Dynamic Dictionary of Mathematical Functions

Bruno Salvy
Bruno.Salvy@inria.fr

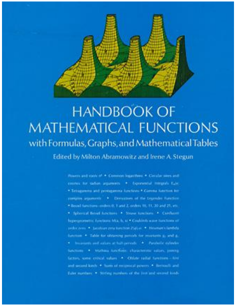
April 29, 2010

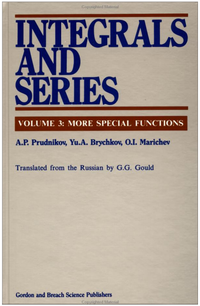
DDMF team: Alin Bostan, Frédéric Chyzak, Bruno Salvy,
Alexandre Benoit & Marc Mezzarobba (PhD students),
Alexis Darrasse (post-doc), Élie de Panafieu (intern)

I Introduction

Context

1. First, there were mathematical handbooks.
   Among the most cited documents in the scientific literature.





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

Context

2. Then, came computer algebra.
   Computation with exact mathematical objects.
   Several million users.

   Mathematical functions implemented from tables.
   30 years of algorithmic progress in effective mathematics.

Context

3. Last, came the Web.

New kinds of interaction with documents.

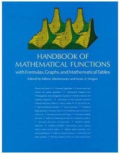
II Dynamic Dictionary of Mathematical Functions

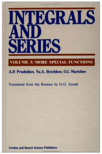
Dynamic Dictionary of Mathematical Functions

Aim of the project

DDMF = Mathematical Handbooks + Computer Algebra + Web

1. Develop and use computer algebra algorithms to generate the formulas;
2. Provide web-like interaction with the document and the computation.



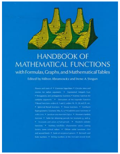


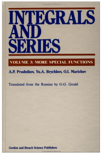
Dynamic Dictionary of Mathematical Functions

Aim of the project

DDMF = Mathematical Handbooks + Computer Algebra + Web

1. Develop and use computer algebra algorithms to generate the formulas;
2. Provide web-like interaction with the document and the computation.





Building Blocks:

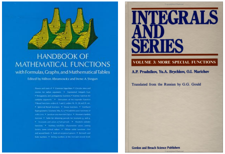
1. Linear differential equations as a data-structure;
2. New language for maths on the web

Dynamic Dictionary of Mathematical Functions

Aim of the project

DDMF = Mathematical Handbooks + Computer Algebra + Web

1. Develop and use computer algebra algorithms to generate the formulas;
2. Provide 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!).

Demo

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

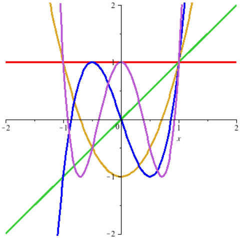
DynaMoW: Example

```
unit singularExpansions ( string s, symbolic def ; ) {
  [s: Local expansions at singularities :s] {
    symbolic eqn := <* op(remove(type, $(def), equation)): *> ;
    integer order := <* nops(indets($(eqn), function)) - 1: *> ;
    <* coeff($(eqn), diff(y(x), x $ $(order))): *> ;
    symbolic sing := <* sort(convert( {solve(%)} minus {0}, list)): *> ;
    integer nb_sing := <* nops($(sing)): *>  ;
    [t: The differential equation above has $(nb_sing) non-zero
        finite singular point(s). :t]
    [t: <ul> :t]
    for integer i in [1..nb_sing] do
      asymptExpansion(s, def, <* $(sing)[$(i)]: *>) ;
    done ;
    asymptExpansion(s, def, <* infinity: *>) ;
    [t: </ul> :t]
  }
}
```

III Algorithmic Ideas: Example of Chebyshev Series

Chebyshev Polynomials and Series

Chebyshev Polynomials: $\boxed{T_n(\cos\theta) := \cos(n\theta)}$



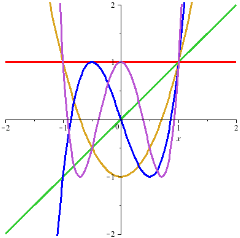
$$2xT_n = T_{n+1} + T_{n-1},$$

$$2(1-x^2)T_n' = n(T_{n-1} - T_{n+1}),$$

$$T_{-n} = T_n.$$

Chebyshev Polynomials and Series

Chebyshev Polynomials: $\boxed{T_n(\cos\theta) := \cos(n\theta)}$



$$\begin{aligned} 2xT_n &= T_{n+1} + T_{n-1}, \\ 2(1-x^2)T_n' &= n(T_{n-1} - T_{n+1}), \\ T_{-n} &= T_n. \end{aligned}$$

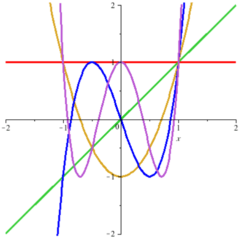
Chebyshev Series: $f = \sum_{n\in\mathbb{Z}} u_n T_n$.

Aim: differential equation for f $\longrightarrow$ recurrence for u_n.

Good approximation properties

Chebyshev Polynomials and Series

Chebyshev Polynomials: $\boxed{T_n(\cos\theta) := \cos(n\theta)}$



$$2xT_n = T_{n+1} + T_{n-1},$$
$$2(1-x^2)T_n' = n(T_{n-1} - T_{n+1}),$$
$$T_{-n} = T_n.$$

Chebyshev Series: $f = \sum_{n\in\mathbb{Z}} u_n T_n.$

Aim: differential equation for f $\rightarrow$ recurrence for u_n.

Basic rules:

$$xf = \sum a_n T_n \quad \Rightarrow \quad a_n = \frac{u_{n-1} + u_{n+1}}{2}$$
$$f' = \sum b_n T_n \quad \Rightarrow \quad b_{n-1} - b_{n+1} = 2nu_n.$$

Examples of Recurrences for Chebyshev Series

Basic rules:

$$xf = \sum a_n T_n \quad \Rightarrow \quad a_n = \frac{u_{n-1} + u_{n+1}}{2}$$

$$f' = \sum b_n T_n \quad \Rightarrow \quad b_{n-1} - b_{n+1} = 2n u_n.$$

Combine:

$$f' + 2xf = \sum c_n T_n \quad \Rightarrow \quad c_{n-1} - c_{n+1} = \mathrm{Rec}_1(u_n).$$

Application: Chebyshev series for $\exp(-x^2)$.

Examples of Recurrences for Chebyshev Series

Basic rules:

$$xf = \sum a_n T_n \quad \Rightarrow \quad a_n = \frac{u_{n-1} + u_{n+1}}{2}$$

$$f' = \sum b_n T_n \quad \Rightarrow \quad b_{n-1} - b_{n+1} = 2n u_n.$$

Combine:

$$f' + 2xf = \sum c_n T_n \quad \Rightarrow \quad c_{n-1} - c_{n+1} = \mathrm{Rec}_1(u_n).$$

Application: Chebyshev series for $\exp(-x^2)$.

$$(f' + 2xf)' = \sum d_n T_n \quad \Rightarrow \quad d_{n-1} - d_{n+1} = 2n c_n,$$

$$\Rightarrow \quad \mathrm{Rec}_2(d_n) = \mathrm{Rec}_3(u_n),$$

$$(f' + 2xf)' - 2f = \sum e_n T_n \quad \Rightarrow \quad \mathrm{Rec}_4(e_n) = \mathrm{Rec}_5(u_n).$$

Application: Chebyshev series for $\mathrm{erf}(x)$.

Combination Rules: General Case

Rule 1: Addition

Given $\mathrm{Rec}_1(a_n) = \mathrm{Rec}_2(u_n)$, $\mathrm{Rec}_3(b_n) = \mathrm{Rec}_4(u_n)$,
find a similar relation between $a_n + b_n$ and u_n.

Rule 2: Composition

Given $\mathrm{Rec}_1(a_n) = \mathrm{Rec}_2(u_n)$, $\mathrm{Rec}_3(b_n) = \mathrm{Rec}_4(a_n)$,
find a similar relation between b_n and u_n.

Combination Rules: General Case

Rule 1: Addition

Given $\mathrm{Rec}_1(a_n) = \mathrm{Rec}_2(u_n)$, $\mathrm{Rec}_3(b_n) = \mathrm{Rec}_4(u_n)$,
find a similar relation between $a_n + b_n$ and u_n.

Rule 2: Composition

Given $\mathrm{Rec}_1(a_n) = \mathrm{Rec}_2(u_n)$, $\mathrm{Rec}_3(b_n) = \mathrm{Rec}_4(a_n)$,
find a similar relation between b_n and u_n.

Algorithm [Benoit-S. 09]: use LCLMs of operators

$$A\,\mathrm{Rec}_1 = B\,\mathrm{Rec}_3 \quad \Rightarrow \quad (A\,\mathrm{Rec}_1)(a_n + b_n) = (A\,\mathrm{Rec}_2 + B\,\mathrm{Rec}_4)(u_n)$$
$$C\,\mathrm{Rec}_1 = D\,\mathrm{Rec}_4 \quad \Rightarrow \quad (D\,\mathrm{Rec}_3)(b_n) = (C\,\mathrm{Rec}_2)(u_n).$$

Unifies and simplifies previous algorithms.

Combination Rules: General Case

Rule 1: Addition

Given $\mathrm{Rec}_1(a_n) = \mathrm{Rec}_2(u_n)$, $\mathrm{Rec}_3(b_n) = \mathrm{Rec}_4(u_n)$,
find a similar relation between $a_n + b_n$ and u_n.

Rule 2: Composition

Given $\mathrm{Rec}_1(a_n) = \mathrm{Rec}_2(u_n)$, $\mathrm{Rec}_3(b_n) = \mathrm{Rec}_4(a_n)$,
find a similar relation between b_n and u_n.

Algorithm [Benoit-S. 09]: use LCLMs of operators

$$A\,\mathrm{Rec}_1 = B\,\mathrm{Rec}_3 \quad \Rightarrow \quad (A\,\mathrm{Rec}_1)(a_n + b_n) = (A\,\mathrm{Rec}_2 + B\,\mathrm{Rec}_4)(u_n)$$
$$C\,\mathrm{Rec}_1 = D\,\mathrm{Rec}_4 \quad \Rightarrow \quad (D\,\mathrm{Rec}_3)(b_n) = (C\,\mathrm{Rec}_2)(u_n).$$

Unifies and simplifies previous algorithms.

Works well beyond Chebyshev series (work in progress)

IV Conclusion

Summary

- A Dictionary of Mathematical Functions that is automatically generated, interactive, and gives proofs;
- A showcase for computer algebra algorithms (including ours);
- Raises new interesting algorithmic questions;
- DynaMoW: a useful language for programming interactive maths on the web (see also http://algo.inria.fr/ecs);
- Next:
  - integral transforms;
  - special functions with parameters (with automatic discussions);
  - generation of numerical code;
  - generation of symbolic code;
  - user-defined functions;
  - more formulas, more pictures,...

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