



Title	スーパーコンピュータと原子核殻模型計算
Author(s)	越桐, 国雄
Citation	大阪大学大型計算機センターニュース. 1987, 66, p. 63-73
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/65746">https://hdl.handle.net/11094/65746</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

# スーパーコンピュータと原子核殻模型計算

大阪教育大学 越 桐 国 雄

## 1. はじめに

このほど、大阪大学大型計算機センターの昭和61年度開発研究計画に採択された「SX-1向け原子核殻模型コードの開発」が終了したので、その結果を報告する。

現在普通に考えられている原子核の描像は、核子（陽子と中性子の総称）がA個（Aは原子核の質量数で2～200程度）集合した多体系で、非相対論的な量子力学によってその構造や運動が支配されているというものである。これは、核子を電子に置き換えて考えれば、原子や分子と同等のものである。しかしその大きな違いは、原子分子の場合は電子間の相互作用が電磁相互作用（クーロン力）であり、その性質がよく解っている簡単なものであるのに対し、核子間の力は「強い相互作用」と呼ばれ、状態に依存する複雑な構造を持ち、また短距離では非常に強い斥力となるため、その性質も取扱い方も十分に理解されたとはいえないことにある。（核力の長距離部分に関しては、 $\pi$ 中間子によって媒介されるとする、湯川理論が現在も出発点になっている。）

さて、原子核はこのように強い核力によって結合された高密度な物質であるが、基底状態近傍ではパウリ原理のため、核子が散乱してその状態を変えるような運動が抑制され、独立粒子描像がよく成立することが知られている。つまり核子が互いに作り出す平均場（Hartree-Fock場）の中に一定の軌道が存在し、そこに核子が順番に詰まっていくことによって、魔法数と呼ばれる特定の核子数の原子核が非常に安定になるなど、原子核の様々な性質が説明されることが解っている。これが原子核の殻模型<sup>1)-3)</sup>と呼ばれるもので、原子における軌道のイメージと同じ様なものを想像してもらえば良い。原子核の殻模型が提案されてから40年近くたつが、現在でも原子核モデルの一つの柱として重要な役割を果たしている。

ところで最近、原子核の描像として核子の自由度を考えるだけでよいのか、中間子や核子の励起状態等（あるいは更に基本的なクォーク、グルーオンの自由度）をあらわに取り入れる必要があるのではないか、という問題提起があるが、これらの問題を実験的に検証しようとするとき、従来の理論の枠組みでどこまで現象を説明できているのかということをおさえておく必要が生じる場合がある。つまり、従来の理論の精密化（例えばより大きな配位空間で殻模型の計算等）が必要となる。一方、原子核は素粒子反応のためのMicro Laboratoryとしての役割も果たし、例えばニュートリノの質量を原子核の2重ベータ崩壊から決めようとする、原子核波動関数に対するかなり大きな計算が必要になってくる。

このように最近、原子核殻模型計算の大型化が要求されてきたが<sup>3)</sup>、必ずしも分子科学における

分子軌道法の計算のように組織化された大規模計算が次々行われるような状況にはなっていない。これは先ほども述べたように、基本となる相互作用や理論の枠組み(核力として2体力のみを考え、核子の運動を非相対論的に扱う近似)が完全には確立しておらず、まだ十分な予言能力を持っているとはいえないためである。

我々は原子核の弱い相互作用や電磁相互作用における、交換電流(中間子自由度)やコア偏極(配位空間の拡張)について研究してきたが<sup>4), 5)</sup>、その計算を軽い核からやや重めの核へ拡張する必要があった<sup>6)</sup>。このためには大規模の原子核殻模型計算が必要で、従来の計算の高速化が必須となった。そこで従来使用してきた3軌道の原子核殻模型コードをSX向けに改良することによって高速化を計ると共に、原子核殻模型計算を進める上での目安(どの程度の規模の計算が比較的簡単に行えるのか)を調べ、また既存のプログラムをベクトル化するには、どのような点に注意すればよいのか<sup>7)</sup>等について検討した。

原子核殻模型計算における基底関数の次元は、粒子の運動する軌道数と粒子数によって定まるが、

表1 基底関数の次元 (J = 2, T = 1 の場合)

軌道数	粒子数	6	12	18
2		14	/	/
3		525	5,768	/
4		6,338	3,067,518	42,128,974

軌道数が増えると共に爆発的に増大する。第1表にいくつかの例を挙げたが、4軌道でもっとも大きいサイズの計算(核子の数が20個)では基底関数の数が6千万を超え(即ち、6千万次元のハミルトニアン行列が必要になる)近い将来にも計算が可能な見通しはほとんどなく、この様な場合には、別の原子核模型から出発するなどの必要がある。現在行なわれている最も規模の大きい計算は、スピナーアイソスピン形式のCFPによる基底関数を用いた場合に換算して、数千～数万次元である。またこの基底でのハミルトニアンは比較的密な行列になっていて、全行列要素に対する0でない行列要素の数の割合は、数千次元のハミルトニアンでも30～40%前後に達している。

ここではスーパーコンピュータの特徴<sup>8), 9)</sup>を生かすためにすべての計算を主記憶上で行い、外部記憶装置とのデータ交換を必要としない程度の計算をベクトル化することを目標にした。具体的には<sup>29</sup>Si(及び<sup>31</sup>P)というsd殻の中央のスピン、アイソスピンが1/2の核を対象とした。この場合核子の全角運動量が1/2、3/2、5/2の3軌道に11(9)個の核子(正確には核子の空孔)が運動している場合に相当し、ハミルトニアン次元は2360(1434)次元となる。次節のプログラムのテストの各数値は核子7個の517次元の状態調べたものである。

## 2. SX-1 向き改良と計算結果

計算は大きくわけて 4 つの部分に分かれている。

- (1) CFP を求める部分
- (2) ハミルトニアンを求める部分
- (3) ハミルトニアンを対角化する部分
- (4) 1 体、2 体の密度行列を計算する部分

以下、この順に沿ってプログラムの変更点を述べていく。なおプログラムの大まかな構成と、最終のプログラムをアナライザ<sup>10), 11)</sup> で解析したサマリーレポートを付録に示す。

### (1) CFP の計算

CFP (Coefficients of Fractional Parentage) は、反対称化された  $N$  体系の波動関数を  $N-1$  (または  $N-2$ ) 体系の反対称化された波動関数で展開した時の係数であり、以下の計算の前に 1 度だけこの係数の組 (今の場合 1 軌道 CFP を考えており、約数万個ある) を求めておけばよく、とくに高速化の障害になるものではない。

最初に SX-1 を走らせてベクトル化率を調べたのが、このモジュールだったが、ベクトル化率は 2 % くらいで驚いてしまった。計算は  $N$  が小さいものから順次漸化式によって求めて行くため、ベクトル化には向いていない。そこで途中に現れる幾何学的係数 (Racah 係数) を計算する関数の部分を DO ループの外に出して、あらかじめ計算し 1 次元配列にストアするようにした。この結果、Racah 係数の計算回数は 30 万回から約 1/10 に減り、最深ループのベクトル化が可能となり (ベクトル率 42 %)、計算時間は 45 秒 (ACOS1000) → 1 秒 (SX-1 : AP) に短縮された。ここでは 1 軌道に対する必要な CFP をすべて求めてファイルにストアし、以下の計算ではこのファイルを初めに 1 回読みだして参照するため、計算時間には含まれない。

原子核の計算において、空間回転に関する幾何学的な係数として現れる Racah 係数や  $9j$  係数と呼ばれるものや、CFP 等は数個 ~ 10 個程度の引数をもった関数となり、全てをストアする事は困難である。またこれらの引数の多い関数の複雑な積和が必要なため、ループの多重度も必然的に高くなりベクトル化を妨げることが多い。

### (2) ハミルトニアンを求める部分

コンピュータを用いた計算において解くべき問題は、ある行列を求めこれを係数とする連立方程式を解いたり、あるいはその行列の固有値を求める問題に帰着できることが多い<sup>8), 9)</sup>。その場合、実行時間の大部分が行列を求める部分に費やされることもしばしばである。原子核の殻模型の計算も多体系のハミルトニアンを表す実対称行列を求め、この固有値および固有ベクトルを固有値の小さいものから数本求めることに帰着され、ハミルトニアンの行列要素の値を求めるために多くの時

間が費やされる。

ハミルトニアンを求めるためにはまずN粒子系の反対称な基底関数の組を求め、つぎに核子間に働く相互作用ポテンシャルの、各基底関数間での行列要素を計算すれば良い。N粒子系の基底関数を、CFPを用いてN-2粒子系の基底関数の組で展開すれば、この計算が簡単に実行できる。この時上述のポテンシャルの2核子間の行列要素が必要となるが、これを第1原理から計算するには、かなり手間が掛かるため、普通、エネルギースペクトルに対する原子核殻模型計算と実験値の比較から最小自乗法によって求められた、経験的有効相互作用を用いることが多い。ここでも、現在最も精度がよいとされている、ブラウンとビルデンタールの有効相互作用<sup>12)</sup>を用いた。これは数十個のデータとして与えられており、計算する必要はない。

つまりハミルトニアンの行列要素は次のような形で与えられるのである。

$$\langle f | H | i \rangle = \sum CFP * CFP * \langle a b | T + V | c d \rangle \quad (1)$$

ここでT、Vはそれぞれ原子核の運動エネルギーとポテンシャルエネルギーを表し、a、b、c、dは核子の存在する軌道を表す。CFPと書いたものは先に述べたCFPそのものではなく、これらを組み合わせて作られる多軌道のCFPで、これらを総て配列として蓄えておくのは困難である。そこでN-2粒子系の状態毎にCFPを計算することになる。状態の指定にスピン、アイソスピンその他の量子数が必要であるため(1)式の和は7重ループとなり、多軌道のCFPの計算に約半分の時間がかかっていた。

そこでまずCFPの計算の中で2軌道分をまとめて計算してストアすることにした。この結果、従来のプログラムより1.5倍高速化されベクトル化率はもとの44%から62%になった(これをVer 2.0と呼ぶ)。しかしこれを詳しく調べてみると、今まで目だたなかった(1)式的最深ループに70%の計算コストが掛かっていることがわかった。またこの中で値が0のCFPについてもそのままループを回しているため余分な時間がかかっている、更に高速化するためには、(1)式的最深ループの回数自身を減らす必要があった。そこでリストベクトルを導入して0でないCFPを圧縮してストアした。これにより最深ループに費やされた時間は15.7秒から1.3秒にまで短縮された。更に多軌道のCFPを計算するサブルーチンでよばれていた関数をサブルーチンの中で展開し整理することによってこの部分が19.0秒から3.9秒になった。このほかに多少改良を加えたVer 3.0はベクトル化率は57%に減ったものVer 2.0より4.5倍高速化された。(最終的には更に若干の改良を加えVer 4.0となっている。)

このように単にベクトル化率が高ければ実行時間が短縮されているということにはなっていない事に注意する必要がある。プログラムを改良していく途中で、ベクトル化率が下っても計算回数を減らすように変更することで計算時間を短縮できる場合が度々あった。

(3) ハミルトニアンを対角化する部分

実対称行列の対角化によって比較的少数の固有値、固有ベクトルを求めるとき、ハウスホルダー法による3重対角化の後、パイセクション法によって固有値を求め、更に逆反復法によって固有ベクトルを求めるといった方法がよく用いられる。センターで利用できるS X用の数値計算ライブラリASLに収録されているプログラムでは、DCSMSSがこれに当たる<sup>13)</sup>。

一方かなり大次元の行列の場合は、ランチョス (LANCZOS) 法によってもとのN次元実対称行列を3重対角化すると、適当なところで3重対角化を打ち切って得られるM次元部分行列の固有値が、Mを増やすと共に元の行列の固有値に収束して行くが、この時 $M \ll N$ で十分であることが知られている<sup>14)</sup>。この方法は特に、行列を列ベクトルに作用させた時の値が、全ての行列要素を求めずに、計算できるときにその威力を発揮し、磁気量子数のみを対角化する基底関数を用いて原子核の殻模型を計算する場合に非常に有効である<sup>15)</sup> (この場合原子核の状態に対する対称性の制限が緩まり、同じ計算でも行列の次元が大変大きくなる)。

ところで、我々のコードではあらかじめ、ハミルトニアンの全ての行列要素を求めてから対角化するのであるが、それでも行列の次元が大きい割に必要な固有値、固有ベクトルの数が少ないので、ランチョス法を用いることによって計算をかなり高速化することができる。表2に実際の計

表2 行列の固有値問題を解く時間 (秒)

次元数	ハウスホルダー	ランチョス	比
5 1 7	1.2 2	0.1 14 (3.3 3)	1 0.7
1 4 3 4	1 6.5 7	0.5 15	3 2.2
2 3 6 0	6 4.3 5	1.1 91	5 4.0

(注) 最小固有値と固有ベクトルを求めたもの。計算は共にS X-1:AP

(カッコ内はCP)による。ランチョス法の5 1 7次元の計算はHFP

(IAP)で0.6 1秒かった。

算時間を比較した例を示すが、ASLライブラリのハウスホルダー法 (DCSMSS)と比べて数十倍に高速化できるのがわかる。このプログラムは従来使っていたものを特に改良せずにそのまま用いたが、HFPのIAPを意識して作っておいたため、ベクトル化率は99.4%に達している。

(4) 密度行列を計算する部分

以上の手続きで、原子核の特定の状態が求るが、系のエネルギー以外の様々な物理量を計算しようとする、この波動関数から必要な物理量を表す1体演算子や2体演算子の行列要素を計算する必要がある。この場合、上の(1)式の右辺でCFPに加えて更に幾何学的な係数が掛かった式の和を

計算する。従ってハミルトニアンを求めたときと、おおむね同じようにアルゴリズムを変更することで高速化が可能である。

以上の結果、改良前のプログラムをSX-1（AP）で実行した場合と、改良後のプログラムをSX-1（AP）で実行した場合のCPU時間が表3のようにまとめられる。全体のベクトル化率

表3      3軌道原子核殻模型プログラムの実行時間（秒）

次元数	改良前	改良後	比
517	25.7	3.4 (18.6)	7.6
1434	345.6	24.7	14.0
2360	1506.4	94.8	15.8
ベクトル化率	44%	75%	/

（注）基底状態のエネルギーと波動関数を求めたもの。計算は共にSX-1：AP（カッコ内はCP）による。改良前の517次元の計算はHFP（IAP）で42秒かかった。

自身は75%とあまり高いわけではないが、プログラムの改良によって従来のプログラムに対して、2000次元程度の行列の場合約15倍に高速化されたことがわかる。更に<sup>29</sup>Si(2360次元)のばあいには磁気能率や電子散乱の形状因子を求めるための密度行列をアイソスカラーとアイソベクターに対して求めると（2体の密度行列まで含めて）約2分強で計算できた。

現在用いられている原子核殻模型のコードの代表的なものとして、ユトレヒト大学のグラウデマンのグループのRITSSCHILという汎用のプログラムがあるが、CYBER205を用いると1000次元の行列の計算に約3分、対角化して固有ベクトルを最小固有値に対応するものから4本求めるのに10秒程度かかるという報告されている<sup>3)</sup>。今回改良したプログラムは3軌道に限られ汎用性に劣り、単純に比較できるものではないが、我々のコードをSX-1で実行すると、RITSSCHILをCYBER205で実行した場合の20倍程度高速に計算できたことになる。

現段階のプログラムで更に高速化するためのネックとなっているのは、TORBI（ハミルトニアン行列要素を求める部分）とSWFP2（3軌道の2粒子CFPを求める部分）である。前者はベクトル化率92%であるが、圧縮してリストベクトルを用いているため、最深DOループの呼び出し回数が多い割に平均ループ長が15と短く、なお全実行時間の30%を占めている。後者では条件判定して配列の0でない要素を圧縮する部分がベクトル化できないため、サブルーチンのベクトル化率は10%程度に留まり、全実行時間の20%程度かかっている。

アナライザーで表示される実行コストの値は高速化のネックを調べる上でそれなりの目安にはな

るが、コストが大きくてベクトル化されている部分とコストは小さいがベクトル化されていない部分を比較してチューニングしようすると、最終的に重要なのは実行時間である。このあたりの情報をもう少し工夫して、得られるようにできないものだろうか。また大阪教育大学のデータステーションでは手順付き端末がサポートされていないため、今回ベクトライザは利用できなかった。

### 3. おわりに

いわゆるスーパーコンピュータが実際にその性能を十分に発揮するのはベクトル化率が90%を超えてからだといわれるが<sup>16)</sup>、問題自身が複雑ではあってもアルゴリズムが単純な構造をもった計算に帰着できる場合にはかなりの部分がベクトル化可能で、スーパーコンピュータの恩恵を十分得ることができる。しかしここで我々が必要としていた計算のように、そのアルゴリズムが必ずしもベクトル化には適していないような問題も少なくない。実際今回のプログラムの最終的なベクトル化率は75%程度に留まった。それにも関わらず、従来のプログラムの高速化という目標はある程度達成することができ、約15倍の高速化が可能になった。その1つの原因は比較的大きな主記憶領域が取れるため、多くの計算結果を配列にストアすることによって、計算回数の重複を減らし、同時に最深ループでの計算を配列の呼び出しによってベクトル化可能とすることができたためである。これらの内のある部分はベクトルプロセッサを用いなくとも高速化できた部分であるが、ベクトル化できるという保証があって初めて安心して高速化のための改良に着手できたという側面もある。またプログラムの高速化という観点からすると、必ずしもベクトル化率のみが目安になるわけではなく、アナライザなどのデバッグツールによって最も実行時間を消費している部分を見つけていくということがポイントになった。

次に今回の経験を通じて感じた点のいくつかを挙げて今後の大型計算機センターに対する要望とさせてもらいたい。

#### (1) スーパーコンピュータをTSSで使いたい。

プログラムの開発の段階でベクトル化について考慮しながら、実行と修正を繰り返していく際に、いちいちバッチ処理を行うのは非常に手間が掛かる。CPではTSSが利用できるが、この時ベクトル化の効果を知ることはできない。従って是非スーパーコンピュータもTSS環境で利用できるようにしてほしい。もし今後もスーパーコンピュータがミッドエンドプロセッサとして使われて、TSSは汎用機だけとするならばできる限り汎用機とスーパーコンピュータを同じ環境で統一的に使えるようにし、またベクトル化による実行時間の高速化を汎用機側で評価できるような工夫をして頂きたい。



(2) ファイルを共有化してほしい

現在、SX-1とACOS 1000のファイルは管理が別になっており、ACOS側のデータをいちいちSX-1側に転送する必要がある、利用に大変不便を感じている。また利用者ライブラリの作成法や管理法もわかりにくい。(できればCPUコストのうち、システム構成上利用者に不便を強いているコンパイルやファイル転送にかかわる費用をただにしてもらえると有難いのですが。)

(3) 主記憶領域を大きくしてほしい

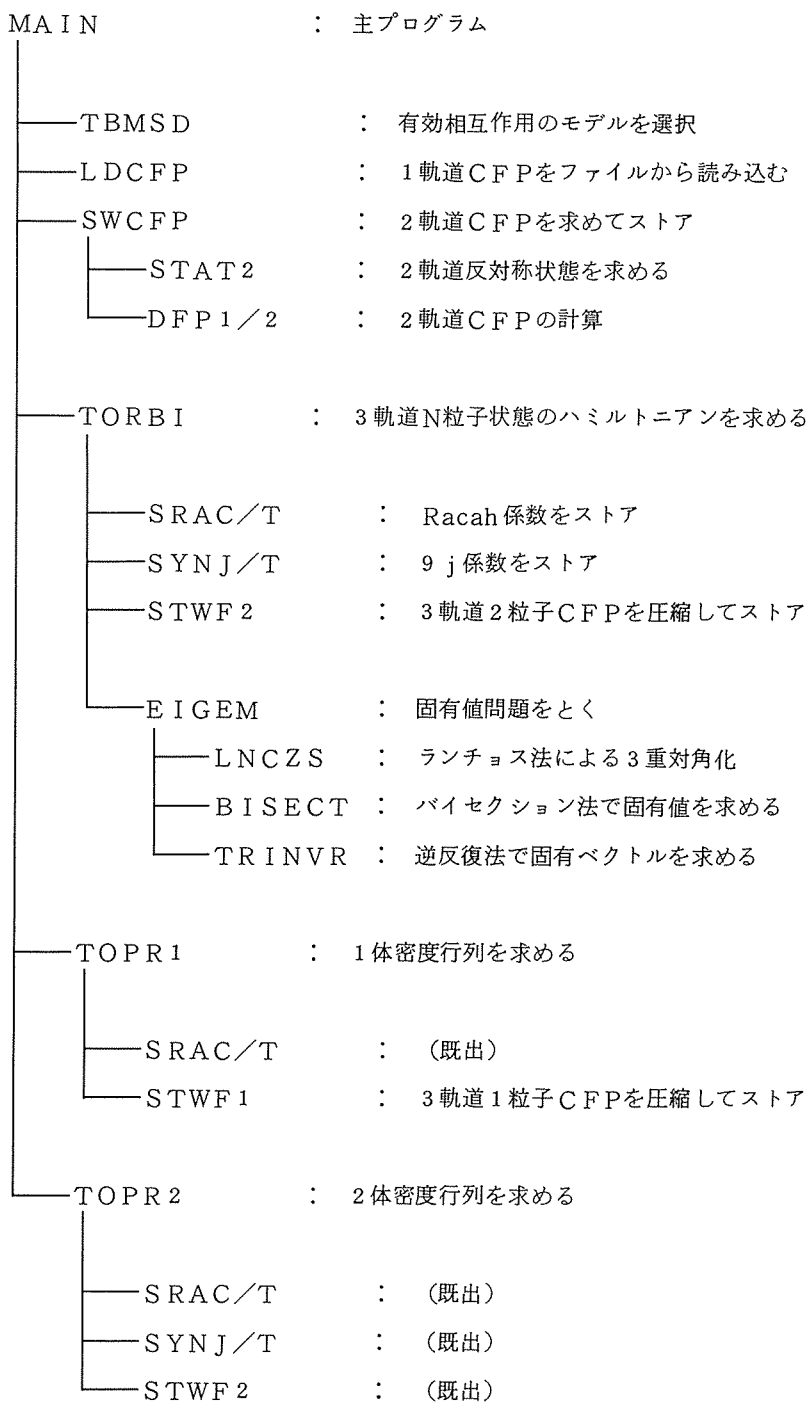
先にも述べたように、高速化のためには同時に多くの変数を配列として主記憶上に準備する必要があった。従ってできるだけ利用者領域が大きく取れるようなシステム構成にしてほしい。今回当初利用者領域が64MBであったので2400次元程度の計算で実対称行列をそのままの形で配列に格納して計算し約54MBのプログラムとなった。現在の最大のジョブクラスWでは、108MBまで可能であるが、もしこれが192MBにまで拡張されると実対称行列を1/2だけ格納することによって、sd殻(3軌道)の全ての計算(行列の最大次元数6707)を主記憶上で計算することができ、現在のコードでも約30分(2体の密度行列迄含めて50分)程度で実行できることになる。

今回の改良によって、もとのプログラムが約15倍に高速化されたが、この結果プログラムの構造は当初のものより複雑になり、プログラムの主要部分は500行から1000行近くに増えて、今後の保守に若干の不安を残した。今後ますます大規模な計算を高速で処理したいという要求が増えてくるだろうが、そのためにはスカラー性能も含めた計算機の高速化や、大容量化など計算機ハードウェアの発展はもちろんのこと、それ以上に利用者がプログラムを作り上げていく上でのソフトウェア環境の一層の発展が望まれる。(スーパーコンピュータ利用における現在の最大のネックはハードウェアの性能やコンパイラの性能以前の、利用者がいかに簡単にプログラムを開発し保守して行けるかというマン・マシンインターフェイス環境にあるといったら言い過ぎであろうか。)

SXを使い初めて十分に時間もなくてまだ使いこなせておらず、また誤った思い込みが多々あると思われるが、皆様のご指導を仰げれば幸いである。

最後にこの間色々お世話になった、大阪大学核物理研究センターの西村道明氏、大阪大学理学部の森田正人教授、大坪久夫助教授はじめ森田研究室の皆さんに厚くお礼を申し上げる次第である。

付録 1 プログラムの概略



[illegible]

## 参 考 文 献

- 1) DeShalit, A., Talmi, I. : "Nuclear Shell Theory", Academic Press, (1964).
- 2) Brussaard, P. J., Glaudemans, P. W. M. : "Shell -Model Applications in Nuclear Spectroscopy", North-Holland, (1977).
- 3) Vallieres, M., Wildenthal, B. H. eds. : "Nuclear Shell Models", World Scientific, (1985).
- 4) Koshigiri, K., et al. : Prog. Theor. Phys. Vol.66, No. 1, 358, (1981).
- 5) Sato, T., et al. : Zeit. Phys. A 320, 507, (1985).
- 6) Nishimura, M., et al. : to be published.
- 7) 片山博、河原久美子、大中幸三郎 : Fortran 77/SXにおける高速化技法 ,  
大阪大学大型計算機センターニュース、Vol. 16, No. 1 (1986).
- 8) 村田健郎、小国力、唐木幸比古 : "スーパーコンピュータ", 丸善, (1985).
- 9) 日本物理学会編 : "スーパーコンピュータ", 培風館, (1985).
- 10) 後藤米子、三原敏敬 : スーパーコンピュータ SX-1 の性能向上支援ツールの利用法、  
大阪大学大型計算機センターニュース、Vol. 16, No. 3 (1986).
- 11) GGB 14-2 ANALYZER/SX説明書、日本電気 (1986).
- 12) Wildenthal, B. H. : Prog. Particle and Nuclear Phys. Vol. 11, Chap. 1,  
Pergamon Press, (1984).
- 13) GYF 12-2 科学技術計算ライブラリ説明書<ASL第2分冊>、日本電気 (1986).
- 14) 森正武、名取亮、鳥居達生 : "岩波講座情報科学 18 数値計算", 岩波書店、(1982).
- 15) Whitehead, R.R., et al. : Advances in Nuclear Phys. Vol. 9, Chap. 2,  
Plenum Press, (1977).
- 16) 渡辺貞、近藤良三、端山伸幸、大中幸三郎、藤井護 : スーパーコンピュータ SX-1 の概要、  
大阪大学大型計算機センターニュース、Vol. 15, No. 4 (1986).