

NAG Library Routine Document

D02PWF

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

D02PWF resets the end point in an integration performed by D02PDF.

2 Specification

```
SUBROUTINE D02PWF(TENDNU, IFAIL)
INTEGER          IFAIL
double precision TENDNU
```

3 Description

D02PWF and its associated routines (D02PDF, D02PVF, D02PXF, D02PYF and 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 (see Brankin *et al.* (1991)), integrate

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

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

D02PWF is used to reset the final value of the independent variable, t_f , when the integration is already underway. It can be used to extend or reduce the range of integration. The new value must be beyond the current value of the independent variable (as returned in TNOW by D02PDF) in the current direction of integration. It is much more efficient to use D02PWF for this purpose than to use D02PVF which involves the overhead of a complete restart of the integration.

If you want to change the direction of integration then you must restart by a call to D02PVF.

4 References

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: TENDNU – **double precision** *Input*
On entry: the new value for t_f .

Constraint: $\text{sign}(\text{TENDNU} - \text{TNOW}) = \text{sign}(\text{TEND} - \text{TSTART})$, where TSTART and TEND are as supplied in the previous call to D02PVF and TNOW is returned by the preceding call to D02PDF. TENDNU must be distinguishable from TNOW for the method and the ***machine precision*** being used.

2: IFAIL – INTEGER *Input/Output*
On entry: IFAIL must be set to 0, –1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.
On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

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 parameter, the recommended value is 0. **When the value -1 or 1 is used it is 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 detected by the routine:

IFAIL = 1

On entry, an invalid input value for TENDNU was detected or an invalid call to D02PWF was made, for example without a previous call to the integration routine D02PDF. You cannot continue integrating the problem.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

This example integrates a two body problem. The equations for the co-ordinates $(x(t), y(t))$ of one body as functions of time t in a suitable frame of reference are

$$x'' = -\frac{x}{r^3}$$

$$y'' = -\frac{y}{r^3}, \quad r = \sqrt{x^2 + y^2}.$$

The initial conditions

$$x(0) = 1 - \epsilon, \quad x'(0) = 0$$

$$y(0) = 0, \quad y'(0) = \sqrt{\frac{1 + \epsilon}{1 - \epsilon}}$$

lead to elliptic motion with $0 < \epsilon < 1$. $\epsilon = 0.7$ is selected and reposed as

$$y'_1 = y_3$$

$$y'_2 = y_4$$

$$y'_3 = -\frac{y_1}{r^3}$$

$$y'_4 = -\frac{y_2}{r^3}$$

over the range $[0, 6\pi]$. Relative error control is used with threshold values of $1.0\text{D}-10$ for each solution component and compute the solution at intervals of length π across the range using D02PWF to reset the end of the integration range. A high-order Runge-Kutta method (METHOD = 3) is also used with tolerances TOL = $1.0\text{D}-4$ and TOL = $1.0\text{D}-5$ in turn so that the solutions may be compared. The value of π is obtained by using X01AAF.

Note that the length of TOL = $1.0\text{D}-4$ and WORK is large enough for any valid combination of input arguments to D02PVF.

9.1 Program Text

```

*      D02PWF Example Program Text
*      Mark 17 Revised. NAG Copyright 1995.
*      .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      INTEGER          NEQ, LENWRK, METHOD
      PARAMETER        (NEQ=4,LENWRK=32*NEQ,METHOD=3)
      DOUBLE PRECISION ZERO, ONE, SIX, ECC
      PARAMETER        (ZERO=0.0D0,ONE=1.0D0,SIX=6.0D0,ECC=0.7D0)
*      .. Local Scalars ..
      DOUBLE PRECISION HNEXT, HSTART, PI, TENDNU, TFINAL, TINC, TNOW,
+      TOL, TSTART, WASTE
      INTEGER          I, IFAIL, J, L, NPTS, STPCST, STPSOK, TOTF
      LOGICAL          ERRASS
*      .. Local Arrays ..
      DOUBLE PRECISION THRES(NEQ), WORK(LENWRK), YNOW(NEQ), YPNOW(NEQ),
+      YSTART(NEQ)
*      .. External Functions ..
      DOUBLE PRECISION X01AAF
      EXTERNAL         X01AAF
*      .. External Subroutines ..
      EXTERNAL         D02PDF, D02PVF, D02PWF, D02PYF, F
*      .. Intrinsic Functions ..
      INTRINSIC        SQRT
*      .. Executable Statements ..
      WRITE (NOUT,*) 'D02PWF Example Program Results'

*
*      Set initial conditions and input for D02PVF
*
      PI = X01AAF(ZERO)
      TSTART = ZERO
      YSTART(1) = ONE - ECC
      YSTART(2) = ZERO
      YSTART(3) = ZERO
      YSTART(4) = SQRT((ONE+ECC)/(ONE-ECC))
      TFINAL = SIX*PI
      DO 20 L = 1, NEQ
          THRES(L) = 1.0D-10
20  CONTINUE
      ERRASS = .FALSE.
      HSTART = ZERO

*
*      Set output control
*
      NPTS = 6
      TINC = TFINAL/NPTS

*
      DO 60 I = 1, 2
          IF (I.EQ.1) TOL = 1.0D-4
          IF (I.EQ.2) TOL = 1.0D-5
          J = NPTS - 1
          TENDNU = TFINAL - J*TINC
          IFAIL = 1
          CALL D02PVF(NEQ,TSTART,YSTART,TENDNU,TOL,THRES,METHOD,
+      'Complex Task',ERRASS,HSTART,WORK,LENWRK,IFAIL)
*
          IF (IFAIL.NE.0) THEN
              WRITE (NOUT,99999) IFAIL
              GO TO 80
          END IF
          WRITE (NOUT,'(A,E8.1)') ' Calculation with TOL = ', TOL
          WRITE (NOUT,'(A/)') ' t          y1          y2'//
+      '          y3          y4'
          WRITE (NOUT,'(1X,F6.3,4(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
        IF (TNOW.LT.TENDNU) GO TO 40
        WRITE (NOUT,'(1X,F6.3,4(3X,F8.4))') TNOW, (YNOW(L),L=1,NEQ)
        IF (TNOW.LT.TFINAL) THEN
          J = J - 1
          TENDNU = TFINAL - J*TINC
          CALL D02PWF(TENDNU,IFAIL)
          GO TO 40
        END IF
      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
      80 CONTINUE
*
999999 FORMAT (1X,/1X,' ** D02PVF returned with IFAIL = ',I5)
      END
      SUBROUTINE F(T,Y,YP)
*      .. Scalar Arguments ..
      DOUBLE PRECISION T
*      .. Array Arguments ..
      DOUBLE PRECISION Y(*), YP(*)
*      .. Local Scalars ..
      DOUBLE PRECISION R
*      .. Intrinsic Functions ..
      INTRINSIC      SQRT
*      .. Executable Statements ..
      R = SQRT(Y(1)**2+Y(2)**2)
      YP(1) = Y(3)
      YP(2) = Y(4)
      YP(3) = -Y(1)/R**3
      YP(4) = -Y(2)/R**3
      RETURN
      END

```

9.2 Program Data

None.

9.3 Program Results

D02PWF Example Program Results

Calculation with TOL = 0.1E-03

t	y1	y2	y3	y4
0.000	0.3000	0.0000	0.0000	2.3805
3.142	-1.7000	0.0000	0.0000	-0.4201
6.283	0.3000	0.0000	0.0001	2.3805
9.425	-1.7000	0.0000	0.0000	-0.4201
12.566	0.3000	-0.0003	0.0016	2.3805
15.708	-1.7001	0.0001	-0.0001	-0.4201
18.850	0.3000	-0.0010	0.0045	2.3805

Cost of the integration in evaluations of F is 571

Calculation with TOL = 0.1E-04

t	y1	y2	y3	y4
0.000	0.3000	0.0000	0.0000	2.3805
3.142	-1.7000	0.0000	0.0000	-0.4201
6.283	0.3000	0.0000	0.0000	2.3805
9.425	-1.7000	0.0000	0.0000	-0.4201

12.566	0.3000	-0.0001	0.0004	2.3805
15.708	-1.7000	0.0000	0.0000	-0.4201
18.850	0.3000	-0.0003	0.0012	2.3805

Cost of the integration in evaluations of F is 748

