

Reliable numerical evaluation of special functions. Algorithms and experiments.

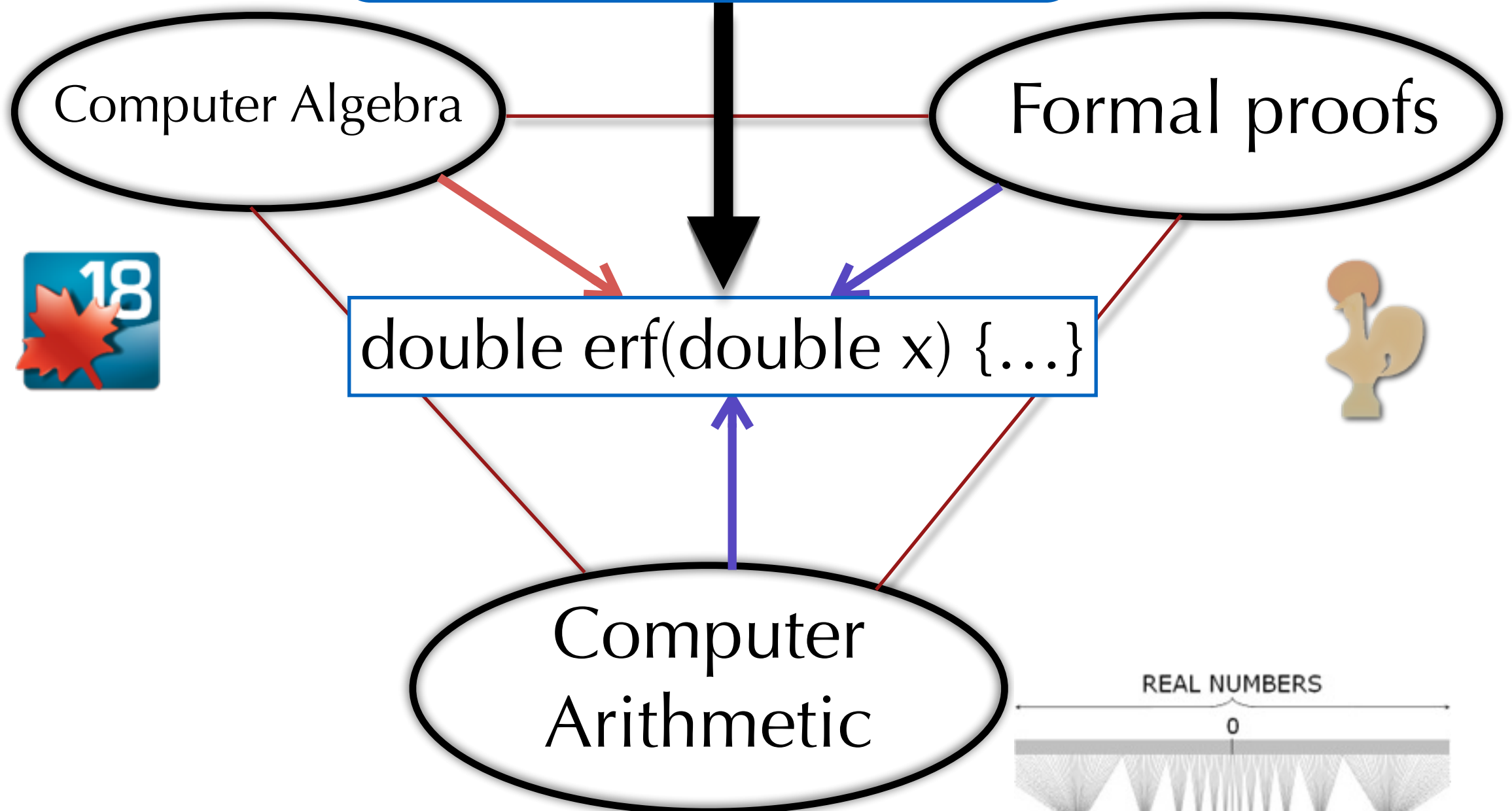
Bruno Salvy
Inria & ENS de Lyon
France



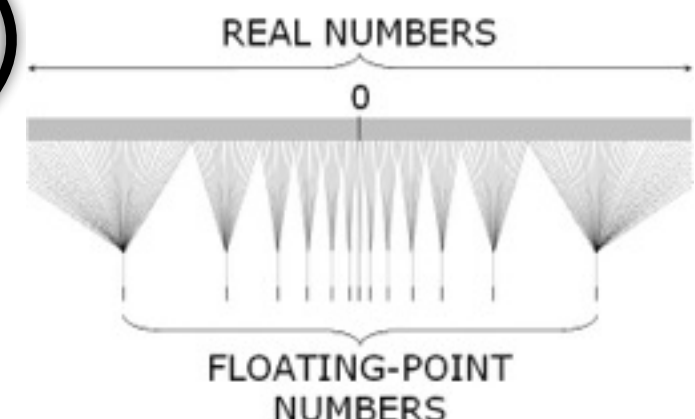
July 21st, 2014

New project: FastRelax (starting this Fall)

$$y'' + 2xy' = 0 + \text{ini. cond.}$$



5 teams, 4 years



I. Equations as a data-structure

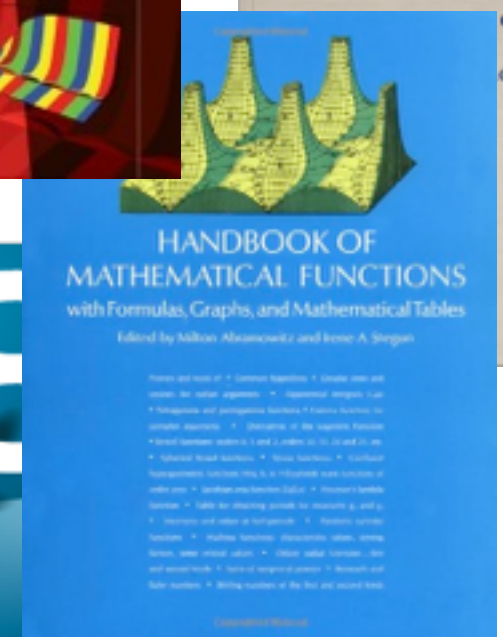
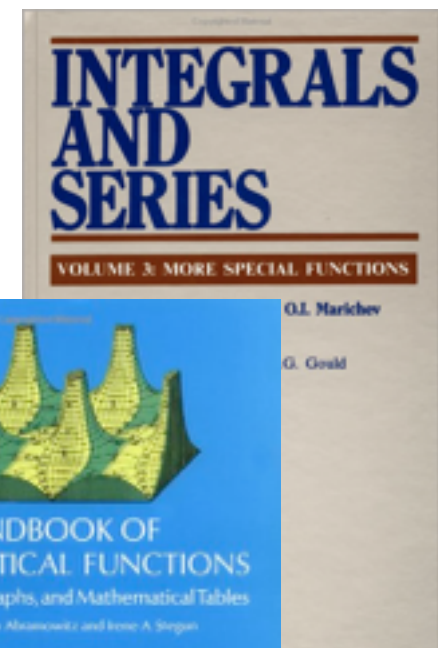
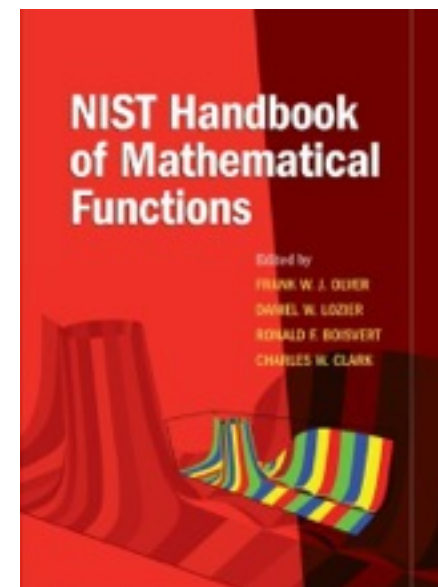
$\text{erf} := (y'' + 2xy' = 0, \text{ini. cond.})$

basis of the [gfun](#) package

Dynamic Dictionary of Mathematical Functions

- User need
- Recent algorithmic progress
- Maths on the web

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



Heavy work by F. Chyzak

Demonstration

Dynamic Dictionary of Mathematical Functions

ddmf.msr-inria.inria.fr/1.9.1/ddmf

Home

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.

This is release 1.9.1 of DDMF
Select a special function from the list

What's new? The main changes in this release 1.9.1, dated May 2013, are:

- Proofs related to Taylor polynomial approximations.

Release [history](#).

More on the project:

- [Help](#) on selecting and configuring the mathematical rendering
- DDMF [developers](#) list
- [Motivation](#) of the project
- [Article](#) on the project at ICMS'2010
- [Source code](#) used to generate these pages
- List of [related projects](#)

Mathematical Functions

- The [Airy function of the first kind](#) $\text{Ai}(x)$
- The [Airy function of the second kind](#) $\text{Bi}(x)$
- The [Anger function](#) $\text{J}_n(x)$
- The [inverse cosine](#) $\arccos(x)$
- The [inverse hyperbolic cosine](#) $\text{arccosh}(x)$
- The [inverse cotangent](#) $\text{arccot}(x)$
- The [inverse hyperbolic cotangent](#) $\text{arccoth}(x)$
- The [inverse cosecant](#) $\text{arccsc}(x)$
- The [inverse hyperbolic cosecant](#) $\text{arccsch}(x)$
- The [inverse secant](#) $\text{arcsec}(x)$
- The [inverse hyperbolic secant](#) $\text{arcsech}(x)$
- The [inverse sine](#) $\arcsin(x)$
- The [inverse hyperbolic sine](#) $\text{arcsinh}(x)$
- The [inverse tangent](#) $\arctan(x)$
- The [inverse hyperbolic tangent](#) $\text{arctanh}(x)$
- The [modified Bessel function of the first kind](#) $I_\nu(x)$
- The [Bessel function of the first kind](#) $J_\nu(x)$
- The [modified Bessel function of the second kind](#) $K_\nu(x)$
- The [Bessel function of the second kind](#) $Y_\nu(x)$
- The [Chebyshev function of the first kind](#) $T_n(x)$
- The [Chebyshev function of the second kind](#) $U_n(x)$
- The [hyperbolic cosine integral](#) $\text{Chi}(x)$
- The [cosine integral](#) $\text{Ci}(x)$
- The [cosine](#) $\cos(x)$
- The [hyperbolic cosine](#) $\cosh(x)$
- The [Coulomb function](#) $F_n(l, x)$
- The [Whittaker's parabolic function](#) $D_a(x)$
- The [parabolic cylinder function](#) $U(a, x)$
- The [parabolic cylinder function](#) $V(a, x)$

II. Numerical evaluation via the Taylor series

From large integers to precise numerical values

Numerical evaluation of solutions of LDEs

Principle:
$$f(x) = \underbrace{\sum_{n=0}^N a_n x^n}_{\text{fast evaluation}} + \underbrace{\sum_{n=N+1}^{\infty} a_n x^n}_{\text{good bounds}}$$

f solution of a LDE with coeffs in $\mathbf{Q}(x)$ (our data-structure!)

1. linear recurrence in N for the first sum (easy);
2. tight bounds on the tail (e.g., [Mezzarobba,S.2010]);
3. no numerical roundoff errors.

The technique used for fast evaluation of constants like

$$\frac{1}{\pi} = \frac{12}{C^{3/2}} \sum_{n=0}^{\infty} \frac{(-1)^n (6n)! (A + nB)}{(3n)! n!^3 C^{3n}} \quad \begin{array}{l} \text{with } A=13591409, \\ B=545140134, \\ C=640320. \end{array}$$

Code available: [NumGfun](#) [Mezzarobba 2010]

Binary Splitting for linear recurrences (70's and 80's)

- $n!$ by divide-and-conquer:

$$n! = \underbrace{n \times \cdots \times \lceil n/2 \rceil}_{\text{size } O(n \log n)} \times \underbrace{\lfloor n/2 \rfloor \times \cdots \times 1}_{\text{size } O(n \log n)}$$

Cost: $O(n \log^3 n \log \log n)$ using FFT

- linear recurrences of order 1 reduce to

$$\textcolor{red}{p!}(n) := (p(n) \times \cdots \times p(\lceil n/2 \rceil)) \times (p(\lfloor n/2 \rfloor) \times \cdots \times p(1))$$

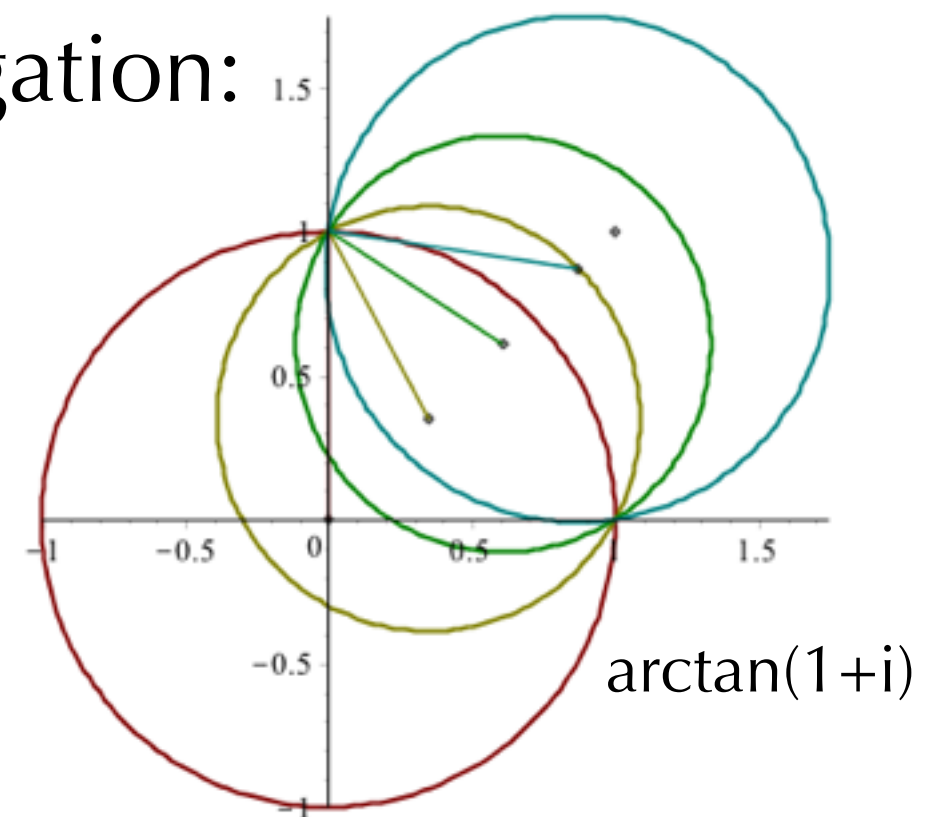
- arbitrary order: same idea, same cost (matrix factorial):

ex $e_n := \sum_{k=0}^n \frac{1}{k!}$ satisfies a 2nd order rec, computed via

$$\begin{pmatrix} e_n \\ e_{n-1} \end{pmatrix} = \frac{1}{n} \underbrace{\begin{pmatrix} n+1 & -1 \\ n & 0 \end{pmatrix}}_{A(n)} \begin{pmatrix} e_{n-1} \\ e_{n-2} \end{pmatrix} = \frac{1}{n!} \textcolor{red}{A!}(n) \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Analytic continuation

Compute $f(x), f'(x), \dots, f^{(d-1)}(x)$ as new initial conditions and handle error propagation:

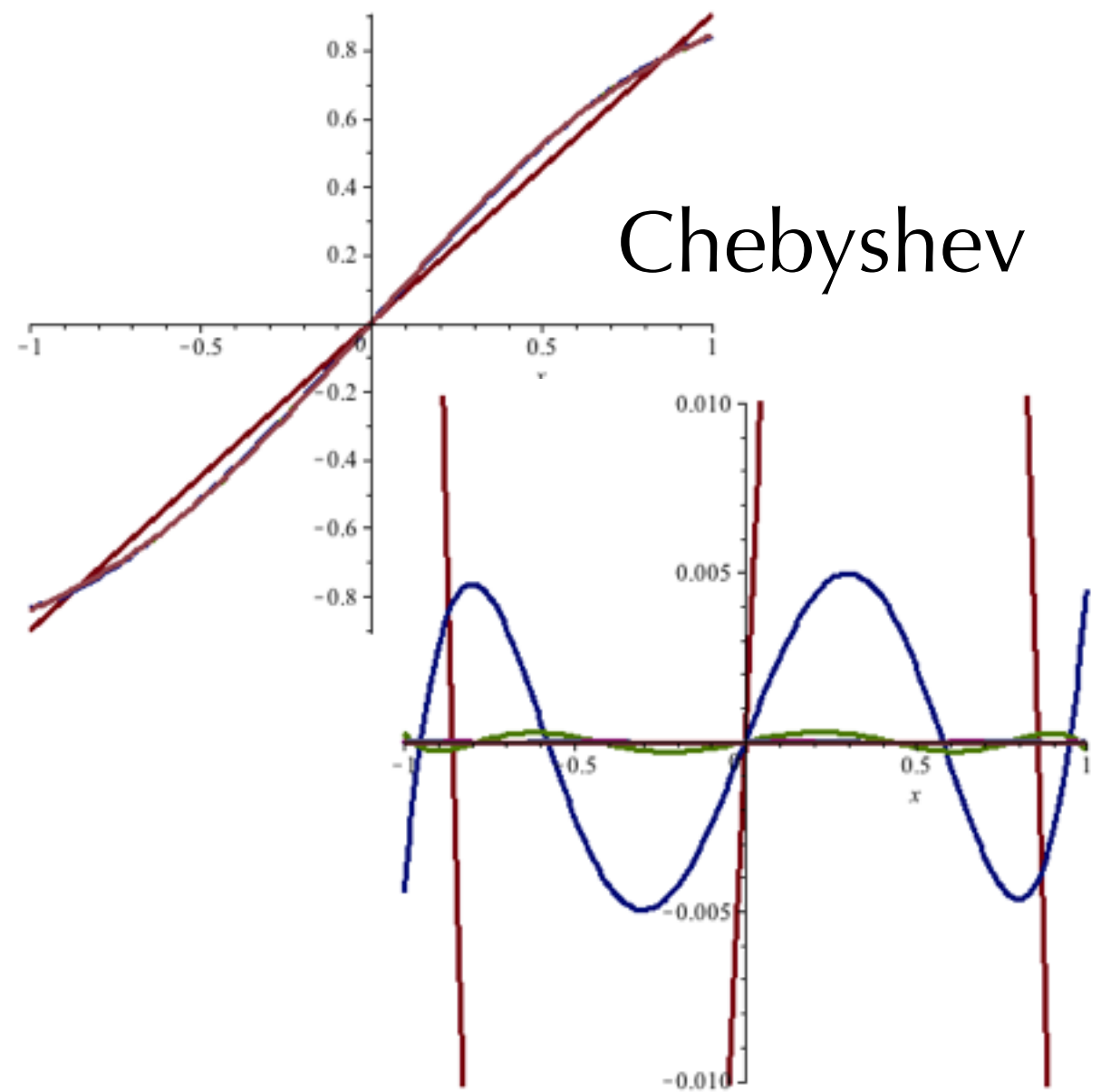
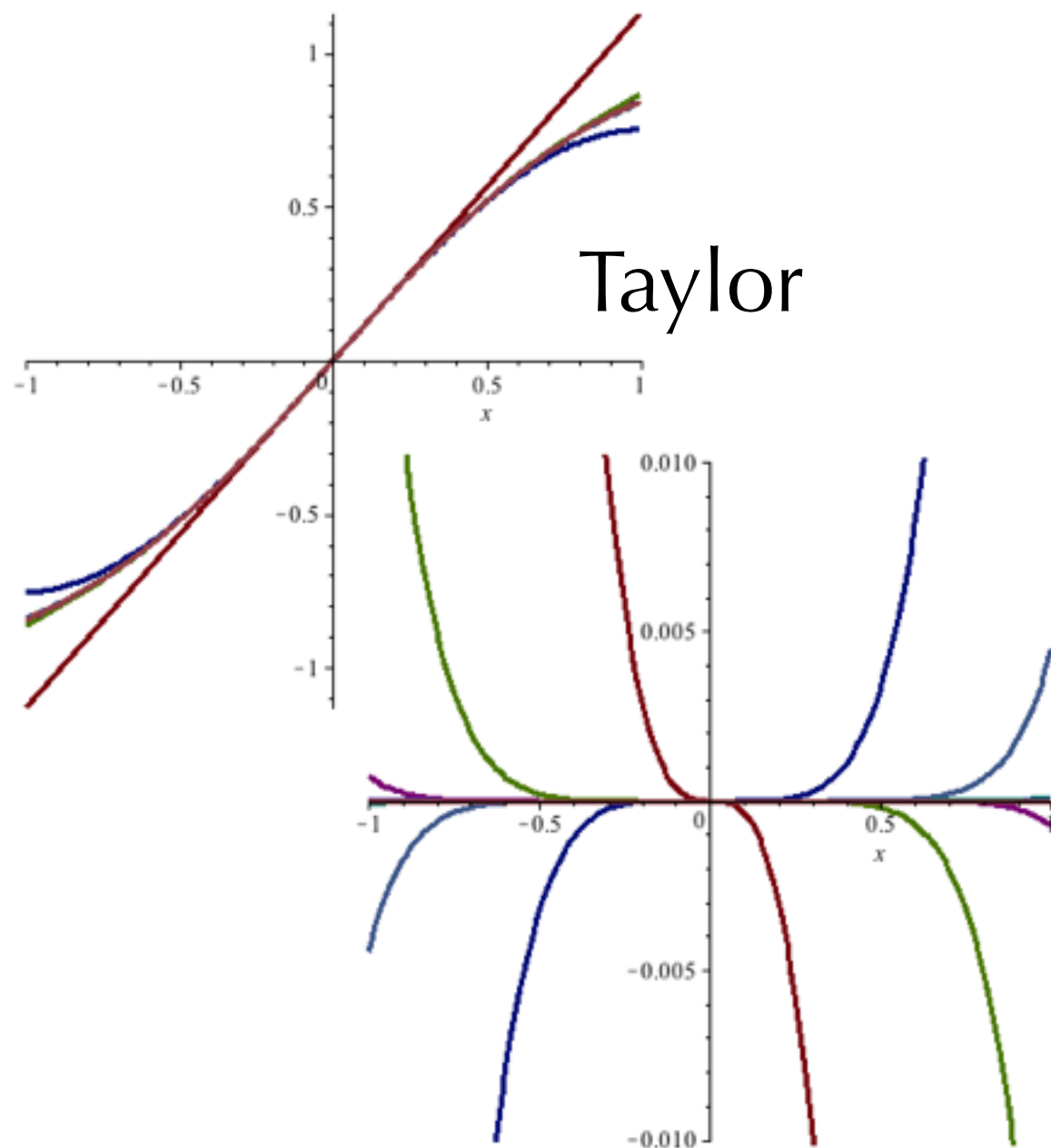


Ex: $\text{erf}(\pi)$ with 15 digits:

$$0 \xrightarrow[200 \text{ terms}]{} 3.1416 \xrightarrow[18 \text{ terms}]{} 3.1415927 \xrightarrow[6 \text{ terms}]{} 3.14159265358979$$

Again: computation on integers. No roundoff errors.

III. Chebyshev expansions



From equations to operators

$$d/dx \leftrightarrow D$$

$$\text{mult by } x \leftrightarrow x$$

$$\text{composition} \leftrightarrow \text{product}$$

$$Dx = xD + 1$$

$$(n \mapsto n+1) \leftrightarrow S$$

$$\text{mult by } n \leftrightarrow n$$

$$\text{composition} \leftrightarrow \text{product}$$

$$Sn = (n+1)S$$

Taylor morphism: $D \mapsto (n+1)S$; $x \mapsto S^{-1}$
produces linear recurrence from LDE

erf: $D^2 + 2xD \mapsto (n+1)S(n+1)S + 2S^{-1}(n+1)S = (n+1)(n+2)S^2 + 2n$

Ore (1933): general framework for these non-commutative polynomials.

Main property: $\deg AB = \deg A + \deg B$.

Consequence 1: (non-commutative) Euclidean division

Consequence 2: (non-commutative) Euclidean algorithm.

Ore fractions ($Q^{-1}P$ with P&Q operators)

Thm. (Ore 1933) Sums and products reduce to that form.

Application: extend Taylor morphism to Chebyshev expansions

Taylor

$$x^{n+1} = x \cdot x^n, (x^n)' = nx^{n-1} \\ \leftrightarrow X := S^{-1}, D := (n+1)S$$

Chebyshev

$$2xT_n(x) = T_{n+1}(x) + T_{n-1}(x), \\ 2(1-x^2)T_n'(x) = -nT_{n+1}(x) + nT_{n-1}(x) \\ \leftrightarrow X := (S + S^{-1})/2, \\ D := (1-X^2)^{-1}n(S - S^{-1})/2 = 2(S^{-1} - S)^{-1}n.$$

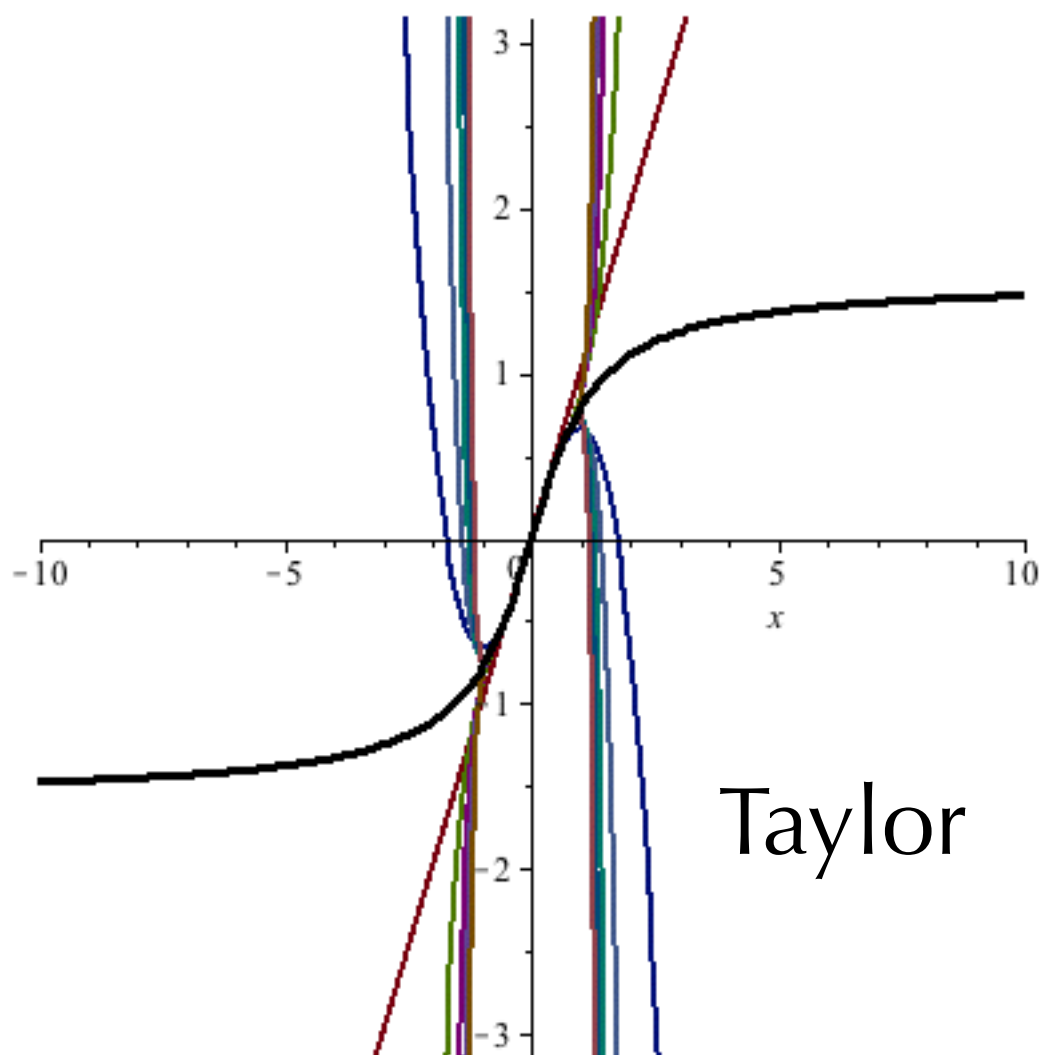
$$\text{erf: } D^2 + 2xD \mapsto (2(S^{-1} - S)^{-1}n)^2 + 2 \frac{S + S^{-1}}{2} 2(S^{-1} - S)^{-1}n \\ = \text{pol}(n, S)^{-1} (2(n+1)(n+4)S^4 - 4(n+2)^3S^2 + 2n(n+3))$$

Prop. [Benoit, S (2009)] If y is a solution of $L(x, d/dx)$, then its Chebyshev coefficients annihilate the **numerator** of $L(X, D)$.

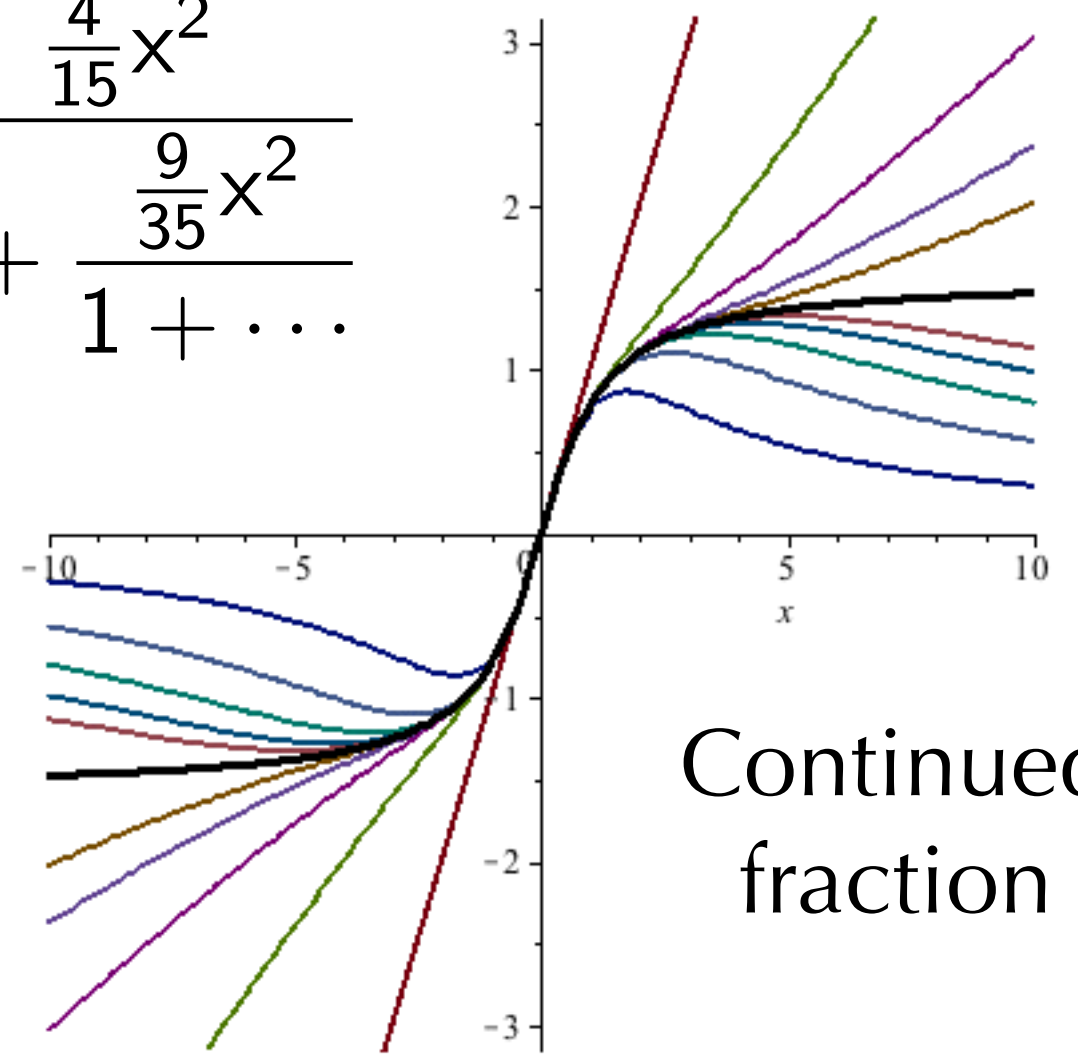
Efficient numerical use: [arXiv:1407.2802](https://arxiv.org/abs/1407.2802) (2 weeks ago).

III. Continued Fractions

$$\arctan x = \frac{x}{1 + \frac{\frac{1}{3}x^2}{1 + \frac{\frac{4}{15}x^2}{1 + \frac{\frac{9}{35}x^2}{1 + \dots}}}}$$



Taylor



Continued
fraction

A guess & prove approach

(Maulat, S. 2014)

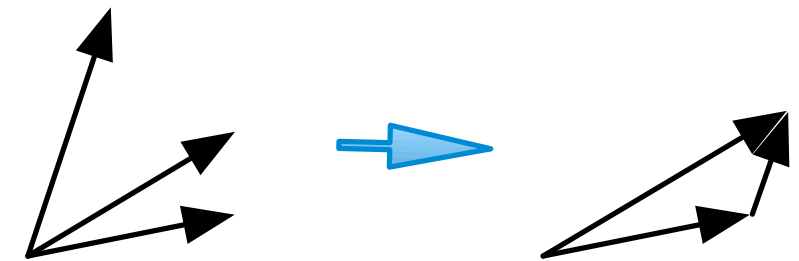
1. Differential equation produces first terms (easy):

$$\arctan x = \frac{x}{1 + \frac{\frac{1}{3}x^2}{1 + \frac{\frac{4}{15}x^2}{1 + \frac{\frac{9}{35}x^2}{1 + \dots}}}}$$

2. **Guess** a formula (easy): $a_n = \frac{n^2}{4n^2 - 1}$
3. **Prove** that the CF with these a_n satisfies the differential equation.

No human intervention needed.

Proof technique



```
> series(sin(x)^2+cos(x)^2-1,x,4);
```

f satisfies a LDE



$f, f', f'', \dots$ live in a
finite-dim. vector space

$O(x^4)$

Why is this a proof?

1. $\sin$ and $\cos$ satisfy a 2nd order LDE: $y''+y=0$;
2. their squares and their sum satisfy a 3rd order LDE;
3. the constant -1 satisfies $y'=0$;
4. thus $\sin^2+\cos^2-1$ satisfies a LDE of order at most 4;
5. Cauchy's theorem concludes.

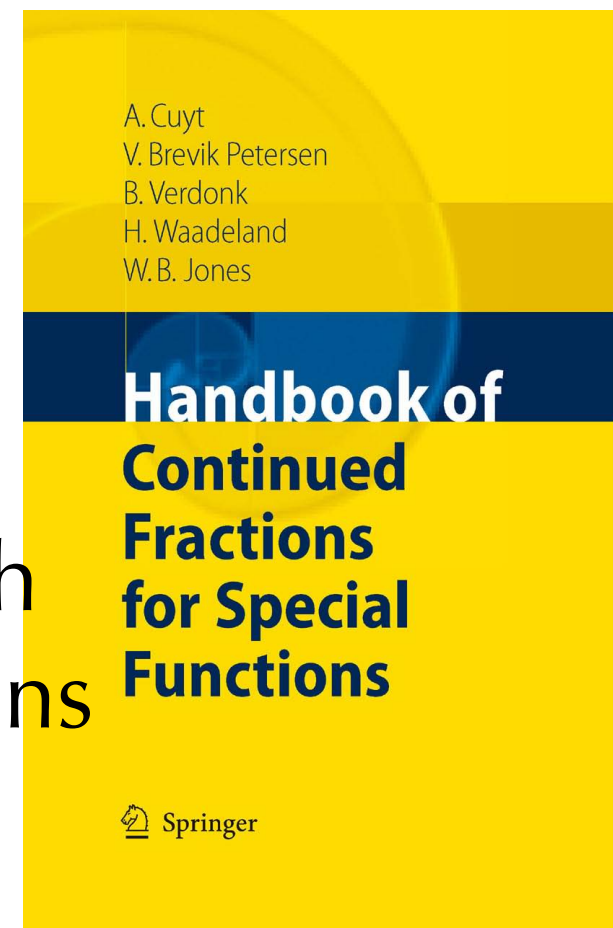
Proofs of non-linear identities by linear algebra!

Automatic Proof of the guessed CF

$$\arctan x \stackrel{?}{=} \frac{x}{1 + \frac{\dots}{1 + \frac{\frac{n^2}{4n^2-1}x^2}{1 + \dots}}}$$

- Aim: RHS satisfies $(x^2+1)y'-1=0$;
- Convergents P_n/Q_n where P_n and Q_n satisfy a LRE (and $Q_n(0) \neq 0$);
- Define $H_n := (Q_n)^2((x^2+1)(P_n/Q_n)' - 1)$;
- H_n is a polynomial in P_n, Q_n and their derivatives;
- **therefore**, it satisfies a LRE that can be computed;
- from it, $H_n = O(x^n)$ visible, ie $\lim P_n/Q_n$ soln;
- conclude $P_n/Q_n \rightarrow \arctan$ (check initial cond.).

More generally: this guess-and-proof approach applies to CF for solutions of (q-)Ricatti equations
→ all explicit C-fractions in Cuyt et alii.



Conclusion

- Linear differential equations and recurrences are a great data-structure;
- Numerous algorithms have been developed in computer algebra;
- Efficient code is available;
- More is true (creative telescoping, diagonals,...);
- More to come in DDME, including formal proofs.

The End