NAG C Library Function Document

nag_dgehrd (f08nec)

1 Purpose

nag_dgehrd (f08nec) reduces a real general matrix to Hessenberg form.

2 Specification

```c
void nag_dgehrd (Nag_OrderType order, Integer n, Integer ilo, Integer ihi,
                double a[], Integer pda, double tau[], NagError *fail)
```

3 Description

nag_dgehrd (f08nec) reduces a real general matrix $A$ to upper Hessenberg form $H$ by an orthogonal similarity transformation: $A = QHQ^T$.

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_dgebal (f08nhc), 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_dgebal (f08nhc) can be supplied to the function directly. If nag_dgebal (f08nhc) 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: $n \geq 0$. 

[NP3645/7]
3:  \text{ilo} \quad \text{Integer} \quad \text{Input}
4:  \text{ihi} \quad \text{Integer} \quad \text{Input}

\text{On entry: if } A \text{ has been output by nag_dgebal (f08nhc), 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 \text{ and } \text{ihi} \text{ to } n.

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

5:  \text{a[\text{dim}]} \quad \text{double} \quad \text{Input/Output}

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

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

\text{On exit: 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 orthogonal matrix } Q.

6:  \text{pda} \quad \text{Integer} \quad \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 \text{max}(1, n).

7:  \text{tau[\text{dim}]} \quad \text{double} \quad \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 orthogonal matrix } Q.

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

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

6 \quad \text{Error Indicators and Warnings}

\text{NE_INT}

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

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

\text{NE_INT_2}

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

\text{NE_BAD_PARAM}

\text{On entry, parameter } \langle \text{value} \rangle \text{ 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 floating-point operations is approximately
\[
\frac{2}{3} q^2 (2q + 3n), \quad \text{where} \quad q = ihi - ilo; \quad \text{if} \quad ilo = 1 \quad \text{and} \quad ihi = n, \quad \text{the number is approximately} \quad \frac{10}{3} n^3.
\]

To form the orthogonal matrix \( Q \) this function may be followed by a call to nag_dorghr (f08nfc):
\[
nag_dorghr (\text{order}, n, ilo, ihi, \& a, pda, tau, \& fail)
\]
To apply \( Q \) to an \( m \) by \( n \) real matrix \( C \) this function may be followed by a call to nag_dormhr (f08ngc). For example,
\[
nag_dormhr (\text{order}, \text{Nag_LeftSide}, \text{Nag_NoTrans}, m, n, ilo, ihi, \& a, pda, tau, \& c, pdc, \& fail)
\]
forms the matrix product \( QC \).
The complex analogue of this function is nag_zgehrd (f08nsc).

9 Example

To compute the upper Hessenberg form of the matrix \( A \), where
\[
A = \begin{pmatrix}
0.35 & 0.45 & -0.14 & -0.17 \\
0.09 & 0.07 & -0.54 & 0.35 \\
-0.44 & -0.33 & -0.03 & 0.17 \\
0.25 & -0.32 & -0.13 & 0.11
\end{pmatrix},
\]

9.1 Program Text

/* nag_dgehrd (f08nec) Example Program.*/
#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;
    Integer exit_status=0;
    NagError fail;
    Nag_OrderType order;
    /* Arrays */
#include <nag.h>

double *a=0, *tau=0;

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

order = Nag_ColMajor;

INIT_FAIL(fail);
Vprintf("f08nec Example Program Results\n\n");

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

/* Allocate memory */
if ( !(a = NAG_ALLOC(n * n, double)) ||
    !(tau = NAG_ALLOC(tau_len, double)) )
{
    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", &A(i,j));
}
Vscanf("%*[^
] ");

/* Reduce A to upper Hessenberg form */
f08nec(order, n, 1, n, a, pda, tau, &fail);
if (fail.code != NE_NOERROR)
{
    Vprintf("Error from f08nec.\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) = 0.0;
}

/* Print upper Hessenberg form */
x04cac(order, Nag_GeneralMatrix, Nag_NonUnitDiag, n, n, a, pda, "Upper Hessenberg form", 0, &fail);
if (fail.code != NE_NOERROR)
{
    Vprintf("Error from x04cac.\n%s\n", fail.message);
    exit_status = 1;
    goto END;
}

END:
if (a) NAG_FREE(a);
if (tau) NAG_FREE(tau);
return exit_status;
9.2  Program Data

f08nec Example Program Data

:Value of N
0.35   0.45   -0.14   -0.17
0.09   0.07   -0.54   0.35
-0.44  -0.33   -0.03   0.17
0.25   -0.32  -0.13   0.11 :End of matrix A

9.3  Program Results

f08nec Example Program Results

Upper Hessenberg form

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>0.3500</td>
<td>-0.1160</td>
<td>-0.3886</td>
</tr>
<tr>
<td>2</td>
<td>-0.5140</td>
<td>0.1225</td>
<td>0.1004</td>
</tr>
<tr>
<td>3</td>
<td>0.0000</td>
<td>0.6443</td>
<td>-0.1357</td>
</tr>
<tr>
<td>4</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.4262</td>
</tr>
</tbody>
</table>