

NAG Library Routine Document

D02NUF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

1 Purpose

D02NUF is a setup routine which must be called prior to an integrator in Sub-chapter D02M–N, if sparse matrix linear algebra is required.

2 Specification

```

SUBROUTINE D02NUF (NEQ, NEQMAX, JCEVAL, NWKJAC, IA, NIA, JA, NJA,      &
                  JACPVT, NJCPVT, SENS, U, ETA, LBLOCK, ISPLIT, RWORK,  &
                  IFAIL)

INTEGER              NEQ, NEQMAX, NWKJAC, IA(NIA), NIA, JA(NJA), NJA,      &
                  JACPVT(NJCPVT), NJCPVT, ISPLIT, IFAIL
REAL (KIND=nag_wp)  SENS, U, ETA, RWORK(50+4*NEQMAX)
LOGICAL              LBLOCK
CHARACTER(1)        JCEVAL

```

3 Description

D02NUF defines the linear algebra to be used as sparse matrix linear algebra, permits you to specify the method for calculating the Jacobian and its structure, and checks the validity of certain input values.

4 References

See the D02M–N Sub-chapter Introduction.

5 Arguments

- 1: NEQ – INTEGER *Input*
On entry: the number of differential equations.
Constraint: $1 \leq \text{NEQ} \leq \text{NEQMAX}$.

- 2: NEQMAX – INTEGER *Input*
On entry: a bound on the maximum number of differential equations to be solved during the integration.
Constraint: $\text{NEQMAX} \geq \text{NEQ}$.

- 3: JCEVAL – CHARACTER(1) *Input*
On entry: specifies the technique to be used to compute the Jacobian.
 JCEVAL = 'N'
 The sparsity structure and the value of the Jacobian are to be determined numerically by the integrator.
 JCEVAL = 'S'
 The sparsity structure of the Jacobian is supplied in the arrays IA and JA but its value is to be determined numerically. This is the recommended mode of operation unless it is a simple matter to supply the Jacobian.

JCEVAL = 'A'

The Jacobian will be evaluated by calls to JAC. The sparsity structure will be estimated by calls to JAC; that is, no explicit sparsity structure need be supplied in the arrays IA and JA.

JCEVAL = 'F'

The sparsity structure of the Jacobian is supplied in IA and JA, and its value will be determined by calls to JAC. This is the recommended mode of operation if the JAC is simple to form.

JCEVAL = 'D'

The default choice is to be made. In this case 'D' is interpreted as 'S'.

If the sparsity structure is supplied in arrays IA and JA, then any evidence from the numerical or analytical formation of the Jacobian that this structure is not correct, is ignored.

Only the first character of the actual argument JCEVAL is passed to D02NUF; hence it is permissible for the actual argument to be more descriptive, e.g., 'Numerical', 'Structural', 'Analytical', 'Full information' or 'Default' in a call to D02NUF.

If the option JCEVAL = 'N', 'S' or 'D' is used then the actual argument corresponding to JAC in the call to D02NDF or D02NJF must be either D02NDZ or D02NJZ respectively.

If integration is to be performed by reverse communication (D02NMF or D02NNF) then JCEVAL should be set to either 'N' or 'A'. In this case IA and JA are not used and their lengths may be set to 1.

Constraint: JCEVAL = 'N', 'S', 'A', 'F' or 'D'.

4: NWKJAC – INTEGER *Input*

On entry: the size of the array WKJAC, which you are supplying to the integrator, as declared in the (sub)program from which D02NUF is called.

Suggested value: $NWKJAC = 4 \times NEQMAX$ if JCEVAL = 'N' or 'A'. If NWKJAC is less than this estimate, then a message is printed on the current advisory message unit (see X04ABF), and execution continues.

Constraint: if JCEVAL = 'S', 'F' or 'D', $NWKJAC \geq nelement + 2 \times NEQ$, where *nelement* is the total number of nonzeros.

5: IA(NIA) – INTEGER array *Input*

On entry: if JCEVAL = 'S', 'F' or 'D', IA must contain details of the sparsity pattern to be used for the Jacobian. See JA.

IA is not used if JCEVAL = 'N' or 'A'.

6: NIA – INTEGER *Input*

On entry: the dimension of the array IA as declared in the (sub)program from which D02NUF is called.

Constraints:

if JCEVAL = 'S', 'F' or 'D', $NIA \geq NEQ + 1$;
otherwise $NIA \geq 1$.

7: JA(NJA) – INTEGER array *Input*

On entry: if JCEVAL = 'S', 'F' or 'D', JA must contain details of the sparsity pattern to be used for the Jacobian. JA contains the row indices where nonzero elements occur, reading in column-wise order, and IA contains the starting locations in JA of the descriptions of columns 1, 2, ..., NEQ in that order, with $IA(1) = 1$. Thus for each column index $j = 1, 2, \dots, NEQ$, the values of the row index i in column j where a nonzero element may occur are given by

$$i = \text{JA}(k)$$

where $\text{IA}(j) \leq k < \text{IA}(j + 1)$.

Thus the total number of nonzeros, *nelement*, must be $\text{IA}(\text{NEQ} + 1) - 1$. For example, for the following matrix

$$\begin{pmatrix} x & 0 & x & 0 & 0 \\ 0 & x & x & x & 0 \\ x & x & x & 0 & 0 \\ x & 0 & 0 & x & x \\ 0 & 0 & 0 & x & x \end{pmatrix}$$

where x represents nonzero elements (13 in all) the arrays IA and JA should be

$$\begin{array}{r} \text{IA}(k) \quad 1 \quad 4 \quad 6 \quad 9 \quad 12 \quad 14 \\ \text{JA}(k) \quad 1 \quad 3 \quad 4 \quad 2 \quad 3 \quad 1 \quad 2 \quad 3 \quad 2 \quad 4 \quad 5 \quad 4 \quad 5 \end{array}$$

JA is not used if JCEVAL = 'N' or 'A'.

8: NJA – INTEGER *Input*

On entry: the dimension of the array JA as declared in the (sub)program from which D02NUF is called.

Constraints:

if JCEVAL = 'S', 'F' or 'D', $\text{NJA} \geq \text{IA}(\text{NEQ} + 1) - 1$;
otherwise $\text{NJA} \geq 1$.

9: JACPVT(NJCPVT) – INTEGER array *Communication Array*

On exit: data relating to the Jacobian sparsity structure.

10: NJCPVT – INTEGER *Input*

On entry: the length of the array JACPVT, which you are supplying to the integrator, as dimensioned in the sub(program) from which D02NUF is called.

Suggested value: $\text{NJCPVT} = 20 \times \text{NEQMAX}$ if JCEVAL = 'N' or 'A'. If NJCPVT is less than this estimate, then a message is printed on the current advisory message unit (see X04ABF), and execution continues.

Constraint: if JCEVAL = 'S', 'F' or 'D', $\text{NJCPVT} \geq 3 \times \text{nelement} + 14 \times \text{NEQ}$, where *nelement* is the total number of nonzeros.

11: SENS – REAL (KIND=nag_wp) *Input*

On entry: a threshold argument used to determine whether or not a matrix element is zero; when SENS is set to 0.0 on entry, the routine will use $\text{SENS} = 100.0 \times \text{machine precision}$. Otherwise the absolute value of SENS is used.

12: U – REAL (KIND=nag_wp) *Input*

On entry: should have a value between 0.0 and 0.9999. Otherwise a default value of 0.1 is used. When the sparsity pattern has been evaluated, the first Jacobian computed is decomposed with U governing the choice of pivots; subsequent Jacobian decompositions use the same pattern of decomposition until the sparsity pattern is re-evaluated. When searching a row for a pivot, any element is excluded from the search which is less than U times the largest of those elements in the row available as pivots. Thus decreasing U biases the algorithm towards maintaining sparsity at the expense of numerical stability.

- 13: ETA – REAL (KIND=nag_wp) *Input*
On entry: a relative pivot threshold, below which on subsequent decompositions (as described under U), an internal error is provoked.
 ETA > 1.0
 No check on pivot size is made.
 ETA ≤ 0.0
 The default value ETA = 1.0E−4 is used.
- 14: LBLOCK – LOGICAL *Input*
On entry: indicates if reordering is used before decomposition.
 If LBLOCK = .TRUE., on entry, the Jacobian matrix is preordered to block lower triangular form before a decomposition is performed (this is the recommended mode). If you know the structure of the Jacobian to be irreducible, that is not permutable to block lower triangular form, then you should set LBLOCK = .FALSE.. For example, a Jacobian arising from using the method of lines for parabolic partial differential equations would normally be irreducible. (See the specification of D02NXF for optional output concerning LBLOCK.)
- 15: ISPLIT – INTEGER *Input*
On entry: this argument is used for splitting the integer workspace JACPVT to effect an efficient decomposition. It must satisfy $1 \leq \text{ISPLIT} \leq 99$. If ISPLIT lies outside this range on entry, a default value of 73 is used. An appropriate value for ISPLIT for subsequent runs on similar problems is available via the optional output D02NXF.
Suggested value: ISPLIT = 73, unless you have information from a previous run of a similar problem.
- 16: RWORK(50 + 4 × NEQMAX) – REAL (KIND=nag_wp) array *Communication Array*
 This must be the same workspace array as the array RWORK supplied to the integrator. It is used to pass information from the setup routine to the integrator and therefore the contents of this array must not be changed before calling the integrator.
- 17: IFAIL – INTEGER *Input/Output*
On entry: IFAIL must be set to 0, −1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.
 For environments where it might be inappropriate to halt program execution when an error is detected, the value −1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value −1 or 1 is used it is essential to test the value of IFAIL on exit.**
On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL = 0 or −1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 1

 On entry, an illegal input was detected.

IFAIL = -99

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in *How to Use the NAG Library and its Documentation* for further information.

IFAIL = -399

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in *How to Use the NAG Library and its Documentation* for further information.

IFAIL = -999

Dynamic memory allocation failed.

See Section 3.7 in *How to Use the NAG Library and its Documentation* for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

D02NUF is not thread safe and should not be called from a multithreaded user program. Please see Section 3.12.1 in *How to Use the NAG Library and its Documentation* for more information on thread safety.

D02NUF is not threaded in any implementation.

9 Further Comments

D02NUF must be called as a setup routine before a call to either D02NDF or D02NJF and may be called as the linear algebra setup routine before a call to D02NMF or D02NNF.

10 Example

See Section 10 in D02NDF, D02NJF and D02NNF.
