SSL II の Fortran 90 インターフェイスの開発

渡部 善隆
(九州大学大型計算機センター)

Fortran 90 から新たに導入された機能を用い，連立 1 次方程式の解・固有値・ Fourier 係数などを組込み関数と同様の手軽さで求めることのできる関数副プログラムを作成し，スーパーコンピュータ VPP700/56 上で公開した．関数副プログラムは内部で数値計算サブルーチンライブラリ SSL II を呼び出すため，精度・速度ともに十分な性能が得られることが期待される．利用者はソースプログラムをコピーし，目的に応じて修正することができる．また， SSL II 以外のサブルーチンライブラリへの差し替え，共存も可能であり，他の計算機システムへの移植の雛型としても利用できる．

本稿では SSL II の Fortran 90 インターフェイスの概要とこれまでに作成した関数副プログラムを紹介し，インターフェイス部分に要するオーバーヘッドの実測値を示す．

1 Fortran 90 と SSL II

1.1 Fortran 90

Fortran(Formula Translation) は 1950 年代に John. W. Backus らによって開発されたプログラム言語であり，科学技術計算において現在広く使われている． 1991 年に ISO(国際標準化規格) の国際規格として制定された Fortran 90 では，それまでの FORTRAN 77 規格を完全に包含しながら，他のプログラム言語 (C, C++, Pascal など) での経験にもとづいた機能を多数採り入れている．追加されたおもな新機能は，配列演算・利用者データ型定義・ポインタ・動的記憶割付けなどである ([3])．日本では 1994 年 1 月 1 日に JIS(日本工業規格)Fortran 90([1]) が制定された．現在の JIS Fortran 規格は 1998 年 10 月 20 日付けで制定された Fortran 95([2]) である．

九州大学大型計算機センターでは 1999 年 8 月現在，スーパーコンピュータ VPP700/56，汎用コンピュータ M-1800/20U，ライブラリサーバー S-4/1000E 上で Fortran 90 が利用できる．また，次期汎用コンピュータシステムでは Fortran 95 の公開を予定している．

1.2 SSL II

SSL II(Scientific Subroutine Libraly II) は富士通株式会社提供の数値計算ライブラリであり，線形方程式や微分方程式などの問題を解く約 230 種類のサブルーチンから構成される ([4], [5], [6], [7])．各サブルーチンは Fortran で記述されており，利用者プログラムから `CALL` 文で使用する．流体力学・構造解析・分子化学・核融合などの分野で行なわれる大規模数値計算では，連立 1 次方程式・固有値計算・ Fourier 変換などの計算に要するコストがプログラム全体のかなりの部分を占めるため，これらの部分を SSL II を用いて高速化することにより格段の性能向上が期待できる．

SSL II の問題点として，ソースプログラムが非公開であることと並び，利用における「敷居」の高さを挙げることができる．SSL II は FORTRAN 77 プログラムからの利用を想定して設計されている．そのため利用者は行列の整合寸法・次数・収束判定パラメータなどの多くの引数をサブルーチンに与えなければならない．また，必要に応じてサブルーチンに渡す作業領域を自身であらかじめ確保することも要求される．たとえば実行列 A, ベクトル b に対する連立 1 次方程式 $Ax = b$ の解 x の数値解を SSL II のサブルーチン `DVLAX`([5]) を用いて倍精度で求める場合，呼び出し例は

```
CALL DVLAX(A,K,N,B,EPSZ,ISW,IS,VW,IP,ICON)
```

となる．利用者は行列 A，ベクトル b の他にも作業用配列 `IP`, `VW` を確保したり，パラメータ `EPSZ`, `ISW` に適当な値を設定したのち 10 個の引数を間違いなく順番に入力する必要がある．

一方， *Mathematica*, Maple V, MATLAB などの統合数学ソフトウェアには最低限の引数で連立 1 次方程式の解を求める関数がサポートされている．たとえば連立 1 次方程式 $Ax = b$ の数値解は *Mathematica*([10]) の組み込みオブジェクト `LinearSolve` を用いると以下の記述で求めることができる．

```
x=LinearSolve(A,b)
```

近年，線形計算ライブラリ LAPACK([8]) のインターフェイスを改良した LAPACK 90([9]) が公開された．LAPACK 90 は `CALL` 文で利用するサブルーチン群であり， Fortran 90 の新機能を用いて引数を減らすことに成功している．そこで， LAPACK 以上の豊富な機能を有する SSL II に対しても *Mathematica* などの組み込みオブジェクトと同じく簡単に利用できる関数副プログラムを提供し， SSL II の利用における「敷居」を低くす

ることを目的として今回の開発を行なった．

2 Fortran 90 の新機能

ここでは，SSL II の Fortran 90 インタフェースに利用した Fortran 90 の機能を簡単に紹介する．

2.1 配列関数

Fortran 90 の新機能として，簡潔な記法を用いて配列を操作するための組込み関数が多数備わった．おもな関数を表 1 に挙げる．

表 1: 配列に関する組込み関数

関数名	機能
MATMUL	行列積
DOT_PRODUCT	ベクトルの内積
TRANSPOSE	行列の転置
MAXVAL	配列要素の最大値を求める
MINVAL	配列要素の最小値を求める
PRODUCT	全配列要素の積
SUM	全配列要素の和
SIZE	配列要素の総数を求める

VPP700/56 で測定した MATMUL, DOT_PRODUCT の処理性能は [11] を参照されたい．

2.2 動的配列割り当て

プログラムの実行中に配列の大きさを決める動的配列割り当て機能がある．配列は ALLOCATABLE 属性を使って宣言し，ALLOCATE 文により配列の大きさを決定する．不要になった配列は DEALLOCATE 文によって解放する．この機能を用いることにより，SSL II で用いる作業用配列を利用者が自身で宣言する手間をなくすことができる．以下は，動的配列 vw, ip を宣言し，何らかの方法で決定された n によって配列の大きさを割り当て，SSL II のサブルーチン DLAXR に作業用配列として渡すプログラム例である．

```
real(kind=8),dimension(:),allocatable    :: vw
integer(kind=4),dimension(:),allocatable :: ip
         :                                ! ↑配列の宣言
allocate(vw(n),ip(n))                     ! 大きさの決定
         :
call DLAXR(x,A,k1,n,Z,b,ip,vw,icon)       ! 作業用配列
         :
deallocate(vw,ip)                         ! 配列の解放
```

2.3 省略可能引数

引数が省略可能であるものを OPTIONAL 属性によって宣言することで，SSL II に省略値を渡すことができる．利用者は自身でパラメータまたは手法を選択する必要のある場合にのみ引数を指定すればよい．その場合，キーワードを用いた引数を書くことができる．利用者が引数を指定したかどうかの判断は組込み関数 PRESENT で知ることができる．

```
function linear_solve_s(A,b,eps,refine) result(x)
              :
  real(kind=4),intent(in),optional       :: eps
  character(len=1),intent(in),optional   :: refine
  real(kind=4)                           :: epsz=0.0
              :
  if(present(eps)) epsz=eps
              :
  if(present(refine).and.refine=='Y' ) then
    call LAXR(x,A,k1,n,Z,b,ip,vw,icon)
  end if
```

この例では，関数 linear_solve_s の引数 eps および refine は省略可能である．利用者が引数を指定する場合にはキーワード引数として

```
x=linear_solve_s(A,b,eps=1.0E-5)
x=linear_solve_s(A,b,refine='Y')
x=linear_solve_s(A,b,refine='Y',eps=1.0E-2)
```

などと指定する．

2.4 総称名

組込み手続きの数学関数として利用できる ABS, MATMUL などは引数の型や精度によらず同一の名前で記述することができる．利用者作成の手続きに対しても，INTERFACE 文を用いることにより複数の関数を同じ総称名として呼び出せるようにできる．

```
interface eigenvalues        ! 総称名の指定
  function eigenvalues_s(A) result(x)
    real(kind=4),dimension(:,:),intent(in)   :: A
    complex(kind=4),dimension(size(A,2))     :: x
  end function eigenvalues_s

  function eigenvalues_d(A) result(x)
    real(kind=8),dimension(:,:),intent(in)   :: A
    complex(kind=8),dimension(size(A,2))     :: x
  end function eigenvalues_d

  function eigenvalues_sc(A) result(x)
    complex(kind=4),dimension(:,:),intent(in):: A
    complex(kind=4),dimension(size(A,2))     :: x
  end function eigenvalues_sc

  function eigenvalues_dc(A) result(x)
    complex(kind=8),dimension(:,:),intent(in):: A
    complex(kind=8),dimension(size(A,2))     :: x
  end function eigenvalues_dc
end interface
```

上の例では，総称名 EIGENVALUES により，単精度実数型・倍精度実数型・単精度複素数型・倍精度複素数型の関数を型と精度に応じてそれぞれ呼び出せるように指定している．個別の関数副プログラム部分には，引数と結果の宣言を記述する．

2.5 モジュール

総称名として記述した関数副プログラムはモジュールと呼ばれる副プログラムにまとめて記述することにより，他のプログラム単位から参照することができる．

```
module SSL2_90
  interface linear_solve    ! 総称名の指定
        :                   ! (連立 1 次方程式)
  end interface
        :
  interface inverse         ! 総称名の指定
        :                   ! (逆行列)
  end interface

end module SSL2_90
```

モジュールに記述した関数副プログラムを利用するには，USE 文を用いる．

3 関数副プログラムの概要

ここでは，SSL II の Fortran 90 インターフェイスとして作成した関数副プログラムを紹介する．精度は単精度・倍精度が利用できる．表 2 に機能の一覧を挙げる．

表 2: 機能一覧

関数名	機能
LINEAR_SOLVE	連立 1 次方程式の解
INVERSE	逆行列
LEAST_SQUARES	最小二乗解，最小二乗最小ノルム解
MATRIX_POWER	行列の乗算
DET	行列式
EIGENVALUES	固有値
FOURIER	Fourier 変換，Fourier 逆変換
INTEGRATE	1 次元実関数の数値積分

● LINEAR_SOLVE

書式

LINEAR_SOLVE(A,B[,EPS=*eps*][,REFINE='Y'])

機能

連立 1 次方程式 $Ax = b$ を部分ピボット選択法による Gauss の消去法で解く．

引数

A ...行列 A．実数型または複素数型 2 次元配列．行および列の大きさが B の大きさ以上であること．

B ...ベクトル b．実数型または複素数型 1 次元配列．B の大きさが方程式の次数となる．

EPS=*eps* ...行列の非正則性判定定数．選択されたピボットの絶対値と A の要素の絶対値最大との相対比が *eps* 以下になった場合，行列は数値的に非正則と判断し処理を打ち切る．省略した場合，標準値が採用される．詳細は SSL II のマニュアルを参照．

REFINE='Y' ...解の反復改良を行なうことを指定．省略可能．倍精度の場合，実行時間が大幅に増加することがあるので注意が必要．

戻り値

x の近似解．B と同じ大きさの実数型または複素数型 1 次元配列．

SSL II

VLAX, LAXR, DVLAX, DLAXR, LCX, LCXR, DLCX, DLCXR

● INVERSE

書式

INVERSE(A,[,EPS=*eps*])

機能

行列 A を部分ピボット選択法による Gauss の消去法を用いて LU 分解し，逆行列 A^{-1} を求める．

引数

A ...行列 A．実数型または複素数型 2 次元配列．列の大きさを次数とする．行の大きさは列の大きさ以上であること．

EPS=*eps* ...行列の非正則性判定定数．選択されたピボットの絶対値と A の要素の絶対値最大との相対比が *eps* 以下になった場合，行列は数値的に非正則と判断し処理を打ち切る．省略した場合，標準値が採用される．詳細は SSL II のマニュアルを参照．

戻り値

逆行列 A^{-1}．A と同じ大きさの実数型または複素数型 2 次元配列．

SSL II

VALU, VLUIV, DVALU, DVLUIV, CLU, CLUIV, DCLU, DCLUIV

● LEAST_SQUARES

書式

```
LEAST_SQUARES(A,B[,EPS=eps][,REFINE='Y'] &
                  [,MINIMAL_NORM='Y'])
```

機能

$m \times n$ の実行列 A を係数とする連立 1 次方程式 $Ax = b$ に対し，Householder 変換による最小二乗解または特異値分解法による最小二乗最小ノルム解を求める．

引数

A ...$m \times n$ の実行列 A．実数型 2 次元配列．$m \geq n \geq 1$ かつ MINIMAL_NORM='Y' の指定がない場合は最

小二乗解を，$m < n$ または MINIMAL_NORM='Y' が指定された場合は最小二乗最小ノルム解を求める．

B ...ベクトル b．大きさ m の実数型 1 次元配列．

EPS=eps ...最小二乗最小ノルム解を求める場合の特異値の相対零判定値．省略可能．最小二乗解を求める場合に指定しても無視される．詳細は SSL II のマニュアルを参照．

REFINE='Y' ...最小二乗解の反復改良を行なうことを指定．省略可能．倍精度の場合，実行時間が大幅に増加することがあるので注意が必要．最小二乗最小ノルム解を求める場合は無視される．

MINIMAL_NORM='Y' ...最小二乗最小ノルム解を求めることを指定．$m \geq n \geq 1$ であっても階数落ち $(\mathrm{rank}(A) < \min\{m,n\})$ がある場合に指定する．

戻り値

最小二乗近似解または最小二乗最小ノルム近似解．大きさ n の実数型 1 次元配列．

SSL II

LAXL, LAXLR, LAXLM, DLAXL, DLAXLR, DLAXLM

● MATRIX_POWER

書式

MATRIX_POWER(A,K)

機能

行列 A の k 乗 A^k を求める．

引数

A ...正方行列 A．実数型または複素数型 2 次元配列．

K ...乗数 k．整数型．

戻り値

行列 A^k． A と同じ大きさの実数型または複素数型 2 次元配列．

組込み関数

MATMUL

● DET

書式

DET(A[,EPS=eps])

機能

正方行列 A の行列式を部分ピボット選択法による LU 分解により計算する．

引数

A ...正方行列 A．実数型または複素数型 2 次元配列．

EPS=eps ...行列の非正則性判定定数．省略した場合，標準値が採用される．詳細は SSL II のマニュアルを参照．

戻り値

行列式の値．実数型または複素数型．

SSL II

VALU, DVALU, CLU, DCLU

● EIGENVALUES

書式

EIGENVALUES(A)

機能

$m \times m$ の行列 A の固有値を実行列の場合 Householder 変換および 2 段 QR 法，複素行列の場合安定化基本相似変換および QR 法により求める．

引数

A ...行列 A．実数型または複素数型 2 次元配列．列の大きさを次数とする．行の大きさは列の大きさ以上であること．

戻り値

近似固有値．A と列と同じ大きさの複素数型 1 次元配列．

SSL II

BLNC, HES1, HSQR, DBLNC, DHES1, DHSQR, CBLNC, CHES2, CHSQR, DCBLNC, DCHES2, DCHSQR,

● FOURIER

書式

FOURIER(X[,INVERSE=='Y'])

機能

項数 $n = 2^l$(l は自然数) の 1 次元複素時系列データ $\{x_j\}$ が与えられたとき，離散型複素 Fourier 変換，または逆変換を計算する．

引数

X ...大きさ n の複素数型 1 次元配列．

INVERSE=='Y' ...Fourier 逆変換を行なうことを指定．省略可能．

戻り値

項数 $n = 2^l$(l は自然数) の 1 次元複素時系列データ．入力された $\{x_j\}$ から

$$n\alpha_k = \sum_{j=0}^{n-1} x_j \omega^{-jk}, \qquad k = 0, 1, \ldots, n-1$$

を計算し，$\{n\alpha_j\}$ が返却される．INVERSE=='Y' が指定された場合，入力された $\{\alpha_j\}$ から

$$x_j = \sum_{k=0}^{n-1} \alpha_k \omega^{jk}, \qquad j = 0, 1, \ldots, n-1$$

を計算し，$\{x_j\}$ が返却される．ここで $\omega = \exp(2\pi i/n)$．

SSL II

VCFT2, DVCFT2

- INTEGRATE

書式

INTEGRATE(F[,A=a][,B=b][,EPS=eps])

機能

1 次元実関数 $f(x)$ の積分の近似値を高橋・森の二重指数関数型積分公式により求める．

引数

F . . . 被積分関数 $f(x)$ を計算する関数副プログラム名．必ず INTEGRATE を呼び出すプログラムにおいて EXTERNAL 文による宣言をすること．

A=a . . . 積分区間の下限．

B=b . . . 積分区間の上限．

【注意】 半無限区間積分を計算する場合は A=0.0(単精度) または A=0.0D0(倍精度) を指定し，B は省略すること．全無限区間積分を計算する場合は A, B の指定を省略すること．

EPS=eps . . . 積分の近似値 S に対する相対誤差の上限を指定する．INTEGRATE は

$$\left| S - \int_a^b f(x)\,dx \right| \Big/ \left| \int_a^b f(x)\,dx \right| \le eps$$

を満たす S を求める．省略した場合，マシンエプシロンが採用される．

戻り値

積分の近似値．実数型．

SSL II

AQE, AQEH, AQEI DAQE, DAQEH, DAQEI

4 利用方法

ここでは，具体的な関数副プログラムの利用方法などについて述べる．なお，コマンド・オプションは処理系に依存する．

4.1 VPP700/56 での利用方法

VPP700/56(ホスト名 kyu-vpp) の /usr/local/lib に総称名を記述したモジュール ssl2_90.mod と関数副プログラムを翻訳したオブジェクトファイルを集めたアーカイブライブラリ libssl2_90.a がある．

モジュールの宣言は PROGRAM 文に続けて記述する．

```
  PROGRAM TEST
  USE SSL2_90        ! モジュールの引用
  IMPLICIT NONE
  INTEGER,PARAMETER :: N=8
        :
```

以下に連立 1 次方程式を LINEAR_SOLVE により解くサンプルプログラムを挙げる．

```
program Linear_solver_test
  use ssl2_90
  implicit none
  real(kind=8),dimension(:,:),allocatable :: A
  real(kind=8),dimension(:),allocatable   :: x,b
  integer(kind=4)                         :: n,i,j
  external linear_solve

  read(5,*) n
  allocate(A(n,n),x(n),b(n))

  do i=1,N
    do j=1,N
      if( i <= j ) then
        a(i,j)=i
      else                ! テスト用行列の作成
        a(i,j)=j
      end if
    end do
  end do
  b=0
  do i=1,n
    do j=1,n
      b(i)=b(i)+a(i,j)    ! テスト用ベクトルの作成
    end do
  end do

  x=linear_solve(A,b,refine='Y')

  write(6,*) 'maximum error=',maxval(abs(x-1.0D0))
  deallocate(A,x,b)
end program Linear_solver_test
```

実行ファイルの作成のためには，モジュールの引用を指定する翻訳時オプション -Am，モジュールの存在するディレクトリを指定する翻訳時オプション -I，および，アーカイブライブラリ libssl2_90.a の場所を指定す

るリンク時オプション -L に続けて関数副プログラム名，SSL II/VP のライブラリ名を指定する．

```
kyu-vpp% frt -Am -I/usr/local/lib program.f90\
? -L/usr/local/lib -lssl2_90 -lssl2vp
```

環境変数 FORT90C，LD_LIBRARY_PATH にあらかじめオプション，ディレクトリ名を指定すると入力の手間を省くことができる．

```
kyu-vpp% setenv FORT90C '-Am -I/usr/local/lib'
kyu-vpp% setenv LD_LIBRARY_PATH /usr/local/lib
```

この場合，以降はリンク時オプション -l のみを指定する．

```
kyu-vpp% frt program.f90 -lssl2_90 -lssl2vp
kyu-vpp% a.out
```

4.2 プログラムのカスタマイズ

kyu-vpp の /usr/local/qlib/src に関数副プログラムとモジュールのソースを公開する．利用者は誰でもプログラムをコピーし，用途に合わせて変更することができる．また，SSL II の呼び出し部分を修正することによって，他の計算機システム，サブルーチンライブラリに対応した関数副プログラムに変更することもできる．

5 オーバーヘッドの実測値

ここでは，インターフェイス部分の処理に要するオーバーヘッドの実測結果について述べる．

5.1 オーバーヘッド

関数副プログラムでは，使い勝手を向上する目的で作業領域を関数が呼び出される都度動的に宣言している．また，通常 SSL II のサブルーチンに引数として渡される行列はその結果が保存されない．関数副プログラムでは行列データを保存するため，行列のコピーを行なっている．これらの部分に要するオーバーヘッドを，関数副プログラムを用いない SSL II のサブルーチンと比較することにより計測した．

比較機種は FUJITSU VPP700/56(1PE; 主記憶 2GB, 最大処理性能 2.2GFLOPS, OS: UXP/V) の Fujitsu Fortran90/VP Compiler Driver L98121 TX01393，Sun ULTRA2(主記憶 256MB, ULTRA SPARC 200MHz, SunOS 5.5.1) の Fujitsu Fortran Compiler Driver Version 4.0，FUJITSU FMV-6500TX2(主記憶 384MB, Intel Pentium III プロセッサ 500MHz, Windows 98) の Fortran & C Academic Package V2.0L10 を用いた．時間は VPP700/56, Sun ULTRA2 は Fortran コンパイラの CLOCK サービスサブルーチンを用いた CPU 時間を，FMV-6500TX2 は TIMER サービスサブルーチンを用いた経過時間を 10 回測った平均値である．

5.2 数値例 1

4.1 節のサンプルプログラムの行列 A, ベクトル b に対する n 次連立 1 次方程式の解を求めるまでの時間を LINEAR_SOLVE と SSL II の DVLAX とで比較した．

数字は表示桁以下を切捨てている．比率は SSL II のサブルーチンの実行時間を関数副プログラムの処理に要する時間で割り，% 表示している．

表 3: 連立 1 次方程式 $n = 1000$ の実行時間 (単位: 秒)

計算機名	LINEAR_SOLVE	SSL II	比率
VPP700/56	0.472	0.470	99%
Sun ULTRA2	7.392	7.200	97%
FMV-6500TX2	5.372	4.803	89%

表 4: 連立 1 次方程式 $n = 2000$ の実行時間 (単位: 秒)

計算機名	LINEAR_SOLVE	SSL II	比率
VPP700/56	3.43	3.40	99%
Sun ULTRA2	56.57	54.57	96%
FMV-6500TX2	38.56	37.32	96%

次元数が大きい場合，SSL II の演算に要するコストが大部分を占めている．

5.3 数値例 2

2 次元の同次境界条件を持つ Poisson 方程式を有限要素法により離散近似して求まる行列の固有値の計算を EIGENVALUES と SSL II の DBLNC+DHES1+DHSQR とで比較した．

表 5: 固有値問題の実行時間: 361 次元 (単位: 秒)

計算機名	EIGENVALUES	SSL II	比率
VPP700/56	0.891	0.873	97%
Sun ULTRA2	8.907	8.822	99%
FMV-6500TX2	4.738	3.961	83%

表 6: 固有値問題の実行時間: 841 次元 (単位: 秒)

計算機名	EIGENVALUES	SSL II	比率
VPP700/56	6.88	6.80	98%
Sun ULTRA2	182.53	182.47	99%
FMV-6500TX2	68.88	66.26	96%

次元が大きくなると，作業領域の割り当て・解放，行列のコピーなどに要するインターフェイス部分のコストはわずかであることがわかる．しかし，同じ手続きを何度も呼び出すことが多い反復計算においては，同様な作業領域を何度も割り当て・解放することは効率が悪い．また，SSL II のサブルーチンには，以前の処理結果を

利用したり固有値・固有ベクトルを同時に返却するなど，関数副プログラムでは実現できない多彩な機能を備えている．したがって，より性能・機能を追求する場合には直接 SSL II を利用するべきである．

6 おわりに

本稿では，より手軽に SSL II を利用するために開発した Fortran 90 インターフェイスを紹介した．今後の課題はマニュアル・サンプルプログラムの整備，ブラックボックスとしてのライブラリの使用に抵抗がある利用者のための代替となるソースプログラムの作成・公開，演算子再定義機能を用いた有理数・区間演算・任意桁演算への拡張，計算結果の誤差限界の評価などである．

参考文献

[1] JIS X 3001(1994) プログラム言語 Fortran[ISO/IEC 1539-91], JIS ハンドブック 情報処理 プログラム言語編, pp.17–346, 日本規格協会 (1994).

[2] JIS X 3001-1(1998) プログラム言語 Fortran— 第 1 部: 基底言語 [ISO/IEC 1539-1], 日本規格協会 (1998).

[3] M. Metcalf, J. Reid (西村 恕彦, 和田 英穂, 西村 和夫, 高田 正之 訳): 詳解 Fortran 90, bit 別冊, 共立出版 (1993).

[4] 富士通 SSL II 使用手引書 (科学用サブルーチンライブラリ), 99SP-4020, 富士通株式会社 (1987).

[5] FIJITSU SSL II 拡張機能使用手引書 (科学用サブルーチンライブラリ), 99SP-4070, 富士通株式会社 (1991)．

[6] FUJITSU SSL II 拡張機能使用手引書 II(科学用サブルーチンライブラリ), J2X0-1360, 富士通株式会社 (1995)．

[7] FUJITSU SSL II/VPP 使用手引書 (科学用サブルーチンライブラリ)V12 用, J2X0-1371, 富士通株式会社 (1997)．

[8] http://www.netlib.org/lapack/

[9] http://www.netlib.org/lapack90/

[10] Stephen Wolfram(スティーブン・ウルフラム): Mathematica ブック 第 3 版, トッパン (1998).

[11] 渡部 善隆: 並列スーパーコンピュータの演算性能 ~ Fortran プログラムでの実測データ ~, 全国共同利用大型計算機センター研究開発論文集, No.19 (1997) pp.100-108.