

# 第一原理分子軌道計算DVX 法の オブジェクト指向化と専用計算機の開発

佐々木 徹<sup>a</sup>, 長嶋 雲兵<sup>b\*</sup>, 塚田 捷<sup>c</sup>

<sup>a</sup> (株) ソニー木原研究所, 〒 141-0022 品川区東五反田 1-14-10

兼 (株) アプリオリ・マイクロシステムズ,

〒 223-8522 横浜市港北区日吉 3-14-1 慶應義塾先端科学技術研究センター 14-605

<sup>b</sup> 産業技術融合領域研究所, 〒 305-8562 つくば市東 1-1-4

<sup>c</sup> 東京大学大学院理学系研究科物理学専攻, 〒 113-8658 文京区本郷 7-3-1

\*e-mail: umpei@nair.go.jp

(Received: April 6, 2000; Accepted for publication: June 20, 2000; Published on Web: October 30, 2000)

DVX 法に基づく第一原理分子軌道計算プログラムをオブジェクト指向 C++ 言語を用いて作成した。さらに計算時間の 90 % 以上の部分を占める行列要素算出部分を並列化し、計算速度の向上を図った。また、Sparc Station10(74MHz) をホストとして TI320C40 プロセッサ 64 台からなるアクセラレータを持つ専用計算機を開発した。

Si<sub>78</sub>B<sub>6</sub>H<sub>53</sub> クラスターの計算をホストのみで実行した場合に比べ、アクセラレータを用いた並列分散処理では 40 倍程度の高速化が実現された。特に並列処理を行った行列要素算出部分に関しては 100 倍以上の性能向上となり、スーパーリニアスピードアップを観測した。

キーワード: the First Principles DVX $\alpha$  Calculation, Special Purpose Computer, Object-Oriented Programming, C++, Super-linear Speed up

## 1 はじめに

非経験的分子軌道法は、さまざまな機能性分子の設計や開発に対して最も基本的でかつ重要な手法である。しかし、その計算量はハートリーフォック計算の場合でも、用いる基底関数の 4 乗に比例するため、生体内や固体表面での化学反応解析等の大規模系の電子状態計算には膨大な計算コストが必要になる。現実的には、非経験的分子軌道法は、設計したい大きさの大規模分子系ではなく、それを簡素化した小規模モデル分子への適用がせいぜいである。1990 年代前半までは数百基底 (数十原子) の分子軌道計算でさえ、その当時のスーパーコンピュータシステムを必要としていた。そのため、現在でも分子設計に必要な情報を十分な精度で得ることができないでいる。

1990年代後半における計算機の進歩、特にスーパーコンピュータや高性能ワークステーションのような大規模並列計算機の進歩により、ようやく生体分子などの高分子を意識した1,000基底を超える分子軌道計算 [1-5] が行われつつあるが、これらの計算機は高価であり、かつ設備の維持管理も大変であるため、研究者が研究室レベルで「現実を反映した大規模分子系」の分子軌道計算を実現することは容易なことではない。

「現実を反映した大規模分子系」の分子軌道計算を、「低コスト = パーソナルユース」で実現するためには、計算量を軽減するための近似法を取り入れ、さらに計算機の性能を飛躍的に向上させることが必要である。

計算量を軽減するためには、2つの方法が考えられる。一つは、経験的または半経験的分子軌道法のように必要な計算をまともにする代わりに実測値を用いて計算量を削減する方法である。この方法は演算量が大幅に軽減されるにも関わらず、炭化水素などの系では計算される物理量が実測値をよく再現することから、MOPAC[6]のように非常に広範囲に利用されている。有機化学反応におけるフロンティア軌道理論や遷移状態理論などの普及は、経験的や半経験的分子軌道法なしには考えられなかった。しかしながら、経験的または半経験的分子軌道法は、実測が無い系や金属のように周辺的环境によって様々な状態を容易にとる系には利用できないことや、計算結果の信頼性にばらつきがあることがよく知られている。他は、実測値をパラメータとして用いる代わりに、本論文で取り扱うDVX 法 [7] のように、ハミルトニアンに近似を導入することで計算すべき積分自体を簡素化して計算量を軽減する方法である。DVX 法の演算量は、基底関数のほぼ3乗に比例する演算量となる。DVX 法のように近似ハミルトニアンを用いる計算法は、計算量や結果の信頼性などの面から非経験的分子軌道法と半経験的分子軌道法の間際に位置づけられている。この方法は経験的パラメータを陽に含まないことから、第一原理分子軌道計算または第一原理計算と呼ばれている。

また計算機の性能を飛躍的に向上させるための試みは、分子軌道計算法の持つ高い並列性を背景にして、パソコンなどによる並列分散処理システム (PC クラスタ) の構築 [8] や専用計算機の開発等が最近進められてきている。我々のグループも大規模分子の非経験的分子軌道計算専用機として、分散小メモリ型並列アーキテクチャである分子軌道計算専用並列計算機 MOE の開発 [9-12] に現在取り組んでいる。しかしながら、その様な専用計算機を用いたとしても生体分子や物質材料等の大規模系の分子軌道計算を行うことは依然として難しい。

そこで、本研究では第一原理分子軌道計算法の一つである DVX 法を用いて計算量を軽減し、さらに専用分散並列計算機を開発して計算速度の向上をはかることで、「現実を反映した大規模分子系」の分子軌道計算を「低コスト = パーソナルユース」で実現するシステムの開発を行ったので報告する。

## 2 オブジェクト指向言語 C++ による DVX 法プログラムの開発

第一原理分子軌道計算法の一つである DVX 法 [7] は計算精度の面ではやや不満があるものの、塚田らの STM シミュレータ [13-17] に使用されるなど物質材料の第一原理計算を行う上では手頃な手法であり、その物質材料設計に対する有用性は大きい。また、手法そのものが高い並列性を持っているため、PC クラスタなどでも容易に並列化が可能である。そこで DVX プログラムをオブジェクト指向 C++ 言語を用いて作成し、汎用プロセッサと並列アクセラレータ

を具備した分散並列専用計算機上に移植し高速化を図った。

## Class Diagram of DVXa Program

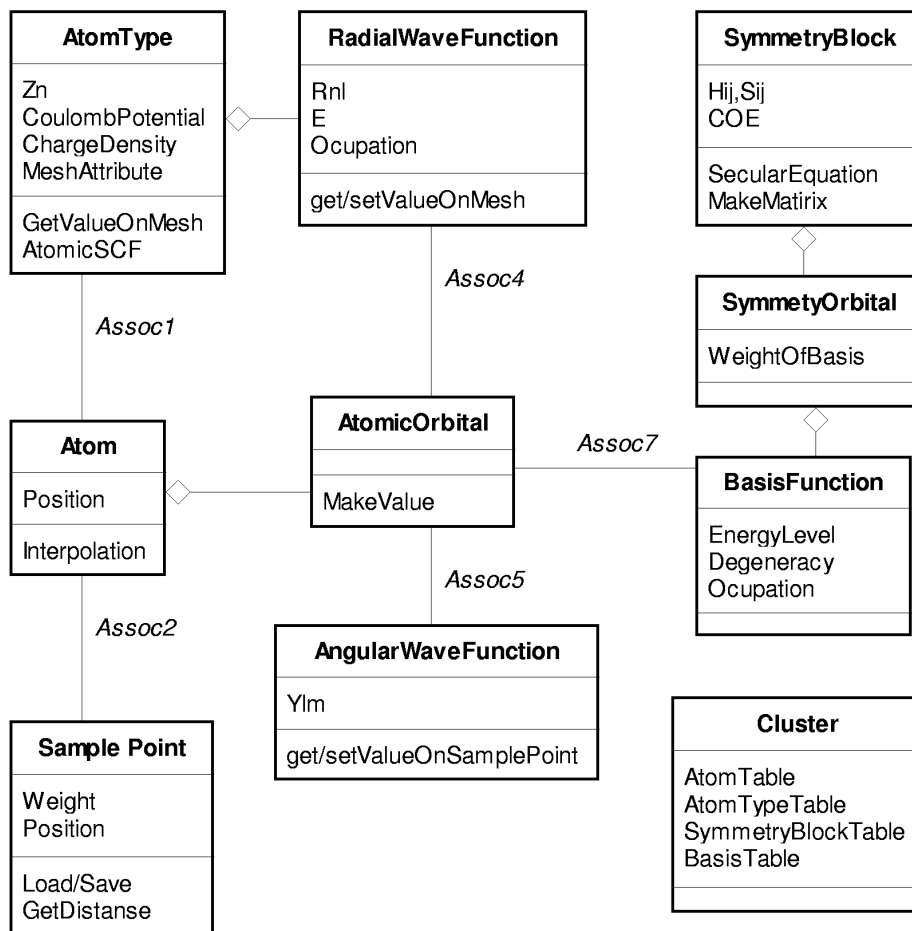


Figure 1. Class Diagram of DVX $\alpha$  Program

塚田らが主宰するCMSフォーラム [18] で公開されている Fortran 版の DVX プログラム [19] を詳しく分析してみると配列データ群が INDEX 値により関連づけられており、データ構造全体がオブジェクトとして定義されている、すなわち明示的ではないがオブジェクト化されていることがわかった。そこでそのデータ構造を参考にして、プログラミング言語にオブジェクト指向言語 C++ を用いて、オブジェクト指向プログラミングを行った。

このようなオブジェクト指向プログラム言語 C++ を用いた分子軌道計算プログラムの開発の例は世界的にみてもまだ少なく、日本では佐藤らの密度汎関数法による分子軌道計算プログラムのオブジェクト指向化 [20] があるのみである。

Figure 1. にクラス構成図を示した。なお、図中、Atom Type は点群対称の位置にある原子をまとめたものであり、Symmetry Group は同一の点群対称を持つ基底関数をまとめたものである。そのため部分行列も Symmetry Group に含めている。

オブジェクト指向化の利点としては、プログラムを小さな部分(オブジェクト)に分割しそれを組み合わせて実行をおこなう構造になるため、プログラム全体の見通しが良くなることである。これによりプログラミング効率だけでなくドキュメンテーションの問題もかなり改善できることがわかった。さらにこのようにすることで、計算科学と計算機工学の研究者間での意志の疎通が図りやすいという利点がある。また、計算負荷の大きい部分を並列計算機システムに移植するには単にオブジェクトを分割して実装すれば良いので、並列化の指針が得られやすいという利点もある。

### 3 DVX プログラムの並列化

DVX 法では、一電子波動関数を  $\phi_i = \sum_k C_{ik} \chi_k$  のように LCAO 型で表現する。ここで  $C_{ik}$  と  $\chi_k$  は、それぞれ基底関数と変分パラメータである。解くべき一電子方程式は以下ようになる。このときの基底関数としては原子軌道の数値解が用いられる。また、X ポテンシャル ( $V_{xa}$ ) のなかに電子密度の 3 乗根を含むため、行列要素は 3 次元の数値積分法を用いて計算する。行列表示をすると以下ようになる。

$$h_{xa} C_i = \varepsilon_i^{xa} S C_i$$

$$h_{kl}^{xa} = \int \chi_k^*(1) \left\{ -\frac{1}{2} \Delta_1 - \sum_l \frac{Z_l}{r_{1l}} + V_c(1) + V_{xa}(1) \right\} \chi_l(1) d\tau_1$$

$$h_{kl}^{xa} = \sum_{s=1}^N \omega(r_s) \chi_k^*(r_s) h^{xa} \chi_l^*(r_s)$$

$$S_{kl} = \sum_{s=1}^N \omega(r_s) \chi_k^*(r_s) \chi_l^*(r_s)$$

ここで、 $C_i$  と  $S$  はそれぞれ、係数ベクトルと重なり積分である。 $\omega(r_s)$  は、乱数で生成したサンプル点  $r_s$  上の体積要素の重みである。サンプル点の数は、通常基底関数の数の 100 倍以上となる。この行列要素の計算が、DVX 法において 90% 以上の計算時間を必要とする部分である。

DVX 法は  $h_{ij}$  と  $S_{ij}$  をサンプルポイント  $r_s$  上の各点に対する重み付きの累積加算として求めるので、行列要素生成部分の並列化は、並列化に際してサンプルポイントに関するループ ( $s=1, N$ ) をプロセッサ数個の部分に分割し、その部分和の実行をプロセッサに振り分け、最後に収集するだけでよい。

$$h_{kl}^{xa} = \sum_{s=1}^{N1} \omega(r_s) \chi_k^*(r_s) h^{xa} \chi_l^*(r_s) + \dots + \sum_{s=1}^{Nm} \omega(r_s) \chi_k^*(r_s) h^{xa} \chi_l^*(r_s)$$

$$S_{kl} = \sum_{s=1}^{N1} \omega(r_s) \chi_k^*(r_s) \chi_l^*(r_s) + \dots + \sum_{s=1}^{Nm} \omega(r_s) \chi_k^*(r_s) \chi_l^*(r_s)$$

$$(N_1 + N_2 + \dots + N_m = N)$$

ただし、原子軌道に関するデータなどを算出するのに必要なデータは、各プロセッサ上に常に置いておく必要がある。

並列処理手順の概略は以下になる。

1. SCF 反復の中で変化しない角度成分や重みは計算開始時にプロセッサに転送する。
2. SCF 反復の中で変化するポテンシャルや動径波動関数はホスト上で計算し、SCF 反復ごとに各プロセッサに送られる。
3. 各プロセッサ上で  $h_{ij}$ 、 $S_{ij}$  の部分和を求め、( 例えば ) 0 番目のプロセッサで総和を取り、積分された  $h_{ij}$ 、 $S_{ij}$  をホストに転送する。このときホストとのデータ転送量は、基底関数の数の 2 乗のオーダーとなる。
4. ホスト上で固有値計算を行い、収束判定を行う。収束していれば計算終了。収束していなければ、電子の再分配を行い、2 に戻る。

一般に並列計算機上のソフトウェア開発は複数のプロセッサが同時に動作するため、逐次計算機上にソフトウェアを開発する場合と比較してデバッグが難しい。そのため、まずホスト CPU のみ使用するバージョンを作成し、その後、分散並列専用計算機に順次移植するという開発ステップでソフトウェアを開発した。これはホスト CPU のみを使用した逐次版と分散並列専用計算機を使用した並列化版との結果が比較でき、また対ワークステーションでの高速化率が簡単に測定できる。

DVX 法の計算ステップをすべて並列化するのは得策ではない。例えば並列処理が向かない部分に並列処理を用いてしまうと計算速度があまり向上しないにも関わらず、かえってソフトウェア開発の難易度が高くなるということになってしまう。

そこで、次項に示すように単一プロセッサでは計算時間の 97% 以上が費やされる行列要素の生成部分を並列化した。ただし、データの初期化や行列の固有値計算等といった並列処理してもあまり効果のないものはホストコンピュータを使用した。

## 4 DVX 専用分散並列計算機

新たに作成した専用計算機のハードウェア構成を Figure 2 に示す。このシステムは、ネットワーク接続や I/O その他の仕事をするホストボード : Sparc Station10(75MHz) と 16 枚のアクセラレータボードが VME-bus で結合されたものであり、それぞれのアクセラレータボード上では 16MB から 64MB のローカルメモリを持つ DSP の TI320C40 プロセッサが 4 つ実装されている。合計 64 プロセッサの並列専用計算機である。ボード間のネットワークポロジはハイパーリンクである。

ホストとアクセラレータの役割分担に関しては、先に述べたようにアクセラレータは  $h_{ij}$  と  $S_{ij}$  の算出のみを行い、原子軌道の算出や、行列計算の固有値計算、収束判定、電子の再分配はホストが行っている。Figure 3 にボードの構成を示した。

ベンチマークテストとして P 型半導体の表面に H 原子が吸着したものをモデル化した 135 原子からなる  $\text{Si}_{78}\text{B}_6\text{H}_{53}$  クラスタを用いて計算時間を測定した。

ホストコンピュータとして使用している Sparc Station10(75MHz) だけで計算した場合と 64 プ

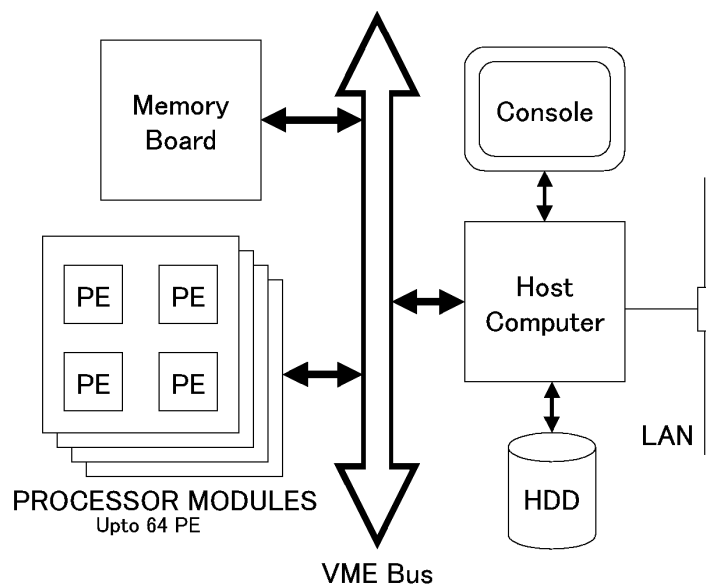


Figure 2. System Configuration of Special Purpose Computer for DVX

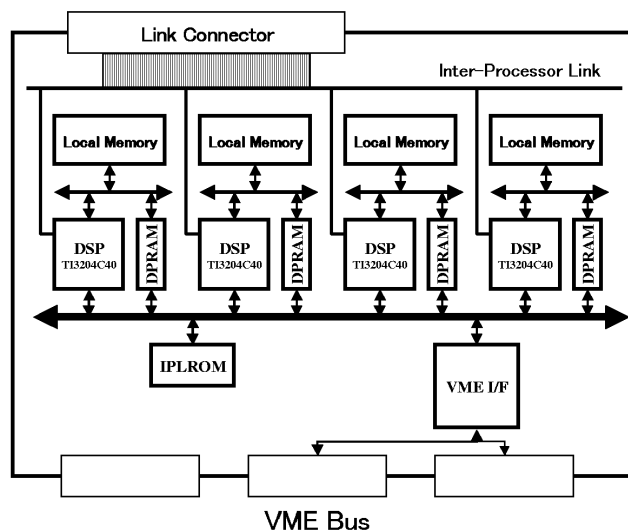


Figure 3. Board Configuration of Special Purpose Computer for DVX

ロセッサからなる分散並列専用計算機を使用した場合の SCF 反復 1 回の計算時間を Table 1. に示す。

Table 1. Wall-Clock Time(sec.)

## 演算時間の内訳

単位: 秒

SPARC単体	ポイント数				
	2048	4096	8192	16382	32762
球対称ポテンシャル生成	4	4	4	5	4
行列要素算出	509	1,019	2,067	4,104	8,196
行列固有値計算	178	173	174	175	176
電子再分配・収束判定	36	35	34	35	34
合計	727	1,231	2,279	4,319	8,410

SPARC+アクセラレータ	ポイント数				
	2048	4096	8192	16382	32762
球対称ポテンシャル生成	5	4	4	5	6
行列要素算出	8	13	23	42	81
行列固有値計算	175	175	174	177	174
電子再分配・収束判定	34	39	34	35	35
合計	222	231	235	259	296



はアクセラレータにより加速している。

逐次計算ではサンプル点の増加に従い行列要素算出のステップの比率が大きくなっていることが分かる。計算結果に対して 5 桁程度の精度を出すためには 1 原子につきサンプル点として 300 点程度必要であるから、表中の 32,768 点が現実的な計算規模である。32,768 点の計算を Sparc Station 単体で実行した場合には、処理時間の 97% 以上が  $h_{ij}$ 、 $S_{ij}$  の生成に費やされている。それに対し、分散並列専用計算機を使用すると、並列化効率が高いため、 $h_{ij}$ 、 $S_{ij}$  生成の処理時間が著しく減少し、対角化のコストが相対的に大きくなってきている。

行列要素算出のステップだけみると分散並列専用計算機により 100 倍以上の高速化が図られており、理論性能限界の 64 倍を超えたスーパーリニアスピードアップが観測されている。これは、並列化によってプロセッサ一台あたりの問題サイズが小さくなり、キャッシュミスなどのメモリアーキテクチャに起因するボトルネックが解消されたためである。

## 5 まとめ

DVX 法をオブジェクト指向プログラミング言語 C++ を用いて開発した。プログラムをオブジェクト指向化することにより、プログラミング効率だけでなくドキュメンテーションの問題もかなり改善できることがわかった。

また、DVX 法の持つ並列性を活かし、分散並列専用計算機で高速化することができた。DVX 法の並列処理は特にサンプルポイントの数が大きくなるほど効果が大きく、計算全体では 64 プロセッサで 40 倍程度の性能向上をみた。並列処理を行った行列要素算出のステップに限れば、100 倍以上の高速化を実現し、スーパーニアスピードアップを観測した。

これをさらに発展させて、計算負荷の大きいオブジェクトのメソッドを抽出し、それを専用 LSI 化することによって高速化する試みに着手した。基盤となるシステムを共通化することにより様々な分野の専用 LSI にも適合できるようなシステム構成を目指している。

本研究の一部は科学技術振興事業団平成 8 年度補正予算と平成 9 年度予算による独創的研究成果等育成事業「走査トンネル顕微鏡シミュレータの開発」および科学技術振興調整費 総合研究「科学技術計算専用ロジック組込み型プラットフォーム・アーキテクチャに関する研究」によるものである。また、研究環境を提供して頂いた慶應義塾大学先端科学技術研究センター、ご支援を頂いたソニー木原研究所会長木原信敏博士および福田安志氏、有用な御議論をいただいた MOE 研究会、日本大学の里子允敏先生を始めとする CMS フォーラムのみなさまに深く感謝する。

## 参考文献

- [1] Alsenoy, C.V., Yu, C.H., Peeters, A., Martin, J.M.L., and Schafer, L., *J. Phys. Chem.*, **102**, 2246-2251 (1998).
- [2] Challacombe, M., and Schwegler, E., *J. Chem. Phys.*, **106**, 5526-5536 (1997).
- [3] Mattson, T. G., *Parallel computing in computational chemistry*, ACS Symp. Series (1995).
- [4] McWeeny, R., *Method of molecular quantum mechanics*, 2nd edition, Academic (1989).
- [5] Sakuma, T., Kashiwagi, H., Takada, T., and Nakamura, H., *Int. J. Quantum Chem.*, **61**, 137-152 (1997).
- [6] Deware, M. J. S., Zoebitsch, E. G., Healy, E. F., and Stewart, J. J. P., *J. Am. Chem. Soc.*, **107**, 3902 (1985).  
Stewart, J. J. P., *MOPAC93.00*, Fujitsu Ltd., Tokyo, Japan (1993).  
Available from Quantum Chemistry Program Exchange, University of Indiana, Bloomington, IN, USA
- [7] 例えば、里子允敏、大西梢平、密度汎関数法とその応用、分子・クラスターの電子状態、講談社サイエンティフィック、(1994).



- [8] 分子軌道計算関係では、サイエンティストパラダイツ社 (<http://www.spara.co.jp>) やベストシステムズ社 (<http://www.bestsystems.co.jp>) 等が、分子軌道計算プログラムを並列化し、それを搭載した PC クラスタを製造販売している。
- [9] 村上和彰, 小原 繁, 長嶋雲兵, 網崎孝志, 田辺和俊, 北尾 修, 北村一泰, 高島 一, 宮川宣明, 稲畑深二郎, 山田 想, *CICSJ Bulletin*, **16**, 6-12 (1998).
- [10] 小原 繁, 村上和彰, 長嶋雲兵, 網崎孝志, 田辺和俊, 北尾 修, 北村一泰, 高島 一, 宮川宣明, 稲畑深二郎, 山田 想, *CICSJ Bulletin*, **16**, 2-5 (1998).
- [11] 稲畑深二郎, 山田 想, 大澤 拓, 沖野晃一, 富田裕人, 橋本浩二, 早川 潔, 宮川宣明, 村上和彰, 電子情報通信学会技術報告, **ICD98-21**, 77-84 (1998).
- [12] [http://kasuga.csce.kyushu-u.ac.jp/~moe/index\\_j.html](http://kasuga.csce.kyushu-u.ac.jp/~moe/index_j.html)
- [13] Watanabe, S., Aono, M., and Tsukada, M., *J. Vac. Sci. Technol. B*, **12**, 2167 (1994).
- [14] Watanabe, S., Aono, M., and Tsukada, M., *Phys. Rev. B*, **44**, 8330 (1991).
- [15] Uchiyama, T. and Tsukada, M., *J. Vac. Sci. Technol. B*, **12**, 2205 (1994).
- [16] Uchiyama, T. and Tsukada, M., *Surface Science*, **313**, 17 (1994).
- [17] Kobayashi, K. and Tsukada, M., *Phys. Rev. B*, **49**, 7660 (1994).
- [18] CMS フォーラム : <http://www-cms.phys.s.u-tokyo.ac.jp/~cms/>
- [19] Adachi, H., Tsukada, M., and Satoko, C., *J. Phys. Soc. Jpn*, **45**, 875 (1978).
- [20] Sato, F., Shigemitsu, Y., Okazaki, I., Yahiro, S., Fukue, M., Kozuru, S., and Kashiwagi, H., *Int. J. Quantum Chem.*, **63**, 245-256 (1997).

# Development of an Object-Oriented Program and a Special Purpose Parallel Computer for First Principles DVX $\alpha$ Calculations

Tohru SASAKI<sup>a</sup>, Umpei NAGASHIMA<sup>b\*</sup> and Masaru TSUKADA<sup>c</sup>

<sup>a</sup>Sony Kihara Research Center Inc.

1-10-4 Higashi-Gotanda, Shinagawa-ku, Tokyo 141-0022, Japan

Also at A-Priori Microsystems, Inc.

14-605, Keio Leading-edge Laboratory of Science and Technology,

3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan

<sup>b</sup>Computational Science, National Institute for Advanced Interdisciplinary Research

1-1-4 Higashi, Tsukuba, Ibaraki 305-8562, Japan

<sup>c</sup>Department of Physics, Graduate School of Science, University of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8658, Japan

*\*e-mail: umpei@nair.go.jp*

We have developed a program for first principles DVX calculations using an object-oriented programming language C++. Based on the program we have also developed a special purpose parallel computer for DVX calculations, which realizes super high performance and cost-down of large scale DVX $\alpha$  calculation.

The special purpose computer consists one host-board (Sparc Station10/74MHz) and 16 acceleration-boards. 4 TI320C40 processors are equipped on a board. The acceleration-board system has 64 processors connected to each other by VME-bus.

The matrix elements generation step where more than 90% calculation time spends was parallelized on the system. In the case of Si<sub>78</sub>B<sub>6</sub>H<sub>53</sub>cluster, the special purpose computer is almost 40 times faster than that by a lone Sparc Station10 processor. Speed up of the matrix elements generation step is more than 100 times though 64 times is the theoretical maximum. Thus, super-linear speed up has been observed.

**Keywords:** the First Principles DVX $\alpha$  Calculation, Special Purpose Computer, Object-Oriented Programming, C++, Super-linear Speed up