# What and Where are Branch Cuts?

James Davenport (Bath)

based on ideas developed in 2000, 2009 with Corless, Jeffrey, Watt and Moreno Maza, work done at INRIA/Microsoft, Saclay, France with Frédéric Chyzak, Christoph Koutschan (RISC), and Bruno Salvy, and at Bath with Russell Bradford, Matthew England (Coventry) and David Wilson

1 February 2016

# What is a "Function" and how do we evaluate it?

Bourbaki A left-total, right-unique relation

o.d.e. analytic continuation from initial conditions

inverse analytic continuation from initial point

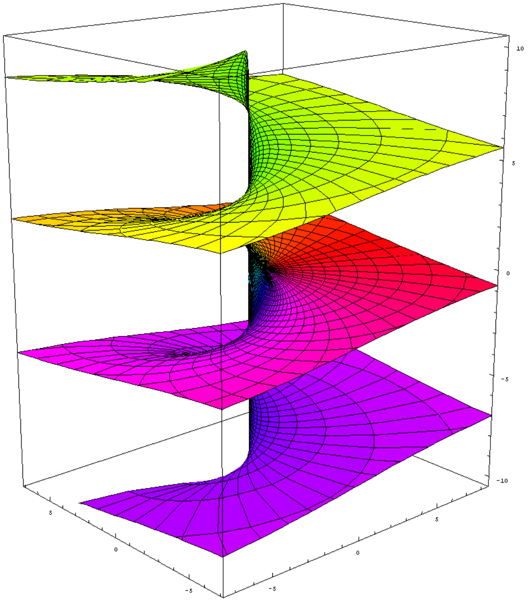
differential algebra "what do you mean, evaluate?"

But these are incompatible: $\log \stackrel{?}{=} \int \frac{1}{x} \stackrel{?}{=} \exp^{-1} \stackrel{?}{=}$ "$\theta : \theta' = \frac{1}{x}$".
Note that computer algebra is fundamentally in the "differential algebra" mindset

# Various solutions

- Multivalued Functions — Passing the buck forwards: "which $f(x)$ do you want"
- Riemann surfaces — Passing the buck backwards: "which Riemann sheet did that $x$ come from"
- "in a suitably chosen open subset" — passing the buck upwards (the traditional textbook method)
- branch cuts — biting the bullet; sacrificing continuity for uniqueness $\mathbf{C} \to \mathbf{C}$

# A Riemann surface example: log



# As well as sacrificing continuity,

We also sacrifice (some) identities

- $\log(1/z) = -\log z$ except on the negative real axis

  $\mathrm{Log}(-1) = \{(2k+1)i\pi\} = -\{(2k+1)i\pi\}$

  $\log(-1) = i\pi \neq -\log(-1)$

Riemann $1/(-1) = -1$ on a different sheet!

- $\sqrt{z-1}\sqrt{z+1} = \sqrt{z^2-1}$ only when $\Re(z) \geq 0$

But $\sqrt{1-z^2} = \sqrt{1-z}\sqrt{1+z}$ everywhere

Hence the question is: which identities, and where?

# This is not just a theorist's paradox [Kah87, pp. 187–188]

With the usual definitions,

$$g(z) := 2\,\mathrm{arccosh}\left(1 + \frac{2z}{3}\right) - \mathrm{arccosh}\left(\frac{5z+12}{3(z+4)}\right)$$

is only the same as the ostensibly more efficient

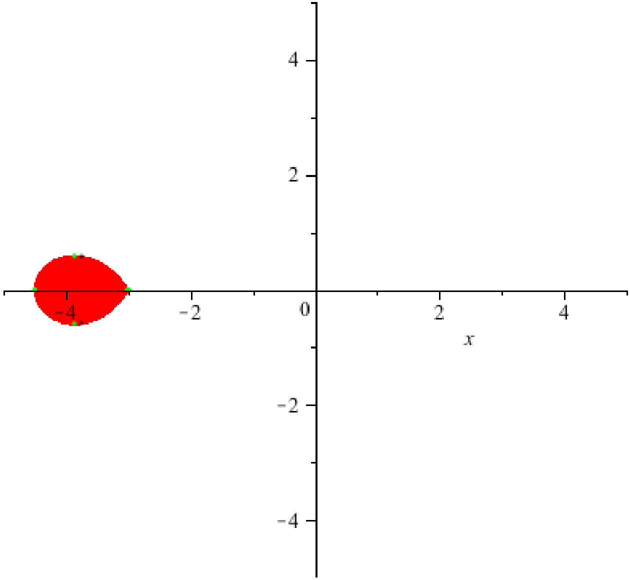
$$q(z) := 2\,\mathrm{arccosh}\left(2(z+3)\sqrt{\frac{z+3}{27(z+4)}}\right),$$

If we avoid the negative real axis and the area

$$\left\{z = x + iy : |y| \le \sqrt{\frac{(x+3)^2(-2x-9)}{2x+5}} \wedge -9/2 \le x \le -3\right\}$$

- So the cuts matter.

# The erroneous region



# Then why branch cuts?

The table maker's/library software writer's challenge: assign a *definite meaning* to log (or $\operatorname{arccot}$ or ...) as a function $\mathbf{C} \to \mathbf{C}$.

If we are to implement a library of functions, they must have a *context-free* definition.

What do we put in our catalogue, and what relationship(s) will exist between the objects in the catalogue?

How will we, or the writer of verified software based on our library, reason about these branch cuts?

# Two design questions, and a use question

positioning Where do we put the cuts?

adherence What are the values on the cuts themselves?

"There can be no dispute about where to put the slits; their locations are deducible. However, Principal Values have too often been left ambiguous on the slits." [Kah87]

* Note that Kahan advocated "signed zeros", so his cuts could adhere to both sides:

$$\frac{1}{1+0i} = 1 - 0i$$

but this only works for cuts along the axes!

$\log(z)$ but not $\log(z+i)$

reasoning How do we reason about these cuts?

# Making a table/dictionary/library/...

History: Chosen by the author(s)

Often poorly documented, and occasionally incompatible (Matlab in 2000!)

A+S/DLMF: `http://dlmf.nist.gov` A serious attempt to systematize history, based on careful hand editing

DDMF: `http://ddmf.msr-inria.inria.fr` A serious attempt to *automate* the analysis of special functions

# Kahan's rules for the table maker

R1 These functions $f$ are extensions to $\mathbf{C}$ of a real elementary function analytic at every interior point of its domain, which is a segment $\mathcal{S}$ of the real axis.

R2 Therefore, to preserve this analyticity (i.e. the convergence of the power series), the slits cannot intersect the interior of $\mathcal{S}$.

R3 Since the power series for $f$ has real coefficients, $f(\overline{z}) = \overline{f(z)}$ in a complex neighbourhood of the segment's interior, so this should extend throughout the range of definition. So complex conjugation should map slits to themselves.

R4 Similarly, the slits of an odd function should be invariant under reflection in the origin, i.e. $z \to -z$.

R5 The slits must begin and end at singularities.

While these rules are satisfied by the branch cuts of elementary building blocks [Nat10], we must add a form of Occam's razor:

R6 The slits might as well be straight lines.

# Wasn't there an issue about $\operatorname{arccot}$?

Indeed so: between the first and ninth printings of [AS64]

$$\operatorname{arccot}_1(x) = \pi/2 - \arctan(x),$$
$$\operatorname{arccot}_9(x) = \arctan(1/x).$$

$\operatorname{arccot}_1(x)$ is defined on $\mathcal{S} = (-\infty, \infty)$, but $\operatorname{arccot}_9(x)$ on $\mathcal{S} = (0, \infty] \overset{\text{cts}}{\cup} [-\infty, 0)$, hence

R1 These functions $f$ are extensions to **C** of a real elementary function analytic at every interior point of its domain, which is a segment $\mathcal{S}$ of the real axis.

R2 Therefore, to preserve this analyticity (i.e. the convergence of the power series), the slits cannot intersect the interior of $\mathcal{S}$.

work differently

# We extend Kahan's rules to o.d.e.s $L(y) = 0$ & initial value

- R2' The branch cuts do not enter the circle of convergence (with respect to the given initial value).
- Therefore different initial values might give rise to different branch cuts
- R3' Complex conjugation is respected.
- R4' Any symmetries inherent in the power series are respected.
- R5' The branch cuts begin and end at singularities.
- R6 The branch cuts are straight lines.
- * Compatible with cuts in [Nat10]

These rules *do not* necessarily completely determine the branch cut: a "random" differential equation with singularities scattered in the complex plane and no special symmetries will not be determined.

# An example: $x(1+x^4)f'' + (3x^4-1)f' = 0$; $f(0) = f'(0) = 0; f''(0) = 2$

The equation has four regular singularities at $z = \pm\sqrt{\pm i} = \frac{\pm 1 \pm i}{\sqrt{2}}$

(R6) These four singularities have to be connected by straight lines.

(R2′) We cannot connect the singularities pairwise (in either way!) without going to infinity.

(R4′) The symmetry $f(ix) = -f(x)$ can be checked directly from the equation, so that branch cuts should be mapped to branch cuts by a rotation of $\pi/2$.

(R3′) Reality implies that branch cuts are also mapped to branch cuts by horizontal symmetry.

We are thus left with only the following choice: cuts that "head northeast" from $\frac{1+i}{\sqrt{2}}$, "northwest" from $\frac{-1+i}{\sqrt{2}}$ etc., all meeting at infinity. This is consistent with $\arctan(x^2)$, a solution of $L$.

# Series?

In the world of (generalized) power series, the branch cuts

- Only appear in the expansions about the singular points
- Their directions are coded in the arguments

a) $\log z =$ branch cut heading west, adhering north

b) $\log(iz) =$ branch cut heading north, etc.

- Their adherence is coded similarly

c) $-\log(1/z) =$ branch cut west, adhering south

# Traditional Classification of Verification Problems

How often are they considered?
Statistics from a verification conference [CE05]

- **blunder (83%)** (of the coding variety) This is the sort of error traditionally addressed in "program verification". Typically independent of the arithmetic.
- **parallelism (13%)** Issues of deadlocks or races occurring due to the parallelism of an otherwise correct sequential program. Again, arithmetic-independent.
- **numerical (3%)** Do truncation and round-off errors, individually or combined, mean that the program computes approximations to the "true" answers which are out of tolerance.

To this, I wish to add a fourth kind

# “The bug that dares not speak its name”

manipulation A piece of algebra, which is “obviously correct”,

(0%!) turns out not to be correct when interpreted, not as abstract algebra, but as the manipulation of functions $\mathbf{R} \to \mathbf{R}$ or $\mathbf{C} \to \mathbf{C}$.

# A note on complex numbers

Most of our examples involve complex numbers, and people say

*real programs don't use complex numbers*

However

- `COMPLEX` in Fortran II (1958–61) was the first programming language data type not corresponding to a machine one
- Even C99 introduced `_Complex`
- Many examples, notably in fluid mechanics.

Some "verified subsets" of programming languages have to exclude `complex` even though it is in the language

# Kahan's example [Kah87]

Flow in a slotted strip, transformed by

$$w = g(z) := 2\operatorname{arccosh}\left(1 + \frac{2z}{3}\right) - \operatorname{arccosh}\left(\frac{5z+12}{3(z+4)}\right) \qquad (1)$$

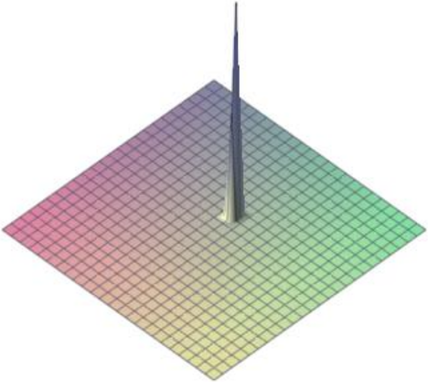
into a more tractable region.
Is this the same transformation as

$$w \stackrel{?}{=} q(z) := 2\operatorname{arccosh}\left(2(z+3)\sqrt{\frac{z+3}{27(z+4)}}\right)? \qquad (2)$$

Or possibly

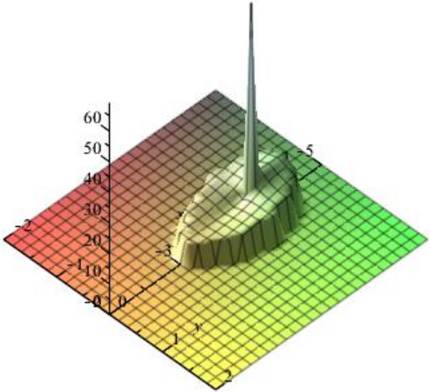
$$w \stackrel{?}{=} h(z) := 2\ln\left(\frac{1}{3}\,\frac{\sqrt{3\,z+12}\left(\sqrt{z+3}+\sqrt{z}\right)^2}{2\sqrt{z+3}+\sqrt{z}}\right)? \qquad (3)$$

# $g - q$ might look OK



"OK apart from a slight glitch."

# But if we look closer




Definitely not OK

# But, in fact $g = h$

Most computer algebra systems (these days!) will refuse to "simplify" $g$ to $q$

But will also refuse to simplify $g$ to $h$.

Indeed Maple's `coulditbe(g<>h);` returns `true`, which *ought* to indicate that there is a counter-example.

If $g = h$ then $g - h$ is zero:

$$\frac{d(g-h)}{dz} = 2\left(\sqrt{\frac{z}{z+4}}\sqrt{\frac{z+3}{z+4}}z^{3/2} - 2\,z^{3/2} + 2\,\sqrt{z+3}\sqrt{\frac{z}{z+4}}\cdot \sqrt{\frac{z+3}{z+4}}z - z\sqrt{z+3} + 4\,\sqrt{\frac{z}{z+4}}\sqrt{\frac{z+3}{z+4}}\sqrt{z} + 8\,\sqrt{z+3}\sqrt{\frac{z}{z+4}}\cdot \sqrt{\frac{z+3}{z+4}} - 6\,\sqrt{z}\right)\frac{1}{\sqrt{z+3}}\frac{1}{\sqrt{z}}\frac{1}{\sqrt{\frac{z}{z+4}}}\frac{1}{\sqrt{\frac{z+3}{z+4}}}(z+4)^{-2}\left(2\,\sqrt{z+3}+\sqrt{z}\right)^{-1}$$

and it's a bold person who would say "$= 0$"

# Methodology [BCD+02]

These branch cuts in $\mathbf{C}$ (more generally $\mathbf{C}^n$) have to be viewed as semi-algebraic curves in $\mathbf{R}^2$ (or $\mathbf{R}^{2n}$), to which we apply the methods of cylindrical algebraic decomposition [Col75] to deduce the geometry of the branch cuts on $\mathbf{R}^{2n}$, and the connected status of $\mathbf{R}^{2n}\setminus\{\text{branch cuts}\}$.

$$\sqrt{z-1}\sqrt{z+1}\stackrel{?}{=}\sqrt{z^2-1} \tag{4}$$

the branch cuts (in fact $z = 0 + iy$ of $\sqrt{z^2-1}$) disconnect $\mathbf{C} = \mathbf{R}^2$, but

$$\sqrt{1-z}\sqrt{1+z}\stackrel{?}{=}\sqrt{1-z^2} \tag{5}$$

does not have this problem, hence "true in a small region" implies "true except perhaps on the branch cuts"

# Challenges

**Challenge (1)**

*Demonstrate automatically that $g$ and $q$ are not equal, by producing a $z$ at which they give different results.*

The technology described in [BBDP07] will isolate the curve $y = \pm\sqrt{\frac{(x+3)^2(-2x-9)}{2x+5}}$ as a potential obstacle (it is the branch cut of $q$), but the geometry questions were too hard in 2009 (1143 cells) for a fully-automated solution. [BDE+14] reduces it to 39 cells.

**Challenge (2)**

*Demonstrate automatically that $g$ and $h$ are equal.*

Solved in principle, but the code has still to be written

# Joukowski (a)

Consider the Joukowski map [Hen74, pp. 294–298]:

$$f : z \mapsto \frac{1}{2}\left(z + \frac{1}{z}\right). \tag{6}$$

**Lemma**

*$f$ is injective as a function from $D := \{z : |z| > 1\}$.*

This is also a function $\mathbf{R}^2 \to \mathbf{R}^2$:

$$f_R : (x, y) \mapsto \left(\frac{1}{2}x + \frac{1}{2}\frac{x}{x^2+y^2}, \frac{1}{2}y - \frac{1}{2}\frac{y}{x^2+y^2}\right) \tag{7}$$

# Joukowski challenge (a1)

## Challenge (3)

*Demonstrate automatically that $f_R$ is injective, i.e.*

$$\forall x_1 \forall x_2 \forall y_1 \forall y_2 \quad \Bigg( x_1^2 + y_1^2 > 1 \wedge x_2^2 + y_2^2 > 1 \wedge x_1 + \frac{x_1}{x_1^2+y_1^2} = x_2 + \frac{x_2}{x_2^2+y_2^2} \wedge y_1 - \frac{y_1}{x_1^2+y_1^2} = y_2 - \frac{y_2}{x_2^2+y_2^2} \Bigg) \Rightarrow \big( x_1 = x_2 \wedge y_1 = y_2 \big). \tag{8}$$

We have failed to do this automatically, but Brown can reformulate manually then solve in QEPCAD ($<$ 12 seconds). Using Brown's reformulation, it was solved by CAD in [CM14, §4].

## Challenge (4)

*Automate these techniques and transforms.*

# Joukowski challenge (a2)

So it's a bijection: what's the inverse?

Figure: Maple's `solve` on inverting Joukowski

```
> [solve(zeta = 1/2*(z+1/z), z)];
```

$$\left[\zeta + \sqrt{\zeta^2 - 1}, \zeta - \sqrt{\zeta^2 - 1}\right]$$

The only challenge might be the choice implicit in the $\pm\sqrt{}$ idea: which do we choose? Unfortunately, the answer is "neither", or at least "neither, uniformly".

# Joukowski challenge (a2 continued)

$$f_1(\zeta) = \zeta \begin{cases} +\sqrt{\zeta^2 - 1} & \Im(\zeta) > 0 \\ -\sqrt{\zeta^2 - 1} & \Im\zeta) < 0 \\ +\sqrt{\zeta^2 - 1} & \Im(\zeta) = 0 \wedge \Re(\zeta) > 1 \\ -\sqrt{\zeta^2 - 1} & \Im(\zeta) = 0 \wedge \Re(\zeta) < -1 \end{cases} \qquad (9)$$

In fact, a better (at least, free of case distinctions) definition is

$$f_2(\zeta) = \zeta + \sqrt{\zeta - 1}\sqrt{\zeta + 1}. \qquad (10)$$

The techniques of [BBDP07] are able to **verify** (10), in the sense of showing that $f_2(f(z)) - z$ is the zero function on $\{z : |z| > 1\}$.

## Challenge (5)

*Derive automatically, and demonstrate the validity of, either (9) or (10). In terms of Maple, this would be ...*

# Joukowski challenge (a2 continued)

Figure: Bad: Maple's actual `solve` on inverting injective Joukowski

```
> [solve(zeta = 1/2*(z+1/z), z)] assuming abs(z) > 1
```

$$\left[\zeta + \sqrt{\zeta^2 - 1}, \zeta - \sqrt{\zeta^2 - 1}\right]$$

Figure: Good: Ideal software inverting injective Joukowski

```
> solve(zeta = 1/2*(z+1/z), z) assuming abs(z) > 1
```

$$\zeta + \sqrt{\zeta - 1}\sqrt{\zeta + 1}$$

As far as I can tell (supported by the documentation), Maple ignores the "`assuming`" as it's on the codomain, not the domain. Currently, we can't solve quadratics!

# Joukowski (b) challenge

## Lemma

*f is injective as a function from* $H := \{z : \Im z > 0\}$.

## Challenge (6)

*Demonstrate automatically the truth of*

$$\forall x_1 \forall x_2 \forall y_1 \forall y_2 \qquad \Big( y_1 > 0 \wedge y_2 > 0 \wedge$$
$$x_1 + \frac{x_1}{x_1^2+y_1^2} = x_2 + \frac{x_2}{x_2^2+y_2^2} \quad \wedge \quad y_1 - \frac{y_1}{x_1^2+y_1^2} = y_2 - \frac{y_2}{x_2^2+y_2^2} \Big)$$
$$\Rightarrow \qquad \big(x_1 = x_2 \wedge y_1 = y_2\big). \tag{11}$$

Brown's ideas apply here as well, and this is solved in [CM14, §4].

# Joukowski (b) challenge continued

So it's a bijection: what's the inverse?
[Hen74, (5.1-13), p. 298] argues for

$$f_3(\zeta) = \zeta + \underbrace{\sqrt{\zeta - 1}}_{\arg\in(-\pi/2,\pi/2]} \underbrace{\sqrt{\zeta + 1}}_{\arg\in(0,\pi]}. \qquad (12)$$

## Challenge (7)

*Find a way to represent functions such as* $\underbrace{\sqrt{\zeta + 1}}_{\arg\in(0,\pi]}$

# Alternative formulations

Fortunately this one is soluble in this case, we can write

$$\underbrace{\sqrt{\zeta+1}}_{\arg\in(0,\pi]} = i \underbrace{\sqrt{-\zeta-1}}_{\arg\in(-\pi/2,\pi/2]},$$

so we have an inverse function

$$f_4(\zeta) = \zeta + \sqrt{\zeta-1}\, i\sqrt{-\zeta-1}. \quad (13)$$

(I hate to think what an optimising compiler would do to this!)

**Challenge (8)**

*Demonstrate automatically that this is an inverse to $f$ on* $\{z : \Im z > 0\}$.

# Computational Problems

- A basic branch cut is simple: $x < 1 \wedge y = 0$ for $\operatorname{arccosh}(x + iy)$.
- For $\operatorname{arccosh}\left(\frac{5z+12}{3(z+4)}\right)$ we have $y = 0$ but

$$\frac{2}{3}\,\frac{29\,x^2 + 29\,y^2 + 122\,x + 24}{x^2 + y^2 + 8\,x + 16} < 1$$

- Not trivial to convert this to a polynomial inequality
- $N$ branch cuts have $2N$ equations, most of whose intersections are spurious
- Techniques of [BDE$^+$14] reduce this to intersections of $N$ equations
- (and reduced $g \neq q$ from 1143 cells to 39 cells)

# [BDE+14] in a nutshell

- Given two branch cuts $f_1 = 0 \wedge g_1 < 0$ and $f_2 = 0 \wedge g_2 < 0$, standard CAD considers 4 polynomials and hence 4 discriminants and 6 resultants.
- But $g_i$ is only relevant when $f_i = 0$
- Hence we only need $\mathrm{disc}(f_1)$, $\mathrm{disc}(f_2)$, $\mathrm{res}(f_1, f_2)$, $\mathrm{res}(f_1, g_1)$ and $\mathrm{res}(f_2, g_2)$ — 2,3 rather than 4,6
- and with $k$ branch cuts, $2k + \frac{k(k-1)}{2}$ derived polynomials rather than $2k + \frac{2k(2k-1)}{2}$ (asymptotically $1/4$)

# Bibliography


M. Abramowitz and I. Stegun.
Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, 9th printing.
*US Government Printing Office*, 1964.

J.C. Beaumont, R.J. Bradford, J.H. Davenport, and N. Phisanbut.
Testing Elementary Function Identities Using CAD.
*AAECC*, 18:513–543, 2007.

# Bibliography II


R.J. Bradford, R.M. Corless, J.H. Davenport, D.J. Jeffrey, and S.M. Watt.
Reasoning about the Elementary Functions of Complex Analysis.
*Annals of Mathematics and Artificial Intelligence*, 36:303–318, 2002.

R.J. Bradford, J.H. Davenport, M. England, S. McCallum, and D.J. Wilson.
Truth Table Invariant Cylindrical Algebraic Decomposition.
*To appear in J. Symbolic Computation*, 2014.

R. Cousot (Ed.).
Verification, Model Checking, and Abstract Interpretation.
*Springer Lecture Notes in Computer Science 3385*, 2005.

# Bibliography III


C. Chen and M. Moreno Maza.
Cylindrical Algebraic Decomposition in the RegularChains Library.
In *Proceedings Mathematical Software — ICMS 2014*, pages 425–433, 2014.

G.E. Collins.
Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition.
In *Proceedings 2nd. GI Conference Automata Theory & Formal Languages*, pages 134–183, 1975.

P. Henrici.
Applied and Computational Complex Analysis I.
*Wiley*, 1974.

# Bibliography IV


W. Kahan.
Branch Cuts for Complex Elementary Functions.
In A. Iserles and M.J.D. Powell, editors, *Proceedings The State of Art in Numerical Analysis*, pages 165–211, 1987.

National Institute for Standards and Technology.
The NIST Digital Library of Mathematical Functions.
http://dlmf.nist.gov, 2010.