

## D02PWF – 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

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

### 2 Specification

```
SUBROUTINE D02PWF(TENDNU, IFAIL)
  INTEGER          IFAIL
  real            TENDNU
```

### 3 Description

D02PWF and its associated routines (D02PVF, D02PDF, 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.

D02PWF is used to reset the 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

- [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: TENDNU — *real* *Input*

*On entry:* the new value for  $t_f$ .

*Constraints:*  $\text{sign}(\text{TENDNU} - \text{TNOW}) = \text{sign}(\text{TEND} - \text{TSTART})$ , 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 precision of the machine being used.

2: 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 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. 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.

## 7 Accuracy

Not applicable.

## 8 Further Comments

None.

## 9 Example

We integrate a two body problem. The equations for the coordinates  $(x(t), y(t))$  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

$$\begin{aligned} x(0) &= 1 - \epsilon, & x'(0) &= 0 \\ y(0) &= 0, & y'(0) &= \sqrt{\frac{1 + \epsilon}{1 - \epsilon}} \end{aligned}$$

lead to elliptic motion with  $0 < \epsilon < 1$ . We select  $\epsilon = 0.7$  and repose 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  $[0, 6\pi]$ . We use relative error control with threshold values of  $1.0\text{E}-10$  for each solution component and compute the solution at intervals of length  $\pi$  across the range using D02PWF to reset the end of the integration range. We use a high order Runge-Kutta method (METHOD = 3) with tolerances TOL =  $1.0\text{E}-4$  and TOL =  $1.0\text{E}-5$  in turn so that we may compare the solutions. The value of  $\pi$  is obtained by using X01AAF.

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

## 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.

```

*   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)
real          ZERO, ONE, SIX, ECC
PARAMETER       (ZERO=0.0e0,ONE=1.0e0,SIX=6.0e0,ECC=0.7e0)
*   .. Local Scalars ..
real         HNEXT, HSTART, PI, TEND, TFINAL, TINC, TNOW, TOL,
+             TSTART, WASTE
INTEGER         I, IFAIL, J, L, NPTS, 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, 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) = 1.0e-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 = 1.0e-4
        IF (I.EQ.2) TOL = 1.0e-5
        J = NPTS - 1
        TEND = TFINAL - J*TINC
        IFAIL = 0

```

```

CALL DO2PVF(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'//
+   '      y3      y4'
WRITE (NOUT,'(1X,F6.3,4(3X,F8.4))') TSTART, (YSTART(L),L=1,NEQ)
*
40  CONTINUE
    IFAIL = -1
    CALL DO2PDF(F,TNOW,YNOW,YPNOW,WORK,IFAIL)
*
    IF (IFAIL.EQ.0) THEN
        IF (TNOW.LT.TEND) 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
            TEND = TFINAL - J*TINC
            CALL DO2PWF(TEND,IFAIL)
            GO TO 40
        END IF
    END IF
*
    IFAIL = 0
    CALL DO2PYF(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(*)
*   .. Local Scalars ..
  real    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

---