# NAG Library Routine Document

# D05ABF

**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

D05ABF solves any linear nonsingular Fredholm integral equation of the second kind with a smooth kernel.

## 2 Specification

```
 SUBROUTINE D05ABF(K, G, LAMBDA, A, B, ODOREV, EV, N, CM, F1, WK, LDCM,
1                  NT2P1, F, C, IFAIL)
 INTEGER           N, LDCM, NT2P1, IFAIL
 double precision  K, G, LAMBDA, A, B, CM(LDCM,LDCM), F1(LDCM,1),
1                  WK(2,NT2P1), F(N), C(N)
 LOGICAL           ODOREV, EV
 EXTERNAL          K, G
```

## 3 Description

D05ABF uses the method of El–Gendi (1969) to solve an integral equation of the form

$$f(x) - \lambda \int_a^b k(x,s) f(s)\, ds = g(x)$$

for the function $f(x)$ in the range $a \leq x \leq b$.

An approximation to the solution $f(x)$ is found in the form of an $n$ term Chebyshev series $\sum_{i=1}^{n}{}' c_i T_i(x)$, where $'$ indicates that the first term is halved in the sum. The coefficients $c_i$, for $i = 1, 2, \ldots, n$, of this series are determined directly from approximate values $f_i$, for $i = 1, 2, \ldots, n$, of the function $f(x)$ at the first $n$ of a set of $m + 1$ Chebyshev points

$$x_i = \tfrac{1}{2}(a + b + (b - a) \times \cos[(i - 1) \times \pi/m]), \qquad i = 1, 2, \ldots, m + 1.$$

The values $f_i$ are obtained by solving a set of simultaneous linear algebraic equations formed by applying a quadrature formula (equivalent to the scheme of Clenshaw and Curtis (1960)) to the integral equation at each of the above points.

In general $m = n - 1$. However, advantage may be taken of any prior knowledge of the symmetry of $f(x)$. Thus if $f(x)$ is symmetric (i.e., even) about the mid-point of the range $(a, b)$, it may be approximated by an even Chebyshev series with $m = 2n - 1$. Similarly, if $f(x)$ is anti-symmetric (i.e., odd) about the mid-point of the range of integration, it may be approximated by an odd Chebyshev series with $m = 2n$.

## 4 References


Clenshaw C W and Curtis A R (1960) A method for numerical integration on an automatic computer *Numer. Math.* **2** 197–205

El–Gendi S E (1969) Chebyshev solution of differential, integral and integro-differential equations *Comput. J.* **12** 282–287

## 5 Parameters

1: K – ***double precision*** FUNCTION, supplied by the user. *External Procedure*

K must compute the value of the kernel $k(x,s)$ of the integral equation over the square $a \leq x \leq b$, $a \leq s \leq b$.

The specification of K is:

```
double precision FUNCTION K(X, S)
double precision            X, S
```

1: X – ***double precision*** *Input*

2: S – ***double precision*** *Input*

*On entry*: the values of $x$ and $s$ at which $k(x,s)$ is to be calculated.

K must be declared as EXTERNAL in the (sub)program from which D05ABF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

2: G – ***double precision*** FUNCTION, supplied by the user. *External Procedure*

G must compute the value of the function $g(x)$ of the integral equation in the interval $a \leq x \leq b$.

The specification of G is:

```
double precision FUNCTION G(X)
double precision            X
```

1: X – ***double precision*** *Input*

*On entry*: the value of $x$ at which $g(x)$ is to be calculated.

G must be declared as EXTERNAL in the (sub)program from which D05ABF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

3: LAMBDA – ***double precision*** *Input*

*On entry*: the value of the parameter $\lambda$ of the integral equation.

4: A – ***double precision*** *Input*

*On entry*: $a$, the lower limit of integration.

5: B – ***double precision*** *Input*

*On entry*: $b$, the upper limit of integration.

*Constraint*: B $>$ A.

6: ODOREV – LOGICAL *Input*

*On entry*: indicates whether it is known that the solution $f(x)$ is odd or even about the mid-point of the range of integration. If ODOREV is .TRUE. then an odd or even solution is sought depending upon the value of EV.

7: EV – LOGICAL *Input*

*On entry*: is ignored if ODOREV is .FALSE. Otherwise, if EV is .TRUE., an even solution is sought, whilst if EV is .FALSE., an odd solution is sought.

8: N – INTEGER *Input*

*On entry*: the number of terms in the Chebyshev series which approximates the solution $f(x)$.

*Constraint*: N $> 0$.

9: CM(LDCM,LDCM) – ***double precision*** array *Workspace*
10: F1(LDCM,1) – ***double precision*** array *Workspace*
11: WK(2,NT2P1) – ***double precision*** array *Workspace*

12: LDCM – INTEGER *Input*

*On entry*: the first dimension of the arrays CM and F1 and the second dimension of the array CM as declared in the (sub)program from which D05ABF is called.

*Constraint*: LDCM $\geq$ N.

13: NT2P1 – INTEGER *Input*

*On entry*: the second dimension of the array WK as declared in the (sub)program from which D05ABF is called. The value $2 \times$ N $+ 1$.

14: F(N) – ***double precision*** array *Output*

*On exit*: the approximate values $f_i$, for $i = 1, 2, \ldots,$ N, of the function $f(x)$ at the first N of $m + 1$ Chebyshev points (see Section 3), where

$m =$ 2N $- 1$ if ODOREV = .TRUE. and EV = .TRUE.

$m =$ 2N if ODOREV = .TRUE. and EV = .FALSE.

$m =$ N $- 1$ if ODOREV = .FALSE.

If ODOREV is .TRUE., then $m = 2 \times$ N $- 1$ if EV is .TRUE. and $m = 2 \times$ N if EV is .FALSE.; otherwise $m =$ N $- 1$.

15: C(N) – ***double precision*** array *Output*

*On exit*: the coefficients $c_i$, for $i = 1, 2, \ldots,$ N, of the Chebyshev series approximation to $f(x)$. When ODOREV is .TRUE., this series contains polynomials of even order only or of odd order only, according to EV being .TRUE. or .FALSE. respectively.

16: 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.

*On exit*: IFAIL $= 0$ unless the routine detects an error (see Section 6).

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.**

## 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, A $\geq$ B or N $< 1$.

IFAIL $= 2$

A failure has occurred due to proximity to an eigenvalue. In general, if LAMBDA is near an eigenvalue of the integral equation, the corresponding matrix will be nearly singular. In the special case, $m = 1$, the matrix reduces to a zero-valued number.

## 7 Accuracy

No explicit error estimate is provided by the routine but it is possible to obtain a good indication of the accuracy of the solution either

(i) by examining the size of the later Chebyshev coefficients $c_i$, or

(ii) by comparing the coefficients $c_i$ or the function values $f_i$ for two or more values of N.

## 8 Further Comments

The time taken by D05ABF depends upon the value of N and upon the complexity of the kernel function $k(x, s)$.

## 9 Example

This example solves Love's equation:

$$f(x) + \frac{1}{\pi}\int_{-1}^{1} \frac{f(s)}{1 + (x - s)^2}\, ds = 1.$$

It will solve the slightly more general equation:

$$f(x) - \lambda \int_a^b k(x, s) f(s)\, ds = 1$$

where $k(x, s) = \alpha / \left(\alpha^2 + (x - s)^2\right)$. The values $\lambda = -1/\pi, a = -1, b = 1, \alpha = 1$ are used below.

It is evident from the symmetry of the given equation that $f(x)$ is an even function. Advantage is taken of this fact both in the application of D05ABF, to obtain the $f_i \simeq f(x_i)$ and the $c_i$, and in subsequent applications of C06DBF to obtain $f(x)$ at selected points.

The program runs for $\mathrm{N} = 5$ and $\mathrm{N} = 10$.

### 9.1 Program Text

```
*     D05ABF Example Program Text
*     Mark 14 Revised. NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          LDCM, NT2P1
      PARAMETER        (LDCM=10,NT2P1=2*LDCM+1)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Scalars in Common ..
      DOUBLE PRECISION ALPHA, W
*     .. Local Scalars ..
      DOUBLE PRECISION A, A1, B, CHEBR, D, E, LAMBDA, S, X
      INTEGER          I, IFAIL, N, SS
      LOGICAL          EV, ODOREV
*     .. Local Arrays ..
      DOUBLE PRECISION C(LDCM), CM(LDCM,LDCM), F(LDCM), F1(LDCM,1),
     +                 WK(2,NT2P1)
*     .. External Functions ..
      DOUBLE PRECISION C06DBF, G, K
      EXTERNAL         C06DBF, G, K
*     .. External Subroutines ..
      EXTERNAL         D05ABF
*     .. Common blocks ..
      COMMON           /AFRED2/ALPHA, W
```

```
*     .. Executable Statements ..
      WRITE (NOUT,*) 'D05ABF Example Program Results'
      WRITE (NOUT,*)
      ODOREV = .TRUE.
      EV = .TRUE.
      LAMBDA = -0.3183D0
      A = -1.0D0
      B = 1.0D0
      ALPHA = 1.0D0
      W = ALPHA*ALPHA
      IF (ODOREV .AND. EV) THEN
         WRITE (NOUT,*) 'Solution is even'
      ELSE
         IF (ODOREV) WRITE (NOUT,*) 'Solution is odd'
      END IF
      DO 60 N = 5, LDCM, 5
         IFAIL = 1
*
         CALL D05ABF(K,G,LAMBDA,A,B,ODOREV,EV,N,CM,F1,WK,LDCM,NT2P1,F,C,
     +               IFAIL)
*
         IF (IFAIL.EQ.0) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99999) 'Results for N =', N
            WRITE (NOUT,*)
            WRITE (NOUT,*) '  I         F(I)         C(I)'
            DO 20 I = 1, N
               WRITE (NOUT,99998) I, F(I), C(I)
   20       CONTINUE
            WRITE (NOUT,*)
            WRITE (NOUT,*) '     X           F(X)'
            IF (ODOREV) THEN
               IF (EV) THEN
                  SS = 2
               ELSE
                  SS = 3
               END IF
            ELSE
               SS = 1
            END IF
            A1 = 0.5D0*(A+B)
            S = 0.5D0*(B-A)
            X = A1
            IF ( .NOT. ODOREV) THEN
               X = X - 5
            ELSE
               X = A1
            END IF
            D = 1.0D0/S
            S = 0.25D0*S
            E = B + 0.1D0*S
   40       CHEBR = C06DBF((X-A1)*D,C,N,SS)
            WRITE (NOUT,99997) X, CHEBR
            X = X + S
            IF (X.LT.E) GO TO 40
         ELSE
            IF (IFAIL.EQ.1) THEN
               WRITE (NOUT,*)
               WRITE (NOUT,*) 'Failure in D05ABF -'
               WRITE (NOUT,*) 'error in integration limits'
            ELSE IF (IFAIL.EQ.2) THEN
               WRITE (NOUT,*)
               WRITE (NOUT,*) 'Failure in D05ABF -'
               WRITE (NOUT,*) 'LAMBDA near eigenvalue'
            ELSE
               WRITE (NOUT,*)
               WRITE (NOUT,*) ' ** D05ABF returned with IFAIL = ', IFAIL
               GO TO 80
            END IF
         END IF
   60 CONTINUE
```

```
   80 CONTINUE
*
99999 FORMAT (1X,A,I3)
99998 FORMAT (1X,I3,F15.5,E15.5)
99997 FORMAT (1X,F8.4,F15.5)
      END
*
      DOUBLE PRECISION FUNCTION K(X,S)
*     .. Scalar Arguments ..
      DOUBLE PRECISION S, X
*     .. Scalars in Common ..
      DOUBLE PRECISION ALPHA, W
*     .. Common blocks ..
      COMMON           /AFRED2/ALPHA, W
*     .. Executable Statements ..
      K = ALPHA/(W+(X-S)*(X-S))
      RETURN
      END
*
      DOUBLE PRECISION FUNCTION G(X)
*     .. Scalar Arguments ..
      DOUBLE PRECISION X
*     .. Executable Statements ..
      G = 1.0D0
      RETURN
      END
```

## 9.2 Program Data

None.

## 9.3 Program Results

```
 D05ABF Example Program Results

 Solution is even

 Results for N =  5

   I         F(I)          C(I)
   1        0.75572    0.14152E+01
   2        0.74534    0.49384E-01
   3        0.71729   -0.10476E-02
   4        0.68319   -0.23282E-03
   5        0.66051    0.20890E-04

      X           F(X)
   0.0000        0.65742
   0.2500        0.66383
   0.5000        0.68319
   0.7500        0.71489
   1.0000        0.75572

 Results for N = 10

   I         F(I)          C(I)
   1        0.75572    0.14152E+01
   2        0.75336    0.49384E-01
   3        0.74639   -0.10475E-02
   4        0.73525   -0.23275E-03
   5        0.72081    0.19986E-04
   6        0.70452    0.98675E-06
   7        0.68825   -0.23796E-06
   8        0.67404    0.18581E-08
   9        0.66361    0.24483E-08
  10        0.65812   -0.16527E-09

      X           F(X)
```

```
   0.0000         0.65742
   0.2500         0.66384
   0.5000         0.68319
   0.7500         0.71489
   1.0000         0.75572
```