NAG Library Routine Document

D02UYF

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

D02UYF obtains the weights for Clenshaw–Curtis quadrature at Chebyshev points. This allows for fast approximations of integrals for functions specified on Chebyshev Gauss–Lobatto points on $[-1, 1]$.

2 Specification

```
SUBROUTINE D02UYF (N, W, IFAIL)
INTEGER              N, IFAIL
REAL (KIND=nag_wp) W(N+1)
```

3 Description

D02UYF obtains the weights for Clenshaw–Curtis quadrature at Chebyshev points.

Given the (Clenshaw–Curtis) weights w_i, for $i = 0, 1, \ldots, n$, and function values $f_i = f(t_i)$ (where $t_i = -\cos(i \times \pi/n)$, for $i = 0, 1, \ldots, n$, are the Chebyshev Gauss–Lobatto points), then $\int_{-1}^{1} f(x)dx \approx \sum_{i=0}^{n} w_i f_i$.

For a function discretized on a Chebyshev Gauss–Lobatto grid on $[a, b]$ the resultant summation must be multiplied by the factor $(b - a)/2$.

4 References

Trefethen L N (2000) *Spectral Methods in MATLAB* SIAM

5 Parameters

1: N – INTEGER *Input*

On entry: n, where the number of grid points is $n + 1$.

Constraint: N > 0 and N is even.

2: W(N + 1) – REAL (KIND=nag_wp) array *Output*

On exit: the Clenshaw–Curtis quadrature weights, w_i, for $i = 0, 1, \ldots, n$.

3: 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, N $= \langle value \rangle$.
Constraint: N > 0.

On entry, N $= \langle value \rangle$.
Constraint: N is even.

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

The accuracy should be close to ***machine precision***.

8 Parallelism and Performance

Not applicable.

9 Further Comments

A real array of length $2n$ is internally allocated.

10 Example

This example approximates the integral $\int_{-1}^{3} 3x^2 dx$ using 65 Clenshaw–Curtis weights and a 65-point Chebyshev Gauss–Lobatto grid on $[-1, 3]$.

10.1 Program Text

```
!   D02UYF Example Program Text
!   Mark 25 Release. NAG Copyright 2014.

    Module d02uyfe_mod

!     D02UYF 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                                :: exact
!     .. Parameters ..
      Real (Kind=nag_wp), Parameter, Public :: a = -1.0_nag_wp
      Real (Kind=nag_wp), Parameter, Public :: b = 3.0_nag_wp
      Integer, Parameter, Public            :: nin = 5, nout = 6
      Logical, Parameter, Public            :: reqerr = .False., reqwgt = .False.
    Contains
      Function exact(x)

!       .. Function Return Value ..
        Real (Kind=nag_wp)                    :: exact
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In)       :: x
!       .. Executable Statements ..
        exact = 3.0_nag_wp*x**2
        Return
      End Function exact
    End Module d02uyfe_mod
    Program d02uyfe

!     D02UYF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: d02ucf, d02uyf, ddot, nag_wp, x02ajf
      Use d02uyfe_mod, Only: a, b, exact, nin, nout, reqerr, reqwgt
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)                    :: integ, scale, uerr
      Integer                               :: i, ifail, iu, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable       :: f(:), w(:), x(:)
!     .. Intrinsic Procedures ..
      Intrinsic                             :: abs, int
!     .. Executable Statements ..
      Write (nout,*) ' D02UYF Example Program Results '
      Write (nout,*)

      Read (nin,*)
      Read (nin,*) n

      Allocate (f(n+1),w(n+1),x(n+1))

!     Set up solution grid
      ifail = 0
      Call d02ucf(n,a,b,x,ifail)

!     Set up problem right hand sides for grid
      Do i = 1, n + 1
        f(i) = exact(x(i))
      End Do
      scale = 0.5_nag_wp*(b-a)

!     Solve on equally spaced grid
      ifail = 0
      Call d02uyf(n,w,ifail)
!     The NAG name equivalent of ddot is f06eaf
      integ = ddot(n+1,w,1,f,1)*scale

!     Print function values and weights if required
      If (reqwgt) Then
        Write (nout,*) ' f(x) and Integral weights'
        Write (nout,*)
        Write (nout,99999)
        Write (nout,99998)(x(i),f(i),w(i),i=1,n+1)
      End If

!     Print approximation to integral
```

```
      Write (nout,99996) a, b, integ

      If (reqerr) Then
        uerr = abs(integ-28.0_nag_wp)
        iu = 10*(int(uerr/10.0_nag_wp/x02ajf())+1)
        Write (nout,99997) iu
      End If

99999 Format (1X,T8,'X',T18,'f(X)',T28,'W')
99998 Format (1X,3F10.4)
99997 Format (/1X,'Integral is within a multiple ',I8, &
        ' of machine precision.')
99996 Format (/1X,'Integral of f(x) from ',F6.1,' to ',F6.2,' = ',F13.5,'.'/)
    End Program d02uyfe
```

10.2 Program Data

```
D02UYF Example Program Data
  64           : N
```

10.3 Program Results

```
  D02UYF Example Program Results


 Integral of f(x) from   -1.0 to   3.00 =      28.00000.
```