NAG Toolbox

nag_roots_lambertw_complex (c05bb)

1 Purpose

nag_roots_lambertw_complex (c05bb) computes the values of Lambert's W function $W(z)$.

2 Syntax

```
[w, resid, ifail] = nag_roots_lambertw_complex(branch, offset, z)
[w, resid, ifail] = c05bb(branch, offset, z)
```

3 Description

nag_roots_lambertw_complex (c05bb) 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. nag_roots_lambertw_complex (c05bb) 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.

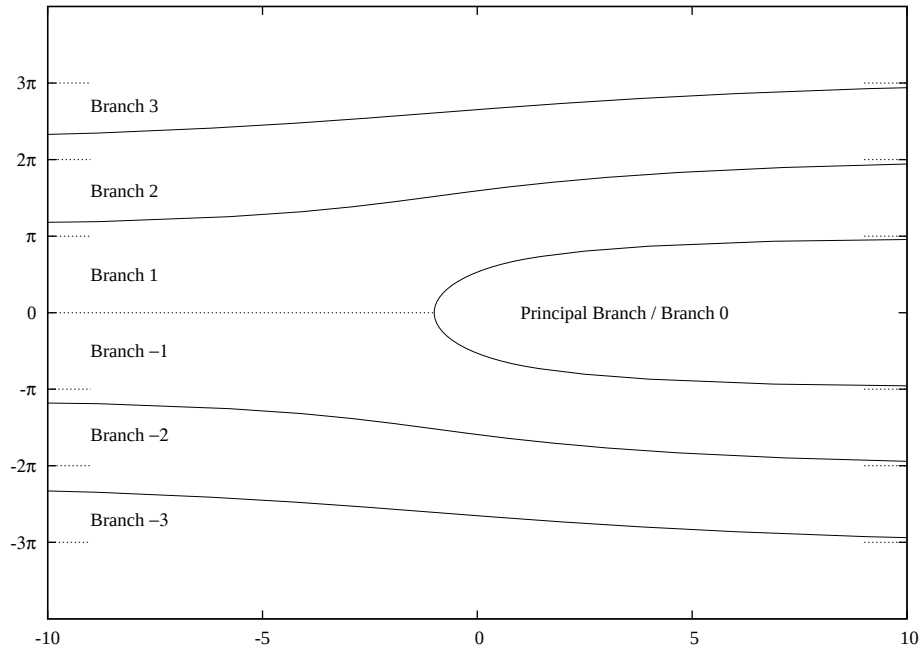

Figure 1
Ranges of the branches of $W(z)$

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 on entry you inform nag_roots_lambertw_complex (c05bb) 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 Parameters

5.1 Compulsory Input Parameters

1: **branch** – INTEGER

The branch required.

2: **offset** – LOGICAL

Controls whether or not **z** is being specified as an offset from $-\exp(-1)$.

3: **z** – COMPLEX (KIND=nag_wp)

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

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **w** – COMPLEX (KIND=nag_wp)

The value $W(\beta)$: see also the description of **z**.

2: **resid** – REAL (KIND=nag_wp)

The residual $|W(\beta)\exp(W(\beta)) - \beta|$: see also the description of **z**.

3: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Note: nag_roots_lambertw_complex (c05bb) may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the function:

ifail $= 1$ (*warning*)

For the given offset **z**, W is negligibly different from -1.

z is close to $-\exp(-1)$.

ifail $= 2$ (*warning*)

The iterative procedure used internally did not converge in ⟨*value*⟩ 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.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

For a high percentage of **z**, nag_roots_lambertw_complex (c05bb) is accurate to the number of decimal digits of precision on the host machine (see nag_machine_decimal_digits (x02be)). 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 Further Comments

The following figures show the principal branch of W.

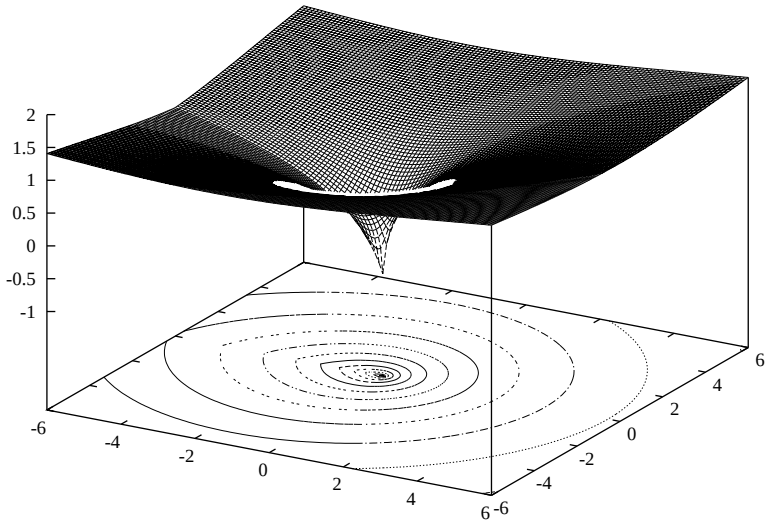

Figure 2
real$(W_0(z))$

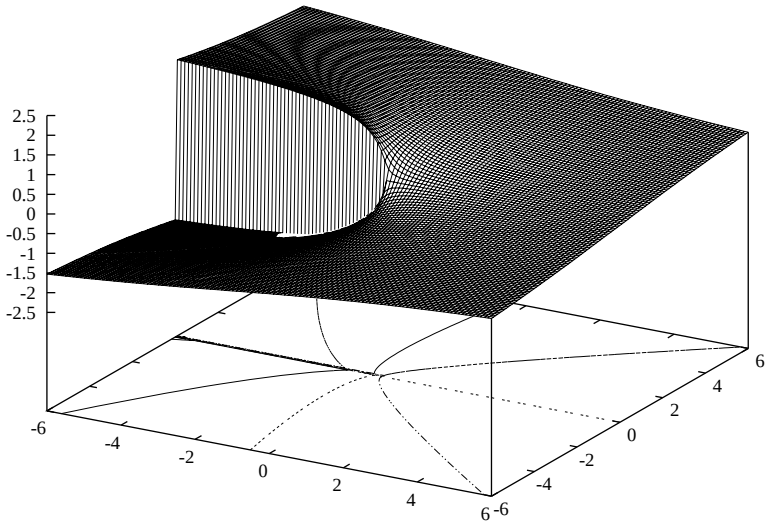

Figure 3
Im$(W_0(z))$

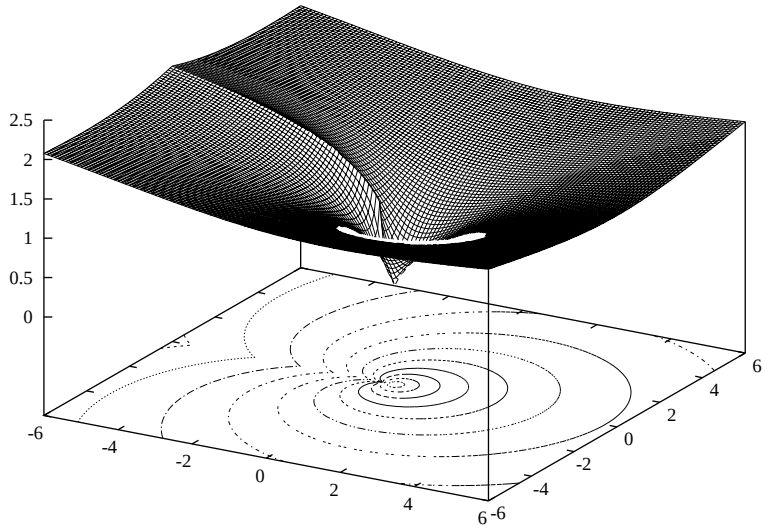

Figure 4
$\mathrm{abs}(W_0(z))$

9 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.

9.1 Program Text

```
function c05bb_example

fprintf('c05bb example results\n\n');

branch = nag_int(0);
offset = false;
z = [0.5-i; 1+2.3*i; 4.5-0.1*i; 6+6*i];
fprintf('\nBranch = %d\n', branch);
if offset
  fprintf('Offset = true\n');
else
  fprintf('Offset = false\n');
end
fprintf('\n%12s%16s%18s%9s\n','z','w','resid','ifail');
for j =1:4
  [w, resid, ifail] = c05bb(branch, offset, z(j));
  fprintf('%10.1f %4.1fi', real(z(j)), imag(z(j)))
  fprintf('%10.5f %8.5fi %12.5e %3d\n', real(w), imag(w), resid, ifail);
end
```

9.2 Program Results

```
c05bb example results

Branch = 0
Offset = false

           z               w              resid    ifail
       0.5 -1.0i   0.51651 -0.42205i  5.55112e-17   0
       1.0  2.3i   0.87361  0.57698i  1.11022e-16   0
       4.5 -0.1i   1.26735 -0.01242i  0.00000e+00   0
       6.0  6.0i   1.61492  0.49051i  1.25607e-15   0
```