The Dynamic Dictionary of Mathematical Functions (DDMF)*

Alexandre Benoit, Frédéric Chyzak, Alexis Darrasse, Stefan Gerhold, Marc Mezzarobba, and Bruno Salvy

Inria Paris-Rocquencourt, France
`http://ddmf.msr-inria.inria.fr`

Abstract. We describe the main features of the Dynamic Dictionary of Mathematical Functions (version 1.5). It is a website consisting of interactive tables of mathematical formulas on elementary and special functions. The formulas are automatically generated by computer algebra routines. The user can ask for more terms of the expansions, more digits of the numerical values, or proofs of some of the formulas.

1 Motivation

Dictionaries of mathematical functions are commonly used by scientists and engineers. Some of the most famous ones are Abramowitz & Stegun's *Handbook of Mathematical Functions* [1]; the Bateman project *Higher Transcendental Functions* [7]; Gradshtein & Ryzhik's *Table of Integrals, Series, and Products* [8]; and the multivolume *Integrals and Series* by Prudnikov, Brychkov, and Marichev [15]. These dictionaries gather formulas such as differential equations, definite and indefinite integrals, inequalities, recurrence relations, power series, asymptotic expansions, approximations, and sometimes graphs and numerical tables, for a large set of functions. They have been prepared by specialists of these functions and carefully checked and proofread. Their success is attested to by the hundreds of thousands of citations they have received [3].

The first editions of those books were published between 60 and 30 years ago.

Since then, the advent of the World Wide Web has changed the way people now look for information. Aware of this change, the NIST has published a new version of [1] in 2010, called the *NIST Handbook of Mathematical Functions* [13] together with a web site, the NIST Digital Library of Mathematical Functions. This site offers navigation in the formulas, active links, export to various formats, and a search engine.

In parallel, computer algebra systems have grown into huge libraries of mathematical algorithms. While the implementation of mathematical functions in these systems is often basically a coding of formulas from the dictionaries mentioned above, the algorithms have matured to a level where many of those formulas can actually be computed automatically.

* This work was supported by the Microsoft Research-Inria Joint Centre.

The aim of the DDMF is to combine recent algorithms in computer algebra together with web interaction into a dictionary of mathematical functions that is automatically generated, easily navigable with export to various formats, and interactive[1]. Interactivity means that formulas or graphics can be adapted to the user's needs; that arbitrary precision can be given on demand; and that proofs can be displayed if desired.

At this stage, the reader is encouraged to have a look at the DDMF at the following url

`http://ddmf.msr-inria.inria.fr`

A typical page is presented in Figure 1.

The rest of this article presents the ideas underlying our current version (1.5), first from the point of view of the document system and then from the computer algebra viewpoint.

2 Dynamic Mathematics on the Web

The language we use to produce the DDMF is called DynaMoW for *Dynamic Mathematics on the Web*. The main principle on which it is based is captured by the following statement:

> *The document being generated by the symbolic computation engine is an object of the language.*

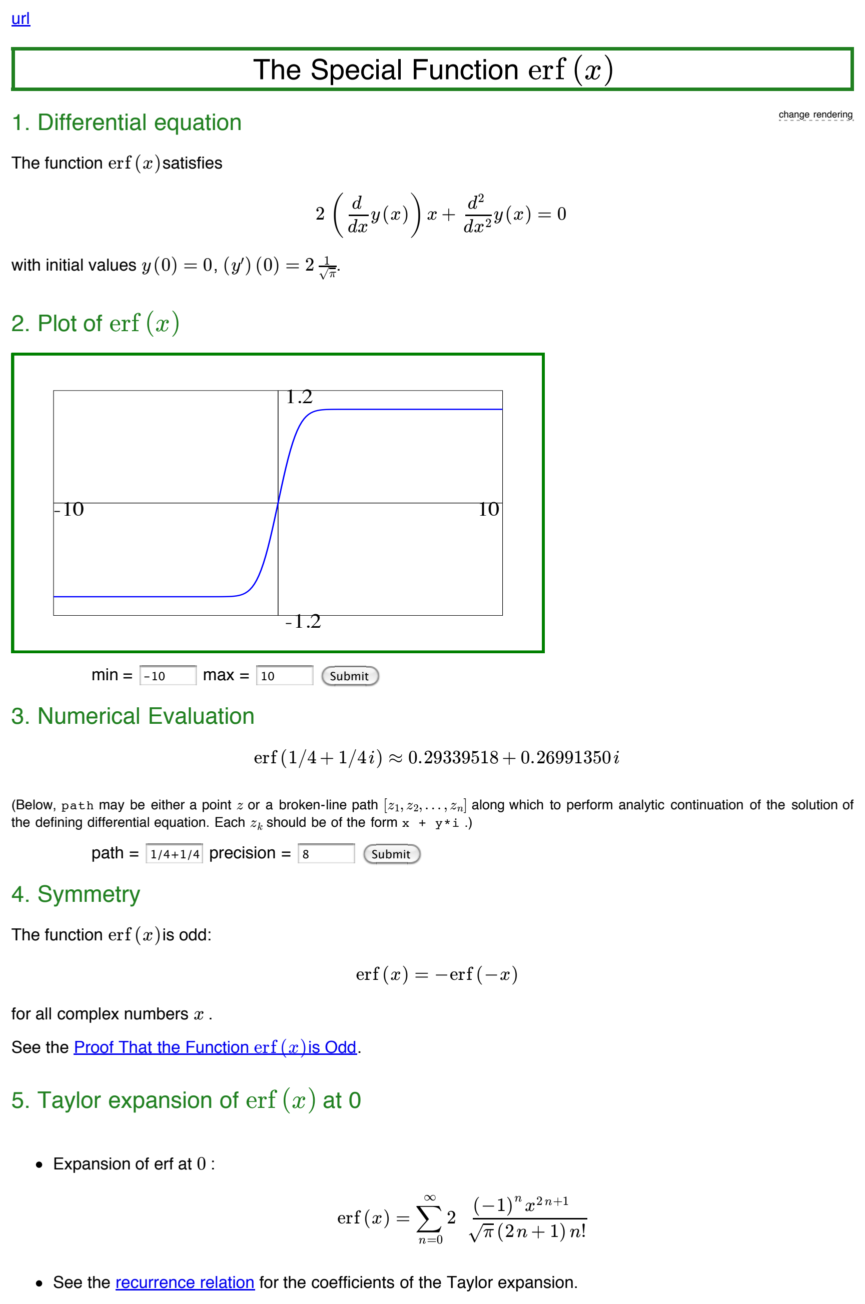
Thus, instead of using a fixed template whose fields are filled in during a computation, the structure of the document itself depends on the results of intermediate computations. For instance, the number of subsections on asymptotic expansions is a result of computing the singularities of the function; the section on symmetries only occurs if the function has been proved even or odd.

DynaMoW is a layer between a symbolic computation engine[2] and a web server. It lets one mix symbolic code together with pieces of documents in a single source code in a natural way. This provides an easy way to showcase computer algebra algorithms to users who do not know the syntax of a computer algebra system: all they need is a web browser; DynaMoW has been designed to be produce pages compatible with the most popular ones.

Moreover, once the document becomes part of the computation, new possibilities arise. For instance, being able to glue together pieces of documents during the computation lets us turn a trace of the computation into a detailed mathematical proof of its result. (See, for instance, the proof of the recurrence formula for the coefficients of the Taylor series of the Airy Ai function.) This answers a frequent request of users of computer algebra systems, who want to be able to understand where the results come from and how they can check or trust them. Traces are not the only type of proof that can be generated. For

[1] An ancestor of these ideas without interactivity was presented in [10].

[2] Currently we use Maple, but DynaMoW is designed so that other systems can be used as well.



url
change rendering
1.2
-10
10
-1.2
min = -10 max = 10 Submit
path = 1/4+1/4 precision = 8 Submit

The Special Function $\operatorname{erf}(x)$

1. Differential equation

The function $\operatorname{erf}(x)$ satisfies

$$2\left(\frac{d}{dx}y(x)\right)x+\frac{d^2}{dx^2}y(x)=0$$

with initial values $y(0)=0,\ (y')(0)=2\frac{1}{\sqrt{\pi}}$.

2. Plot of $\operatorname{erf}(x)$

3. Numerical Evaluation

$$\operatorname{erf}(1/4+1/4i)\approx 0.29339518+0.26991350i$$

(Below, `path` may be either a point z or a broken-line path $[z_1, z_2, \ldots, z_n]$ along which to perform analytic continuation of the solution of the defining differential equation. Each z_k should be of the form `x + y*i` .)

4. Symmetry

The function $\operatorname{erf}(x)$ is odd:

$$\operatorname{erf}(x)=-\operatorname{erf}(-x)$$

for all complex numbers x .

See the Proof That the Function $\operatorname{erf}(x)$ is Odd.

5. Taylor expansion of $\operatorname{erf}(x)$ at 0

- Expansion of erf at 0 :

$$\operatorname{erf}(x)=\sum_{n=0}^{\infty}2\,\frac{(-1)^n x^{2n+1}}{\sqrt{\pi}(2n+1)\,n!}$$

- See the recurrence relation for the coefficients of the Taylor expansion.

Fig. 1. The begining of the page of the DDMF on the error function

instance, we may present a simple proof for the solution of a recurrence once the solution has been found, instead of retracing its computation.

The implementation of the DynaMoW language itself is work in progress and a stable version will be described in due course. We believe that this language will be of interest outside of the DDMF. For instance, we have also used it with success for an encyclopedia of combinatorial structures.

3 Computer Algebra Algorithms

From the computer algebra point of view, what is a good definition of a mathematical function such that all the desired formulas can be computed algorithmically? Our choice is to concentrate on

> *Functions given as solutions of linear differential equations or linear recurrences.*

Our basic data-structure consists of these equations and their initial conditions. In the example of Fig. 1, this is the content of Section 1.

This data-structure has become common in computer algebra, starting with works of Stanley [17], Lipschitz [9], Zeilberger [18] and more recently Chyzak, Salvy *et alii* [4,5,6]. In particular, we rely on the Maple `gfun` package [16] for many of our computations.

Given this data structure, we have used or developed algorithms to compute many relevant properties of mathematical functions. For instance, Section 3 of our example offers numerical approximations of guaranteed quality in good complexity (see [11] for the algorithm). Such approximations can be used to produce graphs as in Section 2 of the example. Section 5 is based on recurrences for the Taylor coefficients that are obtained from the differential equations. When they exist, closed-form hypergeometric solutions of these recurrences can be computed [14]. In all cases, the rest of that Section 5 (beyond the part that is visible in Fig. 1) gives the first terms of these expansions and bounds on tails of power series [12]. The same computations are performed at each singularity including infinity. Further results include Chebyshev expansions [2] and differential equations for the Laplace transform (Sections 7 and 8).

Future Work

Some of our next steps include these tasks: automatic handling of families of functions or functions with parameters, like the Bessel functions, either by letting the user choose values for the parameters, or by performing an automatic discussion according to the possible range of values; automatic generation of good numerical code at fixed precision; more integral transforms; expansions on other bases; information on the zeros of the functions; handling of branch-cuts; and support for user-defined functions.

References

1. Abramowitz, M., Stegun, I.A. (eds.): Handbook of mathematical functions with formulas, graphs, and mathematical tables. Dover Publications Inc., New York (1992), reprint of the 1972 edition. First edition 1964
2. Benoit, A., Salvy, B.: Chebyshev expansions for solutions of linear differential equations. In: May, J. (ed.) Symbolic and Algebraic Computation. pp. 23–30. ACM Press (2009), proceedings of ISSAC'09, Seoul, July 2009
3. Boisvert, R., Lozier, D.W.: A Century of Excellence in Measurements Standards and Technology, chap. Handbook of Mathematical Functions, pp. 135–139. CRC Press (2001)
4. Chyzak, F.: Groebner bases, symbolic summation and symbolic integration. In: Buchberger, B., Winkler, F. (eds.) Groebner Bases and Applications (Proc. of the Conference 33 Years of Gröbner Bases). London Mathematical Society Lecture Notes Series, vol. 251, pp. 32–60. Cambridge University Press (1998), ISBN 0-521-63298-6
5. Chyzak, F.: An extension of Zeilberger's fast algorithm to general holonomic functions. Discrete Mathematics 217(1-3), 115–134 (2000)
6. Chyzak, F., Kauers, M., Salvy, B.: A non-holonomic systems approach to special function identities. In: May, J. (ed.) Symbolic and Algebraic Computation. pp. 111–118. ACM Press (2009), proceedings of ISSAC'09, Seoul, July 2009
7. Erdélyi, A.: Higher Transcendental Functions, vol. 1–3. R. E. Krieger publishing Company, Inc., Malabar, Florida, second edn. (1981). First edition 1953
8. Gradshteyn, I.S., Ryzhik, I.M.: Table of Integrals, Series, and Products. Academic Press (1996). First English edition 1965
9. Lipshitz, L.: *D*-finite power series. Journal of Algebra 122(2), 353–373 (1989)
10. Meunier, L., Salvy, B.: ESF: An automatically generated encyclopedia of special functions. In: Sendra, J.R. (ed.) Symbolic and Algebraic Computation. pp. 199–205. ACM Press (2003), proceedings of ISSAC'03, Philadelphia, August 2003
11. Mezzarobba, M.: NumGfun: a package for numerical and analytic computation with D-finite functions. In: ISSAC'10. ACM Press, `http://arxiv.org/abs/1002.3077`
12. Mezzarobba, M., Salvy, B.: Effective bounds for P-recursive sequences. Journal of Symbolic Computation (To appear)
13. Olver, F.W.J., Lozier, D.W., Boisvert, R.F., Clark, C.W. (eds.): NIST Handbook of Mathematical Functions. Cambridge University Press (2010).
14. Petkovšek, M.: Hypergeometric solutions of linear recurrences with polynomial coefficients. Journal of Symbolic Computation 14(2-3), 243–264 (1992)
15. Prudnikov, A.P., Brychkov, Y.A., Marichev, O.I.: Integrals and Series. Volume 1–6. First edition in Moscow, Nauka, 1981
16. Salvy, B., Zimmermann, P.: 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)
17. Stanley, R.P.: Differentiably finite power series. European Journal of Combinatorics 1(2), 175–188 (1980)
18. Zeilberger, D.: A holonomic systems approach to special functions identities. Journal of Computational and Applied Mathematics 32(3), 321–368 (1990)