Theor Chem Acc (2007) 117:397–405
DOI 10.1007/s00214-006-0171-8

REGULAR ARTICLE

A graphical symmetric group approach for a spin adapted full configuration interaction: partitioning of a configuration graph into sets of closed-shell and open-shell graphs

Kiyoshi Tanaka · Yuji Mochizuki · Takeshi Ishikawa · Hidemi Terashima · Hiroaki Tokiwa

Received: 6 July 2006 / Accepted: 3 August 2006 / Published online: 6 October 2006

Abstract We developed a spin adapted full configuration interaction (FCI) method which was expected to be effective for parallel processing. The graphical symmetric group approach (GSGA) was employed, where a configuration graph was partitioned into several sets of closed-shell and open-shell graphs. The configuration state functions (CSFs) bearing the same number of closed-shells and open-shells were assembled in a configuration group. The graphical approach provided a number to identify each CSF in a sequential order within the group. Combination of this partitioning and an intermediate configuration-driven algorithm in calculating the so-called σ vectors allowed us to use symbolic coupling constants. Furthermore, this combination made it easy to implement an efficient algorithm suitable to task-distributed parallel procedure for evaluating σ vectors. A program was written and some test calculations were carried out with high parallel efficiency. The largest size of FCI used 10 million CSFs (20 million determinants).

Keywords Spin adapted FCI · GSGA · Intermediate configuration-driven algorithm · Parallel processing

K. Tanaka (✉) · Y. Mochizuki · T. Ishikawa · H. Tokiwa
CREST Project, Japan Science and Technology Agency, 4-1-8, Honcho, Kawaguchi, Saitama 332-0012, Japan
e-mail: k-tanaka@fsis.iis.u-tokyo.ac.jp

K. Tanaka · Y. Mochizuki
Advancesoft, Center for Collaborative Research, The University of Tokyo, 4-6-1, Komaba, Meguro-ku, Tokyo 153-8904, Japan

K. Tanaka · Y. Mochizuki
Institute of Industrial Science, The University of Tokyo, 4-6-1, Komaba, Meguro-ku, Tokyo 153-8904, Japan

T. Ishikawa · H. Tokiwa
Department of Chemistry, Faculty of Science, Rikkyo University, 3-34-1, Nishi-ikebukuro, Toshima-ku, Tokyo 171-8501, Japan

H. Terashima
Faculty of Cultural Information Resources, Surugadai University, 698, Asu, Hanno, Saitama 357-8555, Japan

Present Address:
Y. Mochizuki
Department of Chemistry, Faculty of Science, Rikkyo University, 3-34-1, Nishi-ikebukuro, Toshima-ku, Tokyo 171-8501, Japan

1 Introduction

Full configuration interaction (FCI) provides the exact eigenfunction of the non-relativistic Schrödinger equation within a given molecular orbital (MO) space. The method plays a significant role to study electronic structure of molecules and there is a long history of FCI calculations with a valence orbital set (FVCI, in other words). As a special variant of multi-configuration self consistent field (MCSCF) [1,2], Roos and co-workers [3] proposed a complete active space self consistent field (CASSCF), in which FCI is employed to determine CI coefficients of CASSCF wave function. Furthermore FCI has been an important benchmark tool to assess various approximate electron correlation methods. Considerable efforts have been devoted in developing and implementing efficient algorithms for FCI [4–13], because a CASSCF calculation requires large active space in studying complicated chemical systems and quantum dots [13,14] as well as in calibrating calculations of the results from other approximate methods.

Parallelized FCI programs recently developed were mostly based on determinants [9–12] rather than configuration state functions (CSFs) [13], because of the simplicity in calculating Hamiltonian matrix elements and of high efficiency in processing the so-called $\boldsymbol{\sigma}$ vectors. We, sometimes, feel it important to have information about a spin coupling scheme of an open shell wave function, whereas a determinant based wave function does not give directly its spin coupling scheme. Especially, this may become important in studying systems containing transition metal ions, e.g., active regions of a family of cytochrome [15]. Since different spin states are energetically close in such systems and relative stability is sometimes a matter of extensive study, it is desirable to develop a method obtaining a spin projected FCI wave function which provides spin coupling scheme.

Since algorithm of a spin adapted FCI method should include much more complicated logics than the determinant based methods, it is desirable to develop efficient algorithms which realize high performance of parallel processing, needless to say the use of vector processing. In order to achieve a highly efficient spin adapted FCI program, we employed a graphical symmetric group approach (GSGA) [16–18] in which we proposed partitioning of a configuration graph into several sets of closed-shell and open-shell graphs. A process calculating the so-called $\boldsymbol{\sigma}$ vector is the most demanding in CI calculation. As will be shown later, this partitioning allowed us to localize combination of a CI vector and a matrix of symbolic expression for one electron operators (coupling constants) in calculating a $\boldsymbol{\sigma}$ vector. This also allowed us to develop an intermediate configuration-driven algorithm by which the procedure obtaining a $\boldsymbol{\sigma}$ vector was easily parallelized without communication across worker processors.

In the following, we will discuss the developed methods and algorithms. We will also discuss how the partitioning of the configuration graph guarantees data locality suitable to parallel processing. The performance of parallel processing will be presented.

2 Theory and algorithm

2.1 Outline of the method

Before going to discuss details, the outline of the method will be shown in this subsection. As an approximate wave function of the eigenfunction of the Schrödinger equation, an FCI wave function is obtained by solving an eigenvector of the Hamiltonian matrix over a set of CSF's, $\{\Phi_I\}$, which is generated from all the possible electronic configurations over a given active MO set. Since a few lowest eigenvalues are needed as usual, we employed the iterative method to obtain eigenvector(s) proposed by Davidson [19] and extended by Liu [20]. Then, the most time consuming step is the evaluation of the $\boldsymbol{\sigma}$ vector

$$\sigma_I^{(i+1)} = \Sigma_J H_{IJ} c_J^{(i)}, \tag{1}$$

where $\boldsymbol{c}^{(i)}$ is a correction vector used in the ith iteration and $\boldsymbol{H}$ is the Hamiltonian matrix.

The electronic Hamiltonian operator is expressed in the second quantized form as follows;

$$H = \Sigma_{pq}\{h_{pq} - \frac{1}{2}\Sigma_s(ps|sq)\}E_{pq} + \frac{1}{2}\Sigma_{pqrs}(pq|rs)E_{pq}E_{rs}, \tag{2}$$

where indices p, q, r, and s denote ortho-normalized MOs, E_{pq} is the one electron operator of the unitary group generator, h_{pq} represents a one-electron Hamiltonian matrix element, $(pq|rs)$ is a two electron integral.

In evaluating a $\boldsymbol{\sigma}$ vector, we employed the scheme introducing the resolution of the identity, $\Sigma_K \Phi_K >< \Phi_K$ between E_{pq} and E_{rs}, just as done by Siegbahn [4]. The most time consuming part of obtaining $\boldsymbol{\sigma}$ is evaluation of $\boldsymbol{\sigma}'$, which is expressed as;

$$\sigma'_I = \Sigma_{pqrs}\Sigma_K(pq|rs)\langle\Phi_I, E_{pq}\Phi_K\rangle \cdot\Sigma_J\langle\Phi_K, E_{rs}\Phi_J\rangle c_J. \tag{3}$$

Evaluation of the vector $\boldsymbol{\sigma}'$ was carried out by setting intermediate states K as the outermost loop and Eq. (3) was divided into matrix-oriented three parts:

$$D^K_{rs} = \Sigma_J\langle\Phi_K, E_{rs}\Phi_J\rangle c_J, \tag{4}$$

$$G^K_{pq} = \Sigma_{rs}(pq|rs)D^K_{rs}, \tag{5}$$

$$\sigma'_I = \Sigma_K\langle\Phi_I, E_{pq}\Phi_K\rangle G^K_{pq}, \tag{6}$$

In the present algorithm, the list of CSFs were ordered so that the parallel processing was performed efficiently. A CSF was specified as $\Phi^{(Nc,No)}_{Jj}$, where Nc and No represent number of closed-shells and open-shells, respectively, J indicates a configuration belonging to the (Nc, No) group, and j indicates an independent CSF of the configuration J. Then, coupling constants connecting with $\Phi^{(Nc,No)}_{Kk}$ should fall only in the following types:

$$\langle\Phi^{(Nc,No)}_{Kk}, E_{pq}\Phi^{(Nc,No)}_{Ii}\rangle,$$

and

$$\langle\Phi^{(Nc,No)}_{Kk}, E_{pq}\Phi^{(Nc\pm1,No\mp2)}_{Ii}\rangle.$$

If we are able to evaluate J and j easily so as to specify CSFs $\Phi^{(Nc,No)}_{Jj}$ in a sequential order and make use of symbolic expression for the coupling constants, we can localize the data handling in carrying out Eqs. (4)–(6).

This is the reason why we used the GSGA representation, where a configuration graph (or an orbital graph) [18] is partitioned into sets of closed- and open-shell graphs of (Nc, No), i.e. the index J is easily evaluated using the both graphs, and the index j is specified by a branching diagram [18,21].

The program evaluating a σ vector was coded in the form driven by intermediate configurations. The outermost loop was selected to be the configuration groups (Nc, No). After evaluation of the symbolic expression matrices, the loop of intermediate configuration K was set to run. This structure allowed us to employ the technique of a parallel processing by assigning a processor to all the processes concerning a given intermediate configuration K, because Eqs. (4)–(6) do not require communication with the other K and load of all the processes concerning a given K are uniform in a configuration group (Nc, No).

Since we are interested in studying molecules of complicated structure, no symmetry constraint was imposed on the CSFs.

2.2 Partitioning of a configuration graph and address of CSFs

The GSGA approach is described in detail by Duch and Karwowski [18]. We will focus our attention on partitioning of a configuration graph into closed-shell and open-shell graphs using an example of triplet state of six electrons over six active orbitals (designated as 6/6). Figure 1 shows configuration graphs and a branching diagram describing all the possible CSFs of FCI of the triplet state for the case of 6/6. The electronic configurations are classified into sets of configurations which are specified by Nc and No. In this case, CSFs are grouped into three sets: (2, 2), (1, 4), and (0, 6) for (Nc, No), respectively. The figure shows combinations of a closed-shell and an open-shell graphs corresponding to the respective group, (Nc, No). A closed-shell graph of the group (2, 2) indicates all the possible selection of two orbitals as closed shell orbitals among the six active MOs and an open-shell graph of (2, 2) shows all the possible open-shell configurations distributing two electrons among the rest of orbitals, (four for this case). A number in a circle of each graph indicates the total number of configurations expressed by the graph. This quantity is designated as $M\_\mathrm{conf}\_c^{(N\mathrm{c},N\mathrm{o})}$ for closed-shell case or $M\_\mathrm{conf}\_o^{(N\mathrm{c},N\mathrm{o})}$ for open-shell case. A branching diagram shown also in this figure represents spin coupling scheme of CSFs and numbers in circles given for the three values of No are the number of independent CSFs in accordance with the three cases. This quantity is designated as $M\_\mathrm{CSF}\_o^{(N\mathrm{c},N\mathrm{o})}$. The variables, N_el, N_amo, N_omo, and $2S+1$, shown in the figure mean the number of electrons, numbering of the active MOs, and numbering of MOs available for open-shells, and spin multiplicity, respectively. A configuration is represented by a walk from the origin to the outermost point of (N_el, N_amo) or (N_el, N_omo), which is a point with number in a circle. A spin coupling scheme of an open-shell CSF is represented by a walk from the origin to the target points, $2S+1=3$ and $N\mathrm{o}=2,4,6$ in this example.

A CSF is specified by 'walks' in the closed-shell and open-shell graphs, and the branching diagram. The number given on an oblique line indicates a weight assigned to the line. The weight of lines without number is zero. We followed Duch and Karwowski [18] in assigning a weight to a line. By accumulating weights along with a 'walk' from the origin to the terminal, we obtain a walk weight for respective configurations or CSFs. The walk weights of the closed-shell graph, the open-shell graph, and the branching diagram are designated as $\mathrm{Walk\_weight}\_c^{(N\mathrm{c},N\mathrm{o})}$, $\mathrm{Walk\_weight}\_o^{(N\mathrm{c},N\mathrm{o})}$, and $\mathrm{CSF\_weight}\_o^{(N\mathrm{c},N\mathrm{o})}$, respectively. The internal address of a CSF, $\Phi_{J,j}^{N\mathrm{c},N\mathrm{o}}$, belonging to the configuration group (Nc, No) is given by the following equations

$$N\_\mathrm{csf\_address}^{(N\mathrm{c},N\mathrm{o})} = J + j, \tag{7}$$

$$J = \left(\mathrm{Walk\_weight\_c}^{(N\mathrm{c},N\mathrm{o})} M\_\mathrm{conf}\_o^{(N\mathrm{c},N\mathrm{o})} + \mathrm{Walk\_weight}\_o^{(N\mathrm{c},N\mathrm{o})}\right) \mathrm{M\_CSF}\_o^{(N\mathrm{c},N\mathrm{o})}, \tag{8}$$

$$j = \mathrm{CSF\_weight}\_o^{(N\mathrm{c},N\mathrm{o})} + 1. \tag{9}$$

2.3 Symbolic expression of the coupling constant

Since mapping of orbitals in the open-shell graph is dependent on the occupied orbitals in the closed-shell graph, the numbering of open-shell orbitals are given symbolically in the open-shell configuration graph. This means that CSFs are presented symbolically and gives rise to advantage in using symbolic expressions of coupling constants. A coupling constant between CSFs which belongs to the same configuration group (Nc, No) is

(a) Originated by an electron annihilation from an open-shell orbital and creation on an empty orbital, or
(b) Annihilation from a closed-shell (create an open-shell) and creation on an open-shell orbital (create a closed-shell).

Both cases mean change in ordering of open-shell orbitals in the spin coupling scheme. The former type

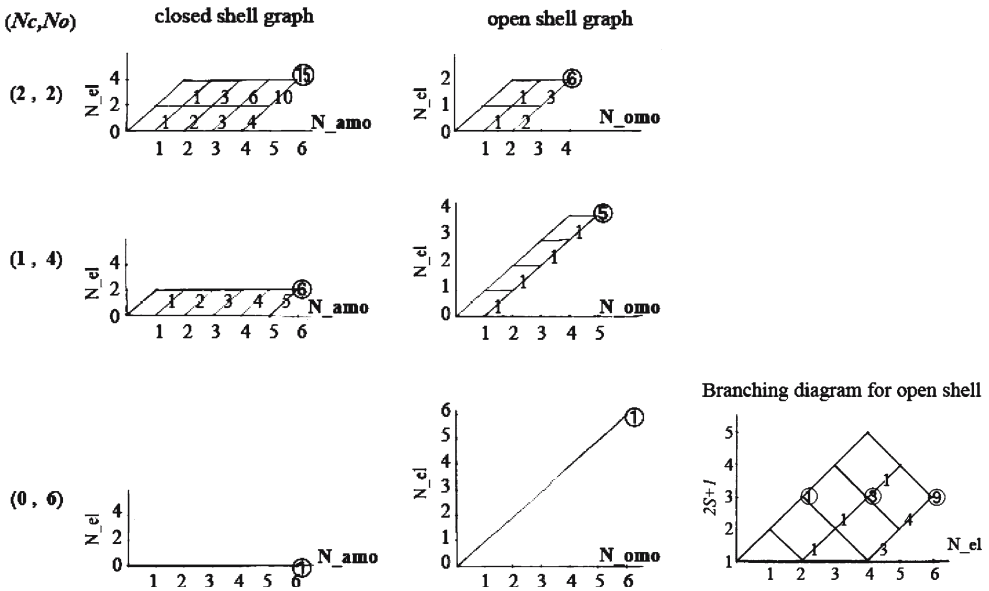

Fig. 1 Configuration graphs and a branching diagram; Nc, No, N_el, N_amo, and N_omo mean the number of closed-shells, the number of open-shells, the number of electrons, numbering of the active MOs, and numbering of MOs available for open-shells, respectively. Details are described in the text

coupling constant is used for the latter type of coupling constant, although the phase factor is inverted. The coupling constants between the groups of (Nc, No) and $(Nc-1, No+2)$ is originated from an electron annihilation from a closed-shell orbital and creation on an empty orbital. Only special attention should be paid to the ordering of resulted open-shell orbitals and similar consideration should be paid to evaluation of the coupling constants between the groups of (Nc, No) and $(Nc+1, No-2)$.

In calculating the symbolic coupling constants, we used a code of CSFs reflecting branching diagram [22,23]. By defining S_J and s_j as indices of symbolic configuration and order of the independent CSFs of the configuration S_J, respectively, a symbolic CSF is specified by S_J, and s_j. Since the CSFs are expanded by determinants, Davidson's scheme [24] was used for fast evaluation of the symbolic coupling constants,

$$\langle \Phi^{Nc,No}_{S_K,s_k}, E_{s_p,s_q}\Phi^{Nc',No'}_{S_J,s_j}\rangle ,$$

where s_p and s_q indicate symbolic orbital pair by which $E_{s_p,s_q}\Phi^{Nc',No'}_{S_J,s_j} >$ belongs to the configuration S_K of (Nc, No). It should be noted that the symbolic expression is given in a matrix form.

2.4 Algorithm and program structure

We discuss algorithms of the portion calculating $\boldsymbol{\sigma}'$ in more detail. Since very large scale fast core memory is available in a modern computer, we developed a program by designing to keep a few vectors of $\{\boldsymbol{c}^{(i)}\}$ and $\{\boldsymbol{\sigma}^{(i+1)}\}$ on the fast memory. Figure 2 illustrates data flow in calculating the $\boldsymbol{\sigma}'$ vector. As was described in the Sect. 2.3, the outermost loop is the configuration group (Nc, No) of the intermediate configuration and the next loop is driven by K. The data flow from right hand side to $\boldsymbol{D}^K$ in the middle corresponds to carrying out Eq. (4). Since the combination of K and E_{pq} specifies the configuration J, this procedure is driven over pq. After completing the process concerning all the possible pq, the calculation of Eq. (5) is carried out, which is the data flow from $\boldsymbol{D}^K$ to $\boldsymbol{G}^K$. The data flow from $\boldsymbol{G}^K$ to the left hand side corresponds to Eq. (6) by which the $\boldsymbol{\sigma}'$ vector is updated by the contributions from the configuration K of (Nc, No). It should be worth noting that

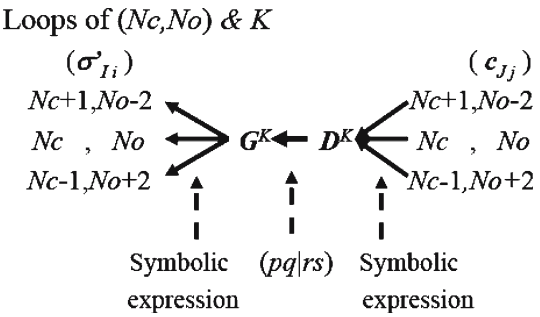

Fig. 2 Illustration of the data flow in calculating $\boldsymbol{\sigma}'$ vector (*solid lines* data flow, *dotted lines* usage of data)

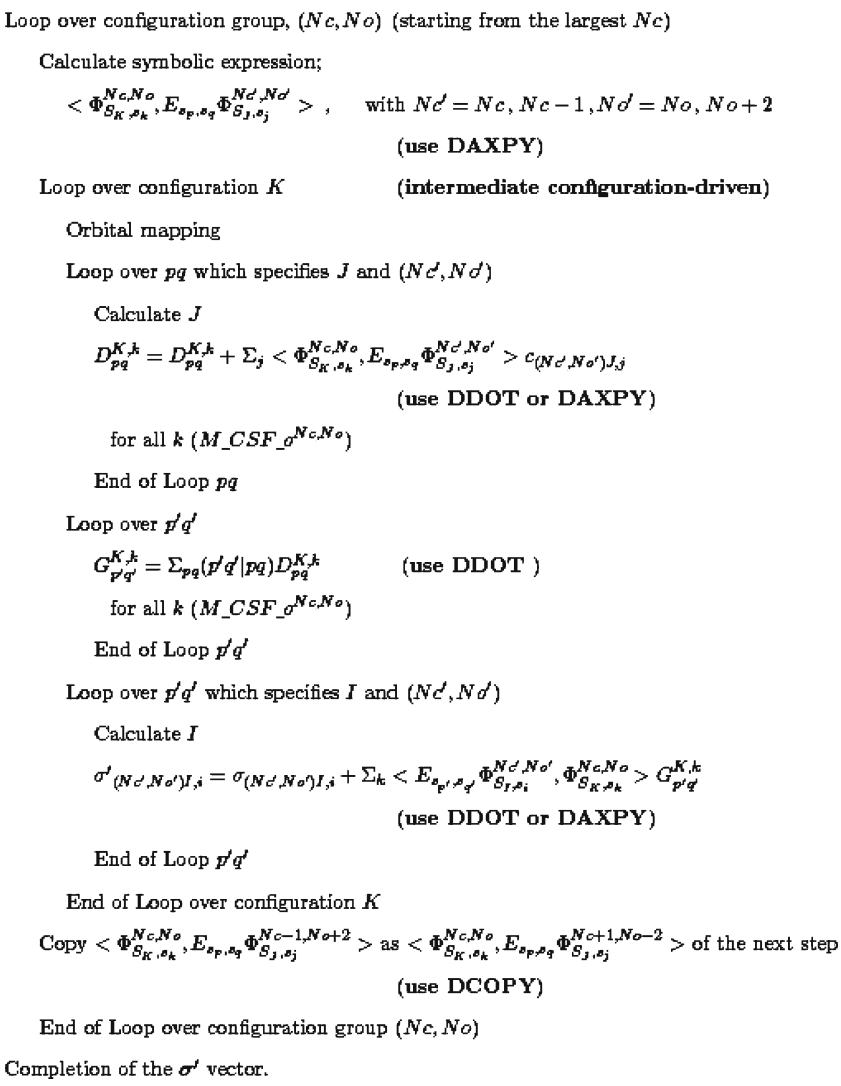

Fig. 3 Outline of the intermediate configuration-driven algorithm

(a) No communication with other intermediate configurations is required.
(b) The length of the list of pair pq is in common for each configuration K of the configuration group $(N\mathrm{c}, N\mathrm{o})$, and this guarantees the load in each K should be uniform.

The outline of the procedure obtaining the σ' vector is illustrated in Fig. 3.

In developing codes, we kept in mind to make use of the Basic Linear Algebra Subroutines (BLAS) [25] for vector processing in the innermost loops as well as possible. The subroutines DAXPY, DDOT and DCOPY [25] of BLAS were employed for the innermost loops as indicated in Fig. 3.

2.5 Parallel implementation

In the procedure presented in Sect. 2.4, the length of the loop K for the intermediate configuration is strongly dependent on a configuration group $(N\mathrm{c}, N\mathrm{o})$, whereas the load within the loop K is uniform for every intermediate configuration of $(N\mathrm{c}, N\mathrm{o})$, i.e., the same length of the procedures for each K after mapping of active orbitals to closed-shell MOs and open-shell MOs. Since no communication with other intermediate configurations is required throughout the process of Eqs. (4)–(6), as shown in Figs. 2 and 3, it should be a good policy to take the index K as the target of the standard message-passing interface (MPI) [26] parallel procedure. When a task list is distributed by the index of the loop K, a piece of the σ' vector is distributed in each processor without redundancy. Suppose the index $K^{(\mu)}$ is designated as an intermediate configuration of $(N\mathrm{c}, N\mathrm{o})$ which is assigned to the μth processor, the computational procedure in the μth processor is performed as follows:

(a) $D^{K^{(\mu)},k}_{rs}$ is obtained, where rs is a single index given in a canonical order and k indicates an independent CSF of the configuration $K^{(\mu)}$.

(b) $G_{pq}^{K(\mu),k}$ is obtained where pq is a single index given in a canonical order.

(c) $\Sigma_k < \Phi_{S_I,s_i}^{Nc',No'}, E_{s_p,s_q}\Phi_{S_K,s_k}^{Nc,No} > G_{pq}^{K(\mu),k}$ is added to ${\sigma'}_{I,i}^{(\mu)}$, where $\boldsymbol{\sigma}'^{(\mu)}$ is contribution from μth processor to the $\boldsymbol{\sigma}'$ vector and I is specified by selecting the pair of p and q.

(d) Accumulation of $\boldsymbol{\sigma}'^{(\mu)}$ distributed on processors can be carried out after the loop (Nc, No). This is done by calling the MPI_ALL_REDUCE [26] with negligibly small barrier operation on worker processes.

The operation, MPI_ALL_REDUCE, is carried out only once an iteration, because the $\boldsymbol{\sigma}$ vector is appended once in each eigenvector iteration.

In this subsection, we have shown how parallel code will be easily developed using intermediate configuration-driven algorithm with the partitioning of configuration graph into groups of closed shell and open shell graphs. In this respect, it is interesting to note that this scheme can be effectively applied to the scheme by Zarrabian, Sarma, and Paldus [7], with the number of electrons being reduced by 2 in intermediate configurations.

3 Performance of a spin adapted parallel FCI

Firstly, we checked if the present code reproduces previous works or results obtained by an existing program. In this respect, we carried out SCF and FCI calculations of the lowest singlet ($^1A_1$) and triplet ($^3B_1$) states of CH_2 and the doublet state of NH_2 using the program code of GAMESS [27], with the K-shell $1a_1$ electrons kept frozen. The present program was invoked with the integrals and MOs resulted from GAMESS for these three states. The basis sets were those of the Dunning double zeta (DZ) contraction [28] of the Huzinaga primitive sets [29] (with the exponents scaled for hydrogen). The present total energies agreed with those given by GAMESS up to the eighth decimal places. The test calculations were continued for the $^1A_1$ and $^3B_1$ states of CH_2 using the integrals over the natural orbitals from the respective FCI calculations. The total energies of these two states reproduced the total energies using integrals over Hartree Fock MOs.

As further test calculations, we took the lowest singlet and triplet states of CH_2 with a larger basis sets [DZ plus polarization functions (DZP)] by Bauschlicher [30] and the lowest doublet state of NH_2 with DZP reported by Bauschlicher [31]. The results using molecular integrals and restricted Hartree Fock (RHF) and restricted open shell Hartree Fock (ROHF) orbitals from GAMESS are shown in Table 1 in comparison with Bauschlicher's results. The ROHF calculations were carried out using a Fock matrix obtained by averaging over the up and down spin Fock matrices of unrestricted Hartree-Fock (UHF) scheme with even weights for the diagonal blocks of closed, open, and virtual spaces. The resulted total energies of the three species agree with Bauschlicher's results, as shown in Table 1. It should be noted that the same type of FCI with the ROHF MOs by Roothaan's scheme [32] gave total energies of -39.046295 a.u. for CH_2 ($^3B_1$) and -55.7426493 a.u. for NH_2 ($^2B_1$) which are a little bit different from the present values owing to the K shell frozen (see Table 1), needless to say that ROHF energies coincided with the present ones.

Parallel performance was tested using these three examples of CH_2 and NH_2 of the DZP basis set, firstly. The timing data was taken using a linux-based cluster of 5 nodes Intel Dual-Xeon (2.8 GHz; dual core version).[1]

The timing data and efficiency of parallel computation are shown in Table 2, where timing data are almost equal to CPU time in calculating a $\boldsymbol{\sigma}$ vector plus the communicating time required in the process of MPI_ALL_REDUCE. The efficiency is given by the ratio of the time carried out with only one processor to the time multiplied by the number of the processors. The timing data are those needed in calculating a $\boldsymbol{\sigma}$ vector plus the process of MPI_ALL_REDUCE.

One may find very high score of the parallel efficiency. Since a node of the present system possesses two CPUs, there are two ways of two CPU parallel calculation; using two nodes with one CPU a node and one node with two CPUs. The performance of the former calculation should be higher than that of the latter, because of no conflict in memory accessing for the former case. The performance of the parallel calculation of NH_2 with four CPUs (four nodes with one CPU a node) is 0.97 as shown in Table 2, whereas the performance was reduced to about 0.90 when we used two CPUs from each node. This is the reason why the performance with CPUs more than or equal to six is lowered a little in comparison with cases of two, four, and five CPUs, because the number of nodes with two CPUs increases as the number of the parallel processes increases.

Furthermore, in the present system, one node is capable of invoking four tasks at the same time, although efficiency should be lowered much in comparison with the case of one task running on a node. The cases of running parallel processes 16 and 20 include such nodes partly and fully, respectively. Even with this restriction, the efficiency of the parallel computation is about 0.75

[1] All test calculations were carried out on this cluster machine at Rikkyo Univeristy.

Table 1 Results of test calculations on CH_2 and NH_2 with DZP basis

Molecule	CH_2	CH_2	NH_2
State	$^1A_1$	$^3B_1$	$^2B_1$
R(AH) (a.u.)[a]	2.11	2.045	1.935
∠ HAH (°)[a]	102.4	132.4	103.4
Number electron/number active MO	6/25	6/25	7/24
Total number of CSF's	1,495,000	2,466,750	9,775,920
Total energy (a.u.)	−39.0271829	−39.0462596	−55.7426204
Coefficient of SCF CSF	0.95248	0.97337	0.96875
Bauschlicher	−39.027183[b]	−39.046260[b]	−55.742620[c]

[a] A = C or N
[b] Ref. [30]
[c] Ref. [31]

Table 2 Parallel performance on CH_2 and NH_2

Molecule	CH_2	CH_2	NH_2
State	$^1A_1$	$^3B_1$	$^2B_1$
Basis set	DZP	DZP	DZP
Number electron/number active MO	6/25	6/25	7/24
Total number of CSFs	1,495,000	2,466,750	9,775,920
Total number of determinants	5,290,000	3,795,000	21,507,024
Number of configuration groups	4	3	4
Number of parallel processes[a]	Time (min)/iter. and efficiency (in parentheses)		
1: (1CPU,1CORE)	2.25 (1.00)	3.44 (1.00)	12.40 (1.00)
2: 2×(1CPU, 1CORE)	1.14 (0.99)	1.74 (0.99)	6.29 (0.99)
4: 4×(1CPU, 1CORE)	0.59 (0.96)	0.89(0.97)	3.20 (0.97)
5: 5×(1CPU, 1CORE)	0.48 (0.95)	0.72 (0.95)	2.60 (0.95)
6: 4×(1CPU, 1CORE)+1×(2CPU, 2CORE)	0.40 (0.93)	0.61 (0.94)	2.20 (0.94)
8: 2×(1CPU, 1CORE)+3×(2CPU, 2CORE)	0.31 (0.91)	0.47 (0.92)	1.69 (0.92)
10:5×(2CPU, 2CORE)	0.26 (0.87)	0.39 (0.88)	1.39 (0.89)
16:2×(2CPU, 2CORE)+3×(2CPU,4CORE)	0.18 (0.80)	0.26 (0.82)	0.96 (0.81)
20:5×(2CPU, 4CORE)	0.15 (0.75)	0.23 (0.76)	0.83 (0.75)

Five nodes Intel Dual-Xeon (dual core version; 2.8 GHz) were used
[a] (1CPU, 1CORE) ≡ 1CPU/NODE and 1CORE/CPU, (2CPU, 2CORE) ≡ 2CPU/NODE and 1CORE/CPU, (2CPU, 4CORE) ≡ 2CPU/NODE and 2CORE/CPU

with 20 (5×(2CPU, 4CORE) = 2processes/CPU) parallel processors.

Further test calculations of larger number of active electrons (including larger number of open shells) were carried out by a model system of $[FeO(NH_3)]^{2+}$ (shown in Fig. 4) with a 6-31G basis set [33]. The number of basic functions was 53.

Geometry parameters optimized by a standard density functional method are given in Table 3. Using quintet state ROHF MOs, triplet and quintet state FCI calculations were carried out: The lowest 15 MOs were kept frozen and the next 15 valence type MOs were taken as active orbitals and 12 valence electrons were distributed among the active orbitals.

Table 3 Geometry parameters of $[FeO(NH_3)]^{2+}$ (in a_0)

Atom	x	y	z
Fe	0.5868279493	−0.8166847698	0.0000000000
O	1.9960136259	1.8898903939	−0.0000056692
N	−2.9281039618	0.5320958155	−0.0000018897
H	−3.2409159737	1.6577696816	1.5825510284
H	−3.2403981888	1.6597217686	−1.5812924709
H	−4.2475900149	−0.9275096401	−0.0011697404

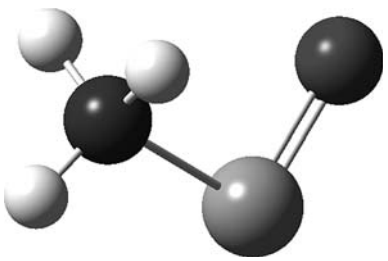

Fig. 4 Structure of $[FeO(NH_3)]^{2+}$ optimized by a standard density functional method

The timing data and efficiency of parallel computation are shown in Table 4 together with total energies and the number of CSFs. The efficiency is decreased a little bit in comparison with the previous case. Since the processes concerning intermediate configurations were simply distributed among processors in each (Nc, No),

Table 4 Parallel performance on $[FeO(NH_3)]^{2+}$

Spin multiplicity	Quintet	Triplet
Number electron/number active MO	12/15	12/15
Total number of CSF's	6,506,500	10,540,530
Total number of determinants	8,783,775	19,324,305
Number of configuration groups	5	6
Total energy (a.u.)	−1,392.4513339	−1,392.4410531
Number of parallel processes[a]	Time (min)/iter. efficiency (in parentheses)	
1: (1CPU,1CORE)	7.82 (1.00)	11.56 (1.00)
2: 2×(1CPU, 1CORE)	3.98 (0.98)	5.89 (0.98)
4: 4×(1CPU, 1CORE)	2.02 (0.97)	3.02 (0.96)
5: 5×(1CPU, 1CORE)	1.66 (0.94)	2.46 (0.94)
6: 4×(1CPU, 1CORE)+1×(2CPU, 2CORE)	1.48 (0.88)	2.18 (0.88)
8: 2×(1CPU, 1CORE)+3×(2CPU, 2CORE)	1.14 (0.87)	1.68 (0.86)
10:5×(2CPU, 2CORE)	0.94 (0.83)	1.40 (0.83)
16:2×(2CPU, 2CORE)+3×(2CPU,4CORE)	0.72 (0.68)	1.13 (0.64)
20:5×(2CPU, 4CORE)	0.64 (0.61)	0.98 (0.59)

Five nodes Intel Dual-Xeon (dual core version; 2.8 GHz) were used

[a] Same as in Table 2

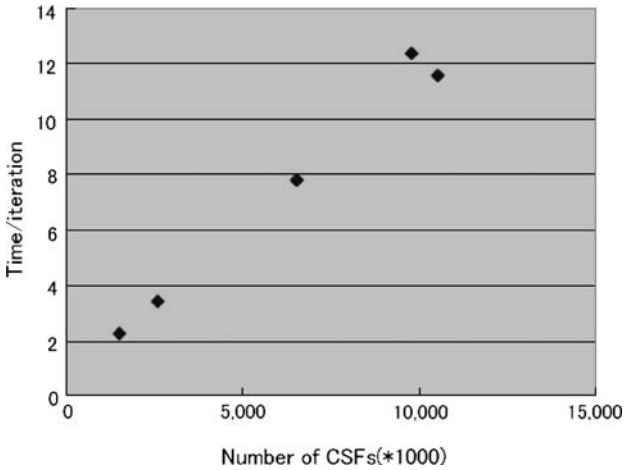

Fig. 5 Used time versus the number of CSFs

deterioration in load-valance should be conspicuous when

(a) Residues of intermediate configurations caused by indivisibility of the number of configurations of some configuration group by the number of processors.
(b) The number of configuration groups is increased.

These features are more enhanced in the case of $[FeO(NH_3)]^{2+}$ in comparison with the cases of CH_2 and NH_2. We are now devising to improve this defect. Also improvements to gain in speed are in progress, for example replacing some first order BLAS routines by the second order ones.

Other than the parallel performance in calculation of the σ vector, we noted that the timing data is almost proportional to the number of CSFs as shown in Fig. 5.

This is originated from the intermediate configuration-driven algorithm with almost uniform computational processes for each intermediate configuration. This is an advantageous aspect of the present algorithm.

4 Concluding remarks

In this paper, we report a GSGA approach of FCI, where a configuration graph is partitioned into several sets of closed- and open-shell graphs in which the number of closed-shells and the number of open-shells are specified. This partitioning allows us to make intermediate configuration-driven strategy advantageous to parallel processing of the σ vector by distributing a task concerning an intermediate configuration to a processor. The efficiency of this way is guaranteed by well-balanced load of the task and no communication among intermediate configurations. The ALL_REDUCE is required only once an iteration in eigenvalue procedure. By the use of a linux-based cluster of Dual-Xeon Processors (dual core version; 2.8 GHz clock-rate), we carried out fairly large scale FCI calculations of CH_2, NH_2, and $[FeO(NH_3)]^{2+}$. High efficiency was obtained by the method. It turned out that process time is nearly proportional to the number of CSFs. This would be helpful to perform large scale FCI including heavier ions.

Acknowledgements K. T. expresses his appreciation to Dr. Tatsuya Nakano for his kind comments on parallel computation. The authors are grateful to Prof. Shigenori Tanaka for

his interest and encouragement on this problem. This work was supported primarily by the CREST project operated by Japanese Science and Technology Agency (JST) and partially supported by the Revolutionary Simulation Software for 21st Century (RSS21) project operated by Ministry of Education, Culture, Sports, Science and Technology.

References

1. Dupis M(ed) (1980) Recent developments and applications of multi-configuration Hartree-Fock methods. LBL-12157 technical report, Lawrence Berkeley Laboratory, Berkeley, CA
2. Werner H-J (1987) Ab initio methods in quantum chemistry-II., pp 1–62, Wiley, New York
3. Roos BO, Taylor PR, Siegbahn PEM (1980) Chem Phys 48:157
4. Siegbahn PEM (1984) Chem Phys Lett 109:417
5. Handy NC (1980) Chem Phys Lett 74:280
6. Knowles PJ, Handy NC (1984) Chem Phys Lett 111:315
7. Zarrabian S, Sarma CR, Paldus J (1989) Chem Phys Lett 155:183
8. Harrison RJ, Zarrabian S (1989) Chem Phys Lett 158:393
9. Olsen J, Roos B, Jørgensen P, Jensen HJA (1988) J Chem Phys 89:2185
10. Ansaloni R, Bendazzoli GL, Evangelisti S, Rossi E (2000) Comput Phys Commun 128:496
11. Klene M, Robb MA, Frisch MJ, Celani P (2000) J Chem Phys 113:5653
12. Gan Z, Alexeev Y, Gordon MS, Kendall RA (2003) J Chem Phys 119:47
13. Recently, CSF based method was published applicable to the few-electron problem in artificial atoms (semiconductor quantum dot). Rontani M, Cavazzoni C, Bellucci D, Goldoni G (2006) J Chem Phys 124:124102
14. Corni S, Braskén M, Lindberg M, Olsen J, Sundholm D (2003) Phys Rev B 67:085314
15. Meunier B, de Visser SP, Shaik S (2004) Chem Rev 104:3947
16. Sasaki F (1974) Int J Quant Chem VIII:605
17. Sasaki F, Tanaka K, Noro T, Togasi M, Nomura T, Sekiya M, Gonoi T, Ohno K (1987) Theor Chim Acta 72:123
18. Duch W, Karwowski J (1985) Comput Phys Rep 2:93
19. Davidson ER (1975) J Comput Phys 17:87
20. Liu B (1978) In: Moler C, Shavitt I (eds) Numerical algorithms in chemistry: algebraic methods. LBL-8158 Technical Report, Lawrence Berkley Laboratory, Berkley, CA
21. Kotani M, Amemiya A, Ishiguro E, Kimura T (1955) Tables of Molecular Integrals (Maruzen Co., Tokyo)
22. Mochizuki Y, Nishi N, Hirahara Y, Takada T (1996) Theor Chim Acta 93:211
23. Pauncz R (2000) The construction of spin eigenfunctions Kluwer Academic/Plenum Publishers, New York
24. Davidson ER (1974) Int J Quant Chem VIII:83
25. Basic Linear Algebra Subroutines (http://www.netlib.org/blas/)
26. Message-Passing Interface (http://www.mpi-forum.org/)
27. Schmidt MW, Baldridge KK, Boatz JA, Elbert ST, Gordon MS, Jensen JH, Koseki S, Matsunaga N, Nguyen KA, Su SJ, Windus TL, Dupuis M, Montgomery JA (1993) J Comp Chem 14:1347 (http://www.msg.ameslab.gov/GAMESS/)
28. Dunning TH (1970) J Chem Phys 53:2823
29. Huzinaga S (1965) J Chem Phys 42:1293
30. Bauschlicher Jr. CW, Taylor PR (1986) J Chem Phys 85: 6510
31. Bauschlicher Jr. CW, Langhoff SR, Taylor PR, Handy NC, Knowles PJ (1986) J Chem Phys 85:1469
32. Roothaan CCJ (1960) Rev Mod Phys 32:179
33. Foresman JB, Frisch A (1996) Exploring chemistry with electronic structure Methods, 2nd edn. Gaussian Inc. Pittsburgh