JCTC Journal of Chemical Theory and Computation

Weak Molecular Interactions Studied with Parallel Implementations of the Local Pair Natural Orbital Coupled Pair and Coupled Cluster Methods

Dimitrios G. Liakos, Andreas Hansen, and Frank Neese*

Lehrstuhl für Theoretische Chemie, Universität Bonn, Wegelerstrasse 12, D-53115 Bonn, Germany

Received August 9, 2010

Abstract: A parallel implementation of the recently developed local pair natural orbital coupled electron pair approximation (LPNO-CEPA/*n*, $n =$ Version 1, 2, or 3) and the corresponding LPNO coupled cluster method with single- and double excitations (LPNO−CCSD) is described. A detailed analysis alongside pseudocode is presented for the most important computational steps. The scaling with respect to the number of processors is reasonable and speedups of about 10 with 14 processors have been found in benchmark calculations (wall-clock time). The most important factor limiting the efficiency of the scaling with respect to the number of processors is probably the limited bandwidth of the presently prevailing multicore machines. The parallel LPNO methods were applied to study weak intermolecular interactions. Initially, the well-established S22 set of molecules was studied. The mean absolute error resulting from the use of the LPNO-CEPA/1 method relative to the most recent CCSD(T) reference data is found to be 0.24 kcal/mol. Thus, LPNO-CEPA/1 holds great promise for the efficient ab initio treatment of weak intermolecular interactions. In order to demonstrate the applicability of the methods to real systems, a two-dimensional potential energy surface for a trimer of 2,4-dihydroxy-3-acetyl-6-methyl acetophenone [$C_{11}H_{12}O_4$] (81 atoms, 1296 basis functions, 133 single points) has been calculated with the LPNO-CEPA/1 method. In this system, a clear distinction can be made between hydrogen bonding and $\pi-\pi$ interactions. The global minimum on the PES obtained from the calculations agrees excellently with the experimentally determined crystal structure. By contrast, popular density functional methods show no discernible minimum.

Introduction

The ability to accurately and efficiently calculate electron correlation effects has been one of the major goals of electronic structure theory during the last decades. Density functional theory (DFT) represents the most popular family of methods in this respect.[1,2] Owing to their excellent price/performance ratio, DFT methods have become the standard tools for the calculation of molecular properties in most chemical applications. However, DFT methods do not represent a systematic hierarchy that converges to the exact solution of the Schrödinger equation, frequently contain empirical parameters, often are constructed with respect to arbitrarily chosen reference systems such as the inhomogeneous electron gas, and incorporate electron correlation effects in a somewhat ad hoc fashion. In particular, DFT methods do not incorporate dispersion effects and empirical corrections are required in order to obtain reasonable results for weakly bound systems (however, see recent efforts to incorporate the van der Waals interaction empirically[3−7] or by models based on physical reasoning[8−13]).

* Corresponding author e-mail: neese@thch.uni-bonn.de.

10.1021/ct100445s © 2011 American Chemical Society
Published on Web 12/10/2010

An arguably more systematic route toward the incorporation of electron correlation effects is offered by wave function based ab initio methods, all of which take the Hartree−Fock (HF) determinant as the starting point. It has been realized that many body perturbation theory (MBPT) is often not robust enough for obtaining accurate results and only the lowest order correlation method, second order many body perturbation theory with Möller-Plesset partitioning (MP2) has survived as a widely used method owing to its comparatively small computational cost (for recent developments see refs 14−22). For accurate results, however, it is important to use iterative methods. The most popular family is based on the coupled cluster (CC) expansion that is size consistent, unitarily invariant, and converges reasonably quickly toward the full CI limit.[23−25] It is well-known that for obtaining accurate results, it is necessary to include connected triple excitations at least in a perturbative way, thus leading to the "gold" standard CCSD(T) method.[26−28] However, this method features seventh order scaling with respect to molecular size and is hence restricted to small molecules. An intermediate approach that lacks the rigor of CC theory but has been proven to lead to usefully accurate results (intermediate between CCSD and CCSD(T))[29] is the coupled electron pair approximation (CEPA[30−34]) which may be viewed as a simple size consistent modification of the configuration interaction methods with single- and double-excitations (CISD). CEPA is of slightly lower computational cost than CCSD but is usually more accurate.[29]

The most expensive steps in these methods involve multiple nested summations over the entire virtual orbital space. Many efforts have been done aiming in the development of low-order scaling correlation methods, like the ones from Werner and co-workers,[22,35−37] Head-Gordon and co-workers,[38−40] Ayala and Scuseria,[41,42] Carter and co-workers,[43,44] Auer and Nooijen,[45] and others. In this spirit, we have recently revived the method of pair natural orbitals (PNOs[30,31,33,34,46,47]) that greatly reduces the computational cost of these contractions and make the associated computational cost asymptotically linear scaling with respect to molecular size.[48,49] The PNO expansion can be viewed as a means of greatly compacting the information content contained in the virtual space thus leading to very compact correlated wave functions.[48] This is particularly important in combination with the large basis sets that are known to be required for obtaining accurate results in correlated ab initio calculations. In combination with localized internal orbitals and an electron pair screening procedure, the correlated singles- and doubles-wave function is expanded in a linear scaling number of wave function parameters. Our actual realization of the method in the framework of the CEPA and CCSD methods is not linearly scaling but rather retains some (small) fifth order steps.[48,49] Nevertheless, the methods have been proven to be efficient, reliable, of black box character and to recover typically 99.9% of the canonical correlation energy. In fact, some test calculations on molecules in the range of 40 to 80 atoms revealed that the actual time required for the correlation calculation was only two to four times larger than that taken by the proceeding Hartree−Fock calculations.[49] Thus, a very large range of chemical applications can be envisioned.

In this work, the applicability of the LPNO methods is further enhanced through the development of a parallel code. Furthermore, we demonstrate that the LPNO-CEPA/1 method provides accurate results for weak intermolecular interactions.

Theory

The underlying theory of the LPNO-CEPA and LPNO−CCSD methods is described in detail elsewhere.[48,49] Here, we will only briefly describe the working equations in order to assist the understanding of the parallel program version.

Working Equations. Starting from the closed-shell CISD wave function and using the generator state formalism,[50] the CISD wave function is written as follows:

$$\Psi = \Psi_{\mathrm{HF}} + \sum_{ia} C_a^i \Psi_i^a + \sum_{i \le j} \sum_{ab} C_{ab}^{ij} \Psi_{ij}^{ab} \tag{1}$$

with *i,j,k,l* referring to occupied orbitals and *a,b,c,d* to unoccupied ones. The residual for the double excitation amplitudes is as follows:

$$\sigma_{ab}^{ij} = \langle \tilde{\Psi}_{ij}^{ab} | H - E_0 - \Delta^{ij} | \Psi \rangle \tag{2}$$

where Δ^{ij} is a method specific shift and $\tilde{\Psi}_{ij}^{ab}$denotes the contravariant configuration state function.[50] As derived in detailed elsewhere the working equation for the amplitudes in the PNO basis (the PNO basis is indicated by an overbar) becomes:

$$\begin{aligned}
\sigma^{ij}_{\bar{a}^{ij}\bar{b}^{ij}} = {} & K^{ij}_{\bar{a}^{ij}\bar{b}^{ij}} + K(\bar{\mathrm{C}}^{ij})_{\bar{a}^{ij}\bar{b}^{ij}} + \{\mathbf{d}^{ij\dagger}\mathbf{F}^{\mathbf{V}}\mathbf{d}^{ij}\bar{\mathrm{C}}^{ij} + \bar{\mathrm{C}}^{ij}\mathbf{d}^{ij\dagger}\mathbf{F}^{V}\mathbf{d}^{ij}\}_{\bar{a}^{ij}\bar{b}^{ij}} \\
& - \sum_{kl} (ik|jl)(\mathbf{S}^{ij,kl}\bar{\mathrm{C}}^{kl}\mathbf{S}^{ij,kl\dagger})_{\bar{a}^{ij}\bar{b}^{ij}} - \sum_{k} (\mathbf{F}_{ik}(\mathbf{S}^{ij,kl}\bar{\mathrm{C}}^{ik}\mathbf{S}^{ij,kl\dagger})_{\bar{a}^{ij}\bar{b}^{ij}} + \mathbf{F}_{ik}(\mathbf{S}^{ij,kj}\bar{\mathrm{C}}^{\mathrm{kj}}\mathbf{S}^{ij,kj\dagger})_{\bar{a}^{ij}\bar{b}^{ij}}) \\
& + \sum_{k} \left\{\mathbf{S}^{ij,ik}(2\bar{\mathrm{C}}^{ik} - \bar{\mathrm{C}}^{ik\dagger})\left(\mathbf{K}^{kj} - \frac{1}{2}\mathbf{J}^{kj}\right)\mathbf{d}^{ij} + \mathbf{d}^{ij\dagger}\left(\mathbf{K}^{ik} - \frac{1}{2}\mathbf{J}^{ik}\right)(2\bar{\mathrm{C}}^{kj} - \bar{\mathrm{C}}^{kj\dagger})\mathbf{S}^{ij,ik\dagger}\right\}_{\bar{a}^{ij}\bar{b}^{ij}} \\
& - \sum_{k} \left\{\frac{1}{2}\mathbf{S}^{ij,ik}\bar{\mathrm{C}}^{ik\dagger}\mathbf{J}^{jk\dagger}\mathbf{d}^{ij} + \frac{1}{2}\mathbf{d}^{ij\dagger}\mathbf{J}^{ik}\bar{\mathrm{C}}^{kj\dagger}\mathbf{S}^{kj,ij} + \mathbf{d}^{ij\dagger}\mathbf{J}^{ik}\bar{\mathrm{C}}^{ik}\mathbf{S}^{ik,ij} + \mathbf{S}^{ij,kj}\bar{\mathrm{C}}^{kj}\mathbf{J}^{ik\dagger}\mathbf{d}^{ij}\right\}_{\bar{a}^{ij}\bar{b}^{ij}} \\
& + C^{i}_{\bar{a}^{ij}}F^{j}_{\bar{b}^{ij}} + C^{j}_{\bar{b}^{ij}}F^{i}_{\bar{a}^{ij}} - \sum_{k} \{(jk|i\bar{a}^{ij})C^{k}_{\bar{b}^{ij}} + (ik|j\bar{b}^{ij})C^{k}_{\bar{a}^{ij}}\} \\
& + \sum_{\bar{c}^{ij}} (i\bar{a}^{ij}|\bar{c}^{ij}\bar{b}^{ij})C^{j}_{\bar{c}^{ij}} + (i\bar{a}^{ij}|\bar{c}^{ij}\bar{b}^{ij})C^{j}_{\bar{c}^{ij}} - \Delta^{ij}\bar{\mathrm{C}}^{ij}_{\bar{a}^{ij}\bar{b}^{ij}}
\end{aligned} \tag{3}$$

where $K_{ab}^{ij} = (ia|jb)$ and $J_{ab}^{ij} = (ij|ab)$are the usual exchange and Coulomb operators and two-electron integrals in round brackets are written in (11|22) notation. $S_{\bar{a}\bar{b}}^{ij} = \langle \bar{a}^{ij}|\bar{b}^{kl}\rangle = (\mathbf{d}^{ij\dagger}\mathbf{d}^{kl})_{\bar{a}\bar{b}}$ denotes the overlap between PNOs of different pairs. The matrices $\mathbf{d}^{ij}$are the transformation matrices from the canonical virtual basis to the PNO basis of pair i,j.[48]

Following Meyer,[47] $K(\bar{\mathbf{C}}^{ij})_{\bar{a}^{ij}\bar{b}^{ij}}$ denotes the 'external exchange' operator:

$$K(\bar{\mathbf{C}}^{ij})_{\bar{a}^{ij}\bar{b}^{ij}} = \sum_{\bar{a}^{ij}\bar{b}^{ij}} (\bar{a}^{ij}\bar{c}^{ij}|\bar{b}^{ij}\bar{d}^{ij})\bar{C}_{\bar{c}^{ij}\bar{d}^{ij}}^{ij} \tag{4}$$

The singles residual becomes:

$$\begin{aligned}\sigma_a^i &= F_a^i + \{\mathbf{F}^V\mathbf{C}^i\}_a - \sum_j F_{ij}C_a^j + \sum_{jkb}(2K_{jb}^{ik} - J_{jb}^{ik})C_{ba}^{kj} \\ &+ \sum_j \{(2\mathbf{K}^{ij} - \mathbf{J}^{ij})\mathbf{C}^j + \mathbf{F}^j(2\mathbf{C}^{ij\dagger} - \mathbf{C}^{ij})\}_a \\ &+ \sum_j \sum_{\bar{a}^{ij}} d_{a\bar{a}}^{ij} \sum_{\bar{b}^{ij}\bar{c}^{ij}} (2(i\bar{b}^{ij}|\bar{a}^{ij}\bar{c}^{ij}) - (i\bar{c}^{ij}|\bar{a}^{ij}\bar{b}^{ij}))C_{\bar{b}^{ij}\bar{c}^{ij}}^{ij} \\ &- \Delta^i C_a^i\end{aligned} \tag{5}$$

As explained in detail in refs,[51,52] the QCISD and CCSD methods can be implemented in a very similar fashion by introducing "dressed" integrals (to follow usual conventional we use cluster amplitudes $\mathbf{t}$ rather than CI coefficients $\mathbf{C}$ in these equations). The doubles residual becomes:

$$\begin{aligned}\sigma_{ab}^{ij} &= K_{ab}^{ij} + K(\tau^{ij})_{ab} + (\tilde{\tilde{\mathbf{F}}}^{\dagger}\tau^{ij} + \tau^{ij}\tilde{\tilde{\mathbf{F}}})_{ab} - \\ &\sum_k (\tilde{\tilde{F}}_{jk}\mathbf{t}^{ki\dagger} + \tilde{\tilde{F}}_{ik}\mathbf{t}^{kj})_{ab} + \sum_{kl}{}''(ik|jl)\tau_{ab}^{kl} \\ &+ \sum_k \left((2\mathbf{t}^{ki\dagger} - \mathbf{t}^{ki})\left(\tilde{\mathbf{K}}^{jk\dagger} - \frac{1}{2}\tilde{\mathbf{J}}^{jk}\right) - \frac{1}{2}\mathbf{t}^{ki}\tilde{\mathbf{J}}^{jk} - \tilde{\mathbf{J}}^{jk\dagger}\mathbf{t}^{ki\dagger}\right)_{ab} \\ &+ \sum_k \left(\left(\tilde{\mathbf{K}}^{ik} - \frac{1}{2}\tilde{\mathbf{J}}^{ik\dagger}\right)(2\mathbf{t}^{kj} - \mathbf{t}^{kj\dagger}) - \frac{1}{2}\tilde{\mathbf{J}}^{ik\dagger}\mathbf{t}^{kj\dagger} - \mathbf{t}^{kj}\tilde{\mathbf{J}}^{ik}\right)_{ab} \\ &- \sum_k ((jk|ia)t_b^k + (ik|jb)t_b^k) \\ &+ \sum_c ((ia|cb)t_c^i + (ib|ac)t_c^j) - \\ &\left\{\sum_k ((\mathbf{K}^{ik}\mathbf{t}^j)_a t_b^k + (\mathbf{K}^{jk}\mathbf{t}^i)_b t_a^k) + (\mathbf{J}^{ik}\mathbf{t}^j)_b t_a^k + (\mathbf{J}^{jk}\mathbf{t}^i)_a t_b^k\right\}\end{aligned} \tag{6}$$

The dressed quantities are defined elsewhere.[49]

The singles sigma vector can be written in an analogous way as follows:

$$\begin{aligned}\sigma_a^i &= F_{ia} + \sum_b \tilde{F}_{ba}t_b^i - \sum_j \tilde{F}_{ij}t_a^j + G(\mathbf{t}_1)_{ia} + \sum_{jb}(2\mathbf{t}^{ji} - \mathbf{t}^{ji\dagger})\tilde{F}_{bj} \\ &- \sum_{kjb}(2(ik|jb) - (ij|kb))\tau_{ab}^{kj} \\ &+ \sum_{jbc}((2(ib|ac) - (ic|ab))\tau_{bc}^{ij} + (2(jb|ac) - (jc|ab))\tau_{cb}^{ij}) \\ &+ \left\{\sum_{jb}(\tilde{F}_{jb} - 2F_{jb})t_b^i t_a^j\right\}\end{aligned} \tag{7}$$

A complicating feature of LPNO-QCISD and LPNO-CCSD is that the dressed operators change in every iteration and can therefore not be precomputed over PNOs as in the case of LPNO-CEPA. How to overcome this problem is described in detail in ref 49.

Parallel Implementation. Our parallel implementation of the LPNO-CEPA and LPNO-CCSD methods, is based on standard Message Passing Interface[53] (MPI) libraries. The reason for this choice is portability of the code to both shared- and distributed memory machines. One important choice in the design of the parallel algorithm is the use of static or dynamic distribution of work across the processors. Dynamic distribution potentially offers the opportunity for better load balancing, especially when the number of processors is increasing. Our target platforms are clusters of personal computers (PCs) with a limited number of cores rather than massively parallel computers. Given that load-balancing problems do not appear to be overwhelming (vide infra), we have chosen the static distribution model, which potentially also offers the significant advantage of data distribution across local hard drives.

There are three computationally expensive parts of a LPNO-CEPA/CCSD implementation: (a) the canonical integral transformation, (b) the transformation of various classes of two-electron electron repulsion integrals into the PNO basis, and (c) the calculation of the sigma vector. The parallelization of these steps will be described in detail below. The algorithm discussed below has been implemented into the ORCA electronic structure package and is already publically available.[54]

Canonical Integrals. The canonical integral transformations in the LPNO methods are based on the Resolution of the Identity (RI) approximation.[55] In the first step of the transformation, the 3-index repulsion integrals over basis functions are calculated, transformed to the molecular orbital basis, and stored on disk. However, we prefer to avoid the generation, storage and resorting of the largest class of such integrals, $(ab|\tilde{K})$ ($\tilde{K}$ refers to an orthogonalized auxiliary basis function in the Coulomb metric[49]). In the second step, the necessary four index integrals are formed from the prestored three index integrals. The first step is computationally insignificant and will therefore not be further described.

In the second half transformation, the integrals that are required for a LPNO calculation are generated: (a) all internal $(ik|jl)$, b) one external $(ik|ja)$, $(ij|ka)$, and c) two external $(ia|jb)$, $(ij|ab)$. By far the largest set of integrals is the two external integrals and thus special attention should be given to their generation. Given that the set of one-external RI integrals $(ia|\tilde{K})$ is rather small, they can conveniently stored on disk. The generation of $(ia|jb)$ then proceeds efficiently (as in the case of RI-MP2[56]) by efficient matrix multiplications. We store matrices ordered by the internal index to this end ($X_{aK}^i =(ia|\tilde{K})$).

The generation of the Coulomb integrals, $(ij|ab)$, is more difficult as a convenient matrix driven strategy is less obvious. Since the generation of the three-index RI integrals over atomic orbitals, $(\mu\nu|K)$, is computationally inexpensive compared to the remaining steps, we have developed an integral direct algorithm for the generation of $(ij|ab)$ with the intentions to: (a) minimize input/output (I/O) overheads, (b) avoid lengthy integral sorts, (c) use efficient BLAS level 3 matrix multiplications to the largest possible extent within

```
Read in all-internal RI integrals
Construct batches according to number of processors and available memory
For batches of indices ab distributed across processors
    For K in auxiliary basis functions
          for $\mu$ in atomic orbitals
                for $\nu$ in atomic orbitals ($\mu \leq \nu$)
                      Prescreen for negligible integrals
                      Calculate $Y^{K}(\mu,\nu)=(\mu\nu|K)$.
                End $\nu$
          End $\mu$
          Transform $X(K,ab)=(K|ab)=(\mathbf{c}_{L}^{T}\ \mathbf{Y}^{K}\ \mathbf{c}_{R})_{ab}$
            (using batch specific truncated MO coefficient matrices $\mathbf{c}$)
    End K.
    Orthogonalize the integrals $\mathbf{Z}= \mathbf{V}^{-1/2}\mathbf{X}$.
    Generate subset of target integrals $\mathbf{J}^{ij}(a,b)=(ij|K)*Z(K,ab)$
                                  (using matrix multiplications)
    Gather the local results
    Store the integrals.
End batches
```

Figure 1. Pseudo code for the parallel generation of (*ij*|*ab*) within the RI approximation as implemented in ORCA.

a given amount of core memory, and (d) achieve good parallel scalability.

This strategy implies a batch algorithm that is structured as shown in Figure 1. It is assumed that the all internal RI integrals, (*ij*|*K*), fit into memory and this assumption has so far never created a bottleneck in actual calculations. In each batch, a loop over all internal pairs is performed and as many external pairs as fit in memory are treated simultaneously. In the parallel algorithm, the work is distributed over external pairs thus leading to fine-grained parallelization. Inside each batch, the RI integral generation is driven by an outer loop over auxiliary functions *K*. For each *K* within a given angular momentum shell, the entire matrix of atomic integrals $Y_{\mu\nu}^{K} = (\mu\nu|K)$ is generated. From these integrals, BLAS level 3 matrix multiplications are used to generated as many (*ab*|*K*) as memory permits. Truncated MO coefficient matrices are set up to this end. These integrals are orthogonalized through a second matrix multiplication with the $\mathbf{V}^{-\ 1/2}$ matrix ($V_{KL} = (K|L)$ is the Coulomb metric) or an equivalent thereof. The orthogonalized integrals are contracted in the third step with $(ij|\tilde{K})$ in order to generate a subset of the target (*ij*|*ab*) integrals. For storage of these and similar integrals, a special data structure is employed (referred to as “Matrix Container” in ORCA), where integrals are stored as matrices ordered by the internal pair labels ($\mathbf{J}^{ij}$,$\mathbf{K}^{ij}$) as required by the matrix formulation of the CCSD and CEPA equations (eqs 3, 5, 6, and 7). In the parallel algorithm, a limited amount of overhead arises from gathering the partial $\mathbf{J}^{ij}$ operators formed at different processors before writing them to disk. This procedure also necessarily synchronizes the processes. The rest of the algorithm does not require communication and proceeds as efficiently as in the single processor case since the size of the matrix multiplications are independent of the number of processors. The disk storage of the integrals is organized in a “per node” fashion, where the integrals are stored only for each node and not for each processor or core. This implies that if all processors are located on the same node, the integrals are stored only once and are available to all processors. As long as the communication overhead remains small and the generation of the RI-integrals over atomic orbitals is not rate limiting, the algorithm is expected to scale nearly linearly with the number of processes.

Table 1. Wall-Clock Timings and Speedups for the Parallel Creation of the (ij|ab) Integrals for **1**[a]

	processors						
direct_Jij	1	2	4	6	8	10	14
time (sec)	71008	33785	19269	13149	10611	8375	7424
speedup	1	2.1	3.7	5.4	6.7	8.5	9.6

[a] Basis Set: def2-TZVP[58,59] (1130 basis functions). Auxilliary Basis Set: def2-TZVP/C (3122 basis functions).

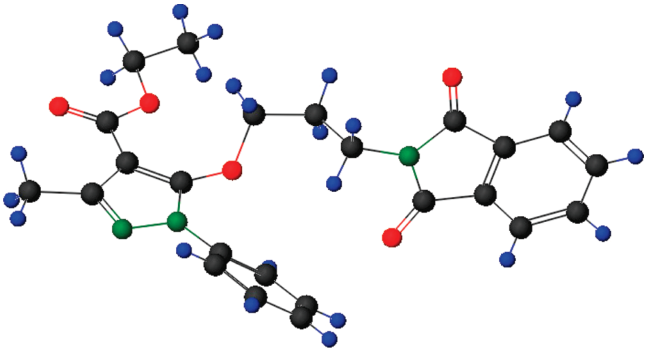

Figure 2. The structure of **1** = 5-[3-(1,3-dioxo-1,3-dihydro-isoindol-2-yl)-propoxy]-3-methyl-1-phenyl-1H-py pyrazole-4-carboxylic acid ethyl ester ($C_{24}H_{23}N_3O_5$).

In Table 1, we present timings for the transformation. For all timings presented in this work, we used the molecule 5-[3-(1,3-dioxo-1,3-dihydro-isoindol-2-yl)-propoxy]-3-methyl-1-phenyl-1H-py pyrazole-4-carboxylic acid ethyl ester ($C_{24}H_{23}N_3O_5 \equiv$ **1**, see Figure 2).[57] The def2-TZVP[58,59] basis set is used (1130 contracted basis functions), together with the corresponding def2-TZVP/C auxiliary basis (3122 auxiliary basis functions).

It is observed that the scaling with the number of processes is, unfortunately, not perfectly linear. We believe that the reason for this is that the calculations were performed on a machine with four AMD 4 Quad-Core AMD Opteron 8354 processors per node. For these machines, it is known that memory bandwidth becomes rate limiting for compute and memory intensive operations.[60,61] However, a speedup of 9.6 is still observed with 14 processors. For parallel machines with better memory bandwidth better scalability is expected.

```
For K=MyID  to K = Naux, K+=NumProcs
    For μ in basis functions
         For ν in basis functions (μ ≤ ν)
              Prescreen for negligible integrals
              Calculate (μν|K)
         End ν
    End μ
    Transform (μν|K) to (ij|K), (ia|K) and (ab|K)
    For ip(ij) in internal pairs
         Create and store three index integrals over the PNOs of pair ip
    End ip
End K
Collect local parts of integrals.
For ip=MyID to ip < Internal pairs. ip += NumProcs
    Copy the relevant part of the metric matrix for pair ip
    Create local orthogonalization matrix for pair ip
    Use this matrix to transform the PNO repulsion integrals for pair ip.
    Construct the final 4-index repulsion integrals for pair ip.
End ip
Gather local parts
```

Figure 3. Pseudo code for the parallel local transformation of the repulsion integrals to the PNO basis.

PNO Transformation. The second computationally expensive step is the generation of electron−electron repulsion integrals over PNOs. Since the number of PNOs is typically much larger than the number of virtual orbitals, the transformation must be carefully arranged in order to avoid computational bottlenecks (Figure 3). As discussed in the original work,[48] the RI approximation is used to generate the integrals over PNOs. The procedure is divided in two steps. In the first step, the RI integrals over AOs are generated ($(\mu\nu|K)$) with the auxiliary index running slow, as described above for the canonical integrals. From these integrals, we first produce the canonical integrals $(ij|K)$, $(ia|K)$, and $(ab|K)$ for a given set of K's that belong to a given angular momentum shell inside the auxiliary basis set. This transformation is done using BLAS level 3 operations. Hence, sparsity and integral prescreening is used in the integral generation but not in the initial transformation. A loop over pairs $P \equiv (ij)$ is next performed. Inside this loop, we first generate and store the integrals $(ij|K)$, $(i\bar{a}ij|K)$, $(j\bar{a}ij|K)$, and $(\bar{a}ij\bar{b}ij|K)$. Since the number of these integrals is linear scaling, the storage creates no bottlenecks despite the fact that there may be tens of thousands of PNOs in the calculation. It should be noted that the algorithm contains pair specific local fitting domains such that only those pairs that have the auxiliary index K inside their domain are treated. As explained in detail in reference,[48] the domain construction is organized using a highly conservative threshold ($T_{\mathrm{CutMKN}} = 10^{-3}$) such that the error introduced is negligible. The loop ends by synchronization in which all integrals are communicated to all processors.

The last step of the PNO integral transformation algorithm is driven by a loop over internal electron pairs. For each pair, the local Coulomb metric is created and the pair specific three index repulsion integrals created in the previous step are orthogonalized to give three-index repulsion integrals over orthogonalized auxiliary functions $\tilde{K}$. This enables the calculation of the target integrals $(i\bar{a}^{ij}|j\bar{b}^{ij})$, $(ij|\bar{a}^{ij}\bar{b}^{ij})$, $(i\bar{a}^{ij}|\bar{b}^{ij}\bar{c}^{ij})$, $(j\bar{a}^{ij}|\bar{b}^{ij}\bar{c}^{ij})$, and $(\bar{a}^{ij}\bar{b}^{ij}|\bar{c}^{ij}\bar{d}^{ij})$ through efficient dense matrix multiplications. These integrals are stored on disk. This step of the algorithm is linear scaling and is, once more, parallelized in a round robin fashion by distributing the loop over pairs over all processors.

Table 2. Wall-Clock Timings and Speedups for the Integral Parallel Transformation to the PNO Basis during a LPNO−CCSD Calculation on **1**[a]

PNO transformation	processors						
	1	2	4	6	8	10	14
time (sec)	39494	25144	11691	8292	6990	7658	4821
speedup	1	1.6	3.4	4.8	5.7	5.2	8.2

[a] Basis Set: def2-TZVP[58,59] (1130 basis functions). Auxilliary Basis Set: def2-TZVP/C (3122 basis functions).

The final step of the PNO integrals transformation consists of the calculation of overlap matrices between the PNOs of different pairs. These overlap matrices are stored in packed form on disk. Only a linear number of overlap integrals is nonvanishing such that no storage bottlenecks arise. The overlap integrals are calculated through efficient matrix multiplications.

In Table 2, we present the time and speed-ups for the PNO transformation. Once more, it is noted that the algorithm does not scale perfectly linearly with the number of processors. Again, we believe that this is due to limited memory bandwidth of the machines used. However, a speedup of 8.2 is still observed for 14 processors.

Sigma Vector Construction. The third major part of the LPNO-CEPA/CCSD methods is the construction of the sigma vector. Below, we will comment on how the computationally most significant terms were implemented in the parallel case.

Singles Fock. The construction of the singles Fock matrix $G(t_1)$ is presently the most time-consuming step in the sigma-vector construction during a LPNO−CCSD calculation. As discussed in ref 49, this step is not necessary for LPNO-CEPA or LPNO-QCISD which hence do have a timing advantage over LPNO−CCSD. The advantages are most pronounced for highly polarized basis sets where integral evaluation becomes computationally expensive. The parallelization of this step is, however, straightforward and follows the techniques that are available throughout the ORCA program. The load balancing obtained during Fock-matrix construction is excellent and the direct computation of the singles-Fock matrix scales nearly linearly with the number of processors up to 14 processors. The only communication required is the gathering of the partial Fock-type matrices after the integral evaluation has been completed. It is to be expected that this step will significantly improve through the use of the RIJCOSX[62] approximation.

Doubles−Doubles Interaction with No External Labels. The computationally next most demanding term is the disjoint doubles−doubles interaction that involves the all-internal exchange integrals (*ik*|*jl*). These terms are written in canonical form as follows:[48]

$$\sum_{kl} K_{kl}^{ij} t_{ab}^{kl} - \sum_{k} \{F_{jk} t_{ab}^{ik} + F_{ik} t_{ab}^{kj}\}$$

and in the LPNO case as follows:

$$\sum_{kl} (ik|jl)(\mathbf{S}^{ij,kl}\bar{\mathbf{t}}^{kl}\mathbf{S}^{ij,kl\dagger})_{\bar{a}^{ij}\bar{b}^{ij}} - \sum_{k} (F_{jk}(\mathbf{S}^{ij,ik}\bar{\mathbf{t}}^{ik}\mathbf{S}^{ij,ik\dagger})_{\bar{a}^{ij}\bar{b}^{ij}} + F_{ik}(\mathbf{S}^{ij,kj}\bar{\mathbf{t}}^{kj}\mathbf{S}^{ij,kj\dagger})_{\bar{a}^{ij}\bar{b}^{ij}})$$

The algorithm commences with a double loop over *k* and *l* followed by reading a matrix of integrals, $K_{ij}^{kl} = (ki|lj)$. This is followed by a loop over pairs P with internal indices *i* and *j*. Pair−pair interactions with negligible values of |(*ki*|*lj*)| are dropped. The remaining significant contributions are calculated by transforming the amplitudes of pair *k*,*l* into the PNO basis of pair *i*,*j* and update of the sigma-vector. Parallelization takes place by distributing the pair loop over processors. There is no communication between processors required in this step as the local sigma vector components are gathered at the end of each coupled pair/coupled cluster iteration. Thus, the only issue to be addressed is load balancing that is once more done in a round-robin fashion. Thus, this step should scale linearly with the number of processors as the load balancing will tend to become perfect for larger molecules.

Doubles-Singles Interaction with One External Label. Quite unexpectedly, a step that becomes computationally significant in the LPNO methods is the interaction of the double excitations with singles through integrals with one-external label. The origin of this is that the singles are left in canonical form. The sigma-vector contribution is written as follows:

$$\sigma_a^i = \sum_j \{(2\mathbf{K}^{ij} - \mathbf{J}^{ij})\mathbf{t}^j + \mathbf{F}^j(2\mathbf{t}^{ij\dagger} - \mathbf{t}^{ij})\}$$

The complication arises from the fact that the doubles amplitudes have to be in the canonical basis for the calculation of this term. Hence, the back transformation from the PNO to the canonical basis must be performed inside a pair loop. It is to be expected that the calculation of this term can be significantly improved through prescreening or a more judicious choice if orbitals to expand the single excitations in.

Parallelization is achieved by dividing the pair loop over processors in a round-robin fashion. During the loop over pairs, there is no communication needed among the processors and the parallel overhead comes just in the form of gathering the individual parts of the singles sigma vectors, which is computationally insignificant. Hence, this part of the calculation should scale linearly with the number of processors.

Table 3. Wall-Clock Timings and Speedups for the Creation of the Sigma Vector in a LPNO−CCSD Calculation on **1**[a]

sigma	processors						
vector	1	2	4	6	8	10	14
time (sec)	188857	99326	53619	34203	24420	22029	15024
speedup	1	1.9	3.5	5.5	7.7	8.6	12.6

[a] Basis Set: def2-TZVP[58,59] (1130 basis functions). Auxilliary Basis Set: def2-TZVP/C (3122 basis functions).

Doubles-Doubles Interaction with Two External Labels. As discussed at lengths in refs 48 and 49 this term can be written as follows:

$$\sum_k \left\{ \begin{matrix} \mathbf{S}^{ij,ik}(2\bar{\mathbf{t}}^{ik} - \bar{\mathbf{t}}^{ik\dagger})\left(\mathbf{K}^{kj} - \frac{1}{2}\mathbf{J}^{kj}\right)\mathbf{d}^{ij} + \\ \mathbf{d}^{ij\dagger}\left(\mathbf{K}^{ik} - \frac{1}{2}\mathbf{J}^{ik}\right)(2\bar{\mathbf{t}}^{kj} - \bar{\mathbf{t}}^{kj\dagger})\mathbf{S}^{ij,ik\dagger} \end{matrix} \right\}_{\bar{a}^{ij}\bar{b}^{ij}} - \sum_k \left\{ \begin{matrix} \frac{1}{2}\mathbf{S}^{ij,ik}\bar{\mathbf{t}}^{ik\dagger}\mathbf{J}^{jk\dagger}\mathbf{d}^{ij} + \frac{1}{2}\mathbf{d}^{ij\dagger}\mathbf{J}^{ik}\bar{\mathbf{t}}^{kj\dagger}\mathbf{S}^{kj,ij} + \\ \mathbf{d}^{ij\dagger}\mathbf{J}^{ik}\bar{\mathbf{t}}^{ik}\mathbf{S}^{ik,ij} + \mathbf{S}^{ij,kj}\bar{\mathbf{t}}^{kj}\mathbf{J}^{ik\dagger}\mathbf{d}^{ij} \end{matrix} \right\}_{\bar{a}^{ij}\bar{b}^{ij}}$$

As explained elsewhere,[49] this term greatly profits from precalculating the contractions of the canonical **J***ij* and **K***ij* operators with the relevant PNO transformation matrices **d**. During the iterations, these partial integrals are read and contracted with the PNO amplitudes of the interacting pair in order to arrive at the final sigma-vector contributions.

In order to parallelize this contribution, the first step consists of the parallelization of the creation of the partial integral file. This is achieved by treating only a subset of internal pairs on each processor. This process creates a local partial pair interaction file. Hence, during sigma-vector construction each processor only processes this partial file which means that no further explicit parallelization is necessary.

Parallel Speedup of Sigma-Vector Contributions. In Table 3 and Figure 4, the absolute times and speedups for the creation of the sigma vector for the test molecule in Figure 2 are shown. It is observed that up to 14 processors the program scales very well, as expected from the analysis presented above. In Table 4 the wall-clock times for the complete LPNO-CCSD calculation are analyzed. An overall speedup of 9.4 is observed for 14 processors. While this is less than the ideal result, we have argued above that the parallel efficiency is presently mainly limited by memory bandwidth of the machines used and hence may well improve for future generation hardware. We have nevertheless chosen to use the present machines as they represent typical hardware for computational chemistry applications. The main result is that the wall clock times required for LPNO-CCSD calculations on standard hardware can be reduced by an order of magnitude through parallelization.

Application to Weak Interactions

Noncovalent interactions play a key role in modern chemical research, for example in supramolecular chemistry. Hence,

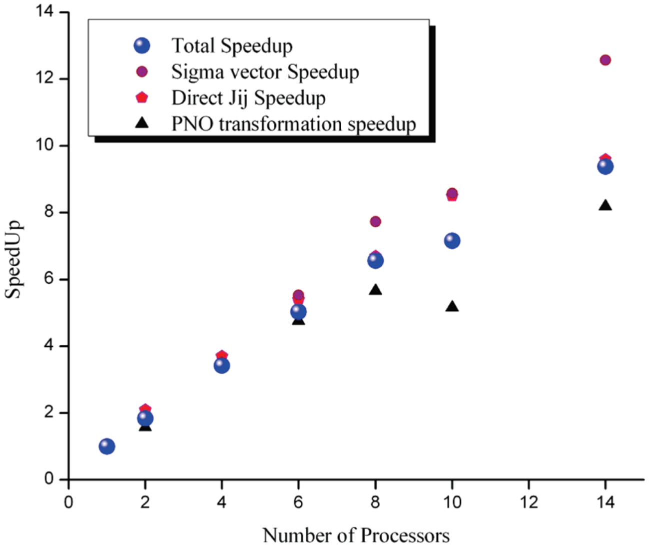

Figure 4. Plot of the speedups for a LPNO-CCSD calculation on **1** versus the number of processors used. A computer with four Quad-Core AMD Opteron 8354 Processors per node was used for the calculations. The size of the cache memory was 512 Kbytes at the L2 level and 2 Mbytes at the L3 level. Overall, 120 GBytes of memory were available for the whole node and the program used 1.5 GBytes per processor.

Table 4. Wall-Clock Timings and Speedups for a Complete LPNO−CCSD Calculation on **1**[a]

total LPNO−CCSD	processors						
	1	2	4	6	8	10	14
time (sec)	336789	183084	98674	66973	51393	47099	35927
speedup	1	1.8	3.4	5.0	6.6	7.2	9.4

[a] Basis Set: def2-TZVP[58,59] (1130 basis functions). Auxilliary Basis Set: def2-TZVP/C (3122 basis functions).

this is also an active area of theoretical research.[63−67] The description of weak intermolecular forces presents a challenge for the theoretical chemistry since highly accurate methods such as CCSD(T) are computationally too expensive for routine use and computationally cheaper methods (mostly based on DFT) are not accurate enough.[68−71] (However, see recent results of Grimme and co-workers[6,56,72,73]). As will be shown below, the LPNO methods offer a computationally tractable route to this problem without introducing any element of empiricism.

S22 Benchmark Calculations. In order to evaluate the accuracy of the LPNO based methods for intermolecular interactions, the defacto standard S22 set of molecules was treated.[74] After the original publication in 2006, two recent articles have been published[75,76] that improve upon the reference values. In this work, the results of Takatani et al.[76] were used as reference. The geometries provided in the original publications were used throughout. We have also followed the same extrapolation scheme for the estimation of the complete basis set (CBS) limit. The complete basis set (CBS) MP2 correlation energy is estimated as follows:[77]

$$E_{\mathrm{MP2}}^{(\infty)} = \frac{X^3 E_{\mathrm{MP2}}^{(X)} - Y^3 E_{\mathrm{MP2}}^{(Y)}}{X^3 - Y^3} \qquad (8)$$

Table 5. Calculated Reaction Energies for the S22 Set Using the LPNO-CEPA/1 Method[a]

complex	ref 76	calculated energy	error
$(NH_3)_2$	−3.17	−3.03	−0.14
$(H_2O)_2$	−5.02	−4.93	−0.09
formic acid dimer	−18.8	−18.28	−0.52
formamide dimer	−16.12	−15.61	−0.51
uracil dimer ($C_{2\ h}$)	−20.69	−19.94	−0.75
2-pyridoxine.. 0.2-amino-pyridine	−17	−16.24	−0.76
adenine...thymine (WC)	−16.74	−15.95	−0.79
$(CH_4)_2$	−0.53	−0.49	−0.04
$(C_2H_4)_2$	−1.5	−1.45	−0.05
C_6H_6...CH_4	−1.45	−1.53	0.08
C_6H_6...C_6H_6(stacked)	−2.62	−2.59	−0.03
pyrazine dimer	−4.2	−4.01	−0.19
uracil dimer (C_2)	−9.74	−9.45	−0.29
indole...benzene (stacked)	−4.59	−4.62	0.03
adenine...thymine (stacked)	−11.66	−11.14	−0.52
C_2H_4...C_2H_2	−1.51	−1.54	0.03
C_6H_6...H_2O	−3.29	−3.27	−0.02
C_6H_6...NH_3	−2.32	−2.37	0.05
C_6H_6...HCN	−4.55	−4.63	0.08
C_6H_6...C_6H_6 (T-Shape)	−2.71	−2.84	0.13
indole...benzene(T-shape)	−5.62	−5.71	0.09
phenol dimer	−7.09	−6.89	−0.20
mean absolute error			0.24
max error			0.79

[a] All values in kcal/mol.

Here X and Y are ($Y > X$) the two cardinal numbers of the two basis sets used for the two-point extrapolation and $E_{\mathrm{MP2}}^{(X)}$,$E_{\mathrm{MP2}}^{(Y)}$ are the MP2 correlation energies calculated with the two basis sets. The total energy in the CBS limit is then estimated as follows:

$$E_{\mathrm{total}}^{(\mathrm{CBS})} \approx E_{\mathrm{SCF}}^{Y} + E_{\mathrm{corr}}^{\mathrm{LPNO};X} + E_{\mathrm{corr}}^{(\mathrm{MP2};\infty)} - E_{\mathrm{corr}}^{(\mathrm{MP2};X)} \qquad (9)$$

Here $E_{\mathrm{corr}}^{(\mathrm{LPNO};X)}$ is the correlation energy calculated with LPNO-CEPA/1 within the smaller basis set and E_{SCF}^{Y} the SCF energy calculated with the larger basis set. Thus, it is assumed that the SCF energy with the larger basis is close enough to the basis set limit and that the difference between the MP2 basis set limit and the MP2 energy calculated with the small basis set is a reliable estimate of the difference in correlation energies that would be obtained with the high-level correlation method (in this case LPNO-CEPA/1). The accuracy of this scheme will be investigated in detail elsewhere.

For all molecules the aug-cc-pVTZ/aug-cc-pVQZ bases[78] together with their corresponding auxiliary basis sets for the RI approximation were employed. No counterpoise correction was employed as this is inconsistent with the notion of CBS extrapolation.

In Table 5, the calculated reaction energies are presented together with the deviations from the reference values and the calculated mean absolute error (MAE). It is observed that relative to the reference CCSD(T) values, the MAE of the LPNO-CEPA/1 method is only 0.24 kcal/mol and the maximum error is 0.79 kcal/mol. These results imply that the LPNO-CEPA/1 method is an accurate and computationally efficient method for the treatment of weak, noncovalent interactions. Importantly, this method can be applied to much larger systems than the canonical CEPA, CCSD, or CCSD(T) methods.

Table 6. Mean Absolute Errors for the S22 Set Using Various Electronic Structure Methods

method	type	mean absolute error (kcal/mol)
LPNO-CEPA/1		0.24
PW86PBE-XDM(BR)	GGA-XDM	0.34
B2PLYP-D	double hybrid-D	0.43
M06−2X/MG3S	hybrid meta-GGA	0.43
B97-D	GGA-D	0.46
B3LYP-D	hybrid GGA-D	0.52
PW86PBE-XDM(XX)	GGA-XDM	0.72
B2PLYP	double hybrid	1.79
B3LYP	hybrid GGA	3.77
BP86	GGA	4.14

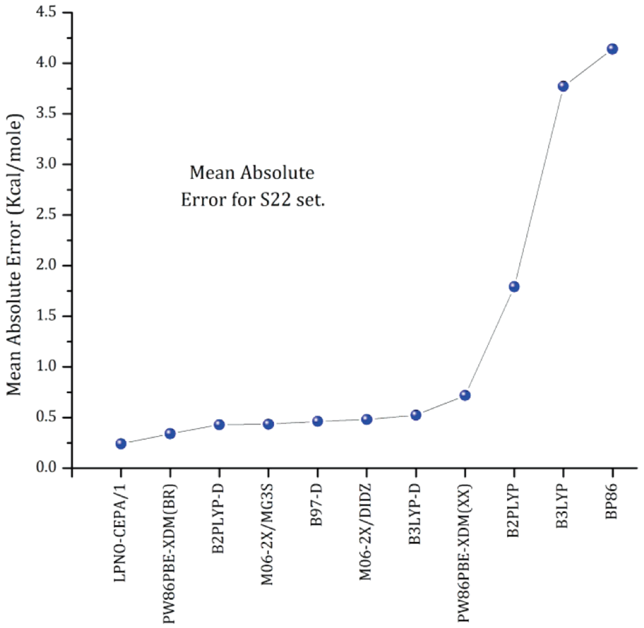

Figure 5. Mean Absolute Error for the 22 reactions of the S22 set.

In order to compare our results with the ones obtained by present day DFT functionals, representative results are collected in Table 6 and are graphically shown in Figure 5. While the collection of functionals represent only a small fraction of the available literature, they do represent the presently prevailing classes of functionals (Zhao et al.[13] present a comparison for a more extensive set of functionals). Specifically, we used BP86[79,80] (values taken from reference[53]) as an example of a GGA functional, B3LYP[81−83] as a typical hybrid functional (values taken from Prof. Grimmes Web site[84]) and B2PLYP[85] as a double hybrid functional (values taken from Prof. Grimmes Web site[84]). These classes of functional were then considered together with the semiempirical van der Waals correction proposed by Grimme (B97-D,[86] B3LYP-D,[3,4] and B2PLYP-D[3,4] functionals as dispersion corrected GGA, hybrid and doubly hybrid functionals respectively. M06−2X/MG3S47[71] and M06−2X/DIDZ47[71] (values taken from ref 71) were used as typical hybrid meta-GGA functionals. Finally, the recently proposed PW86PBE-XDM(BR)[8] and PW86PBE-XDM(XX)[8] functionals that are specifically tailored to the calculation of noncovalent interactions were considered. It is pleasing to observe that of all methods tested, the LPNO-CEPA/1 methods leads to the smallest mean absolute error. Thus it even outperforms functionals that are specifically designed for the calculation of weak interactions. That LPNO-CEPA/1 is also highly successful for many other molecular properties[48] and hence represents an accurate and efficient model chemistry.

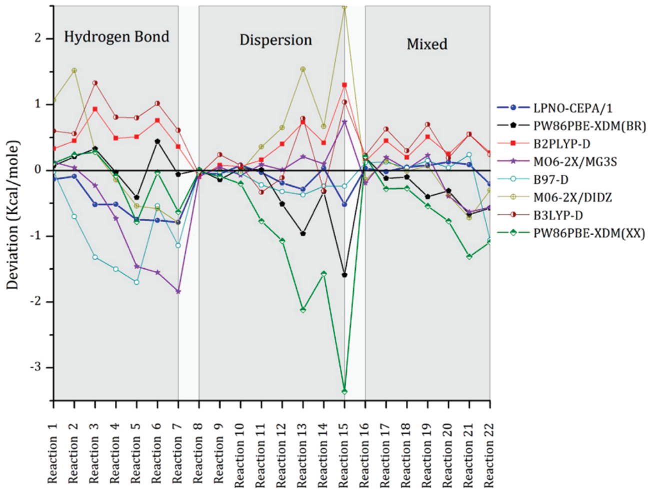

Figure 6. Deviations from the reference values for the S22 set obtained by different computational methods as described in the text.

In order to analyze these results more closely, Figure 6 presents a plot collecting the deviations from the reference values for different methods. The plot has three parts describing reactions concerning complexes with mainly hydrogen bonding interactions, complexes where dispersion forces are dominant and finally complexes with both types of interactions. It is concluded that none of the tested functionals can treat both kind of interactions with consistent accuracy. For example PW86PBE-XBM(BR), which presented the lowest mean absolute error of all functionals, seems to describe hydrogen bonds accurate but shows errors of up to 1.6 kcal/mol for the dispersion dominated cases. Similarly, B97-D describes dispersion very well but is less accurate for hydrogen bonds. The same is true for LPNO-CEPA/1, where however, the largest absolute error is still smaller than 1 kcal/mol.

Application to a Trimer System. In order to demonstrate the application of LPNO-CEPA/1 to a larger "real-life" system, 2,4-dihydroxy-3-acetyl-6-methyl acetophenone [$C_{11}H_{12}O_4$] (≡ **2**, see Figure 7) was treated. This system has been recently synthesized and characterized by spectroscopic as well as X-ray diffraction studies.[87] It was found that in the crystal structure both hydrogen bonds and $\pi-\pi$ stacking interactions are crucial for the formation of a two-dimensional supramolecular assembly (Figure 7).

The crystal structure of **2** shows that there are discrete trimers formed and it is evident that the π-stacking and hydrogen bonding interactions can be nicely differentiated. Monomer A interacts with monomer B through two hydrogen bonds while the interaction between monomers B and C occurs through $\pi-\pi$ stacking interactions.

Since the two interactions are fairly well separated, we have conjectured that only two geometrical parameters are

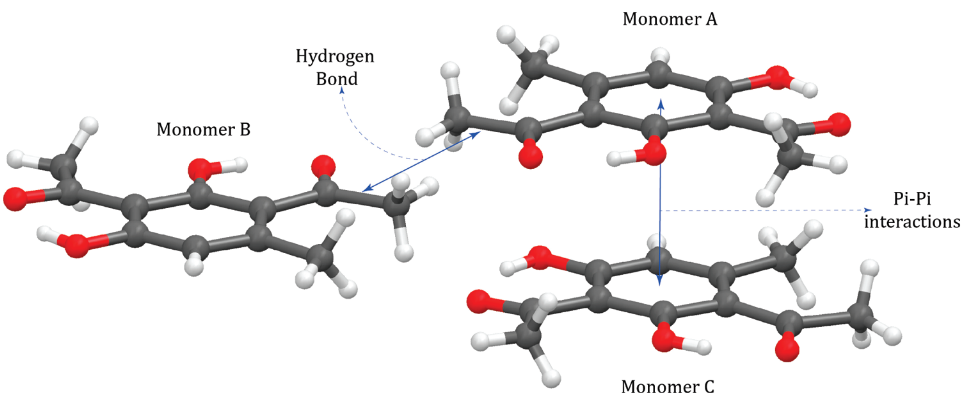

Figure 7. The molcular structure of **2** = (2−4-dihydroxy-3-acetyl-6-methyl acetophenone)$_3$.

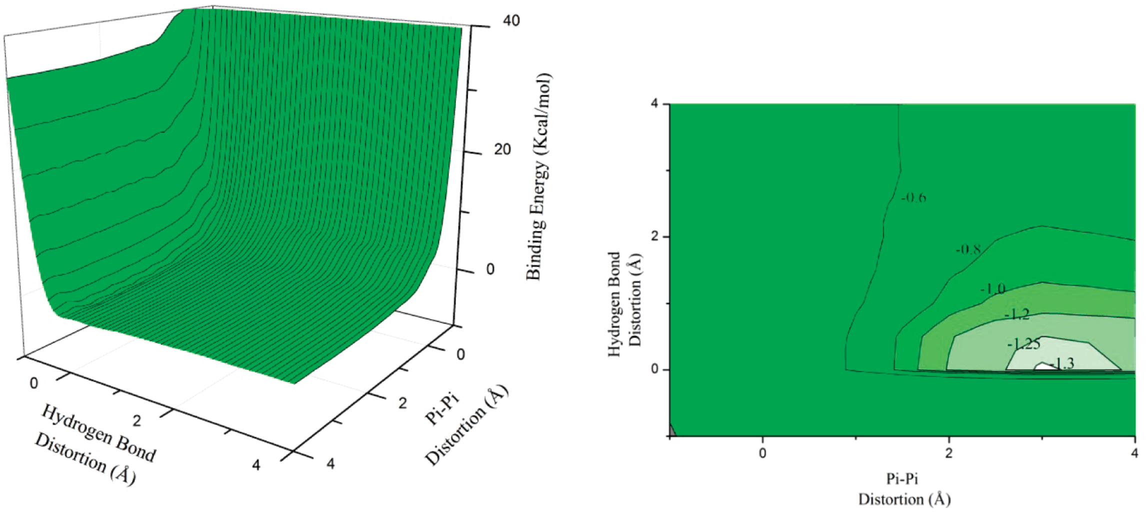

Figure 8. The PES of **2** calculated at the BP86/def2-TZVP level of theory. Left: 3-dimensional surface plot, Right: Contour plot.

of key importance. The hydrogen bond contact interactions and the $\pi-\pi$ stacking distance. The total energy was scanned along these two coordinates (see Figure 5) while keeping the remaining internal structures of the building blocks fixed. Eleven steps between −1 and 4 Å (relative to the experimental structure) were performed for each coordinate. Additionally close to the potential minimum, a finer mesh of 0.2 Å was used. Overall, 133 points were calculated. For comparison, both BP86 and LPNO-CEPA/1 calculations were performed on the grid in combination with the def2-TZVP[59,88] basis set (def2-TZVP(-f) for LPNO-CEPA/1, 1296 basis functions, 4500 auxiliary basis functions). In both cases, we also performed calculations to correct for the basis set superposition error (BSSE). In Figure 8, the resulting potential energy surface for the BP86 case is shown.

Obviously, BP86[80] correctly predicts the location of the minimum for the hydrogen bond distance coordinate but completely fails for $\pi-\pi$ interactions. In Figure 9, the analogous graphs are shown for LPNO-CEPA/1. Pleasingly, clear minima almost exactly at the experimental geometry are found for both, the hydrogen bond and the $\pi-\pi$ interaction coordinate. In Figure 10, one-dimensional cuts are presented for both coordinates while keeping the other at the experimental value.

It appears that the hydrogen bond is calculated ~0.1 Å shorter than the experimental value which appears to be within the experimental error and the resolution of the grid that was used to scan the energy in our calculations. For the case of $\pi-\pi$ interactions, the calculated minimum almost exactly corresponds with the experimental one.

We note in passing that the hydrogen bond energy was found to be 4.0 kcal/mol and the $\pi-\pi$ stacking energy −17.7 kcal/mol, both of which are plausible values. Obviously, our calculations neglect any impact of static disorder or crystal packing effects and they cannot be compared to experimental data. Nevertheless, this application should provide a reasonable feeling for what is presently routinely possible with the LPNO-CEPA/1 method.

Conclusions

In this work, a parallel implementation of the LPNO-CEPA and LPNO-QCISD/LPNO-CCSD family of methods was presented. A detailed analysis of the parallelization scheme was presented together with representative timings. It was argued that the scaling of the algorithm is inherently nearly linear with the number of processors but that less than

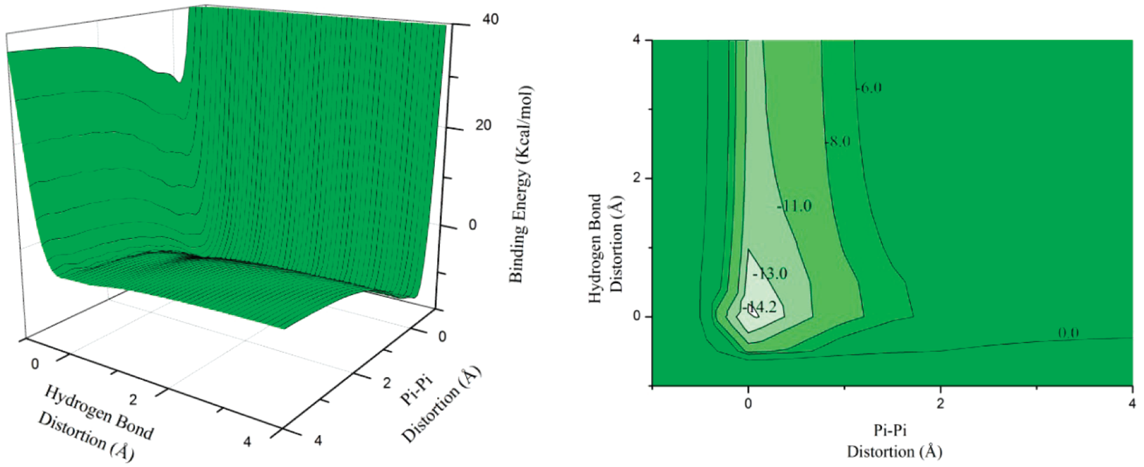

Figure 9. The PES calculated for **2** with LPNO-CEPA/1/def2-TZVP(-f). Left: 3-dimensional plot, Right: Contour plot.

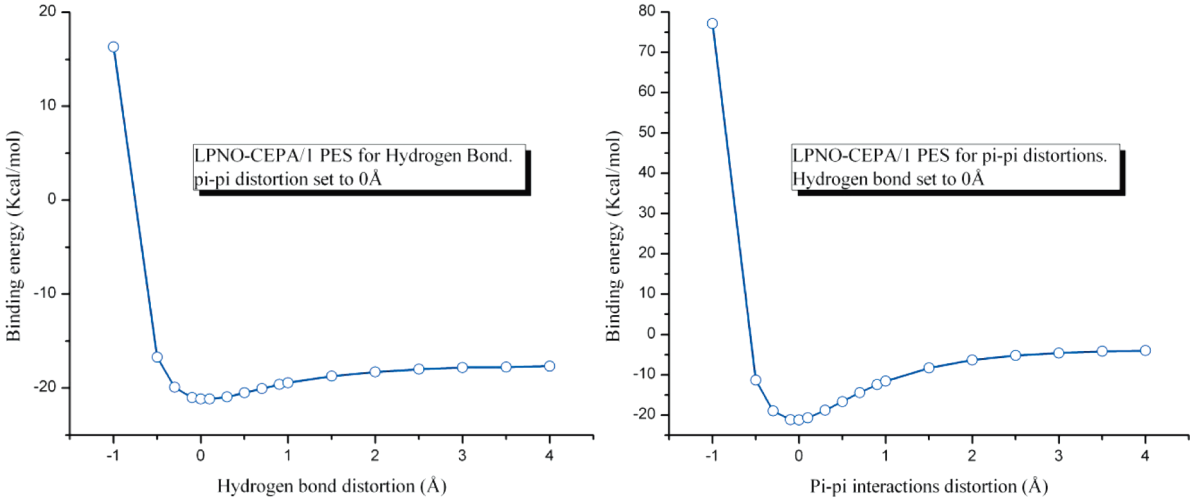

Figure 10. One dimensional plots for hydrogen bond and $\pi-\pi$ interactions for the trimer calculated with LPNO-CEPA/1/def2-TZVP(-f). On the left we set $\pi-\pi$ equal to the experimental value and vary the Hydrogen bond distance while on the left we set the Hydrogen bond distance equal to the experimental one and distort the $\pi-\pi$ interaction distance.

optimum scaling is observed on present day machines due to limited memory bandwidth. Nevertheless, even with clusters built from cheap mass-market computers, speedups of about an order of magnitude can be obtained with 14 processors.

Second, the accuracy of the LPNO-CEPA/1 method was tested for weak intermolecular interactions. It was found that LPNO-CEPA/1 is an accurate method for such applications: the Mean Absolute Error was found to be 0.24 kcal/mol. This shows that no essential physics are lost with the LPNO approximations and that the canonical or LPNO-CEPA/1 methods provide results that are very close to CCSD(T) for such interactions. Given that LPNO-CEPA/1 is applicable to much larger systems than CCSD(T), this demonstrates a high potential of this method for chemical applications involving weak intermolecular interactions. Quite pleasingly, it was found that the accuracy of LPNO-CEPA/1 exceeds that of even the most purpose specific density functionals. The applicability of the LPNO-CEPA/1 method to larger systems was proven for the case of an acetophenone derivative trimer in which both hydrogen bonding and $\pi-\pi$ stacking interactions are prominent. This shows that large scale applications with ~1500 basis functions are rendered a routine application with the parallel version of the LPNO-CEPA/1 program. Together with the fact that the LPNO-CEPA/1 method is as easy to use as its canonical counterpart or a standard DFT functional, we do consider these developments as significant progress. A detailed discussion of the relative merits of LPNO-CEPA/1 and other local correlation schemes has been given in refs 48 and 49.

References

(1) Koch, W.; Holthausen, M. C. *A Chemist's Guide to Density Functional Theory*; Wiley-VCH: Weinheim, 2000.

(2) Neese, F. *Coord. Chem. Rev.* **2009**, *253*, 526–563.

(3) Grimme, S. *J. Comput. Chem.* **2004**, *25*, 1463–1473.

(4) Grimme, S. *J. Comput. Chem.* **2006**, *27*, 1787–1799.

(5) Peverati, R.; Baldridge, K. K. *J. Chem. Theory Comput.* **2008**, *4*, 2030–2048.

(6) Grimme, S.; Antony, J.; Ehrlich, S.; Krieg, H. *J. Chem. Phys.* **2010**, *132*, 154104–154119.

(7) Wodrich, M. D.; Jana, D. F.; Schleyer, P. v. R.; Corminboeuf, C. m. *J. Phys. Chem. A* **2008**, *112*, 11495–11500.

(8) Kannemann, F. O.; Becke, A. D. *J. Chem. Theory Comput.* **2010**, *6*, 1081–1088.

(9) Vydrov, O. A.; Van Voorhis, T. *J. Chem. Phys.* **2010**, *132*, 164113–164116.

(10) Sato, T.; Nakai, H. *J. Chem. Phys.* **2009**, *131*, 224104–224112.

(11) Chai, J.-D.; Head-Gordon, M. *J. Chem. Phys.* **2009**, *131*, 174105–174113.

(12) Román-Pérez, G.; Soler, J. M. *Phys. Rev. Lett.* **2009**, *103*, 096102.

(13) Zhao, Y.; Truhlar, D. G. *J. Chem. Theory Comput.* **2006**, *3*, 289–300.

(14) Kossmann, S.; Neese, F. *J. Chem. Theory Comput.* **2010**, *6*, 2325–2338.

(15) Grimme, S. *J. Chem. Phys.* **2003**, *118*, 9095–9102.

(16) Gerenkamp, M.; Grimme, S. *Chem. Phys. Lett.* **2004**, *392*, 229–235.

(17) Jung, Y.; Lochan, R. C.; Dutoi, A. D.; Head-Gordon, M. *J. Chem. Phys.* **2004**, *121*, 9793–9802.

(18) Lochan, R. C.; Head-Gordon, M. *J. Chem. Phys.* **2007**, *126*, 164101–164111.

(19) Ishimura, K.; Pulay, P.; Nagase, S. *J. Comput. Chem.* **2006**, *27*, 407–413.

(20) Lambrecht, D. S.; Doser, B.; Ochsenfeld, C. *J. Chem. Phys.* **2005**, *123*, 184102–184111.

(21) Doser, B.; Lambrecht, D. S.; Kussmann, J.; Ochsenfeld, C. *J. Chem. Phys.* **2009**, *130*, 064107−064114.

(22) Schutz, M.; Werner, H.-J.; Lindh, R.; Manby, F. R. *J. Chem. Phys.* **2004**, *121*, 737–750.

(23) Bartlett, R. J.; Musial, M. *Rev. Mod. Phys.* **2007**, *79*, 291–262.

(24) Shavitt, I.; Bartlett, R. J. *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*; Cambridge University Press: New York, 2009.

(25) Janowski, T.; Ford, A. R.; Pulay, P. *J. Chem. Theory Comput.* **2007**, *3*, 1368–1377.

(26) Pople, J. A.; Head-Gordon, M.; Raghavachari, K. *J. Chem. Phys.* **1987**, *87*, 5968–5975.

(27) Lee, T. J.; Rendell, A. P.; Taylor, P. R. *J. Phys. Chem.* **1990**, *94*, 5463–5468.

(28) Janowski, T.; Pulay, P. *J. Chem. Theory Comput.* **2008**, *4*, 1585–1592.

(29) Neese, F.; Hansen, A.; Wennmohs, F.; Grimme, S. *Acc. Chem. Res.* **2009**, *42*, 641–648.

(30) Meyer, W. *Int. J. Quantum Chem.* **1971**, *S5*, 341–348.

(31) Meyer, W. *J. Chem. Phys.* **1973**, *58*, 1017–1035.

(32) Meyer, W. *Theor. Chim. Acta* **1974**, *35*, 277–292.

(33) Ahlrichs, R.; Driessler, F.; Lischka, H.; Staemmler, V.; Kutzelnigg, W. *J. Chem. Phys.* **1975**, *62*, 1235–1247.

(34) Ahlrichs, R.; Driessler, F. *Theor. Chim. Acta* **1975**, *36*, 275–287.

(35) Schutz, M.; Hetzer, G.; Werner, H.-J. *J. Chem. Phys.* **1999**, *111*, 5691–5705.

(36) Hampel, C.; Werner, H.-J. *J. Chem. Phys.* **1996**, *104*, 6286–6297.

(37) Mata, R. A.; Werner, H.-J. *J. Chem. Phys.* **2006**, *125*, 184110–184119.

(38) Maslen, P. E.; Head-Gordon, M. *Chem. Phys. Lett.* **1998**, *283*, 102–108.

(39) Subotnik, J. E.; Sodt, A.; Head-Gordon, M. *J. Chem. Phys.* **2006**, *125*, 074116−074122.

(40) Maslen, P. E.; Lee, M. S.; Head-Gordon, M. *Chem. Phys. Lett.* **2000**, *319*, 205–212.

(41) Scuseria, G. E.; Ayala, P. Y. *J. Chem. Phys.* **1999**, *111*, 8330–8343.

(42) Ayala, P. Y.; Scuseria, G. E. *J. Comput. Chem.* **2000**, *21*, 1524–1531.

(43) Venkatnathan, A.; Szilva, A. B.; Walter, D.; Gdanitz, R. J.; Carter, E. A. *J. Chem. Phys.* **2004**, *120*, 1693–1704.

(44) Walter, D.; Venkatnathan, A.; Carter, E. A. *J. Chem. Phys.* **2003**, *118*, 8127–8139.

(45) Auer, A. A.; Nooijen, M. *J. Chem. Phys.* **2006**, *125*, 024104−024114.

(46) Edmiston, C.; Krauss, M. *J. Chem. Phys.* **1965**, *42*, 1119–1120.

(47) Meyer, W. *Configuration Expansion by Means of Pseudonatural Orbitals. In Methods of Electronic Structure Theory*; Schaefer, H. F., III, Ed.; Plenum Press: New York, 1977; Vol. 3, pp 413−445.

(48) Neese, F.; Wennmohs, F.; Hansen, A. *J. Chem. Phys.* **2009**, *130*, 114108–114118.

(49) Neese, F.; Hansen, A.; Liakos, D. G. *J. Chem. Phys.* **2009**, *131*, 064103−064115.

(50) Pulay, P.; Saebo, S.; Meyer, W. *J. Chem. Phys.* **1984**, *81*, 1901–1905.

(51) Scuseria, G. E.; Janssen, C. L.; Schaefer III, H. F. *J. Chem. Phys.* **1988**, *89*, 7382–7387.

(52) Scuseria, G. E.; Schaefer III, H. F. *J. Chem. Phys.* **1989**, *90*, 3700–3703.

(53) Message Passing Interface Forum. www.mpi-forum.org (accessed Oct 15, 2010).

(54) Neese, F.; Becker, U.; Ganyushin, D.; Hansen, A.; Liakos, D. G.; Kollmar, C.; Kossmann, S.; Petrenko, T.; Reimann, C.; Riplinger, C.; Sivalingam, K.; Valeev, E.; Wezisla, B.; Wennmohs, F. *ORCA*; University of Bonn: Bonn, Germany, 2009.

(55) Vahtras, O.; Almlöf, J.; Feyereisen, M. W. *Chem. Phys. Lett.* **1993**, *213*, 514–518.

(56) Neese, F.; Schwabe, T.; Grimme, S. *J. Chem. Phys.* **2007**, *126*, 124115–124115.

(57) Tewari, A. K.; Srivastava, P.; Puerta, C.; Valegra, P. *J. Mol. Struct.* **2009**, *921*, 251–254.

(58) Schäfer, A.; Huber, C.; Ahlrichs, R. *J. Chem. Phys.* **1994**, *100*, 5829–5835.

(59) Weigend, F.; Ahlrichs, R. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3297–3305.

(60) Barker, J. K.; Davis, K.; Hoisie, A.; Kerbyson, J. D.; Lang, M.; Pakin, S.; Carlos, S. J. *In IEEE LSPP'08, LA-UR 07−6855*, 2008.

(61) Barker, J. K.; Davis, K.; Hoisie, A.; Kerbyson, J. D.; Lang, M.; Pakin, S.; Carlos, S. J. *Parallel Proc. Lett.* **2008**, *18*.

(62) Neese, F.; Wennmohs, F.; Hansen, A.; Becker, U. *Chem. Phys.* **2009**, *356*, 98–109.

(63) Muller-Dethlefs, K.; Hobza, P. *Chem. Rev.* **2000**, *100*, 143–168.

(64) Sponer, J.; Riley, K. E.; Hobza, P. *Phys. Chem. Chem. Phys.* **2008**, *10*, 2595.

(65) Stone, A. J. *The Theory of Intermolecular Forces*; Oxford University Press: Oxford, 1997.

(66) Janowski, T.; Pulay, P. *Chem. Phys. Lett.* **2007**, *447*, 27–32.

(67) Janowski, T.; Ford, A. R.; Pulay, P. *Mol. Phys.* **2010**, *108*, 249–257.

(68) Shields, A. E.; van Mourik, T. *J. Phys. Chem. A* **2007**, *111*, 13272–13277.

(69) Goerigk, L.; Grimme, S. *J. Chem. Theory Comput.* **2009**, *6*, 107–126.

(70) Valdes, H.; Pluhackova, K.; Pitonak, M.; Rezac, J.; Hobza, P. *Phys. Chem. Chem. Phys.* **2007**, *10*, 2747–2757.

(71) Zhao, Y.; Truhlar, D. G. *J. Phys. Chem. C* **2008**, *112*, 4061–4067.

(72) Goerigk, L.; Grimme, S. *J. Chem. Theory Comput.* **2009**, *6*, 107–126.

(73) Grimme, S.; Mück-Lichtenfeld, C.; Würthwein, E.-U.; Ehlers, A. W.; Goumans, T. P. M.; Lammertsma, K. *J. Phys. Chem. A* **2006**, *110*, 2583–2586.

(74) Jurecka, P.; Sponer, J.; Cerny, J.; P., H. *Phys. Chem. Chem. Phys.* **2006**, *8*, 1985–1993.

(75) Marchetti, O.; Werner, H.-J. *J. Phys. Chem. A* **2009**, *113*, 11580–11585.

(76) Takatani, T.; Hohenstein, E. G.; Malagoli, M.; Marshall, M. S.; Sherrill, C. D. *J. Chem. Phys.* **2010**, *132*, 144104–5.

(77) Helgaker, T.; Klopper, W.; Koch, H.; Noga, J. *J. Chem. Phys.* **1997**, *106*, 9639–9646.

(78) Dunning, J. T. H. *J. Chem. Phys.* **1989**, *90*, 1007–1023.

(79) Perdew, J. P. *Phys. Rev. B* **1986**, *33*, 8822.

(80) Becke, A. D. *Phys. Rev. A* **1988**, *38*, 3098.

(81) Becke, A. D. *J. Chem. Phys.* **1993**, *98*, 5648–5652.

(82) Stephens, P. J.; Devlin, F. J.; Chabalowski, C. F.; Frisch, M. J. *J. Phys. Chem.* **1994**, *98*, 11623–11627.

(83) Lee, C.; Yang, W.; Parr, R. G. *Phys. Rev. B* **1988**, *37*, 785.

(84) Prof. Stefan Grimme Research Web Site. http://www.uni-muenster.de/Chemie.oc/grimme/en/index.html. (accessed July 6, 2010).

(85) Grimme, S. *J. Chem. Phys.* **2006**, *124*, 034108−034116.

(86) Chai, J.-D.; Head-Gordon, M. *Phys. Chem. Chem. Phys.* **2008**, *10*, 6615–6620.

(87) Seth, S. K.; Hazra, D. K.; Mukherjee, M.; Kar, T. *J. Mol. Struct.* **2009**, *936*, 277–282.

(88) Schaefer, A.; Huber, C.; Ahlrichs, R. *J. Chem. Phys.* **1994**, *100*, 5829–5835.

CT100445S