Scalable electron correlation methods I.: PNO-LMP2 with linear scaling in the molecular size and near-inverse-linear scaling in the number of processors

Hans-Joachim Werner, Gerald Knizia, Christine Krause, Max Schwilk, Mark Dornbach
Institut für Theoretische Chemie, Universität Stuttgart, Pfaffenwaldring 55, 70569 Stuttgart, Germany.

We propose to construct electron correlation methods that are scalable in both molecule size and aggregated parallel computational power, in the sense that the total elapsed time of a calculation becomes nearly independent of the molecular size when the number of processors grows linearly with the molecular size. This is shown to be possible by exploiting a combination of local approximations and parallel algorithms. The concept is demonstrated with a linear scaling PNO-LMP2 (pair natural orbital local second-order Møller-Plesset perturbation theory) method. In this method, both the wave function manifold and the integrals are transformed incrementally from projected atomic orbitals (PAOs) first to orbital specific virtuals (OSVs), and finally to pair natural orbitals (PNOs), which allow for minimum domain sizes and fine-grained accuracy control using very few parameters. A parallel algorithm design is discussed, which is efficient for both small and large molecules, and numbers of processors, although true inverse-linear scaling with compute power is not yet reached in all cases. Initial applications to reactions involving large molecules reveal surprisingly large effects of dispersion energy contributions, as well as large intramolecular basis set superposition errors (BSSE) in canonical MP2 calculations. In order to account for the dispersion effects, the usual selection of PNOs on the basis of natural occupation numbers turns out to be insufficient, and a new energy based criterion is proposed. If explicitly correlated (F12) terms are included, fast convergence to the MP2 complete basis set (CBS) limit is achieved. For the studied reactions, the PNO-LMP2-F12 results deviate from the canonical MP2/CBS and MP2-F12 values by less then 1 kJ mol^{-1}, using triple-ζ (VTZ-F12) basis sets.

I. INTRODUCTION

Despite the enormous success of density functional theory, accurate wave function methods are highly desirable in computational chemistry. However, due to the steep scaling of the computational resources (CPU-time, memory, disk space) with the molecular size, conventional wave function methods such as coupled-cluster CCSD(T) (coupled-cluster with single and double excitations and a perturbative treatment of triple excitations) can be applied only to rather small systems. The CPU-time of CCSD(T) scales as $\mathscr{O}(\mathscr{N}^7)$, and even for the simplest electron correlation method, MP2 (second-order Møller-Plesset perturbation theory), it scales as $\mathscr{O}(\mathscr{N}^5)$, where $\mathscr{N}$ is a measure of the molecular size (e.g. the number of electrons). This causes a "scaling wall" that can only slightly be moved towards larger molecular sizes by parallelization, even with the largest supercomputers. Another problem of the CCSD(T) method is the slow convergence of the correlation energy with the basis set size. Very large basis sets are needed to obtain converged results, and this makes high-accuracy electronic structure calculations extremely expensive.

During the last decade enormous progress has been made to overcome these problems. The scaling problem can be alleviated by exploiting the short-range character of electron correlation using local orbitals,[1–44] and the basis set problem can be overcome by explicitly correlated (F12) methods.[45–78] Particularly interesting is the combination of both approaches, since it has been found that the F12 contributions not only reduce the basis set error, but also much improve the accuracy of local approximations.[23,71,74,75]

Nevertheless, one remaining problem is that most of the existing programs are not well suited to high-performance computing platforms. Even in methods scaling linearly in molecular size, the amount of data, in particular the number of two-electron integrals, may become enormous in calculations for large molecules. To our knowledge, in all current methods beyond MP2 these data are stored on disk and must then be read and processed in each iteration. This leads to an I/O bottleneck preventing efficient parallelization of these methods, and the most powerful available local methods are therefore only moderately parallel[18,40–43] and suitable only for a single or very few compute nodes. However, an electron correlation methods's range of applicability would be massively increased if it would scale linearly in molecule size and and inverse-linearly in aggregated parallel computational power *at the same time*. We will call such methods *scalable*. For example, due to its $\mathscr{O}(\mathscr{N}^7)$ size scaling, even if perfectly paralellized, a canonical CCSD(T) calculation can only treat 10% larger molecules if the computational resources are doubled. But with a *scalable* method, de facto *any* molecular size can be tackled by increasing computing power.

Unfortunately, achieving true scalability by simply modifying the existing codes would be difficult. We have therefore started to write a new generation of local correlation programs from scratch, and the current paper describes the first step, namely PNO-LMP2. All parts of the program are parallelized, and all large data structures are distributed over the processors and compute nodes. Furthermore, communication between the processes is minimized and I/O operations in the integral transformations and iterations are entirely avoided. The new methods are based on a combination of know-how from our previous developments of PAO- and OSV-based LCCSD methods with the ideas of Neese and coworkers[37–43] regarding the use of PNOs. The use of PNOs is essential, since this allows an enormous reduction of the data size and CPU time. This makes it possible to keep all data in memory and thus avoid I/O bottlenecks in the parallel execution. We also include explicitly correlated (F12) terms, but a scalable implementation is still under development and will be described elsewhere.

This paper is organized as follows: in section II we briefly summarize relevant previous work. In section III we present the basic theory. Subsequently, we outline how scalability can be realized and describe details of the local approximations in our present program, such as domain selections (section IV). In section V we outline the linear-scaling integral evaluation using local density fitting techniques. Our parallelization model is described in section VI. The treatment of explicitly correlated terms is summarized in VII. In section VIII we will present benchmarks for the efficiency of our method, as well as for the accuracy of correlation and reaction energies. A short summary closes the paper.

Our method involves a number of different orbital spaces, and for the sake of clarity the abbreviations and index notation used in this paper are summarized in Table I. We assume that the initial occupied molecular orbitals $|i\rangle$ are localized, while the initial virtual orbitals $|a\rangle$ are canonical. In this paper we exclusively consider the closed-shell theory.

II. BACKGROUND

The field of local correlation methods came to live via the pioneering work of Pulay in 1983.[1–5] Over the years many other local correlation approaches[26–44,79–87] have been proposed, including methods for excited states[88–90] and periodic systems,[91–94] but only few of these have been developed up to the CCSD(T) level and have reached the production stage. A review of all methods is beyond the scope of the current paper. Here we only briefly mention those methods that are particularly relevant for the current work.

PAO-based local correlation methods have been developed in our group and by Schütz et al.[6–25] up to LCCSD(T) and even higher levels,[13,42,43] and for all these methods linear scaling of the computational cost with system size has been achieved. Further progress was made by the development of local density fitting techniques[15–18] for the efficient calculation and transformation of the required two-electron integrals. More recently, the extension by explicitly correlated terms[71,74,87,95,96] [LCCSD(T)-F12] has significantly improved the accuracy of these methods, and sub-kcal accuracy has been achieved, e.g. for reaction energies and barrier heights in enzymes.[95] Unfortunately, the PAO-based local correlation methods suffer from the fact that rather large PAO domains are needed to obtain fully converged results; a consistent and balanced selection of these domains for different molecular structures turned out to be a difficult problem. The number of integrals in the PAO basis needed for LCCSD(T) calculation scales with the fourth power of the domain sizes, and despite linear scaling this can cause severe memory and I/O bottlenecks.

This problem has been partly overcome by Neese and co-workers,[37–43] who proposed to use pair natural orbitals (PNOs)[97–101] instead of PAOs to span the virtual space. The PNO domains can be chosen in a fine-grained way, based on the pair natural occupation numbers. PNOs are the optimum orbitals in the sense that the correlation energy converges very quickly with the number of PNOs per pair. Typically, only 50-70 PNOs per pair are needed to recover about 99.8% of the correlation energy (even fewer PNOs are sufficient if F12 terms are included). However, the PNOs are different for each pair, and the total number of required PNOs in large molecules may become huge (10^5 or more). For a long time it was therefore believed that this prevents their use in local methods, since the integral transformations seemed impossible. However, Neese et al. demonstrated that this problem can be overcome using local density fitting approaches, and they recently implemented impressive PNO based methods up to the level of LCCSD(T).[42,43] Following the ideas of Neese and coworkers, a number of other PNO-based local methods have also been developed in recent years.[79–82,88–90]

TABLE I: Index notation used for different orbital spaces. Unless otherwise noted, the orbitals are assumed to be orthonormal and pseudo-canonical (see text). The corresponding non-orthogonal orbitals are denoted by a tilde.

orbital space	indices
Localized occupied orbitals (LMOs)	i, j
Canonical virtual orbitals (VMOs)	a, b
Projected atomic orbitals (PAOs)	r, s
Orbital specific virtuals (OSVs)	r^i, s^i
OSV pair domains	r^{ij}, s^{ij}
Pair natural orbitals (PNOs)	$p^{ij}, q^{ij}, u^{ij}, v^{ij}$

A related approach is to use orbital-specific virtuals (OSVs),[102–105] where one set of virtual orbitals is used for each LMO. The OSVs are identical to the PNOs for the diagonal pairs (i.e. those which describe the correlation of two electrons in the same spatial orbital). As compared to the PNO approach this strongly reduces the total number of virtual orbitals, but still keeps the advantage of a fine-grained domain selection. OSV-based local methods have also been implemented up to the level of LCCSD(T).[103,105] It turned out, however, that the OSV pair domains (which are the union of the OSV domains for two orbitals) must be 2-4 times larger than the PNO domains, and in the end the efficiency of both methods appears to be quite comparable.

In the previously described methods, local approximations are introduced, but the full system wave function is still explicitly constructed. A conceptionally different class of methods treats the correlation problem in overlapping fragments of the molecule independently (fragmentation,[106–116] and incremental methods[117–121]). The various approaches differ in the way in which the fragments are defined and the total correlation energy is assembled. They have in common, however, that for each fragment the corresponding subset of orbitals is canonicalized, so that standard methods such as CCSD(T) can be used. The advantage of such methods is their simplicity. Since each fragment is treated independently of all others, “embarrassingly” parallel implementations are possible, and the methods are in principle scalable, i.e. the size of the molecules that can be treated increases linearly with the available number of processors. However, these methods have three essential disadvantages: first, the fragments must strongly overlap, and this causes an enormous amount of quasi-redundant computations. Secondly, in order to obtain converged results, the fragments must be rather large, and it may be difficult and very expensive to treat them with canonical methods (this could of course be overcome by using a local method for each fragment). Third, the fragments may not always be simple to define, in particular in strongly conjugated systems. Overall, these methods are by construction much more expensive than efficient local correlation methods.

The essential difference of the current work to fragmentation methods is that only the data used by the different groups are overlapping, but redundant calculations of pair residuals and integrals are avoided. Furthermore, the calculation time for each pair or group is very much smaller than in a canonical fragment calculation. The price to pay is that at the end of each iteration the processes must be synchronized, and the updated amplitudes must be communicated between the groups. Thus, optimized load balancing and communication strategies are necessary.

III. LOCAL DESCRIPTION OF ELECTRON CORRELATION

A. Local occupied orbitals

The localized occupied molecular orbitals (LMOs) $|i\rangle$ can in principle be obtained by any localization scheme, e.g., Boys[122] or Pipek-Mezey[123] (PM) localization; also natural localized molecular orbitals (NLMOs) can be used.[21] In the current work we use the recently introduced intrinsic bond orbitals (IBOs).[124] IBOs are based on a molecule-intrinsic minimal basis derived from free-atom atomic orbitals.[124–127] They resemble PM orbitals, but are insensitive to basis set variations; in particular, the commonly encountered artifacts of PM orbitals with diffuse basis sets are absent.[124,127,128] Additionally, the IBO construction yields stable orbital partial charges which can be used for defining domains (see section IV B). Similar localization schemes were also put forward by Lehtola and Jónsson[128] and West et al.[126]

The construction of IBOs is simple and efficient.[124] Even though our current implementation scales cubically with molecular size, the time to generate the IBOs is found to be short as compared to a PNO-LMP2 calculation, even for cases with more than 9000 basis functions (c.f. section VIII B). Core orbitals are not correlated throughout the current work and therefore excluded from the localization.

B. MP2 with canonical and local occupied orbitals

The MP2 correlation energy can be split into contributions E_{ij} associated with individual pairs of occupied orbitals i, j:

$$\begin{aligned} E_{\mathrm{corr}} &= \sum_{i\geq j}(2-\delta_{ij})E_{ij}, \qquad (1)\\ E_{ij} &= \sum_{a,b}\tilde{T}^{ij}_{ab}K^{ij}_{ab}, \end{aligned}$$

where $K^{ij}_{ab}=(ai|bj)$, $\tilde{T}^{ij}_{ab}=2T^{ij}_{ab}-T^{ij}_{ba}$, and T^{ij}_{ab} are the wave function amplitudes. The T^{ij}_{ab} describe the correlation hole associated with orbital pair $|ij\rangle$, and are determined by the LMP2 amplitude equations. If the virtual orbitals are canonical (i.e., diagonalize the Fock matrix such that $f_{ab}=\delta_{ab}\varepsilon_b$), this equation reads

$$\begin{aligned} 0 &= K^{ij}_{ab}+(\varepsilon_a+\varepsilon_b-f_{ii}-f_{jj})T^{ij}_{ab}+G^{ij}_{ab}+G^{ji}_{ba},\\ G^{ij}_{ab} &= -\sum_{k\neq i} f_{ik}T^{kj}_{ab}. \qquad (2) \end{aligned}$$

In general, Eq. (2) must be solved iteratively. But if canonical occupied orbitals are used (i.e., $f_{ij}=\delta_{ij}\varepsilon_i$), it is solved in closed form by

$$T^{ij}_{ab} = -\frac{(ai|bj)}{\varepsilon_a+\varepsilon_b-f_{ii}-f_{jj}}. \qquad (3)$$

The solution in Eq. (3) is also obtained in the local case if the approximation $G^{ij}\approx 0$ is employed; Following Riplinger and Neese[42], this will be denoted as *semi-canonical* approximation in the following. The energies obtained with semi-canonical amplitudes are too inaccurate for practical applications, but these amplitudes will be useful for generating PNOs.

C. Local approximations in electron correlation

If all pairs ij are retained and Eq. (2) is solved exactly (in the full space of virtual orbitals), then Eq. 1 obtains the exact MP2 energy. However, large efficiency gains can be realized by applying local approximations: Once local orbitals i,j are employed, many pair energies E_{ij} can be neglected completely, others can be approximated with multipole approximations, and in the retained E_{ij}, large portions of the virtual space can be neglected.

This is best seen when writing the two-electron integrals $(ai|bj)$ in terms of charge distributions $\rho_{ai}(\mathbf{r})=\phi_a(\mathbf{r})\phi_i(\mathbf{r})$:

$$(ai|bj) = \int_{\mathbb{R}^3}\mathrm{d}\mathbf{r}_1\int_{\mathbb{R}^3}\mathrm{d}\mathbf{r}_2\ \rho_{ai}(\mathbf{r}_1)\frac{1}{r_{12}}\rho_{bj}(\mathbf{r}_2). \qquad (4)$$

Let us assume i and j are local orbitals. If orbital i is far from j, then also the local charge distribution ρ_{ai} is far from ρ_{bj}, and the integral can be approximated by a *multipole expansion*.[7,8,42] This is discussed in section III E. Since the densities $\rho_{ai}(\mathbf{r})$ carry no charge, the lowest-order contribution is the dipole-dipole interaction, which decays with R_{ij}^{-3}, where R_{ij} is the distance between the charge centers of the two distributions. Consequently, in a local orbital basis, the amplitudes and pair energies quickly decay with R_{ij}^{-3} and R_{ij}^{-6}, respectively. This can be used to neglect distant pairs ij in the *pair approximation*.

If the canonical orbitals a and b are replaced by local orbitals r, s, respectively, then the integral also decays quickly (approximately exponentially) with the distance between $\phi_r(\mathbf{r})$ and $\phi_i(\mathbf{r})$ or between $\phi_s(\mathbf{r})$ and $\phi_j(\mathbf{r})$. This forms the basis for *domain approximations*, which means for each pair ij of occupied orbitals, excitations T^{ij}_{rs} are restricted to subspace of local virtual orbitals r,s that are spatially close to i and j. Such subspaces (domains) are denoted as $r,s\in[ij]$; the choice and construction of the virtual spaces will be discussed in section III D.

In the most general case, we may assign completely independent sets of virtual orbitals $\{p^{ij}\}$ to different domains $[ij]$ ($p^{ij}\in[ij]$). This is done in PNO case described next. Assuming that virtual orbitals within a given domain $[ij]$ are orthogonal and *pseudo-canonical* (that is, they diagonalize the Fock matrix in the respective domain), but that virtual orbitals from different domains can be non-orthogonal, eq. (2) turns into

$$\begin{aligned} R^{ij}_{pq} &= K^{ij}_{pq}+(\varepsilon^{ij}_p+\varepsilon^{ij}_q-f_{ii}-f_{jj})T^{ij}_{ab}+G^{ij}_{pq}+G^{ji}_{qp},\\ G^{ij}_{pq} &= -\sum_{k\neq i} f_{ik}\sum_{u,v\in[kj]}\langle p^{ij}|u^{kj}\rangle T^{kj}_{uv}\langle v^{kj}|q^{ij}\rangle, \qquad (5) \end{aligned}$$

where $\langle p^{ij}|u^{kj}\rangle$ is the overlap matrix between the virtual orbitals p in domain $[ij]$ and u in domain $[kj]$. (If virtual orbitals within a pair domain are not orthogonal or not pseudo-canonical, additional terms occur). For the optimized amplitudes, the residual matrices R^{ij}_{pq} must vanish. With a given set of amplitudes, the correlation energy is computed as

$$E_{\mathrm{corr}} = \sum_{i\geq j}(2-\delta_{ij})\sum_{p,q\,\in\,[ij]}(K^{ij}_{pq}+R^{ij}_{pq})\tilde{T}^{ij}_{pq}. \qquad (6)$$

This expression corresponds to the Hylleraas functional and depends only quadratically on the deviation of the amplitudes from the fully optimized ones.

D. Choice of virtual orbitals

The choice of the virtual orbitals is essential for a compact local representation of the correlated wave function. In the current work, projected atomic orbitals (PAOs), orbital specific virtuals (OSVs) and pair natural orbitals (PNOs) are used in a hierarchical treatment. These three orbital types have been discussed and compared in an earlier paper.[23] In order to introduce the notation and approximations used in the current work, we will briefly review them here.

1. *Projected atomic orbitals (PAOs)*

The simplest choice of local virtual orbitals are projected atomic orbitals (PAOs), as first proposed by Pulay[1]

$$|r\rangle \;=\; \big(1-\sum_i |i\rangle\langle i|\big)\,|r^{\mathrm{AO}}\rangle, \tag{7}$$

where $|r^{\mathrm{AO}}\rangle$ are functions representing atomic orbitals, expressed in the orbital basis by a block-diagonal coefficient matrix $\mathbf{C}^{\mathrm{AO}}$. The use of actual atomic orbitals is helpful (as opposed to, for example, directly using segmented basis sets) because they allow for a clear separation of core orbitals, which should be eliminated from the virtual space. The PAO expansion coefficients in terms of the AO basis are

$$\mathbf{P} = \mathbf{1} - \mathbf{L}\mathbf{L}^{\dagger}\mathbf{S}\mathbf{C}^{\mathrm{AO}}, \tag{8}$$

where $\mathbf{L}$ is the $N_{\mathrm{AO}} \times N_{\mathrm{occ}}$ matrix of occupied orbitals (LMOs), and $\mathbf{S}$ is the AO overlap matrix. We employ generally contracted basis sets, in which $\mathbf{C}^{\mathrm{AO}}$ can be taken as a unit matrix. The PAOs are local, but not orthogonal. The full set is linearly dependent. Non-redundant pseudo-canonical PAO domains can be determined as described in more detail in previous work[6,19–21,129] and outlined in section IV B.

In principle, subsets of PAOs can be used directly to define the virtual spaces $[ij]$ associated with ij pairs, and PAO-based local correlation methods have been implemented up to LCCSD(T) and even higher levels.[6–25] It has been found that with PAOs significant smaller domains are sufficient to achieve a prescribed accuracy than with localized orthogonal virtual MOs.[23] However, in order to recover 99.8% of the canonical correlation energy, still 500–800 PAOs per ij pair may be needed (using triple-ζ basis sets). This number increases linearly with the size of the AO basis. The domain sizes can be further reduced by using pair-specific virtual orbitals, as defined next.

2. *Pair natural orbitals (PNOs)*

PNOs[37–43,97–101] are obtained by diagonalizing (approximate) external MP2 pair density matrices ($i \geq j$)

$$D^{ij}_{ab} \;=\; \frac{1}{1+\delta_{ij}}[\tilde{\mathbf{T}}^{ij\dagger}\mathbf{T}^{ij}+\tilde{\mathbf{T}}^{ij}\mathbf{T}^{ij\dagger}]_{ab}. \tag{9}$$

Various choices of approximate amplitudes $\mathbf{T}^{ij}$ will be discussed in section IV A. The diagonalization yields pair-specific transformation matrices $\mathbf{Q}^{ij}$

$$\left[\mathbf{Q}^{ij\dagger}\mathbf{D}^{ij}\mathbf{Q}^{ij}\right]_{pq} \;=\; n^{ij}_{p}\delta_{pq}. \tag{10}$$

To each PNO $|p^{ij}\rangle = \sum_a |a\rangle Q^{ij}_{ap}$ a natural occupation number n^{ij}_p is associated, and this can be used to define the PNO pair domains, cf. sec. IV C. PNOs for a given orbital pair ij are orthonormal, but the PNOs of different pairs are mutually non-orthogonal.

In the following it will be convenient to pseudo-canonicalize the PNOs in each domain $[ij]_{\mathrm{PNO}}$ by a further unitary transformation $\mathbf{U}^{ij}$, so that

$$|\bar{p}^{ij}\rangle \;=\; \sum_{q\in[ij]_{\mathrm{PNO}}} |q^{ij}\rangle U_{q\bar{p}}, \tag{11}$$

$$\left[\mathbf{U}^{ij\dagger}\mathbf{f}^{ij}\mathbf{U}^{ij}\right]_{\bar{p}\bar{q}} \;=\; \varepsilon^{ij}_{\bar{p}}\delta_{\bar{p}\bar{q}}, \qquad \bar{p},\bar{q}\in[ij]_{\mathrm{PNO}}, \tag{12}$$

where $[\mathbf{f}^{ij}]_{pq} = \langle p^{ij}|\hat{f}|q^{ij}\rangle$. This simplifies the iterative solution of the PNO-LMP2 equations but has no effect on the results. The total transformation from canonical MOs to pseudo-canonical PNOs for a pair ij is then given by

$$\bar{Q}^{ij}_{a\bar{p}} = \sum_{q\in[ij]} Q^{ij}_{aq}U^{ij}_{q\bar{p}}. \tag{13}$$

For the sake of simplicity, we will in the following drop the bar and always assume that the PNOs are pseudo-canonical.

The advantage of PNOs over PAOs is that the number of amplitudes to be optimized is minimized. Typically, only 50-70 PNOs per pair are needed to recover about 99.8% of the correlation energy, which is about an order of magnitude less than in the PAO case (and, consequently, the number of amplitudes and coupled equations are reduced by two orders of magnitude). A further important advantage is that this number grows much slower with the number of basis functions than the number of PAOs.[37] For distant pairs the number of PNOs becomes even smaller and drops to zero if the LMOs i and j are sufficiently far apart. These aspects are illustrated in Fig. 1. The total number of significant PNOs scales linearly with molecular size. Nevertheless, it can become huge in large molecules, and therefore special techniques are necessary to compute the necessary integrals in the PNO basis, cf. section V.

3. *Orbital specific virtuals (OSVs)*

Orbital-specific virtuals (OSVs) have first been proposed and used by Yang et al. for LMP2,[102] and more recently

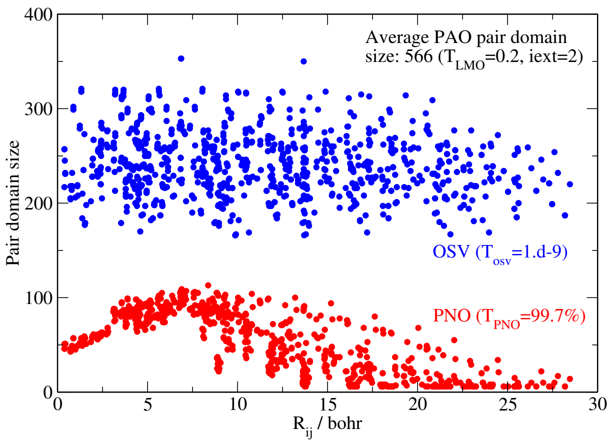

FIG. 1: Sizes of PAO, OSV, and PNO pair domains $[ij]$ in the Gly$_4$ molecule, plotted over the distance R_{ij} between the charge centroids of orbitals i and j. The VTZ-F12 basis is used.

OSV-based methods have been implemented up to the level of LCCSD(T)-F12.[103,105] In this case one set of virtual orbitals is assigned to each LMO i. OSVs correspond to the PNOs for diagonal pairs ii. Since $\mathbf{D}^{ii} = \mathbf{T}^{ii}\mathbf{T}^{ii}$, the OSV transformation matrices $\mathbf{Q}^{i} := \mathbf{Q}^{ii}$ can be obtained simply by diagonalizing the symmetric matrices $\mathbf{T}^{ii}$, and the natural occupation numbers are related to the eigenvalues t_r^{ii} by $n_r^{ii} = (t_r^{ii})^2$. OSV orbital domains $[i]_{\text{OSV}}$ are selected using an occupation number threshold T_{OSV}. Pair domains $[ij]_{\text{OSV}}$ are then obtained by taking the union of the orbital domains $[i]_{\text{OSV}}$ and $[j]_{\text{OSV}}$, followed by orthogonalization and pseudo-canonicalization. The total number of OSVs scales linearly with molecular size. It is much smaller than the number of PNOs, but for a prescribed accuracy the OSV pair domains must typically be larger by a factor of 3-4 than the PNOs ones.[23]

E. Multipole approximations for distant pairs

In order to reduce the computation time and memory requirements, the number of pairs that are included in the PNO-LMP2 method should be minimized. The selection of negligible pairs can be based on multipole approximations, as previously described by Hetzer et al.[7,8] using PAOs, and recently also used by Riplinger and Neese[42] using OSVs. Here we adopt the latter approach. The OSVs $|r^i\rangle$ and $|s^j\rangle$ are assumed to be localized near the LMOs $|i\rangle$ and $|j\rangle$, respectively, and their overlap is neglected. Furthermore, only the integrals $(r^i i|s^j j)$ are kept, where r^i is close to i and s^j close to j. All other integrals, such as the exchange-type integrals $(r^j i|s^i j)$ are neglected. The simplest dipole-dipole approximation for the remaining integrals then is

$$(r^i i|s^j j) = \frac{\sqrt{2}}{(R_{ij})^3}\left[\langle i|\mathbf{r}|r^i\rangle \cdot \langle j|\mathbf{r}|s^j\rangle\right]. \tag{14}$$

This yields for the distant pair energies[42]

$$E_{ij}^{\text{OSV-DIP}} = -\frac{8}{(R_{ij})^6}\sum_{r\in[i]_{\text{OSV}}}\sum_{s\in[j]_{\text{OSV}}}\frac{\left[\langle i|\mathbf{r}|r^i\rangle \cdot \langle j|\mathbf{r}|s^j\rangle\right]^2}{\varepsilon_r^i + \varepsilon_s^j - f_{ii} - f_{jj}}, \tag{15}$$

where R_{ij} is the distance between the charge centers of $|i\rangle$ and $|j\rangle$. In the following this will be denoted as DIP(SC)-approximation. These pair energies can be used to select the distant pairs and to estimate their long-range correlation contribution. As pointed out by Riplinger and Neese,[42] this approximation is computationally very inexpensive, even for large systems. The integrals are computed exactly only for pairs that have energies larger than a threshold T_{dist}, as in the work of Riplinger and Neese. The default value is $T_{\text{dist}} = 10^{-6}$. Alternatively, the LMP2 equations can be solved using all pairs, where either exact integrals $(r^i i|s^j j)$ [NOEXCH approximation] or approximate integrals [DIP(OPT) approximation] are employed. In both cases, all other integral types are neglected. The performance of these three approximations will be compared in section VIII.

The earlier work of Hetzer et al.[7] has shown that higher-order multipole contributions are non-negligible. Typically, the dipole-dipole approximation underestimates long-range pair energies by up to 30%. Including higher-order terms improves the accuracy, but also leads to stronger divergence for small R_{ij}. The optimum compromise recommended in that work is $p = 3$, where p is the highest multipole order l involved, i.e. the above dipole-dipole approximation corresponds to $p = 1$, and the contributions to the integrals decrease with $(R_{ij})^{-(l+l'+1)}$). For more details see Ref. 7.

IV. IMPLEMENTATION OF LOCAL APPROXIMATIONS

A. Linear scaling PNO generation

The CPU time for generating the PNOs from canonical orbitals as described above (section III D 2) scales as $\mathcal{O}(\mathcal{N}^5)$. This is true both for the evaluation of the integrals $(ai|bj)$ needed in eq. (3), as well as for the diagonalization step in eq. (10). Even though it has been argued that the $\mathcal{O}(\mathcal{N}^5)$ scaling is no problem for medium size molecules (up to 50-100 atoms),[37,38] it leads to serious bottlenecks in calculations for larger molecules.

A partial remedy of these problems is to neglect or approximate from the very beginning distant pairs that contribute negligibly to the correlation energy. This reduces the scaling from $\mathcal{O}(\mathcal{N}^5)$ to $\mathcal{O}(\mathcal{N}^4)$. The selection of the neglected very distant pairs can be based on multipole approximations, cf. section III E. The generation of OSVs from amplitudes in the canonical basis also scales as $\mathcal{O}(\mathcal{N}^4)$, though with a much smaller pre-factor than for PNOs.

In order to combine the advantages of OSVs and PNOs, we have previously proposed a hybrid method,[23] in which the PNOs are generated using OSV amplitudes. This significantly reduces the cost for computing the PNOs. The generation of PNOs from OSV amplitudes scales linearly if distant pairs are neglected or treated by multipole approximations. Shortly after, the same idea has also been reported by Hättig et al.[44] Very recently, Riplinger and Neese presented a method in which the PNOs are generated from amplitudes in the PAO basis (using pseudo-canonical PAO domains), which leads to linear scaling as well.[42] We also implemented this method. This will be denoted PAO(SC) approach.

Our current method is closely related to the approach of Riplinger and Neese but introduces an intermediate step in the OSV basis. As will be shown below this reduces the cost for generating the PNOs by about one order of magnitude.

First, PAO orbital domains are created as described in section IV B. The PAOs in each domain are orthogonalized and canonicalized, similar as described above for PNOs. Initial semi-canonical amplitudes T_{rs}^{ii} $(r,s \in [i]_{\text{PAO}})$ are then created using

$$T_{rs}^{ii} \;=\; -\frac{K_{rs}^{ii}}{\varepsilon_r^i + \varepsilon_s^i - 2f_{ii}}. \tag{16}$$

These amplitudes matrices are diagonalized to yield OSVs, and the OSV orbital domains $[i]_{\text{OSV}}$ are selected using an occupation number threshold T_{OSV}. These domains are typically by a factor of 2–4 times smaller than the initial PAO domains. Next OSV pair domains $[ij]_{\text{OSV}}$ are generated by taking the union of $[i]_{\text{OSV}}$ and $[j]_{\text{OSV}}$. The pseudo-canonical orbitals in the resulting pair domains are denoted $|\bar{r}^{ij}\rangle$, with associated orbital energies $\varepsilon_{\bar{r}}^{ij}$. The transformation from the non-orthogonal to the orthogonal pseudo-canonical OSV basis is represented by matrices $\mathbf{W}^{ij}$,

$$|\bar{r}^{ij}\rangle = \sum_{s\in[ij]_{\text{OSV}}} |s^{ij}\rangle W_{s\bar{r}}^{ij}. \tag{17}$$

Next, the integrals are evaluated in this basis. This is done by first computing them in the PAO basis and then transforming them to the pseudo-canonical OSV basis using the transformation matrices $\mathbf{Q}^i$, $\mathbf{Q}^j$ and $\mathbf{W}^{ij}$. This can all be done on the fly in (distributed) memory without any I/O. More details will be give in section V.

A new set of semi-canonical amplitudes T^{ij}_{rs} is then generated according to eq. (16), using the appropriate integrals $(\bar{r}^{ij}i|\bar{s}^{ij}j)$ and orbital energies ε^{ij}_r. The final step is to generate the PNOs using the OSV amplitudes T^{ij}_{rs}. The integrals $(\bar{r}^{ij}i|\bar{s}^{ij}j)$ as well as the overlap integrals $\langle r^{ij}|s^{ij}\rangle$ are transformed from the OSV to the PNO basis accordingly.

Optionally, it is possible to solve the OSV-LMP2 equations iteratively to obtain the OSV amplitudes T^{ij}_{rs} required in the PNO generation. This can be done similarly to eq. (5), but it is much more expensive than solving the PNO-LMP2 equations. In section VIII E it will be demonstrated that the loss of accuracy by using semi-canonical OSV amplitudes to generate the PNOs [OSV(SC) approach], rather than fully optimized ones [OSV(OPT) approach], is small. Therefore semi-canonical OSV-LMP2 amplitudes are used by default.

As already mentioned, there are two advantages of using the PAO-OSV-PNO hybrid approach: First, it is not necessary to generate and store the huge transformation matrices $\mathbf{Q}^{ij}$ that transform from the PAO basis to PNOs. Secondly, the construction and diagonalization of the external pair density matrices $\mathbf{D}^{ij}$ is much faster in the OSV basis than in the larger PAO basis. Since the OSV domains are typically a factor of 2–4 smaller than the PAO ones, the speedup is 1–2 orders of magnitude. As will be demonstrated later, the loss of accuracy is negligible, provided a sufficiently small threshold $T_{\rm OSV}$ is employed.

B. PAO domain selection

The determination of appropriate initial PAO orbital domains is a crucial for the accuracy of the method outlined above. In contrast to the PAO-LMP2 and PAO-LCCSD methods previously developed in our group, these domains are now chosen to be so large that nearly the same accuracy as with the full set of canonical orbitals is achieved.

Each PAO can be associated to the atomic center at which the generating AO (basis function) has its origin. The PAO domains $[i]_{\rm PAO}$ always include all PAOs at a subset of centers, and therefore each domain is defined by a list of centers. To determine these center lists we use a two-step procedure. First, rather small *primary* domains are defined using intrinsic partial charges of IBOs.[124] For each LMO all atoms that have partial charges above a threshold $T_{\rm LMO}$ are included. This threshold has been chosen to be 0.2 for most applications presented in this paper. Typically, the partial charges are either significantly larger or smaller than this value, and therefore these domains should be rather insensitive to changes of the geometry and correspond closely to chemical intuition of molecular bonds. In order to guarantee smooth potentials the domains could also be frozen.[20]

In a second step, the primary domains are extended. A straightforward method would be to simply reduce the threshold $T_{\rm LMO}$, as proposed by Riplinger and Neese[42] (but using Löwdin partial charges that are much more basis set dependent than the IBO charges). We found, however, that even with IBO partial charges this procedure is not fully satisfactory, since very small charges are not physically meaningful and may be scattered over many atoms due to the orthogonalization tails. We therefore extend the center lists by adding all next neighbors (`IEXT=1`), where two atoms are considered to be neighbors (connected by a bond) if their distance is less or equal 1.2 times the sum of their atomic radii. This can be repeated using further shells of neighboring atoms (`IEXT=2`, `IEXT=3` etc.). Using these *extended domains* the canonical limit can be systematically approached.[19] Alternatively, distance criteria (`REXT`) can also be used. This is sometimes advantageous if bonds are stretched (e.g. at transition states) and not recognized as a bond any more by the above criterion. The defaults in our program is `IEXT=2` and `REXT=5` bohr. At least one of these criteria must be fulfilled.

C. PNO domain selection

The conventional method[37–44] to determine the PNO domains is based on the occupation numbers n^{ij}_p, cf. eq. (10). However, as will be demonstrated later, this simple method may lead to too small domains for distant pairs and may therefore significantly underestimate long-range dispersion interactions (unless a very small threshold is used, which then unnecessarily increases the domains for strong and weak pairs). More balanced results are obtained if the PNO domains are determined such that the PNO pair energies closely reproduce the OSV ones for all pairs.

The first step is to generate for a pair ij the full set of PNOs (i.e., as many as OSVs in the pair domain $[ij]_{\rm OSV}$) by constructing and diagonalizing the pair density matrix $\mathbf{D}^{ij}$ in the pseudo-canonical OSV basis for the given pair. Secondly, the integrals K^{ij}_{rs} and the amplitudes T^{ij}_{rs} are transformed from the OSV into the PNO basis. This can be done exactly because the OSV-PNO transformation is unitary and can be inverted. One can then evaluate the pair correlation energy in the PNO basis according to

$$E_{ij} \;=\; \sum_{p,q\in[ij]} \tilde{T}^{ij}_{pq}(p^{ij}i|q^{ij}j). \qquad (18)$$

If the full set of PNOs is used, this yields exactly the OSV pair energy. Reduced PNO domains $[ij]_{\rm PNO}$ are obtained by increasing the domain size one by one (starting with 1 in the order of decreasing natural occupation numbers) until the approximate pair energy reproduces the exact OSV pair energy within a certain threshold $T_{\rm PNO}$, i.e,

$$E^{\rm PNO}_{ij}/E^{\rm OSV}_{ij} \geq T_{\rm PNO} \qquad (19)$$

$T_{\rm PNO}$ is normally chosen to be 0.997-0.998, but it is possible to use smaller thresholds if F12 terms are included (cf. section VIII). Furthermore, it is possible to combine the occupation number and energy criteria, by requiring that both must be fulfilled (e.g. $T_{\rm PNO} = 0.997$ and $T_{\rm PNO\_OCC} = 10^{-8}$). This ensures that the domains of long-range pairs are sufficiently large (by the energy criterion), and increases the accuracy of the short-range pairs (by the occupation number criterion). The procedure also works well if semi-canonical OSV amplitudes are used, despite the fact that the absolute values of the semi-canonical pair energies are not very accurate. Since for diagonal pairs ii the OSVs and PNOs are identical, the corresponding domains are kept unchanged (this is the default; re-adjustment of the PNO domains according to the above criterion is also possible in our program).

Typically, the resulting PNO domain sizes are by a factor of 2-4 smaller than the OSV ones. As will be shown later, even smaller PNO domains can be used in PNO-LMP2-F12 calculations. Furthermore, the PNO domains become smaller for distant pairs.

V. LOCAL DENSITY FITTING INTEGRAL EVALUATION AND TRANSFORMATION

A. Evaluation of PNO-basis integrals

The most time consuming part of a PNO-LMP2 calculation is usually the evaluation of the transformed integrals $K^{ij}_{pq} = (p^{ij}i|q^{ij}j)$. We have succeeded in devising an algorithm that shows excellent scalability with the molecular size and the number of processors/cores,[159] and this algorithm is now described.

In our method, the integrals are first evaluated in the PAO basis and finally transformed on the fly into the OSV basis. We found that this is much more efficient that the direct evaluation of the integrals in the OSV basis, despite large PAO domains. The reason is that that in large molecules there may be many more OSVs than AOs and PAOs, which makes the second transformation step not only very expensive, but also leads to an exceedingly large number of 3-index integrals. Linear scaling is achieved through local density fitting (LDF)[15,17,18,74,130] and using the sparsity of the LMO and PAO expansion coefficients.

In general, density fitting involves the following steps:

$$(A|\mu i) = \sum_{\nu}(A|\mu\nu)L_{\nu i} \quad \text{first half transf.} \tag{20}$$

$$(A|ri) = \sum_{\mu}(A|\mu i)P_{\mu r} \quad \text{second half transf.} \tag{21}$$

$$(\bar{B}|ri) = \sum_{A\in[ij]_{\rm fit}} D_{\bar{B},A}(A|ri) \quad \text{fitting} \tag{22}$$

$$(ri|sj) = \sum_{\bar{B}\in[ij]_{\rm fit}} (\bar{B}|ri)(\bar{B}|sj) \quad \text{assembly} \tag{23}$$

where μ,ν refer to functions in the orbital basis, and A, B are auxiliary fitting functions (DF basis). The integrals are defined as $(A|\mu\nu) = \int \chi_\mu(\mathbf{r}_1)\chi_\nu(\mathbf{r}_1)r_{12}^{-1}\chi_A(\mathbf{r}_2)\,\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2$ with $r_{12} = |\mathbf{r}_1 - \mathbf{r}_2|$. In principle, one could use $\mathbf{D} = \mathbf{J}^{-1/2}$, with $J_{AB} = \int \chi_A(\mathbf{r}_1)r_{12}^{-1}\chi_B(\mathbf{r}_2)d\mathbf{r}_1 d\mathbf{r}_2$. However, it is more efficient (and numerically similarly stable) to instead use a Cholesky decomposition of $\mathbf{J}$:

$$\mathbf{J} = \mathbf{G}\mathbf{G}^T, \tag{24}$$

$$\mathbf{D} = \mathbf{G}^{-1}. \tag{25}$$

In practice, Eq. (22) is then implemented by solving the triangular linear equation system (BLAS3 routine `dtrsm`)

$$(A|ri) = \sum_{B} G_{AB}(\bar{B}|ri) \tag{26}$$

to obtain the transformed integrals $(\bar{B}|ri)$. Compared to the standard density fitting procedure, where one first solves for the fitting coefficients $d_{A,ri} = J^{-1}_{AB}(B|ri)$, followed by the assembly step $(ri|sj) = d_{A,ri}(A|sj)$ (summations over repeated indices implied) the above procedure saves a factor of two in both memory and CPU time (since eq. (26) can be solved *in place* and $\mathbf{G}$ is a triangular matrix).

Linear scaling is achieved as follows: To each LMO i a fitting domain $[i]_{\rm fit}$ of auxiliary functions A is assigned. Note that even though a given LMO i may have significant Coulomb interactions with fitting functions A in the entire molecule, only such functions must be included in $[i]_{\rm fit}$ which are necessary to fit the charge distributions $|i\mu)$, which cannot spatially extend much beyond i itself. The fitting domains are thus determined exactly as the PAO domains (cf. section IV B). The parameters for extending the fitting domains are denoted `IDFDOM`, `RDFDOM` (corresponding to `IEXT`, `REXT` for the PAO domains). The default value are `IDFDOM=3` and `RDFDOM=7` a_0.

In the fitting and assembly steps [eqs. (22), (23)] the summations over A and $\bar{B}$ are restricted to the union of the domains $[i]_{\rm fit}$ and $[j]_{\rm fit}$. Therefore, for a given LMO i, the integral evaluation and transformation [eqs. (20), (21)] must include the united fitting domain $A \in [i]_{\rm ufit}$, which is the union of $[i]_{\rm fit}$ and all fitting domains $[j]_{\rm fit}$ of LMOs j that form pairs with LMO i. Similarly, the range of PAOs in the second half transformation must include the united PAO domains $r \in [i]_{\rm uPAO}$. The sizes of all these domains are asymptotically independent of the molecular size (provided distant pairs are neglected). This leads to linear scaling in the fitting and assembly steps. Note that in order to realize robust density fitting (i.e., obtain errors in the final integrals $(ri|sj)$ which are only quadratic in the incompleteness of the fitting basis $[i]_{\rm fit} \cup [j]_{\rm fit}$), the Cholesky decomposition and the fitting step [Eq. (22)] must be carried out for each ij pair separately, since each pair uses a unique set of fitting functions $[i]_{\rm fit} \cup [j]_{\rm fit}$. We even compute the required domain blocks of $\mathbf{J}$ on the fly for each pair. As will be shown later, this is sufficiently fast and avoids the storage of any quantities that scale quadratically with the molecular size.

Since for a given i the number of r and A are independent of the molecular size, and the integral $(A|\mu i)$ decays exponentially with the distance between i and μ, the second half transformation scales linearly, even without using additional sparsity of the PAO coefficients $P_{\mu r}$. However, the first half transformation still scales quadratically. This can be reduced to linear by exploiting the sparsity of the LMO coefficients $L_{\mu i}$, as described in section V B. In addition, the sparsity of the PAO coefficients can be used, but this has a smaller effect on the efficiency.

The algorithm is driven by center blocks of fitting functions. For given block of fitting functions A one can determine which subsets of LMOs i and AOs ν contribute. The remaining integrals $(A|\mu\nu)$ are prescreened for pairs of AO blocks. Since then ν is bound to A and μ to ν the number of integrals to be evaluated scales linearly. The transformation steps also scale linearly, as explained above.

B. Sparsity of LMO and PAO coefficients

As we have seen in the previous section, linear scaling can only be achieved if the sparsity of the LMO coefficient matrices is exploited in the integral transformation. Simply neglecting small $L_{\mu i}$ leads to rather large errors and is not recommended. However, setting small $L_{\mu i} = 0$ and fitting the resulting approximate LMO $\tilde{\phi}_i$ to the original LMO ϕ_i by minimizing $\int[\phi_i(\mathbf{r}) - \tilde{\phi}_i(\mathbf{r})]^2 d\mathbf{r}$ yields much more accurate results. The fitting procedure leads to a set of linear equations. This is equivalent to what is done in the Boughton-Pulay procedure for domain selection,[129] and similar ap-

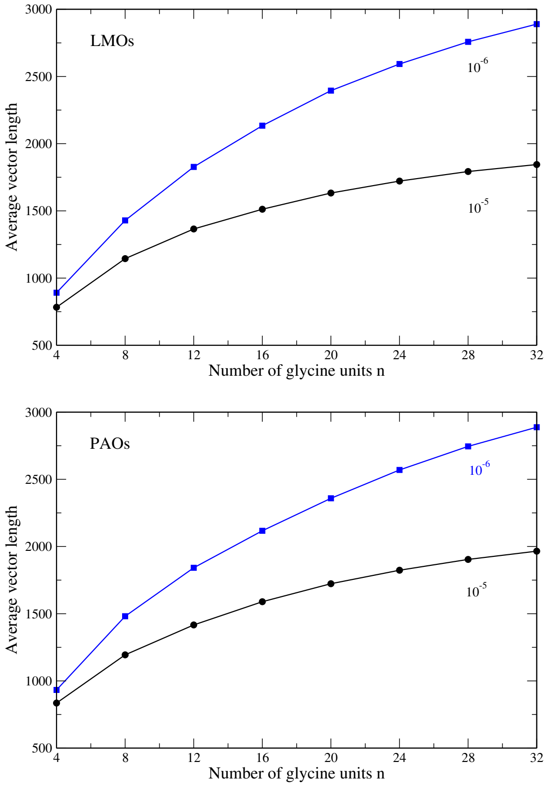

FIG. 2: Average lengths of the truncated LMOs (upper panel) and PAOs (lower panel) for two different threshold $T^{\rm cut}_{\rm LMO}$ and $T^{\rm cut}_{\rm PAO}$, respectively, as a function of the chain length of polyglycine peptides. The aVTZ basis set has been used, see section VIII for details.

proximations have been used in local fragmentation methods as well.[108] PAOs can be truncated and fitted in a similar way.

In detail, we use center based domains and screening algorithms. LMO coefficients $L_{\mu i}$ at a center C are set to zero if

$$\sum_{\mu\in C} L^2_{\mu i} \leq T^{\rm cut}_{\rm LMO}, \qquad (27)$$

where the sum runs over all basis functions at center C. For PAOs, blocks corresponding to center pairs are considered. The coefficients $P_{\mu r}$ are set to zero if for a pair of centers C, D

$$\sum_{\mu\in C} P^2_{\mu r} \leq T^{\rm cut}_{\rm PAO} \qquad \forall r \in D. \qquad (28)$$

The errors caused by these approximations are sufficiently small if $T^{\rm cut}_{\rm LMO} = 10^{-5}$ and $T^{\rm cut}_{\rm PAO} = 10^{-6}$.

Fig. 2 shows the average vector length of the truncated LMOs (upper panel) and PAOs (lower panel) for glycine polypeptide chains for two different thresholds (using the aVTZ basis set, see section VIII). While from theoretical arguments one would expect the vector length to reach a constant, even in the long linear glycine chains studied here we do not see this convergence. This is probably due to both boundary effects and the long orthogonalization tails caused by the diffuse basis functions. Further investigations are required in this regard, in particular to see if more complex criteria than (27) and (28), which take the basis function overlap into account, can improve the situation. For (Gly)$_{32}$ 60.5 % and 73.1% of the PAO coefficients are neglected for thresholds $T^{\rm cut}_{\rm PAO}$ of 10^{-5} and 10^{-6}, respectively. The sparsity of the LMOs for the same values $T^{\rm cut}_{\rm LMO}$ is very similar. The errors in the correlation energies are 0.12 mH and 0.013 mH for the two thresholds, respectively (the total correlation energy without truncation is -24.640580 H).

VI. PARALLELIZATION

Once the LMOs and PAOs are available, our entire PNO-LMP2 program scales nearly linearly with molecular size. Except for the initial overlap matrix and Fock matrix in the PAO basis, the memory requirements for all necessary quantities (integrals, amplitudes, overlap matrices, transformation matrices) scale linearly with molecular size. The memory per pair which is needed in the integral transformation and the LMP2 iterations becomes asymptotically independent of the molecular size. If the number of cores grows proportionally to the number of pairs, the memory requirements per core thus remain constant.

The OSV generation is parallelized over groups of LMOs i, while in most other parts of the program parallelization is over groups of pairs ij. To each processor is assigned a group of residuals $\mathbf{R}^{ij}$ which it needs to compute. If distant pairs are neglected (or treated by multipole approximations, cf section III E), the number of amplitude matrices that contribute to a given group of residuals $\mathbf{R}^{ij}$ [cf. eq. (5)] becomes asymptotically independent of the molecular size. The same is true for the PNO overlap matrices. Thus, the CPU time and memory per residual becomes independent of the molecular size, and a processor only requires a limited amount of data (and thus communication) in order to compute its assigned residuals. The more processors are available, the smaller the memory requirement per core (up to the limit that each residual is computed on a different processor).

At the end of each LMP2 iteration, the updated amplitudes are written to a global array (GA),[131] which can be accessed from each processor independently (for details of the GA software see http://hpc.pnl.gov/globalarrays/). The GA is irregularly distributed over the processors, so that the amplitude matrices of the pairs that are allocated to a given processor are stored locally. Synchronization is necessary only at the end of each iteration, after all updated amplitudes have been written. At the beginning of the next iteration, each processor reads the amplitude matrices which it needs from the GA, except those that it owns and that are already in its local memory. This yields a “scalable” algorithm, i.e. the elapsed time, memory, and communication per processor become (nearly) independent of the molecular size if the number of processors grows linearly with molecular size.

The most demanding part of the calculation is the evaluation and transformation of the two-electron integrals as described in section V. We have implemented and tested various parallelization strategies for the density fitting algorithm. With local density fitting it is most efficient to parallelize the integral evaluation and transformation steps over blocks of fitting functions, and the fitting/assembly steps over pairs of LMOs, using dynamic load balancing in both cases. This requires, however, to communicate the transformed integral blocks to other processors. As for the amplitudes in the iterations, the integrals are stored in a GA and retrieved independently by the individual processors when needed. The communication overhead caused by repeatedly accessing the same integral blocks is reduced by a buffering scheme. The communication time is found to be insignificant on a single node with shared memory. However, on networked multimode systems the communication can become a problem, depending on the network speed. While with an Infiniband network the communication was never a bottleneck, a simple Gbit network was found to be insufficient to achieve reasonable speedups.

A simpler, yet efficient parallelization method is to compute groups of operators $\mathbf{K}^{ij}$ independently on different processors, similar to what has been described above for the LMP2 iterations and other steps. The groups of pairs should be chosen such that the number of LMOs in each group is as small as possible and the amount of work on each processor is as similar as possible, since in this case the distribution on the processors is static. We use the METIS graph partitioning technique[132] to achieve this goal. This minimizes the sizes of the united PAO and fitting domains and also minimizes the amount of redundant integral calculations. This parallelization method avoids any communication between processors. However, due to the unavoidable redundant integral evaluations on different processors as well as the static parallelization scheme it is slower than the internally parallelized transformation program, provided a fast network such as Infiniband is available. It may be advantageous to use with a slow Gigabit network.

As a third alternative, we have also implemented an algorithm that parallelizes the transformation and fitting steps over groups of LMOs, accepting some redundant AO integral evaluations. This algorithm is used by default if local fitting is disabled, since then the fitting step proceeds outside the assembly loop. Again a GA and a buffering system is used to store and retrieve the transformed integrals.

VII. EXPLICIT CORRELATION

In the current work we use the so-called approximation 3**A*** with pair specific projectors, as described in detail in several previous papers.[71,74,75,95] It will therefore not be repeated here. Currently, the integrals are still computed using previously developed programs. They allow for local density fitting and local resolution of the identity (RI) approximations as described in Ref. 74 and scale almost linearly with molecular size. However, even though the algorithms are parallelized, they still involve a lot of disk I/O and are therefore not scalable over several compute nodes. The development of scalable algorithms similar to those described in this work is in progress.

One technical point is worth mentioning: in approximation 3**A*** the evaluation of the F12 correction is uncoupled from the conventional amplitudes and therefore noniterative. However, the strong-orthogonality projector depends on the chosen virtual orbitals and domains.[71,75] But since this affects only the contributions arising from the term $-\sum_{ab\in[ij]}|ab\rangle\langle ab|$ in the projector, it is advantageous to compute and store the energy contributions arising from these and the other terms separately. In case that one is interested both in the OSV-LMP2-F12 and PNO-LMP2-F12 energies, one only needs to re-evaluate the above-mentioned projection terms in the PNO basis. The required integrals such as $F^{ij}_{pq} = \langle pq|\hat{F}_{12}|ij\rangle$ can of course be obtained by transformation from the OSV into the PNO basis. Therefore the computation of the F12 correction for PNO-LMP2-F12 is very cheap once the OSV-LMP2-F12 correction has been computed.

Our method does not explicitly use the complementary auxiliary basis set approach (CABS)[52,133] since this makes it difficult to employ local RI approximations. However, if the union of the orbital basis and the RI basis is used to approximate the RIs, exactly the same results as with the CABS method are obtained. This can be done most conveniently with the VnZ-F12 orbital basis sets[134] along with the corresponding `OPTRI` sets of Yousaf and Peterson.[135] The latter are constructed so that linear dependencies in the united basis sets are avoided. Recently, the construction of local RI spaces from geminal spanning orbitals has been proposed,[87] but this has not yet been considered here.

VIII. BENCHMARK CALCULATIONS

In this section we will demonstrate the accuracy and efficiency of our PNO-LMP2 method. First we will compare different approaches for computing the PNOs and present some timing data for the individual computational steps (VIII A). In section VIII B and VIII C we will report benchmark calculations for glycine and alanine polypeptides that demonstrate the scaling behavior of CPU-time and memory, as well as the parallel efficiency. Subsequently, in section VIII D we will demonstrate the dependence of computed correlation energies and energy differences (reaction energies, barrier heights) on the various parameters that determine the virtual orbitals and domains in our approach. Finally, in section VIII F we will investigate long-range effects and the distant pair approximations. The reactions studied are shown in Fig. 3. The structures of all molecules can be found in the supporting information.

We employed two different basis sets of triple-ζ quality: the first one is denoted aVTZ and consists of the aug-cc-pVTZ basis sets[136] for first-row atoms (C,N,O), aug-cc-pV(T+d)Z basis[137]) for second-row atoms (Cl), and cc-pVTZ[138] for H-atoms. Secondly, the VTZ-F12 basis sets of Peterson et al.[134] were used, which are specially designed for F12 calculations. All calculations were carried out using density fitting (DF) approximations for the evaluation of the two-electron integrals. For computing the Fock matrix an aug-cc-pVTZ/JKFIT basis was used that was derived from Weigend's cc-pVTZ/JKFIT basis[139] by adding for each angular momentum another shell of diffuse functions in an even tempered manner. For all other integral types the aug-cc-pVTZ/MP2FIT sets[140] were used (the diffuse functions on hydrogens were omitted in the calculations for polypeptides). For the resolution of the identity in the F12 calculations the aug-cc-pVTZ/JKFIT were employed [without using the complementary auxiliary basis set (CABS) approach]. We also carried out some tests with CABS and optimized CABS basis sets[135] and found that the differences of the computed reaction energies are very small.

Typically, the aVTZ and VTZ-F12 basis set yields better than aug-cc-pV5Z quality results for energy differences when used in MP2-F12[58,62] or CCSD(T)-F12 calculations.[60,72,141,142] The diffuse functions were included since they are known to be important to reach such accuracy. Furthermore, we wish to demonstrate that low-order scaling is also possible with diffuse basis sets.

The method is implemented in the development version[143] of the Molpro quantum chemistry package.[144]

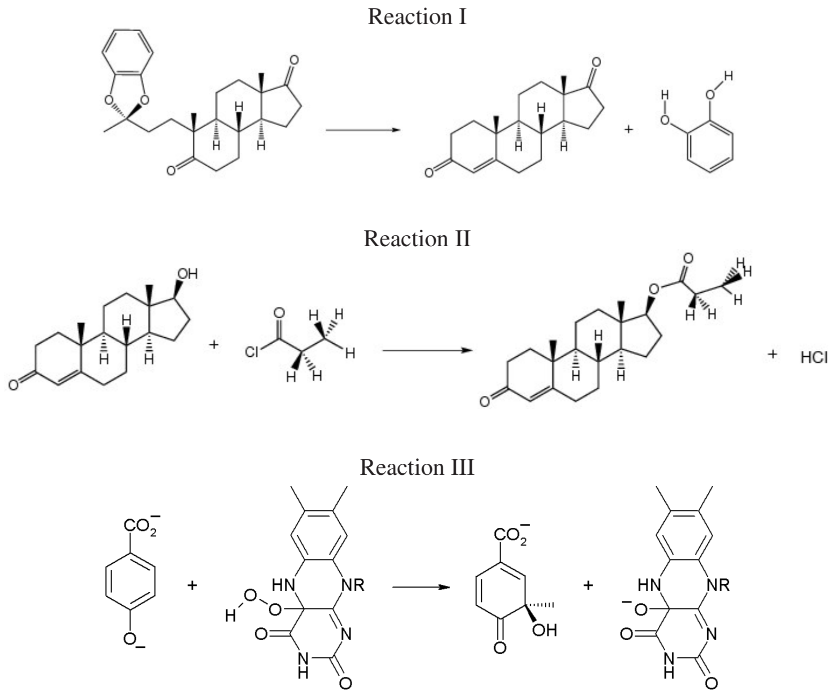

FIG. 3: The three benchmark reactions studied in this paper . For reactions **I** and **II** reaction energies are computed, for **III** the reaction barrier height.

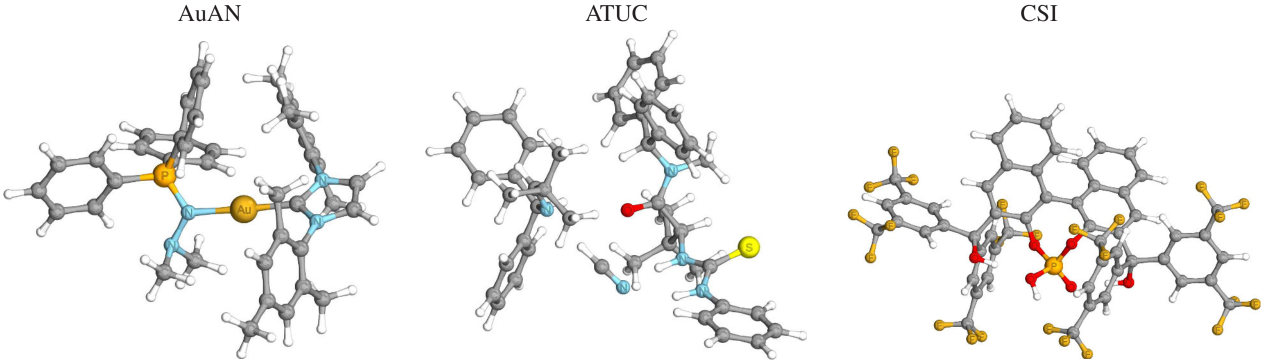

FIG. 4: Three-dimensional benchmark molecules

All benchmark calculations were performed on a compute cluster with Xeon E5-2680 v2 @ 2.8 GHZ processors. Each node has 2 processors with a total of 20 cores and 256 GB of memory. Hyperthreading was not used. The nodes are connected by an QDR-Infiniband network.

A. Analysis of computation times

We start with a comparison of timings for the individual steps of PNO-LMP2 calculations, cf. Table II. The time required to compute the Fock matrix is not included, since it is taken from the last Hartree-Fock (HF) iteration and not recomputed in the PNO-LMP2 program. Furthermore, the F12 terms are not considered since a scalable F12 program is still under development (cf. Section VII). Throughout this paper, only elapsed times will be shown (i.e., differences in wall-clock time including all communication and I/O). The timings are shown for the androstendione precursor molecule, i.e., the reactant of **I**. OSV(SC), OSV(OPT), and PAO(SC) refer to the approaches for PNO generation discussed in section IV A.

The total times for the OSV(SC) approach are dominated by the integral evaluation and transformation (using local density fitting). Since the integrals K_{rs}^{ij} are first computed in the non-orthogonal PAO basis and only at the end transformed to the pseudo-canonical OSV or PAO basis, the computation times are similar in all 5 cases shown in table II. They depend, of course, on the size of the initial PAO domains and on the sizes of the local density fitting domains, which are the same in all cases. The remaining computational steps, such as generation of the PNOs, evaluation of the PNO overlap matrix, and solving the PNO-LMP2 equations take in the OSV(SC) approach very little time. The threshold T_{OSV} has only a rather weak effect on the overall computation time. The additional effort for $T_{\mathrm{OSV}} = 10^{-10}$ as compared to $T_{\mathrm{OSV}} = 10^{-9}$ mainly stems from the diagonalizations and transformations required to determine the pseudo-canonical OSV and PNO orbitals, which depend cubically on the OSV pair domain sizes.

In the PAO(SC) approach (last column) the computation times to generate the pseudo-canonical PAO and PNO pair domains are strongly increased, since the average PAO pair domain size is roughly twice as large as the OSV one, even for the very small threshold $T_{\rm OSV} = 10^{-10}$. As expected, this increases the computation times for the PNO generation by an order of magnitude. Moreover, the memory requirements are significantly larger. The increase of the memory with increasing OSV (or PAO) domain size is due to the fact that the number of integrals and amplitudes depend quadratically on the domain sizes. In the OSV cases, the transformation matrices $\mathbf{W}^{ij}$ grow linearly with the OSV and PNO domain sizes. In the PAO(SC) case these matrices transform from PAO to PNO and are approximately a factor of 2-3 larger. Overall, the PAO(SC) calculations are about 2-3 times as expensive as the OSV(SC) one for $T_{\rm OSV} = 10^{-10}$, while the results agree within 0.2-0.3 KJ mol^{-1} (cf. section VIII D).

The middle columns of Table II show the case that the OSV-LMP2 equations are solved iteratively. Since the evaluation of the LMP2 residuals depends cubically on the domain sizes, the OSV-LMP2 iterations are dramatically more expensive than the PNO-LMP2 ones. In the current case the OSV domains are 4-5 times larger than the PNO ones, and consequently the PNO iterations are 2 orders of magnitude faster than the OSV ones. The same will be true for the LCCSD iterations (currently under development). Even though our program also allows for solving the PAO-LMP2 equations iteratively, this would be extremely expensive and has therefore not been attempted. Note that in our previous PAO based methods[6,9,12,15,18] very much smaller PAO domains were used, and even then the solution of the LMP2 equations was often rather expensive.

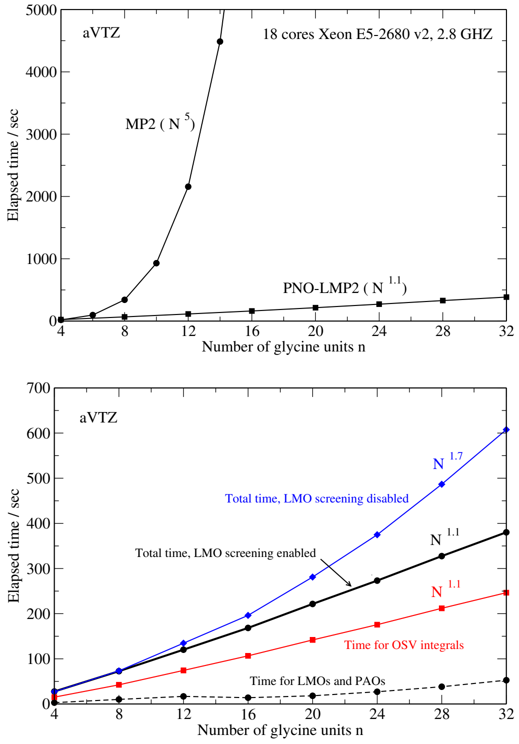

FIG. 5: Total elapsed times of PNO-LMP2 calculations for glycine polypeptides. Upper panel: comparison of canonical and local density fitted MP2 calculations, using 18 cores on one node. Lower panel: PNO-LMP2 elapsed times on a larger scale (using 16 cores on one node). Circles (black): total PNO-LMP2 times with LMO, and PAO screening. Diamonds (blue): total PNO-LMP2 times without LMO/PAO screening. Squares (red): times for OSV integral evaluation. Dashed line: times for non-linear parts (LMO and PAO generation and fit; this part, which formally scales cubically, is not included in the other graphs). All timings were obtained with density fitting approximations.

B. Demonstration of scaling with molecular size

In this section the scaling of elapsed times and memory usage with the molecular size is demonstrated. In order to reach the asymptotic scaling behavior quickly, linear glycine polypeptide chains (Gly)$_n$ (up to $n = 40$) as well as alanine polypeptides (Ala)$_n$ (up to $n = 32$) in the 3_{10}-helical conformation are used. Of course, these model systems are quite unrealistic for real applications. Nevertheless they are helpful for testing and detecting efficiency problems in the algorithms and have often been used by other authors as well. The alanine polypeptide helices are much more dense than the linear glycine chains and contain more complex intramolecular interactions (including hydrogen bonds). A comparison of the two systems can therefore give some insight about the effect of the secondary molecular structure on the timings and memory requirements. Furthermore, in section VIII C we will present some test calculations for the 3-dimensional molecules shown in Fig. 4.

The largest calculations for (Ala)$_{32}$ and (Gly)$_{40}$ were of comparable size and included 323/283 atoms, 9674/9114 contracted basis functions (CGTOs), 21926/20726 auxiliary basis functions for the density fitting, 904/888 correlated electrons, 18113/8535 pairs optimized in LMP2, and 997084/605961 PNOs for (Ala)$_{32}$/(Gly)$_{40}$, respectively. Currently our Hartree-Fock program precludes us from testing the PNO-LMP2 method for even larger molecules (vide infra).

The LMOs were determined using the IBO method,[124] and the domains were determined as described in section IV B (using $T_{\rm LMO}$=0.15 and `IEXT=2`). The fitting domains included all auxiliary functions at the primary atoms and 3 shells of neighboring atoms (`IDFDOM=3`). Distant pair energies were estimated using the dipole-dipole approximation (cf. section III E), and only pairs with pair energies $|E_{ij}| > T_{\rm dist} = 10^{-6}$ where included in the iterative LMP2. The LMO and PAO truncation thresholds (cf. section V B) were set to $T^{\rm cut}_{\rm LMO} = 10^{-5}$ and $T^{\rm cut}_{\rm PAO} = 10^{-6}$. The OSV threshold $T_{\rm OSV}$ was 10^{-9} and the PNO threshold $T_{\rm PNO}$=0.998 (which means a target of 99.8% of the OSV pair energies). The PNO-LMP2 calculations yielded quite consistently 99.7% of the corresponding canonical correlation energies (tested up to (Gly)$_{16}$). This percentage is mainly determined by the PNO domain sizes and can be increased to 99.9 % using tighter thresholds (see supporting information). The effect of the domain sizes on absolute and relative energies will be investigated in more detail in section VIII D.

Fig. 5 demonstrates the scaling of the elapsed times for (Gly)$_n$ as a function of the peptide length. The upper panel shows a comparison of canonical and local MP2 calculations, using 18 cores on a single node. The lower panel shows a comparison of timings with and without using the sparsity of the LMO and PAO coefficients; without this screening quadratic scaling is expected in the integral evaluation and transformation. That the observed scaling exponent of 1.7 is somewhat smaller than 2 is due to the contribution of other parts of the calculations, which scale linearly. If the LMO and PAO screening is enabled, the scaling exponent drops to 1.1 (determined from the (Gly)$_{32}$ vs. (Gly)$_{28}$ elapsed times, using the number of electrons as a measure of the molecular size). The remaining small non-linear over-

TABLE II: Dependence of pair domain sizes and timings for the androstendione precursur on the OSV and PNO thresholds. 20 cores on one node were used in all cases.[a]

Amplitudes for PNO generation	OSV(SC)		OSV(OPT)		PAO(SC)
OSV Threshold:	10^{-9}	10^{-10}	10^{-9}	10^{-10}	
Average PAO pair domain size	795	795	795	795	795
Average OSV pair domain size	270	380	270	380	-
Average PNO pair domain size	77	85	79	87	93
Elapsed times in seconds:					
OSV generation[b]	23	23	23	23	23
OSV or PAO pair domains[c,d]	10	20	10	20	110
Integrals and transformation[d]	159	161	159	160	165
OSV Iterations[e]	1	2	619	1482	1
PNO generation	5	11	5	11	62
PNO S-matrix	13	26	13	27	122
PNO-LMP2 iterations[f]	17	21	9	11	33
Total	231	265	839	1736	524
Maximum memory per core (MW):	126	186	216	378	144
Maximum memory+GA space per core (MW):	131	190	227	401	358

a) Basis VTZ-F12 (2113 CGTOs), $T_{\rm LMO} = 0.2$, IEXT=2, $T_{\rm PNO} = 0.997$.
The PNO domains for diagonal pairs are determined by $T_{\rm OSV}$.
Number of pairs in LMP2: 2145; number of distant pairs (treated by multipole approximation): 936; number of correlated electrons: 156.

b) Including evaluation of integrals K^{ii}_{rs} and multipole treatment.

c) Including redundancy check and transformation to pseudo-canonical basis.

d) Using local fitting (IDFDOM=3).

e) Including evaluation of OSV S-matrix by transforming $S_{\rm PAO}$.

f) PNO-LMP2 with OSV(SC) or PAO(SC) amplitudes.: 5 iterations.
PNO-LMP2 with OSV(OPT) amplitudes: 2 iterations, starting with projected OSV amplitudes. OSV-LMP2: 7 iterations.

head is mainly due to the problem that the LMO and PAO AO-domains are not yet constant for these molecular sizes, cf. Fig. 2. Furthermore, there are always some unavoidable parts, such as integral screening, that scale quadratically, though with a tiny prefactor. It should be noted that the PAO screening reduces the overall times only for very large molecules, since the fitting procedures also takes a non-negligible amount of time. Therefore this screening protocol is not enabled by default.

As shown in the lower panel of Fig. 5 the computation of the integrals $(ri|sj)$ $(r,s \in [ij]_{\rm OSV})$, required for the PNO construction, is by far the most expensive part of the whole calculation. For (Gly)$_{32}$ it takes 63% of the total LMP2 time (not including LMO and PAO calculation); further 17% are used for computing the initial "diagonal" integrals K^{ii}_{rs} $(r,s \in [i]_{\rm PAO})$, which are needed to compute the OSVs. Solving the PNO-LMP2 residual equations (7 iterations) takes only 8%; the remaining 12% are used for generating the OSVs and PNOs, as well as the corresponding overlap matrices.

These PNO-LMP2 timings do not include the computation of the LMOs and PAOs and their screening (using the method outlined in section V B). The time for these parts, which formally scale cubically with molecular size, is shown by the dashed line in Fig. 5. The observed scaling is (for these molecular sizes) much lower than cubic. This is caused by the parallel matrix multiplications used in these steps, which become more efficient as the molecular size increases.

Fig. 6 (upper panel) shows the dynamic memory (local and global) allocated per core as a function of the molecular size. In most cases the maximum amount of memory is used in the integral transformation program. The global array (GA) space scales linearly with molecular size. The amount of local memory is slightly non-linear since for these molecular sizes the number of non-zero LMO and PAO coefficients increases more than linearly with the molecular size, cf. Fig. 2. The lower panel of the figure shows for the (Gly)$_{32}$ case (7306 basis functions) that the memory requirements per core decrease almost linearly with the number of cores (for small numbers of cores). If a single node with 20 cores is used, the program needs 150 MW (1.2 GB) of dynamic memory (including GA space) per core, i.e. 24 GB overall. If 2 or more nodes are used (40 or more cores) the total memory per node drops to less then 16 GB (this does not include memory for the program code and a small amount of static memory, e.g. for common blocks that hold the geometry and basis set information).

In our current program the initial calculation of the LMOs requires the replicated storage of 3 full matrices plus the LMO vectors on each core. The HF program has a similar or even larger memory requirement, and this is still the bottleneck in very large calculations. Clearly, this has to be improved in the future, and this part will not be further discussed in the current paper.

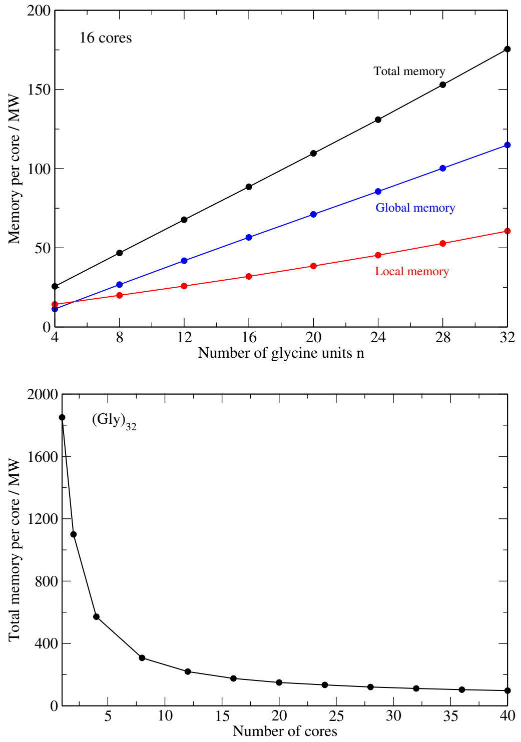

FIG. 6: Upper panel: Dynamic memory per core (triangles down (red), GA-space per core (triangles up, blue), and total memory per core (circles, black) needed for the PNO-LMP2 calculations for glycine polypeptides (16 cores used). Lower panel: Total memory per core as a function of the number of cores used. All data in Megawords (MW); 1 MW=8 MByte (MB).

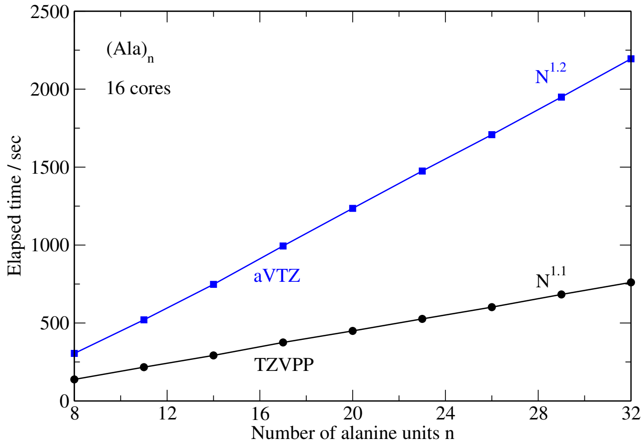

FIG. 7: Total elapsed times of PNO-LMP2 calculations for alanine helices (Ala)$_n$. The aVTZ and TZVPP basis set were used for comparison. The orbital localization and PNO generation is not included in the timings. The calculation for (Ala)$_{32}$ has 323 atoms and 7259 and 9674 basis functions for the TZVPP and aVTZ basis sets, respectively.

In the subsequent calculation of the PAOs and OSVs, and the transformation of the overlap and Fock matrices into these bases, at most two full matrices ($\mathbf{S}_{\rm PAO}$ and $\mathbf{F}_{\rm PAO}$) are currently needed in memory. For example, in the case of (Gly)$_{40}$ (9114 basis functions) these matrices require about 180 MW of memory. This amount is only exceeded by other parts of the calculation if 16 or less cores are used. Therefore the total memory requirement level off for larger numbers of cores. It should be possible, however, to reduce the memory requirements by only storing the domain blocks of $\mathbf{S}_{\rm PAO}$ and $\mathbf{F}_{\rm PAO}$ that are actually needed on a given processor. Further efforts are needed to remove this bottleneck.

The scaling of the elapsed times with molecular size for the (Ala)$_n$ helices in shown in Fig. 7. Using the TZVPP basis a scaling exponent of about 1.1 is reached, while with the more diffuse aVTZ basis it is slightly larger (1.2). This reflects the reduced sparsity of the LMO and PAO coefficients. For (Ala)$_{32}$ the LMO/PAO sparsities are (86.9%/78.1%) and (66.6%/58.7%) for the TZVPP and aVTZ basis sets, respectively. Other conclusions are similar as for the (Gly)$_n$ molecules. However, as expected the elapsed times and memory requirements are larger than for the (Gly)$_n$ chains. This will be further discussed in the next section.

C. Demonstration of scaling with number of processors

Fig. 8 demonstrates the speedup with the number of cores on a single node. The upper panel shows calculations for (Gly)$_{20}$ with a varying number of cores on one node relative to a calculation with 1 core. The observed speedup of 18 with 20 cores is very satisfactory. This indicates that the memory bandwidth is less of a problem than it used to be on older hardware. The lower panel demonstrates for (Gly)$_{40}$ that our program also scales quite well over multiple nodes with up to about 80-100 cores.This may seem surprising, considering that even on a single node with 18 cores the Gly$_{32}$ calculation takes only 6.5 minutes (Fig. 5). Very similar speedups are achieved for (Ala)$_{32}$, cf. Fig. 9. With more cores the speedup curves becomes flatter, but this is mainly a result our integral evaluation and transformation program being parallelized over center blocks of fitting functions. Therefore, near-optimal speedups can only be expected as long as the number of centers is much larger than the number of cores—but in (Gly)$_{40}$ there are only 283 centers [of which 122 are H-atoms and 161 heavy atoms (C, N, O)]. In the limiting case that the number of cores equals the number of centers, the overall time of of this program step will be determined by the center with the largest number of fitting functions; the cores treating smaller atoms (e.g. hydrogens) will be partly idle. This explains why the slopes of the speedup curves become smaller if many cores are used. It should be possible to reduce this problem by splitting center blocks of fitting functions, but has not been attempted so far.

In order to test the performance of our program for systems that are of interest in real-life chemistry we carried out some additional calculations for the molecules shown in Fig. 4. The first is a gold(I)-aminonitrene complex (Au $C_{41}H_{45}N_4P$, in the following denoted AuA), taken from Ref. 145 (but the structure has been re-optimized). The ECP60MDF effective core potential for the inner 60 electrons along with the aug-cc-pVTZ-PP basis set was used for the gold atom. The second system is the reactant state of the amido-thiourea catalyzed enantioselective imine hydrocyanation ($C_{46}H_{53}N_5O$ S, denoted here ATU) from Ref. 146. The third test system is a binaphthyl-based chiral sulfonimide ($C_{54}H_{25}O_6F_{24}P$, denoted CSI) from Ref. 147, which plays a role in a highly enantioselective organocatalysis.[147] The optimized geometry of the structure denoted as “DSI 2” in Ref. 147 has been used. All three structures can be found in the supporting information. Theoretical studies of reactions involving some of these molecules are in progress and will be published elsewhere.

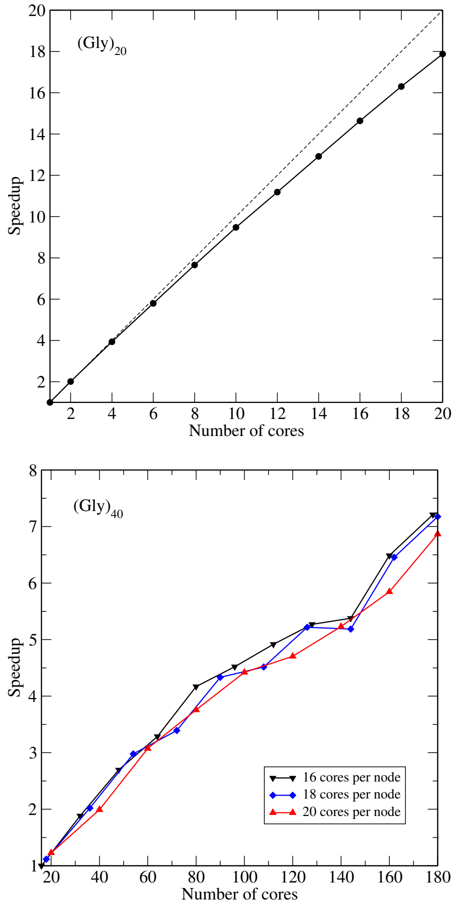

FIG. 8: Upper panel: Speedup for (Gly)$_{20}$ calculations relative to 1 core with the number of processing cores. The dashed line is the ideal speedup. Lower panel: Speedup for (Gly)$_{40}$ calculations relative to 16 cores on one node with the number of processing cores and compute nodes.

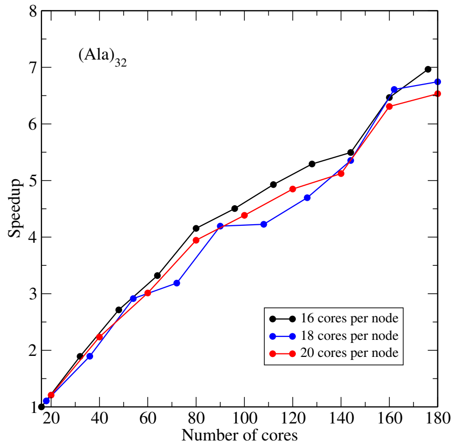

FIG. 9: Speedup for (Ala)$_{32}$ calculations relative to 16 cores on one node with the number of processing cores and compute nodes. The aVTZ basis set was used. The speedups with the TZVPP basis set are very similar and not shown.

Here we are mainly interested in two questions: (i) How large are the memory requirements for such rather dense and 3-dimensional systems; and (ii) do communication bottlenecks occur. In table III some computational details and the memory requirements for the 3 molecules as well as for (Ala)$_{32}$ are listed, using varying numbers of cores (this information is independent of the number of nodes). The elapsed times for varying numbers of cores and nodes are shown in Fig. 10.

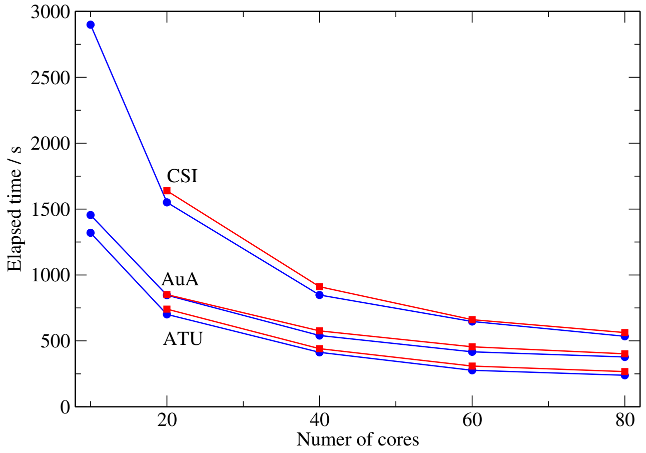

FIG. 10: Total elapsed times of PNO-LMP2 calculations as a function of the number of cores and nodes. Blue lines (circles): 10 cores per node; red lines (squares): 20 cores per node.

The memory requirements decrease with increasing number of cores, though not quite linearly, due to the non-sparse replicated storage of the orbital, Fock, and overlap matrices, as already mentioned. With 80 cores, the calculation for (Ala)$_{32}$ requires less than 4 GB of memory per core (including GA space) and needs about 10 minutes of elapsed time (including orbital localization and PAO generation). This may be compared to the (Gly)$_{40}$ calculation, which only takes about 3 minutes on 80 cores. The difference is due to the more dense structure of the (Ala)$_{32}$ helix. This increases the number of non-distant pairs approximately by a factor of 2. The effect of the molecular structure is even more pronounced when comparing the (Ala)$_{32}$ and CSI calculations. Both need nearly the same memory and time, even though (Ala)$_{32}$ has about 3 times as many atoms and twice as many basis functions. The larger computational resources needed for CSI are partly due to the dense 3-dimensional structure of this very electron-rich molecule. Furthermore, it has several aromatic rings (including 2 naphthyl groups), and the domains in the conjugated π-electron systems are larger than in more saturated molecules. The calculations for the other two molecules took roughly half as long and need half as much memory as the ones for CSI and (Ala)$_{32}$.

In Fig. 10 it can be observed that the calculations with 10 cores per node are always somewhat faster than those with 20 cores per node, if the total number of cores is the same. Similarly, in Figs. 8 and 9 the calculations with 16 cores per node are mostly faster than those with 18 or 20 cores per node. This effect could be due to the limited memory bandwidth or reduced clock speeds if many cores of a CPU are used (the machines run in Turbo mode, i.e. the clock speed may depend on the CPU load). It is not easily possible to measure the communication times directly, since in many places of our program communication and computations are done concurrently. However, since the elapsed times are even shorter when more nodes are used (i.e. more communication procedes through the Infiniband network), and also the speedups are very similar from 10 to 20 cores on one node and from 20 to 40 cores on 2 and 4 nodes we conclude that no significant communication bottlenecks occur in any of these calculations.

In summary, the results in this section show that rather

TABLE III: Computational details and memory requirements of PNO-LMP2 calculations for $(\text{Ala})_{32}$ and the three molecules in Fig. 4

				Correlated	LMP2	Distant		Max. memory per core^b			
Molecule	Basis	Atoms	CGTOs	orbitals	pairs^a	pairs^b	PNOs^c	10	20	40	80
AuA	VTZ-F12	92	3345	122	5128	2375	362712	649	431	280	207
ATU	VTZ-F12	106	3772	137	5279	5174	324384	572	352	229	162
CSI	VTZ-F12	110	4964	225	8201	17224	633079	1245	768	491	322
$(\text{Ala})_{32}$	aVTZ	323	9674	452	18113	84265	997084	1163	721	504	389

a) Number of pairs optimized in PNO-LMP2.

b) Number of distant pairs, treated by non-iterative multipole approximation, $T_{\text{dist}} = 10^{-6}$ E$_h$.

b) Max. memory + GA space per core (in MW) for the number of cores given in the title line.

c) Total number of PNOs, $T_{\text{OSV}} = 10^{-9}$, $T_{\text{PNO}} = 0.998$.

large PNO-LMP2 calculations can be carried out on a single workstation with virtually no disk I/O. Significant further speedups and a reduction of the memory requirements per core are possible by using several nodes. The PNO-LMP2 times are completely negligible as compared to the preceding Hartree-Fock calculation. Without using any local approximations the DF-HF for $(\text{Gly})_{40}$ took nearly 35 hours using 18 cores [the calculation of the exchange matrix scales as $\mathcal{O}(N^4)$]. It is possible, however, to reduce the scaling by local fitting approximations; using the approach of Ref. 130 the CPU time is reduced by a factor of 15. Further improvements, as for example recently described in Ref. 148, should be possible, and the development of a scalable DF-HF implementation is one of our next goals.

D. The effect of domain sizes on correlation, reaction and activation energies

In this section we will present benchmark calculations for three reactions involving medium-size molecules (up to 61 atoms, cf. Fig. 3). Reaction **I** is the last step in the synthesis of androstendione. In reaction **II**, testosterone is derivatized to make it more lipophilic for a longer retention time in the body tissues. Reaction **III** is the hydroxylation of p-hydroxybenzoate by the enzyme *p*-hydroxybenzoate hydroxylase, PHBH). Reactions **I** and **II** have been used for benchmarks of previous PAO-LCCSD(T)-F12 methods in Ref. 95, while reaction **III** has been studied using PAO-LMP2 and PAO-LCCSD(T) methods in Refs. 149,150. In the current paper reaction energies are computed for reactions **I** and **II**, while for reaction **III** the reaction barrier is computed for snapshot 1 of Ref. 150, using the same QM/MM setup as in the previous work (electrostatic embedding with 19233 lattice points). For reactions **I** and **II** environment or solvent effects have not been considered, since here we are merely interested in the convergence of the reaction energies as a function of the PAO, OSV, and PNO domain sizes.

Reactions **I** and **II** have been chosen since in both cases the reactant and product molecules have quite different sizes. This leads to interesting basis set and long-range correlation effects. Reaction **III** is of interest since the correlation effect on the barrier height is huge ($\approx$ 100 kJ/mol). Furthermore, the electronic structure at the transition state is rather different than in the reactants. This has led in the past to difficulties when treating the reaction with PAO-LCCSD(T) methods, since the results were quite sensitive to the choice of the domains, which differed at the two structures.

The previous work has shown that in all three cases the MP2 results are in rather good agreement with LCCSD(T) ones. It has also been found that the effect of the domain approximation is very similar for LMP2 and LCCSD. Therefore, we believe that the PNO-LMP2 results presented here will also be useful for estimating the reliability of PNO-LCCSD(T) methods that are currently under development in our group.

Table IV lists the canonical reference values using basis sets of double-ζ and triple-ζ quality. All results were obtained with density fitting, using the same auxiliary basis sets for canonical and local calculations. The CABS singles correction[60,151] to the HF reference energies is included in all F12 results. These corrections are smaller for the VTZ-F12 basis sets than for the aVTZ ones, This is consistent with the fact that the VTZ-F12 sets yield significantly better Hartree-Fock energies than the aVTZ basis, since the VTZ-F12 *s* and *p* basis sets correspond to the aug-cc-pVQZ ones.[134] Basis set extrapolations of the DF-MP2 correlation energies were carried out using the usual n^{-3} formula[152] with basis sets aVQZ and aV5Z. The Hartree-Fock energies were extrapolated using the Karton-Martin scheme.[153] The MP2-F12 results with the aVTZ basis set agree within less than 0.5 kJ mol^{-1} with the extrapolated CBS values. The F12 results with the VTZ-F12 basis also agree with the CBS estimates within 0.5 kJ mol^{-1}, except for reaction **II**, where the difference amounts to 1 kJ mol^{-1}. Nevertheless, all MP2-F12 results are always closer to the CBS values than the MP2/aV5Z ones. This is remarkable, since the pure MP2 results obtained with the same triple-ζ sets differ significantly among each other and from the CBS limit.

In the case of reaction **I** the MP2/aVTZ reaction energy is about 10 kJ mol^{-1} larger than the CBS limit, i.e. it strongly overshoots the correlation effect. For the aug-cc-pVDZ basis (not shown in Table IV), this effect is even stronger, the overshooting then amounts to 17 kJ mol^{-1}. We believe that this effect is mainly due to the intramolecular basis set superposition error (BSSE). The BSSE favors the large precursor molecule relative to the products, and therefore the MP2 reaction energies converge from above to the extrapolated CBS value (see upper panel of Fig. 11). The opposite is seen for the reaction **II**, where the product molecule is the largest one, see (see lower panel of Fig. 11). As expected, the basis set effects are larger for the correlation energies than for the HF energies.

The remaining PNO-LMP2-F12 calculations in this section were all carried out with the VTZ-F12 basis. No distant pair approximations were applied (these will be investigated

TABLE IV: Canonical reference values. All results are obtained with density fitting. For other details see text.

Method	Reaction I			Reaction II			Reaction III[c]		
	VDZ-F12	VTZ-F12	aVTZ	VDZ-F12	VTZ-F12	aVTZ	VDZ-F12	VTZ-F12	aVTZ
HF	-1.68	-4.94	-2.97	-26.28	-26.82	-27.16	170.73	173.47	173.78
CABS[a]	-3.77	-0.83	-2.61	0.018	0.60	0.90	3.46	0.75	0.28
HF+CABS	-5.45	-5.77	-5.58	-26.27	-26.22	-26.26	174.20	174.22	174.06
MP2	24.44	22.41	29.69	-24.23	-22.98	-24.97	64.12	67.85	67.90
MP2-F12	19.08	18.83	19.19	-19.75	-19.73	-18.92	71.01	70.28	70.72
HF/aVQZ[b]		-5.37			-26.20			173.80	
HF/aV5Z[b]		-5.83			-26.04			174.07	
HF/CBS[45]		-5.90			-26.01			174.11	
MP2/aVQZ[b]		22.42			-21.08			69.37	
MP2/aV5Z[b]		20.48			-19.87			70.00	
MP2/CBS[45]		18.85			-18.75			70.43	

a) CABS Singles correction to the HF energy

b) Basis aug-cc-pVnZ, for Cl aug-cc-pV(n+d)Z, for hydrogen cc-pVnZ .

c) Barrier height excluding the MM correction of 11.72 kJ/mol

in the next section). Unless otherwise noted, the PNOs were generated from fully converged OSV-LMP2 amplitudes.

Table V demonstrates the dependence of the results on the choice of the PAO domains.

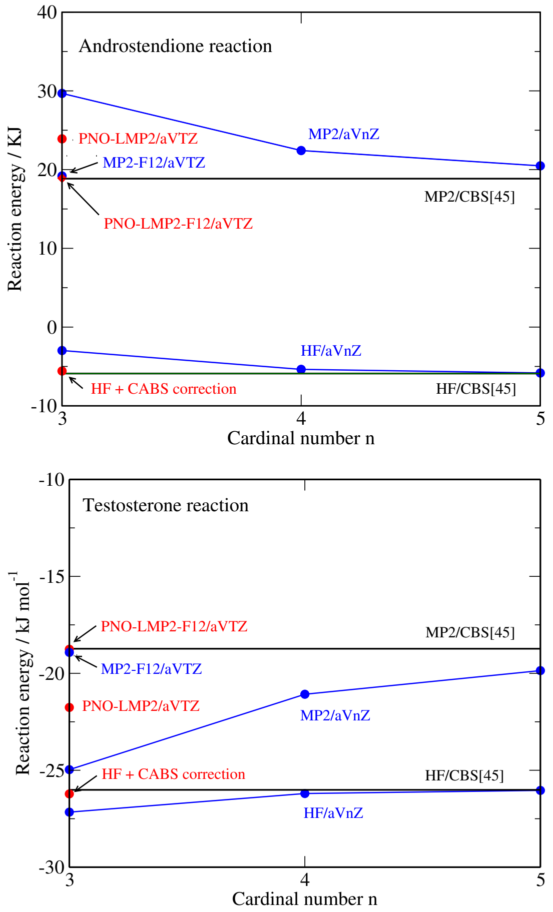

FIG. 11: Basis set dependence of reaction energies for reactions I (upper panel) and II (lower panel) using canonical and local methods. The aVnZ basis set has been used, see text. All calculations employed density fitting.

As outlined in section IV B first primary PAO domains are determined using a threshold T_{LMO} for the IAO partial charges. Secondly, these domains are extended by adding all PAOs at `IEXT` shells of neighboring atoms. We have carried out calculations using T_{LMO} values of 0.2 and 0.1; the domains obtained with these values differ mainly for the aromatic π systems; with the larger value, for each π-bond 3 C-atoms are selected (all partial charges being larger than 0.3), while with the smaller value a 4th atom is added in each case (partial charges close to 0.1). As can be seen, the effect on the results is very small, and we have therefore set the default value to 0.2. All further calculations were carried out with this value. In the case of reaction **III** some bonds are significantly stretched at the transition state and not recognized as ”bonds” any more by the program. The additional distance criterion $R_{\mathrm{ext}} = 5\ a_0$ then had an effect of about -1 kJ mol^{-1} on the barrier height. This criterion does not affect the other reactions, since for normal organic molecules $R_{\mathrm{ext}} = 5\ a_0$ corresponds to `IEXT=2`.

Depending on the PAO domain extensions, the PNO-LMP2 results show variations of up to nearly 4 KJ mol^{-1}, while the PNO-LMP2-F12 values are extremely stable. As already noted, the large differences between MP2/VTZ-F12 and MP2/CBS values is probably due to BSSE effects. It is well known from previous work that the BSSE is strongly reduced by the domain approximation.[154–158] In fact, the PNO-LMP2/VTZ-F12 values for `IEXT=2` are in better agreement with the MP2/CBS ones than the canonical MP2/VTZ-F12 results. Unfortunately, it is not possible to separate the intramolecular BSSE from other correlation effects, and therefore it remains uncertain whether the improvement is an error compensation or really due to the reduction of BSSE in the local case. Another problem is that the BSSE is partly re-introduced when the PAO domains are increased towards the full virtual space by increasing `IEXT`, and there is no clear criterion which tells us where to stop. The results (and probably the BSSE) also depend on the parameters T_{OSV} and T_{PNO}.

Fortunately, all these problems are largely avoided when the F12 terms are included. These reduce the basis set in-

TABLE V: Dependence of PNO-LMP2 and PNO-LMP2-F12 reaction energies on the PAO domain size.[a] AVD is the average PAO orbital domain size of the largest molecule in each reaction. No distant pair approximations are applied.

PAO domain	Reaction I			Reaction II			Reaction III		
extension `IEXT`	AVD	LMP2	LMP2-F12[b]	AVD	LMP2	LMP2-F12[b]	AVD	LMP2	LMP2-F12[b]
Using $T_{\text{LMO}} = 0.2$:									
1	239	16.07	18.09	235	-20.58	-19.48	222	71.18	71.45
2	478	18.31	18.33	474	-20.83	-19.67	446	67.87	70.39
3	751	19.58	18.39	757	-21.24	-19.68	688	68.27	70.36
Using $T_{\text{LMO}} = 0.1$:									
1	243	15.88	18.09	238	-20.63	-19.49	225	72.86	72.07
2	479	18.29	18.34	476	-20.82	-19.74	446	68.23	70.56
3	753	19.60	18.39	758	-20.21	-19.74	688	68.42	70.43
full PAO domains[c]	2006	21.62	18.25	1807	-22.53	-19.55	1855	68.35	70.38
canonical MP2[d]		22.41	18.83		-22.91	-19.73		67.85	70.28

a) Basis VTZ-F12, $T_{\text{OSV}} = 10^{-9}$, $T_{\text{PNO}} = 0.997$

b) The F12 results contain the CABS singles correction, see Table IV.

c) The OSVs are generated from semi-canonical amplitudes T^{ii}_{ab}, cf. eq. (3), without any PAO domain approximation. The PNOs are generated using OSV amplitudes, exactly as in the other cases.

d) The values in the LMP2 and LMP2-F12 columns are the canonical DF-MP2 and DF-MP2-F12 results, respectively, without any local approximations.

completeness errors and therefore also the BSSE. Furthermore, they reduce the domain error, which has been demonstrated in detail earlier for absolute correlation energies as well reaction energies of small[71,75] and large[74,95] molecules, using PAO-LMP2 and PAO-LCCSD methods. The improvement is due to the fact that the F12 contributions approximately account for the correlation energy contributions of the virtual orbitals outside the domains. (In the local case the strong orthogonality projector is pair specific, see Refs. 71,74,75,95 for details). These findings are reinforced for the PNO-LMP2-F12 method by the current results as well as by those in Ref. 23. Using `IEXT=2` convergence of the relative energies to less than 0.2 KJ mol^{-1} is reached, and even with only `IEXT=1` the errors amount less than 1 kJ mol^{-1}. The strongest dependence is seen for the barrier height of reaction **III**, which is not surprising in view of the huge electron correlation effect ($\approx$ 100 kJ mol^{-1}).

In contrast to these results, a recent study of Pavošević et al.[87] found that the reduction of the domain error by F12 terms does not work well if PNOs are employed (using a different F12 approximation). These inconsistent findings are not yet resolved. In this context we have noticed that in small molecules there can be cases where PNOs with near- or exactly zero occupation numbers still have non-negligible contributions to the F12 energy contribution (but not on the conventional PNO-LMP2 energy). In such cases the absolute F12 energies can depend rather sensitively on even small PNO thresholds. In the present study this effect is visible in the HCl and propionyl chloride molecules (see supporting information). The problem is apparently related to (local) symmetry around a given atom, but its precise origin is so far not understood and needs further investigation. Nevertheless, we again find that for relative energies the F12 correction effectively corrects the domain error, even if small molecules are involved where the symmetry problem occurs (c.f. reaction **II**). This is also in line with the results of Ref. 23, in which the reaction energies for 52 reactions of smaller molecules were computed using a simulated PNO-LCCSD-F12 program. The maximum and root mean square deviations from the canonical CCSD-F12 results were only 1.4 and 0.5 kJ mol^{-1}, respectively, using a PNO threshold of 10^{-7}.

The convergence of the relative PNO-LMP2-F12 energies on the thresholds T_{OSV} and T_{PNO} is shown in Table VI. More detailed results, including those for PNO-LMP2, can be found in the supporting information. The convergence with T_{OSV} is found to be rather slow, even with the explicitly correlated methods. This problem has been noted earlier,[23] and it has probably been underestimated in the original OSV papers.[102,103] Reaction energy **I** is most sensitive to this threshold. Reducing the threshold from 10^{-9} to 10^{-10} still affects the reaction energy for **I** by 0.3 kJ mol^{-1}. Nevertheless, we have set the default value to 10^{-9}, which should be accurate enough for most purposes. The average OSV pair domain sizes for this threshold are 267, 263, and 267 for the three reactions, respectively.

Reaction **I** is also most sensitive to the threshold T_{PNO}. The important finding here is that using the occupation number threshold leads to much slower convergence of the reaction energies with the PNO domain size than the alternative method, where as many PNOs are selected as needed to recover a certain fraction (e.g. 0.997) of the OSV pair energies (cf. section IV C). This is particularly the case for the PNO-LMP2 values without F12 correction (see supporting information). The slow convergence seen with the occupation number threshold is due to the fact that the PNO domains become too small (or even zero) for long-range pairs, and then significant dispersion contributions are missing. The energy criterion fixes this problem, and the PNO-LMP2-F12 results are then rather insensitive to the choice of the threshold. The default value for T_{PNO} has been set to 0.997. The average PNO domain sizes for this threshold are 55, 63, and 60 for reactions **I-III**, respectively.

That the improvement of the relative energies by the

TABLE VI: Dependence of PNO-LMP2-F12 reaction energies on type of amplitudes used to compute the PNOs. The values for $T_{\mathrm{PNO}} = 0$ are the OSV-LMP2-F12 results.

		Reaction I			Reaction II			Reaction III		
T_{OSV}	T_{PNO}	OSV	OSV(SC)	PAO(SC)	OSV	OSV(SC)	PAO(SC)	OSV	OSV(SC)	PAO(SC)
10^{-9}	10^{-7}	18.03	16.78	17.21	-19.30	-19.22	-19.46	70.38	71.07	70.89
10^{-9}	10^{-8}	18.40	17.93	18.43	-19.78	-19.62	-19.88	69.87	70.41	70.21
10^{-9}	10^{-9}	18.53	18.32	18.83	-19.78	-19.74	-20.04	70.20	70.41	70.20
10^{-9}	10^{-10}	18.67	18.61	19.15	-19.49	-19.44	-19.75	70.42	70.46	70.24
10^{-9}	0	18.47			-19.64			70.42		
10^{-9}	0.990	18.27	17.77	18.24	-19.67	-19.68	-19.98	69.73	70.36	70.12
10^{-9}	0.995	18.30	17.87	18.38	-19.53	-19.60	-19.91	70.17	70.72	70.68
10^{-9}	0.997	18.33	18.03	18.55	-19.67	-19.80	-20.09	70.39	70.43	70.65
10^{-9}	0.999	18.35	18.27	18.86	-19.63	-19.73	-20.04	70.76	70.54	70.13
10^{-10}	0.990	18.64	18.05	18.26	-19.87	-19.88	-19.99	69.57	70.24	70.12
10^{-10}	0.995	18.65	18.19	18.40	-19.75	-19.81	-19.92	69.99	70.60	70.68
10^{-10}	0.997	18.67	18.35	18.57	-19.89	-20.00	-20.10	70.25	70.28	70.65
10^{-10}	0.999	18.76	18.62	18.87	-19.87	-19.95	-20.05	70.61	70.39	70.13
10^{-10}	0	18.79			-19.86			70.28		
MP2-F12^{c}		18.83			-19.73			70.28		

a) Basis VTZ-F12, IEXT=2; for reaction **III** in addition REXT=5 a_0, see text.

b) All results contain the CABS singles correction, see Table IV.

c) Canonical MP2-F12 results.

F12 contributions is not a fortuitous error compensation is demonstrated by the dependence of the absolute correlation energies on the parameters that determine the domains. In Table VII the percentages of correlation energy recovered by the PNO-LMP2 and PNO-LMP2-F12 methods relative to the corresponding canonical values are listed for the molecules of reaction **I**. Corresponding (and more complete) data for all molecules can be found in the supporting information. As expected, the PNO-LMP2 correlation energies converge towards the canonical limit with increasing OSV and PNO domain sizes. Using the default parameters (IEXT=2, $T_{\mathrm{OSV}} = 10^{-9}$, $T_{\mathrm{PNO}} = 0.997$) 99.6-99.8 % of the canonical correlation energies are recovered. The PNO-LMP2-F12 correlation energies are much closer to the canonical MP2-F12 ones and very insensitive to the domain thresholds; for the default domains they deviate by no more than 0.02 % from the canonical MP2-F12 values. Using tighter thresholds this can be reduced to less than 0.01%. In contrast to the PNO-LMP2 correlation energies they mostly converge from above with increasing domain sizes, which is due to the well-known slight overshooting of the F12/3*A approximation.

It should be pointed out that the F12 calculation is not for free. With our current program, which is not yet fully optimized, a PNO-LMP2-F12 calculation takes 5-6 times more time than a corresponding PNO-LMP2 calculation. For example, the PNO-LMP2 and PNO-LMP2-F12 calculations with the VTZ-F12 basis for the 4 molecules of the Testosterone reaction take 431 and 2154 sec, respectively (on a single node using 18 cores, with local fitting and local RI enabled, excluding Hartree-Fock). This is comparable to an analogous PNO-LMP2 calculation using the VQZ-F12 basis, which takes 1587 sec and yields a reaction energy of -19.32 kJ mol^{-1}. Including the Hartree-Fock time the LMP2/VQZ-F12 calculation is about 1.5 times as expensive as the LMP2-F12/VTZ-F12 one (4817 sec vs. 3214 sec, respectively).

In summary, using the chosen default parameters ($T_{\mathrm{LMO}} = 0.2$, IEXT=2, $T_{\mathrm{OSV}} = 10^{-9}$, $T_{\mathrm{PNO}} = 0.997$, all computed energy differences agree with the canonical MP2-F12 values as well as with the extrapolated MP2/CBS values within 1 kJ mol^{-1} (0.25 kcal mol^{-1}). The deviations from the canonical MP2-F12 values can be reduced to less than 0.3 kJ mol^{-1} by setting $T_{\mathrm{OSV}} = 10^{-10}$ and $T_{\mathrm{PNO}} = 0.998$.

Finally, we note that even with the VDZ-F12 basis set results of excellent accuracy are obtained. Some results are shown in Table VIII. In some of these calculations the energy criterion as well as the occupation number criterion have been used for PNO domain selection, i.e. both must be fulfilled simultaneously (cf. section IV C). This ensures accurate results both for close and distant pairs. Even though the LMP2-F12 correlation energies somewhat overshoot the canonical MP2-F12 values, the relative energies are still very accurate. The absolute truncation errors of the PNO-LMP2-F12 values decrease with increasing PNO-domains; for the tightest thresholds (0.997/1.d-9) the absolute truncation error of the PNO-LMP2-F12 correlation energy for the androstendione precursor (relative to MP2-F12) amounts to only 0.5 mH, while the PNO-LMP2 one (relative to MP2) amounts to about 14 mH.

E. Effect of approximations in the PNO generation

As discussed in section IV A, various approximations can be used in the generation of PNOs. The effect of using semi-canonical (non-iterative) OSV-LMP2 amplitudes [OSV(SC)] rather than the converged ones [OSV(OPT)] to generate the PNOs can be seen by comparing the first two columns for each reaction in Table VI. This effect is rather

TABLE VII: Absolute and relative correlation energies for the molecules in reaction **I** as a function of the domain selection parameters. The VTZ-F12 is used. For the local methods the percentage relative to the corresponding canonical values is given. The canonical values are in hartree (E_h). A value $T_{\text{LMO}} = 0.2$ has been used in all cases. Fully optimized OSV-LMP2 amplitudes have been used to generate the PNOs. The corresponding values obtained if semi-canonical OSV-LMP2 amplitudes are used to generate PNOs, as well as additional data for these and other molecules, can be found in the supporting information.

			PNO-LMP2			PNO-LMP2-F12		
`IEXT`	T_{OSV}	T_{PNO}	Precursor	Androstendione	o-Hydroyphenol	Precursor	Androstendione	o-Hydroyphenol
Reference energies[a]			-1265.293819	-884.712689	-380.583011	-1265.311431	-884.724895	-380.588733
Canonical correlation energies:[b]			-5.245171	-3.757905	-1.476848	-5.687985	-4.071195	-1.607420
1	10^{-9}	0.997	99.6357	99.4098	99.4285	99.9932	100.0227	100.0093
2	10^{-9}	0.997	99.7648	99.6586	99.6594	99.9995	100.0163	100.0082
3	10^{-9}	0.997	99.7978	99.7483	99.7421	100.0001	100.0057	100.0011
2	10^{-10}	0.995	99.6500	99.5705	99.5674	100.0268	100.0631	100.0515
2	10^{-10}	0.997	99.7758	99.6875	99.6875	100.0027	100.0282	100.0199
2	10^{-10}	0.999	99.9008	99.8025	99.8052	99.9831	100.0079	100.0004
2	10^{-10}	10^{-7}	99.7170	99.5281	99.5302	100.0178	100.0407	100.0310
2	10^{-10}	10^{-8}	99.9198	99.7920	99.7965	100.0106	100.0190	100.0159
2	10^{-10}	10^{-9}	99.9563	99.8479	99.8520	99.9927	100.0113	100.0063
2	10^{-10}	10^{-10}	99.9619	99.8583	99.8621	99.9909	100.0064	100.0032

a) The values under PNO-LMP2 are pure DF-HF energies; the ones under PNO-LMP2-F12 include the CABS singles correction.

b) The values under PNO-LMP2 are DF-MP2 correlation energies; the ones under PNO-LMP2-F12 include the F12 correction.

small and systematically decreases with increasing PNO domain sizes. In most cases the convergence with the PNO threshold is somewhat faster when optimized OSV amplitudes are used. This effect is most clearly seen for reaction **I**. Using the default threshold $T_{\text{PNO}} = 0.997$ the differences are smaller or equal to 0.3 kJ mol^{-1}, and we use the semi-canonical methods as the default. As has been shown in section VIII A, this strongly reduces the computational effort.

Results obtained with the PAO(SC) approach, in which the OSV step is skipped, are also shown in Table VI. As expected, the PAO(SC) results converge somewhat faster than the OSV(SC) ones towards the canonical results. The tiny dependence on the threshold T_{OSV} is due to the fact that in our program the PNOs for the diagonal pairs (i.e. the OSVs) are selected with this threshold.

F. Effect of long-range correlations and multipole approximations on reaction energies

In this section we will demonstrate the effect of long-range correlation effects on the reaction energies and investigate the effect of the dipole-dipole approximation for approximating the distant pair energies (cf. section III E). As an example we will discuss reaction **I**; benchmark results for all three reactions can be found in the supporting information. Throughout this section the F12 corrections are included, but these corrections are neglected for the distant pairs.

Fig. 12 shows the effect of long-range correlation effects on the reaction energy of **I** as a function of a distance parameter R_{dist}. The pair energies of pairs for which $R_{ij} > R_{\text{dist}}$ are either neglected or approximated. Here R_{ij} is the minimum distance between any atom in the primary orbital domain $[i]$ from those in domain $[j]$ (using $T_{\text{LMO}} = 0.2$ to select these domains). Completely neglecting the long-range pairs has a strong effect: for $R_{\text{dist}} = 8$ the error amounts to nearly 6 kJ mol^{-1}. Including the distant pairs but keeping only the (exact) integrals $(r^i i|s^j j)$ in the LMP2 iterations almost eliminates the error beyond $R_{\text{dist}} = 9\ a_0$. Even for $R_{\text{dist}} = 8\ a_0$ the error is reduced to about 1 kJ mol^{-1}. Also approximating these integrals by the dipole-dipole approximation has a rather small effect. These results show that the dispersion interaction, which are described by these integrals, plays a dominant role. It stabilizes the large reactant molecule, and therefore makes the reaction energy more positive. If the distant pair energies are treated by the non-iterative dipole-dipole [DIP(SC)] approximation, the errors are somewhat

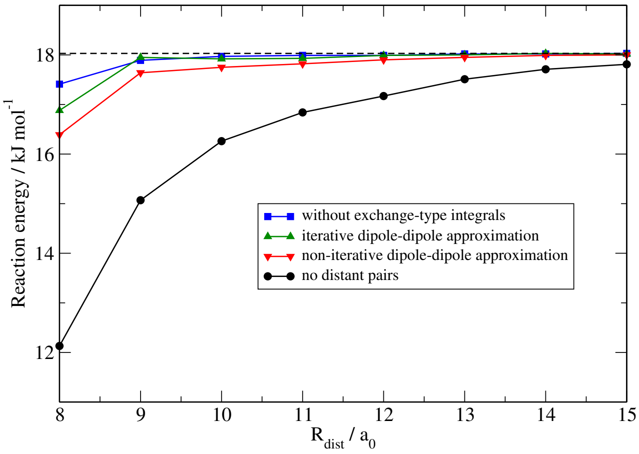

FIG. 12: Effect of long range correlation effects and dipole-dipole approximations on the PNO-LMP2-F12 reaction energy **I** as a function of the distance threshold R_{dist} (see text). The dashed line is the result obtained with all pairs. Defaults have been used for all other parameters (see footnote of Table IX).

TABLE VIII: Reaction energies and correlation energies using the VDZ-F12 basis set. The PNOs have been computed using optimized OSV amplitudes. $T_{\mathrm{LMO}=0.2}$, `IEXT=2`, $T_{\mathrm{OSV}} = 10^{-9}$, $T_{\mathrm{PNO}} = 0.997$. $T_{\mathrm{PNO}}^{\mathrm{occ}}$ is an occupation number threshold for the PNO selection. Where given, this be fulfilled in addition to the energy threshold T_{PNO}.

Method	$T_{\mathrm{PNO}}^{\mathrm{occ}}$	Reaction **I**	Reaction **II**	Reaction **III**
Reaction energies (kJ mol^{-1}):				
PNO-LMP2		17.82	-20.01	63.91
PNO-LMP2	10^{-8}	17.53	-19.95	63.70
PNO-LMP2	10^{-9}	17.72	-20.00	63.58
MP2		24.44	-24.23	64.12
PNO-LMP2-F12		18.41	-19.34	71.29
PNO-LMP2-F12	10^{-8}	18.86	-19.69	70.97
PNO-LMP2-F12	10^{-9}	18.79	-19.50	71.19
MP2-F12		19.08	-19.75	71.01
MP2/CBS[45]		18.85	-18.75	70.43
Correlation energies:[a]				
MP2		-4.709626	-4.085582	-5.248180
PNO-LM2		99.5297	99.5123	99.6060
PNO-LMP2	10^{-8}	99.6841	99.6635	99.7710
PNO-LMP2	10^{-9}	99.7038	99.6828	99.7882
MP2-F12		-5.699752	-4.937962	-6.399285
PNO-LMP2-F12		100.1326	100.1273	100.1163
PNO-LMP2-F12	10^{-8}	100.0392	100.0387	100.0137
PNO-LMP2-F12	10^{-9}	100.0086	100.0085	99.9910

a) For the largest molecules of **I** and **II**, and for the reactant of **III**. MP2 and MP2-F12 values in E_h, PNO-LMP2 values in percent.

larger, but quickly converge to negligible values at long distances. For R_{dist} values of 8, 10, and 12 the additional errors caused by the DIP(SC) treatment amount to 0.5, 0.2, and 0.1 kJ mol^{-1}, respectively. The total error of the DIP(SC) method for $R_{\mathrm{dist}} = 12$ is only 0.1 kJ mol^{-1}, while complete neglect of the pairs beyond this distance causes an error of 0.9 kJ mol^{-1}.

An alternative method to select the distant pairs is to use their pair energies as a criterion. The pair energies are then computed for all pairs for which the distance R_{ij} between the charge centers of the two orbital domains is nonzero (or above some minimum value), and then all pairs for which $E_{ij} < T_{\mathrm{dist}}$ are selected as distant. Table IX shows the number of distant pairs (for the largest molecule of **I**, **II**, and for the reactant of **III**) and the convergence of the reaction energies with this threshold. The numbers of distant pairs obtained with $T_{\mathrm{dist}} = 10^{-6}$ correspond to those for $R_{dist} = 11 - 12\,a_0$, and the errors with this threshold are between 0.2 and 0.4 kJ mol^{-1}. Thus, this threshold appears to be appropriate for most cases. A safer choice is $3 \cdot 10^{-7}$, but this reduces the number of distant pairs approximately by a factor of 2 and therefore makes the calculations more expensive.

TABLE IX: PNO-LMP2-F12 energy differences as a function of the energy criterion T_{dist} for the selection of distant pairs. The DIP(SC) approximation[a] has been used, other parameters are default values[a].

	Reaction I		Reaction II		Reaction III	
T_{dist}	N_{dist}	ΔE	N_{dist}	ΔE	N_{dist}	ΔE
0	0	18.03	0	-19.80	0	70.43
$1 \cdot 10^{-7}$	201	18.03	209	-19.80	113	70.43
$3 \cdot 10^{-7}$	522	17.99	384	-19.78	432	70.45
$1 \cdot 10^{-6}$	941	17.90	634	-19.62	931	70.81
$3 \cdot 10^{-6}$	1301	17.34	891	-19.12	1438	72.14

a) $T_{\mathrm{LMO}} = 0.2$, `IEXT=2`, $T_{\mathrm{OSV}} = 10^{-9}$, $T_{\mathrm{PNO}} = 0.997$ The PNOs were generated from OSV(SC) amplitudes.

IX. SUMMARY AND CONCLUSIONS

As a first step towards the development of new scalable coupled-cluster methods we have demonstrated that almost perfect linear scaling of the elapsed-time and memory requirements with the molecular size can be achieved for PNO-LMP2, without significant loss of accuracy as compared to canonical MP2. If combined with explicitly correlated terms, we have shown that extrapolated MP2/CBS limits for reaction energies or barrier heights can be reproduced by PNO-LMP2-F12 with triple-ζ basis sets within less than 1 kJ mol^{-1} error. The accuracy of the method can be controlled by very few parameters, and based on extensive benchmark calculations reliable default values for these have been established. It has been demonstrated that long-range dispersion interactions can have a significant effect on reaction energies involving large molecules. These long-range correlation effects can be well approximated by multipole expansions.

Our new implementation is well parallelized, and good speedups with 100 and more processing cores have been achieved on a compute cluster with Infiniband network, even for calculations that require only a few minutes of elapsed time. Due to successive transformations of the canonical virtual orbitals via PAOs and OSVs to PNOs, an enormous reduction of the amount of data (integrals, amplitudes) is achieved, which makes it possible to keep all data in the distributed memory. Disk I/O is entirely avoided, apart from reading the LMOs and PAOs. This is essential for efficient use of massively parallel machines. The memory requirements also decrease almost linearly with the number of processors used, which means the system size that can be treated within a certain time and a fixed amount of memory per processor increases almost linearly with the number of processors. Currently this is still limited by the replicated storage of the full overlap and Fock matrices in the PAO basis as well as non-perfect load balancing, but it should be possible to eliminate these problems in the future.

The next steps will be the implementation of a scalable F12 correction as well as an PNO-LCCSD program. At the time of writing this paper, PNO-LCCD is already working, using very similar techniques as presented here. With this new program we have already been able to carry out calculations for 3-dimensional molecules with over 100 atoms and more than 2000 basis functions on one compute node in less than 1 hour of elapsed time. The details and benchmarks will be presented in a another publication.

At least with our current program, the bottleneck in very large calculations is the preceding Hartree-Fock calculation, in particular the evaluation of the exchange part of the Fock matrix. This applies both to CPU-time and memory. It was shown previously[130,148] that this problem can be alleviated using local density fitting approximations. Several of the improved techniques presented in this paper should be equally applicable in Hartree-Fock, and the development of a scalable HF implementation is another important goal for the near future.

Acknowledgements

This work has been funded by the ERC Advanced Grant 320723 (ASES).

Supporting information

Supporting information is available and contains: (a) Detailed tables of individual Hartree-Fock and correlation energies for the reactions described in the main text (using various basis sets, parameters and thresholds), (b) Tables illustrating the dependence of the reaction energies on the OSV and PNO thresholds, (c) Geometries of all used molecules. This information is available free of charge via the Internet at http://pubs.acs.org/.

[1] P. Pulay, Chem. Phys. Lett. **100**, 151 (1983).

[2] S. Saebø and P. Pulay, Chem. Phys. Lett. **113**, 13 (1985).

[3] P. Pulay and S. Saebø, Theor. Chim. Acta **69**, 357 (1986).

[4] S. Saebø and P. Pulay, J. Chem. Phys. **86**, 914 (1987).

[5] S. Saebø and P. Pulay, J. Chem. Phys. **88**, 1884 (1988).

[6] C. Hampel and H.-J. Werner, J. Chem. Phys. **104**, 6286 (1996).

[7] G. Hetzer, P. Pulay, and H.-J. Werner, Chem. Phys. Lett. **290**, 143 (1998).

[8] G. Hetzer, M. Schütz, H. Stoll, and H.-J. Werner, J. Chem. Phys. **113**, 9443 (2000).

[9] M. Schütz, G. Hetzer, and H.-J. Werner, J. Chem. Phys. **111**, 5691 (1999).

[10] M. Schütz and H.-J. Werner, Chem. Phys. Lett. **318**, 370 (2000).

[11] M. Schütz, J. Chem. Phys. **113**, 9986 (2000).

[12] M. Schütz and H.-J. Werner, J. Chem. Phys. **114**, 661 (2001).

[13] M. Schütz, J. Chem. Phys. **116**, 8772 (2002).

[14] M. Schütz, Phys. Chem. Chem. Phys. **4**, 3941 (2002).

[15] H.-J. Werner, F. R. Manby, and P. J. Knowles, J. Chem. Phys. **118**, 8149 (2003).

[16] M. Schütz, H.-J. Werner, R. Lindh, and F. R. Manby, J. Chem. Phys. **121**, 737 (2004).

[17] M. Schütz and F. R. Manby, Phys. Chem. Chem. Phys. **5**, 3349 (2003).

[18] H.-J. Werner and M. Schütz, J. Chem. Phys. **135**, 144116 (2011).

[19] H.-J. Werner and K. Pflüger, Ann. Reports in Comput. Chem. **2**, 53 (2006).

[20] R. Mata and H.-J. Werner, J. Chem. Phys. **125**, 184110 (2006).

[21] R. Mata and H.-J. Werner, Mol. Phys. **105**, 2753 (2007).

[22] R. Mata, H.-J. Werner, and M. Schütz, J. Chem. Phys. **128**, 144106 (2008).

[23] C. Krause and H.-J. Werner, Phys. Chem. Chem. Phys. **14**, 7591 (2012).

[24] D. Kats and M. Schütz, J. Chem. Phys. **131**, 124117 (2009).

[25] K. Freundorfer, D. Kats, T. Korona, and M. Schütz, J. Chem. Phys. **133**, 244110 (2010).

[26] P. E. Maslen and M. Head-Gordon, Chem. Phys. Lett. **283**, 102 (1998).

[27] P. E. Maslen and M. Head-Gordon, J. Chem. Phys. **109**, 7093 (1998).

[28] P. Y. Ayala and G. E. Scuseria, J. Chem. Phys. **110**, 3660 (1999).

[29] G. E. Scuseria and P. Y. Ayala, J. Chem. Phys. **111**, 8330 (1999).

[30] P. E. Maslen, A. Dutoi, M. S. Lee, Y. H. Shao, and M. Head-Gordon, Mol. Phys. **103**, 425 (2005).

[31] N. J. Russ and T. D. Crawford, Chem. Phys. Lett. **400**, 104 (2004).

[32] R. A. DiStasio, Y. S. Jung, and M. Head-Gordon, J. Chem. Theory Comput. **1**, 862 (2005).

[33] J. E. Subotnik and M. Head-Gordon, J. Chem. Phys. **123**, 064108 (2005).

[34] A. Auer and M. Nooijen, J. Chem. Phys. **125**, 024104 (2006).

[35] K. V. Lawler, J. A. Parkhill, and M. Head-Gordon, Mol. Phys. **106**, 2309 (2008).

[36] T. S. Chwee, A. B. Szilva, R. Lindh, and E. A. Carter, J. Chem. Phys. **128**, 224106 (2008).

[37] F. Neese, F. Wennmohs, and A. Hansen, J. Chem. Phys. **130**, 114108 (2009).

[38] F. Neese, A. Hansen, and D. G. Liakos, J. Chem. Phys. **131**, 064103 (2009).

[39] A. Hansen, D. G. Liakos, and F. Neese, J. Chem. Phys. **135**, 214102 (2011).

[40] D. G. Liakos, A. Hansen, and F. Neese, J. Chem. Theory Comput. **7**, 76 (2011).

[41] R. Izsak, A. Hansen, and F. Neese, Mol. Phys. **110**, 2413 (2012).

[42] C. Riplinger and F. Neese, J. Chem. Phys. **138**, 034106 (2013).

[43] C. Riplinger, B. Sandhoefer, A. Hansen, and F. Neese, J. Chem. Phys. **139**, 134101 (2013).

[44] C. Hättig, D. P. Tew, and B. Helmich, J. Chem. Phys. **136**, 204105 (2012).

[45] W. Klopper and W. Kutzelnigg, Chem. Phys. Lett. **134**, 17 (1987).

[46] W. Kutzelnigg and W. Klopper, J. Chem. Phys. **94**, 1985 (1991).

[47] W. Klopper and C. C. M. Samson, J. Chem. Phys. **116**, 6397 (2002).

[48] F. R. Manby, J. Chem. Phys. **119**, 4607 (2003).

[49] S. Ten-no, Chem. Phys. Lett. **398**, 56 (2004).

[50] S. Ten-no, J. Chem. Phys. **121**, 117 (2004).

[51] A. J. May and F. R. Manby, J. Chem. Phys. **121**, 4479 (2004).

[52] E. F. Valeev, Chem. Phys. Lett. **395**, 190 (2004).

[53] D. P. Tew and W. Klopper, J. Chem. Phys. **123**, 074101 (2005).

[54] S. Kedžuch, M. Milko, and J. Noga, Int. J. Quantum Chem. **105**, 929 (2005).

[55] H. Fliegl, W. Klopper, and C. Hättig, J. Chem. Phys. **122**, 084107 (2005).

[56] H. Fliegl, C. Hättig, and W. Klopper, Int. J. Quantum Chem. **106**, 2306 (2006).

[57] F. R. Manby, H.-J. Werner, T. B. Adler, and A. J. May, J. Chem. Phys. **124**, 094103 (2006).

[58] H.-J. Werner, T. B. Adler, and F. R. Manby, J. Chem. Phys. **126**, 164102 (2007).

[59] J. Noga, S. Kedžuch, and J. Šimunek, J. Chem. Phys. **127**, 034106 (2007).

[60] T. B. Adler, G. Knizia, and H.-J. Werner, J. Chem. Phys. **127**,

221106 (2007).

[61] D. P. Tew, W. Klopper, C. Neiss, and C. Hättig, Phys. Chem. Chem. Phys. **9**, 1921 (2007).

[62] G. Knizia and H.-J. Werner, J. Chem. Phys. **128**, 154103 (2008).

[63] T. Shiozaki, M. Kamiya, S. Hirata, and E. F. Valeev, J. Chem. Phys. **129**, 071101 (2008).

[64] T. Shiozaki, M. Kamiya, S. Hirata, and E. F. Valeev, Phys. Chem. Chem. Phys. **10**, 3358 (2008).

[65] J. Noga, S. Kedžuch, J. Šimunek, and S. Ten-no, J. Chem. Phys. **128**, 174103 (2008).

[66] D. P. Tew, W. Klopper, and C. Hättig, Chem. Phys. Lett. **452**, 326 (2008).

[67] E. F. Valeev, Phys. Chem. Chem. Phys. **10**, 106 (2008).

[68] E. F. Valeev and T. D. Crawford, J. Chem. Phys. **128**, 244113 (2008).

[69] M. Torheyden and E. F. Valeev, Phys. Chem. Chem. Phys. **10**, 3410 (2008).

[70] D. Bokhan, S. Ten-no, and J. Noga, Phys. Chem. Chem. Phys. **10**, 3320 (2008).

[71] H.-J. Werner, J. Chem. Phys. **129**, 101103 (2008).

[72] G. Knizia, T. B. Adler, and H.-J. Werner, J. Chem. Phys. **130**, 054104 (2009).

[73] T. Shiozaki, M. Kamiya, S. Hirata, and E. F. Valeev, J. Chem. Phys. **130**, 054101 (2009).

[74] T. B. Adler, H.-J. Werner, and F. R. Manby, J. Chem. Phys. **130**, 054106 (2009).

[75] T. B. Adler and H.-J. Werner, J. Chem. Phys. **130**, 241101 (2009).

[76] D. Bokhan, S. Bernadotte, and S. Ten-no, Chem. Phys. Lett. **469**, 214 (2009).

[77] C. Hättig, D. P. Tew, and A. Köhn, J. Chem. Phys. **132**, 231102 (2010).

[78] H.-J. Werner, G. Knizia, and F. R. Manby, Mol. Phys. **109**, 407 (2011).

[79] D. P. Tew, B. Helmich, and C. Hättig, J. Chem. Phys. **135**, 074107 (2011).

[80] C. Hättig, D. P. Tew, and B. Helmich, J. Chem. Phys. **136**, 204105 (2012).

[81] G. Schmitz, B. Helmich, and C. Hättig, Mol. Phys. **111**, 2463 (2013).

[82] D. P. Tew and C. Hättig, Int. J. Quantum Chem. **113**, 224 (2013).

[83] D. S. Hollman, J. J. Wilke, and H. F. Schaefer, J. Chem. Phys. **138**, 064107 (2013).

[84] B. Doser, D. S. Lambrecht, J. Kussmann, and C. Ochsenfeld, J. Chem. Phys. **130**, 064107 (2009).

[85] B. Doser, J. Zienau, L. Clin, D. S. Lambrecht, and C. Ochsenfeld, Z. Phys. Chem. **224**, 397 (2010).

[86] S. A. Maurer, D. S. Lambrecht, J. Kussmann, and C. Ochsenfeld, J. Chem. Phys. **138**, 014101 (2013).

[87] F. Pavošević, F. Neese, and E. F. Valeev, J. Chem. Phys. **141**, 054106 (2014).

[88] B. Helmich and C. Hättig, J. Chem. Phys. **135**, 214106 (2011).

[89] B. Helmich and C. Hättig, J. Chem. Phys. **139**, 084114 (2013).

[90] B. Helmich and C. Hättig, Comput. Theor. Chem. **1040–1041**, 35 (2014).

[91] L. Maschio, J. Chem. Theory Comput. **7**, 2818 (2011).

[92] M. Del Ben, J. Hutter, and J. VandeVondele, J. Chem. Theory Comput. **9**, 2654 (2013).

[93] D. Usvyat, L. Maschio, C. Pisani, and M. Schütz, Z. Phys. Chem. **224**, 441 (2010).

[94] D. Usvyat, J. Chem. Phys. **139**, 194101 (2013).

[95] T. B. Adler and H.-J. Werner, J. Chem. Phys. **135**, 144117 (2011).

[96] G. Schmitz, C. Hättig, and D. P. Tew, Phys. Chem. Chem. Phys. **16**, 22167 (2014).

[97] C. Edmiston and M. Krauss, J. Chem. Phys. **42**, 1119 (1965).

[98] W. Meyer, Int. J. Quantum Chem. **5**, 341 (1971).

[99] W. Meyer, J. Chem. Phys. **58**, 1017 (1973).

[100] R. Ahlrichs, F. Driessler, H. Lischka, V. Staemmler, and W. Kutzelnigg, J. Chem. Phys. **62**, 1235 (1975).

[101] V. Staemmler and R. Jaquet, Theor. Chim. Acta **59**, 487 (1981).

[102] J. Yang, Y. Kurashige, F. R. Manby, and G. K. L. Chan, J. Chem. Phys. **134**, 044123 (2011).

[103] J. Yang, F. R. Manby, G. K. L. Chan, M. Schütz, and H.-J. Werner, J. Chem. Phys. **136**, 144105 (2012).

[104] Y. Kurashige, J. Yang, G. K. L. Chan, and F. R. Manby, J. Chem. Phys. **136** (2012).

[105] M. Schütz, J. Yang, G. K.-L. Chan, F. R. Manby, and H.-J. Werner, J. Chem. Phys. **138**, 054109 (2013).

[106] D. G. Fedorov and K. Kitaura, J. Chem. Phys. **123**, 134103 (2005).

[107] T. F. Hughes, N. Flocke, and R. J. Bartlett, J. Phys. Chem. A **112**, 5994 (2008).

[108] W. Li, P. Piecuch, J. R. Gour, and S. Li, J. Chem. Phys. **131**, 114109 (2009).

[109] W. Li and P. Piecuch, J. Phys. Chem. A **114**, 8644 (2010).

[110] W. Li and P. Piecuch, J. Phys. Chem. A **114**, 6721 (2010).

[111] W. Li, Y. Guo, and S. Li, Phys. Chem. Chem. Phys. **14**, 7854 (2012).

[112] M. Ziółkowski, B. Jansìk, T. Kjærgaard, and P. Jørgensen, J. Chem. Phys. **133**, 014107 (2010).

[113] K. Kristensen, M. Ziółkowski, B. Jansìk, T. Kjærgaard, and P. Jørgensen, J. Chem. Theory Comput. **7**, 1677 (2011).

[114] K. Kristensen, P. Jørgensen, B. Jansìk, T. Kjærgaard, and S. Reine, J. Chem. Phys. **137**, 114102 (2012).

[115] I.-M. Høyvik, K. Kristensen, B. Jansìk, and P. Jørgensen, J. Chem. Phys. **136**, 014105 (2012).

[116] K. Kristensen, T. Kjærgaard, I.-M. Høyvik, P. Ettenhuber, P. Jørgensen, B. Jansik, S. Reine, and J. Jakowski, Mol. Phys. **111**, 1196 (2013).

[117] H. Stoll, J. Chem. Phys. **97**, 8449 (1992).

[118] J. Friedrich, M. Hanrath, and M. Dolg, J. Chem. Phys. **126**, 154110 (2007).

[119] J. Friedrich and M. Dolg, J. Chem. Phys. **129**, 244105 (2008).

[120] J. Friedrich, S. Coriani, T. Helgaker, and M. Dolg, J. Chem. Phys. **131**, 154102 (2009).

[121] J. Friedrich and M. Dolg, J. Chem. Theory Comput. **5**, 287 (2009).

[122] S. F. Boys, in *Quantum Theory of Atoms, Molecules, and the Solid State*, edited by P. O. Löwdin (Academic Press, New York, 1966), p. 253.

[123] J. Pipek and P. G. Mezey, J. Chem. Phys. **90**, 4916 (1989).

[124] G. Knizia, J. Chem. Theory Comput. **9**, 4834 (2013).

[125] W. C. Lu, C. Z. Wang, M. W. Schmidt, L. Bytautas, K. M. Ho, and K. Ruedenberg, J. Chem. Phys. **120**, 2629 (2004).

[126] A. C. West, M. W. Schmidt, M. S. Gordon, and K. Ruedenberg, J. Chem. Phys. **139**, 234107 (2013).

[127] T. Janowski, J. Chem. Theory Comput. **10**, 3085 (2014).

[128] S. Lehtola and H. Jónsson, J. Chem. Theory Comput. **9**, 5365 (2013).

[129] J. W. Boughton and P. Pulay, J. Comp. Chem. **14**, 736 (1993).

[130] R. Polly, H.-J. Werner, F. R. Manby, and P. J. Knowles, Mol. Phys. **102**, 2311 (2004).

[131] J. Nieplocha, B. Palmer, V. Tipparaju, M. Krishnan, H. Trease, and E. Aprà, International Journal of High Performance Computing Applications **20**, 203 (2006).

[132] G. Karypis and V. Kumar, SIAM J. Sci. Comput. **20**, 359 (1998).

[133] E. F. Valeev and C. L. Janssen, J. Chem. Phys. **121**, 1214 (2004).

[134] K. A. Peterson, T. B. Adler, and H.-J. Werner, J. Chem. Phys. **128**, 084102 (2008).

[135] K. E. Yousaf and K. A. Peterson, J. Chem. Phys. **129**, 184108 (2008).

[136] R. A. Kendall, T. H. Dunning, and R. J. Harrison, J. Chem. Phys. **96**, 6796 (1992).

[137] T. Dunning, Jr., K. A. Peterson, and A. Wilson, J. Chem. Phys. **114**, 9244 (2001).

[138] T. H. Dunning, Jr., J. Chem. Phys. **90**, 1007 (1989).

[139] F. Weigend, Phys. Chem. Chem. Phys. **4**, 4285 (2002).

[140] F. Weigend, A. Köhn, and C. Hättig, J. Chem. Phys. **116**, 3175 (2002).

[141] H.-J. Werner, T. B. Adler, G. Knizia, and F. R. Manby, in *Recent Progress in Coupled Cluster Methods*, edited by P. Čársky, J. Paldus, and J. Pittner (Springer, Dordrecht, Heidelberg, London, New York, 2010), p. 573.

[142] H.-J. Werner, G. Knizia, T. B. Adler, and O. Marchetti, Z. Phys. Chem. **224**, 493 (2010).

[143] H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, M. Schütz, and . and others, *Molpro, development version 2014.2, a package of ab initio programs* (2014), see http://www.molpro.net.

[144] H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, and M. Schütz, WIREs Comput. Mol. Sci. **2**, 242 (2011).

[145] A. Fedorov, L. Batiste, E. P. A. Couzijn, and P. Chen, Chem. Phys. Chem. **11**, 1002 (2010).

[146] S. J. Zuend and E. N. Jacobsen, J. Am. Chem. Soc. **131**, 15358 (2009).

[147] L. Ratjen, M. van Gemmeren, F. Pesciaioli, and B. List, Angew. Chem. **126**, 8910 (2014).

[148] D. Mejía-Rodríguez and A. M. Köster, J. Chem. Phys. **141**, 124114 (2014).

[149] F. Claeyssens, J. N. Harvey, F. R. Manby, R. A. Mata, A. J. Mulholland, K. E. Ranaghan, M. Schütz, S. Thiel, W. Thiel, and H.-J. Werner, Angew. Chem. **118**, 7010 (2006).

[150] R. A. Mata, H.-J. Werner, S. Thiel, and W. Thiel, J. Chem. Phys. **128**, 025104 (2008).

[151] G. Knizia and H.-J. Werner, J. Chem. Phys. **128**, 154103 (2008).

[152] T. Helgaker, W. Klopper, H. Koch, and J. Noga, J. Chem. Phys. **106**, 9639 (1997).

[153] A. Karton and J. Martin, Theor. Chem. Acc. **115**, 330 (2006).

[154] S. Saebø, W. Tong, and P. Pulay, J. Chem. Phys. **98**, 2170 (1993).

[155] M. Schütz, G. Rauhut, and H.-J. Werner, J. Phys. Chem. A **102**, 5997 (1998).

[156] G. Rauhut, A. E. Azhary, F. Eckert, U. Schumann, and H.-J. Werner, Spectrochim. Acta **A 55**, 647 (1999).

[157] G. Rauhut and H.-J. Werner, Phys. Chem. Chem. Phys. **5**, 2001 (2003).

[158] T. Hrenar, G. Rauhut, and H.-J. Werner, J. Phys. Chem. A **110**, 2060 (2006).

[159] For simplicity we will not distinguish between “processors” and “cores” and consider each core as a processor.