D02PZF – 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

D02PZF provides details about global error assessment computed during an integration with either D02PCF or D02PDF.

2 Specification

```
SUBROUTINE D02PZF(RMSERR, ERRMAX, TERRMX, WORK, IFAIL)
INTEGER             IFAIL
real                RMSERR(*), ERRMAX, TERRMX, WORK(*)
```

3 Description

D02PZF and its associated routines (D02PCF, D02PDF, D02PVF, D02PWF, D02PXF, D02PYF) 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.

After a call to D02PCF or D02PDF, D02PZF can be called for information about error assessment, if this assessment was specified in the setup routine D02PVF. A more accurate 'true' solution $\hat{y}$ is computed in a secondary integration. The error is measured as specified in D02PVF for local error control. At each step in the primary integration, an average magnitude μ_i of component y_i is computed, and the error in the component is

$$\frac{|y_i - \hat{y}_i|}{\max(\mu_i, \mathrm{THRES}(i))}.$$

It is difficult to estimate reliably the true error at a single point. For this reason the RMS (root-mean-square) average of the estimated global error in each solution component is computed. This average is taken over all steps from the beginning of the integration through to the current integration point. If all has gone well, the average errors reported will be comparable to TOL (see D02PVF). The maximum error seen in any component in the integration so far and the point where the maximum error first occurred are also reported.

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: RMSERR(∗) — ***real*** array *Output*

Note: the dimension of the array RMSERR must be at least n.

On exit: RMSERR(i) approximates the RMS average of the true error of the numerical solution for the ith solution component, for $i = 1, 2, \ldots, n$. The average is taken over all steps from the beginning of the integration to the current integration point.

2: ERRMAX — ***real*** *Output*

On exit: the maximum weighted approximate true error taken over all solution components and all steps.

3: TERRMX — ***real*** *Output*

On exit: the first value of the independent variable where an approximate true error attains the maximum value, ERRMAX.

4: WORK($*$) — ***real*** array *Input*

On entry: this **must** be the same array as supplied to D02PCF or D02PDF and **must** remain unchanged between calls.

5: 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 $= 1$

An invalid call to D02PZF has been made, for example without a previous call to D02PCF or D02PDF, or without error assessment having been specified in a call to 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.

7 Accuracy

Not applicable.

8 Further Comments

If the integration has proceeded 'well' and the problem is smooth enough, stable and not too difficult then the values returned in the arguments RMSERR and ERRMAX should be comparable to the value of TOL specified in the prior call to D02PVF.

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 $[0, 3\pi]$. We use relative error control with threshold values of 1.0E−10 for each solution component and a high order Runge–Kutta method (METHOD = 3) with tolerance TOL = 1.0E−6. 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. Note also, for illustration purposes since it is not necessary for this problem, we choose to integrate to the end of the range regardless of efficiency concerns (i.e., returns from D02PCF with IFAIL = 2, 3, 4).

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.

```
*     D02PZF 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, THREE, ECC
      PARAMETER        (ZERO=0.0e0,ONE=1.0e0,THREE=3.0e0,ECC=0.7e0)
*     .. Local Scalars ..
      real             ERRMAX, HNEXT, HSTART, PI, TEND, TERRMX, TGOT,
     +                 TOL, TSTART, TWANT, WASTE
      INTEGER          IFAIL, L, STPCST, STPSOK, TOTF
      LOGICAL          ERRASS
*     .. Local Arrays ..
      real             RMSERR(NEQ), THRES(NEQ), WORK(LENWRK), YGOT(NEQ),
     +                 YMAX(NEQ), YPGOT(NEQ), YSTART(NEQ)
*     .. External Functions ..
      real             X01AAF
      EXTERNAL         X01AAF
*     .. External Subroutines ..
      EXTERNAL         D02PCF, D02PVF, D02PYF, D02PZF, F
*     .. Intrinsic Functions ..
      INTRINSIC        SQRT
*     .. Executable Statements ..
      WRITE (NOUT,*) 'D02PZF 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))
```

```
      TEND = THREE*PI
      DO 20 L = 1, NEQ
         THRES(L) = 1.0e-10
   20 CONTINUE
      ERRASS = .TRUE.
      HSTART = ZERO
      TOL = 1.0e-6
      IFAIL = 0
      CALL D02PVF(NEQ,TSTART,YSTART,TEND,TOL,THRES,METHOD,'Usual 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)
*
      TWANT = TEND
*
   40 CONTINUE
      IFAIL = 1
      CALL D02PCF(F,TWANT,TGOT,YGOT,YPGOT,YMAX,WORK,IFAIL)
*
      IF (IFAIL.GE.2 .AND. IFAIL.LE.4) THEN
         GO TO 40
      ELSE IF (IFAIL.NE.0) THEN
         WRITE (NOUT,'(A,I2)') ' D02PCF returned with IFAIL set to',
     +     IFAIL
      ELSE
         WRITE (NOUT,'(1X,F6.3,4(3X,F8.4))') TGOT, (YGOT(L),L=1,NEQ)
*
         IFAIL = 0
         CALL D02PZF(RMSERR,ERRMAX,TERRMX,WORK,IFAIL)
         WRITE (NOUT,'(/A/9X,4(2X,E9.2))')
     +     ' Componentwise error '//'assessment', (RMSERR(L),L=1,NEQ)
         WRITE (NOUT,'(/A,E9.2,A,F6.3)')
     +     ' Worst global error observed '//'was ', ERRMAX,
     +     ' - it occurred at T = ', TERRMX
*
         IFAIL = 0
         CALL D02PYF(TOTF,STPCST,WASTE,STPSOK,HNEXT,IFAIL)
         WRITE (NOUT,'(/A,I6)')
     +     ' Cost of the integration in evaluations of F is', TOTF
      END IF
*
      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

```
 D02PZF Example Program Results

 Calculation with TOL =  0.1E-05

    t         y1         y2         y3         y4

  0.000     0.3000     0.0000     0.0000     2.3805
  9.425    -1.7000     0.0000     0.0000    -0.4201

 Componentwise error assessment
           0.38E-05   0.71E-05   0.69E-05   0.21E-05

 Worst global error observed was  0.34E-04 - it occurred at T =  6.302

 Cost of the integration in evaluations of F is  1361
```