1] proved to be an extremely helpful tool for accelerating the convergence of numerous iterative processes. In this method, an interpolation in the subspace spanned by the results of several previous iterations is performed to find the next approximation to the solution; the only information used for the purpose of such an interpolation being the so-called "error vector" associated with every iteration. This strategy worked very well for different kinds of problems, including the self-consistent field (SCF) procedure[1-6] and molecular geometry optimization.[5,7]

*Author to whom all correspondence should be addressed.

The DIIS algorithm suggests that the choice of an error vector for a particular implementation should have significant impact on the DIIS behavior. However, no study comparing different choices of an error vector and/or offering a specific prescription for error vector construction has been reported. The absence of a unified approach to error vector construction is clearly illustrated by the variety of error vectors used in particular implementations of DIIS.[1-7]

To address this issue, we consider a connection between the DIIS procedure and damping. (This connection was pointed out by Pulay[1] with regard to the simplest DIIS and the damping algorithm of Neilsen,[8] but it did not receive further consideration.) We will show formally that DIIS can be interpreted as consistently optimal damping; i.e., a damping in which the damping coefficients at all previous iterations result in optimal properties of

Journal of Computational Chemistry, Vol. 17, No. 16, 1836–1847 (1996)

CCC 0192-8651 / 96 / 161836-12

the current iteration. This formulation not only sheds further light on DIIS's favorable effect on convergence, but it also provides support for particular error vector choices and justifies the use of a novel error vector that is computationally efficient and applicable to practically every iterative process. In this article, we will discuss these error vectors in detail and illustrate our conclusions by considering the performance of DIIS with different error vectors. These examples involve molecular geometry optimizations of a Na_6 cluster and the Si_2H_6 molecule.

Relation of DIIS to Damping

SCF iterations and most molecular geometry optimization schemes can be described as an iterative process:

$$x_i = T(x_{i-1}) \tag{1}$$

that aims at finding a solution x^* of the problem:

$$x^* = T(x^*) \tag{2}$$

For example, in case of SCF iterations, x_i is either a (composite) Fock matrix or a matrix of expansion coefficients of molecular orbitals over basis functions at iteration i, while the operator T corresponds to the process of transforming x_{i-1} into x_i that takes place during one SCF iteration.

In practice, process (1) may converge very slowly or even diverge; to remedy such situations, it is common to use either DIIS or damping techniques, where the former is considered mostly as a convergence accelerator and the latter as a convergence stabilizer. In the DIIS method,[1] an error vector Δ_i is associated with every x_i (except, maybe, the initial guess, x_0), and at every iteration, n, the linear combination:

$$\tilde{x}_n = \sum_{i=i_0}^{n} c_i^{(n)} x_i \tag{3}$$

is formed subject to the following conditions:

$$\sum_{i=i_0}^{n} c_i^{(n)} = 1 \tag{4}$$

$$\left\| \sum_{i=i_0}^{n} c_i^{(n)} \Delta_i \right\| \rightarrow \min \tag{5}$$

The current parameter vector x_n is then substituted by $\tilde{x}_n$ which, in case of Δ_i being a linear function of x_i, yields the minimal norm for the corresponding error vector. Such a process generally results in faster convergence than prescription (1).

When one deals with convergence problems of process (1) by means of damping, the iterative sequence $x_1, \ldots, x_{n-1}$ is augmented by:

$$\tilde{x}_n = \tau x_n + (1 - \tau) x_{n-1} \tag{6}$$

which means that, instead of taking a full step from x_{n-1} to x_n at the nth iteration according to eq. (1), one takes a fraction τ of this step, arriving at $\tilde{x}_n$. Eq. (6) plays a major role in several damping techniques developed for stabilization and/or acceleration of SCF iterations,[1,2,8,10-12] in which a damping parameter τ is chosen according to either empirical[10] or analytical[1,2,8,11,12] schemes. The latter are based on the asymptotic behavior of $(x_i - x_{i-1})$ near the solution x^*.

Unlike the assumption of linearity of Δ that provides grounds for DIIS, the underlying justification for damping follows from Banach's fixed point principle (see, e.g., Evtushenko[9]). According to this principle, process (1) will converge if the operator T is "contracting"; i.e., if it satisfies the condition:

$$\|T(x) - T(y)\| \leq q\|x - y\|, \tag{7}$$

where $0 < q < 1$. The smaller the "contraction constant," q, the faster the convergence of process (1). Therefore, it is advantageous to introduce another operator, $\tilde{T}$, that has the same fixed point x^* as does the operator T but has a lower contraction coefficient q. The simplest way to form such an operator $\tilde{T}$ is the following[9]:

$$\tilde{T} = \tau T + (1 - \tau) I \tag{8}$$

Here τ is a parameter that is adjusted to get the smallest possible contraction coefficient, q, and I is the identity operator. Then, for some known $x, y, T(x)$, and $T(y)$, one can estimate the optimal τ from the condition:

$$\begin{aligned} &\|\tilde{T}(x) - \tilde{T}(y)\| \\ &\quad = \|\tau(T(x) - T(y)) + (1 - \tau)(x - y)\| \\ &\quad \rightarrow \min_{\tau} \end{aligned} \tag{9}$$

Note that the operator $\tilde{T}$ obtained from this condition generally depends on the x and y used, but it is always more contracting than the original opera-

tor T when applied to both x and y. Moreover, the result of $\tilde{T}$ operating on x or y is readily available: $\tilde{T}(x) = \tau T(x) + (1 - \tau)x$. Of course, to guarantee the existence of such an operator $\tilde{T}$, the operator T has to be uniformly monotonic (see Evtushenko[9] for details), but this requirement is much weaker as compared to the requirement of T being linear.

Once the operator $\tilde{T}$ is obtained for the current iteration n, it is advantageous to substitute the current point x_n obtained via $x_n = T(x_{n-1})$ by the point $\tilde{x}_n$ obtained as:

$$\tilde{x}_n = \tilde{T}(x_{n-1}) \tag{10}$$

This step can be interpreted as an implementation of damping in the original process (1) at the nth iteration, since, according to eq. (8):

$$\begin{aligned} \tilde{x}_n &= \tau T(x_{n-1}) + (1 - \tau)x_{n-1} \\ &= \tau x_n + (1 - \tau)x_{n-1} \end{aligned} \tag{11}$$

which corresponds exactly to eq. (6). Here it is important to note that condition (9) provides a way to determine an optimal damping parameter τ at *every* stage of the iterative process, not just toward the end of it, as in previously proposed damping schemes.[1,2,8,10-12]

Indeed, at iteration n, x_n, x_{n-1}, and x_{n-2} are available, and we can rewrite eq. (9), in which $x = x_{n-1}$ and $y = x_{n-2}$, as:

$$\|\tau_n(x_n - x_{n-1}) + (1 - \tau_n)(x_{n-1} - x_{n-2})\| \to \min_{\tau_n} \tag{12}$$

to determine τ_n and, hence, operator $\tilde{T}$. Now, if we take the point from which to continue iterations as $\tilde{x}_n = \tilde{T}(x_{n-1})$ in accord with eq. (10), it will correspond to eqs. (3)–(5) where $i_0 = (n - 1)$ and $\Delta_n = (x_n - x_{n-1})$, as follows from eqs. (10), (11), and (12). However, if the point from which to continue iterations is taken as:

$$\tilde{x}_n = \tilde{T}(x_{n-2}) = \tau_n x_{n-1} + (1 - \tau_n)x_{n-2} \tag{13}$$

it will correspond to the error vector $\Delta_{n-1} = x_n - x_{n-1}$ and, hence, $\Delta_n = (x_{n+1} - x_n)$, as follows from the comparison between eqs. (12) and (13) and eqs. (3)–(5). Last, if y is chosen as $y = x^*$ (and $x = x_{n-1}$), it is easy to see from eq. (9) that the error vector corresponding to eqs. (10) or (11) will be $\Delta_n = (x_n - x^*)$.

Instead of searching for the most contracting operator $\tilde{T}$ by means of eq. (9), one can also search for an optimal linear combination $x_\alpha = (\alpha x + (1 - \alpha)y)$ to which T should be applied, based on the condition:

$$\|T(x_\alpha) - x_\alpha\| \to \min_{\alpha} \tag{14}$$

Note that x_α optimal for T will be optimal for $\tilde{T}$ and vice versa, but to obtain α from condition (14) one needs to assume that T is linear. In this case, expression (14) leads to:

$$\|\alpha(T(x) - x) + (1 - \alpha)(T(y) - y)\| \to \min_{\alpha} \tag{15}$$

which is equivalent to DIIS with the error vector $\Delta(x) = (T(x) - x)$, exactly the error vector proposed in ref. 1. However, since the (assumed) linearity of T was already used to obtain α, it is reasonable to obtain $T(\tilde{x}_\alpha)$ as $(\alpha T(x) + (1 - \alpha)T(y))$ and use *that* point as the one from which to continue iterations. In this case, eq. (15) becomes equivalent to the DIIS's condition (5) where $\Delta(T(x)) = (T(x) - x)$. This means that the optimal point, $(\alpha T(x) + (1 - \alpha)T(y))$, is chosen based on the error vector $\Delta(x) = (x - T^{-1}(x))$ which, in the case of process (1), is expressed as $\Delta_n = (x_n - x_{n-1})$.

Thus, the uniform monotonicity of the operator T makes it possible to determine a more contracting operator $\tilde{T}$ by utilizing damping. If one assumes further that T is linear, it becomes possible to find an optimal point to which T (or $\tilde{T}$) should be applied, and this is accomplished by DIIS. It is easy to show that either DIIS or damping will result in the same iterative sequence whenever T is linear and the x and y used to estimate the damping parameter τ via (9) are related as $y = T(x)$, because only in this case is the α obtained from condition (15) equal to the τ obtained from condition (12). It turns out that this relationship between DIIS and damping is not limited to the simplest DIIS with $n = (i_0 + 1)$ but holds for general DIIS as well, as shown in the Appendix. In this case, the linear combination (3) is formed as the result of either DIIS or damping, where, in the latter, the previous damping parameters τ_i are allowed to adjust at each iteration. We will refer to such a damping as consistently optimal damping, in order to reflect the requirement of consistency between consecutive damping parameters.

From the discussion above, it becomes clear that the occasionally robust behavior of DIIS in early iterations must result from (near) linearity of the

operator T in the local neighborhood of the current parameter vector. It is important to note that T is rigorously linear for many nontrivial applications such as SCF iterations that involve closed-shell Hartree–Fock wave functions (see Stanton[13]). Another example is the quasi-Newton optimization method without line searches in which the corresponding operator T is just $T = I - H\nabla$, where H is an approximate inverse Hessian matrix and ∇ is the differential operator, so that such T can be considered practically linear within the quadratic region of the solution.

We also have seen that the correspondence between DIIS and damping results in formal identification of the error vectors suitable for DIIS according to Banach's principle. However, since this principle *per se* does not indicate which (if any) of the three different error vectors obtained is the best, we consider this issue in the following section.

Error Vector Analysis

As shown previously, the three different error vector choices that follow from Banach's principle are:

$$\Delta'(x_i) = \Delta'_i = x_{i+1} - x_i = T(x_i) - x_i \quad (16)$$

$$\Delta''(x_i) = \Delta''_i = x_i - x^* \quad (17)$$

and:

$$\Delta'''(x_i) = \Delta'''_i = x_i - x_{i-1} = x_i - T^{-1}(x_i) \quad (18)$$

It turns out that all successful implementations of DIIS proposed by others[1-5,7] involve error vectors of either types (16) or (17). Indeed, in the first article on DIIS,[1] the error vector (16) was proposed and shown to be useful. However, for an application of DIIS to Hartree–Fock SCF such a choice required the costly evaluation of an additional Fock matrix that was not used for any other part of the calculation. This was the main reason for abandoning such an error vector choice in future applications of DIIS.

In subsequent work,[2] a very convenient choice for an error vector was suggested; that is, the error vector at iteration n was formed from the off-diagonal part of composite Fock matrix (CFM) in the orbital basis of this iteration. (The off-diagonal elements F_{ij} of the CFM are, in general, proportional to $\langle i|\mathscr{F}_j - \mathscr{F}_i|j\rangle$, where $\mathscr{F}_i$ is the Fock operator associated with an orbital $|i\rangle$). Of course, to apply the DIIS technique, error vectors so obtained need to be transformed to a common basis, for instance, atomic or some fixed molecular orbital basis. Such a choice of an error vector has been successfully implemented for converging different wave functions ever since.[2-4] It is easy to see that this error vector approximates error vector (17), where x^* is an optimal CFM. Indeed, if all error vectors are considered in the optimal molecular basis where the optimal CFM is diagonal, then the error vector taken as the off-diagonal part of the CFM becomes equal to Δ''_i if the differences between the diagonal matrix elements of x_i and x^* are neglected. Note, that by choosing diagonal elements of the CFM in the iteration-dependent molecular basis as constants, as implemented in Muller et al.,[4] one still cannot achieve exact correspondence between error vector (17) and the error vector which is the off-diagonal part of the CFM because the diagonals of these CFMs will be different after the necessary transformation to a common basis.

A similar approach to error vector formation is taken when a wave function or an equilibrium structure of the molecule is sought via direct minimization of the total energy.[5,7] In these cases, a quadratic approximation around x^* is used to obtain the error vector as:

$$\Delta'''_i = x_i - x^* \approx Hg_i \quad (19)$$

where g_i is the gradient at x_i and H is an approximation to the inverse Hessian matrix at x^*.

We would like to point out that when a good approximation to the inverse Hessian H is not available, one can still improve convergence by using DIIS with the error vector $\Delta_n = g_n$ instead of $\Delta_n = Hg_n \approx (x_n - x^*)$. In this case, instead of minimizing $\|\tilde{x}_n - x^*\|$ via condition (5), one minimizes its estimated upper bound:

$$\|\tilde{x}_n - x^*\| \approx \left\| \sum_{i=i_0}^{n} c_i^{(n)} Hg_i \right\| \leq \|H\| \cdot \left\| \sum_{i=i_0}^{n} c_i^{(n)} g_i \right\| \to \min \quad (20)$$

As far as error vector (18) is concerned, it was implemented in the orbital-based DIIS we introduced recently.[6] Compared to error vector (16), it has the important advantage of requiring only one evaluation of the operator T per iteration (which, for an SCF process, means only one Fock matrix evaluation per iteration). It is tempting to consider error vector (18) as error vector (16) used at the

"wrong" iteration, since for a given x_n, x_{n+1}, one obtains $\Delta'_n = \Delta'''_{n+1}$. However, the connection between the three error vectors (16)–(18) and Banach's principle established in the previous section indicates that error vectors (16) and (18) correspond to the optimal $\tilde{T}$ operating on x_{n-2} and x_{n-1}, respectively, as already mentioned. Thus, if the original process (1) diverges, then x_{n-2} is closer to the solution x^* than x_{n-1} is, and it is advantageous to use error vector (16). Similarly, if process (1) converges slowly, then error vector (18) is preferable. This, however, is not a crucial issue, since Banach's principle shows the legitimacy of both.

It is also important to note a certain subtlety of the requirement for the error vector to vanish when convergence is achieved. One can argue that the error vector (18) does not satisfy this requirement, since one can "accidentally" arrive at the exact solution $x^* = x_n$, but in this case Δ'''_n is nonzero and iterations will continue. [Also, a scenario can be devised where DIIS employing error vector (16) continues even after visiting the exact solution x^*, if x_{n+1} "accidentally" equals x^*, but the method will produce some $\tilde{x}_n$ according to DIIS prescription (3)–(5), and the process will go on.] However, to guarantee uniqueness of a solution x^*, the operator T has to be uniformly monotonic and, hence, the inverse operator T^{-1} has to exist. In this case, it is not possible to arrive exactly at the solution x^* at some iteration n by means of process (1), since eq. (2) implies $T^{-1}(x^*) = x^*$, which would contradict $T^{-1}(x^*) = x_n$. It is possible indeed to arrive at the exact solution x^* after a finite number of iterations utilizing DIIS, but this can be achieved by virtue of $\tilde{x}_n = x^*$ only, and such a situation will be detected by either error vector considered.

From the error vector analysis carried out above, it is clear that there is a marked difference between error vectors (17) and (18), whereas error vectors (16), apart from their computational inefficiency, are similar to error vectors (18). Hence, from now on we will compare only error vectors (17) with error vectors (18). The difference mentioned above results from the fact that one can evaluate error vector (18) directly, while some approximation is required to obtain error vector (17). Additionally, the choice of error vector (18) is applicable to every iterative process of the type (1), and it does not involve the ambiguity associated with error vector (17) that can be formed in different ways depending on the particular approximation chosen. However, condition (5), which defines the DIIS coefficients, corresponds exactly to minimization of the norm of the error vector (17) associated with a new parameter vector $\tilde{x}_n$, while in the case of the error vector (18), this correspondence is approximate, since one has to assume that the operator T is linear within a neighborhood containing $x_{i_0}, \ldots, x_n$.

This trade-off indicates that both error vectors (17) and (18) should be roughly similar for a generic DIIS algorithm. However, for a particular DIIS implementation it would be desirable to estimate when and if the operator T can be considered linear (as mentioned previously, it can be linear even for nontrivial applications), and what quality of approximation used to evaluate error vector (17) is expected [e.g., based on the quality of the approximate inverse Hessian matrix, H, see eq. (19)]. Based on this information, one can make an intelligent error vector choice for a particular DIIS application, although such a choice will not be crucial *as long as it is either (17) or (18)*, as illustrated in the next section.

Examples

To test how the error vector choice effects DIIS performance, we consider several cases of molecular geometry optimization. Existing combinations of DIIS and molecular geometry optimization[5,7] have been implemented only for the quasi-Newton method without line searches. This corresponds to the iterative process:

$$x_{n+1} = x_n - H_n g_n \tag{21}$$

where x_n is a vector of geometrical parameters at iteration n, g_n is the corresponding gradient, and H_n is the approximate inverse of the Hessian matrix. The matrix H_n can be chosen as the unit matrix at every iteration,[7] or it can be updated based on H_{n-1}, x_n, x_{n-1}, g_n, and g_{n-1}, where H_1 is either the unit matrix or is calculated from empirical rules.[5] These applications of DIIS had initial guesses that were close to x^*; that is, the molecular geometry optimization was started from either an experimental or a low-level theoretical geometry; the guess for a wave function in Fischer and Almlöf[5] was, apparently, also close to the solution as indicated by the low (no more than 12) number of iterations required by Gaussian 88[14] to reach convergence in every case considered.

It is well known (see, e.g., Fletcher[15]) that the quasi-Newton method without line searches can

diverge when the initial guess x_1 is far from the equilibrium geometry x^* or when the initial guess for the inverse Hessian matrix is inadequate. To make method (21) more robust, one has to provide for a decrease of the total energy at each iteration.[15] This can be accomplished by adjusting the step λ_n along the direction $-H_n g_n$ by either finding a minimum along this direction or just taking a step that leads to some decrease in the total energy. (Note that if the matrix H_n is positive definite and $g_n \neq 0$, it is always possible to find such a step.) The former corresponds to the quasi-Newton method with exact line searches:

$$x_{n+1} = x_n - \lambda_n H_n g_n \tag{22}$$

$$\lambda_n: E(x_n - \lambda_n H_n g_n) \rightarrow \min \tag{23}$$

which is the only derivative-based method that is guaranteed to converge in no more than N iterations on a quadratic potential energy surface of dimension N, and where H_n is guaranteed to converge to the exact inverse Hessian. This method is also robust, since at early iterations, where the quadratic approximation is not valid, the equilibrium geometry is approached by virtue of the decrease of the total energy.

Based on the properties of the quasi-Newton method with line searches outlined above, it becomes clear that this method is already very efficient for a general problem of finding an equilibrium structure. Thus, the question arises what (if any) speedup can be gained from combining DIIS with this method. The extensive study described by Fischer and Almlöf[5] shows that when the unit matrix is used as an initial Hessian, DIIS results in at most a 43% speedup even for the less efficient quasi-Newton method without line searches, thus the effect of DIIS should be even less for the quasi-Newton method with line searches. The results from Császár and Pulay[7] indicate, however, that even when H_n is the unit matrix at every iteration, DIIS results in a factor of two or three fewer iterations than method (21) alone. We attribute this discrepancy to a rather low number of iterations (3–17) required to achieve convergence in all the cases considered in ref. 7; the range of the number of iterations for the cases studied in ref. 5 was 7–28.

We implement DIIS combined with the quasi-Newton method with line searches as follows. At the nth iteration, we find x_n according to eqs. (22) and (23), evaluate g_n, and update H_n according to the BFGS updating formula.[15] Then we form the DIIS linear combination $\tilde{x}_n$ according to eqs. (3)–(5), where for each optimization run we use either eq. (18) or eq. (19) to determine the error vector Δ_n. In the latter case, the most recent approximation to the inverse Hessian, H_n, is used to obtain the error vectors $\Delta'_i \approx H_n g_i$, $i = 1, \ldots, n$. Then we calculate the total energy at $\tilde{x}_n$ and, if it is lower than the one at x_n, we use $\tilde{x}_n$ instead of x_n and $\Sigma c_i g_i$ instead of g_n in eqs. (22) and (23) to obtain x_{n+1}; otherwise, we obtain x_{n+1} in the usual way from eqs. (22) and (23).

In the method described above, the total energy decreases at every iteration, which is an important property when optimization starts far from the optimal structure x^*. This is achieved at the expense of not allowing the DIIS procedure to take place if it results in a total energy increase. We used this method to find optimal structures for Na_6 and Si_2H_6, which have potential energy surfaces (PESs) of different kinds: the PES for the former is rather flat, while the PES for the latter has a pronounced potential well. The initial guesses for equilibrium geometries were far from the actual stable structures for both Na_6 and Si_2H_6, and the initial Hessians were taken as unit matrices. The total energy for Si_2H_6 was calculated at the HF/6-31G** level, and the total energy for Na_6 was calculated at the generalized valence bond level with perfect-pairing restrictions (GVB-PP),[16] where the basis set and effective core potential are from Melius and Goddard[17] and the six valence electrons were correlated as three GVB pairs with two natural orbitals per pair [GVB(3/6)-PP].

The results of these optimizations are presented in Table I. From this table, we see that DIIS can improve performance of not only the quasi-Newton method without line searches as has been established in refs. 5 and 7, but that it is also beneficial for the already very efficient quasi-Newton method with line searches, which is applicable to arbitrary initial geometries. The decrease in the number of iterations due to DIIS is similar for all choices of the error vectors presented in Table I, and it is comparable to the decrease in the number of iterations achieved via combining DIIS with the quasi-Newton method without line searches when the initial inverse Hessian is taken as the unit matrix and the optimization is started close to the equilibrium structure.[5]

The similarity of DIISs with the error vectors defined by eqs. (17) and (18) is, of course, the expected behavior, as was explained in the previous section. However, when we tried to use an error vector such as the maximum component of the gradient or the gradient's norm [which does

TABLE I.
Number of Iterations Required to Converge to Equilibrium Geometry as Function of Number of Terms in DIIS Expansion and Choice of Error Vector, Δ_i.

N_{store}[a]	$\Delta_i = H_n g_i \approx x_i - x^*$	$\Delta_i = x_i - x_{i-1}$	$\Delta_i = g_i$
Si_2H_6, "normal" PES			
0	32	32	32
1	19	27	20
2	28	23	28
3	31	39	20
4	21	18	38
Na_6, "flat" PES			
0	24	24	24
1	16	16	20
2	19	21	18
3	17	18	13
4	24	21	16

[a] N_{store} is the number of previous iterations used in the DIIS procedure, so that the number of terms in the DIIS expansion is $N_{store} + 1$.

not correspond to either of eqs. (17) or (18)], there was no DIIS linear combination (3) that yielded a total energy less than that at the regular point x_n at iteration n, and, thus, no acceleration due to DIIS was possible. [Note that in this case the dimension of the error vector is 1, and, hence, only DIIS with two terms in the expansion (3) makes sense.] Thus, these error vectors are not suitable choices for DIIS purposes, although they do vanish at convergence and are related to the gradient, while error vector choices made according to eqs. (17) or (18) result in improved performance.

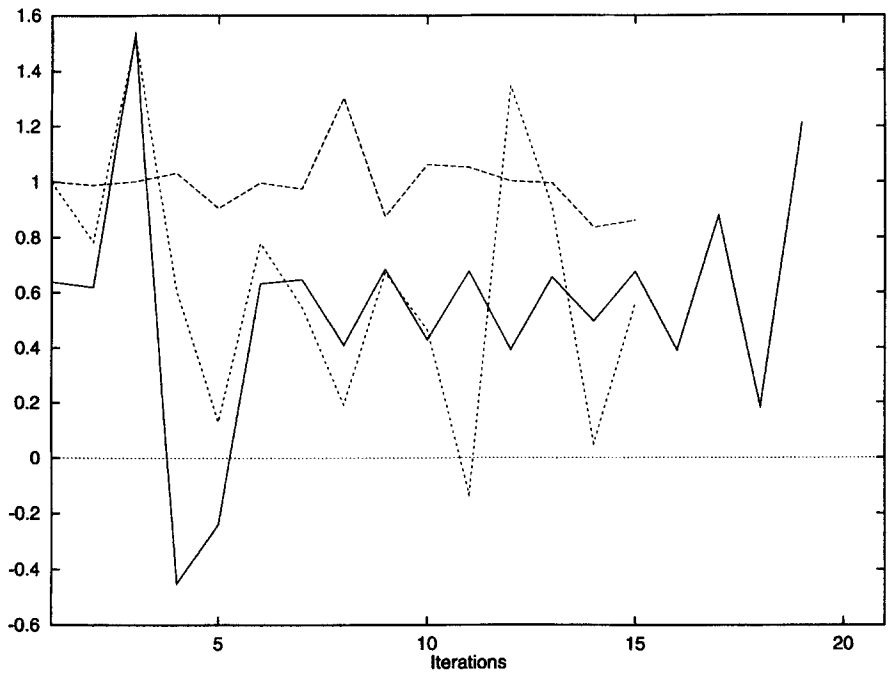

FIGURE 1. DIIS coefficient c_n obtained in the course of the simplest ($n = i_0 + 1$) DIIS combined with the quasi-Newton method with line searches. Solid, dashed, and dotted lines correspond to the error vectors g_n, $H_n g_n$, and $(x_n - x_{n-1})$, respectively. Optimization performed for a Na_6 cluster.

It was observed in ref. 5 that one has to store parameter and error vectors from about four previous iterations before DIIS becomes counterproductive. We found that this trend is even stronger for the PESs we considered where the most efficient DIIS expansion (3) consists of two or three terms, and, therefore, up to two previous parameter and error vectors are required. This is due to the fact that our initial guesses are far from the final structure, and hence the corresponding PESs are farther from being quadratic than the ones considered in ref. 5. The highly nonquadratic character of the PESs studied is also manifested by nonmonotonic dependence of the number of iterations required to converge to the equilibrium structure on the number of terms in the DIIS expansion (3). This results from the fact that the larger the value of N_{store}, the more counterproductive the corresponding DIIS will be at the beginning of the iterative process, but the more efficient it will be near convergence.

The similarity in performance of DIIS employing error vectors of the form $(x_n - x_{n-1})$, $H_n g_n$, or g_n follows from the similarity of the DIIS coefficients obtained for each of these choices, as illustrated in Figures 1–5. In these figures, we plot the DIIS coefficient c_n corresponding to the current parameter vector x_n for the simplest DIIS (i.e., $N_{store} = 1$, so that c_n is also an optimal damping coefficient). Figures 1 and 2, which correspond to DIIS combined with the quasi-Newton method with line searches, show that all three error vectors result in c_n being in the range $(-0.5, 1.5)$. Considering that the three error vectors utilized different iterative sequences, $x_1, \ldots, x_n, \ldots$, this agreement is rather good. Also note that c_n being in the area of 1 indicates that the quasi-Newton method with line searches is already very efficient, so that DIIS results only in slight corrections.

The strong dependence of the DIIS coefficients on the operator T and their weak dependence on the choice of error vector [made among $(x_n - x_{n-1})$, $H_n g_n$, and g_n] is illustrated in Figures 3–5. These figures correspond to DIIS combined with the quasi-Newton method without line searches, where an energy decrease on every iteration was accomplished by halving the step size until this condition was met. In this case, the operator T was not even smooth because of this procedure to ensure total energy lowering, but DIIS still resulted in a slight decrease in the number of iterations [36, 37, and 40 iterations for the simplest DIIS with error vectors g_n, $H_n g_n$, and $(x_n - x_{n-1})$, respectively, versus 42 iterations for the case of no DIIS].

When DIIS is turned off, the quasi-Newton method without line searches (but with energy

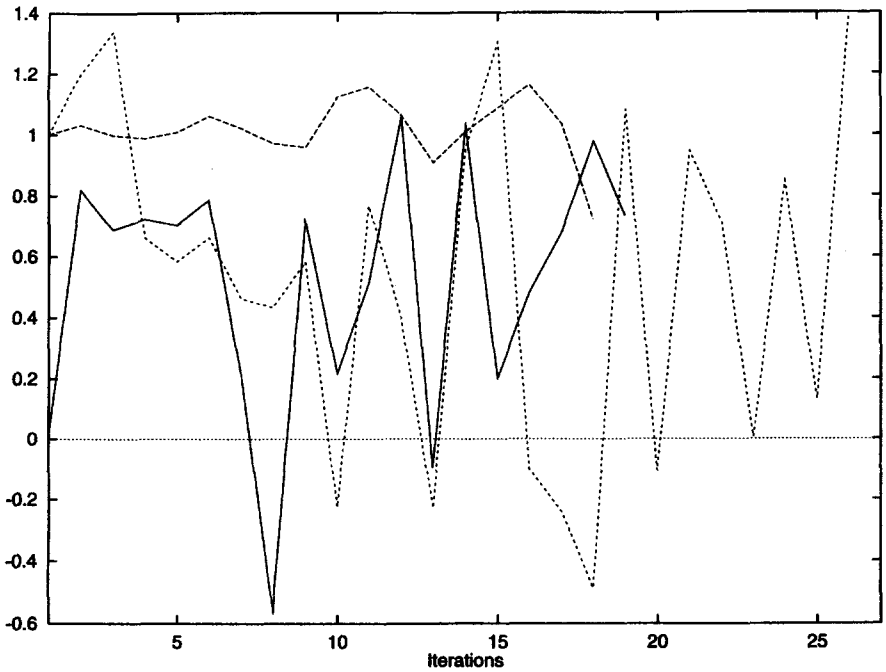

FIGURE 2. Same as Figure 1, for the Si_2H_6 molecule.

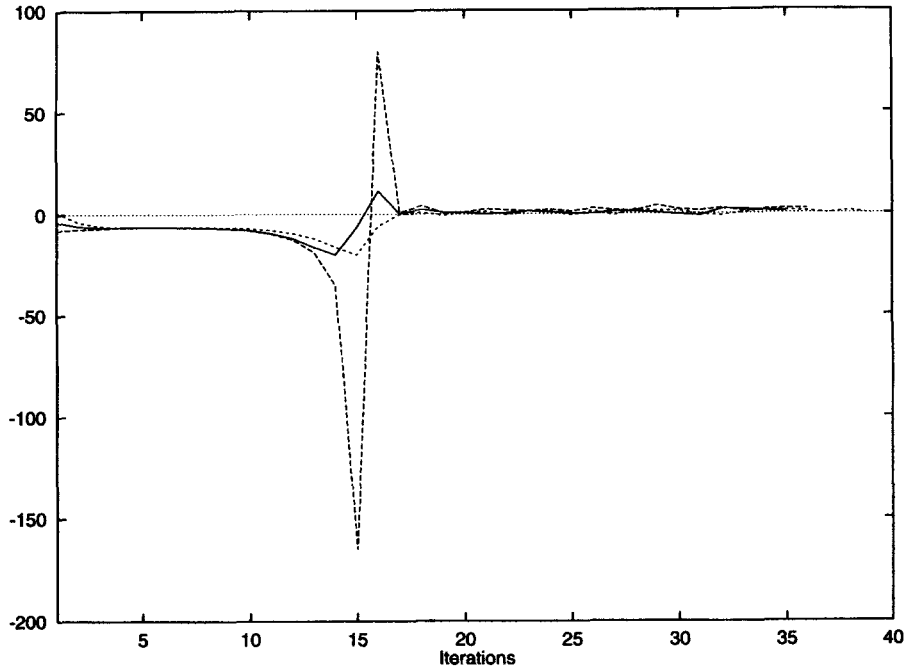

FIGURE 3. DIIS coefficient c_n obtained in the course of the simplest DIIS combined with the quasi-Newton method without line searches. Solid, dashed, and dotted lines correspond to the error vectors g_n, $H_n g_n$, and $(x_n - x_{n-1})$, respectively. Optimization performed for the Si_2H_6 molecule.

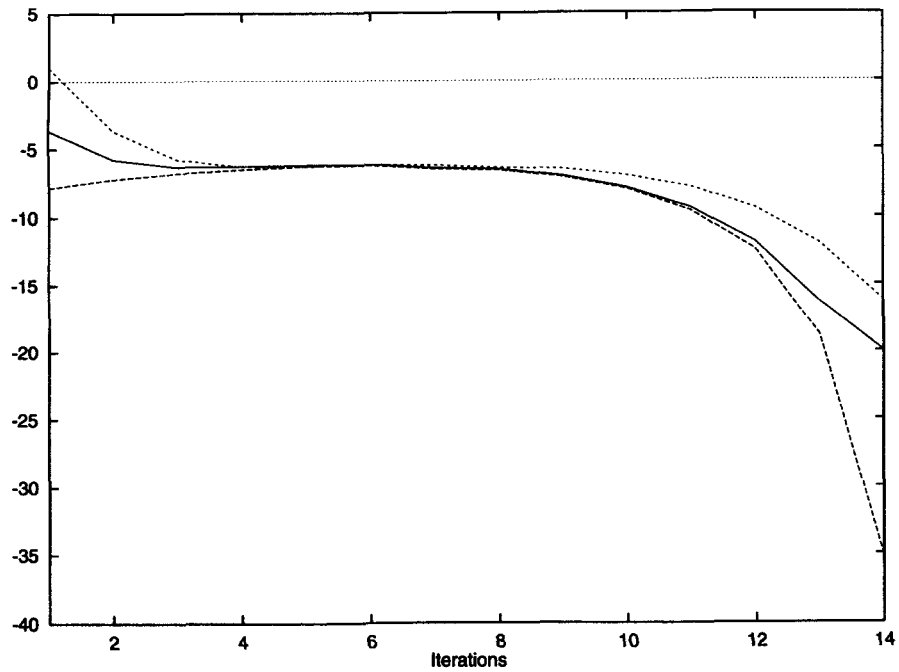

FIGURE 4. First 14 iterations of the process shown in Figure 3.

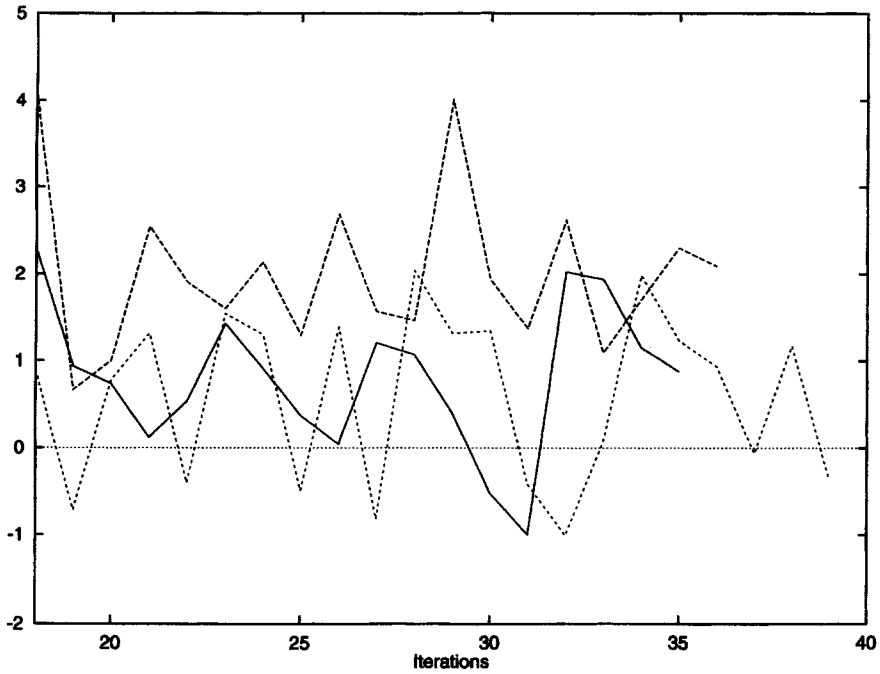

FIGURE 5. Iterations of the process shown in Figure 3, from iteration 18 onward.

lowering scheme described above) produces parameter vectors that are farther away from the solution as compared to those produced when line searches are employed. This is indicated not only by the higher number of iterations (42 vs. 32) required to achieve convergence starting from the same initial structure without DIIS, but also by the value of an optimal damping coefficient c_n that is very far from unity in the beginning of the process when the simplest DIIS is employed (see Fig. 3). At this stage of the iterations, the total energy at $\tilde{x}_n$ produced by DIIS was higher than the total energy at x_n, so that the $\tilde{x}_n$ were not included into iterative sequence, and, hence, the information used by DIIS to obtain c_n was the same for the three error vectors considered. In this case, the agreement between c_n obtained based on these different error vectors is remarkable (see Fig. 4). Toward the end of the iterative process, the quadratic region of the solution is reached and DIIS becomes productive; that is, $\tilde{x}_n$ are now accepted into iterative sequence and, hence, this sequence becomes different for every error vector (thus contributing to slight differences among DIIS coefficients obtained in the course of DIISs with different error vectors). Also, the matrix H_n in eq. (21) becomes a good approximation to the inverse Hessian matrix, yielding good quality parameter vectors from eq. (21), as indicated by the proximity of the optimal damping coefficient c_n to 1 (see Fig. 5).

Conclusions

We have shown that DIIS can be interpreted as consistently optimal damping, which allowed us to derive error vectors suitable for DIIS directly from Banach's principle. This principle not only provides support for the known error vectors (16) and (17), but it also confirms the legitimacy of the error vector (18) recently introduced by us.[6] This error vector is as easily constructed as the error vector (16), but it has the important advantage of being computationally efficient.

We would also like to note that one can accelerate convergence of an iterative process (1) by means of damping whenever the operator T is uniformly monotonic. When T is linear (at least locally, within a neighborhood of a current parameter vector), this scheme becomes equivalent to DIIS. From this we conclude that the cases where DIIS was found to be useful in early iterations[3] involved (near)linear operators T. (In fact, ref. 3 deals with DIIS for a small multiconfigurational SCF where the linearity of T is, probably, a good approximation given that T is rigorously linear for closed-shell SCF[13]).

Our analysis of the error vectors revealed the approximations associated with them. Indeed, one can evaluate error vectors (16) and (18) directly as a difference between consecutive parameter vectors, but in this case the DIIS condition (5) corresponds to minimization of the error vector norm only within a linear approximation of the operator T. On the other hand, error vector (17) can only be estimated based on some approximation, but in this case, condition (5) describes the error vector norm minimization exactly. Due to this trade-off, the effect of a generic DIIS should be roughly similar for either type of error vector used.

We have illustrated the expected weak dependence of DIIS performance and DIIS coefficients on the choice of (proper) error vector that we examined [i.e., either $(x_n - x_{n-1})$, $H_n g_n$, or g_n] and, simultaneously, their strong dependence on the operator T in several examples. These examples involved a novel application of DIIS to the quasi-Newton method with line searches to allow an optimization to start from an arbitrary initial guess.

We have also shown that other error vector choices that are related to the gradient and vanish at convergence (but do not follow from Banach's principle) do not result in performance improvement due to DIIS.

As a final remark, it is the error vector analysis presented here that enabled us to significantly improve the performance of our Ridge method for transition states searches[18] where the error vector used was $\Delta_n = g_n$. We also expect that the use of the new error vector $\Delta'''_n = (x_n - x_{n-1})$ will allow application of the DIIS to problems not currently amenable to error vectors Δ' and Δ'', for example, orbital-free density-functional methods.[19]

Acknowledgments

We are grateful to the Office of Naval Research for primary support of this work. E. A. C. also acknowledges support from the Camille and Henry Dreyfus Foundation and the Alfred P. Sloan Foundation via a Teacher-Scholar Award and a Research Fellowship, respectively.

Appendix

Let us show that consistently optimal damping with appropriate choices of x and y in condition (9) is equivalent to general DIIS when the operator T is linear. First, it is easy to see that eqs. (3) and (4) describe not just DIIS but also damping, when the latter is performed at every iteration. Indeed, consider the following damping process starting from an initial guess $x_0 = \tilde{x}_0$:

$$x_n = T(\tilde{x}_{n-1}) \tag{24}$$

$$\tilde{x}_n = \tilde{T}_n(\tilde{x}_{n-1}) = \tau_n x_n + (1 - \tau_n)\tilde{x}_{n-1} \tag{25}$$

where $\tau_1 = 1$. Then, by denoting:

$$c_n^{(n)} = \tau_n, \qquad c_i^{(n)} = (1 - \tau_n)c_i^{(n-1)}, \qquad i < n \tag{26}$$

one can easily obtain eqs. (3) and (4) that describe DIIS.

Now, to make the comparison between DIIS and damping complete, we need to consider the relationship between condition (5) (minimization of error) and condition (9) (maximization of the contraction of the operator $\tilde{T}$) that determine the DIIS coefficients, $c_i^{(k)}$, and the damping coefficients, τ_i, respectively. However, since condition (5) is formulated in terms of error vectors, Δ_i, it is the error vector choice that determines the essence of this condition.

Since the error vector (17) is linear with respect to x_i, we immediately obtain that conditions (5) and (9) are equivalent when the latter uses $x = \tilde{x}_{n-1}$ and $y = x^*$:

$$\left\| \sum_{i=i_0}^{n} c_i^{(n)} \Delta'''_i \right\| = \left\| \sum_{i=i_0}^{n} c_i^{(n)} (x_i - x^*) \right\| = \| \tilde{x}_n - x^* \|$$

$$= \left\| \tilde{T}_n(\tilde{x}_{n-1}) - \tilde{T}_n(x^*) \right\| \to \min \tag{27}$$

Thus, when using error vector (17), the same DIIS coefficients will be obtained from either minimization of the norm of the error vector associated with $\tilde{x}_n$ in (5) or from maximizing the contraction of the operator $\tilde{T}_n$ based on $x = \tilde{x}_{n-1}$ and $y = x^*$ in (9).

For a linear operator T [and, hence, linear error vector $\Delta'''_i = \Delta'''(x_i)$, see eq. (18)], we obtain, based on eqs. (3), (4), (24), (25), and (18):

$$\left\| \sum_{i=i_0}^{n} c_i^{(n)} \Delta'''_i \right\| = \left\| \Delta''' \left(\sum_{i=i_0}^{n} c_i^{(n)} x_i \right) \right\| = \| \Delta'''(\tilde{x}_n) \| \tag{28}$$

$$= \| \tilde{x}_n - T^{-1}(\tilde{x}_n) \|$$

$$= \| \tilde{x}_n - (\tau_n T^{-1} T(\tilde{x}_{n-1}) + (1 - \tau_n) T^{-1}(\tilde{x}_{n-1})) \| \tag{29}$$

If, in addition to x_n and $\tilde{x}_n$ that determine the damping process (24)–(25), we introduce auxiliary

vectors $\hat{x}_n$ formed as:

$$\hat{x}_n = \tau_n \tilde{x}_{n-1} + (1 - \tau_n)\hat{x}_{n-1} \tag{30}$$

where $\hat{x}_0 = x_0$, then it is easy to show that $T\hat{x}_n = \tilde{x}_n$ and $\tilde{T}_n(\hat{x}_{n-1}) = \hat{x}_n$, and, thus, by continuing eq. (29) we obtain:

$$\left\| \sum_{i=i_0}^{n} c_i^{(n)} \Delta_i'' \right\| = \| \tilde{x}_n - \hat{x}_n \| = \left\| \tilde{T}_n(\tilde{x}_{n-1}) - \tilde{T}_n(\hat{x}_{n-1}) \right\| \tag{31}$$

Thus, whenever condition (5) can be treated as minimization of the norm of error vector (18) associated with $\tilde{x}_n$, it can also be interpreted as maximization of contraction of operator $\tilde{T}_n$ based on using $x = \tilde{x}_{n-1}$ and $y = \hat{x}_{n-1}$ in (9). A similar relationship is easy to establish for error vector (16) by modifying eqs. (28)–(31) to account for the use of eq. (16) instead of (18); we do not present these redundant derivations here.

References

1. P. Pulay, *Chem. Phys. Lett.*, **73**, 393 (1980).
2. P. Pulay, *J. Comput. Chem.*, **3**, 556 (1982).
3. T. P. Hamilton and P. Pulay, *J. Chem. Phys.*, **84**, 5728 (1986).
4. R. P. Muller, J.-M. Langlois, M. Ringnalda, R. A. Friesner, and W. A. Goddard III, *J. Chem. Phys.*, **100**, 1226 (1994).
5. T. H. Fischer and J. Almlöf, *J. Phys. Chem.*, **96**, 9768 (1992).
6. I. V. Ionova and E. A. Carter, *J. Chem. Phys.*, **102**, 1251 (1995).
7. P. Császár and P. Pulay, *J. Mol. Struct.*, **114**, 31 (1984).
8. W. B. Neilsen, *Chem. Phys. Lett.*, **18**, 225 (1973).
9. Y. G. Evtushenko, *Numerical Optimization Techniques*, Optimization Software, Inc., Publication Division, New York, 1985.
10. P. Pulay, *Theor. Chim. Acta*, **50**, 299 (1979).
11. H. Hsu, E. R. Davidson, and R. M. Pitzer, *J. Chem. Phys.*, **65**, 609 (1976).
12. C. C. J. Roothaan and P. S. Bagus, *Meth. Comput. Phys.*, **2**, 62 (1963).
13. R. E. Stanton, *J. Chem. Phys.*, **75**, 5416 (1981).
14. M. J. Frisch, M. Head-Gordon, H. B. Schlegel, K. Raghavachari, J. S. Binkley, C. Gonzalez, D. J. DeFrees, D. J. Fox, R. A. Whiteside, R. Seeger, C. F. Melius, J. Baker, L. R. Kahn, J. J. P. Stewart, E. M. Fluder, S. Topiol, and J. A. Pople, GAUSSIAN 88, Gaussian, Inc., Pittsburgh, PA, 1988.
15. R. Fletcher, *Practical Methods of Optimization*, Wiley, New York, 1987.
16. F. W. Bobrowicz and W. A. Goddard III, In *Modern Theoretical Chemistry*, H. F. Schaefer, Ed., Plenum Press, New York, 1977, p. 79.
17. C. F. Melius and W. A. Goddard III, *Phys. Rev. A*, **10**, 1528 (1974).
18. I. V. Ionova and E. A. Carter, *J. Chem. Phys.*, **103**, 5437 (1995).
19. M. Pearson, I. Smargiassi, and P. A. Madden, *J. Phys. Condens. Matter*, **5**, 3321 (1993).