NAG Library Routine Document

D01TEF

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

Given the $2n + l$ moments of the weight function, D01TEF generates the recursion coefficients needed by D01TDF to calculate a Gaussian quadrature rule.

2 Specification

```
SUBROUTINE D01TEF (N, MU, A, B, C, IFAIL)
INTEGER            N, IFAIL
REAL (KIND=nag_wp) MU(0:2*N), A(N), B(N), C(N)
```

3 Description

D01TEF should only be used if the three-term recurrence cannot be determined analytically. A system of equations are formed, using the moments provided. This set of equations becomes ill-conditioned for moderate values of n, the number of abscissae and weights required. In most implementations quadruple precision calculation is used to maintain as much accuracy as possible.

4 References

Golub G H and Welsch J H (1969) Calculation of Gauss quadrature rules *Math. Comput.* **23** 221–230

5 Arguments

1: N – INTEGER *Input*

On entry: n, the number of weights and abscissae required.

Constraint: N > 0.

2: MU$(0 : 2 * \text{N})$ – REAL (KIND=nag_wp) array *Input*

On entry: MU(i) must contain the value of the moment with respect to x^i i.e., $\int w(x)x^i\,dx$, for $i = 0, 1, \ldots, 2n$.

3: A(N) – REAL (KIND=nag_wp) array *Output*

On exit: values helping define the three term recurrence used by D01TDF.

4: B(N) – REAL (KIND=nag_wp) array *Output*

On exit: values helping define the three term recurrence used by D01TDF.

5: C(N) – REAL (KIND=nag_wp) array *Output*

On exit: values helping define the three term recurrence used by D01TDF.

6: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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 $= 1$

The number of weights and abscissae requested (N) is less than 1: N $= \langle value \rangle$.

IFAIL $= 2$

The problem is too ill conditioned, it breaks down at row $\langle value \rangle$.

IFAIL $= -99$

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

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Internally quadruple precision is used to minimize loss of accuracy as much as possible.

8 Parallelism and Performance

D01TEF is not threaded in any implementation.

9 Further Comments

Because the routine cannot check the validity of all the data presented, the user is advised to independently check the result, perhaps by integrating a function whose integral is known, using D01TEF and subsequently D01TDF, to compare answers.

10 Example

This example program uses D01TEF and moments to calculate a three-term recurrence relationship appropriate for Gauss–Legendre quadrature. It then uses the recurrence relationship to derive the weights and abscissae by calling D01TDF.

10.1 Program Text

```
    Program d01tefe
!     D01TEF Example Program Text
!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: d01tdf, d01tef, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: n = 4, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: muzero
      Integer                          :: i, ifail
!     .. Local Arrays ..
      Real (Kind=nag_wp)               :: a(1:n), abscissae(1:n), b(1:n),      &
                                          c(1:n), mu(0:2*n), weights(1:n)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: real
!     .. Executable Statements ..
      Write (nout,*) 'D01TEF Example Program Results'
      Do i = 0, 2*n
        mu(i) = 0.0_nag_wp
      End Do
      Do i = 0, 2*n, 2
        mu(i) = 2.0_nag_wp/real(i+1,kind=nag_wp)
      End Do

      ifail = 0
      Call d01tef(n,mu,a,b,c,ifail)
      muzero = mu(0)
      Write (nout,*)
      Write (nout,*) '      a         b         c'
      Write (nout,99999)(a(i),b(i),c(i),i=1,n)
99999 Format (1X,3F10.5)

      ifail = 0
      Call d01tdf(n,a,b,c,muzero,weights,abscissae,ifail)
      Write (nout,*)
      Write (6,*) '   weights       abscissae '
      Write (6,99998)(weights(i),abscissae(i),i=1,4)
      Write (nout,*)
99998 Format (1X,F10.5,5X,F10.5)
    End Program d01tefe
```

10.2 Program Data

None.

10.3 Program Results

```
 D01TEF Example Program Results

       a         b         c
   -1.73205   0.00000   1.73205
   -1.93649   0.00000   1.11803
   -1.97203   0.00000   1.01835
   -1.00000   0.00000   0.50709

    weights       abscissae
    0.34785       -0.86114
    0.65215       -0.33998
    0.65215        0.33998
    0.34785        0.86114
```