

[03_04]九州大学大型計算機センター広報 : 3(4)

<https://doi.org/10.15017/1467969>

出版情報 : 九州大学大型計算機センター広報. 3 (4), pp.1-56, 1970-08-26. 九州大学大型計算機センター
バージョン :
権利関係 :



ライブラリプログラムの紹介

No. 251 F2/QU/F/HER4

EIGENVALUES AND EIGENVECTORS OF HERMITIAN MATRIX

複素共役行列 (HERMITE 行列) の固有値および固有ベクトル

作 成	作成者 国 宗 真	作成年月日 昭和45年6月17日
形 式	a、コンプリートプログラム d、手続き	⑥ サブルーチン c、関数 e、関数手続き
使 用 言 語	④、FORTRAN d、PL/I	b、ALGOL c、FASP e、その他
使 用 機 種	FACOM 230-60	
使用メモリ数	a、コア()K語 b、ディスクバック()K語 c、その他()	
使用機器構成	a、カードリーダー ⑥、ラインプリンタ c、カードパンチ d、紙テープリーダー e、紙テープパンチ f、磁気テープ()ユニット g、ディスクバック h、その他	
利用者の義務	a、プログラム名と作成者名を明記する ⑥、明記する必要はない	
公 表	④ ソースプログラムを公表する。 b、ソースプログラムの公表は一定期間保留する(年 月 日まで)	

§ 1. 概 要

1.1 目 的

複素共役行列の固有値と固有ベクトルを求める。

1.2 計算方法

与えられた Hermite 行列に段階的にユニタール変換を行なって、元の行列を対角化する。
この結果の対角要素が固有値であり、対角化するために必要であったユニタール行列の各行が、固有ベクトルを与える。

参考文献： 宇野利雄：計算機のための数値計算 p. 62 (朝倉書店 S. 43)

山口、森口、一松：電子計算機のための数値計算法 I p. 187 (培風館)

§ 2. 使用法

2.1 呼び出し方法

CALL HER4(K, N, EPS, IVEC, ARI, ERI, ILL)

2.2 パラメータ

ARI 複素数型配列名。大きさ $K \times K$ の 2 次元配列。ARI(K, K)

入力行列である Hermite 行列を与える。

$$ARI(I, J) = CONJG(ARI(J, I)) \quad (I, J = 1, 2, \dots, N)$$

サブルーチンから戻る時、ARIの内容はこわされ、結果の固有値がARI(J, J) (J=1, 2, ..., N)にセットされる。

ERI 複素数型配列名。大きさK×Kの2次元配列・ERI(K, K)結果の固有ベクトルがセットされる。

ARI(J, J)に対する固有ベクトルは、ERI(J, I) (I=1, 2, ..., N)で与えられる。

K 配列ARI, ERIの大きさを与える。K ≥ N

整数型変数名又は整数。

N 入力行列ARIの演算対象となる小行列の元数を与える。

整数型変数名又は整数。(N ≤ 100)

EPS 実数型変数名又は実定数。EPS > 0.0

収束判定値を与える。

収束判定は $\max(|\text{非対角要素}|) < \text{EPS}$ で行なう。

IVEC 整数型変数名又は整数。

解の与え方を指示する。

IVEC = 1 固有値だけ求める。

IUEC = 1 固有値及び固有ベクトルを求める。

ILL 整数型変数名。

入力パラメータが制限条件外で計算が行えない時、ILL=1として、その理由を印刷する。正常に解が求まった時 ILL=0 が与えられる。

注1) EPSの標準値は 10^{-5} 位

注2) 演算後 ARIの内容はこわされる。

2.3 エラー処理

ILL CONDITION になった時 (即ち、ILL=1) 以下のメッセージをプリントして、呼び出しプログラムに戻る。

(i) ARIがHermite 行列でない時、

[ILL=1 HERMITIAN DE NAI

KISOKUNI HANSITA GYOO TO RETU ARI(I, J)=(,)

(ii) EPS ≤ 0の時

ARI(J, I)=(,)]

「 I L L = 1 E P S ≤ 0 * * * * E P S = * * * * 」

(iii) $K < N$ の時

「 I L L = 1 $K < N$ * * * * $K = \dots\dots\dots$, $N = \dots\dots\dots$ * * * * 」

(iv) $N > 100$ の時

「 I L L = 1 $N > 100$ (SUBROUTINE NO HAIRETU MAXCOL
(100), NO DIMENSION O HUYASUKOTO * * * * $N = \dots\dots\dots$ * * * * 」

(v) $N \leq 0$ の時

「 I L L = 1 * * * * $N =$ * * * * 」

(vi) サブルーチン内の D O ループを $3 N^2$ 回繰り返しても収斂しない時、

「 * * * * H E R M T 3 W A S H U U R E N S I N A I . G U R U G U R U
M A W A T T A K A I S U U W A \dots\dots\dots * * * * 」

2.4 備 考

(i) このサブルーチンでは、以下の組込み関数、基本外部関数を使用している。

C A B S , C O N J G , R E A L , S Q R T , S I G N

(ii) 所要時間

一概には言えないが、1例として9行9列の計算で約0.7秒。

No. 252 I 9 / QU / F / INLIST

INPUT, NON-FORMAT

LIST型入力

作 成	作成者 上田耕平	作成年月日 昭和45年5月15日
形 式	a、コンプリートプログラム d、手続き	(b)、サブルーチン e、関数手続き c、関数
使 用 言 語	(a)、FORTRAN d、PL/I	b、ALGOL e、その他 c、FASP
使 用 機 種	FACOM 230-60	
使用メモリ数	a、コア(1) K語	b、ディスクバック() K語 c、その他()
使用機器構成	(a)、カードリーダー d、紙テープリーダー f、磁気テープ() ユニット g、ディスクバック h、その他 b、ラインプリンタ e、紙テープパンチ c、カードパンチ	
利用者の義務	a、プログラム名と作成者名を明記する (b) 明記する必要はない	
公 表	(a)、ソースプログラムを公表する。 b、ソースプログラムの公表は一定期間保留する(年 月 日まで)	

§ 1. 概 要

FORTRANのプログラムにおいて、FORMATなしでデータ入力を行なう。データの区切りは、,(コンマ)又は1桁以上の空白である。

プログラムは、FORTRANで書かれたエレメントINPUT1と、FASPで書かれたエレメントINPUT2の2つのサブルーチン副プログラムから成り、エンタリー名は、INPUTR, INPUTD, INPUTI, INPUTA, INPUTB, INPUTK及びALREADの7つで、前6個はそれぞれ引数の型に対応して入力を行ない、ALREADは、入力方法の変更を行なう。

§ 2. 使用法

2.1 呼び出し方法

GALL INPUTR(R₁, R₂, ..., R_n) (1)
CALL INPUTD(D₁, D₂, ..., D_n) (2)
CALL INPUTI(I₁, I₂, ..., I_n) (3)
CALL INPUTA(A, M₁, M₂, M₃) (4)
CALL INPUTB(B, M₁, M₂, M₃) (5)
CALL INPUTK(K, M₁, M₂, M₃) (6)

CALL ALREAD(L) (7)

2.2 パラメータ

R_j 単精度実数型変数または配列要素名 (for $1 \leq j \leq 20$)

D_j 倍精度実数型変数または配列要素名 (for $1 \leq j \leq 20$)

I_j 整数型変数または配列要素名 (for $1 \leq j \leq 20$)

A 単精度実数型配列名 (1次元)

B 倍精度実数型配列名 (1次元)

K 整数型配列名 (1次元)

$\left. \begin{array}{l} L \\ M_1 \\ M_2 \\ M_3 \end{array} \right\}$ 整数または既に値を持った整変数名

(1), (2), (3)は引数の順序に従ってリスト型入力を行なう。

(4), (5), (6)は、READ (,) (A(M), M=M₁, M₂, M₃)に相当して、1次元配列名Aに添字Mの初期値をM₁として、きざみ巾M₃で、M₂を越えない範囲で入力を行なう。

(7)については後述。

2.3 入力データ形式

- (i) データは1テキスト(カード入力の場合は1枚のカード)1~72桁として、1桁以上の空白または、(コンマ)で区切ったもので、FORTRANで許されるデータ形式はすべて良く、実変数に読み込む場合に整数型、整変数に読み込む場合に実数型の記法を用いてもよい。但し、1つのデータが2つのテキストにまたがってはならない。(この場合、72桁で区切ったものとして、2つのデータとみなす。)新たに入力用サブルーチンが呼び出された場合は、通常のFORTRAN入力(READ文)と同様、新しいテキストから入力を始める。

(注) 空白をデータの区切りとみなすため、たとえば15.0に対して、15E01は許されない。

(ii) ALREAD(L)

通常、リスト型入力は新しい入力ごとに(新たにサブルーチンがCALLされるたび)新しいテキストから入力を始めるが、CALL ALREAD(L) (L≠0)を用いることによって、ALGOLと同様の形で、既に読み込んだテキストに余りのデータがあれば、そのデータから入力を行なうことができ、またCALL ALREAD(L) (L=0)によって元に戻すことができる。

(例)

データカードに

1 2 , 1.4 , 1.6 , 1.7 1.8 1.9 , 2 (テキスト1)

3 , 4.1 , 5.1 , 6 , 7.5 , 8 (テキスト2)

8.5 9 9.5 , 10 (テキスト3)

がパンチしてある時

CALL INPUTR(A,B,C)

CALL INPUTA(D,1,5,1)

CALL INPUTI(J,K)

を実行すると、入力後は、

A=1.2 , B=1.4 , C=1.6 , D(1)=3.0 , D(2)=4.1 , D(3)=5.1 ,

D(4)=6.0 , D(5)=7.5 , J=8 , K=9

となり、また、

CALL ALREAD(3)

CALL INPUTR(A,B,C)

CALL INPUTA(D,1,5)

CALL ALREAD(0)

CALL INPUTI(J,K)

を実行すると、入力後は、

A=1.2 , B=1.4 , C=1.6 , D(1)=1.7 , D(2)=1.8 , D(3)=1.9 , D(4)=2.0 ,

D(5)=3.0 , J=8 , K=9

となる。

なお、CALL ALREAD(L)を用いる時は、特に次の点に注意する必要がある。

(i) CALL INPUTR(R₁,R₂,...).....(1)

CALL ALREAD(3)

CALL INPUTR(R'₁,R'₂,...).....(2)

上の例の場合、(1)で入力を行なったテキストに余りのデータがある場合は、(2)の入力では、

(1)の残りのデータから入力を開始する。

(ii) CALL ALREAD(2)

CALL INPUTR(A,B,C).....(1)

READ (5,5)D,E,F(2)

5 FORMAT(3F5.0)

CALL INPUTR(P,Q,R)(3)

この場合、データカードが

1.0, 2.0, 3.0, 4.0, 5.0

1 1.0 1 2.0 1 3.0 1 4.0 1 5.0

2 1, 2 2, 2 3, 2 4, 2 5

となっていると、

A=1.0, B=2.0, C=3.0, D=1 1.0, E=1 2.0, F=1 3.0, P=4.0,

Q=5.0, R=2 1.0

となる。この様に、(1)で入力を行なったテキストに余りのデータがあると(3)の入力において、(1)の入力の残りから始まり、(2)の入力テキストを除いて、新しいテキストから入力を行なう。

原 稿 募 集

利用者の皆さんの「もの申す」という手段として、センターでは「問合せ票」を準備しておりますが、センターを利用して見て受けられた感じなどを「利用者の声」の欄に投稿してください。

その他、書かれたプログラムでライブラリとして役に立ちそうなものなども投稿してください。歓迎いたします。

投稿先 8 1 2 福岡市箱崎

九州大学大型計算機センター広報編集掛

No. 253 F4/QU/F/SWPMLE

INVERSE MATRIX AND LINEAR EQUATION, USING DISK OR MT

逆行列及び、連立一次方程式 補助記憶装置使用

作 成	作成者 上田耕平	作成年月日 昭和45年2月 日
形 式	a、コンプリートプログラム d、手続き	b、サブルーチン c、関数 e、関数手続き
使 用 言 語	a、FORTRAN d、PL/I	b、ALGOL c、FASP e、その他

使 用 機 種	FACOM 230-60
使用メモリ数	a、コア(1.5)K語 b、ディスクパック()K語 c、その他()
使用機器構成	a、カードリーダー b、ラインプリンタ c、カードパンチ d、紙テープリーダー e、紙テープパンチ f、磁気テープ(2)ユニット (g又はf) g、ディスクパック h、その他
利用者の義務	a、プログラム名と作成者名を明記する b、明記する必要はない
公 表	a、ソースプログラムを公表する b、ソースプログラムの公表は一定期間保留する(年 月 日まで)

§ 1. 概 要

1.1 目 的

係数が同じm組の連立一次方程式の解や逆行列を求める場合で、主記憶装置のメモリ数だけでは不足する様な大きなものについて、補助記憶装置を並用して計算する。

プログラムは、エレメント SWPMD1 から成り、 サブルーチン副プログラムの形式である。

プログラムのエントリは、SWPMD1, SWPMD2, SWPMD3 の3つである。

連立一次方程式

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{1,n+1} & a_{1,n+2} & \cdots & a_{1,n+m} \\ a_{2,n+1} & a_{2,n+2} & \cdots & a_{2,n+m} \\ \vdots & & & \\ a_{n,n+1} & a_{n,n+2} & \cdots & a_{n,n+m} \end{bmatrix}$$

逆行列の場合は $m=n$ かつ、 $a_{in+j} = \begin{cases} 1 & (i=j) \\ 0 & (i \neq j) \end{cases}$

1.2 計算方法

Gauss - Jordan の掃き出し法を用い、Pivot は、列方向に絶対値最大のものを選ぶ。

§ 2. 使用方法

2.1 呼び出し方法

```
CALL SWPMD1(N,M,PA,PB,PC,PD,WA,M1,WB,M2,NS,
            EPS,KS1,KS2)
```

```
CALL SWPMD2(PA)
```

```
CALL SWPMD3(S,DS,ISS,ILL)
```

2.2 パラメータ

N 整数型定数または、変数名。

連立一次方程式の元数 n を与える。

NM 整数型定数または、変数名。

同時に求める解の組数 + 連立一次方程式の元数を与える。($n + m$)

M1 整数型定数または、変数名。

配列WAの宣言時の大きさを与える。

M2 整数型定数または、変数名。

配列WBの宣言時の大きさを与える。

NS 整数型定数または、変数名。

配列Sまたは、SDの宣言時の第1添字の大きさ、即ち、

$\text{DIMENSION } S(NS, M), M \geq M \text{ の } NS. NS \geq N.$

ISS 整数型定数または、変数名。

結果の取り出し方を指定する。ISS > 0

ISS = 1 解をSに引き渡す。(Sは単精度)

ISS = 2 解をSDに引き渡す。(SDは倍精度)

ISS ≥ 3 解はWBの中にそのまま置く。

PA }
PB } 倍精度実数型配列名。
PC } 大きさ N + M 以上を宣言しておく。
PD }

WA } 倍精度実数型配列名。
WB } 大きさは取り得るだけ大きく宣言しておく。

SD 倍精度実数型配列名 } 解の結果がセットされる。

S 単精度実数型配列名 } ISS で指定した方を必ず宣言しておく。

必要のない方はdummy でよい。

ILL 整数型変数名。

サブルーチンから戻った時の状態がセットされる。

正常に解が求まった時、0 がセットされる。

EPS 単精度実数型変数名または、実定数。

ゼロ判定値を与える。Pivot の絶対値がこれより小さくなった場合、ILL=1

として呼び出しプログラムに戻る。

KS1 } 整数または、整数型変数名。

KS2 } 補助記憶装置への入出力に用いる機番を指定する。

KS1≡KS2 かつ、これらは1, 2, 3, 4, 8 のいずれかでなくてはならない。

また、実行中に、これらのファイルを指定するコントロールカードを必要とする。

2.3 ジョブ制御文

このサブルーチンを呼び出す場合は、標準的には、以下の様なコントロールカードが必要である。

¥NO

¥QJOB

¥FORTRAN

[CALL SWPMD1(N, , KS1, KS2)

¥LIEDRUN

[データ

¥F.WORK n_1

¥F.WORK n_2

¥JEND

(ここで n_1 は1, 2, 3, 4, 8 のいずれかで $n_1 = KS1$, $n_2 = KS2$ でなければならない。)

2.4

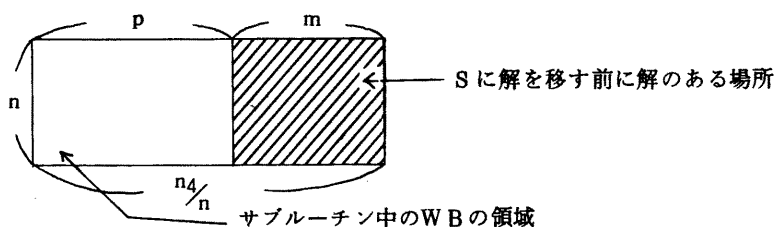
(1) 呼び出しプログラムの宣言文

DOUBLE PRECISION PA(n_a), PB(n_b), PC(n_c), PD(n_d),

$WA(n_1), WB(n_2), SD(n_3, n_4)$ において、 $n_a, n_b, n_c, n_d \geq n+m$,
 $n_1 \geq n, n_4 \geq m$, かつ、 n_1, n_2 はできる限り大きくとらなければならない。 n_1, n_2
 を大きくとるために、次の EQUIVALENCE 文を用いてよい。

EQUIVALENCE (PA, SD, WB), (PB, WB(n')), (PC, WB(n'')),
 (PD, WB(n''')), (WA, WB(n''''))

但し、 $n' = n+m+1, n'' = n' + n+m, n''' = n'' + n+m, n'''' = n''' + n+m$
 n_3 の値が n と異なる時は、次の事に注意する必要がある。



WB から S への解の転送は、 $S(i, j) = (WB(i, p+j), j=1, m),$
 $i=1, n)$ の形を用いている。

(ii) 各呼び出し文の使い方と意味

(a) CALL SWPMD1 ()

元数、組数、ゼロ判定値、配列の大きさ、領域等をセットしておく。

SWPMD2, SWPMD3 の呼び出しに先だって呼び出されねばならない。

(b) CALL SWPMD2 (PA)

係数を一行ずつ ($a_{i1}, a_{i2}, \dots, a_{in}, a_{in+1}, \dots, a_{in+m}$)
 転送する。

従って、配列 PA には

$PA(1) = a_{i1}, PA(2) = a_{i2}, \dots, PA(n+m) = a_{in+m}$

の形で係数の値を入れておかなければならない。

(c) CALL SWPMD3 (S, SD, ISS, ILL)

計算を実行し、return condition および、正常な場合には解を S 又は SD にセ
 ットする。

(iii) 使用方法例 ($n=200, m=50$)

DOUBLE PRECISION PA(250), PB(250), PC(250), WA(27000),
 WB(28000).

DIMENSION S(200, 50).

EQUIVALENCE(S, WB, PA), (WB, (251), PB), (WB(501), PC),

```
(WB(751),PD),1(WB(1001),WA)
CALL SWPMD1(200,250,PA,PB,PC,PD,WA,27000,WB,28000,
200,1.0E-9,3,4)
DO 10 I=1,200
DO 20 J=1,250
20 PA(J)=
    CALL SWPMD2(PA)
```

10 CONTINUE

```
CALL SWPMD3(S,S,1,ILL)
```

```
IF(ILL,NE,0)STOP
```

```
END
```

(注) CALL SWPMD2()から、CALL SWPMD3()までの間で
PC,PDに値を入れてはならない。

(Ⅳ) 備 考

このサブルーチンは、補助記憶装置を使用するため時間がかかるので、主記憶上で計算できる場合は、なるべく別の方法を使う方がよい。