LAPACK95
Users' Guidei
Draft

Vincent A. Barker[1] L. Susan Blackford[2]
Jack J. Dongarra[3] Jeremy Du Croz[4] Sven Hammarling[5]
Jerzy Waśniewski[6] Plamen Yalamov[7]

February 11, 1999

[1]Department of Mathematical Modelling (IMM), Technical University of Denmark, Building 305, DK-2800 Lyngby, Denmark; email: vab@imm.dtu.dk

[2]Department of Computer Science, University of Tennessee, 107 Ayres Hall, Knoxville, TN 37996-1301, USA

[3]Department of Computer Science, University of Tennessee, 107 Ayres Hall, Knoxville, TN 37996-1301, USA and Mathematical Sciences Section, Oak Ridge National Laboratory, P.O.Box 2008, Bldg. 6012, Oak Ridge, TN 37831-6367, USA; email: dongarra@cs.utk.edu

[4]Numerical Algorithms Group Ltd, Wilkinson House, Jordan Hill Road, Oxford OX2 8DR, UK; email: jeremy@nag.co.uk

[5]Numerical Algorithms Group Ltd, Wilkinson House, Jordan Hill Road, Oxford OX2 8DR, UK; email: sven@nag.co.uk

[6]The Danish Computing Center for Research and Education (UNI•C), Technical University of Denmark, Bldg. 304, DK-2800 Lyngby, Denmark; Email: jerzy.wasniewski@uni-c.dk

[7]Center of Applied Mathematics and Informatics, University of Rousse, Bulgaria; Email: yalamov@ami.ru.acad.bg

Contents

Chapter 1

Driver Routines for Linear Systems

1.1 General Linear Systems

1.1.1 LA_GESV

Purpose

LA_GESV computes the solution to a real or complex system of linear equations $AX = B$, where A is a square matrix and X and B are rectangular matrices or vectors. Gaussian elimination with row interchanges is used to factor A as $A = P^T LU$, where P is a permutation matrix, L is unit lower triangular, and U is upper triangular. The factored form of A is then used to solve the above system.

Specification

SUBROUTINE LA_GESV(A, B, IPIV=ipiv, **INFO**=info **)**
  type(*wp*), **INTENT(INOUT) :: A(:,:),** *rhs*
  INTEGER, INTENT(OUT), OPTIONAL :: IPIV(:)
  INTEGER, INTENT(OUT), OPTIONAL :: INFO
  where
  type **::= REAL** | **COMPLEX**
  wp **::= KIND**(1.0) | **KIND**(1.0D0)
  rhs **::= B(:,:)** | **B(:)**

Arguments

A – (*input/output*) **REAL** or **COMPLEX** square array, shape $(:,:)$.

- On entry, the matrix A.
- On exit, the factors L and U from the factorization $A = P^T LU$; the unit diagonal elements of L are not stored.

B – (*input/output*) **REAL** or **COMPLEX** array, shape (:, :) or (:), and $size(\mathbf{B}, 1) = size(\mathbf{A}, 1)$ or $size(\mathbf{B}) = size(\mathbf{A}, 1)$.

- On entry, the matrix B.
- On exit, the solution matrix X.

IPIV – *Optional* (*output*) **INTEGER** array, shape (:), $size(\mathbf{IPIV}) = size(\mathbf{A}, 1)$.

- The indices that define the permutation matrix P; row i of the matrix was interchanged with row $\mathbf{IPIV}_i$.

INFO – *Optional* (*output*) **INTEGER**.

- $= 0$: successful exit.
  < 0 : if $\mathbf{INFO} = -i$, the i^{th} argument has an illegal value.
  > 0 : if $\mathbf{INFO} = i$, then $U_{i,i} = 0$. A is singular and no solution was computed.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_GESV_EXAMPLE)

$$A = \begin{pmatrix} 0 & 2 & 3 & 5 & 4 \\ 1 & 0 & 5 & 6 & 6 \\ 7 & 6 & 8 & 0 & 5 \\ 4 & 6 & 0 & 3 & 9 \\ 5 & 9 & 0 & 0 & 8 \end{pmatrix}, \qquad B = \begin{pmatrix} 14 & 28 & 42 \\ 18 & 36 & 54 \\ 26 & 52 & 78 \\ 22 & 44 & 66 \\ 22 & 44 & 66 \end{pmatrix}$$

Arrays **A** and **B** on entry:

A

0	2	3	5	4
1	0	5	6	6
7	6	8	0	5
4	6	0	3	9
5	9	0	0	8

B

14	28	42
18	36	54
26	52	78
22	44	66
22	44	66

The call:

CALL LA_GESV(A, B)

B on exit:

B

1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000

The solution of the system $A\,X = B$ is:

$$X = \begin{pmatrix} 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \end{pmatrix}.$$

Example 2 (from Program LA_GESV_EXAMPLE)

A on entry: As in Example 1.
B on entry: $B_{:,1}$, where B is the input matrix in Example 1.

The call:

CALL LA_GESV(A, B(:,1), IPIV, INFO)

A, $\mathbf{B}(:,\mathbf{1})$, **IPIV** and **INFO** on exit:

A

7.00000	6.00000	8.00000	0.00000	5.00000
$7.14286_{10^{-1}}$	4.71429	−5.71429	0.00000	4.42857
0.00000	$4.24242_{10^{-1}}$	5.42424	5.00000	2.12121
$5.71429_{10^{-1}}$	$5.45454_{10^{-1}}$	$-2.68157_{10^{-1}}$	4.34078	4.29609
$1.42857_{10^{-1}}$	$-1.81818_{10^{-1}}$	$5.19553_{10^{-1}}$	$7.83784_{10^{-1}}$	1.62162

$\mathbf{B}(:,\mathbf{1})$

1.00000
1.00000
1.00000
1.00000
1.00000

IPIV

3
5
3
4
5

INFO = 0

Matrices L, U, P and x, where x is the solution of the system $Ax = b$:

$$L = \begin{pmatrix} 1.00000 & & & & \\ 7.14286_{10^{-1}} & 1.00000 & & & \\ 0.00000 & 4.24242_{10^{-1}} & 1.00000 & & \\ 5.71429_{10^{-1}} & 5.45454_{10^{-1}} & -2.68157_{10^{-1}} & 1.00000 & \\ 1.42857_{10^{-1}} & -1.81818_{10^{-1}} & 5.19553_{10^{-1}} & 7.83784_{10^{-1}} & 1.00000 \end{pmatrix}$$

$$U = \begin{pmatrix} 7.00000 & 6.00000 & 8.00000 & 0.00000 & 5.00000 \\ & 4.71429 & -5.71429 & 0.00000 & 4.42857 \\ & & 5.42424 & 5.00000 & 2.12121 \\ & & & 4.34078 & 4.29609 \\ & & & & 1.62162 \end{pmatrix}$$

$$P = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad x = \begin{pmatrix} 1.00000 \\ 1.00000 \\ 1.00000 \\ 1.00000 \\ 1.00000 \end{pmatrix}.$$

1.1.2 LA_GBSV

Purpose

LA_GBSV computes the solution to a real or complex system of linear equations $AX = B$, where A is a band matrix and X and B are rectangular matrices or vectors. Gaussian elimination with row interchanges is used to factor A as $A = P^T LU$, where P is a permutation matrix, L is unit lower triangular, and U is upper triangular. The factored form of A is then used to solve the above system.

Specification

SUBROUTINE LA_GBSV(AB, B, KL=kl, **IPIV**=ipiv, **INFO**=info **)**
  type(*wp*), **INTENT(INOUT) :: AB(:,:)**, *rhs*
  INTEGER, INTENT(IN), OPTIONAL :: KL
  INTEGER, INTENT(OUT), OPTIONAL :: IPIV(:)
  INTEGER, INTENT(OUT), OPTIONAL :: INFO
  type **::= REAL | COMPLEX**
  wp **::= KIND**(1.0) | **KIND**(1.0D0)
  rhs **::= B(:,:)** | **B(:)**

Arguments

AB – (*input/output*) **REAL** or **COMPLEX** rectangular array, shape (:, :), and $size(\mathbf{AB}, 1) = 2\ kl + ku + 1$, $size(\mathbf{AB}, 2) = n$, where kl and ku are, respectively, the numbers of subdiagonals and superdiagonals in the band of A, and n is the order of A.

- On entry, the matrix A in band storage. The $(kl + ku + 1)$ diagonals of A are stored rowwise in **AB**, begining in row $kl + 1$ as follows:

$$\mathbf{AB}_{kl+ku+1+i-j,j} = A_{i,j} :$$

$$\max(1, j - ku) \leq i \leq \min(n, j + kl), \;\; 1 \leq j \leq n.$$

  The remaining elements in **AB** need not be set.

- On exit, details of the factorization. U is an upper triangular band matrix with $(kl + ku + 1)$ diagonals. These are stored rowwise in the first $(kl + ku + 1)$ rows of **AB**. The multipliers that arise during the factorization are stored in the following kl rows.

B – (*input/output*) **REAL** or **COMPLEX** array, shape (:, :) or (:), $size(\mathbf{B}, 1) = n$ or $size(\mathbf{B}) = n$.

- On entry, the matrix B.
- On exit, the solution matrix X.

KL – *Optional* (*input*) **INTEGER**.

- **KL** is the number of subdiagonals in the band of A ($\mathbf{KL} = kl$).

If **KL** is not present the system assumes equal numbers of subdiagonals and superdiagonals, $kl = ku = (size(\mathbf{AB}, 1) - 1)/3$.

IPIV – *Optional* (*output*) **INTEGER** array, shape (:), $size(\mathbf{IPIV}) = n$.

- The indices that define the permutation matrix P; row i of the matrix was interchanged with row $\mathbf{IPIV}_i$.

INFO – *Optional* (*output*) **INTEGER**.

- $= 0$: successful exit.
  < 0: if $\mathbf{INFO} = -i$, the i^{th} argument has an illegal value.
  > 0: if $\mathbf{INFO} = i$, then $U_{i,i} = 0$. A is singular and no solution was computed.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

Example 1

The band storage scheme is illustrated by the following example, where $n = 9$, $kl = 2$ and $ku = 1$, making $size(\mathbf{AB}, 1) = 6$ and $size(\mathbf{AB}, 2) = 9$.

$$A = \begin{pmatrix} a_{11} & a_{12} & & & & & & & \\ a_{21} & a_{22} & a_{23} & & & & & & \\ a_{31} & a_{32} & a_{33} & a_{34} & & & & & \\ & a_{42} & a_{43} & a_{44} & a_{45} & & & & \\ & & a_{53} & a_{54} & a_{55} & a_{56} & & & \\ & & & a_{64} & a_{65} & a_{66} & a_{67} & & \\ & & & & a_{75} & a_{76} & a_{77} & a_{78} & \\ & & & & & a_{86} & a_{87} & a_{88} & a_{89} \\ & & & & & & a_{97} & a_{98} & a_{99} \end{pmatrix}$$

AB on entry

$*$	$*$	$*$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$
$*$	$*$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$
$*$	a_{12}	a_{23}	a_{34}	a_{45}	a_{56}	a_{67}	a_{78}	a_{89}
a_{11}	a_{22}	a_{33}	a_{44}	a_{55}	a_{66}	a_{77}	a_{88}	a_{99}
a_{21}	a_{32}	a_{43}	a_{54}	a_{65}	a_{76}	a_{87}	a_{89}	$*$
a_{31}	a_{42}	a_{53}	a_{64}	a_{75}	a_{86}	a_{97}	$*$	$*$

AB on exit

$*$	$*$	$*$	u_{14}	u_{25}	u_{36}	u_{47}	u_{58}	u_{69}
$*$	$*$	u_{13}	u_{24}	u_{35}	u_{46}	u_{57}	u_{68}	u_{79}
$*$	u_{12}	u_{23}	u_{34}	u_{45}	u_{56}	u_{67}	u_{78}	u_{89}
u_{11}	u_{22}	u_{33}	u_{44}	u_{55}	u_{66}	u_{77}	u_{88}	u_{99}
m_{21}	m_{32}	m_{43}	m_{54}	m_{65}	m_{76}	m_{87}	m_{98}	$*$
m_{31}	m_{42}	m_{53}	m_{64}	m_{75}	m_{86}	u_{97}	$*$	$*$

Elements marked $*$ are not referenced by the routine. Elements marked $\circ$ need not be set on entry. This extra space is needed for U because of the row interchanges.

Example 2 (from Program LA_GBSV_EXAMPLE)

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

$$A = \begin{pmatrix} 5 & 7 & & & & \\ 6 & 10 & 1 & & & \\ 3 & 4 & 6 & 5 & & \\ & 6 & 7 & 7 & 1 & \\ & & 1 & 6 & 7 & 3 \\ & & & 10 & 5 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 12 & 24 \\ 17 & 34 \\ 18 & 36 \\ 21 & 42 \\ 17 & 34 \\ 16 & 32 \end{pmatrix}.$$

Arrays **AB** and **B** on entry:

AB

	$*$	$*$	$*$	$\circ$	$\circ$	$\circ$
	$*$	$*$	$\circ$	$\circ$	$\circ$	$\circ$
	$*$	7	1	5	1	3
A	5	10	6	7	7	1
	6	4	7	6	5	$*$
	3	6	1	10	$*$	$*$

B

12	24
17	34
18	36
21	42
17	34
16	32

The call:

CALL LA_GBSV(AB, B, 2**, IPIV, INFO)**

AB, **B**, **IPIV** and **INFO** on exit:

AB

	$*$	$*$	$*$	0.00000	1.00000	0.00000
U	$*$	$*$	1.00000	7.00000	$1.66667_{10^{-1}}$	1.00000
	$*$	10.0000	7.00000	6.16667	5.00000	2.49250
	6.00000	6.00000	6.66667	10.0000	4.43750	$4.58216_{10^{-2}}$
Multipliers	$8.33333_{10^{-1}}$	$-1.66667_{10^{-1}}$	$1.08333_{10^{-1}}$	$5.07500_{10^{-1}}$	$-5.39906_{10^{-2}}$	$*$
	$5.00000_{10^{-1}}$	$-2.22222_{10^{-1}}$	$1.50000_{10^{-1}}$	$8.87500_{10^{-2}}$	$*$	$*$

B

1.00000	2.00000
1.00000	2.00000
1.00000	2.00000
1.00000	2.00000
1.00000	2.00001
$9.99993_{10^{-1}}$	2.00000

IPIV

2
4
3
6
5
6

INFO $= 0$

Matrices P, L, U and X, where X is the solution of the system $AX = B$:

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

L is obtained from P and the multipliers:

$$L = \begin{pmatrix} 1 & & & & & \\ 0.00000 & 1 & & & & \\ 5.00000_{10^{-1}} & -1.66667_{10^{-1}} & 1 & & & \\ & 0.00000 & 0.00000 & 1 & & \\ & & 1.50000_{10^{-1}} & 5.07500_{10^{-1}} & 1 & \\ 8.33333_{10^{-1}} & -2.22222_{10^{-1}} & 1.08333_{10^{-1}} & 8.87500_{10^{-2}} & -5.39906_{10^{-2}} & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} 6.00000 & 10.0000 & 1.00000 & 0.00000 & & \\ & 6.00000 & 7.00000 & 7.00000 & 1.00000 & \\ & & 6.66667 & 6.16667 & 1.66667_{10^{-1}} & 0.00000 \\ & & & 10.0000 & 5.00000 & 1.00000 \\ & & & & 4.43750 & 2.49250 \\ & & & & & 4.58216_{10^{-2}} \end{pmatrix}$$

$$X = \begin{pmatrix} 1.00000 & 2.00000 \\ 1.00000 & 2.00000 \\ 1.00000 & 2.00000 \\ 1.00000 & 2.00000 \\ 1.00000 & 2.00001 \\ 9.99993_{10^{-1}} & 2.00000 \end{pmatrix}.$$

1.1.3 LA_GTSV

Purpose

LA_GTSV computes the solution to a real or complex system of linear equations $AX = B$, where A is a tridiagonal matrix and X and B are rectangular matrices or vectors. Gaussian elimination with row interchanges is used to factor A as $A = P^T LU$, where P is a permutation matrix, L is unit lower triangular, and U is upper triangular. The factored form of A is then used to solve the above system.

The system $A^T X = B$ may be solved by interchanging the order of the arguments **DU** and **DL**.

Specification

SUBROUTINE LA_GTSV(DL, D, DU, B, INFO=info **)**
type(*wp*), **INTENT(INOUT) :: DL(:), D(:), DU(:),** *rhs*
INTEGER, INTENT(OUT), OPTIONAL :: INFO
type **::= REAL | COMPLEX**
wp **::= KIND**(1.0) | **KIND**(1.0D0)
rhs **::= B(:,:) | B(:)**

Arguments

DL – (*input/output*) **REAL** or **COMPLEX** array, shape (:), $size(\mathbf{DL}) = n-1$, where n is the order of **A**.

- On entry, the $n-1$ subdiagonal elements of A.
- On exit, the $n-2$ elements of the second superdiagonal of U in $\mathbf{DL}_1, \cdots, \mathbf{DL}_{n-2}$.

D – (*input/output*) **REAL** or **COMPLEX** array, shape (:), where $size(\mathbf{D}) = n$.

- On entry, the diagonal elements of A.
- On exit, the diagonal elements of U.

DU – (*input/output*) **REAL** or **COMPLEX** array, shape (:), where $size(\mathbf{DL}) = n-1$.

- On entry, the $n-1$ superdiagonal elements of A.
- On exit, the $n-1$ elements of the first superdiagonal of U.

B – (*input/output*) **REAL** or **COMPLEX** array, shape (:, :) or (:), and $size(\mathbf{B}, 1) = n$ or $size(\mathbf{B}) = n$.

- On entry, the matrix B.
- On exit, the solution matrix X.

INFO – *Optional* (*output*) **INTEGER**.

- $= 0$: successful exit.
  < 0 : if $\mathbf{INFO} = -i$, the i^{th} argument has an illegal value.
  > 0 : if $\mathbf{INFO} = i$, then $U_{i,i} = 0$. A is singular and no solution was computed.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Example (from Program LA_GTSV_EXAMPLE)

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

$$A = \begin{pmatrix} 2 & 2 & & & & \\ 1 & 5 & 4 & & & \\ & 4 & 8 & 4 & & \\ & & 6 & 8 & 6 & \\ & & & 6 & 4 & 3 \\ & & & & 4 & 9 \end{pmatrix}, \quad B = \begin{pmatrix} 4 & 8 & 12 \\ 10 & 20 & 30 \\ 16 & 32 & 48 \\ 20 & 40 & 60 \\ 13 & 26 & 39 \\ 13 & 26 & 39 \end{pmatrix}$$

Arrays **DL**, **D**, **DU** and **B** on entry:

DL	D	DU	B		
1	2	2	4	8	12
4	5	4	10	20	30
6	8	4	16	32	48
6	8	6	20	40	60
4	4	3	13	26	39
	9		13	26	39

The call:

CALL LA_GTSV(DL, D, DU, B, INFO)

DL, **D**, **DU**, **B** and **INFO** on exit:

DL	D	DU	B		
0	2	2	1.00000	2.00000	3.00000
0	4	4	1.00000	2.00000	3.00000
6	6	8	1.00000	2.00000	3.00000
3	6	4	1.00000	2.00000	3.00000
∘	4	9	1.00000	2.00000	3.00000
	7.66667		1.00000	2.00000	3.00000

INFO $= 0$

Element marked ∘ is not used on exit.

Matrices U and X, where X is the solution of the system $A\,X = B$:

$$U = \begin{pmatrix} 2 & 2 & 0 & & & \\ & 4 & 4 & 0 & & \\ & & 6 & 8 & 6 & \\ & & & 6 & 4 & 3 \\ & & & & 4 & 9 \\ & & & & & 7.66667 \end{pmatrix}$$

$$X = \begin{pmatrix} 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \end{pmatrix}.$$

1.2 Symmetric/Hermitian Positive Definite Linear Systems

1.2.1 LA_POSV

Purpose

LA_POSV computes the solution to a system of linear equations $AX = B$, where A is real symmetric or complex Hermitian and, in either case, positive definite, and where X and B are rectangular matrices or vectors.

The Cholesky decomposition is used to factor A as

$$A = U^H U, \ \text{if } \mathbf{UPLO} = \text{'U'}$$

or

$$A = LL^H, \ \text{if } \mathbf{UPLO} = \text{'L'}$$

where U is an upper triangular matrix and L is lower triangular ($L = U^H$). The factored form of A is then used to solve the above system.

Specification

SUBROUTINE LA_POSV(A, B, UPLO=uplo, **INFO**=info **)**
type(*wp*), **INTENT(INOUT) :: A(:,:)**, *rhs*
CHARACTER(LEN=1), INTENT(IN), OPTIONAL :: UPLO
INTEGER, INTENT(OUT), OPTIONAL :: INFO
type **::= REAL** | **COMPLEX**
wp **::= KIND**(1.0) | **KIND**(1.0D0)
rhs **::= B(:,:)** | **B(:)**

Arguments

A – (*(input/output)* **REAL** or **COMPLEX** square array, shape (:,:).

- On entry, the matrix A. If **UPLO** = 'U', the upper triangular part of **A** contains the upper triangular part of the matrix A, and the strictly lower triangular part of **A** is not referenced. If **UPLO** = 'L', the lower triangular part of **A** contains the lower triangular part of the matrix A, and the strictly upper triangular part of **A** is not referenced.
- On exit, the factor U or L from the Cholesky factorization $A = U^H U = LL^H$.

B – *(input/output)* **REAL** or **COMPLEX** array, shape (:, :) or (:), and $size(\mathbf{B}, 1) = size(\mathbf{A}, 1)$ or $size(\mathbf{B}) = size(\mathbf{A}, 1)$.

- On entry, the matrix B.
- On exit, the solution matrix X.

UPLO – *Optional, (input)* **CHARACTER(LEN=1)**

- = 'U': Upper triangle of A is stored;
  = 'L': Lower triangle of A is stored.

UPLO = 'U' is assumed if **UPLO** is not present.

INFO – *Optional (output)* **INTEGER**.

- $= 0$: successful exit
  < 0 : if **INFO** $= -i$, the i^{th} argument has an illegal value
  > 0 : if **INFO** $= i$, the leading minor of order i of A is not positive definite and no factorization or solution was computed.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_POSV_EXAMPLE)

$$A = \begin{pmatrix} 38 & 3 & 4 & 6 & 5 \\ 3 & 48 & 6 & 7 & 7 \\ 4 & 6 & 52 & 1 & 6 \\ 6 & 7 & 1 & 56 & 10 \\ 5 & 7 & 6 & 10 & 74 \end{pmatrix}, \quad B = \begin{pmatrix} 56 & 112 & 168 \\ 71 & 142 & 213 \\ 69 & 138 & 207 \\ 80 & 160 & 240 \\ 102 & 204 & 306 \end{pmatrix}.$$

Arrays **A** and **B** on entry:

A

38	3	4	6	5
∗	48	6	7	7
∗	∗	52	1	6
∗	∗	∗	56	10
∗	∗	∗	∗	74

B

56	112	168
71	142	213
69	138	207
80	160	240
102	204	306

Elements marked ∗ are not used by the routine.

The call:

CALL LA_POSV(A, B)

Arrays **A** and **B** on exit:

A

6.16441	$4.86664_{10^{-1}}$	$6.48886_{10^{-1}}$	$9.73329_{10^{-1}}$	$8.11107_{10^{-1}}$
∗	6.91109	$8.22477_{10^{-1}}$	$9.44325_{10^{-1}}$	$9.55748_{10^{-1}}$
∗	∗	7.13460	$-5.72232_{10^{-2}}$	$6.57024_{10^{-1}}$
∗	∗	∗	7.35919	1.13404
∗	∗	∗	∗	8.40898

B

1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000

Matrices U and X, where $A = U^T U$ and X is the solution of the system $AX = B$:

$$U = \begin{pmatrix} 6.16441 & 4.86664_{10^{-1}} & 6.48886_{10^{-1}} & 9.73329_{10^{-1}} & 8.11107_{10^{-1}} \\ & 6.91109 & 8.22477_{10^{-1}} & 9.44325_{10^{-1}} & 9.55748_{10^{-1}} \\ & & 7.13460 & -5.72232_{10^{-2}} & 6.57024_{10^{-1}} \\ & & & 7.35919 & 1.13404 \\ & & & & 8.40898 \end{pmatrix}$$

$$X = \begin{pmatrix} 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \end{pmatrix}.$$

Example 2 (from Program LA_POSV_EXAMPLE)

A on entry: As in Example 1.

B on entry: $B_{:,1}$, where B is the input matrix in Example 1.

A

38	∗	∗	∗	∗
3	48	∗	∗	∗
4	6	52	∗	∗
6	7	1	56	∗
5	7	6	10	74

B(:, **1**)
56
71
69
80
102

Elements marked ∗ are not used by the routine.

The call:

CALL LA_POSV(A, B(:,1)**, UPLO** = 'L')

A and **B**(:, **1**) on exit:

A

6.16441	∗	∗	∗	∗
$4.86664_{10^{-1}}$	6.91109	∗	∗	∗
$6.48886_{10^{-1}}$	$8.22477_{10^{-1}}$	7.13460	∗	∗
$9.73329_{10^{-1}}$	$9.44325_{10^{-1}}$	$-5.72232_{10^{-2}}$	7.35919	∗
$8.11107_{10^{-1}}$	$9.55748_{10^{-1}}$	$6.57024_{10^{-1}}$	1.13404	8.40898

B(:, **1**)
1.00000
1.00000
1.00000
1.00000
1.00000

Matrices L and x, where x is the solution of the system $A\,x = b$:

$$L = \begin{pmatrix} 6.16441 & & & & \\ 4.86664_{10^{-1}} & 6.91109 & & & \\ 6.48886_{10^{-1}} & 8.22477_{10^{-1}} & 7.13460 & & \\ 9.73329_{10^{-1}} & 9.44325_{10^{-1}} & -5.72232_{10^{-2}} & 7.35919 & \\ 8.11107_{10^{-1}} & 9.55748_{10^{-1}} & 6.57024_{10^{-1}} & 1.13404 & 8.40898 \end{pmatrix},$$

$$x = \begin{pmatrix} 1.00000 \\ 1.00000 \\ 1.00000 \\ 1.00000 \\ 1.00000 \end{pmatrix}.$$

1.2.2 LA_PPSV

Purpose

LA_PPSV computes the solution to a system of linear equations $A\,X = B$, where A is real symmetric or complex Hermitian and, in either case, positive definite, and where X and B are rectangular matrices or vectors. A is stored in packed format.

The Cholesky decomposition is used to factor A as

$$A = U^H U, \quad \text{if } \mathbf{UPLO} = \text{'U'}$$

or

$$A = LL^H, \quad \text{if } \mathbf{UPLO} = \text{'L'}$$

where U is an upper triangular matrix and L is lower triangular ($L = U^H$). The factored form of A is then used to solve the above system.

Specification

SUBROUTINE LA_PPSV(AP, B, UPLO=uplo, **INFO**=info **)**
type(*wp*), **INTENT(INOUT) :: AP(:)**, *rhs*
CHARACTER(LEN=1), INTENT(IN), OPTIONAL :: UPLO
INTEGER, INTENT(OUT), OPTIONAL :: INFO
type **::= REAL** | **COMPLEX**
wp **::= KIND**(1.0) | **KIND**(1.0D0)
rhs **::= B(:,:)** | **B(:)**

Arguments

AP – *(input/output)* **REAL** or **COMPLEX** array, shape (:), and $size(\mathbf{AP}) = n(n+1)/2$, where n is the order of A.

- On entry, the upper or lower triangle of matrix A in packed storage. The elements are stored columnwise in a linear array as follows:

$A_{i,j}$	i,j	**UPLO**
$\mathbf{AP}_{i+(j-1)j/2}$	$1 \le j \le n$ $1 \le i \le j$	'U'
$\mathbf{AP}_{i+(j-1)(2n-j)/2}$	$1 \le j \le n$ $j \le i \le n$	'L'

- On exit, the factor U or L from the Cholesky factorization $A = U^H U$ or $A = LL^H$, in the same storage format as A.

B – *(input/output)* **REAL** or **COMPLEX** array, shape (:, :) or (:), $size(\mathbf{B}, 1) = n$ or $size(\mathbf{B}) = n$.

- On entry, the matrix B.
- On exit, the solution matrix X.

UPLO – *Optional, (input)* **CHARACTER(LEN=1)**

- = 'U': Upper triangle of A is stored;
  = 'L': Lower triangle of A is stored.

UPLO = 'U' is assumed if **UPLO** is not present.

INFO *Optional (output)* **INTEGER**.

- $= 0$: successful exit
  < 0 : if **INFO** $= -i$, the i^{th} argument has an illegal value
  > 0 : if **INFO** $= i$, the leading minor of order i of A is not positive definite and no factorization or solution was completed.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

Example 1

The packed storage scheme is illustrated by the following example, where $n = 4$ and **UPLO** = 'U':

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

where $a_{ij} = \mathsf{conj}(a_{ji})$ for $i,j = 1,\cdots,4$.

Packed storage of the upper triangle of A:

AP on entry

a_{11}	a_{12}	a_{22}	a_{13}	a_{23}	a_{33}	a_{14}	a_{24}	a_{34}	a_{44}

Packed storage of the upper triangle matrix U:

AP on exit

u_{11}	u_{12}	u_{22}	u_{13}	u_{23}	u_{33}	u_{14}	u_{24}	u_{34}	u_{44}

where $size(\mathbf{AP}) = n(n+1)/2 = 10$.

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix}.$$

Example 2 (from Program LA_PPSV_EXAMPLE)

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

$$A = \begin{pmatrix} 38 & 3 & 4 & 6 & 5 \\ 3 & 48 & 6 & 7 & 7 \\ 4 & 6 & 52 & 1 & 6 \\ 6 & 7 & 1 & 56 & 10 \\ 5 & 7 & 6 & 10 & 74 \end{pmatrix}, \quad B = \begin{pmatrix} 56 \\ 71 \\ 69 \\ 80 \\ 102 \end{pmatrix}.$$

Arrays **AP** and **B** on entry:

AP

38	3	4	6	5	48	6	7	7	52	1	6	56	10	74

B
56
71
69
80
102

The call:

CALL LA_PPSV(AP, B, 'L' **)**

Arrays **AP** and **B** on exit:

AP

6.16441	$4.86664_{10^{-1}}$	$6.48886_{10^{-1}}$	$9.73329_{10^{-1}}$	$8.11107_{10^{-1}}$
6.91109	$8.22477_{10^{-1}}$	$9.44325_{10^{-1}}$	$9.55748_{10^{-1}}$	7.13460
$-5.72232_{10^{-2}}$	$6.57024_{10^{-1}}$	7.35919	1.13404	8.13404

B

1.00000
1.00000
1.00000
1.00000
1.00000

Matrices L and x, where x is the solution of the system $A\,x = b$:

$$L = \begin{pmatrix} 6.16441 & & & & \\ 4.86664_{10^{-1}} & 6.91109 & & & \\ 6.48886_{10^{-1}} & 8.22477_{10^{-1}} & 7.13460 & & \\ 9.73329_{10^{-1}} & 9.44325_{10^{-1}} & -5.72232_{10^{-2}} & 7.35919 & \\ 8.11107_{10^{-1}} & 9.55748_{10^{-1}} & 6.57024_{10^{-1}} & 1.13404 & 8.40898 \end{pmatrix},$$

$$x = \begin{pmatrix} 1.00000 \\ 1.00000 \\ 1.00000 \\ 1.00000 \\ 1.00000 \end{pmatrix}.$$

1.2.3 LA_PBSV

Purpose

LA_PBSV computes the solution to a system of linear equations AX =B, where A has band form and is real symmetric or complex Hermitian and, in either case, positive definite, and where X and B are rectangular matrices or vectors.

The Cholesky decomposition is used to factor A as

$$A = U^H U, \ \text{ if } \mathbf{UPLO} = \text{'U'}$$

or

$$A = LL^H, \ \text{ if } \mathbf{UPLO} = \text{'L'}$$

where U is an upper triangular band matrix and L is a lower triangular band matrix, each with the same number of superdiagonals or subdiagonals as A. The factored form of A is then used to solve the above system.

Specification

SUBROUTINE LA_PBSV(AB, B, UPLO=uplo, **INFO**=info)
type(*wp*), **INTENT(INOUT) :: AB(:,:)**, *rhs*
CHARACTER(LEN=1), INTENT(IN), OPTIONAL :: UPLO
INTEGER, INTENT(OUT), OPTIONAL :: INFO
type **::= REAL** | **COMPLEX**
wp **::= KIND**(1.0) | **KIND**(1.0D0)
rhs **::= B(:,:)** | **B(:)**

Arguments

AB – *(input/output)* **REAL** or *COMPLEX* rectangular array, shape $(:,:)$, and $size(\mathbf{AB},1) = kd+1$, $size(\mathbf{AB},2) = n$, where kd is the number subdiagonals or superdiagonals and n is the order of A.

- On entry, the upper (if **UPLO** = 'U') or lower (if **UPLO** = 'L') triangle of matrix A in band storage. The $kd+1$ diagonals of A are stored rowwise in **AB**, begining in row 1. The j^{th} column of A is stored in the j^{th} column of the array **AB** as follows:

$A_{i,j}$	i,j	**UPLO**
$\mathbf{AB}_{kd+1+i-j,j}$	$\max(1,j-kd) \leq i \leq j$ $1 \leq j \leq n$	'U'
$\mathbf{AB}_{1+i-j,j}$	$j \leq i \leq min(n,j+kd)$ $1 \leq j \leq n$	'L'

- On exit, the triangular factor U or L from the Cholesky factorization $A = U^H U = LL^H$ in the same storage format as A.

B – *(input/output)* **REAL** or **COMPLEX** array, shape $(:,:)$ or $(:)$, $size(\mathbf{B},1) = n$ or $size(\mathbf{B}) = n$.

- On entry, the matrix B.
- On exit, the solution matrix X.

UPLO – *Optional*, **(input) CHARACTER(LEN=1)**

- = 'U': Upper triangle of A is stored;
  = 'L': Lower triangle of A is stored.

UPLO = 'U' is assumed if **UPLO** is not present.

INFO *Optional (output)* **INTEGER**.

- $= 0$: successful exit
  < 0 : if **INFO** $= -i$, the i^{th} argument has an illegal value

> 0 : if **INFO** $= i$, the leading minor of order i of A is not positive definite and no factorization or solution was computed.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

Example 1

The band storage scheme is illustrated by the following example, where $n = 9$ and $kd = 2$, making $size(\mathbf{AB}, 1) = 3$ and $size(\mathbf{AB}, 2) = 9$:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & & & & & & \\ a_{21} & a_{22} & a_{23} & a_{24} & & & & & \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & & & & \\ & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & & & \\ & & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & & \\ & & & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & \\ & & & & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\ & & & & & a_{86} & a_{87} & a_{88} & a_{89} \\ & & & & & & a_{97} & a_{98} & a_{99} \end{pmatrix}$$

where $a_{ij} = \mathsf{conjg}(a_{ji})$ for $i, j = 1, \cdots, 9$.

Banded storage of A when **UPLO** = 'L':

AB on entry

a_{11}	a_{22}	a_{33}	a_{44}	a_{55}	a_{66}	a_{77}	a_{88}	a_{99}
a_{21}	a_{32}	a_{43}	a_{54}	a_{65}	a_{76}	a_{87}	a_{98}	$*$
a_{31}	a_{42}	a_{53}	a_{64}	a_{75}	a_{86}	a_{97}	$*$	$*$

Banded storage of L:

AB on exit

l_{11}	l_{22}	l_{33}	l_{44}	l_{55}	l_{66}	l_{77}	l_{88}	l_{99}
l_{21}	l_{32}	l_{43}	l_{54}	l_{65}	l_{76}	l_{87}	l_{98}	$*$
l_{31}	l_{42}	l_{53}	l_{64}	l_{75}	l_{86}	l_{97}	$*$	$*$

Elements marked $*$ are not used by the routine.

Matrix L:

$$L = \begin{pmatrix} l_{11} & & & & & & & & \\ l_{21} & l_{22} & & & & & & & \\ l_{31} & l_{32} & l_{33} & & & & & & \\ & l_{42} & l_{43} & l_{44} & & & & & \\ & & l_{53} & l_{54} & l_{55} & & & & \\ & & & l_{64} & l_{65} & l_{66} & & & \\ & & & & l_{75} & l_{76} & l_{77} & & \\ & & & & & l_{86} & l_{87} & l_{88} & \\ & & & & & & l_{97} & l_{98} & l_{99} \end{pmatrix}$$

Example 2 (from Program LA_PBSV_EXAMPLE)

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

$$A = \begin{pmatrix} 16 & 3 & 2 & 1 & & & \\ 3 & 22 & 3 & 2 & 1 & & \\ 2 & 3 & 26 & 3 & 2 & 1 & \\ 1 & 2 & 3 & 28 & 3 & 2 & 1 \\ & 1 & 2 & 3 & 26 & 3 & 2 \\ & & 1 & 2 & 3 & 22 & 3 \\ & & & 1 & 2 & 3 & 16 \end{pmatrix}, \quad B = \begin{pmatrix} 22 & 44 & 66 \\ 31 & 62 & 93 \\ 37 & 74 & 111 \\ 40 & 80 & 120 \\ 37 & 74 & 111 \\ 31 & 62 & 93 \\ 22 & 44 & 66 \end{pmatrix}.$$

AB and **B** on entry:

AB

*	*	*	1	1	1	1
*	*	2	2	2	2	2
*	3	3	3	3	3	3
16	22	26	28	26	22	16

B

22	44	66
31	62	93
37	74	111
40	80	120
37	74	111
31	62	93
22	44	66

The call:

CALL LA_PBSV(AB, B, INFO=INFO **)**

AB, **B** and **INFO** on exit:

AB

*	*	*	2.50000_{10-1}	2.15980_{10-1}	1.98307_{10-1}	1.90667_{10-1}
*	*	5.00000_{10-1}	3.91463_{10-1}	3.72332_{10-1}	3.61440_{10-1}	3.76143_{10-1}
*	7.50000_{10-1}	5.66947_{10-1}	5.26122_{10-1}	5.18530_{10-1}	5.41866_{10-1}	5.87680_{10-1}
4.00000	4.63007	5.04268	5.24475	5.05429	4.64073	3.93406

B

1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000

INFO $= 0$

Matrices U and X, where $A = U^T U$ and X is the solution of the system $AX = B$:

$$U = \begin{pmatrix} 4.00000 & 7.50000_{10-1} & 5.00000_{10-1} & 2.50000_{10-1} & & & \\ & 4.63007 & 5.66947_{10-1} & 3.91463_{10-1} & 2.15980_{10-1} & & \\ & & 5.04268 & 5.26122_{10-1} & 3.72332_{10-1} & 1.98307_{10-1} & \\ & & & 5.24475 & 5.18530_{10-1} & 3.61440_{10-1} & 1.90667_{10-1} \\ & & & & 5.05429 & 5.41866_{10-1} & 3.76143_{10-1} \\ & & & & & 4.64073 & 5.87680_{10-1} \\ & & & & & & 3.93406 \end{pmatrix}$$

$$X = \begin{pmatrix} 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 2.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \end{pmatrix}.$$

1.2.4 LA_PTSV

Purpose

LA_PTSV computes the solution to a system of linear equations $AX = B$, where A has tridiagonal form and is real symmetric or complex Hermitian and, in either case, positive definite, and where X and B are rectangular matrices or vectors. A is factored as $A = LDL^H$, where L is a unit lower bidiagonal matrix and D is a diagonal matrix. The factored form of A is then used to solve the above system.

Specification

SUBROUTINE LA_PTSV(D, E, B, INFO=info **)**
  REAL(*wp***), INTENT(INOUT) :: D(:)**
  type(*wp*)**, INTENT(INOUT) :: E(:)**, *rhs*
  INTEGER, INTENT(OUT), OPTIONAL :: INFO
  type **::= REAL** | **COMPLEX**
  wp **::= KIND**(1.0) | **KIND**(1.0D0)
  rhs **::= B(:,:)** | **B(:)**

Arguments

D – *(input/output)* **REAL** array, shape (:), and $size(\mathbf{D}) = n$, where n is the order of A.

- On entry, the n diagonal elements of the tridiagonal matrix A.
- On exit, the n diagonal elements of the diagonal matrix D in the factorization $A = LDL^H$.

E – *(input/output)* **REAL** or **COMPLEX** array, shape (:), where $size(\mathbf{E}) = size(\mathbf{D}) - 1$.

- On entry, the $n-1$ subdiagonal elements of the tridiagonal matrix A.
- On exit, the $n-1$ subdiagonal elements of the unit bidiagonal factor L in the LDL^H factorization of A.

Note: **E** can also be regarded as the superdiagonal of the unit bidiagonal factor U in the factorization $A = U^H DU$.

B – *(input/output)* **REAL** or **COMPLEX** rectangular array, shape $(:,:)$ or (:), and $size(\mathbf{B}, 1) = n$ or $size(\mathbf{B}) = n$.

- On entry, the matrix B.
- On exit, the solution matrix X.

INFO – *Optional (output)* **INTEGER**.

- $= 0$: successful exit
  < 0 : if **INFO** $= -i$, the i^{th} argument has an illegal value
  > 0 : if **INFO** $= i$, then $U_{i,i} \leq 0$. A is not positive definite or Hermitian, and no solution was computed.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Example (from Program LA_PTSV_EXAMPLE)

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

$$A = \begin{pmatrix} 7 & 3 & & & \\ 3 & 7 & 3 & & \\ & 3 & 7 & 3 & \\ & & 3 & 7 & 3 \\ & & & 3 & 7 \end{pmatrix}, \quad B = \begin{pmatrix} 10 & 20 & 30 \\ 13 & 26 & 39 \\ 13 & 26 & 39 \\ 13 & 26 & 39 \\ 10 & 20 & 30 \end{pmatrix}$$

Arrays **D**, **E** and **B** on entry:

D

7
7
7
7
7

E

3
3
3
3

B

10	20	30
13	26	39
13	26	39
13	26	39
10	20	30

The call:

```
CALL LA_PTSV( D, E, B, INFO )
```

D, **E**, **B** and **INFO** on exit:

D

7.00000
5.71429
5.42500
5.34101
5.31493

E

$4.28571_{10^{-1}}$
$5.25000_{10^{-1}}$
$5.52995_{10^{-1}}$
$5.61691_{10^{-1}}$

B

1.00000	2.00000	3.00000
1.00000	2.00000	2.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000

INFO $= 0$

Matrices L, U and X, where $A = L\,D\,L^H$ and X is the solution of the system $A\,X = B$:

$$L = \begin{pmatrix} 7.00000 & & & & \\ 4.28571_{10^{-1}} & 5.71429 & & & \\ & 5.25000_{10^{-1}} & 5.42500 & & \\ & & 5.52995 & 5.34101 & \\ & & & 5.61691_{10^{-1}} & 5.31493 \end{pmatrix}$$

$$U = \begin{pmatrix} 7.00000 & 4.28571_{10^{-1}} & & & \\ & 5.71429 & 5.25000_{10^{-1}} & & \\ & & 5.42500 & 5.52995_{10^{-1}} & \\ & & & 5.34101 & 5.61691_{10^{-1}} \\ & & & & 5.31493 \end{pmatrix}$$

$$X = \begin{pmatrix} 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 2.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 2.00000 \\ 1.00000 & 2.00000 & 3.00000 \end{pmatrix}.$$

1.3 Symmetric Indefinite Linear Systems

1.3.1 LA_SYSV / LA_HESV

Purpose

LA_SYSV computes the solution to a system of linear equations $A\,X = B$, where A is a real symmetric matrix and X and B are rectangular matrices or vectors. A diagonal pivoting method is used to factor A as

$$A = P^T\,U\,D\,U^T\,P, \ \text{ if } \mathbf{UPLO} = \text{'U'}$$

or

$$A = P^T\,L\,D\,L^T\,P, \ \text{ if } \mathbf{UPLO} = \text{'L'}$$

where P is a permutation matrix, U and L are unit upper and lower triangular matrices, respectively, and D is a symmetric block diagonal with 1×1 and 2×2 diagonal blocks. The factored form of A is then used to solve the above system.

LA_HESV computes the solution to a system of linear equations $A\,X = B$, where A is a complex Hermitian matrix and X and B are rectangular matrices or vectors. A diagonal pivoting method is used to factor A as

$$A = P^T\,U\,D\,U^H\,P, \text{ if } \mathbf{UPLO} = \text{'U'}$$

or

$$A = P^T\,L\,D\,L^H\,P, \text{ if } \mathbf{UPLO} = \text{'L'}$$

where P is a permutation matrix, U and L are unit upper and lower triangular matrices, respectively, and D is a complex Hermitian block diagonal with 1×1 and 2×2 diagonal blocks. The factored form of A is then used to solve the above system.

Specification

SUBROUTINE LA_SYSV / LA_HESV(A, B, UPLO=uplo, **&**
IPIV=ipiv, **INFO**=info **)**
type(*wp*)**, INTENT(INOUT) :: A(:,:),** *rhs*
CHARACTER(LEN=1), INTENT(IN), OPTIONAL :: UPLO
INTEGER, INTENT(OUT), OPTIONAL :: IPIV(:)
INTEGER, INTENT(OUT), OPTIONAL :: INFO
type **::= REAL | COMPLEX**
wp **::= KIND**(1.0) | **KIND**(1.0D0)
rhs **::= B(:,:) | B(:)**

Arguments

A – *(input/output)* **REAL** or **COMPLEX** square array, shape $(:,:)$.

- On entry, the matrix A. If **UPLO** = 'U', the upper triangular part of array **A** contains the upper triangular part of the matrix A, and the strictly lower triangular part of **A** is not referenced. If **UPLO** = 'L', the lower triangular part of **A** contains the lower triangular part of A, and the strictly upper triangular part of **A** is not referenced.
- On exit, the block diagonal matrix D and the multipliers used to obtain U or L in the factorization of A.

B – *(input/output)* **REAL** or **COMPLEX** rectangular array, shape $(:,:)$ or $(:)$, and $size(\mathbf{B},1) = size(\mathbf{A},1)$ or $size(\mathbf{B}) = size(\mathbf{A},1)$.

- On entry, the matrix B.
- On exit, the solution matrix X.

UPLO – *Optional, (input)* **CHARACTER(LEN=1)**

- = 'U': Upper triangle of A is stored;

= 'L': Lower triangle of A is stored.

UPLO = 'U' is assumed if **UPLO** is not present.

IPIV – *Optional (output)* **INTEGER** array, shape (:), $size(\mathbf{IPIV}) = size(\mathbf{A}, 1)$.

- The indices that define the permutation matrix P and details of the block structure of D, as determined by **LA_SYTRF** or **LA_HETRF**.
  - If $\mathbf{IPIV}_k > 0$, then rows and columns k and $\mathbf{IPIV}_k$ are interchanged, and $D_{k,k}$ is a 1×1 diagonal block.
  - If $\mathbf{IPIV}_k < 0$ and
    * If **UPLO** = 'U' and $\mathbf{IPIV}_k = \mathbf{IPIV}_{k-1} < 0$, then rows and columns $k-1$ and $-\mathbf{IPIV}_k$ are interchanged and $D_{k-1:k,k-1:k}$ is a 2×2 diagonal block.
    * If **UPLO** = 'L' and $\mathbf{IPIV}_k = \mathbf{IPIV}_{k+1} < 0$, then rows and columns $k+1$ and $-\mathbf{IPIV}_k$ are interchanged and $D_{k:k+1,k:k+1}$ is a 2×2 diagonal block.

INFO – *Optional (output)* **INTEGER**.

- $= 0$: successful exit

  < 0 : if $\mathbf{INFO} = -k$, the k^{th} argument has an illegal value

  > 0 : if $\mathbf{INFO} = k$, $D_{k,k} = 0$. The factorization has been completed, but the matrix D is singular, and no solution was computed.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_SYSV_EXAMPLE)

$$A = \begin{pmatrix} 0 & 2 & 3 & 5 & 4 \\ 2 & 0 & 5 & 6 & 6 \\ 3 & 5 & 8 & 0 & 5 \\ 5 & 6 & 0 & 3 & 9 \\ 4 & 6 & 5 & 9 & 8 \end{pmatrix}, \qquad B = \begin{pmatrix} 14 & 28 & 42 \\ 19 & 38 & 57 \\ 21 & 42 & 63 \\ 23 & 46 & 69 \\ 32 & 64 & 96 \end{pmatrix}$$

Arrays **A** and **B** on entry:

A

0	2	3	5	4
*	0	5	6	6
*	*	8	0	5
*	*	*	3	9
*	*	*	*	8

B

14	28	42
18	36	54
26	52	78
22	44	66
22	44	66

Elements marked * are not used by the routine.

The call:

CALL LA_SYSV(A, B, IPIV=IPIV)

A, **B** and **IPIV** on exit:

A

-1.72531	$2.24321_{10^{-1}}$	$1.12994_{10^{-2}}$	$-7.01754_{10^{-2}}$	$5.00000_{10^{-1}}$
*	-4.78531	$1.97740_{10^{-1}}$	$1.05263_{10^{-1}}$	$7.50000_{10^{-1}}$
*	*	9.31579	$7.89474_{10^{-1}}$	$6.25000_{10^{-1}}$
*	*	*	-7.12500	1.12500
*	*	*	*	8.00000

B

1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000
1.00000	2.00000	3.00000

IPIV

1
2
3
4
5

Matrices U, D, P and X, where $A = P^T U\, D\, U^T P$ and X is the solution of the system $A\,X = B$:

$$U = \begin{pmatrix} 1 & 2.24321_{10^{-1}} & 1.12994_{10^{-2}} & -7.01754_{10^{-2}} & 5.00000_{10^{-1}} \\ 0 & 1 & 1.97740_{10^{-1}} & 1.05263_{10^{-1}} & 7.50000_{10^{-1}} \\ 0 & 0 & 1 & 7.89474_{10^{-1}} & 6.25000_{10^{-1}} \\ 0 & 0 & 0 & 1 & 1.12500 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$D = \begin{pmatrix} -1.72531 & 0 & 0 & 0 & 0 \\ 0 & -4.78531 & 0 & 0 & 0 \\ 0 & 0 & 9.31579 & 0 & 0 \\ 0 & 0 & 0 & -7.12500 & 0 \\ 0 & 0 & 0 & 0 & 8.00000 \end{pmatrix}$$

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$X = \begin{pmatrix} 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \\ 1.00000 & 2.00000 & 3.00000 \end{pmatrix}.$$

Example 2 (from Program LA_SYSV_EXAMPLE)

Matrix A as in Example 1 and the first column of matrix B in Example 1.

Arrays **A** and $\mathbf{B}(:,\mathbf{1})$ on entry:

A

0	∗	∗	∗	∗
2	0	∗	∗	∗
3	5	8	∗	∗
5	6	0	3	∗
4	6	5	9	8

$\mathbf{B}(:,\mathbf{1})$

14
19
21
23
32

Elements marked ∗ are not used by the routine.
The call:

CALL LA_SYSV(A, B(:,1), 'L', **IPIV, INFO)**

A, **B**(:, **1**), **IPIV** and **INFO** on exit:

A

0.00000	*	*	*	*
5.00000	3.00000	*	*	*
$-3.60000_{10^{-1}}$	$6.00000_{10^{-1}}$	9.08000	*	*
$9.60000_{10^{-1}}$	$4.00000_{10^{-1}}$	$2.33480_{10^{-1}}$	−4.81498	*
1.32000	8.00000	$1.14537_{10^{-1}}$	$3.49497_{10^{-1}}$	−4.01098

B(:, **1**)
1.00000
1.00000
1.00000
1.00000
1.00000

IPIV
−4
−4
3
4
5

INFO = 0

Matrices L, D, P and x, where $A = P^T L\, D\, L^T P$ and x is the solution of the system $A\,x = b$:

$$L = \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ -3.60000_{10^{-1}} & 6.00000_{10^{-1}} & 1 & & \\ 9.60000_{10^{-1}} & 4.00000_{10^{-1}} & 2.33480_{10^{-1}} & 1 & \\ 1.32000 & 8.00000_{10^{-1}} & 1.14537_{10^{-1}} & 3.49497_{10^{-1}} & 1 \end{pmatrix}$$

$$D = \begin{pmatrix} 0.00000 & 5.00000 & & & \\ 5.00000 & 3.00000 & & & \\ & & 9.08000 & & \\ & & & -4.81498 & \\ & & & & -4.01098 \end{pmatrix}$$

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \qquad x = \begin{pmatrix} 1.00000 \\ 1.00000 \\ 1.00000 \\ 1.00000 \\ 1.00000 \end{pmatrix}.$$

Example 3 (from Program LA_HESV_EXAMPLE)

$$A = \begin{pmatrix} 1 & 4+6i & 5+7i & 10+8i & 3+10i \\ 4-6i & 9 & 6+10i & 3+i & 8+8i \\ 5-7i & 6-10i & 9 & 8+4i & 7+i \\ 10-8i & 3-i & 8-4i & 7 & 7+9i \\ 3-10i & 8-8i & 7-i & 7-9i & 3 \end{pmatrix}, \; B = \begin{pmatrix} -8+54i \\ 17+43i \\ 47+23i \\ 39+31i \\ 56 \end{pmatrix}$$

Arrays **A** and **B** on entry:

A

$(1,0)$	$(4,6)$	$(5,7)$	$(10,8)$	$(3,10)$
$*$	$(9,0)$	$(6,10)$	$(3,1)$	$(8,8)$
$*$	$*$	$(9,0)$	$(8,4)$	$(7,1)$
$*$	$*$	$*$	$(7,0)$	$(7,9)$
$*$	$*$	$*$	$*$	$(3,0)$

B

$(-8,54)$
$(17,43)$
$(47,23)$
$(39,31)$
$(56,0)$

Elements marked $*$ are not used by the routine.

The call:

CALL LA_HESV(A, B, IPIV=IPIV **)**

A, **B** and **IPIV** on exit:

A

$(-11.2808,0)$	$(4.42315_{10^{-1}},3.02316_{10^{-1}})$	$(1.40419_{10^{-1}},-4.40477_{10^{-1}})$
$*$	$(11.5941,0)$	$(5.60396,-12.7129)$
$*$	$*$	$(3.86140_{10^{-1}},0)$
$*$	$*$	$*$
$*$	$*$	$*$

A(continued)

$(9.10891_{10^{-1}},3.76238_{10^{-1}})$	$(-4.25743_{10^{-1}},-9.90099_{10^{-2}})$
$(1.17822,1.88119_{10^{-1}})$	$(-3.06931_{10^{-1}},-6.43564_{10^{-1}})$
$(4.55446_{10^{-1}},-1.78218_{10^{-1}})$	$(6.43564_{10^{-1}},-4.05941_{10^{-1}})$
$(9.00000,0)$	$(8.00000,8.00000)$
$*$	$(3.00000,0)$

B	IPIV
(1.00000, 1.00000)	1
(1.00000, 1.00000)	−2
(1.00000, 1.00000)	−2
(1.00000, 1.00000)	−2
(1.00000, 1.00000)	−2

Matrices U, D, P and X, where $A = P^T U\, D\, U^H P$ and X is the solution of the system $A\,X = B$:

$$U = \begin{pmatrix} 1 & 4.42315_{10^{-1}} + 3.02316_{10^{-1}}i & 1.40419_{10^{-1}} - 4.40477_{10^{-1}}i \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix.$$

$$\begin{matrix} 9.10891_{10^{-1}} + 3.76238_{10^{-1}}i & -4.25743_{10^{-1}} - 9.90099_{10^{-2}}i \\ 1.17822 + 1.88119_{10^{-1}}i & -3.06931_{10^{-1}} - 6.43564_{10^{-1}}i \\ 4.55446_{10^{-1}} - 1.78218_{10^{-1}}i & 6.43564_{10^{-1}} - 4.05941_{10^{-1}}i \\ 1 & 0 \\ 0 & 1 \end{matrix}\right)$$

$$D = \begin{pmatrix} -11.2808 & 0 & 0 \\ 0 & 11.5941 & 5.60396 - 12.7129i \\ 0 & 5.60396 + 12.7129i & 3.86140_{10^{-1}} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix.$$

$$\begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 9.00000 & 8.00000 + 8.00000i \\ 8.00000 - 8.00000i & 3.00000 \end{matrix}\right)$$

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \qquad X = \begin{pmatrix} 1.00000 + 1.00000i \\ 1.00000 + 1.00000i \\ 1.00000 + 1.00000i \\ 1.00000 + 1.00000i \\ 1.00000 + 1.00000i \end{pmatrix}.$$

1.3.2 LA_SPSV / LA_HPSV

Purpose

LA_SPSV computes the solution to a system of linear equations $AX = B$, where A is real symmetric and X and B are rectangular matrices or vectors. A diagonal pivoting method is used to factor A as

$$A = P^T\, U\, D\, U^T\, P, \ \text{ if } \ \textbf{UPLO} = \text{'U'}$$

or

$$A = P^T\, L\, D\, L^T\, P, \ \text{ if } \ \textbf{UPLO} = \text{'L'}$$

where P is a permutation matrix, U and L are unit upper and lower triangular matrices, respectively, and D is a symmetric block diagonal with 1×1 and 2×2 diagonal blocks. The factored form of A is then used to solve the above system. A is stored in packed format.

LA_HPSV computes the solution to a system of linear equations $AX = B$, where A is complex Hermitian and X and B are rectangular matrices or vectors. A diagonal pivoting method is used to factor A as

$$A = P^T\, U\, D\, U^H\, P, \ \text{ if } \ \textbf{UPLO} = \text{'U'}$$

or

$$A = P^T\, L\, D\, L^H\, P, \ \text{ if } \ \textbf{UPLO} = \text{'L'}$$

where P is a permutation matrix, U and L are unit upper and lower triangular matrices, respectively, and D is a complex Hermitian block diagonal with 1×1 and 2×2 diagonal blocks. The factored form of A is then used to solve the above system. A is stored in packed format.

Specification

SUBROUTINE LA_SPSV / LA_HESV(AP, B, UPLO=uplo, **&**
IPIV=ipiv, **INFO**=info **)**
type(*wp*), **INTENT(INOUT) :: AP(:),** *rhs*
CHARACTER(LEN=1), INTENT(IN), OPTIONAL :: UPLO
INTEGER, INTENT(OUT), OPTIONAL :: IPIV(:)
INTEGER, INTENT(OUT), OPTIONAL :: INFO
type **::= REAL | COMPLEX**
wp **::= KIND**(1.0) | **KIND**(1.0D0)
rhs **::= B(:,:) | B(:)**

Arguments

AP – *(input/output)* **REAL** or **COMPLEX** array, shape (:), and $size(A) = n(n+1)/2$, where n is the order of the matrix A.

- On entry, the upper or lower triangle of the real symmetric or complex Hermitian matrix A, packed columnwise in a linear array. The j^{th} column of A is stored in array **AP** as follows:

$A_{i,j}$	i,j	**UPLO**
$\mathbf{AP}_{i+(j-1)j/2}$	$1 \le j \le n$ $1 \le i \le j$	'U'
$\mathbf{AP}_{i+(j-1)(2n-j)/2}$	$1 \le j \le n$ $j \le i \le n$	'L'

- On exit, the block diagonal matrix D and the multipliers used to obtain U or L in the factorization of A stored as a packed triangular matrix in the same storage format as A.

B – *(input/output)* **REAL** or **COMPLEX** rectangular array, shape $(:,:)$ or $(:)$, and $size(B,1) = n$ or $size(B) = n$, where n is the order of matrix A.

- On entry, the matrix B.
- On exit, the solution matrix X.

UPLO – *Optional, (input)* **CHARACTER(LEN=1)**

- = 'U': Upper triangle of A is stored;
  = 'L': Lower triangle of A is stored.

UPLO = 'U' is assumed if **UPLO** is not present.

IPIV – *Optional (output)* **INTEGER** array, shape $(:)$, $size(\mathbf{IPIV}) = n$.

- The indices that define the permutation matrix P and details of the block structure of D, as determined by **LA_SPTRF** or **LA_HPTRF**.
  - If $\mathbf{IPIV}_k > 0$, then rows and columns k and $\mathbf{IPIV}_k$ are interchanged, and $D_{k,k}$ is a 1×1 diagonal block.
  - If $\mathbf{IPIV}_k < 0$ and
    * If **UPLO** = 'U' and $\mathbf{IPIV}_k = \mathbf{IPIV}_{k-1} < 0$, then rows and columns $k-1$ and $-\mathbf{IPIV}_k$ are interchanged and $D_{k-1:k,k-1:k}$ is a 2×2 diagonal block.
    * If **UPLO** = 'L' and $\mathbf{IPIV}_k = \mathbf{IPIV}_{k+1} < 0$, then rows and columns $k+1$ and $-\mathbf{IPIV}_k$ are interchanged and $D_{k:k+1,k:k+1}$ is a 2×2 diagonal block.

INFO – *Optional (output)* **INTEGER**.

- $= 0$: successful exit
  < 0 : if **INFO** $= -k$, the k^{th} argument has an illegal value
  > 0 : if **INFO** $= k$, $D_{k,k} = 0$. The factorization has been completed, but the matrix D is singular, and no solution was computed.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

Example 1

The packed storage scheme is illustrated by the following example, where $n = 4$ and **UPLO** = 'U':

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

where $a_{ij} = a_{ji}$ for $i, j = 1, \cdots, 4$.

Packed storage of the upper triangle of A:

AP on entry

a_{11}	a_{12}	a_{22}	a_{13}	a_{23}	a_{33}	a_{14}	a_{24}	a_{34}	a_{44}

AP on exit

d_{11}	u_{12}	d_{22}	u_{13}	u_{23}	d_{33}	u_{14}	u_{24}	u_{34}	d_{44}

where $size(\mathbf{AP}) = n(n + 1)/2 = 10$.

Example 2 (from Program LA_SPSV_EXAMPLE)

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

$$A = \begin{pmatrix} 0 & 2 & 3 & 5 & 4 \\ 2 & 0 & 5 & 6 & 6 \\ 3 & 5 & 8 & 0 & 5 \\ 5 & 6 & 0 & 3 & 9 \\ 4 & 6 & 5 & 9 & 8 \end{pmatrix}, \qquad B = \begin{pmatrix} 14 \\ 19 \\ 63 \\ 69 \\ 96 \end{pmatrix}$$

Arrays **AP** and **B** on entry:

AP

0	2	3	5	4	0	5	6	6	8	0	5	3	9	8

B

14
19
63
69
96

The call:

CALL LA_SPSV(AP, B, 'L', IPIV)

AP and **B** on exit:

AP

0.00000	5.00000	$-3.60000_{10^{-1}}$	$9.60000_{10^{-1}}$	1.32000	3.00000
$6.00000_{10^{-1}}$	$4.00000_{10^{-1}}$	$8.00000_{10^{-1}}$	9.80000	$2.33480_{10^{-1}}$	
$1.14537_{10^{-1}}$	-4.81498	$3.49497_{10^{-1}}$	-4.01098		

B	IPIV
1.00000	−4
1.00000	−4
1.00000	3
1.00000	4
1.00000	5

Matrices L, D, P and x, where $A = P^T L\,D\,L^T P$ and x is the solution of the system $A\,x = b$:

$$L = \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ -3.60000_{10^{-1}} & 6.00000_{10^{-1}} & 1 & & \\ 9.60000_{10^{-1}} & 4.00000_{10^{-1}} & 2.33480_{10^{-1}} & 1 & \\ 1.32000 & 8.00000_{10^{-1}} & 1.14537_{10^{-1}} & 3.49497_{10^{-1}} & 1 \end{pmatrix}$$

$$D = \begin{pmatrix} 0.00000 & 5.00000 & & & \\ 5.00000 & 3.00000 & & & \\ & & 9.08000 & & \\ & & & -4.81498 & \\ & & & & -4.01098 \end{pmatrix}$$

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \qquad x = \begin{pmatrix} 1.00000 \\ 1.00000 \\ 1.00000 \\ 1.00000 \\ 1.00000 \end{pmatrix}.$$

Chapter 2

Driver Routines for Least Squares Problems

2.1 Linear Least Squares Problems

2.1.1 LA_GELS

Purpose

LA_GELS solves overdetermined and underdetermined linear systems of equations $AX = B$, $A^TX = B$ and $A^HX = B$ using a QR or LQ factorization of A. Matrix A is real or complex, rectangular, and of full rank, and B is a real or complex vector or rectangular matrix.

Specification

SUBROUTINE LA_GELS(A, B, TRANS=trans, **INFO**=info **)**
type(*wp*), **INTENT(INOUT) :: A(:, :)**, *rhs*
CHARACTER(LEN=1), INTENT(IN), OPTIONAL :: TRANS
INTEGER, INTENT(OUT), OPTIONAL :: INFO
type **::= REAL** | **COMPLEX**
wp **::= KIND(1.0)** | **KIND(1.0D0)**
rhs **::= B(:,:)** | **B(:)**

Arguments

A – (*Input/Output*) **REAL** or **COMPLEX** rectangular array, shape $(:,:)$.

- On entry, the matrix A.
- On exit:

- If $size(\mathbf{A},1) \geq size(\mathbf{A},2)$, **A** is overwritten by details of its QR factorization as returned by **LA_GEQRF**.
- If $size(\mathbf{A},1) < size(\mathbf{A},2)$, **A** is overwritten by details of its LQ factorization as returned by **LA_GELQF**.

B – (*input/output*) **REAL** or **COMPLEX** array, shape (:,:) or (:), and $size(\mathbf{B},1) = \max(size(\mathbf{A},1), size(\mathbf{A},2))$ or $size(\mathbf{B}) = \max(size(\mathbf{A},1), size(\mathbf{A},2))$, and $size(\mathbf{B},2)$ is the number of right-hand side vectors.

- On entry, the matrix B.
- On exit, **B** is overwritten by the solution matrx **X** as follows:
  - If **TRANS** = 'N' and $size(\mathbf{A},1) \geq size(\mathbf{A},2)$, rows 1 to $size(\mathbf{A},2)$ of **B** contain, columnwise, the least squares solution vector(s); the residual sum of squares for the solution vector in each column is given by the sum of squares of elements in rows $size(\mathbf{A},2)+1$ to $size(\mathbf{A},1)$ of that column.
  - If **TRANS** = 'N' and $size(\mathbf{A},1) < size(\mathbf{A},2)$, rows 1 to $size(\mathbf{A},2)$ of B contain, columnwise, the minimum norm solution vector(s).
  - If (**TRANS** = 'T' or **TRANS** = 'C') and $size(\mathbf{A},1) \geq size(\mathbf{A},2)$, rows 1 to $size(\mathbf{A},1)$ of B contain, columnwise, the minimum norm solution vector(s).
  - If (**TRANS** = 'T' or **TRANS** = 'C') and $size(\mathbf{A},1) < size(\mathbf{A},2)$, rows 1 to $size(\mathbf{A},1)$ of B contain, columnwise, the least squares solution vector(s); the residual sum of squares for the solution vector in each column is given by the sum of squares of elements in rows $size(\mathbf{A},1)+1$ to $size(\mathbf{A},2)$ of that column.

TRANS – *Optional* (*Input*) **CHARACTER(LEN=1)**.

- Specifies the form of the system of equations:
  = 'N': the system solved is $AX = B$;
  = 'T': the system solved is $A^T X = B$;
  = 'C': the system solved is $A^H X = B$;

  TRANS = 'N' is assumed if **TRANS** is not present.

INFO – *Optional* (*Output*) **INTEGER**.

- = 0: successful exit
  < 0: if **INFO** = $-i$, the i^{th} argument had an illegal value

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_GELS_EXAMPLE)

$$A = \begin{pmatrix} -3 & -1 & 6 & -5 \\ 5 & 4 & -6 & 8 \\ 7 & 5 & 0 & -4 \\ -7 & 4 & 0 & 3 \\ -7 & 7 & -8 & 2 \\ 3 & 4 & 2 & -4 \end{pmatrix}, \qquad B = \begin{pmatrix} -3 & -6 & 9 \\ 1 & 2 & 3 \\ 8 & -6 & 2 \\ -2 & 0 & 7 \\ 6 & -2 & -8 \\ -5 & 10 & 1 \end{pmatrix}$$

Arrays **A** and **B** on entry:

A

−3	−1	6	−5
5	4	−6	8
7	5	0	−4
−7	4	0	3
−7	7	−8	2
3	4	2	−4

B

−3	−6	9
1	2	3
8	−6	2
−2	0	7
6	−2	−8
−5	10	1

The call:

CALL LA_GELS(A, B)

B on exit:

B

$1.48869_{10^{-1}}$	$1.86313_{10^{-1}}$	$7.37811_{10^{-2}}$
$-8.41163_{10^{-2}}$	$1.96520_{10^{-1}}$	1.07181
-1.18204	$3.09642_{10^{-1}}$	2.28277
$-7.33727_{10^{-1}}$	$4.17938_{10^{-1}}$	1.32373
1.56013	2.28673	-3.05718
-7.29547	12.6517	-3.71209

The least squares solution of the system $AX = B$ is:

$$X = \begin{pmatrix} 1.48869_{10^{-1}} & 1.86313_{10^{-1}} & 7.37811_{10^{-2}} \\ -4.81163_{10^{-2}} & 1.96520_{10^{-1}} & 1.07181 \\ -1.18204 & 3.09642_{10^{-1}} & 2.28277 \\ -7.33727_{10^{-1}} & 4.17938_{10^{-1}} & 23.1260 \end{pmatrix}.$$

The residual sums-of-square are:

$$\begin{pmatrix} 55.6579 & 165.295 & 23.1260 \end{pmatrix}.$$

Example 2 (from Program LA_GELS_EXAMPLE)

A on entry: As in Example 1.
B on entry: $B_{1:4,1}$, where B is the input matrix in Example 1.

The call:

CALL LA_GELS(A, B(1:4,1), TRANS='T', INFO)

$\mathbf{B(:,1)}$ and **INFO** on exit:

$\mathbf{B(:,1)}$
$5.85640_{10^{-1}}$
$7.96301_{10^{-2}}$
$-8.99154_{10^{-3}}$
$9.08716_{10^{-1}}$
$-5.32638_{10^{-1}}$
$3.51419_{10^{-1}}$

INFO $= 0$

The minimum-norm solution of the system $A^T x = b$ is:

$$x = \begin{pmatrix} 5.85640_{10^{-1}} \\ 7.96301_{10^{-2}} \\ -8.99154_{10^{-3}} \\ 9.08716_{10^{-1}} \\ -5.32638_{10^{-1}} \\ 3.51419_{10^{-1}} \end{pmatrix}.$$

2.1.2 LA_GELSX

Purpose

LA_GELSX computes the minimum-norm solution to one or more linear least squares problems

$$\min_x \|b - A\,x\|_2$$

using a complete orthogonal factorization of A. A is a real or complex rectangular matrix which may be rank-deficient, and b and x are columns of real or complex rectangular matrices B and X, respectively.

The routine first computes a QR factorization of A with column pivoting:

$$A\,P = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}$$

with R_{11} defined as the largest leading submatrix whose estimated condition number is less than 1/**RCOND**. The order of R_{11}, **RANK**, is the effective rank of A.

Then, R_{22} is considered to be negligible, and R_{12} is annihilated by orthogonal transformations from the right, arriving at the complete orthogonal factorization:

$$A\ P = Q \begin{pmatrix} T_{11} & 0 \\ 0 & 0 \end{pmatrix} Z$$

The minimum-norm solution is then

$$X = P\ Z^H \begin{pmatrix} T_{11}^{-1} Q_1^H B \\ 0 \end{pmatrix}$$

where Q_1 consists of the first **RANK** columns of Q.

Specification

SUBROUTINE LA_GELSX(A, B, RANK, JPVT=jpvt, **&**
RCOND=rcond, **INFO**=info **)**
type(*wp*), **INTENT(INOUT) :: A(:, :)**, *rhs*
INTEGER, INTENT(OUT) :: RANK
INTEGER, INTENT(OUT), OPTIONAL :: JPVT(:)
REAL(*wp***), INTENT(IN), OPTIONAL :: RCOND**
INTEGER, INTENT(OUT), OPTIONAL :: INFO
type **::= REAL | COMPLEX**
wp **::= KIND(1.0) | KIND(1.0D0)**
rhs **::= B(:,:) | B(:)**

Arguments

A – (*Input/Output*) **REAL** or **COMPLEX** array, shape $(:,:)$.

- On entry, the matrix A.
- On exit, A has been overwritten by details of its complete orthogonal factorization.

B – (*Input/Output*) **REAL** or **COMPLEX** array, shape $(:,:)$ or $(:)$, and $size(\mathbf{B},1)$ $= max(size(\mathbf{A},1), size(\mathbf{A},2))$ or $size(\mathbf{B}) = max(size(\mathbf{A},1), size(\mathbf{A},2))$, and $size(\mathbf{B},2)$ is the number of vectors b.

- On entry, the matrix B.
- On exit, rows 1 to $size(A,2)$ contain the solution matrix X.
  - If $size(\mathbf{A},1) \geq size(\mathbf{A},2)$ and $\mathbf{RANK} = size(\mathbf{A},2)$, the residual sum-of-squares for the solution in the i^{th} column is given by the sum of squares of elements in rows $size(\mathbf{A},2)+1 : size(\mathbf{A},1)$ in that column.

RANK – (*Output*) **INTEGER**.

- The effective rank of A, i.e., the order of the submatrix R_{11}. This is the same as the order of the submatrix T_{11} in the complete orthogonal factorization of A.

JPVT – *Optional* (*Input/Output*) **INTEGER** array, shape (:), and $size(\mathbf{JPVT}) = size(\mathbf{A}, 2)$

- On entry, if $\mathbf{JPVT}_i \neq 0$, the i^{th} column of A is an initial column, otherwise it is a free column. Before the QR factorization of A, all initial columns are permuted to the leading positions; only the remaining free columns are moved as a result of column pivoting during the factorization.
- On exit, if $\mathbf{JPVT}_i = k$, then the i^{th} column of the matrix product AP was the k^{th} column of A.

RCOND – *Optional* (*Input*) **REAL**.

- **RCOND** is used to determine the effective rank of A, which is defined as the order of the largest leading triangular submatrix R_{11} in the QR factorization of A, with pivoting, whose estimated condition number $< 1/\mathbf{RCOND}$. $\mathbf{RCOND} > 0$.

If **RCOND** is not present, **RCOND** is set to

$$10 \times \max(size(A,1), size(A,2)) \times \mathbf{EPSILON}(1.0\_wp),$$

where *wp* is the working precision.

INFO – *Optional* (*Output*) **INTEGER**.

- $= 0$: successful exit
  < 0: if $\mathbf{INFO} = -i$, the i^{th} argument had an illegal value

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_GELSX_EXAMPLE)

$$A = \begin{pmatrix} 2-i & 6-6i & -2+3i & 2-2i \\ 2 & -3 & 4 & -1 \\ -1-2i & 3-6i & -3+2i & 1-2i \\ -2i & 3-6i & -2+2i & 1-2i \\ -2+3i & 3 & -4+3i & 1 \\ -4+i & -6+6i & -3i & -2+2i \end{pmatrix},$$

$$B = \begin{pmatrix} 0 & 3-i & -1 \\ -3-i & 2i & 5+2i \\ 1+i & 1+3i & 2i \\ 1-2i & 1-3i & 1+2i \\ -2+2i & -2i & 3+3i \\ 3+4i & 2+2i & 1+2i \end{pmatrix}$$

Arrays **A** and **B** on entry:

A

$(2,-1)$	$(6,-6)$	$(-2,3)$	$(2,-2)$
$(2,0)$	$(-3,0)$	$(4,0)$	$(-1,0)$
$(-1,-2)$	$(3,-6)$	$(-3,2)$	$(1,-2)$
$(0,-2)$	$(3,-6)$	$(-2,2)$	$(1,-2)$
$(-2,3i)$	$(3,0)$	$(-4,3)$	$(1,0)$
$(-4,1)$	$(-6,6)$	$(0,-3)$	$(-2,2)$

B

$(0,0)$	$(3,-1)$	$(-1,0)$
$(-3,-1)$	$(0,2)$	$(5,2)$
$(1,1)$	$(1,3)$	$(0,2)$
$(1,-2)$	$(1,-3)$	$(1,2)$
$(-2,2)$	$(0,-2)$	$(3,3)$
$(3,4)$	$(2,2)$	$(1,2)$

The call:

CALL LA_GELSX(A, B, RANK)

B and **RANK** on exit:

B

$(-8.05497_{10^{-2}}, -1.64579_{10^{-1}})$	$(-1.54141_{10^{-1}}, \ 6.26305_{10^{-2}})$
$(\ 3.54906_{10^{-2}}, -8.50731_{10^{-2}})$	$(\ 6.68059_{10^{-2}}, -9.39458_{10^{-3}})$
$(-1.04210_{10^{-1}}, -1.07864_{10^{-1}})$	$(-1.98678_{10^{-1}}, \ 6.88935_{10^{-2}})$
$(\ 1.18302_{10^{-2}}, -2.83577_{10^{-2}})$	$(\ 2.22686_{10^{-2}}, -3.13153_{10^{-3}})$
$(\ 1.27752, \ 8.35313_{10^{-1}})$	$(\ 2.26723, \ -1.59868)$
$(\ 2.86091_{10^{-1}}, \ 2.21345)$	$(\ 1.02186, \ 4.11708)$

B

$(\ 7.58525_{10^{-2}}, -3.43772_{10^{-1}})$
$(-9.18581_{10^{-2}}, -6.99374_{10^{-2}})$
$(\ 1.37091_{10^{-1}}, -2.97147_{10^{-1}})$
$(-3.06193_{10^{-2}}, -2.33125_{10^{-2}})$
$(\ 2.29379, \ 1.97094)$
$(\ 3.98605, \ 2.27148)$

$$\mathbf{RANK} = 2$$

The solution matrix is:

$$X = \left(\begin{array}{rr} -8.05497_{10^{-2}} - 1.64579_{10^{-1}}i & -1.54141_{10^{-1}} + 6.26305_{10^{-2}}i \\ 3.54906_{10^{-2}} - 8.50731_{10^{-2}}i & 6.68059_{10^{-2}} - 9.39458_{10^{-3}}i \\ -1.04210_{10^{-1}} - 1.07864_{10^{-1}}i & -1.98678_{10^{-1}} + 6.88935_{10^{-2}}i \\ 1.18302_{10^{-2}} - 2.83577_{10^{-2}}i & 2.22686_{10^{-2}} - 3.13153_{10^{-3}}i \end{array} \right.$$

$$\left. \begin{array}{r} 7.58525_{10^{-2}} - 3.43772_{10^{-1}}i \\ -9.18581_{10^{-2}} - 6.99374_{10^{-2}}i \\ 1.37091_{10^{-1}} - 2.97147_{10^{-1}}i \\ -3.06193_{10^{-2}} - 2.33125_{10^{-2}}i \end{array} \right).$$

Example 2 (from Program LA_GELSX_EXAMPLE)

A on entry: As in Example 1.
B on entry: $B_{:,1}$, where B is the input matrix in Example 1.
RCOND on entry: **RCOND** = 0.001

The call:

CALL LA_GELSX(A, B(:,1), RANK, JPVT, RCOND = 0.001, &
INFO = INFO)

B(:, **1**), **RANK**, **JPVT**, and **INFO** on exit:

$\mathbf{B}(:,\mathbf{1})$
$(-8.05497_{10^{-2}}, -1.64579_{10^{-1}})$
$(3.54906_{10^{-2}}, -8.50731_{10^{-2}})$
$(-1.04210_{10^{-1}}, -1.07864_{10^{-1}})$
$(1.18302_{10^{-2}}, -2.83577_{10^{-2}})$
$(1.27752, 8.35313_{10^{-1}})$
$(2.86091_{10^{-1}}, 2.21345)$

RANK = 2

JPVT
2
1
4
3

INFO = 0

The solution vector is:

$$x = \begin{pmatrix} -8.05497_{10^{-2}} - 1.64579_{10^{-1}}i \\ 3.54906_{10^{-2}} - 8.50731_{10^{-2}}i \\ -1.04210_{10^{-1}} - 1.07864_{10^{-1}}i \\ 1.18302_{10^{-2}} - 2.83577_{10^{-2}}i \end{pmatrix}.$$

2.1.3 LA_GELSS

Purpose

LA_GELSS computes the minimum norm solution to one or more linear least squares problems

$$\min_x \|b - A\,x\|_2$$

using the singular value decomposition (SVD) of A. A is a real or complex rectangular matrix which may be rank-deficient, and b and x are columns of real or complex rectangular matrices B and X, respectively. The effective rank of A is determined by treating as zero those singular values which are less than **RCOND** times the largest singular value. In addition to X, **LA_GELSS** also returns the right singular vectors and numerical rank of A and, optionally, the singular values of A. (see "Purpose" of **LA_GESVD** for details of the SVD).

Specification

SUBROUTINE LA_GELSS(A, B, RANK, S=s, **&**
RCOND=rcond, **INFO**=info **)**
type(*wp*), **INTENT(INOUT) :: A(:, :),** *rhs*
INTEGER, INTENT(OUT) :: RANK
REAL(*wp***), INTENT(OUT), OPTIONAL :: S(:)**
REAL(*wp***), INTENT(IN), OPTIONAL :: RCOND**
INTEGER, INTENT(OUT), OPTIONAL :: INFO
type **::= REAL | COMPLEX**
wp **::= KIND(1.0) | KIND(1.0D0)**
rhs **::= B(:,:) | B(:)**

Arguments

A – (*Input/Output*) **REAL** or **COMPLEX** array, shape $(:,:)$.

- On entry, the matrix A.
- On exit, the first $\min(size(\mathbf{A},1), size(\mathbf{A},2))$ rows of A are overwritten with its right singular vectors, stored rowwise.

B – (*Input/Output*) **REAL** or **COMPLEX** array, shape $(:,:)$ or $(:)$, and $size(\mathbf{B},1) = max(size(\mathbf{A},1), size(\mathbf{A},2))$ or $size(\mathbf{B}) = max(size(\mathbf{A},1), size(\mathbf{A},2))$, and $size(\mathbf{B},2)$ is the number of columns in matrix B.

- On entry, the matrix B.
- On exit, the solution matrix X.
  - If $size(\mathbf{A},1) \geq size(\mathbf{A},2)$ and $\mathbf{RANK} = size(\mathbf{A},2)$, the residual sum-of-squares for the solution in the i^{th} column is given by the sum of squares of elements in rows $size(\mathbf{A},2)+1 : size(\mathbf{A},1)$ in that column.

RANK – (*Output*) **INTEGER**.

- The effective rank of A, i.e., the number of singular values of A which are greater than the product $\mathbf{RCOND} \times \sigma_1$, where σ_1 is the largest singular value.

S – *Optional* (*Output*) **REAL** array, shape (:), and $size(\mathbf{S}) = \min(size(\mathbf{A},1), size(\mathbf{A},2))$

- The singular values of A in decreasing order.
- The condition number of A in the 2-norm is:

$$\kappa_2(A) = \sigma_1/\sigma_{\min(size(\mathbf{A},1),size(\mathbf{A},2))}$$

.

RCOND – *Optional* (*Input*) **REAL**.

- **RCOND** is used to determine the effective rank of A. Singular values $\sigma_i \leq \mathbf{RCOND} \times \sigma_1$ are treated as zero. $\mathbf{RCOND} > 0$.

If **RCOND** is not present, **RCOND** is set to

$$10 \times \max(size(A,1), size(A,2)) \times \mathbf{EPSILON}(1.0\_wp),$$

where *wp* is the working precision.

INFO – *Optional* (*Output*) **INTEGER**.

- $= 0$: successful exit
  < 0: if $\mathbf{INFO} = -i$, the i^{th} argument had an illegal value
  > 0: the algorithm for computing the SVD failed to converge;
  - if $\mathbf{INFO} = i$, i off-diagonal elements of an intermediate bidiagonal form did not converge to zero.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Example (from Program LA_GELSS_EXAMPLE)

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

$$A = \begin{pmatrix} 0 & 9 & -6 & 3 \\ -3 & 0 & -3 & 0 \\ 1 & 3 & -1 & 1 \\ 1 & 3 & -1 & 1 \\ -2 & 0 & -2 & 0 \\ 3 & 6 & -1 & 2 \end{pmatrix}, \; B = \begin{pmatrix} 4 & -3 & -6 \\ -3 & 5 & 0 \\ 0 & -2 & -3 \\ 0 & 2 & 2 \\ -4 & 1 & 2 \\ 1 & -2 & 6 \end{pmatrix}$$

Arrays **A** and **B** and **RCOND** on entry:

A

0	9	−6	3
−3	0	−3	0
1	3	−1	1
1	3	−1	1
−2	0	−2	0
3	6	−1	2

B

4	−3	−6
−3	5	0
0	−2	−3
0	2	2
−4	1	2
1	−2	6

RCOND = 0.00001

The call:

CALL LA_GELSS(A, B, RANK, S, RCOND=0.00001, &
INFO=INFO)

A, **B**, **S**, **RANK**, and **INFO** on exit:

A

$-1.14937_{10^{-1}}$	$-8.41912_{10^{-1}}$	$4.46337_{10^{-1}}$	$-2.80637_{10^{-1}}$
$-7.55065_{10^{-1}}$	$-2.02940_{10^{-1}}$	$-6.19771_{10^{-1}}$	$-6.76469_{10^{-2}}$
$-5.00000_{10^{-1}}$	$5.00000_{10^{-1}}$	$5.00000_{10^{-1}}$	$-5.00000_{10^{-1}}$
$4.08248_{10^{-1}}$	0.00000	$-4.08248_{10^{-1}}$	$-8.16497_{10^{-1}}$
$-4.08248_{10^{-1}}$	$2.97907_{10^{-1}}$	$-3.53509_{10^{-1}}$	$-1.48151_{10^{-1}}$
$6.12372_{10^{-1}}$	$-1.31850_{10^{-1}}$	$7.06835_{10^{-1}}$	$6.90588_{10^{-1}}$

B

$4.20045_{10^{-1}}$	$-5.10980_{10^{-1}}$	$4.18074_{10^{-1}}$
$3.07432_{10^{-1}}$	$-3.11655_{10^{-1}}$	$-1.77365_{10^{-2}}$
$2.15090_{10^{-1}}$	$-3.03210_{10^{-1}}$	$4.29899_{10^{-1}}$
$1.02477_{10^{-1}}$	$-1.03885_{10^{-1}}$	$-5.91214_{10^{-3}}$
-3.60201	$8.81953_{10^{-3}}$	4.35676
$2.72425_{10^{-1}}$	-2.77266	$3.04325_{10^{-1}}$

S

13.7210
6.14279
$2.64806_{10^{-7}}$
$7.52832_{10^{-15}}$

$$\textbf{RANK} = 2 \qquad \textbf{INFO} = 0$$

The singular values of A are:

$$\begin{pmatrix} 13.7210 & 6.14279 & 2.64806_{10^{-7}} & 7.52832_{10^{-15}} \end{pmatrix}$$

The right singular vectors are (columnwise):

$$\begin{pmatrix} -1.14937_{10^{-1}} & -7.55065_{10^{-1}} & -5.00000_{10^{-1}} & 4.08248_{10^{-1}} \\ -8.41912_{10^{-1}} & -2.02940_{10^{-1}} & 5.00000_{10^{-1}} & 0.00000 \\ 4.46337_{10^{-1}} & -6.19771_{10^{-1}} & 5.00000_{10^{-1}} & -4.082483_{10^{-1}} \\ -2.80637_{10^{-1}} & -6.76469_{10^{-2}} & -5.00000_{10^{-1}} & -8.16497_{10^{-1}} \end{pmatrix}$$

The solution matrix is:

$$X = \begin{pmatrix} 4.20045_{10^{-1}} & -5.10980_{10^{-1}} & 4.18074_{10^{-1}} \\ 3.07432_{10^{-1}} & -3.11655_{10^{-1}} & -1.77365_{10^{-2}} \\ 2.15090_{10^{-1}} & -3.03210_{10^{-1}} & 4.29899_{10^{-1}} \\ 1.02477_{10^{-1}} & -1.03885_{10^{-1}} & -5.91214_{10^{-3}} \end{pmatrix}.$$

2.2 Generalized Linear Least Squares Problems

2.2.1 LA_GGLSE

Purpose

LA_GGLSE solves the linear equality-constrained least squares (LSE) problem:

$$\min \|c - A\,x\|_2 \ \text{ subject to } \ B\,x = d$$

where A and B are real or complex rectangular matrices and c and d are real or complex vectors. Further, A is $m \times n$, B is $p \times n$, c is $m \times 1$ and d is $p \times 1$, where it is assumed that

$$p \le n \le m + p, \quad \text{rank}(B) = p, \text{ and } \ \text{rank}\begin{pmatrix} A \\ B \end{pmatrix} = n.$$

These conditions ensure that the LSE problem has a unique solution. This is obtained using a generalized QR factorization of the matrices B and A.

Specification

SUBROUTINE LA_GGLSE(A, B, C, D, X, INFO=info **)**
type(*wp*)**, INTENT(INOUT) :: A(:, :), B(:,:), C(:), D(:)**
type(*wp*)**, INTENT(OUT) :: X(:)**
INTEGER, INTENT(OUT), OPTIONAL :: INFO
type **::= REAL | COMPLEX**
wp **::= KIND(1.0) | KIND(1.0D0)**

Arguments

A – (*Input/Output*) **REAL** or **COMPLEX** array, shape $(:,:)$, and $size(\mathbf{A},1) = m$, $size(\mathbf{A},2) = n$.

- On entry, the matrix A.
- On exit, the contents of **A** are destroyed.

B – (*Input/Output*) **REAL** or **COMPLEX** array, shape $(:,:)$, and $size(\mathbf{B},1) = p$, $size(\mathbf{B},2) = n$.

- On entry, the matrix B.
- On exit, the contents of **B** are destroyed.

C – (*Input/Output*) **REAL** or **COMPLEX** array, shape $(:)$, and $size(\mathbf{C}) = m$.

- On entry, the vector c.
- On exit, the residual sum of squares for the solution is given by the sum of squares of elements $n - p + 1$ to m.

D – (*Input/Output*) **REAL** or **COMPLEX** array, shape $(:)$, and $size(\mathbf{D}) = p$.

- On entry, The vectors d.
- On exit, the contents of **D** are destroyed.

X – (*Output*) **REAL** or **COMPLEX** array, shape $(:)$, and $size(\mathbf{X}) = n$.

- The solution vector x.

INFO – *Optional* (*Output*) **INTEGER**.

- $= 0$: successful exit
  < 0: if **INFO** $= -i$, the i^{th} argument had an illegal value

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Example (from Program LA_GGLSE_EXAMPLE)

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

$$A = \begin{pmatrix} -6 & 7 & 4 \\ 4 & -7 & 5 \\ 9 & -7 & 4 \\ -3 & 3 & 7 \\ 5 & -1 & 4 \end{pmatrix}, \; B = \begin{pmatrix} 3 & 0 & -3 \\ -2 & 0 & 1 \end{pmatrix}$$

$$c = \begin{pmatrix} 2 \\ -3 \\ 0 \\ 0 \\ 1 \end{pmatrix} \qquad d = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

Arrays $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and $\mathbf{D}$ on entry:

A

-6	7	4
4	-7	5
9	-7	4
-3	3	7
5	-1	4

B

3	0	-3
-2	0	1

C

2
-3
0
0
1

D

2
-1

The call:

CALL LA_GGLSE(A, B, C, D, X, INFO)

$\mathbf{C}, \mathbf{X}$, and **INFO** on exit:

C

$4.90446_{10^{-1}}$
-3.36713
$8.25615_{10^{-2}}$
$-2.79510_{10^{-1}}$
$-8.37885_{10^{-1}}$

X

$3.33333_{10^{-1}}$
$4.90446_{10^{-1}}$
$-3.33333_{10^{-1}}$

INFO $= 0$

The solution vector x is:

$$x = \begin{pmatrix} 3.33333_{10^{-1}} \\ 4.90446_{10^{-1}} \\ -3.33333_{10^{-1}} \end{pmatrix}.$$

The residual sum-of-squares is:

$$\|c - Ax\|_2^2 = 12.12456.$$

2.2.2 LA_GGGLM

Purpose

LA_GGGLM solves a General (Gauss-Markov) Linear Model (GLM) problem:

$$\min_x \|y\|_2 \quad \textsf{subject to} \quad d = A\,x + B\,y$$

where A and B are real or complex rectangular matrices and d is a real or complex vector. Further, A is $n \times m$, B is $n \times p$, and d is $n \times 1$, where it is assumed that

$$m \le n \le m + p, \quad \textsf{rank}(A) = m, \quad \textsf{rank}(A\ B) = n.$$

Under these assumptions, the constrained equation is always consistent, and there is a unique solution x and a minimal 2-norm solution y, which is obtained using a generalized QR factorization of A and B.

In particular, if matrix B is square and nonsingular, then the GLM problem is equivalent to the weighted linear least squares problem

$$\min_x \|B^{-1}\,(d - A\,x)\|_2.$$

Specification

SUBROUTINE LA_GGGLM(A, B, D, X, Y, INFO=info **)**
  type(*wp*)**, INTENT(INOUT) :: A(:, :), B(:,:), D(:)**
  type(*wp*)**, INTENT(OUT) :: X(:), Y(:)**
  INTEGER, INTENT(OUT), OPTIONAL :: INFO
  type **::= REAL | COMPLEX**
  wp **::= KIND(1.0) | KIND(1.0D0)**

Arguments

A – (*Input/Output*) **REAL** or **COMPLEX** array, shape $(:,:)$, and $size(\mathbf{A},1) = n$, $size(\mathbf{A},2) = m$,

- On entry, the matrix A.
- On exit, the contents of **A** are destroyed.

B – (*Input/Output*) **REAL** or **COMPLEX** array, shape (:, :), and $size(\mathbf{B},1) = n$, $size(\mathbf{B},2) = p$,

- The matrix B.
- On exit, the contents of **B** are destroyed.

D – (*Input/Output*) **REAL** or **COMPLEX** array, shape (:), and $size(\mathbf{D}) = n$.

- The vector d.
- On exit, the contents of **D** are destroyed.

X – (*Output*) **REAL** or **COMPLEX** array, shape (:), and $size(\mathbf{X}) = m$.

- The solution vector x.

Y – (*Output*) **REAL** or **COMPLEX** array, shape (:), and $size(\mathbf{Y}) = p$.

- The solution vector y.

INFO – *Optional* (*Output*) **INTEGER**.

- $= 0$: successful exit
  < 0: if **INFO** $= -i$, the i^{th} argument had an illegal value

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Example (from Program LA_GGGLM_EXAMPLE)

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

$$A = \begin{pmatrix} -5 & -3 & 1 & 5 \\ 4 & 2 & -1 & 4 \\ 3 & -5 & 6 & 3 \\ 9 & 0 & -9 & 1 \\ 3 & 0 & 3 & 4 \end{pmatrix}, \qquad B = \begin{pmatrix} 3 & -1 \\ -1 & 1 \\ -1 & 2 \\ -1 & 8 \\ -7 & -1 \end{pmatrix}, \qquad d = \begin{pmatrix} 2 \\ 9 \\ 9 \\ 15 \\ -6 \end{pmatrix}$$

Arrays $\mathbf{A}, \mathbf{B}$, and **D** on entry:

A

−5	−3	1	5
4	2	−1	4
3	−5	6	3
9	0	−9	1
3	0	3	4

B

3	−1
−1	1
−1	2
−1	8
−7	−1

D

2
9
9
15
−6

The call:

CALL LA_GGGLM(A, B, D, X, Y, INFO)

X, **Y**, and **INFO** on exit:

X
1.85042
$8.56464_{10^{-2}}$
$4.19959_{10^{-1}}$
$9.26821_{10^{-1}}$

Y
2.29734
$4.37049_{10^{-1}}$

INFO $= 0$

The solution vectors x and y are:

$$x = \begin{pmatrix} 1.85042 \\ 8.56464_{10^{-2}} \\ 4.19959_{10^{-1}} \\ 9.26821_{10^{-1}} \end{pmatrix}, \quad y = \begin{pmatrix} 2.29734 \\ 4.37049_{10^{-1}} \end{pmatrix}.$$

Chapter 3

Driver Routines for Standard Eigenvalue Problems

3.1 Standard Symmetric Eigenvalue Problems

3.1.1 LA_SYEV / LA_HEEV / LA_SYEVD / LA_HEEVD

Purpose

LA_SYEV computes all eigenvalues and, optionally, all eigenvectors of a real symmetric matrix A.

LA_HEEV computes all eigenvalues and, optionally, all eigenvectors of a complex Hermitian matrix A.

LA_SYEVD computes all eigenvalues and, optionally, all eigenvectors of a real symmetric matrix A. Eigenvectors are computed by a divide and conquer algorithm.

LA_HEEVD computes all eigenvalues and, optionally, all eigenvectors of a complex Hermitian matrix A. Eigenvectors are computed by a divide and conquer algorithm.

Specification

SUBROUTINE LA_SYEV / LA_HEEV / LA_SYEVD /
LA_HEEVD(A, W, JOBZ=jobz, **UPLO**=uplo, **INFO**=info **)**
type(*wp*)**, INTENT(INOUT) :: A(:,:)**
REAL(_wp_**), INTENT(OUT) :: W(:)**
CHARACTER(LEN=1), INTENT(IN), OPTIONAL :: &
JOBZ, UPLO
INTEGER, INTENT(OUT), OPTIONAL :: INFO
where
type **::= REAL | COMPLEX**
wp **::= KIND(1.0) | KIND(1.0D0)**

Arguments

A – (*Input/Output*) **REAL** or **COMPLEX** square array, shape (:, :).

- On entry, the matrix A.
  - If **UPLO** = 'U', the upper triangular part of **A** contains the upper triangular part of the matrix A.
  - If **UPLO** = 'L', the lower triangular part of **A** contains the lower triangular part of the matrix A.
- On exit:
  - If **JOBZ** = 'V', then the columns of **A** contain the orthonormal eigenvectors of the matrix A.

W – (*Output*) **REAL** array, shape (:), $size(W) = size(\mathbf{A}, 1)$.

- If **INFO** = 0, the eigenvalues in ascending order.

JOBZ – *Optional*, (*Input*) **CHARACTER(LEN=1)**.

- = 'N': Compute eigenvalues only;
  = 'V': Compute eigenvalues and eigenvectors.

If **JOBZ** is not present, **JOBZ** = 'N' is assumed.

UPLO – *Optional*, (*Input*) **CHARACTER(LEN=1)**.

- = 'U': Upper triangle of A is stored;
  = 'L': Lower triangle of A is stored.

If **UPLO** is not present, **UPLO** = 'U' is assumed.

INFO – *Optional*, (*Output*) **INTEGER**.

- = 0: successful exit

< 0: if **INFO** $= -i$, the i^{th} argument had an illegal value

> 0: if **INFO** $= i$, i off-diagonal elements of an intermediate tridiagonal form did not converge to zero.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_SYEV_EXAMPLE)

$$A = \begin{pmatrix} -5 & -3 & 1 & 5 & 3 \\ -3 & 2 & -1 & 4 & -1 \\ 1 & -1 & 6 & 3 & -1 \\ 5 & 4 & 3 & 1 & -1 \\ 3 & -1 & -1 & -1 & -7 \end{pmatrix},$$

Array **A** on entry:

A

−5	−3	1	5	3
*	2	−1	4	−1
*	*	6	3	−1
*	*	*	1	−1
*	*	*	*	−7

Elements marked * are not used by the routine.

The call:

CALL LA_SYEV(A, W)

W on exit:

W
−11.2462
−6.24975
1.18524
5.00130
8.30939

The eigenvalues of matrix A are:

$$\begin{pmatrix} -11.2462 \\ -6.24975 \\ 1.18524 \\ 5.00130 \\ 8.30939 \end{pmatrix}.$$

Example 2 (from Program LA_SYEVD_EXAMPLE)

Matrix A as in Example 1.

Array **A** on entry:

A

-5	$*$	$*$	$*$	$*$
-3	2	$*$	$*$	$*$
1	-1	6	$*$	$*$
5	4	3	1	$*$
3	-1	-1	-1	-7

Elements marked * are not used by the routine.

The call:

CALL LA_SYEVD(A, W, 'V', 'L', INFO)

A, **W** and **INFO** on exit:

A

$-6.99428_{10^{-1}}$	$-2.64011_{10^{-1}}$	$6.14343_{10^{-1}}$	$1.08891_{10^{-1}}$	$2.27649_{10^{-1}}$
$-2.43909_{10^{-1}}$	$-3.81203_{10^{-1}}$	$-3.44718_{10^{-1}}$	$-8.12481_{10^{-1}}$	$1.27422_{10^{-1}}$
$-1.45216_{10^{-2}}$	$-1.60115_{10^{-1}}$	$-4.46798_{10^{-1}}$	$3.92579_{10^{-1}}$	$7.87659_{10^{-1}}$
$4.12103_{10^{-1}}$	$3.48500_{10^{-1}}$	$4.84928_{10^{-1}}$	$-4.05806_{10^{-1}}$	$5.55774_{10^{-1}}$
$5.30348_{10^{-1}}$	$-7.98682_{10^{-1}}$	$2.62621_{10^{-1}}$	$9.60214_{10^{-2}}$	$-5.14657_{10^{-2}}$

INFO $= 0$

W: As in Example 1.

The eigenvectors of matrix A are:

$$\begin{pmatrix} -6.99428_{10^{-1}} & -2.64011_{10^{-1}} & 6.14343_{10^{-1}} & 1.08891_{10^{-1}} & 2.27649_{10^{-1}} \\ -2.43909_{10^{-1}} & -3.81203_{10^{-1}} & -3.44718_{10^{-1}} & -8.12481_{10^{-1}} & 1.27422_{10^{-1}} \\ -1.45216_{10^{-2}} & -1.60115_{10^{-1}} & -4.46798_{10^{-1}} & 3.92579_{10^{-1}} & 7.87659_{10^{-1}} \\ 4.12103_{10^{-1}} & 3.48500_{10^{-1}} & 4.84928_{10^{-1}} & -4.05806_{10^{-1}} & 5.55774_{10^{-1}} \\ 5.30348_{10^{-1}} & -7.98682_{10^{-1}} & 2.62621_{10^{-1}} & 9.60214_{10^{-2}} & -5.14657_{10^{-2}} \end{pmatrix}$$

3.1.2 LA_SPEV / LA_HPEV / LA_SPEVD / LA_HPEVD

Purpose

LA_SPEV computes all eigenvalues and, optionally, all eigenvectors of a real symmetric matrix A in packed storage.
LA_HPEV computes all eigenvalues and, optionally, all eigenvectors of a complex Hermitian matrix A in packed storage.
LA_SPEVD computes all eigenvalues and, optionally, all eigenvectors of a real symmetric matrix A in packed storage. Eigenvectors are computed by a divide and conquer algorithm.
LA_HPEVD computes all eigenvalues and, optionally, all eigenvectors of a complex Hermitian matrix A in packed storage. Eigenvectors are computed by a divide and conquer algorithm.

Specification

SUBROUTINE LA_SPEV / LA_HPEV / LA_SPEVD /
LA_HPEVD(AP, W, UPLO=uplo, **Z**=z, **INFO**=info **)**
type(*wp*), **INTENT(INOUT) :: AP(:)**
REAL(*wp***), INTENT(OUT) :: W(:)**
CHARACTER(LEN=1), INTENT(IN), OPTIONAL :: UPLO
type(*wp*), **INTENT(OUT), OPTIONAL :: Z(:,:)**
INTEGER, INTENT(OUT), OPTIONAL :: INFO
type **::= REAL | COMPLEX**
wp **::= KIND(1.0) | KIND(1.0D0)**

Arguments

AP – (*Input/Output*) **REAL** or **COMPLEX** array, shape (:), and $size(\mathbf{AP}) = n(n+1)/2$, where n is the order of A.

- On entry, the upper or lower triangle of matrix A in packed storage.

The elements are stored columnwise in a linear array as follows:

$A_{i,j}$	i,j	**UPLO**
$\mathbf{AP}_{i+(j-1)j/2}$	$1 \leq j \leq n$ $1 \leq i \leq j$	'U'
$\mathbf{AP}_{i+(j-1)(2n-j)/2}$	$1 \leq j \leq n$ $j \leq i \leq n$	'L'

- On exit, **AP** is overwritten by values generated during a reduction of A to tridiagonal matrix T. If **UPLO** = 'U', the diagonal and first superdiagonal of T overwrite the corresponding diagonals of A. If **UPLO** = 'L', the diagonal and first subdiagonal of T overwrite the corresponding diagonals of A.

W – (*Output*) **REAL** array, shape (:), and $size(\mathbf{W}) = n$.

- The eigenvalues in ascending order.

UPLO – *Optional* (*Input*) **CHARACTER(LEN=1)**.

- = 'U': Upper triangle of A is stored;
  = 'L': Lower triangle of A is stored.

If **UPLO** is not present, **UPLO** = 'U' is assumed.

Z – *Optional* (*Output*) **REAL** or **COMPLEX** array, shape (:, :), and $size(\mathbf{Z}, 1) = size(\mathbf{Z}, 2) = n$.

- The columns of **Z** contain the orthonormal eigenvectors of A.

INFO – *Optional* (*Output*) **INTEGER**.

- = 0: successful exit.
  < 0: if **INFO** = $-i$, the i^{th} argument had an illegal value.
  > 0: if **INFO** = i, i off-diagonal elements of an intermediate tridiagonal form did not converge to zero.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

Example 1

The packed storage scheme is illustrated below for the matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

where $a_{ij} = \mathsf{conjg}(a_{ji})$ for $i, j = 1, \cdots, 4$.

If **UPLO** = 'U' then:

AP on entry									
a_{11}	a_{12}	a_{22}	a_{13}	a_{23}	a_{33}	a_{14}	a_{24}	a_{34}	a_{44}

If **UPLO** = 'L' then:

AP on entry									
a_{11}	a_{21}	a_{31}	a_{41}	a_{22}	a_{32}	a_{42}	a_{33}	a_{43}	a_{44}

Example 2 (from Program LA_HPEV_EXAMPLE)

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

$$A = \begin{pmatrix} -1 & 4+i & 2-2i & 2 & -4+i \\ 4-i & 4 & -5-2i & 1-2i & 4+2i \\ 2+2i & -5+2i & -3 & -3-2i & 3-2i \\ 2 & 1+2i & -3+2i & -2 & -4 \\ -4-i & 4-2i & 3+2i & -4 & -1 \end{pmatrix},$$

Array **AP** on entry (**UPLO** = 'U' by default):

AP							
$(-1,0)$	$(4,1)$	$(4,0)$	$(2,-2)$	$(-5,-2)$	$(-3,3)$	$(2,0)$	$(1,-2)$

AP (*continued*)						
$(-3,-2)$	$(-2,0)$	$(-4,1)$	$(4,2)$	$(3,-2)$	$(-4,0)$	$(-1,0)$

The call:

CALL LA_HPEV(A, W)

W on exit:

W
-12.5551
-5.54001
$3.40775_{10^{-1}}$
4.40089
10.3534

The eigenvalues of matrix A are:

$$\begin{pmatrix} -12.5551 \\ -5.54001 \\ 3.40775_{10^{-1}} \\ 4.40089 \\ 10.3534 \end{pmatrix}.$$

Example 3 (from Program LA_HPEVD_EXAMPLE)

Matrix A as in Example 2.

Array **AP** on entry:

A

$(-1,0)$	$(4,-1)$	$(2,2)$	$(2,0)$	$(-4,-1)$	$(4,0)$	$(-5,2)$	$(1,2)$

A continued

$(4,-2)$	$(-3,0)$	$(-3,2)$	$(3,2)$	$(-2,0)$	$(-4,0)$	$(-1,0)$

The call:

CALL LA_HPEVD(A, W, 'L', Z, INFO)

W, **Z**, and **INFO** on exit:

W: As in Example 2.

Z

$(\ \ 3.98486_{10^{-1}}, +0.00000)$	$(\ \ 4.02021_{10^{-1}}, -0.00000)$
$(-3.70709_{10^{-1}}, -1.91586_{10^{-1}})$	$(-1.25732_{10^{-1}}, +6.12544_{10^{-2}})$
$(-5.87136_{10^{-1}}, -2.48810_{10^{-2}})$	$(\ \ 1.14956_{10^{-3}}, -3.18444_{10^{-1}})$
$(-9.05485_{10^{-2}}, +2.84778_{10^{-1}})$	$(-9.06732_{10^{-2}}, -6.94968_{10^{-1}})$
$(\ \ 4.16215_{10^{-1}}, +2.43307_{10^{-1}})$	$(\ \ 2.17064_{10^{-1}}, -4.23193_{10^{-1}})$

Z continued

$(\ \ 7.03213_{10^{-1}}, +0.00000)$	$(-1.81322_{10^{-1}}, +0.00000)$
$(\ \ 1.22276_{10^{-2}}, +2.63003_{10^{-1}})$	$(-9.39151_{10^{-2}}, -4.68567_{10^{-1}})$
$(\ \ 5.35173_{10^{-1}}, +1.99132_{10^{-2}})$	$(\ \ 3.85155_{10^{-1}}, -1.12678_{10^{-1}})$
$(-1.30605_{10^{-1}}, +3.17015_{10^{-1}})$	$(-3.01614_{10^{-1}}, +3.62524_{10^{-1}})$
$(-1.11745_{10^{-1}}, +1.39001_{10^{-1}})$	$(\ \ 3.70267_{10^{-1}}, -4.67133_{10^{-1}})$

Z continued
$(\ \ 3.90130_{10^{-1}}, +0.00000)$
$(\ \ 4.42522_{10^{-1}}, -5.59275_{10^{-1}})$
$(-1.87118_{10^{-1}}, +2.65300_{10^{-1}})$
$(\ \ 2.81158_{10^{-1}}, +2.23413_{10^{-2}})$
$(-2.75297_{10^{-1}}, -2.80089_{10^{-1}})$

INFO = 0

The eigenvectors of A are:

$$\left(\begin{array}{ll} 3.98486_{10^{-1}} & 4.02021_{10^{-1}} \\ -3.70709_{10^{-1}} - 1.91586_{10^{-1}}i & -1.25732_{10^{-1}} + 6.12544_{10^{-2}}i \\ -5.87136_{10^{-1}} - 2.48810_{10^{-2}}i & 1.14956_{10^{-3}} - 3.18444_{10^{-1}}i \\ -9.05485_{10^{-2}} + 2.84778_{10^{-1}}i & -9.06732_{10^{-2}} - 6.94968_{10^{-1}}i \\ 4.16215_{10^{-1}} + 2.43307_{10^{-1}}i & 2.17064_{10^{-1}} - 4.23193_{10^{-1}}i \end{array}\right.$$

$$\begin{array}{ll} 7.03213_{10^{-1}} & 1.81322_{10^{-1}} \\ 1.22276_{10^{-2}} + 2.63003_{10^{-1}}i & -9.39151_{10^{-2}} - 4.68567_{10^{-1}}i \\ 5.35173_{10^{-1}} + 1.99132_{10^{-2}}i & 3.85155_{10^{-1}} - 1.12678_{10^{-1}}i \\ -1.30605_{10^{-1}} + 3.17015_{10^{-1}}i & -3.01614_{10^{-1}} + 3.62524_{10^{-1}}i \\ -1.11745_{10^{-1}} + 1.39001_{10^{-1}}i & 3.70267_{10^{-1}} - 4.67133_{10^{-1}}i \end{array}$$

$$\left.\begin{array}{l} 3.90130_{10^{-1}} \\ 4.42522_{10^{-1}} - 5.59275_{10^{-1}}i \\ -1.87118_{10^{-1}} + 2.65300_{10^{-1}}i \\ 2.81158_{10^{-1}} + 2.23413_{10^{-2}}i \\ -2.75297_{10^{-1}} - 2.80089_{10^{-1}}i \end{array}\right)$$

3.1.3 LA_SBEV / LA_HBEV / LA_SBEVD / LA_HBEVD

Purpose

LA_SBEV computes all eigenvalues and, optionally, all eigenvectors of a real symmetric matrix A in band form.
LA_HBEV computes all eigenvalues and, optionally, all eigenvectors of a complex Hermitian matrix A in band form.
LA_SBEVD computes all eigenvalues and, optionally, all eigenvectors of a real symmetric matrix A in band form. Eigenvectors are computed by a divide and conquer algorithm.
LA_HBEVD computes all eigenvalues and, optionally, all eigenvectors of a complex Hermitian matrix A in band form. Eigenvectors are computed by a divide and conquer algorithm.

Specification

SUBROUTINE LA_SBEV / LA_HBEV / LA_SBEVD /
LA_HBEVD(AB, W, UPLO=uplo, **Z**=z, **INFO**=info **)**
type(*wp*)**, INTENT(INOUT) :: AB(:,:)**
REAL(*wp***), INTENT(OUT) :: W(:)**
CHARACTER(LEN=1), INTENT(IN), OPTIONAL :: UPLO
type(*wp*)**, INTENT(OUT), OPTIONAL :: Z(:,:)**
INTEGER, INTENT(OUT), OPTIONAL :: INFO
where
type **::= REAL | COMPLEX**
wp **::= KIND(1.0) | KIND(1.0D0)**

Arguments

AB – (*Input/Output*) **REAL / COMPLEX** array, shape $(:,:)$, and $size(\mathbf{AB},1) = kd+1$, $size(\mathbf{AB},2) = n$, where kd is the number of subdiagonals or superdiagonals in the band and n is the order of A.

- On entry, the upper (if **UPLO** = 'U') or lower (if **UPLO** = 'L') triangle of matrix A in band storage. The $kd+1$ diagonals of A are stored rowwise in **AB** as follows:

$A_{i,j}$	i,j	**UPLO**
$\mathbf{AB}_{kd+1+i-j,j}$	$\max(1, j-kd) \le i \le j$ $1 \le j \le n$	'U'
$\mathbf{AB}_{1+i-j,j}$	$j \le i \le \min(n, j+kd)$ $1 \le j \le n$	'L'

- On exit, **AB** is overwritten by values generated during a reduction of A to a tridiagonal matrix T.
  - If **UPLO** = 'U' the first superdiagonal and the diagonal of T are returned in rows kd and $kd+1$ of **AB**.
  - If **UPLO** = 'L', the diagonal and first subdiagonal of T are returned in the first two rows of **AB**.

W – (*Output*) **REAL** array, shape $(:)$, and $size(\mathbf{W}) = n$.

- If **INFO** = 0, the eigenvalues in ascending order.

UPLO – *Optional* (*Input*) **CHARACTER(LEN=1)**.

- = 'U': Upper triangle of A is stored;
  = 'L': Lower triangle of A is stored.

If **UPLO** is not present, **UPLO** = 'U' is assumed.

Z – *Optional* (*Output*) **REAL** or **COMPLEX** array, shape $(:,:)$, and $size(\mathbf{Z},1) = size(\mathbf{Z},2) = n$.

- The columns of **Z** contain the orthonormal eigenvectors of A.

INFO – *Optional*, (*Output*) **INTEGER**.

- $= 0$: successful exit.
  < 0: if **INFO** $= -i$, the i^{th} argument had an illegal value.
  > 0: if **INFO** $= i$, i off-diagonal elements of an intermediate tridiagonal form did not converge to zero.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

Example 1

The band storage scheme is illustrated by the following example, where $n = 9$ and $kd = 2$, making $size(\mathbf{AB},1) = 3$ and $size(\mathbf{AB},2) = 9$:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & & & & & & \\ a_{21} & a_{22} & a_{23} & a_{24} & & & & & \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & & & & \\ & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & & & \\ & & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & & \\ & & & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & \\ & & & & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\ & & & & & a_{86} & a_{87} & a_{88} & a_{89} \\ & & & & & & a_{97} & a_{98} & a_{99} \end{pmatrix}$$

where $a_{ij} = \mathsf{conj}(a_{ji})$ for $i,j = 1,\cdots,9$.

Banded storage of A when **UPLO** = 'U':

AB on entry

$*$	$*$	a_{13}	a_{24}	a_{35}	a_{46}	a_{57}	a_{68}	a_{79}
$*$	a_{12}	a_{23}	a_{34}	a_{45}	a_{56}	a_{67}	a_{78}	a_{89}
a_{11}	a_{22}	a_{33}	a_{44}	a_{55}	a_{66}	a_{77}	a_{88}	a_{99}

Banded storage of A when **UPLO** = 'L':

AB on entry

a_{11}	a_{22}	a_{33}	a_{44}	a_{55}	a_{66}	a_{77}	a_{88}	a_{99}
a_{21}	a_{32}	a_{43}	a_{54}	a_{65}	a_{76}	a_{87}	a_{98}	$*$
a_{31}	a_{42}	a_{53}	a_{64}	a_{75}	a_{86}	a_{97}	$*$	$*$

Array elements marked $*$ are not used by the routine.

Example 2 (from Program LA_SBEV_EXAMPLE)

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

$$A = \begin{pmatrix} -6 & -2 & -5 & 0 & 0 \\ -2 & 2 & 6 & 4 & 0 \\ -5 & 6 & -1 & 10 & -1 \\ 0 & 4 & 10 & 3 & 6 \\ 0 & 0 & -1 & 6 & -5 \end{pmatrix},$$

Array **AB** on entry (**UPLO** = 'U' by default):

AB

0	0	−5	4	−1
0	−2	6	10	6
−6	2	−1	3	−5

The call:

CALL LA_SBEV(A, W)

W on exit:

W
−13.6526
−6.98025
−3.71265
1.20058
16.1449

The eigenvalues of matrix A are:

$$\begin{pmatrix} -13.6526 \\ -6.98025 \\ -3.71265 \\ 1.20058 \\ 16.1449 \end{pmatrix}.$$

Example 3 (from Program LA_SBEVD_EXAMPLE)

Matrix A as in Example 2.

Array **AB** on entry:

A

−6	2	−1	3	−5
−2	6	10	6	0
−5	4	−1	0	0

The call:

CALL LA_SBEVD(A, W, 'L', Z, INFO)

W, **Z** and **INFO** on exit:

W: As in Example 2.

Z

$-3.93202_{10^{-1}}$	$7.24993_{10^{-1}}$	$4.59408_{10^{-1}}$	$2.82426_{10^{-1}}$	$-1.70171_{10^{-1}}$
$5.72489_{10^{-2}}$	$-5.11449_{10^{-2}}$	$6.61920_{10^{-1}}$	$-5.94760_{10^{-1}}$	$4.49699_{10^{-1}}$
$-6.24702_{10^{-1}}$	$1.62592_{10^{-1}}$	$-4.74933_{10^{-1}}$	$-1.68822_{10^{-1}}$	$5.73803_{10^{-1}}$
$5.16428_{10^{-1}}$	$2.33432_{10^{-1}}$	$-3.22674_{10^{-3}}$	$5.13312_{10^{-1}}$	$6.44447_{10^{-1}}$
$-4.30307_{10^{-1}}$	$-6.25175_{10^{-1}}$	$3.53885_{10^{-1}}$	$5.23933_{10^{-1}}$	$1.55729_{10^{-1}}$

INFO $= 0$

The eigenvectors of A are:

$$\begin{pmatrix} -3.93202_{10^{-1}} & 7.24993_{10^{-1}} & 4.59408_{10^{-1}} & 2.82426_{10^{-1}} & -1.70171_{10^{-1}} \\ 5.72489_{10^{-2}} & -5.11449_{10^{-2}} & 6.61920_{10^{-1}} & -5.94760_{10^{-1}} & 4.49699_{10^{-1}} \\ -6.24702_{10^{-1}} & 1.62592_{10^{-1}} & -4.74933_{10^{-1}} & -1.68822_{10^{-1}} & 5.73803_{10^{-1}} \\ 5.16428_{10^{-1}} & 2.33432_{10^{-1}} & -3.22674_{10^{-3}} & 5.13312_{10^{-1}} & 6.44447_{10^{-1}} \\ -4.30307_{10^{-1}} & -6.25175_{10^{-1}} & 3.53885_{10^{-1}} & 5.23933_{10^{-1}} & 1.55729_{10^{-1}} \end{pmatrix}$$

3.1.4 LA_STEV / LA_STEVD

Purpose

LA_STEV computes all eigenvalues and, optionally, all eigenvectors of a real symmetric tridiagonal matrix A.

LA_STEVD computes all eigenvalues and, optionally, all eigenvectors of a real symmetric tridiagonal matrix A. Eigenvectors are computed by a divide and conquer algorithm.

Specification

SUBROUTINE LA_STEV / LA_STEVD(D, E, Z=z, **INFO**=info)
 REAL(*wp*), INTENT(INOUT) :: D(:), E(:)
 REAL(*wp*), INTENT(OUT), OPTIONAL :: Z(:,:)
 INTEGER, INTENT(OUT), OPTIONAL :: INFO
 where
 wp **::= KIND(1.0)** | **KIND(1.0D0)**

Arguments

D – (*Input/Output*) **REAL** array shape (:). $size(\mathbf{D}) = n$, where n is the order of A.

- On entry, the diagonal elements of the matrix A.
- On exit, the eigenvalues in ascending order.

E – (*Input/Output*) **REAL** array, shape (:), and $size(\mathbf{E}) = n$.

- On entry, the $n-1$ subdiagonal elements of A, stored in elements 1 to $n-1$ of **E**; $\mathbf{E}_n$ need not be set, but is used by the routine.
- On exit, the contents of **D** are destroyed.

Z – *Optional*, (*Output*) **REAL** array, shape(:,:), and $size(Z,1) = size(Z,2) = n$.

- The columns of **Z** contain the orthonormal eigenvectors of the matrix A.

INFO – *Optional*, (*Output*) **INTEGER**.

- $= 0$: successful exit
  < 0: if **INFO** $= -i$, the i^{th} argument had an illegal value
  > 0: if **INFO** $= i$, i off-diagonal elements of **E** did not converge to zero.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_STEV_EXAMPLE)

$$A = \begin{pmatrix} 0 & 9 & 0 & 0 & 0 \\ 9 & 5 & 2 & 0 & 0 \\ 0 & 2 & 4 & 3 & 0 \\ 0 & 0 & 3 & 7 & -11 \\ 0 & 0 & 0 & -11 & 9 \end{pmatrix},$$

Arrays **D** and **E** on entry:

D

0	5	4	7	9

E

9	2	3	−11	*

The call:

CALL LA_STEV(D, E)

D on exit:

D
−6.99558
−3.67499
4.20213
12.1448
19.3236

The eigenvalues of matrix A are:

$$\begin{pmatrix} -6.99558 \\ -3.67499 \\ 4.20213 \\ 12.1448 \\ 19.3236 \end{pmatrix}.$$

Example 2 (from Program LA_STEVD_EXAMPLE)

Matrix A as in Example 1.

The call:

CALL LA_STEVD(D, E, Z, INFO)

D, **Z**, and **INFO** on exit:

D: As in Example 1.

Z

$7.81147_{10^{-1}}$	$9.95755_{10^{-2}}$	$1.97493_{10^{-1}}$	$-5.83724_{10^{-1}}$	$-1.25199_{10^{-2}}$
$-6.07175_{10^{-1}}$	$-4.06599_{10^{-2}}$	$9.22101_{10^{-2}}$	$-7.87693_{10^{-1}}$	$-2.68811_{10^{-2}}$
$1.26546_{10^{-1}}$	$-2.71728_{10^{-1}}$	$-9.25504_{10^{-1}}$	$-1.87215_{10^{-1}}$	$-1.36177_{10^{-1}}$
$-5.90327_{10^{-2}}$	$7.22276_{10^{-1}}$	$-1.23832_{10^{-1}}$	$1.68506_{10^{-2}}$	$-6.77654_{10^{-1}}$
$-4.05962_{10^{-2}}$	$6.26828_{10^{-1}}$	$-2.83907_{10^{-1}}$	$-5.89398_{10^{-2}}$	$7.22054_{10^{-1}}$

INFO $= 0$

The eigenvectors of A are:

$$\begin{pmatrix} 7.81147_{10^{-1}} & 9.95755_{10^{-2}} & 1.97493_{10^{-1}} & -5.83724_{10^{-1}} & -1.25199_{10^{-2}} \\ -6.07175_{10^{-1}} & -4.06599_{10^{-2}} & 9.22101_{10^{-2}} & -7.87693_{10^{-1}} & -2.68811_{10^{-2}} \\ 1.26546_{10^{-1}} & -2.71728_{10^{-1}} & -9.25504_{10^{-1}} & -1.87215_{10^{-1}} & -1.36177_{10^{-1}} \\ -5.90327_{10^{-2}} & 7.22276_{10^{-1}} & -1.23832_{10^{-1}} & 1.68506_{10^{-2}} & -6.77654_{10^{-1}} \\ -4.05962_{10^{-2}} & 6.26828_{10^{-1}} & -2.83907_{10^{-1}} & -5.89398_{10^{-2}} & 7.22054_{10^{-1}} \end{pmatrix}$$

3.2 Standard Nonsymmetric Eigenvalue Problems

3.2.1 LA_GEES

Purpose

LA_GEES computes for a real/complex square matrix A, the eigenvalues, the real-Schur/complex-Schur form T, and, optionally, the matrix of Schur vectors Z, where Z is orthogonal/unitary. This gives the Schur factorization

$$A = Z\,T\,Z^H.$$

Optionally, it also orders the eigenvalues on the diagonal of the Schur form so that selected eigenvalues are at the top left. The leading columns of Z then form an orthonormal basis for the invariant subspace corresponding to the selected eigenvalues.

A real matrix is in real-Schur form if it is block upper triangular with 1×1 and 2×2 blocks along the main diagonal. 2×2 blocks are standardized in the form

$$\begin{bmatrix} a & b \\ c & a \end{bmatrix}$$

where $b\,c < 0$. The eigenvalues of such a block are $a \pm \sqrt{bc}$.

A complex matrix is in complex-Schur form if it is upper triangular.

Specification

SUBROUTINE LA_GEES(A, *w*, **VS**=vs, **SELECT**=select, **&**
SDIM=sdim, **INFO**=info **)**
type(*wp*), **INTENT(INOUT) :: A(:,:)**
type(*wp*), **INTENT(OUT) ::** *w(:)*
type(*wp*), **INTENT(OUT), OPTIONAL :: VS(:,:)**
INTERFACE
LOGICAL FUNCTION SELECT(*w***)**
type(*wp*), **INTENT(IN) ::** *w*
END FUNCTION SELECT
END INTERFACE
OPTIONAL :: SELECT
INTEGER, INTENT(OUT), OPTIONAL :: SDIM, INFO
where
type **::= REAL | COMPLEX**
wp **::= KIND(1.0) | KIND(1.0D0)**
w **::= WR, WI | W**
w(:) **::= WR(:), WI(:) | W(:)**

Arguments

A – (*Input/Output*) **REAL** or **COMPLEX** square array, shape $(:,:)$.

- On entry, the matrix A.
- On exit, **A** has been overwritten by its Schur form T.

w – (*Output*) **REAL** or **COMPLEX** array, shape $(:)$, and $size(w) = size(\mathbf{A}, 1)$.

- The computed eigenvalues in the order that they appear on the diagonal of the Schur form T.
- $w(:)$ ::= **WR(:), WI(:) | W(:)**,
  where
  WR(:), WI(:) are of **REAL** *type* (for the real and imaginary parts)
  and **W(:)** is of **COMPLEX** *type*.

VS – *Optional* (*Output*) **REAL** or **COMPLEX** array, shape $(:,:)$, and $size(\mathbf{VS}, 1) = size(\mathbf{VS}, 2) = size(\mathbf{A}, 1)$.

- **VS** contains the orthogonal/unitary matrix Z of Schur vectors.

SELECT – *Optional* (*Input*) **LOGICAL FUNCTION**

LOGICAL FUNCTION SELECT(*w* **)**
type(*wp*), **INTENT(IN) ::** *w(:)*
where
type **::= REAL | COMPLEX**

wp ::= **KIND(1.0)** | **KIND(1.0D0)**
w ::= **WR, WI** | **W**

1. **SELECT** must be declared as **EXTERNAL** or as an explicit interface in the calling (sub)program.
2. **SELECT** is called by **LA_GEES** for every computed eigenvalue w_j (but only once for a complex conjugate pair). It is used to select eigenvalues to order to the top left of the Schur form. The eigenvalue w_j $(1 \leq j \leq size(\mathbf{A}, 1))$ is selected if **SELECT**(w_j) has a value **.TRUE.**.
3. Note that a selected complex eigenvalue may no longer satisfy **SELECT**$(w_j) =$ **.TRUE.** after ordering, since ordering may change the value of complex eigenvalues (especially if the eigenvalue is ill-conditioned); in this case **INFO** is set to $size(\mathbf{A}, 1) + 2$ (see **INFO** below).
4. The absence of **SELECT** is equvalent to **SELECT**$(w_j) =$ **.FALSE.**, for $1 \leq j \leq size(\mathbf{A}, 1)$.

SDIM – *Optional* (*Output*) **INTEGER**.

- **SDIM** is a number of eigenvalues for which **SELECT** is true.

INFO – *Optional*, (*Output*) **INTEGER**.

- $= 0$: successful exit
  < 0: if **INFO** $= -i$, the i^{th} argument had an illegal value
  > 0: if **INFO** $= i$, and i is
  $\leq size(\mathbf{A}, 1)$: the QR algorithm failed to compute all the eigenvalues; elements $1 : ilo - 1$ and $i + 1 : size(\mathbf{A}, 1)$ of w contain those eigenvalues which have converged; **VS** contains the matrix which reduces A to its partially converged Schur form.
  $= size(\mathbf{A}, 1) + 1$: the eigenvalues could not be reordered because some eigenvalues were too close to separate (the problem is very ill-conditioned);
  $= size(\mathbf{A}, 1) + 2$: after reordering, roundoff changed values of some complex eigenvalues so that leading eigenvalues in the Schur form no longer satisfy **SELECT** = **.TRUE.**. This could also be caused by underflow due to scaling.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_GEES_EXAMPLE)

$$A = \begin{pmatrix} -2 & -8 & 1 & 2 & -5 \\ 6 & 6 & 0 & 2 & 1 \\ 0 & 4 & -2 & 11 & 0 \\ 1 & 6 & 1 & 0 & -4 \\ 2 & -6 & 4 & 9 & -3 \end{pmatrix}$$

Array **A** on entry:

A

−2	−8	1	2	−5
6	6	0	2	1
0	4	−2	11	0
1	6	1	0	−4
2	−6	4	9	−3

The call:

CALL LA_GEES(A, WR, WI)

A, **WR** and **WI** on exit:

A

−2.21691	5.49380	−1.14455	−2.80402	−1.54849
−13.4518	−2.21691	−4.98520	3.07957	−3.07004
0.00000	0.00000	4.45961	11.8468	4.30683
0.00000	0.00000	−1.21940	4.45961	−3.07536
0.00000	0.00000	0.00000	0.00000	−5.48541

WR	WI
−2.21691	8.59661
−2.21691	−8.59661
4.45961	3.80078
4.45961	−3.80078
−5.48541	0.00000

The real-Schur form T of matrix A is:

$$T = \begin{pmatrix} -2.21691 & 5.49380 & -1.14455 & -2.80402 & -1.54849 \\ -13.4518 & -2.21691 & -4.98520 & 3.07957 & -3.07004 \\ 0.00000 & 0.00000 & 4.45961 & 11.8468 & 4.30682 \\ 0.00000 & 0.00000 & -1.21940 & 4.45961 & -3.07536 \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 & -5.48541 \end{pmatrix}.$$

The eigenvalues of matrix A are:

$$\begin{pmatrix} -2.21691 + 8.59661i \\ -2.21691 - 8.59661i \\ 4.45961 + 3.80078i \\ 4.45961 - 3.80078i \\ -5.48541 \end{pmatrix}.$$

Example 2 (from Program LA_GEES_EXAMPLE)

Matrix A as in Example 1.

Function **SELECT** is:

```
LOGICAL FUNCTION SELECT(X,Y)
   USE LA_PRECISION, ONLY: WP => SP
   INTRINSIC :: EPSILON
   REAL(WP), INTENT(IN) :: X,Y
!  Select the real eigenvalues
!  within the working precision
   IF (ABS(Y) <= EPSILON(1.0_WP)) THEN
      SELECT = .TRUE.
   ELSE
      SELECT = .FALSE.
   ENDIF
END FUNCTION SELECT
```

The call:

CALL LA_GEES(A, WR, WI, VS, SELECT, SDIM, INFO)

A, **WR**, **WI**, **VS**, **SDIM**, and **INFO** on exit:

A

−5.48541	−1.02666	−1.53935	3.61397	7.99755
0.00000	−2.21691	13.3890	4.35515	−4.25895
0.00000	−5.51959	−2.21691	1.90098	1.50834
0.00000	0.00000	0.00000	4.45961	−1.43694
0.00000	0.00000	0.00000	10.0533	4.45961

WR
-5.48541
-2.21691
-2.21691
4.45961
4.45961

WI
0.00000
8.59661
-8.59661
3.80078
-3.80078

VS

$3.27118_{10^{-2}}$	$6.30875_{10^{-1}}$	$3.60917_{10^{-1}}$	$-5.46254_{10^{-1}}$	$4.15058_{10^{-1}}$
$-8.25041_{10^{-2}}$	$-1.65375_{10^{-1}}$	$-5.77973_{10^{-1}}$	$-7.58461_{10^{-1}}$	$-2.37756_{10^{-1}}$
$-9.35865_{10^{-1}}$	$1.32935_{10^{-1}}$	$1.85062_{10^{-1}}$	$1.58972_{10^{-2}}$	$-2.68297_{10^{-1}}$
$3.26536_{10^{-1}}$	$5.20012_{10^{-2}}$	$4.99294_{10^{-1}}$	$-1.82926_{10^{-1}}$	$-7.79689_{10^{-1}}$
$9.82508_{10^{-2}}$	$7.44495_{10^{-1}}$	$-5.02140_{10^{-1}}$	$3.04349_{10^{-1}}$	$-3.02161_{10^{-1}}$

$$\mathbf{SDIM} = 1, \qquad \mathbf{INFO} = 0$$

The real-Schur form T, the matrix of Schur vectors Z, and the eigenvalues of matrix A are:

$$T = \begin{pmatrix} -5.48541 & -1.02666 & -1.53935 & 3.61397 & 7.99755 \\ 0.00000 & -2.21691 & 13.3890 & 4.35515 & -4.25895 \\ 0.00000 & -5.51959 & -2.21691 & 1.90098 & 1.50834 \\ 0.00000 & 0.00000 & 0.00000 & 4.45961 & -1.43694 \\ 0.00000 & 0.00000 & 0.00000 & 10.05331 & 4.45961 \end{pmatrix},$$

$$Z = \begin{pmatrix} 3.27118_{10^{-2}} & 6.30875_{10^{-1}} & 3.60917_{10^{-1}} & -5.46254_{10^{-1}} & 4.15057_{10^{-1}} \\ -8.25041_{10^{-2}} & -1.65375_{10^{-1}} & -5.77973_{10^{-1}} & -7.58461_{10^{-1}} & -2.37756_{10^{-1}} \\ -9.35865_{10^{-1}} & 1.32935_{10^{-1}} & 1.85062_{10^{-1}} & 1.58972_{10^{-2}} & -2.68297_{10^{-1}} \\ 3.26536_{10^{-1}} & 5.20012_{10^{-1}} & 4.99294_{10^{-1}} & -1.82926_{10^{-1}} & -7.79689_{10^{-1}} \\ 9.82508_{10^{-2}} & 7.44495_{10^{-1}} & -5.02140_{10^{-1}} & 3.04349_{10^{-1}} & -3.02161_{10^{-1}} \end{pmatrix},$$

and

$$\begin{pmatrix} -5.48541 \\ -2.21691 + 8.59661i \\ -2.21691 - 8.59661i \\ 4.45961 + 3.80078i \\ 4.45961 - 3.80078i \end{pmatrix}.$$

3.2.2 LA_GEEV

Purpose

LA_GEEV computes for a real or complex square matrix A, the eigenvalues and, optionally, the left and/or right eigenvectors.

The right eigenvector v_j of A satisfies

$$A\,v_j = \lambda_j\,v_j$$

where λ_j is its eigenvalue. The left eigenvector u_j of A satisfies

$$u_j^H\,A = \lambda_j\,u_j^H$$

where u_j^H denotes the conjugate-transpose of u_j. The computed eigenvectors are normalized to have Euclidean norm equal to 1 and largest component real.

Specification

SUBROUTINE LA_GEEV(A, *w*, **VL**=vl, **VR**=vr, **INFO**=info **)**
type(*wp*), **INTENT(INOUT) :: A(:,:)**
type(*wp*), **INTENT(OUT) ::** *w(:)*
type(*wp*), **INTENT(OUT), OPTIONAL :: VL(:,:), VR(:,:)**
INTEGER, INTENT(OUT), OPTIONAL :: INFO
where
type **::= REAL | COMPLEX**
wp **::= KIND(1.0) | KIND(1.0D0)**
w **::= WR, WI | W**
w(:) **::= WR(:), WI(:) | W(:)**

Arguments

A – (*Input/Output*) **REAL** or **COMPLEX** square array, shape $(:,:)$.

- On entry, the matrix A.

w – (*Output*) **REAL** or **COMPLEX** array, shape $(:)$, and $size(w) = size(\mathbf{A},1)$.

- The computed eigenvalues. A complex-conjugate pair appear consecutively, with the eigenvalue having the positive imaginary part first.
- $w(:)$ ::= **WR(:), WI(:) | W(:)**,
  where
  WR(:), WI(:) are of **REAL** *type* (for the real and imaginary parts) and **W(:)** is of **COMPLEX** *type*.

VL – *Optional* (*Output*) **REAL** or **COMPLEX** square array, shape $(:,:)$, $size(\mathbf{VL},1) = size(\mathbf{A},1)$. The left eigenvectors u_j are stored in the columns of **VL**.

VR – *Optional* (*Output*) **REAL** or **COMPLEX** square array, shape $(:,:)$, $size(\mathbf{VL},1) = size(\mathbf{A},1)$. The right eigenvectors v_j are stored in the columns of **VR**.

INFO – *Optional*, (*Output*) **INTEGER**.

- $= 0$: successful exit
  < 0: if **INFO** $= -i$, the i^{th} argument had an illegal value
  > 0: if **INFO** $= i$, the QR algorithm failed to compute all the eigenvalues, and no eigenvectors have been computed; elements $i+1:n$ of w contain eigenvalues which have converged.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_GEEV_EXAMPLE)

$$A = \begin{pmatrix} -2 & -5 & 1 & -1 & -3 \\ 5 & 0 & -4 & -1 & 9 \\ 8 & -1 & 1 & 6 & 0 \\ 2 & 0 & 3 & 1 & -2 \\ -4 & -3 & 4 & -4 & -2 \end{pmatrix}$$

Array **A** on entry:

A

−2	−5	1	−1	−3
5	0	−4	−1	9
8	−1	1	6	0
2	0	3	1	−2
−4	−3	4	−4	−2

The call:

CALL LA_GEEV(A, WR, WI)

WR and **WI** on exit:

WR	WI
-3.74980	5.88964
-3.74980	-5.88964
5.79150	0.00000
-2.34478	0.00000
2.05287	0.00000

The eigenvalues of matrix A are:

$$\begin{pmatrix} -3.74980 + 5.88964i \\ -3.74980 - 5.88964i \\ 5.79150 \\ -2.34478 \\ 2.05287 \end{pmatrix}.$$

Example 2 (from Program LA_GEEV_EXAMPLE)

$$A = \begin{pmatrix} -6 & 5 & 8+i & -1+2i & -4+3i \\ -6 & 2 & 6-i & -4 & -1-3i \\ 5+i & -4+4i & -3+i & 3-4i & 4 \\ 1-i & 7-i & -6-i & 8+i & -4-i \\ -3 & 2-2i & 2i & -2+4i & -6-2i \end{pmatrix}$$

A

$(-6,0)$	$(5,0)$	$(8,1)$	$(-1,2)$	$(-4,3)$
$(-6,0)$	$(2,0)$	$(6,-1)$	$(-4,0)$	$(-1,-3)$
$(5,1)$	$(-4,4)$	$(-3,1)$	$(3,-4)$	$(4,0)$
$(1,-1)$	$(7,-1)$	$(-6,-1)$	$(8,1)$	$(-4,-1)$
$(-3,0)$	$(2,-2)$	$(0,2)$	$(-2,4)$	$(-6,-2)$

The call:

CALL LA_GEEV(A, W, VL, VR, INFO)

W, **VL**, **VR**, and **INFO** on exit:

W

$(-11.3885, -4.99204)$
$(-5.08344_{10^{-1}}, -3.70385)$
$(5.39333, -1.70267)$
$(2.47931, 7.98679)$
$(-9.75794_{10^{-1}}, 2.41177)$

VL

$(-4.42699_{10^{-1}}, -1.09152_{10^{-1}})$	$(-2.90780_{10^{-1}}, 4.13348_{10^{-1}})$
$(8.76973_{10^{-2}}, 4.11936_{10^{-1}})$	$(5.12723_{10^{-1}}, -1.81871_{10^{-1}})$
$(5.61990_{10^{-1}}, 0.00000)$	$(3.14212_{10^{-1}}, 4.61661_{10^{-2}})$
$(5.36108_{10^{-2}}, -1.48074_{10^{-1}})$	$(1.18891_{10^{-1}}, 6.62182_{10^{-2}})$
$(-1.48308_{10^{-1}}, -5.02089_{10^{-1}})$	$(5.73801_{10^{-1}}, 0.00000)$

VL (*continued*)

$(-2.41800_{10^{-1}}, 1.14309_{10^{-1}})$	$(2.03059_{10^{-1}}, -4.36209_{10^{-2}})$
$(6.91175_{10^{-1}}, 0.00000)$	$(5.78956_{10^{-1}}, 0.00000)$
$(2.59987_{10^{-2}}, 2.27635_{10^{-1}})$	$(-9.63880_{10^{-2}}, 4.99523_{10^{-1}})$
$(5.31360_{10^{-1}}, 1.13176_{10^{-1}})$	$(2.48811_{10^{-2}}, -3.86579_{10^{-1}})$
$(-1.40565_{10^{-1}}, 2.88686_{10^{-1}})$	$(-3.41421_{10^{-1}}, 3.10209_{10^{-1}})$

VL (*continued*)

$(2.32683_{10^{-1}}, 4.03721_{10^{-1}})$
$(-3.27815_{10^{-1}}, -1.75589_{10^{-1}})$
$(4.24576_{10^{-1}}, 1.76178_{10^{-1}})$
$(-1.18324_{10^{-1}}, 7.56740_{10^{-2}})$
$(6.43074_{10^{-1}}, 0.00000)$

VR

$(6.16325_{10^{-1}}, 0.00000)$	$(-2.03992_{10^{-1}}, -1.52540_{10^{-1}})$
$(3.79401_{10^{-1}}, 4.30749_{10^{-2}})$	$(4.25472_{10^{-1}}, 2.10096_{10^{-1}})$
$(-4.24464_{10^{-1}}, -2.28062_{10^{-1}})$	$(-2.53433_{10^{-1}}, -3.20503_{10^{-1}})$
$(-2.22189_{10^{-1}}, 9.43230_{10^{-2}})$	$(-2.50176_{10^{-1}}, -1.67329_{10^{-1}})$
$(2.72903_{10^{-1}}, 3.30787_{10^{-1}})$	$(6.72618_{10^{-1}}, 0.00000)$

VR (*continued*)

$(1.74764_{10^{-1}}, -8.11960_{10^{-2}})$	$(-4.16383_{10^{-1}}, 2.18125_{10^{-1}})$
$(-8.30785_{10^{-2}}, -1.00695_{10^{-1}})$	$(-4.91021_{10^{-1}}, 1.22142_{10^{-1}})$
$(4.12709_{10^{-1}}, -7.98013_{10^{-2}})$	$(-2.17545_{10^{-1}}, -3.72154_{10^{-1}})$
$(7.73319_{10^{-1}}, 0.00000)$	$(5.52272_{10^{-1}}, 0.00000)$
$(-1.90435_{10^{-1}}, 3.67204_{10^{-1}})$	$(1.43970_{10^{-1}}, 1.07102_{10^{-1}})$

VR (*continued*)
$(\ 7.45969_{10^{-1}},\ \ 0.00000)$
$(\ 2.54283_{10^{-1}},\ \ 4.21799_{10^{-1}})$
$(\ 3.79400_{10^{-1}}, -5.27684_{10^{-2}})$
$(-2.19337_{10^{-2}}, -1.29830_{10^{-1}})$
$(\ 1.00520_{10^{-1}},\ \ 1.63665_{10^{-1}})$

$$\mathbf{INFO} = 0$$

The eigenvalues of matrix A are:

$$\begin{pmatrix} -11.3885 & -\,4.99204i \\ -5.08344_{10^{-1}} & -\,3.70385i \\ 5.39333 & -\,1.70267i \\ 2.47931 & +\,7.98679i \\ -9.75794_{10^{-1}} & +\,2.41177i \end{pmatrix}.$$

The left and right eigenvectors of matrix A are (columnwise):

$$\left(\begin{array}{ll} -4.42699_{10^{-1}} - 1.09152_{10^{-1}}i & -2.90780_{10^{-1}} + 4.13348_{10^{-1}}i \\ 8.76973_{10^{-2}} + 4.11936_{10^{-1}}i & 5.12723_{10^{-1}} - 1.81871_{10^{-1}}i \\ 5.61990_{10^{-1}} & 3.14212_{10^{-1}} + 4.61661_{10^{-2}}i \\ 5.36108_{10^{-2}} - 1.48074_{10^{-1}}i & 1.18891_{10^{-1}} + 6.62182_{10^{-2}}i \\ -1.48308_{10^{-1}} - 5.02089_{10^{-1}}i & 5.738012_{10^{-1}} \end{array} \right.$$

$$\begin{array}{ll} -2.41800_{10^{-1}} + 1.14309_{10^{-1}}i & -2.03059_{10^{-1}} - 4.36209_{10^{-2}}i \\ 6.91175_{10^{-1}} & 5.78956_{10^{-1}} \\ 2.59987_{10^{-2}} + 2.27635_{10^{-1}}i & -9.63880_{10^{-2}} + 4.99523_{10^{-1}}i \\ 5.31360_{10^{-1}} + 1.13176_{10^{-1}}i & -2.48811_{10^{-2}} - 3.86579_{10^{-1}}i \\ -1.40565_{10^{-1}} + 2.88686_{10^{-1}}i & -3.41421_{10^{-1}} + 3.10209_{10^{-1}}i \end{array}$$

$$\left. \begin{array}{l} 2.32683_{10^{-1}} + 4.03721_{10^{-1}}i \\ -3.27815_{10^{-1}} - 1.75589_{10^{-1}}i \\ 4.24576_{10^{-1}} + 1.76178_{10^{-1}}i \\ -1.18324_{10^{-1}} + 7.56740_{10^{-2}}i \\ 6.43074_{10^{-1}} \end{array} \right)$$

and

$$\left(\begin{array}{ll} 6.16325_{10^{-1}} & -2.03992_{10^{-1}} - 1.52540_{10^{-1}}i \\ 3.79401_{10^{-1}} + 4.30749_{10^{-2}}i & 4.25472_{10^{-1}} + 2.10096_{10^{-1}}i \\ -4.24464_{10^{-1}} - 2.28062_{10^{-1}}i & -2.53433_{10^{-1}} - 3.20503_{10^{-1}}i \\ -2.22189_{10^{-1}} + 9.43230_{10^{-2}}i & -2.50176_{10^{-1}} - 1.67329_{10^{-1}}i \\ 2.72903_{10^{-1}} + 3.30787_{10^{-1}}i & 6.72618_{10^{-1}} \end{array} \right.$$

$$
\begin{array}{cc}
1.74764_{10^{-1}} - 8.11960_{10^{-2}}i & -4.16383_{10^{-1}} + 2.18125_{10^{-1}}i \\
-8.30785_{10^{-2}} - 1.00695_{10^{-1}}i & -4.91021_{10^{-1}} + 1.22142_{10^{-1}}i \\
4.12709_{10^{-1}} - 7.98013_{10^{-2}}i & -2.17545_{10^{-1}} - 3.72154_{10^{-1}}i \\
7.73319_{10^{-1}} & 5.52272_{10^{-1}} \\
-1.90435_{10^{-1}} + 3.67204_{10^{-1}}i & 1.43970_{10^{-1}} + 1.07102_{10^{-1}}i
\end{array}
$$

$$
\left.
\begin{array}{c}
7.45969_{10^{-1}} \\
2.54283_{10^{-1}} + 4.21799_{10^{-1}}i \\
3.79400_{10^{-1}} - 5.27684_{10^{-2}}i \\
-2.19337_{10^{-2}} - 1.29830_{10^{-1}}i \\
1.00520_{10^{-1}} + 1.63665_{10^{-1}}i
\end{array}
\right) .
$$

Chapter 4

Driver Routines for Generalized Eigenvalue Problems

4.1 Generalized Symmetric Eigenvalue Problems

4.1.1 LA_SYGV / LA_HEGV

Purpose

LA_SYGV and **LA_HEGV** compute all the eigenvalues, and optionally, the eigenvectors of generalized eigenvalue problems of the form

$$A\,x = \lambda\,B\,x, \quad A\,Bx = \lambda\,x, \quad \text{or} \quad B\,A\,x = \lambda\,x,$$

where A and B are real symmetric in the case of **LA_SYGV** and complex Hermitian in the case of **LA_HEGV**. In both cases B is positive definite.

Specification

SUBROUTINE LA_SYGV / LA_HEGV(A, B, W, ITYPE=itype,
JOBZ=jobz, **UPLO**=uplo, **INFO**=info **)**

type(*wp*)**, INTENT(INOUT) :: A(:,:), B(:,:)**
REAL(*wp***), INTENT(OUT) :: W(:)**
INTEGER, INTENT(IN), OPTIONAL :: ITYPE
CHARACTER(LEN=1), INTENT(IN), OPTIONAL :: &
JOBZ, UPLO
INTEGER, INTENT(OUT), OPTIONAL :: INFO
where
type **::= REAL | COMPLEX**
wp **::= KIND(1.0) | KIND(1.0D0)**

Arguments

A – (*Input/Output*) **REAL** or **COMPLEX** square array, shape $(:,:)$.

- On entry, the matrix A.
  - If **UPLO** = 'U', the upper triangular part of **A** contains the upper triangular part of matrix A.
  - If **UPLO** = 'L', the lower triangular part of **A** contains the lower triangular part of matrix A.
- On exit:
  - If **JOBZ** = 'V' and **INFO** = 0, then the columns of **A** contain the eigenvectors, normalized as follows:
    * if **ITYPE** = 1 or 2: $Z^H B Z = I$;
    * if **ITYPE** = 3: $Z^H B^{-1} Z = I$.

B – (*Input/output*) **REAL** or **COMPLEX** square array, shape $(:,:)$, and $size(\mathbf{B},1) = size(\mathbf{A},1)$.

- On entry, the matrix B.
  - If **UPLO** = 'U', the upper triangular part of **B** contains the upper triangular part of matrix B.
  - If **UPLO** = 'L', the lower triangular part of **B** contains the lower triangular part of matrix B.
- On exit, if **INFO** $\leq size(A,1)$, the part of **B** containing the matrix is overwritten by the triangular factor U or L of the Cholesky factorization $B = U^H U$ or $B = L L^H$.

W – (*Output*) **REAL** array, shape $(:)$, $size(\mathbf{W}) = size(\mathbf{A},1)$.

- The eigenvalues in ascending order.

ITYPE – *Optional*, (*Input*) **INTEGER**. Specifies the problem type to be solved.

- $= 1 \; : \; A\,x = \lambda\,B\,x$
  $= 2 \; : \; A\,B\,x = \lambda\,x$
  $= 3 \; : \; B\,A\,x = \lambda\,x$
- If **ITYPE** is not present then **ITYPE** = 1 is assumed.

JOBZ – *Optional*, (*Input*) **CHARACTER(LEN=1)**.

- = **'N'** : Compute eigenvalues only;
  = **'V'** : Compute eigenvalues and eigenvectors.
- If **JOBZ** is not present then **JOBZ** = 'N' is assumed.

UPLO – *Optional*, (*Input*) **CHARACTER(LEN=1)**.

- = **'U'** : Upper triangle of A is stored and B is factored as $U^H U$;
  = **'L'** : Lower triangle of A is stored and B is factored as $L\,L^H$.
- If **UPLO** is not present **UPLO** = 'U' is assumed.

INFO – *Optional*, (*Output*) **INTEGER**.

- $= 0$: successful exit
  < 0: if **INFO** $= -i$, the i^{th} argument had an illegal value
  > 0:
  $\leq size(\mathbf{A}, 1)$: if **INFO** $= i$, then i off-diagonal elements of an intermediate tridiagonal form did not converge to zero;
  $> size(\mathbf{A}, 1)$: if **INFO** $= size(\mathbf{A}, 1) + i$, then the leading minor of order i of B is not positive definite. The factorization of B could not be completed and no eigenvalues or eigenvectors were computed.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_SYGV_EXAMPLE)

$$A = \begin{pmatrix} 4 & -10 & -3 & 2 & -5 \\ -10 & -2 & -11 & 5 & 0 \\ -3 & -11 & -6 & -10 & 8 \\ 2 & 5 & -10 & -3 & 0 \\ -5 & 0 & 8 & 0 & 7 \end{pmatrix}, \quad B = \begin{pmatrix} 8 & 2 & 2 & 2 & -1 \\ 2 & 13 & 0 & 0 & 1 \\ 2 & 0 & 13 & 0 & -2 \\ 2 & 0 & 0 & 14 & 3 \\ -1 & 1 & -2 & 3 & 3 \end{pmatrix}$$

Arrays **A** and **B** on entry:

A

4	−10	−3	2	−5
*	−2	−11	5	0
*	*	−6	−10	8
*	*	*	−3	0
*	*	*	*	7

B

8	2	2	2	−1
*	13	0	0	1
*	*	13	0	−2
*	*	*	14	3
*	*	*	*	3

Elements marked * are not used by the routine.

The call:

CALL LA_SYGV(A, B, W)

B and **W** on exit:

B

2.82843	$7.07107_{10^{-1}}$	$7.07107_{10^{-1}}$	$7.07107_{10^{-1}}$	$-3.53553_{10^{-1}}$
2.00000	3.53553	$-1.41421_{10^{-1}}$	$-1.41421_{10^{-1}}$	$3.53553_{10^{-1}}$
2.00000	0.00000	3.53270	$-1.47196_{10^{-1}}$	$-4.81218_{10^{-1}}$
2.00000	0.00000	0.00000	3.66856	$8.80227_{10^{-1}}$
−1.00000	1.00000	−2.00000	3.00000	1.32047

W

−1.72115
−1.18720
$1.29015_{10^{-1}}$
1.58752
6.01579

The eigenvalues of the problem $A\,x = \lambda\,B\,x$ are:

$$\begin{pmatrix} -1.72115 \\ -1.18720 \\ 1.29015_{10^{-1}} \\ 1.58752 \\ 6.01579 \end{pmatrix}.$$

The triangular factor U of the Cholesky factorization of B is:

$$U = \begin{pmatrix} 2.82843 & 7.07107_{10^{-1}} & 7.07107_{10^{-1}} & 7.07107_{10^{-1}} & -3.53553_{10^{-1}} \\ 0 & 3.53553 & -1.41421_{10^{-1}} & -1.41421_{10^{-1}} & 3.53553_{10^{-1}} \\ 0 & 0 & 3.53270 & -1.47196_{10^{-1}} & -4.81218_{10^{-1}} \\ 0 & 0 & 0 & 3.66856 & 8.80227_{10^{-1}} \\ 0 & 0 & 0 & 0 & 1.32047 \end{pmatrix}.$$

Example 2 (from Program LA_SYGV_EXAMPLE)

Matrices A and B as in Example 1.

Arrays **A** and **B** on entry:

A

4	*	*	*	*
−10	−2	*	*	*
−3	−11	−6	*	*
2	5	−10	−3	*
−5	0	8	0	7

B

8	*	*	*	*
2	13	*	*	*
2	0	13	*	*
2	0	0	14	*
−1	1	−2	3	3

Elements marked * are not used by the routine.

The call:

CALL LA_SYGV(A, B, W, 2, ’V’, ’L’, INFO)

A, **B**, **W**, and **INFO** on exit:

A

$-7.28022_{10^{-2}}$	$-1.43685_{10^{-1}}$	$2.27996_{10^{-1}}$	$2.56489_{10^{-1}}$	$1.25081_{10^{-1}}$
$-1.55449_{10^{-1}}$	$-1.09085_{10^{-1}}$	$-1.01512_{10^{-1}}$	$-4.64006_{10^{-2}}$	$-1.94946_{10^{-1}}$
$-1.89317_{10^{-1}}$	$8.05219_{10^{-2}}$	$5.68352_{10^{-2}}$	$-1.59253_{10^{-1}}$	$1.35932_{10^{-1}}$
$-5.58084_{10^{-2}}$	$2.40823_{10^{-1}}$	$-1.03728_{10^{-1}}$	$1.10736_{10^{-1}}$	$-1.52279_{10^{-1}}$
$4.42886_{10^{-2}}$	$-1.08632_{10^{-1}}$	$6.92130_{10^{-1}}$	$-2.81321_{10^{-1}}$	$3.95988_{10^{-2}}$

B

2.82843	2.00000	2.00000	2.00000	−1.00000
$7.07107_{10^{-1}}$	3.53553	0.00000	0.00000	1.00000
$7.07107_{10^{-1}}$	$-1.41421_{10^{-1}}$	3.53270	0.00000	−2.00000
$7.07107_{10^{-1}}$	$-1.41421_{10^{-1}}$	$-1.47196_{10^{-1}}$	3.66856	3.00000
$-3.53553_{10^{-1}}$	$3.53553_{10^{-1}}$	−4.81218	$8.80227_{10^{-1}}$	1.32047

W

−284.343
−117.700
5.14901
78.7262
159.168

INFO = 0

The eigenvalues of the problem $A\,x = \lambda\,B\,x$ are:

$$\begin{pmatrix} -284.343 \\ -117.700 \\ 5.14901 \\ 78.7262 \\ 159.168 \end{pmatrix}.$$

The eigenvectors are:

$$\begin{pmatrix} -7.28022_{10^{-2}} & -1.43685_{10^{-1}} & 2.27996_{10^{-1}} & 2.56489_{10^{-1}} & 1.25081_{10^{-1}} \\ -1.55449_{10^{-1}} & -1.09085_{10^{-1}} & -1.01512_{10^{-1}} & -4.64006_{10^{-2}} & -1.94946_{10^{-1}} \\ -1.89317_{10^{-1}} & 8.05219_{10^{-2}} & 5.68352_{10^{-2}} & -1.59253_{10^{-1}} & 1.35932_{10^{-1}} \\ -5.58084_{10^{-2}} & 2.40823_{10^{-1}} & -1.03728_{10^{-1}} & 1.10736_{10^{-1}} & -1.52279_{10^{-1}} \\ 4.42886_{10^{-2}} & -1.08632_{10^{-1}} & 6.92130_{10^{-1}} & -2.81321_{10^{-1}} & 3.95988_{10^{-2}} \end{pmatrix}.$$

The triangular factor L of the Cholesky factorization of B is:

$$L = \begin{pmatrix} 2.82843 & 0 & 0 & 0 & 0 \\ 7.07107_{10^{-1}} & 3.53553 & 0 & 0 & 0 \\ 7.07107_{10^{-1}} & -1.41421_{10^{-1}} & 3.53270 & 0 & 0 \\ 7.07107_{10^{-1}} & -1.41421_{10^{-1}} & -1.47196_{10^{-1}} & 3.66856 & 0 \\ -3.53553_{10^{-1}} & 3.53553_{10^{-1}} & -4.81218 & 8.80227_{10^{-1}} & 1.32047 \end{pmatrix}.$$

4.1.2 LA_SPGV / LA_HPGV

Purpose

LA_SPGV and LA_HPGV compute all the eigenvalues and, optionally, the eigenvectors of generalized eigenvalue problems of the form

$$A\,x = \lambda\,B\,x, \quad A\,Bx = \lambda\,x, \quad \text{or} \quad B\,A\,x = \lambda\,x,$$

where A and B are real symmetric in the case of **LA_SPGV** and complex Hermitian in the case of **LA_HPGV**. In both cases B is positive definite. Here A and B are stored in a packed format.

Specification

SUBROUTINE LA_SPGV / LA_HPGV(AP, BP, W, &
ITYPE=itype, **UPLO**=uplo, **Z**=z, **INFO**=info)
type(*wp*), **INTENT(INOUT) :: AP(:), BP(:)**
REAL(*wp*), INTENT(OUT) :: W(:)
INTEGER, INTENT(IN), OPTIONAL :: ITYPE
CHARACTER(LEN=1), INTENT(IN), OPTIONAL :: UPLO
type(*wp*), **INTENT(OUT), OPTIONAL :: Z(:,:)**
INTEGER, INTENT(OUT), OPTIONAL :: INFO
where
type **::= REAL | COMPLEX**
wp **::= KIND(1.0) | KIND(1.0D0)**

Arguments

AP – (*Input/Output*) **REAL** or **COMPLEX** array, shape (:).

- On entry, the upper or lower triangle of matrix A in packed storage. $size(\mathbf{AP}) = n(n+1)/2$, where n is the order of A and B. The elements are stored columnwise as follows:

$A_{i,j}$	i,j	**UPLO**
$\mathbf{AP}_{i+(j-1)j/2}$	$1 \le j \le n$ $1 \le i \le j$	'U'
$\mathbf{AP}_{i+(j-1)(2n-j)/2}$	$1 \le j \le n$ $j \le i \le n$	'L'

- On exit, the contents of **AP** are destroyed.

BP – (*Input/Output*) **REAL** or **COMPLEX** array, shape (:), and $size(\mathbf{BP}) = size(\mathbf{AP})$.

- On entry, the upper or lower triangle of matrix B in packed storage. The elements are stored columnwise as follows:

$B_{i,j}$	i,j	**UPLO**
$\mathbf{BP}_{i+(j-1)j/2}$	$1 \le j \le n$ $1 \le i \le j$	'U'
$\mathbf{BP}_{i+(j-1)(2n-j)/2}$	$1 \le j \le n$ $j \le i \le n$	'L'

- On exit, the triangular factor U or L of the Cholesky factorization $B = U^T U$ or $B = L\,L^T$, in the same storage format as B.

W – (*Output*) **REAL** array, shape (:), and $size(\mathbf{W}) = n$.

- The eigenvalues in ascending order.

ITYPE – *Optional* (*Input*) **INTEGER**. Specifies the problem type to be solved:

- $= 1$: $A\,x = \lambda\, B\, x$
  $= 2$: $A\, B\, x = \lambda\, x$
  $= 3$: $B\, A\, x = \lambda\, x$

If **ITYPE** is not present, **ITYPE** $= 1$ is assumed.

UPLO – *Optional* (*Input*) **CHARACTER(LEN=1)**.

- $=$ **'U'** : Upper triangles of A and B are stored;
  $=$ **'L'** : Lower triangles of A and B are stored.

If **UPLO** is not present, **UPLO** = 'U' is assumed.

Z – *Optional* (*Output*) **REAL** or **COMPLEX** square array, shape $(:,:)$, and $size(Z,1) = n$.

- **Z** contains the matrix Z of eigenvectors, normalized as follows:
  - if **ITYPE** $= 1$ or 2: $Z^T\, B\, Z = I$;
  - if **ITYPE** $= 3$: $Z^T\, B^{-1}\, Z = I$.

INFO – *Optional*, (*Output*) **INTEGER**.

- $= 0$: successful exit
  < 0 : if **INFO** $= -i$, the i^{th} argument had an illegal value
  > 0 :
  $\leq n$: if **INFO** $= i$, then i off-diagonal elements of an intermediate tridiagonal form did not converge to zero.
  $> n$: if **INFO** $= n + i$, then the leading minor of order i of B is not positive definite. The factorization of B could not be completed and no eigenvalues or eigenvectors were computed.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_HPGV_EXAMPLE)

$$A = \begin{pmatrix} 0 & 9-2i & 5-2i & 3-i & 7-4i \\ 9+2i & 5 & 2+3i & -9-i & 7-i \\ 5+2i & 2-3i & 4 & 3 & -4+i \\ 3+i & -9+i & 3 & 7 & -11 \\ 7+4i & 7+i & -4-i & -11 & 9 \end{pmatrix},$$

$$B = \begin{pmatrix} 10 & -1-4i & -2-3i & -6+4i & -1+i \\ -1+4i & 11 & 1-2i & i & 1 \\ -2+3i & 1+2i & 10 & 5-3i & -1 \\ -6-4i & -i & 5+3i & 12 & -1-i \\ -1-i & 1 & -1 & -1+i & 11 \end{pmatrix}$$

Arrays **AP** and **BP** on entry:

AP

$(0,0)$	$(9,-2)$	$(5,0)$	$(5,-2)$	$(2,3)$	$(4,0)$	$(3,-1)$	$(-9,-1)$

AP continued

$(3,0)$	$(7,0)$	$(7,-4)$	$(7,-1)$	$(-4,1)$	$(-11,0)$	$(9,0)$

BP

$(10,0)$	$(-1,-4)$	$(11,0)$	$(-2,-3)$	$(1,-2)$	$(10,0)$	$(-6,4)$	$(0,1)$

BP continued

$(5,-3)$	$(12,0)$	$(-1,1)$	$(1,0)$	$(-1,0)$	$(-1,-1)$	$(11,0)$

The call:

CALL LA_HPGV(AP, BP, W)

BP and **W** on exit:

BP

$(3.16228, 0.00000)$	$(-3.16228_{10^{-1}}, -1.26491)$	$(3.04959, 0.00000)$

BP continued

$(-6.32456_{10^{-1}}, -9.48683_{10^{-1}})$	$(-1.31165_{10^{-1}}, -4.91869_{10^{-1}})$	$(2.90532, 0.00000)$

BP continued

$(-1.89737, 1.26491)$	$(3.27913_{10^{-1}}, 1.24607)$	$(1.94675, -1.36939_{10^{-1}})$

BP continued

$(1.15378, 0.00000)$	$(-3.16228_{10^{-1}}, 3.16228_{10^{-1}})$	$(4.26287_{10^{-1}}, 1.63957_{10^{-1}})$

BP continued

$(-2.62774_{10^{-1}}, 1.07330_{10^{-1}})$	$(-1.57555, -4.29493_{10^{-1}})$	$(2.80072, 0.00000)$

W
-1.02534
$-4.96729_{10^{-1}}$
$6.13611_{10^{-1}}$
1.56312
16.6680

The eigenvalues of the problem $A\,x = \lambda\,B\,x$ are:

$$\begin{pmatrix} -1.02534 \\ -4.96729_{10^{-1}} \\ 6.13611_{10^{-1}} \\ 1.56312 \\ 16.6680 \end{pmatrix}.$$

The triangular factor U of the Cholesky factorization of B is:

$$U = \begin{pmatrix} 3.16228 & -3.16228_{10^{-1}} - 1.26491i & -6.32456_{10^{-1}} - 9.48683_{10^{-1}}i \\ 0 & 3.04959 & -1.31165_{10^{-1}} - 4.91869_{10^{-1}}i \\ 0 & 0 & 2.90532 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix.$$

$$\begin{matrix} -1.89737 + 1.26491i & -3.16228_{10^{-1}} + 3.16228_{10^{-1}}i \\ 3.27913_{10^{-1}} + 1.24607i & 4.26287_{10^{-1}} + 1.63957_{10^{-1}}i \\ 1.94675 - 1.36939_{10^{-1}}i & -2.62774_{10^{-1}} + 1.07330_{10^{-1}}i \\ 1.15378 & -1.57555 - 4.29493_{10^{-1}}i \\ 0 & 2.80072 \end{matrix}\right).$$

Example 2 (from Program LA_HPGV_EXAMPLE)

Matrices A and B as in Example 1.

Arrays **AP** and **BP** on entry:

AP

$(0,0)$	$(9,2)$	$(5,2)$	$(3,1)$	$(7,4)$	$(5,0)$	$(2,-3)$	$(-9,1)$

AP continued

$(7,1)$	$(4,0)$	$(3,0)$	$(-4,-1)$	$(7,0)$	$(-11,0)$	$(9,0)$

BP

$(10,0)$	$(-1,4)$	$(-2,3)$	$(-6,-4)$	$(-1,-1)$	$(11,0)$	$(1,2)$

BP continued

$(0,-1)$	$(1,0)$	$(10,0)$	$(5,3)$	$(-1,0)$	$(12,0)$	$(-1,1)$	$(11,0)$

The call:

CALL LA_HPGV(AP, BP, W, 3, 'L', Z, INFO)

BP, **W**, **Z**, and **INFO** on exit:

BP

$(3.16228, 0.00000)$	$(-3.16228_{10^{-1}}, 1.26491)$	$(-6.32456_{10^{-1}}, 9.48683_{10^{-1}})$

BP continued

$(-1.89737, -1.26491)$	$(-3.16228_{10^{-1}}, -3.16228_{10^{-1}})$	$(3.04959, 0.00000)$

BP continued

$(-1.31165_{10^{-1}}, 4.91869_{10^{-1}})$	$(3.27913_{10^{-1}}, -1.24607)$	$(4.26287_{10^{-1}}, -1.63956_{10^{-1}})$

BP continued

$(2.90532, 0.000000)$	$(1.94675, 1.36938_{10^{-1}})$	$(-2.62774_{10^{-1}}, -1.07330_{10^{-1}})$

BP continued

$(1.15378, 0.00000)$	$(-1.57555, 4.29493_{10^{-1}})$	$(2.80072, 0.00000)$

W

-203.206
-35.5916
1.73180
111.573
391.493

Z

$(2.42010, 0.00000)$	$(-1.00862_{10^{-1}}, 0.00000)$
$(-1.40369, 2.00023_{10^{-1}})$	$(-8.68867_{10^{-1}}, -1.90827)$
$(-9.31750_{10^{-1}}, 1.28187_{10^{-1}})$	$(1.13129, 5.89736_{10^{-1}})$
$(-2.22278, -7.68252_{10^{-1}})$	$(-4.87970_{10^{-1}}, 2.77526_{10^{-1}})$
$(-1.41174, -8.15877_{10^{-1}})$	$(3.00437_{10^{-1}}, 1.63311)$

Z

$(2.60276_{10^{-1}}, 0.00000)$	$(-1.50507, 0.00000)$
$(-1.31807_{10^{-1}}, -3.63407_{10^{-1}})$	$(-5.59433_{10^{-1}}, -1.30728)$
$(-1.98489_{10^{-1}}, 6.70745_{10^{-2}})$	$(-1.38829, -1.98335)$
$(3.62049_{10^{-1}}, 1.23805_{10^{-1}})$	$(-6.22834_{10^{-1}}, -3.34956_{10^{-1}})$
$(2.45206_{10^{-1}}, 3.51769_{10^{-1}})$	$(-7.92847_{10^{-2}}, 6.16168_{10^{-1}})$

Z continued

(−1.34162, 0.00000)
(−1.11936, −1.08115)
(1.24386, $1.88800_{10^{-1}}$)
(1.26824, 1.97457)
(−1.68730, −1.47214)

INFO = 0

The eigenvalues of the problem $BAx = \lambda x$ are:

$$\begin{pmatrix} -203.206 \\ -35.5916 \\ 1.73180 \\ 111.573 \\ 391.493 \end{pmatrix}.$$

The eigenvectors are:

$$\begin{pmatrix} 2.42010 & -1.00862_{10^{-1}} \\ -1.40369 + 2.00023_{10^{-1}}i & -8.68867_{10^{-1}} - 1.90827i \\ -9.31750_{10^{-1}} + 1.28187_{10^{-1}}i & 1.13129 + 5.89736_{10^{-1}}i \\ -2.22278 - 7.68252_{10^{-1}}i & -4.87970_{10^{-1}} + 2.77526_{10^{-1}}i \\ -1.41174 - 8.15877_{10^{-1}}i & 3.00437_{10^{-1}} + 1.63311i \end{pmatrix.}$$

$$\begin{matrix} 2.60276_{10^{-1}} & -1.50507 \\ -1.31807_{10^{-1}} - 3.63407_{10^{-1}}i & -5.59433_{10^{-1}} - 1.30728i \\ -1.98489_{10^{-1}} + 6.70745_{10^{-2}}i & -1.38829 - 1.98335i \\ 3.62049_{10^{-1}} + 1.23805_{10^{-1}}i & -6.22834_{10^{-1}} - 3.34956_{10^{-1}}i \\ 2.45206_{10^{-1}} + 3.51769_{10^{-1}}i & -7.92847_{10^{-2}} + 6.16168_{10^{-1}}i \end{matrix}$$

$$\left.\begin{matrix} -1.34162 \\ -1.11936 - 1.08115i \\ 1.24386 + 1.88800_{10^{-1}}i \\ 1.26824 + 1.97457i \\ -1.68730 - 1.47214i \end{matrix}\right).$$

The triangular factor L from the Cholesky factorization of B is:

$$L = \begin{pmatrix} 3.16228 & 0 \\ -3.16228_{10^{-1}} + 1.26491i & 3.04959 \\ -6.32456_{10^{-1}} + 9.48683_{10^{-1}}i & -1.31165_{10^{-1}} + 4.91869_{10^{-1}}i \\ -1.89737 - 1.26491i & 3.27913_{10^{-1}} - 1.24607i \\ -3.16228_{10^{-1}} - 3.16228_{10^{-1}}i & 4.26287_{10^{-1}} - 1.63956_{10^{-1}}i \end{matrix}$$

$$
\left.\begin{matrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
2.90532 & 0 & 0 \\
1.94675 + 1.36938_{10^{-1}}\, i & 1.15378 & 0 \\
-2.62774_{10^{-1}} - 1.07330_{10^{-1}}\, i & -1.57555 + 4.29493_{10^{-1}}\, i & 2.80072
\end{matrix}\right) .
$$

4.1.3 LA_SBGV / LA_HBGV

Purpose

LA_SBGV and LA_HBGV compute all the eigenvalues and, optionally, the eigenvectors of the generalized eigenvalue problem

$$A\,x = \lambda\,B\,x,$$

where A and B are real symmetric in the case of **LA_SBGV** and complex Hermitian in the case of **LA_HBGV**. Matrix B is positive definite. Here A and B are stored in a band format.

Specification

SUBROUTINE LA_SBGV / LA_HBGV(AB, BB, W, &
UPLO=uplo, **Z**=z, **INFO**=info)
  type(*wp*), **INTENT(INOUT) :: AB(:,:), BB(:,:)**
  REAL(*wp*), **INTENT(OUT) :: W(:)**
  CHARACTER(LEN=1), INTENT(IN), OPTIONAL :: UPLO
  type(*wp*), **INTENT(OUT), OPTIONAL :: Z(:,:)**
  INTEGER, INTENT(OUT), OPTIONAL :: INFO
  where
  type **::= REAL | COMPLEX**
  wp **::= KIND(1.0) | KIND(1.0D0)**

Arguments

AB – (*Input/Output*) **REAL** or **COMPLEX** array, shape (:, :).

- On entry, the upper (if **UPLO** = 'U') or lower (if **UPLO** = 'L') triangle of matrix A in band storage. The dimensions of **AB** are

  $$size(\mathbf{AB}, 1) = ka + 1, \quad size(\mathbf{AB}, 2) = n$$

  where ka is the number subdiagonals or superdiagonals in the band of A and n is the order of A and B. The $ka + 1$ diagonals of A are stored rowwise in **AB**. The j^{th} column of A is stored in the j^{th} column of the array **AB** as follows:

$A_{i,j}$	i, j	**UPLO**
$\mathbf{AB}_{ka+1+i-j,j}$	$\max(1, j-ka) \le i \le j$ $1 \le j \le n$	'U'
$\mathbf{AB}_{1+i-j,j}$	$j \le i \le min(n, j+ka)$ $1 \le j \le n$	'L'

- On exit, the contents of **AB** are destroyed.

BB – (*Input/Output*) **REAL / COMPLEX** array, shape (:, :), and $size(\mathbf{BB}, 1) = size(\mathbf{AB}, 1)$, $size(\mathbf{BB}, 2) = size(\mathbf{AB}, 2)$.

- On entry, the upper (if **UPLO** = 'U') or lower (if **UPLO** = 'L') triangle of matrix B in band storage. The dimensions of **BB** are

  $$size(\mathbf{BB}, 1) = kb + 1, \quad size(\mathbf{BB}, 2) = n.$$

  The $kb+1$ diagonals of B are stored rowwise in **BB**. The j^{th} column of B is stored in the j^{th} column of the array **BB** as follows:

$B_{i,j}$	i, j	**UPLO**
$\mathbf{BB}_{kb+1+i-j,j}$	$\max(1, j-kb) \leq i \leq j$ $1 \leq j \leq n$	$'\mathsf{U}'$
$\mathbf{BB}_{1+i-j,j}$	$j \leq i \leq min(n, j+kb)$ $1 \leq j \leq n$	$'\mathsf{L}'$

- On exit, the factor S from the split Cholesky factorization

  $$B = S^T S$$

  .

W – (*Output*) **REAL** array, shape (:), and $size(\mathbf{W}) = size(\mathbf{AB}, 2)$.

- The eigenvalues in ascending order.

UPLO – UPLO *Optional* (*Input*) **CHARACTER(LEN=1)**.

- = 'U': Upper triangles of A and B are stored;
  = 'L': Lower triangles of A and B are stored.

If **UPLO** is not present, **UPLO** = 'U' is assumed.

Z – *Optional* (*Output*) **REAL** or **COMPLEX** square array, shape (:, :), and $size(\mathbf{Z}, 1) = size(\mathbf{AB}, 2)$.

- **Z** contains the matrix Z of eigenvectors, normalized so that

  $$Z^T B Z = I$$

  .

INFO – *Optional* (*Output*) **INTEGER**.

- $= 0$: successful exit
  < 0 : if **INFO** $= -i$, the i^{th} argument had an illegal value
  > 0 :
  - $\leq n$: if **INFO** $= i$, then i off-diagonal elements of an intermediate tridiagonal form did not converge to zero.
  - $> n$: if **INFO** $= n+i$, for $1 \leq i \leq n$, then the leading minor of order i of B is not positive definite. The factorization of B could not be completed and no eigenvalues or eigenvectors were computed.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_SBGV_EXAMPLE)

$$A = \begin{pmatrix} 6 & 5 & -4 & 0 & 0 \\ 5 & -5 & 4 & -3 & 0 \\ -4 & 4 & 0 & 4 & -4 \\ 0 & -3 & 4 & 0 & 4 \\ 0 & 0 & -4 & 4 & 9 \end{pmatrix}, \quad B = \begin{pmatrix} 10 & 0 & -5 & 0 & 0 \\ 0 & 8 & 2 & 2 & 0 \\ -5 & 2 & 8 & 2 & 0 \\ 0 & 2 & 2 & 8 & 3 \\ 0 & 0 & 0 & 3 & 10 \end{pmatrix}$$

Arrays **AB** and **BB** on entry:

AB

0	0	-4	-3	-4
0	5	4	4	4
6	-5	0	0	9

BB

0	0	-5	2	0
0	0	2	2	3
10	8	8	8	10

The call:

CALL LA_SBGV(AB, BB, W)

BB and **W** on exit:

BB

0.00000	0.00000	-1.58114	$7.50587_{10^{-1}}$	0.00000
0.00000	0.00000	$5.26810_{10^{-1}}$	$7.50587_{10^{-1}}$	$9.48683_{10^{-1}}$
3.16228	2.72702	2.15849	2.66458	3.16228

W

-2.95028
$-2.60316_{10^{-1}}$
$3.37961_{10^{-1}}$
$8.63341_{10^{-1}}$
1.12617

The eigenvalues of the problem $A\,x = \lambda\,B\,x$ are:

$$\begin{pmatrix} -2.95028 \\ -2.60316_{10^{-1}} \\ 3.37961_{10^{-1}} \\ 8.63341_{10^{-1}} \\ 1.12617 \end{pmatrix}.$$

The split Cholesky factor S is:

$$\begin{pmatrix} 0.00000 & 0.00000 & -1.58114 & 7.50587_{10^{-1}} & 0.00000 \\ 0.00000 & 0.00000 & 5.26810_{10^{-1}} & 7.50587_{10^{-1}} & 9.48683_{10^{-1}} \\ 3.16228 & 2.72702 & 2.15849 & 2.66458 & 3.16228 \end{pmatrix}.$$

Example 2 (from Program LA_SBGV_EXAMPLE)

Matrices A and B as in Example 1.

Arrays **AB** and **BB** on entry:

AB

6	−5	0	0	9
5	4	4	4	0
−4	−3	−4	0	0

BB

10	8	8	8	10
0	2	2	3	0
−5	2	0	0	0

The call:

CALL LA_SBGV(AB, BB, W, 'L', Z, INFO)

W, **Z**, and **INFO** on exit:

W

-2.95028
$-2.60316_{10^{-1}}$
$3.37961_{10^{-1}}$
$8.63341_{10^{-1}}$
1.12617

Z

$-2.27298_{10^{-1}}$	$-8.01166_{10^{-2}}$	$9.84314_{10^{-2}}$	$2.70025_{10^{-1}}$	$1.13901_{10^{-1}}$
$2.29460_{10^{-1}}$	$2.49110_{10^{-1}}$	$4.58133_{10^{-2}}$	$1.43622_{10^{-1}}$	$7.10354_{10^{-2}}$
$-3.69168_{10^{-1}}$	$1.04930_{10^{-1}}$	$2.10803_{10^{-1}}$	$-2.21748_{10^{-2}}$	$1.49698_{10^{-1}}$
$1.80306_{10^{-1}}$	$-2.63376_{10^{-1}}$	$2.28500_{10^{-1}}$	$-8.68158_{10^{-2}}$	$-3.12299_{10^{-2}}$
$-9.85318_{10^{-2}}$	$1.44694_{10^{-1}}$	$2.86247_{10^{-2}}$	$9.19529_{10^{-2}}$	$-2.73340_{10^{-1}}$

INFO $= 0$

The eigenvalues of the problem $A\,x = \lambda\,B\,x$ are:

$$\begin{pmatrix} -2.95028 \\ -2.60316_{10^{-1}} \\ 3.37961_{10^{-1}} \\ 8.63341_{10^{-1}} \\ 1.12617 \end{pmatrix}.$$

The eigenvectors are:

$$\begin{pmatrix} -2.27298_{10^{-1}} & -8.01166_{10^{-2}} & 9.84314_{10^{-2}} & 2.70025_{10^{-1}} & 1.13901_{10^{-1}} \\ 2.29460_{10^{-1}} & 2.49110_{10^{-1}} & 4.58133_{10^{-2}} & 1.43622_{10^{-1}} & 7.10354_{10^{-2}} \\ -3.69168_{10^{-1}} & 1.04930_{10^{-1}} & 2.10803_{10^{-1}} & -2.21748_{10^{-2}} & 1.49698_{10^{-1}} \\ 1.80306_{10^{-1}} & -2.63376_{10^{-1}} & 2.28500_{10^{-1}} & -8.68158_{10^{-2}} & -3.12299_{10^{-2}} \\ -9.85318_{10^{-2}} & 1.44694_{10^{-1}} & 2.86247_{10^{-2}} & 9.19529_{10^{-2}} & -2.73340_{10^{-1}} \end{pmatrix}.$$

4.2 Generalized Nonsymmetric Eigenvalue Problems

4.2.1 LA_GEGS

Purpose

LA_GEGS computes for a pair of $n \times n$ real or complex matrices (A, B) the (generalized) real or complex Schur form, the generalized eigenvalues in the form of scalar pairs (α, β), and, optionally, the left and/or right Schur vectors.

If A and B are real then the real-Schur form is computed, otherwise the complex-Schur form is computed. The real-Schur form is a pair of real matrices $(\tilde{A}, \tilde{B})$ such that 1) $\tilde{A}$ has block upper triangular form, with 1×1 and 2×2 blocks along the main diagonal, 2) $\tilde{B}$ has upper triangular form with nonnegative elements on the main diagonal, and 3) $\tilde{A} = Q_1^T A Q_2$ and $\tilde{B} = Q_1^T B Q_2$, where Q_1 and Q_2 are orthogonal matrices. The 2×2 blocks of $\tilde{A}$ are "standardized" by making the corresponding elements of $\tilde{B}$ have the form

$$\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$$

The complex-Schur form is a pair of matrices $(\tilde{A}, \tilde{B})$ such that 1) $\tilde{A}$ has upper triangular form, 2) $\tilde{B}$ has upper triangular form with nonnegative elements on the main diagonal, and 3) $\tilde{A} = Q_1^H A Q_2$ and $\tilde{B} = Q_1^H B Q_2$, where Q_1 and Q_2 are unitary matrices.

In both cases the columns of Q_1 and Q_2 are called, respectively, the left and right Schur vectors.

A generalized eigenvalue of the pair (A, B) is, roughly speaking, a scalar of the form $\lambda = \alpha/\beta$ such that the matrix $A - \lambda B$ is singular. It is usually represented as the pair (α, β), as there is a reasonable interpretation of the case $\beta = 0$ (even if $\alpha = 0$).

Note: If only the generalized eigenvalues are needed, use the driver **LA_GEGV** instead.

Specification

SUBROUTINE LA_GEGS(A, B, *alpha*, **BETA, &**
VSL=vsl, **VSR**=vsr, **INFO**=info **)**
type(*wp*), **INTENT(INOUT) :: A(:,:), B(:,:)**
type(*wp*), **INTENT(OUT) ::** *alpha(:)*, **BETA(:)**
type(*wp*), **INTENT(OUT), OPTIONAL :: VSL(:,:), VSR(:,:)**
INTEGER, INTENT(OUT), OPTIONAL :: INFO
where
type ::= **REAL** | **COMPLEX**
wp ::= **KIND(1.0)** | **KIND(1.0D0)**
alpha ::= **ALPHAR, ALPHAI** | **ALPHA**
alpha(:) ::= **ALPHAR(:), ALPHAI(:)** | **ALPHA(:)**

Arguments

A – (*Input/Output*) **REAL** or **COMPLEX** square array, shape $(:,:)$.

- On entry, the matrix A.
- On exit, the matrix $\tilde{A}$.

Note: to avoid overflow, the Frobenius norm of the matrix A should be less than the overflow threshold.

B – (*Input/Output*) **REAL** or **COMPLEX** square array, shape $(:,:)$, and $size(\mathbf{B},1) = size(\mathbf{A},1)$.

- On entry, the matrix B.
- On exit, the matrix $\tilde{B}$.

Note: to avoid overflow, the Frobenius norm of the matrix B should be less than the overflow threshold.

alpha – (*Output*) **REAL** or **COMPLEX** array, shape $(:)$, and $size(alpha) = size(\mathbf{A},1)$.

- The values of α.
- *alpha*(:) ::= **ALPHAR(:), ALPHAI(:)** | **ALPHA(:)**,
  where
  ALPHAR(:), ALPHAI(:) are of **REAL** *type* (for the real and imaginary parts) and **ALPHA(:)** is of **COMPLEX** *type*.

BETA – (*Output*) **REAL** or **COMPLEX** array, shape $(:)$, and $size(\mathbf{BETA}) = size(\mathbf{A},1)$.

- The values of β.

Note: The generalized eigenvalues of the pair (A, B) are the scalars $\lambda = \alpha/\beta$.

1. If the imaginary part of $alpha_j$ is zero, then the j^{th} eigenvalue is real; if positive, then the j^{th} and $(j+1)^{st}$ eigenvalues are a complex conjugate pair, with the imaginary part of $alpha_{j+1}$ negative.

2. The quotients $alpha_j/\mathbf{BETA}_j$ may easily over- or underflow, and $\mathbf{BETA}_j$ may even be zero. Thus, the user should avoid naively computing the ratio α/β. However, *alpha* will be always less than and usually comparable with $\|A\|$ in magnitude, and **BETA** always less than and usually comparable with $\|B\|$.

VSL – *Optional* (*Output*) **REAL** or **COMPLEX** square array, shape $(:,:)$, and $size(\mathbf{VSL},1) = size(\mathbf{A},1)$.

- The left Schur vectors.

VSR – *Optional* (*Output*) **REAL** or **COMPLEX** square array, shape $(:,:)$, $size(\mathbf{VSR},1) = size(\mathbf{A},1)$.

- The right Schur vectors.

INFO – *Optional*, (*Output*) **INTEGER**.

- $= 0$: successful exit

  < 0 : if $\mathbf{INFO} = -i$, the i^{th} argument had an illegal value

  $= 1, \cdots, n$, where $n = size(\mathbf{A},1)$: The computation of the Schur form could not be completed, but $alpha_j$ and $\mathbf{BETA}_j$ should be correct for $j = \mathbf{INFO}+1, \mathbf{INFO}+2, \cdots, n$.

  $> n$: errors that usually indicate **LAPACK** problems:

  $= n+1$: error return from **LA_GGBAL**

  $= n+2$: error return from **LA_GEQRF**

  $= n+3$: error return from **LA_ORMQR**

  $= n+4$: error return from **LA_ORGQR**

  $= n+5$: error return from **LA_GGHRD**

  $= n+6$: error return from **LA_HGEQZ** (other than failed iteration)

  $= n+7$: error return from **LA_GGBAK** (computing **VSL**)

  $= n+8$: error return from **LA_GGBAK** (computing **VSR**)

  $= n+9$: error return from **LA_LASCL** (various places)

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_GEGS_EXAMPLE)

$$A = \begin{pmatrix} 0 & 2 & -1 & -1 & -1 \\ -4 & 2 & 7 & -2 & -2 \\ 4 & 1 & 8 & -8 & -5 \\ -3 & 11 & 3 & -2 & 1 \\ -1 & 5 & 1 & 5 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} -1 & 0 & 1 & -3 & -1 \\ 0 & -5 & 5 & -5 & -3 \\ -5 & 6 & -3 & -8 & 5 \\ 8 & 4 & 1 & -4 & -3 \\ 7 & 6 & -3 & 5 & 2 \end{pmatrix}$$

Arrays **A** and **B** on entry:

A

0	2	−1	−1	−1
−4	2	7	−2	−2
4	1	8	−8	−5
−3	11	3	−2	1
−1	5	1	5	4

B

−1	0	1	−3	−1
0	−5	5	−5	−3
−5	6	−3	−8	5
8	4	1	−4	−3
7	6	−3	5	2

The call:

CALL LA_GEGS(A, B, ALPHAR, ALPHAI, BETA)

A, **B**, **ALPHAR**, **ALPHAI** and **BETA** on exit:

A

5.08955	−11.6023	3.62763	$1.85041_{10^{-1}}$	−6.63232
3.96582	$-4.49878_{10^{-1}}$	−5.57311	5.96274	−3.02299
0.00000	0.00000	−3.52057	8.33637	−4.37685
0.00000	0.00000	0.00000	5.15666	2.47689
0.00000	0.00000	0.00000	0.00000	4.77975

B

3.61233	−0.00000	−3.75779	1.38528	−8.41591
0.00000	1.49775	9.44494	−4.82032	−3.98364
0.00000	0.00000	7.54035	10.2008	−2.16966
0.00000	0.00000	0.00000	7.50417	4.11258
0.00000	0.00000	0.00000	0.00000	3.83158

ALPHAR	**ALPHAI**	**BETA**
1.58394	7.96773	2.85764
1.04942	−5.27891	1.89329
−3.52057	0.00000	7.54035
5.15666	0.00000	7.50417
4.77975	0.00000	3.83158

The matrix $\tilde{A}$ in the real-Schur form of (A, B) is:

$$\begin{pmatrix} 5.08955 & -11.6023 & 3.62763 & 1.85041_{10^{-1}} & -6.63232 \\ 3.96582 & -4.49878_{10^{-1}} & -5.57311 & 5.96274 & -3.02299 \\ 0 & 0 & -3.52057 & 8.33637 & -4.37685 \\ 0 & 0 & 0 & 5.15666 & 2.47689 \\ 0 & 0 & 0 & 0 & 4.77975 \end{pmatrix}.$$

The matrix $\tilde{B}$ in the real-Schur form of (A, B) is:

$$\begin{pmatrix} 3.61233 & -0.00000 & -3.75779 & 1.38528 & -8.41591 \\ 0 & 1.49775 & 9.44494 & -4.82032 & -3.98364 \\ 0 & 0 & 7.54035 & 10.2008 & -2.16966 \\ 0 & 0 & 0 & 7.50417 & 4.11258 \\ 0 & 0 & 0 & 0 & 3.83158 \end{pmatrix}.$$

The generalized eigenvalues of $(A,\ B)$ are:

$$\begin{pmatrix} 5.54283_{10^{-1}} + 2.78823i \\ 5.54283_{10^{-1}} - 2.78822i \\ -4.66897_{10^{-1}} \\ 6.87172_{10^{-1}} \\ 1.24746 \end{pmatrix}.$$

Example 2 (from Program LA_GEGS_EXAMPLE)

$$A = \begin{pmatrix} i & -4+3i & 4+-2i & -3+7i & -1-2i \\ 2+6i & 2-2i & 1+10i & 11-3i & 5+6i \\ -1+2i & 7 & 8-i & 3+4i & 1+8i \\ -1+4i & -2+7i & -8+4i & -2+i & 5 \\ -i & -2+2i & -5i & 1+9i & 4+i \end{pmatrix},$$

$$B = \begin{pmatrix} -1 & -3i & -5+i & 8-2i & 7+3i \\ 4i & -5i & 6+7i & 4+3i & 6-2i \\ i & 5+3i & -3+3i & 1+2i & -3-5i \\ -3+9i & -5+4i & -8-6i & -4i & 5+3i \\ -1-7i & -3+4i & 5+2i & -3+i & 1+2i \end{pmatrix}$$

Arrays **A** and **B** on entry:

A

$(0,1)$	$(-4,3)$	$(4,-2)$	$(-3,7)$	$(-1,-2)$
$(2,6)$	$(2,-2)$	$(1,10)$	$(11,-3)$	$(5,6)$
$(-1,2)$	$(7,0)$	$(8,-1)$	$(3,4)$	$(1,8)$
$(-1,4)$	$(-2,7)$	$(-8,4)$	$(-2,1)$	$(5,0)$
$(0,-1)$	$(-2,2)$	$(0,-5)$	$(1,9)$	$(4,1)$

B

$(-1,0)$	$(0,-3)$	$(-5,1)$	$(8,-2)$	$(7,3)$
$(0,4)$	$(0,-5)$	$(6,7)$	$(4,3)$	$(6,-2)$
$(0,1)$	$(5,3)$	$(-3,3)$	$(1,2)$	$(-3,-5)$
$(-3,9)$	$(-5,4)$	$(-8,-6)$	$(0,-4)$	$(5,3)$
$(-1,-7)$	$(-3,4)$	$(5,2)$	$(-3,1)$	$(1,2)$

The call:

CALL LA_GEGS(A, B, ALPHA, BETA, VSL, VSR, INFO)

Arrays **A**, **B**, **ALPHA**, **BETA**, **VSL**, **VSR**, and **INFO** on exit:

A

$(-8.17153, -8.96301)$	$(5.04434, 1.57791)$	$(1.54856, 11.0471)$
$(0.00000, 0.00000)$	$(12.1798, -2.17465)$	$(-3.50080_{10^{-1}}, 1.09048)$
$(0.00000, 0.00000)$	$(0.00000, 0.00000)$	$(5.53578, -7.48624)$
$(0.00000, 0.00000)$	$(0.00000, 0.00000)$	$(0.00000, 0.00000)$
$(0.00000, 0.00000)$	$(0.00000, 0.00000)$	$(0.00000, 0.00000)$

A continued

$(-4.67583, 2.85761_{10^{-1}})$	$(3.53262_{10^{-1}}, 4.55776)$
$(5.15747_{10^{-2}}, 5.09233)$	$(1.61123, 4.19652)$
$(3.22650, 3.01791)$	$(9.51798_{10^{-1}}, 6.80497)$
$(4.59900, 1.62754)$	$(-5.66130, -7.09712)$
$(0.00000, 0.00000)$	$(-11.7012, 5.62050)$

B

$(6.11500, 0.00000)$	$(9.85319, -1.93926)$	$(-1.14575, 1.24398)$
$(0.00000, 0.00000)$	$(8.03778, 0.00000)$	$(-2.35088, -3.62784)$
$(0.00000, 0.00000)$	$(0.00000, 0.00000)$	$(16.1276, 0.00000)$
$(0.00000, 0.00000)$	$(0.00000, 0.00000)$	$(0.00000, 0.00000)$
$(0.00000, 0.00000)$	$(0.00000, 0.00000)$	$(0.00000, 0.00000)$

B continued

$(-2.23480, 4.16477_{10^{-1}})$	$(4.38479, 3.32310)$
$(-1.73601, 1.85724)$	$(1.25656, 3.78571)$
$(3.24918, 2.02950)$	$(-2.22585, -7.81959_{10^{-1}})$
$(10.7795, 0.00000)$	$(2.04957, -7.02463)$
$(0.00000, 0.00000)$	$(10.6164, 0.00000)$

ALPHA	BETA
$(-8.17153, -8.96301)$	$(6.11500, 0.00000)$
$(12.1798, -2.17465)$	$(8.03778, 0.00000)$
$(5.53578, -7.48624)$	$(16.1276, 0.00000)$
$(4.59900, 1.62754)$	$(10.7795, 0.00000)$
$(-11.7012, 5.62050)$	$(10.6164, 0.00000)$

VSL

$(-2.31376_{10^{-1}}, -2.93443_{10^{-1}})$	$(-1.40754_{10^{-1}}, 1.55212_{10^{-1}})$
$(-3.21580_{10^{-1}}, 4.92178_{10^{-1}})$	$(-4.15637_{10^{-1}}, -5.24378_{10^{-2}})$
$(1.10683_{10^{-1}}, 6.13581_{10^{-1}})$	$(-1.92758_{10^{-1}}, -1.52761_{10^{-1}})$
$(-3.30552_{10^{-2}}, -1.85065_{10^{-1}})$	$(-7.33556_{10^{-1}}, -1.55672_{10^{-1}})$
$(1.02541_{10^{-1}}, -2.83045_{10^{-1}})$	$(-3.96246_{10^{-1}}, -2.74412_{10^{-2}})$

VSL *continued*

$(-3.15548_{10^{-1}},\ 1.46823_{10^{-1}})$	$(-1.35120_{10^{-1}},\ -1.05099_{10^{-1}})$
$(-5.36435_{10^{-1}},\ 8.38246_{10^{-2}})$	$(-1.20159_{10^{-1}},\ -3.13488_{10^{-1}})$
$(1.24232_{10^{-1}},\ -4.12928_{10^{-1}})$	$(1.72548_{10^{-1}},\ 3.67860_{10^{-1}})$
$(4.37650_{10^{-1}},\ 3.33711_{10^{-1}})$	$(3.10038_{10^{-1}},\ -3.23662_{10^{-2}})$
$(-1.01167_{10^{-1}},\ -2.91556_{10^{-1}})$	$(-6.42288_{10^{-1}},\ 4.28001_{10^{-1}})$

VSL *continued*

$(-3.78747_{10^{-1}},\ -7.22893_{10^{-1}})$
$(1.78997_{10^{-1}},\ 1.98245_{10^{-1}})$
$(-2.31354_{10^{-1}},\ -3.82379_{10^{-1}})$
$(-3.17373_{10^{-2}},\ 3.52331_{10^{-2}})$
$(-7.40088_{10^{-2}},\ 2.34884_{10^{-1}})$

VSR

$(1.88445_{10^{-1}},\ -1.93223_{10^{-1}})$	$(2.79403_{10^{-1}},\ 2.77106_{10^{-1}})$
$(-2.93219_{10^{-2}},\ -4.54877_{10^{-1}})$	$(1.56626_{10^{-1}},\ 4.75309_{10^{-1}})$
$(2.57476_{10^{-1}},\ -1.28734_{10^{-1}})$	$(-5.52404_{10^{-2}},\ 3.28628_{10^{-1}})$
$(4.16922_{10^{-1}},\ 2.46221_{10^{-1}})$	$(-5.44198_{10^{-2}},\ 1.99182_{10^{-1}})$
$(-5.18184_{10^{-1}},\ -3.65445_{10^{-1}})$	$(-6.31330_{10^{-1}},\ 2.05996_{10^{-1}})$

VSR *continued*

$(4.10231_{10^{-1}},\ -1.41946_{10^{-1}})$	$(-4.15273_{10^{-1}},\ -6.39695_{10^{-1}})$
$(-2.93882_{10^{-1}},\ -4.95559_{10^{-1}})$	$(4.15164_{10^{-1}},\ 6.22410_{10^{-2}})$
$(-4.42575_{10^{-1}},\ 2.49795_{10^{-1}})$	$(-2.46834_{10^{-1}},\ 6.23311_{10^{-3}})$
$(-4.06737_{10^{-1}},\ 1.46332_{10^{-1}})$	$(-3.08148_{10^{-1}},\ 5.48922_{10^{-2}})$
$(-1.64044_{10^{-2}},\ 1.85012_{10^{-1}})$	$(-2.74666_{10^{-1}},\ -8.79069_{10^{-2}})$

VSR *continued*

$(-3.47124_{10^{-2}},\ -3.15312_{10^{-2}})$
$(3.15289_{10^{-2}},\ -1.80562_{10^{-1}})$
$(3.12882_{10^{-1}},\ 6.23662_{10^{-1}})$
$(-8.98236_{10^{-2}},\ -6.55768_{10^{-1}})$
$(-1.12438_{10^{-1}},\ -1.63134_{10^{-1}})$

INFO = 0

The matrix $\tilde{A}$ in the complex-Schur form of $(A,\, B)$ is:

$$\begin{pmatrix} -8.17153 - 8.96301i & 5.04434 + 1.57791i & 1.54856 + 11.0471i & -4.67583 + 2.85761_{10^{-1}}i & 3.53262_{10^{-1}} + 4.55776i \\ 0 & 12.1798 - 2.17465i & -3.50080_{10^{-1}} + 1.09048i & 5.15747_{10^{-2}} + 5.09233i & 1.61123 + 4.19652i \\ 0 & 0 & 5.53578 - 7.48624i & 3.22650 + 3.01791i & 9.51798_{10^{-1}} + 6.80497i \\ 0 & 0 & 0 & 4.59900 + 1.62754i & -5.66130 - 7.09712i \\ 0 & 0 & 0 & 0 & -11.7012 + 5.62050i \end{pmatrix}.$$

The matrix $\tilde{B}$ in the complex-Schur form of $(B,\, B)$ is:

$$\begin{pmatrix} 6.11500 & 9.85319 - 1.93926i & -1.14575 + 1.24398i & -2.23480 + 4.16477_{10^{-1}}\,i & 4.38479 + 3.32310i \\ 0 & 8.03778 & -2.35088 - 3.62784i & -1.73601 + 1.85724i & 1.25656 + 3.78571i \\ 0 & 0 & 16.1276 & 3.24918 + 2.02950i & -2.22585 - 7.81959_{10^{-1}}\,i \\ 0 & 0 & 0 & 10.7795 & 2.04957 - 7.02463i \\ 0 & 0 & 0 & 0 & 10.6164 \end{pmatrix}.$$

The left and right Schur vectors are:

$$\begin{pmatrix} -2.31376_{10^{-1}} - 2.93443_{10^{-1}}\,i & -1.40754_{10^{-1}} + 1.55212_{10^{-1}}\,i & -3.15548_{10^{-1}} + 1.46823_{10^{-1}}\,i & -1.35120_{10^{-1}} - 1.05099_{10^{-1}}\,i & -3.78747_{10^{-1}} - 7.22893_{10^{-1}}\,i \\ -3.21580_{10^{-1}} + 4.92178_{10^{-1}}\,i & -4.15637_{10^{-1}} - 5.24378_{10^{-2}}\,i & -5.36435_{10^{-1}} + 8.38246_{10^{-2}}\,i & -1.20159_{10^{-1}} - 3.13488_{10^{-1}}\,i & 1.78997_{10^{-1}} + 1.98245_{10^{-1}}\,i \\ 1.10683_{10^{-1}} + 6.13581_{10^{-1}}\,i & -1.92758_{10^{-1}} - 1.52761_{10^{-1}}\,i & 1.24232_{10^{-1}} - 4.12928_{10^{-1}}\,i & 1.72548_{10^{-1}} + 3.67860_{10^{-1}}\,i & -2.31354_{10^{-1}} - 3.82379_{10^{-1}}\,i \\ -3.30552_{10^{-2}} - 1.85065_{10^{-1}}\,i & -7.33556_{10^{-1}} - 1.55672_{10^{-1}}\,i & 4.37650_{10^{-1}} + 3.33711_{10^{-1}}\,i & 3.10038_{10^{-1}} - 3.23662_{10^{-2}}\,i & -3.17373_{10^{-2}} + 3.52331_{10^{-2}}\,i \\ 1.02541_{10^{-1}} - 2.83045_{10^{-1}}\,i & -3.96246_{10^{-1}} - 2.74412_{10^{-2}}\,i & -1.01167_{10^{-1}} - 2.91556_{10^{-1}}\,i & -6.42288_{10^{-1}} + 4.28001_{10^{-1}}\,i & -7.40088_{10^{-2}} + 2.34884_{10^{-1}}\,i \end{pmatrix}$$

and

$$\begin{pmatrix} 1.88445_{10^{-1}} - 1.93223_{10^{-1}}\,i & 2.79403_{10^{-1}}, 2.77106_{10^{-1}}\,i & 4.10231_{10^{-1}} - 1.41946_{10^{-1}}\,i & -4.15273_{10^{-1}} - 6.39695_{10^{-1}}\,i & -3.47124_{10^{-2}} - 3.15312_{10^{-2}}\,i \\ -2.93219_{10^{-2}} - 4.54877_{10^{-1}}\,i & 1.56626_{10^{-1}}, 4.75309_{10^{-1}}\,i & -2.93882_{10^{-1}} - 4.95559_{10^{-1}}\,i & 4.15164_{10^{-1}} + 6.22410_{10^{-2}}\,i & 3.15289_{10^{-2}} - 1.80562_{10^{-1}}\,i \\ 2.57476_{10^{-1}} - 1.28734_{10^{-1}}\,i & -5.52404_{10^{-2}}, 3.28628_{10^{-1}}\,i & -4.42575_{10^{-1}} + 2.49795_{10^{-1}}\,i & -2.46834_{10^{-1}} + 6.23311_{10^{-3}}\,i & 3.12882_{10^{-1}} + 6.23662_{10^{-1}}\,i \\ 4.16922_{10^{-1}} + 2.46221_{10^{-1}}\,i & -5.44198_{10^{-2}}, 1.99182_{10^{-1}}\,i & -4.06737_{10^{-1}} + 1.46332_{10^{-1}}\,i & -3.08148_{10^{-1}} + 5.48922_{10^{-2}}\,i & -8.98236_{10^{-2}} - 6.55768_{10^{-1}}\,i \\ -5.18184_{10^{-1}} - 3.65445_{10^{-1}}\,i & -6.31330_{10^{-1}}, 2.05996_{10^{-1}}\,i & -1.64044_{10^{-2}} + 1.85012_{10^{-1}}\,i & -2.74666_{10^{-1}} - 8.79069_{10^{-2}}\,i & -1.12438_{10^{-1}} - 1.63134_{10^{-1}}\,i \end{pmatrix}.$$

The generalized eigenvalues are:

$$\begin{pmatrix} -1.33631 & -\,1.46574i \\ 1.51532 & -\,2.70554_{10^{-1}}\,i \\ 3.43248_{10^{-1}} & -\,4.64187_{10^{-1}}\,i \\ 4.26642_{10^{-1}} & +\,1.50984_{10^{-1}}\,i \\ -1.10218 & +\,5.29416_{10^{-1}}\,i \end{pmatrix}.$$

4.2.2 LA_GEGV

Purpose

LA_GEGV computes for a real or complex matrix pair $(A,\ B)$ the generalized eigenvalues in the form of scalar pairs $(\alpha,\ \beta)$, and, optionally, the left and/or right generalized eigenvectors.

A generalized eigenvalue of the pair (A, B) is, roughly speaking, a scalar of the form $\lambda = \alpha/\beta$ such that the matrix $A - \lambda B$ is singular. It is usually represented as the pair (α, β), as there is a reasonable interpretation of the case $\beta = 0$ (even if $\alpha = 0$).

A right generalized eigenvector corresponding to a generalized eigenvalue λ is a vector x such that $(A - \lambda\, B)\, x = 0$. A left generalized eigenvector is a vector y such that $y^H\, (A - \lambda\, B) = 0$, where y^H is the conjugate-transpose of y.

The computation is based on the real or complex Schur form of $(A,\ B)$. See **LA_GEGS** for details of this form.

Note: This routine performs "full balancing" on A and B.

Specification

SUBROUTINE LA_GEGV(A, B, *alpha*, **BETA, &**
VL=vl, **VR**=vr, **INFO**=info)
INTEGER, INTENT(OUT), OPTIONAL :: INFO
type(*wp*), **INTENT(INOUT) :: A(:,:), B(:,:)**
type(*wp*), **INTENT(OUT) ::** *alpha(:)*, **BETA(:)**
type(*wp*), **INTENT(OUT), OPTIONAL :: VL(:,:), VR(:,:)**
where
type **::= REAL | COMPLEX**
wp **::= KIND(1.0) | KIND(1.0D0)**
alpha **::= ALPHAR, ALPHAI | ALPHA**
alpha(:) **::= ALPHAR(:), ALPHAI(:) | ALPHA(:)**

Arguments

A – (*Input/Output*) **REAL** or **COMPLEX** square array, shape $(:, :)$.

- On entry, the matrix A.
- On exit, the matrix $\tilde{A}$ in the Schur form $(\tilde{A},\ \tilde{B})$ if eigenvectors are computed, otherwise just the diagonal blocks of $\tilde{A}$.

B – (*Input/Output*) **REAL** or **COMPLEX** square array, shape $(:, :)$, and $size(\mathbf{B}, 1) = size(\mathbf{A}, 1)$.

- On entry, the matrix B.
- On exit, the matrix $\tilde{B}$ in the Schur form (A, B) if eigenvectors are computed, otherwise just the diagonal blocks of $\tilde{B}$.

alpha – (*Output*) **REAL** or **COMPLEX** array, shape (:), and $size(alpha) = size(\mathbf{A}, 1)$.

- The values of α.
- *alpha*(:) ::= **ALPHAR(:), ALPHAI(:)** | **ALPHA(:)**, where **ALPHAR(:), ALPHAI(:)** are of **REAL** *type* (for the real and imaginary parts) and **ALPHA(:)** is of **COMPLEX** *type*.

BETA – (*Output*) **REAL** or **COMPLEX** array, shape (:), and $size(\mathbf{BETA}) = size(\mathbf{A}, 1)$.

- The values of β.

Note: The generalized eigenvalues of the pair (A, B) are the scalars $\lambda = \alpha/\beta$.

1. If the imaginary part of $alpha_j$ is zero, then the j^{th} eigenvalue is real; if positive, then the j^{th} and $(j+1)^{st}$ eigenvalues are a complex conjugate pair, with the imaginary part of $alpha_{j+1}$ negative.
2. The quotients $alpha_j/\mathbf{BETA}_j$ may easily over- or underflow, and $\mathbf{BETA}_j$ may even be zero. Thus, the user should avoid naively computing the ratio α/β. However, *alpha* will be always less than and usually comparable with $\|A\|$ in magnitude, and **BETA** always less than and usually comparable with $\|B\|$.

VL – *Optional* (*Output*) **REAL / COMPLEX** array, shape (:,:), $size(\mathbf{VL}, 1) = size(\mathbf{VL}, 2) = size(\mathbf{A}, 1)$.

- The left generalized eigenvectors. Real eigenvectors take one column, complex take two columns, the first for the real part and the second for the imaginary part. Complex eigenvectors correspond to an eigenvalue with positive imaginary part. Each eigenvector will be scaled so the largest component will have $\mid realpart \mid + \mid imag.part \mid = 1$, except that for eigenvalues with $\alpha = \beta = 0$, a zero vector will be returned as the corresponding eigenvector.

VR – *Optional* (*Output*) **REAL / COMPLEX** array, shape (:,:), $size(\mathbf{VR}, 1) = size(\mathbf{VR}, 2) = size(\mathbf{A}, 1)$.

- The right generalized eigenvectors. Real eigenvectors take one column, complex take two columns, the first for the real part and the second for the imaginary part. Complex eigenvectors correspond to an eigenvalue with positive imaginary part. Each eigenvector will be scaled so the largest component will have $\mid realpart \mid + \mid imag.part \mid = 1$, except that for eigenvalues with $\alpha = \beta = 0$, a zero vector will be returned as the corresponding eigenvector.

INFO – *Optional*, (*Output*) **INTEGER**.

- $= 0$: successful exit
  < 0 : if $\mathbf{INFO} = -i$, the i^{th} argument had an illegal value

$= 1, \cdots, n$, where $n = size(\mathbf{A}, 1)$: The iteration failed. No eigenvectors have been calculated, but $\mathbf{ALPHAR}_j$, $\mathbf{ALPHAI}_j$, and $\mathbf{BETA}_j$ should be correct for $j = \mathbf{INFO} + 1, \mathbf{INFO} + 2, \cdots, n$.

$> n$: errors that usually indicate **LAPACK** problems:

- $= n + 1$: error return from **LA_GGBAL**
- $= n + 2$: error return from **LA_GEQRF**
- $= n + 3$: error return from **LA_ORMQR**
- $= n + 4$: error return from **LA_ORGQR**
- $= n + 5$: error return from **LA_GGHRD**
- $= n + 6$: error return from **LA_HGEQZ** (other than failed iteration)
- $= n + 7$: error return from **LA_TGEVC**
- $= n + 8$: error return from **LA_GGBAK** (computing **VL**)
- $= n + 9$: error return from **LA_GGBAK** (computing **VR**)
- $= n + 10$: error return from **LA_LASCL** (various calls)

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_GEGV_EXAMPLE)

$$A = \begin{pmatrix} 3+i & 6+i & 2+4i & 1-i & -8 \\ 3+2i & -3i & -5+2i & 4+i & 3+2i \\ -3+i & -8-i & -2-i & 5+i & -3-2i \\ 5-2i & -6-2i & 7+i & 1-2i & 3-i \\ -11 & 2+3i & -2 & -8-i & -6 \end{pmatrix},$$

$$B = \begin{pmatrix} 5 & 1+i & -5 & 2+3i & -1 \\ 9+2i & -7-i & -3 & 8+3i & -3+i \\ 8-3i & -2+i & 3 & -5 & 7-3i \\ 5 & -i & -5 & 4-i & 5+4i \\ 2 & 2-2i & -3-3i & -3+i & -10+i \end{pmatrix}$$

Arrays **A** and **B** on entry:

A

$(3,1)$	$(6,1)$	$(2,4)$	$(1,-1)$	$(-8,0)$
$(3,2)$	$(0,-3)$	$(-5,2)$	$(4,1)$	$(3,2)$
$(-3,1)$	$(-8,-1)$	$(-2,-1)$	$(5,1)$	$(-3,-2)$
$(5,-2)$	$(-6,-2)$	$(7,1)$	$(1,-2)$	$(3,-1)$
$(-11,0)$	$(2,3)$	$(-2,0)$	$(-8,-1)$	$(-6,0)$

B

$(5,0)$	$(1,1)$	$(-5,0)$	$(2,3)$	$(-1,0)$
$(9,2)$	$(-7,-1)$	$(-3,0)$	$(8,3)$	$(-3,1)$
$(8,-3)$	$(-2,1)$	$(3,0)$	$(-5,0)$	$(7,-3)$
$(5,0)$	$(0,-1)$	$(-5,0)$	$(4,-1)$	$(5,4)$
$(2,0)$	$(2,-2)$	$(-3,-3)$	$(-3,1)$	$(-10,1)$

The call:

CALL LA_GEGV(A, B, ALPHA, BETA)

ALPHA and **BETA** on exit:

ALPHA	BETA
$(5.38188, -10.1568)$	$(4.74767, 0.00000)$
$(9.76004, -2.37831)$	$(7.80194, 0.00000)$
$(-8.40953, 4.70817)$	$(8.08745, 0.00000)$
$(2.18308, 6.14928_{10^{-2}})$	$(14.9588, 0.00000)$
$(-6.11817, -4.76947)$	$(11.0013, 0.00000)$

The generalized eigenvalues of the matrix pair (A, B) are:

$$\begin{pmatrix} 1.13358 & -\,2.13931i \\ 1.25098 & -\,3.04836_{10^{-1}}\,i \\ -1.03982 & +\,5.82158_{10^{-1}}\,i \\ 1.45940_{10^{-1}} & +\,4.11082_{10^{-3}}\,i \\ -5.56130_{10^{-1}} & -\,4.33536_{10^{-1}}\,i \end{pmatrix}.$$

Example 2 (from Program LA_GEGV_EXAMPLE)

Matrices A and B as in Example 1.

Arrays **A** and **B** on entry: as in Example 1.

The call:

CALL LA_GEGV(A, B, ALPHA, BETA, VL, VR, INFO)

ALPHA and **BETA** on exit: as in Example 1.

VL, **VR**, and **INFO** on exit:

VL

($2.42481_{10^{-1}}$, $-7.57520_{10^{-1}}$)	($2.28589_{10^{-1}}$, $-7.14509_{10^{-1}}$)
($7.11078_{10^{-2}}$, $8.00779_{10^{-2}}$)	($5.26561_{10^{-1}}$, $4.73439_{10^{-1}}$)
($-1.48777_{10^{-1}}$, $3.35853_{10^{-2}}$)	($-9.96893_{10^{-2}}$, $1.87584_{10^{-1}}$)
($-1.27857_{10^{-1}}$, $3.36957_{10^{-1}}$)	($1.42497_{10^{-1}}$, $6.68439_{10^{-1}}$)
($-2.70491_{10^{-1}}$, $1.48541_{10^{-1}}$)	($-3.22790_{10^{-1}}$, $-4.26320_{10^{-1}}$)

VL *continued*

($-7.81628_{10^{-2}}$, $7.57742_{10^{-2}}$)	($1.71564_{10^{-1}}$, $1.41906_{10^{-1}}$)
($3.69128_{10^{-1}}$, $3.30618_{10^{-1}}$)	($5.09198_{10^{-1}}$, $4.90802_{10^{-1}}$)
($2.08501_{10^{-1}}$, $-5.30135_{10^{-1}}$)	($3.58223_{10^{-2}}$, $-0.18843_{10^{-2}}$)
($-9.63995_{10^{-1}}$, $-3.60048_{10^{-2}}$)	($-1.37470_{10^{-2}}$, $4.91049_{10^{-2}}$)
($-1.61771_{10^{-1}}$, $-4.47743_{10^{-2}}$)	($8.54211_{10^{-2}}$, $2.02605_{10^{-1}}$)

VL *continued*

($-4.19396_{10^{-1}}$, $-4.08450_{10^{-1}}$)
($2.85816_{10^{-1}}$, $3.96209_{10^{-1}}$)
($-9.14546_{10^{-1}}$, $-8.54544_{10^{-2}}$)
($7.05706_{10^{-1}}$, $-6.60841_{10^{-2}}$)
($3.97135_{10^{-1}}$, $1.15260_{10^{-1}}$)

VR

($-2.41334_{10^{-1}}$, $-7.97233_{10^{-2}}$)	($5.53201_{10^{-2}}$, $-4.09357_{10^{-2}}$)
($-4.55653_{10^{-1}}$, $-5.44347_{10^{-1}}$)	($5.89935_{10^{-2}}$, $2.76644_{10^{-1}}$)
($-3.89841_{10^{-1}}$, $-2.09902_{10^{-1}}$)	($4.20171_{10^{-2}}$, $5.08865_{10^{-1}}$)
($-3.67653_{10^{-1}}$, $-4.25132_{10^{-1}}$)	($3.22591_{10^{-1}}$, $6.77409_{10^{-1}}$)
($-5.47522_{10^{-2}}$, $-8.42053_{10^{-2}}$)	($2.08910_{10^{-1}}$, $4.10870_{10^{-1}}$)

VR*continued*

($1.80334_{10^{-1}}$, $-5.23297_{10^{-1}}$)	($2.53231_{10^{-1}}$, $6.25690_{10^{-1}}$)
($-5.57299_{10^{-1}}$, $-2.18461_{10^{-1}}$)	($-3.63023_{10^{-1}}$, $-3.72963_{10^{-1}}$)
($7.22664_{10^{-1}}$, $-2.77336_{10^{-1}}$)	($-2.20737_{10^{-1}}$, $-5.50388_{10^{-1}}$)
($-1.13110_{10^{-1}}$, $2.14123_{10^{-1}}$)	($-2.24819_{10^{-1}}$, $-7.75181_{10^{-1}}$)
($-5.04181_{10^{-1}}$, $-6.63034_{10^{-2}}$)	($-1.93321_{10^{-1}}$, $-4.39131_{10^{-1}}$)

VR*continued*

($-3.08733_{10^{-1}}$, $-2.46359_{10^{-1}}$)
($8.51386_{10^{-2}}$, $-2.38417_{10^{-1}}$)
($-4.95046_{10^{-2}}$, $3.62534_{10^{-1}}$)
($6.08798_{10^{-1}}$, $3.91202_{10^{-1}}$)
($-1.02984_{10^{-1}}$, $-3.23649_{10^{-1}}$)

INFO $= 0$

The generalized eigenvalues of (A, B) are as in Example 1. The left and right generalized eigenvectors are:

$$\left(\begin{array}{rr}
2.42481_{10^{-1}} - 7.57520_{10^{-1}}\,i & 2.28589_{10^{-1}} - 7.14509_{10^{-1}}\,i \\
7.11078_{10^{-2}} + 8.00779_{10^{-2}}\,i & 5.26561_{10^{-1}} + 4.73439_{10^{-1}}\,i \\
-1.48777_{10^{-1}} + 3.35853_{10^{-2}}\,i & -9.96893_{10^{-2}} + 1.87584_{10^{-1}}\,i \\
-1.27857_{10^{-1}} + 3.36957_{10^{-1}}\,i & 1.42497_{10^{-1}} + 6.68439_{10^{-1}}\,i \\
-2.70491_{10^{-1}} + 1.48541_{10^{-1}}\,i & -3.22790_{10^{-1}} - 4.26320_{10^{-1}}\,i
\end{array}\right.$$

$$
\begin{array}{ll}
-7.81628_{10^{-2}} + 7.57742_{10^{-2}}\,i & 1.71564_{10^{-1}} + 1.41906_{10^{-1}}\,i \\
3.69128_{10^{-1}} + 3.30618_{10^{-1}}\,i & 5.09198_{10^{-1}} + 4.90802_{10^{-1}}\,i \\
2.08501_{10^{-1}} - 5.30135_{10^{-1}}\,i & 3.58223_{10^{-2}} - 0.18843_{10^{-2}}\,i \\
-9.63995_{10^{-1}} - 3.60048_{10^{-2}}\,i & -1.37470_{10^{-2}} + 4.91049_{10^{-2}}\,i \\
-1.61771_{10^{-1}} - 4.47743_{10^{-2}}\,i & 8.54211_{10^{-2}} + 2.02605_{10^{-1}}\,i
\end{array}
$$

$$
\left.
\begin{array}{l}
-4.19396_{10^{-1}} - 4.08450_{10^{-1}}\,i \\
2.85816_{10^{-1}} + 3.96209_{10^{-1}}\,i \\
-9.14546_{10^{-1}} - 8.54544_{10^{-2}}\,i \\
7.05706_{10^{-1}} - 6.60841_{10^{-2}}\,i \\
3.97135_{10^{-1}} + 1.15260_{10^{-1}}\,i
\end{array}
\right)
$$

and

$$
\left(
\begin{array}{ll}
-2.41334_{10^{-1}} - 7.97233_{10^{-2}}\,i & 5.53201_{10^{-2}} - 4.09357_{10^{-2}}\,i \\
-4.55653_{10^{-1}} - 5.44347_{10^{-1}}\,i & 5.89935_{10^{-2}} + 2.76644_{10^{-1}}\,i \\
-3.89841_{10^{-1}} - 2.09902_{10^{-1}}\,i & 4.20171_{10^{-2}} + 5.08865_{10^{-1}}\,i \\
-3.67653_{10^{-1}} - 4.25132_{10^{-1}}\,i & 3.22591_{10^{-1}} + 6.77409_{10^{-1}}\,i \\
-5.47522_{10^{-2}} - 8.42053_{10^{-2}}\,i & 2.08910_{10^{-1}} + 4.10870_{10^{-1}}\,i
\end{array}
\right.
$$

$$
\begin{array}{ll}
1.80334_{10^{-1}} - 5.23297_{10^{-1}}\,i & 2.53231_{10^{-1}} + 6.25690_{10^{-1}}\,i \\
-5.57299_{10^{-1}} - 2.18461_{10^{-1}}\,i & -3.63023_{10^{-1}} - 3.72963_{10^{-1}}\,i \\
7.22664_{10^{-1}} - 2.77336_{10^{-1}}\,i & -2.20737_{10^{-1}} - 5.50388_{10^{-1}}\,i \\
-1.13110_{10^{-1}} + 2.14123_{10^{-1}}\,i & -2.24819_{10^{-1}} - 7.75181_{10^{-1}}\,i \\
-5.04181_{10^{-1}} - 6.63034_{10^{-2}}\,i & -1.93321_{10^{-1}} - 4.39131_{10^{-1}}\,i
\end{array}
$$

$$
\left.
\begin{array}{l}
-3.08733_{10^{-1}} - 2.46359_{10^{-1}}\,i \\
8.51386_{10^{-2}} - 2.38417_{10^{-1}}\,i \\
-4.95046_{10^{-2}} + 3.62534_{10^{-1}}\,i \\
6.08798_{10^{-1}} + 3.91202_{10^{-1}}\,i \\
-1.02984_{10^{-1}} - 3.23649_{10^{-1}}\,i
\end{array}
\right)
$$

Chapter 5

Driver Routines for Singular Value Problems

5.1 Standard Singular Value Problems

5.1.1 LA_GESVD

Purpose

LA_GESVD computes the singular value decomposition (SVD) of a real or complex $m \times n$ matrix A and, optionally, the left and/or right singular vectors. The SVD of A is written

$$A = U\,\Sigma\,V^T \quad (A = U\,\Sigma\,V^H)$$

where Σ is an $m \times n$ matrix which is zero except for its $\min(m, n)$ diagonal elements, U is an $m \times m$ orthogonal (unitary) matrix, and V is an $n \times n$ orthogonal (unitary) matrix. The diagonal elements of Σ, i.e., the values

$$\sigma_i \equiv \Sigma_{i,i}, \quad i = 1, 2, \ldots, \min(m, n)$$

are the singular values of A; they are real and non-negative, and are returned in descending order. The first $\min(m, n)$ columns of U and V are the left and right singular vectors of A, respectively.

For brevity, we hereafter use the conjugate-transpose notation "H" for both versions of the SVD.

Note that the routine returns V^H, not V.

Specification

SUBROUTINE LA_GESVD(A, S, U=u, **VT**=vt, **WW**=ww, **&**
JOB=job, **INFO**=info)
type(*wp*), **INTENT(INOUT) :: A(:,:)**
REAL(*wp*), **INTENT(OUT) :: S(:)**
type(*wp*), **INTENT(OUT), OPTIONAL :: U(:,:), VT(:,:)**
REAL(*wp*), **INTENT(OUT), OPTIONAL :: WW(:)**
CHARACTER(LEN=1), OPTIONAL, INTENT(IN) :: JOB
INTEGER, INTENT(OUT), OPTIONAL :: INFO
where
type **::= REAL | COMPLEX**
wp **::= KIND(1.0) | KIND(1.0D0)**

Arguments

A – (*Input/Output*) **REAL / COMPLEX** array, shape $(:,:)$, $size(A,1) = m$, $size(A,2) = n$.

- On entry, the $m \times n$ matrix A.
- On exit,
  - If **U** is not present and **JOB** = 'U', then **A** is overwritten with the first $\min(m,n)$ columns of U (the left singular vectors, stored columnwise).
  - If **VT** is not present and **JOB** = 'V', then **A** is overwritten with the first $\min(m,n)$ rows of V^H (the right singular vectors, stored rowwise).
  - In all cases the original contents of **A** are destroyed.

S – (*Output*) **REAL** array, shape $(:)$, $size(\mathbf{S}) = \min(m,n)$.

- The singular values of A, sorted so that $\mathbf{S}_i \geq \mathbf{S}_{i+1}$.

U – *Optional* (*Output*) **REAL / COMPLEX** array, shape $(:,:)$, $size(\mathbf{U},1) = m$, $size(\mathbf{U},2) = m$ or $\min(m,n)$.

- If $size(\mathbf{U},2) = m$, **U** contains the $m \times m$ matrix U;
- if $size(\mathbf{U},2) = \min(m,n)$, **U** contains the first $\min(m,n)$ columns of U (the left singular vectors, stored columnwise);

VT – *Optional* (*Output*) **REAL / COMPLEX** array, shape $(:,:)$, $size(\mathbf{VT},1) = n$ or $\min(m,n)$, $size(\mathbf{VT},2) = n$.

- If $size(\mathbf{VT},1) = n$, **VT** contains the $n \times n$ matrix V^H;
- if $size(\mathbf{VT},1) = \min(m,n)$, **VT** contains the first $\min(m,n)$ rows of V^H (the right singular vectors, stored rowwise);

WW – *Optional* (*Output*) **REAL** array, shape $(:)$, $size(\mathbf{WW}) = \min(m,n) - 1$

- If **INFO** > 0, **WW** contains the unconverged superdiagonal elements of an upper bidiagonal matrix B whose diagonal is in Σ (not necessarily sorted). B has the same singular values as A.

JOB – *Optional* (*Input*) **CHARACTER(LEN=1)**.

- = 'U': and **U** is not present, the first $\min(m, n)$ columns of U (the left singular vectors) are returned in array **A**;
  = 'V': and **VT** is not present, the first $\min(m, n)$ rows of V^H (the right singular vectors) are returned in array **A**;
  = 'N': neither columns of U nor rows of V^H are stored in array **A**.

If **JOB** is not present, **JOB** = 'N' is assumed.

INFO – *Optional*, (*Output*) **INTEGER**.

- $= 0$: successful exit
  < 0 : if **INFO** $= -i$, the i^{th} argument had an illegal value
  > 0 : **INFO** specifies how many superdiagonals of an intermediate bidiagonal form B did not converge to zero.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_GESVD_EXAMPLE)

$$A = \begin{pmatrix} -2 & 1 & -5 & 11 & 2 \\ 2 & -2 & 0 & 1 & 4 \\ 6 & -8 & 0 & 6 & 0 \end{pmatrix}$$

Array **A** on entry:

A

−2	1	−5	11	2
2	−2	0	1	4
6	−8	0	6	0

The call:

CALL LA_GESVD(A, S)

S on exit:

S

14.0828
10.1124
3.92600

The singular values of matrix A are:

$$\begin{pmatrix} 14.0828 \\ 10.1124 \\ 3.92600 \end{pmatrix}.$$

Example 2 (from Program LA_GESVD_EXAMPLE)

Matrix A as in Example 1.

The call:

**CALL LA_GESVD(A, S, VT=VT, WW=WW, JOB='U', &
INFO=INFO)**

S on exit: as in Example 1.

A, **VT**, **WW**, and **INFO** on exit:

A

$-7.40942_{10^{-1}}$	$6.71569_{10^{-1}}$	$2.36981_{10^{-4}}$	$-8.67066_{10^{-1}}$	$-1.57023_{10^{-1}}$
$-1.82360_{10^{-1}}$	$-2.00858_{10^{-1}}$	$-9.62497_{10^{-1}}$	$2.87490_{10^{-1}}$	$5.33710_{10^{-2}}$
$-6.46336_{10^{-1}}$	$-7.13198_{10^{-1}}$	$2.71291_{10^{-1}}$	$1.70111_{10^{-1}}$	$-9.80518_{10^{-1}}$

VT

$-1.96044_{10^{-1}}$	$3.40447_{10^{-1}}$	$2.63066_{10^{-1}}$	$-8.67066_{10^{-1}}$	$-1.57023_{10^{-1}}$
$-5.95709_{10^{-1}}$	$6.70353_{10^{-1}}$	$-3.32052_{10^{-1}}$	$2.87490_{10^{-1}}$	$5.33710_{10^{-2}}$
$-7.58338_{10^{-2}}$	$-6.24287_{10^{-2}}$	$-3.01914_{10^{-4}}$	$1.70111_{10^{-1}}$	$-9.80517_{10^{-1}}$
$4.27700_{10^{-1}}$	$5.59256_{10^{-1}}$	$6.34830_{10^{-1}}$	$3.17975_{10^{-1}}$	$-1.37157_{10^{-2}}$
$6.46544_{10^{-1}}$	$3.43595_{10^{-1}}$	$-6.46165_{10^{-1}}$	$-1.88417_{10^{-1}}$	$-1.04370_{10^{-1}}$

WW

0.00000
0.00000

INFO = 0

The singular values of A: as in Example 1.

The left singular vectors of A are:

$$\begin{pmatrix} -7.40942_{10^{-1}} & 6.71569_{10^{-1}} & 2.36981_{10^{-4}} \\ -1.82360_{10^{-1}} & -2.00858_{10^{-1}} & -9.62497_{10^{-1}} \\ -6.46336_{10^{-1}} & -7.13198_{10^{-1}} & 2.71291_{10^{-1}} \end{pmatrix}.$$

The right singular vectors of A are (columnwise):

$$\begin{pmatrix} -1.96044_{10^{-1}} & -5.95709_{10^{-1}} & -7.58338_{10^{-2}} \\ 3.40447_{10^{-1}} & 6.70353_{10^{-1}} & -6.24287_{10^{-2}} \\ 2.63066_{10^{-1}} & -3.32052_{10^{-1}} & -3.01914_{10^{-4}} \\ -8.67066_{10^{-1}} & 2.87490_{10^{-1}} & 1.70111_{10^{-1}} \\ -1.57023_{10^{-1}} & 5.33710_{10^{-2}} & -9.80517_{10^{-1}} \end{pmatrix}.$$

5.2 Generalized Singular Value Problems

5.2.1 LA_GGSVD

Purpose

LA_GGSVD computes the generalized singular value decomposition ($GSVD$) of a real or complex matrix pair (A, B), where A is $m \times n$ and B is $p \times n$. In the real case the $GSVD$ is of the form

$$U^T A Q = \Sigma_1 \,(0 \quad R), \qquad\qquad V^T B Q = \Sigma_2 \,(0 \quad R)$$

where U, V and Q are orthogonal, and in the complex case it is given by

$$U^H A Q = \Sigma_1 \,(0 \quad R), \qquad\qquad V^H B Q = \Sigma_2 \,(0 \quad R)$$

where U, V and Q are unitary. For brevity, we hereafter use the conjugate-transpose notation "H" for both cases.

Let l be the rank of B and r the rank of the $(m+p) \times n$ matrix $\begin{pmatrix} A \\ B \end{pmatrix}$. Let $k = r - l$. Then R is a $(k+l) \times (k+l)$ nonsingular upper triangular matrix, Σ_1 and Σ_2 are $m \times (k+l)$ and $p \times (k+l)$ "diagonal" matrices, respectively, with the following structures:

If $m - k - l \geq 0$,

$$\Sigma_1 \;=\; \begin{array}{c c} & \begin{array}{cc} k & l \end{array} \\ \begin{array}{c} k \\ l \\ m-k-l \end{array} & \begin{pmatrix} I & 0 \\ 0 & C \\ 0 & 0 \end{pmatrix} \end{array}$$

$$\Sigma_2 \;=\; \begin{array}{c c} & \begin{array}{cc} k & l \end{array} \\ \begin{array}{c} l \\ p-l \end{array} & \begin{pmatrix} 0 & S \\ 0 & 0 \end{pmatrix} \end{array}$$

$$(0 \quad R) \;=\; \begin{array}{c c} & \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{array}{c} k \\ l \end{array} & \begin{pmatrix} 0 & R_{11} & R_{12} \\ 0 & 0 & R_{22} \end{pmatrix} \end{array}$$

where $C^2 + S^2 = I$. R is stored in $\mathbf{A}_{1:k+l,n-k-l+1:n}$ on exit. Let

$$\alpha_1 = \alpha_2 = \cdots = \alpha_k = 1, \; \alpha_{k+i} = c_{ii}, \quad i = 1, 2, \ldots, l$$

$$\beta_1 = \beta_2 = \cdots = \beta_k = 0, \; \beta_{k+i} = s_{ii}, \quad i = 1, 2, \ldots, l$$

If $m - k - l < 0$,

$$\Sigma_1 \;=\; \begin{array}{c c} & \begin{array}{ccc} k & m-k & k+l-m \end{array} \\ \begin{array}{c} k \\ m-k \end{array} & \begin{pmatrix} I & 0 & 0 \\ 0 & C & 0 \end{pmatrix} \end{array}$$

$$\Sigma_2 = \begin{array}{c} \\ m-k \\ k+l-m \\ p-l \end{array} \begin{array}{c} \begin{array}{ccc} k & m-k & k+l-m \end{array} \\ \begin{pmatrix} 0 & S & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$(\,0 \quad R\,) = \begin{array}{c} \\ k \\ m-k \\ k+l-m \end{array} \begin{array}{c} \begin{array}{cccc} n-k-l & k & m-k & k+l-m \end{array} \\ \begin{pmatrix} 0 & R_{11} & R_{12} & R_{13} \\ 0 & 0 & R_{22} & R_{23} \\ 0 & 0 & 0 & R_{33} \end{pmatrix} \end{array}$$

where $C^2 + S^2 = I$. The matrix

$$\begin{pmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \end{pmatrix}$$

is stored in $\mathbf{A}_{1:m,n-k-l+1:n}$ and R_{33} is stored in $\mathbf{B}_{m-k+1:l,n+m-k-l+1:n}$ on exit. Let

$$\alpha_1 = \alpha_2 = \cdots = \alpha_k = 1,\ \alpha_{k+i} = c_{ii}, \quad i = 1, 2, \ldots, m{-}k, \quad \alpha_{m+1} = \alpha_{m+2} = \cdots = \alpha_{k+l} = 0$$

$$\beta_1 = \beta_2 = \cdots = \beta_k = 0,\ \beta_{k+i} = s_{ii}, \quad i = 1, 2, \ldots, m{-}k, \quad \beta_{m+1} = \beta_{m+2} = \cdots = \beta_{k+l} = 1$$

The generalized singular values of the pair $(A,\, B)$ are the ratios

$$\sigma_i = \alpha_i / \beta_i, \quad i = 1, 2, \ldots, k + l$$

Note that first k singular values are infinite. The finite singular values are real and nonnegative.

LA_GGSVD computes R, the real nonnegative scalars $\alpha_i, \beta_i, \quad i = 1, 2, \ldots, k+l$ and optionally, the transformation matrices U, V and Q. If B is an $n \times n$ nonsingular matrix, then the $GSVD$ of A and B implicitly gives the SVD of $A\,B^{-1}$:

$$A\,B^{-1} \;=\; U\,(\Sigma_1\,\Sigma_2^{-1})\,V^H.$$

If $\begin{pmatrix} A \\ B \end{pmatrix}$ has orthonormal columns, then the $GSVD$ of A and B is also equal to the CS (Cosine-Sine) decomposition of $(A,\, B)$. Furthermore, the $GSVD$ can be used to derive the solution of the eigenvalue problem:

$$A^H A x \;=\; \lambda\, B^H B x.$$

In some of the literature, the $GSVD$ of $(A,\, B)$ is presented in the form

$$U^H A X \;=\; (\,0 \quad \Sigma_1\,), \qquad\qquad V^H B X \;=\; (\,0 \quad \Sigma_2\,)$$

where U and V are orthogonal/unitary, X is nonsingular, and Σ_1 and Σ_2 are "diagonal". The former $GSVD$ form can be converted to the latter form by taking the nonsingular matrix X as

$$X \;=\; Q \begin{pmatrix} I & 0 \\ 0 & R^{-1} \end{pmatrix}.$$

Specification

SUBROUTINE LA_GGSVD(A, B, ALPHA, BETA, K, L, &
U=u, **V**=v, **Q**=q, **INFO**=info)
type(*wp*), **INTENT(INOUT) :: A(:,:), B(:,:)**
REAL(*wp*), INTENT(OUT) :: ALPHA(:), BETA(:)
INTEGER, INTENT(OUT) :: K, L
type(*wp*), **INTENT(OUT), OPTIONAL :: U(:,:), V(:,:), Q(:,:)**
INTEGER, INTENT(OUT), OPTIONAL :: INFO
type **::= REAL | COMPLEX**
wp **::= KIND(1.0) | KIND(1.0D0)**

Arguments

A – (*Input/Output*) **REAL / COMPLEX** array, shape (:, :), $size(\mathbf{A},1) = m$, $size(\mathbf{A},2) = n$.

- On entry, the $m \times n$ matrix A.
- On exit, A contains the triangular matrix R, or part of R. (See "Purpose" for details).

B – (*Input/Output*) **REAL / COMPLEX** array, shape (:, :), $size(\mathbf{B},1) = p$, $size(\mathbf{B},2) = n$.

- On entry, the $p \times n$ matrix B.
- On exit, B contains the triangular matrix R_{33} if $m - \mathbf{K} - \mathbf{L} < 0$. (See "Purpose" for details).

ALPHA, BETA – (*Output*) **REAL** array, shape (:), $size(\mathbf{ALPHA}) = size(\mathbf{BETA}) = n$.

- The generalized singular values of the pair (A, B) are

$$\sigma_i = \mathbf{ALPHA}_i/\mathbf{BETA}_i, \quad i = 1, 2, \ldots, \mathbf{K}+\mathbf{L}.$$

  If $\mathbf{K}+\mathbf{L} < n$ then $\mathbf{ALPHA}_{\mathbf{K}+\mathbf{L}+1:n} = \mathbf{BETA}_{\mathbf{K}+\mathbf{L}+1:n} = 0$.

K, L – (*Output*) **INTEGER**.

- The dimension parameters k and l described in "Purpose". $\mathbf{K}+\mathbf{L}$ is the effective numerical rank of the matrix $\begin{pmatrix} A \\ B \end{pmatrix}$.

U – *Optional* (*Output*) **REAL / COMPLEX** array, shape (:, :), $size(\mathbf{U},1) = size(\mathbf{U},2) = m$.

- The $m \times m$ matrix U.

V – *Optional* (*Output*) **REAL / COMPLEX** array, shape (:, :), $size(\mathbf{V},1) = size(\mathbf{V},2) = p$.

- The $p \times p$ matrix V.

Q – *Optional* (*Output*) **REAL / COMPLEX** array, shape (:, :), $size(\mathbf{Q},1) = size(\mathbf{Q},2) = n$.

- The $n \times n$ matrix Q.

INFO – *Optional*, (*Output*) **INTEGER**.

- $= 0$: successful exit

  < 0 : if **INFO** $= -i$, the i^{th} argument had an illegal value

  > 0 : if **INFO** $= 1$, the iteration failed to converge. For further details, see subroutine **LA_TGSJA**.

If **INFO** is not present and an error occurs, then the program is terminated with an error message.

Internal Parameters

TOLA, TOLB – **REAL**.

- **TOLA** and **TOLB** are the thresholds to determine the effective rank of $\begin{pmatrix} A \\ B \end{pmatrix}$. Generally, they are set to
  - **TOLA** $= \max(m, n)\, \|A\|$ **EPSILON**(1.0_wp),
  - **TOLB** $= \max(p, n)\, \|B\|$ **EPSILON**(1.0_wp).
- The size of **TOLA** and **TOLB** may affect the size of backward errors of the decomposition.

Further Details

Report number 2-96, "Based on Modifications by Ming Gu and Huan Ren, Computer Science Division, University of California at Berkeley, USA.

Examples

The results below are computed with $\epsilon = 1.19209_{10^{-7}}$.

Example 1 (from Program LA_GGSVD_EXAMPLE)

$$A = \begin{pmatrix} 1 & 2+2i & 5-3i \\ 6+2i & -6+3i & 7+i \\ 1+2i & 4-2i & -7-2i \\ 2i & 9 & -7+3i \\ -4+3i & -3 & 3-i \end{pmatrix}, \quad B = \begin{pmatrix} 3+2i & -4+4i & -i \\ 2+2i & -10+2i & -5+i \end{pmatrix}$$

Arrays **A** and **B** on entry:

A

$(1, 0)$	$(2, 2)$	$(5, -3)$
$(6, 2)$	$(-6, 3)$	$(7, 1)$
$(1, 2)$	$(4, -2)$	$(-7, -2)$
$(0, 2)$	$(9, 0)$	$(-7, 3)$
$(-4, 3)$	$(-3, 0)$	$(3, -1)$

B

$(3, 2)$	$(-4, 4)$	$(0, -1)$
$(2, 2)$	$(-10, 2)$	$(-5, 1)$

The call:

CALL LA_GGSVD(A, B, ALPHA, BETA, K, L)

A, **ALPHA**, **BETA**, **K** and **L** on exit:

A

$(12.8700, 0.00000)$	$(3.78645_{10^{-1}}, 6.30547_{10^{-1}})$	$(-9.52308, 3.00962_{10^{-2}})$
$(0.00000, 0.00000)$	$(-13.8124, 0.00000)$	$(-5.37275, -1.55730)$
$(0.00000, 0.00000)$	$(0.00000, 0.00000)$	$(-12.1680, 0.00000)$
$(0.00000, 0.00000)$	$(0.00000, 0.00000)$	$(0.00000, 0.00000)$
$(0.00000, 0.00000)$	$(0.00000, 0.00000)$	$(0.00000, 0.00000)$

ALPHA
1.00000
$4.91662_{10^{-1}}$
$9.45822_{10^{-1}}$

BETA
0.00000
$8.70786_{10^{-1}}$
$3.24686_{10^{-1}}$

$\mathbf{K} = 1$
$\mathbf{L} = 2$

The generalized singular values of (A, B) are:

$$\begin{pmatrix} \infty \\ 5.64618 \\ 2.91303 \end{pmatrix}$$

Matrix R is:

$$R = \begin{pmatrix} 12.8700 & 3.78645 + 6.30547_{10^{-1}}\, i & -9.52308 + 3.00962_{10^{-1}}\, i \\ 0 & -13.8124 & -5.37275 - 1.55730i \\ 0 & 0 & -12.1680 \end{pmatrix}.$$

Example 2 (from Program LA_GGSVD_EXAMPLE)

Arrays **A** and **B** on entry: As in Example 1.

The call:

CALL LA_GGSVD(A, B, ALPHA, BETA, K, L, U, V, Q, &
INFO)

A, **ALPHA**, **BETA**, **K** and **L** on exit: As in Example 1.

U, **V**, **Q** and **INFO** on exit:

U

$(-1.20287_{10^{-1}}, \ 8.48472_{10^{-2}})$	$(\ 5.68144_{10^{-1}}, -1.61279_{10^{-1}})$
$(-3.74600_{10^{-1}}, -2.61951_{10^{-1}})$	$(-2.03812_{10^{-1}}, \ 2.36176_{10^{-1}})$
$(\ 4.56687_{10^{-1}}, -1.13901_{10^{-1}})$	$(-2.53457_{10^{-1}}, -2.93179_{10^{-1}})$
$(\ 6.48896_{10^{-1}}, -2.41744_{10^{-1}})$	$(\ 4.63500_{10^{-1}}, \ 3.54403_{10^{-1}})$
$(\ 3.12241_{10^{-2}}, \ 2.59560_{10^{-1}})$	$(\ 2.85223_{10^{-3}}, -2.51491_{10^{-1}})$

U *continued*

$(-2.47529_{10^{-1}}, \ 2.87380_{10^{-1}})$	$(\ 1.16535_{10^{-1}}, \ 6.86407_{10^{-1}})$
$(-4.16748_{10^{-1}}, \ 4.76240_{10^{-1}})$	$(\ 2.13112_{10^{-1}}, -1.74097_{10^{-1}})$
$(\ 1.43392_{10^{-1}}, \ 2.90525_{10^{-1}})$	$(-4.52024_{10^{-1}}, \ 2.63089_{10^{-1}})$
$(\ 4.70580_{10^{-3}}, \ 6.08956_{10^{-2}})$	
$(\ 3.07325_{10^{-1}}, -2.27184_{10^{-1}})$	
$(-4.71865_{10^{-1}}, -3.52570_{10^{-1}})$	$(\ 7.88587_{10^{-2}}, -1.17121_{10^{-1}})$

U *continued*

$(-2.47529_{10^{-1}}, \ 2.87380_{10^{-1}})$
$(-4.16748_{10^{-1}}, \ 4.76240_{10^{-1}})$
$(\ 1.43392_{10^{-1}}, \ 2.90525_{10^{-1}})$
$(\ 4.70580_{10^{-3}}, \ 6.08956_{10^{-2}})$
$(-4.71865_{10^{-1}}, -3.52570_{10^{-1}})$

V

$(-3.13546_{10^{-1}}, 1.847105_{10^{-1}})$	$(0.00000, 0.00000)$
$(-9.02092_{10^{-1}}, 2.31953_{10^{-1}})$	$(0.00000, 0.00000)$

Q

$(-1.88389_{10^{-1}}, -7.50513_{10^{-1}})$	$(\ 2.21487_{10^{-1}}, -5.58294_{10^{-2}})$
$(\ 3.84084_{10^{-1}}, -1.53059_{10^{-1}})$	$(-7.45109_{10^{-1}}, \ 3.15974_{10^{-2}})$
$(-4.79744_{10^{-1}}, -1.17578_{10^{-2}})$	$(-6.09710_{10^{-1}}, \ 1.41051_{10^{-1}})$

Q *continued*

$(\ 1.03882_{10^{-2}}, -5.90727_{10^{-1}})$
$(-4.74368_{10^{-1}}, -2.18728_{10^{-1}})$
$(\ 6.11610_{10^{-1}}, -6.32460_{10^{-2}})$

$$\mathbf{INFO} = 0$$

The generalized singular values of (A, B) are as in Example 1.

Matrices U, V and Q are:

$$U = \left(\begin{array}{rr} -1.20287_{10^{-1}} + 8.48472_{10^{-2}}i & 5.68144_{10^{-1}} - 1.61279_{10^{-1}}i \\ -3.74600_{10^{-1}} - 2.61951_{10^{-1}}i & -2.03812_{10^{-1}} + 2.36176_{10^{-1}}i \\ 4.56687_{10^{-1}} - 1.13901_{10^{-1}}i & -2.53457_{10^{-1}} - 2.93179_{10^{-1}}i \\ 6.48896_{10^{-1}} - 2.41744_{10^{-1}}i & 4.63500_{10^{-1}} + 3.54403_{10^{-1}}i \\ 3.12241_{10^{-2}} + 2.59560_{10^{-1}}i & 2.85223_{10^{-3}} - 2.51491_{10^{-1}}i \end{array} \right.$$

$$
\begin{array}{rr}
-2.47529_{10^{-1}} + 2.87380_{10^{-1}}\,i & 1.16535_{10^{-1}} + 6.86407_{10^{-1}}\,i \\
-4.16748_{10^{-1}} + 4.76240_{10^{-1}}\,i & 2.13112_{10^{-1}} - 1.74097_{10^{-1}}\,i \\
1.43392_{10^{-1}} + 2.90525_{10^{-1}}\,i & -4.52024_{10^{-1}} + 2.63089_{10^{-1}}\,i \\
4.70580_{10^{-3}} + 6.08956_{10^{-2}}\,i & 3.07325_{10^{-1}} - 2.27184_{10^{-1}}\,i \\
-4.71865_{10^{-1}} - 3.52570_{10^{-1}}\,i & 7.88587_{10^{-2}} - 1.17121_{10^{-1}}\,i
\end{array}
$$

$$
\left.
\begin{array}{r}
-2.47529_{10^{-1}} + 2.87380_{10^{-1}}\,i \\
-4.16748_{10^{-1}} + 4.76240_{10^{-1}}\,i \\
1.43392_{10^{-1}} + 2.90525_{10^{-1}}\,i \\
4.70580_{10^{-3}} + 6.08956_{10^{-2}}\,i \\
-4.71865_{10^{-1}} - 3.52570_{10^{-1}}\,i
\end{array}
\right),
$$

$$
V = \begin{pmatrix}
-3.13546_{10^{-1}} + 1.84710_{10^{-1}}\,i & 0.00000 \\
-9.02092_{10^{-1}} + 2.31953_{10^{-1}}\,i & 0.00000
\end{pmatrix},
$$

$$
Q = \left(
\begin{array}{rr}
-1.88389_{10^{-1}} - 7.50513_{10^{-1}}\,i & 2.21487_{10^{-1}} - 5.58294_{10^{-2}}\,i \\
3.84084_{10^{-1}} - 1.53059_{10^{-1}}\,i & -7.45109_{10^{-1}} + 3.15974_{10^{-2}}\,i \\
-4.79744_{10^{-1}} - 1.17578_{10^{-2}}\,i & -6.09710_{10^{-1}} + 1.41051_{10^{-1}}\,i
\end{array}
\right.
$$

$$
\left.
\begin{array}{r}
1.03882_{10^{-2}} - 5.90727_{10^{-1}}\,i \\
-4.74368_{10^{-1}} - 2.18728_{10^{-1}}\,i \\
6.11610_{10^{-1}} - 6.32460_{10^{-2}}\,i
\end{array}
\right).
$$