

NAG Library Routine Document

C06DCF

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

C06DCF evaluates a polynomial from its Chebyshev series representation at a set of points.

2 Specification

```
SUBROUTINE C06DCF (X, LX, XMIN, XMAX, C, N, S, RES, IFAIL)
  INTEGER          LX, N, S, IFAIL
  REAL (KIND=nag_wp) X(LX), XMIN, XMAX, C(N), RES(LX)
```

3 Description

C06DCF evaluates, at each point in a given set X , the sum of a Chebyshev series of one of three forms according to the value of the parameter S :

$$S = 1: \quad 0.5c_1 + \sum_{j=2}^n c_j T_{j-1}(\bar{x})$$

$$S = 2: \quad 0.5c_1 + \sum_{j=2}^n c_j T_{2j-2}(\bar{x})$$

$$S = 3: \quad \sum_{j=1}^n c_j T_{2j-1}(\bar{x})$$

where \bar{x} lies in the range $-1.0 \leq \bar{x} \leq 1.0$. Here $T_r(x)$ is the Chebyshev polynomial of order r in \bar{x} , defined by $\cos(ry)$ where $\cos y = \bar{x}$.

It is assumed that the independent variable \bar{x} in the interval $[-1.0, +1.0]$ was obtained from your original variable $x \in X$, a set of real numbers in the interval $[x_{\min}, x_{\max}]$, by the linear transformation

$$\bar{x} = \frac{2x - (x_{\max} + x_{\min})}{x_{\max} - x_{\min}}.$$

The method used is based upon a three-term recurrence relation; for details see Clenshaw (1962).

The coefficients c_j are normally generated by other routines, for example they may be those returned by the interpolation routine E01AEF (in vector A), by a least squares fitting routine in Chapter E02, or as the solution of a boundary value problem by D02JAF, D02JBF or D02UEF.

4 References

Clenshaw C W (1962) Chebyshev Series for Mathematical Functions *Mathematical tables* HMSO

5 Arguments

- 1: X(LX) – REAL (KIND=nag_wp) array *Input*
On entry: $x \in X$, the set of arguments of the series.
Constraint: $X_{\min} \leq X(i) \leq X_{\max}$, for $i = 1, 2, \dots, LX$.

- 2: LX – INTEGER *Input*
On entry: the number of evaluation points in X .
Constraint: $LX \geq 1$.
- 3: XMIN – REAL (KIND=nag_wp) *Input*
 4: XMAX – REAL (KIND=nag_wp) *Input*
On entry: the lower and upper end points respectively of the interval $[x_{\min}, x_{\max}]$. The Chebyshev series representation is in terms of the normalized variable \bar{x} , where
- $$\bar{x} = \frac{2x - (x_{\max} + x_{\min})}{x_{\max} - x_{\min}}.$$
- Constraint:* $XMIN < XMAX$.
- 5: C(N) – REAL (KIND=nag_wp) array *Input*
On entry: $C(j)$ must contain the coefficient c_j of the Chebyshev series, for $j = 1, 2, \dots, n$.
- 6: N – INTEGER *Input*
On entry: n , the number of terms in the series.
Constraint: $N \geq 1$.
- 7: S – INTEGER *Input*
On entry: determines the series (see Section 3).
 $S = 1$
 The series is general.
 $S = 2$
 The series is even.
 $S = 3$
 The series is odd.
Constraint: $S = 1, 2$ or 3 .
- 8: RES(LX) – REAL (KIND=nag_wp) array *Output*
On exit: the Chebyshev series evaluated at the set of points X .
- 9: 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, $LX = \langle value \rangle$.
Constraint: $LX \geq 1$.

$IFAIL = 2$

On entry, $N = \langle value \rangle$.
Constraint: $N \geq 1$.

$IFAIL = 3$

On entry, $S = \langle value \rangle$.
Constraint: $S = 1, 2$ or 3 .

$IFAIL = 4$

On entry, $XMAX = \langle value \rangle$ and $XMIN = \langle value \rangle$.
Constraint: $XMIN < XMAX$.

$IFAIL = 5$

On entry, element $X(\langle value \rangle) = \langle value \rangle$, $XMIN = \langle value \rangle$ and $XMAX = \langle value \rangle$.
Constraint: $XMIN \leq X(i) \leq XMAX$, for all i .

$IFAIL = -99$

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

There may be a loss of significant figures due to cancellation between terms. However, provided that n is not too large, C06DCF yields results which differ little from the best attainable for the available *machine precision*.

8 Parallelism and Performance

C06DCF is not threaded in any implementation.

9 Further Comments

The time taken increases with n .

C06DCF has been prepared in the present form to complement a number of integral equation solving routines which use Chebyshev series methods, e.g., D05AAF and D05ABF.

10 Example

This example evaluates

$$0.5 + T_1(x) + 0.5T_2(x) + 0.25T_3(x)$$

at the points $X = [0.5, 1.0, -0.2]$.

10.1 Program Text

```

Program c06dcfe

!      C06DCF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
Use nag_library, Only: c06dcf, nag_wp
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
Real (Kind=nag_wp)         :: xmax, xmin
Integer                    :: i, ifail, lx, n, s
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: c(:), res(:), x(:)
!      .. Executable Statements ..
Write (nout,*) 'C06DCF Example Program Results'
!      Skip heading in data file
Read (nin,*)
Read (nin,*) n, lx
Allocate (c(n),res(lx),x(lx))

Read (nin,*) x(1:lx)
Read (nin,*) xmin, xmax
Read (nin,*) s
Read (nin,*) c(1:n)

!      ifail: behaviour on error exit
!      =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
ifail = 0
Call c06dcf(x,lx,xmin,xmax,c,n,s,res,ifail)

Write (nout,*)
Write (nout,*) '      x      sum of series'
Write (nout,*)
Write (nout,99999)(x(i),res(i),i=1,lx)

99999 Format (1X,F8.4,4X,F8.4)
End Program c06dcfe

```

10.2 Program Data

```

C06DCF Example Program Data
 4  3      : n, lx
0.5  1.0 -0.2 : x
-1.0  1.0      : xmin, xmax
 1      : s
1.0  1.0  0.5  0.25 : c

```

10.3 Program Results

C06DCF Example Program Results

x	sum of series
0.5000	0.5000
1.0000	2.2500
-0.2000	-0.0180
