D02PDF – 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

D02PDF is a one-step routine for solving the initial value problem for a first-order system of ordinary differential equations using Runge–Kutta methods.

2 Specification

```
SUBROUTINE D02PDF(F, TNOW, YNOW, YPNOW, WORK, IFAIL)
INTEGER           IFAIL
real              TNOW, YNOW(*), YPNOW(*), WORK(*)
EXTERNAL          F
```

3 Description

D02PDF and its associated routines (D02PVF, D02PWF, D02PXF, D02PYF, D02PZF) solve the initial value problem for a first-order system of ordinary differential equations. The routines, based on Runge–Kutta methods and derived from RKSUITE [1], integrate

$$y' = f(t, y) \text{ given } y(t_0) = y_0$$

where y is the vector of n solution components and t is the independent variable.

D02PDF is designed to be used in complicated tasks when solving systems of ordinary differential equations. You must first call D02PVF to specify the problem and how it is to be solved. Thereafter you (repeatedly) call D02PDF to take one integration step at a time from TSTART in the direction of TEND (as specified in D02PVF). In this manner D02PDF returns an approximation to the solution YNOW and its derivative YPNOW at successive points TNOW. If D02PDF encounters some difficulty in taking a step, the integration is not advanced and the routine returns with the same values of TNOW, YNOW and YPNOW as returned on the previous successful step. D02PDF tries to advance the integration as far as possible subject to passing the test on the local error and not going past TEND. In the call to D02PVF you can specify either the first step size for D02PDF to attempt or that it compute automatically an appropriate value. Thereafter D02PDF estimates an appropriate step size for its next step. This value and other details of the integration can be obtained after any call to D02PDF by a call to D02PYF. The local error is controlled at every step as specified in D02PVF. If you wish to assess the true error, you must set ERRASS = .TRUE. in the call to D02PVF. This assessment can be obtained after any call to D02PDF by a call to D02PZF.

If you want answers at specific points there are two ways to proceed:

(1) The more efficient way is to step past the point where a solution is desired, and then call D02PXF to get an answer there. Within the span of the current step, you can get all the answers you want at very little cost by repeated calls to D02PXF. This is very valuable when you want to find where something happens, e.g.,, where a particular solution component vanishes. You cannot proceed in this way with METHOD = 3.

(2) The other way to get an answer at a specific point is to set TEND to this value and integrate to TEND. D02PDF will not step past TEND, so when a step would carry it past, it will reduce the step size so as to produce an answer at TEND exactly. After getting an answer there (TNOW = TEND), you can reset TEND to the next point where you want an answer, and repeat. TEND could be reset by a call to D02PVF, but you should not do this. You should use D02PWF instead because it is both easier to use and much more efficient. This way of getting answers at specific points can be used with any of the available methods, but it is the only way with METHOD = 3. It can be inefficient. Should this be the case, the code will bring the matter to your attention.

4 References

[1] Brankin R W, Gladwell I and Shampine L F (1991) RKSUITE: A suite of Runge–Kutta codes for the initial value problems for ODEs *SoftReport 91–S1* Southern Methodist University

5 Parameters

1: F — SUBROUTINE, supplied by the user. *External Procedure*

F must evaluate the functions f_i (that is the first derivatives y_i') for given values of the arguments t, y_i.

Its specification is:

```
SUBROUTINE F(T, Y, YP)
real          T, Y(*), YP(*)
```

1: T — ***real*** *Input*
On entry: the current value of the independent variable, t.

2: Y(∗) — ***real*** array *Input*
On entry: the current values of the dependent variables, y_i for $i = 1, 2, \ldots, n$.

3: YP(∗) — ***real*** array *Output*
On exit: the values of f_i for $i = 1, 2, \ldots, n$.

F must be declared as EXTERNAL in the (sub)program from which D02PDF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

2: TNOW — ***real*** *Output*
On exit: the value of the independent variable t at which a solution has been computed.

3: YNOW(∗) — ***real*** array *Output*
Note. Normal the dimension of YNOW must be at least n.
On exit: an approximation to the solution at TNOW. The local error of the step to TNOW was no greater than permitted by the specified tolerances (see D02PVF).

4: YPNOW(∗) — ***real*** array *Output*
Note. Normal the dimension of YPNOW must be at least n.
On exit: an approximation to the derivative of the solution at TNOW.

5: WORK(∗) — ***real*** array *Input/Output*
On entry: this **must** be the same array as supplied to D02PVF. It **must** remain unchanged between calls.
On exit: information about the integration for use on subsequent calls to D02PDF or other associated routines.

6: IFAIL — INTEGER *Input/Output*
On entry: IFAIL must be set to 0, −1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.
On exit: IFAIL = 0 unless the routine detects an error or gives a warning (see Section 6).

For this routine, because the values of output parameters may be useful even if IFAIL $\neq 0$ on exit, users are recommended to set IFAIL to −1 before entry. **It is then essential to test the value of IFAIL on exit**.

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 specified by the routine:

IFAIL = 1

On entry, an invalid call to D02PDF was made, for example without a previous call to the setup routine D02PVF. If on entry IFAIL = 0 or −1, the precise form of the error will be detailed on the current error message unit (as defined by X04AAF). You cannot continue integrating the problem.

IFAIL = 2

D02PDF is being used inefficiently because the step size has been reduced drastically many times to obtain answers at many points TEND. If you really need the solution at this many points, you should use D02PXF to obtain the answers inexpensively. If you need to change from METHOD = 3 to do this, restart the integration from TNOW, YNOW by a call to D02PVF. If you wish to continue as before, call D02PDF again. The monitor of this kind of inefficiency will be reset automatically so that the integration can proceed.

IFAIL = 3

A considerable amount of work has been expended in the (primary) integration. This is measured by counting the number of calls to the subroutine F. At least 5000 calls have been made since the last time this counter was reset. Calls to F in a secondary integration for global error assessment (when ERRASS = .TRUE. in the call to D02PVF) are not counted in this total. The integration was interrupted. If you wish to continue on towards TEND, just call D02PDF again. The counter measuring work will be reset to zero automatically.

IFAIL = 4

It appears that this problem is stiff. The methods implemented in D02PDF can solve such problems, but they are inefficient. You should change to another code based on methods appropriate for stiff problems. The integration was interrupted. If you want to continue on towards TEND, just call D02PDF again. The stiffness monitor will be reset automatically.

IFAIL = 5

It does not appear possible to achieve the accuracy specified by TOL and THRES in the call to D02PVF with the precision available on the computer being used and with this value of METHOD. You cannot continue integrating this problem. A larger value for METHOD, if possible, will permit greater accuracy with this precision. To increase METHOD and/or continue with larger values of TOL and/or THRES, restart the integration from TNOW, YNOW by a call to D02PVF.

IFAIL = 6

(This error exit can only occur if ERRASS = .TRUE. in the call to D02PVF.) The global error assessment may not be reliable beyond the current integration point TNOW. This may occur because either too little or too much accuracy has been requested or because $f(t, y)$ is not smooth enough for values of t just beyond TNOW and current values of the solution y. The integration cannot be continued. This return does not mean that you cannot integrate past TGOT, rather that you cannot do it with ERRASS = .TRUE.. However, it may also indicate problems with the primary integration.

7 Accuracy

The accuracy of integration is determined by the parameters TOL and THRES in a prior call to D02PVF. Note that only the local error at each step is controlled by these parameters. The error estimates obtained are not strict bounds but are usually reliable over one step. Over a number of steps the overall error may accumulate in various ways, depending on the properties of the differential system.

8 Further Comments

If D02PDF returns with IFAIL = 5 and the accuracy specified by TOL and THRES is really required then you should consider whether there is a more fundamental difficulty. For example, the solution may contain a singularity. In such a region the solution components will usually be large in magnitude. Successive output values of YNOW should be monitored with the aim of trapping the solution before the singularity. In any case numerical integration cannot be continued through a singularity, and analytical treatment may be necessary.

Performance statistics are available after any return from D02PDF (except when IFAIL = 1) by a call to D02PYF. If ERRASS = .TRUE. in the call to D02PVF, global error assessment is available after any return from D02PDF (except when IFAIL = 1) by a call to D02PZF.

After a failure with IFAIL = 5 or 6 the diagnostic routines D02PYF and D02PZF may be called only once.

If D02PDF returns with IFAIL = 4 then it is advisable to change to another code more suited to the solution of stiff problems. D02PDF will not return with IFAIL = 4 if the problem is actually stiff but it is estimated that integration can be completed using less function evaluations than already computed.

9 Example

We solve the equation

$$y'' = -y, \quad y(0) = 0, y'(0) = 1$$

reposed as

$$y_1' = y_2 \;\; y_2' = -y_1$$

over the range $[0, 2\pi]$ with initial conditions $y_1 = 0.0$ and $y_2 = 1.0$. We use relative error control with threshold values of 1.0E−8 for each solution component and print the solution at each integration step across the range. We use a medium order Runge–Kutta method (METHOD = 2) with tolerances TOL = 1.0E−4 and TOL = 1.0E−5 in turn so that we may compare the solutions. The value of π is obtained by using X01AAF.

Note that the length of WORK is large enough for any valid combination of input arguments to D02PVF.

See also the example programs for D02PWF and D02PXF.

9.1 Program Text

Note. The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     D02PDF Example Program Text
*     Mark 17 Revised.  NAG Copyright 1995.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      INTEGER          NEQ, LENWRK, METHOD
      PARAMETER        (NEQ=2,LENWRK=32*NEQ,METHOD=2)
      real             ZERO, ONE, TWO
      PARAMETER        (ZERO=0.0e0,ONE=1.0e0,TWO=2.0e0)
*     .. Local Scalars ..
      real             HNEXT, HSTART, PI, TEND, TNOW, TOL, TSTART, WASTE
      INTEGER          I, IFAIL, L, STPCST, STPSOK, TOTF
      LOGICAL          ERRASS
*     .. Local Arrays ..
      real             THRES(NEQ), WORK(LENWRK), YNOW(NEQ), YPNOW(NEQ),
     +                 YSTART(NEQ)
```

```
*     .. External Functions ..
      real             X01AAF
      EXTERNAL         X01AAF
*     .. External Subroutines ..
      EXTERNAL         D02PDF, D02PVF, D02PYF, F
*     .. Executable Statements ..
      WRITE (NOUT,*) 'D02PDF Example Program Results'
*
*  Set initial conditions and input for D02PVF
*
      PI = X01AAF(ZERO)
      TSTART = ZERO
      YSTART(1) = ZERO
      YSTART(2) = ONE
      TEND = TWO*PI
      DO 20 L = 1, NEQ
         THRES(L) = 1.0e-8
   20 CONTINUE
      ERRASS = .FALSE.
      HSTART = ZERO
*
      DO 60 I = 1, 2
         IF (I.EQ.1) TOL = 1.0e-4
         IF (I.EQ.2) TOL = 1.0e-5
         IFAIL = 0
         CALL D02PVF(NEQ,TSTART,YSTART,TEND,TOL,THRES,METHOD,
     +               'Complex Task',ERRASS,HSTART,WORK,LENWRK,IFAIL)
*
         WRITE (NOUT,'(/A,D8.1)') ' Calculation with TOL = ', TOL
         WRITE (NOUT,'(/A/)') '    t          y1          y2'
         WRITE (NOUT,'(1X,F6.3,2(3X,F8.4))') TSTART, (YSTART(L),L=1,NEQ)
*
   40    CONTINUE
         IFAIL = -1
         CALL D02PDF(F,TNOW,YNOW,YPNOW,WORK,IFAIL)
*
         IF (IFAIL.EQ.0) THEN
            WRITE (NOUT,'(1X,F6.3,2(3X,F8.4))') TNOW, (YNOW(L),L=1,NEQ)
            IF (TNOW.LT.TEND) GO TO 40
         END IF
*
         IFAIL = 0
         CALL D02PYF(TOTF,STPCST,WASTE,STPSOK,HNEXT,IFAIL)
         WRITE (NOUT,'(/A,I6)')
     +     ' Cost of the integration in evaluations of F is', TOTF
*
   60 CONTINUE
*
      STOP
      END
      SUBROUTINE F(T,Y,YP)
*     .. Scalar Arguments ..
      real         T
*     .. Array Arguments ..
      real         Y(*), YP(*)
*     .. Executable Statements ..
      YP(1) = Y(2)
```

```
      YP(2) = -Y(1)
      RETURN
      END
```

9.2 Program Data

None.

9.3 Program Results

```
D02PDF Example Program Results

Calculation with TOL =  0.1E-03

   t          y1          y2

 0.000     0.0000     1.0000
 0.785     0.7071     0.7071
 1.519     0.9987     0.0513
 2.282     0.7573    -0.6531
 2.911     0.2285    -0.9735
 3.706    -0.5348    -0.8450
 4.364    -0.9399    -0.3414
 5.320    -0.8209     0.5710
 5.802    -0.4631     0.8863
 6.283     0.0000     1.0000

Cost of the integration in evaluations of F is    78

Calculation with TOL =  0.1E-04

   t          y1          y2

 0.000     0.0000     1.0000
 0.393     0.3827     0.9239
 0.785     0.7071     0.7071
 1.416     0.9881     0.1538
 1.870     0.9557    -0.2943
 2.204     0.8062    -0.5916
 2.761     0.3711    -0.9286
 3.230    -0.0880    -0.9961
 3.587    -0.4304    -0.9026
 4.022    -0.7710    -0.6368
 4.641    -0.9974    -0.0717
 5.152    -0.9049     0.4256
 5.521    -0.6903     0.7235
 5.902    -0.3718     0.9283
 6.283     0.0000     1.0000

Cost of the integration in evaluations of F is   118
```