

# 3 G - 7 行列演算用言語 LAMAX-S の開発 - 言語の特徴とその応用範囲 -

井上 美明<sup>1</sup> 田中 和彦<sup>1</sup> 内田 智史<sup>2</sup>  
八巻 直一<sup>1</sup> 本郷 茂<sup>3</sup> 間野 浩太郎<sup>2</sup>

1) システム計画研究所      2) 青山学院大学理工学部      3) 専修大学経営学部

## 1 はじめに

我々は、現在、スーパーコンピュータを効率良く利用できるように工夫された行列演算用言語、LAMAX-S (a Language for MAtRiX calculation on Supercomputer) を開発している。LAMAX-S は、前身の LAMAX-E<sup>1</sup> の流れをくむ FORTRAN のプリコンパイラ言語であり、強力な行列演算記述能力を持っている。さらに、LAMAX-S はその最適化機能により、チューニング作業にわずらわされることなくスーパーコンピュータの性能を十分引き出すことができる。

言語仕様の点からみれば、LAMAX-S は、FORTRAN に次の機能が追加されたと考えてよい。

### 各種の行列構造 (structure) の導入

例： 密行列・対角行列・バンド行列・スパース行列など  
さまざまな行列演算機能

例： 四則演算、連立方程式の求解、LU 分解など

## 2 LAMAX-S 文法の概要

LAMAX-S の文法は、基本的には FORTRAN と同じである。ここでは、行列の記述に関する部分をはじめとして概要を紹介することにする。

### 2.1 LAMAX-S における行列属性

一般に FORTRAN には、データ型 (type)、類 (class) の区別がある。

データ型 : INTEGER, REAL, COMPLEX, CHARACTER など

類 : VARIABLE, ARRAY, FUNCTION, SUBROUTINE など

LAMAX-S では、これに加えて行列属性を持つ。行列属性には、行列の構造、要素の密度、対称性がある。その一覧を表 1 に示す。

表 1 LAMAX-S の行列属性

構造	要素の密度	対称性
行列 (m × n)		
正方行列		
列ベクトル	密行列	対称行列
行ベクトル		
帯行列	スパース行列	非対称行列
対角行列		
3 重対角行列		
上三角行列		
下三角行列		

行列属性は、ここに示した構造、要素の密度、対称性をそれぞれ指定したものになる。下線で示したものが既定値となっている。

### 2.2 行列の宣言

行列の宣言は、次の形式に従う。

要素のデータ型 変数名: 構造 [構造に関する指定値  
{, 要素の密度 [その記述]} {, 対称性} ]

たとえば、要素が倍精度で、上バンド幅が 3、下バンド幅が 2 の 100 × 100 のバンド行列 X を宣言したければ、

```
REAL*8 X: BAND [100, 3, 2]
```

とする。この場合には、密行列、非対称性が仮定される。以下に若干の例を示す。また、各構造の宣言構文を表 2, 3, 4 に示す。

```
INTEGER A: MATRIX [10, 20]
```

要素が整数型で 10 × 20 の密行列

```
REAL*16 B: VECTOR [10]
```

4 倍精度型の要素を 10 個持つベクトル

```
COMPLEX C: SQUARE [1000, SPARSE [200]]
```

要素が複素数型で、1000 × 1000 の正方  
スパース行列、その中に最大 200 個の非  
ゼロ要素を含む

表 2 構造名とその宣言法

構造名	宣言法
密行列	MATRIX [行数, 列数]
列ベクトル	VECTOR [要素数]
行ベクトル	RVECTOR [要素数]
帯行列	BAND [行数 (= 列数), 上バンド幅, 下バンド幅]
対角行列	DIAGONAL [行数 (= 列数)]
3 重対角行列	DIAG_3 [行数 (= 列数)]
上三角行列	UPPER.TR [行数 (= 列数)]
下三角行列	LOWER.TR [行数 (= 列数)]

表 3 要素の密度、対称性の指定

密行列: DENSITY	スパース行列: SPARSE [スパース記述]
対称行列: SYMMETRIC	非対称行列: ASYMMETRIC

表 4 スパース記述

スパース記述	非ゼロ要素の個数    あるいは    非ゼロ要素の割合%
--------	-------------------------------

The Implementation of LAMAX-S (Language for MAtRiX calculation on Supercomputer) - the language design and its application -

Yoshiaki Inoue<sup>1</sup>, Kazuhiko Tanaka<sup>1</sup>, Satoshi Uchida<sup>2</sup>

Naokazu Yamaki<sup>1</sup>, Shigeru Hongo<sup>3</sup>, Koutarou Mano<sup>1</sup>

1) Research Institute of Systems Planning 2) Aoyama Gakuin Univ.

3) Senshu Univ.

### 2.3 行列の演算

行列の演算は、普通の計算式とまったく同じように記述できる。さらにいくつかの演算子を加え、FORTRAN や LAMAX-E に比べてかなり拡張されたものになっている。

なお、一般のプログラミング言語では、加算や乗算の演算では、ある特定の型とその型で表現されたデータが結果として生成されるのに対し、LAMAX-S では、行列属性、型、データの3つが生成される。

表5 LAMAX-S における演算子 (S:スカラ、M:行列)

演算子	意味	形式	結果の類
+	正符号	+S,+M	S,M
-	負符号	-S,-M	S,M
'	転置	M'	M
!	LU分解	!M	M
**	べき乗	S**S,M**S	S,M
*	行列の乗算	S*S,S*M,M*S,M*M	S,M,M,M
&	要素ごとの乗算	M&M	M
/	除算	S/S, M/S	S,M
/	逆行列	S/M, M/M	M,M
%	要素ごとの除算	M%M	M
\$	連立方程式の解	M\$M	M
+	加算	S+S,M+M	S,M
-	減算	S-S,M-M	S,M
=	代入	S=S,M=M	S,M
:=	強制代入	M:=M	
<=	挿入	M{S,S}<=M	
=>	抜き出し	M{S,S}=>M	
<=>	交換	M<=>M	
<<	分割	M<<M	

### 2.4 連立方程式の求解と LU 分解

SOLVE 文で連立方程式を解くことができる。たとえば、A がバンド行列、X が未知ベクトル、Y は係数ベクトルとすると

```
SOLVE A * X = Y
```

は、この方程式を解いて、その答を X にセットする。このような処理に対し、LAMAX-S は、メーカー作成の標準的な (スーパーコンピュータ用にコーディングされた) サブルーチン呼び出すようにするので、行列 A の特性を十分に活かした処理が行われる。

行列の三角分解は、DECOMP 文を用いる。

```
DECOMP A = L * U USING LU (LU 分解の例)
```

### 3 LAMAX-S の応用範囲

自然科学系・社会科学系を問わず、各種の問題の数値解を求める際に行列演算が用いられることは多い。したがって、LAMAX-S の適用範囲もきわめて幅広いものとなる。例えば、社会科学系では

- 計量経済・統計
- OR

など、自然科学系では、原子力、気象、分子科学などの分野での

- 構造解析
- シミュレーション

などである。

### 4 LAMAX-S のプログラム例

ここで、実際に LAMAX-S のプログラム例を示すことにする。太字部分が行列である。

(例)

C:表題 Adler-Karmarker法のプログラム

```
integer m,n
parameter (m=10,n=10)

real*8 A:matrix[m,n]
real*8 D:diagonal[m]
real*8 (B,V,DV,DX):vector[m]
real*8 (C,X,DX):vector[n]
real*8 obj,objold,eps,gamma,alpha,rati
integer itera <* 繰り返し数
```

C:[1] ステップ1 データの入力

```
MINPUT A,B,X,C <* LAMAX-Sの行列入力文
```

C:[2] ステップ2 各種パラメタの設定

```
objold = -999999. <* -999999.:最小値
eps = 1.0d-6 <* 打ち切り誤差
gamma = 0.99
itera = 0
```

C:[3] ステップ3 計算

```
100 continue
```

```
itera = itera + 1
if( itera .gt. 10 ) gamma = 0.97
V = B - A * X
D = diag(1%V)
DX = 1/(A' * D ** 2 * A) * C
DV = - A * DX
```

C:[4] ステップ4 判定

```
alpha = maxelm(V % DV, 'minus')
X = X + gamma*alpha*DX
obj = C' * X
rati = dabs((obj-objold)/obj)
oldobj = obj
if(rati .gt. eps ) goto 100
```

C:[5] ステップ5 最適解の表示

```
write(*,*) '最適解が見つかりました。
          繰り返し数=',itera,' 最適解=',obj
do 110 i=1,n
write(*,*) 'X[' ,i, ' ]=',X[i]
110 continue
```

参考文献

- 1) 本郷, 内田; 行列演算用言語 LAMAX-E 利用の手引 (第2版) 東京大学大型計算機センター (1987.12)
- 2) 内田, 井上, 田中, 八巻, 本郷, 間野; 行列演算用言語 LAMAX-S の開発-実現方法- 情報処理学会 第40回全国大会一般講演論文集 (1990.3)