NAG C Library Function Document
nag_zgehdr (f08nsc)

1 Purpose

nag_zgehdr (f08nsc) reduces a complex general matrix to Hessenberg form.

2 Specification

void nag_zgehdr (Nag_OrderType order, Integer n, Integer ilo, Integer ihi, Complex a[], Integer pda, Complex tau[], NagError *fail)

3 Description

nag_zgehdr (f08nsc) reduces a complex general matrix $A$ to upper Hessenberg form $H$ by a unitary similarity transformation: $A = QH{Q}^{H}$. $H$ has real subdiagonal elements.

The matrix $Q$ is not formed explicitly, but is represented as a product of elementary reflectors (see the f08 Chapter Introduction for details). Functions are provided to work with $Q$ in this representation (see Section 8).

The function can take advantage of a previous call to nag_zgebal (f08nvc), which may produce a matrix with the structure:

$$
\begin{pmatrix}
A_{11} & A_{12} & A_{13} \\
A_{22} & A_{23} \\
A_{33}
\end{pmatrix}
$$

where $A_{11}$ and $A_{33}$ are upper triangular. If so, only the central diagonal block $A_{22}$, in rows and columns $i_{lo}$ to $i_{hi}$, needs to be reduced to Hessenberg form (the blocks $A_{12}$ and $A_{23}$ will also be affected by the reduction). Therefore the values of $i_{lo}$ and $i_{hi}$ determined by nag_zgebal (f08nvc) can be supplied to the function directly. If nag_zgebal (f08nvc) has not previously been called however, then $i_{lo}$ must be set to 1 and $i_{hi}$ to $n$.

4 References


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:  n – Integer

   Input

   On entry: $n$, the order of the matrix $A$.

   Constraint: $n \geq 0$. 
3:  \text{ilo} – Integer  
\text{Input}

4:  \text{ihi} – Integer  
\text{Input}

\text{On entry: if } A \text{ has been output by nag_zgebal (f08nvc), then } \text{ilo} \text{ and } \text{ihi} \text{ must contain the values returned by that function. Otherwise, } \text{ilo} \text{ must be set to 1 and } \text{ihi} \text{ to } n. 

\text{Constraints:}
\begin{align*}
& \text{if } n > 0, 1 \leq \text{ilo} \leq \text{ihi} \leq n; \\
& \text{if } n = 0, \text{ilo} = 1 \text{ and } \text{ihi} = 0.
\end{align*}

5:  \text{a}[\text{dim}] – Complex  
\text{Input/Output}

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

\text{If } \text{order} = \text{Nag\_ColMajor}, \text{ the } (i, j)\text{th element of the matrix } A \text{ is stored in } \text{a}[(j - 1) \times \text{pda} + i - 1] \text{ and if } \text{order} = \text{Nag\_RowMajor}, \text{ the } (i, j)\text{th element of the matrix } A \text{ is stored in } \text{a}[(i - 1) \times \text{pda} + j - 1]. 

\text{On entry: the } n \text{ by } n \text{ general matrix } A. 

\text{On exit: } A \text{ is overwritten by the upper Hessenberg matrix } H \text{ and details of the unitary matrix } Q. \text{ The subdiagonal elements of } H \text{ are real.}

6:  \text{pda} – Integer  
\text{Input}

\text{On entry: the stride separating matrix row or column elements (depending on the value of } \text{order} \text{) in the array } \text{a}. 

\text{Constraint: } \text{pda} \geq \max(1, n). 

7:  \text{tau}[\text{dim}] – Complex  
\text{Output}

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

\text{On exit: further details of the unitary matrix } Q. 

8:  \text{fail} – NagError *  
\text{Output}

\text{The NAG error parameter (see the Essential Introduction).}

6  \text{Error Indicators and Warnings}

\text{NE\_INT}

\text{On entry, } n = \langle \text{value} \rangle. 
\text{Constraint: } n \geq 0. 
\text{On entry, } \text{pda} = \langle \text{value} \rangle. 
\text{Constraint: } \text{pda} > 0.

\text{NE\_INT\_2}

\text{On entry, } \text{pda} = \langle \text{value} \rangle, \text{ n} = \langle \text{value} \rangle. 
\text{Constraint: } \text{pda} \geq \max(1, n).

\text{NE\_INT\_3}

\text{On entry, } n = \langle \text{value} \rangle, \text{ ilo} = \langle \text{value} \rangle, \text{ ihi} = \langle \text{value} \rangle. 
\text{Constraint: if } n > 0, 1 \leq \text{ilo} \leq \text{ihi} \leq n; \\
\text{if } n = 0, \text{ilo} = 1 \text{ and } \text{ihi} = 0.

\text{NE\_ALLOC\_FAIL}

\text{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 Hessenberg matrix \( H \) is exactly similar to a nearby matrix \( A + E \), where
\[
\| E \|_2 \leq c(n)\epsilon\| A \|_2,
\]
c\( (n) \) is a modestly increasing function of \( n \), and \( \epsilon \) is the machine precision.
The elements of \( H \) themselves may be sensitive to small perturbations in \( A \) or to rounding errors in the
computation, but this does not affect the stability of the eigenvalues, eigenvectors or Schur factorization.

8 Further Comments

The total number of real floating-point operations is approximately
\[
\frac{8}{3} q^2 (2q + 3n), \quad \text{where} \quad q = ihi - ilo; \text{ if} \quad ilo = 1 \text{ and } ihi = n, \text{ the number is approximately } \frac{40}{3} n^3.
\]
To form the unitary matrix \( Q \) this function may be followed by a call to nag_zunghr (f08ntc):
\[
nag\_zunghr (\text{order}, n, ilo, ihi, \& a, pda, tau, \& fail)
\]
To apply \( Q \) to an \( m \) by \( n \) complex matrix \( C \) this function may be followed by a call to nag_zunmhr
(f08nuc). For example,
\[
nag\_zunmhr (\text{order}, Nag\_LeftSide, Nag\_NoTrans, m, n, ilo, ihi, \& a, pda, tau, \& c, pdc, \& fail)
\]
forms the matrix product \( QC \).
The real analogue of this function is nag_dgehrd (f08nec).

9 Example

To compute the upper Hessenberg form of the matrix \( A \), where
\[
A = \begin{pmatrix}
-3.97 - 5.04i & -4.11 + 3.70i & -0.34 + 1.01i & 1.29 - 0.86i \\
0.34 - 1.50i & 1.52 - 0.43i & 1.88 - 5.38i & 3.36 + 0.65i \\
3.31 - 3.85i & 2.50 + 3.45i & 0.88 - 1.08i & 0.64 - 1.48i \\
-1.10 + 0.82i & 1.81 - 1.59i & 3.25 + 1.33i & 1.57 - 3.44i
\end{pmatrix}
\]

9.1 Program Text

/* nag_zgehrd (f08nsc) 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, tau_len;
```c
Integer exit_status=0;
NagError fail;
Nag_OrderType order;
/* Arrays */
Complex *a=0, *tau=0;

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

INIT_FAIL(fail);
Vprintf("f08nsc Example Program Results\n\n");

/* Skip heading in data file */
Vscanf("%*[^
] ");
Vscanf("%ld%*[^
] ", &n);
#ifdef NAG_COLUMN_MAJOR
pda = n;
#else
pda = n;
#endif
tau_len = n - 1;

/* Allocate memory */
if ( !(a = NAG_ALLOC(n * n, Complex)) ||
     !(tau = NAG_ALLOC(tau_len, Complex)) )
{
    Vprintf("Allocation failure\n");
    exit_status = -1;
    goto END;
}

/* Read A from data file */
for (i = 1; i <= n; ++i)
{
    for (j = 1; j <= n; ++j)
        Vscanf(" ( %lf , %lf )", &A(i,j).re, &A(i,j).im);
}
Vscanf("%*[^
] ");

/* Reduce A to upper Hessenberg form */
f08nsc(order, n, 1, n, a, pda, tau, &fail);
if (fail.code != NE_NOERROR)
{
    Vprintf("Error from f08nsc.\n%s\n", fail.message);
    exit_status = 1;
    goto END;
}

/* Set the elements below the first sub-diagonal to zero */
for (i = 1; i <= n - 2; ++i)
{
    for (j = i + 2; j <= n; ++j)
        A(j, i).re = 0.0, A(j, i).im = 0.0;
}

/* Print upper Hessenberg form */
x04dbc(order, Nag_GeneralMatrix, Nag_NonUnitDiag, n, n, a, pda, Nag_BracketForm, "%7.4f",
"Upper Hessenberg form", 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;
}
```
if (a) NAG_FREE(a);
if (tau) NAG_FREE(tau);
return exit_status;
}

9.2 Program Data

f08nsc Example Program Data

4 :Value of N

(-3.97,-5.04) (-4.11, 3.70) (-0.34, 1.01) ( 1.29,-0.86)
( 0.34,-1.50) ( 1.52,-0.43) ( 1.88,-5.38) ( 3.36, 0.65)
( 3.31,-3.85) ( 2.50, 3.45) ( 0.88,-1.08) ( 0.64,-1.48)
(-1.10, 0.82) ( 1.81,-1.59) ( 3.25, 1.33) ( 1.57,-3.44) :End of matrix A

9.3 Program Results

f08nsc Example Program Results

Upper Hessenberg form

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(-3.9700,-5.0400)</td>
<td>(-1.1318,-2.5693)</td>
<td>(-4.6027,-0.1426)</td>
<td>(-1.4249, 1.7330)</td>
</tr>
<tr>
<td>2</td>
<td>(-5.4797, 0.0000)</td>
<td>( 1.8585,-1.5502)</td>
<td>( 4.4145,-0.7638)</td>
<td>(-0.4805,-1.1976)</td>
</tr>
<tr>
<td>3</td>
<td>( 0.0000, 0.0000)</td>
<td>( 6.2673, 0.0000)</td>
<td>(-0.4504,-0.0290)</td>
<td>(-1.3467, 1.6579)</td>
</tr>
<tr>
<td>4</td>
<td>( 0.0000, 0.0000)</td>
<td>( 0.0000, 0.0000)</td>
<td>(-3.5000, 0.0000)</td>
<td>( 2.5619,-3.3708)</td>
</tr>
</tbody>
</table>