

JAERI-M

7573

J
A
E
R
I
·
M

7
5
7
3

大次元行列演算用サブルーチン・パッケージ

"A T L A S"

1978年3月

常松俊秀・竹田辰興・藤田恵一・松浦俊彦・田原伸夫

日 本 原 子 力 研 究 所
Japan Atomic Energy Research Institute

この報告書は、日本原子力研究所がJAERI-Mレポートとして、不定期に刊行している研究報告書です。入手、複製などのお問い合わせは、日本原子力研究所技術情報部（茨城県那珂郡東海村）あて、お申しこしてください。

JAERI-M reports, issued irregularly, describe the results of research works carried out in JAERI. Inquiries about the availability of reports and their reproduction should be addressed to Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, Japan.

大次元行列演算用サブルーチン・パッケージ "ATLAS"

日本原子力研究所東海研究所核融合研究部
常松俊秀、竹田辰興、藤田憲一*、松浦俊彦*
田原伸夫*

(1978年1月31日 受理)

核融合プラズマの挙動解析のための流体シミュレーション・コード用大次元行列演算ルーチン群の開発を行った。このルーチン群は "ATLAS" と呼ばれ、基本演算ルーチン群、連立一次方程式の求解ルーチン群、一般固有値問題の求解ルーチン群の3つの演算ルーチン群とユーティリティ・ルーチン群から成る。本報告書では、これらのルーチン群の内容、使用例およびATLASと計算機システムとの関連について述べる。

* 計算センター外来研究員、富士通株式会社

Large-Scale Matrix-Handling Subroutines "ATLAS"

Toshihide TSUNEMATSU, Tatsuoki TAKEDA, Keiichi FUJITA*,
Toshihiko MATSUURA*, and Nobuo TAHARA*
Division of Thermonuclear Fusion Research,
Tokai Research Establishment, JAERI

(Received January 31, 1978)

Subroutine package "ATLAS" has been developed for handling large-scale matrices. The package is composed of four kinds of subroutines, i.e., basic arithmetic routines, routines for solving linear simultaneous equations and for solving general eigenvalue problems and utility routines. The subroutines are useful in large scale plasma-fluid simulations.

Keywords : Large-Scale Matrix, Arithmetic Routine, Linear Simultaneous Equation, General Eigenvalue Problem, Utility Routine, Plasma-Fluid Simulation, ATLAS Code

目 次

1. はじめに.....	1
2. ファイル形式.....	2
2.1 データ構造.....	2
2.2 ファイル形式.....	3
3. ATLAS-A ルーチン群.....	4
3.1 ATLAS-A ルーチン群の概要.....	4
3.2 ATLAS-A ルーチン群の使用例.....	4
4. ATLAS-L ルーチン群.....	5
4.1 ATLAS-L ルーチン群の概要.....	5
4.2 ATLAS-L ルーチン群の使用例.....	7
5. ATLAS-E ルーチン群.....	8
5.1 ATLAS-E ルーチン群の概要.....	8
5.2 ATLAS-E ルーチン群の使用例.....	9
6. 計算機システムとの関連.....	10
7. まとめと今後の課題.....	11
謝 辞.....	11
参考文献.....	12

Contents

1. Introduction	1
2. File Formats	2
2.1. Data Structure	2
2.2. File Formats	3
3. ATLAS-A Routines	4
3.1. Summary of ATLAS-A Routines	4
3.2. Applications of ATLAS-A Routines	4
4. ATLAS-L Routines	5
4.1. Summary of ATLAS-L Routines	5
4.2. Applications of ATLAS-L Routines	7
5. ATLAS-E Routines	8
5.1. Summary of ATLAS-E Routines	8
5.2. Applications of ATLAS-E Routines	8
6. ATLAS and Computer System	10
7. Conclusions and Discussions	11
Acknowledgement	11
References	12

1. はじめに

核融合実験装置が大型化し複雑化するにつれ、装置の設計や実験結果の解析のための計算機コードも大型かつ複雑になってきている。特に核融合プラズマの挙動解析のための流体シミュレーション・コードでは、大次元行列演算を効率良く行うことが必須とされている。例えば、変分法によるMHD安定性解析コードでは、大次元帯行列の一般固有値問題を、また、非線形安定性解析コードでは大次元帯行列の連立一次方程式を効率良く解くことが必要な要請となっている。

大次元行列の場合、たとえその行列が帯行列であっても全要素をいちどに主記憶装置に格納することは不可能である。そのため、磁気ディスクや磁気テープなどの補助記憶装置と主記憶装置との間でデータの転送を行わなければならない。行列要素が補助記憶装置に存在する場合には、数値解析の教科書に載っているアルゴリズムをそのままプログラムにしたのでは、入出力のための時間も計算時間も多くなり、行列の次元が大きくなると事実上計算が不可能になる。そこで、教科書的な主記憶内の演算のみのアルゴリズムを大次元向きに変更し、さらに行列の構造に応じて入出力アルゴリズムも十分に検討した上でプログラムを作成しなければならない。

既に、構造解析の分野では、大次元帯行列の逆行列を計算するために、入出力を十分に考慮に入れたサブルーチン群が開発され、汎用コードに組み込まれている。しかしながら、これらのサブルーチン群は、構造解析の特殊性を考慮してつくられているため、汎用コードから取りはずして単体で使用することは、技術的にかなり難しい問題となる。さらに、プラズマの流体シミュレーションに現われる行列は、構造解析の場合と違って各自由度間の結合が密なために帯内の非零要素の数が非常に多い。そのため、プラズマの流体シミュレーション・コードの開発には、非零要素が多い場合に効果的な大次元帯行列演算用サブルーチン群の作成が必要となる。このような理由から、今回開発され概略が完成したルーチン群が本報告書で述べる“ATLAS”(Assembly of The Large-scale Array-handling Subroutines)である。

この開発は、汎用流体シミュレーション・コード・システム“TRITON⁻¹⁾”の開発計画(TRITON計画)の一環として行われたが、現在未開発の部分もあり、ATLASの今後の改良、機能強化、さらに行列演算用プリプロセッサの開発は、TRITON計画の中で最も重要な要因の一つになっている。

このように、ATLASは、プラズマの流体シミュレーション・コード用に作成されたサブルーチン群であるが、もちろん、汎用サブルーチンとして他の分野の計算のためにもすぐに利用者のプログラムから参照できる形式になっている。ただし、現在のATLASでは、すべて倍精度演算を行っているので、単精度データを取り扱うにはATLASルーチン群を多少修正する必要がある。

ATLASは次の3種類のサブルーチン群から成り立っている。

- (1) ATLAS-Aルーチン群：ベクトル、行列間の加減乗算を行う。
- (2) ATLAS-Lルーチン群：連立一次方程式の求解を行う。
- (3) ATLAS-Eルーチン群：固有値問題を解く。

これらのルーチン群では、入力データも演算結果も補助記憶装置に格納されるため、入出力の効率が上がるように、上記のルーチン群内には、利用者のデータを適当なデータ構成に変更するためのユーティリティ・ルーチンも備えられている。

本報告書では、ATLASの各ルーチンについて、その機能とアルゴリズムの概説をする。各ルーチンの利用法については“ATLAS利用者マニュアル”²⁾で詳しく説明する。第2章では、使用されるファイルの形式について述べ、第3章から第5章までの各章で、前述した各ルーチン群についてアルゴリズムの概説と使用例、計算時間の例などについて述べる。第6章では、大次元行列演算を行う上で必要な計算機システムの機能について述べ、最後に第7章でまとめと今後の課題について述べる。

2. ファイル形式

ATLASの各ルーチンでは、図1に示すように入力データ、計算結果を格納する各種媒体、中間データを格納する作業領域はいずれも補助記憶装置（磁気ディスク、磁気テープなど）を使用している。したがってATLASを利用するには、まず演算の対象となるデータを補助記憶装置に格納し、さらに、出力結果や中間データを格納する補助記憶装置を確保しておかなければならない。以後、補助記憶装置のことを“ファイル”と呼び、入力、出力、中間データを格納するファイルを、それぞれ“入力ファイル”、“出力ファイル”、“中間ファイル”と呼ぶことにする。

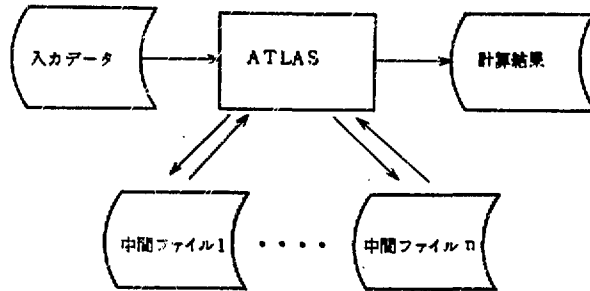


図1 ATLAS とファイル

2.1 データ構造

ATLASでは、記憶領域を節約するために、図2(a)に示す正方行列Aの帯内の要素のみをファイルに格納する。要素の格納順は、図2(b)に示すように各行毎に列方向に格納し、対角行列については、対角要素を含めて右上側の要素のみを格納する。ATLASでは、各ルーチンで使用するファイルへの入出力には、FORTRANの書式なし入出力文を使い、行列の1行、およびベクトルの全要素を1レコード（論理レコード）³⁾としている。

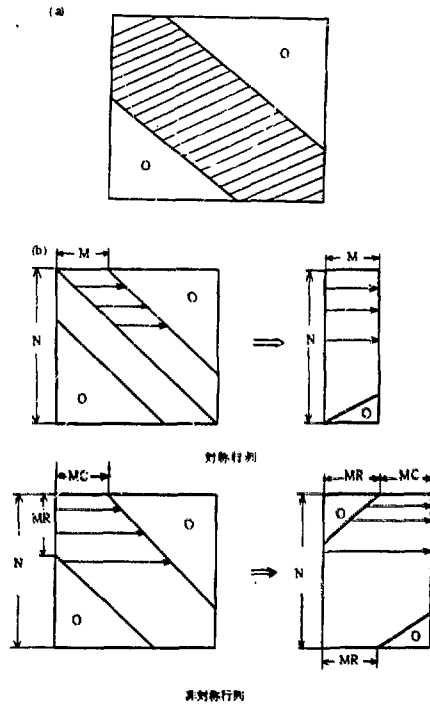


図2 ファイル形式

2.2 ファイル形式

前に述べたように、ATLASでは書式なし入出力文を使っているので、ファイルへのデータの格納の形式について特別の考慮をしなくとも演算はできる。しかしながら、大量のデータを取り扱う場合入出力の効率が計算の効率を高める上で重要な要因となるので、入出力の効率を上げるためには「ブロッキング」³⁾を行うべきである。

ブロッキングを行うには、ファイルの定義を行うジョブ制御文(TPDISK文、DFDISK文、FD文など)でレコード長(RCDSIZE)とブロック長(BLKSIZE)とを指定すればよい。RCDSIZEとBLKSIZEの決め方は次の通りである。

$$RCDSIZE = (i \cdot w + 2) \cdot \frac{9}{2} \text{ (バイト)}$$

$$BLKSIZE = n \cdot RCDSIZE \text{ (バイト)}$$

ここで、 i 、 w 、 n の決め方は次の通りである。

$i : i = 2$ 実数データの時、

$i = 4$ 複素数データの時、

w : ベクトルまたは行列の次元数.

n : 整数.

ファイルが磁気ディスク上にある場合には、ハードウェア上の制限からBLKSIZEは13023以下でなければならない。したがってベクトルや行列の次元数が大きくなりRCDSIZEが13023より大きくなる時には

RCDSIZE = 13023

BLKSIZE = 13023

とすればよい。

また、バッファ領域の大きさは、標準値が5 K語(20 Kバイト)になっているので、一度に使用するファイルのBLKSIZEが標準値より大きくなる時には、RUNコマンド(HRUN, HLIEDRUN)のパラメータとして"SIZE=XXXX"の形でK語単位にバッファ領域の大きさを指定しなければならない。

3. ATLAS-Aルーチン群

3.1 ATLAS-Aルーチン群の概要

このルーチン群は、行列間の加減乗算、行列とベクトル間の乗算、転置行列の生成を行うためのルーチン群である。また、これらのことが効率よく行えるように各種のユティリティ・ルーチンも備わっている。表1に各ルーチンの機能とルーチン名を示す。各ルーチンの詳しい説明はATLASのマニュアル²⁾で述べる。

表1 ATLAS-Aルーチン群に属するルーチン名。
この表でA, Bは行列, Vはベクトルを表わす。

ルーチン名 演算	対称行列			非対称行列	
	実	複素	エルミート	実	複素
加算 (A+B)	R P S	C P S	H P S	R P A S	C P A S
減算 (A-B)	R S S	C S S	H S S	R S A S	C S A S
乗算 (A×B)	R M S	C M S	H M S	R M A S	C M A S
乗算 (A×V)	R S M V E C	C S M V E C	H S M V E C	R A M V E C	C A M V E C
転置行列	R L T O L T	C L T O L T	H L T O L T	R A T R A N	C A T R A N
ブロッキング				R A F I L E	C A F I L E
行の要素数				L R E C O D	
上三角行列の行の要素数	L U R C D				
下三角行列の行の要素数	L L R C D				
半サイズを全サイズにする	R S T O A S	C S T O A S	H S T O A S		

3.2 ATLAS-Aルーチン群の使用例

このルーチン群を使用する場合、利用者はあらかじめ対象となる行列やベクトルをファイル上

に格納しておき、各ルーチンでは利用者が指定した大きさのデータをプログラム中の配列に読み込んで処理を行う。その際に必要な作業領域は、利用者のプログラム中で全作業領域分の1次元配列を確保しておけば、ATLASルーチン側で適宜に分割して使用する。

例1：対称行列とベクトルとの乗算

```

REAL* 8 W(7000)          .....作業領域の宣言
      .
      .
N = 1000                  .....行列の次元数
M = 500                   .....行列の帯幅
L = 5                     .....行列の読み込み行数
NIN1 = 10                 .....行列のファイル機番
NIN2 = 11                 .....ベクトルのファイル機番
NOUT = 13                 .....結果ベクトルのファイル機番
      .
      .
CALL RSMVEC(N, M, L, W(1), W(M*L+1), W(M*L+N+1), NIN1,
           NIN2, NOUT)
      .
      .

```

この結果、機番 10, 11 のファイルにそれぞれ格納されていた実対称行列と実ベクトルの乗算が行われ、結果が機番 13 のファイルに格納される。

4. ATLAS-Lルーチン群

4.1 ATLAS-Lルーチン群の概要

このルーチン群は、大次元連立一次方程式を解くためのもので、係数行列は、実対称、複素対称、エルミート対称、実非対称、複素非対称のいずれでもよい。連立一次方程式の解法には、大別して直接法と反復法があるが、ATLAS-Lルーチン群では直接法を採用し、帯内に非零要素が多い場合を考慮して直接法の中のブロック消去法⁴⁾を採用している。

連立一次方程式

$$Ax = b \quad (1)$$

を解くには、行列Aの逆行列 A^{-1} を求めればよいが、Aが帯行列の場合、帯である性質を保存しつつ A^{-1} を求めることを考える。そのためにAが対称行列の場合、Aを

$$A = LDL^T \quad (2)$$

の形に分解する(LDL^T分解)。ここでLは対角要素が1で下半分のみに非零要素がある行列、L^Tはその転置行列である。また、Dは対角行列である。行列Aがエルミート行列の場合は(2)の分解においてL^TをL^H(エルミート共役行列)にすればよい。このようにAを分解しておくこと、方程式(1)を解く過程は次の3つに分割される。

1) $Lz = b$ を解く(前進代入)

- 2) $Dy = z$ を解く(除算)
- 3) $L^T x = y$ を解く(後退代入)

このようにすれば、Aが帯行列の時はLも帯行列となるから、帯である性質を保存して方程式(1)を解くことができる。Aが非対称行列の時は(2)の分解の代わりに

$$A = LU \tag{3}$$

を使えばよい。ここでUは上半分のみに非零要素がある行列である(LU分解)*。

ATLASでは、分解(2)および(3)を行う際に、入出力の効率を上げるために、行列Aを図3に示すようにいくつかのブロックに分割している。このブロックの数だけファイルを必要とするが、利用者のデータからブロック化されたデータをつくりだすためのユーティリティ・ルーチンが備えられ、利用者への便宜をはかっている。

ATLAS-L ルーチン群に含まれるルーチンは表2に示すとおりである。各ルーチンの詳しい説明は、ATLASのマニュアル²⁾で述べらる。

表2 ATLAS-Lルーチン群に属するルーチン名

(1) 対称行列

演算 \ ルーチン名	実	複 素	エルミート
ブロック化	RARRNG	CARRNG	HARRNG
ゼロクリアー	RCSUBS	CCSUBS	HCSUBS
LDL ^H 分解	RTRIDE	CTRIDE	HTRIDE
前進代入	RLOWER	CLOWER	HLOWER
後退代入	RUPPER	CUPPER	CUPPER

(2) 非対称行列

演算 \ ルーチン名	実	複 素
L U 分解	RALUDE	CALUDE
前進代入	RAFSUB	CAFSUB
逆順ファイル作成	RCDINV	CRCDIV
後退代入	RABSUB	CABSUB

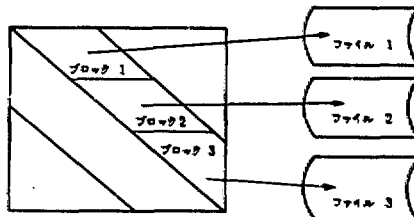


図3 ブロック分割

* LDL^T分解とLU分解の基本的アルゴリズムについては例えば文献4)を参照のこと。

4.2 ATLAS-L ルーチン群の使用例

例2: 複素対称行列係数の連立一次方程式の解法

ATLAS-L ルーチン群を使って大次元帯行列係数の連立一次方程式を解くには、次のようにすればよい。

```

COMPLEX * 16  A ( 22000 ), SUBS ( 6000 ), MT ( 4 )
M = 300          .....帯幅
N = 1478        .....次元数
NBF = 4         .....ブロック数
}
その他のパラメータ
}
.....
CALL  CARRNG ( ..... ) .....データのブロック化
CALL  CTRIDE ( ..... ) .....係数行列の分解
CALL  CCSUBS ( ..... ) .....配列のクリアー
CALL  CLOWER ( ..... ) .....前進代入**
CALL  CUPPER ( ..... ) .....後退代入
}
.....

```

この例からわかるように、ATLAS-L ルーチン群では、連立一次方程式を解く過程をいくつかのサブルーチンに分割している。これは、係数行列が同じで定数ベクトルが違う場合に前進代入と後退代入のみを行って計算の効率を良くするためである。

参考までにATLAS-L ルーチン群を使って複素対称行列係数の連立一次方程式を解いた際の計算時間、使用記憶の大きさなどを表3に示す。この結果はFORTRANの最適化レベルOPT0でコンパイルした目的プログラムを使って得られたものであるが、最適化レベルOPT2で行うと計算時間が10~30%短くてすむ。

表3 連立一次方程式の求解のテストケース。時間はFACOM230/75を使った結果。

次元	帯幅	ブロック数	記憶容量 (kW)	CPU時間 (sec)	CORE時間 (sec)
322	66	1	85	29	100
322	66	2	70	29	100
1478	300	4	180	2307	3810

** ATLAS-L ルーチン群では、前進代入と除算を同時に行っている。

5. ATLAS-Eルーチン群

5.1 ATLAS-Eルーチン群の概要

このルーチン群は、大次元帯行列の固有値問題を解くためのルーチン群である。プラズマの流体シミュレーションには、標準的な固有値問題

$$Ax = \lambda x \quad (4)$$

よりも、一般固有値問題

$$Ax = \lambda Bx \quad (5)$$

の方がよく現われる。そのため、ATLAS-Eルーチン群で取扱うのは、一般固有値問題であるが、わずかな修正で標準固有値問題用のルーチン群も作成できる。

ATLAS-Eルーチン群では、一般固有値問題(5)の係数Aが実対称またはエルミート対称、Bが正定値実対称行列である場合を取り扱い、A、Bが帯行列であるという性質を積極的に利用するために、解法としてブロック逆反復法⁵⁾を採用している。

よく知られているように、逆反復法は連立一次方程式をくりかえし解き固有ベクトルと固有値を求める解法で、ブロック逆反復法は、いくつかのベクトルをまとめて逆反復する方法である。逆反復法で固有ベクトルを求めるには、連立一次方程式

$$Ax_{k+1}^j = Bx_k^j \quad (6)$$

を解けばよい。ここでkは逐次近次の番号、jは同時に求める固有ベクトルの番号である。求めた固有ベクトル群 $\{x_k^j\}$ が互いに独立になるようにするには、反復の過程で正規直交化条件

$$(x_k^j, Bx_k^l) = \delta_{jl} \quad (7)$$

を満たすようにすればよい。行列A、Bが帯行列の場合、“帯である性質”を保存して連立一次方程式を解くにはATLAS-Lルーチン群で用いた方法を使えばよい。まず、A、Bを次のように分解する。

$$A = LDL^H \quad (8)$$

$$B = R^T R \quad (9)$$

ここで、Lは下半分のみに非零要素がある行列で、 L^H はそのエルミート共役行列である。行列Aが実対称行列の時は L^H は転置行列 L^T になる。また、Bは正定値実対称行列であるから、上半分行列Rとその転置行列 R^T とに分解できる。

方程式(6)を条件(7)の下で解く過程は次の6つに分解される。

$$1. \quad u_k^j = R x_k^j \quad (R \text{ の乗算}) \quad (8)$$

$$2. \quad (u_k^l, u_k^j) = \delta_{jl} \quad (\text{正規直交化}) \quad (9)$$

$$3. \quad v_k^j = R^T u_k^j \quad (R^T \text{ の乗算}) \quad (10)$$

$$4. Ly_{k+1}^j = v_k^j \quad (\text{前進代入}) \quad (11)$$

$$5. Dw_{k+1}^j = y_{k+1}^j \quad (\text{除算}) \quad (12)$$

$$6. L^H x_{k+1}^j = w_{k+1}^j \quad (\text{後退代入}) \quad (13)$$

収束の判定は、ステップ2の後でベクトル u_{k+1}^j と u_k^j を使い

$$\max |u_{k+1}^j - u_k^j| < \epsilon \quad (14)$$

で行い、収束条件(14)が満された後、固有値をレイリー商を使って計算する。

$$\begin{aligned} \lambda_j &= \frac{(x_{k+1}^j, Ax_{k+1}^j)}{(x_{k+1}^j, Bx_{k+1}^j)} \\ &= \frac{(x_{k+1}^j, Bx_k^j)}{(u_{k+1}^j, u_{k+1}^j)} \\ &= (x_{k+1}^j, Bx_k^j) \end{aligned} \quad (15)$$

ATLAS-E ルーチン群には、ブロック逆反復法を使って一般固有値問題を解くルーチンとして "EHYMNIA" と "GEPsi" とがある。前者は、Gruber によって作成され CPC ライブラリーに登録された "HYMNIA"⁶⁾ を拡張したもので、一方、後者は我々が独自に開発したルーチンである。EHYMNIA も GEPsi も大筋は同じであるが、最後に固有値を計算する部分が多少異っている。EHYMNIA はブロック逆反復法をそのままプログラムにしたルーチンであるが、GEPsi では、 p 個のベクトル (p は求めるべき固有値の個数よりも大きくとっておく)

$x_{k+1}^1, \dots, x_{k+1}^p$ を使い、 $p \times p$ の行列

$$Y_{k+1} = \begin{pmatrix} 1^T \\ x_{k+1}^1 \\ \vdots \\ x_{k+1}^p \\ x_{k+1}^T \end{pmatrix} (x_{k+1}^1, \dots, x_{k+1}^p) \quad (16)$$

の固有値を求めることによって、固有値および固有ベクトルを計算している。⁵⁾

EHYMNIA, GEPsi の使い方について詳しい説明は ATLAS のマニュアルで述べる。

5.2 ATLAS-E ルーチン群の使用例

例3：一般固有値問題の解法

プラズマの電磁流体力学的安定性を変分法を使って解析する際には、一般固有値問題

$$Ax = \lambda Bx \quad (17)$$

を解く必要がある。ここで、 A はエルミート行列、 B は実対称正定値行列である。この問題を EHYMNIA ルーチンを使って解くには次のようにすればよい。

DIMENSION WRK1 (45150), WRKBUF (300)

COMPLEX * 16 WRK1, WRKBUF

```

.....
N= 1478                    .....次元数
M= 300                    .....帯幅
.....
CALL INIT (.....)        .....ファイル機番のデフォルト値設定
CALL CCLEAR(.....)        .....作業域のクリアー
CALL HESIVI(.....)        .....固有値, 固有ベクトル計算
.....

```

参考までに演算時間を表 4 に示す。現在のところ EHYMNIA, GEPS I 共に改良の余地があり、特に EHYMNIA では、連立一次方程式を解く部分を改良することで大次元行列の場合に約 20%程度演算時間が速くなることが予想される。

表 4 一般固有値問題の求解のテストケース。
時間は FACOM 230/75 を使った結果。

次元	帯幅	固有値の個数	くりかえし数	記憶容量 (kW)	CPU時間 (sec)	CORE時間 (sec)
322	66	1	19	67	140	798
1128	228	1	19	172	3143	5741
1478	300	1	19	246	6546	9461

6. 計算機システムとの関連

大次元行列演算の場合、計算が必ずしも 1つのジョブで完結するとは限らない。演算時間(CPU時間)が何時間にもなる時には、途中でジョブを打ち切って再開始処理をしなければならない。その際、中間のデータをファイルに出力しておかなければならないが、このデータは短時間(例えば 1日程度)の間保存されなければならない。再開始用の中間データを保存するファイルは、比較的低速度の装置でよいが、記憶容量が大きくなければならない。例えば、次元数 2000、帯幅 500 の複素係数の一般固有値問題の場合、倍長演算を行うと

$$A \text{ のサイズ} = 2000 \times 500 \times 16 = 16 \text{ MB}$$

$$B \text{ のサイズ} = 2000 \times 500 \times 8 = 8 \text{ MB}$$

の記憶領域が必要となる。また、A、B の分解の途中でジョブを打ち切る必要がある場合には、同程度の容量のファイルをいくつか作成しなければならない。現在のところ、再開始用のファイルとして磁気テープを使っているが、いくつかの磁気テープ・ファイルを使うことは、利用者にとって負担になり、できれば磁気ディスクと同様に使用できる大容量の記憶装置があると便利である。この必要性は、次元数によって中間ファイルの個数が異なるプログラム(例えば ATLAS-Lルーチン群の一部)の場合、更に高くなる。

一方、ソフトウェアでは、ファイルの確保や定義がプログラム中から行える機能が必要である。中間ファイルに対しては、入出力の効率を上げるために、行列の大きさによってレコード長やブロック長を変えなければならない。これをいくつかのファイルについて行うことは利用者にとってかなり負担になることである。また、行列の大きさによって中間ファイルの個数が異なる場合、何個のファイル定義文を使うか考えるのも利用者にとっては負担になることである。したがって、ATLASルーチン群の中で行列の大きさから中間ファイルの個数やレコード長、ブロック長などを算出し、その結果に基づいてファイルの確保や定義をしておけば利用者は中間作業用のファイルをほとんど意識しないですむ。

7. まとめと今後の課題

今回のATLASの開発では、プラズマの流体シミュレーションに現われる大次元帯行列を対象とする行列演算用サブルーチンを作成した。前にも述べたように、我々が現在念頭においている行列は、帯の中に零要素が非常に少い行列で、この型の行列演算に関しては、今回開発されたルーチン群は、性能の上でほぼ満足すべきものであった。しかしながら、核融合研究における行列演算全般を見渡すと、構造解析に現われる粗行列（帯中に零要素が多く存在する行列）に対して有効なルーチン群の開発も必要となる。更につきすすんで考えると、利用者のデータから行列の構造を読み取り、適当なアルゴリズムを決定する、いわば、“大次元行列演算ソフトウェアシステム”の開発も必要となろう。我々は、現在、行列の性質とそれに適したアルゴリズムについて精力的な研究調査を行っている。

謝 辞

このルーチン群の開発に当って豊富な御助言と御支援を下された計算センター外来研究員の田子幹男氏に感謝致します。また、EHYMNIAとGEPSIとはそれぞれ、富士通株式会社、日本アイ・ビー・エム株式会社に発注して製作したもので、その際御助力いただいた佐々木幹夫氏（富士通）と棚町芳弘氏（日本IBM）に感謝致します。さらに、森茂部長と田中正俊室長の御激励は開発期間全般にわたって我々にとって大きな支えでありました。

参 考 文 献

- 1) Takeda T., Tsunematsu T., Kurita G., and Tanaka M. : " Project TRITON: in preparation .
- 2) 常松俊秀, 竹田辰興, 藤田恵一, 松浦俊彦, 田原伸夫 : " ATLAS利用手引書 ", in preparation .
- 3) 富士通 : FACOM 230 M-VII FORTRAN VI-H 使用手引書, 富士通株式会社 (1976)
- 4) 戸川隼人 : " マトリクスの数値計算 ", オーム社 (1971) .
- 5) Bathe K. J. and Wilson E. L. : International Journal for Numerical Methods in Engineering , 6 , (1973) 213
- 6) Gruber R. : Comput . Phys . Commun . 10 (1975) 30