NAG 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
#include <nag.h>
#include <nagf07.h>
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 \( \infty \)-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 Arguments

1: order – Nag_OrderType

On entry: the order argument 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 3.2.1.3 in the Essential Introduction for a more detailed explanation of the use of this argument.

Constraint: order = Nag_RowMajor or Nag_ColMajor.

2: norm – Nag_NormType

On entry: indicates whether \( \kappa_1(A) \) or \( \kappa_\infty(A) \) is estimated.

norm = Nag_OneNorm
\( \kappa_1(A) \) is estimated.
norm = Nag_InfNorm
κ∞(A) is estimated.

*Constraint:* norm = Nag_OneNorm or Nag_InfNorm.

3: n – Integer
*Input*

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

*Constraint:* n ≥ 0.

4: a[ dim ] – const double
*Input*

*Note:* the dimension, dim, of the array a must be at least max(1, pda × n).

The (i, j)th element of the matrix A is stored in

\[ a[(j - 1) \times pda + i - 1] \text{ when order} = \text{Nag ColMajor; } \]

\[ a[(i - 1) \times pda + j - 1] \text{ when order} = \text{Nag RowMajor.} \]

*On entry:* the LU factorization of A, as returned by nag_dgetrf (f07adc).

5: pda – Integer
*Input*

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

*Constraint:* pda ≥ max(1, n).

6: anorm – double
*Input*

*On entry:* if norm = Nag_OneNorm, the 1-norm of the original matrix A.

If norm = Nag_InfNorm, the ∞-norm of the original matrix A.

anorm may be computed by calling nag_dge_norm (f16rac) with the same value for the argument norm.

anorm must be computed either before calling nag_dgetrf (f07adc) or else from a copy of the original matrix A (see Section 10).

*Constraint:* anorm ≥ 0.0.

7: rcond – double *
*Output*

*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 *machine precision*, A is singular to working precision.

8: fail – NagError *
*Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

**NE_ALLOC_FAIL**
Dynamic memory allocation failed.

See Section 3.2.1.2 in the Essential Introduction for further information.

**NE_BAD_PARAM**

On entry, argument ⟨value⟩ had an illegal value.
NE_INT
On entry, \(n = \langle value\rangle\).
Constraint: \(n \geq 0\).
On entry, \(pda = \langle value\rangle\).
Constraint: \(pda > 0\).

NE_INT_2
On entry, \(pda = \langle value\rangle\) and \(n = \langle value\rangle\).
Constraint: \(pda \geq \max(1, n)\).

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 contact NAG for assistance.
An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE
Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

NE_REAL
On entry, \(anorm = \langle value\rangle\).
Constraint: \(anorm \geq 0.0\).

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 Parallelism and Performance
\(\text{nag_dgecon (f07agc)}\) is not threaded by NAG in any implementation.
\(\text{nag_dgecon (f07agc)}\) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.
Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this function. Please also consult the Users’ Note for your implementation for any additional implementation-specific information.

9 Further Comments
A call to \(\text{nag_dgecon (f07agc)}\) involves solving a number of systems of linear equations of the form \(Ax = b\) or \(A^T x = b\); 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 \(\text{nag_dgetrs (f07aec)}\) with one right-hand side, because extra care is taken to avoid overflow when \(A\) is approximately singular.
The complex analogue of this function is \(\text{nag_zgecon (f07auc)}\).
10  Example

This example estimates 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.

10.1  Program Text

```c
#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, n, pda;
    NagError fail;
    Nag_OrderType order;
    /* Arrays */
    double *a = 0;
    Integer *ipiv = 0;

    INIT_FAIL(fail);
    printf("nag_dgecon (f07agc) Example Program Results\n");
    /* Skip heading in data file */
    #ifdef _WIN32
    scanf_s("%*[\n] ");
    #else
    scanf("%*[\n] ");
    #endif
    #ifdef _WIN32
    scanf_s("%"NAG_IFMT"%*[\n] ", &n);
    #else
    scanf("%"NAG_IFMT"%*[\n] ", &n);
    #endif
    pda = n;
    ipiv_len = n;
```

/* Allocate memory */
if (!(a = NAG_ALLOC(n * n, double)) ||
    !(ipiv = NAG_ALLOC(ipiv_len, Integer)))
{
    printf("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)
#ifdef _WIN32
        scanf_s("%lf", &A(i, j));
#else
        scanf("%lf", &A(i, j));
#endif
#ifdef _WIN32
    scanf_s("%*[^
\] ");
#else
    scanf("%*[\n] ");
#endif
}

/* Compute norm of A */
/* nag_dge_norm (f16rac).
 * 1-norm, infinity-norm, Frobenius norm, largest absolute
 * element, real general matrix
 */
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_dge_norm (f16rac).
%s
", fail.message);
    exit_status = 1;
    goto END;
}

/* Factorize A */
/* nag_dgetrf (f07adc).
 * LU factorization of real m by n matrix
 */
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_dgetrf (f07adc).
%s
", fail.message);
    exit_status = 1;
    goto END;
}

/* Estimate condition number */
/* nag_dgecon (f07agc).
 * Estimate condition number of real matrix, matrix already
 * factorized by nag_dgetrf (f07adc)
 */
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_dgecon (f07agc).
%s
", fail.message);
    exit_status = 1;
    goto END;
}

/* nag_machine_precision (x02ajc).
 * The machine precision
 */
if (rcond >= nag_machine_precision)
{
    printf("Estimate of condition number =%11.2e\n", 1.0/rcond);
}
else
}


```c
    printf("A is singular to working precision\n");
END:
    NAG_FREE(a);
    NAG_FREE(ipiv);
    return exit_status;
}
```

### 10.2 Program Data

nag_dgecon (f07agc) Example Program Data

<table>
<thead>
<tr>
<th>Value of N</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

| 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

### 10.3 Program Results

nag_dgecon (f07agc) Example Program Results

Estimate of condition number = 1.52e+02