NAME

ctrevc.f −

SYNOPSIS

Functions/Subroutines

subroutine **ctrevc** (SIDE, HOWMNY, SELECT, N, T, LDT, VL, LDVL, VR, LDVR, MM, M, WORK, RWORK, INFO)

CTREVC

Function/Subroutine Documentation

subroutine ctrevc (characterSIDE, characterHOWMNY, logical, dimension(*)SELECT, integerN, complex, dimension(ldt, *)T, integerLDT, complex, dimension(ldvl, *)VL, integerLDVL, complex, dimension(ldvr, *)VR, integerLDVR, integerMM, integerM, complex, dimension(*)WORK, real, dimension(*)RWORK, integerINFO)

CTREVC

Purpose:

CTREVC computes some or all of the right and/or left eigenvectors of
a complex upper triangular matrix T.
Matrices of this type are produced by the Schur factorization of
a complex general matrix: A = Q*T*Q**H, as computed by CHSEQR.

The right eigenvector x and the left eigenvector y of T corresponding
to an eigenvalue w are defined by:

T*x = w*x, (y**H)*T = w*(y**H)

where y**H denotes the conjugate transpose of the vector y.
The eigenvalues are not input to this routine, but are read directly
from the diagonal of T.

This routine returns the matrices X and/or Y of right and left
eigenvectors of T, or the products Q*X and/or Q*Y, where Q is an
input matrix. If Q is the unitary factor that reduces a matrix A to
Schur form T, then Q*X and Q*Y are the matrices of right and left
eigenvectors of A.

Parameters:

SIDE

SIDE is CHARACTER*1
= ’R’: compute right eigenvectors only;
= ’L’: compute left eigenvectors only;
= ’B’: compute both right and left eigenvectors.

HOWMNY

HOWMNY is CHARACTER*1
= ’A’: compute all right and/or left eigenvectors;
= ’B’: compute all right and/or left eigenvectors,
backtransformed using the matrices supplied in
VR and/or VL;
= ’S’: compute selected right and/or left eigenvectors,
as indicated by the logical array SELECT.

SELECT

SELECT is LOGICAL array, dimension (N)
If HOWMNY = ’S’, SELECT specifies the eigenvectors to be
computed.
The eigenvector corresponding to the j-th eigenvalue is
computed if SELECT(j) = .TRUE..
Not referenced if HOWMNY = ’A’ or ’B’.

N

N is INTEGER
The order of the matrix T. N >= 0.

T

T is COMPLEX array, dimension (LDT,N)
The upper triangular matrix T. T is modified, but restored
on exit.

LDT

LDT is INTEGER
The leading dimension of the array T. LDT >= max(1,N).

VL

VL is COMPLEX array, dimension (LDVL,MM)
On entry, if SIDE = 'L' or 'B' and HOWMNY = 'B', VL must
contain an N-by-N matrix Q (usually the unitary matrix Q of
Schur vectors returned by CHSEQR).
On exit, if SIDE = 'L' or 'B', VL contains:
if HOWMNY = 'A', the matrix Y of left eigenvectors of T;
if HOWMNY = 'B', the matrix Q*Y;
if HOWMNY = 'S', the left eigenvectors of T specified by
SELECT, stored consecutively in the columns
of VL, in the same order as their
eigenvalues.
Not referenced if SIDE = 'R'.

LDVL

LDVL is INTEGER
The leading dimension of the array VL. LDVL >= 1, and if
SIDE = 'L' or 'B', LDVL >= N.

VR

VR is COMPLEX array, dimension (LDVR,MM)
On entry, if SIDE = 'R' or 'B' and HOWMNY = 'B', VR must
contain an N-by-N matrix Q (usually the unitary matrix Q of
Schur vectors returned by CHSEQR).
On exit, if SIDE = 'R' or 'B', VR contains:
if HOWMNY = 'A', the matrix X of right eigenvectors of T;
if HOWMNY = 'B', the matrix Q*X;
if HOWMNY = 'S', the right eigenvectors of T specified by
SELECT, stored consecutively in the columns
of VR, in the same order as their
eigenvalues.
Not referenced if SIDE = 'L'.

LDVR

LDVR is INTEGER
The leading dimension of the array VR. LDVR >= 1, and if
SIDE = 'R' or 'B'; LDVR >= N.

MM

MM is INTEGER
The number of columns in the arrays VL and/or VR. MM >= M.

M

M is INTEGER
The number of columns in the arrays VL and/or VR actually
used to store the eigenvectors. If HOWMNY = 'A' or 'B', M
is set to N. Each selected eigenvector occupies one

column.

WORK

WORK is COMPLEX array, dimension (2*N)

RWORK

RWORK is REAL array, dimension (N)

INFO

INFO is INTEGER
= 0: successful exit
< 0: if INFO = -i, the i-th argument had an illegal value

Author:

Univ. of Tennessee

Univ. of California Berkeley

Univ. of Colorado Denver

NAG Ltd.

Date:

November 2011

Further Details:

The algorithm used in this program is basically backward (forward) substitution, with scaling to make the the code robust against possible overflow.

Each eigenvector is normalized so that the element of largest magnitude has magnitude 1; here the magnitude of a complex number (x,y) is taken to be |x| + |y|.

Definition at line 218 of file ctrevc.f.

Author

Generated automatically by Doxygen for LAPACK from the source code.