



Large-scale correlated electronic structure calculations: the RI-MP2 method on parallel computers

David E. Bernholdt¹, Robert J. Harrison²

Environmental Molecular Sciences Laboratory, Pacific Northwest Laboratory, P.O. Box 999, Richland, WA 99352-0999, USA

Received 23 October 1995; in final form 15 January 1996

Abstract

The approximate 'resolution of the identity' second-order many-body perturbation theory method (RI-MP2) recently introduced by Feyereisen, Fitzgerald and Komornicki utilizes a combination of two- and three-center integrals to approximate the usual four-center two-electron repulsion integrals. Like the exact MP2, the overall cost of the RI-MP2 method scales with the fifth power of the number of basis functions, however the balance of the work shifts in such a way as to make the RI-MP2 method particularly well suited for implementation on massively parallel computers. We describe such an implementation and examine its parallel performance for several chemical systems. We are able to accurately reproduce the exact MP2 binding energy of K⁺ to 12-crown-4 ether in roughly 5% of the time.

1. Introduction

The so-called 'resolution of the identity' (RI) approach has been exploited in several recent papers to simplify the calculation and processing of two-electron integrals in the MP2 [1], SCF [2], and CCSD(T) [3] methods. The RI approach approximates the four-center two-electron repulsion integrals,

$$(pq|rs) = \int \chi_p(r_1)\chi_q(r_1) \frac{1}{|r_1 - r_2|} \times \chi_r(r_2)\chi_s(r_2) d^3r_1 d^3r_2, \quad (1)$$

with combinations of two- and three-center one- and two-electron integrals. It is very closely related to the fitting of the density commonly used in density func-

tional theory [4,5] and to the multiplicative integral approximation of Van Alsenoy [6].

Pairwise products of the atomic orbital (AO) basis functions, $\{\chi_p\chi_q\}$, are replaced by linear combinations of a 'fitting' basis, $\{\alpha_i\}$. If $\{\alpha_i\}$ is the Gaussian product theorem (GPT) result of the AO basis $\{\chi_p\}$ with itself, the substitution is exact, while smaller fitting bases result in approximations. The success of this approach depends on the observation that the GPT-AO space is usually nearly linearly dependent and can be spanned (or nearly so) by a smaller set of functions [7,8]. Such an approximation should work particularly well for calculations involving large atomic basis sets, which provide more redundancy in the GPT-AO space and at the same time would provide little sparsity that might normally be exploited to reduce the computational effort.

If used for all integrals, the RI approximation reduces the amount of data that must be manipulated from the $O(N^4)$ four-center two-electron integrals

¹ Current address: Northeast Parallel Architectures Center, 111 College Place, Syracuse University, Syracuse, NY 13244-4100, USA. E-Mail: bernhold@npac.syr.edu

² E-Mail: rj.harrison@pnl.gov

(for N AO basis functions) to just $\mathcal{O}(N^2 N_{\text{fit}})$ quantities. Thus as long as the size of the fitting basis, N_{fit} , is kept to less than N^2 , the RI-MP2 will involve fewer integrals than the exact MP2. Because this method is only recently introduced, little experience has been gained in choosing fitting basis sets, but it has been suggested that $N_{\text{fit}} \approx (2-5) \times N$ might provide adequate results [9,10], and clearly even $N_{\text{fit}} \approx (10-20) \times N$ offers excellent potential for the RI-MP2 approach for N of at least 100.

The RI approach, and RI-MP2 in particular, is especially well suited for use on modern massively parallel processors (MPPs). Current MPP machines typically offer rather limited memory per processor, and low aggregate I/O bandwidth, both of which can be very restrictive to traditional correlated methods, which must manipulate large amounts of data. Utilizing RI approximations, the amount of I/O that must be performed in a disk-based implementation can be reduced by an order of magnitude or more, and algorithms can be designed to require a limited amount of memory.

In this Letter, we describe a parallel algorithm for the RI-MP2 method suitable for both RHF and UHF references and demonstrate the scalability of this approach. Our algorithm is formulated to use as little as $\mathcal{O}(N^2)$ aggregate memory, and $\mathcal{O}(N^3)$ disk storage, putting it closer to an SCF calculation than a traditional MP2 in terms of resource requirements. Finally, we calculate the binding energy of K^+ to 12-crown-4 ether using a basis of more than 400 functions as a demonstration of the speed and efficiency of the method relative to the exact MP2.

2. Theory

As stated above, the essence of the RI approach is the replacement of a basis of pairs of AO basis functions with a different fitting basis. Several criteria can be chosen for optimizing the fitting basis representation of the GPT-AO space, resulting in different expansions of the four-center ERIs [2],

$$(pq|rs) = \sum_{t,u,v,w} (pqt) S_{tu}^{-1} V_{uv} S_{vw}^{-1} (wrs), \quad (2)$$

$$(pq|i;s) = \sum_{t,u} (pqt) S_{tu}^{-1} (u|rs), \quad (3a)$$

$$(pq|rs) = \sum_{t,u} (pqt) S_{tu}^{-1} (urs), \quad (3b)$$

$$(pq|rs) = \sum_{t,u} (pqt) V_{tu}^{-1} (u|rs), \quad (4)$$

known as the SVS, S, and V approximations respectively. Observe that all three approximations have the same basic form: a two-index quantity multiplied on both sides by three-index objects. To avoid redundant work later on when the integrals are processed, the two-index quantity can be subsumed into the three-center integrals in a transformation step, and any of the approximations above can be represented schematically as

$$(pq|rs) = \sum_{t'} I_{pqt'} J_{t'rs}. \quad (5)$$

For the SVS and V approximations, in which both of the three-index quantities are of the same type (overlaps for SVS, ERIs for V), the symmetry of the approximation can be maintained by using $(S^{-1}VS^{-1})^{1/2}$ or $V^{-1/2}$ to transform the fitting basis index of both of the three-index integrals [3]. Since I and J are identical in this case, this is a straightforward way to further reduce both the computational cost as well as the amount of data that must be manipulated.

The RI-MP2 equations are obtained by the straightforward insertion of Eq. (5) into the MP2 energy expression,

$$E^{(2)} = \frac{1}{2} \sum_{i,j,a,b} \frac{(ia|jb)[(ia|jb) - (ib|ja)]}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}. \quad (6)$$

The formula above is presented in spin-orbital form, following the usual notation³. For the case of a UHF reference, the resulting equations are

$$E_{\text{UHF}}^{(2)} = \frac{1}{2} \sum_{i,j,a,b,t'} \frac{I_{iat'} J_{t'jb} [I_{iat'} J_{t'jb} - I_{ibt'} J_{t'ja}]}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b} \quad (7a)$$

$$+ \frac{1}{2} \sum_{i,j,\bar{a},\bar{b},t'} \frac{I_{i\bar{a}t'} J_{t'\bar{b}} [I_{i\bar{a}t'} J_{t'\bar{b}} - I_{i\bar{b}t'} J_{t'\bar{a}}]}{\epsilon_i + \epsilon_j - \epsilon_{\bar{a}} - \epsilon_{\bar{b}}} \quad (7b)$$

$$+ \sum_{i,j,a,\bar{b},t'} \frac{I_{iat'} J_{t'\bar{b}} [I_{iat'} J_{t'\bar{b}}]}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_{\bar{b}}}, \quad (7c)$$

³ i and j represent occupied orbitals, a and b virtuals, and the ϵ_p are the canonical SCF eigenvalues.

where i, j, a, b represent α spin orbitals and $\bar{i}, \bar{j}, \bar{a}, \bar{b}$ β spin. The fitting basis indices, t' are independent of spin. It is worth noting that in the UHF RI-MP2, all of the three-center integrals are pure spin, and the mixed spin four-center integrals are approximated by the product of a pure- α integral with a pure- β integral. This means that the UHF RI-MP2 manipulates even less data relative to its exact counterpart than the RHF version. For RHF references, the three terms shown above collapse to a single one, since the α and β spin orbitals have the same spatial component, just as in the exact method.

The RI-MP2 approach has a different distribution of the computational costs compared to the exact MP2. Taking N_{fit} to be $\mathcal{O}(N)$, the $\mathcal{O}(N^5)$ four-index transformation is replaced by an $\mathcal{O}(N^4)$ three-index transformation (assuming the fitting basis is also transformed, as discussed above), while the $\mathcal{O}(N^4)$ exact energy evaluation now becomes $\mathcal{O}(N^5)$ due to the need to multiply I and J to form approximate four-center integrals. These differences are well suited to the strengths of modern parallel processors. Distributed matrix multiplication [11–13] performs much more efficiently than most parallel four-index transformation algorithms which have been developed [14–25], and the reduced number of integrals means disk-based implementations can be used much more readily than for the exact MP2.

3. Parallel implementation

Our parallel RI-MP2 implementation is based on a distributed globally-addressable non-uniform access memory programming model. Our 'global array tools' (GA) library [26–28] implements this model on a variety of shared memory and distributed memory MPPs. Basic GA routines provide asynchronous read/write operations on distributed (global) arrays and provide each processor with direct access to the local patch of the global array. Building on this base, the GA tools offer a variety of basic linear algebra routines modeled on the widely-used BLAS and LAPACK libraries as well as the capabilities of Fortran90.

The RI-MP2 algorithm maps quite straightforwardly to the GA model, both in the three-center integral transformation and in the energy evaluation. The three-index transformation, summarized in Fig.

1 is split into two parts, the first comprising the transformation of the two AO basis indices, and the second the transformation of the fitting basis index with the appropriate two-index matrix according to the approximation chosen. In the first part, each node generates the raw integrals $(\mu\nu|t)$ or $(uv|t)$ for the unique $\mu\nu$, and a subset of t , so that the complete space of $\{t\}$ is distributed across the machine. The integrals are directly transformed to the MO basis, $(ai|t)$ or (ait) and stored in a global array. No inter-processor communication is involved in this step, so it is completely parallel aside from possible load imbalance. The second part of the transformation is a distributed matrix multiplication operation, which is provided by the GA library. The transformed integrals can be stored in memory (given enough space) or written to disk. Disk I/O is also done in parallel, with each node writing its subset of the integrals to a private file. (Although machine-specific optimizations are possible for many platforms, file I/O is currently done using only standard Fortran.) Each record holds the local subset of t and all virtual orbitals a for a given occupied orbital i , consequently the integral file must be written and read by the same number of nodes. More general solutions are beyond the scope of standard Fortran.

As shown in Fig. 2, the energy evaluation is driven by an outer loop over pairs of occupied orbitals, ij . Integrals $I_{ai,t'}$ and $J_{bj,t'}$ are multiplied to produce a matrix A_{ab}^{ij} , which are the approximate four-center integrals for this particular ij . In the RHF case, and the pure spin UHF terms, the $(ia|jb) - (ib|ja)$ term can be constructed by taking $B^{ij} = A^{ij} - A^{ijT}$. Denominators are applied to B^{ij} , and then A^{ij} and B^{ij} are contracted in a dot product-like operation to produce the pair correlation energy. In this case, all matrices and operations are handled in distributed fashion using GA library routines. Application of the denominators to B^{ij} occurs in parallel fashion as well, each processor operating on the subset of the matrix which is held in local memory and accessed via the GA library. All steps following creation of the approximate four-center integral matrix A_{ab}^{ij} are mathematically identical to the exact method, and apart from differences in data distribution could be evaluated with the same code.

In all phases of the code, blocking or 'stripmining' is used to improve performance within the amount

Form $V_{ii'}^{-\frac{1}{2}}$

Determine block size for i (occupied orbital) based on available memory

Loop over blocks of i

Determine size of buffer available to accumulate raw integrals

Loop over buffer-loads of AO shell blocks $s_\mu s_\nu$

Fill buffer with integrals $(\mu\nu|t)$ (t distributed across processors)

$(a\nu|t) = C_{\mu a}(\mu\nu|t)$ (local matrix multiply)

$(ai|t) = C_{\nu i}(\mu\nu|t)$ (local matrix multiply)

End of $s_\mu s_\nu$ loop

$I_{aii'} = (ai|t') = (ai|t)V_{ii'}^{-\frac{1}{2}}$ (distributed matrix multiply)

Write $I_{aii'}$ block to disk

End of i loop

Fig. 1. RI-MP2 transformation algorithm. Although shown specifically for the 'V approximation', the algorithm remains substantially the same for the others. C are the SCF orbital coefficients.

Determine block sizes for i and j (occupied orbitals) based on available memory

Loop over unique blocks of ij

Read three-center integral blocks $I_{aii'}$ and $J_{bjj'}$ (local)

$A_{ab}^{ij} = I_{aii'} \times J_{bjj'}$ (distributed matrix multiply)

$B^{ij} = A^{ij} - A^{ijT}$ (data parallel)

Apply denominators to B^{ij} (data parallel)

$E_{ij} = A^{ij} \cdot B^{ij}$ (data parallel)

End of ij loop

Fig. 2. RI-MP2 energy evaluation algorithm.

of memory available. In the integral transformation shell triplets of raw integrals are collected into larger blocks before calls to BLAS matrix multiplication routines are made for the first two steps. Block dimensions are determined so as to maximize the contraction lengths of the matrix multiplications. The transformation is also capable of multi-pass operation for instances where there is insufficient memory for a single pass. Stripmining the energy evaluation loops allows the distributed matrix multiplication, which produces the approximate four-center integrals, to be made as

large as possible. Also, the number of times the transformed three-center integrals must be read from disk is determined by the number of blocks in the outer loop (over i). This can have an important impact on the performance of the code.

As a result of this flexibility with respect to memory usage, these algorithms can work with anywhere from $\mathcal{O}(N^2)$ memory and $\mathcal{O}(N^3)$ disk space to a fully in-core calculation requiring $\mathcal{O}(N^3)$ memory.

4. K⁺ Binding to 12-Crown-4 Ether

To demonstrate the potential of this method for a sizable calculation, we have chosen to study the binding of K⁺ to 12-crown-4 ether. The basis sets on H, C, and O are aug-cc-pVDZ [29,30] without the diffuse *s* of the hydrogen basis to prevent linear dependence problems (designated aug-cc-pVDZ'). The potassium basis has been designed in analogy with the Li and Na cc-pVDZ basis sets [31], and is 15s12p2d contracted to 6s5p2d [32]. Due to limitations of the current integral code in our parallel computational chemistry package, NWChem [33], the RI-MP2 calculations were carried out using the Cartesian representation of the *d* shells rather than the spherical harmonics in which the basis sets were originally optimized, resulting in AO basis sets of 428, 33, and 461 functions for the bare ether, K⁺, and the complex, respectively. For these systems, 12, 5, and 17 core orbitals were frozen in all correlated calculations. The corresponding exact-MP2 calculations were of course calculated in the same basis set. All calculations used exact MP2 optimized geometries obtained with the cc-pVDZ basis (using spherical harmonic *d* shells) [34].

The fitting basis was chosen to reproduce the exact MP2 binding energy of the cation to the crown ether ($-44.46 \text{ kcal mol}^{-1}$ at this geometry) to within $0.02 \text{ kcal mol}^{-1}$. The core of the fitting basis is the uncontracted version of the AO basis, which is augmented by additional primitive functions detailed in Table 1. These functions were chosen based on combinations of exponents of the uncontracted AO functions as suggested by the Gaussian Product Theorem. The result is fitting basis sets of size 1132, 95, and 1227 for the bare ether, cation, and complex. The resulting total energies, along with the reference exact MP2 results are given in Table 2.

There are several points that should be made with respect to the fitting basis. First, per our stated goal, the fitting basis reproduces the binding energy, not the individual energies. The individual total energies differ from the exact result by approximately $1-2 \text{ mE}_h$. Secondly, in the course of this work, and in separate efforts focusing explicitly on construction of fitting basis sets [10] it has become clear that quite a wide range of results can be obtained depending on the choice of fitting basis. This should not be too surprising given that changing the AO basis set in the exact method

can also effect the results. It is not yet clear whether the situation for RI fitting basis sets is better or worse than for the AO basis itself in the exact method, but there are indications that it will be possible to obtain reliable results with efficient (relatively small) fitting basis sets systematically derived from the chosen AO set [10]. Finally, we must note that using the unrefined AO basis as the fitting basis provides generally unsatisfactory results. This is significant because the majority of electronic structure codes are not designed to use more than one basis set in a given calculation, so a natural first implementation of RI methods in such codes would restrict the fitting basis to be identical to the AO basis. In summary, as with any newly introduced approximation, due caution must be exercised to insure that reliable results are obtained.

Once a suitable fitting basis is found, however, performance of the RI-MP2 method is quite good. Running on 70 nodes of the Kendall Square Research KSR-2, the RI-MP2 calculation on the K⁺:12-crown-4 complex required 1530 s compared to 29478 s for NWChem's direct exact MP2 [25]. Even on ten nodes, the smallest partition for which this calculation could be performed on the KSR, the RI-MP2 required only 9967 s. In a study of K⁺-ether complexes by Feller, Aprà, Nichols, and Bernholdt [35], K⁺:12-crown-4 calculations are extended to other geometries, and counterpoise corrections for basis set superposition error, made feasible by the performance of the RI-MP2 method, are included. Feller and co-workers conclude that RI-MP2, because of its ability to mimic the exact MP2 results, represents the binding energies of the complexes more reliably than density functional theory calculations.

5. Parallel performance

In order to gauge the parallel performance of the method, we have carried out calculations on K⁺:12-crown-4 using three massively parallel machines with different architectures, the Cray-T3D from Cray Research, the Intel Paragon, and Kendall Square Research KSR-2. Key features of these three machines are summarized in Table 3.

The parallel speedup for the K⁺:12-crown-4 calculation is shown in Fig. 3. As indicated by the error bars, repeated runs of the same job can give sizable

Table 1

Basis functions used to augment the uncontracted AO basis functions in the fitting basis set for the K^+ :12-crown-4 binding energy calculation

Atom	Exponents			
	s	p	d	f
H		1.454	1.454	
C	0.0235	0.0202	0.3034, 1.1, 2.5476	1.1
O	0.0395	0.343	0.5506, 2.370, 4.900	2.370
K	0.0091	0.0872	0.3490, 1.5000, 1.8080	1.8080

Table 2

Exact and RI-MP2 total energies for K^+ , 12-crown-4, and K^+ :12-crown-4. The AO and fitting basis sets are described in the text

Molecule	Core orbitals frozen	Exact MP2 (E_h)	RI-MP2 (E_h)
K^+	5	-599.172243376	-599.172691398
12-crown-4	12	-613.712329452	-613.713518948
K^+ :12-crown-4	17	-1212.955581872	-1212.957180496

Table 3

Key characteristics of the massively parallel processors used in this study. Communications performance is described by observed results of the basic global array *put* and *get* operations, taken from Ref. [28]

Characteristic		Cray-T3D	Intel Paragon	KSR-2
CPU		DEC Alpha	Intel i860XP	proprietary RISC
clock rate	(MHz)	150	50	40
memory per node	(MB)	64	32	32
interconnect topology		3d torus	2d mesh	hierarchy of rings
GA remote <i>put</i>				
latency	(μ s)	18	62	36
bandwidth	(MB/s)	110	20	29
GA remote <i>get</i>				
latency	(μ s)	35	497	37
bandwidth	(MB/s)	32	9.5	28

variations in timings depending on the load on the remainder of the machine and the architecture. The Cray-T3D exhibited minimal variation because the interconnect topology and restrictions to partitions of 2^n nodes means that jobs partitions do not share bandwidth on the interconnection network.

For all three platforms, the algorithm produces super-linear speedups over a large range of machine sizes. This is primarily due to the fact that each CPU applied to the problem brings with it a certain amount of local memory and the fact that the algorithm is designed to be scalable in both CPU and memory utilization. Stripmining of the transformation step results

in fewer passes as memory is increased, which means fewer evaluations of the three-center two-electron integrals. By efficiently using the additional memory as well as processors, the algorithm is able to eliminate work that is effectively redundant, resulting in additional speedup over that gained solely by applying more CPUs. This effect is typical of multi-pass algorithms in which a significant amount of computation must be repeated in each pass.

Timings of important components of the code, shown in Fig. 4, reveal several features of the present implementation. Although the transformation is formally N^4 and energy evaluation (dominated by the

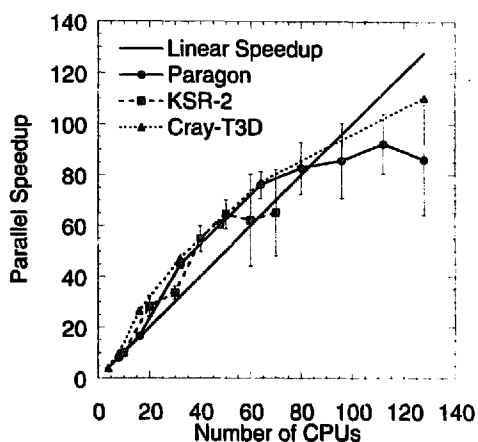


Fig. 3. Parallel speedup for the $K^+ : 12\text{-crown-4}$ RI-MP2 energy on three different MPP platforms. Error bars resulting from the uncertainty associated with the load on the remainder of the machine were obtained from analysis of repeated runs of the same job. Error bars for the Cray-T3D data are too small to be distinguished at this scale.

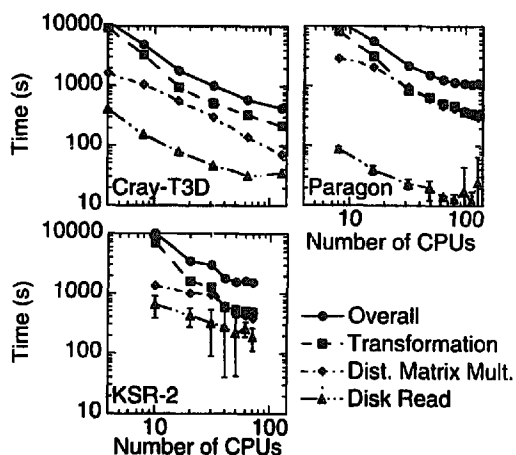


Fig. 4. Actual timings of important components of the RI-MP2 calculation on three MPP platforms. The distributed matrix multiplication is the dominant component of the energy evaluation step.

distributed matrix multiplication) is N^5 , the transformation is comparable to, or even more expensive than the energy evaluation for this size of system due to a larger prefactor in the N^4 step. Clearly although the importance of the transformation is reduced relative to the exact case, improvements in integral evaluation and the transformation itself will benefit the RI-MP2. Another feature displayed in Fig. 4 is the disk I/O demands of the method, characterized by the time

required to read the three-center integrals from disk during the energy evaluation. Despite rather different I/O performance on the three platforms, I/O is a relatively small component of the overall timing in general. At larger processor counts, the total I/O time (writing and reading integrals) is as much as 10–20% of the overall calculation as a consequence of the relatively poor scaling in the I/O performance observed on all three platforms. In a non-dedicated situation, there can also be substantial variation in timings of repeated calculations, as shown in the error bars on the KSR-2 and Paragon curves. A disk-based exact MP2 would require about an order of magnitude more data for this calculation.

6. Conclusions

We have described a parallel implementation of the RI-MP2 method of Feyereisen, Fitzgerald, and Komornicki and its application to the binding of K^+ to 12-crown-4 ether. The method is well-suited to the strengths of massively parallel computers, and maps quite straightforwardly onto a shared non-uniform-access memory programming model which is both efficient and portable.

The exact MP2 $K^+ : 12\text{-crown-4}$ binding energy can be reproduced to within $0.02 \text{ kcal mol}^{-1}$ by a fitting basis less than three times the size of the AO basis in roughly 5% of the time required by the exact calculation. The parallel performance of the implementation is also quite good, as we have demonstrated on MPPs with rather different architectures.

Acknowledgements

This work benefited from discussions with Jan Almlöf (University of Minnesota), David Feller (PNL), Jarek Nieplocha (PNL), and Alistair Rendell (Australian National University). We also appreciate Rick Kendall's critical reading of the manuscript. Some of the calculations reported here were performed at the Center for Computational Science (CCS) at Oak Ridge National Laboratory and at the National Energy Research Supercomputer Center (NERSC) at Lawrence Livermore National Laboratory. Access to the CCS Paragon was provided by

the High Performance Computing and Communications Initiative Grand Challenge Program. Access to the CRAY-T3D was provided by the MPP Access Program at NERSC. Support for this work was provided by the Environmental and Molecular Sciences Laboratory project managed by the Office of Energy Research, and by the High Performance Computing and Communications Initiative Grand Challenge Program of the Division of Mathematical, Information, and Computational Sciences, Office of Computer and Technology Research, U. S. Department of Energy under contract DE-AC06-76RLO 1830 with Battelle Memorial Institute which operates the Pacific Northwest laboratory, and contract DE-FG06-89ER-75522 with Associated Western Universities, Inc., Northwest Division.

References

- [1] M. Feyereisen, G. Fitzgerald and A. Komornicki, *Chem. Phys. Letters* 208 (1993) 359.
- [2] O. Vahtras, J. Almlöf and M.W. Feyereisen, *Chem. Phys. Letters* 213 (1993) 514.
- [3] A.P. Rendell and T.J. Lee, *J. Chem. Phys.* 101 (1994) 400.
- [4] B.I. Dunlap, J.W.D. Connolly and J.R. Sabin, *J. Chem. Phys.* 71 (1979) 3396.
- [5] J.W. Mintmire and B.I. Dunlap, *Phys. Rev. A* 25 (1982) 88.
- [6] C.V. Alsenoy, *J. Comput. Chem.* 9 (1988) 620.
- [7] N.H.F. Beebe and J. Linderberg, *Intern. J. Quantum Chem.* 12 (1977) 683.
- [8] D.W. O'Neal and J. Simons, *Intern. J. Quantum Chem.* 34 (1989) 673.
- [9] J. Almlöf, private communication.
- [10] D.E. Bernholdt and R.J. Harrison, to be published.
- [11] J. Choi, J.J. Dongarra and D.W. Walker, PUMMA: Parallel universal matrix multiplication algorithms on distributed memory concurrent computers, Technical Report ORNL/TM-12252, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA, 1993 (URL <http://www.netlib.org/scalapack/pumma.ps>).
- [12] S. Huss-Lederman, E.M. Jacobson, A. Tsao and G. Zhang, *Concurrency: Practice and Experience* 6 (1994) 571.
- [13] R.A. van de Geijn and J. Watts, SUMMA: Scalable universal matrix multiplication algorithm, Technical Report TR-95-13, Department of Computer Sciences, Univ. of Texas, Austin, TX, USA, 1995 (URL [file://net.cs.utexas.edu/pub/techreports/tr95-13.ps](http://net.cs.utexas.edu/pub/techreports/tr95-13.ps)).
- [14] E. Clementi, S. Chin and D. Logan, *Lect. Notes Chem.* 44 (1986) 130.
- [15] R.A. Whiteside, J.S. Binkley, M.E. Colvin and H.F. Schaefer III, *J. Chem. Phys.* 86 (1987) 2185.
- [16] J.N. Hurley, D.L. Huestis and W.A. Goddard III, *J. Phys. Chem.* 92 (1988) 4880.
- [17] C.W. Bauschlicher Jr., *Theoret. Chim. Acta* 76 (1989) 187.
- [18] J.D. Watts and M. Dupuis, Vector and parallel implementations of a fourth-order Moller-Plesset perturbation theory (MP4) program, Technical Report KGN-197, IBM Corporation, Data Systems Division, Kingston, NY 12401, USA, 1989.
- [19] L.A. Covick and K.M. Sando, *J. Comput. Chem.* 11 (1990) 1151.
- [20] R. Wiest, J. Demuyneck, M. Bénard, M. Rohmer and R. Ernenwein, *Computer Phys. Commun.* 62 (1991) 107.
- [21] R.J. Harrison, Ab initio chemistry in parallel, Technical Report CCSF-14-92, Caltech Concurrent Supercomputing Facility, Pasadena, CA, IUSA, 1992.
- [22] A.C. Limaye and S.R. Gadre, *J. Chem. Phys.* 100 (1994) 1303.
- [23] M. Dupuis, S. Chin and A. Marquez, Modern tools for including electron correlation in electronic structure studies: HONDO and CHEM-STATION, in: *Relativistic and electronic correlation effects in molecules and clusters*, ed. G.L. Malli, NATO ASI Series B (Physics), Vol. 318 (Plenum Press, New York, 1994) pp. 315-338.
- [24] T.L. Windus, M.E. Schmidt and M.S. Gordon, *Theoret. Chim. Acta* 89 (1994) 77.
- [25] A.T. Wong, R.J. Harrison and A.P. Rendell, *Theoret. Chim. Acta*, in press.
- [26] J. Nieplocha and R.J. Harrison, Global array tools library, available for anonymous FTP via URL <ftp://ftp.pnl.gov/pub/global>.
- [27] J. Nieplocha, R.J. Harrison and R.J. Littlefield, Global arrays: A portable 'shared-memory' programming model for distributed memory computers, in *Supercomputing'94*, Los Alamitos, CA, USA, 1994, Institute of Electrical and Electronics Engineers and Association for Computing Machinery, IEEE Computer Society Press.
- [28] J. Nieplocha, R.J. Harrison and R.J. Littlefield, *J. Supercomputing*, submitted.
- [29] T.H. Dunning Jr., *J. Chem. Phys.* 90 (1989) 1007.
- [30] R.A. Kendall, T.H. Dunning Jr. and R.J. Harrison, *J. Chem. Phys.* 96 (1992) 6796.
- [31] D.E. Woon and T.H. Dunning Jr., to be published.
- [32] D. Feller, E.D. Glendening, D.E. Woon and M.W. Feyereisen, *J. Chem. Phys.* 103 (1995) 3526.
- [33] High Performance Computational Chemistry Group, Pacific Northwest Laboratory, Richland, Washington 99352, USA, NWChem, a computational chemistry package for parallel computers, version 1.0, 1994.
- [34] D. Feller, private communication.
- [35] D. Feller, A. Aprà, J.A. Nichols and D.E. Bernholdt, The structure and binding energy of K⁺-ether complexes: A comparison of MP2, RI-MP2 and density functional methods, in preparation.