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,          &
                   NT2P1, F, C, IFAIL)

INTEGER            N, LDCM, NT2P1, IFAIL
REAL (KIND=nag_wp) K, G, LAMBDA, A, B, CM(LDCM,LDCM), F1(LDCM,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 – REAL (KIND=nag_wp) 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:

```
FUNCTION K (X, S)

REAL (KIND=nag_wp) K
REAL (KIND=nag_wp) X, S
```

1: X – REAL (KIND=nag_wp) *Input*

2: S – REAL (KIND=nag_wp) *Input*

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

K must either be a module subprogram USEd by, or 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 – REAL (KIND=nag_wp) 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:

```
FUNCTION G (X)

REAL (KIND=nag_wp) G
REAL (KIND=nag_wp) X
```

1: X – REAL (KIND=nag_wp) *Input*

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

G must either be a module subprogram USEd by, or 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 – REAL (KIND=nag_wp) *Input*

On entry: the value of the parameter λ of the integral equation.

4: A – REAL (KIND=nag_wp) *Input*

On entry: a, the lower limit of integration.

5: B – REAL (KIND=nag_wp) *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: $\mathrm{N} \geq 1$.

9: CM(LDCM,LDCM) – REAL (KIND=nag_wp) array *Workspace*
10: F1(LDCM,1) – REAL (KIND=nag_wp) array *Workspace*
11: WK(2,NT2P1) – REAL (KIND=nag_wp) 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: $\mathrm{LDCM} \geq \mathrm{N}$.

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 $2 \times \mathrm{N} + 1$.

14: F(N) – REAL (KIND=nag_wp) array *Output*

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

$m = 2\mathrm{N} - 1$ if ODOREV = .TRUE. and EV = .TRUE..

$m = 2\mathrm{N}$ if ODOREV = .TRUE. and EV = .FALSE..

$m = \mathrm{N} - 1$ if ODOREV = .FALSE..

15: C(N) – REAL (KIND=nag_wp) array *Output*

On exit: the coefficients c_i, for $i = 1, 2, \ldots, \mathrm{N}$, 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.

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: $\mathrm{IFAIL} = 0$ 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, 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 C06DCF to obtain $f(x)$ at selected points.

The program runs for N $= 5$ and N $= 10$.

9.1 Program Text

```
!   D05ABF Example Program Text
!   Mark 24 Release. NAG Copyright 2012.

    Module d05abfe_mod

!     D05ABF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
```

```
      Implicit None
!     .. Parameters ..
      Integer, Parameter                   :: nmax = 10, nout = 6
    Contains
      Function k(x,s)

!       .. Function Return Value ..
        Real (Kind=nag_wp)                   :: k
!       .. Parameters ..
        Real (Kind=nag_wp), Parameter        :: alpha = 1.0_nag_wp
        Real (Kind=nag_wp), Parameter        :: w = alpha**2
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In)      :: s, x
!       .. Executable Statements ..
        k = alpha/(w+(x-s)*(x-s))

        Return
      End Function k
      Function g(x)

!       .. Function Return Value ..
        Real (Kind=nag_wp)                   :: g
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In)      :: x
!       .. Executable Statements ..
        g = 1.0_nag_wp

        Return
      End Function g
    End Module d05abfe_mod
    Program d05abfe

!     D05ABF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: c06dcf, d05abf, nag_wp, x01aaf
      Use d05abfe_mod, Only: g, k, nmax, nout
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)                   :: a, b, lambda, x0
      Integer                              :: i, ifail, ldcm, lx, n, nt2p1, ss
      Logical                              :: ev, odorev
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable      :: c(:), chebr(:), cm(:,:), f(:),   &
                                              f1(:,:), wk(:,:), x(:)
!     .. Intrinsic Procedures ..
      Intrinsic                            :: cos, int, real
!     .. Executable Statements ..
      Write (nout,*) 'D05ABF Example Program Results'

      odorev = .True.
      ev = .True.
      lambda = -0.3183_nag_wp
      a = -1.0_nag_wp
      b = 1.0_nag_wp

      If (odorev) Then
        Write (nout,*)
        If (ev) Then
          Write (nout,*) 'Solution is even'
          ss = 2
        Else
          Write (nout,*) 'Solution is odd'
          ss = 3
        End If
        x0 = 0.5_nag_wp*(a+b)
      Else
        ss = 1
        x0 = a
      End If
```

```
!     Set up uniform grid to evaluate Chebyshev polynomials.
      lx = int(4.000001_nag_wp*(b-x0)) + 1
      Allocate (x(lx),chebr(lx))
      x(1) = x0
      Do i = 2, lx
        x(i) = x(i-1) + 0.25_nag_wp
      End Do

      Do n = 5, nmax, 5
        ldcm = n
        nt2p1 = 2*n + 1
        Allocate (c(n),cm(ldcm,ldcm),f(n),f1(ldcm,1),wk(2,nt2p1))

        ifail = -1
        Call d05abf(k,g,lambda,a,b,odorev,ev,n,cm,f1,wk,ldcm,nt2p1,f,c,ifail)

        If (ifail==0) Then
          Write (nout,*)
          Write (nout,99999) 'Results for N =', n
          Write (nout,*)
          Write (nout,99996) 'Solution on first ', n, &
            ' Chebyshev points and Chebyshev coefficients'
          Write (nout,*) '  I            X             F(I)          C(I)'
          Write (nout,99998)(i,cos(x01aaf(a)*real(i,kind=nag_wp)/real(2*n-1, &
            kind=nag_wp)),f(i),c(i),i=1,n)


!         Evaluate and print solution on uniform grid.
          ifail = 0
          Call c06dcf(x,lx,a,b,c,n,ss,chebr,ifail)

          Write (nout,*)
          Write (nout,*) 'Solution on evenly spaced grid'
          Write (nout,*) '     X           F(X)'
          Write (nout,99997)(x(i),chebr(i),i=1,lx)

        End If

        Deallocate (c,cm,f,f1,wk)
      End Do
      Deallocate (x,chebr)

99999 Format (1X,A,I3)
99998 Format (1X,I3,2F15.5,E15.5)
99997 Format (1X,F8.4,F15.5)
99996 Format (1X,A,I2,A)
    End Program d05abfe
```

9.2 Program Data

None.

9.3 Program Results

```
 D05ABF Example Program Results

 Solution is even

 Results for N =  5

 Solution on first  5 Chebyshev points and Chebyshev coefficients
   I            X             F(I)          C(I)
   1        0.93969        0.75572    0.14152E+01
   2        0.76604        0.74534    0.49384E-01
   3        0.50000        0.71729   -0.10476E-02
   4        0.17365        0.68319   -0.23282E-03
   5       -0.17365        0.66051    0.20890E-04

 Solution on evenly spaced grid
```

```
     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

 Solution on first 10 Chebyshev points and Chebyshev coefficients
   I           X             F(I)          C(I)
   1        0.98636        0.75572    0.14152E+01
   2        0.94582        0.75336    0.49384E-01
   3        0.87947        0.74639   -0.10475E-02
   4        0.78914        0.73525   -0.23275E-03
   5        0.67728        0.72081    0.19986E-04
   6        0.54695        0.70452    0.98675E-06
   7        0.40170        0.68825   -0.23796E-06
   8        0.24549        0.67404    0.18581E-08
   9        0.08258        0.66361    0.24483E-08
  10       -0.08258        0.65812   -0.16527E-09

 Solution on evenly spaced grid
     X           F(X)
   0.0000        0.65742
   0.2500        0.66384
   0.5000        0.68319
   0.7500        0.71489
   1.0000        0.75572
```