NAG C Library Function Document

nag_zungtr (f08ftc)

1 Purpose

nag_zungtr (f08ftc) generates the complex unitary matrix Q, which was determined by nag_zhetrd (f08fsc) when reducing a Hermitian matrix to tridiagonal form.

2 Specification

```
void nag_zungtr (Nag_OrderType order, Nag_UploType uplo, Integer n, Complex a[],
     Integer pda, const Complex tau[], NagError *fail)
```

3 Description

nag_zungtr (f08ftc) is intended to be used after a call to nag_zhetrd (f08fsc), which reduces a complex Hermitian matrix A to real symmetric tridiagonal form T by a unitary similarity transformation: $A = QTQ^H$. nag_zhetrd (f08fsc) represents the unitary matrix Q as a product of $n-1$ elementary reflectors.

This function may be used to generate Q explicitly as a square matrix.

4 References

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Parameters

1: **order** – Nag_OrderType *Input*

On entry: the **order** parameter specifies the two-dimensional storage scheme being used, i.e., row-major ordering or column-major ordering. C language defined storage is specified by **order** = **Nag_RowMajor**. See Section 2.2.1.4 of the Essential Introduction for a more detailed explanation of the use of this parameter.

Constraint: **order** = **Nag_RowMajor** or **Nag_ColMajor**.

2: **uplo** – Nag_UploType *Input*

On entry: this **must** be the same parameter **uplo** as supplied to nag_zhetrd (f08fsc).

Constraint: **uplo** = **Nag_Upper** or **Nag_Lower**.

3: **n** – Integer *Input*

On entry: n, the order of the matrix Q.

Constraint: **n** ≥ 0.

4: **a**[dim] – Complex *Input/Output*

Note: the dimension, dim, of the array **a** must be at least $\max(1, \textbf{pda} \times \textbf{n})$.

If **order** = **Nag_ColMajor**, the (i,j)th element of the matrix A is stored in **a**$[(j-1) \times \textbf{pda} + i - 1]$ and if **order** = **Nag_RowMajor**, the (i,j)th element of the matrix A is stored in **a**$[(i-1) \times \textbf{pda} + j - 1]$.

On entry: details of the vectors which define the elementary reflectors, as returned by nag_zhetrd (f08fsc).

On exit: the n by n unitary matrix Q.

5: **pda** – Integer *Input*

On entry: the stride separating row or column elements (depending on the value of **order**) of the matrix A in the array **a**.

Constraint: **pda** $\geq \max(1, \mathbf{n})$.

6: **tau**$[dim]$ – const Complex *Input*

Note: the dimension, dim, of the array **tau** must be at least $\max(1, \mathbf{n} - 1)$.

On entry: further details of the elementary reflectors, as returned by nag_zhetrd (f08fsc).

7: **fail** – NagError * *Output*

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

On entry, **n** = ⟨*value*⟩.
Constraint: **n** ≥ 0.

On entry, **pda** = ⟨*value*⟩.
Constraint: **pda** > 0.

NE_INT_2

On entry, **pda** = ⟨*value*⟩, **n** = ⟨*value*⟩.
Constraint: **pda** $\geq \max(1, \mathbf{n})$.

NE_ALLOC_FAIL

Memory allocation failed.

NE_BAD_PARAM

On entry, parameter ⟨*value*⟩ had an illegal value.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please consult NAG for assistance.

7 Accuracy

The computed matrix Q differs from an exactly unitary matrix by a matrix E such that

$$\|E\|_2 = O(\epsilon),$$

where ϵ is the ***machine precision***.

8 Further Comments

The total number of real floating-point operations is approximately $\frac{16}{3}n^3$.

The real analogue of this function is nag_dorgtr (f08ffc).

9 Example

To compute all the eigenvalues and eigenvectors of the matrix A, where

$$A = \begin{pmatrix} -2.28 + 0.00i & 1.78 - 2.03i & 2.26 + 0.10i & -0.12 + 2.53i \\ 1.78 + 2.03i & -1.12 + 0.00i & 0.01 + 0.43i & -1.07 + 0.86i \\ 2.26 - 0.10i & 0.01 - 0.43i & -0.37 + 0.00i & 2.31 - 0.92i \\ -0.12 - 2.53i & -1.07 - 0.86i & 2.31 + 0.92i & -0.73 + 0.00i \end{pmatrix}.$$

Here A is Hermitian and must first be reduced to tridiagonal form by nag_zhetrd (f08fsc). The program then calls nag_zungtr (f08ftc) to form Q, and passes this matrix to nag_zsteqr (f08jsc) which computes the eigenvalues and eigenvectors of A.

9.1 Program Text

```
/* nag_zungtr (f08ftc) Example Program.
 *
 * Copyright 2001 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagf08.h>
#include <nagx04.h>

int main(void)
{
  /* Scalars */
  Integer  i, j, n, pda, pdz, d_len, e_len, tau_len;
  Integer  exit_status=0;
  NagError fail;
  Nag_UploType  uplo;
  Nag_OrderType order;
  /* Arrays */
  char    uplo_char[2];
  Complex *a=0, *tau=0, *z=0;
  double  *d=0, *e=0;

#ifdef NAG_COLUMN_MAJOR
#define A(I,J) a[(J-1)*pda + I - 1]
#define Z(I,J) z[(J-1)*pdz + I - 1]
  order = Nag_ColMajor;
#else
#define A(I,J) a[(I-1)*pda + J - 1]
#define Z(I,J) z[(I-1)*pdz + J - 1]
  order = Nag_RowMajor;
#endif

  INIT_FAIL(fail);
  Vprintf("f08ftc Example Program Results\n");

  /* Skip heading in data file */
  Vscanf("%*[^\n] ");
  Vscanf("%ld%*[^\n] ", &n);
#ifdef NAG_COLUMN_MAJOR
  pda = n;
  pdz = n;
#else
  pda = n;
  pdz = n;
#endif

  tau_len = n-1;
  d_len = n;
  e_len = n-1;
  /* Allocate memory */
```

```
if ( !(a = NAG_ALLOC(n * n, Complex)) ||
     !(tau = NAG_ALLOC(tau_len, Complex)) ||
     !(z = NAG_ALLOC(n * n, Complex)) ||
     !(d = NAG_ALLOC(d_len, double)) ||
     !(e = NAG_ALLOC(e_len, double)) )
  {
    Vprintf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }

/* Read A from data file */
Vscanf(" ' %1s '%*[^\n] ", uplo_char);
if (*(unsigned char *)uplo_char == 'L')
  uplo = Nag_Lower;
else if (*(unsigned char *)uplo_char == 'U')
  uplo = Nag_Upper;
else
  {
    Vprintf("Unrecognised character for Nag_UploType type\n");
    exit_status = -1;
    goto END;
  }
if (uplo == Nag_Upper)
  {
    for (i = 1; i <= n; ++i)
      {
        for (j = i; j <= n; ++j)
          Vscanf(" ( %lf , %lf )", &A(i,j).re, &A(i,j).im);
      }
    Vscanf("%*[^\n] ");
  }
else
  {
    for (i = 1; i <= n; ++i)
      {
        for (j = 1; j <= i; ++j)
          Vscanf(" ( %lf , %lf )", &A(i,j).re, &A(i,j).im);
      }
    Vscanf("%*[^\n] ");
  }

/* Reduce A to tridiagonal form T = (Q**H)*A*Q */
f08fsc(order, uplo, n, a, pda, d, e, tau, &fail);
if (fail.code != NE_NOERROR)
  {
    Vprintf("Error from f08fsc.\n%s\n", fail.message);
    exit_status = 1;
    goto END;
  }

/* Copy A into Z */
if (uplo == Nag_Upper)
  {
    for (i = 1; i <= n; ++i)
      {
        for (j = i; j <= n; ++j)
          {
            Z(i,j).re = A(i,j).re;
            Z(i,j).im = A(i,j).im;
          }
      }
  }
else
  {
    for (i = 1; i <= n; ++i)
      {
        for (j = 1; j <= i; ++j)
          {
            Z(i,j).re = A(i,j).re;
            Z(i,j).im = A(i,j).im;
```

```
          }
        }
    }
  /* Form Q explicitly, storing the result in Z */
  f08ftc(order, uplo, n, z, pdz, tau, &fail);
  if (fail.code != NE_NOERROR)
    {
      Vprintf("Error from f08ftc.\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }

  /* Calculate all the eigenvalues and eigenvectors of A */
  f08jsc(order, Nag_UpdateZ, n, d, e, z, pdz, &fail);
  if (fail.code != NE_NOERROR)
    {
      Vprintf("Error from f08jsc.\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
  /* Print eigenvalues and eigenvectors */
  Vprintf("\nEigenvalues\n");
  for (i = 1; i <= n; ++i)
    Vprintf("%9.4f%s", d[i-1], i%4==0 ?"\n":" ");
  Vprintf("\n");
  x04dbc(order, Nag_GeneralMatrix, Nag_NonUnitDiag, n, n,
         z, pdz, Nag_BracketForm, "%7.4f", "Eigenvectors",
         Nag_IntegerLabels, 0, Nag_IntegerLabels, 0, 80, 0,
         0, &fail);
  if (fail.code != NE_NOERROR)
    {
      Vprintf("Error from x04dbc.\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
 END:
  if (a) NAG_FREE(a);
  if (tau) NAG_FREE(tau);
  if (z) NAG_FREE(z);
  if (d) NAG_FREE(d);
  if (e) NAG_FREE(e);

  return exit_status;
}
```

9.2 Program Data

```
f08ftc Example Program Data
  4                                                         :Value of N
  'L'                                                       :Value of UPLO
 (-2.28, 0.00)
 ( 1.78, 2.03) (-1.12, 0.00)
 ( 2.26,-0.10) ( 0.01,-0.43) (-0.37, 0.00)
 (-0.12,-2.53) (-1.07,-0.86) ( 2.31, 0.92) (-0.73, 0.00)   :End of matrix A
```

9.3 Program Results

```
f08ftc Example Program Results

Eigenvalues
  -6.0002   -3.0030    0.5036    3.9996

 Eigenvectors
                   1                   2                   3                   4
 1  ( 0.7299, 0.0000)  (-0.2120, 0.1497)  ( 0.1000,-0.3570)  ( 0.1991, 0.4720)
 2  (-0.1663,-0.2061)  ( 0.7307, 0.0000)  ( 0.2863,-0.3353)  (-0.2467, 0.3751)
 3  (-0.4165,-0.1417)  (-0.3291, 0.0479)  ( 0.6890, 0.0000)  ( 0.4468, 0.1466)
 4  ( 0.1743, 0.4162)  ( 0.5200, 0.1329)  ( 0.0662, 0.4347)  ( 0.5612, 0.0000)
```