

# NAG Library Routine Document

## D02PTF

**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

D02PTF provides details about an integration performed by either D02PEF or D02PFF.

### 2 Specification

```
SUBROUTINE D02PTF (FEVALS, STEPCOST, WASTE, STEPSOK, HNEXT, IWSAV,           &
                  RWSAV, IFAIL)
INTEGER          FEVALS, STEPCOST, STEPSOK, IWSAV(130), IFAIL
REAL (KIND=nag_wp) WASTE, HNEXT, RWSAV(350)
```

### 3 Description

D02PTF and its associated routines (D02PEF, D02PFF, D02PQF, D02PRF, D02PSF and D02PUF) 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.

After a call to D02PEF or D02PFF, D02PTF can be called to obtain information about the cost of the integration and the size of the next step.

### 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: FEVALS – INTEGER *Output*  
*On exit:* the total number of evaluations of  $f$  used in the integration so far; this includes evaluations of  $f$  required for the secondary integration necessary if D02PQF had previously been called with METHOD > 0.
- 2: STEPCOST – INTEGER *Output*  
*On exit:* the cost in terms of number of evaluations of  $f$  of a typical step with the method being used for the integration. The method is specified by the parameter METHOD in a prior call to D02PQF.
- 3: WASTE – REAL (KIND=nag\_wp) *Output*  
*On exit:* the number of attempted steps that failed to meet the local error requirement divided by the total number of steps attempted so far in the integration. A ‘large’ fraction indicates that the integrator is having trouble with the problem being solved. This can happen when the problem is ‘stiff’ and also when the solution has discontinuities in a low-order derivative.

- 4: STEPSOK – INTEGER *Output*  
*On exit:* the number of accepted steps.
- 5: HNEXT – REAL (KIND=nag\_wp) *Output*  
*On exit:* the step size the integrator will attempt to use for the next step.
- 6: IWSAV(130) – INTEGER array *Communication Array*  
 7: RWSAV(350) – REAL (KIND=nag\_wp) array *Communication Array*
- Note:** the communication RWSAV used by the other routines in the suite must be used here however, only the first 350 elements will be referenced.
- On entry:* these must be the same arrays supplied in a previous call to D02PEF or D02PFF. They must remain unchanged between calls.
- On exit:* information about the integration for use on subsequent calls to D02PEF or D02PFF or other associated routines.
- 8: 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.
- 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.**
- On exit:* IFAIL = 0 unless the routine detects an error or a warning has been flagged (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 or warnings detected by the routine:

IFAIL = 1

On entry, a previous call to the setup routine has not been made or the communication arrays have become corrupted, or a catastrophic error has already been detected elsewhere.

You cannot continue integrating the problem.

You cannot call this routine before you have called the integrator.

You have already made one call to this routine after the integrator could not achieve specified accuracy.

You cannot call this routine again.

IFAIL = –99

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL = –399

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL = -999

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

## **7 Accuracy**

Not applicable.

## **8 Parallelism and Performance**

Not applicable.

## **9 Further Comments**

When a secondary integration has taken place, that is when global error assessment has been specified using  $\text{METHOD} > 0$  in a prior call to D02PQF, then the approximate number of evaluations of  $f$  used in this secondary integration is given by  $2 \times \text{STEPSOK} \times \text{STEP COST}$  for  $\text{METHOD} = 2$  or  $3$  and  $3 \times \text{STEPSOK} \times \text{STEP COST}$  for  $\text{METHOD} = 1$ .

## **10 Example**

See Section 10 in D02PEF, D02PFF, D02PRF, D02PSF and D02PUF.

---