BLAS LAPACK USER'S GUIDE

Third Edition, January 2004
The contents of this manual may be revised without prior notice.

Preface

This manual describes the usage of the following two software.

BLAS version 4.0
LAPACK version 4.0

The products are the implementations of the public domain BLAS (Basic Linear Algebra Subprograms) and LAPACK (Linear Algebra PACKage), which have been developed by groups of people such as Prof. Jack Dongarra, University of Tennessee, USA and all published on the WWW (URL: http://www.netlib.org/) .

The structure of the manual is as follows.

1 Overview

The products are outlined with emphasis on the differences from the public domain versions.

2 Documentation

The manual is not intended to give calling sequences of individual subprograms. Instead, the reader is requested to refer to a set of documentation available to the general public. The section describes where and how to access it.

3 Example program

This section shows an example code that uses subroutines from BLAS and LAPACK. The example is simple enough to understand and intended for use in order to describe principles of calling BLAS and LAPACK subroutines from a Fortran program.

4 Compile and Link

The way of compiling the user's program containing calls to the products and of link-editing with the products is described.

5 Notes on Use

Several notes are provided on usage of the products.

Appendix A Routines List

The entire list of subprograms provided is given.

For general usage of BLAS and LAPACK please see the documentation mentioned in:

2. Documentation”.

For a detailed specification of OpenMP Fortran please refer *OpenMP Fortran Application Program Interface Nov 2000 2.0* (http://www.openmp.org/).

Acknowledgement

BLAS and LAPACK are collaborative effort involving several institution and it is distributed on Netlib.

Contents

List of Figures

List of Tables

1. Overview

The optimized BLAS and LAPACK libraries are based on the public domain BLAS (Basic Linear Algebra Subprograms) and LAPACK (Linear Algebra PACKage) libraries. Each routine can be called from user programs written in Fortran with the CALL statement.

1.1 BLAS

BLAS is a library for vector and matrix operations. BLAS source code is provided on Netlib. BLAS includes 57 functions. The total number of routines for all precision types amounts to approximately 170.

BLAS provides the following routines.

Level 1 BLAS	:	Vector operations
Level 2 BLAS	:	Matrix and vector operations
Level 3 BLAS	:	Matrix and matrix operations
Sparse-BLAS	:	Sparse vector operations

1.2 LAPACK

LAPACK is a library of linear algebra routines. LAPACK source code is provided on Netlib. LAPACK includes approximately 320 functions. The total number of routines for all precision types amounts to approximately 1300.

LAPACK provides the following routines.

- Linear equations
- Linear least squares problems
- Eigenvalue problems
- Singular value decomposition

LAPACK contains driver routines, computational routines and auxiliary routines. Driver routines are those which deal with general linear algebraic problems such as a system of linear equations, while computational routines serve to work as components of driver routines such as LU decomposition of matrices. Auxiliary routines perform certain subtask or common low-level computation.

2. Documentation

The calling interfaces (routine name, parameter sequence) of subroutines provided in the products are unchanged from the public domain counterparts. For that reason, the manual does not include the calling specifications of individual subroutines. Instead, the user is requested to refer to the publicly available documentation to be listed below.

2.1 Users' Guide

Refer to the following manual for usage descriptions of the individual routines provided by this software:

- *LAPACK Users' Guide, Third Edition* (SIAM, 1999)

  The book describes in detail the usage of LAPACK, including calling specifications, purposes, parameter descriptions, performances, and accuracy of driver routines and computational routines. Also included is a quick reference of BLAS.

2.2 On-line Documentation

Several documentation of BLAS and LAPACK are available online from the domain http://www.netlib.org/. The following names are all as of January, 2003.

- LAPACK Users' Guide.

  The following URL provides an overview of calling forms for LAPACK routines, which includes the same information described in “Part 1 Guide” in the book of *LAPACK Users' Guide, Third Edition*.

  http://www.netlib.org/lapack/lug/lapack_lug.html

- Manual pages (i.e. “man” pages) for BLAS and LAPACK routines (a gzip tar file).

  http://www.netlib.org/lapack/manpages.tgz

3. Example program

This section describes how to call subroutines of BLAS and by using a simple example code.

3.1 Linear equations

3.1.1 Problem

Let's consider a system of linear equations

$Ax = b$

, where A is a real dense matrix of order n, b the right hand side vector of order n, and x the solution vector. An interesting case is that we have multiple right hand side vectors for each of which we need the solution. This problem can be written in a matrix form,

$AX = B$

, where each column vector of B , denoted by b_i (i = 1,2,...,m) ,stands for each right hand side vector, and each column of X , denoted by x_i (i = 1,2,...,m) , the solution vector corresponding to b_i

3.1.2 Example code

In general, the user can choose the subroutine DGESV from LAPACK, a driver routine, to solve equations of the matrix form above. But for explanation purpose here, let's use kind of component routines DGETRF and DGETRS, which are called *computational* routines in the LAPACK book. The former is to compute an LU factorization of a given coefficient matrix A, and the latter to solve the factorized system for the solutions X.

Now let's assume we are going to solve the equations of order 200 with 80 different right hand side vectors. An example program is shown below, where subroutine **inita** sets up the coefficient matrix in array **a**, the right hand side matrix consisting of b_i is set up using a given solutions for explanation purpose, and subroutine **check** checks the solutions x_i obtained back for correctness.

Example

```
      implicit real*8 (a-h,o-z)
      parameter(maxn=200,m=80,k=maxn+1)
      parameter(zero=0.0d0,one=1.0d0)
      real*8 a(k,maxn),aa(k,maxn),x(k,m),b(k,m)
      integer ip(maxn)
C     ============================================================
C     Define the matrix
C     ============================================================
      n=maxn
      call inita(a,k,n)
      do i=1,n
        do j=1,n
          aa(j,i)=a(j,i)
        end do
      end do
C     ============================================================
C     LU decomposition
C     ============================================================
      call dgetrf(n,n,a,k,ip,info)
C     ============================================================
C     Defeine the vectors
C     ============================================================
      do jm=1,m
        do jn=1,n
```

```
            x(jn,jm)=jn+jm
          end do
        end do
        call dgemm('N','N',n,m,n,one,aa,k,x,k,zero,b,k)
C       ===========================================================
C       Solution
C       ===========================================================
        call dgetrs('N',n,m,a,k,ip,b,k,info)
        if(info.ne.0) then
          write(6,*) 'error in dgetrs   info = ',info
          stop
        end if
C       ===========================================================
C       Check result
C       ===========================================================
        call check(a,b,k,n,m)
        end
```

4. Compile and Link

BLAS and LAPACK thread-safe versions are to be called from an OpenMP Fortran program. Compilation should be done using Fujitsu Fortran compiler. This section describes procedures of compilation through execution for these programs.

For users calling both BLAS and LAPACK routines, please follow information given in "4.2 LAPACK" and BLAS routines can be linked implicitly.

4.1 BLAS

BLAS can be used by linking it with user programs written in Fortran language. It is linked to the user program when `-blas` is specified on the compile command line. Two distinct version of library are provided, a general version and a Pentium®4 version which is tuned by using SSE2 instructions. If the option `–sse2` is specified in conjunction with `–blas,` the P4 optimized version of the library is linked. Other options relevant to optimization can be specified when needed.

Example 1:

Compile a user's program a.f90, and link it with the general version of BLAS library.

```
lf95 a.f90 –blas
```

Example 2:

Compile a user's program a.f90, and link it with the Pentium®4 version of BLAS library.

```
lf95 a.f90 –tp4 –sse2 –blas
```

4.2 LAPACK

LAPACK can be used by linking it with user programs written in Fortran language. It is linked to the user program when `-lapack` is specified on the compiler command line. Since LAPACK calls BLAS, when `-lapack` is specified, the BLAS libraries are automatically linked also.

Example 1:

Compile a user's program a.f90, and link it with LAPACK library and the general version BLAS library.

```
lf90 a.f90 -lapack
```

Example 2:

Compile a user's program a.f90, and link it with LAPACK library and the Pentium®4 version of BLAS library.

```
lf95 a.f90  -tp4 –sse2 -lapack
```

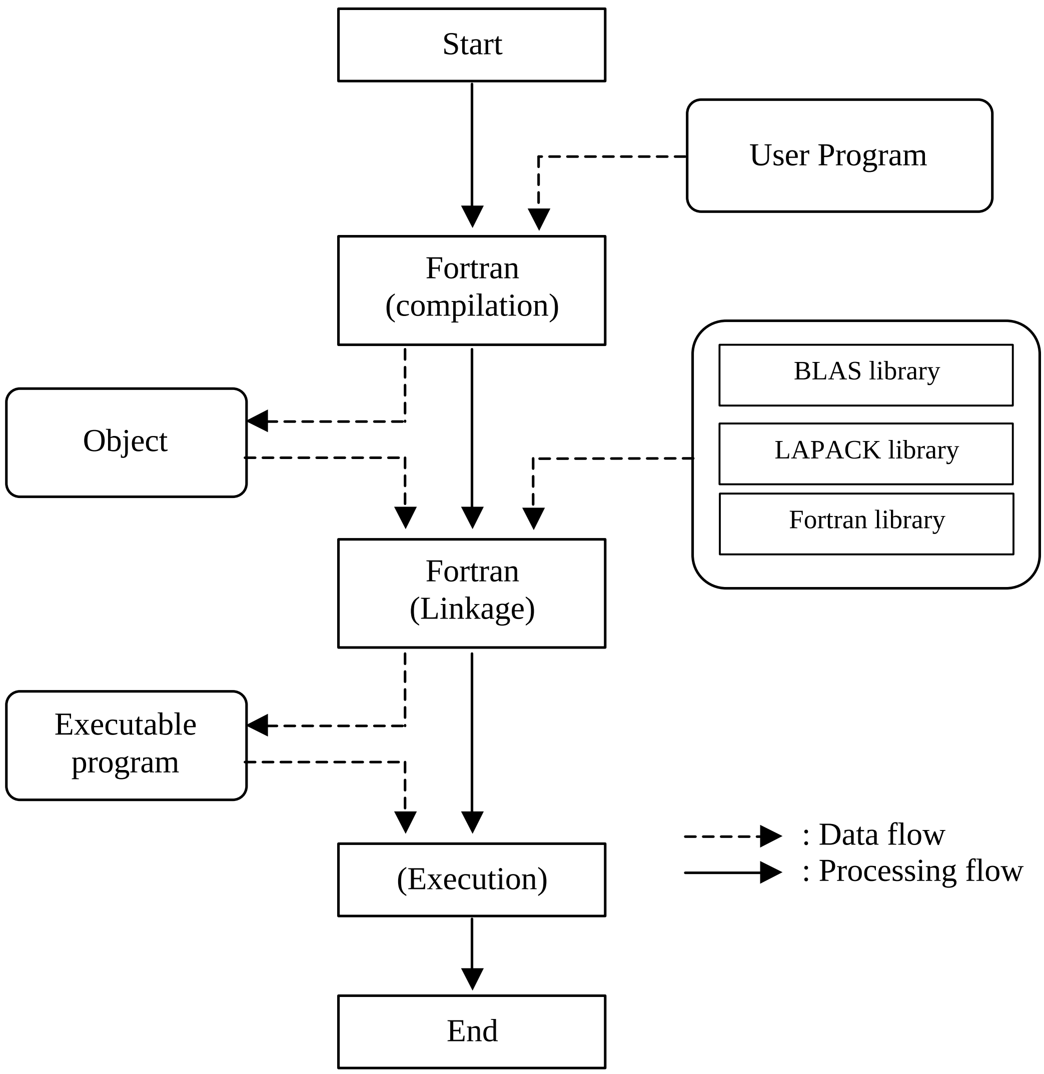

Figure 1 Flow of processing from compilation to execution (BLAS, LAPACK)

5. Notes on Use

Following are notes the user should be aware of for producing correct results from routines.

5.1 Infinity and NaN

In the LAPACK version 3.0 from netlib, a set of subroutines have been added that expect infinities and NaN (not a number) defined in the IEEE standard, to be returned as results of zero-divide or overflow and not to terminate the computation.

Lahey-Fujitsu Fortran fully conforms to the standard of such numbers. However, when the user specifies the option -trap, error conditions may occur. Make sure to avoid using the option when compiling programs that call LAPACK routines.

5.2 Routines in the archive file

The library includes slave routines, the names of which starts #L_(# means S, D, C, S, I or X). The user needs to be careful not to duplicate subroutine names with them.

Appendix A Routines L ist

In order to help the user make sure the routines interested are provided in the product, the entire lists of routines are provided here. The user is asked to check with the following lists whenever he/she feels uncertain about availability. Note, however, that the coverage of routines here does not always keep up with the latest versions from Netlib. The lists below are intended to be used to check the difference, if any, from the Netlib.

A.1 BLAS

The routines that are supplied with BLAS are listed in Table A.1 to Table A.4. The slave routines included in this software are listed in Table A.5

The placekeeper “#” may be replaced by:

S : REAL
D : DOUBLE PRECISION
C : COMPLEX
Z : COMPLEX*16

A combination of precisions means to use more than one precision. For example, “SC” of SCNRM2 is a function returning real and complex entries.

Table A.1 Level 1 BLAS routines

Routine name	Supported precision
#ROTG	S, D
#ROTMG	S, D
#ROT	S, D
#ROTM	S, D
#SWAP	S, D, C, Z
#SCAL	S, D, C, Z, CS, ZD
#COPY	S, D, C, Z
#AXPY	S, D, C, Z
#DOT	S, D, DS
#DOTU	C, Z
#DOTC	C, Z
##DOT	SDS
#NRM2	S, D, SC, DZ
#ASUM	S, D, SC, DZ
I#AMAX	S, D, C, Z

Table A.2 Level 2 BLAS routines

Routine name	Supported precision
#GEMV	S, D, C, Z
#GBMV	S, D, C, Z
#HEMV	C, Z
#HBMV	C, Z
#HPMV	C, Z
#SYMV	S, D
#SBMV	S, D
#SPMV	S, D
#TRMV	S, D, C, Z
#TBMV	S, D, C, Z
#TPMV	S, D, C, Z
#TRSV	S, D, C, Z
#TBSV	S, D, C, Z
#TPSV	S, D, C, Z
#GER	S, D
#GERU	C, Z
#GERC	C, Z
#HER	C, Z
#HPR	C, Z
#HER2	C, Z
#HPR2	C, Z
#SYR	S, D
#SPR	S, D
#SYR2	S, D
#SPR2	S, D

Table A.3 Level 3 BLAS routines

Routine name	Supported precision
#GEMM	S, D, C, Z
#SYMM	S, D, C, Z
#HEMM	C, Z
#SYRK	S, D, C, Z
#HERK	C, Z
#SYR2K	S, D, C, Z
#HER2K	C, Z
#TRMM	S, D, C, Z
#TRSM	S, D, C, Z

Table A.4 Sparse BLAS

Routine name	Supported precision
#AXPYI	S, D, C, Z
#DOTI	S, D
#DOTCI	C, Z
#DOTUI	C, Z
#GTHR	S, D, C, Z
#GTHRZ	S, D, C, Z
#SCTR	S, D, C, Z
#ROTI	S, D

Table A.5 Slave routines for BLAS

Routine name	Contents
DCABS1, LSAME, XREBLA	Included opened BLAS

A.2 LAPACK

The routines that are supplied with LAPACK are listed in Table A.6 to Table A.7.

The routine names in the following tables are the names of real and complex routines. The character indicating the supported precision is the first character of the routine name. For a double precision routine, replace the “S” of the real routine with “D”. For a double complex routine, replace the “C” of the complex routine with “Z”.

Auxiliary routine XERBLA is unique and do not depend on data type.

The symbol * shows new routines in version 3.0 of LAPACK on Netlib.

Table A.6 Driver and Computational routines of LAPACK

Real routine	Complex routine
SBDSDC*	—
SBDSQR	CBDSQR
SDISNA	—
SGBBRD	CGBBRD
SGBCON	CGBCON
SGBEQU	CGBEQU
SGBRFS	CGBRFS
SGBSV	CGBSV
SGBSVX	CGBSVX
SGBTRF	CGBTRF
SGBTRS	CGBTRS
SGEBAK	CGEBAK
SGEBAL	CGEBAL

Real routine	Complex routine
SGEBRD	CGEBRD
SGECON	CGECON
SGEEQU	CGEEQU
SGEES	CGEES
SGEESX	CGEESX
SGEEV	CGEEV
SGEEVX	CGEEVX
SGEHRD	CGEHRD
SGELQF	CGELQF
SGELS	CGELS
SGELSD *	CGELSD *
SGELSS	CGELSS
SGELSY *	CGELSY *

Table A.6 Driver and Computational routines of LAPACK (continued)

Real routine	Complex routine	Real routine	Complex routine
SGEQLF	CGEQLF	SORMBR	CUNMBR
SGEQP3 *	CGEQP3 *	SORMHR	CUNMHR
SGEQRF	CGEQRF	SORMLQ	CUNMLQ
SGERFS	CGERFS	SORMQL	CUNMQL
SGERQF	CGERQF	SORMQR	CUNMQR
SGESDD *	CGESDD *	SORMRQ	CUNMRQ
SGESV	CGESV	SORMRZ *	CUNMRZ *
SGESVD	CGESVD	SORMTR	CUNMTR
SGESVX	CGESVX	SPBCON	CPBCON
SGETRF	CGETRF	SPBEQU	CPBEQU
SGETRI	CGETRI	SPBRFS	CPBRFS
SGETRS	CGETRS	SPBSTF	CPBSTF
SGGBAK	CGGBAK	SPBSV	CPBSV
SGGBAL	CGGBAL	SPBSVX	CPBSVX
SGGES *	CGGES *	SPBTRF	CPBTRF
SGGESX *	CGGESX *	SPBTRS	CPBTRS
SGGEV *	CGGEV *	SPOCON	CPOCON
SGGEVX *	CGGEVX *	SPOEQU	CPOEQU
SGGHRD	CGGHRD	SPORFS	CPORFS
SGGGLM	CGGGLM	SPOSV	CPOSV
SGGLSE	CGGLSE	SPOSVX	CPOSVX
SGGQRF	CGGQRF	SPOTRF	CPOTRF
SGGRQF	CGGRQF	SPOTRI	CPOTRI
SGGSVD	CGGSVD	SPOTRS	CPOTRS
SGGSVP	CGGSVP	SPPCON	CPPCON
SGTCON	CGTCON	SPPEQU	CPPEQU
SGTRFS	CGTRFS	SPPRFS	CPPRFS
SGTSV	CGTSV	SPPSV	CPPSV
SGTSVX	CGTSVX	SPPSVX	CPPSVX
SGTTRF	CGTTRF	SPPTRF	CPPTRF
SGTTRS	CGTTRS	SPPTRI	CPPTRI
SHGEQZ	CHGEQZ	SPPTRS	CPPTRS
SHSEIN	CHSEIN	SPTCON	CPTCON
SHSEQR	CHSEQR	SPTEQR	CPTEQR
SOPGTR	CUPGTR	SPTRFS	CPTRFS
SOPMTR	CUPMTR	SPTSV	CPTSV
SORGBR	CUNGBR	SPTSVX	CPTSVX
SORGHR	CUNGHR	SPTTRF	CPTTRF
SORGLQ	CUNGLQ	SPTTRS	CPTTRS
SORGQL	CUNGQL	SSBEV	CHBEV
SORGQR	CUNGQR	SSBEVD	CHBEVD
SORGRQ	CUNGRQ	SSBEVX	CHBEVX
SORGTR	CUNGTR	SSBGST	CHBGST

Table A.6 Driver and Computational routines of LAPACK (continued)

Real routine	Complex routine	Real routine	Complex routine
SSBGV	CHBGV	SSYEVX	CHEEVX
SSBGVD *	CHBGVD *	SSYGST	CHEGST
SSBGVX *	CHBGVX *	SSYGV	CHEGV
SSBTRD	CHBTRD	SSYGVD *	CHEGVD *
SSPCON	CSPCON	SSYGVX *	CHEGVX *
—	CHPCON	SSYRFS	CSYRFS
SSPEV	CHPEV	—	CHERFS
SSPEVD	CHPEVD	SSYSV	CSYSV
SSPEVX	CHPEVX	—	CHESV
SSPGST	CHPGST	SSYSVX	CSYSVX
SSPGV	CHPGV	—	CHESVX
SSPGVD *	CHPGVD *	SSYTRD	CHETRD
SSPGVX *	CHPGVX *	SSYTRF	CSYTRF
SSPRFS	CSPRFS	—	CHETRF
—	CHPRFS	SSYTRI	CSYTRI
SSPSV	CSPSV	—	CHETRI
—	CHPSV	SSYTRS	CSYTRS
SSPSVX	CSPSVX	—	CHETRS
—	CHPSVX	STBCON	CTBCON
SSPTRD	CHPTRD	STBRFS	CTBRFS
SSPTRF	CSPTRF	STBTRS	CTBTRS
—	CHPTRF	STGEVC	CTGEVC
SSPTRI	CSPTRI	STGEXC *	CTGEXC *
—	CHPTRI	STGSEN *	CTGSEN *
SSPTRS	CSPTRS	STGSJA	CTGSJA
—	CHPTRS	STGSNA *	CTGSNA *
SSTEBZ	—	STGSYL *	CTGSYL *
SSTEDC	CSTEDC	STPCON	CTPCON
SSTEGR *	CSTEGR *	STPRFS	CTPRFS
SSTEIN	CSTEIN	STPTRI	CTPTRI
SSTEQR	CSTEQR	STPTRS	CTPTRS
SSTERF	—	STRCON	CTRCON
SSTEV	—	STREVC	CTREVC
SSTEVD	—	STREXC	CTREXC
SSTEVR *	—	STRRFS	CTRRFS
SSTEVX	—	STRSEN	CTRSEN
SSYCON	CSYCON	STRSNA	CTRSNA
—	CHECON	STRSYL	CTRSYL
SSYEV	CHEEV	STRTRI	CTRTRI
SSYEVD	CHEEVD	STRTRS	CTRTRS
SSYEVR *	CHEEVR *	STZRZF	CTZRZF

Table A.7 Auxiliary routines of LAPACK

Real routine	Complex routine	Real routine	Complex routine
—	CLACGV	SLAED9	—
—	CLACRM	SLAEDA	—
—	CLACRT	SLAEIN	CLAEIN
—	CLAESY	SLAEV2	CLAEV2
—	CROT	SLAEXC	—
—	CSPMV	SLAG2	—
—	CSPR	SLAGS2 *	—
—	CSROT	SLAGTF	—
—	CSYMV	SLAGTM	CLAGTM
—	CSYR	SLAGTS	—
—	ICMAX1	SLAGV2 *	—
ILAENV	—	SLAHQR	CLAHQR
LSAME	—	SLAHRD	CLAHRD
LSAMEN	—	SLAIC1	CLAIC1
—	SCSUM1	SLALN2	—
SGBTF2	CGBTF2	SLALS0 *	CLALS0 *
SGEBD2	CGEBD2	SLALSA *	CLALSA *
SGEHD2	CGEHD2	SLALSD *	CLALSD *
SGELQ2	CGELQ2	SLAMCH	—
SGEQL2	CGEQL2	SLAMRG	—
SGEQR2	CGEQR2	SLANGB	CLANGB
SGERQ2	CGERQ2	SLANGE	CLANGE
SGESC2 *	CGESC2 *	SLANGT	CLANGT
SGETC2 *	CGETC2 *	SLANHS	CLANHS
SGETF2	CGETF2	SLANSB	CLANSB
SGTTS2 *	CGTTS2 *	—	CLANHB
SLABAD	—	SLANSP	CLANSP
SLABRD	CLABRD	—	CLANHP
SLACON	CLACON	SLANST	CLANHT
SLACPY	CLACPY	SLANSY	CLANSY
SLADIV	CLADIV	—	CLANHE
SLAE2	—	SLANTB	CLANTB
SLAEBZ	—	SLANTP	CLANTP
SLAED0	CLAED0	SLANTR	CLANTR
SLAED1	—	SLANV2	—
SLAED2	—	SLAPLL	CLAPLL
SLAED3	—	SLAPMT	CLAPMT
SLAED4	—	SLAPY2	—
SLAED5	—	SLAPY3	—
SLAED6	—	SLAQGB	CLAQGB
SLAED7	CLAED7	SLAQGE	CLAQGE
SLAED8	CLAED8	SLAQP2 *	CLAQP2 *

Table A.7 Auxiliary routines of LAPACK (continued)

Real routine	Complex routine	Real routine	Complex routine
SLAQPS *	CLAQPS *	SLASQ3	—
SLAQSB	CLAQSB	SLASQ4	—
SLAQSP	CLAQSP	SLASQ5 *	—
SLAQSY	CLAQSY	SLASQ6 *	—
SLAQTR	—	SLASR	CLASR
SLAR1V *	CLAR1V *	SLASRT	—
SLAR2V	CLAR2V	SLASSQ	CLASSQ
SLARF	CLARF	SLASV2	—
SLARFB	CLARFB	SLASWP	CLASWP
SLARFG	CLARFG	SLASY2	—
SLARFT	CLARFT	SLASYF	CLASYF
SLARFX	CLARFX	—	CLAHEF
SLARGV	CLARGV	SLATBS	CLATBS
SLARNV	CLARNV	SLATDF *	CLATDF *
SLARRB *	—	SLATPS	CLATPS
SLARRE *	—	SLATRD	CLATRD
SLARRF *	—	SLATRS	CLATRS
SLARRV *	CLARRV	SLATRZ *	CLATRZ *
SLARTG	CLARTG	SLAUU2	CLAUU2
SLARTV	CLARTV	SLAUUM	CLAUUM
SLARUV	—	SORG2L	CUNG2L
SLARZ *	CLARZ *	SORG2R	CUNG2R
SLARZB *	CLARZB *	SORGL2	CUNGL2
SLARZT *	CLARZT *	SORGR2	CUNGR2
SLAS2	—	SORM2L	CUNM2L
SLASCL	CLASCL	SORM2R	CUNM2R
SLASD0 *	—	SORML2	CUNML2
SLASD1 *	—	SORMR2	CUNMR2
SLASD2 *	—	SORMR3 *	CUNMR3 *
SLASD3 *	—	SPBTF2	CPBTF2
SLASD4 *	—	SPOTF2	CPOTF2
SLASD5 *	—	SPTTS2 *	CPTTS2 *
SLASD6 *	—	SRSCL	CSRSCL
SLASD7 *	—	SSYGS2	CHEGS2
SLASD8 *	—	SSYTD2	CHETD2
SLASD9 *	—	SSYTF2	CSYTF2
SLASDA *	—	—	CHETF2
SLASDQ *	—	STGEX2 *	CTGEX2 *
SLASDT *	—	STGSY2 *	CTGSY2 *
SLASET	CLASET	STRTI2	CTRTI2
SLASQ1	—	XERBLA	—
SLASQ2	—		