[日本ＯＲ学会　第７回事例研究奨励賞　ソフトウェア部門受賞作品]

# 行列演算用言語 LAMAX-S

## ——数学ソフトウェア用の言語に向けて——

八巻 直一，内田 智史，本郷　茂

## 1. はじめに

1980年代に，スーパーコンピュータが出現したことによって，科学技術計算の分野は急に前途が開けた感がある．超大容量のメモリー空間と，超高速計算によって，これまで事実上不可能であった問題の多くが現実に解けるようになったのである．これからも，スーパーコンピュータの発達は進むであろうし，それにともなって数値計算技術も大いに進展するに違いない．

ところが，スーパーコンピュータの能力を十分に引き出すためには，個々のハードウェア知識がどうしても必要となる．しかも，個々のスーパーコンピュータは相当に異なる構造をもっているので，あるコンピュータで十分な性能を示すプログラムが，他のコンピュータ上では期待ほどではないことがむしろ普通である．

上のような背景から，われわれは新しい言語を提案するにいたった．それが，LAMAX-S［１］であり，その設計のコンセプトは，以下のとおりである．

A．数学的表現が素直にできる言語を実現する．

B．個々のスーパーコンピュータのハードウェア知識を活かして，ユーザーが意図する性能を引き出す能力を実現する．

C．有用な数学ライブラリ（LINPACK［２］など）を有効に活かす能力を実現する．

D．過去の社会的資産を継承するとともに，技術計算のプログラミング文化を継承する．

数学的表現能力を期待するならば，どうしても新しい言語仕様を設計しなければならない．ハードウェアや数学ライブラリを有効にしかも自動的に利用するためには，個々のコンピュータやライブラリの特性を知識ベースとして保持し，ユーザーの目的を理解してインターフェースを実現する機構が，言語処理系に必要である．また，過去の社会的資産や技術計算のプログラミング文化は，ほとんどが FORTRAN を中心としているので，新しい言語は FORTRAN と十分な親和性が保たれていなければならない．

FORTRAN には，ある程度の数学的表現能力があり，かつ社会的資産も豊富である．しかしながら，その数学的表現能力はまだ不十分である．LAMAX-S の言語仕様は，上記の理由を動機として『FORTRAN＋行列表現』とされた．そして，処理系は FORTRAN ソースコードを生成するプリプロセッサとして実現された．

LAMAX-S 以外にも，行列型の変数を扱うことのできる言語はいくつか存在するが，あまり普及していないのが現状である．そのことには，２つの理由が挙げられる．第1は，それらの言語の多くが，FORTRANに代表される技術計算の文化とうまく融合できていないことである．行列表現が可能な言語のほとんどは，本質的にFORTRANとは異質な言語であり，FORTRANの資産の継承が困難なばかりでなく，プログラミング文化が異なることが大きな問題となるのである．第2はもっと重要だと思われるが，言語仕様に行列型変数を導入しただけでは，数学的ソフトウェアの表現機能は満たされないということである．数学的ソフトウェアに適応するため，とりわけ数値計算ソフトウェアに適応するためには，以下の機能が重要な要素となる．

A．行列の構造（対称，バンドなど）に対応する．

B．ベクトル化などの最適化を，自動的に行なう．このとき，各機種の特性を自動的に考慮する．

C．数学ライブラリを自動的に有効利用する．このとき，各種ライブラリの特性とインタフェースを自動的に考慮する．

上記のような機能が供給されることによって，はじめて本格的な数学プログラムに対応することが可能になるのであって，単に行列型変数が扱えるだけでは，小さな

やまき　なおかず　㈱システム計画研究所
〒150 渋谷区桜丘町２－９　カスヤビル
うちだ　さとし　神奈川大学工学部
ほんごう　しげる　専修大学経営学部

実験的プログラムには適用できるが，実用的なソフトウェアの構築は難しい．まして，個々のスーパーコンピュータの性能を引き出すような工夫は困難であろう．

LAMAX-S の計画は，この点の打開をめざしたものである．なお，本稿では，LAMAX-S の基本的な機能についての概要を述べるにとどまるが，さらに深い内容については別に発表する機会をもちたい．

## 2. 言語仕様と機能

LAMAX-S の文法は，FORTRAN の上に行列に関する豊富な表現能力を付加したものである．LAMAX-S の処理系（プリプロセッサ）は，LAMAX-S のソースコードを解析し，オブジェクトとして FORTRAN77 のソースコードを生成する．以下処理系をコンパイラと呼ぶ．

LAMAX-S コンパイラの特徴の 1 つは，構文解析機能の中に FORTRAN77 を完全に包含することである．もう 1 つの特徴は，これが本質的であるが，用いるコンピュータに対応した最適化を施したソースコードを自動生成することと，用いる数学ライブラリと自動的にインターフェースをとることである．そのためには，各種のコンピュータの特性を知識ベースとして保持すること，および各種の数学ライブラリの使い方の知識ベースをもつことが必要となる．そのことは，最近のプログラミング思想であるリポジトリベースの活用とも合致する．あるいはユーザーから見ると，ライブラリ利用のためのシンタックスシュガー（より使いやすいように考慮された文法）と考えることも可能であろう．

このような考え方の，さらに重要な効用は，スーパーコンピュータ間のソフトウェアの可搬性の実現であろう．LAMAX-S のソースコードは，ハードウェアとは完全に独立しているので，ハードウェア間の可搬性の問題は解消される．

以下，主として行列型変数の取扱いと，数学的表現の記述を中心に，代表的な機能について述べる．LAMAX-S の特徴には，数学的表現の記述性の他に，メモリーの動的管理や最適アンローリング（繰り返し計算を工夫して実行の並列性を高める手法）の自動実現があり，数学的意味を利用したプログラムの最適化，数学的ソフトウェアの開発環境，あるいは TeX [3]（数学文書の整形組版ソフトウェア）形式との相互変換などの研究が付随している．これらは，本稿では割愛し，他の機会にゆずりたい．また，表記の目標が現時点ですべて達成されたわけではなく，その意味で現バージョンは評価用と位置づけられるものと認識している．

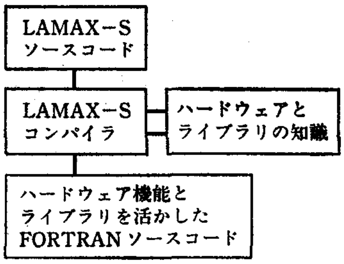


**図 1　LAMAX-S の処理概念**

### 2.1 行列型変数の導入

LAMAX-S の最大の特徴は，行列を明示的に取り扱うことができる点である．ここでは，言語仕様の上で行列がどのように定義され，どのように扱われるかの概要を示す．

行列あるいはベクトルを表わす変数または定数には，型と構造および数学的性質が付随する．型は，もともとの FORTRAN77 で定義されている型である．構造とは，対角行列，バンド行列，3 角行列などの，要素の配置に関する構造を意味する．数学的性質は，対称性，交代性，正定値性などの数学的意味づけを表わす．また，スパース行列はその数値計算上の取扱いが密行列と異なるために，数学的性質と規定している．以下に，行列の定義の例を示す．

**real : matrix [3,5]** $A$
**real : vector [6]** $x$
**real : rvector [7]** $y$
**real : matrix [5,5 diagonal]** $D$

上の定義では，$A$は 3 行 5 列の行列，$x$は 6 次元の縦ベクトル，$y$は 7 次元横ベクトル，および$D$は 5 行 5 列の対角行列となる．このように，行列は自然に定義される．行列の構造には，『対角，三重対角，上 3 角，下 3 角，バンド』がある．数学的性質には，『密，スパース，対称，交代，正定値，エルミート』がある．以下に，行列宣言のいくつかの例を示す．

1. **real : matrix [100, 100 : lower_tr(1)]** $A$
2. **real : matrix [100, 100 : sparse(200), symmetric]** $B$
3. **real : matrix [10, 10 : skew]** $C$
4. **complex : matrix [5, 5 : hermitian]** $D$
5. **real : matrix [10, 10 : band(2, 2)]** $E$

上記の各行列は次表のように定義される．

表 2　行列の定義

| 行列 | 型 | 構造と数学的性質 |
|---|---|---|
| $A$ | 実　数 | 下3角（対角要素は0） |
| $B$ | 実　数 | スパース（要素数<200），対称 |
| $C$ | 実　数 | 交代（$D'=-D$） |
| $D$ | 複素数 | エルミート |
| $E$ | 実　数 | バンド（5重） |

lower_tr(1)は，下3角行列で(1)は対角から1離れたところまで零となることを意味する．lower_tr(2)ならば，対角要素と対角要素の隣の要素が0となる．band(2,2)は，対角要素から上に2，下に2の幅であることを意味する．したがって5重対角行列になる．上に1，下に3の幅を与えるときは，band(1,3)とする．

### 2.2　動的な行列の定義

LAMAX-S の特徴の1つに，行列のダイナミックな展開がある．FORTRANにおいては，配列の便宜的に行列変数とした場合，次元の大きさはあらかじめ定めておかなければならない．実行前に大きさがわからないときは，配列の領域を安全と思われる大きな値にしておく必要がある．LAMAX-S では，動的行列の宣言をすることによって，代入の実行時に次元を動的に決定させることが可能である．動的行列の宣言は，次元の大きさをすべて＊に置き換えればよく，次のように行なう．

real : matrix [*,*]　$A$

動的行列に対して，次元の大きさを定数で与えた場合，静的行列と呼ぶ．LAMAX-S では，動的行列を処理するために，行列用のメモリー空間を管理するライブラリをもっている．このライブラリは，動的行列のための巨大なメモリー空間を確保し，必要に応じてそのメモリーの一部を切り出して行列に割り当てたり，不要になったメモリー領域を再利用できるように戻す機能を備えている．LAMAX-S では，このメモリー領域の大きさを利用者が与えるようにしている．

### 2.3　行列型関数の導入

LAMAX-S では，行列を値として返す関数を定義することができる．たとえば

real : matrix [*,*] FUNCTION $P$(X)

は，$p$が行列であることを表わす．行列関数は，数学ソフトウェアのためには大変便利である．

### 2.4　行列の演算

LAMAX-S における行列の取扱いは，行列に関して必要なすべてを実現することを目標としている．本節では，現在実現されている機能を列挙する．行列に対する操作には，以下のものがある．

- $i$，$j$ 要素を示す．　$A[i,j]$
- $i$ 行を示す．　$A[i,*]$
- 転置を表わす．　$A'$
- 部分行列を示す．　$A[i:j,k:1]$　$i$ 行から $j$ 行，$k$ 列から1列までの部分行列
- 対角部分指定．　$A\langle i\rangle$　対角からの変位が $i$ の要素からなる縦ベクトル

行列の演算式は，自然な表現をほぼそのまま用いることができる．以下に例を示そう．

1.　$A=B'*C$
2.　$A=B\hat{}*C$
3.　$k=x'*Q*x+b'*x$

1.は$B$の転置と$C$の積を$A$に代入する．2.は$B$の逆行列と$C$の積を$A$に代入する．3.の右辺は，2次形式である．このように，行列の演算式は自然に表現できる．しかし，LAMAX-S の特徴は，行列の演算が自然に表現できることに留まらず，ライブラリの積極的な利用を自動的に実現することにある．たとえば，上記例の2.では逆行列を求めているが，用いるコンピュータの性能を十分に引き出す逆行列ライブラリが存在すれば，自動的にそれを利用することができる．

スーパーコンピュータの場合には，専用のライブラリが用意されているのでそれを用いるべきである．さらに，$C$がベクトルであれば，$A$は連立1次方程式$B*A=C$の解なので，LU分解のサブプログラムを用いるほうがより効率がよいだろう．LAMAX-S は，文脈から最も適した手法を選択し，実行効率の高いプログラムを生成する．このような戦略は数学的最適化と呼ばれていて，重要な要素となっている．（数学的最適化の機能は，現バージョンでは実現していない）

## 3.　アルゴリズム記述能力

数学ソフトウェアに必要な機能の最大の要素は，アルゴリズムの記述能力であろう．行列表現の導入と行列ライブラリの隠ぺいは，アルゴリズム表現にとって予想以上の効果を発揮し，われわれが当初意図したよりも記述性能は高いものが得られたと考えられる．

### 3.1　連立1次方程式の解法

連立1次方程式 $Ax=b$ の数値解法には，大きく分け

れば消去法と繰り返し法の2つがある．ここでは，その代表的な方法であるガウスの消去法をとりあげる．なおアルゴリズムの詳細とプログラムの詳細の説明は省略するが，LAMAX-S の記述性のよさはおのずと認識できるであろう．

［アルゴリズム］ガウスの消去法

ステップ 1. 以下の，ステップ 1-1. とステップ 1-2. を $k=1$ から $n$(変数の次元) $-1$ まで行なう

ステップ 1-1. $A$ と $b$ の $k$ 行を $A_{k,k}$ で割る

ステップ 1-2. $A$ の $k$ 行 $k$ 列以降の下3角要素を0にする．

ステップ 2. 上3角行列をもつ連立1次方程式を解いて解 $x$ を得る

アルゴリズム中のステップ1-1. および1-2. は，いわゆる行列の基本演算によって実現できる．

［LAMAX-S］ガウスの消去法

```
    real : matrix [*,*]  A,D
    real : matrix [*,* : upper_tr(*)]  U
    real : vector [*]  b
    real : vector [*]  x
c
    call minput(A)
    n=icol(A)
    call minput(b)
c ステップ 1.
    do 100 k=1, n-1
c ステップ 1-1.
    D : 1. d00 : matrix [n,n : diagonal]
    D[k,k]=1./A[k,k]
c ステップ 1-2.
    D[k+1 : n,k]=-A[k+1 : n,k]/A[k,k]
    A=D*A
    b=D*b
  100 continue
c ステップ 2.
    U=A
    solve U*x=b
c
    call mprint(x)
    end
```

## 4. 数学ソフトウェア用の言語に向けて

LAMAX-S は，現在の評価バージョンにおいても，その数学アルゴリズムの記述能力の高さが，ある程度実証されたものと考えられる．しかし，数学ソフトウェアのための言語としては，さらにいくつかの機能を付加しなければならない．

まず第1は，ヒューマンインターフェースの強化である．数学記号やその意味の明示的な表現が，どうしても必要である．たとえば TeX や，いくつかの文書ソフトウェア，あるいはある種の数式処理システムにその萌芽が認められる．しかしそれらは，数値計算アルゴリズムの表現とは，直接つながってはいない．かつ，表記の意味までは完全に認識できているわけではない．

第2は，もっと多くの数学的要素の導入である．集合，区間などの型も，ぜひ必要である．さまざまな関係演算，射影などの表現，空間の概念の導入なども重要である．

第3には，計算誤差の管理能力の確立である．高速自動微分［4］などの数学ツールや，区間演算［5］などの，誤差評価を積極的に取り扱う手法の研究が現在急速に進展している．それを受けて，今後はそれらの成果を十分に取り込みながら，数値計算における誤差管理機能を，言語レベルで実現する必要があろう．

われわれは，上記目標に少しでも近づくために，これからも LAMAX-S プロジェクトを中心として推進していきたい．

［謝辞］

本研究に対してＯＲ学会賞をいただきましたことに，深く感謝いたします．今後も，なお一層努力を重ねて，ＯＲの発展のために少しでも寄与したいと思います．

### 参考文献

［1］ ASNOP 研究会，非線形最適化プログラミング，日刊工業新聞社，1991

［2］ Dongarra, J. J., Bunch. J. R., Moler, C. B. and Stewart, G. W., LINPACK Users' Guide, SIAM, 1979

［3］ Knuth, D. E., The TeXbook, Addison-Wesley, 1984

［4］ 伊理正夫，久保田光一，高速自動微分法（Ｉ）（II），応用数理，Vol. 1, No. 1, pp. 17-35, 1991, Vol. 1, No. 2, pp. 53-63, 1991

［5］ 久保田光一，伊理正夫，区間演算を用いた丸め誤差解析，情報処理，Vol. 31, No. 1, pp. 1212-1219, 1990