

Copyright 2014 American Institute of Physics. This article may be downloaded for personal use only. Any other use requires prior permission of the author and the American Institute of Physics".

The following article appeared in J. Chem. Phys. 141, 061101 (2014) and may be found at <http://dx.doi.org/10.1063/1.4892792>.

Speeding up local correlation methods

Daniel Kats

Citation: *The Journal of Chemical Physics* **141**, 244101 (2014); doi: 10.1063/1.4903983

View online: <http://dx.doi.org/10.1063/1.4903983>

View Table of Contents: <http://scitation.aip.org/content/aip/journal/jcp/141/24?ver=pdfcov>

Published by the [AIP Publishing](#)

Articles you may be interested in

[Accurate all-electron correlation energies for the closed-shell atoms from Ar to Rn and their relationship to the corresponding MP2 correlation energies](#)

J. Chem. Phys. **134**, 044102 (2011); 10.1063/1.3547262

[Scalar relativistic explicitly correlated R12 methods](#)

J. Chem. Phys. **132**, 214104 (2010); 10.1063/1.3417984

[Making the random phase approximation to electronic correlation accurate](#)

J. Chem. Phys. **131**, 154115 (2009); 10.1063/1.3250347

[Explicitly correlated second-order Møller–Plesset methods with auxiliary basis sets](#)

J. Chem. Phys. **116**, 6397 (2002); 10.1063/1.1461814

[Analysis of the linear response function along the adiabatic connection from the Kohn–Sham to the correlated system](#)

J. Chem. Phys. **115**, 6827 (2001); 10.1063/1.1405011



AIP | APL Photonics

APL Photonics is pleased to announce
Benjamin Eggleton as its Editor-in-Chief



Speeding up local correlation methods

Daniel Kats

Institut für Theoretische Chemie, Universität Stuttgart, Pfaffenwaldring 55, D-70569 Stuttgart, Germany

(Received 24 October 2014; accepted 1 December 2014; published online 22 December 2014)

We present two techniques that can substantially speed up the local correlation methods. The first one allows one to avoid the expensive transformation of the electron-repulsion integrals from atomic orbitals to virtual space. The second one introduces an algorithm for the residual equations in the local perturbative treatment that, in contrast to the standard scheme, does not require holding the amplitudes or residuals in memory. It is shown that even an interpreter-based implementation of the proposed algorithm in the context of local MP2 method is faster and requires less memory than the highly optimized variants of conventional algorithms. © 2014 AIP Publishing LLC. [<http://dx.doi.org/10.1063/1.4903983>]

I. INTRODUCTION

Linearly scaling local correlation methods have gained much attention in the recent years. The developments in computer architectures allow one to overcome large prefactors in the calculations and treat very large molecules at a high quantum chemical level.

There are several versions of the local correlation approach. The main difference between these methods is the treatment of the virtual space. One of the most successful type of the local treatment is based on projected atomic orbitals (PAOs).¹⁻⁴ It has been extensively used for ground- and excited state methods for molecules and periodic systems.⁵⁻¹⁶ Other common choices for the orbitals to span virtual space are the pair-natural orbitals (PNOs)¹⁷⁻²¹ or orbital-specific virtuals (OSVs).^{22,23} There are also approaches employing pure atomic orbitals (AOs).²⁴⁻²⁶ Recently, we presented a local integrated-tensor framework (LITF)²⁷ that allows rapid implementations of local methods and exploration of novel algorithms.

The perturbation theory based on the Møller-Plesset partitioning forms an essential platform for the quantum chemical hierarchy of methods. Among the perturbative techniques, popular in quantum chemical applications, one can mention the second-order perturbation theory (MP2) or perturbative triples correction to coupled cluster with singles and doubles (CCSD(T))²⁸ for the ground state, or perturbative coupled-cluster response methods CCn^{29,30} or UCC[n]^{31,32} for excited states.

Within the perturbative treatment, the diagonality of the Fock matrix in the canonical representation allows for the trivial inversion of amplitude equations, but in the local representation, this is no longer the case. This difficulty can be in principle circumvented by means of the Laplace transform technique.^{24,26,33,34} However, conventionally the local amplitude equations are solved iteratively. An excessive I/O of the amplitudes due to the inter-pair coupling of the residual equations is avoided by keeping all the amplitudes in memory,^{6,8,14} which for very large systems can constitute a memory bottleneck.

With the amplitude equations scaling linearly evaluation of the integral transformations can become a computational

bottleneck of the whole calculation (or at least form a significant share of the total computational time). A substantial improvement^{7,35,36} in this respect has been achieved by the density-fitting (DF) technique³⁷⁻³⁹ for calculation of the two-electron integrals. Nevertheless, even with DF, the problem of the high computational cost of the integral transformation remains, now for the three-index two-electron integrals, especially in case of the two-external ones.

In this contribution, we address both mentioned bottlenecks of local correlation methods and present two techniques that considerably improve the computational performance. In the first subsection of the theory section, we introduce a semi-direct integral transformation, which eliminates the need for explicit integral transformations to the virtual space. Additionally, we introduce a new *a priori* prescreening criterion, which can also be used in other parts of local correlation calculations, e.g., local density fitting. In the second subsection, we present a general algorithm for local perturbative methods that does not require to hold amplitudes or residuals in memory and is considerably faster than the conventional algorithms (cf. Sec. III).

II. THEORY

A. Semi-direct integral transformations

Since the full AO space in single-reference methods consists of occupied and virtual spaces only, the transformation from AO to the virtual subspace can be achieved through projection^{36,40} rather than real orbital transformation. This can be done also for truncated virtual spaces, i.e., we project the occupied space out of the AO space before the truncation of the latter. Within the DF formalism, for the three-index integrals, one then obtains

$$\begin{aligned} (ia|P)|_{a \in [i^*]} &= \{(i\mu|P) - S_{\mu\rho}C_{\rho k}C_{vk}(iv|P)\}_{\mu=a; a \in [i^*]} \\ &= \{(i\mu|P) - \bar{C}_{\mu k}(ik|P)\}_{\mu=a; a \in [i^*]} \end{aligned}$$

with

$$\bar{C}_{\mu k} = S_{\mu\rho}C_{\rho k}, \quad (1)$$

which guarantees that the virtual space, truncated or not, will be orthogonal to the occupied space. Here and in the following, $i, j, k, \dots, a, b, c, \dots, \mu, \nu, \rho, \dots$, and P, Q, \dots indices denote localized occupied orbitals, projected virtual orbitals, AOs, and DF-functions, respectively. Furthermore, we assume summation over repeated indices and utilize notation for local restrictions from Ref. 27, i.e., $a \in [i^*]$ means that the virtual index a is restricted to the united domain of i . $S_{\mu\nu}$ is the AO-overlap matrix, $C_{\mu i}$ are the occupied orbital coefficients, and we use chemical notation for the electron-repulsion integrals.

This type of projection results in the PAO space initially suggested by Pulay.¹ However, taking the PAO coefficients apart as in Eq. (1) results in a new very favorable form of integral transformations. Indeed, in this formulation, there is no longer need to carry out the expensive AO to PAO transformation of the integrals. Note that the resulting integrals, even in a truncated space, are exactly equal to the explicitly transformed ones, since the projection term is applied before the truncation. The truncation itself can be done according to, e.g., Boughton-Pulay domains,⁴¹ and the redundancies in the virtual basis are eliminated as usual during the update procedure.⁵

The $(i\nu|P)$ integrals in the second term on the right hand side can be efficiently prescreened using Hartree-Fock density matrix multiplied with the AO-overlap, i.e., only ν are needed for which $S_{\mu\rho}D_{\rho\nu}^{\text{HF}}|_{\mu \in [i^*]}$ are not zero.

A further way to reduce the cost of the transformation is to prescreen the $(ik|P)$ integrals. Since the occupied orbitals are mutually orthogonal, simple overlap-based criteria are not possible here. The pseudo-overlaps^{9,42,43} can in principle be used, but in case of occupied-orbital products, they can be prone to artificial zeroing, when the product of the two orbitals is not totally symmetric. Therefore, we employ a different criterion, a product of the orbitals' Löwdin partial charges, which is related to an overlap of absolute values of orbitals,

$$O_{ik} = \sum_A \sum_{\nu \in A} \left(\sum_{\mu} C_{\mu i} S_{\mu\nu}^{\frac{1}{2}} \right)^2 \sum_{\nu' \in A} \left(\sum_{\rho} S_{\nu'\rho}^{\frac{1}{2}} C_{\rho k} \right)^2, \quad (2)$$

where A denotes an atom. This prescreening can be applied in an *a priori* manner defining domains of k indices for a given i index,⁴³ and thus allows linear-scaling implementations of integral transformations.

The new type of the integral factorization is especially useful for the two-external integrals, needed in local coupled-cluster methods. Furthermore, in the case of local CC2 method,^{12,44} it facilitates a semidirect approach, which drastically reduces the I/O rate and is very instrumental for efficient parallelization. Following this strategy, one rewrites two-external integrals as

$$(ab|P) = \{(\mu\nu|P) - \bar{C}_{\mu k}(k\nu|P) - \bar{C}_{\nu k}(\mu k|P) + \bar{C}_{\mu k}\bar{C}_{\nu l}(kl|P)\}_{\mu=a, \nu=b}. \quad (3)$$

This allows one to completely avoid the expensive AO to PAO transformation of the $(\mu\nu|P)$ and $(a\nu|P)$ integrals and the storage of the $(ab|P)$ integrals in the case of density-fitted local CC2.

Finally, we note that the projector in Eq. (1) can also utilize non-orthogonal occupied orbitals^{45,46} and thus benefit from their additional locality. In order to further speed up the transformations and achieve linear scaling, one can combine this semi-direct transformation with local density-fitting techniques.^{7,42,43}

B. Fast and memory-friendly iterative algorithm for the amplitude equations in local perturbative methods

In general, the orbital-invariant amplitude equations for single-reference perturbative methods based on the Møller-Plesset partitioning of the Hamiltonian in orthogonal occupied space can be written as

$$0 = R_{ab\dots}^{ij\dots} = K_{ab\dots}^{ij\dots} + \mathcal{P}(ia; jb; \dots) \left\{ T_{a'b'\dots}^{ij\dots} f_{aa'} S_{b'b} \dots - f_{i'i} T_{a'b'\dots}^{i'j\dots} S_{a'a} S_{b'b} \dots \right\} \quad (4)$$

with

$$\mathcal{P}(ia; jb; \dots) X_{ab\dots}^{ij\dots} = X_{ab\dots}^{ij\dots} + X_{ba\dots}^{ji\dots} + \dots, \quad (5)$$

and $K_{ab\dots}^{ij\dots}$, f , S_{ab} , and $T_{ab\dots}^{ij\dots}$ are the permutationally symmetrical right-hand-side, Fock matrices, PAO-overlap matrix, and amplitudes, respectively. From the form of Eq. (4), it is apparent that the amplitudes are coupled to the residual only by one-electron two-index quantities. It is possible to derive an algorithm for solving these equations with one occupied-space index free, while all other occupied indices are fixed in a loop (in the conventional algorithm all occupied indices of the residual are fixed, and amplitudes are held in the primary memory⁶).

In Figure 1, we show such an algorithm for the local MP2 equations. We again use the notation for local restrictions from Ref. 27: $a \in [ij]$ denotes the restriction of index a to the pair domain ij , and $ij \in \{\mathbf{ij}\}$ denotes that the indices i and j are restricted according to the pair list. First, the right-hand-side (i.e., the 4-index integrals) is written to the residual

```

copy  $\mathbf{K}_{ab}^{ij}$  to  $\mathbf{R}_{ab}^{ij} |_{ab \in [ij]; ij \in \{\mathbf{ij}\}}$ 
for all  $j$  do
  read  $\mathbf{T}_{ab}^{kj} |_{ab \in [kj]; kj \in \{\mathbf{kj}\}}$ 
  calculate  $(\mathbf{ST})_{ab}^{kj} |_{a \in [j^*]; b \in [kj]; kj \in \{\mathbf{kj}\}}$ 
   $\mathbf{Q}_{ac}^{kj} |_{ac \in [kj]; kj \in \{\mathbf{kj}\}} = (\mathbf{ST})_{ab}^{kj} \times \mathbf{F}_{bc}$ 
  for all  $i \in \{\mathbf{ij}\}$  do
     $(\mathbf{fST})_{ab}^{ij} |_{a \in [ij]; b \in [j^*]; ij \in \{\mathbf{ij}\}} = (\mathbf{ST})_{ab}^{kj} \times \mathbf{f}_{ki}$ 
     $\mathbf{Q}_{ac}^{ij} = \mathbf{Q}_{ac}^{ij} - (\mathbf{fST})_{ab}^{ij} \times \mathbf{S}_{bc}$ 
    if  $i \leq j$  then
       $\mathbf{R}_{ac}^{ij} = \mathbf{R}_{ac}^{ij} + \mathbf{Q}_{ac}^{ij}$ 
    end if
    if  $i \geq j$  then
       $\mathbf{R}_{ca}^{ji} = \mathbf{R}_{ca}^{ji} + \mathbf{Q}_{ac}^{ij}$ 
    end if
  end for
end for
end for

```

FIG. 1. Pseudocode for evaluation of the LMP2 residual.

file. The residual is then accumulated and the amplitudes iteratively updated within a loop over *one* occupied index j . At each cycle, the current LMP2 amplitudes for a given j (with resolved index-permutational symmetry) are read and contracted with the overlap matrix. The resulting intermediate has one virtual index in the united domain of j and the other in the corresponding pair domain. In the next step, the contribution from the external Fock-matrix contraction is calculated and added to the \mathbf{Q}_{ac}^{kj} intermediate. The latter is effectively a three-index quantity with the same structure as the residual. Next, one starts a loop over another occupied index i and calculates the internal Fock-matrix contribution. Finally, one symmetrises the \mathbf{Q}_{ac}^{kj} quantity and adds it to the residual.

The largest intermediate within this algorithm is $(\mathbf{ST})_{ab}^{kj}$, which has one virtual index in the united domain (\mathbf{a}), one virtual index in pair domain of kj (\mathbf{b}), and an occupied index restricted according to a pair list (\mathbf{k}). All other intermediates are much smaller in size. The overall primary memory requirement is thus around $N_v^{u.d.} \cdot N_v^{p.d.} \cdot N_o^{p.l.}$ (with $N_v^{u.d.}$ we denote the maximal number of virtuals in united domains, $N_v^{p.d.}$ —averaged number of virtuals in pair domains, and $N_o^{p.l.}$ —maximal number of occupied orbitals making a pair with a given orbital according to the pair list). For very large systems, the size of Fock and overlap matrices could become an issue, but in the local virtual basis they are essentially sparse. Therefore, the memory demands of this algorithm are asymptotically either independent from the molecular size or, for extremely large systems, scale with the latter linearly.

Another important advantage of this algorithm is that in all multiplications with the overlap matrices, one index is in the united domain. It is especially beneficial to employ it with large PAO domains, where it considerably reduces redundancies in the calculations (cf. Sec. III). Additionally, one benefits from larger matrix multiplications, compared to the conventional case, i.e., Algorithm 1 from Ref. 6. We note that there are versions of LMP2 algorithms, e.g., Algorithm 2 from Ref. 6, in which one builds an intermediate with two virtual indices in united domains.^{6,14} However, this intermediate usually contains zero-blocks, since pair-domains united over one orbital generally do not yield all blocks of a squared united domain and, therefore, these algorithms will be generally slower than the suggested one (and require more memory than even the Algorithm 1 from Ref. 6).

Iterative local perturbative triples correction⁸ can be implemented in the similar way. There the main loop should run over a pair of indices from the triple list. After the multiplication, symmetrization of the third index is required (as in the LMP2-algorithm).

One can go even further and derive an analogous algorithm for non-perturbative methods. For example, in the case of CCSD with full triples (CCSDT), the triples amplitudes are coupled to the residual through a two-electron quantity. It means that it would be possible to calculate the residual without holding the triples amplitudes in memory, i.e., by looping over one occupied index and restoring the index-permutational symmetry for the two remaining indices afterwards.

III. IMPLEMENTATION AND RESULTS

The semi-direct generation of one-external three-index integrals, which are required in DF-LMP2 calculations, has been implemented in MOLPRO^{47,48} in the integral-transformation part of DF-LMP2 code⁷ (without local density fitting). Note that the projection has to be done using the full occupied space, i.e., including core orbitals. The procedure is outlined below.

1. First transformation: prescreening with maximal PAO coefficients as was done in the original PAO-transformation code of MOLPRO.
2. Second transformation:
 - (a) copy $(i\mu|P)$ to $(ia|P)|_{a \in [i^*]}$;
 - (b) calculate $(ik|P) = C_{\nu k}(i\nu|P)$ with prescreening according to the O_{ik} -criterion, Eq. (2);
 - (c) calculate $-\bar{C}_{\mu k}(ik|P)$ and accumulate on $(ia|P)|_{a \in [i^*]}$.

The algorithm presented in Sec. II B has been implemented using LITF.²⁷

As a test system, we chose a 12-unit glycine chain ($[\text{gly}]_{12}$) in aug-cc-pVTZ basis (68 correlated electrons and 3128 AOs) with aug-cc-pVTZ/MP2FIT as the fitting basis set. Three types of calculations have been performed: with pure Boughton-Pulay (BP) domains⁴¹ (threshold 0.985), and with BP domains extended to the neighbouring (the `ixt=1` option in MOLPRO) and the second neighbouring (`ixt=2`) atoms. These test cases represent common applications of local methods to large organic molecules. Additionally, we have performed a `ixt=1` calculation on a $[\text{gly}]_{18}$ system (101 correlated electrons and 4646 AOs) and on a 3D-like dye dyad solvent cluster from Ref. 34 (123 correlated electrons and 3962 AOs) denoted as DA, the latter one in cc-pVTZ basis. Pairs with inter-orbital distances larger than 15 bohrs were neglected. In the case of aug-cc-pVTZ calculations, the contributions of the most diffuse functions of each angular momentum have been discarded in the Pipek-Mezey orbital-localization procedure⁴⁹ (the `cpdel=1` option). In Table I, we show dependence of the averaged number of occupied orbitals k and accuracy of the calculation from the choice of the threshold for the O_{ik} -criterion. In the case of the glycine chain, the relative error of the correlation energy follows closely the value of the threshold. But in the DA case, the error is much larger, which seems to indicate that in three-dimensional systems, one should choose tighter thresholds. In the following calculations, we set the threshold to 10^{-9} (for which the error in DA

TABLE I. Average number of k indices per i index in the prescreened integrals $(ik|P)$ and relative correlation energy error $\Delta E^{\text{corr}} = (E_{\text{thr}}^{\text{corr}} - E_0^{\text{corr}}) / E_0^{\text{corr}}$ for different threshold values for the O_{ik} -criterion.

O_{ik} -thr.	10^{-7}	10^{-8}	10^{-9}	10^{-10}	0
	[gly] ₁₂ , ixt=1				
\bar{N}_o	110.5	139.2	163.4	175.8	185.0
ΔE^{corr}	10^{-7}	10^{-8}	10^{-9}	10^{-10}	0
	DA, ixt=1				
\bar{N}_o	150.7	195.0	238.5	272.8	346.0
ΔE^{corr}	10^{-2}	10^{-4}	10^{-6}	10^{-7}	0

correlation energy is around $10^{-4}\%$), although tighter values did not make the calculations noticeably slower, since main savings come from the projection technique rather than pre-screening (at least for this size of molecules). Timings of relevant steps and the overall minimal memory requirements are compiled in Table II, which also contains the corresponding numbers for the Algorithm 1 and the Algorithm 2 from Ref. 6 and a slightly modified version of the Algorithm 2 (algo3), where an intermediate is constructed for a union of united domains of i and j (the options `lmp2algo=1, 0, and 3` in MOLPRO, respectively). Additionally, timings for Algorithm 2 implemented in LITF are provided in order to estimate the overhead of a LITF implementation.

The presented algorithm is noticeably faster than the highly-optimized conventional algorithms. The time for one iteration of the new algorithm is shorter than of all other algorithms in all tested cases, apart from the `[gly]12, iext=2` calculation, where the native implementation of `algo3` turned out to be slightly faster. However, a comparison of a native implementation of `algo2` and a LITF implementation suggests that a native implementation of the new algorithm would outperform the `algo3` also in this case. The current implementation usually requires two or three iterations more, since the MOLPRO implementation does a dynamic update of amplitudes during the iterations, but it would be possible to do it also in the new algorithm.

The whole calculation on `[gly]12` with `iext=1` can be done using less than one gigabyte of memory with batching required only in the integral transformation part, in contrast to the conventional algorithms that require nearly four and more than four gigabytes. And at the same time, the iterations (using less optimal LITF-implementation) are faster than the hand-coded

conventional algorithms, and the gain in time is even larger if one goes to larger systems or smaller domains. Note that the memory requirements for the `[gly]18` calculation are the same as for the `[gly]12` one (both with `iext=1`), i.e., the memory demand indeed does not increase with the system size. The new scheme requires a somewhat larger I/O communication than the conventional algorithm in MOLPRO, since the amplitudes now are stored on disk. However, the I/O rate is independent of the molecular size (in contrast to paging implementations of conventional algorithms). Furthermore, if the memory is indeed sufficient to hold the doubles amplitudes, this issue can be completely eliminated.

The DA calculation shows that even for more three-dimensional systems with larger united domains, the primary-memory requirements of the proposed algorithm are much smaller than of the conventional algorithms and the iteration time for this system is nearly halved.

IV. CONCLUSION

Two ways of speeding up local correlated calculations have been presented. The first one allows one to avoid explicit integral transformations with the virtual-orbitals coefficients. Apart from the obvious benefits for the PAO-integral transformations, it opens a way to implement semi-direct AO methods with an explicit elimination of occupied-orbitals contamination of the virtual space (as in pure AO methods). It can also significantly speed up local coupled-cluster implementations, where integral transformations often consume a considerable amount of the total computational time. Since the two-internal three-index integrals are computed anyway in the density-fitted coupled-cluster calculations, evaluation of

TABLE II. Timings for AO to PAO transformation step ($t_{\text{PAO transf.}}$), one PAO-LMP2 iteration ($t_{\text{iter.}}$) and total wall time of the calculation ($t_{\text{wall tot.}}$) (in s), as well as minimal memory requirement rounded up to the next integer (in gigabytes) and fraction of correlation energy recovered by the respective local approximation for conventional implementations (denoted as “algo1,” “algo2,” and “algo3”) and new implementation. The calculations have been performed on one Intel Core i5-3550 CPU, 3.30 GHz processor unit.

	<code>[gly]₁₂,BP</code>					<code>[gly]₁₂,iext=1</code>					<code>[gly]₁₂,iext=2</code>				
	Native			LITF		Native			LITF		Native			LITF	
	algo1	algo2	algo3	algo2	New	algo1	algo2	algo3	algo2	New	algo1	algo2	algo3	algo2	New
$t_{\text{PAO transf.}}^a$	264			57		307			70		347			71	
$t_{\text{iter.}}$	201	167	128	187	76	2279	644	480	734	470	9107	1810	1423	1989	1517
$t_{\text{wall tot.}}^a$	2941	2845	2507	3660	2401	16 251	6481	5408	10 335	7953	59 236	15 417	13 130	25 532	22 093
Memory	1 ^b	2 ^b	2 ^b	2 ^b	1(1) ^c	4 ^b	5 ^b	5 ^b	5 ^b	2(1) ^c	10 ^b	10 ^b	10 ^b	10 ^b	4(1) ^c
E^{corr} frac	0.987						0.997						0.999		
	<code>[gly]₁₈,iext=1</code>					DA,iext=1									
	Native			LITF		Native			LITF						
	algo1	algo3		New		algo3	New								
$t_{\text{PAO transf.}}^a$	787			179		1884			348						
$t_{\text{iter.}}$	3042			838		3217			1694						
$t_{\text{wall tot.}}^a$	28 672			11 951		15 214			28 347						
Memory	6 ^b			7 ^b		2(1) ^c			8 ^b						
						8(3) ^c									

^aCalculations have been performed without local density fitting. In the new PNO-based code of MOLPRO, the calculations require only a fraction of this time.²¹

^bMemory required for doubles amplitudes and intermediates.

^cIn the current implementation using LITF, loops are running in batches of occupied orbitals with a fixed size of four; therefore, in a native implementation, the memory consumption could be significantly reduced further. The numbers in parentheses show the memory requirement in the case of single-orbital loops.

integrals with external indices would be even cheaper than in the case of LMP2. This method is also relevant for some OSV and PNO implementations,^{21,50} which, in order to preserve linear scaling, start from a larger virtual space, spanned by PAOs.

The second part presents an algorithm for iterative local perturbative methods. The primary-memory requirements of this technique are independent from the molecular-system size. In the test calculations, the new LMP2 equation solver needed substantially less memory and was faster than the conventional ones. This algorithm represents a valuable alternative to Laplace-transform approach to local perturbative methods.^{26,34,44,51–53} Indeed, it preserves the features of the conventional iterative local scheme (i.e., it provides an upper-bound of the full-domain energy, it is free from possible pitfalls in the generation of Laplace-quadrature points,⁵⁴ etc.). At the same time, its memory requirements are as small as in the Laplace-transform-based LMP2.³⁴ This algorithm is also promising for efficient evaluation of the iterative local perturbative triples correction.

ACKNOWLEDGMENTS

The author is grateful to D. Usvyat and H.-J. Werner for valuable comments on the manuscript. Financial support from ERC (Advanced Grant ASES 320723) is acknowledged.

- ¹P. Pulay, *Chem. Phys. Lett.* **100**, 151 (1983).
- ²S. Saebø and P. Pulay, *Chem. Phys. Lett.* **113**, 13 (1985).
- ³S. Saebø and P. Pulay, *J. Chem. Phys.* **86**, 914 (1987).
- ⁴S. Saebø and P. Pulay, *J. Chem. Phys.* **88**, 1884 (1988).
- ⁵C. Hampel and H.-J. Werner, *J. Chem. Phys.* **104**, 6286 (1996).
- ⁶M. Schütz, G. Hetzer, and H.-J. Werner, *J. Chem. Phys.* **111**, 5691 (1999).
- ⁷H.-J. Werner, F. R. Manby, and P. J. Knowles, *J. Chem. Phys.* **118**, 8149 (2003).
- ⁸M. Schütz, *J. Chem. Phys.* **113**, 9986 (2000).
- ⁹H.-J. Werner and M. Schütz, *J. Chem. Phys.* **135**, 144116 (2011).
- ¹⁰T. D. Crawford and R. A. King, *Chem. Phys. Lett.* **366**, 611 (2002).
- ¹¹T. Korona and H.-J. Werner, *J. Chem. Phys.* **118**, 3006 (2003).
- ¹²D. Kats, T. Korona, and M. Schütz, *J. Chem. Phys.* **125**, 104106 (2006).
- ¹³D. Kats, T. Korona, and M. Schütz, *J. Chem. Phys.* **127**, 064107 (2007).
- ¹⁴L. Maschio, D. Usvyat, F. R. Manby, S. Casassa, C. Pisani, and M. Schütz, *Phys. Rev. B* **76**, 075101 (2007).
- ¹⁵M. Lorenz, D. Usvyat, and M. Schütz, *J. Chem. Phys.* **134**, 094101 (2011).
- ¹⁶T. Korona, D. Kats, M. Schütz, T. B. Adler, Y. Liu, and H.-J. Werner, *Linear-Scaling Techniques in Computational Chemistry and Physics*, Challenges and Advances in Computational Chemistry and Physics No. 13, edited by R. Zalesny, M. G. Papadopoulos, P. G. Mezey, and J. Leszczynski (Springer, Netherlands, 2011), pp. 345–407.
- ¹⁷W. Meyer, *Int. J. Quantum Chem. Symp.* **5**, 341 (1971).
- ¹⁸F. Neese, F. Wennmohs, and A. Hansen, *J. Chem. Phys.* **130**, 114108 (2009).
- ¹⁹C. Hättig, D. P. Tew, and B. Helmich, *J. Chem. Phys.* **136**, 204105 (2012).
- ²⁰C. Riplinger, B. Sandhoefer, A. Hansen, and F. Neese, *J. Chem. Phys.* **139**, 134101 (2013).
- ²¹H.-J. Werner, G. Knizia, C. Krause, M. Schwilk, and M. Dornbach, “Scalable electron correlation methods I.: PNO-LMP2 with linear scaling in the molecular size and inverse-linear scaling in the number of processors,” *J. Chem. Theory Comput.* (published online).
- ²²J. Yang, Y. Kurashige, F. R. Manby, and G. K. L. Chan, *J. Chem. Phys.* **134**, 044123 (2011).
- ²³J. Yang, G. K. L. Chan, F. R. Manby, M. Schütz, and H.-J. Werner, *J. Chem. Phys.* **136**, 144105 (2012).
- ²⁴M. Häser and J. Almlöf, *J. Chem. Phys.* **96**, 489 (1992).
- ²⁵G. E. Scuseria and P. Y. Ayala, *J. Chem. Phys.* **111**, 8330 (1999).
- ²⁶D. Lambrecht, B. Doser, and C. Ochsenfeld, *J. Chem. Phys.* **123**, 184102 (2005).
- ²⁷D. Kats and F. R. Manby, *J. Chem. Phys.* **138**, 144101 (2013).
- ²⁸K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon, *Chem. Phys. Lett.* **157**, 479 (1989).
- ²⁹O. Christiansen, P. Jørgensen, and C. Hättig, *Int. J. Quantum Chem.* **68**, 1 (1998).
- ³⁰O. Christiansen, H. Koch, and P. Jørgensen, *J. Chem. Phys.* **103**, 7429 (1995).
- ³¹D. Kats, D. Usvyat, and M. Schütz, *Phys. Rev. A* **83**, 062503 (2011).
- ³²G. Wälz, D. Kats, D. Usvyat, T. Korona, and M. Schütz, *Phys. Rev. A* **86**, 052519 (2012).
- ³³J. Almlöf, *Chem. Phys. Lett.* **181**, 319 (1991).
- ³⁴D. Kats, D. Usvyat, and M. Schütz, *Phys. Chem. Chem. Phys.* **10**, 3430 (2008).
- ³⁵M. Schütz and F. R. Manby, *Phys. Chem. Chem. Phys.* **5**, 3349 (2003).
- ³⁶M. Schütz, H.-J. Werner, R. Lindh, and F. R. Manby, *J. Chem. Phys.* **121**, 737 (2004).
- ³⁷E. J. Baerends, D. E. Ellis, and P. Ros, *Chem. Phys.* **2**, 41 (1973).
- ³⁸J. L. Whitten, *J. Chem. Phys.* **58**, 4496 (1973).
- ³⁹B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, *J. Chem. Phys.* **71**, 3396 (1979).
- ⁴⁰P. Umari, G. Stenuit, and S. Baroni, *Phys. Rev. B* **81**, 115104 (2010).
- ⁴¹J. W. Boughton and P. Pulay, *J. Comput. Chem.* **14**, 736 (1993).
- ⁴²M. Schütz, D. Usvyat, M. Lorenz, C. Pisani, L. Maschio, S. Casassa, and M. Halo, in *Accurate Condensed Phase Quantum Chemistry*, edited by F. R. Manby (CRC Press, Boca Raton, FL, 2010), p. 29.
- ⁴³D. Usvyat, *J. Chem. Phys.* **139**, 194101 (2013).
- ⁴⁴D. Kats and M. Schütz, *J. Chem. Phys.* **131**, 124117 (2009).
- ⁴⁵V. P. Smirnov and D. E. Usvyat, *Phys. Rev. B* **64**, 245108 (2001).
- ⁴⁶D. Usvyat and M. Schütz, *Theor. Chem. Acc.* **114**, 276 (2005).
- ⁴⁷H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, M. Schütz *et al.*, MOLPRO, version 2012.1, a package of *ab initio* programs, 2012, see <http://www.molpro.net>.
- ⁴⁸H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, and M. Schütz, *WIREs. Comput. Mol. Sci.* **2**, 242 (2012).
- ⁴⁹J. Pipek and P. G. Mezey, *J. Chem. Phys.* **90**, 4916 (1989).
- ⁵⁰C. Riplinger and F. Neese, *J. Chem. Phys.* **138**, 034106 (2013).
- ⁵¹D. Kats and M. Schütz, *Z. Phys. Chem.* **224**, 601 (2010).
- ⁵²K. Freundorfer, D. Kats, T. Korona, and M. Schütz, *J. Chem. Phys.* **133**, 244110 (2010).
- ⁵³K. Ledermüller, D. Kats, and M. Schütz, *J. Chem. Phys.* **139**, 084111 (2013).
- ⁵⁴D. Kats, D. Usvyat, S. Loibl, T. Merz, and M. Schütz, *J. Chem. Phys.* **130**, 127101 (2009).