ESF: An Automatically Generated Encyclopedia of Special Functions

Ludovic Meunier
Algorithms Project
INRIA Rocquencourt
78153 Le Chesnay Cedex, France
Ludovic.Meunier@inria.fr

Bruno Salvy
Algorithms Project
INRIA Rocquencourt
78153 Le Chesnay Cedex, France
Bruno.Salvy@inria.fr

ABSTRACT

We present our on-going work on the automatic generation of an encyclopedia of special functions on the web, called *The Encyclopedia of Special Functions* (ESF)[1]. All mathematical formulæ in the ESF are computed, typeset and displayed without any human intervention. This is achieved by exploiting a collection of computer algebra algorithms in a systematic way, on top of a specially designed data structure for a class of special functions.

Categories and Subject Descriptors

I.1.2 [**Computing Methodologies**]: Symbolic and Algebraic ManipulationAlgorithms[Algebraic algorithms]

General Terms

Algorithms

Keywords

Special functions, D-finite functions

1. INTRODUCTION

Abramowitz and Stegun's *Handbook of Mathematical Functions* [1] is one of the most cited references not only in mathematics, but also in physics, chemistry and engineering sciences. For about one hundred functions, this book lists properties such as series expansion, asymptotic expansion, graphs, zeroes, tables of numerical values, identities, integral transforms, transformation formulæ, ...

The formulæ in this book and other references on special functions, such as the recent DLMF project[2], were computed, written and proof-read by hand. The progress made in symbolic computation over the last twenty years makes it possible to automate completely the derivation of *many* of these formulæ. This opens the way to developing an automatically generated encyclopedia of special functions that features some level of interactivity.

We concentrate on special functions that are solutions of linear differential equations with polynomial coefficients. These functions are called differentially finite, in short D-finite (or sometimes holonomic), and many of their properties are algorithmically computable [16, 21, 15, 5, 6]. Numerous elementary functions and many so-called special functions fall into the D-finite class. Examples are: exponential, logarithm, all *algebraic functions*, trigonometric and hyperbolic trigonometric functions, inverse trigonometric and inverse hyperbolic trigonometric functions, hypergeometric, confluent hypergeometric and generalized hypergeometric functions (which include Bessel J, H and K functions, Airy Ai and Bi functions, dilogarithms and polylogarithms), Struve, Weber and Anger functions. This class contains about 60% of the functions described in [1].

A special function is represented by a data structure consisting of a linear differential equation and a set of initial conditions (see Section 3 for a more precise description). This forms the input of the engine of our encyclopedia (the Encyclopedia Generator, written in Maple) that produces a document on the function. This document contains mathematical formulæ such as Taylor expansion, asymptotic expansion, closed-form for the coefficients in these expansions when available and graphical representations. The document is output as a web page and becomes part of the ESF. Other output formats are also available (LaTeX, PDF and PostScript). The end of this article shows the document on the Airy Ai function produced with the current state of our generator.

Another outcome of this project is feedback to computer algebra systems themselves: not only formulæ can be generated, but also procedures. Indeed, in a recent work of ours (together with A. Sedoglavic), we wrote a complete package for series and asymptotic expansions in Maple. In this package, all the code concerning the special functions listed above has been generated automatically by the same tools that are used by our Encyclopedia Generator.

As an encyclopedia, the ESF can be seen as complementary to the DLMF and mathematical references linked with computer algebra systems, such as CRC-Maple[3]. These "human-made" encyclopedias provide mathematical insights, methods, formulæ whose derivation cannot be automated

[1] `http://algo.inria.fr/esf`
[2] `http://dlmf.nist.gov`
[3] `http://www.maplesoft.com/products/Crc/crc.shtml`

ISSAC'03, August 3–6, 2003, Philadelphia, Pennsylvania, USA.

(many identities, inequalities, functions that are not D-finite, ...), plus bibliographical and historical references. The ESF focuses on providing *validated* formulæ for the *entire* class of D-finite functions, and not only for a limited set of known and named functions. We emphasize that our encyclopedia is *not* a *static* database of formulæ managed by some search engine; an interesting discussion about this alternative design can be found in *the Hypergeometric Database* [13, §3]. Instead, the ESF is a software that can compute and display formulæ for *any* D-finite function. The web site will let the user define her own function and the corresponding document will be generated *on the fly*.

We feel that our approach shows very attractive features. First, it offers the exciting prospect of presenting ultimately error-free documents. Indeed, mathematical errors and software bugs are shifted from individual formulæ to general algorithms and thus are more likely to be spotted and fixed. Second, the content of a document is easily extended by adding or modifying routines in the generator. In short, the ESF offers access to the algorithmic body of computer algebra on special functions to users who do not know (or wish to learn) the syntax of computer algebra systems. Thus, to some extent, our work fulfils the prediction of [11].

This article is structured as follows. In Section 2, we introduce a typical document of the ESF. We then describe the mathematical foundations of our work in Section 3, where we specify precisely the input of our Encyclopedia Generator. The underlying algorithms are detailed in Section 4. Most of them are well-known, at least in the computer algebra community. In Section 5, we comment on the design issues behind the interface of the ESF.

Notation. This article is a description both of the design and of the current state of the ESF. To distinguish those features described here that are not available at the moment of writing, these are followed by the symbol ($\circ$).

2. WORKED EXAMPLE

In Appendix, we present the current state of the generated entry for the Airy Ai function. The corresponding input for the Encyclopedia Generator is described in Section 3.3.

Section 1 of this generated entry summarizes most of the input of the generator: linear differential equation, notation and initial condition at the origin. The field "Related function" is obtained by looking up the other functions in the database sharing the same differential equation. Section 2 deals with series and asymptotic expansions. In Section 2.1, the Encyclopedia Generator has computed a third order linear recurrence satisfied by the Taylor coefficients of $\mathrm{Ai}(z)$ at the origin and the general formula for these coefficients, together with the first few terms. In Section 3, this recurrence has been used to generate numerical code that is exploited to display graphs of the function in the real and complex domain. The rest of Section 2 deals with the singularity at infinity and Section 4 provides recurrences for expansions bases of orthogonal polynomials.

3. DATA STRUCTURE

R. Askey once remarked that a special function is simply a mathematical function that has been used often enough to deserve a name [2]. Indeed, there is no standard mathematical definition for special functions. Our approach here is based on the observation that many of the classical special functions can be represented in a uniform way using the linear differential equation they satisfy. These functions are called D-finite (see below). In order to permit a complete algorithmic treatment of these functions, some information about initial conditions is also necessary. The purpose of this section is to review the relevant mathematical material and make more precise the data structure used for the input of our Encyclopedia Generator.

3.1 From Special Functions to Linear Algebra

D-finite Series. The definitions and results in this section are classical and can be found for instance in [16]. We use K to denote a computable subfield of $\mathbb{C}$ (the field of complex numbers). Here, computable means that there are algorithms to check for equality and to perform the field operations $(+, -, \times, \div)$. Typical computable fields are $\mathbb{Q}$ (the field of rational numbers) and its finite extensions.

DEFINITION 1. *A formal power series $y \in K[[z]]$ is called* D-finite *if there exist polynomials $p_0, \dots, p_n \in K[z]$ such that y satisfies the linear homogeneous differential equation*

$$p_n(z)\frac{d^n y(z)}{dz^n} + p_{n-1}(z)\frac{d^{n-1}y(z)}{dz^{n-1}} + \cdots + p_0(z) = 0. \quad (1)$$

An equivalent statement of this definition is that the vector space spanned by y and all its derivatives over the rational functions $K(z)$ is finite-dimensional. An important consequence of this finite dimension is the following property.

PROPERTY 1. *The set of D-finite series in $K[[z]]$ forms a subalgebra of $K[[z]]$.*

This means that linear combinations and products of D-finite series are D-finite. Indeed, by repeated differentiation and reduction using the linear differential equation to rewrite high-order derivatives, the product fg of D-finite series f and g lives in a finite-dimensional vector space generated by the series $f^{(i)}g^{(j)}$ and a similar argument applies to sums. These arguments are easily translated into algorithms relying on *linear algebra* that compute a differential equation of type (1) for fg or $f + g$ given similar equations for f and g. Such algorithms are implemented, for instance in the `gfun` package [15][4], together with other closure properties that we recall here.

PROPERTY 2. *Algebraic series are D-finite. The set of D-finite series is closed under differentiation, integration, composition with algebraic series, Hadamard (i.e., termwise) product.*

D-finite Functions

DEFINITION 2. *Let y be an analytic function on a simply connected opened set $\mathcal{D} \subset \mathbb{C}$. The function y is called D-finite if there exists a point $a \in \mathcal{D}$ in whose neighbourhood the Taylor series of y is a D-finite series.*

By Cauchy's theorem and by classical properties of analytic continuation, the function y is actually solution of the same differential equation (1) on any simply connected subset of $\mathcal{D} \setminus \Omega$, where Ω is the (necessarily finite) set of roots of the polynomial p_n. By choosing appropriate points a in this definition, we get the following consequence.

[4] `http://algo.inria.fr/libraries/software.html`

Property 3. *Algebraic functions are D-finite. The set of D-finite functions is closed under sum, product, differentiation, integration, and composition with algebraic functions.*

In this work, linear algebra plays an important role in another way, via the following classical property of solutions of linear differential equations.

Property 4. *The analytic functions that are solutions of the differential equation* (1) *over* $\mathcal{D}$ *form a vector space* $\mathcal{E}$ *over* K*, whose dimension is* n*.*

A D-finite function solution of (1) is thus classically specified by this equation and a set of n "initial conditions". From the computer algebra viewpoint, y is regarded as a zero of the differential operator $\mathcal{L}_z$ of the Weyl algebra $K\langle D, z\rangle$ defined by

$$\mathcal{L}_z = p_n(z)D^n + p_{n-1}(z)D^{n-1} + \cdots + p_0(z),$$

and operations on these operators involve linear algebra over $K(z)$, while initial conditions involve linear algebra over K.

3.2 Singularities and Initial Conditions

Mathematically, initial conditions can be given at any ordinary (i.e., nonsingular) point. From the computational point of view however, there is no general algorithm in this setting to perform analytic continuation other than numerically. Thus in order to produce a database containing information about the function at each of its singularities, we adopt a different notion of initial conditions. Informally, we specify a D-finite function by fixing its behaviour in the neighbourhood of each of its singularities. This is made more precise in the following. Classical references for the results of this section are [8, 20]. Using "generalized initial conditions" as a data-structure is our only contribution here.

Recall that the finite singularities of a solution of (1) can only be found among the roots of the leading polynomial p_n. By shifting the variable (which might involve computations in an algebraic extension of K), we reduce the question to specifying initial conditions at the origin.

Statement of the problem. Consider the Bessel equation

$$z^2\frac{d^2y(z)}{dz^2} + z\frac{dy(z)}{dz} + \left(z^2 - \nu^2\right)y(z) = 0, \qquad (3)$$

with $\nu \in \mathbb{N}^*$; we take $\nu = 1$. The two Bessel functions J_1 and Y_1 form a basis of solutions of (3). The function J_1 is analytic in a neighbourhood of 0 and it is defined by $\mathrm{J}_1(0) = 0$ and $\mathrm{J}_1'(0) = 1/2$, while Y_1 is singular at 0 with a logarithmic behaviour. In order to define Y_1, it is classical to invoke a limit process such as [1, chap. 9]:

$$\mathrm{Y}_1(z) = \lim_{\nu\to 1}\frac{\mathrm{J}_\nu(z)\cos(\nu\pi) - \mathrm{J}_{-\nu}(z)}{\sin(\nu\pi)}.$$

However, such a definition for Y_1 cannot be found automatically from the mere knowledge of (3) specialized at $\nu = 1$.

We choose to express initial conditions in terms of a basis of *formal solutions*, whose definition and properties we now proceed to recall.

Indicial Polynomial. The indicial polynomial $f \in K[\sigma]$ associated to equation (1) at the origin is defined as the leading term in

$$\left(p_n^{-1}(z)z^{n-\sigma}\right)\mathcal{L}_z z^\sigma = f(\sigma) + \cdots \in K[\sigma][[z]].$$

It is such that $\deg(f) \le n$. This polynomial plays an important role in the nature of the possible behaviour of solutions in the neighbourhood of the origin. Its zeroes are the possible valuations of formal series solutions of (1). If the origin is not a singular point of the equation, (i.e., $p_n(0) \neq 0$), then clearly $f(\sigma) = \sigma(\sigma-1)\cdots(\sigma-n+1)$. Otherwise, one distinguishes two types of singularities.

Definition 3. *Assume the origin is a singular point of equation (1) (i.e.,* $p_n(0) = 0$*) and let* $f(\sigma)$ *be the associated indicial polynomial. Then if* $\deg(f) = n$*, 0 is called a* regular singular *point; otherwise* $\deg(f) < n$ *and 0 is called an* irregular singular *point.*

Regular Singular Points

Property 5. *If 0 is a regular singular point of equation (1), this equation admits a basis of formal solutions of the form*

$$y_i(z) = z^{\sigma_i}\sum_{j=0}^{d_i}\log^j(z)\Phi_{i,j}(z), \qquad i = 1,\ldots,n, \qquad (6)$$

where σ_i *are the roots of the indicial polynomial* f*,* $d_i < n$*. Moreover, the formal power series* $\Phi_{i,j}(z) \in K[[z]]$ *are convergent power series.*

The vector space over K of formal solutions generated by the y_i is isomorphic to the vector space $\mathcal{E}$ of Property 4, thanks to the convergence of the $\Phi_{i,j}$. Indeed, after selecting a determination for the logarithms and algebraic exponents in (6), one obtains *analytic functions* in a slit neighbourhood of 0. These n functions are solutions of (1) and thus belong to $\mathcal{E}$. Linear independence of these functions follows from the fact that their Wronskian only vanishes at roots of p_n different from 0, which does not occur in a sufficiently small neighbourhood of 0. This identification of formal series with analytic solutions makes it possible to specify initial conditions as the coordinates of the vector y (the solution under consideration) in a basis of formal solutions. It turns out to be convenient to proceed as follows. If σ_i is a root of the indicial polynomial of multiplicity m_i, we consider the set of monomials

$$\delta_i(z) = \left\{z^{\sigma_i}, z^{\sigma_i}\log(z), \ldots, z^{\sigma_i}\log^{m_i-1}(z)\right\}.$$

Now, $\Delta(z) = \cup_i\delta_i(z)$ is a basis of n linearly independent monomials, which is such that the matrix of coordinates of the y_i's in this basis is non-singular. This leads us to the following.

Definition 4. *When 0 is a regular singular point of (1), the* generalized initial conditions *of a solution are its coordinates in the canonical basis* $\Delta(z)$*.*

It should be noted that this definition coincides with the classical initial conditions when 0 is an ordinary point: in this case $\Delta(z) = \{1, z, \ldots, z^{n-1}\}$. The advantage of this choice in the more general case is that the determination of $\Delta(z)$ only requires the computation of the indicial polynomial.

In the case of the Bessel function Y_1 defined by equation (3), the indicial polynomial at 0 is $(\sigma-1)(\sigma+1)$. The singular point 0 is thus regular, since (3) is of order 2, and the canonical basis $\Delta(z)$ is $\{z^{-1}, z\}$. This means that Y_1 is totally defined by the coefficients of the 2 monomials z^{-1} and z in its series expansion at 0 [5]:

$$[z^{-1}]\,\mathrm{Y}_1(z) = -\frac{2}{\pi},$$
$$[z]\,\mathrm{Y}_1(z) = \frac{2\gamma - 2\ln 2 - 1}{2\pi}.$$

From these 2 coefficients, our Generator produces the classical expansion at 0

$$\mathrm{Y}_1(z) = -\frac{2}{\pi z} + \left(\frac{\ln z}{\pi} + \frac{2\gamma - 2\ln 2 - 1}{2\pi}\right) z - \left(\frac{\ln z}{8\pi} + \frac{4\gamma - 4\ln 2 - 5}{32\pi}\right) z^3 + \mathrm{O}\big(z^5\big).$$

Note in particular that the monomials that are used to define a function at a singular point do not necessarily correspond to the dominant terms in the local expansion (z^{-1} and $\ln z$ in this example).

Irregular Singular Points

PROPERTY 6. *If 0 is an irregular singular point of equation (1), this equation admits a basis of formal solutions of the form*

$$y_i(t) = \mathrm{e}^{P_i(1/t)}\,\Psi_i(t), \tag{8}$$

where $t^{\mu_i} = z$ *for some* $\mu_i \in \mathbb{N}^*$, P_i *is a polynomial with coefficients in the algebraic closure* $\overline{K}$ *of* K *and* Ψ_i *are series of the form* (6). *Moreover, the formal power series* $\Phi_{i,j}$ *contained in the* Ψ_i *are not necessarily convergent.*

The divergence of these formal power series makes the definition of generalized initial conditions more delicate than in the regular singular case. This is achieved thanks to an asymptotic property (the Main Asymptotic Existence Theorem [20]): the formal solutions (8) are asymptotic expansions of actual *analytic solutions* of the equation in any angular sector of amplitude smaller than $\pi/(\mu_i \deg(P_i))$ (see for instance [9]). Thus again, we get an isomorphism between the vector space of analytic solutions $\mathcal{E}$ and the vector space of formal solutions generated by the y_i in (8), provided we specify an angular sector of sufficiently small amplitude.

Since for any polynomial P, $\exp(-P(1/t))$ satisfies a linear (first order) differential equation, closure under product (Property 3) shows that the series $\Psi_i(t)$ is formal solution of a linear differential operator $\mathcal{M}_{t,i}$. Although $\mathcal{M}_{t,i}$ still features an irregular singularity *a priori*, a subspace of the vector space of its formal solutions have the form (6) and can be computed in the same way as in the regular singular case. In particular, to each $\mathcal{M}_{t,i}$ is associated a canonical basis Δ_i defined in the same manner as Δ above. We are thus led to the following definition.

DEFINITION 5. *When 0 is an irregular singular point of equation (1), the* generalized initial conditions *of a solution are its coordinates in the canonical basis*

$$\mathrm{e}^{P_i(1/t)}\,\Delta_i(t), \quad \textit{with} \quad t^{\mu_i} = z,$$

together with a real interval defining an angular sector with vertex 0.

[5] We use the notation $[u(z)]y(z)$ to denote the coefficient of the monomial $u(z)$ in the formal solution $y(z)$.

3.3 Data Structure

We choose to represent a special (D-finite) function by a differential operator it annihilates, together with generalized initial conditions at each of the singularities of this operator. These generalized initial conditions are actually redundant: they are related by so-called *connection matrices* that express the change of coordinates between the set of analytic functions associated with the formal bases at each singular point. However, except for special classes of functions, only numerical approximations of these matrices can be computed.

As an example, the input used for generating the document presented in the appendix on the Airy Ai function consists of: the list of polynomial coefficients of the differential operator; three names `Airy Ai`, `&operatorname(Ai)`, `AI` (the first name is used in plain text, the second one in mathematical formulæ (see §5.2), the last one in cross-references); generalized initial conditions at 0, specifying the constant coefficient and the coefficient of z; generalized initial conditions at ∞ (asymptotic sector $(-\pi/3, \pi/3)$, ramification index 2 and coefficient of $\exp(-2/3/z^{3/2})z^{-1/4}$).

It is important to note that the data we use specifies only one set of coordinates at each singular point and thus actually encodes an implicit choice of *branch cuts* (or rather of homotopy classes of branch cuts in $\mathbb{C} \setminus \Omega$). The actual location of these branch cuts and the way they are handled numerically is an important issue in this area (see in particular [3] and its references for interesting discussions of branch cuts in computer algebra). We come back to this issue in the next section.

4. SURVEY OF ALGORITHMS

Many algorithms are known for various operations on linear differential operators, and implementations are available. Our work here consists in implementing the necessary programming glue and extend these algorithms to our data structure that includes generalized initial conditions. In the following, we denote by y the special function to which the algorithms are applied and give pointers to the algorithms that we use.

– *Taylor expansion.* It is well-known that a series is D-finite if and only if the sequence of its coefficients satisfies a linear recurrence equation. This equation is computed via the algebra morphism defined by $z\,d/dz \mapsto n$, $z \mapsto S_n^{-1}$, where S_n denotes the shift operator. Initial conditions require a bit of care, since in general the differential equation and the recurrence equation do not have the same order. This is implemented in the package `gfun` [15].

– *Closed form for Taylor coefficients.* Petkovšek's algorithm computes a basis of hypergeometric solutions of the recurrence above (i.e., solutions u_n such that u_{n+1}/u_n is a rational function of n) [12]. Then, using our initial conditions reduces the matter of finding a closed form for the Taylor coefficients of y to linear algebra. In MAPLE, this is implemented in the `LREtools` package. Success then leads to a closed form for the coefficients in terms of the Gamma function. Note that failure of this algorithm is a *proof* that the sequence of coefficients is not a linear combination of hypergeometric sequences. The classical method of reduction of order makes it possible to extend this to closed forms of the d'Alembertian type [12].

– *Asymptotic expansion.* At regular singular points, a basis of *formal* solutions of the form (6) is computed by Frobenius' algorithm [8, 17]. The variant of this method that we have implemented expresses all coefficients of the expansion by non-homogeneous linear recurrences [18]. The *asymptotic* expansion is then given by a linear combination of the basis elements whose coefficients are computed from the generalized initial conditions.

  At irregular points, a basis of *formal* solutions is of the form (8). The exponential parts and ramification indices are computed by means of Newton polygons [10, 7, 19]. In MAPLE, this is available in the `DEtools` package. From there, the closure properties implemented in `gfun` give the associated operators $\mathcal{M}_{t,i}$, to which the algorithms for regular singular points apply.

– *Closed form for the asymptotic expansions.* Petkovšek's algorithm applies to the non-homogeneous recurrences above, when they turn out to have hypergeometric right hand-side, which is decided by that same algorithm.

– *Expansion in series of orthogonal polynomial families.* Instead of expanding the function y on the basis $\{x^i, i \in \mathbb{N}\}$, which correspond to Taylor series, it is sometimes useful for approximation purposes to use expansions in other basis, such as those provided by orthogonal polynomials. For instance, Chebyshev series are known to provide good approximations over real intervals. It turns out that the coefficients of D-finite series on the bases provided by the classical families of orthogonal polynomials also satisfy linear recurrence equations that can be computed. In general, this computation can be performed using integral representations and elimination in a multivariate D-finite setting, which is implemented in MGFUN [5]. Again, Petkovšek's algorithm can then be applied to determine closed forms when available.

– *Numerical values and graphics.* The method of binary splitting is an algorithm evaluating D-finite series at complex points with rational coordinates with a bit-complexity that is optimal (up to logarithmic factors) with respect to the required precision. This algorithm is extended in [4, 18] to a low-cost evaluation algorithm (∘) with arbitrary precision (hence, with guaranteed results) for D-finite functions at any point of their Riemann surface. From there, it is possible to produce graphics and tables of numerical values. Also, it is possible to compute numerical approximations of the connection matrices and thus recover the branch cuts that are implicit in our generalized initial conditions (∘).

  Currently, the generated code uses a simple summation of Taylor series in their disk of convergence. Ultimately, it would be desirable that the generated code be able to select an approximation scheme (e.g., Chebyshev series, Taylor series, Taylor series with binary splitting) depending on the required precision and the point where the function has to be evaluated.

– *Integral transforms* (∘). The classical Laplace, Fourier, Hankel and Stieltjes transforms of a D-finite function are also D-finite, when the corresponding integral converges. The multivariate algorithms implemented in MGFUN apply. In many cases, it is however possible to derive the differential equation satisfied by the integral transform by simple univariate manipulations. This is the case for the Laplace transform, for instance.

5. INTERFACE

We aim to make our encyclopedia available to a wide audience of users of special functions: mathematicians, physicists and engineers. It is therefore important that the mathematical content of our encyclopedia is presented and communicated in commonly understood formats. This section explains the choices we made and the solutions we developed in order to achieve this goal.

5.1 Design Issues

The content of our encyclopedia mostly consists of mathematical formulæ. The main issues are the *display* of formulæ to a human reader and the expression of their *semantics*, so that they can be reused (translated into other formats or communicated between systems). Both these questions have received a lot of attention in the past few years.

The MathML standard[6] is concerned with the display of mathematical formulæ. It is more and more widely available in recent versions of browsers, but at the time of this writing, there are still many people whose browser does not support this standard. Since we want to reach the largest possible audience, we are reluctant to force users to switch browser or install plug-ins. We are thus faced with the need for a design accommodating (most) current readers without MathML support while at the same time providing an extension mechanism so that formats such as MathML or others be easy to implement. In the same vein, this extension mechanism should make it easy to produce a format conveying (some) semantics, for instance using the OpenMath standard[7].

Our design is that the algorithmic part of the Encyclopedia Generator is kept isolated from the interface. The Generator produces a "mathematical abstract document" (MAD, see §5.2). This document is encoded in an XML-like format, expressing both content and document structure. From there, it is relatively easy to implement simple translators to various formats (although good quality output is more demanding, see §5.3).

5.2 Mathematical Abstract Document

An abstract document is expressed by a reduced set of markup tags using LaTeX-like keywords (see Table 1). As an example, the document for the Airy Ai function shown in the appendix of this article is encoded as follow

$$\&\texttt{document}\Big(\&\texttt{chapter}\Big(\text{“The AiryAi function”},\\ \qquad \&\texttt{section}\Big(\text{“Introduction”},\ldots,\&\texttt{equation}\Big(\ldots$$

The semantics of mathematical objects is kept in MAPLE. The display, however, is encoded in a format that expresses relative positionning (see Table 2). This format makes it possible to produce pretty-printers supporting automatic line-breaking of large expressions.

5.3 Translators

A translator to a target format translates the MAD format into the target format.

[6]`http://www.w3.org/Math`

[7]`http://openmath.org`

`&document`	physical sectioning
`&chapter`, `&section`	logical sectioning
`&paragraph`	container
`&imath`, `&equation`	math mode
`&ref`, `&link`	cross-references

Table 1: Mathematical Abstract Document

`&sepsequence`	sequence with separator
`&sizedsep`	variable-sized separator
`&delimiter`	pair of delimiters
`&subscript`, `&superscript`	vertical positioning

Table 2: Mathematical formulæ

5.3.1 AMS-LaTeX

LaTeX is a well-known *standard* language for communicating scientific material. Moreover, a whole suite of tools for processing LaTeX documents is available. In order to gain more control over spacing and line-breaking, we adopted the AMS-LaTeX dialect (i.e., LaTeX plus the packages provided by the AMS).

This translator is available as a separate Maple package, called `amslatex`[8]. The most important part of the package lies in its handling of mathematical formulæ. This is achieved by recursive computation of approximate widths and heights of TeX boxes, so that appropriate sizes for parentheses and brackets can be determined, as well as the appropriate locations of line-breaks.

5.3.2 HTML

In previous versions of the ESF, HTML was generated from the LaTeX source (e.g., LaTeX2HTML[9], HeVeA[10]). For more flexibility, we developped our own translator from the MAD format to HTML 4.0. This translator displays mathematical formulæ as GIF or PNG images, which are produced from LaTeX by the script `htmx`[11]. Note that Maple includes a MathML generator that we will use, but we have not experimented with it yet.

5.3.3 Other Formats

The generated page also contains links to the better looking PDF and postscript versions produced at no cost from the AMS-LaTeX source.

5.4 Interactivity (◦)

Obviously, one of the most attractive features of HTML lies in the capability of providing some level of interactivity, which cannot be achieved with hardcopy. At the moment, the web server is not available to the public.

The Encyclopedia of Special Functions is not limited to the known and named special functions. Documents of our encyclopedia are generated for *any* function defined by a linear differential equation and the corresponding set of initial conditions: documents on *user-defined* functions can then be created as well.

[8] `http://algo.inria.fr/meunier/index.html#amslatex`
[9] `http://www.latex2html.org`
[10] `http://pauillac.inria.fr/~maranget/hevea/`
[11] `http://www.math.uic.edu/~fields/htmx/`

Also, the generated documents should be customizable: for instance, the number of explicit terms of the series expansions, the number of digits of numerical approximations have to be user-defined.

5.5 Generated Code

As already mentioned, we also have a generator for series and asymptotic expansions in our multiseries package (not distributed yet). From a data-structure of the type described in Section 3, representing a function F, a set of procedures is generated so that series expansions of functions involving F can be computed efficiently. Because of the nature of irregular singular points and of multiseries expansions [14], this requires adding the "resummed" functions corresponding to the divergent series to the knowledge of the package. Report on this is yet to be written.

In parallel to its generation of formulæ, the Encyclopedia Generator also produces Maple code for numerical evaluation. The code is available from the HTML page. This is a direction in which we want to put much effort in the future, aiming at generating efficient numerical code in arbitrary precision, both in Maple and in C, with the use of a multiprecision package such as MPFR[12].

6. REFERENCES

[1] Milton Abramowitz and Irene A. Stegun, editors. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Dover Publications Inc., New York, 1992. Reprint of the 1972 edition.

[2] George E. Andrews, Richard Askey, and Ranjan Roy. *Special functions*, volume 71. Cambridge University Press, 1999.

[3] R.J. Bradford and J.H. Davenport. Towards better simplification of elementary functions. In Teo Mora, editor, *ISSAC 2002*, pages 16–22. ACM Press, July 2002. Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, July 07–10, 2002, Université de Lille, France.

[4] D. V. Chudnovsky and G. V. Chudnovsky. Approximations and complex multiplication according to Ramanujan. In *Ramanujan revisited*, pages 375–472. Academic Press, Boston, MA, 1988.

[5] Frédéric Chyzak. *Fonctions holonomes en calcul formel*. Phd thesis, École polytechnique, 1998.

[6] Frédéric Chyzak and Bruno Salvy. Non-commutative elimination in Ore algebras proves multivariate holonomic identities. *Journal of Symbolic Computation*, 26(2):187–227, August 1998.

[7] J. Della Dora and E. Tournier. Solutions formelles d'équations différentielles au voisinage de points singuliers réguliers. Technical Report 239, IMAG, Grenoble, 1981.

[8] E. L. Ince. *Ordinary differential equation*. Dover Publications, New York, 1956. Reprint of the 1926 edition.

[9] Michèle Loday-Richaud. Solutions formelles des systèmes différentiels linéaires méromorphes et sommation. *Expositiones Mathematicæ*, 13:116–162, 1995.

[12] `http://www.loria.fr/projets/mpfr`

[10] B. Malgrange. Sur la réduction formelle des équations différentielles linéaires à singularités irrégulières. Grenoble, 1979.

[11] Joel Moses. Toward a general theory of special functions. *Communications of the ACM*, 15:550–554, 1972.

[12] Marko Petkovšek. Hypergeometric solutions of linear recurrences with polynomial coefficients. *Journal of Symbolic Computation*, 14(2-3):243–264, 1992.

[13] Marko Petkovšek, Herbert S. Wilf, and Doron Zeilberger. $A = B$. A. K. Peters Ltd., Wellesley, MA, 1996.

[14] Dan Richardson, Bruno Salvy, John Shackell, and Joris Van der Hoeven. Asymptotic expansions of exp-log functions. In Y. N. Lakshman, editor, *ISSAC'96*, pages 309–313. ACM Press, 1996. Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation. July 24–26, 1996. Zürich, Switzerland.

[15] Bruno Salvy and Paul Zimmermann. Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable. *ACM Transactions on Mathematical Software*, 20(2):163–177, 1994.

[16] Richard P. Stanley. *Enumerative combinatorics*, volume 2. Cambridge University Press, 1999.

[17] Évelyne Tournier. *Solutions formelles d'équations différentielles*. Doctorat d'état, Université scientifique, technologique et médicale de Grenoble, 1987.

[18] Joris Van der Hoeven. Fast evaluation of holonomic functions. *Theoretical Computer Science*, 210(1):199–216, 1999.

[19] Mark van Hoeij. Factorization of differential operators with rational functions coefficients. *Journal of Symbolic Computation*, 24(5):537–561, 1997.

[20] Wolfgang Wasow. *Asymptotic expansions for ordinary differential equations*. Dover Publications Inc., New York, 1987. Reprint of the John Wiley 1976 edition.

[21] Doron Zeilberger. A holonomic systems approach to special functions identities. *Journal of Computational and Applied Mathematics*, 32(3):321–368, 1990.

Appendix: Generated Document for Airy Ai

The AiryAi function

1. Introduction

Let z be a complex variable of $\mathbb{C}\setminus\{\infty\}$. The function AiryAi (noted Ai) is defined by the second order differential equation

$$\frac{\partial^2 y(z)}{\partial z^2} - z\,y(z) = 0. \tag{1}$$

The initial conditions of (1) are given by

$$\mathrm{Ai}(0) = \frac{\frac{1}{3}\sqrt[3]{3}}{\Gamma\left(\frac{2}{3}\right)},\qquad \frac{\partial\,\mathrm{Ai}(z)}{\partial z}(0) = \frac{\left(-\frac{1}{2}\right)\sqrt[6]{3}\,\Gamma\left(\frac{2}{3}\right)}{\pi}. \tag{2}$$

Related function: AiryBi (noted Bi).

2. Series and asymptotic expansions

2.1. Taylor expansion at 0.

2.1.1. *First terms.*

$$\begin{aligned}\mathrm{Ai}(z) = {}& \frac{\frac{1}{3}\sqrt[3]{3}}{\Gamma\left(\frac{2}{3}\right)} - \frac{\frac{1}{2}\sqrt[6]{3}\,\Gamma\left(\frac{2}{3}\right)z}{\pi} + \\ & \frac{\frac{1}{18}\sqrt[3]{3}\,z^3}{\Gamma\left(\frac{2}{3}\right)} - \frac{\frac{1}{24}\sqrt[6]{3}\,\Gamma\left(\frac{2}{3}\right)z^4}{\pi} + \\ & \frac{\frac{1}{540}\sqrt[3]{3}\,z^6}{\Gamma\left(\frac{2}{3}\right)} - \frac{\frac{1}{1008}\sqrt[6]{3}\,\Gamma\left(\frac{2}{3}\right)z^7}{\pi} \\ & + \frac{\frac{1}{38880}\sqrt[3]{3}\,z^9}{\Gamma\left(\frac{2}{3}\right)} + \mathrm{O}\left(z^{10}\right)\end{aligned} \tag{3}$$

2.1.2. *General form.*

$$\mathrm{Ai}(z) = \sum_{n=0}^{\infty} u(n) z^n. \tag{4}$$

The coefficients $u(n)$ satisfy the recurrence

$$-u(n) + \left(n^2 + 5n + 6\right)u(n+3) = 0. \tag{5}$$

Initial conditions of (5) are given by

$$u(0) = \frac{\frac{1}{3}\sqrt[3]{3}}{\Gamma\left(\frac{2}{3}\right)},\quad u(1) = \frac{\left(-\frac{1}{2}\right)\sqrt[6]{3}\,\Gamma\left(\frac{2}{3}\right)}{\pi},\quad u(2) = 0. \tag{6}$$

The recurrence (5) has the closed form solution

$$u(3n) = \frac{\frac{1}{3}\,3^{\left(\frac{1}{3}-2n\right)}}{\Gamma\left(n+\frac{2}{3}\right)\Gamma(n+1)},\quad u(3n+1) = \frac{\left(-\frac{1}{9}\right)3^{\left(\frac{2}{3}-2n\right)}}{\Gamma\left(n+\frac{4}{3}\right)\Gamma(n+1)},\quad u(3n+2) = 0. \tag{7}$$

2.2. Asymptotic expansion at ∞.

2.2.1. *First terms.*

$$\begin{aligned}\mathrm{Ai}(z) \approx {}& \mathrm{e}^{\left(\frac{-\frac{2}{3}}{\xi^3}\right)}\sqrt{\xi}\left(\frac{\frac{1}{2}}{\sqrt{\pi}} - \frac{\frac{5}{96}\xi^3}{\sqrt{\pi}}\right. \\ & + \frac{\frac{385}{9216}\xi^6}{\sqrt{\pi}} - \frac{\frac{85085}{1327104}\xi^9}{\sqrt{\pi}} \\ & \left. + \frac{\frac{37182145}{254803968}\xi^{12}}{\sqrt{\pi}} + \cdots\right),\end{aligned} \tag{8}$$

where $\xi = \sqrt{z}$ and $-\frac{\pi}{3} < \arg(z) < \frac{\pi}{3}$.

2.2.2. *General form.*

$$\mathrm{Ai}(z) \approx \mathrm{e}^{\left(\frac{-\frac{2}{3}}{\xi^3}\right)}\sqrt{\xi}\sum_{n=0}^{\infty} u(n)\xi^n, \tag{9}$$

where $\xi = \sqrt{z}$. The coefficients $u(n)$ satisfy the recurrence

$$\left(5 + 12n + 4n^2\right)u(n) + (16n + 48)\,u(n+3) = 0 \tag{10}$$

The initial conditions of (10) are given by

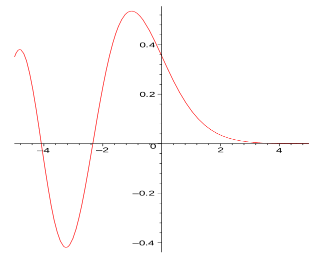
$$u(0) = \frac{\frac{1}{2}}{\sqrt{\pi}},\quad u(1) = 0,\quad u(2) = 0. \tag{11}$$

The recurrence (10) has the closed form solution

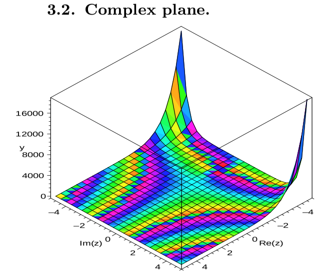
$$u(3n) = \left(\frac{1}{4}(-1)^n\,6^{(2n)}\,\Gamma\left(n+\frac{5}{6}\right)\Gamma\left(n+\frac{1}{6}\right)\right)\Big/\left(\pi^{\frac{3}{2}}\,48^n\,\Gamma(n+1)\right),\quad u(3n+1) = 0,\quad u(3n+2) = 0 \tag{12}$$

3. Graphs

3.1. Real axis.



3.2. Complex plane.



4. Other series expansions

4.1. Chebyshev polynomials.

$$\mathrm{Ai}(z) = \sum_{n=0}^{\infty} u(n) T_n(z), \tag{13}$$

where T_n is the n-th Chebyshev polynomial. The coefficients $u(n)$ satisfy the recurrence

$$\begin{aligned}&\left(-n - n^2\right)u(n) + \\ &\left(-14n - 12 - 4n^2\right)u(n+2) + \\ &8\,u(n+3) + \left(-23n - 12 - 5n^2\right)u(n+4) \\ &- 24\,u(n+5) + (28n + 168)\,u(n+6) + \\ &24\,u(n+7) + \left(97n + 456 + 5n^2\right)u(n+8) \\ &- 8\,u(n+9) + \left(82n + 420 + 4n^2\right)u(n+10) + \\ &\left(n^2 + 23n + 132\right)u(n+12) = 0.\end{aligned} \tag{14}$$

4.2. Hermite polynomials.

$$\mathrm{Ai}(z) = \sum_{n=0}^{\infty} u(n) H_n(z), \tag{15}$$

where H_n is the n-th Hermite polynomial. The coefficients $u(n)$ satisfy the recurrence

$$\begin{aligned}&\left(8n^2 + 24n + 16\right)u(n) + \\ &(-2 - 2n)\,u(n+1) - u(n+3) = 0.\end{aligned} \tag{16}$$