D02NVF – NAG Fortran Library Routine Document

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

D02NVF is a setup routine which must be called by the user, prior to an integrator in the subchapter D02M–D02N, if Backward Differentiation Formulae (BDF) are to be used.

2 Specification

```
 SUBROUTINE D02NVF(NEQMAX, NY2DIM, MAXORD, METHOD, PETZLD, CONST,
1                  TCRIT, HMIN, HMAX, H0, MAXSTP, MXHNIL, NORM,
2                  RWORK, IFAIL)
 INTEGER           NEQMAX, NY2DIM, MAXORD, MAXSTP, MXHNIL, IFAIL
 real              CONST(6), TCRIT, HMIN, HMAX, H0,
1                  RWORK(50+4*NEQMAX)
 LOGICAL           PETZLD
 CHARACTER*1       METHOD, NORM
```

3 Description

An integrator setup routine must be called before the call to any integrator in this subchapter. The setup routine D02NVF makes the choice of the BDF integrator and permits the user to define options appropriate to this choice.

4 References

None.

5 Parameters

1: NEQMAX — INTEGER *Input*

On entry: a bound on the maximum number of differential equations to be solved.

Constraint: NEQMAX ≥ 1.

2: NY2DIM — INTEGER *Input*

On entry: the second dimension of the array YSAVE that will be supplied to the integrator, as declared in the (sub)program from which the integrator is called.

Constraint: MNY2DIM $\geq$ MAXORD + 1.

3: MAXORD — INTEGER *Input*

On entry: the maximum order to be used for the BDF method.

Constraint: $0 <$ MAXORD ≤ 5.

4: METHOD — CHARACTER*1 *Input*

On entry: specifies the method to be used to solve the system of nonlinear equations arising on each step of the BDF code. If METHOD = 'N', a modified Newton iteration is used. If METHOD = 'F', functional iteration is used. If METHOD = 'D', the modified Newton iteration is used.

Note. A linear algebra setup routine must be called even when using functional iteration, since if difficulty is encountered a switch is made to a modified Newton method.

Only the first character of the actual argument METHOD is passed to D02NVF; hence it is permissible for the actual argument to be more descriptive e.g., 'Newton', 'Functional iteration' or 'Default' in a call to D02NVF.

5: PETZLD — LOGICAL *Input*

On entry: specifies whether the Petzold local error test is to be used. If PETZLD is set to .TRUE. on entry, then the Petzold local error test is used, otherwise a conventional test is used. The Petzold test results in extra overhead cost but is more stable and reliable for differential/algebraic equations.

6: CONST(6) — ***real*** array *Input/Output*

On entry: values to be used to control step size choice during integration. If any CONST(i) = 0.0 on entry, it is replaced by its default value described below. In most cases this is the recommended setting.

CONST(1), CONST(2), and CONST(3) are factors used to bound step size changes. If the current step size h fails, then the modulus of the next step size is bounded by CONST(l)$\times|h|$. The default value of CONST(1) is 2.0. Note that the new step size may be used with a method of different order to the failed step. If the initial step size is h, then the modulus of the step size on the second step is bounded by CONST(3) $\times\ |h|$. At any other stage in the integration, if the current step size is h, then the modulus of the next step size is bounded by CONST(2) $\times\ |h|$. The default values are 10.0 for CONST(2) and 1000.0 for CONST(3).

CONST(4), CONST(5) and CONST(6) are 'tuning' constants used in determining the next order and step size. They are used to scale the error estimates used in determining whether to keep the same order of the BDF method, decrease the order or increase the order respectively. The larger the value of CONST(i), i = 4, 5, 6 the less likely the choice of the corresponding order. The default values are: CONST(4) = 1.2, CONST(5) = 1.3, CONST(6) = 1.4.

Constraints: the following constraints must be satisfied after any zero values have been replaced by their default values:

$$0.0 < \text{CONST}(1) < \text{CONST}(2) < \text{CONST}(3);$$
$$\text{CONST}(i) > 1.0, \text{ for } i = 2, 3, \ldots, 6.$$

On exit: the values actually used by the routine.

7: TCRIT — ***real*** *Input*

On entry: a point beyond which integration must not be attempted. The use of TCRIT is described under the parameter ITASK in the specification for the integrator. A value, 0.0 say, must be specified even if ITASK subsequently specifies that TCRIT will not be used.

8: HMIN — ***real*** *Input*

On entry: the minimum absolute step size to be allowed. Set HMIN = 0.0 if this option is not required.

9: HMAX — ***real*** *Input*

On entry: the maximum absolute step size to be allowed. Set HMAX = 0.0 if this option is not required.

10: H0 — ***real*** *Input*

On entry: the step size to be attempted on the first step. Set H0 = 0.0 if the initial step size is to be calculated internally.

11: MAXSTP — INTEGER *Input*

On entry: the maximum number of steps to be attempted during one call to the integrator after which it will return with IFAIL = 2. Set MAXSTP = 0 if no limit is to be imposed.

12: MXHNIL — INTEGER *Input*

On entry: the maximum number of warnings printed (if ITRACE ≥ 0) per problem when $t + h = t$ on a step ($h =$ current step size). If MXHNIL ≤ 0, a default value of 10 is assumed.

13: NORM — CHARACTER*1 *Input*

On entry: indicates the type of norm to be used. Three options are available:

'M'	maximum norm
'A'	averaged L2 norm.
'D'	is the same as 'A'

If VNORM denotes the norm of the vector v of length n, then for the averaged L2 norm

$$\text{VNORM} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(v_i/w_i)^2},$$

while for the maximum norm

$$\text{VNORM} = \max_i |v_i/w_i|.$$

If the user wishes to weight the maximum norm or the L2 norm, then RTOL and ATOL should be scaled appropriately on input to the integrator (see under ITOL in the specification of the integrator for the formulation of the weight vector w_i from RTOL and ATOL).

Only the first character to the actual argument NORM is passed to D02NVF; hence it is permissible for the actual argument to be more descriptive e.g., 'Maximum', 'Average L2' or 'Default' in a call to D02NVF.

Constraint: NORM = 'M', 'A' or 'D'.

14: RWORK(50+4∗NEQMAX) — ***real*** array *Workspace*

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.

15: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (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 detected by the routine:

IFAIL = 1

On entry, an illegal input was detected.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

See the example for Section 9 of the document for D02NBF.