NAG Library Routine Document

D01UBF

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

D01UBF returns the Gaussian quadrature approximation for the specific problem $\int_0^\infty \exp\left(-x^2\right) f(x)\, dx$. The degrees of precision catered for are: 1, 3, 5, 7, 9, 19, 29, 39 and 49, corresponding to values of $n = 1$, 2, 3, 4, 5, 10, 15, 20 and 25, where n is the number of weights.

2 Specification

```
SUBROUTINE D01UBF (FUN, N, ANS, IUSER, RUSER, IFAIL)
INTEGER            N, IUSER(*), IFAIL
REAL (KIND=nag_wp) ANS, RUSER(*)
EXTERNAL           FUN
```

3 Description

D01UBF uses the weights w_i and the abscissae x_i such that $\int_0^\infty \exp\left(-x^2\right) f(x)$ is approximated by $\sum_{i=1}^{n} w_i f(x_i)$ to maximum precision i.e., it is exact when $f(x)$ is a polynomial of degree $2n-1$.

4 References

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

5 Arguments

1: FUN – SUBROUTINE, supplied by the user. *External Procedure*

FUN must return the integrands $f(x_i)$ in F(i) for each x_i in X(i), for $i = 1, 2, \ldots,$ N at a given point.

The specification of FUN is:

```
SUBROUTINE FUN (X, F, N, IUSER, RUSER, ISTOP)
INTEGER            N, IUSER(*), ISTOP
REAL (KIND=nag_wp) X(N), F(N), RUSER(*)
```

1: X(N) – REAL (KIND=nag_wp) array *Input*

On entry: the points at which the integrand function f must be evaluated.

2: F(N) – REAL (KIND=nag_wp) array *Output*

On exit: F(i) must contain the value of the integrand $f(x_i)$ evaluated at the point X(i), for $i = 1, 2, \ldots,$ N.

3: N – INTEGER *Input*

On entry: N specifies the number of weights and abscissae to be used.

4: IUSER(∗) – INTEGER array *User Workspace*
5: RUSER(∗) – REAL (KIND=nag_wp) array *User Workspace*

FUN is called with the arguments IUSER and RUSER as supplied to D01UBF. You should use the arrays IUSER and RUSER to supply information to FUN.

6: ISTOP – INTEGER *Input/Output*

On entry: ISTOP = 0.

On exit: you may set ISTOP to a negative number if at any time it is impossible to evaluate the function $f(x)$. In this case D01UBF halts with IFAIL set to the value of ISTOP and the value returned in ANS will be that of a non-signalling NaN.

FUN must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub) program from which D01UBF is called. Arguments denoted as *Input* must **not** be changed by this procedure.

2: N – INTEGER *Input*

On entry: N specifies the number of weights and abscissae to be used.

Constraint: N = 1, 2, 3, 4, 5, 10, 15, 20 or 25.

3: ANS – REAL (KIND=nag_wp) *Output*

On exit: if IFAIL = 0, ANS contains an approximation to the integral. Otherwise, ANS will be a non-signalling NaN.

4: IUSER(∗) – INTEGER array *User Workspace*
5: RUSER(∗) – REAL (KIND=nag_wp) array *User Workspace*

IUSER and RUSER are not used by D01UBF, but are passed directly to FUN and should be used to pass information to this routine.

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 < 0

The user has halted the calculation.

IFAIL = 1

On entry, N = ⟨*value*⟩.
Constraint: 1 ≤ N ≤ 25.

IFAIL $= 2$

On entry, N $= \langle value \rangle$.
N is not one of the allowed values.

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

The weights and abscissae have been calculated using quadruple precision arithmetic.

8 Parallelism and Performance

D01UBF is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example computes an approximation to $\displaystyle\int_0^\infty \exp\left(-x^2\right)x\ dx$.

10.1 Program Text

```
!   Mark 26 Release. NAG Copyright 2016.

    Module d01ubfe_mod
!     D01UBF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Accessibility Statements ..
      Private
      Public                             :: fun
!     .. Parameters ..
      Integer, Parameter, Public         :: nout = 6
    Contains
      Subroutine fun(x,f,n,iuser,ruser,ifail)

!       .. Scalar Arguments ..
        Integer, Intent (Inout)          :: ifail
        Integer, Intent (In)             :: n
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (Out) :: f(n)
```

```
        Real (Kind=nag_wp), Intent (Inout) :: ruser(*)
        Real (Kind=nag_wp), Intent (In) :: x(n)
        Integer, Intent (Inout)        :: iuser(*)
!       .. Executable Statements ..
        f = x
      End Subroutine fun

    End Module d01ubfe_mod

    Program d01ubfe

!     .. Use Statements ..
      Use nag_library, Only: d01ubf, nag_wp
      Use d01ubfe_mod, Only: fun, nout
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: ans
      Integer                          :: ifail, n
!     .. Local Arrays ..
      Real (Kind=nag_wp)               :: ruser(1)
      Integer                          :: iuser(1)
!     .. Executable Statements ..
      Write (nout,*) 'D01UBF Example Program Results'
      n = 10
      ifail = 0
      Call d01ubf(fun,n,ans,iuser,ruser,ifail)
      Write (nout,*)
      Write (nout,99999) 'Approximation to the integral = ', ans

99999 Format (1X,A,F12.5)
    End Program d01ubfe
```

10.2 Program Data

None.

10.3 Program Results

```
 D01UBF Example Program Results

 Approximation to the integral =   0.50000
```