RFC: DLMF Content Dictionaries

Bruce R. Miller
National Institute of Standards and Technology
Gaithersburg, MD, USA
bruce.miller@nist.gov

1 Overview

The Digital Library of Mathematical Functions (DLMF[1]) covers the definitions and properties of a wide variety of special functions with applications in physics and engineering — several hundred, depending on how you count them. Although initially focused as a resource for human readers, the DLMF's long-term goal is to support machine-readable content to enable interoperability between other digital libraries, computer algebra and theorem proving systems.

But, the long history of special functions across and among different communities of interest has led to the potential for confusion and error. When different practitioners speak about apparently the same function, their different histories and conventions may include different assumptions of scalings, arguments (order or definition), branch cuts, assumed values in special cases, and so on. A simple example is the Jacobian elliptic function sn seen as a function either of the modulus k or the parameter $m = k^2$; each form is preferred for certain purposes. While neither party is necessarily *wrong*, failure to account for their differences in communications is guaranteed to lead to error.

It is thus critical for interoperability between systems to establish each system's conventions and assumptions. This is true enough for a human reader transcribing DLMF's results into a computer algebra system, but all the more so when these processes are automated and hidden from view. Ideally these differences can be formalized to the extent that enables automatic conversion of formula across the different views. Indeed, the World Digital Mathematics Library[2] has founded an effort to develop such a Special Function Concordance.

Towards these ends, this note presents a proposed set of (virtual) OpenMath[3] Content Dictionaries (CD) to characterize the choices made in the DLMF.

2 Organization and Naming Conventions

An unexpected challenge was an organization and naming of the CDs and symbols in a fashion appropriate to OpenMath applications. The functions can be grouped according to mathematical or historical features, or applications. The proper names of functions can become quite verbose with strings of significant adjectives before they become sufficiently unique. Consider "Legendre's incomplete elliptic integral of the first kind" and then add modifiers such as "zeros of the derivatives of". Given that some functions are ubiquitous while others are truly esoteric, one would even hope for a Huffman-type encoding.

Yet, the functions have already been grouped into chapters in a way appropriate for the DLMF's purposes. Moreover, each function has a unique LaTeX macro defined for it to simplify the markup and preserve the semantics during conversion to web formats [4]. For example the two functions mentioned above have, macros `\Jacobiellsnk` (encoding "the Jacobian elliptic function sn, of modulus k") and `\incellintFk` (encoding "(Legendre's) incomplete elliptic integral (of the first kind) of modulus k"; See Appendix A for details). While these

In: O. Hasan, J. Davenport, M. Kohlhase (eds.): Proceedings of the 29th OpenMath Workshop, Hagenberg, Austria, 13-Aug-2018, published at http://ceur-ws.org

[1] `https://dlmf.nist.gov/`

[2] `https://www.mathunion.org/ceic/library/world-digital-mathematics-library-wdml`

[3] `https://openmath.org/`

[4] a DLMF Macro set, to be released, is under development

Table 1: Types used in the special function type signature, where $\mathcal{T}$ stands for any arbitrary set or type.

Notation	*Meaning*
$\mathcal{T} \to \mathcal{T}'$	function (mapping) from type $\mathcal{T}$ to $\mathcal{T}'$
$\mathcal{T} \times \mathcal{T}'$	product set of multiple types
$\mathbb{R}$	the set of real numbers (excluding ∞)
$\mathbb{C}$	the set of complex numbers (excluding ∞)
$\mathbb{Z}$	the set of integers
$\mathbb{Z}^+$	the set of integers > 0
$\mathbb{Z}^*$	the set of integers ≥ 0
$\mathbb{Q}$	set of rational numbers
$\mathbb{D}$	the complex numbers in the open unit disc, $\|z\| < 1$
$\{a_0, \ldots\}$	one of a finite set (of symbols)
$\mathcal{T}^n$	n-tuples with elements of type $\mathcal{T}$ (e.g. $\mathbb{R}^2$ for pairs of reals)
$\mathcal{T}^\bullet$	tuples with elements of type $\mathcal{T}$, unspecified length
$\boldsymbol{\mathcal{T}}^n$	vectors of dimension n, with elements of type $\mathcal{T}$
$\boldsymbol{\mathcal{T}}^\bullet$	vectors with elements of type $\mathcal{T}$, unspecified dimension
$\mathcal{T}^{n\times m}$	$n \times m$ matrices with elements of type $\mathcal{T}$
$\mathcal{T}^{\bullet\times\bullet}$	matrices with elements of type $\mathcal{T}$ (unspecified dimension)
$\mathbf{L}$	lattices in the complex plane (in the sense of elliptic functions)

may not roll off the tongue, they are unique, reasonably type-able and blend with TeX's macro conventions. And while this organization and naming may not be optimized for OpenMath purposes, it seems better to reuse the one scheme than to introduce redundant ones.

We have therefore followed DLMF's organization for the primary grouping of functions. The most important functions in a chapter are covered in a base CD, such as `DLMF_BS` (for Bessel functions). In most cases, progressively esoteric functions are grouped into subcategories according to: generalizations (`DLMF_BS_gen`), q-analogs (`DLMF_BS_q`), magnitudes, zeros, matrix argument and so on, as well as some special case such as `DLMF_GH_Appell` for Appell functions.

3 Characterizing the Functions

The more fundamental challenge is to properly characterize the functions. This, of course, is exactly what any proper 'definition' ought to be. But here the point is that the definition be sufficiently complete, explicit and formalized, to enable easily determining the equivalancy of functions from different systems. Ultimately, the goal would be to enable automatic conversion between, for example, the two different 'flavors' of elliptic functions, sn.

At this stage of development, we are providing URLs as the definitions of each function, being pointers into the DLMF where the definition is to be found. This is obviously an informal definition, and may require digging for some details. Definitions may be either explicit or implicit (such as a function defined by a differential equation along with boundary conditions).

Additionally, we have provided a simple type signature for each function to characterize its domain and range (See Table 1). Note that in many cases functions are undefined for isolated values of some arguments, e.g. singularities; these cases are not always reflected in the current signatures. Other properties, such as branch cuts, multivaluedness, have not yet been made explicit.

4 Conclusions and Request for Comments

We have provided here a catalog of the special functions covered by the DLMF — a set of informal, virtual Content Dictionaries. It should serve as a reasonable starting point for establishing a concordance between the sets of functions covered by the several interested parties. This is a continuing process; our CDs will continue to be refined and gradually extended and formalized as needed.

The current status can be found at `https://math.nist.gov/~BMiller/DLMF-CDS/`, where also a JSON encoding of the data may be downloaded for processing.

We welcome suggestions about which features and characteristics function are important to the notion of a concordance, as well as how best to encode and formalize that information. Any other comments about the catalog are also welcome.

Acknowledgements: The author would like to thank Patrick Ion, Howard Cohl and Florian Rabe for constructive comments.

A DLMF Macro Naming conventions

Briefly, the names of the various mathematical function macros are derived from the descriptive 'Proper Name' of the function according to:

$$\mathit{macro} \equiv \backslash\ \mathit{prefix}^{*}\ \mathit{name}\ \mathit{class}^{?}\ \mathit{symbol}^{?} \mathit{suffix}^{*}$$

The *name* is the 'conventional' name or based on the "inventor's" name. The *class* indicates function (generally omitted), integrals, polynomials, and so on. The *symbol* is the latinized form of the notation, upper or lower case as appropriate. The *prefix* modifier includes *all* significant characteristics that may distinguish functions (e.g. 'modified Bessel' vs. simply 'Bessel'). The *suffix* generally indicates limitations or special cases regarding arguments. The abbreviations used for *prefix*, *class* and *suffix* are given in Table 2. For predictability, we avoid abbreviating people's names.

Table 2: Abbreviations used for DLMF macros

class	*Meaning*
	function (omitted by default)
char	characteristic
eigval	eigenvalues
eigvec	eigenvectors
int	integral
mod	modulus
number	number
phase	phase (or phase shift)
poly	polynomial
sum	sum
sym	symbol
trans	transform
wave	wavefunction

suffix	*Meaning*
imag	imaginary argument or order
k	elliptic functions of k, modulus
m	elliptic functions of parameter $m = k^2$
mat	matrix argument
real	of real argument or order
invar	on invariants (Weierstrass)
latt	on lattice (Weierstrass)
q	functions of q, nome
tau	functions of τ

prefix	*Meaning*
a	arc, inverse (circular functions)
A	arc, multi-valued-inverse
aff	affine
ass	associated
aux	auxiliary
big	big
canon	canonical
comp	complete
ccomp	complete complementary
cont	continuous
cusp	cuspoid
deriv	derivative(s) of
diff	differential
diffr	diffraction
dil	dilated
disc	discrete
div	dividing
dual	dual
ell	elliptic
env	envelope of
exp	exponential
gen	general | generalized
hyper	hyperbolic | hypergeometric
inc	incomplete
inv	inverse
irreg	irregular
little	little
log	logarithm(ic)
mod	modified (or modular?)
multivar	multivariate
n	number of
norm	normalized or normalization
para	parabolic
per	periodic
q	q-variant of
rad	radial
reg	regular
rest	restricted
sc	scaled
shift	shifted
sph	spherical | spheroidal
sym	symmetric
umb	umbilic
usph	ultraspherical
z	zeros (of)