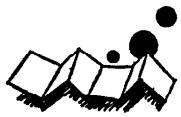
解　説



スーパコンピュータと数学ライブラリ†

二　宮　市　三††

1. はじめに

1976年に，アメリカで世界最初のスーパコンピュータ CRAY-1 が発表されてから，すでに10年が経過した．この間に，アメリカの CRAY，CDC，我が国の日立[1]，富士通[2]，日本電気[3]の諸社が，相次いで新機種を発表し，熾烈な速度競争を展開している．表-1に，現在稼働中の各種主要スーパコンピュータの速度性能を掲げる．

このようなスーパコンピュータの出現と発展にともない，従来からの汎用機のスピードの頭打ち現象のために，その捌け口を見失っていた，多くの研究分野での計算需要が，堰を切ったようにスーパコンピュータに向って奔り始めた．そして，原子力，宇宙開発，気象，航空，建設，分子化学，核物理，生物，計量経済などの巨大計算を必須とする諸分野では，すでに，汎用機時代とは質の異なる新しい成果が着々として，蓄積されつつある．しかしながら，一方では，スーパコンピュータの利用法についての認識が十分でなく，その性能が生かし切れていないというのも，偽らざる現状である．

表-1　スーパコンピュータの最大性能

メーカ	機種	最大性能	発表年
クレイ・リサーチ社	CRAY-1	80 MFLOPS	1976
	CRAY-XMP	400 MFLOPS	1983
	CRAY-2	1,950 MFLOPS	1985
CDC社	Cyber 205	400 MFLOPS	1981
日立	S-810/10	315 MFLOPS	1983
	S-810/20	630 MFLOPS	1983
富士通	VP-100	250 MFLOPS	1983
	VP-200	500 MFLOPS	1983
	VP-400	1,140 MFLOPS	1985
日本電気	SX-1	570 MFLOPS	1985
	SX-2	1,300 MFLOPS	1985

† Supercomputer and Mathematical Library by Ichizo NINOMIYA (Department of Business Administration and Information Science, Chubu University).

†† 中部大学経営情報学部経営情報学科

以上の見地から，今回本小特集が企画されたことは，まことに時宜を得たものである．筆者は，スーパコンピュータの専門家ではないが，数学ソフトウェア・ライブラリに関連する，ささやかな経験と実績を頼りに，その責めを果たしたいと考えるものである．

2. スーパコンピュータとその特徴

スーパコンピュータ (Supercomputer) という名前は，字義通り解釈すれば，普通の計算機を超越した計算機ということになるから，論理的な推論をそのまま行わせようとする第5世代計算機や，多数の計算機を空間的に配置して，文字どおりの並列計算を目指す，パラレルコンピュータも，スーパコンピュータと言えないことはない．しかしながら，世上ではスーパコンピュータといえば，別名ベクトルプロセッサまたはアレイプロセッサとも呼ばれ，いわゆるパイプライン制御方式によって，汎用機の数十倍以上の高速化を目指す，数値計算専用機を意味する．

それでは，スーパコンピュータと汎用機とではどこが違うか？　一般に，計算機には，置数 (LOAD)，格納 (STORE)，加減算，乗除算などの命令がある．これらの命令は，その種類によって多少の相違があるが，少なくとも4個程度の基本ステージから成り立っている．汎用機では，これらの命令はおのおの一体として，逐次的に実行される．すなわち，一つの命令の完了を待って次の命令が発動され，それが完了するのを待って，さらに次の命令が発動されるという具合である．ところが，スーパコンピュータでは，各命令の基本ステージが独立の単位として扱われる．スーパコンピュータ内のベクトルユニットの中には，各命令専門のパイプラインがあり，この中ではそれぞれの命令の各ステージが逐次的に実行される．今 $C(I)=A(I)+B(I)$, $I=1,2,\cdots,N$ という計算が行われるところを想像しよう．二つのベクトルの成分 $A(I)$ と $B(I)$ は順次それぞれのベクトルレジスタから供給され，加算パイプライン内をとおって和 $C(I)$ となり，別のベクトル

レジスタに運ばれていく．$A(1)$と$B(1)$が，パイプラインに入り，その第1ステージを通過して，第2ステージに入るや否や，第2のデータ対$A(2)$と$B(2)$が第1ステージに入る．さらに1ピッチ（パイプライン1ステージ処理時間）経過すると，$A(1)$と$B(1)$は第3ステージに，$A(2)$と$B(2)$は第2ステージに，そして$A(3)$と$B(3)$とが新たに第1ステージに入ってくる．このように，データ対は相連なって，1ピッチごとにパイプライン中を移動し，すべてのステージを完了したものは，和となって順次所定のベクトルレジスタに送られる．したがって，すべてのステージが働き始めるまでの過渡的な時間—立ち上り—を無視すると，1ピッチ（マシンサイクルとも言う）当り一つの演算が行われるという勘定になる．スーパコンピュータの最大性能というのは，このような意味での計算速度を，1秒当り百万回の浮動小数点演算を意味するMFLOP単位で表したものである．いわば，瞬間風速というような，理想状態の速度であって，一応の目安にはなるが，常時そんなスピードを出せるものではない．

スーパコンピュータの最高速度が発揮できない理由の一つは，立ち上り時間が無視できないことである．特にベクトルの長さが短いときには，その影響が大きい．しかし，スーパコンピュータの実効速度を左右するより大きな要因は，ベクトル化率である．一つの数値計算ジョブには，スーパコンピュータ向きの，ベクトルデータに対する一様な計算処理のほかに，スカラデータに対する種々の雑多な計算や，入出力処理も含まれている．全体の計算量の中で，スーパコンピュータのベクトル処理が有効に利用できる部分の割合をベクトル化率という．今，ベクトル化率をV，スカラ処理に対するベクトル処理の速度倍率をαとすると，計算全体に対するスーパコンピュータのスカラコンピュータに対する速度倍率pは

$$p=\frac{1}{1-V+V/\alpha}$$

で与えられる．この式から，pを大きくするには，Vをできるだけ大きくすることが大切であることがわかる．大まかに言えば，Vは75%以上でないと大した効果はあげられない．たとえばV=50%では，$\alpha=\infty$でも，p=2となるにすぎない．

要するに，スーパコンピュータはハードウェアとしては，極めて高い性能を有するものであるが，その反面，その性能を十分に発揮させるためには，ソフトウェアの側にも高度の技術が要求される，かなりの難物ということができる．

3. 基本的なプログラミング技法

今まで汎用機で行ってきた計算を，単にスーパコンピュータで行いたいというのであれば，別段なんらの努力もいらない．今まで汎用機で使ってきたプログラムをそのままスーパコンピュータにかければよい．FORTRANコンパイラが，ベクトル処理が可能な部分に対しては，自動的にしかるべき目的プログラムを生成してくれる．しかし，これでは単にスーパコンピュータを動かしたというに止り，まともな成果を期待することはできない．そのためには，利用者側での，プログラミングに対するある程度の工夫と努力が必要である．もちろん，メーカ側での，コンパイラの改良による支援が重要であり，実際にそのような活動が行われていて，かなりの効果をあげているのも確かであるが，それにも限界があり，あまり多くを望むのは無理である．

本章では，著者が数学ライブラリNUMPAC[4)~6)]の中の，いくつかの線形計算サブルーチンのベクトル化で実際に経験し，効果が認められたプログラミング技法について説明し，各位の参考に供したい．

3.1 ベクトル化制御行の利用

スーパコンピュータのベクトルユニットの中での動きは，概略前章で述べたとおりである．しかしながら，プログラマの立場からは，端的に次のように考えても差し支えないであろう．「スーパコンピュータは，プログラムの中の最も内側のDOループだけをベクトル化し，その部分をあたかも一体としてのベクトル間の演算として実行する」．たとえば，

```
      DO 10 I=1, N
      Z(I)=A*X(I)+Y(I)
   10 CONTINUE
```

というDOループは，あたかも数学でのベクトル演算

$$\boldsymbol{z}=a\cdot\boldsymbol{x}+\boldsymbol{y}$$

そのままの形で実行される．ところが，

```
      DO 20 I=2, N
      A(I)=A(I-1)*X+B(I)
   20 CONTINUE
```

のようなループでは，ベクトルAの要素間に，ベクトル化を阻害する依存関係があるために，これを無理にベクトル化すると，プログラムの意図に反する結果

が得られてしまう．実際には，現行のすべてのコンパイラには，このような依存関係を識別する能力があって，ベクトル化を差し控え，「ベクトル要素間の依存関係のために，ベクトルができない」旨のメッセージをベクトル化情報の中に出してくれるので問題はない．さて，正方行列を上ヘッセンベルグ形に変換する場合に現れる次の例を見てみよう．

```
      DO 30 K=1, N-2
      .....
      DO 20 J=K+1, N
      DO 10 I=K+2, N
      A(I,J)=A(I,J)-A(I,K)*A(K+1,J)
   10 CONTINUE
   20 CONTINUE
   30 CONTINUE
```

このプログラムを，国内メーカのスーパコンピュータで走らせたところ，「データの依存関係が不明のため，ベクトル化できない」というメッセージが出た．

これは，一番内側のループで，変化を受ける側の $A(I,J)$ が，変化を与える側の $A(I,K)$ や $A(K+1,J)$ としての役目にまわることがあるかどうかが，コンパイラの側からは見通せないので，ベクトル化を断念するという意味である．事実は，変化を受ける側と，変化を与える側は完全に分離していて，ベクトル化をしても全く差し支えない．このような場合には，ベクトル化制御行を利用し，問題のループの直前に，「ベクトル化を阻害する要素間の依存関係がない」旨の指示をコンパイラに与えることにより，強制的にベクトル化を促すことができる．ベクトル化制御行には，このほかに，IF 文の成立する割合など，ベクトル化のために有益な情報を指示できる機能もあるが，さして重要ではないので省略する．

3.2 ループ・アンローリング

正方行列の LU-分解法の中に出てくる，次のプログラム例を見よう．

```
      DO 30 K=1, N
      .....
      DO 20 J=1, K-1
      DO 10 I=K+1, N
      A(I,K)=A(I,K)-A(I,J)*A(J,K)
   10 CONTINUE
   20 CONTINUE
   30 CONTINUE
```

最も外側のループの制御変数 K が固定されると，最も内側のループでは，長さ $N-K$ の定ベクトル $A(I,K)$ から，同じ長さの変ベクトル $A(I,J)$ のスカラ $A(J,K)$ 倍を引くという，いわゆる“外積型”の，スーパコンピュータとしては最適の計算になっている．さて，このプログラムで，内側から2番目のループの制御変数 J の動き方をまばらにして二刻みとし，その代りに最も内側で，連続する二つの J の値に対する計算を一つにまとめ，さらに，この措置によって生ずる余剰部分に対する処理を加えると，次のようになる．

```
      DO 30 K=1, N
      .....
      JMAX=(K-1)/2*2
      DO 20 J=1, JMAX, 2
      DO 10 I=K+1, N
      A(I,K)=A(I,K)-A(I,J)*A(J,K)-A(I,
     *       J+1)*A(J+1,K)
   10 CONTINUE
   20 CONTINUE
      DO 50 J=JMAX+1, K-1
      DO 40 I=K+1, N
      A(I,K)=A(I,K)-A(I,J)*A(J,K)
   40 CONTINUE
   50 CONTINUE
   30 CONTINUE
```

これが，ループ・アンローリングという技術で，これにより，最も内側のループ内での計算密度が増加し，今まで遊んでいた演算装置が有効に活用できるようになり，ベクトル化の効果は著しく向上する．上に示した例は，内側から2番目の制御変数の動きを二刻みにした，いわゆる2重のアンローリングであるが，これを m 刻みにすれば，全く同様にして m 重のアンローリングが得られる．アンローリングの多重度 m を上げれば，それだけ計算密度が増すわけで，遊休装置がある間は性能は上昇するが，ある限度を越えれば当然無効となる．アンローリングがどの程度の多重度まで有効であるかは，パイプラインの数によって大きく左右される．著者の経験では，VP 系列では，効果は多重度2でおおむね飽和するのに対して，S-810 系列や SX 系列では，4までは確実に増大し，プログラムによっては8以上でもなお有効な場合があることが観察された．なお，アンローリングによって生ずる余剰部分の処理は，多重度と共に増加する負要因であるが，これも著者の経験に関する限り，ある程度以上

の大きさの問題では，これをどのように書いても同様であって，ほとんど無害である．

以上のように，アンローリングはたしかに効果があるけれども，簡潔なプログラムをわざと複雑にしなければ達成できない，いわば“汚い手”であって，このような美的感覚に反する手段を講じなければ，高性能を実現できないというのは実になげかわしい．このようなことは，コンパイラの領分に属することであるという考えもあり，一部には簡単なアンローリングを自動的に行うコンパイラもあると聞くが，果たしてコンパイラに全面的に任せてよいものかどうか，疑問に思われる．

3.3 ベクトル長の増加

3.2 節のプログラム例は，最も内側のループ内で，外積型のベクトル計算が行われていて，スーパコンピュータにとっては理想的な場合であった．しかし，アンローリングも含めて，そのまま実行に移すのは賢明でない．なぜなら，最も内側のループのベクトル長 $N-K$ は，K が小さい間は大きく好都合だが，K が大きくなると小さくなって，ベクトル化の大原則「ベクトル長を大きくせよ」に反する．そこで，第2と第3のループをそっくり入れ換えてみると，

```
      DO 30 K=1, N
      .....
      DO 20 I=K+1, N
      DO 10 J=1, K-1
      A(I,K)=A(I,K)-A(I,J)*A(J,K)
   10 CONTINUE
   20 CONTINUE
   30 CONTINUE
```

となる．今度は，最も内側のループでは，K と I の値を固定すると定まるスカラ量 $A(I,K)$ から長さ $K-1$ の二つのベクトルの内積を引くという計算が行われることになる．このような計算は本来は，ベクトル化に適しない逐次的な計算であるが，線形計算にはしばしば現れる重要な型の計算であるので，コンパイラには，これを全体的に内積計算であると認識してベクトル化する特別の機能がそなえられている．したがって，外積型ほどではないにしても，アンローリングの可能性をも含めて，K の値が大きいときはかなり有効である．以上を要約すると，現在の例では，K の値が小さい間は前節の外積型の計算を行い，K が大きくなると本節の内積型の計算に移行するのが得策である．

このときの最適の切り換えの K の値の N の値に対する依存性は，実験的に調べるより方法がないように思われる．

3.4 コンパイラへのデータ構造の明示

3.2 のプログラム例をもう一度観察しよう．最も内側のループで計算されるベクトル $A(I,K)$ は，最も外側のループの制御変数 K だけで定まり，中間のループの制御変数 J には無関係である．したがって，一番外のループで K の値が定まったとき，一度だけメモリからベクトルレジスタに移せば，中間のループの内部では，同じベクトルレジスタに置いたままで計算が続けられるはずである．ところが，人間の眼には明白なこの事実も，コンパイラにはこれを見抜くほどの洞察力がないために，中間のループ内で J の値が変化するたびごとに，メモリとベクトルレジスタの間でロードとストアを反復する能率の悪い目的プログラムが作られてしまう．そこで，ベクトル $A(I,K)$ の代りに，作業ベクトル W を用いて，

```
      DO 30 K=1, N
      .....
      DO 40 I=K+1, N
      W(I)=A(I,K)
   40 CONTINUE
      DO 20 J=1, K-1
      DO 10 I=K+1, N
      W(I)=W(I)-A(I,J)*A(J,K)
   10 CONTINUE
   20 CONTINUE
      DO 50 I=K+1, N
      A(I,K)=W(I)
   50 CONTINUE
   30 CONTINUE
```

というふうにプログラムを書き換えて，データの依存関係を明示してやると，コンパイラは安心して，ベクトル $W(I)$ をベクトルレジスタに入れっ放しにする理想的な目的プログラムを作ってくれるというわけである．

3.5 メモリ競合の防止

スーパコンピュータでは，メモリとベクトルユニット間のデータの転送を高速化するために，メモリをいくつかのメモリバンクに分けて，並列的にアクセスを行うようになっている．ところがデータの読み出しや書き込みの順序によっては，同一のメモリバンクにアクセスが集中することがあり，データ転送に大きな遅延を生ずる．この現象をメモリ競合あるいはバンクコ

ンフリクトという．データを連続的にアクセスする場合には，すべてのメモリバンクが並列的に動作して理想的なデータ転送ができるようになっているので，なるべくそのようになるようにするのがよい．したがって，最も内側のループでは行列要素はできるだけ列方向に辿るようにすべきである．しかし，すべての計算をそのように配置することは一般には不可能で，どうしても行方向へ辿る場合が必要になる．このとき行列要素は，その配列宣言の第1寸法おきにアクセスされることになる．この意味で，2次元配列の宣言で，第1寸法をどのように定めるかが重要になってくる．著者の知る限りでは，この点についての最良の方策は，第1寸法を倍精度データについては奇数に，単精度データについては奇数の2倍の偶数に定めることである．

たとえば，200 次の行列 A を扱う場合には，単精度ならば $A(202, 200)$，倍精度ならば $A(201, 200)$ と宣言するのが最良である．このようにして，A の要素を行方向に辿るときのメモリ競合を完全に防止することができる．

4. 数学ライブラリ NUMPAC のベクトル化

名古屋大学数学ライブラリ NUMPAC (Nagoya University Mathematical Package)[4)~6)] は，1971年名古屋大学大型計算機センター創設以来，15年の長年月にわたり，著者を代表とする名古屋大学工学部情報工学数値解析研究グループならびに名古屋大学大型計算機センター研究開発部ライブラリ担当者の協力の下に，開発，改良，整備，普及の努力が重ねられた結果，今や，約300種類，延べ総数900に及ぶ高性能プログラムの大集合体に成長を遂げた．現在，国立七大学大型計算機センターを始めとする，約30の計算施設に移植され，広く江湖の研究者の間に流通し，その研究活動に貢献しようとする態勢を整えるに至った．

さて，スーパコンピュータの出現と普及にともなって，NUMPAC も当然の対応を迫られることになった．そこで，1983年末頃よりそのベクトル化に着手した．手始めに，最もベクトル化の効果が高いと思われる倍精度用線形計算ルーチンの中から，次の7個を選んだ．

LEQLUW…LU-分解法
CHOLFW…コレスキー分解法
MCHLFW…改訂コレスキー分解法
} 連立一次方程式

HOQRVW…ハウスホルダ・QR 法
HQRIIW…ハウスホルダ・QR・逆反復法
HOBSVW…ハウスホルダ・二分・逆反復法
HEQRVW…ヘッセンベルグ・ダブル QR 法
} 固有値

上にあげた，7個のサブルーチン名は，ベクトル化によって新たに作成されたスーパコンピュータ版の正式名で，対応する元の汎用機版の名前の末尾のDをWに入れ換えたものである．

ベクトル化のための書き換え作業は，最初のうちは経験不足のため難航した．しかし，経験を深め，勘所がつかめてくるにつれて，次第に好転し，ハードウェアとコンパイラの内部を熟知した実力者の援助も得て，約1年の間に，S 810 版と VP 版ができ上った．これらのプログラムは，それぞれのメーカが提供している同種類のプログラムに比べても遜色のない性能を有することが確かめられ，スーパコンピュータの威力に感動すると共に，NUMPAC の優秀性に大いに自信を深めた次第である．最近は，新顔の SX 系列についても同様の試みを行い，満足な成果がえられた．

以下に，4個のプログラム LEQLUW，CHOLFW HOQRVW 及び HOBSVW の汎用機 M-380，M-680 H，ならびに国産各スーパコンピュータにおける性能テストの結果を報告する．おのおのの表の見出しの意味は次のとおりである．

N…テスト行列の次数
M380…富士通　M-380（名大センター）
M680…日立　M-680 H（分子研センター）
VP1 …VP 100（京大センター）
VP2 …VP 200（京大センター）
S81 …S-810/10（分子研センター）
S82 …S-810/20（東大センター）
SX1 …SX-1（阪大センター）
SX2 …SX-2（日本電気）

表の中の数値は，所要計算時間をミリ秒単位で測定したものである．

4.1 LEQLUW の性能

このプログラムは，LU-分解法（Crout 法）により，一般の実係数行列の連立一次方程式を解くためのもので，汎用機版 LEQLUD は，数あるサブルーチンの中でも最も利用頻度の高い有用なプログラムである．テスト問題としては，

$$a_{ij}=n+1-\max(i, j),\ i, j=1, 2, \cdots, n$$

を要素とする対称正値行列，すなわちフランク行列を係数とし，$b_i=\sum_{j=1}^{n} a_{ij},\ i=1, 2, \cdots, n$ を右辺ベクトル

とする（解ベクトルは $x_i=1$, $i=1, 2, \cdots, n$），方程式 $Ax=b$ を選んだ．結果は表-2 に示されている．なお，解の精度については，詳細は省略するが，行列の条件数 $16n^2/\pi^2$ に見合う程度のものが得られていることだけを報告しておく．

4.2 CHOLFW の性能

このプログラムはコレスキ分解法により，実対称正値行列を係数とする連立一次方程式を解くためのものである．テスト問題は，LEQLUW の場合と全く同じフランク方程式を採用した．結果は表-3 に示されている．当然のことであるが，LEQLUW の場合と比較すると，所要時間はすべて約半分になっている．

また改訂コレスキ分解法による類似のサブルーチン MCHLFW についても，同様の性能がえられている．

4.3 HOQRVW の性能

このプログラムは，ハウスホルダ・QR 法によって，実対称行列のすべての固有値と，要求に応じて，すべての固有ベクトルを計算するためのものである．テスト問題は，連立一次方程式の場合と同じくフランク行列を選んだ．この行列の固有値は，大きい方から番号をつけて

$$\lambda_k=1/4\sin^2\left(\frac{2k-1}{2n+1}\frac{\pi}{2}\right) \quad k=1, 2, \cdots, n$$

で与えられる．したがって，最大固有値と最小固有値はそれぞれ $\lambda_1\approx\frac{4n^2}{\pi^2}$, $\lambda_n\approx\frac{1}{4}$ となり，固有値は最小固有値の付近に密集している．なお要求精度は 10^{-15} とした．速度性能は，固有値のみの場合は表-4 に，固有ベクトルを含む場合は表-5 に示されている．連立一次方程式ルーチンの性能は，テスト問題に依存しないのに比べて，固有値ルーチンの性能はテスト問題と要求精度に依存することに留意していただきたい．

表-2 LEQLUW の速度性能 (ms)

OBS	N	M380	M680	VP1	VP2	S81	S82	SX1	SX2
1	100	77	43	13	9	12	10	6	5
2	200	529	328	40	32	52	41	25	17
3	300	1986	1159	103	73	142	102	68	40
4	400	5147	2759	220	147	307	204	144	80
5	500	11114	5504	404	264	556	349	264	141
6	600	20212	9796	701	420	920	552	438	227
7	700	33589	15769	1160	617	1417	814	673	342
8	800	52182	24014	1821	879	2066	1154	977	489
9	900	77496	.	2699	1213	2857	1572	1362	675
10	1000	110461	.	3809	1608	3884	2091	1836	899

表-3 CHOLFW の速度性能 (ms)

OBS	N	M380	M680	VP1	VP2	S81	S82	SX1	SX2
1	100	41	26	5	5	7	6	2	3
2	200	297	179	20	17	33	20	12	7
3	300	979	628	53	38	89	48	31	18
4	400	2368	1527	110	74	192	96	66	35
5	500	5251	3024	198	124	352	166	121	62
6	600	10127	5317	349	198	584	265	201	99
7	700	17800	8558	592	295	904	397	310	150
8	800	28782	13026	946	421	1322	567	452	215
9	900	44244	.	1413	579	1846	776	633	298
10	1000	64873	.	2013	772	2499	1031	858	399

4.4 HOBSVW の性能

このプログラムは，ハウスホルダ・二分・逆反復法によって，実対称行列の，指定された個数の固有値と，要求に応じて，対応する固有ベクトルを計算するためのものである．元来，スツルム列の符号変化の数を利用する二分法の部分は，データの依存関係のある逐次式の計算で，そのままではベクトル化できない．HOBSVW では，二分法を制御する内側のループと，その外側の固有値の番号を制御するループを入れ換えることによってベクトル化に成功した．HOQRVW

表-4 HOQRVW の速度性能 (ms)
（全固有値）

OBS	N	M380	M680	VP1	VP2	S81	S82	SX1	SX2
1	50	32	22	15	13	18	19	6	5
2	100	203	137	53	46	65	66	24	19
3	150	650	431	120	97	146	147	59	42
4	200	1494	1016	224	172	267	261	115	75
5	250	2896	1957	367	271	431	414	194	121
6	300	5062	3381	562	400	652	606	311	191
7	350	8148	5381	805	560	930	839	457	277
8	400	12223	7958	1109	754	1268	1119	646	378
9	450	17477	11396	1479	984	1672	1448	882	500
10	500	23989	15685	1981	1252	2145	1826	1160	643
11	550	31958	20729	2490	1584	2710	2263	1506	827
12	600	41469	26923	3163	1972	3361	2754	1902	1037

表-5 HOQRVW の速度性能 (ms)
（全固有値，ベクトル）

OBS	N	M380	M680	VP1	VP2	S81	S82	SX1	SX2
1	50	99	59	28	23	29	30	12	9
2	100	696	404	117	85	117	118	45	34
3	150	2262	1318	289	196	285	285	135	76
4	200	5258	3149	569	371	555	548	269	144
5	250	10136	6118	1022	610	942	924	466	238
6	300	17513	10499	1561	941	1567	1444	816	441
7	350	27850	.	2326	1343	2308	2051	1204	639
8	400	41564	.	3277	1876	3243	2801	1823	883
9	450	59362	.	4501	2535	4395	3699	2464	1185
10	500	82031	.	5934	3292	5773	4813	3246	1541
11	550	110152	.	7940	4232	7658	6338	4394	2170
12	600	144284	.	10135	5310	9653	7973	5511	2699

表-6 HOBSVW の速度性能 (ms)
(全固有値)

OBS	N	M380	M680	VP1	VP2	S81	S82	SX1	SX2
1	50	139	94	26	24	20	21	11	9
2	100	632	416	87	59	62	59	34	25
3	150	1617	1053	185	113	132	116	89	50
4	200	3229	2118	330	191	233	198	158	85
5	250	5613	3691	527	288	374	306	253	134
6	300	8979	5856	791	416	603	452	411	228
7	350	13464	8762	1106	564	851	625	580	318
8	400	19199	12346	1513	759	1155	849	831	419
9	450	26348	16901	2003	980	1519	1050	1099	545
10	500	34957	22374	2578	1251	1949	1336	1409	692
11	550	45261	28930	3341	1585	2531	1684	1834	916
12	600	57314	36666	4154	1950	3137	2073	2271	1127

表-7 HOBSVW の速度性能 (ms)
(全固有値，ベクトル)

OBS	N	M380	M680	VP1	VP2	S81	S82	SX1	SX2
1	50	192	128	48	43	51	48	27	18
2	100	1015	649	177	138	196	170	109	63
3	150	2896	1821	406	301	459	374	283	136
4	200	6299	3958	756	538	864	673	527	245
5	250	11524	7301	1250	859	1437	1075	862	400
6	300	19554	12203	1890	1279	2277	1616	1359	649
7	350	30825	18838	2714	1791	3290	2272	1929	912
8	400	45531	27264	3710	2438	4549	3076	2724	1233
9	450	64198	38274	4981	3206	6076	3961	3582	1617
10	500	87056	51855	6454	4123	7891	5045	4592	2068
11	550	114701	67937	8553	5236	10201	6360	5923	2751
12	600	147742	87521	10732	6566	12744	7810	7280	3369

のときと同様に，テスト問題はフランク行列，要求精度は 10^{-15} とし，すべての固有値を求める場合(表-6)とすべての固有値と固有ベクトルを求める場合(表-7)とについて速度性能の測定を行った．予想に反して，このような場合でも，HOBSVW の性能は HOQRVW に比べてほとんど遜色がないことがわかった．

以上で，NUMPAC スーパコンピュータ版の代表的なプログラムの性能についての報告を終る．著者の本意は，NUMPAC の優秀性を明らかにすることであったが，図らずも各機種の競演の形となってしまった．ここで，誤解を防ぐために次のことを強調しておく．すなわち，おのおののプログラムはそれぞれの機種に応じてチューニングされているが，その度合は，各機種に対する習熟度の差によって一様ではないこと，したがってこの競演は必ずしも公平ではないことである．しかしながら，ともかく，一断面の比較にはなっていることは否めない．この報告が，良い意味での競争を促す契機となれば幸甚である．

NUMPAC のベクトル化は，逆行列，非線形連立方程式，そして連立一次方程式の複素数版へと，緩やかではあるが，着実な歩みをつづけている．各位のご期待を乞う次第である．

5. おわりに

スーパコンピュータはたしかに驚嘆すべきスピードをもっている．これを理想的に操れば，汎用機の百倍にも及ぶ高速を達成することができる．しかし，そのためには，奇矯ともいえる種々の高度のプログラミング技術を自由自在に駆使しなければならない．これは，専門家以外の計算者には至難の技ではあるまいか．ここに NUMPAC や各メーカ提供の数学ライブラリの存在意義がある．スーパコンピュータを利用される計算者各位には，それぞれの計算ジョブの中の，連立一次方程式，固有値問題，フーリエ解析などの定型的な部分については，積極的に数学ライブラリを利用されんことを切望する．

終りにのぞみ，NUMPAC ならびにそのスーパコンピュータ版の開発と管理の両面において，絶大なる尽力をいただき，今回の性能調査表の編集の労をとられた，中京大学秦野甯世助教授に深甚の感謝を捧げる．また，著者にこの小論を草する機会を与えられた，本小特集企画学会誌編集委員会に謝意を表明する．

参考文献

1) 小高俊彦，小林二三幸，河辺 俊，長島重夫：最大性能が 630 MFLOPS で 1G バイトの半導体拡張記憶が付くスーパーコンピュータ HITAC S-810，日経エレクトロニクス，4月号，pp. 159-184 (1983).
2) 平栗俊男，田畑 晃，槌本隆光，田口尚三：マシン・サイクル 7.5 ns を達成した並列パイプライン処理方式のスーパコンピュータ FACOM VP，日経エレクトロニクス，4月号，pp. 131-155 (1983).
3) 古勝紀誠，渡辺 貞，近藤良三：最大性能 1.3 GFLOPS，マシン・サイクル 6 ns のスーパーコンピュータ SX システム，日経エレクトロニクス，11月号，pp. 237-271 (1984).
4) 二宮市三，秦野甯世：数学ライブラリ NUMPAC，情報処理，Vol. 26, No. 9, pp. 1033-1042 (1985).
5) 名古屋大学大型計算機センター，ライブラリー・プログラム利用の手引（数値計算編）(1982).
6) 名古屋大学大型計算機センター，ライブラリー・プログラム利用の手引（増補版）(1982).

（昭和 61 年 7 月 7 日受付）