Development of Efficient Algorithms for Quantum Chemistry Calculations of Large Molecules

Kazuya Ishimura

2007

Contents

General Introduction

Quantum chemistry plays an important role in elucidating molecular geometries, electronic states, and reaction mechanisms, because of the developments of a variety of theoretical methods, such as Hartree-Fock (HF), Møller-Plesset (MP) perturbation, configuration interaction (CI), coupled-cluster (CC), and density functional theory (DFT) methods.[1] Electronic structure calculations have been carried out by not only theoretical chemists but also experimental chemists. DFT is currently most widely used to investigate large molecules in the ground state as well as small molecules because of the low computational cost. However, the generally used functionals fail to describe correctly non-covalent interactions that are important for host-guest molecules, self-assembly, and molecular recognition, and they tend to underestimate reaction barriers.[2] Many attempts have been made to develop new functionals and add semiempirical or empirical correction terms to standard functionals, but no generally accepted DFT method has emerged yet.

Second-order Møller-Plesset perturbation theory (MP2)[3] is the simplest method that includes electron correlation important for non-covalent interactions and reaction barriers within the single determinant model. However, the computational cost of MP2 is considerably higher than that of DFT. In addition, much larger sizes of fast memory and hard disk are required in MP2 calculations. These make MP2 calculations increasingly difficult for larger molecules. Since workstation or personal computer (PC) clusters have become popular for quantum chemistry calculations, an efficient parallel calculation is a solution of the problem. Therefore, new parallel algorithms for MP2 energy and gradient calculations are presented in this thesis. Furthermore, an efficient algorithm for the generation of two-electron repulsion integrals (ERIs) which is important in quantum chemistry calculations is also presented.

For the calculations of excited states, different approaches are required: for

example, CI, multi-configuration self-consistent field (MCSCF), time-dependent DFT (TDDFT), and symmetry adapted cluster (SAC)/SAC-CI methods.[4,5] One of the most accurate methods is SAC/SAC-CI, as demonstrated for many molecules. In this thesis, SAC/SAC-CI calculations of ground, ionized, and excited states are presented.

This thesis consists of five chapters: a new algorithm of two-electron repulsion integral calculations (Chapter I), a new parallel algorithm of MP2 energy calculations (Chapter II), a new parallel algorithm of MP2 energy gradient calculations (Chapter III), applications of MP2 calculations (Chapter IV), and SAC/SAC-CI calculations of ionized and excited states (Chapter V).

In quantum chemistry calculations, the generation of ERIs is one of the most basic subjects and is the most time-consuming step especially in direct SCF calculations. Many algorithms have been developed to reduce the computational cost. In Pople-Hehre algorithm,[6] Cartesian axes are rotated to make several coordinate components zero or constant, so that these components are skipped in the generation of ERIs. In McMurchie-Davidson algorithm,[7] ERIs are generated from (*ss*|*ss*) type integrals using a recurrence relation derived from Hermite polynomials. By combining these two algorithms, a new algorithm is developed in Chapter I.

In quantum mechanics, perturbation methods can be used for adding corrections to reference solutions. In the MP perturbation method, a sum over Fock operators is used as the reference term, and the exact two-electron repulsion operator minus twice the average two-electron repulsion operator is used as the perturbation term. It is the advantage that the MP perturbation method is size consistent and size extensive, unlike truncated CI methods. The zero-order wave function is the HF Slater determinant, and the zero-order energy is expressed as a sum of occupied molecular orbital (MO) energies. The first-order perturbation is the correction for the overcounting of two-electron repulsions at zero-order, and the first-order energy corresponds to the HF energy. The MP correlation starts at second-order. In general, second-order (MP2)

accounts for 80 – 90% of electron correlation. Therefore, MP2 is focused in this thesis since it is applicable to large molecules with considerable reliability and low computational cost.

The formal computational scaling of MP2 energy calculations with respect to molecular size is fifth order, much higher than that of DFT energy calculations. Therefore, less expensive methods, such as Local MP2,[8] density fitting (resolution of identity, RI) MP2,[9-12] and Laplace Transform MP2,[13-16] have been developed. However, all of these methods include approximations or cut-offs that need to be checked against full MP2 energies. An alternative approach to reduce the computational cost is to parallelize MP2 energy calculations. A number of papers on parallel MP2 energy calculations have been published. Almost all of them are based on simple parallelization methods that distribute only atomic orbital (AO)[17] or MO[18] indices to each processor. These methods have a disadvantage since intermediate integrals are broadcasted to all CPUs or the same AO ERIs are generated in all processors. Baker and Pulay[19,20] developed a new parallel algorithm using Saebø-Almlöf integral transformation method.[21] This algorithm parallelizes the first half transformation by AO indices and the second half transformation by MO indices. The advantages are that the total amount of network communication is independent of the number of processors and the AO integrals are generated only once. The disadvantage is the I/O overhead for the sorting of half-transformed integrals. A new parallel algorithm for MP2 energy calculations based on the two-step parallelization idea is presented in Chapter II. In this algorithm, AO indices are distributed in the AO integral generation and the first three quarter transformation, and MO indices are distributed in the last quarter transformation and MP2 energy calculation. Because the algorithm makes the sorting of intermediate integrals very simple, the parallel efficiency is highly improved and the I/O overhead is removed. Furthermore, the algorithm reduces highly the floating point operation

(FLOP) count as well as the required memory and hard disk space, in comparison with other algorithms.

Determination of molecular geometries and reaction paths is a fundamental task in quantum chemistry and requires energy gradients with respect to nuclear coordinates. In Chapter III, a new parallel algorithm for MP2 energy gradient calculations is presented. The algorithm consists of 5 steps, the integral transformation, the MP2 amplitude calculation, the MP2 Lagrangian calculation, the coupled-perturbed HF calculation, and the integral derivative calculation. All steps are parallelized by distributing AO or MO indices. The algorithm also reduces the FLOP count, the required memory, and hard disk space.

In Chapter IV, several applications of MP2 are performed using the program developed in Chapters II and III. Some molecules that DFT cannot treat well are optimized at the MP2 level. Geometry optimization is also carried out using the spin-component scaled (SCS) MP2 method.[22] In this method, a different scaling is employed for the same and opposite spin components of the MP2 energy, so that SCS-MP2 performs as well as the much more costly CCSD(T) method at a high level of theory.

SAC theory is developed for ground states and based on CC theory that describes higher-order electron correlation. The main factor of electron correlation is collisions of two electrons. In CC theory, most collisions of four electrons can be taken in as the product of collisions of two electrons. Only a symmetry adapted excitation operator is used for the SAC expansion. Since the operator of the SAC expansion is totally symmetric, the unlinked terms (the products of the operators) are also totally symmetric. SAC-CI is developed to treat excited states. SAC and SAC-CI wave functions are orthogonal and Hamiltonian-orthogonal to each other. These orthogonalities are especially important for the calculations of transitions and relaxations. In general, the SAC-CI operators R are restricted to single and double excitations. This is called the

SAC-CI SD-R method. For the calculations of high-spin states and multiple excitation processes, triple, quadruple, and higher excitation operators are included. This is called the SAC-CI general-R method. In Chapter V, the ground, singlet and triplet excited, ionized and electron attached states of ferrocene ($Fe(C_5H_5)_2$) are calculated using the SAC/SAC-CI SD-R method. The assignment of *d* electron ionizations and the difference of singlet and triplet *d-d* transitions are discussed.

References

1. 永瀬茂, 平尾公彦, “分子理論の展開”, 岩波書店, 2002.
2. Schreiner, P. R. Angew Chem Int Ed 2007, 46, 4217.
3. Møller, C.; Plesset, M. S. Phys Rev 1934, 46, 618.
4. Nakatsuji, H.; Hirao, K. J Chem Phys 1978, 68, 2053.
5. Nakatsuji, H. Chem Phys Lett 1978, 59, 362; 1979, 67, 329; 1979, 67, 334.
6. Pople, J. A.; Hehre, W. J. J Comput Phys 1978, 27, 161.
7. McMurchie, L. E.; Davidson, E. R. J Comput Phys 1978, 26, 218.
8. Saebø, S.; Pulay, P. Ann Rev Phys Chem 1993, 44, 213.
9. Feyereisen, M.; Fitzgerald, G.; Komornicki, A. Chem Phys Lett 1993, 208, 359.
10. Vahtras, O.; Almlof, J.; Feyereisen, M. W. Chem Phys Lett 1993, 213, 514.
11. Bernholdt, D. E.; Harrison, R. J. Chem Phys Lett 1996, 250, 477.
12. Weigend, F.; Haser, M.; Patzelt, H.; Ahlrichs, R. Chem Phys Lett 1998, 294, 143.
13. Almlöf, J. Chem Phys Lett 1991, 181, 319.
14. Haser, M.; Almlöf, J. J Chem Phys 1992, 96, 489.
15. Wilson, A. K.; Almlöf, J. Theor Chem Acc 1997, 95, 49.
16. Ayala, P. Y.; Scuseria, G. E. J Chem Phys 1999, 110, 3660.
17. Nielsen, I. M. B.; Seidl, E. T. J Comput Chem 1995, 16, 1301.

18. Schütz, M.; Lindh, R. Theoret Chim Acta 1997, 95, 13.

19. Pulay, P.; Saebø, S.; Wolinski, K. Chem Phys Lett 2001, 344, 543.

20. Baker, J.; Pulay, P. J Comput Chem 2002, 23, 1150.

21. Saebø, S.; Almlöf, J. Chem Phys Lett 1989, 154, 83.

22. Grimme, S. J Chem Phys 2003, 118, 9095.

Chapter I

A new algorithm of two-electron repulsion integral calculations

(Ishimura, K.; Nagase, S. Theoret Chem Acc in press.)

1.1 Abstract

A new algorithm of two-electron repulsion integral (ERI) calculations has been developed. In this algorithm, Cartesian axes are rotated to make several coordinate components zero or constant using the Pople-Hehre algorithm, and ERIs are evaluated at the rotated coordinate by the McMurchie-Davidson algorithm. The new algorithm applicable to (*ss*|*ss*) to (*dd*|*dd*) ERIs is implemented in the quantum chemistry program GAMESS. Test calculations show that the new algorithm reduces the computational times by 10-40%, as compared with the Pople-Hehre and the Rys quadrature algorithms.

1.2 Introduction

In quantum chemistry, it is an important subject to calculate two-electron repulsion integrals (ERIs) at a high speed, because ERIs are fundamental in all *ab initio* self-consistent field (SCF) and post-SCF calculations. The generation of ERIs is especially time-consuming in widely used direct SCF calculations because ERIs are generated in every SCF iteration. Since Boys proposed the use of Gaussian functions as basis functions in 1950,[1] many algorithms have been developed to reduce the computational time.[2-17]

In the Pople-Hehre algorithm,[2] Cartesian axes are rotated so that several coordinate components become zero or constant. The Floating-Point Operation (FLOP) count of the fourth and second power terms is reduced. The Rys quadrature developed by Dupuis, Rys, and King[3-5] is based on a set of orthogonal polynomials and is used especially for ERIs including high angular momentum functions. In the McMurchie-Davidson algorithm,[6] the efficient recurrence relation derived from Hermite polynomials was adopted to generate all ERIs from (*ss*|*ss*) type integrals. Obara and Saika[7] developed an efficient algorithm of a vertical recurrence relation (VRR) in which required ERIs are generated from actually needed and auxiliary ERIs of lower angular momentum functions. Head-Gordon and Pople (HGP)[8] derived a horizontal recurrence relation (HRR) to make several terms in VRR always vanish. Gill, Head-Gordon, and Pople[9] derived another relation by combining the MD and HGP algorithms. Lindh, Ryu, and Liu[10] derived an algorithm from the Rys quadrature and HRR. Ishida[11] developed the accompanying coordinate expansion (ACE) algorithm based on the Rys quadrature, which can treat up to (*hh*|*hh*) ERIs.[12] A different algorithm was proposed by Gill and Pople, and named PRISM.[13] In PRISM, the path from $(ss|ss)^{(m)}$ integrals to required ERIs is selected to minimize the FLOP count. The bra and ket transformations in the

algorithms such as MD and HGP as well as the bra and ket contractions depend on the degrees of bra and ket contractions of the primitive functions.

In this paper, we report a new algorithm of ERI calculations by combining the PH and MD algorithms because MD can use all advantages of PH. The new algorithm is considerably faster than the original PH and Rys, as demonstrated by test calculations.

1.3 Methods

An unnormalized primitive Gaussian function $\varphi_a(\mathbf{r})$ with the exponent α at point **A** is given by

$$\varphi_a(\mathbf{r}) = (x - A_x)^{a_x} (y - A_y)^{a_y} (z - A_z)^{a_z} \exp\left[-\alpha_A (\mathbf{r} - \mathbf{A})^2\right]. \quad (1)$$

A primitive ERI can be written by

$$[\mathbf{ab} \mid \mathbf{cd}] = \iint \varphi_a(\mathbf{r}_1)\varphi_b(\mathbf{r}_1) r_{12}^{-1} \varphi_c(\mathbf{r}_2)\varphi_d(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 \,. \quad (2)$$

The bra part, a product of two primitive functions, can be written by

$$\exp\left[-\alpha_A (\mathbf{r} - \mathbf{A})^2 - \alpha_B (\mathbf{r} - \mathbf{B})^2\right] = L_P \exp\left[-\alpha_P (\mathbf{r} - \mathbf{P})^2\right] \quad (3)$$

where point **P** is defined as

$$\mathbf{P} = (\alpha_A \mathbf{A} + \alpha_B \mathbf{B}) / \alpha_P \,, \quad (4)$$

$$\alpha_P = \alpha_A + \alpha_B \,, \quad (5)$$

and the constant L_P is

$$L_P = \exp\left[-\alpha_A \alpha_B (\mathbf{A} - \mathbf{B})^2 / \alpha_P\right]. \quad (6)$$

In the same manner, the ket part in Eq. 2 can be written as follows:

$$\exp\left[-\alpha_C (\mathbf{r} - \mathbf{C})^2 - \alpha_D (\mathbf{r} - \mathbf{D})^2\right] = L_Q \exp\left[-\alpha_Q (\mathbf{r} - \mathbf{Q})^2\right] \,, \quad (7)$$

$$\mathbf{Q} = (\alpha_C \mathbf{C} + \alpha_D \mathbf{D}) / \alpha_Q \,, \quad (8)$$

$$\alpha_Q = \alpha_C + \alpha_D \,, \quad (9)$$

$$L_Q = \exp\left[-\alpha_C \alpha_D (\mathbf{C} - \mathbf{D})^2 / \alpha_Q\right]. \quad (10)$$

Primitive functions are usually contracted in the following way,

$$\phi_a(\mathbf{r}) = \sum_k^K D_{ak}\varphi_{ak} \tag{11}$$

where D_{ak} is a contraction coefficient and K is the degree of contraction. A contracted ERI can be written from Eqs. 2 and 3,

$$(\mathbf{ab}\,|\,\mathbf{cd}) = \sum_i^{K_A}\sum_j^{K_B}\sum_k^{K_C}\sum_l^{K_D} D_{ai}D_{bj}D_{ck}D_{dl}\left[\mathbf{a}_{Ai}\mathbf{b}_{Bj}\,|\,\mathbf{c}_{Ck}\mathbf{d}_{Dl}\right]. \tag{12}$$

In the Pople-Hehre (PH) algorithm,[2] Cartesian axes are rotated to make z- and y-directions along **AB** and perpendicular to **AB** and **CD**, respectively, as shown in Fig. 1, which are named axes-2 in the original paper. Point **P** lies on the **AB** line and depends on exponents α_A and α_B. In the rotated coordinate, PA_x, PA_y, PB_x, and PB_y are zero and PQ_x and PQ_y become constant at any point **P**. QC_y, and QD_y are zero at any point **Q**.

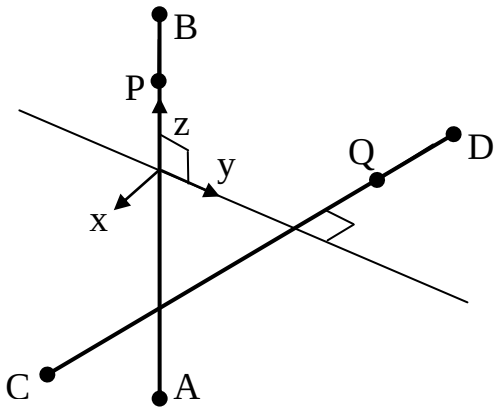


Figure 1. Rotated Cartesian axes.

In the McMurchie-Davidson (MD) algorithm,[6] the bra part of the primitive [**ab**|**cd**] (=[**ab0**|**cd0**]) is formed from the following recurrence relation,

$$\left[(\mathbf{a}+\mathbf{1}_i)\mathbf{bp}\,|\right. = p_i\left[\mathbf{ab}(\mathbf{p}-\mathbf{1}_i)\right| + (P_i - A_i)\left[\mathbf{abp}\,|\right. + (2\alpha_P)^{-1}\left[\mathbf{ab}(\mathbf{p}+\mathbf{1}_i)\right| \tag{13}$$

where subscript i denotes a Cartesian direction (x, y, or z) and $\mathbf{1}_i$ is the unit vector of the i-direction. The ket part is also formed in the following way,

$$|\left(\mathbf{c}+\mathbf{1}_i\right)\mathbf{dq}] = q_i\,|\,\mathbf{cd}\left(\mathbf{q}-\mathbf{1}_i\right)] + \left(Q_i - C_i\right)|\,\mathbf{cdq}] + \left(2\alpha_Q\right)^{-1}|\,\mathbf{cd}\left(\mathbf{q}+\mathbf{1}_i\right)]. \quad (14)$$

[**p**|**q**] (=[**00p**|**00q**]) necessary to generate [**ab**|**cd**] is evaluated by

$$[\mathbf{p}\,|\,\mathbf{q}] = \left(-1\right)^q[\mathbf{p}+\mathbf{q}]. \quad (15)$$

Furthermore, [**r**] (=[**r**]$^{(0)}$) integrals can be obtained from [**0**]$^{(m)}$ (=(ss|ss)$^{(m)}$) integrals,

$$[\mathbf{r}]^{(m)} = \left(Q_i - P_i\right)[\mathbf{r}-\mathbf{1}_i]^{(m+1)} - \left(r_i - 1\right)[\mathbf{r}-\mathbf{2}_i]^{(m+1)} \quad (16)$$

where $\mathbf{2}_i$ is twice the unit vector in the i-direction and [**0**]$^{(m)}$ is generated from

$$[\mathbf{0}]^{(m)} = D_a D_b D_c D_d L_P L_Q 2^{m+1}\pi^{5/2}\frac{\alpha_P^{\,m-1}\alpha_Q^{\,m-1}}{\left(\alpha_P+\alpha_Q\right)^{m+1/2}}F_m\left(T\right) \quad (17)$$

where

$$F_m\left(T\right) = \int_0^1 t^{2m}e^{-Tt^2}dt \quad (18)$$

$$T = \frac{\alpha_P\alpha_Q}{\alpha_P+\alpha_Q}\left(\mathbf{PQ}\right)^2. \quad (19)$$

When T is small, $F_m(T)$ for the largest m value is evaluated using Taylor or Chebyshev expansions and the others are evaluated using the following recurrence relation,[18]

$$F_m\left(T\right) = \frac{1}{2m+1}\left[2TF_{m+1}\left(T\right)+\exp\left(-T\right)\right]. \quad (20)$$

When T is large, $F_0(T)$ is evaluated using the following equation,

$$F_0\left(T\right) = \sqrt{\pi/4T}\,, \quad (21)$$

and the others are evaluated using the following recurrence relation,

$$F_{m+1}\left(T\right) = \left(2m+1\right)F_m\left(T\right)/2T\,. \quad (22)$$

We developed the PH+MD algorithm by combining the PH and MD algorithms. Cartesian axes are rotated using the PH algorithm, as described above, and then ERIs are calculated using MD algorithm at the rotated coordinate. The calculated ERIs are

transformed to the original coordinate. The second rotation in the original PH algorithm, axes-1, is not exploited in the PH+MD algorithm because the rotation is not useful for contracted basis functions. The pseudocode is shown in Fig. 2. Cartesian axes are first rotated, as shown in Fig. 1, and the quadruple loop starts at the rotated coordinate. The outer loop is over **c** and **d**, and the inner loop is over **a** and **b**. In the quadruple loop, $[\mathbf{0}]^{(m)}$ and the related integrals such as $(P_z\text{-}A_z)[\mathbf{0}]^{(m)}$, $(P_z\text{-}B_z)[\mathbf{0}]^{(m)}$, $(2\alpha_P)^{-1}[\mathbf{0}]^{(m)}$, and $(Q_z\text{-}P_z)[\mathbf{0}]^{(m)}$ are calculated and accumulated. Calculations of integrals including terms such as $(P_x\text{-}A_x)$ and $(Q_y\text{-}P_y)$ are skipped because PA_x, PA_y, PB_x, and PB_y are all zero and PQ_x and PQ_y are constant at the rotated coordinate. After the loop, the bra-contracted integrals $(\mathbf{0}]^{(m)}$ are obtained. In the double loop, (**r**] and the related integrals such as $(P_z\text{-}A_z)(\mathbf{r}]$, $(Q_x\text{-}C_x)(\mathbf{r}]$, and $(2\alpha_Q)^{-1}(\mathbf{r}]$ are calculated and accumulated. Calculations of integrals including terms such as $(P_x\text{-}A_x)$ and $(Q_y\text{-}C_y)$ are again skipped because PA_x, PA_y, PB_x, PB_y, QC_y, and QD_y are all zero. After the loop, the contracted integrals (**r**) are obtained. The (**ab**|**cd**) integrals are calculated from (**r**) using Eqs. 13-15 in the rotated coordinate and finally transformed into the original coordinate. (*sp*,*sp*|*sp*,*sp*) and (*dd*|*dd*) integrals have $90K^4$ and $830K^4$ constant values and $250K^2$+380 and $3900K^2$+31000 zero values, respectively (*K* is the degree of contraction).

As an example, we describe the (*ps*|*ss*) generation. In the quadruple loop, $[\mathbf{0}]^{(0)}$ and $[\mathbf{0}]^{(1)}$ are calculated for each primitive function, and $(2\alpha_P)^{-1}[\mathbf{0}]^{(1)}$, $(Q_z\text{-}P_z)(2\alpha_P)^{-1}[\mathbf{0}]^{(1)}$, and $(P_z\text{-}A_z)[\mathbf{0}]^{(0)}$ are calculated and accumulated for the bra part. In the double loop, $(2\alpha_P)^{-1}(\mathbf{x}]$, $(2\alpha_P)^{-1}(\mathbf{y}]$, $(2\alpha_P)^{-1}(\mathbf{z}]$, and $(P_z\text{-}A_z)(\mathbf{0}]$ are calculated and accumulated for the ket part. After the loop, (*ps*|*ss*) integrals are calculated using the following recurrence relations derived from Eqs. 13-15,

$$\left(p_x s \mid ss\right) = \left(2\alpha_P\right)^{-1}\left(\mathbf{x}\right), \tag{23}$$

$$\left(p_y s \mid ss\right) = \left(2\alpha_P\right)^{-1}\left(\mathbf{y}\right), \tag{24}$$

$$(p_z s \mid ss) = (P_z - A_z)(\mathbf{0}) + (2\alpha_P)^{-1}(\mathbf{x}). \quad (25)$$

At the end, (*ps*|*ss*) integrals are transformed to the original coordinate.

rotate Cartesian axes

do **c** shells

  do **d** shells

    do **a** shells

      do **b** shells

        calculate $F_m(T)$

        calculate and accumulate $[\mathbf{0}]^{(m)}$ and the related integrals

      enddo **b**

    enddo **a**

    calculate and accumulate $(\mathbf{r}]^{(m)}$ and the related integrals

  enddo **d**

enddo **c**

calculate (**ab**|**cd**)

rotate (**ab**|**cd**) to original Cartesian axes

Figure 2. Pseudocode of PH+MD algorithm.

1.4 Results

Tables 1 and 2 show the FLOP count parameters for (*sp*,*sp*|*sp*,*sp*) and (*dd*|*dd*), respectively. The FLOP count is given by

$$\text{FLOPs} = xK^4 + yK^2 + z. \tag{26}$$

Table 1. FLOP count parameters for (*sp*,*sp* |*sp*,*sp*).

	x	y	z
PH+MD	180	1100	5330
PH[a]	220	2300	4000
MD[b]	1500	1700	0
Dupuis, Rys, and King[c]	1056	30	800
Head-Gordon and Pople[d]	1400	30	800
Gill, Head-Gordon, and Pople[e]	450	1300	1700
Lindh, Ryu, and Liu[f]	753	30	800
ACE-RR[g]	155	774	4548

[a]Reference 2.

[b]Reference 6.

[c]Reference 3-5.

[d]Reference 8.

[e]Reference 9.

[f]Reference 10.

[g]Reference 17.

Table 2. FLOP count parameters for (*dd* |*dd*).

	x	y	z
PH+MD	850	11860	173500
MD[a]	27300	24000	0
Dupuis, Rys, and King[b]	30900	220	0
Head-Gordon and Pople[c]	14600	30	11300
Gill, Head-Gordon, and Pople[d]	2450	25800	28900
Lindh, Ryu, and Liu[e]	10255	30	11300
ACE-b3k3[f]	327	2281	163000

[a]Reference 6.

[b]Reference 3-5.

[c]Reference 8.

[d]Reference 9.

[e]Reference 10.

[f]Reference 12.

Table 3 shows the FLOP counts of (*sp*,*sp*|*sp*,*sp*) for several *K*. The parameters *x* and *y* for K^4 and K^2 become very small in the PH+MD algorithm, compared with the original algorithms, because of zero components, constant components, and efficient recurrence relations. Instead, the zeroth power parameter *z* becomes large because the operations of the last recurrence relations, Eqs. 13-15, are performed. This PH+MD algorithm is especially suited for the moderate degree of contraction, for instance, STO-3G and 6-31G(d). The FLOP count of the PH+MD algorithm is not the least of all the algorithms. However more than 80% operations of the algorithm are performed in do

loops. This is a significant advantage for practical calculations because of the sequential access to fast memory.

Table 3. FLOP counts for (*sp*,*sp* |*sp*,*sp*).

	K=1	K=2	K=3	K=6
PH+MD	6610	12610	29810	278210
PH[a]	6520	16720	42520	371920
MD[b]	3200	30800	136800	2005200
Dupuis, Rys, and King[c]	1886	17816	86606	1370456
Head-Gordon and Pople[d]	2230	23320	114470	1816280
Gill, Head-Gordon, and Pople[e]	3450	14100	49850	631700
Lindh, Ryu, and Liu[f]	1583	12968	62063	977768
ACE-RR[g]	5477	10124	24069	233292

[a]Reference 2.

[b]Reference 6.

[c]Reference 3-5.

[d]Reference 8.

[e]Reference 9.

[f]Reference 10.

[g]Reference 17.

The PH+MD algorithm derived for (*ss*|*ss*) to (*dd*|*dd*) was implemented into the quantum chemistry package GAMESS[19] by adding 39000 lines. Test calculations were performed for PH+MD and compared with the PH and Rys algorithms available in GAMESS using a 3.0GHz Pentium4 machine. PH is used for ERIs including *s* and *sp* functions and Rys is used for ERIs including *d* functions. The computational times for the Fock matrix generation in the first direct SCF iteration are shown in Table 4 for taxol ($C_{47}H_{51}NO_{14}$, no symmetry) and luciferin ($C_{11}H_8N_2O_3S_2$, no symmetry). STO-3G, 6-31G, and 6-31G(d) basis sets were used for taxol and 6-31G, 6-31G(d), and aug-cc-pVDZ were used for luciferin. An integral prescreening threshold of 10^{-9} was adopted in all calculations. As shown in Table 4, the PH+MD algorithm reduces computational times by 10-20% in comparison with the PH algorithm for the STO-3G and 6-31G basis sets. This is because the FLOP count of (*sp*,*sp*|*sp*,*sp*) is 25-30% reduced in PH+MD though the FLOP counts of (*ss*|*ss*) and (*sp*,*s*|*ss*) are almost the same in PH+MD and PH. It is interesting that PH+MD reduces the computational times by 32-43% in comparison with PH and Rys for 6-31G(d) and aug-cc-pVDZ. This is owing to the fact that ERIs of *sp* and *d* functions such as (*d*,*sp*|*d*,*sp*) and (*dd*|*sp*,*sp*) are generated very fast. Only for primitive (*dd*|*dd*) calculations PH+MD is somewhat slower than Rys. The number of primitive (*dd*|*dd*) ERIs is usually very small, so that the disadvantage is negligible in practical calculations. Due to the small *x* coefficient for quartic work, the strength of this new approach must be even more apparent for heavy atoms with valence or core *d* electrons, whose basis sets would contain contracted *d* functions.

Table 4. Computational times of PH+MD, PH, and Rys algorithms (sec).

		PH+MD	PH	PH and Rys
taxol	STO-3G	69.9	85.7	
	6-31G	379.4	475.4	
	6-31G(d)	1361.8		2015.2
luciferin	6-31G	8.6	9.6	
	6-31G(d)	39.3		62.2
	aug-cc-pVDZ	1154.5		2014.9

1.5 Conclusions

A new algorithm to calculate (*ss*|*ss*) to (*dd*|*dd*) ERIs is developed by combining the PH and MD algorithms and implemented into the GAMESS program. The FLOP count is highly reduced because of the axis-switch in PH and the recurrence relation in MD. The PH+MD algorithm is considerably faster than the PH and Rys algorithms, especially for the basis sets including *d* functions such as 6-31G(d) and aug-cc-pVDZ.

1.6 References

1. Boys, S. F. Proc Roy Soc London A 1950, 200, 542.

2. Pople, J. A.; Hehre, W. J. J Comput Phys 1978, 27, 161.

3. King, H. F.; Dupuis, M. J Comput Phys 1976, 21, 144.

4. Dupuis, M.; Rys, J.; King, H. F. J Chem Phys 1976, 65, 111.

5. Rys, J.; Dupuis, M.; King, H. F. J Comput Chem 1983, 4, 154.

6. McMurchie, L. E.; Davidson, E. R. J Comput Phys 1978, 26, 218.

7. Obara, S.; Saika, A. J Chem Phys 1986, 84, 3963.

8. Head-Gordon, M.; Pople, J. A. J Chem Phys 1988, 89, 5777.

9. Gill, P. M. W.; Head-Gordon, M.; Pople, J. A. J Phys Chem 1990, 94, 5564.

10. Lindh, R.; Ryu, U.; Liu, B. J Chem Phys 1991, 95, 5889.

11. Ishida, K. Int J Quantum Chem 1996, 59, 209.

12. Ishida, K. J Comput Chem 1998, 19, 923.

13. Gill, P. M. W.; Pople, J. A. Int J Quantum Chem 1991, 40, 753.

14. Schlegel, H. B. J Chem Phys 1982, 77, 3676.

15. Hamilton, T. P.; Schaefer, III H. F. Chem Phys 1991, 150, 163.

16. Ten-no, S. Chem Phys Lett 1993, 211, 259.

17. Kobayashi, M.; Nakai, H. J Chem Phys 2004, 121, 4050.

18. Gill, P. M. W.; Johnson, B. G.; Pople, J. A. Int J Quantum Chem 1991, 40, 745.

19. Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S.; Windus, T. L.; Dupuis, M.; Montgomery, J. A. J Comput Chem 1993, 14, 1347.

Chapter II

A New Parallel Algorithm of MP2 Energy Calculations

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

2.1 Abstract

A new parallel algorithm has been developed for second order Møller-Plesset perturbation theory (MP2) energy calculations. Its main projected applications are for large molecules, for instance for the calculation of dispersion interaction. Tests on a moderate number of processors (2-16) show that the program has high CPU and parallel efficiency. Timings are presented for two relatively large molecules, taxol ($C_{47}H_{51}NO_{14}$) and luciferin ($C_{11}H_8N_2O_3S_2$), the former with the 6-31G* and 6-311G** basis sets (1032 and 1484 basis functions, 164 correlated orbitals), and the latter with the aug-cc-pVDZ and aug-cc-pVTZ basis sets (530 and 1198 basis functions, 46 correlated orbitals). An MP2 energy calculation on $C_{130}H_{10}$ (1970 basis functions, 265 correlated orbitals) completed in less than 2 hours on 128 processors.

2.2 Introduction

Density functional theory (DFT) is currently the most widely used method to calculate molecular properties. However, the generally used local and semilocal DFT methods fail to describe the dispersion interaction that plays an important role for large molecules, and usually underestimate reaction barriers. The simplest method to include dispersion nonempirically is second order Møller-Plesset perturbation theory (MP2).[1] The formal scaling of MP2 with molecular size (assuming constant basis set quality) is fifth order, $O(n^5)$, higher than the $O(n^4)$ formal scaling of Hartree-Fock theory. Moreover, natural sparsity is more difficult to use in canonical MP2 theory than in Hartree-Fock theory. These factors together made MP2 theory significantly more costly for large systems than, say, Hartree-Fock theory. Much effort has been expended to create less expensive approximations to MP2. Local MP2[2-9] can be much less expensive than full MP2 for large systems. Other efficient methods include Density Fitting (Resolution of Identity, RI) MP2[10-13] and Laplace Transform MP2.[14-17] All of these methods include approximations that need to be carefully checked against full calculations. Most introduce a new model chemistry, and have limitations. For instance, the efficiency of local MP2 decreases for genuinely delocalized systems, like π-π interactions, and for diffuse basis sets. The recent implementation of RI-MP2 by the Ahlrichs group[13] has no problems with large basis sets and promises to be an excellent tool for routine calculations. However, it also introduces a slightly different model chemistry, and its ultimate scaling is fifth order, like that of canonical MP2. A Laplace Transform MP2 utilizing natural sparsity has been implemented[17] but its computational superiority has yet to be demonstrated.

An alternative method of overcoming the long computational times of MP2 is parallel implementation. This has become particularly useful with the advent of

inexpensive high-speed processors. A number of papers have been published on parallel integral transformation, the main part of parallel MP2.[18-28] The simplest parallelization methods use a single variable, say atomic orbital (AO) or molecular orbital (MO) indices to distribute the workload, or use shared memory computers. These methods have significant shortcomings. For instance, in the AO-based parallelization of Nielsen and Seidl[23] progressively more network communication is needed as the number of processors increases. The MO-based parallelization, for instance the methods devised by Schütz and Lindh[26] require the repeated evaluation of AO integrals on different nodes or to broadcast integrals to all nodes, limiting parallel efficiency. Shared memory computers are much less widely available and more expensive than clusters, and their size is limited.

Recently, Baker and Pulay[29] have described a parallel MP2 implementation, based on an efficient canonical program[30] using the Saebo-Almlöf integral transformation method.[31] This program parallelizes the first half transformation by AO indices and the second half transformation by MO indices. Its advantages are that the fast memory needed scales only with the square of the system size, the total amount of communication is independent of the number of processors, and the AO integrals are generated only once. One of its disadvantages is that it uses only one permutational symmetry during AO integral generation, quadrupling the computational effort for this task. However, the main disadvantage of the Baker-Pulay code is the I/O overhead involved in the sorting of the half-transformed integrals. While disk I/O on the current generation of computers cannot be eliminated, the current project aims at reducing this overhead. The required memory in the present code has a cubic component. However, it is proportional to o^2n only, where o is the number of correlated MOs. In most MP2 calculations, o is about an order of magnitude smaller than n, the number of basis

functions, and therefore memory becomes a limiting factor only for very large calculations in the present program.

The algorithm described in this paper has been implemented in the freely distributed quantum chemistry program system GAMESS.[32] GAMESS has already a parallel MP2 implementation,[28] based on the Distributed Data Interface.[33] However, this implementation stores the transformed integrals, an array that scales with the fourth power of the molecular size, in distributed fast memory. If there is not enough memory, it resorts to expensive multiple passes. This is an efficient strategy on massively parallel computers where the aggregate memory is sufficient to hold all integrals, but it becomes inefficient for larger calculations on small or medium-sized clusters. By making use of ample and inexpensive disk storage, the program presented in this paper is able to perform large calculations on modest clusters with high efficiency.

2.3 Theory

The closed-shell MP2 energy can be written as

$$E(\mathrm{MP2}) = \sum_{i \geq j}^{occ.} (2 - \delta_{ij}) \sum_{ab}^{virt.} \frac{(ia \mid jb)[2(ia \mid jb) - (ib \mid ja)]}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b}, \qquad (1)$$

where *i* and *j* are doubly occupied MOs, *a* and *b* are virtual MOs, and *ε's* are corresponding the orbital energies. (*ia*|*jb*) denotes a two-electron MO integral that is generated as

$$(ia \mid jb) = \sum_{\mu\nu\lambda\sigma} C_{\mu i} C_{\nu a} C_{\lambda j} C_{\sigma b} (\mu\nu \mid \lambda\sigma). \qquad (2)$$

Here, and in the following Greek letters denote AOs, ($\mu\nu|\lambda\sigma$) is a two-electron AO integral, and *C* is the matrix of the MO coefficients. The most time-consuming step in

an MP2 energy calculation is the transformation of the AO integrals to MOs, Eq. (2).

The outline of the new algorithm is summarized in Figure 1 and the required memory, disk space requirements, and formal computational costs are shown in Table 1. First, we describe the serial version. Capital Greek letters M, N, Λ, and Σ denote shells of AOs; computational efficiency mandates the utilization of shell structure. The outermost loop up to the third quarter transformation is over AO shell M. In the loop, AO integrals $(MN|\Lambda\Sigma)$ are calculated for one M, Λ, and Σ, and all N. $(MN|\Lambda\Sigma)$ denotes all integrals $(\mu\nu|\lambda\sigma)$, $\mu\in M$, $\nu\in N$, $\lambda\in\Lambda$, and $\sigma\in\Sigma$. Schwartz prescreening[34-36] is used in this phase to discard insignificant integrals, just as in the SCF calculation. This results in significant computational savings. The required memory size in this step is s^3n, where s is the maximum number of basis functions in a shell, for instance, 1 for s function and 4 for sp function and n is the total number of basis functions. The formal computational cost is $O(n^4)$; for large molecules, this diminishes to $O(n^2)$. Only one of the three permutational symmetries, $(\mu\nu|\lambda\sigma)=(\mu\nu|\sigma\lambda)$, is exploited in our algorithm. This means that AO integrals are effectively evaluated four times. The SCF algorithm and some other MP2 algorithms can use three symmetries. However, we find, in agreement with Pulay *et al.*,[30] that sacrificing permutational symmetry of the AO integrals incurs only a modest penalty. After the calculation of AO integral blocks, the first quarter transformation,

$$(\mu i|\lambda\sigma)=\sum_{\nu}C_{\nu i}(\mu\nu|\lambda\sigma) \qquad (3)$$

is performed for all i, and $\mu\in M$, $\lambda\in\Lambda$, and $\sigma\in\Sigma$. The memory size required to store the transformed integrals is s^3o, where o is the number of correlated MOs, and the formal computational cost is $O(on^4)$. The second quarter transformation,

$$(\mu i|\lambda j)=\sum_{\sigma}C_{\sigma j}(\mu i|\lambda\sigma) \qquad (4)$$

is performed for all i and $j(\le i)$, and for $\mu \in M$ and $\lambda \in \Lambda$. The required memory size is $so^2n/2$ and the formal computational cost is $O(o^2n^3)$. This is the step with the largest memory demand in our algorithm. On a 32-bit computer, memory limits restrict the calculation to ~2000 basis functions and 200 occupied orbitals if the highest angular momentum functions are 6-component *d* functions. This limit can be exceeded (see below) at the cost of introducing multiple integral batches. The half-transformed $(\mu i|\lambda j)$ integrals are accumulated for all i, $j(\le i)$, and λ, and for $\mu \in M$. The quarter-transformed $(\mu i|\lambda\sigma)$ integrals are screened prior to the transformation.

The third quarter transformation,

$$(\mu i \mid bj) = \sum_{\lambda} C_{\lambda b} (\mu i \mid \lambda j) \tag{5}$$

is performed for all i, $j(\le i)$, and b, and for $\mu \in M$. The three-quarter transformed integrals are collected in fixed-size bins that are written to disk when full. We use the compact DDI library[33] in GAMESS for communication, and achieve high CPU efficiency by choosing the bin size the same size of disk cache, for instance, 2 or 8 MB. This reduces disk I/O time significantly, as writing to the disk cache is more than ten times faster than to the disk itself. The computational cost of this step, both formal and actual, is $O(o^2vn^2)$, where v is the number of virtual MOs. No screening of the $(\mu i|\lambda j)$ integrals is used in this transformation, as the canonical orbitals are usually delocalized, making the screening ineffective. The total disk storage required is $o^2vn/2$, somewhat less that in Ref. 30 ($o^2n^2/2$). After the M loop has finished, all $(\mu i|bj)$ integrals have been calculated and stored.

In the fourth quarter transformation,

$$(ai \mid bj) = \sum_{\nu} C_{\mu a} (\mu i \mid bj), \tag{6}$$

the outermost loop step is over batches of *ij* pairs. A block of ($\mu i|bj$) integrals for all μ, *b* and a batch of *ij* pairs are read from disk and transformed. For a single (*ij*) orbital pair, this step requires a minimum array size of *vn* for storing the ($\mu i|bj$) integrals and v^2 for ($ai|bj$), in addition to the storage of the SCF coefficient matrix. However, efficiency is higher if as many pairs are processed together as possible. The computational cost of this step is $O(o^2v^2n)$. The computational cost of the MP2 energy calculation is $O(o^2v^2)$. If the memory demand is too high, the program can use multiple passes in the calculation of AO integrals to the third quarter transformation, with the AO integrals recalculated in each pass. The maximum memory size depends on the shell size *s*.

This algorithm uses a two steps parallelization like in Ref. 29. In our algorithm, AO indices, (in reality shells, *M* in Figure 1), are distributed dynamically to each processor in the AO integral generation phase and subsequent phases up to the third quarter transformation. A subset of the three-quarter transformed ($\mu i|bj$) integrals (all *b*s, all *ij*, $i \geq j$ and a subset of μs) is stored on each node, with *b* running fastest, followed by *ij* and finally μ. The fourth quarter transformation and the MP2 energy calculation are parallelized by batches of occupied MO pairs *ij*. This parallelization is static, i.e. each process gets approximately the same number of *ij* pairs, because no cut-offs are included, and therefore the timings are the same for each pair. Prior to the fourth quarter transformation, each process reads its own three-quarter transformed integrals, and sends them, according to the *ij* indices, to the appropriate node. As a result, the μ indices become scrambled; rather than reorder the integrals, each process reorders its copy of the AO coefficient matrix accordingly. In the final, computationally insignificant stage, the partial MP2 energies are accumulated on the master process. The required memory size for each quarter transformation is the same as the serial version and the disk storage size per process is $o^2vn/2n_{\mathrm{proc}}$, where n_{proc} is the number of the processors. The total

amount of network communication is o^2vn, and it does not depend on the number of processors. This feature is suitable for the parallel computation. An in-core version is available for small molecules or a large number of processors, in which all ($\mu i|bj$) integrals are stored in memory, not on disk.

Table 1. Flop count, required memory, and total disk space for each step.

	Flop count	memory	disk
AO integral generation	$O(n^4)$	s^3n	
first quarter transformation	$O(on^4)$	s^3o	
second quarter transformation	$O(o^2n^3)$	$so^2n/2$	
third quarter transformation	$O(o^2vn^2)$	k_1sv (=2 or 8MB)	$o^2vn/2$
fourth quarter transformation	$O(o^2v^2n)$	$k_2(vn+v^2)$	
MP2 energy calculation	$O(o^2v^2)$		

k_1, k_2: batches of ij pair.

do M (distributed dynamically)

do Λ = 1, nshells

do Σ= 1, Λ

Schwartz prescreening of ($MN|\Lambda\Sigma$) integrals

calculate $(\mu\nu \mid \lambda\sigma)$ and store as elements of the matrix $\mathbf{T}_{\nu,\sigma\lambda\mu}$ in memory for all ν, $\mu \in M$, $\lambda\in\Lambda$, and $\sigma\in\Sigma$

transform to $(\mu i \mid \lambda\sigma) = (\mathbf{T}\cdot\mathbf{C}_{occ})_{i,\sigma\lambda\mu}$ and store $(\mu i \mid \lambda\sigma) = \mathbf{U}_{i,\sigma\lambda\mu}$ in memory for all i, $\mu\in M$, $\lambda\in\Lambda$, and $\sigma\in\Sigma$

screen of $(\mu i \mid \lambda\sigma)$ integrals

transform to $(\mu i \mid \lambda j) = (\mathbf{U}\cdot\mathbf{C}_{occ})_{ij,\lambda,\mu}$ and store $(\mu i \mid \lambda j) = \mathbf{V}_{ij,\lambda,\mu}$ in memory for all i and $j(\leq i)$, $\mu\in M$, and $\lambda\in\Lambda$

end do Σ

end do Λ

do all ij batch

transform to $(\mu i \mid bj) = (\mathbf{V}\cdot\mathbf{C}_{vir})_{b,ij,\mu}$ for all b and $\mu\in M$

write $(\mu i \mid bj) = \mathbf{W}_{b,ij,\mu}$ on disk

end do ij batch

end do M

do ij batch (distributed statically)

read $(\mu i \mid bj) = \mathbf{X}_{b,ij,\mu}$ from disk and send to an appropriate node for all *b*, partial *ij* of the node to send, and all μ on this node

receive $(\mu i \mid bj) = \mathbf{Y}_{b,ij,\mu}$ from other nodes for all *b* and μ, and a batch of *ij*

transform to $(ai \mid bj) = (\mathbf{Y} \cdot \mathbf{C}_{vir})_{a,b,ij}$ and store $(ai \mid bj) = \mathbf{Z}_{a,b,ij}$ in memory for all *a* and *b*, and a batch of *ij*

calculate partial MP2 energy

end do *ij* batch

accumulate MP2 energy

Figure 1. Outline of the parallel algorithm.

2.4 Results and Discussion

Test calculations on taxol ($C_{47}H_{51}NO_{14}$, no symmetry) and luciferin ($C_{11}H_8N_2O_3S_2$, no symmetry) were performed on a cluster of 3.0 GHz Pentium4 computers connected by gigabit Ethernet. Each node has 2GB dual-PC3200 DDR memory and a striped pair of 200GB disks, each with 8MB cache.

6-31G* and 6-311G** basis sets were used for taxol, and aug-cc-pVDZ and aug-cc-pVTZ basis sets were used for luciferin. Only the valence orbitals were correlated. Table 2 summarizes the details of the calculation (the number of basis functions, shells, correlated orbitals, virtual orbitals, and SCF cycles), and the required memory and disk size per process, and the total amount of network communication. Table 3 shows elapsed timings and speed-ups of SCF and MP2 single point calculations. An integral screening threshold of 10^{-10} was used in all calculations on taxol and 10^{-11} on luciferin. The tighter threshold is required in the latter calculation because of the near-singularity of the augmented basis sets for luciferin (the lowest eigenvalue of the overlap matrix is 6.6×10^{-8}).

As the data in Table 2 show, even the large calculation (taxol 6-311G**, almost 1500 basis functions) can be easily accommodated by a single PC. This calculation needs less than 1 GB of fast memory and about 200 GB of disk space.

Table 2. Details of the calculations.

	taxol		luciferin	
	6-31G*	6-311G**	aug-cc-pVDZ	aug-cc-pVTZ
Number of contracted basis functions	1032	1484	530	1198
Number of basis shells	350	514	206	328
Number of correlated orbitals	164	164	46	46
Number of virtual orbitals	806	1196	422	948
Number of SCF cycles	14	14	15	14
Required memory size per processor /GB	0.67	0.96	0.05	0.16
Total required disk size /GB	90	192	2	10
Total amount of network communication /GB	90	192	2	10

As the timings in Table 3 show, in spite of its steeper formal scaling, the computational time for MP2 energy is commensurate with the SCF time. The parallel scaling of the code is excellent up to the largest number of nodes we have tried: For instance, on 16 processors the elapsed time for the MP2 calculation is 15.4 times faster (in average) than the single-processor time. This is a consequence of the high CPU efficiency of the code (88-98% on 16 processors) which is defined as the ratio of master node CPU and elapsed times.

Table 3. Elapsed times[a] and speed-ups of SCF and MP2 calculations (minutes).

Number of processors		1	2	4	8	16
Taxol	6-31G*					
	t_{scf}[b]	323.7(100%)	160.6(99%)	82.6(99%)	42.6(96%)	25.0(84%)
	s_{scf}[c]	1.00	2.02	3.92	7.60	12.94
	t_{mp2}[d]	611.3(90%)	305.0(90%)	152.1(91%)	78.5(88%)	38.7(89%)
	s_{mp2}[e]	1.00	2.00	4.02	7.78	15.80
	6-311G**					
	t_{scf}[b]	1051.5(100%)	520.2(100%)	263.8(99%)	137.6(97%)	73.5(89%)
	s_{scf}[c]	1.00	2.02	3.99	7.64	14.31
	t_{mp2}[d]	1898.2(92%)	975.0(90%)	483.7(91%)	242.9(91%)	123.3(88%)
	s_{mp2}[e]	1.00	1.95	3.92	7.81	15.40
Luciferin	aug-cc-pVDZ					
	t_{scf}[b]	305.3(100%)	159.5(100%)	77.8(100%)	39.1(99%)	20.0(97%)
	s_{scf}[c]	1.00	1.91	3.92	7.81	15.26
	t_{mp2}[d]	114.3(99%)	58.2(99%)	29.0(98%)	14.8(98%)	7.5(97%)
	s_{mp2}[e]	1.00	1.96	3.94	7.73	15.19
	aug-cc-pVTZ					
	t_{scf}[b]	3814.1(100%)	2062.4(100%)	971.0(100%)	487.1(100%)	244.0(99%)
	s_{scf}[c]	1.00	1.85	3.93	7.83	15.63
	t_{mp2}[d]	1452.7(100%)	710.5(99%)	367.9(98%)	182.3(96%)	96.1(98%)
	s_{mp2}[e]	1.00	2.04	3.95	7.97	15.12

[a]CPU efficiency is shown in parentheses.

[b]Elapsed time of SCF calculation.

[c]Speed-up of SCF calculation.

[d]Elapsed time of MP2 calculation.

[e]Speed-up of MP2 calculation.

Table 4 shows the master node CPU and elapsed timings of the individual steps for the 6-311G** calculation on taxol. It is not possible to break down the elapsed time of the first steps (from AO integral generation to third transformation) and only the total is shown. Though the CPU timing ratios of the AO integral generation and each quarter transformation vary with the number of processors, the speed-up of the first step is almost proportional to the number of processors. The CPU efficiency of the first step is over 99%, in spite of writing the intermediate integrals on disk. This high efficiency is achieved by using an array size in the third transformation that is the same as the disk cache size. This way, the disk writing effectively overlaps with the CPU calculation. The lower CPU efficiency of the fourth transformation (including MP2 energy calculation) comes from reading the intermediate integrals from disk and from network communication. In this example, the CPU efficiency of the fourth transformation varies between 52% and 58%; it shows no systematic variation with the number of processors. The most time-consuming step is the first transformation for the 6-311G** basis, and the second transformation for the 6-31G* basis. As Table 4 shows, the ratio of the first to the second quarter transformation for the 6-311G** basis is only about 1.5, much less than the ratio n/o=9, showing the efficiency of integral screening. The percentages of the $(\mu\nu|\lambda\sigma)$ integrals skipped in the first quarter transformation, and the $(\mu i|\lambda\sigma)$ integrals skipped in the second quarter transformation are 94.9% and 75.9% for 6-311G** and 93.4% and 72.6% for 6-31G* basis sets, respectively. The speed-ups on 2 processors, for instance, in luciferin aug-cc-pVTZ, are slightly higher than the

theoretical upper limit. It may happen that processor cache hit ratios on 2 processors are better than the ratios on 1 processor.

Table 4. CPU and elapsed times of each step for 6-311G** calculation on taxol (minutes).

Number of processors	1	2	4	8	16
CPU time					
AO integral generation	318.3	162.6	79.1	40.7	19.6
1st transformation	535.6	269.6	134.1	63.7	29.8
2nd transformation	372.7	185.4	93.2	48.6	25.1
3rd transformation	266.1	136.0	68.1	35.3	18.4
AO-3rd transformation	1535.3	776.0	385.2	193.7	95.6
4th transformation + MP2 energy calculation	205.0	104.9	53.2	26.4	13.3
total	1740.2	880.9	438.5	220.1	108.8
Elapsed time					
AO-3rd transformation	1544.9	773.5	388.3	194.9	97.9
4th transformation + MP2 energy calculation	353.2	201.6	95.4	48.0	25.4
total	1898.2	975.0	483.7	242.9	123.3

Figure 2 shows the speed-ups, defined as the ratios of elapsed times, of the 6-311G** calculation on taxol and the aug-cc-pVTZ calculation on luciferin. The speed-ups are

almost linear, indicating the high parallel efficiency.

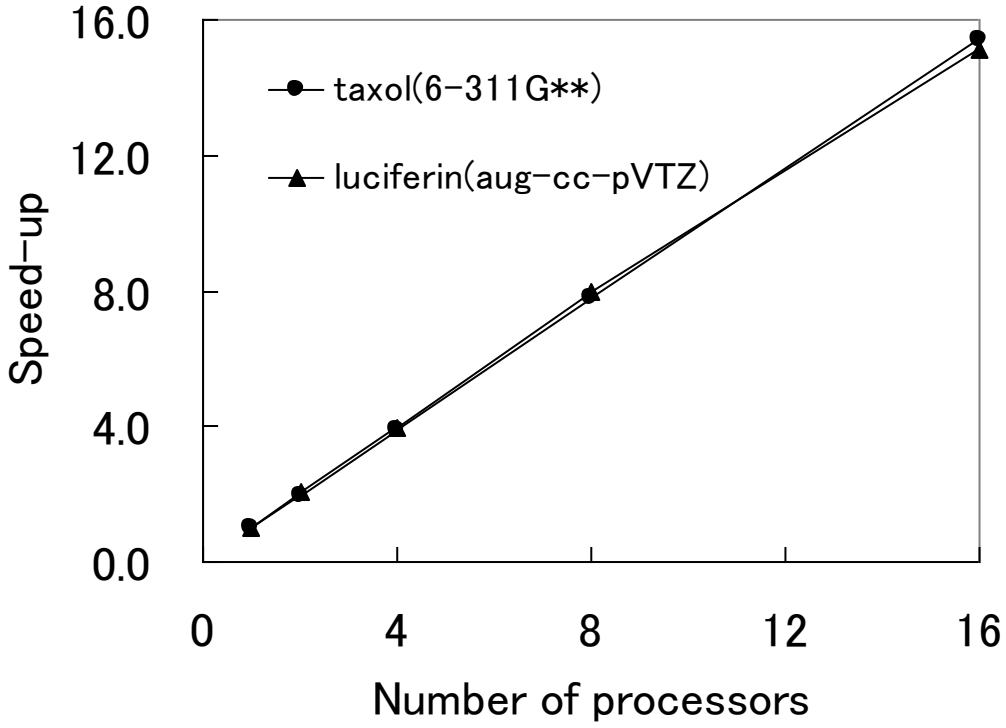


Figure 2. Speed-up ratios of the 6-311G** calculation on taxol and the aug-cc-pVTZ calculation on luciferin.

As Table 3 shows, for a medium-sized molecule with large basis set (luciferin aug-cc-pVTZ), MP2 is significantly less expensive than SCF. Moreover, with three marginal exceptions (luciferin aug-cc-pVDZ on 8 or 16 processors and aug-cc-pVTZ on 16 processors), the parallel speed-up of MP2 is higher than that of SCF. The slightly lower parallel speed-ups for the luciferin calculations arise from imperfect load balancing, caused by the relatively few and large shells in these calculations.

As a preliminary application to grid computing, a larger calculation (a segment of hydrogen terminated (5,0) carbon nanotube, $C_{130}H_{10}$) with the 6-31G* basis (1970 contracted basis functions) was run in a GRID computing environment on a total of 128 processors (64 Hitachi SR-11000 and 64 Hitachi HA-8000) at the NAREGI computer center (Okazaki, Japan). The SR-11000 is IBM Power-4 compatible while the HA-8000 is essentially a 3 GHz Intel Xeon processor. Elapsed time for the MP2 energy calculation was less than 2 hours (117 min), and CPU efficiency was high (94 %), even on this heterogeneous system. For comparison, the elapsed time for the SCF procedure was 52 minutes using a new, highly efficient two-electron integral program for GAMESS.

2.5 Conclusions

We have developed a parallel algorithm for MP2 energy calculation with two-step parallelization and dynamic load-balancing. The parallel efficiency of the algorithm is very good. For moderately large calculations (1000-2000 basis functions), the cost of MP2 is comparable to that of SCF. The CPU efficiency of the present algorithm is high, i.e. the elapsed time is only slightly longer than the CPU time. Using this algorithm, it is entirely practical to calculate the MP2 energies of moderately large molecules.

2.6 References

1. Møller, C.; Plesset M. S. Phys Rev 1934, 46, 618.

2. Saebo, S.; Pulay, P. Ann Rev Phys Chem 1993, 44, 213.

3. Boughton, J. W.; Pulay, P. Int J Quantum Chem 1993, 47, 49.

4. Rauhut, G; Pulay, P.; Werner, H.-J. J Comput Chem 1998, 19, 1241.

5. Hetzer, G; Pulay, P.; Werner, H.-J. Chem Phys Lett 1998, 290, 143.

6. Maslen, P. E.; Head-Gordon, M. Chem Phys Lett 1998, 293, 102.

7. Hetzer, G; Schütz, M; Stoll, H; Werner, H.-J. J Chem Phys 2000, 113, 9443.

8. Saebo, S.; Pulay, P. J Chem Phys 2001, 115, 3975.

9. Nakao, Y.; Hirao, K. J Chem Phys 2004, 120, 6375.

10. Feyereisen, M. W.; Fitzgerald, G.; Komonicki, A. Chem Phys Lett 1993, 208, 359.

11. Vahtras, O.; Almlöf, J.; Feyereisen, M. W. Chem Phys Lett 1993, 213, 514.

12. Bernholdt, D. E.; Harrison, R. J. Chem Phys Lett 1996, 250, 477.

13. Weigend, F; Häser, M; Patzelt, H; Ahlrichs, R. Chem Phys Lett 1998 294, 143.

14. Almlöf, J. Chem Phys Lett 1991, 181, 319.

15. Häser, M.; Almlöf, J. J Chem Phys 1992, 96, 489.

16. Wilson, A. K.; Almlöf, J Theor Chem Acc 1997, 95, 49.

17. Ayala, P. Y.; Scuseria, G. E. J Chem Phys 1999, 110, 3660.

18. Whiteside, R. A.; Binkley, J. S.; Colvin, M. E.; Schaefer, H. F. III. J Chem Phys 1987, 86, 2185.

19. Watts, J. D.; Dupuis, M. J Comput Chem 1988, 8, 158.

20. Limaye, A. C.; Gadre, S. R. J Chem Phys 1994, 100, 1303.

21. Limaye, A. C. J Comput Chem 1997, 18, 552.

22. Márquez, A. M.; Dupuis, M. J Compt Chem 1995, 16, 395.

23. Nielsen, I. M. B.; Seidl, E. T. J Comput Chem 1995, 16, 1301.

24. Bernholdt, D. E.; Harrison, R. J. J Chem Phys 1995, 102, 9582.

25. Wong, A. T.; Harrison, R. J.; Rendell, A. P. Theoret Chim Acta 1996, 93, 317.

26. Schütz, M.; Lindh, R. Theoret Chim Acta 1997, 95, 13.

27. Sosa, C. P.; Ochterski, J.; Carpenter, J.; Frisch, M. J. J Comput Chem 1998, 19,

1053.

28. Fletcher, G. D.; Schmidt, M. W.; Gordon, M. S. Adv Chem Phys 1999, 110, 267.

29. Baker, J.; Pulay, P. J Comput Chem 2002, 23, 1150.

30. Pulay, P.; Saebo, S.; Wolinski, K. Chem Phys Lett 2001, 344, 543.

31. Saebo, S.; Almlöf, J. Chem Phys Lett 1989, 154, 83.

32. Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. J.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S.; Windus, T. L.; Dupuis, M.; Montgomery, J. A. J Comput Chem 1993, 14, 1347.

33. Fletcher, G. D.; Schmidt, M. W.; Bode, B. M.; Gordon, M. S. Comput Phys Commun 2000, 128, 190.

34. Abramowitz, M.; Stegun, I. A., Eds.; Handbook of Mathematical Functions; Dover: New York, 1972.

35. Horn, H.; Weiss, H.; Häser, M.; Ehrig, M.; Ahlrichs, R. J Comput Chem 1991, 12, 1058.

36. Almlöf, J. in Lecture notes in Quantum Chemistry II, Roos, B. O., editor, Springer, Berlin, 1994, pp. 1-90.

Chapter III

A New Parallel Algorithm for MP2 Energy Gradient Calculations

(Ishimura,K; Pulay, P.; Nagase S. J Comput Chem 2007, 28, 2034.)

3.1 Abstract

A new parallel algorithm has been developed for calculating the analytic energy derivatives of full accuracy second order Møller-Plesset perturbation theory (MP2). Its main projected application is optimization of geometries of large molecules in which non-covalent interactions play a significant role. The algorithm is based on the two-step MP2 energy calculation algorithm developed recently and implemented into the quantum chemistry program GAMESS. Timings are presented for test calculations on taxol ($C_{47}H_{51}NO_{14}$) with the 6-31G and 6-31G(d) basis sets (660 and 1032 basis functions, 328 correlated electrons) and luciferin ($C_{11}H_8N_2O_3S_2$) with aug-cc-pVDZ and aug-cc-pVTZ (530 and 1198 basis functions, 92 correlated electrons). The taxol 6-31G(d) calculations are also performed with up to 80 CPU cores. The results demonstrate the high parallel efficiency of the program.

3.2 Introduction

The determination of molecular geometries is one of the basic tasks for quantum chemistry. The most widely used computational method for large molecules is density functional theory (DFT). However, the commonly used functionals are unable to describe the dispersive component of non-covalent interaction which is important for host-guest molecules, self-assembly, molecular recognition, etc. There is much work carried out in this area, both by designing new functionals and by adding semiempirical or empirical correction terms to standard DFT but no generally accepted method has emerged yet. The least expensive *ab initio* method that includes dispersion in a theoretically correct way is second-order Møller-Plesset perturbation theory (MP2)[1]. If dispersion is a significant component of the intermolecular interaction, it is important to optimize molecular geometries at the MP2 level, and this generally requires the calculation of analytical forces of the MP2 energy. Both MP2 energies and in particular MP2 forces are computationally much more demanding and higher scaling than DFT forces.

Analytical MP2 gradients were first implemented by Pople *et al*. in 1979,[2] and since then several groups[3-11] have proposed modified algorithms. In general, direct and semi-direct algorithms require very large memory space, of the order $O(n^4)$ for the storage of two-electron molecular orbital (MO) integrals and intermediate data, where n is the number of basis functions. To reduce the required memory size, multiple pass techniques[3-6] were introduced in which two-electron atomic orbital (AO) integrals and their derivatives are generated many times. This reduces the required memory size to $O(n^2)$ or $O(n^3)$ at the expense of CPU time. Saebø *et al*.[7] developed an AO based MP2 gradient algorithm in which intermediate data of $O(n^4)$ order are stored on disk. This reduces the required random-access memory size to $O(n^2)$ without increasing the

number of CPU cycles, allowing larger calculations. However, all are serial algorithms and limited by CPU, memory and disk capacity. Nielsen[8] developed a parallel direct algorithm in which multiple passes are introduced. The penalty is that AO integrals and their derivatives are recomputed many times. Fletcher *et al.*[9-11] also developed a parallel algorithm using Distributed Data Interface,[12] which was implemented in the quantum chemistry program GAMESS.[13] This algorithm stores all intermediate data in a distributed fashion in the aggregate fast memory of the computer cluster. The total fast memory needed still scales as $O(n^4)$ but individual CPUs need only $O(n^4)/P$ where P is the number of CPUs. This algorithm is best suited to massively parallel computers. Fletcher *et al.*[9-11] report high parallel efficiency using fast interconnect between the CPUs. Several less expensive MP2 gradient methods have been proposed in order to reduce the computational time and resources. Local MP2[14,15] and Density Fitting (Resolution of Identity, RI) MP2[16,17] can treat very large molecules by localizing orbitals or introducing auxiliary basis sets. However, these methods have to be calibrated against the full MP2 gradient because they include approximations or cut-offs.

In this paper, we present a new parallel closed-shell MP2 energy gradient algorithm based on the MP2 energy algorithm that we have developed recently.[18] In the energy algorithm, a two-step parallelization strategy was used. In the first step, the parallelized outermost loop goes over an AO shell index. The first step includes AO integral generation and the first three quarter transformations. The second step includes the final quarter transformation and MP2 energy calculation, and is parallelized by the occupied MO index pair. We have obtained high parallel efficiency and overall computational speed using a modest PC cluster connected by gigabit Ethernet because there are no redundant calculations, and the total amount of network communication is

constant regardless of the number of CPUs. Based on the two-step parallelization idea, we have developed a fast MP2 gradient program with high parallel efficiency that is appropriate for intermediate parallelization.

3.3 Theory

In the following, *i*, *j*, and *k* denote occupied MOs, *I* and *J* denote frozen-core occupied MOs, *a*, *b*, and *c* denote virtual MOs, *A* and *B* denote frozen virtual MOs, *p* and *q* denote any MOs, and μ, ν, λ, and σ denote AOs. The summation ranges are denoted as *occ*, *oact*, *vall*, *vact*, and *all*, and refer to all occupied MOs, active occupied MOs, all virtual MOs, active virtual MOs, and all MOs, respectively. The closed-shell MP2 energy is given by

$$E_{\mathrm{MP2}} = \sum_{ij}^{oact} \sum_{ab}^{vact} t_{ij}^{ab} (ia \mid jb), \tag{1}$$

where the spatial orbital amplitude is defined as

$$t_{ij}^{ab} = \frac{2(ia \mid jb) - (ib \mid ja)}{D_{ij}^{ab}}, \tag{2}$$

$$(ia \mid jb) = \sum_{\mu\nu\lambda\sigma}^{\mathrm{AO}} C_{\mu i} C_{\nu a} C_{\lambda j} C_{\sigma b} (\mu\nu \mid \lambda\sigma), \tag{3}$$

$$D_{ij}^{ab} = \varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b . \tag{4}$$

$(ia \mid jb)$ is a two-electron integral in Mulliken notation and ε_p is an orbital energy. Following Aikens *et al.*,[11] the derivative of the closed-shell SCF+MP2 energy with respect to a nuclear Cartesian coordinate *x* can be written in the MO basis as

$$E_{\mathrm{SCF+MP2}}^{x} = 2\sum_{i}^{occ} H_{ii}^{x} + \sum_{ij}^{occ} \left[2(ii \mid jj)^{x} - (ij \mid ij)^{x} \right] + \sum_{pq}^{all} P_{pq}^{(2)} \left\{ 2H_{pq}^{x} + \sum_{k}^{occ} \left[2(pq \mid kk)^{x} - (pk \mid qk)^{x} \right] \right\}$$

$$+\sum_{ij}^{occ} S_{ij}^{x} W_{ij}^{(2)}[I]+\sum_{ab}^{vall} S_{ab}^{x} W_{ab}^{(2)}[I]+\sum_{i}^{occ}\sum_{a}^{vact} S_{ai}^{x} W_{ai}^{(2)}[I]+$$

$$+\sum_{ij}^{occ} S_{ij}^{x} W_{ij}^{(2)}[II]+\sum_{ab}^{vall} S_{ab}^{x} W_{ab}^{(2)}[II]+\sum_{i}^{occ}\sum_{a}^{vact} S_{ai}^{x} W_{ai}^{(2)}[II]+\sum_{ij}^{occ} S_{ij}^{x} W_{ij}^{(2)}[III]+2\sum_{ij}^{oact}\sum_{ab}^{vact} t_{ij}^{ab}(ia\,|\,jb)^{x}$$

(5)

where

$$W_{ij}^{(2)}[I]=-2\sum_{k}^{oact}\sum_{ab}^{vact} t_{ik}^{ab}(ja\,|\,kb), \tag{6}$$

$$W_{ab}^{(2)}[I]=-2\sum_{ij}^{oact}\sum_{c}^{vact} t_{ij}^{ac}(ib\,|\,jc), \tag{7}$$

$$W_{ai}^{(2)}[I]=-4\sum_{jk}^{oact}\sum_{b}^{vact} t_{jk}^{ab}(ij\,|\,kb), \tag{8}$$

$$W_{ij}^{(2)}[II]=-\frac{1}{2}P_{ij}^{(2)}(\varepsilon_i+\varepsilon_j), \tag{9}$$

$$W_{ab}^{(2)}[II]=-\frac{1}{2}P_{ab}^{(2)}(\varepsilon_a+\varepsilon_b), \tag{10}$$

$$W_{ai}^{(2)}[II]=-P_{ai}^{(2)}\varepsilon_i, \tag{11}$$

and

$$W_{ij}^{(2)}[III]=-\sum_{pq}^{all} P_{pq}^{(2)}[2(pq\,|\,ij)-(pi\,|\,qj)]. \tag{12}$$

The occupied-occupied and virtual-virtual one-particle density matrix elements are defined as

$$P_{IJ}^{(2)}=0, \tag{13}$$

$$P_{iJ}^{(2)}=P_{Ji}^{(2)}=2\sum_{k}^{oact}\sum_{ab}^{vact} t_{ik}^{ab}\frac{(Ja\,|\,kb)}{\varepsilon_i-\varepsilon_J}, \tag{14}$$

$$P_{ij}^{(2)} = -2\sum_{k}^{oact}\sum_{ab}^{vact} t_{ik}^{ab}\frac{(ja\,|\,kb)}{D_{jk}^{ab}}, \tag{15}$$

$$P_{AB}^{(2)} = 0, \tag{16}$$

$$P_{aB}^{(2)} = P_{Ba}^{(2)} = 2\sum_{ij}^{oact}\sum_{c}^{vact} t_{ij}^{ac}\frac{(iB\,|\,jc)}{\varepsilon_a - \varepsilon_B}, \tag{17}$$

and

$$P_{ab}^{(2)} = 2\sum_{ij}^{oact}\sum_{c}^{vact} t_{ij}^{ac}\frac{(ib\,|\,jc)}{D_{ij}^{bc}}. \tag{18}$$

The virtual-occupied one-particle density matrix terms are obtained by solving the following coupled perturbed Hartree-Fock (CPHF) equations derived from the Z-vector method,[19]

$$\sum_{j}^{oall}\sum_{b}^{vall}\left\{4(ai\,|\,bj)-(aj\,|\,bi)-(ab\,|\,ij)+\delta_{ij}\delta_{ab}(\varepsilon_b-\varepsilon_j)\right\}P_{bj}^{(2)} = -L_{ai}, \tag{19}$$

where the MP2 Lagrangian is written as

$$L_{ai} = \sum_{jk}^{occ} P_{jk}^{(2)}\left\{4(ia\,|\,jk)-2(ik\,|\,aj)\right\} + \sum_{bc}^{vall} P_{bc}^{(2)}\left\{4(ia\,|\,bc)-2(ib\,|\,ac)\right\} - 4N_a\sum_{jk}^{oact}\sum_{b}^{vact} t_{jk}^{ab}(ij\,|\,kb) + 4N_i\sum_{j}^{oact}\sum_{bc}^{vact} t_{ij}^{bc}(ab\,|\,jc), \tag{20}$$

$$N_i \equiv \begin{cases} 1, & \text{for } i = \text{active occupied} \\ 0, & \text{for } i = \text{core occupied} \end{cases}$$

and

$$N_a \equiv \begin{cases} 1, & \text{for } a = \text{active virtual} \\ 0, & \text{for } a = \text{frozen virtual} \end{cases}.$$

Although most of the terms in Eqs. 5-20 are calculated in MO basis, some terms are evaluated in AO basis to reduce the computational time or disk space in this new algorithm. The first, second, and third terms in Eq. 5 are calculated in the AO basis,

$$2\sum_{i}^{occ} H_{ii}^{x} + \sum_{ij}^{occ}\left[2(ii\,|\,jj)^{x} - (ij\,|\,ij)^{x}\right] + \sum_{pq}^{all} P_{pq}^{(2)}\left\{H_{pq}^{x} + \sum_{k}^{occ}\left[2(pq\,|\,kk)^{x} - (pk\,|\,qk)^{x}\right]\right\}$$

$$= 2\sum_{\mu\nu}^{AO}\left(P_{\mu\nu}^{(2)} + P_{\mu\nu}^{HF}\right)H_{\mu\nu}^{x} + \sum_{\mu\nu\lambda\sigma}^{AO}\left[2\left(P_{\mu\nu}^{(2)} + P_{\mu\nu}^{HF}\right)P_{\lambda\sigma}^{HF} - \left(P_{\mu\lambda}^{(2)} + P_{\mu\lambda}^{HF}\right)P_{\nu\sigma}^{HF}\right](\mu\nu\,|\,\lambda\sigma)^{x}, \qquad (21)$$

where $P_{\mu\nu}^{HF}$ is the SCF density matrix and

$$P_{\mu\nu}^{(2)} = \sum_{pq}^{\text{all}} C_{\mu p} C_{\nu q} P_{pq}^{(2)}\,. \qquad (22)$$

The first, second, and fourth terms in the MP2 Lagrangian (Eq. 19) are calculated in the AO basis,

$$\sum_{jk}^{occ} P_{jk}^{(2)}\left\{4(ia\,|\,jk) - 2(ij\,|\,ak)\right\} + \sum_{bc}^{vall} P_{bc}^{(2)}\left\{4(ia\,|\,bc) - 2(ib\,|\,ac)\right\} = \sum_{\mu\nu}^{AO} C_{\mu i} C_{\nu a} L_{\mu\nu}^{1,2} \qquad (23)$$

and

$$\sum_{j}^{oact}\sum_{bc}^{vact} t_{ij}^{bc}(ab\,|\,jc) = \sum_{\mu}^{AO} C_{\mu a} L_{\mu i}^{4} \qquad (24)$$

where

$$L_{\mu\nu}^{1,2} = \sum_{\lambda\sigma}^{AO} P_{\lambda\sigma}^{'(2)}\left\{4(\mu\nu\,|\,\lambda\sigma) - 2(\mu\lambda\,|\,\nu\sigma)\right\}, \qquad (25)$$

$$P_{\lambda\sigma}^{'(2)} = \sum_{jk}^{occ} C_{\lambda j} C_{\sigma k} P_{jk}^{(2)} + \sum_{bc}^{vall} C_{\lambda b} C_{\sigma c} P_{bc}^{(2)}\,, \qquad (26)$$

$$L_{\mu i}^{4} = \sum_{j}^{oact}\sum_{\nu\sigma}^{AO} t_{ij}^{\nu\sigma}(\mu\nu\,|\,j\sigma), \qquad (27)$$

and

$$t_{ij}^{\nu\sigma} = \sum_{ab}^{\text{vact}} C_{\nu a} C_{\sigma b} t_{ij}^{ab}\,. \qquad (28)$$

$W_{ij}^{(2)}[III]$ in Eq. 5 is calculated in AO basis,

$$W_{ij}^{(2)}[III] = \sum_{\lambda\sigma}^{AO} C_{\lambda i} C_{\sigma j} W_{\lambda\sigma}^{(2)}[III], \tag{29}$$

where

$$W_{\lambda\sigma}^{(2)}[III] = -\sum_{\mu\nu}^{AO} P_{\mu\nu}^{(2)} \left[2(\mu\nu|\lambda\sigma) - (\mu\lambda|\nu\sigma) \right]. \tag{30}$$

The overlap integral terms are also evaluated in AO basis,

$$\sum_{pq}^{\text{all}} S_{pq}^{x} W_{pq}^{(2)}[I] + \sum_{pq}^{\text{all}} S_{pq}^{x} W_{pq}^{(2)}[II] + \sum_{ij}^{\text{occ}} S_{ij}^{x} W_{ij}^{(2)}[III] = \sum_{\mu\nu}^{\text{AO}} S_{\mu\nu}^{x} W_{\mu\nu}^{(2)} \tag{31}$$

where

$$W_{\mu\nu}^{(2)} = \sum_{pq}^{\text{AO}} C_{p\mu} C_{q\nu} W_{pq}^{(2)}[I] + \sum_{pq}^{\text{AO}} C_{p\mu} C_{q\nu} W_{pq}^{(2)}[II] + \sum_{ij}^{\text{occ}} C_{i\mu} C_{j\nu} W_{ij}^{(2)}[III]. \tag{32}$$

The last term in Eq. 5 is calculated in the AO basis,

$$\sum_{ij}^{\text{oact}} \sum_{ab}^{\text{vact}} t_{ij}^{ab} (ia|jb)^{x} = \sum_{\mu\nu\lambda\sigma}^{\text{AO}} t_{\mu\lambda}^{\nu\sigma} (\mu\nu|\lambda\sigma)^{x} \tag{33}$$

where

$$t_{\mu\lambda}^{\nu\sigma} = \sum_{ij}^{\text{oact}} \sum_{ab}^{\text{vact}} C_{\mu i} C_{\nu a} C_{\lambda j} C_{\sigma b} t_{ij}^{ab} \,. \tag{34}$$

3.4 MP2 gradient algorithm

The pseudocode of the new algorithm is shown in Figure 1. The formal computational costs, required memory and disk sizes, and the total amount of network communication are shown in Table 1. We first describe the serial version. The algorithm consists of 5 steps. Capital Greek letters, *M*, *N*, *Λ*, and *Σ* denote shells of AOs.

do Λ (distributed dynamically)

do M

do $N(\leq M)$

Schwartz prescreening of $(MN|\Lambda\Sigma)$

calculate $(\mu\nu|\lambda\sigma)$ for all σ, $\mu\in M$, $\nu\in N$, and $\lambda\in\Lambda$

transform to $(\mu\nu|\lambda j)$ for all j, $\mu\in M$, $\nu\in N$, and $\lambda\in\Lambda$

screen of $(\mu\nu|\lambda j)$

transform to $(\mu i|\lambda j)$ for all i, j, $\mu\in M$, and $\lambda\in\Lambda$

end do N

end do M

transform to $(ai|\lambda j)$ and save on disk for all i, j, a, and $\lambda\in\Lambda$

end do Λ

do a (distributed statically)

read $(ai|\lambda j)$ from disk and send to an appropriate CPU for all i, j, λ on this CPU, and one a

receive $(ai|\lambda j)$ from other CPUs for all i, j, λ, and one a

transform $(ai|bj)$ for all i, j, b, and one a

transform $(ai|kj)$ for all i, j, k, and one a

calculate MP2 energy

calculate t_{ij}^{ab}, $W_{ij}^{(2)}[I]$, $W_{ab}^{(2)}[I]$, $W_{ai}^{(2)}[I]$, $P_{ij}^{(2)}$, and $P_{ab}^{(2)}$ for all i, j, b, and one a

transform to $t_{ij}^{a\sigma}$ and send to an appropriate CPU for all i, j, σ, and one a

receive $t_{ij}^{a\sigma}$ and store on disk for all i, j, σ on this CPU, and a from other CPUs

end do a

ccumulate and broadcast MP2 energy, $W_{ij}^{(2)}[I]$, $W_{ab}^{(2)}[I]$, $W_{ai}^{(2)}[I]$, $P_{ij}^{(2)}$, and $P_{ab}^{(2)}$ for all i, j, a, and b

calculate $W_{ij}^{(2)}[II]$, $W_{ab}^{(2)}[II]$, and $P_{\lambda\sigma}^{(2)}$ for all i, j, a, b, λ, and σ

do Σ (distributed statically)

read $t_{ij}^{a\sigma}$ from disk for all i, j, a, and $\sigma \in \Sigma$

transform to $t_{ij}^{\nu\sigma}$ and store on disk for all i, j, ν, and $\sigma \in \Sigma$

do M

do $N (\leq M)$

Schwartz prescreening of $(MN|\Lambda\Sigma)$

calculate $(\mu\nu|\lambda\sigma)$ for all λ, $\mu \in M$, $\nu \in N$, and $\sigma \in \Sigma$

calculate $L^{1,2}_{\mu\nu}$ for all $\mu \in M$, and $\nu \in N$

transform to $(\mu\nu|j\sigma)$ for all j, $\mu \in M$, $\nu \in N$, and $\sigma \in \Sigma$

calculate $L^{4}_{\mu i}$ for all i and $\mu \in M$

end do N

end do M

end do Σ

accumulate and broadcast $L^{1,2}_{\mu\nu}$ and $L^{4}_{\mu i}$ for all i, μ, and ν

calculate L_{ai} for all i and a

do CPHF cycle

do M

do $N(\leq M)$ (distributed dynamically)

Schwartz prescreening of $(MN|\Lambda\Sigma)$

calculate $(\mu\nu|\lambda\sigma)$ for all λ, σ, $\mu \in M$, and $\nu \in N$

calculate Fock-like matrix of CPHF equations

end do N

end do M

accumulate Fock-like matrix

solve CPHF equations in the AO basis

end do CPHF cycle

obtain and broadcast $P_{ai}^{(2)}$

calculate $W_{ai}^{(2)}[II]$, $P_{\mu\nu}^{(2)}$, $W_{ij}^{(2)}[III]$, and $W_{\mu\nu}^{(2)}$ all i, j, a, μ, and ν

do M (distributed dynamically)

calculate $H_{\mu\nu}^{x}$ and $S_{\mu\nu}^{x}$ terms for all $\mu \in M$ and $\nu \leq \mu$

end do M

do Σ (distributed statically)

do N

transform to $t_{i\lambda}^{\nu\sigma}$ for all i, λ, $\nu \in N$, and $\sigma \in \Sigma$

do Λ

prescreen $(MN|\Lambda\Sigma)^{x}$

transform to $t_{\mu\lambda}^{\nu\sigma}$ for all μ, $\nu \in N$, $\lambda \in \Lambda$, and $\sigma \in \Sigma$

calculate $(\mu\nu|\lambda\sigma)^x$ terms for all $\mu(\leq\nu)$, $\nu \in N$, $\lambda \in \Lambda$, and $\sigma \in \Sigma$

end do Λ

end do N

end do Σ

Figure 1. Outline of the MP2 Gradient Algorithm.

Table 1. Formal Flop Count, Required Memory Size, Total Disk Size, and Total Amount of Network Communication of Each Step[a].

	Flop	Memory	Disk	Communication
Step 1	$O(n^4)+O(o'n^4)+O(o'on^3)+O(o'ovn^2)$	$s^3n+s^3o'+so'on$+disk cache	$o'ovn$	
Step 2	$O(o'ovn^2)+O(o'v'n^3)+O(o'^2v'^2n)$	$o'on+2o'^2v'+o'^2n+3n^2+[o'^2n]$[b]	$o'ovn+o'^2v'n$	$o'ovn+o'^2v'n$
Step 3	$O(o'^2v'n^2)+O(n^4)+O(o'n^4)+O(o'^2n^3)$	$so'^2v'+o'^2v'+s^3n+n^2+s^3o'+o'n$	o'^2n^2	
Step 4	$O(n^4)$	$4n^2$		
Step 5	$O(n^2)+O(o'n^4)+O(o'^2n^3)+O(n^4)$	$s^2o'^2+s^2o'n+s^4+2n^2$	o'^2n^2	

[a]Abberviations: n = number of basis functions; o = number of occupied MOs; o' = number of active occupied MOs; s = number of basis functions in a shell, e.g., 4 for a *sp* shell; v = number of virtual MOs, v' = number of active virtual MOs

[b]Extra memory required for parallel calculation is shown in brackets.

Step 1: The integral transformation part is based on the algorithm of MP2 energy calculations developed recently.[18] The outermost loop up to the third quarter transformation is over AO shell Λ. An AO integral block ($MN|\Lambda\Sigma$) is generated for one M, N, Λ, and all Σ. ($MN|\Lambda\Sigma$) denotes all AO integrals ($\mu\nu|\lambda\sigma$) for $\mu\in M$, $\nu\in N$, $\lambda\in\Lambda$, and $\sigma\in\Sigma$. Before the AO integral generation, Schwartz prescreening[20-22] is employed to skip the calculation of insignificant integrals, as in the SCF calculation. The computational cost in this step is formally O(n^4), but actually O(n^2~n^3) because of the screening, where n is the total number of basis functions. The required memory size is s^3n, where s is the maximum number of basis functions in a shell, for instance, 1 for s function and 4 for sp function. Only one of the three permutation symmetries, ($\mu\nu|\lambda\sigma$)=($\nu\mu|\lambda\sigma$) is used in the algorithm, that is, the same AO integrals are generated 4 times. This penalty is, however, small, as Pulay *et al*. pointed out for MP2 energy calculations.[23] After the generation of AO integral blocks, the first quarter transformation,

$$(\mu\nu \mid \lambda j) = \sum_{\sigma}^{AO} C_{\sigma j} (\mu\nu \mid \lambda\sigma) \qquad (35)$$

is performed for all active occupied MOs, j, $\mu\in M$, $\nu\in N$, and $\lambda\in\Lambda$. The formal computational cost is O($o'n^4$) and the memory size is s^3o', where o' is the number of active occupied MOs. The second quarter transformation,

$$(\mu i \mid \lambda j) = \sum_{\nu}^{AO} C_{\nu i} (\mu\nu \mid \lambda j) \qquad (36)$$

is performed for all occupied MOs, i, j, $\mu\in M$, and $\lambda\in\Lambda$, then the half-transformed integrals ($\mu i|\lambda j$) are accumulated. The formal computational cost is O($o'on^3$) and the memory size is $so'on$, where o is the number of all occupied MOs. The ($\mu\nu|\lambda j$) integrals are screened prior to this transformation. The third quarter transformation,

$$(ai \mid \lambda j) = \sum_{\mu}^{\mathrm{AO}} C_{\mu a} (\mu i \mid \lambda j) \qquad (37)$$

is performed for all virtual MOs, *a*, *i*, *j*, and $\lambda \in \Lambda$, then the integrals ($ai|\lambda j$) are stored on disk. The computational cost is O($o'ovn^2$), where *v* is the number of all virtual MOs. The memory size is the same size of disk cache, for instance, 8 or 16MB, to reduce disk I/O time. High CPU efficiency is achieved in this step because writing data to the disk cache is more than 10 times faster than to the disk itself. Screening of ($\mu i|\lambda j$) is not exploited in this transformation, as canonical orbitals are delocalized, making the screening ineffective. The disk storage requirement is $o'ovn$.

Step 2: The outermost loop is over virtual MO, *a*. A block of ($ai|\lambda j$) integrals for all *i*, *j*, λ, and one *a*, is read from disk. The fourth quarter transformations,

$$(ai \mid bj) = \sum_{\lambda}^{AO} C_{\lambda b} (ai \mid \lambda j), \qquad (38)$$

for all *i*, *j*, virtual MOs, *b*, and one *a* and

$$(ai \mid kj) = \sum_{\lambda}^{AO} C_{\lambda k} (ai \mid \lambda j), \qquad (39)$$

for all *i*, *j*, occupied MOs, *k*, and one *a* are performed. The computational cost is O($o'ovn^2$) and the memory size is $2o'on$. Using these MO integrals, the MP2 energy, t_{ij}^{ab}, $W_{ij}^{(2)}[I]$, $W_{ab}^{(2)}[I]$, $W_{ai}^{(2)}[I]$, $P_{ij}^{(2)}$, and $P_{ab}^{(2)}$ in Eqs. 1, 2, 6-8, and 13-18 are calculated. The computational cost is O($o'v'n^3$) and the memory size is $o'on+2o'^2v'+o'^2n+3n^2$, where v' is the number of active virtual MOs. The first back-transformation,

$$t_{ij}^{a\sigma} = \sum_{b}^{vact} C_{\sigma b} t_{ij}^{ab} \qquad (40)$$

is performed for all active MOs *i*, *j*, σ and one *a*, and $t_{ij}^{a\sigma}$ is stored on disk. The

computational cost is O($o'^2v'^2n$) and the memory size is o'^2n. The disk storage size is $o'^2v'n$. At the end of this step, $W_{ij}^{(2)}[II]$, $W_{ab}^{(2)}[II]$, and $P_{\lambda\sigma}^{\prime(2)}$ in Eqs. 9, 10, and 26 are calculated.

Step 3: The outermost loop is over AO shell Σ. $t_{ij}^{a\sigma}$ is read from disk and the second back-transformation,

$$t_{ij}^{\nu\sigma} = \sum_{a}^{vact} C_{\nu a} t_{ij}^{a\sigma} \tag{41}$$

is performed for all active MOs, i and j, ν, and $\sigma\in\Sigma$, and $t_{ij}^{\nu\sigma}$ is stored on disk. $t_{ij}^{\nu\sigma}$ is overwritten on the $t_{ij}^{a\sigma}$ file. The computational cost is O($o'^2v'n^2$) and the memory size is (s+1)o'^2v'. The disk storage size is o'^2n^2. Schwartz prescreening for AO integrals is performed, then an AO integral block ($MN|\Lambda\Sigma$) is generated for one M, N, Σ, and all Λ. One permutation symmetry ($\mu\nu|\lambda\sigma$)=($\nu\mu|\lambda\sigma$) is also used in this step. $L_{\mu\nu}^{1,2}$ in Eq. 25 is calculated for all $\mu\in M$ and $\nu\in N$. The formal computational cost and the memory size for ($MN|\Lambda\Sigma$) and $L_{\mu\nu}^{1,2}$ are O(n^4) and s^3n+n^2, respectively. The first transformation,

$$(\mu\nu \mid j\sigma) = \sum_{\lambda}^{AO} C_{\lambda j}(\mu\nu \mid \lambda\sigma) \tag{42}$$

is performed for all j, $\mu\in M$, $\nu\in N$, and $\sigma\in\Sigma$. $L_{\mu i}^{4}$ in Eq. 27 is calculated for all i and $\mu\in M$. The formal computational cost and the memory size for the first transformation and $L_{\mu i}^{4}$ are O($o'n^4$)+O(o'^2n^3) and s^3o'+$o'n$. After the Σ loop finishes, full L_{ai} in Eq. 20 is calculated using $L_{\mu\nu}^{1,2}$ and $L_{\mu i}^{4}$.

Step 4: CPHF equations are solved in AO basis[24-26] using the DIIS method[27] to calculate $P_{ai}^{(2)}$. The outermost loop is over AO shell, M, and the next is over AO shell, N. The formal computational cost is O(n^4) and the memory size is $4n^2$. After $P_{ai}^{(2)}$ is converged, $W_{ai}^{(2)}[II]$, $P_{\mu\nu}^{(2)}$, $W_{ij}^{(2)}[III]$, and $W_{\mu\nu}^{(2)}$ in Eqs. 11, 22, 29, 30, and 32 are

calculated.

Step 5: The derivative terms of the core Hamiltonian integral $H^x_{\mu\nu}$ and the overlap integral $S^x_{\mu\nu}$ are calculated for all μ and ν. The outermost loop is over AO shell, M, during this calculation. The computational cost and the memory size are O(n^2) and $4n^2$, respectively. The third and fourth back-transformations,

$$t^{\nu\sigma}_{i\lambda} = \sum_{j}^{oact} C_{\lambda j} t^{\nu\sigma}_{ij} \tag{43}$$

and

$$t^{\nu\sigma}_{\mu\lambda} = \sum_{i}^{oact} C_{\mu i} t^{\nu\sigma}_{i\lambda} \tag{44}$$

are performed and the derivative terms of the two-electron integral $(\mu\nu|\lambda\sigma)^x$ are calculated for $\mu\in M$, $\nu\in N$, $\sigma\in\Sigma$, and $\lambda\in\Lambda$. The outermost loop is over AO shell Σ. Only one permutation symmetry, $(\mu\nu|\lambda\sigma)^x=(\nu\mu|\lambda\sigma)^x$, is used. The formal computational cost and the memory is O($o'n^4$)+O(o'^2n^3)+O(n^4) and $s^2o'^2+s^2o'n+s^4+2n^2$, respectively. This step yields the final SCF+MP2 energy gradient values.

The required memory sizes in Steps 1 and 3 can be reduced by introducing multiple passes, in which AO integrals are calculated several times up to the number of basis function in a shell. The penalty is small compared with the total cost of the MP2 gradient calculation.

The framework of the parallel version is the same as that of the serial version. In Step 1, AO shells Λ of the outermost loop are dynamically distributed to each CPU. In Step 2, virtual MOs a of the outermost loop are statically distributed. ($ai|\lambda j$) is read from disk and sent to an appropriate CPU before the fourth transformations. After the first back-transformation, $t^{a\sigma}_{ij}$ is sent to an appropriate CPU. The AO shell indices Σ for $t^{a\sigma}_{ij}$ are statically distributed. The extra required memory sizes for receiving data

are $o'on$ for $(ai|\lambda j)$ and o'^2n for $t_{ij}^{a\sigma}$. The MP2 energy, $P_{ij}^{(2)}$, $P_{ab}^{(2)}$, $W_{ij}^{(2)}[I]$, $W_{ab}^{(2)}[I]$, and $W_{ai}^{(2)}[I]$ are accumulated to the master CPU at the end of the loop and broadcasted to all CPUs. $W_{ij}^{(2)}[II]$, $W_{ab}^{(2)}[II]$, and $P_{\lambda\sigma}^{\prime(2)}$ are calculated in all CPUs using full $P_{ij}^{(2)}$ and $P_{ab}^{(2)}$. The penalty is negligible because the cost is O(n^3). In Step 3, AO shells Σ of the outermost loop are distributed as decided in Step 2. The information of this Σ distribution is kept until Step 5. At the end of the step, $L_{\mu\nu}^{1,2}$ and $L_{\mu i}^{4}$ are accumulated and broadcasted. Finally, L_{ai} is calculated in all CPUs. In Step 4, AO shells N of the second outermost loop are distributed dynamically and the CPHF equations are solved iteratively. After $P_{ai}^{(2)}$ is converged, $W_{ai}^{(2)}[II]$, $P_{\mu\nu}^{(2)}$, $W_{ij}^{(2)}[III]$, and $W_{\mu\nu}^{(2)}$ are calculated in all CPUs. In Step 5, AO shells M of the outermost loop are dynamically distributed during the derivative calculation of the one-electron and overlap integrals. AO shells Σ of the outermost loop are statically distributed as decided in Step 2 during the derivative calculation of the two-electron integrals. At the end of the step, partial MP2 gradient values are accumulated.

Because we generate each AO integral only once, and do not broadcast all intermediate integrals to all CPUs in the two-step parallelization, total computational cost and the total disk storage size are the same as those of the serial version and the total amount of data communication is essentially constant, $o'ovn$ for $(ai|\lambda j)$ and $o'^2v'n$ for $t_{ij}^{a\sigma}$ in Step 2. Furthermore, all fourth and fifth order calculations are parallelized by distributing AO or MO indices.

3.5 Results and Discussion

The algorithm was implemented into the GAMESS package. Test calculations on taxol ($C_{47}H_{51}NO_{14}$, no symmetry) and luciferin ($C_{11}H_8N_2O_3S_2$, no symmetry) were

performed on a 3.2-GHz Pentium4 cluster connected by a gigabit Ethernet. Each CPU has 4-GB dual-PC3200 DDR2 memory and a 400-GB hard disk with an 8-MB cache.

The 6-31G and 6-31G(d) basis sets were used for taxol, and the aug-cc-pVDZ and aug-cc-pVTZ basis sets were used for luciferin. Only the occupied core MOs were frozen. Table 2 summarizes the details of the calculations, the required memory size per CPU, the total required disk size, and the total amount of network communication. An integral screening threshold of 10^{-10} was used in all calculations on taxol and 10^{-11} on luciferin. The tighter threshold is required in the luciferin calculation because of the near-singularity of the augmented basis sets. The required memory and disk sizes are moderate, so that test calculations can be performed on one or two PCs.

Table 2. Details of the Calculations.

	Taxol		Luciferin	
	6-31G	6-31G(d)	aug-cc-pVDZ	aug-cc-pVTZ
Number of contracted basis functions	660	1032	530	1198
Number of shells in the basis set	288	350	206	328
Number of correlated electrons	328	328	92	92
Number of occupied MOs	226	226	72	72
Number of active occupied MOs	164	164	46	46
Number of virtual MOs[a]	434	806	422	946
Number of SCF cycles	16	16	24	23
Number of CPHF cycles	21	21	22	21
Required memory size per CPU/GB	0.78	1.84	0.09	0.33
Total required disk size/GB	147	426	10	49
Total amount of network communication/GB	147	426	10	49

[a]Same as the number of active virtual MOs.

Table 3 shows single processor CPU and elapsed timings for the SCF procedure and for each step of the MP2 gradient calculation as described in the previous section. We could not complete the calculation for taxol with the 6-31G(d) basis (n=1032) because the total required disk space was over 400GB. The most CPU intensive steps are Step 1 for taxol and Step 4 for luciferin. The ratios of the number of occupied MOs and basis functions are 2.9 for taxol, 7.4 for luciferin aug-cc-pVDZ, and 16.6 for luciferin aug-cc-pVTZ. The number of occupied MOs is relatively large for taxol. Step 1 (from the AO integral generation to the third quarter integral transformation) takes much time for taxol because the maximum formal computational cost during the transformations has terms that scale formally as O($o'on^3$), in addition to the first quarter transformation that scales as O($o'n^4$). In luciferin, the number of basis functions is large but the number of correlated orbitals is small, and the calculations are dominated by the formally O(n^4) integral evaluation step that has to be repeated in every CPHF step. The difference between CPU and elapsed timings comes from disk I/O, especially in Steps 2 and 3. Reading ($ai|\lambda j$) and writing $t_{ij}^{a\sigma}$ are performed in Step 2, and reading $t_{ij}^{a\sigma}$ and writing $t_{ij}^{\nu\sigma}$ are performed in Step 3. The difference of timings in Step 1 is negligible in spite of writing huge amounts of data ($ai|\lambda j$) to disk, for instance, 85GB for the taxol 6-31G calculation. As in the MP2 energy calculation, ($ai|\lambda j$) is written to disk in blocks that match the size of the disk cache. This measure reduces the timing for disk writing very much.

Table 3. Single Processor CPU and Elapsed Times for the SCF Procedure and for the Steps of the MP2 Gradient, in Hours

			SCF	Step 1	Step 2	Step 3	Step 4	Step 5	MP2 gradient (Step 1 to Step 5)
Taxol	6-31G	CPU	1.22	4.14	1.91	2.62	2.19	1.26	12.11
		Elapsed	1.22	4.20	5.01	4.00	2.19	1.96	17.36
Luciferin	aug-cc-pVDZ	CPU	5.75	2.06	0.11	1.66	6.76	2.95	13.54
		Elapsed	5.75	2.06	0.55	2.01	6.76	2.95	14.33
	aug-cc-pVTZ	CPU	67.72	24.54	1.10	22.51	78.67	31.31	158.12
		Elapsed	67.74	24.56	3.76	25.05	78.67	31.54	163.58

Table 4 shows elapsed timing, speedup ratios, and CPU efficiency of the master CPU for parallel calculations from 1 CPU to 32 CPUs and Figure 2 shows speedup ratios for taxol calculations. The speedup on 2 CPUs is set to be 2.0 for taxol 6-31G(d) calculation. The speedups are very good up to the maximum 32 CPUs we have tried. For instance, on 32 CPUs we get speedup ratios ranging from 28.2 to 33.0, corresponding to 88 to 103 % of linear speedup. This indicates that the present program has high parallel efficiency. In some cases, particularly on a small number of CPUs, we got superlinear speedups. CPU efficiency increases at first with the number of CPUs because the block size of the disk reading in Step 2 increases with increasing number of CPUs. Over 4 or 8 CPUs, we experience a slight deterioration in CPU efficiency because it becomes difficult to achieve completely uniform load balancing. However, the effect is small up to 32 CPUs, as shown in Table 4 and Figure 2.

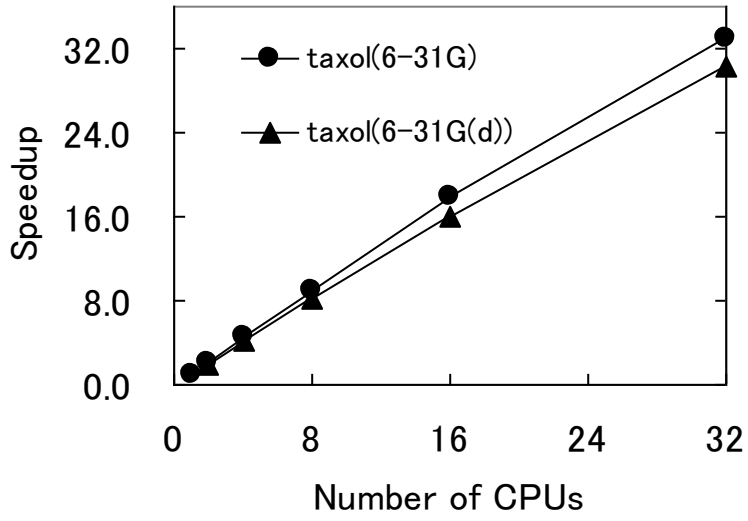


Figure 2. Speedup Ratios for Taxol 6-31G and 6-31G(d) Calculations.

Table 4. Elapsed Times (Hours), Speedups, and CPU efficiencies (%) of the Master CPU for parallel MP2 gradient calculations.

Number of CPUs			1	2	4	8	16	32
Taxol	6-31G	Time	17.36	8.09	3.82	1.92	0.97	0.53
		Speedup	1.0	2.1	4.5	9.0	17.9	33.0
		CPU effi.	69.8	76.9	81.2	78.1	76.7	72.9
	6-31G(d)	Time		31.11	15.11	7.57	3.87	2.05
		Speedup		2.0	4.1	8.2	16.1	30.4
		CPU effi.		80.2	85.6	82.3	78.6	75.8
Luciferin	aug-cc-pVDZ	Time	14.33	6.94	3.44	1.74	0.92	0.50
		Speedup	1.0	2.1	4.2	8.2	15.5	28.9
		CPU effi.	94.5	97.5	97.2	97.8	94.2	91.8
	aug-cc-pVTZ	Time	163.58	81.06	40.66	20.59	10.75	5.80
		Speedup	1.0	2.0	4.0	7.9	15.2	28.2
		CPU effi.	96.7	97.3	98.0	98.2	98.0	95.6

The taxol 6-31G(d) calculation was also performed using dual-core, dual-processor 3.0GHz Xeon workstations to check the parallel efficiency on a larger number of processors. Each node has 4 CPU cores, 8-GB FB-DIMM 667MHz memory, and two 500-GB hard disks. Note that this is an unfavorable configuration for correlated calculations because four processes compete for both fast memory and disk access on the same node. In spite of this, as shown in Figure 3, the program maintains high parallel efficiency for up to 80 CPU cores. The speedup for 4 cores was set to be 4. From these data, we expect that in a more advantageous 2 processor core configuration, reasonably high parallel efficiency would be maintained for 64-128 processors.

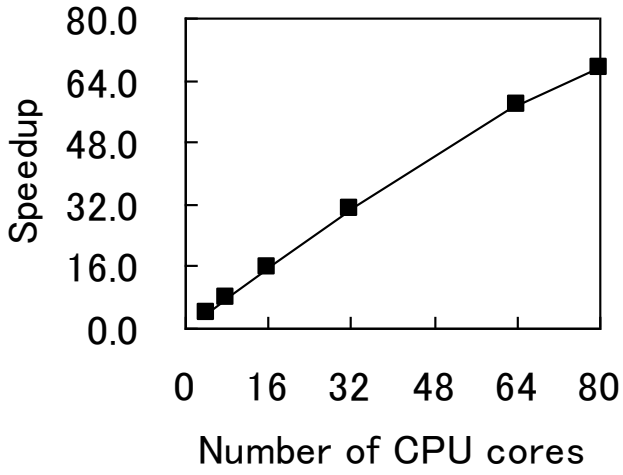


Figure 3. Speedup Ratios for Taxol 6-31G(d) Calculation Using a Dual-core Dual-Processor Xeon Cluster.

3.6 Conclusions

We have developed a new parallel algorithm for MP2 energy gradient calculation

based on the parallel algorithm recently developed for MP2 energy calculation using two-step parallelization. The program allows routine calculation of MP2 gradients for relatively large basis sets (660-1200 basis functions) on medium-sized PC or workstation clusters with good parallel efficiency and CPU utilization.

3.7 References

1. Møller, C.; Plesset, M. S. Phys Rev 1934, 46, 618.

2. Pople, J. A.; Krishnan, R.; Schlegel, H. B.; Binkley, J. S. Int J Quantum Chem, Symp 1979, 13, 225.

3. Frisch, M. J.; Head-Gordon, M.; Pople, J. A. Chem Phys Lett 1990, 166, 275.

4. Frisch, M. J.; Head-Gordon, M.; Pople, J. A. Chem Phys Lett 1990, 166, 281.

5. Head-Gordon, M Mol Phys 1999, 96, 673.

6. Haase, F.; Ahlrichs, R. J Comput Chem 1993, 14, 907.

7. Saebø, S.; Baker, J.; Wolinski, K.; Pulay, P. J Chem Phys 2004, 120, 11423.

8. Nielsen, I. M. B. Chem Phys Lett 1996, 255, 210.

9. Fletcher, G.. D.; Rendell, A. P.; Sherwood, P. Mol Phys 1997, 91, 431.

10. Fletcher, G.. D.; Schmidt, M. W.; Gordon, M. S. Advances in Chemical Physics, Vol. 110, edited by Prigogine I. and Rice S. A. (Wiley, New York, 1999).

11. Aikens, C. M.; Webb, S. P.; Bell, R. L.; Fletcher, G.. D.; Schmidt, M. W.; Gordon, M. S. Theor Chem Acc 2003, 110, 233.

12. Fletcher, G. D.; Schmidt, M. W.; Bode, B. M.; Gordon, M. S. Comput Phys Commun 2000, 128, 190.

13. Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S.; Windus, T. L.; Dupuis, M.;

Montgomery, J. A. J Comput Chem 1993, 14, 1347.

14. ElAzhary, A.; Rauhut, G.; Pulay, P.; Werner, H.-J. J Chem Phys 1998, 108, 5185.

15. Schütz, M.; Wemer, H.-J.; Lindh, R.; Manby, F. R. J Chem Phys 2004, 121, 737.

16. Weigend, F.; Häser, M. Theor. Chem. Acc. 1997, 97, 331.

17. Hättig, C.; Hellweg, A.; Köhn, A. Phys Chem Chem Phys 2006, 8, 1159.

18. Ishimura, K.; Pulay, P.; Nagase S. J Comput Chem 2006, 27, 407.

19. Handy, N. C.; Schaefer, H. F. III J Chem Phys 1984, 81, 5031.

20. Abramowitz, M.; Stegun, I. A., Eds.; Handbook of Mathematical Functions; Dover: New York, 1972.

21. Horn, H.; Weiss, H.; Häser, M.; Ehrig, M.; Ahlrichs, R. J Comput Chem 1991, 12, 1058.

22. Almlöf, J. in Lecture notes in Quantum Chemistry II, Roos, B. O., editor, Springer, Berlin, 1994, pp. 1-90.

23. Pulay, P.; Saebø, S.; Wolinski, K. Chem Phys Lett 2001, 344, 543.

24. Pulay, P.; J Chem Phys 1983, 78, 5043.

25. Osamura, Y.; Yamaguchi, Y.; Saxe, P.; Fox, D. J.; Vincent, M. A.; Schaefer III, H. F. J Mol Struct 1983, 103, 183.

26. Amos, R. D.; Rice, J. E. Comput Phys Rep 1989, 10, 147.

27. Pulay, P. J Comput Chem 1982, 3, 556.

Chapter IV

Applications of MP2 calculations

4.1 Introduction

New parallel algorithms for MP2 energy and gradient calculations are developed in Chapters II and III. In this chapter, these are applied to the energy and geometry calculations of several molecular systems.

Geometries of the lead analogues of alkynes with very bulky substituents are optimized using the developed programs, and are compared with those optimized using the DFT method at the B3PW91 level. MP2 is the simplest method that includes electron correlation important for non-covalent interactions. However, MP2 tends to overestimate the interactions. This shortcoming can be improved by introducing different scaling parameters for the same-spin and opposite-spin components of MP2 correlation energies, as demonstrated by Grimme.[1] To perform the SCS (spin-component scaled)-MP2 calculations, an efficient parallel program is implemented. Geometry optimization and binding energy calculations are performed for platinum complexes as well as benzene-benzene and naphthalene-naphthalene dimers, using both MP2 and SCS-MP2 methods.

4.2 Structures of the lead analogue of alkynes

Multiple bonds between heavier elements are of wide interest in main-group chemistry. Among these, the heavier group 14 element analogues of alkynes, REER (E = Si, Ge, Sn, Pb), have attracted great interest as challenging synthetic targets.[2] Up to now, $R^{si}SiSiR^{si}$ (R^{si} = Si*i*Pr-{$CH(SiMe_3)_2$}$_2$),[3] Ar'GeGeAr' (Ar' = C_6H_3-2,6-(C_6H_3-2,6-*i*Pr_2)$_2$),[4] BbtGeGeBbt (Bbt = C_6H_2-2,6-{$CH(SiMe_3)_2$}$_2$-4-$C(SiMe_3)_3$),[5] Ar'SnSnAr',[6] and Ar*PbPbAr* (Ar* = C_6H_3-2,6-(C_6H_2-2,4,6-*i*Pr_3)$_2$)[7] have been synthesized and isolated. Bulky silyl and aryl groups play an important role in making these heavier analogues synthetically accessible and isolatable as stable compounds because they suppress isomerization and dimerization.[8] As shown by X-ray crystal analysis, the heavier analogues have a trans-bent core skeleton, unlike the alkyne case. The trans-bending is due to the mixing of the low-lying vacant σ* orbital into the in-plane π orbital. This σ* mixing makes the π orbital slipped and weakened. However, the central Si-Si distance of $R^{si}SiSiR^{si}$ is considerably shorter than those of Si-Si double bonds,[3] while the Ge-Ge and Sn-Sn distances of Ar'GeGeAr', BbtGeGeBbt, and Ar'SnSnAr' are close to those of Ge-Ge and Sn-Sn double bonds.[4-6] In sharp contrast, the X-ray crystal analysis of the heaviest analogue, Ar*PbPbAr*, has shown that the trans-bending is greatly increased and the Pb-Pb distance is much longer than the Pb-Pb single-bond distances of diplumbanes such as Ph_3Pb-$PbPh_3$.[7] The singly bonded structure having no Pb-Pb π bond has been also confirmed by model calculations.[9] The exceptional structure of Ar*PbPbAr* has been explained by the fact that the heaviest Pb atom has the strongest tendency to preserve the valence 6s electrons as lone-pair electrons in making bonds (Figure 1).[2,7]

S M

Figure 1. Singly (S) and multiply (M) bonded structures.

In an attempt to investigate whether Ar*PbPbAr* takes a multiply bonded structure or not, density functional calculations at the B3PW91 level have been very recently carried out because the molecular size is considerably large.[8g] Here, geometry optimization was carried out at the MP2 level. The triple-ζ basis set[11] augmented by two sets of d polarization functions (d exponents 0.213 and 0.062)[12] and relativistic effective core potential[11] were used for Pb, while the 6-31G(d) basis set was used for other atoms.[13] These provide a total of 1,324 basis functions for Ar*PbPbAr*. Since all valence electrons were correlated in the MP2 calculations, the numbers of active occupied and virtual orbitals are 269 and 979, respectively. The time-dependent (TD)-B3PW91 method in the Gaussian 03 program[14] was employed to calculate excitation energies.

To calibrate the reliability of the MP2 method, the singly bonded structure of Ar*PbPbAr* found from the X-ray crystal study was first optimized. As Table 1 shows, the Pb-Pb distance and Pb-Pb-C trans-bent angle optimized by the MP2 method are 3.217 Å and 94.4º, respectively.

Table 1. Calculated and experimental geometrical parameters of the singly bonded structure of Ar*PbPbAr*.

	B3PW91[a]	MP2	Exptl.[b]
Pb-Pb (Å)	3.260	3.217	3.188
Pb-Pb-C (º)	100.4	94.4	94.3
C-Pb-Pb-C (º)	175.2	178.6	

[a] Reference 8g.

[b] Reference 7.

These values agree much better with the experimental values of 3.188 Å and 94.3º than the values of 3.260 Å and 100.4º optimized by the B3PW91 method. The core skeleton is nearly planar, as indicated by the C-Pb-Pb-C dihedral angle of 178.6º. Interestingly, a new structure was located as an energy minimum by switching the HOMO and LUMO levels of the singly bonded structure, as in the previous B3PW91 calculations.[8g] As Table 2 and Figure 2 show, the newly located structure is much less trans-bent (112.1º) and has the shorter Pb-Pb distance of 3.024 Å, while it is 118.5º twisted around the Pb-Pb bond because of the bulk of the Ar* group. The Pb-Pb distance is longer than those of typical Pb-Pb single bonds. As is apparent from Figure 3, however, the newly located structure corresponds to a multiply bonded structure, since the central Pb-Pb bond consists of a σ bond, a somewhat distorted π_{dis} bond (resulting from the twisting of the out-of-plane π orbital), and a slipped π_{slip} bond (resulting from the mixing of σ* and in-plane π orbitals due to trans-bending).

Table 2. Geometrical parameters calculated for the multiply bonded structure of Ar*PbPbAr*.

	B3PW91[a]	MP2
Pb-Pb (Å)	3.071	3.024
Pb-Pb-C (º)	117.7	112.1
C-Pb-Pb-C (º)	119.8	118.5

[a] Reference 8g.

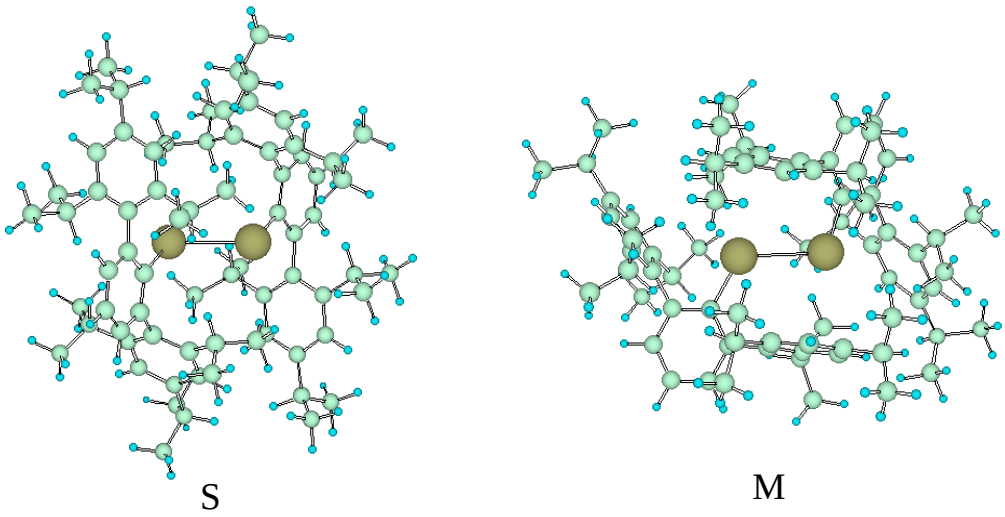
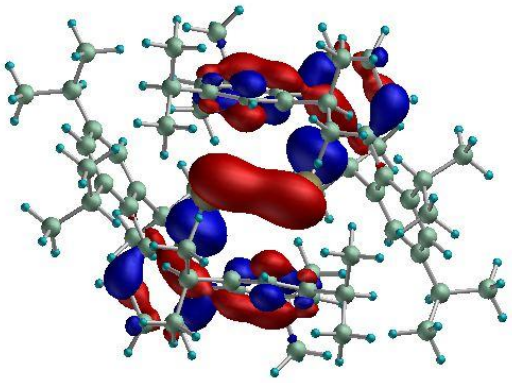
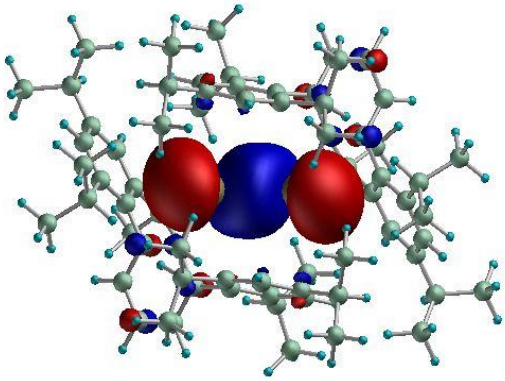


Figure 2. Singly (S) and multiply (M) bonded structures of Ar*PbPbAr* optimized at the MP2 level.

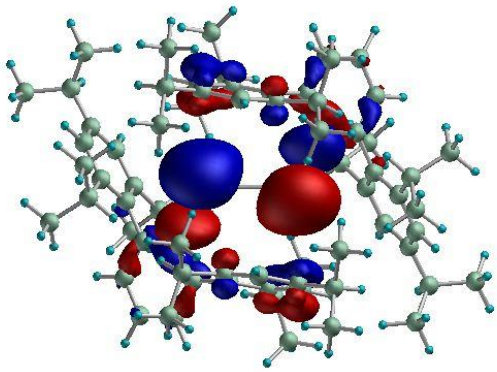
UV-vis spectra provide important information. For Ar*PbPbAr*, two absorptions have been observed at 397 nm (ε = 29000) and 719 nm (ε = 5200) in n-hexane solution,[7] as in the cases of Ar'GeGeAr' (371 and 501 nm)[3] and Ar'SnSnAr' (410 and 597 nm).[6] Table 3 compares the absorptions calculated using the TD-B3PW91 method and the observed values. For the multiply bonded structure of Ar*PbPbAr* optimized at the



σ



π_{dis}



π_{slip}

Figure 3. σ, π_{dis}, and π_{slip} orbitals of the multiply bonded structure of Ar*PbPbAr* at the HF level, plotted with a value of 0.03 au.

MP2 level, the two absorptions were calculated at 414 nm (*f* = 0.065) and 784 nm (*f* = 0.029), which are assignable to the ($\pi_{slip} \rightarrow \pi_{slip}^* + \pi_{dis} \rightarrow \sigma^* + \pi_{dis} \rightarrow Ar^*$) and ($\pi_{dis} \rightarrow \pi_{dis}^*$) transitions, respectively. These values are closer to the observed values of 397 and 719 nm (especially the $\pi_{dis} \rightarrow \pi_{dis}^*$ absorption) than the values of 413 and 822 nm calculated using the B3PW91 optimized structure. This may suggest that the MP2 structure is more reliable than the B3PW91 structure. For the MP2 and B3PW91

optimized singly bonded structures, only one strong absorption was calculated at 405 (*f* = 0.350) and 416 nm (*f* = 0.383), respectively, which is assignable to the $\pi_{dis} \rightarrow \sigma^*$ transition. Obviously, these results indicate that Ar*PbPbAr* has a multiply bonded structure in solution.

Table 3. Calculated and observed UV-vis absorptions (nm) of the singly bonded structure of Ar*PbPbAr*.[a]

Excitation	Geometries		Exptl.
	B3PW91[b]	MP2	
$\pi_{slip} \rightarrow \pi_{slip}^* + \pi_{dis} \rightarrow \sigma^* + \pi_{dis} \rightarrow Ar^*$	413 (*f*=0.141)	414 (*f*=0.065)	397 (ε=29000)
$\pi_{dis} \rightarrow \pi_{dis}^*$	822 (*f*=0.025)	784 (*f*=0.029)	719 (ε=5200)

[a] *f*=oscillator strength, ε in $Lmol^{-1}cm^{-1}$.

[b] Reference 8g.

In conclusion, MP2 provides more reliable structures for Ar*PbPbAr* than B3PW91. As verified from the calculations of UV-vis absorptions, Ar*PbPbAr* takes a multiple bonded structure in solution, although a singly bonded structure has been reported from the X-ray crystal study.

4.3 Binding energies of platinum complexes with π-conjugate systems

Platinum complexes of organic molecules have attracted considerable interest since the synthesis of $Pt(PPh_3)_2(C_{60})$ in 1991.[15-24] Recently, calculations have been performed for the platinum complexes of corannulene ($C_{20}H_{10}$), sumanene ($C_{21}H_{12}$), fullerene (C_{60}), and small π-conjugate molecules, $C_2H_{4-n}(CH=CH_2)_n$ (n = 0 - 4), using a variety of

theoretical methods. For $Pt(PH_3)_2(C_2H_{4-n}(CH=CH_2)_n)$ (n = 0 - 4), various DFT methods, such as SVWN, BLYP, B3LYP, B3PW91, MPW1PW91, and PBE1PBE, were employed to calculate the binding energies. However, the binding energies were highly underestimated by all the DFT methods, as compared with the MP2 and CCSD(T) results. In addition, the underestimation was greatly enhanced as "n" increases. All these results are based on the geometries optimized by the DFT (B3LYP and B3PW91) methods.

Therefore, geometry optimization was carried out without symmetry constraint for platinum complexes, $Pt(PH_3)_2(C_2H_{4-n}(CH=CH_2)_n)$ (n = 0 and 2) (Figure 4), at the B3LYP,[25] MP2, and SCS-MP2 levels.

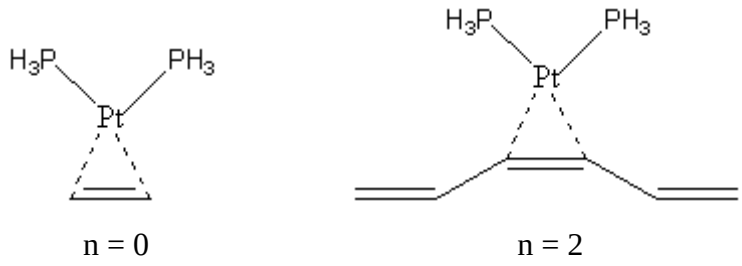


Figure 4. Structures of $Pt(PH_3)_2(C_2H_{4-n}(CH=CH_2)_n)$ (n = 0 and 2).

The binding energies were calculated with CCSD(T) as well as B3LYP, MP2, and SCS-MP2. The (2111/2111/111) basis set[26] augmented by a set of f polarization functions (f exponent 0.993)[27] and relativistic effective core potential[26] were used for Pt, and the 6-311G(d,p) basis set was used for other atoms.[28,29] B3LYP and CCSD(T) calculations were performed using Gaussian 03 program.[14] A dual-core, dual-processor Xeon 3.0GHz machine was used for all calculations.

In the SCS-MP2 method,[1] different scaling parameters are adapted for the same-spin (E_{SS}) and opposite-spin (E_{OS}) components of MP2 correlation energies,

$$E_{SCS-MP2} = p_{SS}E_{SS} + p_{OS}E_{OS}, \quad (1)$$

where E_{SS} and E_{OS} are given as

$$E_{SS} = \frac{1}{2}\sum_{ij}\sum_{ab}\frac{(ia \mid jb)\{(ia \mid jb)-(ib \mid ja)\}}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} + \frac{1}{2}\sum_{\bar{i}\bar{j}}\sum_{\bar{a}\bar{b}}\frac{(\bar{i}\bar{a} \mid \bar{j}\bar{b})\{(\bar{i}\bar{a} \mid \bar{j}\bar{b})-(\bar{i}\bar{b} \mid \bar{j}\bar{a})\}}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \quad (2)$$

and

$$E_{OS} = \sum_{i\bar{j}}\sum_{a\bar{b}}\frac{(ia \mid \bar{j}\bar{b})(ia \mid \bar{j}\bar{b})}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b}. \quad (3)$$

The scaling parameters used generally are p_{SS} = 1/3 and p_{OS} = 6/5, which are determined from the calculations of various molecules.

Table 4 shows the calculated bond distances optimized at the B3LYP, MP2, and SCS-MP2 levels for $Pt(PH_3)_2(C_2H_{4-n}(CH=CH_2)_n)$ (n = 0 and 2). The geometries optimized at the MP2 and SCS-MP2 levels are very similar. Furthermore, the Pt-P, Pt-C, and C=C distances optimized at the MP2 and SCS-MP2 levels for n = 0 are much closer to those observed for $Pt(PPh_3)_2(C_2H_4)$ than those optimized at the B3LYP level. The same is also expected for n = 2. As Table 5 shows, the binding energies are highly underestimated at the B3LYP level, as already pointed out in the recent study.[24] MP2 provides significantly larger binding energies than CCSD(T). However, it is notable that the SCS-MP2 binding energies are almost equal to the CCSD(T) binding energies. The binding energies for n = 2 are much smaller at the B3LYP level than those for n = 0, unlike the MP2, SCS-MP2, and CCSD(T) cases. This reflects the fact that the dispersion forces between the substituents on the Pt atom and the C=C bond are not well taken into account at the B3LYP level.

Table 4. Bond distances (Å) optimized at the B3LYP, MP2, and SCS-MP2 levels for platinum complexes.

	$Pt(PH_3)_2(C_2H_4)$			$Pt(PH_3)_2(C_2H_2(CH=CH_2)_2)$		
	Pt-P	Pt-C	C=C	Pt-P	Pt-C	C=C
B3LYP	2.321	2.150	1.425	2.323	2.173	1.441
MP2	2.252	2.088	1.434	2.251	2.103	1.445
SCS-MP2	2.275	2.094	1.438	2.274	2.107	1.449
Exptl.[a]	2.268	2.112	1.434			

[a] $Pt(PPh_3)_2(C_2H_4)$.[30]

Table 5. Binding energies (kcal/mol) of $Pt(PH_3)_2(C_2H_{4-n}(CH=CH_2)_n)$ (n=0 and 2) calculated with B3LYP, MP2, and SCS-MP2 optimized geometries.

$Pt(PH_3)_2(C_2H_{4-n}(CH=CH_2)_n)$	B3LYP	MP2	SCSMP2	CCSD(T)
MP2 optimized geometries				
n=0	12.1	38.1	29.5	27.3
n=2	4.7	40.2	30.0	27.3
SCS-MP2 optimized geometries				
n=0	12.7	38.0	29.6	27.6
n=2	5.7	40.0	30.2	27.8

Table 6 shows the elapsed times for the energy calculations of $Pt(PH_3)_2(C_2H_{4-n}(CH=CH_2)_n)$ (n = 0 and 2) at the MP2, SCS-MP2, and CCSD(T) levels. The SCS-MP2 calculations are as fast as the MP2 calculations, both being completed within 1 minute for n = 0 and 2. In contrast, the CCSD(T) calculations take 2 hours for n = 0 and 1 day for n = 2, and are 800 – 1800 times more time-consuming than the

SCS-MP2 calculations, despite no significant difference in accuracy between CCSD(T) and SCS-MP2.

Table 6. Elapsed times (second) for the energy calculations of $Pt(PH_3)_2(C_2H_{4-n}(CH=CH_2)_n)$ (n = 0 and 2) using a Xeon 3.0GHz machine (4 CPU-cores).

n	MP2	SCS-MP2	CCSD(T)
0	10	10	8,220
2	46	46	85,800

In conclusion, the SCS-MP2 method differs little in accuracy from the CCSD(T) method. Since SCS-MP2 is much faster than CCSD(T), it is suited to the calculations of large molecular systems.

4.4 Binding energies of benzene-benzene and naphthalene-naphthalene dimers

π-π interactions play an important role in many fields of supramolecular chemistry and biochemistry (i.e., DNA stacking). The widely used DFT fails to describe correctly π-π interactions, while the MP2 method is the simplest approach to treat them. However, the MP2 method tends to overestimate π-π interactions. The binding energies of benzene-benzene and naphthalene-naphthalene dimers with parallel-displaced structures (Figure 5) were calculated using the MP2, SCS-MP2, and CCSD(T) methods to check the performance of SCS-MP2 for π-π interactions. Geometry optimization was carried out at the MP2 and SCS-MP2 levels. The 6-31G(d),[13] 6-311G(d),[31] and aug-cc-pVDZ[32] basis sets were used for all calculations.

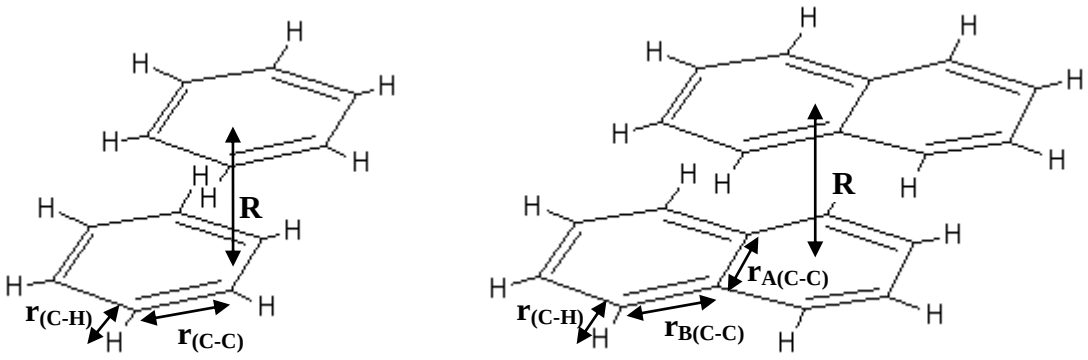


Figure 5. Structures of benzene-benzene and naphthalene-naphthalene dimers

As Table 7 shows, the largest difference in C-C and C-H bond distances between MP2 and SCS-MP2 is only 0.005 Å. However, the intermolecular distances (R) optimized at the SCS-MP2 level are 0.12 - 0.15 Å longer than those at MP2, indicating that the SCS-MP2 method corrects the overestimation of π-π interactions at the MP2 level.

As is obvious from Table 8, binding energies calculated at the MP2 level for the benzene-benzene dimer are considerably larger than those at the CCSD(T) level. In contrast, the binding energies at the SCS-MP2 level are close to those at the CCSD(T) level for all the basis sets employed. The same is also true for the binding energies for the naphthalene-naphthalene dimer, as shown in Table 9. In contrast, only repulsive interactions were calculated for the benzene-benzene and naphthalene-naphthalene dimers by the standard DFT methods such as B3LYP and B3PW91.

Table 7. Bond distances (Å) optimized for benzene-benzene and naphthalene-naphthalene dimers.

	6-31G(d)		6-311G(d)		aug-cc-pVDZ	
	MP2	SCS-MP2	MP2	SCS-MP2	MP2	SCS-MP2
Benzene-benzene dimer						
$r_{(C-C)}$	1.397	1.399	1.399	1.400	1.408	1.409
$r_{(C-H)}$	1.088	1.088	1.087	1.088	1.094	1.095
R	3.390	3.538	3.252	3.388	3.203	3.323
Naphthalene-naphthalene dimer						
$r_{A(C-C)}$	1.434	1.430	1.437	1.432		
$r_{B(C-C)}$	1.419	1.424	1.421	1.426		
$r_{(C-H)}$	1.091	1.091	1.090	1.091		
R	3.265	3.408	3.164	3.297		

Table 8. Binding energies (kcal/mol) calculated for the benzene-benzene dimer.

	Energies								
	6-31G(d)			6-311G(d)			aug-cc-pVDZ		
Geometries	MP2	SCS-MP2	CCSD(T)	MP2	SCS-MP2	CCSD(T)	MP2	SCS-MP2	CCSD(T)
MP2/6-31G(d)	3.17	1.88	1.57	4.33	2.70	2.50	8.01	6.08	6.03
SCS-MP2/6-31G(d)	3.04	2.01	1.73	3.92	2.59	2.37	7.33	5.73	5.70
MP2/6-311G(d)	3.02	1.43	1.07	4.50	2.52	2.31	8.37	6.06	5.97
SCS-MP2/6-311G(d)	3.16	1.94	1.65	4.28	2.74	2.55	7.81	5.97	5.95
MP2/aug-cc-pVDZ	2.78	0.99	0.57	4.44	2.22	1.95	8.45	5.89	5.71
SCS-MP2/aug-cc-pVDZ	3.16	1.71	0.99	4.48	2.66	2.40	8.25	6.09	7.58

Table 9. Binding energies (kcal/mol) calculated for the naphthalene-naphthalene dimer.

	Energies						
	6-31G(d)			6-311G(d)		aug-cc-pVDZ	
Geometries	MP2	SCS-MP2	CCSD(T)	MP2	SCS-MP2	MP2	SCS-MP2
MP2/6-31G(d)	8.56	5.05	3.61	11.80	7.52	18.14	13.37
SCS-MP2/6-31G(d)	7.98	5.39	4.29	11.95	7.05	19.11	13.62
MP2/6-311G(d)	8.38	4.30	2.67	10.38	7.18	16.77	13.05
SCS-MP2/6-311G(d)	8.55	5.30	3.95	11.46	7.51	18.29	13.77

In summary, the SCS-MP2 method reproduces well the binding energies at the CCSD(T) level. The SCS-MP2 method is suitable for the calculations of π-π interactions in large molecules, because of the much lower computational cost.

4.5 References

1. Grimme, S. J Chem Phys 2003, 118, 9095.

2. (a) Power, P. P. Chem Commun 2003, 2091. (b) Weidenbruch, M. Angew Chem Int Ed 2003, 42, 2222. (c) Weidenbruch, M. Angew Chem Int Ed 2005, 44, 514.

3. Sekiguchi, A.; Kinjo, R.; Ichinohe, M. Science 2004, 305, 1755.

4. Stender, M.; Phillips, A. D.; Wright, R. J.; Power, P. P. Angew Chem Int Ed 2002, 41, 1785.

5. Sugiyama, Y.; Sasamori, T.; Hosoi, Y.; Furukawa, Y.; Takagi, N.; Nagase, S.; Tokitoh, N. J Am Chem Soc 2006, 128, 1023.

6. Phillips, A. D.; Wright, R. J.; Olmstead, M. M.; Power, P. P. J Am Chem Soc 2002, 124, 5930.

7. Pu, L.; Twamley, B.; Power, P. P. J Am Chem Soc 2000, 122, 3524.

8. (a) Kobayashi, K.; Nagase, S. Organometallics 1997, 16, 2489. (b) Nagase, S.; Kobayashi, K.; Takagi, N. J Organomet Chem 2000, 611, 264. (c) Kobayashi, K.; Takagi, N.; Nagase, S. Organometallics 2001, 20, 234. (d) Takagi, N.; Nagase, S. Organometallics 2001, 20, 5498. (e) Takagi, N.; Nagase, S. Chem Lett 2001, 966. (f) Takagi, N.; Nagase, S. Eur J Inorg Chem 2002, 2775. (g) Takagi, N.; Nagase, S. J. Organomet Chem 2007, 692, 217.

9. (a) Chen, Y.; Hartmann, M.; Diedenhofen, M.; Frenking, G. Angew Chem Int Ed 2001, 40, 2051. (b) Lein, M.; Krapp, A.; Frenking, G. J Am Chem Soc 2005, 127, 6290.

10. (a) Becke, A. D. Phys Rev 1988, A38, 3098. (b) Becke, A. D. J Chem Phys 1993, 98, 5648. (c) Perdew, J. P.; Wang, Y. Phys Rev 1992, B45, 13244.

11. Wadt, W. R.; Hay, P. J. J Chem Phys 1985, 82, 284.

12. Huzinaga, S.; Andzelm, J.; Klobukowski, M.; Radzio-andzerm, E.; Sakai, Y.; Tatewaki, H. Gaussian Basis Sets for Molecular Calculations; Elsevier: Amsterdam, 1984.

13. Francl, M. N.; Pietro, W. J.; Hehre, W. J.; Binkley, J. S.; Gordon, M. S.; Defrees, D. J.; Pople, J. A. J Chem Phys 1982, 77, 3654.

14. Gaussian 03, Revision C.01; Gaussian, Inc., Wallingford, CT, 2004.

15. Fagan, P. J.; Calabrese, J. C.; Malone, B. J Am Chem Soc 1991, 113, 9408.

16. Lichtenberger, D. L.; Wright, L. L.; Gruhn, N. E.; Rempe, M. E. Synth Metals 1993, 59, 353.

17. Lichtenberger, D. L.; Wright, L. L.; Gruhn, N. E.; Rempe, M. E. J Organomet Chem 1994, 478, 213.

18. Koga, N.; Morokuma, K. Chem Phys Lett 1993, 202, 330.

19. Fujimoto, H.; Nakao, Y.; Fukui, K. J Mol Struct (THEOCHEM) 1993, 300, 425.

20. Bo, C.; Costas, M.; Poblet, J. M. J Phys Chem 1995, 99, 5914.

21. Lopez, J. A.; Mealli, C. J Organomet Chem 1994, 478, 161.

22. Nunzi, F.; Sgamellotti, A.; Re, N.; Floriani, C. Organometallics 2000, 19, 1628.

23. Kaneno, Y.; Ikeda, A.; Nakao, Y.; Sato, H.; Sakaki, S. J Phys Chem A 2005, 109, 8055.

24. Ikeda, A.; Nakao, Y.; Sato, H.; Sakaki, S. J Phys Chem A in press.

25. (a) Becke, A. D. J Chem Phys 1993, 98, 5648. (b) Lee, C.; Yang, W.; Parr, R. G. Phys Rev B 1988, 37, 785.

26. Hay, P. J.; Wadt, W. R. J Chem Phys 1985, 82, 299.

27. Ehlers, A. W.; Böhme, M.; Dapprich, S.; Gobbi, A.; Höllwarth, A.; Jonas, V.; Köhler, K. F.; Stegmann, R.; Veldkamp, A.; Frenking, G. Chem Phys Lett 1993, 208, 111.

28. Krishnan, R.; Binkley, J. S.; Seeger, R.; Pople, J. A. J Chem Phys 1980, 72, 650.

29. McLean, D.; Chandler, G. S. J Chem Phys 1980, 72, 5639.

30. Cheng, P. –T.; Cook, C. D.; Nyburg, S. C.; Wan, K. Y. Inorg Chem 1971, 10, 2210.

31. Krishnan, R.; Binkley, J. S.; Seeger, R.; Pople, J. A. J Chem Phys 1980, 72, 650.

32. Dunning, Jr. T. H. J Chem Phys 1989, 90, 1007.

Chapter V

SAC/SAC-CI calculations of ionized and excited states

(Ishimura, K.; Hada, M.; Nakatsuji, H. J Chem Phys, 2002, 117, 6533.)

5.1 Abstract

The ground state, singlet, and triplet excited states, and ionized states of ferrocene $Fe(C_5H_5)_2$ were studied by the symmetry adapted cluster (SAC)/SAC-configuration-interaction method. The calculated ionization energies and intensities fairly well reproduced the observed photoelectron spectrum in the wide region of 6-14 eV. In particular, the first two peaks ($^2E_2$' and $^2A_1$') were assigned to the ionizations from the occupied 3*d* orbitals of Fe, mixed already with the two-electron shake-up processes. This is the first *ab initio* quantitative assignment that is consistent with the experimental data. For the singlet states, three *d-d* transitions were calculated at 2.12, 2.26, and 4.02 eV, which correspond to the experimental peaks observed at 2.69, 2.97, and 3.82 eV. We propose possible assignments for other absorption bands in the range of 2.12-6.57 eV. In another three triplet *d–d* transition states we calculated, we found that the energy order of these states (1 $^3E_1$", 1 $^3E_2$", 2 $^3E_1$") differs from that of singlet states (1 $^1E_2$", 1 $^1E_1$", 2 $^1E_1$").

5.2 Introduction

Since the discovery of ferrocene, $Fe(C_5H_5)_2$,[1,2] a number of studies have clarified its structure, reactions, and properties.[3,4] Ultraviolet and visible absorption spectra, in particular, have been studied extensively to clarify the electronic structure of ferrocene in its ground and excited states.[5-11] Photoelectron spectra of various ferrocene-type compounds have also been observed and are well documented.[7–9] Armstrong *et al.*[10] and Sohn *et al.*[11] reported that ferrocene exhibits at least 11 absorption bands in the range of 18 000-53 000 cm^{-1} (2.2-6.5 eV). Using the ligand-field theory, three *d-d* transitions were expected for d^6-ferrocene, though there is considerable ambiguity in the location of these three transitions. Further, the positions of the spin-forbidden *d-d* transitions are unclear because of their small intensities. In the ionization spectrum, the first and second peaks were inferred from experimental data to be the ionizations from the 3*d* orbitals.[8,9]

A number of theoretical studies reported the bonding character of the ferrocene ground state.[4] The bonding nature between Fe and two cyclopentadienyl (Cp) rings is reasonably understood as the dative bond between Fe^{2+} and Cp^-.[3,12] Recently, the mechanism of the protonation of ferrocene in the ground state has been studied using the CCSD(T) method.[13]

For ferrocene ionization energies, ΔSCF calculations,[14,15] *Xα* scattered-wave calculations,[16,17] and intermediate neglect of differential overlap (INDO) Green's function calculation[18] have been done. Ohno *et al.*[19] calculated the ionization energies with the *ab initio* third-order algebraic diagrammatic construction [ADC(3)] Green's function method and proposed the lowest four peaks to be E_2' (metal), E_1' (ligand), E_1'' (ligand), and A_1' (metal). All calculations have contradicted with the experimental assignment, the lowest E_2' (metal) and A_1' (metal) states, either qualitatively or

quantitatively.

For ultraviolet-visible absorption spectra of ferrocene, on the other hand, Rohmer *et al.*[20] proposed a theoretical assignment based on *ab initio* singly excited configuration interaction (SECI) calculations. Zerner *et al.*[21] also proposed a different assignment based on semiempirical INDO-SECI calculations. The calculated results did not satisfactorily agree with the experimental transition energies due to the small basis sets and/or the lack of electron correlations. More reliable *ab initio* studies are necessary to truly understand the electronic structures of the ground, excited, and ionized states of ferrocene.

This paper addresses this subject using the symmetry adapted cluster (SAC)[22] and SAC-configuration-interaction (SAC-CI)[23] methods, which have been applied successfully to the spectroscopies of various molecules[24] including transition metal complexes.[25-27] We assign the photoelectron spectra and the ultraviolet-visible absorption spectra of ferrocene, $Fe(C_5H_5)_2$ by calculating the ground state, singlet, and triplet excited states, and ionized doublet states of this molecule.

5.3 Method

We used the Gaussian basis functions of triple-zeta quality for Fe and double-zeta quality for C and H; Huzinaga's (533111/52111/311) basis set is used for Fe,[28] Huzinaga's (5121/41) set for C,[28] and Huzinaga-Dunning's (31) set for H.[29] The basis set for Fe was augmented with Hay's flexible *d* function (ζ=0.1133)[30] and the augmented double-zeta ANO *f* functions of Roos *et al.*[31] The basis set for C was augmented with a polarization *d* function (ζ=0.6)[28] and that of H with a polarization *p* function (ζ=1.0).[29] On the center of each Cp ring, the *s*, *p*, and *d* Rydberg functions were

added in double-zeta quality.[32] The total number of basis functions is 315.

Experimental geometrical parameters of $Fe(C_5H_5)_2$ for the ground state are used throughout the present calculations: 2.064 Å for Fe-C, 1.440 Å for C-C in the Cp ring, and 1.104 Å for C-H.[33] The Cp ring is assumed to be planar and the two Cp rings are in an eclipsed conformation (D_{5h}), as observed in a gas phase.[34] The *z*-axis is defined to be parallel to the Cp-Fe-Cp axis.

The Hartree-Fock (HF) calculations were carried out using the GAUSSIAN 98 program.[35] The electron correlation in the ground state was taken into account by the SAC theory, and those in the singlet and triplet excited states and ionized doublet states by the SAC–CI theory. The active space in SAC and SAC-CI calculations involves 285 HF orbitals; the 1*s*, 2*s*, and 2*p* orbitals of Fe and 1*s* orbital of C were treated as frozen-core orbitals. To reduce the configuration number, perturbation selection technique[36] is used with energy thresholds of 1.0×10^{-5} for the ground state, 1.0×10^{-6} for the doublet ionized state, and 2.0×10^{-6} for singlet and triplet excited states.

5.4 Ground State

The bonding nature of the ground state of ferrocene has been well documented as described above. Table 1 summarizes the energies and their characters of some important occupied and unoccupied orbital obtained by the present Hartree–Fock calculation. The highest occupied molecular orbital (homo) $4e_1$" and the next homo $6e_1$' are essentially the π orbitals of the Cp rings (Cp-π) and have small components of the $3d_{xz}$ and $3d_{yz}$ or $4p_x$ and $4p_y$ orbitals of Fe. We may consider that ferrocene has dative bonds between Fe^{2+} and Cp^-, electron donation being from Cp^- to Fe^{2+} through these orbitals. Occupied 3*d* orbitals of Fe ($4e_2$' and $8a_1$') are lower in energy than the Cp-π

Table 1. Orbital energies and orbital characters of some important occupied and unoccupied MOs.

occupied				unoccupied			
No.	Symmetry	Orbital energy(eV)	Character	No.	Symmetry	Orbital energy(eV)	Character
47,48	$4e_1$"	-9.11	Cp(π)+Fe(d_{xz},d_{yz})	49	$9a_1$'	0.29	Fe(s)+Cp(Ryd(s,d_{z2}))
45,46	$6e_1$'	-9.25	Cp(π)+Fe(p_x,p_y)	50	$7a_2$"	0.44	Fe(p_z)+Cp(Ryd(s,p_z,d_{z2}))
43,44	$4e_2$'	-11.44	Fe(d_{xy},d_{x2-y2})	51,52	$5e_1$"	0.79	Fe(d_{xz},d_{yz})+Cp(Ryd(p_x,p_y))
42	$6a_2$"	-13.31	Cp(π)+Fe(p_z)	53	$10a_1$'	0.80	Fe(s)+Cp(Ryd(s,d_{z2}))
41	$8a_1$'	-13.82	Fe(d_{z2},s)	56,57	$7e_1$'	0.94	Fe(p_x,p_y)+Cp(Ryd(p_x,p_y))
39,40	$3e_2$"	-14.04	Cp(σ)	63	$11a_1$'	1.31	Fe(s)+Cp(Ryd(s,p_z,d_{z2}))
37,38	$3e_2$'	-14.30	Cp(σ)	64	$9a_2$"	1.49	Fe(p_z)+Cp(Ryd(s,p_z,d_{z2}))
35,36	$5e_1$'	-14.95	Cp(σ)	65	$12a_1$'	2.03	Fe(s)+Cp(Ryd(s,p_z,d_{z2}))
33,34	$3e_1$"	-15.10	Cp(σ)	66,67	$6e_1$"	2.14	Fe(d_{xz},d_{yz})+Cp(Ryd(p_x,p_y))
32	$7a_1$'	-15.24	Cp(σ)	78,79	$7e_1$"	3.90	Fe(d_{xz},d_{yz})+Cp(Ryd(p_x,p_y))
				83,84	$7e_2$'	5.28	Cp(π*)+Fe(d_{xy},d_{x2-y2})
				86,87	$8e_1$"	6.18	Cp(π+Ryd(p_x,p_y))+Fe(d_{xz},d_{yz})
				92,93	$9e_1$"	7.43	Fe(d_{xz},d_{yz})+Cp(π+Ryd(p_x,p_y))

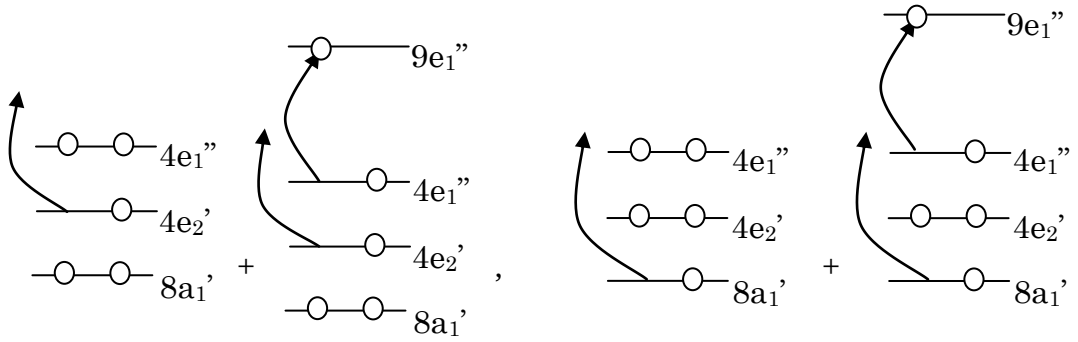
orbitals ($4e_1$" and $6e_1$'). The $4e_2$' orbitals are essentially the $3d_{xy}$ and $3d_{x2-y2}$ atomic orbitals of Fe with a small Cp-π character, while the $8a_1$' orbital has $3d_{z2}$ and 4*s* character of Fe. The unoccupied $3d_{xz}$ and $3d_{yz}$ orbitals appear as $9e"_1$. Therefore, in a simple ligand-field picture, three singly excited *d-d* transitions are expected in this electronic configuration. The total energy at the HF level was -1646.4843 hartree and

the ground state correlation energy obtained by the SAC method was -0.8361 hartree.

5.5 Ionized doublet states

Figure 1 shows the calculated SAC-CI ionization energies compared to the observed spectrum,[9] together with the HF orbital energies shown as Koopmans ionization energies. The observed spectrum in Figure 1 has three bands below 16 eV. First and second bands have two peaks, and the third one has several peaks. The present SAC-CI result satisfactorily reproduces the observed first two bands including four ionized states. The SAC-CI method predicts the third band to have six electronic states. The ionization energies calculated by the SAC-CI method agree well with the observed ones.

Table 2 lists the ionization energies and the intensities (in parentheses) calculated by the SAC-CI method and the HF method (Koopmans). The ionization energies calculated by the *ab initio* ADC(3) Green's function method[19] and those observed[8,9] are also shown for comparison. The first peak observed at 6.86 eV was calculated at 6.26 eV (1 $^2E'_2$) as the ionization mainly from $4e_2$' MO. The second peak observed at 7.23 eV was calculated at 7.27 eV (1 $^2A_1$') as the ionization from $8a_1$' MO. These states have significant components of two-electron shake-up processes, which are from $4e_1$", $4e_2$' to $9e_1$", and from $4e_1$", $8a_1$' to $9e_1$", respectively, as shown below:



1^2E_2' 1^2A_1'

The extents of mixing of these shake-up configurations are very large as seen from the main configuration shown in Table 2. It is very remarkable that even the first ionization peak already has a strong mixing of the shake-up ionizations. It is for this reason that the calculated ionization energy differs by more than 0.5 eV from the experimental value.

The next band at around 9 eV is assigned to be composed of the two peaks of the ionizations from the π orbitals of the Cp rings and the *d* or *p* orbital of Fe. The third peak observed at 8.72 or 8.87 eV was calculated at 8.78 eV (1 $^2E_1$'), which is the ionization from the 6e'$_1$ MO. The fourth peak observed at 9.14-9.39 eV was calculated at 9.05 eV (1 $^2E_1$"), which is the ionization from 4e_1". These two peaks do not have the shake-up mixing. The present assignment of the lowest four peaks agrees with the experimental prediction, though these peaks have not been reproduced quantitatively or even qualitatively in the previous theoretical calculations.[14-19] The assignment of the ADC(3) Green's function method differs from ours because the first and second peaks by ADC(3) were not caused by the ionizations from the Fe *d* orbitals. The Koopmans picture completely breaks down as seen from Figure 1; it fails to describe the ionizations from the *d* orbitals of Fe, due to the strong orbital relaxation and/or large electron correlation effects. This is seen from the main configurations of the 1 $^2E_2$' and 1 $^2A_1$' ionized doublet states shown in Table 3.

For the next broad band in 11-14 eV, six ionized states were calculated by the SAC-CI method. The fifth peak calculated at 11.86 eV corresponds to the peak observed at 12.2-12.3 eV and originates from the π orbitals of the Cp rings mixed with the *p* orbital of Fe. The next higher two pairs of peaks calculated at 12.63, 12.65, and 13.46, 13.69 eV originate from the σ orbitals of the Cp rings, and the peak at 13.34 eV, which lies median between these two pairs of peaks, is the ionization from the π orbital

of the Cp rings. Note that the ordering of these six ionized states is quite different from that of the Koopmans picture, as seen from Figure 1 and from the comparison between Tables 1 and 2. Electron correlations are certainly very important for describing the electronic structures of the ionized states of ferrocene.

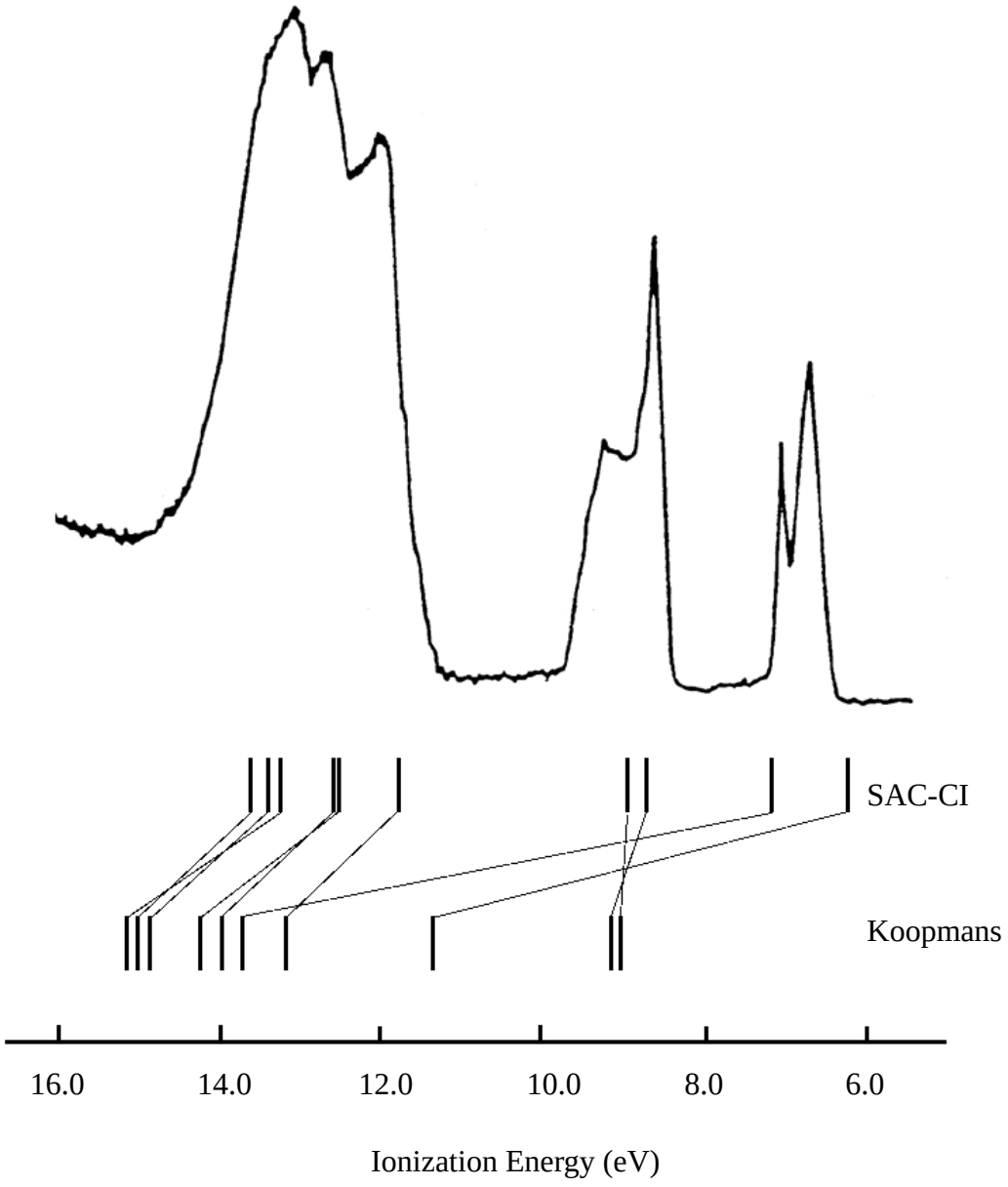


Figure 1. Ionization energies SAC-CI and Koopmans ionization energies of Ferrocene.

Table 2. Ionized doublet states and ionization energies (eV) of ferrocene obtained by SAC-CI, ADC(3), Hartree-Fock, and experiment.

State	Main configurations	SAC-CI	ADC(3)[a]	Koopmans.[b]	Exp.
$1^2E_2'$	$0.854(4e_2')+0.39(4e_1'',4e_2'\rightarrow 9e_1'')$	6.26(1.494)	7.85	11.44	6.86[c],6.88[d]
$1^2A_1'$	$0.82(8a_1')+0.41(4e_1'',8a_1'\rightarrow 9e_1'')$	7.27(0.747)	9.36	13.82	7.23 [c,d]
$1^2E_1'$	$6e_1'$	8.78(1.917)	8.70	9.25	8.72 [c,d],8.87[d]
$1^2E_1''$	$4e_1''$	9.05(1.919)	9.08	9.11	9.14[d],9.38 [c],9.39[d]
$1^2A_2''$	$6a_2''$	11.86(0.919)	12.13	13.31	12.2[c],12.3 [d]
$2^2E_2'$	$3e_2'$	12.63(1.910)	13.08	14.30	13.0 [d],13.6[c]
$1^2E_2''$	$3e_2''$	12.65(1.887)	12.96	14.04	13.0 [d],13.6[c]
$2^2A_1'$	$7a_1'$	13.34(0.920)	13.71	15.24	13.46[d],13.6 [c]
$2^2E_1'$	$5e_1'$	13.46(1.956)	13.89	14.95	13.46[d],13.6 [c]
$2^2E_1''$	$3e_1''$	13.69(1.916)	14.07	15.10	13.46[d],13.6 [c]

[a]Reference 18.

[b]Our work

[c]Reference 8.

[d]Refercence 9.

5.6 Singlet Excited states

Table 3 shows the singlet transition energies and the corresponding main configurations calculated by the SAC-CI method. The results calculated by the *ab initio* SECI method reported by Rohmer *et al.*,[20] and by the INDO-SECI method reported by Zerner *et al.*,[21] and the experimental transition energies[10,11] are also shown in the same

table. As for the experimental values, those observed in vapor phase[10,11] and in a solution[10] are shown, since they are complementary with each other.

As expected from the ground-state electronic configuration, three *d-d* transitions may appear in the low-lying singlet excited states of ferrocene. The present work certainly gave the lowest three states, 1 $^1E_2$" (2.11 eV), 1 $^1E_1$" (2.27 eV), and 2 $^1E_1$" (4.03 eV), to be essentially the *d-d* transitions in character, as expected. However, this energy ordering (1 $^1E_2$" , 1 $^1E_1$" , 2 $^1E_1$") is different from that given by the INDO-SECI method (1 $^1E_1$" , 1 $^1E_2$" , 2 $^1E_1$").[21] These transitions are dipole-forbidden, so that the observed peaks are very weak. In fact, the lowest two singlet transitions were observed at 2.70 and 2.98 eV in solution, but observed as one peak at 2.81 eV in a vapor phase. The next three states calculated by the SAC-CI method are 1 $^1A_2$' (5.25 eV), 1 $^1A_1$' (5.29 eV), and 1 $^1E_2$' (5.48 eV), which are the transitions from Cp-π to 3d_{xz} , 3d_{xz} of Fe. The experimental values are 5.02 and 5.23 eV. The next state is 2 $^1A_2$' calculated at 5.60 eV in comparison with the experimental value of 5.82 eV, and this is the π-π* transition of the Cp ring. These transitions are dipole-forbidden and therefore, their experimental peaks have only small intensities, but they are reasonably well reproduced by the present SAC-CI calculations.

The first dipole-allowed transition is the 1 $^1E_1$' state that is calculated at 6.34 eV by the SAC-CI method; the calculated energy well reproduces the observed ones, 6.31 eV in vapor and 6.20 eV in solution. The second dipole-allowed state is 2 $^1E_1$" state calculated at 6.43 eV, which is assigned to the experimental peak observed at 6.58 eV. Thus, the dipoleallowed transitions in this energy region are also well reproduced by the present SAC-CI calculations.

Table 3. Singlet excited states and their excitation energies of ferrocene (eV) obtained by SAC-CI, SECI, INDO-SECI, and experiment.

State	Main configurations	SAC-CI (eV)	SECI[a]	INDO-SECI[b]	Exp.(eV)
1^1E_2"	$4e_2$'→$9e_1$"	2.11	2.63	2.96	2.81[c],2.70[d]
1^1E_1"	0.58($4e_2$'→$9e_1$")-0.54($8a_1$'→$9e_1$")	2.27	3.31	2.69	2,81 [c],2.98[d]
2^1E_1"	0.57($8a_1$'→$9e_1$")+0.54($4e_2$'→$9e_1$")	4.03	5.74	3.96	3.82[c,d]
			9.05	4.58,4.92,4.95	4.67[c,d]
1^1A_2'	$4e_1$"→$9e_1$"	5.25		5.11	5.02[c]
1^1A_1'	$4e_1$"→$9e_1$"	5.29		5.26.5.32.5.54	5.23[c]
1^1E_2'	$4e_1$"→$9e_1$"	5.48			
2^1A_2'	$4e_1$"→$7e_2$'	5.60	10.17	5.62,5.69,5.91	5.82[c]
				6.22	6.16[c]
1^1E_1'	0.69($6e_1$'→$9a_1$')+0.53($6e_1$'→$11a_1$')	6.34	10.39	6.24	6.20[d],6.31[c]
2^1E_1'	$6e_1$'→$9a_1$'	6.43		6.48	6.58[c]
1^1A_1"	$6e_1$'→$9e_1$"	6.45		7.04	
1^1A_2"	$6e_1$'→$9e_1$"	6.47			
2^1A_2"	$6e_1$'→$9e_1$"	6.56			

[a]Reference 19.

[b]Reference 20.

[c]Reference 10.

[d]Reference 11.

5.7 Triplet Excited States

Table 4 displays three triplet *d-d* transition energies calculated by the SAC-CI, SECI,[20] and INDO-SECI[21] methods. The SAC-CI main configurations and their coefficients are also listed along with the observed excitation energies,[10] which were assigned experimentally to be spin-forbidden transitions. The present SAC-CI calculations indicate that the lowest three triplet excitations are *d-d* transitions in nature, as expected similarly from the case of the singlet excitations. The 1 $^3E_1$" is located lower by 0.87 eV than the corresponding singlet 1 $^1E_1$" state, while 1 $^3E_2$" and 2 $^3E_1$" states are lower by 0.43 and 1.43 eV than the corresponding 1 $^1E_2$" and 2 $^1E_1$" states. The large splitting of the singlet and triplet $^2E_1$" states implies a considerable mixing of the Cp-π states to the *d* states in the triplet state 2 $^3E_1$". Thus, the $^3E_1$" state has the lowest energy of these triplet states and the energy ordering of the triplet states (1 $^3E_1$", 1 $^3E_2$", 2 $^3E_1$") is different from that of the singlet states (1 $^1E_2$", 1 $^1E_1$", 2 $^1E_1$"). The total energy order, including both singlet and triplet states, is 1 $^3E_1$", 1 $^3E_2$", 1 $^1E_2$", 1 $^1E_1$", 2 $^3E_1$", and 2 $^1E_1$". We assign the calculated 2 $^3E_1$" state (2.60 eV) to the observed peak of 2.29 eV considering its weak intensity and the spin-forbidden nature experimentally suggested, though the calculated 1 $^1E_2$" state (2.11 eV in Table 4) is closer to the 2.29 eV peak than the 2 $^3E_1$" state.

Table 4. Triplet d-d excitation energies of ferrocene (eV) obtained by SAC-CI, SECI, and INDO-SECI, and experiment.

State	Main configurations	SAC-CI (eV)	SECI[a]	INDO-SECI[b]	Exp.(eV)[c]
1^3E_1"	0.72($4e_2$'→$9e_1$")-0.33($8a_1$'→$9e_1$")	1.40	1.81-1.87	2.54	1.74
1^3E_2"	$4e_2$'→$9e_1$"	1.68	1.81-1.87	2.55	2.05
2^3E_1"	0.70($8a_1$'→$9e_1$")+0.32($8a_1$'→$8e_1$")	2.60	4.56	2.55	2.29

[a]Reference 19.

[b]Reference 20.

[c]Reference 10.

5.8 Conclusions

This paper summarizes the SAC/SAC-CI study on the ground state, singlet and triplet excited states, and doublet ionized states of ferrocene, $Fe(C_5H_5)_2$. The main results of this paper are as follows:

(1) The ionization spectrum in 6-12 eV energy region is reproduced reasonably by the present calculations. The first and second peaks are assigned to the ionization from occupied 3*d* orbitals of Fe. This assignment is consistent with the experimental one. A remarkable point is the strong mixing of the shake-up processes *even to these lowest ionization states*, showing a break-down of the Koopmans picture.

(2) The second and third broad bands in the ionization spectrum experimentally observed are composed of the two and six ionized states and were reproduced satisfactorily by the SAC–CI method. The Koopmans ordering was wrong for these states.

(3) The observed peaks at 2.70, 2.98, and 3.82 eV were assigned to the singlet *d-d* transitions calculated at 2.11, 2.27, and 4.03 eV. Other dipole-forbidden states were also assigned satisfactorily to the weak peaks experimentally observed.

(4) The absorption observed at 6.31 eV with strong intensity is assigned to the first dipole-allowed $^1E_1$' state, whose calculated energy is 6.34 eV. This agreement is much improved over those in the previous works. The second strong absorption peak is also well reproduced and assigned to the second $^1E_1$' state.

(5) Three triplet *d-d* transition energies are calculated; they well reproduced the three weak peaks observed in 1.74 - 2.29 eV.

5.9 References

1. Kealy, T. J.; Paulson, P. L. Nature (London) 1951, 168, 1039.

2. Miller, S. A.; Tebboth, J. A.; Tremaine, J. F. J Chem Soc 1952, 632.

3. Veilard, A.; Demuynck, J. in *Modern Theoretical Chemistry*, edited by Schaefer III, H. F. (Plenum, New York, 1997), Vol. 4.

4. Haaland, A. Acc Chem Res 1979, 11, 415, and references therein.

5. Scott, D. R.; Becker, R. S. J Chem Phys 1961, 35, 516.

6. Smith, J. J.; Meyer, B. J Chem Phys 1968, 48, 5436.

7. Green, J. C.; Struct Bonding (Berlin) 1981, 43, 37.

8. Rabalais, J. W.; Werme, L. O.; Bergmark, T.; Karlsson, L.; Hussain, M.; Siegbahn, K. J Chem Phys 1972, 57, 1185.

9. Evans, S.; Green, M. L. H.; Jewitt, B.; Orchard, A. F.; Pygall, C. F. J Chem Soc Faraday Trans 2 1972, 68, 1847.

10. Armstrong, A. T.; Smith, F.; Elder, E.; McGlynn, S. P. J Chem Phys 1967, 46, 4321.

11. Sohn, Y. S.; Hendrickson, D. N.; Gray, H. B. J Am Chem Soc 1971, 93, 3603.

12. Lauher, J. W.; Hoffmann, R. J Am Chem Soc 1976, 98, 1729.

13. Mayor-Lopez, M. J.; Lüthi, H. P.; Koch, H.; Morgantini, P. Y.; Weber, J. J Chem Phys 2000, 113, 8009.

14. Coutiere, M.-M.; Demuynck, J.; Veillard, A. Theor Chim Acta 1972, 27, 281.

15. Bagus, P. S.; Walgren, U. I.; Almlöf, J. J Chem Phys 1976, 64, 2324.

16. Baerends, E. J.; Ross, P. Chem Phys Lett 1973, 23, 391.

17. Röcsh, N.; Johnson, K. H. Chem Phys Lett 1974, 24, 179.

18. Böhm, M. C.; Z Naturforsch 1982, 37A, 1193.

19. Ohno, M.; Niessen, W. von; Schüle, J. Chem Phys 1991, 158, 1.

20. Rohmer, M.-M.; Veillard, A.; Wood, M. H. Chem Phys Lett 1974, 29, 466.

21. Zerner, M. C.; Loew, G. H.; Kirchner, R. F.; Mueller-Westerhoff, U. T. J Am Chem Soc 1980, 102, 589.

22. Nakatsuji, H.; Hirao, K. J Chem Phys 1978, 68, 2053.

23. Nakatsuji, H. Chem Phys Lett 1978, 59, 362; 1979, 67, 329; 1979, 67, 334.

24. Nakatsuji, H. Acta Chim Hung 1992, 129, 719.

25. Nakatsuji, H.; Hasegawa, J.; Hada, M. J Chem Phys 1996, 104, 2321; Toyota, K.; Hasegawa, J.; Nakatsuji, H. J Phys Chem A 1997, 101,446; Hada, M.; Imai, Y.; Hidaka, M.; Nakatsuji, H. J Chem Phys 1995, 103, 6993.

26. Nakai, H.; Ohmori, Y.; Nakatsuji, H. J Chem Phys 1991, 95, 8287; Nakatsuji, H.; Ehara, M.; Palmer, M. H.; Guest, M. F. *ibid.* 1992, 97, 2561; Yasuda K.; Nakatsuji, H. *ibid.* 1993, 99, 1945; Nakai H.; Nakatsuji, H. J Mol Struct: THEOCHEM 1994, 331, 141.

27. Jitsuhiro, S.; Nakai, H.; Hada, M.; Nakatsuji, H. J Chem Phys 1994, 101, 1029; Nakatsuji, H.; Ehara, M. *ibid.* 1994, 101, 7658.

28. Huzinaga, S.; Andzelm, J.; Klobukowski, M.; Radzio-Andzelm, E.; Sakai, Y.; Tatewaki, H. *Gaussian Basis Sets for Molecular Calculations* (Elsevier, Amsterdam, 1984).

29. Dunning, Jr. T. H. J Chem Phys 1970, 53, 2823.

30. Hay, P. J. J Chem Phys 1977, 66, 4377.

31. Pou-Amerigo, R.; Merchan, M.; Nebot-Gil, I.; Widmark, P. O.; Roos, B. Theor Chim Acta 1995, 92, 149.

32. Schaefer III, H. F. *Methods of Electronic Structure Theory* (Plenum, New York and London).

33. Bohn, R. K.; Haaland, A. J Organomet Chem 1968, 5, 2653; Haaland, A.; Nilsson, J. E. Acta Chem Scand (1947–1973) 1968, 22, 2653; Haaland, A.; Lusztyk, J.; Novak, D. P.; Brunvoll, J.; Starowieyski, K. B. J Chem Soc Chem Commun 1974, 1974, 54.

34. Orsenblum, M. *Chemistry of Iron Group Metallocenes* (Interscience, New York, 1965), Chap. 3, Part I.

35. Frisch, M. J.; Trucks, G. W.; Schlegel, H. B. *et al.*, GAUSSIAN 98, Revision A.8, Gaussian, Inc., Pittsburgh, PA, 1998.

36. Nakatsuji, H. Chem Phys 1983, 75, 425.

Conclusions

In Chapter I, a new algorithm for the generation of two-electron repulsion integrals (ERIs) was developed by combining Pople-Hehre and McMurchie-Davidson algorithms. The algorithm derived for (*ss*|*ss*) to (*dd*|*dd*) ERIs was tested using STO-3G, 6-31G, 6-31G(d), and aug-cc-pVDZ basis sets. The results show that the new algorithm reduces the computational cost by 10–40%, as compared with the original algorithms. It is notable that the generation of ERIs including *d* functions is considerably fast. The program implemented officially in GAMESS in 2004 has been used all over the world.

In Chapter II, a new two-step parallel algorithm for MP2 energy calculations was developed. Test calculations of taxol ($C_{47}H_{51}NO_{14}$) and luciferin ($C_{11}H_8N_2O_3S_2$) were performed on a cluster of Pentium 4 computers connected by gigabit Ethernet. The parallel scaling of the developed code is excellent up to the largest number of processors we have tested. For instance, the elapsed time for the MP2 energy calculations on 16 processors is on average 15.4 times faster than that on the single-processor. As an application to grid computing, a larger calculation (a segment of hydrogen terminated (5,0) carbon nanotube, $C_{130}H_{10}$) with the 6-31G(d) basis (1970 contracted basis functions) was performed using a total of 128 processors (64 Hitachi SR-11000 and 64 Hitachi HA-8000) at the NAREGI computer center. Elapsed time for the MP2 energy calculation was less than 2 hours and CPU efficiency was high (94%) even on this heterogeneous system.

In Chapter III, a new parallel algorithm for MP2 energy gradient calculations was developed. Test calculations of MP2 energy gradients were performed for taxol and luciferin on a cluster of Pentium 4 computers. The speedups are very good up to 80 CPU cores we have tested. For instance, the speedup ratios are 28.2 - 33.0 on 32 processors, corresponding to 88%–103% of linear speedup. This indicates the high parallel efficiency of the present algorithm. The calculation of taxol with 6-31G(d)

(1032 contracted basis functions) finishes within 2 hours on 32 processors, which requires only 1.8GB memory and 13.4GB hard disk per processor. Therefore, geometry optimization of molecules with 1000 basis functions can be easily performed using standard PC clusters.

In Chapter IV, MP2 and SCS-MP2 calculations were applied to several molecules. MP2 provides geometries that are much closer to experimental ones than DFT. SCS-MP2 performs as well as CCSD(T), though the former is much less time-consuming than the latter.

The algorithms and programs developed in this thesis make MP2 calculations practically feasible for considerably large molecules. It is planned to investigate host-guest molecules, self-assembly, and molecular recognition using the present programs, and to develop an efficient parallel algorithm for open-shell MP2 calculations of large molecules.

In Chapter V, the ionized and excited states of ferrocene were calculated using the SAC/SAC-CI theory. The calculated results are in good agreement with experimental values. It is found that shake-up processes (one electron ionization and one electron excitation) contribute to the first two ionization peaks.

List of Acronyms

AO	atomic orbital
B3LYP	Becke 3-parameter, Lee, Yang, Parr
B3PW91	Becke 3-parameter, Perdew-Wang 1991
CI	configuration interaction
CC	coupled-cluster
CCSD(T)	coupled-cluster with single, double, and perturbative triple excitation
DFT	density functional theory
ERI	electron repulsion integral
FLOP	floating point operation
GAMESS	the general atomic and molecular electronic structure system
HF	Hartree-Fock
MCSCF	multi-configurational self-consistent field
MO	molecular orbital
MP2	2nd order Møller-Plesset perturbation theory
PH	Pople-Hehre (scheme)
RI	resolution of identity
SAC	symmetry adapted cluster
SAC-CI	symmetry adapted cluster configuration interaction
SCF	self-consistent field
SCS-MP2	spin-component scaled 2nd order Møller-Plesset perturbation theory
TDDFT	time-dependent density functional theory

List of Publications

Chapter I:

"A new algorithm of two-electron repulsion integral calculations: a combination of Pople-Hehre and McMurchie-Davidson methods", Ishimura, K.; Nagase, S. Theoret Chem Acc in press.

Chapter II:

"A new parallel algorithm of MP2 energy calculations", Ishimura, K; Pulay, P.; Nagase S. J Comput Chem 2006, 27, 407.

Chapter III:

"New parallel algorithm for MP2 energy gradient calculations", Ishimura, K; Pulay, P.; Nagase S. J Comput Chem 2007, 28, 2034.

Chapter V:

"Ionized and excited states of ferrocene: Symmetry adapted cluster-configuration-interaction study", Ishimura, K.; Hada, M.; Nakatsuji, H. J Chem Phys 2002, 117, 6533.

Other publications:

"C72 isomers: the IPR-satisfying cage is disfavored by both energy and entropy" Slanina, Z.; Ishimura, K.; Kobayashi, K.; Nagase, S. Chem Phys Lett 2004, 384, 114.

"Synthesis and characterization of stable hypervalent carbon compounds (10-C-5) bearing a 2,6-bis(*p*-substituted phenyloxymethyl)benzene ligand"

Akiba, K.-y.; Moriyama, Y.; Mizozoe, M.; Inohara, H.; Nishii, T.; Yamamoto, Y.; Minoura, M.; Hashizume, D.; Iwasaki, F.; Takagi, N.; Ishimura, K.; Nagase, S. J Am Chem Soc 2005, 127, 5893.

"The aromaticity of the stannole dianion"
Saito, M.; Haga, R.; Yoshioka, M.; Ishimura, K.; Nagase, S. Angew Chem Int Ed 2005, 44, 6553.

"Synthesis of Stannaindenyl anions and a dianion"
Saito, M.; Shimosawa, M.; Yoshioka, M.; Ishimura, K.; Nagase, S. Organometallics 2006, 25, 2967.

"Synthesis and characterization of dimetallostannafluorenes"
Saito, M.; Shimosawa, M.; Yoshioka, M.; Ishimura, K.; Nagase, S. Chem Lett 2006, 35, 940.

"Computations of the energetic of $C_{60}F_{36}$ isomers"
Slanina, Z.; Uhlik, F.; Ishimura, K.; Nagase, S. Fullerenes Nanotubes Carbon Nanostruct 2006, 14, 57.

"Accuracy of the three-body fragment molecular orbital method applied to Møller-Plesset perturbation theory",
Fedorov, D. G.; Ishimura, K.; Ishida, T.; Kitaura, K.; Pulay, P.; Nagase, S. J Comput Chem 2007, 28, 1476.

“Synthesis and structure of pentaorganostannate having five carbon substituents” Saito, M.; Imaizumi, S.; Tajima, T.; Ishimura, K.; Nagase, S. J Am Chem Soc 2007, 129, 10974.

Acknowledgment

The thesis is a summary of author's studies carried out at Department of Theoretical and Computational Molecular Science, Institute for Molecular Science from 2002 to 2007. The author would like to express his sincere gratitude to Professor Shigeru Nagase for his guidance, valuable discussion, and warm encouragement.

The author is deeply grateful to Professor Peter Pulay and Dr. Michael W. Schmidt for their constructive discussion and kind encouragement.

The author appreciates the members of Nagase Group for valuable discussions and encouragement and the members of theoretical groups for their kindness and friendship.

The author is sincerely grateful to Professor Hiroshi Nakatsuji for his kind encouragement.

Finally, the author wishes to express his heartfelt gratitude to his parents and brother for their continuous support.

April, 2007

Kazuya Ishimura