

I M S L の 新 バ ー ジ ョ ン に つ い て

福井義成¹⁾ 中島美也子²⁾ 大井理絵²⁾¹⁾東芝CAEシステムズ(株) ²⁾東芝総合情報システム部

I M S L (International Mathematical and Statistical Libraries)がエディション9.2からエディション10.0にバージョンアップされ, 大幅な変更が行われた。変更があった主な項目は次の点である。

(1) FORTRAN77化, (2) ライブラリをMATH, STAT, SFUNに細分化, (3) すべての計算機での倍精度版のサポート, (4) 定数の表現の変更(8進または16進を10進表現にし共通のサブルーチンにまとめた), (5) リリース形態をソースプログラムから原則としてオブジェクトプログラムに変更, (6) 契約形態の変更(MATH, STAT, SFUNを別々に契約), (7) 使用方法の変更(全面的な倍精度化のため), (8) マニュアルの変更, (9) 作業領域のアロケーション方法の変更

New Version of IMSL

¹⁾Yoshinari Fukui, ²⁾Miyako Nakajima and ²⁾Rie Ooi¹⁾TOSHIBA CAE Systems Inc.²⁾TOSHIBA CORPORATION

580-1, HORIKAWA-CHO, SAIWAI-KU, KAWASAKI 210, JAPAN

New version of IMSL is described. Major changes are the following.

- (1) FORTRAN77 support
- (2) decomposition of the library to three separate parts
- (3) double precision support on all available computers
- (4) distribution of the library by object program
- (5) new usages.

1. はじめに

IMSL (International Mathematical and Statistical Libraries) がエディション9.2からエディション10.0にバージョンアップされ、大幅な変更が行われた。ここでは変更された項目の紹介と問題点などについて述べる。

2. バージョンアップの概要

IMSLのエディション9.2からエディション10.0のバージョンアップで使用方法、リリース形態などに大幅な変更が行われた。変更があった主な項目は下記の通りである。

(1) FORTRAN 77化

エディション9.2まではFORTRAN 66対応であったものが、エディション10.0からはFORTRAN 77対応になった。すべてのソースプログラムが“IF-THEN-ELSE”などFORTRAN 77の文法で書換えられている。

表1. IMSLエディション9.2のルーチン

分 野		本数
A	分散分析	17
B	基本統計	33
C	カテゴリーデータの分析	5
D	微分方程式、求積法、微分法	14
E	固有値解析	11
F	予測、計量経済分析、時系列解析、変換	23
G	乱数の生成と検定	54
I	補間、近似、平滑化	16
L	線形代数方程式	36
M	特殊関数	67
N	ノンパラメトリック統計	26
O	観測構造、多変量解析	18
R	回帰分析	29
S	標本抽出	8
U	ユーティリティ機能	29
V	ベクトル・行列演算	110
Z	根と極値、線型計画法	23

(2) ライブラリの分類の変更

エディション9.2まではIMSLという一つのライブラリで、その中が17章(全519本)に区分されていた。エディション10.0からはIMSLが

MATH/LIBRARY
STAT/LIBRARY
SFUN/LIBRARY

の3つの独立したライブラリに分けられた。それぞれ11, 21, 13の小分類があり、サブルーチンの本数は803本, 660本, 307本である(表1, 2, 3, 4を参照)。

(3) 倍精度版の全面的サポート

エディション9.2までは単精度浮動小数点数が32ビットである計算機では単精度版と倍精度版の2つのライブラリがサポートされていたが、単精度浮動小数点数が36ビット以上の計算機では単精度版のライブラリしかサポートされていなかった。たとえば、

表2. IMSLエディション10.0 (MATH/LIBRARY)のルーチン

分 野	本数	
	単精度	倍精度
線形代数方程式	98	98
固有値解析	40	40
補間、近似	44	44
求積法、微分法	17	17
微分方程式	8	8
データ変換	19	19
非線形方程式	8	8
最適化	32	32
基本行列・ベクトル演算	101	100
ユーティリティ	44	18
ライブラリ環境ユーティリティ	7	1

REAL FUNCTION AMACH(I)

AMACH retrieves real machine constants which define the computer's real arithmetic.

Assume that single-precision floating-point numbers are represented in N_s -digit, base B form as $\sigma B^E \sum_{k=1}^{N_s} x_k B^{-k}$, where σ is the sign, $0 \leq x_k < B$ for $k = 1, \dots, N_s$, and $E_{\min_s} \leq E \leq E_{\max_s}$. Note that $B = \text{IMACH}(6)$, $N_s = \text{IMACH}(7)$, $E_{\min_s} = \text{IMACH}(8)$, and $E_{\max_s} = \text{IMACH}(9)$.

The values returned by AMACH are approximate.

The proposed IEEE standard for binary arithmetic (see Stevenson 1982) uses NaN (not a number) as the result of various otherwise illegal operations, such as computing 0/0. On computers that do not support NaN, a value larger than DMACH(2) is returned for AMACH(6). On computers that do not have a special representation for infinity, AMACH(2) returns the same value as AMACH(7).

AMACH is defined by the following table.

- AMACH(1) = $B^{E_{\min_s}-1}$, the smallest positive number.
- AMACH(2) = $B^{E_{\max_s}}(1 - B^{-N_s})$, the largest number.
- AMACH(3) = B^{-N_s} , the smallest relative spacing.
- AMACH(4) = B^{1-N_s} , the largest relative spacing.
- AMACH(5) = $\log_{10}(B)$.
- AMACH(6) = NaN (not a number).
- AMACH(7) = positive machine infinity.
- AMACH(8) = negative machine infinity.

図 1. 計算機定数に関する記述

REAL FUNCTION AMACH (N)

```

INTEGER    N
REAL       RMACH(8)
INTEGER    IRMACH(8)
EQUIVALENCE (RMACH, IRMACH)
DATA RMACH(1)/4.66254E-156/
DATA RMACH(2)/8.37795E152/
DATA RMACH(3)/7.45230E-9/           = 2-27
DATA RMACH(4)/1.49046E-8/           = 2-28
DATA RMACH(5)/.301029995663981195E0/ = LOG10(2.0)
DATA IRMACH(6)/34225495975/
DATA RMACH(7)/8.37795E152/
DATA RMACH(8)/-8.37795E152/
    
```

図 2. AMACHの例 (HIS版)

表 3. IMSLエディション10.0 (STAT/LIBRARY)のルーチン

分野	本数	
	単精度	倍精度
基本統計量	13	13
回帰分析	26	26
相関関係	8	8
分散分析	12	12
カテゴリ、離散データ分析	4	4
ノンパラメトリック統計	12	12
適合度、ランダム度検定	7	7
時系列、予測	25	25
共分散構造と因子分析	13	13
判別分析	3	3
クラスター分析	4	3
抽出	8	8
生存解析、生命検定、信頼性	2	2
多次元スケーリング	2	2
密度、Hazard予測	6	6
プリンタ表示機能	11	11
確率密度関数とその逆関数	24	24
乱数生成	49	35
ユーティリティ	30	10
補助的数学機能	89	80
ライブラリ環境ユーティリティ	7	1

表 4. IMSLエディション10.0 (SFUN/LIBRARY)のルーチン

分野	本数	
	単精度	倍精度
初等関数	17	8
三角関数、双曲線関数	30	16
積分指数関数	10	10
ガンマ関数	22	16
誤差関数	7	7
ベッセル関数	22	20
ケルビン関数	12	12
有理数次ベッセル関数	8	8
楕円積分	6	6
Weierstrassの楕円関数	4	4
確率密度関数とその逆関数	24	24
その他の関数	3	3
ライブラリ環境ユーティリティ	7	1

3 2 ビット: I B M , V A X
3 6 ビット: H I S , A C O S 6
6 0 ビット: C D C 7 6 0 0
6 4 ビット: C R A Y

などである。しかし、エディション 10.0からはすべての計算機で単精度版と倍精度版が1つのライブラリの中でサポートされるようになった。

(4) 定数の表現の変更

エディション 9.2までは収束判定定数や特殊関数の中で使用されている係数が8進や16進でDATA文の中に記述されており、計算機依存性が高いものであった。また、これらの中で共通に使われる定数までも各サブルーチンの中に散在していた。エディション 10.0からはこれらも定数が10進表現になり、共通の定数はすべてまとめられ、サブルーチン

I M A C H
A M A C H
D M A C H

の中に集約された(図1, 2参照)。

(5) リリース形態の変更

エディション 9.2までは、リリース形式がソースプログラムであったが、エディション 10.0からは、原則として、オブジェクトプログラムに変更された(要求すればソースプログラムも入手できる)。エディション 9.2まで、ソースプログラムはほとんどの計算機でラベルなしのIBM形式の固定長(ASCIIまたはEBCDIC)であった。しかし、オブジェクトプログラムによるリリースとなったため、リリース用のテープが各計算機に強く

依存する形式になった。オブジェクトプログラムの影響を受け、ソースプログラムの形式も各計算機に強く依存する形式になった。

リリーステープの本数もエディション 9.2までは単精度版または倍精度版ごとに1本であったが、エディション 10.0からはMATH, STAT, SFUNの各ライブラリごとに1本になった。単精度版と倍精度版は同じ1本のテープに入っている。

(6) 契約形態の変更

エディション 9.2まではIMSLとして1本の契約であったが、エディションからはMATH, STAT, SFUNを別々に契約することになる。

(7) 使用方法の変更

全面的な倍精度化に伴い使用方法がすべて変更になった。詳細については後で述べる。

(8) マニュアルの変更

全面的な倍精度化・使用方法の変更により、マニュアルも全面改訂された。エディション 9.2までは、バインダ形式ルだけであったが、エディション 10.0からはバインダ形式とそれをコンパクトにまとめて本の形(Soft-cover Edition)にしたものも提供されている。また使い方を簡単にまとめたリファレンスブックも提供されている。図3, 4, 5にエディション 9.2と10.0のマニュアルの例とリファレンスブックの例を示す。

(9) 作業領域のアロケーション

エディション 9.2までは、各サブ

IMSL ROUTINE NAME - LEQTLF

PURPOSE - LINEAR EQUATION SOLUTION - FULL STORAGE
MODE - SPACE ECONOMIZER SOLUTION.

USAGE - CALL LEQTLF (A,M,N,IA,B,IDGT,WKAREA,IER)

ARGUMENTS A - INPUT MATRIX OF DIMENSION N BY N CONTAINING
THE COEFFICIENT MATRIX OF THE EQUATION
AX = B.
ON OUTPUT, A IS REPLACED BY THE LU
DECOMPOSITION OF A ROWWISE PERMUTATION OF
A.

M - NUMBER OF RIGHT-HAND SIDES. (INPUT)

N - ORDER OF A AND NUMBER OF ROWS IN B. (INPUT)

IA - ROW DIMENSION OF A AND B EXACTLY AS SPECIFIED
IN THE DIMENSION STATEMENT OF THE CALLING
PROGRAM. (INPUT)

図 3. エディション 9. 2 のマニュアルの例

LSLRG/DLSLRG (Single/Double precision)

Purpose: Solve a real general system of linear equations without iterative refinement.

Usage: CALL LSLRG (N, A, LDA, B, IPATH, X)

Arguments

N - Number of equations. (Input)

A - N by N matrix containing the coefficients of the linear system. (Input)

LDA - Leading dimension of A exactly as specified in the dimension statement of the calling program. (Input)

図 4. エディション 10. 0 のマニュアルの例

Chapter 1: Linear Systems

Solution of Linear Systems, Matrix Inversion, and Determinant Evaluation

Real General Matrices

LSARG (p. 11) Solve a real general system of linear equations with iterative refinement.

CALL LSARG (N, A_{N,N}, LDA, B_N, IPATH, X_N)

LSLRG (p. 13) Solve a real general system of linear equations without iterative refinement.

CALL LSLRG (N, A_{N,N}, LDA, B_N, IPATH, X_N)

LFCRG (p. 16) Compute the LU factorization of a real general matrix and estimate its L₁ condition number.

CALL LFCRG (N, A_{N,N}, LDA, FAC_{N,N}, LDFAC, IPVT_N, RCOND)

For example, consider a real 5 × 5 band matrix with 1 lower and 2 upper codiagonals, stored in the FORTRAN array declared by the following statements.

PARAMETER (N=5, NLCA=1, NUCA=2)
REAL A(NLCA+NUCA+1,N)

The matrix A has the form

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 & 0 \\ A_{21} & A_{22} & A_{23} & A_{24} & 0 \\ 0 & A_{32} & A_{33} & A_{34} & A_{35} \\ 0 & 0 & A_{43} & A_{44} & A_{45} \\ 0 & 0 & 0 & A_{54} & A_{55} \end{bmatrix}$$

As a FORTRAN array it has the form

$$A = \begin{bmatrix} \times & \times & A_{13} & A_{24} & A_{35} \\ \times & A_{12} & A_{23} & A_{34} & A_{45} \\ A_{11} & A_{22} & A_{33} & A_{44} & A_{55} \\ A_{21} & A_{32} & A_{43} & A_{54} & \times \end{bmatrix}$$

The entries marked with an × in above array are not referenced by the IMSL band routines.

図 5. エディション 10. 0 のリファレンスカードの例

ルーチンの中で使用する作業領域は引数として与えていたが、エディション10.0からはCOMMON領域

COMMON / WORKSP /

を利用して割当てている。この方法は単純な使い方をしていないユーザには便利であるが、複雑な使い方をしていないユーザには傷害となることも考えられる。

3. 使用方法の変更

エディション9.2までは単精度版と倍精度版はそれぞれまったく独立な2本のライブラリであった。そのため単精度版から倍精度版への変更はライブラリを代えて行わなければならないかった。

ユーザの単精度版のプログラムを倍精度版に変更するような場合はユーザプログラムに“IMPLICIT REAL*8”を挿入するか、コンパイラの自動倍精度機能を利用し、IMSLの部分はライブラリファイルを切り替えるだけでよかった。しかし、単精度版と倍精度版を同時に使うことはほとんど不可能であった。

エディション10.0では1本のライブラリの中に単精度版と倍精度版が入っているため、単精度版と倍精度版を同時に使うことは簡単にできる。しかし、ユーザの単精度版のプログラムを倍精度版に変更するような場合はユーザプログラムの中でIMSLを使用している部分をすべて変更しなければならない(図3, 4参照)。

どちらの形式が良いかはIMSLを使う状況によって異なるであろう。また外部から導入したソフトウェアが中で旧バージョンのIMSLを使用している場合など両方のエディションを容

易しておかなければならない状況もあるであろう。

このような問題をユーザにとって使いやすいするためには、FORTRANの組み込み関数のように総称サブルーチンの機能をFORTRANコンパイラが持ってくれることが望ましい。しかし、FORTRANの組み込み関数のように引数が少なければよいが、数値計算のサブルーチンのように引数が多いと問題である。なお、エディション9.2は1989年6月30日までではサポートされるそうである。

4. ACOS6の場合

米国より日本の計算機の機能が進んだために起こった現象の例としてACOS6の場合について述べる。我々はACOS6でHIS版のIMSL(FORTRAN66, 2進版)を使用していた。FORTRAN66でもHISとACOS6のFORTRAN66コンパラの仕様が一部異なるためエディション9.2でも一部のDATA文を変更していた。さらにACOS6のFORTRAN77は浮動小数点数が16進表現になったため、特殊関数などの計数を約500ヶ所ほど変更して使用していた。

エディション10.0のIMSLは16進表現のFORTRAN77版となったため、ACOS6でもFORTRAN77版はあまり問題はなくなつた(16進浮動小数点数の定数の一部に問題は残っている)。しかし、FORTRAN66ではFORTRAN77とオブジェクトレベルの互換性がないため、そのままでは使用できず、ソースレベルでも“IF-THEN-ELSE”で記述されているためコンパイルできない。