NAG C Library Function Document

nag_dgecon (f07agc)

1 Purpose

nag_dgecon (f07agc) estimates the condition number of a real matrix $A$, where $A$ has been factorized by nag_dgetrf (f07adc).

2 Specification

```c
void nag_dgecon (Nag_OrderType order, Nag_NormType norm, Integer n,
                const double a[], Integer pda, double anorm, double *rcond, NagError *fail)
```

3 Description

nag_dgecon (f07agc) estimates the condition number of a real matrix $A$, in either the 1-norm or the infinity-norm:

$$
\kappa_1(A) = \| A \|_1 \| A^{-1} \|_1 \quad \text{or} \quad \kappa_\infty(A) = \| A \|_\infty \| A^{-1} \|_\infty.
$$

Note that $\kappa_\infty(A) = \kappa_1(A^T)$.

Because the condition number is infinite if $A$ is singular, the function actually returns an estimate of the reciprocal of the condition number.

The function should be preceded by a call to nag_dge_norm (f16rac) to compute $\| A \|_1$ or $\| A \|_\infty$, and a call to nag_dgetrf (f07adc) to compute the $LU$ factorization of $A$. The function then uses Higham’s implementation of Hager’s method (see Higham (1988)) to estimate $\| A^{-1} \|_1$ or $\| A^{-1} \|_\infty$.

4 References

Higham N J (1988) FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation ACM Trans. Math. Software 14 381–396

5 Parameters

1:  
   ```
   order – Nag_OrderType
   ```
   
   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:  
   ```
   norm – Nag_NormType
   ```
   
   On entry: indicates whether $\kappa_1(A)$ or $\kappa_\infty(A)$ is estimated as follows:

   - if norm = Nag_OneNorm, $\kappa_1(A)$ is estimated;
   - if norm = Nag_InfNorm, $\kappa_\infty(A)$ is estimated.

   Constraint: norm = Nag_OneNorm or Nag_InfNorm.

3:  
   ```
   n – Integer
   ```
   
   On entry: $n$, the order of the matrix $A$.

   Constraint: $n \geq 0$. 

5 References

Higham N J (1988) FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation ACM Trans. Math. Software 14 381–396
4: \( a[dim] \) – const double  
   \textit{Input}

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

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

   \textit{On entry:} the \( LU \) factorization of \( A \), as returned by \texttt{nag_dgetrf} (f07adc).

5: \( pda \) – Integer  
   \textit{Input}

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

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

6: \( anorm \) – double  
   \textit{Input}

   \textit{On entry:} if \textit{norm} = Nag_OneNorm, the 1-norm of the original matrix \( A \); if \textit{norm} = Nag_InfNorm, the infinity-norm of the original matrix \( A \). \( anorm \) may be computed by calling \texttt{nag_dge_norm} (f16rac) with the same value for the parameter \textit{norm}. \( anorm \) must be computed either \textbf{before} calling \texttt{nag_dgetrf} (f07adc) or else from a \textit{copy} of the original matrix \( A \).

   \textbf{Constraint:} \( anorm \geq 0.0 \).

7: \( rcond \) – double *  
   \textit{Output}

   \textit{On exit:} an estimate of the reciprocal of the condition number of \( A \). \( rcond \) is set to zero if exact singularity is detected or the estimate underflows. If \( rcond \) is less than \textit{machine precision}, \( A \) is singular to working precision.

8: \( fail \) – NagError *  
   \textit{Output}

   The NAG error parameter (see the Essential Introduction).

6 \textbf{Error Indicators and Warnings}

**NE_INT**

On entry, \( n = \langle \text{value} \rangle \).

Constraint: \( n \geq 0 \).

On entry, \( pda = \langle \text{value} \rangle \).

Constraint: \( pda > 0 \).

**NE_INT_2**

On entry, \( pda = \langle \text{value} \rangle, n = \langle \text{value} \rangle \).

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

**NE_REAL**

On entry, \( anorm = \langle \text{value} \rangle \).

Constraint: \( anorm \geq 0.0 \).

**NE_ALLOC_FAIL**

Memory allocation failed.

**NE_BAD_PARAM**

On entry, parameter \( \langle \text{value} \rangle \) 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 estimate $rcond$ is never less than the true value $\rho$, and in practice is nearly always less than
$10\rho$, although examples can be constructed where $rcond$ is much larger.

8 Further Comments

A call to nag_dgecon (f07agc) involves solving a number of systems of linear equations of the form
$Ax=b$ or $A^Tx=b$; the number is usually 4 or 5 and never more than 11. Each solution involves
approximately $2n^2$ floating-point operations but takes considerably longer than a call to nag_dgetrs
(f07aec) with 1 right-hand side, because extra care is taken to avoid overflow when $A$ is approximately
singular.

The complex analogue of this function is nag_zgecon (f07auc).

9 Example

To estimate the condition number in the 1-norm of the matrix $A$, where

$$A = \begin{pmatrix}
1.80 & 2.88 & 2.05 & -0.89 \\
5.25 & -2.95 & -0.95 & -3.80 \\
1.58 & -2.69 & -2.90 & -1.04 \\
-1.11 & -0.66 & -0.59 & 0.80
\end{pmatrix}$$

Here $A$ is nonsymmetric and must first be factorized by nag_dgetrf (f07adc). The true condition number
in the 1-norm is 152.16.

9.1 Program Text

/* nag_dgecon (f07agc) Example Program. */
/* Copyright 2001 Numerical Algorithms Group. */
/* Mark 7, 2001. */
/
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagf07.h>
#include <nagf16.h>
#include <nagx02.h>
#include <math.h>

int main(void)
{
    /* Scalars */
    double anorm, rcond;
    Integer exit_status=0;
    Integer i, ipiv_len, j, m, n, pda;
    NagError fail;
    Nag_OrderType order;
    
    /* Arrays */
    double *a=0;
    Integer *ipiv=0;

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

    // Rest of the program code...
#else
#define A(I,J) a[(I-1)\*pda + J - 1]

order = Nag_RowMajor;
#endif

INIT_FAIL(fail);
Vprintf("f07agc Example Program Results\n");
 /* Skip heading in data file */
Vscanf("%*[^
] ");
Vscanf("%ld%*[^
] ", &n);
pda = n;
m = n;
ipiv_len = n;

/* Allocate memory */
if ( !(a = NAG_ALLOC(n * n, double)) |
     !(ipiv = NAG_ALLOC(ipiv_len, Integer)) )
{
    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("%*[^
] ");
/* Compute norm of A */
f16rac(order, Nag_OneNorm, n, n, a, pda, &anorm, &fail);
if (fail.code != NE_NOERROR)
{
    Vprintf("Error from f16rac.\n\n", fail.message);
    exit_status = 1;
    goto END;
}
/* Factorize A */
f07adc(order, n, n, a, pda, ipiv, &fail);
if (fail.code != NE_NOERROR)
{
    Vprintf("Error from f07adc.\n\n", fail.message);
    exit_status = 1;
    goto END;
}
Vprintf("\n");
/* Estimate condition number */
f07agc(order, Nag_OneNorm, n, a, pda, anorm, &rcond, &fail);
if (fail.code != NE_NOERROR)
{
    Vprintf("Error from f07agc.\n\n", fail.message);
    exit_status = 1;
    goto END;
}
if (rcond >= X02AJC)
{
    Vprintf("Estimate of condition number =%10.2e\n", 1.0/rcond);
}
else
    Vprintf("A is singular to working precision\n");
END:
if (a) NAG_FREE(a);
if (ipiv) NAG_FREE(ipiv);
return exit_status;
}
9.2 Program Data

f07agc Example Program Data
4 : Value of N
 1.80  2.88  2.05 -0.89
 5.25 -2.95 -0.95 -3.80
 1.58 -2.69 -2.90 -1.04
-1.11 -0.66 -0.59  0.80 : End of matrix A

9.3 Program Results

f07agc Example Program Results

Estimate of condition number = 1.52e+02