The Dynamic Dictionary of Mathematical Functions

Bruno Salvy
Bruno.Salvy@inria.fr

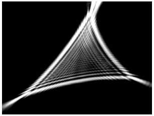
MICROSOFT RESEARCH INRIA JOINT CENTRE

ICMS 2010. September 14, 2010

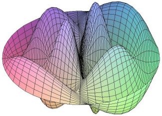
Joint work with: Alexandre Benoit, Frédéric Chyzak, Alexis Darrasse, Stefan Gerhold, Marc Mezzarobba

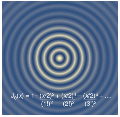
I Motivation

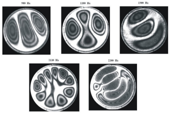
Mathematical Functions



exp, log, sin, cos, tan,
sinh, cosh, tanh, arcsin, arccos, arctan,
arcsinh, arccosh, arctanh,
Bessel $J_\nu, I_\nu, Y_\nu, K_\nu$, Airy Ai and Bi,
hypergeometric, generalized hypergeometric, classical orthogonal polynomials,
Struve $\mathbf{H}_\nu, \mathbf{L}_\nu, \mathbf{M}_\nu, \mathbf{K}_\nu$, Lommel $s_{\mu,\nu}$ and $S_{\mu,\nu}$,
Anger $\mathbf{J}_\nu$, Weber $\mathbf{E}_\nu$, Whittaker $W_{\kappa,\mu}, M_{\kappa,\mu}\dots$







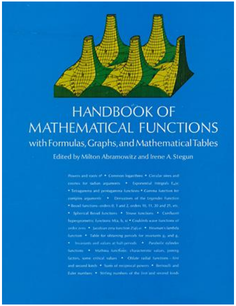
Special functions

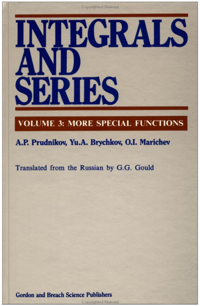
Functions that have been met sufficiently often to deserve a name.

Scientists need help with these functions

Dictionaries

Among the most cited documents in the scientific literature.





Thousands of useful mathematical formulas,
computed, compiled and edited by hand.

Started between 60 and 30 years ago.

Progress in the Past 30 Years

Two important changes in the way we work:

- Symbolic Computation. Several million users.

  Mathematical functions implemented from these dictionaries.

- The Web

  New kinds of interaction with documents.

First Step: the NIST DLMF

Navigation, Exports, Search Engine

Digital Library of Mathematical Functions

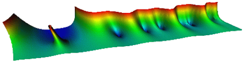
- Index
- Notations

Search

- Need Help?
- How to Cite
- Customize

About the Project

NIST National Institute of Standards and Technology



NIST Digital Library of Mathematical Functions

companion to the NIST Handbook of Mathematical Functions

Project News

2010-05-11 *Handbook published and DLMF goes public*
2010-05-06 *Firefox 3.6 slow on Windows*

- *More news*

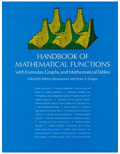
Still computed, compiled and edited by hand.

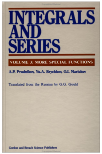
The Dynamic Dictionary of Mathematical Functions

Aim of the project

DDMF = Mathematical Handbooks + Computer Algebra + Web

1. Computer algebra algorithms to generate the formulas;
2. Web-like interaction with the document and the computation.





Building Blocks:

1. Linear differential equations as a data-structure;
2. New language for maths on the web. (compatible with browsers!).

II Demonstration

Demo

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

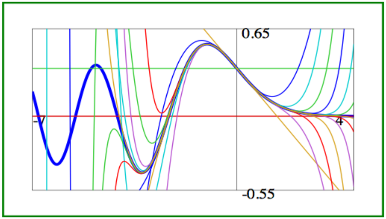
Dynamic Dictionary of Mathematical Functions

Welcome to this interactive site on Mathematical Functions, with properties, truncated expansions, numerical evaluations, plots, and more. The functions currently presented are elementary functions and special functions of a single variable. More functions — special functions with parameters, orthogonal polynomials, sequences — will be added with the project advances.

Select a special function from the list

- Help on selecting and configuring the mathematical rendering
- DDMF developers list
- Motivation of the project
- List of related projects
- Release history

The DDMF project (2008–2010) is hosted and supported by the Microsoft Research – INRIA Joint Centre.



Generated on 2010-06-25 17:03:34 using commit b9253b3..., Fri Jun 25 16:21:12 2010 +0200 (modified).
Powered by DynaMoW.

Contents

rendering link

III DynaMoW: Dynamic Mathematics on the Web

DynaMoW

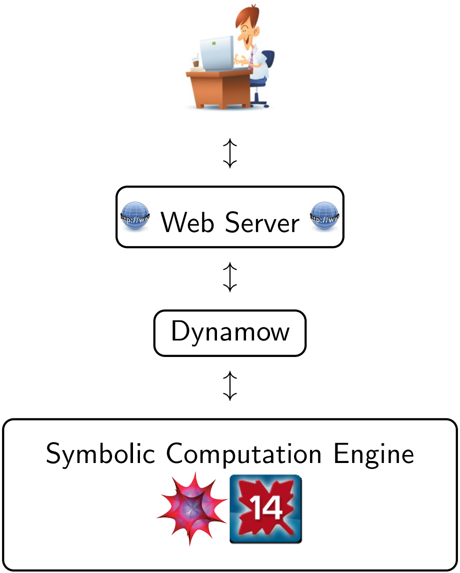
Principle

The document being generated by the computer algebra system is an object of the language.

Consequences:

1. the structure of the document may depend on values that have been computed;
2. intermediate steps can be turned into a mathematical proof in natural language;
3. easy to write demo code.

Architecture



DynaMoW, an OCaml Library

DynaMoW = ocaml + quotations + antiquotations

Symbolic result, converted to LaTeX, put into a paragraph

```
let res = <:symb< symbolic expression >> in
  <:par< some text <:imath< some latex $(symb:res) >>
>>
```

Using ocaml values in symbolic computations

```
let n = 23 and s = "foo" in
  <:symb< f($(int:n), $(str:s)) >>
```

Symbolic objects cast to ocaml types

```
let n = 23 + <:int< symbolic expression >> in ...
if <:bool< symbolic expression >> then ... else ...
<:unit< f := symbolic expression >>
```

[ChyzakDarrasse]

Example

The proof that a function is odd

Sample Implementation with DynaMoW

equation into a homogeneous one.

The homogeneous differential equation and the fact that

$$\frac{d^i}{dx^i} y(-x) = (-1)^i \frac{d^i}{dx^i} y(x), \quad \text{for } i \in \mathbf{N},$$

imply that the function $y_1(x) = -y(-x)$ satisfies the differential equation metadata

$$\left(-1 - x^2\right) \frac{d^2}{dx^2} y_1(x) - 2x \frac{d}{dx} y_1(x) = 0$$

with initial conditions metadata

$$y_1(0) = 0, (y_1')(0) = 1.$$

metadata

The functions $\arctan(x)$ and $-\arctan(-x)$ thus satisfy the same differential equation, and their derivatives at $x = 0$ agree up to order 1 . Since $x = 0$ is an ordinary point of the equation, these functions are analytic and equal in a neighborhood of 0:

$$\arctan(x) = -\arctan(-x).$$

This identity extends to the whole common domain of definition of these functions by uniqueness of the analytic continuation. metadata

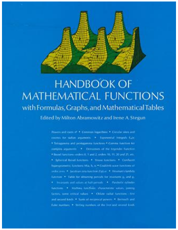
jsMath

```
let ending = Wording.ending_of_seq <:symb< inicondsAlt >> in
let body =
  <:par< The $(str:homStr) differential equation and the fact that
         <:dmath< <:symb< diff(y(-x),[x$i]) = (-1)^i*diff(y(x),[x$i])
              \quad \text{for } i\in\mathbb{N}, >>
         imply that the function <:imath< y_1(x) = $(str:sign)y(-x) >>
         satisfies the differential equation
         <:dsymb< add(op(i+2,altDiffeq) * diff(y[1](x),[x$i]),
                  i=seq(nops(altDiffeq)-2..0,-1)) = -op(1,altDiffe
         with initial condition$(str:ending)
         <:dmath< <:symb< op(inicondsAlt) >> . >> >> :: body in

let sf_name_as_math =
  if sf_name = "" then "y" else <:isymb< $(str:sf_name) >> in

let body =
  let ordinary_point = DC.Text (Glossary.g "ordinary point") in
  <:par< The functions <:imath< $(str:sf_name_as_math)(x) >>
         and <:imath< $(str:sign)$(str:sf_name_as_math)(-x) >>
         thus satisfy the same differential
         equation, and their derivatives at <:imath< x = 0 >> agree up
         order  <:isymb< nops(iniconds)-1 >>.  Since <:imath< x = 0 >>
         is an $(par_entity:ordinary_point) of the equation, these fun
         are analytic and equal in a neighborhood of 0:
         <:dmath< $(str:sf_name_as_math)(x) =
           $(str:sign)$(str:sf_name_as_math)(-x) . >>
         This identity extends to the whole common domain of definitio
         of these functions by uniqueness of the analytic continuation
  >> :: body
in
((title req_params, List.rev body), ret)
```

IV Symbolic Computation

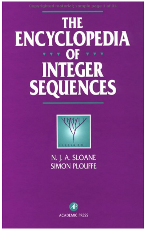
Defining a Mathematical Function by an Equation

- Classical:
  polynomials represent their roots better than radicals.
  Algorithms: Euclidean division and algorithm, Gröbner bases.
- More Recent:
  same for linear differential or recurrence equations.
  Algorithms: non-commutative analogues & gen. func.



About 25% of Sloane's encyclopedia,
60% of Abramowitz & Stegun.

eqn+ini. cond.=data structure



Recent Progress

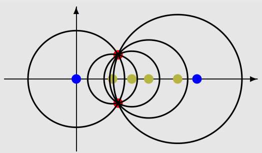
- Automatic computation of bounds, fast evaluation at high precision; [MezzarobbaSalvy2010]
- Chebyshev expansions; [BenoitSalvy2009]
- More formulas accessible to computer algebra [ChyzakKauersSalvy2009].

Guaranteed Numerical Evaluation

1. Inside the disk of convergence

1. Effective majorant-series analysis;
2. Efficient evaluation of truncated series;
3. Time complexity quasi-linear wrt precision.

2. Effective analytic continuation



Path: $0 \to 1.5 \to 2.3 \to 3 \to 4.22 \to 5.$

Computation of Identities by Confinement

$$\int_0^\infty x^{k-1}\zeta(n,\alpha+\beta x)\,dx = \beta^{-k}B(k,n-k)\zeta(n-k,\alpha),$$

$$\int_0^\infty x^{\alpha-1}\,\mathrm{Li}_n(-xy)\,dx = \frac{\pi(-\alpha)^n y^{-\alpha}}{\sin(\alpha\pi)},$$

$$\int_0^\infty x^{k-1}\exp(xy)\Gamma(n,xy)\,dx = \frac{\pi y^{-k}}{\sin((n+k)\pi)}\frac{\Gamma(k)}{\Gamma(1-n)},$$

$$\int_0^\infty \frac{x}{(x^2+y^2)\sin(xz)}\,dx = \frac{\pi}{2\sinh(yz)},$$

not accessible to computer algebra before.

- How can we compute them?
- Why do they exist?

Algorithm:

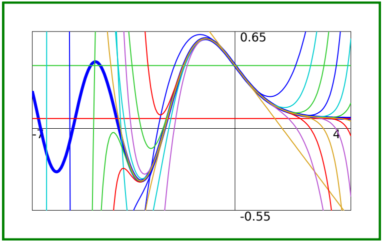
1. Creative telescoping;
2. Confinement in finite-dim vector spaces

Future Work

- More expansions on bases of functions;
- more integral transforms;
- families of functions or functions with parameters;
- automatic generation of numerical code;
- information on the zeros of functions;
- user-defined functions.

Summary: you want to bookmark

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



THE END