Branch Cuts in Maple 17

Matthew England, Russell Bradford, James Davenport and David Wilson
Department of Computer Science, University of Bath, Bath, UK, BA2 7AY
{M.England,R.J.Bradford,J.H.Davenport,D.J.Wilson}@bath.ac.uk

Abstract

Accurate and comprehensible knowledge of branch cuts is essential for correctly working with multi-valued functions, such as the square root and logarithm. We present new tools in Maple 17 for calculating and visualising the branch cuts of such functions, and others built up from them. The cuts are described in a more intuitive and accurate form, offering substantial improvement on the descriptions previously available. The package is rigorous, clearly identifying the situations where output can be guaranteed and the issues encountered.

1 Introduction

When defining multi-valued functions (such as the natural logarithm and the square root) choices must be made for the positioning of the branch cuts. The standard choices are given in [1] and [10] and Maple agrees for all elementary functions except arccot, (for reasons explained in [5]). There are different, often unstated, viewpoints for dealing with multi-valued functions [6]. Maple along with most computer algebra software (and indeed most users), works with multi-valued functions by defining their single-valued counterparts which will have discontinuities over the branch cuts. It is important that users of these functions understand the position of such cuts.

We present the `BranchCuts` package, a new piece of code introduced in Maple 17 as part of the `FunctionAdvisor` project, [4]. The code calculates and visualises the branch cuts of functions of a single complex variable. It is used by default in Maple 17 while readers with older versions can download a version with the main functionality from `http://opus.bath.ac.uk/32511/`.

In [9] we presented in detail the generic algorithms which could be used for calculating branch cuts. Here we focus on the practical improvements offered by the new Maple routines.

2 Branch cuts in Maple

Information on Maple's predefined functions may be accessed with the `FunctionAdvisor` tool: a handbook for special functions, designed to be both human and machine readable, and process output to fit the query [4]. We can view the entries of the table of branch cut definitions by querying a function name without argument. For example,

```
> FunctionAdvisor(branch_cuts, ln);
```

$$[\ln(z), z < 0] \tag{1}$$

indicates the logarithm has a branch cut along the negative real axis. In MAPLE 16, executing the command with an argument returns the same output with z replaced by the argument. For example,

```
> FunctionAdvisor(branch_cuts, ln(z^2));
```

$$[\ln(z^2), z^2 < 0] \tag{2}$$

indicating that the function has a branch cut when z^2 takes negative real values.

From this we can infer that the cuts are along the imaginary axis excluding zero. The statement is accurate but it would be more useful to return the imaginary axis directly. Worse still is the example,

```
> FunctionAdvisor(branch_cuts, ln(-sqrt(z)));
```

$$[\ln(-\sqrt{z}), -\sqrt{z} < 0] \tag{3}$$

from which we could infer that the function has a branch cut along the positive real axis. While this branch cut exists, the function is also discontinuous along the negative real axis, due to the branch cut of the square root function. Hence the output did not imply all branch cuts of the function.

Figure 1 presents 3d plots of the imaginary part of these three functions, from which the actual discontinuities are clear. The imaginary part is given because moving across the discontinuity causes the logarithm to change value by $2\pi\mathrm{i}$. For other functions such as arctan the discontinuity may appear in the real part. While users can infer the actual position of branch cuts from such plots, for more complicated examples this would become trickier and tiresome. The `BranchCuts` package gives accurate and intuitive descriptions of such branch cuts, used in turn to give simple 2d visualisations.

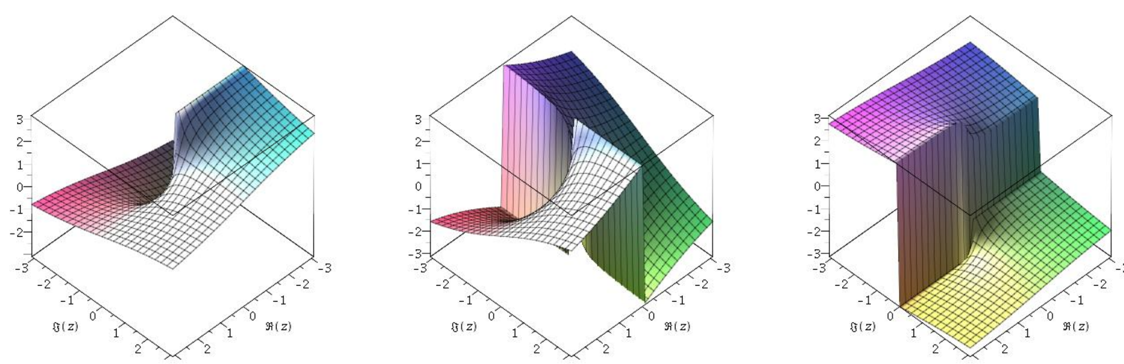

Figure 1: Plots of the imaginary parts of the three functions from Section 2.

3 Improved representing and calculation of branch cuts

Two approaches have been implemented for representing and calculating branch cuts. The first moves the problem from one complex variable z to two real variables, $z = \Re(z) + i\Im(z)$. Then each branch cut is represented as an equation defining either $\Re(z)$ or $\Im(z)$ in terms of the other along with inequations bounding the curve to the appropriate regions. So for example the branch cut for the natural logarithm is given by the equation $\Im(z) = 0$ and inequality $\Re(z) < 0$. Then when considering more complicated functions we solve the resulting semi-algebraic set to find a set of solutions. Hence in MAPLE 17 the second example from Section 2 becomes

```
> FunctionAdvisor(branch_cuts, ln(z^2));
```

$$[\ln(z^2), And(\Re(z) = 0, 0 < \Im(z)), And(\Re(z) = 0, \Im(z) < 0)] \tag{2'}$$

which is as accurate as (2) but is more useful for the user and subsequent computations.

This approach works well when the argument of the multi-valued function is a polynomial. When the argument contains fractional powers then it requires a de-nesting procedure, for which there is currently only a limited implementation. For these examples a second approach is used where cuts are stored as a (possibly complex) function of a real parameter and a range for that parameter, similar to [8]. So for example the branch cut for the natural logarithm is given by $z = a$ and the range $a \in (-\infty, 0)$. Then when considering more complicated functions we need to work only with the equation, applying the solutions with the range.

Both approaches compute the branch cuts of combinations of functions (compositions, sums, products and relations) by taking the union of the individual cuts. This, coupled with the second approach result in MAPLE 17 giving the following output for the third example of Section 2.

```
> FunctionAdvisor(branch_cuts, ln(-sqrt(z)));
```

$$[\ln(-\sqrt{(}z)), And(z = \alpha^2, \alpha \in RealRange(-\infty, 0)), z < 0] \tag{3'}$$

which is both more intuitive and more accurate than (3).

Detailed descriptions of the algorithms we used are presented in [9]. We also developed plotting procedures to turn output from both representations into simple 2d visualisations of the branch cuts, as on the right of Figure 2. These, along with various 3d plots, are available from the `FunctionAdvisor` in MAPLE 17 using an optional argument.

We note that the implementation can consider any function whose defining cuts are stored in the `FunctionAdvisor` table. So as well as elementary functions such as the logarithm and square root we may also consider inverse hyperbolic functions and integral functions for example. There is also functionality for multivariate functions whose branch cuts are in only one variable. Hence many univariate functions with parameters including Bessel functions, Chebyshev polynomials and Jacobi θ-functions are covered, especially important given the increased use of such special functions in modelling.

We note that understanding the position of branch cuts can also have applications in computer algebra itself, for example, in the safe application of identities. Consider

$$2\arcsin(z) = \arcsin(2z\sqrt{1 - z^2}). \tag{4}$$

The values of the real part of LHS(4)−RHS(4) are plotted on the left of Figure 2, while the image on the right is derived from the description of the branch cuts produced by the new code. We find

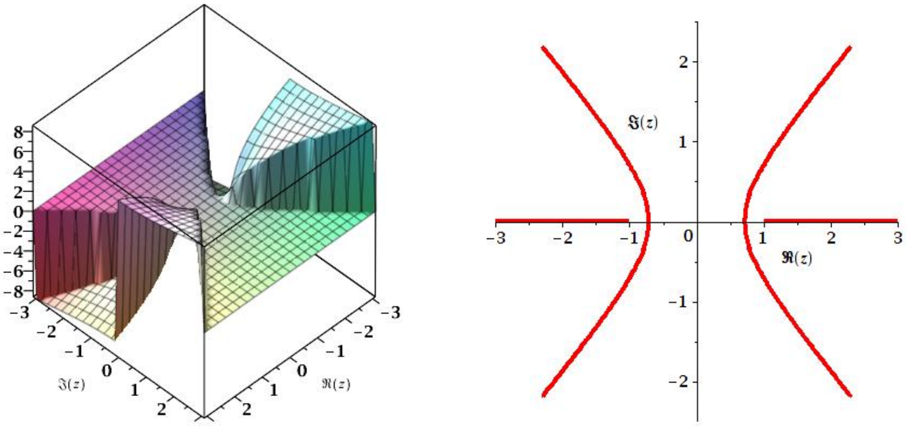

Figure 2: Visualising the branch cuts of equation (4).

that the identity is true in the hour glass shaped region bounded by the hyperbola $y^2 - x^2 + 1 = 0$ and false otherwise. Algorithmic approaches to identifying the regions where such identities hold using branch cuts and cylindrical algebraic decompositions have been studied in a series of papers starting with [2], with recent progress reported in [7]. We note that the new Cylindrical algebraic decomposition algorithm reported in [3] is particularly well suited for dealing with input from our first branch cuts representation.

4 When is a branch cut not a branch cut?

For very complicated functions failure could occur due to time/memory constraints, or the MAPLE solve commands being unable to identify all solutions. However, for all practical examples encountered these problems have not occurred. A more prevalent issue is branch cuts being returned which do not correspond to discontinuities of the input. We call these *spurious cuts* and there are two reasons for their creation. First, when functions are combined the discontinuities introduced may cancel with each other. Second, when the argument of a function contains fractional powers then these need to be de-nested before solutions can be found, with the solution set then corresponding to all possible roots. These issues are examined in more detail in [9] with methods to remove these spurious cuts currently under development.

The package in MAPLE 17 uses a cautious approach, preferring to include branch cuts that may be spurious rather than risk losing true cuts. This approach has also led to a warning system, so that users may apply the algorithms widely, but are informed of the risks. For example,

```
> FunctionAdvisor(branch_cuts, ln(exp(z)));
Warning, branch cuts have been returned for ln(exp(z)) but we do not guarantee
an exhaustive list
```

$$[\ln(\exp(z)), And(z = \ln(\alpha), \alpha \in RealRange(-\infty, 0))] \tag{5}$$

Only the cut in the principal domain has been returned, while the function has infinitely many. Future releases of the code will return the full set, or the set within a specified region.

Acknowledgements

This work was supported by the EPSRC grant: EP/J003247/1. The code in MAPLE 17 is a collaboration between the University of Bath and Maplesoft. We would particularly like to thank Edgardo Cheb-Terrab from Maplesoft for his interest in the work and contribution to the code.

References

[1] M. Abramowitz and I.A. Stegun. *Handbook of mathematical functions.* National Bureau of Standards, 1964.

[2] R. Bradford and J.H. Davenport. Towards better simplification of elementary functions. In *Proceedings of the 2002 international symposium on symbolic and algebraic computation*, ISSAC '02. ACM, 2002.

[3] R. Bradford, J.H. Davenport, M. England, S. McCallum, and D. Wilson. Cylindrical algebraic decompositions for boolean combinations. *In Press: Proc. ISSAC 2013. Preprint at* *http://opus.bath.ac.uk/33926/*, 2013.

[4] E.S. Cheb-Terrab. The function wizard project: A computer algebra handbook of special functions. In *Proceedings of the Maple Summer Workshop. University of Waterloo, Canada*, 2002.

[5] R.M. Corless, J.H. Davenport, D.J. Jeffrey, and S.M. Watt. According to Abramowitz and Stegun. *SIGSAM Bulletin*, 34(3):58–65, 2000.

[6] J.H. Davenport. The challenges of multivalued "functions". *Lecture Notes in Computer Science*, 6167:1–12, 2010.

[7] J.H. Davenport, R. Bradford, M. England, and D. Wilson. Program verification in the presence of complex numbers, functions with branch cuts etc. In *14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, SYNASC 2012, pages 83–88, 2012.

[8] A. Dingle and R.J. Fateman. Branch cuts in computer algebra. In *Proceedings of the international symposium on symbolic and algebraic computation*, ISSAC '94, pages 250–257. ACM, 1994.

[9] M. England, R. Bradford, J.H. Davenport, and D. Wilson. Understanding branch cuts of expressions. *Submitted for publication.* Preprint at http://opus.bath.ac.uk/32511/, 2013.

[10] National Institute for Standards and Technology. The NIST digital library of mathematical functions. http://dlmf.nist.gov, 2010.