NAG Library Routine Document

C05BBF

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

C05BBF computes the values of Lambert's W function $W(z)$.

2 Specification

```
SUBROUTINE C05BBF (BRANCH, OFFSET, Z, W, RESID, IFAIL)
INTEGER                 BRANCH, IFAIL
REAL (KIND=nag_wp)      RESID
COMPLEX (KIND=nag_wp)   Z, W
LOGICAL                 OFFSET
```

3 Description

C05BBF calculates an approximate value for Lambert's W function (sometimes known as the 'product log' or 'Omega' function), which is the inverse function of

$$f(w) = we^w \quad \text{for} \quad w \in C.$$

The function f is many-to-one, and so, except at 0, W is multivalued. C05BBF allows you to specify the branch of W on which you would like the results to lie by using the argument BRANCH. Our choice of branch cuts is as in Corless *et al.* (1996), and the ranges of the branches of W are summarised in Figure 1.

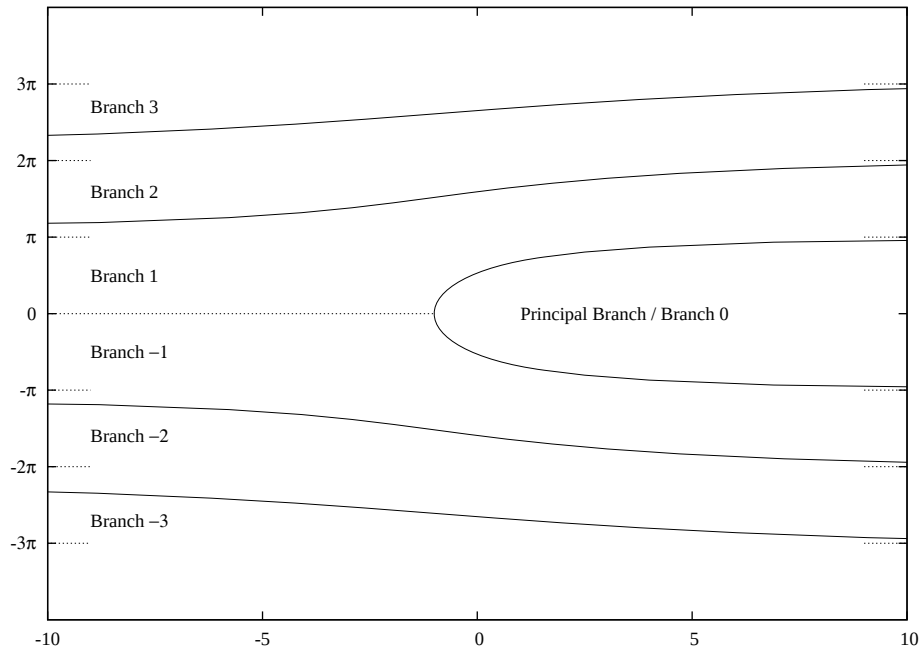

Figure 1
Ranges of the branches of $W(z)$

For more information about the closure of each branch, which is not displayed in Figure 1, see Corless *et al.* (1996). The dotted lines in the Figure denote the asymptotic boundaries of the branches, at multiples of π.

The precise method used to approximate W is as described in Corless *et al.* (1996). For z close to $-\exp(-1)$ greater accuracy comes from evaluating $W(-\exp(-1)+\Delta z)$ rather than $W(z)$: by setting OFFSET = .TRUE. on entry you inform C05BBF that you are providing Δz, not z, in Z.

4 References

Corless R M, Gonnet G H, Hare D E G, Jeffrey D J and Knuth D E (1996) On the Lambert W function *Advances in Comp. Math.* **3** 329–359

5 Arguments

1: BRANCH – INTEGER *Input*

On entry: the branch required.

2: OFFSET – LOGICAL *Input*

On entry: controls whether or not Z is being specified as an offset from $-\exp(-1)$.

3: Z – COMPLEX (KIND=nag_wp) *Input*

On entry: if OFFSET = .TRUE., Z is the offset Δz from $-\exp(-1)$ of the intended argument to W; that is, $W(\beta)$ is computed, where $\beta = -\exp(-1)+\Delta z$.

If OFFSET = .FALSE., Z is the argument z of the function; that is, $W(\beta)$ is computed, where $\beta = z$.

4: W – COMPLEX (KIND=nag_wp) *Output*

On exit: the value $W(\beta)$: see also the description of Z.

5: RESID – REAL (KIND=nag_wp) *Output*

On exit: the residual $|W(\beta)\exp(W(\beta))-\beta|$: see also the description of Z.

6: 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, because for this routine the values of the output arguments may be useful even if IFAIL $\neq 0$ on exit, the recommended value is -1. **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).

Note: C05BBF may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

IFAIL $= 1$

For the given offset Z, W is negligibly different from -1: Re(Z) $= \langle value \rangle$ and Im(Z) $= \langle value \rangle$.

Z is close to $-\exp(-1)$. Enter Z as an offset to $-\exp(-1)$ for greater accuracy: $\mathrm{Re(Z)} = \langle value \rangle$ and $\mathrm{Im(Z)} = \langle value \rangle$.

IFAIL $= 2$

The iterative procedure used internally did not converge in $\langle value \rangle$ iterations. Check the value of RESID for the accuracy of W.

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

For a high percentage of Z, C05BBF is accurate to the number of decimal digits of precision on the host machine (see X02BEF). An extra digit may be lost on some platforms and for a small proportion of Z. This depends on the accuracy of the base-10 logarithm on your system.

8 Parallelism and Performance

C05BBF is not threaded in any implementation.

9 Further Comments

The following figures show the principal branch of W.

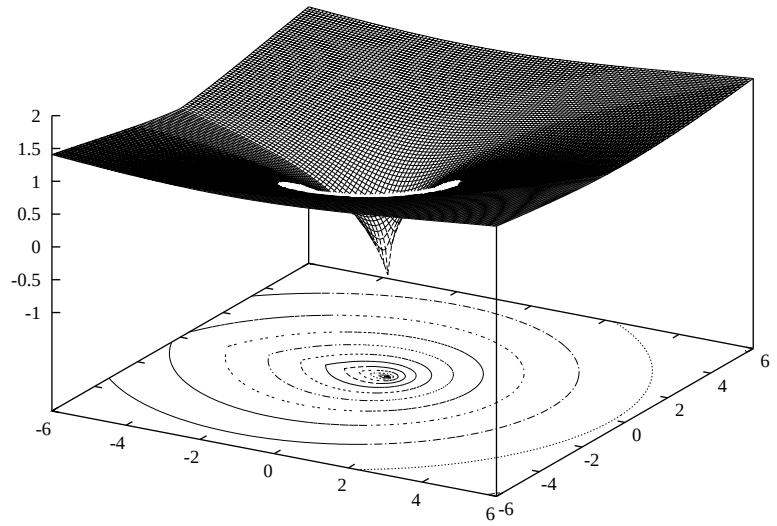

Figure 2
real$(W_0(z))$

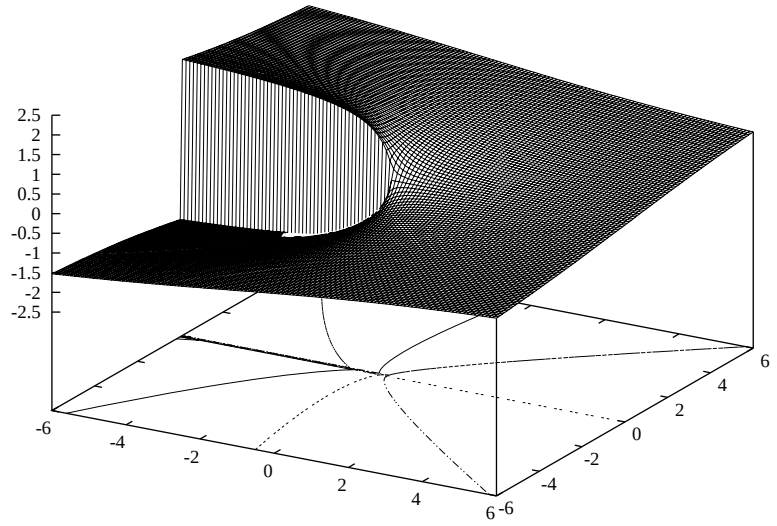

Figure 3
$\mathrm{Im}(W_0(z))$

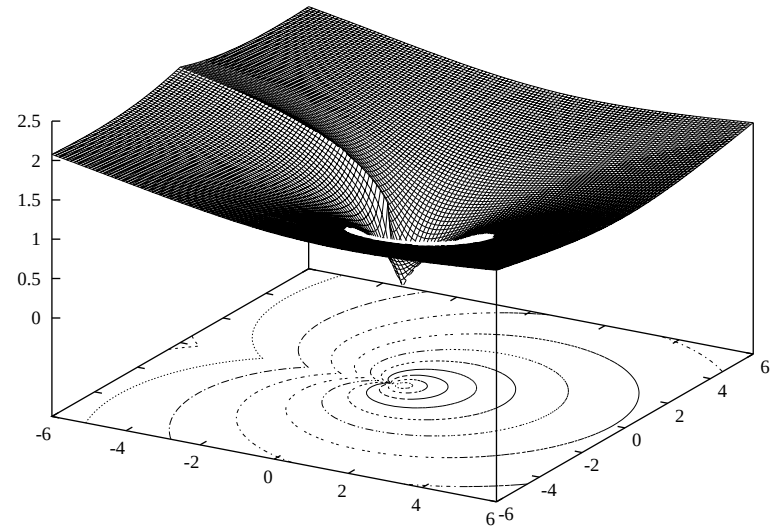

Figure 4
$\mathrm{abs}(W_0(z))$

10 Example

This example reads from a file the value of the required branch, whether or not the arguments to W are to be considered as offsets to $-\exp(-1)$, and the arguments Z themselves. It then evaluates the function for these sets of input data Z and prints the results.

10.1 Program Text

```
    Program c05bbfe

!     C05BBF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: c05bbf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                :: nin = 5, nout = 6
```

```
!     .. Local Scalars ..
      Complex (Kind=nag_wp)            :: w, z
      Real (Kind=nag_wp)               :: resid
      Integer                          :: branch, ifail, ioerr
      Logical                          :: offset
!     .. Executable Statements ..
      Write (nout,*) 'C05BBF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

      Read (nin,*) branch
      Read (nin,*) offset

      Write (nout,*)
      Write (nout,99997) 'BRANCH = ', branch

      If (offset) Then
        Write (nout,99996) 'OFFSET = .TRUE.'
      Else
        Write (nout,99996) 'OFFSET = .FALSE.'
      End If

      Write (nout,*)
      Write (nout,99999)
      Write (nout,*)

data: Do
        Read (nin,*,Iostat=ioerr) z

        If (ioerr<0) Then
          Exit data
        End If

        ifail = -1
        Call c05bbf(branch,offset,z,w,resid,ifail)

        If (ifail<0) Then
          Exit data
        End If

        Write (nout,99998) z, w, resid, ifail
      End Do data

99999 Format (1X,14X,'Z',28X,'W(Z)',18X,'RESID',4X,'IFAIL')
99998 Format (1X,1P,2('(',E13.5,',',E13.5,')',1X),E13.5,1X,I5)
99997 Format (1X,A,I3)
99996 Format (1X,A)
    End Program c05bbfe
```

10.2 Program Data

```
C05BBF Example Program Data
  0                                                    : BRANCH
  .FALSE.                                              : OFFSET
  (0.5, -1.0)
  (1.0, 2.3)
  (4.5, -0.1)
  (6.0, 6.0)                                           : Z
```

10.3 Program Results

```
 C05BBF Example Program Results

 BRANCH =   0
 OFFSET = .FALSE.

              Z                            W(Z)                  RESID    IFAIL

 (  5.00000E-01, -1.00000E+00) (  5.16511E-01, -4.22053E-01)   5.55112E-17     0
 (  1.00000E+00,  2.30000E+00) (  8.73606E-01,  5.76978E-01)   1.11022E-16     0
 (  4.50000E+00, -1.00000E-01) (  1.26735E+00, -1.24194E-02)   0.00000E+00     0
 (  6.00000E+00,  6.00000E+00) (  1.61492E+00,  4.90515E-01)   1.25607E-15     0
```