

NAG Library Routine Document

D02UAF

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

D02UAF obtains the Chebyshev coefficients of a function discretized on Chebyshev Gauss–Lobatto points. The set of discretization points on which the function is evaluated is usually obtained by a previous call to D02UCF.

2 Specification

```
SUBROUTINE D02UAF (N, F, C, IFAIL)
  INTEGER          N, IFAIL
  REAL (KIND=nag_wp) F(N+1), C(N+1)
```

3 Description

D02UAF computes the coefficients c_j , for $j = 1, 2, \dots, n + 1$, of the interpolating Chebyshev series

$$\frac{1}{2}c_1T_0(\bar{x}) + c_2T_1(\bar{x}) + c_3T_2(\bar{x}) + \dots + c_{n+1}T_n(\bar{x}),$$

which interpolates the function $f(x)$ evaluated at the Chebyshev Gauss–Lobatto points

$$\bar{x}_r = -\cos((r-1)\pi/n), \quad r = 1, 2, \dots, n + 1.$$

Here $T_j(\bar{x})$ denotes the Chebyshev polynomial of the first kind of degree j with argument \bar{x} defined on $[-1, 1]$. In terms of your original variable, x say, the input values at which the function values are to be provided are

$$x_r = -\frac{1}{2}(b-a)\cos(\pi(r-1)/n) + \frac{1}{2}(b+a), \quad r = 1, 2, \dots, n + 1,$$

where b and a are respectively the upper and lower ends of the range of x over which the function is required.

4 References

Canuto C (1988) *Spectral Methods in Fluid Dynamics* 502 Springer

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

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 $n + 1$. This is also the largest order of Chebyshev polynomial in the Chebyshev series to be computed.

Constraint: $N > 0$ and N is even.

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

On entry: the function values $f(x_r)$, for $r = 1, 2, \dots, n + 1$.

3: $C(N + 1)$ – REAL (KIND=nag_wp) array *Output*
On exit: the Chebyshev coefficients, c_j , for $j = 1, 2, \dots, n + 1$.

4: 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, $N = \langle \text{value} \rangle$.
 Constraint: $N > 1$.

On entry, $N = \langle \text{value} \rangle$.
 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 Chebyshev coefficients computed should be accurate to within a small multiple of *machine precision*.

8 Parallelism and Performance

D02UAF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

D02UAF 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

See Section 10 in D02UEF.
