NAG Library Routine Document

D02XKF

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

D02XKF interpolates components of the solution of a system of first-order ordinary differential equations from information provided by the integrators in Sub-chapter D02M–N. It provides C^1 interpolation suitable for general use.

2   Specification

```
SUBROUTINE D02XKF (XSOL, SOL, M, YSAV, LDYSAV, SDYSAV, ACOR, NEQ, X,       &
                   NQU, HU, H, IFAIL)
INTEGER           M, LDYSAV, SDYSAV, NEQ, NQU, IFAIL
REAL (KIND=nag_wp) XSOL, SOL(M), YSAV(LDYSAV,SDYSAV), ACOR(NEQ), X,        &
                   HU, H
```

3   Description

D02XKF evaluates the first m components of the solution of a system of ordinary differential equations at any point using C^1 polynomial interpolation based on information generated by the integrator. This information must be passed unchanged to D02XKF. D02XKF should not normally be used to extrapolate outside the range of values obtained from the above routines.

It may be used with the D02N routines only when the BDF integration method is being employed (setup routine D02NVF), provided the Petzold error test was not selected.

4   References

None.

5   Arguments

1:    XSOL – REAL (KIND=nag_wp)   *Input*

   On entry: the point at which the first m components of the solution are to be evaluated. XSOL should not be an extrapolation point, that is XSOL should satisfy $(\text{XSOL} - \text{X}) \times \text{HU} \le 0.0$. Extrapolation is permitted but not recommended.

2:    SOL(M) – REAL (KIND=nag_wp) array   *Output*

   On exit: the calculated value of the ith component of the solution at XSOL, for $i = 1, 2, \ldots, m$.

3:    M – INTEGER   *Input*

   On entry: the number of components of the solution whose values at XSOL are required. The first m components are evaluated.

   Constraint: $1 \le \text{M} \le \text{NEQ}$.

4:    YSAV(LDYSAV, SDYSAV) – REAL (KIND=nag_wp) array   *Input*

   On entry: the values provided in the argument YSAV on return from the integrator.

5:    LDYSAV – INTEGER   *Input*

      On entry: the value used for the argument LDYSAV when calling the integrator.

      Constraint: LDYSAV ≥ 1.

6:    SDYSAV – INTEGER   *Input*

      On entry: the value used for the argument SDYSAV when calling the integrator.

      Constraint: SDYSAV $\geq$ NQU $+ 1$.

7:    ACOR(NEQ) – REAL (KIND=nag_wp) array   *Input*

      On entry: the value returned in position (LDYSAV $+ 50 + i$), for $i = 1, 2, \ldots,$ NEQ, of the
      argument RWORK returned by the integrator. If one of the direct communication D02N routines
      is being employed and D02XKF is to be used in MONITR, then ACOR(i) must contain the value
      given in position $(i, 2)$ of the MONITR argument ACOR, for $i = 1, 2, \ldots,$ NEQ (e.g., see
      D02NBF).

8:    NEQ – INTEGER   *Input*

      On entry: the value used for the argument NEQ when calling the integrator.

      Constraint: $1 \leq$ NEQ $\leq$ LDYSAV.

9:    X – REAL (KIND=nag_wp)   *Input*

      On entry: the latest value at which the solution has been computed, as provided in the argument
      TCUR on return from the optional output D02NYF.

10:   NQU – INTEGER   *Input*

      On entry: the order of the method used up to the latest value at which the solution has been
      computed, as provided in the argument NQU on return from the optional output D02NYF.

      Constraint: NQU ≥ 1.

11:   HU – REAL (KIND=nag_wp)   *Input*

      On entry: the last successful step used, that is the step used in the integration to get to X, as
      provided in the argument HU on return from the optional output D02NYF.

12:   H – REAL (KIND=nag_wp)   *Input*

      On entry: the next step size to be attempted in the integration, as provided in the argument H on
      return from the optional output D02NYF.

13:   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).

      If D02XKF is to be used for extrapolation, IFAIL must be set to 1 before entry. It is then
      essential to test the value of IFAIL on exit for IFAIL $= 1$ or 2.

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, M < 1,
> or NEQ < 1,
> or LDYSAV < 1,
> or NEQ $>$ LDYSAV,
> or M $>$ NEQ,
> or NQU < 1,
> or SDYSAV $<$ NQU $+ 1$,
> or the BDF integrator was not previously used,
> or the Petzold error test, if applicable, was used.

IFAIL $= 2$

> On entry, HU $= 0.0$ or H $= 0.0$. This error can only occur if H and HU have been changed by you or possibly if the integrator has failed before calling D02XKF.

IFAIL $= 3$

> D02XKF has been called for extrapolation. Before returning with this error exit, the value of the solution at XSOL is calculated and placed in SOL.

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

The solution values returned will be of a similar accuracy to those computed by the integrator.

8 Parallelism and Performance

D02XKF 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.

D02XKF is not threaded in any implementation.

9 Further Comments

D02XKF provides a C^1 interpolant and as such is ideal for most applications, for example for tabulation and root-finding. In general D02XKF should be preferred to D02XJF for interpolation as the latter provides only a C^0 interpolant. D02XJF is the natural interpolant employed by the BDF method and it is supplied only to permit you to reproduce the internal values used by the integrator.

10 Example

See Section 10 in D02NDF and D02NMF.