Introduction to D-finite series with a view towards reliable numerical evaluation

Bruno Salvy
Inria & ENS de Lyon

I. Equations as a data-structure

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

The objects of study

Def. A power series is called D-finite when it is the solution of a linear differential equation with polynomial coefficients.

Exs: sin, cos, exp, log, arcsin, arccos, arctan, arcsinh, hypergeometric series, Bessel functions,…

Def. A sequence is P-recursive when it is the solution of a linear recurrence with polynomial coefficients

Prop. $f = \sum_{n=0}^{\infty} f_n z^n$ D-finite $\iff$ f_n P-recursive.

Example

Coefficient of X^{20000} in $P(X)=(1+X)^{20000}(1+X+X^2)^{10000}$?

Example

Coefficient of X^{20000} in $P(X)=(1+X)^{20000}(1+X+X^2)^{10000}$?

Linear differential equation of order 1
→ linear recurrence of order 2
→ unroll (cleverly).

Example

Coefficient of X^{20000} in $P(X)=(1+X)^{20000}(1+X+X\char`^2)^{10000}$?

Linear differential equation of order 1

→ linear recurrence of order 2

→ unroll (cleverly).

```
> P := (1+x)^(2*N)*(1+x+x^2)^N:
> deq := gfun:-holexprtodiffeq(P,y(x)):
> rec := gfun:-diffeqtorec(%,y(x),u(k)):
> p := gfun:-rectoproc(subs(N=10000,rec),u(k)):
> p(20000);
```

Example

Coefficient of X^{20000} in $P(X)=(1+X)^{20000}(1+X+X^2)^{10000}$?

Linear differential equation of order 1
→ linear recurrence of order 2
→ unroll (cleverly).

```
> P := (1+x)^(2*N)*(1+x+x^2)^N:
> deq := gfun:-holexprtodiffeq(P,y(x)):
> rec := gfun:-diffeqtorec(%,y(x),u(k)):
> p := gfun:-rectoproc(subs(N=10000,rec),u(k)):
> p(20000);
```

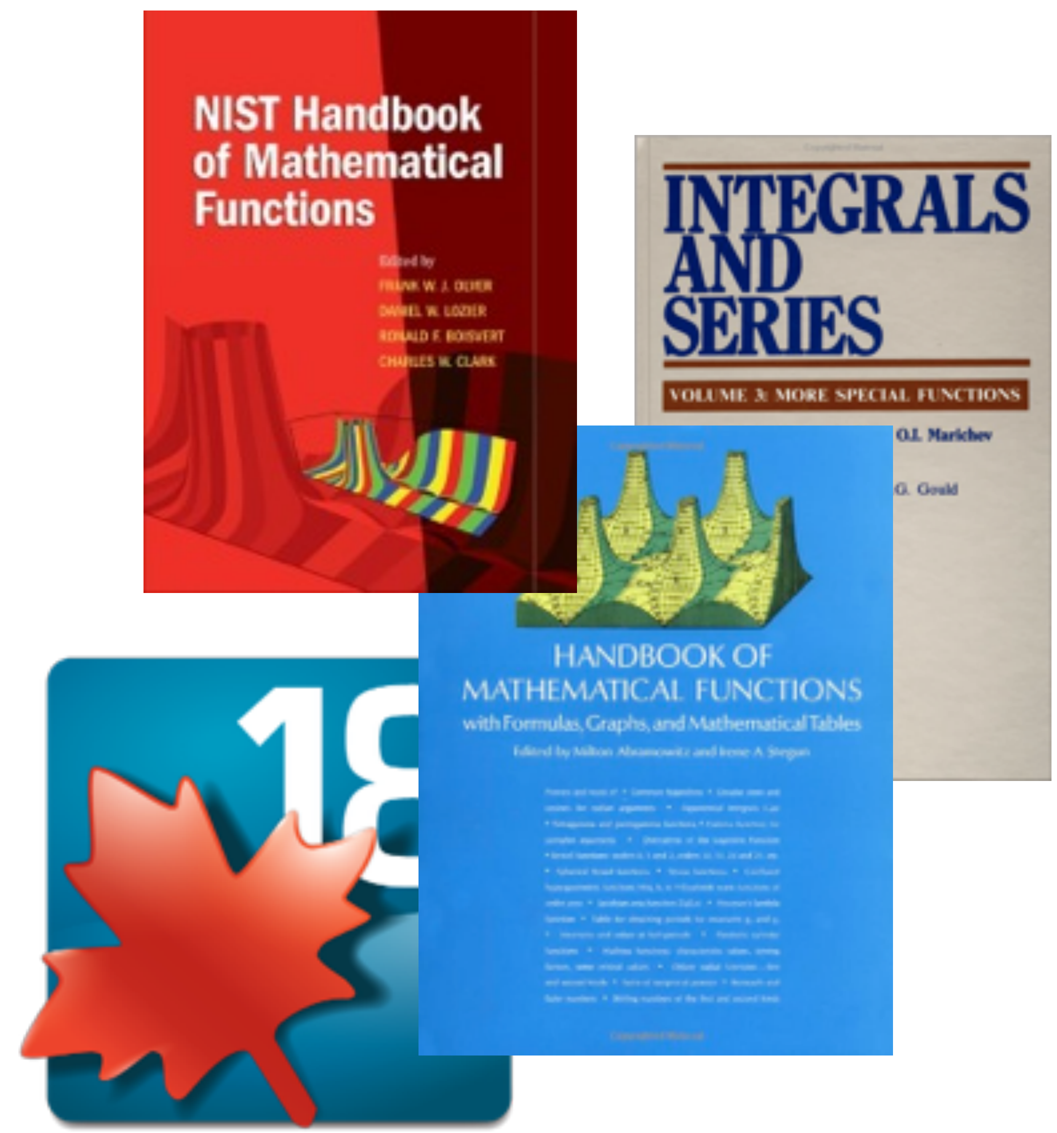
23982[...10590 digits...]33952

Total time: 0.5 sec

Dynamic Dictionary of Mathematical Functions

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

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



Heavy work by F. Chyzak

II. Numerical evaluation at large precision

From large integers to precise numerical values

Fast multiplication

Fast Fourier Transform (Gauss, Cooley-Tuckey, Schönhage-Strassen).
Two integers of n digits can be multiplied with
$O(n \log(n) \log\log(n))$ bit operations.

Fast multiplication

Fast Fourier Transform (Gauss, Cooley-Tuckey, Schönhage-Strassen). Two integers of n digits can be multiplied with O(n log(n) loglog(n)) bit operations.

Applications (in the 70′s and the 80′s):

Fast multiplication

Fast Fourier Transform (Gauss, Cooley-Tuckey, Schönhage-Strassen).
Two integers of n digits can be multiplied with
O(n log(n) loglog(n)) bit operations.

Applications (in the 70's and the 80's):

- inverses, square-roots,… by Newton iteration, same cost;

Fast multiplication

Fast Fourier Transform (Gauss, Cooley-Tuckey, Schönhage-Strassen).
Two integers of n digits can be multiplied with
O(n log(n) loglog(n)) bit operations.

Applications (in the 70's and the 80's):

- inverses, square-roots,… by Newton iteration, same cost;
- n! by divide-and-conquer:

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

Cost: O(n $\log^3$n loglog n)

Fast multiplication

Fast Fourier Transform (Gauss, Cooley-Tuckey, Schönhage-Strassen).
Two integers of n digits can be multiplied with
$O(n \log(n) \log\log(n))$ bit operations.

Applications (in the 70's and the 80's):

- inverses, square-roots,… by Newton iteration, same cost;
- 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: $O(n \log^3 n \log\log n)$
- any linear recurrence of order 1 (coeffs in $\mathbf{Q}(n)$);

Fast multiplication

Fast Fourier Transform (Gauss, Cooley-Tuckey, Schönhage-Strassen). Two integers of n digits can be multiplied with O(n log(n) loglog(n)) bit operations.

Applications (in the 70's and the 80's):

- inverses, square-roots,… by Newton iteration, same cost;
- n! by divide-and-conquer:

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

  Cost: $O(n \log^3 n \operatorname{loglog} n)$
- any linear recurrence of order 1 (coeffs in $\mathbf{Q}(n)$);
- arbitrary order: same idea, same cost (matrix factorial).

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!)

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 (technical);
3. no numerical roundoff errors.

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 (technical);
3. no numerical roundoff errors.

The technique used for fast evaluation of constants like

$$\frac{1}{\pi} = \frac{12}{C^{3/2}} \sum_{n=0}^{\infty} \frac{(-1)^n (6n)!(A+nB)}{(3n)!n!^3 C^{3n}}$$

with A=13591409, B=545140134, C=640320.

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 (technical);
3. no numerical roundoff errors.

The technique used for fast evaluation of constants like

$$\frac{1}{\pi} = \frac{12}{C^{3/2}} \sum_{n=0}^{\infty} \frac{(-1)^n (6n)!(A+nB)}{(3n)!n!^3 C^{3n}}$$

with A=13591409, B=545140134, C=640320.

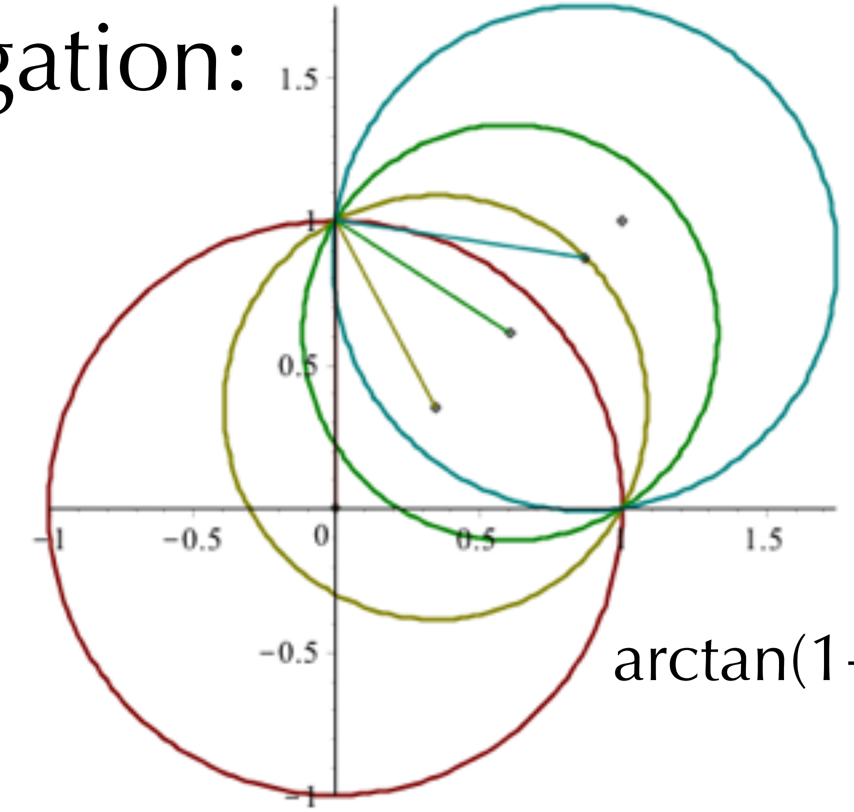
Code available: NumGfun [Mezzarobba 2010]

$$f(x), f'(x), \ldots, f^{(d-1)}(x)$$

arctan(1+i)

Analytic continuation

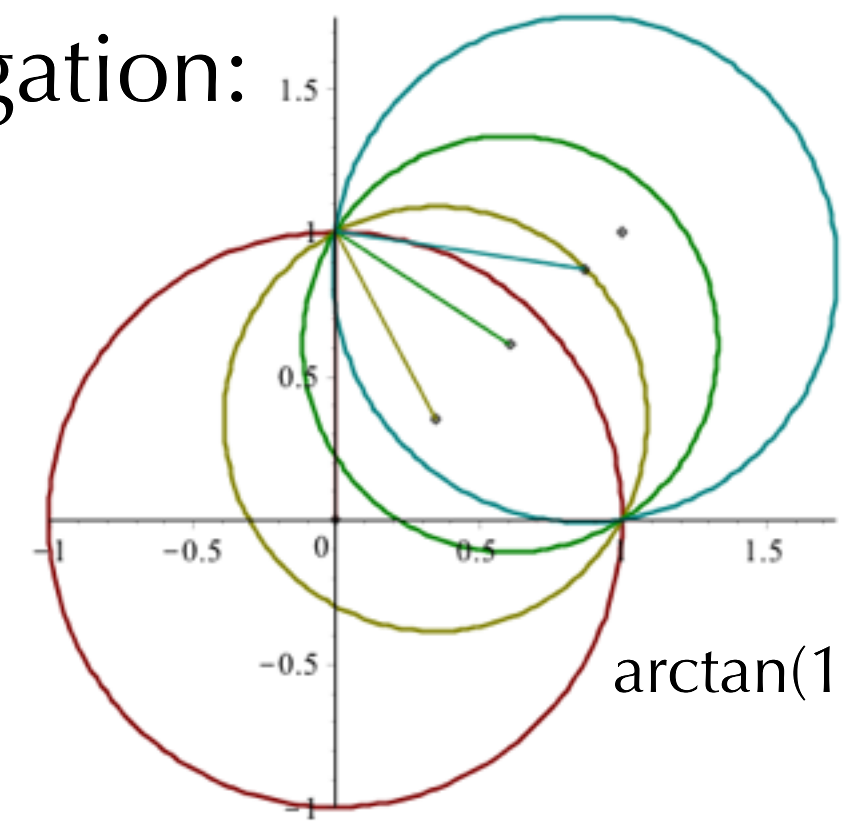
Compute $f(x), f'(x), \ldots, f^{(d-1)}(x)$ as new initial conditions and handle error propagation:



arctan(1+i)

Analytic continuation

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



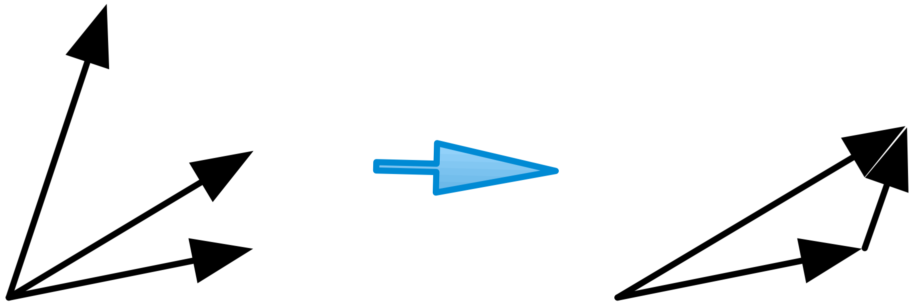
arctan(1+i)

Ex: erf(π) with 15 digits:

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

Again: computation on integers. No roundoff errors.

III. Closure properties



k+1 vectors in dimension k → an identity

Confinement

LDE $\Longleftrightarrow$ the function and all its derivatives are confined in a finite dimensional vector space

$\Rightarrow$ the sum and product of solutions of LDEs satisfy LDEs

$\Rightarrow$ same property for P-recursive sequences

Proofs of identities

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

$$O(x^4)$$

Why is this a proof?

Proofs of identities

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

$$O(x^4)$$

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.

Application to continued fraction expansions

$$e^z = 1 + \cfrac{a_1 z}{1 + \cfrac{a_2 z}{1 + \cfrac{a_3 z}{\ddots}}}$$

$$a_{2k} = \frac{1}{2(2k+1)}, \quad a_{2k+1} = \frac{-1}{2(2k-1)}.$$

(easily found by guessing)

Automatic proof of the continued fraction (Maulat-S. 2014):

Application to continued fraction expansions

$$e^z = 1 + \cfrac{a_1 z}{1 + \cfrac{a_2 z}{1 + \cfrac{a_3 z}{\ddots}}} \qquad a_{2k} = \frac{1}{2(2k+1)},\ a_{2k+1} = \frac{-1}{2(2k-1)}.$$

(easily found by guessing)

Automatic proof of the continued fraction (Maulat-S. 2014):

- Convergents P_k/Q_k where P_k and Q_k satisfy a LRE (and $Q_k(0)\neq 0$);
- $H_k := (Q_k)^2((P_k/Q_k)' - (P_k/Q_k))$ satisfies a LRE from which val $H_k \geq k$ visible.

Application to continued fraction expansions

$$e^z = 1 + \cfrac{a_1 z}{1 + \cfrac{a_2 z}{1 + \cfrac{a_3 z}{\ddots}}} \qquad a_{2k} = \frac{1}{2(2k+1)}, \quad a_{2k+1} = \frac{-1}{2(2k-1)}.$$

(easily found by guessing)

Automatic proof of the continued fraction (Maulat-S. 2014):

- Convergents P_k/Q_k where P_k and Q_k satisfy a LRE (and $Q_k(0)\neq 0$);
- $H_k:=(Q_k)^2((P_k/Q_k)'-(P_k/Q_k))$ satisfies a LRE from which val $H_k \geq k$ visible.

More generally: guess-and-proof approach to CF for solutions of Ricatti equations

Mehler's identity for Hermite polynomials

$$\sum_{n=0}^{\infty} H_n(x)H_n(y)\frac{u^n}{n!} = \frac{\exp\left(\frac{4u(xy-u(x^2+y^2))}{1-4u^2}\right)}{\sqrt{1-4u^2}}$$

1. Definition of Hermite polynomials: recurrence of order 2;

2. Product by linear algebra: $H_{n+k}(x)H_{n+k}(y)/(n+k)!,\ k \in \mathbf{N}$ generated over $\mathbf{Q}(x,n)$ by

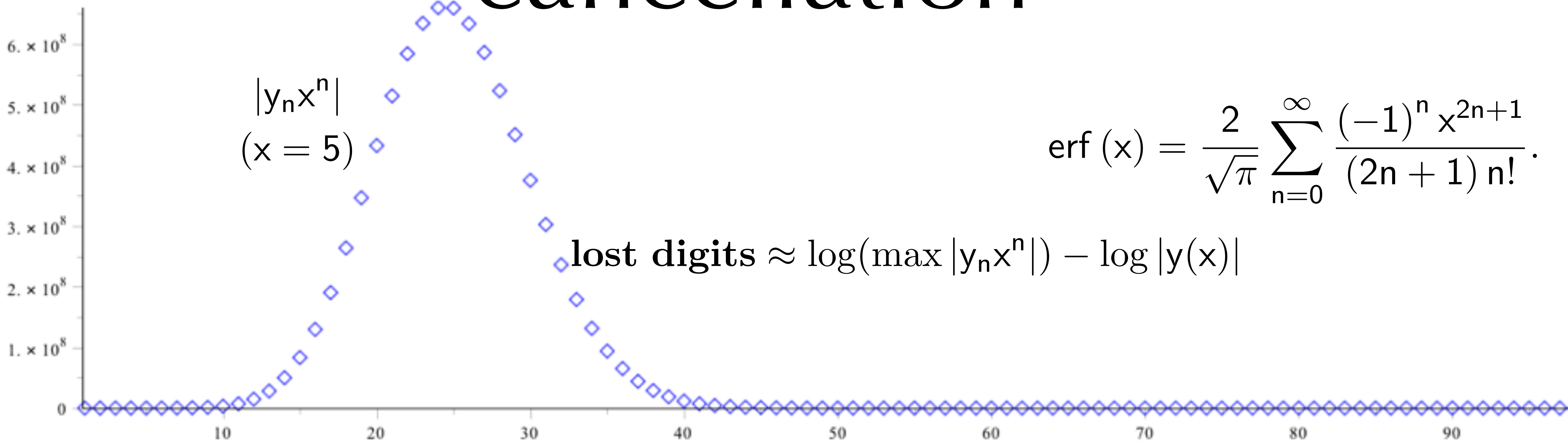
$$\frac{H_n(x)H_n(y)}{n!}, \frac{H_{n+1}(x)H_n(y)}{n!}, \frac{H_n(x)H_{n+1}(y)}{n!}, \frac{H_{n+1}(x)H_{n+1}(y)}{n!}$$

→ recurrence of order at most 4;

3. Translate into a differential equation.

demo

Preventing catastrophic cancellation



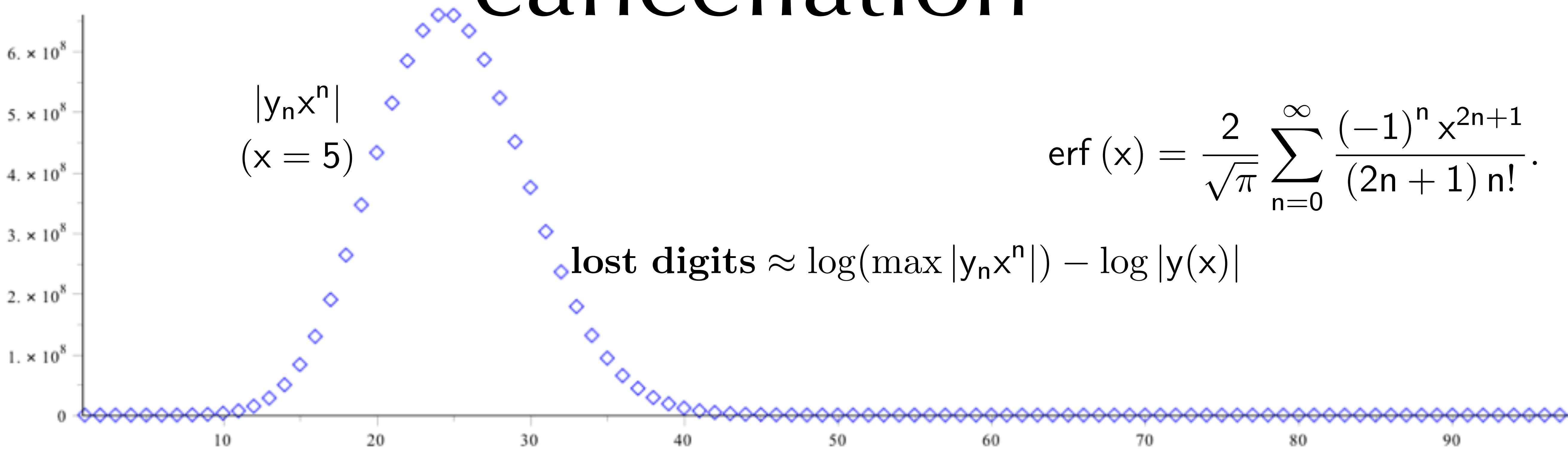
$|y_n x^n|$

$(x = 5)$

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

$$\textbf{lost digits} \approx \log(\max |y_n x^n|) - \log |y(x)|$$

Preventing catastrophic cancellation

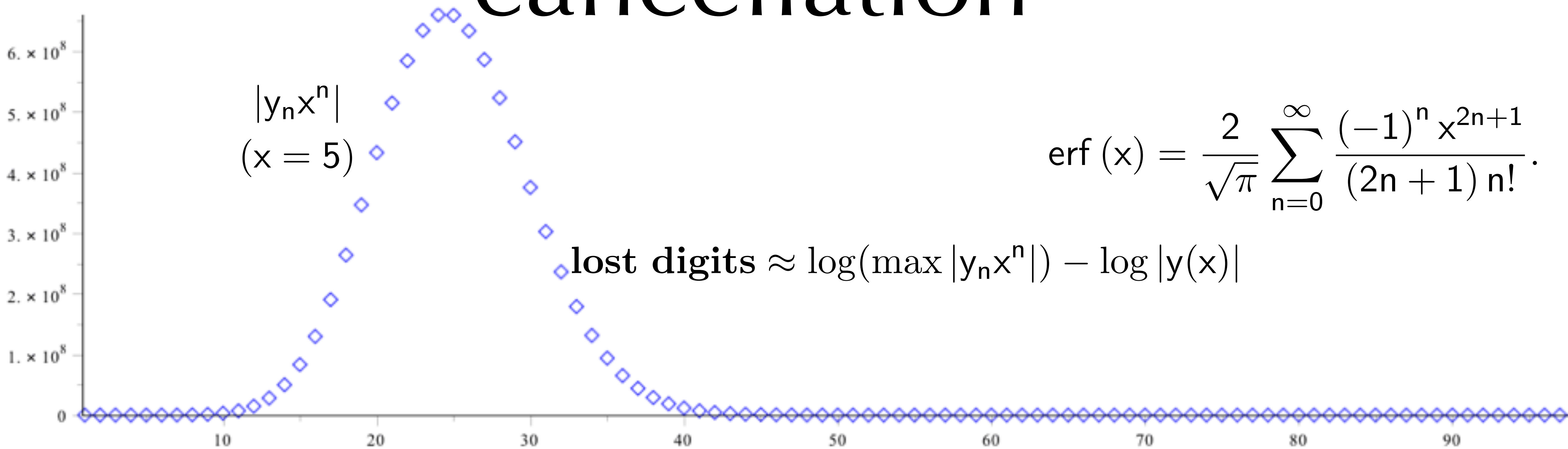


$$\mathrm{erf}\,(x) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)\, n!}.$$

$$\textbf{lost digits} \approx \log(\max|y_n x^n|) - \log|y(x)|$$

- Solution: $\exp(x^2)\mathrm{erf}(x)$ has positive coefficients.

Preventing catastrophic cancellation

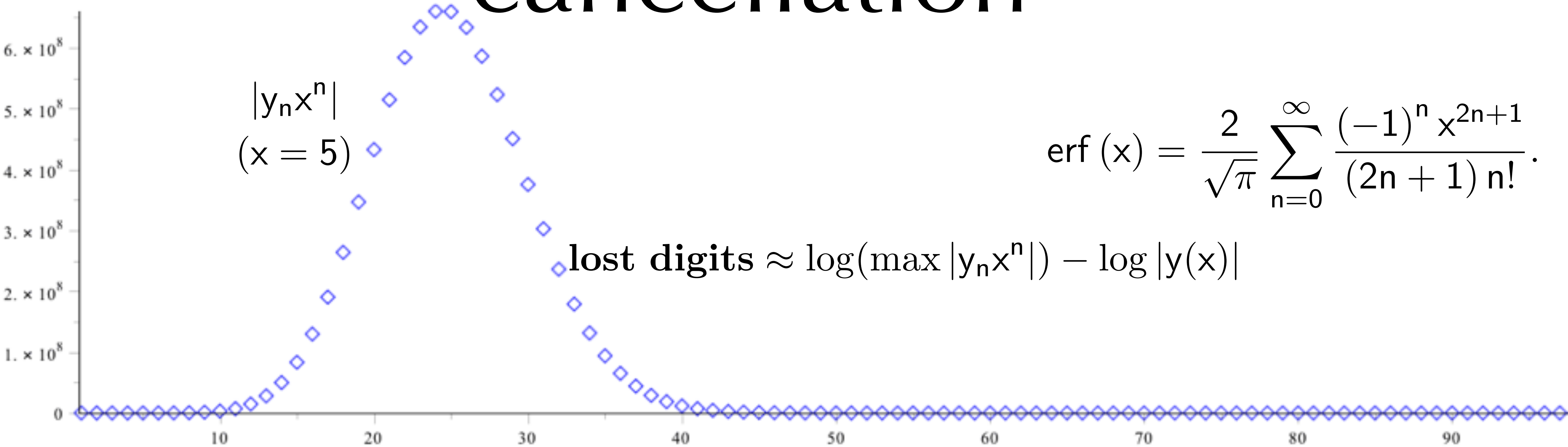


$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)\, n!}.$$

$$\textbf{lost digits} \approx \log(\max |y_n x^n|) - \log |y(x)|$$

- Solution: $\exp(x^2)\operatorname{erf}(x)$ has positive coefficients.
- Can be generalized [GawronskiMullerReinhard2007].

Preventing catastrophic cancellation

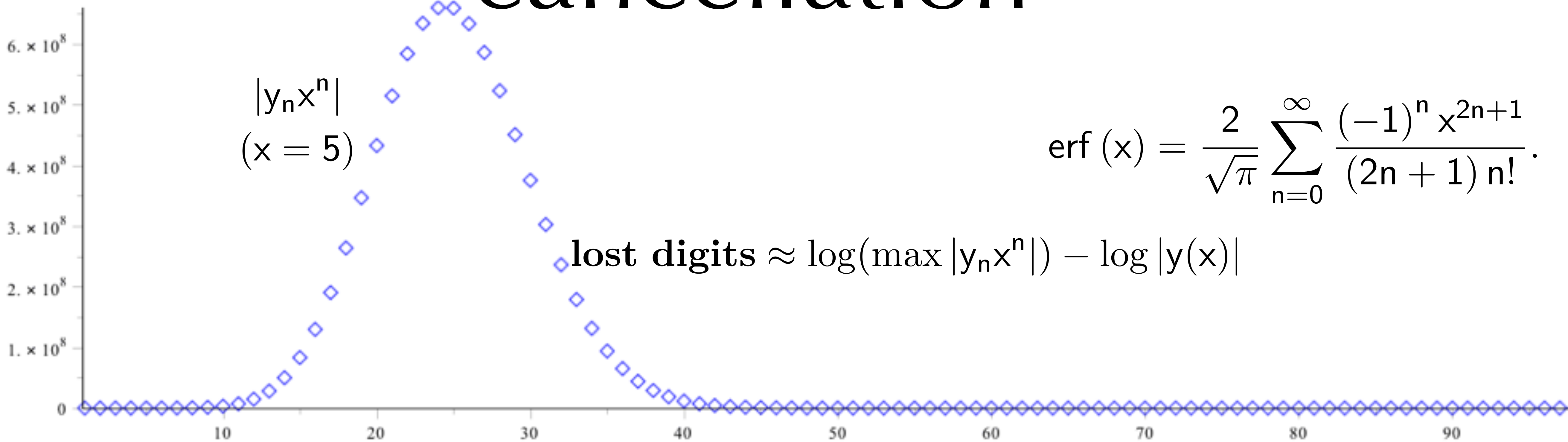


$$\mathrm{erf}\,(x) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)\, n!}.$$

$$\textbf{lost digits} \approx \log(\max |y_n x^n|) - \log |y(x)|$$

- Solution: $\exp(x^2)\mathrm{erf}(x)$ has positive coefficients.
- Can be generalized [GawronskiMullerReinhard2007].
- Chevillard-Mezzarobba2013 use Ai(jx)Ai(x/j)Ai(x) to compute Ai(x).

Preventing catastrophic cancellation



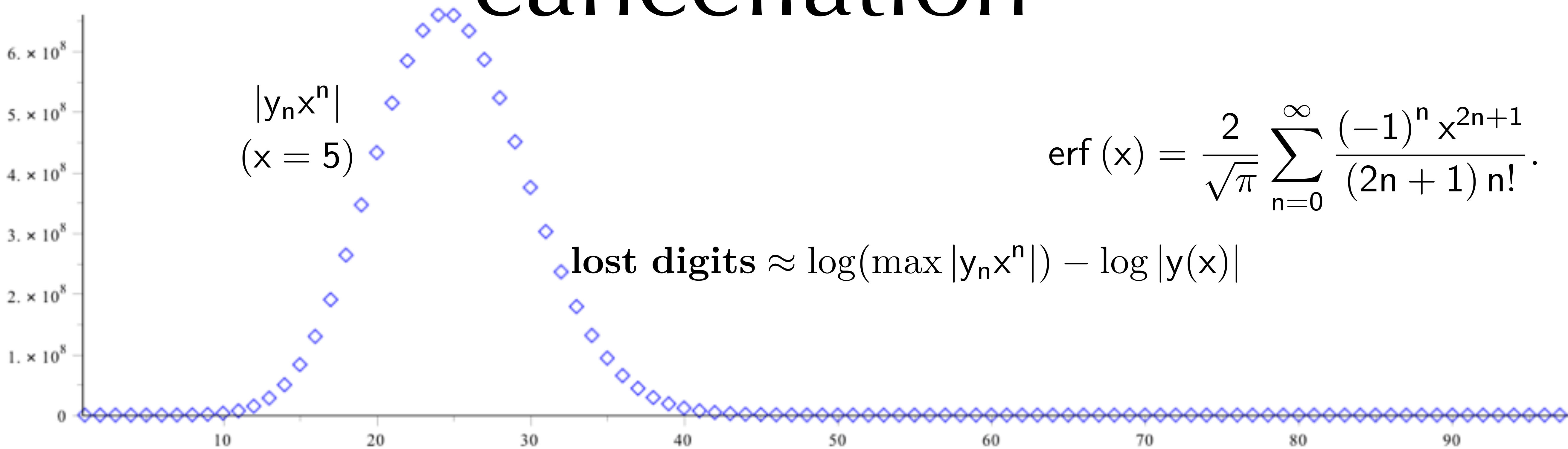
$|y_n x^n|$
$(x = 5)$

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

$$\textbf{lost digits} \approx \log(\max|y_n x^n|) - \log|y(x)|$$

- Solution: $\exp(x^2)\mathrm{erf}(x)$ has positive coefficients.
- Can be generalized [GawronskiMullerReinhard2007].
- Chevillard-Mezzarobba2013 use Ai(jx)Ai(x/j)Ai(x) to compute Ai(x).
- Serra *et alii* 2014 use it for probability of collisions.

Preventing catastrophic cancellation



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

$$\textbf{lost digits} \approx \log(\max |y_n x^n|) - \log |y(x)|$$

- Solution: $\exp(x^2)\operatorname{erf}(x)$ has positive coefficients.
- Can be generalized [GawronskiMullerReinhard2007].
- Chevillard-Mezzarobba2013 use Ai(jx)Ai(x/j)Ai(x) to compute Ai(x).
- Serra *et alii* 2014 use it for probability of collisions.

In all cases: closure by product gives recurrences

IV. Ore polynomials

From equations to operators

$D \leftrightarrow d/dx$
$x \leftrightarrow$ mult by x
product $\leftrightarrow$ composition
$Dx=xD+1$

$S \leftrightarrow (n \mapsto n+1)$
$n \leftrightarrow$ mult by n
product $\leftrightarrow$ composition
$Sn=(n+1)S$

From equations to operators

$D \leftrightarrow d/dx$	$S \leftrightarrow (n \mapsto n+1)$
$x \leftrightarrow$ mult by x	$n \leftrightarrow$ mult by n
product $\leftrightarrow$ composition	product $\leftrightarrow$ composition
$Dx=xD+1$	$Sn=(n+1)S$

Taylor morphism: $D \mapsto (n+1)S$; $x \mapsto S^{-1}$

produces linear recurrence from LDE

From equations to operators

$D \leftrightarrow d/dx$
$x \leftrightarrow$ mult by x
product $\leftrightarrow$ composition
$Dx=xD+1$

$S \leftrightarrow (n \mapsto n+1)$
$n \leftrightarrow$ mult by n
product $\leftrightarrow$ composition
$Sn=(n+1)S$

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

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.

Reduction of order

Reduction of order

Input: a (large) linear differential equation + ini. cond
Output: a factor annihilating this solution

Reduction of order

Input: a (large) linear differential equation + ini. cond
Output: a factor annihilating this solution

- Step 1: use the recurrence to compute a large number of terms;

Reduction of order

Input: a (large) linear differential equation + ini. cond
Output: a factor annihilating this solution

- Step 1: use the recurrence to compute a large number of terms;

- Step 2: guess a linear differential equation annihilating this series (linear algebra, Padé-Hermite approximants);

Reduction of order

Input: a (large) linear differential equation + ini. cond
Output: a factor annihilating this solution

- Step 1: use the recurrence to compute a large number of terms;

- Step 2: guess a linear differential equation annihilating this series (linear algebra, Padé-Hermite approximants);

- Step 3: prove the guess by *Euclidean division*.

GCRD & LCLM

greatest common right *divisor &*
least common left *multiple*

GCRD(A,B): minimal operator whose solutions are common to A and B.
LCLM(A,B): minimal operator having the solutions of A and B for solutions.

Example: closure by sum.

Computation: Euclidean algorithm or linear algebra.

Example from a continued fraction expansion

$$\mathsf{P}_k = \mathsf{a}_k x^2 \mathsf{P}_{k-2} + \mathsf{P}_{k-1}, \quad \mathsf{a}_k = \begin{cases} \frac{2k}{(2k+1)(2k+3)}, & k \text{ even}, \\ \frac{-2(k+2)}{(2k+1)(2k+3)}, & k \text{ odd}. \end{cases}$$

Aim: a recurrence for all k.

Example from a continued fraction expansion

$$\mathsf{P}_\mathsf{k} = \mathsf{a}_\mathsf{k}\mathsf{x}^2\mathsf{P}_{\mathsf{k}-2} + \mathsf{P}_{\mathsf{k}-1}, \quad \mathsf{a}_\mathsf{k} = \begin{cases} \frac{2\mathsf{k}}{(2\mathsf{k}+1)(2\mathsf{k}+3)}, & \mathsf{k} \text{ even}, \\ \frac{-2(\mathsf{k}+2)}{(2\mathsf{k}+1)(2\mathsf{k}+3)}, & \mathsf{k} \text{ odd}. \end{cases}$$

Aim: a recurrence for all k.

- Step 1: use both recurrences to find a relation between P_k, P_{k+2}, P_{k+4} for even k and one for odd k;
- Step 2: compute their LCLM (order 8);
- Step 3: use the initial conditions to reduce (order 4).

Ore fractions

Generalize commutative case:

$R=Q^{-1}P$ with P & Q operators.
Sum and product reduce to that form using LCLM.

Ore fractions

Generalize commutative case:

$R=Q^{-1}P$ with P & Q operators.
Sum and product reduce to that form using LCLM.

Application: extend Taylor morphism to Chebyshev expansions

Ore fractions

Generalize commutative case:

$R=Q^{-1}P$ with P & Q operators.
Sum and product reduce to that form using LCLM.

Application: extend Taylor morphism to Chebyshev expansions

Taylor

$x^{n+1}=x\cdot x^n \leftrightarrow x \mapsto X:=S^{-1}$

$(x^n)'=nx^{n-1} \leftrightarrow d/dx \mapsto D:=(n+1)S$

Ore fractions

Generalize commutative case:

$R=Q^{-1}P$ with P & Q operators.
Sum and product reduce to that form using LCLM.

Application: extend Taylor morphism to Chebyshev expansions

Taylor

$x^{n+1}=x\cdot x^n \leftrightarrow x \mapsto X:=S^{-1}$

$(x^n)'=nx^{n-1} \leftrightarrow d/dx \mapsto D:=(n+1)S$

Chebyshev

$2xT_n(x)=T_{n+1}(x)+T_{n-1}(x)$

$\leftrightarrow x \mapsto X:=(S+S^{-1})/2$

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

$\leftrightarrow d/dx \mapsto D:=(1-X^2)^{-1}n(S-S^{-1})/2.$

Ore fractions

Generalize commutative case:

$R=Q^{-1}P$ with P & Q operators.
Sum and product reduce to that form using LCLM.

Application: extend Taylor morphism to Chebyshev expansions

Taylor

$x^{n+1}=x\cdot x^n \leftrightarrow x \mapsto X:=S^{-1}$

$(x^n)'=nx^{n-1} \leftrightarrow d/dx \mapsto D:=(n+1)S$

Chebyshev

$2xT_n(x)=T_{n+1}(x)+T_{n-1}(x)$

$\leftrightarrow x \mapsto X:=(S+S^{-1})/2$

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

$\leftrightarrow d/dx \mapsto D:=(1-X^2)^{-1}n(S-S^{-1})/2.$

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).

Conclusion

Summary

- 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 DDMF, including formal proofs.

New project: FastRelax

$y'' + 2xy' = 0$