JCTC Journal of Chemical Theory and Computation

Article

pubs.acs.org/JCTC

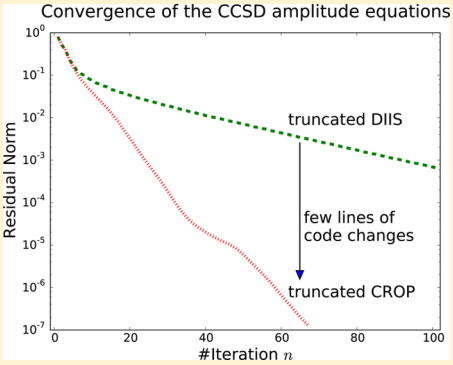
Discarding Information from Previous Iterations in an Optimal Way To Solve the Coupled Cluster Amplitude Equations

Patrick Ettenhuber* and Poul Jørgensen

qLEAP Center for Theoretical Chemistry, Department of Chemistry, Aarhus University, Langelandsgade 140, DK−8000 Aarhus C, Denmark

Supporting Information

ABSTRACT: The direct inversion of the iterative subspace (DIIS) convergence acceleration algorithm is used in most electronic structure programs to solve the nonlinear coupled cluster amplitude equations. When the DIIS algorithm is used, the storage of previous trial vectors may become a bottleneck and the discarding of trial vectors may lead to a degradation of the convergence or even divergence. We discuss an alternative way of storing information from trial vectors where only the last three trial vectors are needed to maintain the convergence of the full set of previous trial vectors, and which requires only minor modifications of an existing DIIS code.



INTRODUCTION

The coupled cluster amplitude equations constitute a set of nonlinear equations that, in most electronic structure programs, are solved using the direct inversion of the iterative subspace (DIIS) algorithm.[1] To solve the coupled pair and cluster amplitude equations, the DIIS algorithm was introduced by Ahlrichs et al.[2] and Scuseria et al.,[3] respectively. In the DIIS algorithm, trial vectors and corresponding vector function values (error vectors) from the previous iterations are stored and used to obtain an improved trial vector of the next iteration. The storage requirement of these trial vectors may become a bottleneck in the DIIS algorithm. To circumvent this bottleneck, the trial vectors of the initial iterations have been discarded, storing only the trial vectors of the last iterations. However, this may lead to a serious abatement in the convergence and even divergence when the DIIS algorithm is used.

In this paper, we discuss an optimal way of discarding information from the previous iterations of a DIIS sequence when solving the CC (coupled cluster) amplitude equations. In particular, we describe how a DIIS sequence may be turned into a conjugate residual with optimal trial vectors (CROP) sequence by modifying a few lines of code. If no trial vectors are discarded, the CROP sequence and the DIIS sequence become identical. However, the advantage of the CROP sequence versus the DIIS sequence is that for the CROP sequence and a set of linear equations information from all but the last three iterations may be discarded without affecting the convergence, while doing the same in a DIIS sequence may strongly affect the convergence. This means that, for a set of nonlinear equations, the CROP algorithm will allow trial vectors to be stored in an optimal way, with respect to the linear part of the equations. When CC theory is applied for a state dominated by a single configuration, the CC equations have a weak nonlinearity and, in practice, solving these only requires information from the last three trial vectors. However, for a more pronounced nonlinearity, it also is sufficient to keep the information from the last three trial vectors, as will be demonstrated.

The CROP algorithm was introduced in 2008 by Ziolkowski et al.[4] and despite the obvious advantage of using the CROP algorithm, compared to the DIIS algorithm, and considering the fact that a standard DIIS implementation can be changed to a CROP implementation by changing a few lines of code, it is surprising that the CROP algorithm is not implemented in any of the standard electronic structure codes. We ascribe this to the fact that, when the CROP algorithm was introduced, it was difficult to recognize, from the derivation, that a DIIS implementation could be changed to a CROP implementation, by modifying only a few lines of code. We present a simplified route for deriving the CROP algorithm, that clearly demonstrates how a DIIS implementation may be changed to become a CROP implementation. When the CROP algorithm was introduced, it was not applied for solving standard CC singles and doubles (CCSD) amplitude equations but for CCSD equations in the projected Löwdin orthogonalized atomic orbital (PLAO) basis.[5] We report applications of the CROP algorithm where the standard CCSD amplitude equations are solved.

When solving a set of linear equations $Ax + b = 0$, two algorithms—the conjugate gradient (CG) and the conjugate

Received: December 11, 2014
Published: March 20, 2015

DOI: 10.1021/ct501114q
J. Chem. Theory Comput. 2015, 11, 1518−1524

residual (CR) algorithms—share the common features, that information from the last three iterations are sufficient to maintain the convergence of the algorithm. In the CG method, this unique feature is obtained as a result of minimizing the quadratic function $f(\boldsymbol{x}) = (^1/_2)\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{b}$, while for the CR algorithm, it is obtained as a result of minimizing the residual norm $\mathbf{r}^T\mathbf{r}$ (with $\mathbf{r} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$). For the CG algorithm, it is the conjugacy of the directions, with respect to the matrix $\mathbf{A}$, that makes the subspace truncations possible, whereas, for the CR algorithm, it is the conjugacy with respect to the matrix $\mathbf{A}^T\mathbf{A}$.

The CG algorithm was introduced into quantum chemistry more than 30 years ago by Wormer, Visser, and Paldus.[6] When the configuration interaction (CI) eigenvalue equation is solved using an iterative algorithm, the convergence in the energy is quadratic in the error in the eigenvector. Van Lenthe and Pulay used this fact to transform the iterative solution of the CI eigenvalue equation to the solution of a set of linear equations.[7] Furthermore, by using the CG algorithm to solve the set of linear equations, they showed that the iterative solution of the CI eigenvalue equation could be expressed in a form where only information from the last three iterations is required to maintain the convergence of the full iterative subspace.[7] To solve sets of nonlinear equations, Ziolkowski et al.[4] introduced the CROP algorithm as a subspace generalization of the CR algorithm and showed the connection between the CROP and DIIS algorithms.

To describe the simplified route for deriving the CROP algorithm, we start out summarizing the properties of the CR subspace method. We then consider a standard DIIS implementation, highlighting the connection to the CR subspace method, and we describe how the DIIS algorithm may be changed to a CR subspace implementation, and thereby turned into the CROP algorithm. Finally, numerical results are presented for solving the nonlinear CCSD equations, demonstrating the advantage of using the CROP algorithm compared to the DIIS algorithm.

SOLUTION OF THE NONLINEAR COUPLED CLUSTER AMPLITUDE EQUATIONS

The CC amplitude equations[8] constitute a set of nonlinear equations in the cluster parameters $\mathbf{t}$

$$\langle \boldsymbol{\mu}|\exp(-T)H\exp(T)|\mathrm{HF}\rangle = \mathbf{0} \tag{1}$$

where T is the cluster operator, H is the molecular Hamiltonian, $|\mathrm{HF}\rangle$ is the Hartree−Fock state, and $|\boldsymbol{\mu}\rangle$ is an excitation manifold. By introducing the vector function

$$\mathbf{w}(\mathbf{t}) = \langle \boldsymbol{\mu}|\exp(-T)H\exp(T)|\mathrm{HF}\rangle \tag{2}$$

our task is to determine the solution vector $\mathbf{t}^*$, which satisfies

$$\mathbf{w}(\mathbf{t}^*) = \mathbf{0} \tag{3}$$

employing an iterative algorithm.

In iteration n of an algorithm for solving eq 3, we assume that a trial solution $\mathbf{t}^{(n)}$ is known and try to find an update $\Delta\mathbf{t}^{(n)}$ that gives an improved trial solution of iteration $n + 1$:

$$\mathbf{t}^{(n+1)} = \mathbf{t}^{(n)} + \Delta\mathbf{t}^{(n)} \tag{4}$$

Expanding the vector function $\mathbf{w}(\mathbf{t})$ around $\mathbf{t}^{(n)}$ yields

$$\mathbf{w}(\mathbf{t}) = \mathbf{w}^{(n)} + \mathbf{W}^{(n)}\Delta\mathbf{t} + O(\|\Delta\mathbf{t}^2\|) \tag{5}$$

where $\mathbf{w}^{(n)}$ and $\mathbf{W}^{(n)}$ are the vector function and the Jacobian at $\mathbf{t}^{(n)}$, respectively, and

$$\Delta\mathbf{t} = \mathbf{t} - \mathbf{t}^{(n)} \tag{6}$$

we may determine an iteration function for the update $\Delta\mathbf{t}^{(n)}$ by setting eq 5 equal to zero and neglecting the second-order terms $O(\|\Delta\mathbf{t}^2\|)$,

$$\mathbf{w}^{(n)} + \mathbf{W}^{(n)}\Delta\mathbf{t} = \mathbf{0} \tag{7}$$

The accuracy of the solution vector $\mathbf{t}$ for the set of linear equations described by eq 7 may be measured by the residual

$$\mathbf{r}^{(n)}(\mathbf{t}) = \mathbf{w}^{(n)} + \mathbf{W}^{(n)}(\mathbf{t} - \mathbf{t}^{(n)}) \tag{8}$$

When the linear equations in eq 7 are solved and $\mathbf{t}^{(n+1)}$ is determined such that

$$\left\|\mathbf{r}^{(n)}(\mathbf{t}^{(n+1)})\right\|^2 = 0 \tag{9}$$

the iterative procedure defined by eq 4 will be quadratically convergent.

The linear equations eq 7 are usually solved using an iterative procedure. Standard iterative algorithms, as the conjugate gradient (CG) and conjugate residual (CR) methods[9] require that a linear transformation of the Jacobian on a trial vector $\mathbf{W}^{(n)}\mathbf{t}_{\mathrm{trial}}$, is calculated in each iteration. Since (i) such a linear transformation is as costly as the evaluation of the vector function $\mathbf{w}(\mathbf{t}_{\mathrm{trial}})$ and (ii) a large number of iterations are required to obtain a solution vector to eq 7 that satisfies eq 9, in practice, it is too costly to use a quadratically convergent algorithm to obtain $\mathbf{t}^*$.

In the next section, we summarize the CR subspace algorithm for solving the linear equations in eq 7. In the subsequent section, we describe a conventional DIIS implementation and how it may be viewed in the context of the iterative scheme developed in this section. This then allows us to show how the DIIS algorithm may be modified to become a CR subspace algorithm that in turn identifies the CROP algorithm.

A SUBSPACE IMPLEMENTATION OF THE CONJUGATE RESIDUAL ALGORITHM

We now summarize the CR subspace algorithm for solving the set of linear equations in eq 7. We assume that we have carried out p iterations of a CR subspace algorithm and we thus know the trial solution $\mathbf{t}_{[p]}$ and the optimal directions of the previous iterations

$$P_{[p-1]} = \{\mathbf{p}_{[1]}, \mathbf{p}_{[2]}, ..., \mathbf{p}_{[p-1]}\} \tag{10}$$

where $\mathbf{p}_{[i]} = \mathbf{t}_{[i]} - \mathbf{t}_{[i-1]}$, for $i = 1, ..., p-1$, ($\mathbf{t}_{[0]} = \mathbf{t}^{(n)}$). Since the $\mathbf{p}_{[i]} = \mathbf{t}_{[i]} - \mathbf{t}_{[i-1]}$ are optimal directions (including length) in iteration i, the trial vectors $\mathbf{t}_{[i]}$ may be interpreted as optimal points. We further know the Jacobian linear transformed vectors $\mathbf{s}_{[i]} = \mathbf{W}^{(n)}\mathbf{p}_{[i]}$ for the optimal directions in eq 10:

$$S_{[p-1]} = \{\mathbf{s}_{[1]}, \mathbf{s}_{[2]}, ..., \mathbf{s}_{[p-1]}\} \tag{11}$$

Iteration p starts with generating the residual for the trial solution

$$\mathbf{r}_{[p]} = \mathbf{r}^{(n)}(\mathbf{t}_{[p]}) = \mathbf{w}^{(n)} + \mathbf{W}^{(n)}(\mathbf{t}_{[p]} - \mathbf{t}^{(n)}) \tag{12}$$

The trial solution of CR subspace iteration $p + 1$ may be parametrized as

$$\mathbf{t}_{[p+1]} = \mathbf{t}_{[p]} + \sum_{i=1}^{p-1} \alpha_i\mathbf{p}_{[i]} + \alpha_p\mathbf{r}_{[p]} \tag{13}$$

DOI: 10.1021/ct501114q
J. Chem. Theory Comput. 2015, 11, 1518−1524

Therefore, the optimal search direction of iteration p is

$$\mathbf{p}_{[p]} = \sum_{i=1}^{p-1} \alpha_i \mathbf{p}_{[i]} + \alpha_p \mathbf{r}_{[p]} \tag{14}$$

and may be determined minimizing the residual norm $\|\mathbf{r}_{[p+1]}\|^2$. The minimization may be carried out solving a set of linear equations (see ref 4 for details)

$$\begin{pmatrix} \langle \mathbf{s}_{[1]}|\mathbf{s}_{[1]}\rangle & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 \\ \vdots & \langle \mathbf{s}_{[p-2]}|\mathbf{s}_{[p-2]}\rangle & 0 & 0 \\ 0 & 0 & \langle \mathbf{s}_{[p-1]}|\mathbf{s}_{[p-1]}\rangle & \langle \mathbf{s}_{[p-1]}|\mathbf{b}_{[p]}\rangle \\ 0 & 0 & \langle \mathbf{b}_{[p]}|\mathbf{s}_{[p-1]}\rangle & \langle \mathbf{b}_{[p]}|\mathbf{b}_{[p]}\rangle \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_{p-2} \\ \alpha_{p-1} \\ \alpha_p \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \langle \mathbf{r}_{[p]}|\mathbf{b}_{[p]}\rangle \end{pmatrix} \tag{15}$$

where

$$\mathbf{b}_{[p]} = \mathbf{W}^{(n)} \mathbf{r}_{[p]} \tag{16}$$

$\mathbf{p}_{[p]}$ and $\mathbf{t}_{[p+1]}$ may now be determined and $\mathbf{s}_{[p]} = \mathbf{W}^{(n)}\mathbf{p}_{[p]}$ can be calculated using eq 16 and $\mathbf{s}_{[i]}$ (for $i = 1, ..., p-1$) can be determined from eq 11. $\mathbf{p}_{[p]}$ and $\mathbf{s}_{[p]}$ may now be added to $P_{[p-1]}$ and $S_{[p-1]}$ and the sequence of CR subspace iterations is thus established.

The important feature of the CR subspace algorithm is the structure of the subspace equation (eq 15), where the solution is not dependent on the initial $p-2$ optimal trial directions. Therefore, the trial vectors from these $p-2$ iterations may be discarded without affecting the subspace solution. It is this feature that may be taken over in the DIIS sequence of iterations when it is transferred to a CROP sequence.

To understand how a DIIS sequence of iterations may be modified to a CR subspace sequence, we write the CR subspace equation in eq 15 in a slightly modified form. The CR subspace equations arise from a minimization of the residual norm in the basis spanned by $\mathbf{p}_{[i]}$ (for $i = 1, ..., p-1$) and $\mathbf{r}_{[p]}$ (see eqs 14 and 15). Since the norm of a vector is independent of the basis in which it is expressed, any basis that spans the same space produces the same trial solution $\mathbf{t}_{[p+1]}$. Below, we describe a transformation to a basis that makes it simple to understand the connection between the DIIS and CR subspace algorithm.

Considering $\mathbf{t}_{[p+1]}$ in eq 13 and introducing a preliminary improvement,

$$\tilde{\mathbf{t}}_{[p+1]} = \mathbf{t}_{[p]} + \mathbf{r}_{[p]} \tag{17}$$

we may rewrite eq 13 as

$$\begin{aligned} \mathbf{t}_{[p+1]} &= \mathbf{t}_{[p]} + \sum_{i=1}^{p-1} \alpha_i(\mathbf{t}_{[i+1]} - \mathbf{t}_{[p]} + \mathbf{t}_{[p]} - \mathbf{t}_{[i]}) + \alpha_p(\tilde{\mathbf{t}}_{[p+1]} - \mathbf{t}_{[p]}) \\ &= \mathbf{t}_{[p]} + \sum_{i=2}^{p-1} (\alpha_i - \alpha_{i+1})(\mathbf{t}_{[i]} - \mathbf{t}_{[p]}) - \alpha_1(\mathbf{t}_{[1]} - \mathbf{t}_{[p]}) \\ &\quad + \alpha_p(\tilde{\mathbf{t}}_{[p+1]} - \mathbf{t}_{[p]}) \end{aligned} \tag{18}$$

By introducing a new set of variational parameters c_i (for $i = 1, ..., p-1$) and c_{p+1}, we may rewrite eq 18 as

$$\mathbf{t}_{[p+1]} = \mathbf{t}_{[p]} + \sum_{i=1}^{p-1} c_i(\mathbf{t}_{[i]} - \mathbf{t}_{[p]}) + c_{p+1}(\tilde{\mathbf{t}}_{[p+1]} - \mathbf{t}_{[p]}) \tag{19}$$

which describes a linear parametrization of the basis

$$T^{\mathrm{CR}}_{[p+1]} = \{\mathbf{t}_{[1]}, ..., \mathbf{t}_{[p]}, \tilde{\mathbf{t}}_{[p+1]}, \} \tag{20}$$

with $\mathbf{t}_{[p]}$ as the reference point. From the above derivation, we see that $T^{\mathrm{CR}}_{[p+1]}$ spans the same space as that used to determine the optimal trial vector in eq 14.

The residual of $\mathbf{t}_{[p+1]}$ in eq 19 becomes

$$\mathbf{r}_{[p+1]} = \mathbf{r}_{[p]} + \sum_{i=1}^{p-1} c_i(\mathbf{r}_{[i]} - \mathbf{r}_{[p]}) + c_{p+1}(\tilde{\mathbf{r}}_{[p+1]} - \mathbf{r}_{[p]}) \tag{21}$$

where

$$\tilde{\mathbf{r}}_{[p+1]} = \mathbf{w}^{(n)} + \mathbf{W}^{(n)}(\tilde{\mathbf{t}}_{[p+1]} - \mathbf{t}^{(n)}) \tag{22}$$

Equation 21 may be written as

$$\begin{aligned} \mathbf{r}_{[p+1]} &= (1 - \sum_{i=1}^{p-1} c_i - c_{p+1})\mathbf{r}_{[p]} + \sum_{i=1}^{p-1} c_i \mathbf{r}_{[i]} + c_{p+1}\tilde{\mathbf{r}}_{[r+1]} \\ &= \sum_{i=1}^{p} c_i \mathbf{r}_{[i]} + c_{p+1}\tilde{\mathbf{r}}_{[p+1]} \end{aligned} \tag{23}$$

where we have introduced the coefficient

$$c_p = 1 - \sum_{i=1}^{p-1} c_i - c_{p+1} \tag{24}$$

Minimizing $\|\mathbf{r}_{[p+1]}\|^2$ of eq 23, with the constraint eq 24 will give the same subspace solution as solving eq 15, since the subspaces of eqs 13 and 19 are identical.

From eq 15 we know, that the optimal solution is spanned by $\mathbf{r}_{[p]}$ and $\mathbf{p}_{[p-1]} = \mathbf{t}_{[p]} - \mathbf{t}_{[p-1]}$. The optimal trial vector is thus contained in the space that is spanned by $\mathbf{t}_{[p]}$, $\mathbf{t}_{[p-1]}$, and $\mathbf{r}_{[p]} = \tilde{\mathbf{t}}_{[p+1]} - \mathbf{t}_{[p]}$. Therefore, the coefficients c_i (for $i = 1, ..., p-2$) are vanishing and the optimal coefficients may thus be determined in the subspace,

$$T^{\mathrm{Red}}_{[p+1]} = \{\mathbf{t}_{[p-1]}, \mathbf{t}_{[p]}, \tilde{\mathbf{t}}_{[p+1]}\} \tag{25}$$

by minimizing the residual

$$\mathbf{r}_{[p+1]} = c_{p-1}\mathbf{r}_{[p-1]} + c_p\mathbf{r}_{[p]} + c_{p+1}\tilde{\mathbf{r}}_{[p+1]} \tag{26}$$

where the variational coefficients satisfy

$$c_{p-1} + c_p + c_{p+1} = 1 \tag{27}$$

In iteration p of a conjugate residual subspace iteration, the subspace solution may thus be obtained either from a basis containing $\mathbf{p}_{[p-1]}$ and $\mathbf{r}_{[p]}$ solving eq 15 or from the basis $T^{\mathrm{Red}}_{[p+1]}$ minimizing the residual norm $\|\mathbf{r}_{[p+1]}\|^2$ of eq 26 with the constraint described by eq 27.

THE DIIS ALGORITHM AND ITS CONNECTION TO A SUBSPACE CONJUGATE RESIDUAL ALGORITHM

When the DIIS method is used to determine the solution vector $\mathbf{t}^*$ in eq 3, the vector function value $\mathbf{w}(\mathbf{t})$ may be

DOI: 10.1021/ct501114q
J. Chem. Theory Comput. 2015, 11, 1518−1524

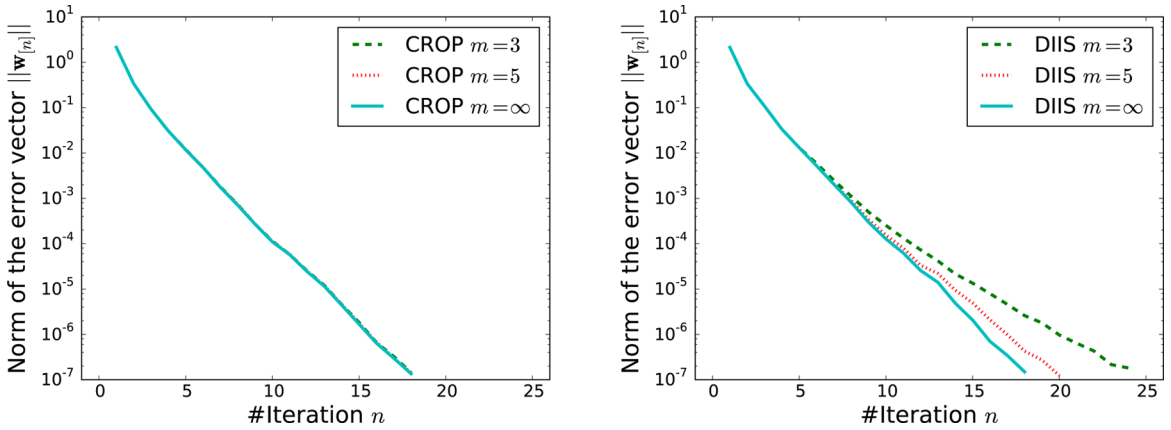


Figure 1. CROP (left) and DIIS (right) iteration sequences for a CCSD calculation on anthracene using a cc-pVDZ basis in the canonical basis for the subspace truncation parameters m = 3, 5, ∞.

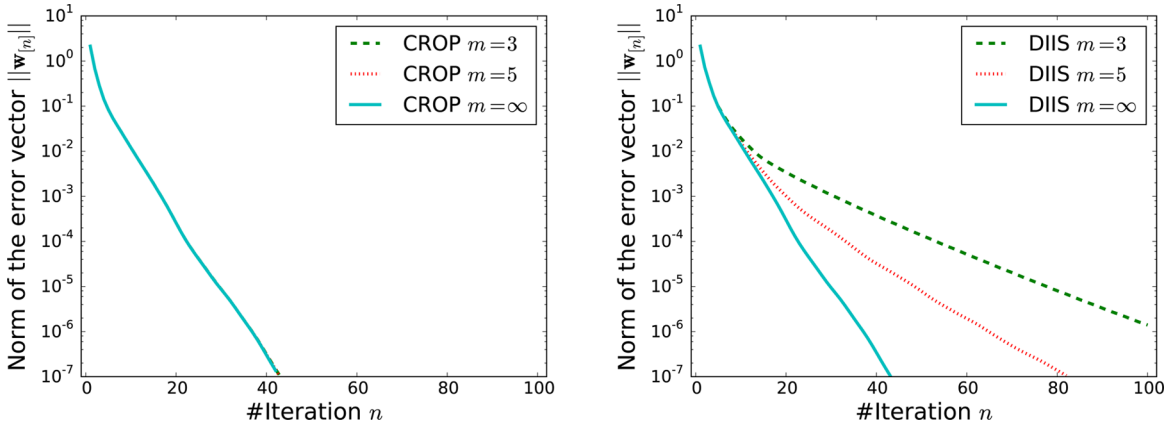


Figure 2. CROP (left) and DIIS (right) iteration sequences for a CCSD calculation on anthracene using a cc-pVDZ basis in the local basis[11] for the subspace truncation parameters m = 3, 5, ∞.

interpreted as an error vector, since $\mathbf{w}(\mathbf{t}) - \mathbf{w}(\mathbf{t}^*) = \mathbf{w}(\mathbf{t})$. At iteration p of a DIIS sequence, the sets of previous trial vectors

$$T_{[p]}^{\mathrm{DIIS}} = \{\mathbf{t}_{[1]}, \mathbf{t}_{[2]}, ..., \mathbf{t}_{[p]}\} \tag{28}$$

and their associated error vectors

$$W_{[p]}^{\mathrm{DIIS}} = \{\mathbf{w}_{[1]}, \mathbf{w}_{[2]}, ..., \mathbf{w}_{[p]}\} \tag{29}$$

are known, where $\mathbf{w}_{[i]} = \mathbf{w}(\mathbf{t}_{[i]})$ (for i = 1, ..., p). The minimal error in the space of $W_{[p]}^{\mathrm{DIIS}}$ is determined from a linear parametrization of the error vector space

$$\mathbf{w}_{[p]}^{\mathrm{opt}} = \sum_{i=1}^{p} \omega_i \mathbf{w}_{[i]} \tag{30a}$$

$$\sum_{i=1}^{p} \omega_i = 1 \tag{30b}$$

where the weights ω_i are determined from a constrained minimization using eq 30b as a constraint. The optimal trial vector $\mathbf{t}_{[p]}^{\mathrm{opt}}$ in DIIS is obtained assuming a linear relationship between trial vectors and error vectors,

$$\mathbf{t}_{[p]}^{\mathrm{opt}} = \sum_{i=1}^{p} \omega_i \mathbf{t}_{[i]} \tag{31}$$

and the trial solution $\mathbf{t}_{[p+1]}$ is constructed as the sum of the optimal contributions in their respective spaces:

$$\mathbf{t}_{[p+1]} = \mathbf{t}_{[p]}^{\mathrm{opt}} + \mathbf{w}_{[p]}^{\mathrm{opt}} \tag{32}$$

$\mathbf{w}_{[p+1]} = \mathbf{w}(\mathbf{t}_{[p+1]})$ is then calculated and the basis $T_{[p]}^{\mathrm{DIIS}}$ and $W_{[p]}^{\mathrm{DIIS}}$ are updated with $\mathbf{t}_{[p+1]}$ and $\mathbf{w}_{[p+1]}$, respectively, giving

$$T_{[p+1]}^{\mathrm{DIIS}} = \{\mathbf{t}_{[1]}, ..., \mathbf{t}_{[p+1]}\} \tag{33a}$$

$$W_{[p+1]}^{\mathrm{DIIS}} = \{\mathbf{w}_{[1]}, ..., \mathbf{w}_{[p+1]}\} \tag{33b}$$

and the DIIS sequence of subspace iterations is thus established.

Let us now consider a sequence of iterations where the optimal trial vectors $\mathbf{t}_{[i]}^{\mathrm{opt}}$ of eq 31 are stored together with $\mathbf{t}_{[p+1]}$ of eq 32. We thus consider the basis of trial vectors and error vectors:

$$T_{[p+1]}^{\mathrm{aux}} = \{\mathbf{t}_{[1]}^{\mathrm{opt}}, ..., \mathbf{t}_{[p]}^{\mathrm{opt}}, \mathbf{t}_{[p+1]}\} \tag{34a}$$

$$W_{[p+1]}^{\mathrm{aux}} = \{\mathbf{w}_{[1]}^{\mathrm{opt}}, ..., \mathbf{w}_{[p]}^{\mathrm{opt}}, \mathbf{w}_{[p+1]}\} \tag{34b}$$

From eq 31 it is seen, that the basis $T_{[p+1]}^{\mathrm{aux}}$ and $T_{[p+1]}^{\mathrm{DIIS}}$ are spanning the same subspace and the same subspace solution is therefore obtained whether the basis in eq 33 or the basis in eq 34 is used, assuming a linear relationship between trial vectors and error vectors. From eqs 5 and 8, we further obtain

DOI: 10.1021/ct501114q
J. Chem. Theory Comput. 2015, 11, 1518−1524

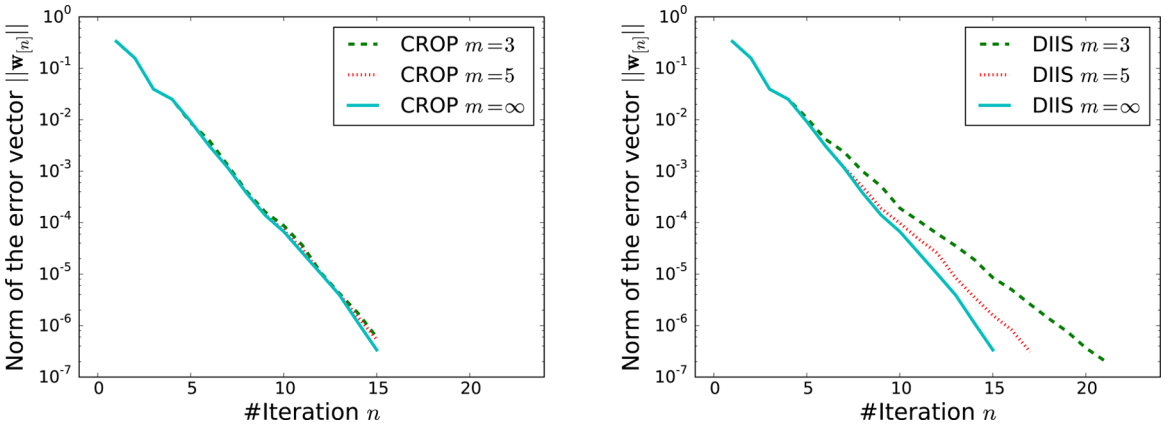


Figure 3. CROP (left) and DIIS (right) iteration sequences for a CCSD calculation on ozone using a cc-pVTZ basis in the canonical basis for the subspace truncation parameters m = 3, 5, ∞.

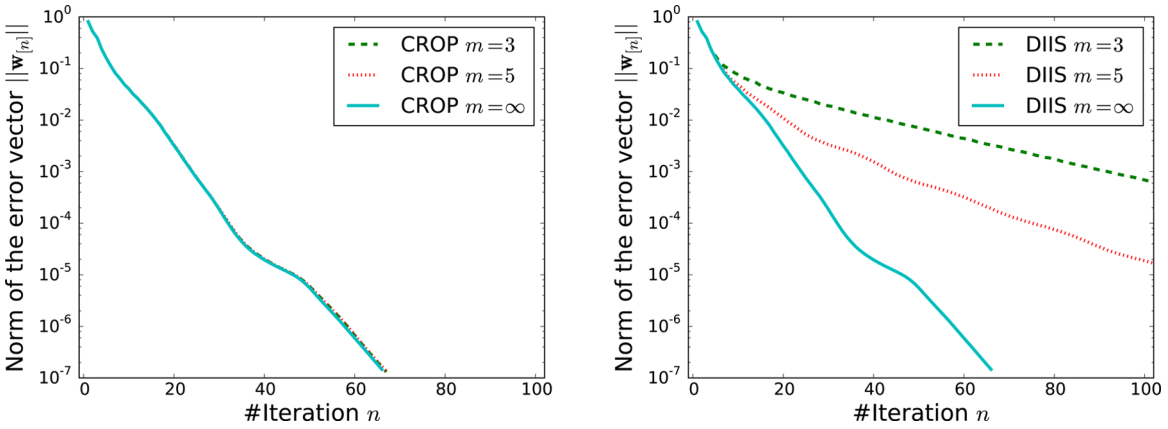


Figure 4. CROP (left) and DIIS (right) iteration sequences for a CCSD calculation on ozone using a cc-pVTZ basis in the local basis[11] for the subspace truncation parameters m = 3, 5, ∞.

$$\begin{aligned}\mathbf{w}(\mathbf{t}_{[i]}) &= \mathbf{w}^{(n)} + \mathbf{W}^{(n)}(\mathbf{t}_{[i]} - \mathbf{t}^{(n)}) + O(\left\|(\mathbf{t}_{[i]} - \mathbf{t}^{(n)})\right\|^2) \\ &= \mathbf{r}^{(n)}(\mathbf{t}_{[i]}) + O(\left\|(\mathbf{t}_{[i]} - \mathbf{t}^{(n)})\right\|^2)\end{aligned} \tag{35}$$

Neglecting terms of order $O(\|(\mathbf{t}_{[i]} - \mathbf{t}^{(n)})\|^2)$, we can thus formally replace error vectors $\mathbf{w}_{[i]}$ with residuals $\mathbf{r}^{(n)}(\mathbf{t}_{[i]}) = \mathbf{r}_{[i]}$.

New trial vectors in DIIS are obtained from eq 32 and in the CR subspace algorithm from eq 17. For both the DIIS and the CR subspace algorithm, an improved trial vector is thus generated by adding the residual for the optimal trial vector to the optimal trial vector. Furthermore, the optimal trial vectors are obtained by minimizing residual norms in both the DIIS and the CR subspace algorithm. If the trial vectors of the DIIS algorithm are stored as in eq 34, the DIIS and the CR subspace algorithm become, therefore, identical. When trial vectors in DIIS are stored in accordance with eq 34, we may neglect trial vectors from all but the last two iterations and thus determine the optimal expansion coefficients using eqs 25, 26, and 27. Of course, the above discussion is valid under the assumption of a weak nonlinearity in the CC equation, but, nevertheless, this is the fundamental assumption that we have made in DIIS by using a linear relationship between trial vectors and error vectors. The basis of eq 34 was originally introduced by Ziolkowski et al.[4] and the algorithm that uses these vectors was denoted the conjugate residual with the optimal trial vectors (CROP) algorithm. The above derivation clearly shows that a standard DIIS implementation may be changed to a CROP implementation by altering a few lines of code (see the Appendix).

NUMERICAL RESULTS

In this section, we compare the convergence of the DIIS and CROP algorithms numerically. To illustrate the convergence properties, we report CCSD (coupled cluster singles and doubles) calculations on anthracene/ozone using Dunning's cc-pVDZ/cc-pVTZ[10] basis set and solving the CCSD equations in the canonical HF basis and in a local basis[11] obtained by minimizing the second power of the second moment localization function, for both the occupied and virtual orbital space.

For the anthracene and ozone molecules, the norms of the error vectors are plotted in Figures 1−4. Figure 1 shows the norms of the error vectors on anthracene, both for the CROP and DIIS algorithms, against the number of iterations in the canonical basis, while Figure 2 shows a similar plot against the number of iterations in the local MO basis.[11] Similarly, Figure 3 shows the norms of the error vectors on ozone, both for the CROP and DIIS algorithms, against the number of iterations in the canonical basis, and Figure 4 shows a similar plot against the number of iterations in the local MO basis.[11] In this paper, we report calculations for subspaces containing the last m = 3, 5, ∞ trial vectors, where in the limit m = ∞, DIIS and CROP

DOI: 10.1021/ct501114q
J. Chem. Theory Comput. 2015, 11, 1518−1524

Chart A1. Comparison of the DIIS and CROP Algorithms

Algorithm 1: The DIIS algorithm	Algorithm 2: The CROP algorithm
1: Diagonalize Fock and transform to the diagonal basis	1: Diagonalize Fock and transform to the diagonal basis
2: get starting guess $\mathbf{t}_{[1]}$ and store $T^{\mathrm{DIIS}}_{[1]}$	2: get starting guess $\mathbf{t}_{[1]}$ and store $T^{\mathrm{aux}}_{[1]}$
3: $f = 1$	3: $f = 1$
4: for n=1,maxiter	4: for n=1,maxiter
5: if(n>m):	5: if(n>m):
6: truncate $W^{\mathrm{DIIS}}_{[n-1]}$ and $T^{\mathrm{DIIS}}_{[n]}$	6: truncate $W^{\mathrm{opt}}_{[n-1]}$ and $T^{\mathrm{aux}}_{[n]}$
7: $f = n - m + 1$	7: $f = n - m + 1$
8: get $\mathbf{w}_{[n]} = \mathbf{w}(\mathbf{t}_{[n]})$	8: get $\mathbf{w}_{[n]} = \mathbf{w}(\mathbf{t}_{[n]})$
9: if ($\|\mathbf{w}_{[n]}\| <$ thr)exit	9: if ($\|\mathbf{w}_{[n]}\| <$ thr)exit
10: add $\mathbf{w}_{[n]}$ to $W^{\mathrm{DIIS}}_{[n-1]} \rightarrow W^{\mathrm{DIIS}}_{[n]}$	10: add $\mathbf{w}_{[n]}$ to $W^{\mathrm{opt}}_{[n-1]} \rightarrow W^{\mathrm{aux}}_{[n]}$
11: solve for ω_i	11: solve for ω_i
12: $\mathbf{w}^{\mathrm{opt}}_{[n]} = \sum_{i=f}^{n} \omega_i \mathbf{w}_{[i]}$	12: $\mathbf{w}^{\mathrm{opt}}_{[n]} = \sum_{i=f}^{n-1} \omega_i \mathbf{w}^{\mathrm{opt}}_{[i]} + \omega_n \mathbf{w}_{[n]}$
13: $\mathbf{t}^{\mathrm{opt}}_{[n]} = \sum_{i=f}^{n} \omega_i \mathbf{t}_{[i]}$	13: $\mathbf{t}^{\mathrm{opt}}_{[n]} = \sum_{i=f}^{n-1} \omega_i \mathbf{t}^{\mathrm{opt}}_{[i]} + \omega_n \mathbf{t}_{[n]}$
	14: replace $\mathbf{w}_{[n]}$ by $\mathbf{w}^{\mathrm{opt}}_{[n]}$ in $W^{\mathrm{aux}}_{[n]} \rightarrow W^{\mathrm{opt}}_{[n]}$
	15: replace $\mathbf{t}_{[n]}$ by $\mathbf{t}^{\mathrm{opt}}_{[n]}$ in $T^{\mathrm{aux}}_{[n]} \rightarrow T^{\mathrm{opt}}_{[n]}$
14: get $\mathbf{t}_{[n+1]} = \mathbf{t}^{\mathrm{opt}}_{[n]} + \mathbf{w}^{\mathrm{opt}}_{[n]}$	16: get $\mathbf{t}_{[n+1]} = \mathbf{t}^{\mathrm{opt}}_{[n]} + \mathbf{w}^{\mathrm{opt}}_{[n]}$
15: add $\mathbf{t}_{[n+1]}$ to $T^{\mathrm{DIIS}}_{[n]} \rightarrow T^{\mathrm{DIIS}}_{[n+1]}$	17: add $\mathbf{t}_{[n+1]}$ to $T^{\mathrm{opt}}_{[n]} \rightarrow T^{\mathrm{aux}}_{[n+1]}$
16: end iter	18: end iter
17: transform integrals and amplitudes back to the original basis	19: transform integrals and amplitudes back to the original basis

give the same sequence of iterations. The iteration sequences for the subspace sizes m = 3, 5, ∞ are basically superimposed using the CROP algorithm, because of the weak nonlinearity of the CCSD vector function, which is independent of the basis. For DIIS, the choice of subspace does have considerable influence on the convergence, where the effect of discarding subspace information is more severe in the local basis than in the canonical basis. Because of the weaker diagonal dominance of the CCSD Jacobian in the local basis than in the canonical basis, more iterations are necessary to solve the CCSD amplitude equations in the local basis.

In contrast to anthracene, the ground state of ozone has multiconfigurational character, and, therefore, the nonlinear parts in the CCSD amplitude equations become more prominent. Still, the CROP algorithm is capable of handling this situation and the iteration sequences for CROP and subspace truncation parameters m = 3, 5, ∞ are almost indistinguishable. In particular, for ozone using the local basis, the DIIS algorithm suffers from the subspace truncation. For the subspace truncation parameter m = 3, 445 iterations are required, while for m = 5, convergence is obtained in 182 iterations. The calculations reported in Figures 1−4 clearly demonstrate the efficiency of carrying out subspace truncations using the CROP algorithm.

CONCLUSION

In this article, we have presented a simplified route for deriving the conjugate residual with optimal trial vectors (CROP) algorithm and demonstrated that the direct inversion of the iterative subspace (DIIS) sequence of iterations may be changed to a CROP sequence of iterations with only a few changes in the code of an existing DIIS implementation. When all trial vectors of the previous iterations are kept, the DIIS and CROP algorithms give the same convergence series. However, when subspace truncations are carried out, the DIIS algorithm gives a degraded convergence, whereas, for the CROP sequence, only information from the last three iterations are required to maintain the convergence of the full iterative subspace. The above conclusions were obtained by solving the nonlinear CCSD amplitude equations, but, of course, are also valid for solving CCSDT (CCSDTQ, ...) amplitude equations. For these equations, it becomes even more important to use CROP to reduce the number of trial vectors that must be stored, because of the length of the vectors. Considering that a DIIS sequence may be changed to a CROP sequence by changing only a few lines of code, we recommend that the CROP algorithm be used to solve CC amplitude equations.

DOI: 10.1021/ct501114q
J. Chem. Theory Comput. 2015, 11, 1518−1524

APPENDIX

Implementation of DIIS and CROP

The purpose of this appendix is (i) to highlight the differences and similarities in a CROP and DIIS implementation and (ii) to provide a canned CROP algorithm for the implementation when no DIIS algorithm is available from which to start.

In the previous discussion, common features of the CR subspace and DIIS algorithms were identified and it was shown that merely the subspaces $T_{[n]}^{\mathrm{DIIS}}$ and $W_{[n]}^{\mathrm{DIIS}}$ of eq 33 must be replaced by $T_{[n]}^{\mathrm{aux}}$ and $W_{[n]}^{\mathrm{aux}}$ of eq 34 to move from a DIIS implementation to a CR subspace implementation. In the CR section, new trial vectors are obtained from eq 17, while in the DIIS section, they are obtained from the corresponding eq 32. To align the notation of eqs 17 and 32 in the algorithms, we use the notation of eq 32, i.e., $\tilde{\mathbf{t}}_{[p+1]} = \mathbf{t}_{[p]} + \mathbf{r}_{[p]}$ becomes $\mathbf{t}_{[p+1]} = \mathbf{t}_{[p]}^{\mathrm{opt}} + \mathbf{w}_{[p]}^{\mathrm{opt}}$ ($\mathbf{r}_{[p]}$ is replaced by $\mathbf{w}_{[p]}^{\mathrm{opt}}$ according to eq 35). The notation of the equations referenced in the CR section must be changed accordingly. Furthermore, it is convenient to introduce a third type of basis, consisting only of the optimal trial/error vectors:

$$T_{[p]}^{\mathrm{opt}} = \{\mathbf{t}_{[1]}^{\mathrm{opt}}, ..., \mathbf{t}_{[p]}^{\mathrm{opt}}\} \tag{A-1}$$

$$W_{[p]}^{\mathrm{opt}} = \{\mathbf{w}_{[1]}^{\mathrm{opt}}, ..., \mathbf{w}_{[p]}^{\mathrm{opt}}\} \tag{A-2}$$

The DIIS (Algorithm 1) and the CROP (Algorithm 2) algorithms are shown side by side in Chart A1 to highlight the code changes necessary to move from an existing DIIS implementation to a CROP implementation. In both algorithms, code lines 1−3 are identical initial steps, where we have chosen to solve the (CCSD) equations in a (pseudo-)canonical basis. Note that $\mathbf{t}_{[1]}$ in $T_{[1]}^{\mathrm{DIIS}}$ is identical to $\mathbf{t}_{[1]} = \mathbf{t}_{[1]}^{\mathrm{opt}}$ in $T_{[1]}^{\mathrm{aux}}$. The first steps in the iterative loop are identical:

(i) perform space truncations, if necessary, in lines 5−7,
(ii) obtain the error vector in line 8,
(iii) check for convergence in line 9,
(iv) add the error vector to the iterative subspace in line 10, and
(v) solve for an optimal error vector in accordance with eq 30/23 for DIIS/CROP in line 11.

The optimal error/trial vectors in DIIS are constructed according to eqs 30/31 and for CROP according to eqs 23/19 with 24 in code lines 12/13. In the CROP algorithm, the error/trial vector $\mathbf{w}_{[n]}/\mathbf{t}_{[n]}$ in $W_{[n]}^{\mathrm{aux}}/T_{[n]}^{\mathrm{aux}}$ must be replaced by $\mathbf{w}_{[n]}^{\mathrm{opt}}/\mathbf{t}_{[t]}^{\mathrm{opt}}$ to arrive at $W_{[n]}^{\mathrm{opt}}/T_{[n]}^{\mathrm{opt}}$ in lines 14/15 of Algorithm 2. All the following steps in the algorithm are identical in DIIS and CROP, i.e., the construction of the new trial vector according to eq 32, and adding the trial vector to the set of previous trial vectors.

ASSOCIATED CONTENT

Supporting Information

This material is available free of charge via the Internet at http://pubs.acs.org.

AUTHOR INFORMATION

Corresponding Author

*E-mail: pett@chem.au.dk.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We would like to acknowledge support from The European Research Council under the European Union's Seventh Framework Programme (FP/2007−2013)/ERC Grant Agreement No. 291371, and support from The Danish Council for Independent Research−Natural Sciences. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy, under Contract No. DE-AC05-00OR22725, provided through an INCITE allocation.

REFERENCES

(1) Pulay, P. *Chem. Phys. Lett.* **1980**, *73*, 393−398.
(2) Ahlrichs, R.; Scharf, P.; Ehrhardt, C. *J. Chem. Phys.* **1985**, *82*, 890−898.
(3) Scuseria, G. E.; Lee, T. J.; Schaefer, H. F., III. *Chem. Phys. Lett.* **1986**, *130*, 236−239.
(4) Ziolkowski, M.; Weijo, V.; Jorgensen, P.; Olsen, J. *J. Chem. Phys.* **2008**, *128*, 204105.
(5) Christiansen, O., Manninen, P., Jørgensen, P., Olsen, J. *J. Chem. Phys.* **2006***124*.
(6) Wormer, P.; Visser, F.; Paldus, J. *J. Comput. Phys.* **1982**, *48*, 23−44.
(7) van Lenthe, J. H.; Pulay, P. *J. Comput. Chem.* **1990**, *11*, 1164−1168.
(8) Helgaker, T., Jørgensen, P., Olsen, J. The projected coupled-cluster equations. In *Molecular Electronic Structure Theory*; John Wiley & Sons, Ltd.: Chichester, West Sussex, England, 2000; Chapter 13.2.3, p 657.
(9) Saad, Y. The Conjugate Gradient Algorithm. In *Iterative Methods for Sparse Linear Systems*; 2nd Edition;Society for Industrial and Applied Mathematics: Philadelphia, PA, 2003; Chapter 6.7, pp 187−194.
(10) Dunning, T. H. *J. Chem. Phys.* **1989**, *90*, 1007−1023.
(11) Jansík, B., Høst, S., Kristensen, K., Jørgensen, P. *J. Chem. Phys.* **2011**, *134*.

DOI: 10.1021/ct501114q
J. Chem. Theory Comput. 2015, 11, 1518−1524