

スーパーコンピュータのプログラム技術

中 村 絹 代* 吉 田 正 廣*

Programming Techniques for High-Speed Processing of Large Scale Numerical Simulation on Supercomputer

by

Kinuyo NAKAMURA and Masahiro YOSHIDA
National Aerospace Laboratory

ABSTRACT

For the purpose of high-speed processing of large scale numerical simulation, the development of high performance computers is required. But it is also important to make good use of computer resources and architecture through programming techniques and the improvement of algorithms.

This paper presents a method for improving algorithms and programming techniques which enables us to use supercomputers efficiently. The method is illustrated by giving an example of a numerical analysis of codes of transonic full potential flows about wings.

1. はじめに

近年、科学技術計算の大規模化に伴って計算の高速化の要求が高まっている。これに応え、半導体技術を中心としたコンピュータハードウェア技術のめざましい進歩を基盤として、計算速度、記憶容量の両面において、これまでの大型汎用計算機を大幅に上まわるスーパーコンピュータの開発がなされている。

大規模な科学技術計算の高速化の実現のためには、

- (1) ハードウェアのスカラ演算およびベクトル演算の性能
- (2) スーパーコンピュータを活かす解法（プログラムすべき問題自体の並列処理可能性やアルゴリズム等）
- (3) プログラム実行におけるベクトル演算の比率（ベクトル化率）

が重要である。スカラ演算およびベクトル演算の性能はハードウェアおよび計算機アーキテクチャにより決定される。また、ベクトル化率は

- (a) プログラム設計レベル（(2)を含む）
- (b) プログラムコーディングレベル
- (c) コンパイラレベル

の各段階での検討および改善により大きくすることが可能である。

スーパーコンピュータを有効に使用するためには並列処理可能な範囲をできるだけ広くすることが基本であるが、対象とする問題をプログラム設計、プログラムコーディング、コンパイル、スーパーコンピュータで実行するまでのそれぞれの過程でその並列性はさまざまな要因により減少し、その後の過程で復元されることはない。このように計算の高速化の実現のためには、問題の持っている並列性をできるだけ保持することが重要である。特に、計算機ユーザーにとっては、プログラム設計およびプログラムコー

* 航空宇宙技術研究所

ディングの段階でスーパーコンピュータを活かす解法
 の選択やプログラムコーディングを行うことにより、
 並列性を保持することが重要である。

本報告ではスーパーコンピュータを使用して数値解
 析コードの高速化を図ることを目的として、三次元
 遷音速翼まわり完全ポテンシャル流数値解析コード
 の方程式解法部をベクトル計算機向きのアルゴリズム
 をもつ、

- (イ) ODD-EVEN SLOR法
- (ロ) ODD-EVEN SOR XSWEEEP法
- (ハ) ODD-EVEN SOR法

の三つの数値計算法を用いて書き換えた。さらに、
 ベクトル計算機を活かすプログラム技術を用いるこ
 とにより、数値解析コードの高速化効果が得られた。

2. スーパーコンピュータの概要

実用化されているスーパーコンピュータの多くはパ
 イプライン方式とよばれる構成方式が採用されてい
 る。¹⁻³⁾

図1にパイプライン方式のスーパーコンピュータの

ひとつであるFACOM VP-400 システムのハード
 ウェア構成図を示す。VP-400 システムの本体系装
 置はベクトル処理装置、主記憶装置、チャンネル装置
 およびサービスプロセッサから成る。ベクトル処理
 装置はベクトルユニットおよびスカラユニットから
 成る。

ベクトルユニットはベクトル命令を高速に処理す
 るユニットであり、加算/論理演算パイプライン、
 乗算パイプラインおよび除算パイプラインの内の2
 本のパイプライン、およびロード/ストアパイプ
 ライン、マスクパイプラインの4本のパイプラインが
 並列動作可能である。スカラユニットはスカラ命令
 を実行するユニットである。

ベクトルユニットで処理されるベクトル命令はス
 カラ命令に比較して格段に高速なため、プログラ
 ムの実行においてベクトルユニットで処理される部
 分ができるだけ多い、すなわち、ベクトル化率が高
 い方がよい。また、並列動作可能なパイプラインの
 数が多い程、そしてベクトル命令の要素数(ベクトル
 長)が多い程、スーパーコンピュータを効率良く利用

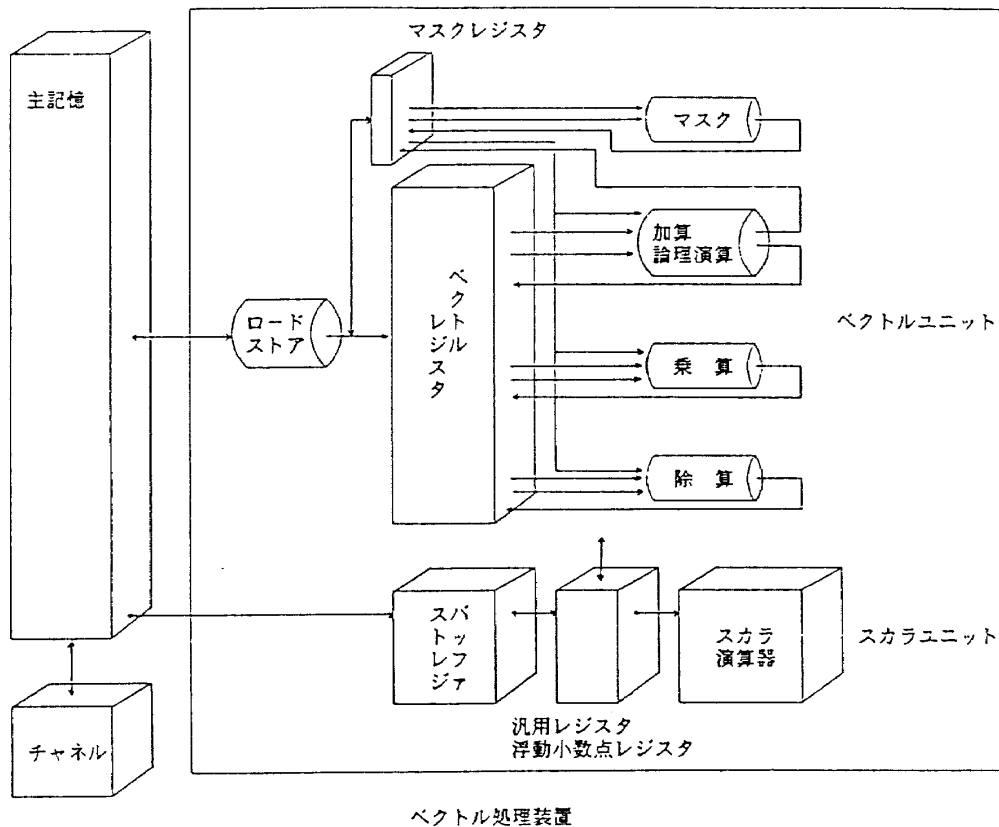


図1 FACOM VP400 システムハードウェア構成図

することができる。従って、プログラム作成を行う際には、ベクトル化率を高め、パイプラインを並列に動作させ、ベクトル長を長くするように配慮することがプログラムの高速化につながる。^{4,5)}この他、主記憶装置とベクトルレジスタとの間のデータの移動(ロード/ストア)をできるだけ減少させ、効率の良いロード/ストアを行えるように配慮することなども必要である。

3. 三次元遷音速完全ポテンシャル流数値解析コードの高速化

3.1 高速化対象プログラムの概要

高速化対象プログラムは三次元遷音速翼まわりのポテンシャル流の解析を行うプログラム⁶⁾であり、それはポテンシャル方程式

$$(a^2 - u^2)\phi_{xx} + (a^2 - v^2)\phi_{yy} + (a^2 - w^2)\phi_{zz} - 2uv\phi_{xy} - 2uw\phi_{xz} - 2vw\phi_{yz} = 0$$

ここで、

$$u, v, w : x, y, z \text{ 方向の速度成分}$$

$$(\phi_x, \phi_y, \phi_z)$$

$$a^2 = a_0^2 - (\gamma - 1)(u^2 + v^2 + w^2)/2$$

a_0 : よどみ点音速

γ : 比熱比

がSLOR法を用いて解かれている。格子点数は、

COARSE	48 × 7 × 8
MEDIUM	96 × 13 × 16
FINE	192 × 25 × 32

とし、反復回数は各100回としている。

3.2 ベクトル計算機向きのアルゴリズムをもつ数値計算法について

ベクトル計算機向きのアルゴリズム^{7,8)}として

- (イ) ODD-EVEN SLOR法
- (ロ) ODD-EVEN SOR XSWEEEP法
- (ハ) ODD-EVEN SOR法

の三つがあげられる。

図2にこれらの計算法の計算空間を示す。X軸、Y軸およびZ軸の要素をそれぞれI、JおよびKとする。

図2の(a)は(イ)の計算法の計算空間である。I方向の連立方程式をJおよびK方向にスイープする場合、

J+K: ODDならば白丸、J+K: EVENならば黒丸とし、はじめに白丸のI方向の方程式を一度に並列に計算した後、黒丸の場合を同様に計算して1回の反復とする。白丸を計算する時、隣接の黒丸の点の値は古い値を使い、黒丸を計算する時には隣接の白丸の点の値は今求められた新しい値を使う。本計算法ではY-Z平面はチェッカーボード、X-Y平面およびX-Z平面は白と黒の縞になる。

図2の(b)は(ロ)の計算法の計算空間である。あるIに対して、Y-Z平面のJ+K: ODDならば白丸、J+K: EVENならば黒丸とし、はじめに白丸を一度に並列にSOR法で解き、次に黒丸について同様に解き、これをX方向にスイープし、1回の反復とする。白丸を計算する時、隣接の黒丸の点の値は古い値を使い、黒丸を計算する時には隣接の白丸の点の値は新しい値を使う。ODD-EVEN SLOR法同様、Y-Z平面はチェッカーボード、X-Y平面およびX-Z平面は白と黒の縞になる。

図2の(c)は(ハ)の計算法の計算空間である。I+J+K: ODDならば白丸、I+J+K: EVENならば黒丸とし、はじめに白丸の場合を一度に並列にSOR法で解き、次に黒丸について同様に解いて1回の反復とする。白丸を計算する時、隣接の黒丸の点の値は古い値を使い、黒丸を計算する時には隣接の白丸の点の値は新しい値を使う。前に述べた二つの計算法と異なり、本計算法の計算空間はどの平面をとってもチェッカーボードになる。

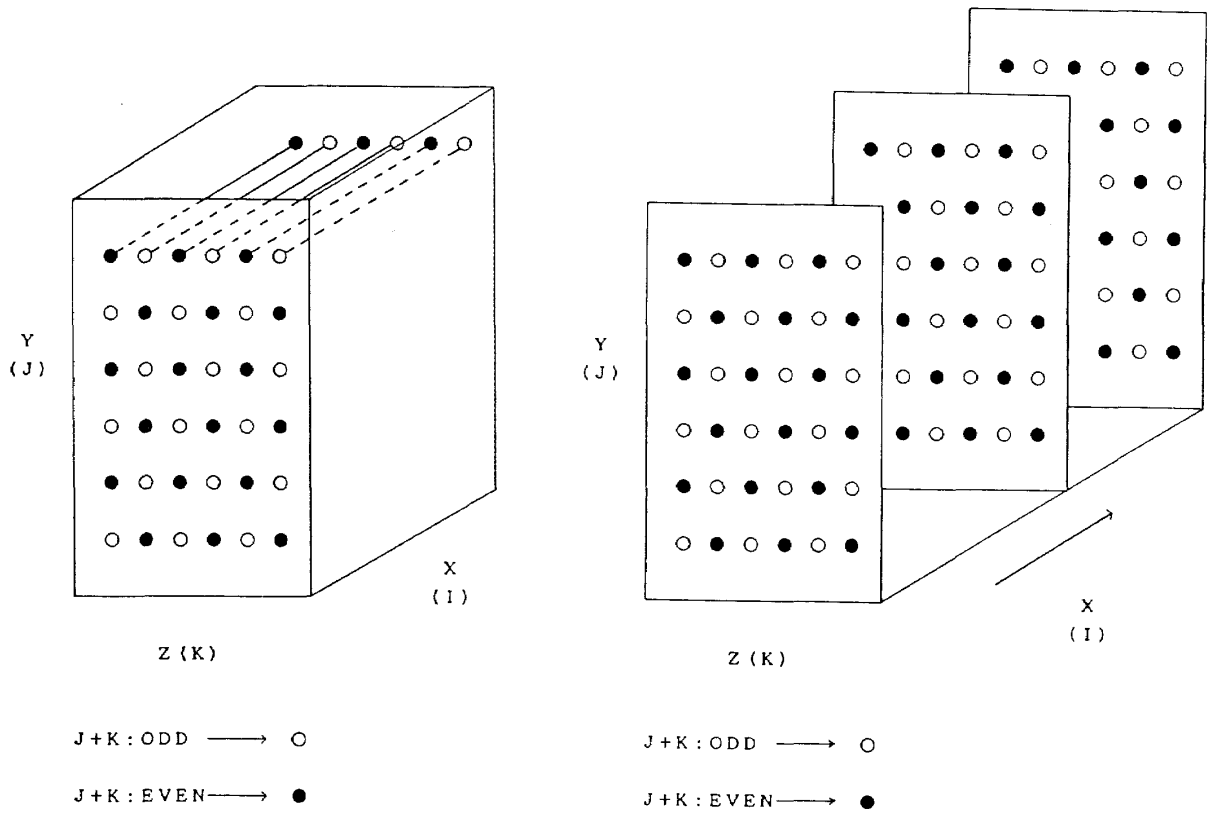
上に述べた三つの計算法に変更することにより回帰演算を回避することができる。回帰演算とは

```
DO 10 I=1, II
  P(I)=A*P(I-1)
10 CONTINUE
```

のような演算を言い、このDOループはベクトル化非対象となる。

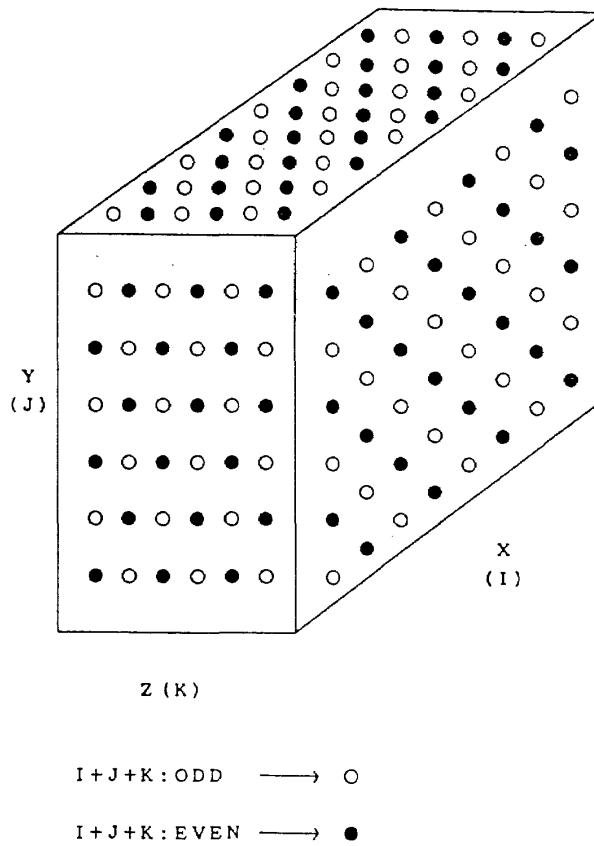
3.3 プログラム技術について

アルゴリズムを変更するだけでめざましい高速化を期待するのには限度があり、スーパーコンピュータおよびアルゴリズムを活かすプログラム技術が必要である。本節ではODD-EVEN SLOR法に施したプログラム技術について示す。



(a) ODD-EVEN SLOR法

(b) ODD-EVEN SOR XSWEEP法



(c) ODD-EVEN SOR法

図2 数值計算法別計算空間

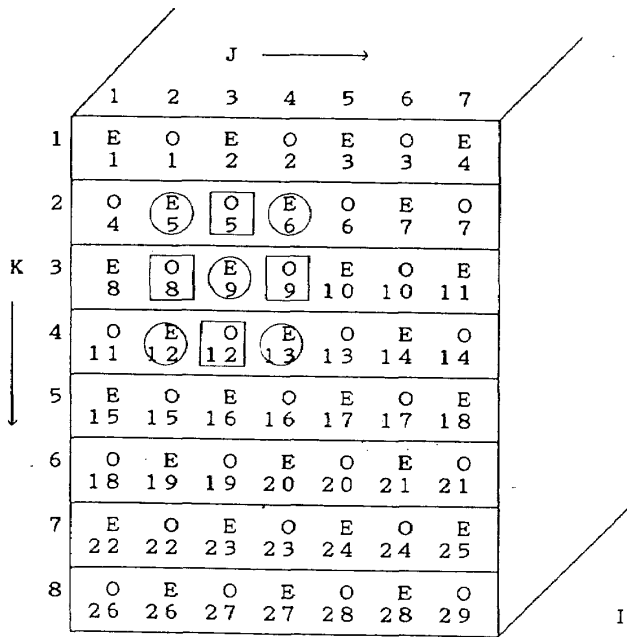


図3 Y-Z平面

(1) 計算空間の格子点数 $II \times JJ \times KK$ の JJ を奇数化

Y方向の要素数 JJ を奇数にとることにより隣接要素位置の決定を簡単に行うことができる。図3は計算空間のY-Z平面を表わしたものであり、EおよびOは $J+K$ の値がEVENであるかODDであるかを示す。EおよびOの下の数字はEおよびO別の要素の順番を示す。たとえば、E9の位置の差分計算を行う場合には隣接のO8, O9, O5, O12および、斜め隣接のE5, E6, E12, E13を使う。この際の隣接要素位置、特に斜め隣接要素位置の決定を簡単に行うことができる。

Y方向の要素数が偶数の場合には斜め隣接要素位置の決定方法が複雑になるので、プログラム実行上の負荷が大きい。

(2) 配列処理

計算空間の各格子でのポテンシャルを配列Pを用いて表わす。配列Pに次の変更を施す。

(a) 配列Pの低次元化

三次元配列 $P(I, J, K)$ を二次元配列 $P(I, JK)$ にする。

(b) 配列Pの次元の入れ換え

$P(I, JK)$ を $P(JK, I)$ として次元を入れ換える。

(c) 配列PのODD, EVEN別の配列への置き換

え

$P(JK, I)$ を $POD(JK1, I)$ および $PEV(JK2, I)$ に分けてODD, EVEN別の配列を用意する。

たとえば、

$$P(200, 30, 30)$$

$$P(200, 900)$$

$$P(900, 200)$$

$$POD(450, 200), PEV(450, 200)$$

のようにして配列Pから配列PODおよびPEVを新しくつくる。

低次元化し、次元を入れ換えることによりベクトル長を拡大し、配列の参照パターンを効率の良いものにできるので、効率的にベクトル計算が行われる。その理由はベクトルデータには連続データ、等間隔データおよび間接指標データがあり、連続データは主記憶のバンクの競合を回避できるので最高速で主記憶からのデータの読み出しまたは書き込みを行うことができ、最も効率の良い参照パターンとなるからである。

さらに、ODD, EVEN別に配列を用意することにより効率的な配列の参照パターンを維持できる。

(3) 係数処理

方程式の係数部の計算を次のように変更する。

(a) 係数の一括計算

係数計算部分は反復毎に同じ値を使用するので反復毎に計算する必要はない。あらかじめ一度計算し、その後の反復では参照する形式にすることにより実行時間の短縮を図ることができる。

(b) 配列Pに合わせた係数の処理

係数の配列を $P(JK, I)$ に合わせて、JおよびKに関する係数は $B(JK)$ 、Iに関する係数は $A(I)$ 、I, JおよびKに関する係数は $C(JK, I)$ のように配列Pと次元を同じにする。また、配列Pと同様に、JおよびKに関する係数とI, JおよびKに関する係数はODD, EVEN別の配列を用意する。

係数の配列の次元を配列 P の次元に合わせない場合には、I に関する係数以外の係数の配列は間接指標ベクトル（リストベクトル）となる。間接指標ベクトルデータを主記憶装置からベクトルレジスタにもってくる場合には LOAD 命令を 2 回必要とする。乗算命令や加算命令等の演算命令に比較して LOAD 命令に要する時間は長い。

また、間接指標データであるため、配列の要素の主記憶上の位置が同一バンク上に存在する可能性もあり、この条件が重なる場合にはさらに実効効率の悪いものとなる。

(4) 内点計算の一括処理

反復において求めた解を新しい解として置き換えるか否かの判定のために、内点であれば 1、内点でなければ 0 を示す配列 L (JK, I) を用意し、これを使用して値の置き換えを制御する。配列 L は ODD, EVEN 別に用意する。

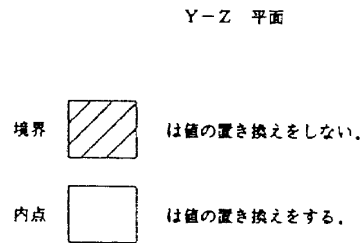
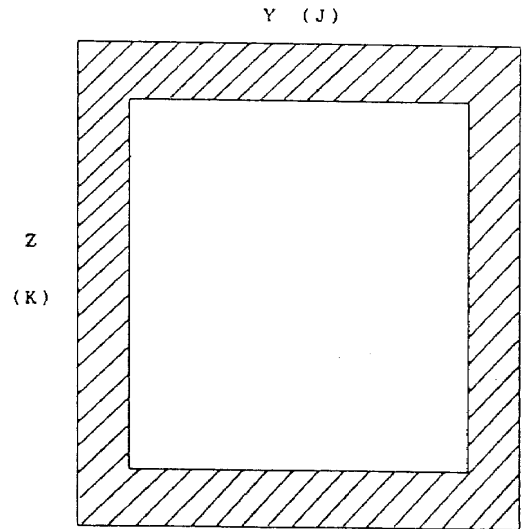
図 4 の(a)に計算空間の Y-Z 平面の内点および境界点を示す。図 4 の(b)に(a)の Y-Z 平面の各要素について計算を行う際の計算順序を示す。図において斜線部分に対応する要素には 0、他に対応する要素には 1 を格納した配列 L を用意して値の置き換えを制御する。これにより、拡大したベクトル長を維持できる。

3.4 高速化効果について

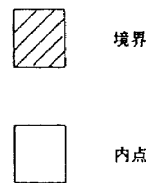
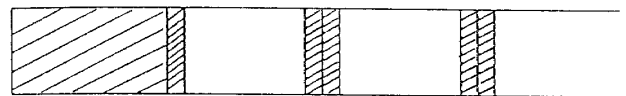
3.2 および 3.3 で述べた数値計算法およびプログラム技術を用いて 3.1 に示す数値解析コードの高速化を図った。表 1 に実測値を示す。

表 1 より、各計算法の M380 計算機システムの GO ジョブステップの実行時間に対する倍率では 50 倍近く、また、オリジナルプログラムの M380 システムの実行時間に対する倍率では 30 ~ 45 倍の高速化が達成されたことがわかる。

表 2 の(a)に数値計算法別 GO ジョブステップの主記憶使用量を示す。計算法 1 ~ 3 の主記憶使用量はオリジナルプログラムの主記憶使用量 1.32MB の 17 ~ 20 倍の主記憶量を必要とする。これは、配列 P を ODD, EVEN 別の配列にしていることに付随して係数配列および次元を配列 POD や PEV に合わせ



(a) Y-Z 平面



(b) Y-Z 平面の計算順序

図 4

たこと、および新しい配列を加えたことに原因があり、プログラムを高速化するために利用可能な主記憶を十分活用していることがわかる。

特に、スーパーコンピュータでは最大 256MB の主

表1 数値計算法別スーパーコンピュータ実測値

計算法	M380システム 実行時間 (秒)	VP400システム 実行時間 (秒)	倍率	倍率 対オリジナル
オリジナル	430.37	44.71	9.6	9.6
計算法1	484.09	10.13	47.8	42.5
計算法2	425.31	9.44	45.1	45.6
計算法3	649.60	13.71	47.4	31.4

計算法1 ODD-EVEN SLOR 法
 計算法2 ODD-EVEN SOR XSWEEP 法
 計算法3 ODD-EVEN SOR 法

記憶の利用が可能であるのでそれを十分活用してプログラムの高速化を図ることができる。

表2の(b)に数値計算法別ベクトル化率を示す。いずれもベクトル化率の大変高いプログラムであることがわかる。

4. む す び

大規模計算の高速化の実現を目的として、計算機ユーザに関わる部分、すなわち、スーパーコンピュータを活かす解法やプログラム技術を用いて三次元遷音速完全ポテンシャル流数値解析コードの高速化を図り、オリジナルプログラムのM380 計算機システムにおける実行時間の30～45倍の高速化が達成された。これにより、スーパーコンピュータを有効利用してプログラムの高速化を図るためには、プログラム設計やプログラムコーディングの段階で、スーパーコンピュータを活かす解法やプログラム技術を用いて並列性を保持するための検討を十分に行うことが必要であることが実証された。

このようにプログラムの高速化を徹底的に図り数十倍の高速化効果が得られることは、汎用計算機でその仕事を行うのに費やす時間の数十分の一の時間で同じ成果が得られることなので、プログラムの高速化は研究者にとって重要問題の一つであろう。

今後の課題として、3.2で述べた計算法を用いた数値解析コードの実用化に向けて、適用範囲の調査を行う予定である。計算法(i), (ii), (iii)の順に適用範囲が狭くなることが予想される。

最後に、本報告がスーパーコンピュータを利用される研究者にとってプログラムの高速化の参考となれば幸いである。

表2

(a) 数値計算法別GO-JOBSTEP主記憶使用量

計算法	プログラム領域 (MB)	データ領域 (MB)	合計 (MB)
オリジナル	—	—	1.32
計算法1	0.348	23.432	23.78
計算法2	0.192	22.548	22.74
計算法3	0.192	25.668	25.86

計算法1 ODD-EVEN SLOR 法
 計算法2 ODD-EVEN SOR XSWEEP 法
 計算法3 ODD-EVEN SOR 法

表2(b) 数値計算法別ベクトル化率

計算法	ベクトル化率 (%)
オリジナル	94.4
計算法1	99.6
計算法2	99.6
計算法3	99.6

計算法1 ODD-EVEN SLOR 法
 計算法2 ODD-EVEN SOR XSWEEP 法
 計算法3 ODD-EVEN SOR 法

参 考 文 献

- 1) 小高俊彦, 小林二三彦, 長島重夫: 最大性能が630MFLOPSで1Gバイトの半導体記憶が付くスーパーコンピュータHITACS-810, 日経エレクトロニクス, 1983, pp.159-184
- 2) 平栗俊男, 田畑 晃, 槌本隆光, 田口尚三: マシンサイクル7.5nsを達成した並列処理パイプライン処理方式のスーパーコンピュータFACOM VP, 日経エレクトロニクス, 1983, 4.11
- 3) 古勝紀誠, 渡辺 貞, 近藤良三: スーパーコンピュータSXシステム/高出力半導体レーザ, 日経エレクトロニクス, 1984, 11.19
- 4) 中村絹代, 吉田正廣, 峯尾真一: プログラム高速化技術とスーパーコンピュータSXシステムによる検証, 航空宇宙技術研究所報告TR-909

- 5) 吉田正廣, 中村絹代, 内田啓一郎, 棚倉由行
: スーパーコンピュータ VP システムによるプログラム
の高速化技術, 航空宇宙技術研究所報告 TR-915
- 6) Antony Jameson and D.A. Caughey:
Numerical Calculation of the Transonic Flow Past a Swept Wing, NASA-CR-153297.
- 7) Jerry C. South Jr. and James D. Keller:
Vector Processor Algorithms for Transonic Flow Calculations, AIAA JOURNAL Vol. 18, No. 7, 79-1457R.
- 8) N.D. Melson and James D. Keller: Use of CYBER 203 and CYBER 205 Computer for Three-Dimensional Transonic Flow Calculations.