## NAME

realGEauxiliary

## SYNOPSIS

### Functions

subroutine **sgesc2** (N, A, LDA, RHS, IPIV, JPIV, SCALE)
**SGESC2** solves a system of linear equations using the LU factorization with complete pivoting computed by sgetc2.

subroutine **sgetc2** (N, A, LDA, IPIV, JPIV, INFO)
**SGETC2** computes the LU factorization with complete pivoting of the general n-by-n matrix.

real function **slange** (NORM, M, N, A, LDA, WORK)
**SLANGE** returns the value of the 1-norm, Frobenius norm, infinity-norm, or the largest absolute value of any element of a general rectangular matrix.

subroutine **slaqge** (M, N, A, LDA, R, C, ROWCND, COLCND, AMAX, EQUED)
**SLAQGE** scales a general rectangular matrix, using row and column scaling factors computed by sgeequ.

subroutine **stgex2** (WANTQ, WANTZ, N, A, LDA, B, LDB, Q, LDQ, Z, LDZ, J1, N1, N2, WORK, LWORK, INFO)
**STGEX2** swaps adjacent diagonal blocks in an upper (quasi) triangular matrix pair by an orthogonal equivalence transformation.

## Detailed Description

This is the group of real auxiliary functions for GE matrices

## Function Documentation

### subroutine sgesc2 (integer N, real, dimension( lda, * ) A, integer LDA, real, dimension( * ) RHS, integer, dimension( * ) IPIV, integer, dimension( * ) JPIV, real SCALE)

**SGESC2** solves a system of linear equations using the LU factorization with complete pivoting computed by sgetc2.

**Purpose:**

```
SGESC2 solves a system of linear equations

          A * X = scale* RHS

with a general N-by-N matrix A using the LU factorization with
complete pivoting computed by SGETC2.
```

**Parameters**

*N*

```
N is INTEGER
The order of the matrix A.
```

*A*

```
A is REAL array, dimension (LDA,N)
On entry, the  LU part of the factorization of the n-by-n
matrix A computed by SGETC2:  A = P * L * U * Q
```

*LDA*

```
LDA is INTEGER
The leading dimension of the array A.  LDA >= max(1, N).
```

*RHS*

```
RHS is REAL array, dimension (N).
On entry, the right hand side vector b.
On exit, the solution vector X.
```

*IPIV*

```
IPIV is INTEGER array, dimension (N).
The pivot indices; for 1 <= i <= N, row i of the
matrix has been interchanged with row IPIV(i).
```

*JPIV*

JPIV is INTEGER array, dimension (N).
The pivot indices; for 1 <= j <= N, column j of the
matrix has been interchanged with column JPIV(j).

*SCALE*

SCALE is REAL
On exit, SCALE contains the scale factor. SCALE is chosen
0 <= SCALE <= 1 to prevent overflow in the solution.

**Author**

Univ. of Tennessee

Univ. of California Berkeley

Univ. of Colorado Denver

NAG Ltd.

**Date**

December 2016

**Contributors:**

Bo Kagstrom and Peter Poromaa, Department of Computing Science, Umea University, S-901 87 Umea, Sweden.

**subroutine sgetc2 (integer N, real, dimension( lda, * ) A, integer LDA, integer, dimension( * ) IPIV, integer, dimension( * ) JPIV, integer INFO)**

**SGETC2** computes the LU factorization with complete pivoting of the general n-by-n matrix.

**Purpose:**

SGETC2 computes an LU factorization with complete pivoting of the
n-by-n matrix A. The factorization has the form A = P * L * U * Q,
where P and Q are permutation matrices, L is lower triangular with
unit diagonal elements and U is upper triangular.

This is the Level 2 BLAS algorithm.

**Parameters**

*N*

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

*A*

A is REAL array, dimension (LDA, N)
On entry, the n-by-n matrix A to be factored.
On exit, the factors L and U from the factorization
A = P*L*U*Q; the unit diagonal elements of L are not stored.
If U(k, k) appears to be less than SMIN, U(k, k) is given the
value of SMIN, i.e., giving a nonsingular perturbed system.

*LDA*

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

*IPIV*

IPIV is INTEGER array, dimension(N).
The pivot indices; for 1 <= i <= N, row i of the
matrix has been interchanged with row IPIV(i).

*JPIV*

JPIV is INTEGER array, dimension(N).
The pivot indices; for 1 <= j <= N, column j of the

matrix has been interchanged with column JPIV(j).

*INFO*

```
INFO is INTEGER
 = 0: successful exit
 > 0: if INFO = k, U(k, k) is likely to produce overflow if
     we try to solve for x in Ax = b. So U is perturbed to
     avoid the overflow.
```

**Author**

Univ. of Tennessee

Univ. of California Berkeley

Univ. of Colorado Denver

NAG Ltd.

**Date**

June 2016

**Contributors:**

Bo Kagstrom and Peter Poromaa, Department of Computing Science, Umea University, S-901 87 Umea, Sweden.

**real function slange (character NORM, integer M, integer N, real, dimension( lda, * ) A, integer LDA, real, dimension( * ) WORK)**

**SLANGE** returns the value of the 1-norm, Frobenius norm, infinity-norm, or the largest absolute value of any element of a general rectangular matrix.

**Purpose:**

```
 SLANGE  returns the value of the one norm,  or the Frobenius norm, or
 the  infinity norm,  or the  element of  largest absolute value  of a
 real matrix A.
```

**Returns**

SLANGE

```
    SLANGE = ( max(abs(A(i,j))), NORM = 'M' or 'm'
             (
             ( norm1(A),         NORM = '1', 'O' or 'o'
             (
             ( normI(A),         NORM = 'I' or 'i'
             (
             ( normF(A),         NORM = 'F', 'f', 'E' or 'e'

 where  norm1  denotes the  one norm of a matrix (maximum column sum),
 normI  denotes the  infinity norm  of a matrix  (maximum row sum) and
 normF  denotes the  Frobenius norm of a matrix (square root of sum of
 squares).  Note that  max(abs(A(i,j)))  is not a consistent matrix norm.
```

**Parameters**

*NORM*

```
NORM is CHARACTER*1
Specifies the value to be returned in SLANGE as described
above.
```

*M*

```
M is INTEGER
The number of rows of the matrix A.  M >= 0.  When M = 0,
SLANGE is set to zero.
```

*N*

```
N is INTEGER
```

The number of columns of the matrix A. N >= 0. When N = 0,
SLANGE is set to zero.

*A*

A is REAL array, dimension (LDA,N)
The m by n matrix A.

*LDA*

LDA is INTEGER
The leading dimension of the array A. LDA >= max(M,1).

*WORK*

WORK is REAL array, dimension (MAX(1,LWORK)),
where LWORK >= M when NORM = 'I'; otherwise, WORK is not
referenced.

**Author**

Univ. of Tennessee

Univ. of California Berkeley

Univ. of Colorado Denver

NAG Ltd.

**Date**

December 2016

**subroutine slaqge (integer M, integer N, real, dimension( lda, * ) A, integer LDA, real, dimension( * ) R, real, dimension( * ) C, real ROWCND, real COLCND, real AMAX, character EQUED)**

**SLAQGE** scales a general rectangular matrix, using row and column scaling factors computed by sgeequ.

**Purpose:**

SLAQGE equilibrates a general M by N matrix A using the row and
column scaling factors in the vectors R and C.

**Parameters**

*M*

M is INTEGER
The number of rows of the matrix A. M >= 0.

*N*

N is INTEGER
The number of columns of the matrix A. N >= 0.

*A*

A is REAL array, dimension (LDA,N)
On entry, the M by N matrix A.
On exit, the equilibrated matrix. See EQUED for the form of
the equilibrated matrix.

*LDA*

LDA is INTEGER
The leading dimension of the array A. LDA >= max(M,1).

*R*

R is REAL array, dimension (M)
The row scale factors for A.

*C*

C is REAL array, dimension (N)
The column scale factors for A.

*ROWCND*

ROWCND is REAL
Ratio of the smallest R(i) to the largest R(i).

*COLCND*

COLCND is REAL
Ratio of the smallest C(i) to the largest C(i).

*AMAX*

AMAX is REAL
Absolute value of largest matrix entry.

*EQUED*

EQUED is CHARACTER*1
Specifies the form of equilibration that was done.
= 'N': No equilibration
= 'R': Row equilibration, i.e., A has been premultiplied by
diag(R).
= 'C': Column equilibration, i.e., A has been postmultiplied
by diag(C).
= 'B': Both row and column equilibration, i.e., A has been
replaced by diag(R) * A * diag(C).

**Internal Parameters:**

THRESH is a threshold value used to decide if row or column scaling
should be done based on the ratio of the row or column scaling
factors. If ROWCND < THRESH, row scaling is done, and if
COLCND < THRESH, column scaling is done.

LARGE and SMALL are threshold values used to decide if row scaling
should be done based on the absolute size of the largest matrix
element. If AMAX > LARGE or AMAX < SMALL, row scaling is done.

**Author**

Univ. of Tennessee

Univ. of California Berkeley

Univ. of Colorado Denver

NAG Ltd.

**Date**

December 2016

**subroutine stgex2 (logical WANTQ, logical WANTZ, integer N, real, dimension( lda, * ) A, integer LDA, real, dimension( ldb, * ) B, integer LDB, real, dimension( ldq, * ) Q, integer LDQ, real, dimension( ldz, * ) Z, integer LDZ, integer J1, integer N1, integer N2, real, dimension( * ) WORK, integer LWORK, integer INFO)**

**STGEX2** swaps adjacent diagonal blocks in an upper (quasi) triangular matrix pair by an orthogonal equivalence transformation.

**Purpose:**

STGEX2 swaps adjacent diagonal blocks (A11, B11) and (A22, B22)
of size 1-by-1 or 2-by-2 in an upper (quasi) triangular matrix pair
(A, B) by an orthogonal equivalence transformation.

(A, B) must be in generalized real Schur canonical form (as returned
by SGGES), i.e. A is block upper triangular with 1-by-1 and 2-by-2
diagonal blocks. B is upper triangular.

Optionally, the matrices Q and Z of generalized Schur vectors are updated.

Q(in) * A(in) * Z(in)**T = Q(out) * A(out) * Z(out)**T
Q(in) * B(in) * Z(in)**T = Q(out) * B(out) * Z(out)**T

**Parameters**

*WANTQ*

WANTQ is LOGICAL
.TRUE. : update the left transformation matrix Q;
.FALSE.: do not update Q.

*WANTZ*

WANTZ is LOGICAL
.TRUE. : update the right transformation matrix Z;
.FALSE.: do not update Z.

*N*

N is INTEGER
The order of the matrices A and B. N >= 0.

*A*

A is REAL array, dimension (LDA,N)
On entry, the matrix A in the pair (A, B).
On exit, the updated matrix A.

*LDA*

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

*B*

B is REAL array, dimension (LDB,N)
On entry, the matrix B in the pair (A, B).
On exit, the updated matrix B.

*LDB*

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

*Q*

Q is REAL array, dimension (LDQ,N)
On entry, if WANTQ = .TRUE., the orthogonal matrix Q.
On exit, the updated matrix Q.
Not referenced if WANTQ = .FALSE..

*LDQ*

LDQ is INTEGER
The leading dimension of the array Q. LDQ >= 1.
If WANTQ = .TRUE., LDQ >= N.

*Z*

Z is REAL array, dimension (LDZ,N)
On entry, if WANTZ =.TRUE., the orthogonal matrix Z.
On exit, the updated matrix Z.
Not referenced if WANTZ = .FALSE..

*LDZ*

LDZ is INTEGER
The leading dimension of the array Z. LDZ >= 1.
If WANTZ = .TRUE., LDZ >= N.

*J1*

J1 is INTEGER
The index to the first block (A11, B11). 1 <= J1 <= N.

*N1*

N1 is INTEGER
The order of the first block (A11, B11). N1 = 0, 1 or 2.

*N2*

N2 is INTEGER
The order of the second block (A22, B22). N2 = 0, 1 or 2.

*WORK*

WORK is REAL array, dimension (MAX(1,LWORK)).

*LWORK*

LWORK is INTEGER
The dimension of the array WORK.
LWORK >= MAX( N*(N2+N1), (N2+N1)*(N2+N1)*2 )

*INFO*

INFO is INTEGER
=0: Successful exit
>0: If INFO = 1, the transformed matrix (A, B) would be
too far from generalized Schur form; the blocks are
not swapped and (A, B) and (Q, Z) are unchanged.
The problem of swapping is too ill-conditioned.
<0: If INFO = -16: LWORK is too small. Appropriate value
for LWORK is returned in WORK(1).

**Author**

Univ. of Tennessee

Univ. of California Berkeley

Univ. of Colorado Denver

NAG Ltd.

**Date**

June 2017

**Further Details:**

In the current code both weak and strong stability tests are performed. The user can omit the strong stability test by changing the internal logical parameter WANDS to .FALSE.. See ref. [2] for details.

**Contributors:**

Bo Kagstrom and Peter Poromaa, Department of Computing Science, Umea University, S-901 87 Umea, Sweden.

**References:**

[1] B. Kagstrom; A Direct Method for Reordering Eigenvalues in the Generalized Real Schur Form of a Regular Matrix Pair (A, B), in M.S. Moonen et al (eds), Linear Algebra for Large Scale and Real-Time Applications, Kluwer Academic Publ. 1993, pp 195-218.

[2] B. Kagstrom and P. Poromaa; Computing Eigenspaces with Specified Eigenvalues of a Regular Matrix Pair (A, B) and Condition Estimation: Theory, Algorithms and Software, Report UMINF - 94.04, Department of Computing Science, Umea University, S-901 87 Umea, Sweden, 1994. Also as LAPACK Working Note 87. To appear in Numerical Algorithms, 1996.

**Author**

Generated automatically by Doxygen for LAPACK from the source code.