NAG Library Routine Document

D02UDF

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

D02UDF differentiates a function discretized on Chebyshev Gauss–Lobatto points. The grid points on which the function values are to be provided are normally returned by a previous call to D02UCF.

2 Specification

```
SUBROUTINE D02UDF (N, F, FD, IFAIL)
INTEGER            N, IFAIL
REAL (KIND=nag_wp) F(N+1), FD(N+1)
```

3 Description

D02UDF differentiates a function discretized on Chebyshev Gauss–Lobatto points on $[-1,1]$. The polynomial interpolation on Chebyshev points is equivalent to trigonometric interpolation on equally spaced points. Hence the differentiation on the Chebyshev points can be implemented by the Fast Fourier transform (FFT).

Given the function values $f(x_i)$ on Chebyshev Gauss–Lobatto points $x_i = -\cos((i-1)\pi/n)$, for $i = 1,2,\ldots,n+1$, f is differentiated with respect to x by means of forward and backward FFTs on the function values $f(x_i)$. D02UDF returns the computed derivative values $f'(x_i)$, for $i = 1,2,\ldots,n+1$. The derivatives are computed with respect to the standard Chebyshev Gauss–Lobatto points on $[-1,1]$; for derivatives of a function on $[a,b]$ the returned values have to be scaled by a factor $2/(b-a)$.

4 References

Canuto C, Hussaini M Y, Quarteroni A and Zang T A (2006) *Spectral Methods: Fundamentals in Single Domains* Springer

Greengard L (1991) Spectral integration and two-point boundary value problems *SIAM J. Numer. Anal.* **28(4)** 1071–80

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

5 Arguments

1: N – INTEGER *Input*

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

Constraint: N >0 and N is even.

2: F(N + 1) – REAL (KIND=nag_wp) array *Input*

On entry: the function values $f(x_i)$, for $i = 1,2,\ldots,n+1$

3: FD(N + 1) – REAL (KIND=nag_wp) array *Output*

On exit: the approximations to the derivatives of the function evaluated at the Chebyshev Gauss–Lobatto points. For functions defined on $[a,b]$, the returned derivative values (corresponding to the domain $[-1,1]$) must be multiplied by the factor $2/(b-a)$ to obtain the correct values on $[a,b]$.

4: 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 $= 1$

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 3.9 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 accuracy is close to ***machine precision*** for small numbers of grid points, typically less than 100. For larger numbers of grid points, the error in differentiation grows with the number of grid points. See Greengard (1991) for more details.

8 Parallelism and Performance

D02UDF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

D02UDF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The number of operations is of the order $n\log(n)$ and the memory requirements are $O(n)$; thus the computation remains efficient and practical for very fine discretizations (very large values of n).

10 Example

The function $2x + \exp(-x)$, defined on $[0, 1.5]$, is supplied and then differentiated on a grid.

10.1 Program Text

```
!   D02UDF Example Program Text
!   Mark 26 Release. NAG Copyright 2016.

    Module d02udfe_mod

!     D02UDF 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                           :: deriv, fcn
!     .. Parameters ..
      Real (Kind=nag_wp), Parameter, Public :: a = 0.0_nag_wp
      Real (Kind=nag_wp), Parameter, Public :: b = 1.5_nag_wp
      Real (Kind=nag_wp), Parameter    :: one = 1.0_nag_wp
      Integer, Parameter, Public       :: nin = 5, nout = 6
      Logical, Parameter, Public       :: reqerr = .False.
    Contains
      Function fcn(x)

!       .. Function Return Value ..
        Real (Kind=nag_wp)             :: fcn
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: x
!       .. Intrinsic Procedures ..
        Intrinsic                      :: exp
!       .. Executable Statements ..
        fcn = (one+one)*x + exp(-x)
        Return
      End Function fcn
      Function deriv(x)

!       .. Function Return Value ..
        Real (Kind=nag_wp)             :: deriv
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: x
!       .. Intrinsic Procedures ..
        Intrinsic                      :: exp
!       .. Executable Statements ..
        deriv = one + one - exp(-x)
        Return
      End Function deriv

    End Module d02udfe_mod
    Program d02udfe

!     D02UDF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: d02ucf, d02udf, nag_wp, x02ajf
      Use d02udfe_mod, Only: a, b, deriv, fcn, nin, nout, reqerr
!     .. Implicit None Statement ..
      Implicit None
```

```
!     .. Local Scalars ..
      Real (Kind=nag_wp)              :: scale, teneps, uxerr
      Integer                         :: i, ifail, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: f(:), fd(:), x(:)
!     .. Intrinsic Procedures ..
      Intrinsic                       :: abs, int, max
!     .. Executable Statements ..
      Write (nout,*) ' D02UDF Example Program Results '
      Write (nout,*)

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

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

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

!     Evaluate fcn on Chebyshev grid.
      Do i = 1, n + 1
        f(i) = fcn(x(i))
      End Do

!     Calculate derivative of fcn.
      ifail = 0
      Call d02udf(n,f,fd,ifail)

      scale = 2.0_nag_wp/(b-a)
      fd(1:n+1) = scale*fd(1:n+1)

!     Print function and its derivative
      Write (nout,*) ' Original Function F and numerical derivative Fx'
      Write (nout,*)
      Write (nout,99999)
      Write (nout,99998)(x(i),f(i),fd(i),i=1,n+1)

      If (reqerr) Then
        uxerr = 0.0_nag_wp
        Do i = 1, n + 1
          uxerr = max(uxerr,abs(fd(i)-deriv(x(i))))
        End Do
        teneps = 100.0_nag_wp*x02ajf()
        Write (nout,99997) 100*(int(uxerr/teneps)+1)
      End If

99999 Format (1X,T8,'X',T18,'F',T28,'Fx')
99998 Format (1X,3F10.4)
99997 Format (1X,'Fx is within a multiple ',I8,' of machine precision.')

    End Program d02udfe
```

10.2 Program Data

```
D02UDF Example Program Data
  16              : N
```

10.3 Program Results

```
  D02UDF Example Program Results

  Original Function F and numerical derivative Fx

       X         F         Fx
    0.0000    1.0000    1.0000
    0.0144    1.0145    1.0143
    0.0571    1.0587    1.0555
    0.1264    1.1341    1.1187
    0.2197    1.2421    1.1972
```

```
    0.3333    1.3832    1.2835
    0.4630    1.5554    1.3706
    0.6037    1.7542    1.4532
    0.7500    1.9724    1.5276
    0.8963    2.2007    1.5919
    1.0370    2.4285    1.6455
    1.1667    2.6448    1.6886
    1.2803    2.8386    1.7221
    1.3736    3.0004    1.7468
    1.4429    3.1221    1.7638
    1.4856    3.1975    1.7736
    1.5000    3.2231    1.7769
```