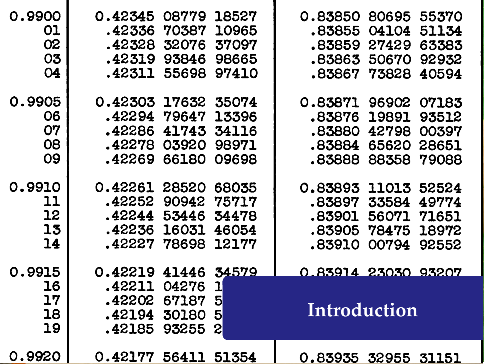
Around the Numerical Evaluation of D-Finite Functions

Marc Mezzarobba

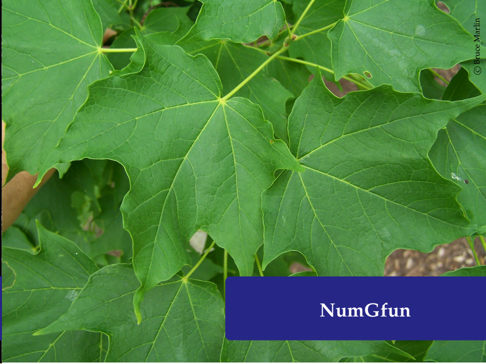
AriC Team – Inria

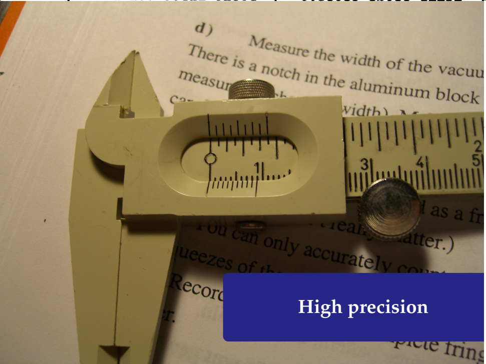
ENS DE LYON

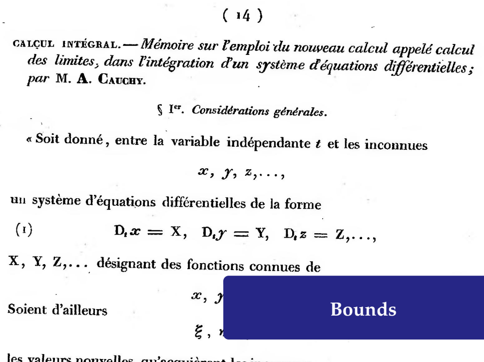
CAPA Seminar, Uppsala universitet
June 12, 2012



0.9900	0.42345 08779 18527	0.83850 80695 55370
01	.42336 70387 10965	.83855 04104 51134
02	.42328 32076 37097	.83859 27429 63383
03	.42319 93846 98665	.83863 50670 92932
04	.42311 55698 97410	.83867 73828 40594
0.9905	0.42303 17632 35074	0.83871 96902 07183
06	.42294 79647 13396	.83876 19891 93512
07	.42286 41743 34116	.83880 42798 00397
08	.42278 03920 98971	.83884 65620 28651
09	.42269 66180 09698	.83888 88358 79088
0.9910	0.42261 28520 68035	0.83893 11013 52524
11	.42252 90942 75717	.83897 33584 49774
12	.42244 53446 34478	.83901 56071 71651
13	.42236 16031 46054	.83905 78475 18972
14	.42227 78698 12177	.83910 00794 92552
0.9915	0.42219 41446 34579	0.83914 23030 93207
16	.42211 04276 [illegible]	[illegible]
17	.42202 67187 [illegible]	[illegible]
18	.42194 30180 [illegible]	[illegible]
19	.42185 93255 [illegible]	[illegible]
0.9920	0.42177 56411 51354	0.83935 32955 31151







(14)

CALCUL INTÉGRAL. — *Mémoire sur l'emploi du nouveau calcul appelé calcul des limites, dans l'intégration d'un système d'équations différentielles;* par M. A. Cauchy.

§ Ier. *Considérations générales.*

« Soit donné, entre la variable indépendante t et les inconnues

$$x, y, z, \dots,$$

un système d'équations différentielles de la forme

$$(1) \qquad D_t x = X, \quad D_t y = Y, \quad D_t z = Z, \dots,$$

X, Y, Z, ... désignant des fonctions connues de

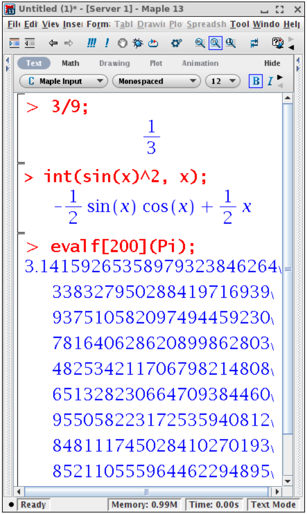
Soient d'ailleurs x, y ... ξ, η ...

les valeurs nouvelles qu'acquièrent les in...

0.9900	0.42345	08779	18527	0.83850	80695	55370
01	.42336	70387	10965	.83855	04104	51134
02	.42328	32076	37097	.83859	27429	63383
03	.42319	93846	98665	.83863	50670	92932
04	.42311	55698	97410	.83867	73828	40594
0.9905	0.42303	17632	35074	0.83871	96902	07183
06	.42294	79647	13396	.83876	19891	93512
07	.42286	41743	34116	.83880	42798	00397
08	.42278	03920	98971	.83884	65620	28651
09	.42269	66180	09698	.83888	88358	79088
0.9910	0.42261	28520	68035	0.83893	11013	52524
11	.42252	90942	75717	.83897	33584	49774
12	.42244	53446	34478	.83901	56071	71651
13	.42236	16031	46054	.83905	78475	18972
14	.42227	78698	12177	.83910	00794	92552
0.9915	0.42219	41446	34579	0.83914	23030	93207
16	.42211	04276	1[illegible]	[illegible]	[illegible]	[illegible]
17	.42202	67187	5[illegible]	[illegible]	[illegible]	[illegible]
18	.42194	30180	5[illegible]	[illegible]	[illegible]	[illegible]
19	.42185	93255	2[illegible]	[illegible]	[illegible]	[illegible]
0.9920	0.42177	56411	51354	0.83935	32955	31151

Introduction

Computer Algebra



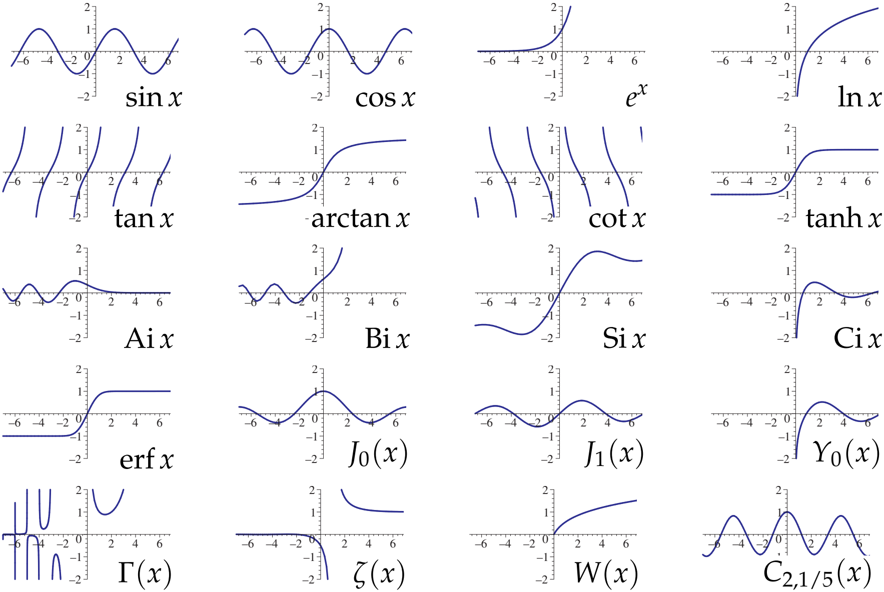
- exact + arbitrary precision computations
  “symbolic” tools
- computational complexity
- objects: polynomials, matrices…
  this talk: special functions
- symbolic-numeric algorithms
  accuracy issues → links with rigorous computing

Complexity: Fast Multiplication

$$
\begin{array}{rrrrr}
 & & 1 & 2 & 3 \\
\times & & 4 & 5 & 6 \\
\hline
 & & 7 & 3 & 8 \\
 & 6 & 1 & 5 & \\
4 & 9 & 2 & & \\
\hline
5 & 6 & 0 & 8 & 8
\end{array}
$$

- $\Theta(n^2)$ is not the best possible [Karatsuba, 1963]
- One can multiply two integers of $\leqslant n$ bits in $O(n \log n \, 2^{O(\log^* n)})$ bit ops [Fürer 2007].

Elementary and Special Functions



D-Finite Functions

An analytic function $y(z) : \mathbb{C} \to \mathbb{C}$ is said to be D-finite (holonomic) iff it satisfies a linear (homogeneous) ODE with polynomial coefficients:

$$a_r(z)\, y^{(r)}(z) + \cdots + a_1(z)\, y'(z) + a_0(z)\, y(z) = 0, \quad a_j \in \mathbb{K}[z].$$

- The sequence of Taylor coefficients of a D-finite functions obeys a linear *recurrence relation* with polynomial coefficients.

Example : $y(z) = \sin z$

$$y''(z) + y(z) = 0 \qquad y(0) = 0, \quad y'(0) = 1$$

D-Finite Functions

An analytic function $y(z) : \mathbb{C} \to \mathbb{C}$ is said to be D-finite (holonomic) iff it satisfies a linear (homogeneous) ODE with polynomial coefficients:

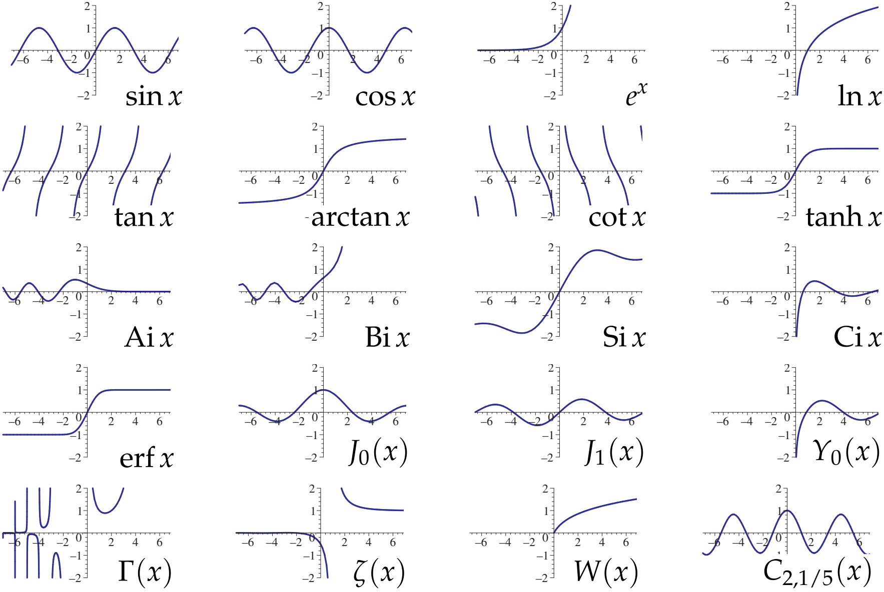
$$a_r(z)\, y^{(r)}(z) + \cdots + a_1(z)\, y'(z) + a_0(z)\, y(z) = 0, \quad a_j \in \mathbb{K}[z].$$

- The sequence of Taylor coefficients of a D-finite functions obeys a linear *recurrence relation* with polynomial coefficients.

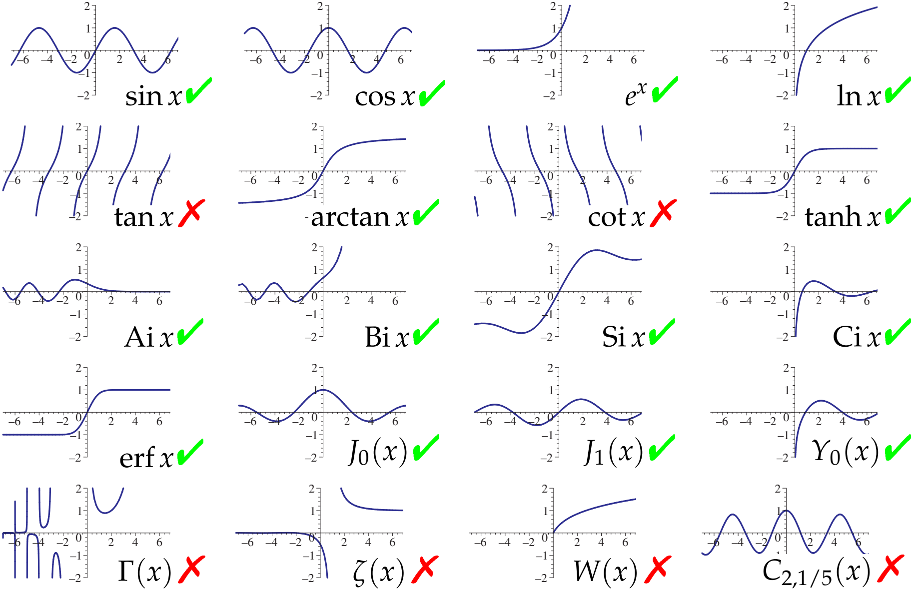
Example : $y(z) = K_0(z)$ (modified Bessel function)

$$z\, y''(z) + y'(z) - z\, y(z) = 0$$

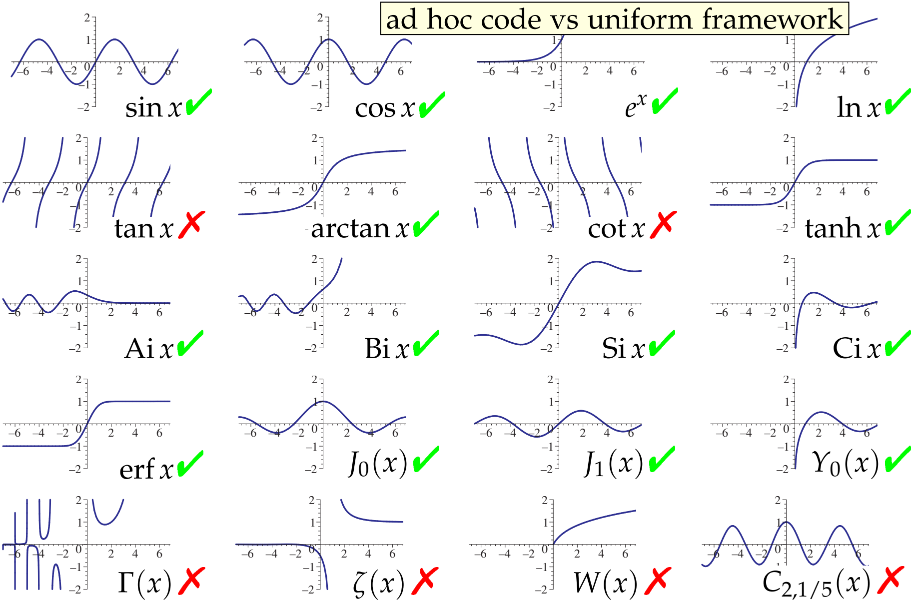
Elementary and Special Functions



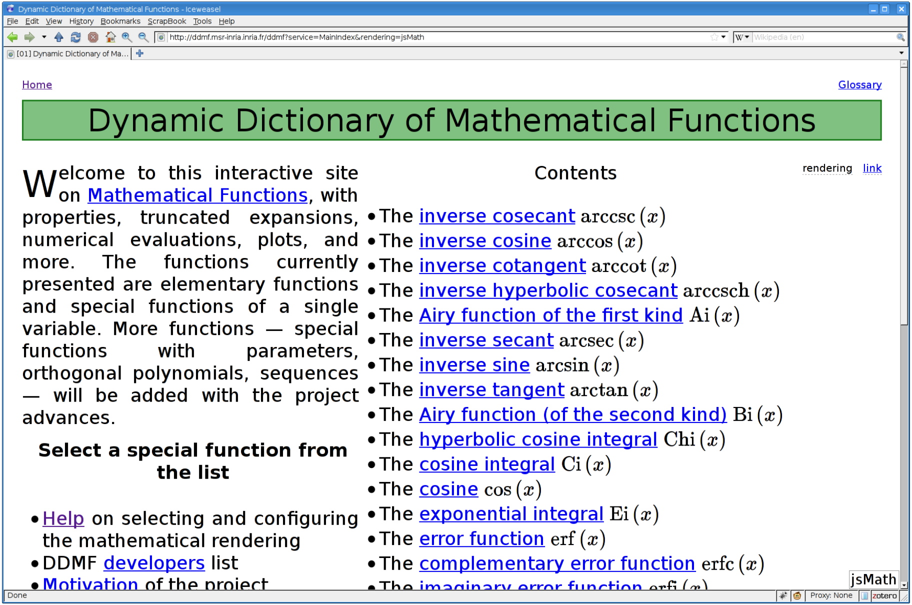
Elementary and Special Functions



Elementary and Special Functions



A Dictionary of D-Finite Functions



Dynamic Dictionary of Mathematical Functions - Iceweasel

File Edit View History Bookmarks ScrapBook Tools Help

http://ddmf.msr-inria.inria.fr/ddmf?service=MainIndex&rendering=jsMath

[01] Dynamic Dictionary of Ma...

Home

Glossary

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

Contents

rendering link

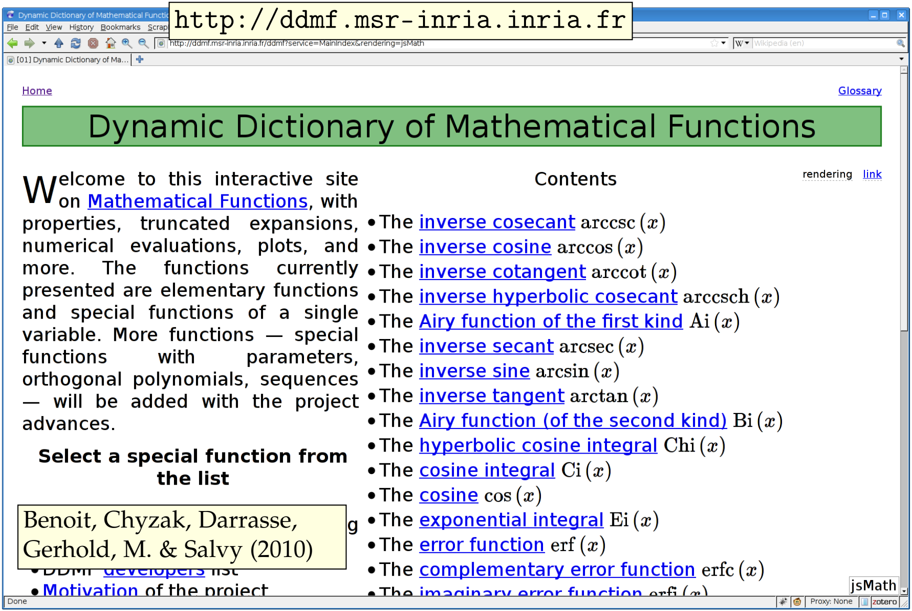
- The inverse cosecant $\operatorname{arccsc}(x)$
- The inverse cosine $\arccos(x)$
- The inverse cotangent $\operatorname{arccot}(x)$
- The inverse hyperbolic cosecant $\operatorname{arccsch}(x)$
- The Airy function of the first kind $\operatorname{Ai}(x)$
- The inverse secant $\operatorname{arcsec}(x)$
- The inverse sine $\arcsin(x)$
- The inverse tangent $\arctan(x)$
- The Airy function (of the second kind) $\operatorname{Bi}(x)$
- The hyperbolic cosine integral $\operatorname{Chi}(x)$
- The cosine integral $\operatorname{Ci}(x)$
- The cosine $\cos(x)$
- The exponential integral $\operatorname{Ei}(x)$
- The error function $\operatorname{erf}(x)$
- The complementary error function $\operatorname{erfc}(x)$
- The imaginary error function $\operatorname{erfi}(x)$

jsMath

Done Proxy: None zotero

A Dictionary of D-Finite Functions

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



Benoit, Chyzak, Darrasse, Gerhold, M. & Salvy (2010)

A Dictionary of D-Finite Functions

The Special Function Ai(x) - Iceweasel

File Edit View History Bookmarks ScrapBook Tools Help

http://ddmf.msr-inria.inria.fr/ddmf?service=SpecialFunction&rendering=jsMath&sf_name=AiryAi&parameters=({ })

[01] Loading...

Home Glossary

The Special Function $\mathrm{Ai}(x)$

1. Differential equation

rendering link

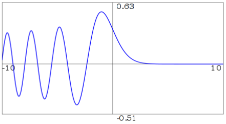
The function $\mathrm{Ai}(x)$ satisfies

$$\frac{d^2}{dx^2}y(x) - xy(x) = 0$$

with initial values $y(0) = 1/3\,\frac{\sqrt[3]{3}}{\Gamma(2/3)}$, $(y')(0) = -1/2\,\frac{\sqrt[6]{3}\,\Gamma(2/3)}{\pi}$.

metadata

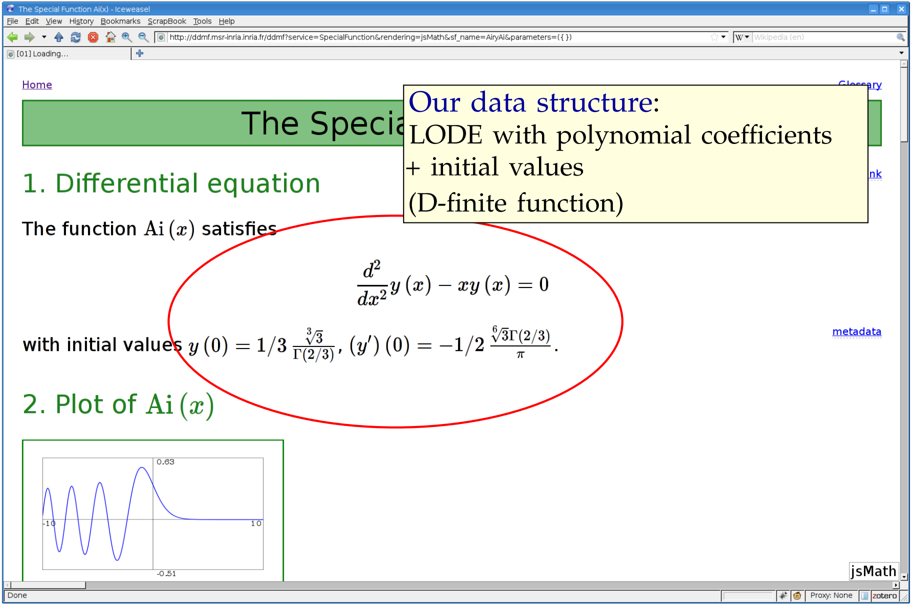
2. Plot of $\mathrm{Ai}(x)$



jsMath

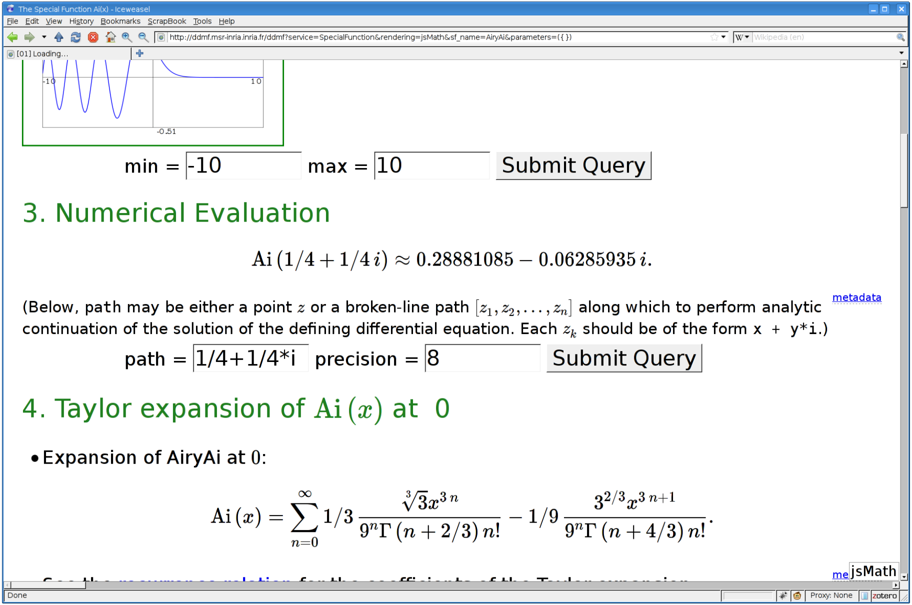
Done Proxy: None zotero

A Dictionary of D-Finite Functions



Our data structure:
LODE with polynomial coefficients
+ initial values
(D-finite function)

A Dictionary of D-Finite Functions



min = -10 max = 10 Submit Query

3. Numerical Evaluation

$$\mathrm{Ai}\,(1/4+1/4\,i) \approx 0.28881085 - 0.06285935\,i.$$

metadata

(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.)

path = 1/4+1/4*i precision = 8 Submit Query

4. Taylor expansion of $\mathrm{Ai}\,(x)$ at 0

- Expansion of AiryAi at 0:

$$\mathrm{Ai}\,(x) = \sum_{n=0}^{\infty} 1/3\,\frac{\sqrt[3]{3}x^{3\,n}}{9^n\Gamma\,(n+2/3)\,n!} - 1/9\,\frac{3^{2/3}x^{3\,n+1}}{9^n\Gamma\,(n+4/3)\,n!}.$$

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

me jsMath

Done Proxy: None zotero

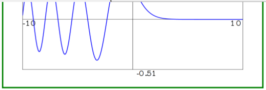
A Dictionary of D-Finite Functions

The Special Function Ai(x) - Iceweasel

File Edit View History Bookmarks ScrapBook Tools Help

http://ddmf.msr-inria.inria.fr/ddmf?service=SpecialFunction&rendering=jsMath&sf_name=AiryAi&parameters=({ })

[01] Loading...



min = -10 max = 10 Submit Query

3. Numerical Evaluation

$$\mathrm{Ai}\,(1/4 + 1/4\,i) \approx 0.28881085 - 0.06285935\,i.$$

metadata

(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.)

path = 1/4+1/4*i precision = 8 Submit Query

4. Taylor expansion of $\mathrm{Ai}\,(x)$ at 0

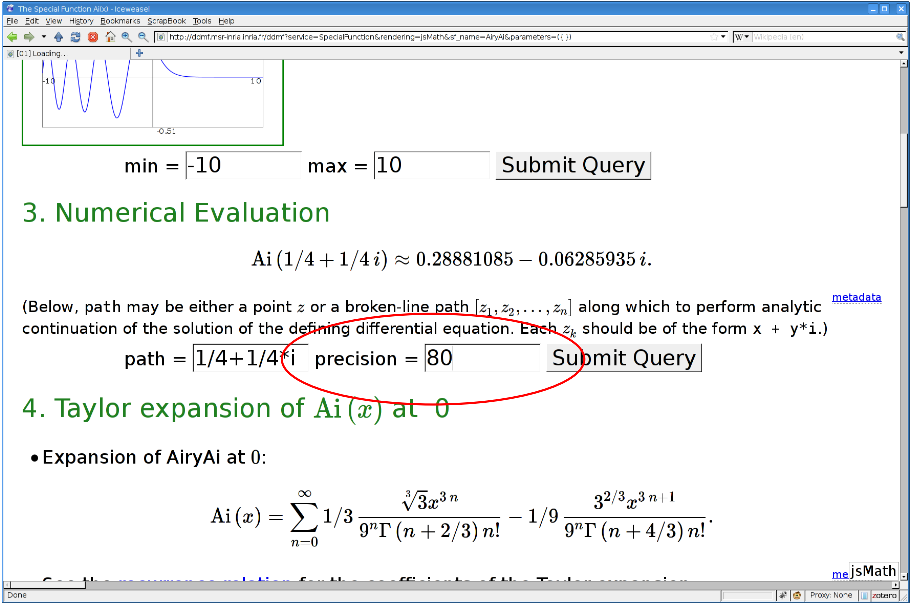
- Expansion of AiryAi at 0:

$$\mathrm{Ai}\,(x) = \sum_{n=0}^{\infty} 1/3\,\frac{\sqrt[3]{3}x^{3\,n}}{9^n\Gamma\,(n+2/3)\,n!} - 1/9\,\frac{3^{2/3}x^{3\,n+1}}{9^n\Gamma\,(n+4/3)\,n!}.$$

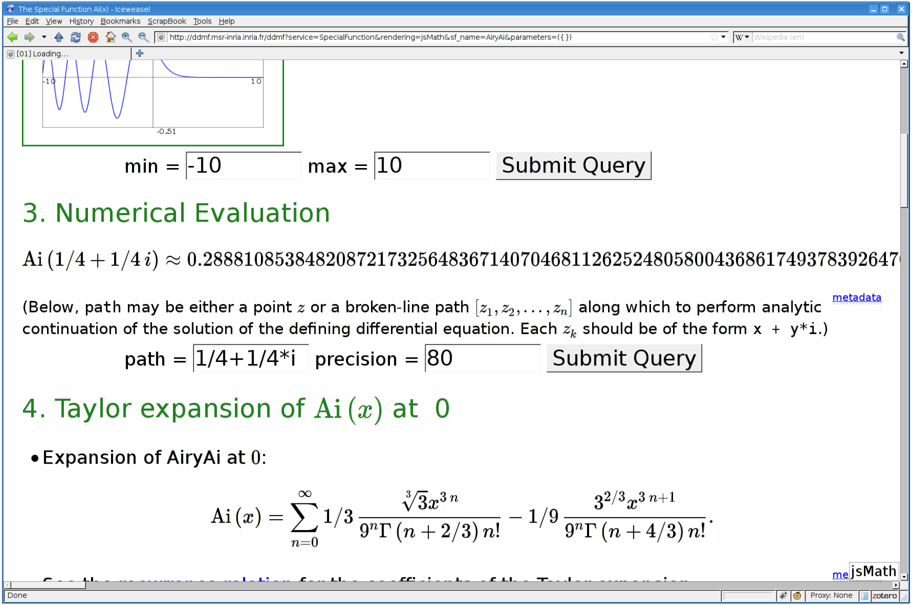
me jsMath

Done Proxy: None zotero

A Dictionary of D-Finite Functions



A Dictionary of D-Finite Functions


The Special Function Ai(x) - Iceweasel

File Edit View History Bookmarks ScrapBook Tools Help

http://ddmf.msr-inria.inria.fr/ddmf?service=SpecialFunction&rendering=jsMath&sf_name=AiryAi&parameters=({ })

[01] Loading...

min = -10 max = 10 Submit Query

3. Numerical Evaluation

$$\mathrm{Ai}\,(1/4 + 1/4\,i) \approx 0.28881085384820872173256483671407046811262524805800436861749378392647$$

(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.) metadata

path = 1/4+1/4*i precision = 80 Submit Query

4. Taylor expansion of $\mathrm{Ai}\,(x)$ at 0

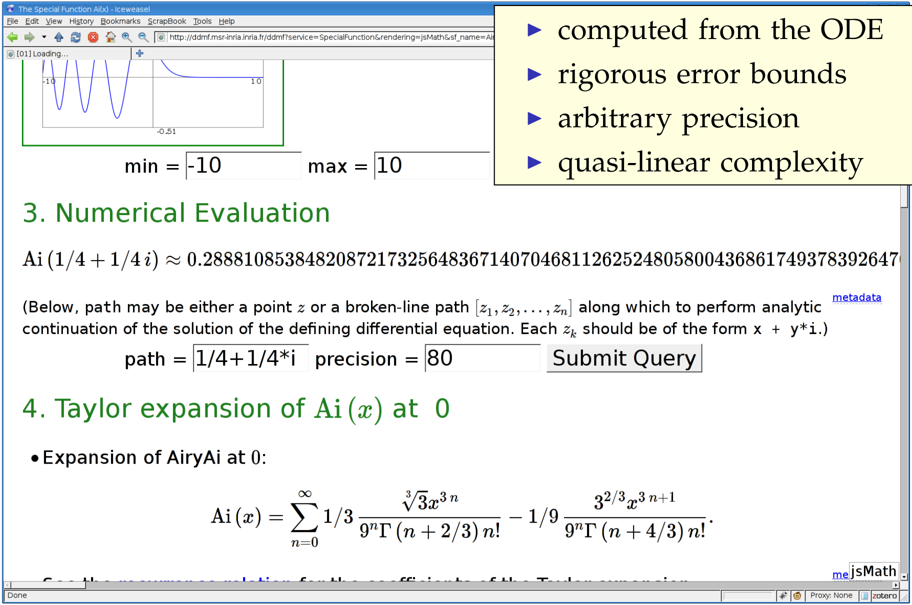
- Expansion of AiryAi at 0:

$$\mathrm{Ai}\,(x) = \sum_{n=0}^{\infty} 1/3\, \frac{\sqrt[3]{3} x^{3\,n}}{9^n \Gamma\,(n+2/3)\, n!} - 1/9\, \frac{3^{2/3} x^{3\,n+1}}{9^n \Gamma\,(n+4/3)\, n!}.$$

me jsMath

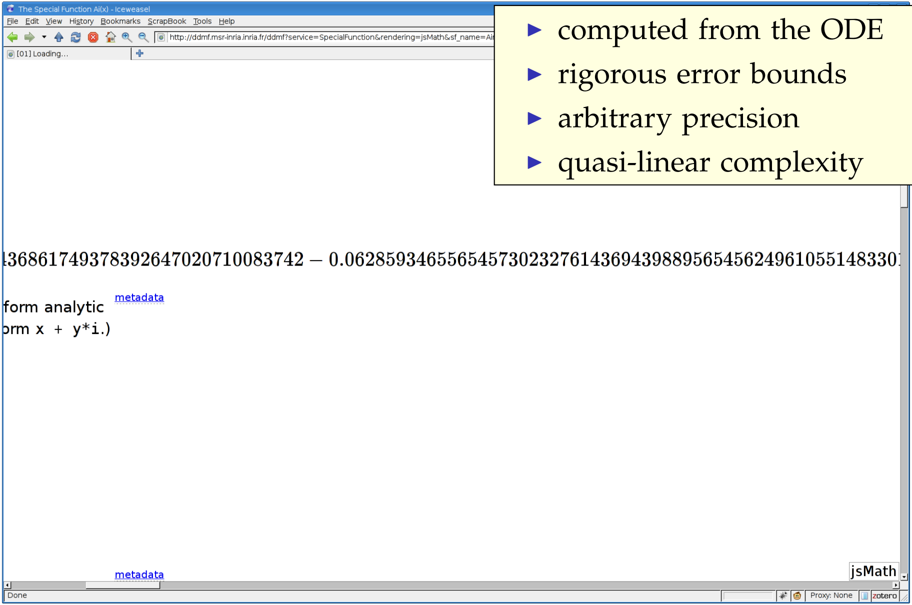
Done Proxy: None zotero

A Dictionary of D-Finite Functions



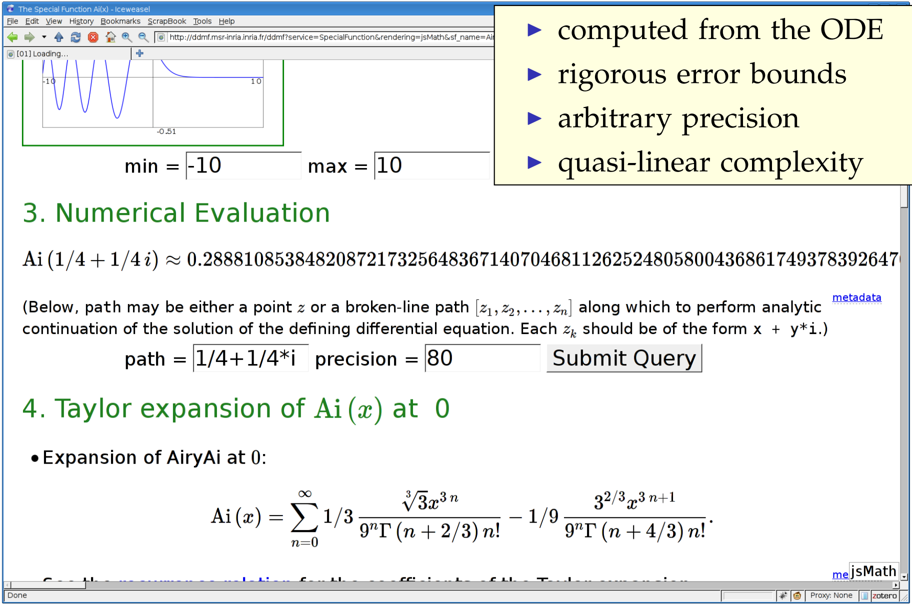
- computed from the ODE
- rigorous error bounds
- arbitrary precision
- quasi-linear complexity

A Dictionary of D-Finite Functions



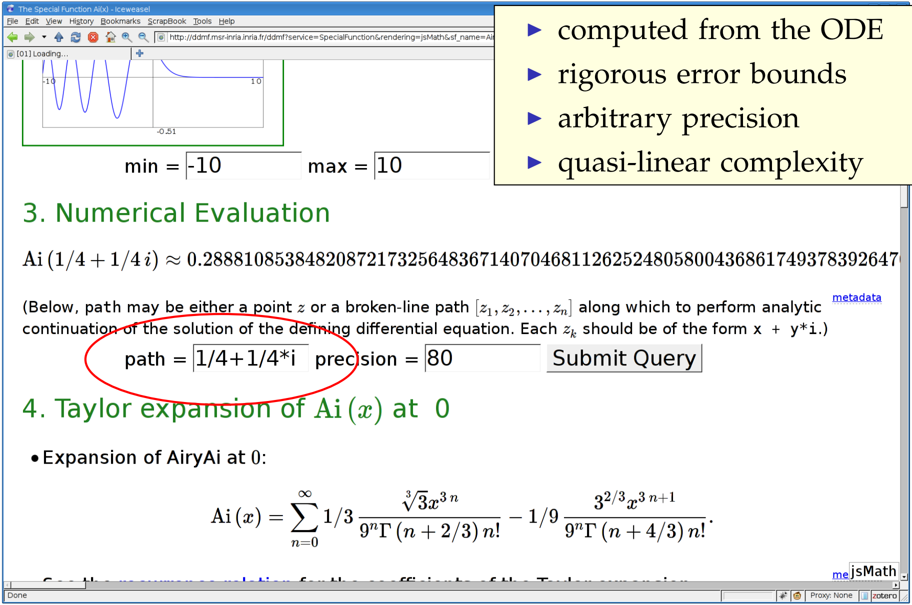
- computed from the ODE
- rigorous error bounds
- arbitrary precision
- quasi-linear complexity

A Dictionary of D-Finite Functions



- computed from the ODE
- rigorous error bounds
- arbitrary precision
- quasi-linear complexity

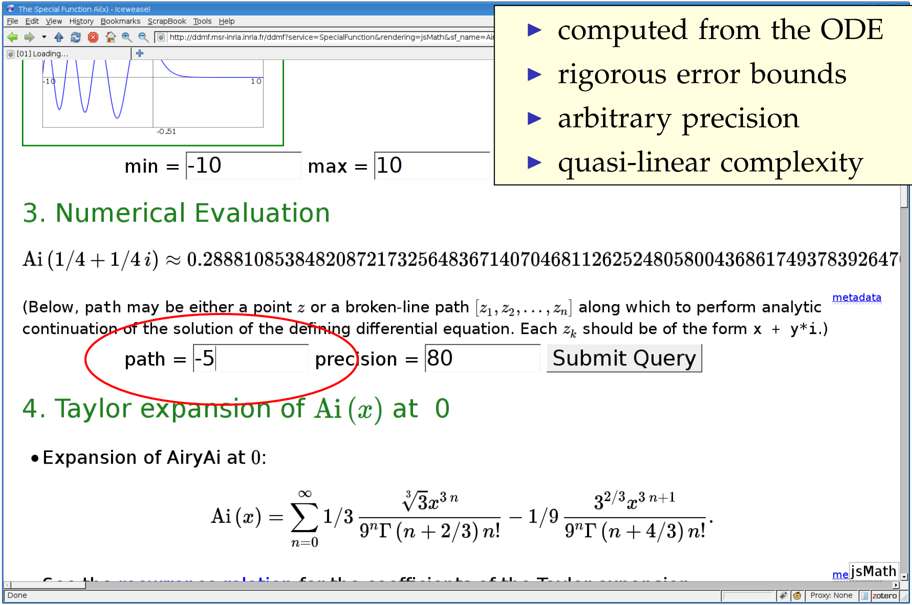
A Dictionary of D-Finite Functions



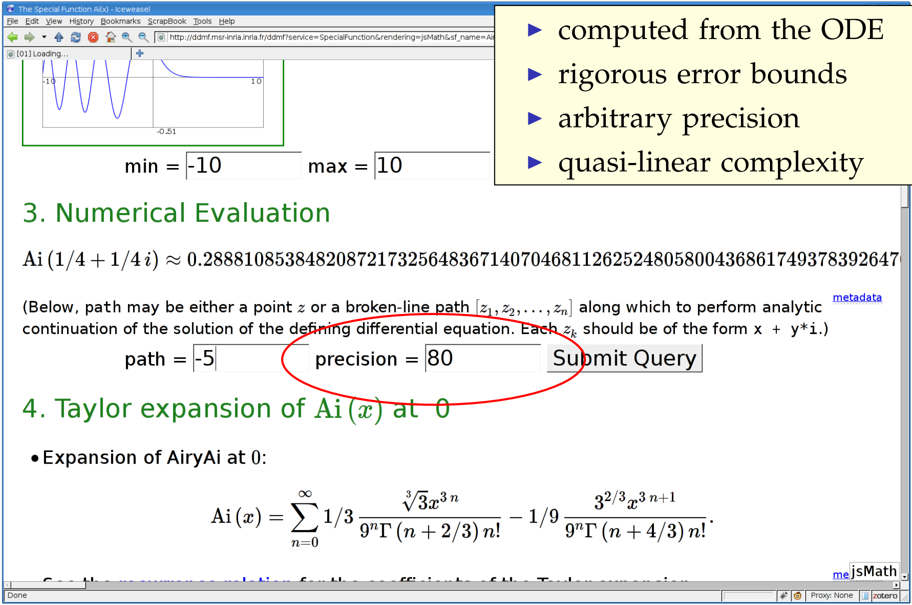
- computed from the ODE
- rigorous error bounds
- arbitrary precision
- quasi-linear complexity

A Dictionary of D-Finite Functions

- computed from the ODE
- rigorous error bounds
- arbitrary precision
- quasi-linear complexity



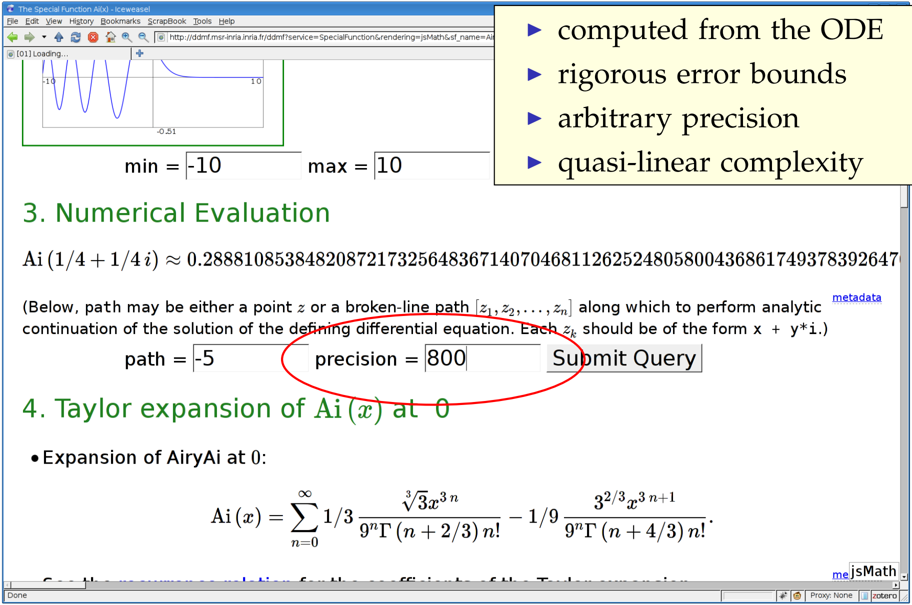
A Dictionary of D-Finite Functions



- computed from the ODE
- rigorous error bounds
- arbitrary precision
- quasi-linear complexity

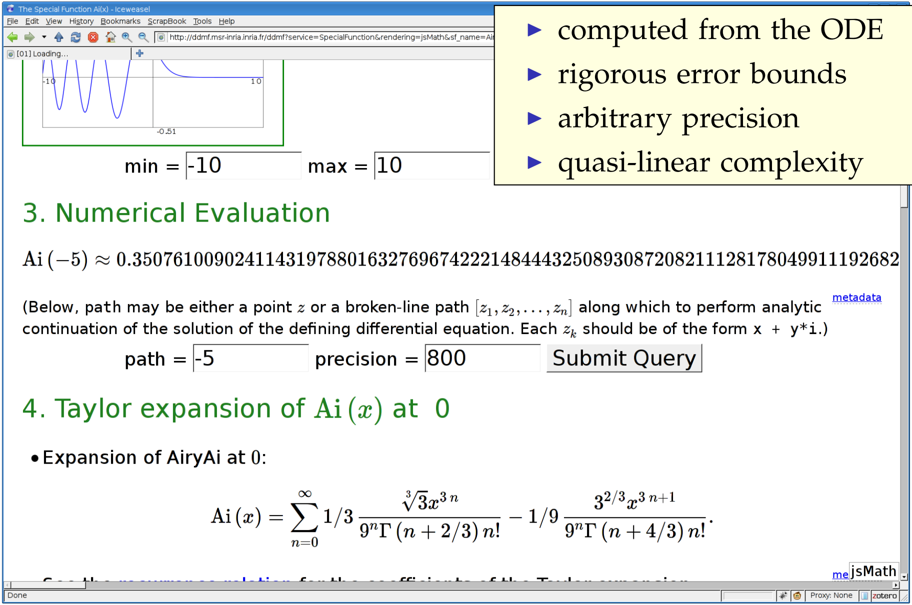
A Dictionary of D-Finite Functions

- computed from the ODE
- rigorous error bounds
- arbitrary precision
- quasi-linear complexity



A Dictionary of D-Finite Functions

- computed from the ODE
- rigorous error bounds
- arbitrary precision
- quasi-linear complexity



The Special Function Ai(x) - Iceweasel

File Edit View History Bookmarks ScrapBook Tools Help

http://ddmf.msr-inria.inria.fr/ddmf?service=SpecialFunction&rendering=jsMath&sf_name=Ai

[01] Loading...

min = -10 max = 10

3. Numerical Evaluation

$$\mathrm{Ai}\,(-5) \approx 0.350761009024114319788016327696742221484443250893087208211128178049911192682$$

(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.) metadata

path = -5 precision = 800 Submit Query

4. Taylor expansion of $\mathrm{Ai}\,(x)$ at 0

- Expansion of AiryAi at 0:

$$\mathrm{Ai}\,(x) = \sum_{n=0}^{\infty} 1/3\,\frac{\sqrt[3]{3}x^{3\,n}}{9^n\Gamma\left(n+2/3\right)n!} - 1/9\,\frac{3^{2/3}x^{3\,n+1}}{9^n\Gamma\left(n+4/3\right)n!}.$$

Done Proxy: None zotero


© Bruce Marlin
NumGfun

NumGfun

http://algo.inria.fr/libraries/ (GNU LGPL)

http://algo.inria.fr/libraries/papers/gfun.html

B. Salvy and P. Zimmermann. Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable. ACM TOMS, 1994.

M. Mezzarobba. NumGfun: a Package for Numerical and Analytic Computation with D-finite functions. ISSAC 2010.

M. Mezzarobba. Autour de l'évaluation numérique des fonctions D-finies. PhD thesis, École polytechnique, 2011.

The Double Confluent Heun Function

```
> diffeq := {diff(diff(y(z),z),z)+(2*z^3-
  z^2*a-2*z-a)/((z+1)^2*(z-1)^2)*diff(y(z)
  ,z)+(z^2*b+z*c+2*z*a+d)*y(z)/((z-1)^3*
  (z+1)^3),y(0)=1,(D(y))(0)=0};
```

The Double Confluent Heun Function

```
> diffeq := {diff(diff(y(z),z),z)+(2*z^3-
  z^2*a-2*z-a)/((z+1)^2*(z-1)^2)*diff(y(z)
  ,z)+(z^2*b+z*c+2*z*a+d)*y(z)/((z-1)^3*
  (z+1)^3),y(0)=1,(D(y))(0)=0};
```

$$diffeq := \left\{ \frac{d^2}{dz^2}\, y(z) + \frac{(2\,z^3 - z^2\,a - 2\,z - a)\left(\frac{d}{dz}\, y(z)\right)}{(z+1)^2\,(z-1)^2} + \frac{(z^2\,b + z\,c + 2\,z\,a + d)\; y(z)}{(z-1)^3\,(z+1)^3},\; y(0) = 1,\; \mathrm{D}(y)(0) = 0 \right\}$$

```
>
```

The Double Confluent Heun Function

```
> diffeq := {diff(diff(y(z),z),z)+(2*z^3-
  z^2*a-2*z-a)/((z+1)^2*(z-1)^2)*diff(y(z)
  ,z)+(z^2*b+z*c+2*z*a+d)*y(z)/((z-1)^3*
  (z+1)^3),y(0)=1,(D(y))(0)=0};
```

$$diffeq := \left\{ \frac{d^2}{dz^2}\, y(z) + \frac{(2\,z^3 - z^2\,a - 2\,z - a)\left(\frac{d}{dz}\, y(z)\right)}{(z+1)^2\,(z-1)^2} + \frac{(z^2\,b + z\,c + 2\,z\,a + d)\; y(z)}{(z-1)^3\,(z+1)^3},\, y(0) = 1,\, D(y)(0) = 0 \right\}$$

```
> a, b, c, d := 1, 1/3, 1/2, 3;
```

The Double Confluent Heun Function

```
> diffeq := {diff(diff(y(z),z),z)+(2*z^3-
z^2*a-2*z-a)/((z+1)^2*(z-1)^2)*diff(y(z)
,z)+(z^2*b+z*c+2*z*a+d)*y(z)/((z-1)^3*
(z+1)^3),y(0)=1,(D(y))(0)=0};
```

$$diffeq := \left\{ \frac{d^2}{dz^2}\, y(z) + \frac{(2\,z^3 - z^2\,a - 2\,z - a)\left(\frac{d}{dz}\, y(z)\right)}{(z+1)^2\,(z-1)^2} + \frac{(z^2\,b + z\,c + 2\,z\,a + d)\, y(z)}{(z-1)^3\,(z+1)^3},\, y(0) = 1,\, D(y)(0) = 0 \right\}$$

```
> a, b, c, d := 1, 1/3, 1/2, 3;
```

$$a, b, c, d := 1, \frac{1}{3}, \frac{1}{2}, 3$$

```
>
```

Accuracy Issues

```
> evalf[51](HeunD(a, b, c, d, 1/3));
```

Accuracy Issues

```
> evalf[51](HeunD(a, b, c, d, 1/3));
  1.23715744756395253918007831405821000395447403052069
>
```

Accuracy Issues

```
> evalf[51](HeunD(a, b, c, d, 1/3));
  1.23715744756395253918007831405821000395447403052069
> myHeunD := diffeqtoproc(diffeq, y(z)):
```

Accuracy Issues

```
> evalf[51](HeunD(a, b, c, d, 1/3));
  1.23715744756395253918007831405821000395447403052069
> myHeunD := diffeqtoproc(diffeq, y(z)):
>
```

Accuracy Issues

```
> evalf[51](HeunD(a, b, c, d, 1/3));
  1.23715744756395253918007831405821000395447403052069
> myHeunD := diffeqtoproc(diffeq, y(z)):
> myHeunD(1/3, 50);
```

Accuracy Issues

```
> evalf[51](HeunD(a, b, c, d, 1/3));
  1.23715744756395253918007831405821000395447403052069
> myHeunD := diffeqtoproc(diffeq, y(z)):
> myHeunD(1/3, 50);
  1.23715744756395253918007831405821000395447403052075
>
```

Accuracy Issues

```
> evalf[51](HeunD(a, b, c, d, 1/3));
  1.23715744756395253918007831405821000395447403052069
> myHeunD := diffeqtoproc(diffeq, y(z)):
> myHeunD(1/3, 50);
  1.23715744756395253918007831405821000395447403052075
>
```

Accuracy Issues

```
> evalf[51](HeunD(a, b, c, d, 1/3));
  1.23715744756395253918007831405821000395447403052069
> myHeunD := diffeqtoproc(diffeq, y(z)):
> myHeunD(1/3, 50);
  1.23715744756395253918007831405821000395447403052075
> myHeunD(1/3, 2000);
```

Accuracy Issues

```
> evalf[51](HeunD(a, b, c, d, 1/3));
  1.23715744756395253918007831405821000395447403052069
> myHeunD := diffeqtoproc(diffeq, y(z)):
> myHeunD(1/3, 50);
  1.23715744756395253918007831405821000395447403052075
> myHeunD(1/3, 2000);
```

(1.3 s later. . .)

Accuracy Issues

```
> evalf[51](HeunD(a, b, c, d, 1/3));
  1.23715744756395253918007831405821000395447403052069
> myHeunD := diffeqtoproc(diffeq, y(z)):
> myHeunD(1/3, 50);
  1.23715744756395253918007831405821000395447403052075
> myHeunD(1/3, 2000);
1.237157447563952539180078314058210003954474030520747249\
    77368122339910479272634279104260366917046868224326693\
    22058740005957868869065637255063771378117634825003548\
    [illegible]
    96170152380808246265230916158732964496323766777357428\
    28214335810166903875586333320334746574757060060591160\
    33361999970684428816250827723506800809
>
```

Accuracy Issues

```
> evalf[51](HeunD(a, b, c, d, 1/3));
  1.23715744756395253918007831405821000395447403052069
> myHeunD := diffeqtoproc(diffeq, y(z)):
> myHeunD(1/3, 50);
  1.23715744756395253918007831405821000395447403052075
> myHeunD(1/3, 2000);
1.23715744756395253918007831405821000395447403052074 7249\
    77368122339910479272634279104260366917046868224326693\
    22058740005957868869065637255063771378117634825003548\
  . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    [illegible]
    96170152380808246265230916158732964496323766777357428\
    28214335810166903875586333320334746574757060060591160\
    333619999706844288162508277235068008 09
>
```

more general code = less bugs!

Approaching a Singular Point

```
> evalf(HeunD(a, b, c, d, -0.9));
```

Approaching a Singular Point

```
> evalf(HeunD(a, b, c, d, -0.9));
```

2.695836763

```
>
```

Approaching a Singular Point

```
> evalf(HeunD(a, b, c, d, -0.9));
```

2.695836763

```
> myHeunD(-0.9, 9);
```

Approaching a Singular Point

```
> evalf(HeunD(a, b, c, d, -0.9));
                    2.695836763
> myHeunD(-0.9, 9);
                    2.695836219
>
```

Approaching a Singular Point

```
> evalf(HeunD(a, b, c, d, -0.9));
```

2.695836763

```
> myHeunD(-0.9, 9);
```

2.695836219

```
>
```

Approaching a Singular Point

```
> evalf(HeunD(a, b, c, d, -0.9));
                    2.695836763
> myHeunD(-0.9, 9);
                    2.695836219
> evalf(HeunD(a, b, c, d, -0.99));
```

Approaching a Singular Point

```
> evalf(HeunD(a, b, c, d, -0.9));
                        2.695836763
> myHeunD(-0.9, 9);
                        2.695836219
> evalf(HeunD(a, b, c, d, -0.99));
Warning, breaking after 2000 terms, the series
is not converging
                        undefined
>
```

Approaching a Singular Point

```
> evalf(HeunD(a, b, c, d, -0.9));
```

2.695836763

```
> myHeunD(-0.9, 9);
```

2.695836219

```
> evalf(HeunD(a, b, c, d, -0.99));
```

Warning, breaking after 2000 terms, the series is not converging

undefined

```
> myHeunD(-0.99);
```

Approaching a Singular Point

```
> evalf(HeunD(a, b, c, d, -0.9));
```
2.695836763

```
> myHeunD(-0.9, 9);
```
2.695836219

```
> evalf(HeunD(a, b, c, d, -0.99));
Warning, breaking after 2000 terms, the series
is not converging
```
undefined

```
> myHeunD(-0.99);
```
4.6775585280

```
>
```

Approaching a Singular Point

```
> evalf(HeunD(a, b, c, d, -0.9));
                        2.695836763
> myHeunD(-0.9, 9);
                        2.695836219
> evalf(HeunD(a, b, c, d, -0.99));
Warning, breaking after 2000 terms, the series
is not converging
                         undefined
> myHeunD(-0.99);
                       4.6775585280
> myHeunD(-0.99, 500);
```

Approaching a Singular Point

```
> evalf(HeunD(a, b, c, d, -0.9));
```

2.695836763

```
> myHeunD(-0.9, 9);
```

2.695836219

```
> evalf(HeunD(a, b, c, d, -0.99));
Warning, breaking after 2000 terms, the series is not converging
```

undefined

```
> myHeunD(-0.99);
```

4.6775585280

```
> myHeunD(-0.99, 500);
```

(6.1 s later. . .)

Approaching a Singular Point

```
> evalf(HeunD(a, b, c, d, -0.9));
```

2.695836763

```
> myHeunD(-0.9, 9);
```

2.695836219

```
> evalf(HeunD(a, b, c, d, -0.99));
Warning, breaking after 2000 terms, the series
is not converging
```

undefined

```
> myHeunD(-0.99);
```

4.6775585280

```
> myHeunD(-0.99, 500);
```

4.677558527966890481646371616414130565650323560409922037\
. . . .
89542201276207762696563032189351846152496641167932588\
4660460023972873078881

```
>
```

Approaching a Singular Point

```
> evalf(HeunD(a, b, c, d, -0.9));
                    2.695836763
> myHeunD(-0.9, 9);
                    2.695836219
> evalf(HeunD(a, b, c, d, -0.99));
Warning, breaking after 2000 terms, the series
is not converging
                    undefined
> myHeunD(-0.99);
                    4.6775585280
> myHeunD(-0.99, 500);
4.677558527966890481646371616414130565650323560409922037\
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    8954220127620776269656[illegible]
    4660460023972873078881
>
```

no numerical instability issues
(price to pay: computation time)

A Random Example

```
> diffeq := random_diffeq(3, 2);
```

A Random Example

```
> diffeq := random_diffeq(3, 2);
```

$$diffeq := \left\{\left(\frac{13}{30} + \frac{8}{15} z + \frac{7}{30} z^2\right) y(z) + \left(-\frac{9}{20} + \frac{29}{30} z - \frac{1}{12} z^2\right) \left(\frac{d}{dz} y(z)\right) + \left(-\frac{43}{60} + \frac{49}{60} z + \frac{11}{30} z^2\right) \left(\frac{d^2}{dz^2} y(z)\right) + \left(-\frac{7}{12} + \frac{17}{30} z - \frac{3}{5} z^2\right) \left(\frac{d^3}{dz^3} y(z)\right), y(0) = 0, D(y)(0) = \frac{7}{30}, D^{(2)}(y)(0) = -\frac{43}{60}\right\}$$

A Random Example

```
> diffeq := random_diffeq(3, 2);
```

$$diffeq := \left\{ \left(\frac{13}{30} + \frac{8}{15} z + \frac{7}{30} z^2 \right) y(z) + \left(-\frac{9}{20} + \frac{29}{30} z - \frac{1}{12} z^2 \right) \left(\frac{d}{dz} y(z) \right) + \left(-\frac{43}{60} + \frac{49}{60} z + \frac{11}{30} z^2 \right) \left(\frac{d^2}{dz^2} y(z) \right) + \left(-\frac{7}{12} + \frac{17}{30} z - \frac{3}{5} z^2 \right) \left(\frac{d^3}{dz^3} y(z) \right), y(0) = 0, D(y)(0) = \frac{7}{30}, D^{(2)}(y)(0) = -\frac{43}{60} \right\}$$

```
> evaldiffeq(diffeq, y(z), (1+I)/5, 40);
```

A Random Example

```
> diffeq := random_diffeq(3, 2);
```

$$diffeq := \left\{ \left(\frac{13}{30} + \frac{8}{15} z + \frac{7}{30} z^2\right) y(z) + \left(-\frac{9}{20} + \frac{29}{30} z - \frac{1}{12} z^2\right) \left(\frac{d}{dz} y(z)\right) + \left(-\frac{43}{60} + \frac{49}{60} z + \frac{11}{30} z^2\right) \left(\frac{d^2}{dz^2} y(z)\right) + \left(-\frac{7}{12} + \frac{17}{30} z - \frac{3}{5} z^2\right) \left(\frac{d^3}{dz^3} y(z)\right), y(0) = 0, D(y)(0) = \frac{7}{30}, D^{(2)}(y)(0) = -\frac{43}{60} \right\}$$

```
> evaldiffeq(diffeq, y(z), (1+I)/5, 40);
```

$$0.04485557487767843131893308147593115486 63 + 0.01990489830212805305047897725810997882 82\, I$$

High Precision

```
> evaldiffeq(diffeq, y(z), 1/5, 1000000);
```

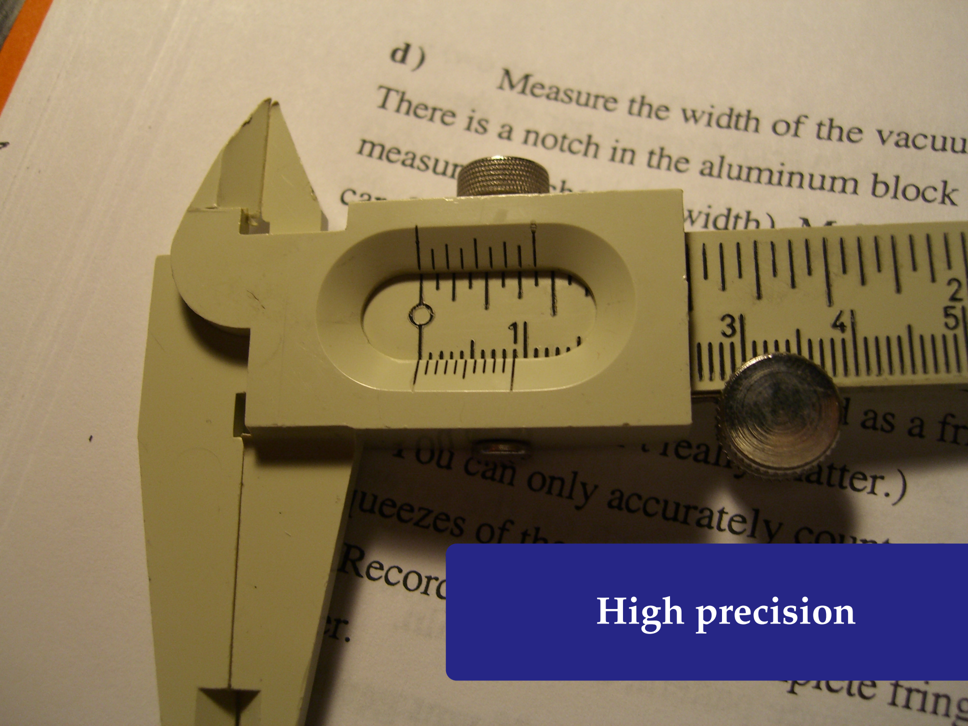
High Precision

```
> evaldiffeq(diffeq, y(z), 1/5, 1000000);
```

(29 min later…)

High Precision

```
> evaldiffeq(diffeq, y(z), 1/5, 1000000);
0.0332532812575675067724593819200243943910659613472928 63\
    1361178559307565437161078471985962090680571076277606 1\
    6599384479391829794197618862065053669108217914960590 4\
    3108048298855823993517550511176819489159174044677130 4\
    7473025189635972756153431009580734363927305651896233 3\
    9721759513884230988401642563243102957713043147210864 6\
    9548515476762402429734385158441412605623777191148968 0\
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    9793325825997236646657321960250165021813974778115734 8\
    7832262865574719581820528242814824080037691356145556 4\
    2959879449123182803958425643066993236588095610171972 7\
    3380613024394057453999112187785110527075237813842272 8\
    7617685959250804078177163720506043190222743767328690 1\
    7129257409846695090670592759003049446015009928821012 1\
    868701569
```



High precision

Some Previous Work

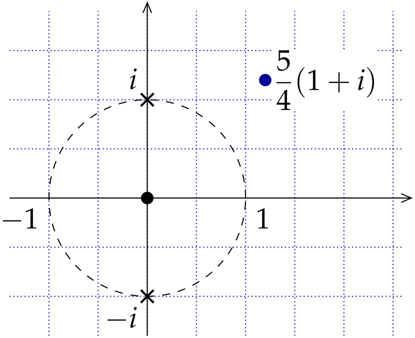
- Schroeppel (1972) – Special evaluation points
- Brent (1976) – Special case of exp (+ variants)
- Chudnovsky & Chudnovsky (1986-1988) – General method (incl. a sketch of the case of regular singular points)
- van der Hoeven (1999, 2001) – General algorithm with error bounds
- M. – Implementation, efficiency improvements, fully automatic error control based on tighter bounds

Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

0 fast integer multiplication
1 binary splitting
2 analytic continuation
3 *bit burst*

2. Taylor series method for ODEs



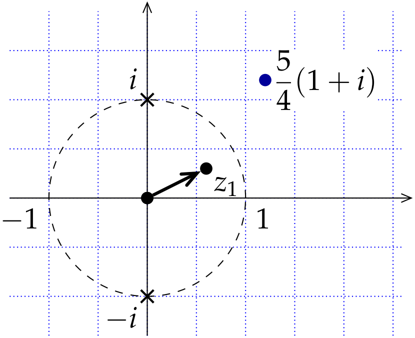
$$\arctan\left(\frac{5}{4}(1+i)\right) = ?$$

Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

0 fast integer multiplication
1 binary splitting
2 analytic continuation
3 *bit burst*

2. Taylor series method for ODEs



$$\arctan\left(\frac{5}{4}(1+i)\right) = ?$$

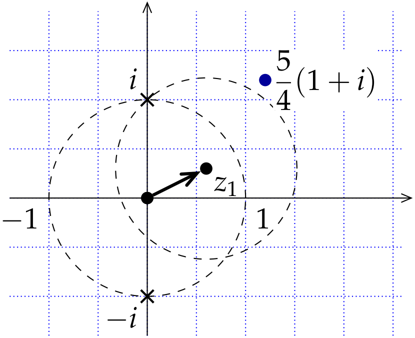
$$\begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix} = \begin{bmatrix} 1 & 0{,}570... + 0{,}220...\, i \\ 0 & 0{,}728... - 0{,}206...\, i \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

0 fast integer multiplication
1 binary splitting
2 analytic continuation
3 *bit burst*

2. Taylor series method for ODEs



$$\arctan\left(\frac{5}{4}(1+i)\right) = ?$$

$$\begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix} = \begin{bmatrix} 1 & 0{,}570...+0{,}220...\,i \\ 0 & 0{,}728...-0{,}206...\,i \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

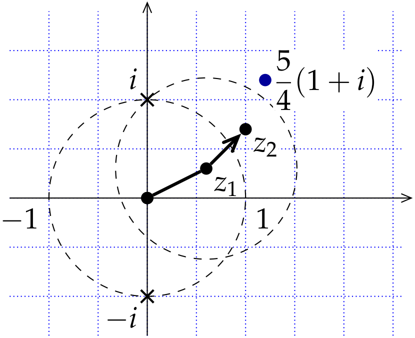
0 fast integer multiplication

1 binary splitting

2 analytic continuation

3 *bit burst*

2. Taylor series method for ODEs



$$\arctan\left(\frac{5}{4}(1+i)\right) = ?$$

$$\begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix} = \begin{bmatrix} 1 & 0{,}570... + 0{,}220...\,i \\ 0 & 0{,}728... - 0{,}206...\,i \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

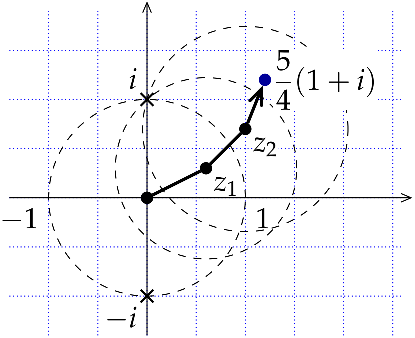
$$\begin{bmatrix} y(z_2) \\ y'(z_2) \end{bmatrix} = \begin{bmatrix} 1 & 0{,}365... + 0{,}329...\,i \\ 0 & 0{,}751... - 0{,}079...\,i \end{bmatrix} \begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix}$$

Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

0 fast integer multiplication
1 binary splitting
2 analytic continuation
3 *bit burst*

2. Taylor series method for ODEs



$$\arctan\left(\frac{5}{4}(1+i)\right) = ?$$

$$\begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix} = \begin{bmatrix} 1 & 0{,}570\ldots + 0{,}220\ldots i \\ 0 & 0{,}728\ldots - 0{,}206\ldots i \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

$$\begin{bmatrix} y(z_2) \\ y'(z_2) \end{bmatrix} = \begin{bmatrix} 1 & 0{,}365\ldots + 0{,}329\ldots i \\ 0 & 0{,}751\ldots - 0{,}079\ldots i \end{bmatrix} \begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix}$$

$\ldots$

Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

0 fast integer multiplication

1 binary splitting

2 analytic continuation

3 *bit burst*

0. One can multiply two integers of $\leqslant n$ bits in $M(n) = O(n \log n \, 2^{O(\log^* n)})$ bit ops [Fürer 2007].

Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

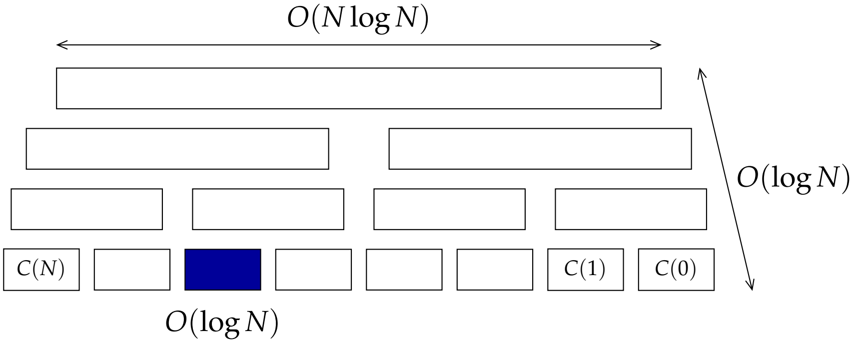
0 fast integer multiplication

1 binary splitting

2 analytic continuation

3 *bit burst*

1. Within the disk of convergence of a Taylor expansion: fast series summation algorithm based on the recurrence



Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

0 fast integer multiplication

1 binary splitting

2 analytic continuation

3 *bit burst*

3. High-precision inputs:
   use analytic continuation even if the series converges!

$$
\begin{aligned}
z_0 = 10_2 &\to z_1 = 10{,}1_2 \\
&\to z_2 = 10{,}101_2 \\
&\to z_3 = 10{,}1011011_2 \\
&\to z_4 = 10{,}101101110010100_2 \\
&\to \ldots \\
&\to z \ = 10.1011011100101001100000\ldots\ldots_2 \simeq e
\end{aligned}
$$

$\sin(e) = \sin(2{,}718\ldots) = ?$

Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

0 fast integer multiplication
1 binary splitting
2 analytic continuation
3 *bit burst*

Theorem (Chudnovsky^2)

The evaluation point z being fixed, one may compute $y(z)$ with error bounded by 2^{-n} in

$$O\Big(M\Big(n \cdot (\log n)^3\Big)\Big)$$

bit operations.

Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

0 fast integer multiplication

1 binary splitting

2 analytic continuation

3 *bit burst*

Theorem (Chudnovsky^2, van der Hoeven)

The evaluation point z being fixed, one may compute $y(z)$ with error bounded by 2^{-n} in

$$O\left(M\left(n \cdot \cancel{(\log n)^3} (\log n)^2 \cdot \log\log n\right)\right)$$

bit operations.

Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

0 fast integer multiplication

1 binary splitting

2 analytic continuation

3 *bit burst*

Theorem (Chudnovsky^2, van der Hoeven, M.)

The evaluation point z being fixed, one may compute $y(z)$ with error bounded by 2^{-n} in

$$O\Big(M\Big(n \cdot \cancel{(\log n)^3} (\log n)^2 \cdot \cancel{\log\log n}\Big)\Big)$$

bit operations.

Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

0 fast integer multiplication

1 binary splitting

2 analytic continuation

3 *bit burst*

Theorem (Chudnovsky², van der Hoeven, M.)

The evaluation point z being fixed, one may compute $y(z)$ with error bounded by 2^{-n} in

$$O\Big(M\Big(n \cdot \cancel{(\log n)^3}(\log n)^2 \cdot \cancel{\log\log n}\Big)\Big)$$

bit operations using $O(n)$ bits of memory.

M. Mezzarobba. A note on the space complexity of fast D-finite function evaluation. CASC 2012 (to appear).

Improvements Towards A Practical Algorithm

Error control

- Precision of intermediate steps
- Tight a priori bounds on truncation orders

"Constant factor"

- Structure of recurrence matrices
- Fast simultaneous computations of several derivatives

Regular singular points

- "Operator version" of the Heffter-Poole method
- Specific binary splitting algorithm (faster in "hard" cases)

Regular Singular Points

```
> diffeq := diffeqtohomdiffeq(
    holexprtodiffeq(
      arctan(z), y(z)), y(z));
```

Regular Singular Points

```
> diffeq := diffeqtohomdiffeq(
    holexprtodiffeq(
      arctan(z), y(z)), y(z));
```

$$diffeq := \left\{ -2\,z \left(\frac{d}{dz}\, y(z) \right) + \left(-1 - z^2 \right) \left(\frac{d^2}{dz^2}\, y(z) \right),\ y(0) = 0,\ D(y)(0) = 1 \right\}$$

```
>
```

Regular Singular Points

```
> diffeq := diffeqtohomdiffeq(
    holexprtodiffeq(
      arctan(z), y(z)), y(z));
```

$$diffeq := \left\{ -2\, z \left(\frac{d}{dz}\, y(z) \right) + (-1 - z^2) \left(\frac{d^2}{dz^2}\, y(z) \right),\ y(0) = 0,\ \mathrm{D}(y)(0) = 1 \right\}$$

```
> subs(z=z-I, evaldiffeq(diffeq, y(z),
                                 [0, I], ord=3));
```

Regular Singular Points

```
> diffeq := diffeqtohomdiffeq(
    holexprtodiffeq(
      arctan(z), y(z)), y(z));
```

$$diffeq := \left\{ -2\, z \left(\frac{d}{dz}\, y(z) \right) + (-1 - z^2) \left(\frac{d^2}{dz^2}\, y(z) \right),\ y(0) = 0,\ D(y)(0) = 1 \right\}$$

```
> subs(z=z-I, evaldiffeq(diffeq, y(z),
                            [0, I], ord=3));
```

$$(-0.5000000000\, I) \left(\ln(z - I) + \frac{1}{2}\, I\, (z - I) - \frac{1}{8}\, (z - I)^2 \right) + (0.7853981634 + 0.3465735903\, I)$$

```
>
```

Regular Singular Points

```
> diffeq := diffeqtohomdiffeq(
    holexprtodiffeq(
      arctan(z), y(z)), y(z));
```

$$diffeq := \left\{ -2\,z \left(\frac{d}{dz}\,y(z)\right) + (-1 - z^2)\left(\frac{d^2}{dz^2}\,y(z)\right),\ y(0) = 0,\ D(y)(0) = 1 \right\}$$

```
> subs(z=z-I, evaldiffeq(diffeq, y(z),
                           [0, I], ord=3));
```

$$(-0.5000000000\,\mathrm{I})\left(\ln\right.$$

$$+ (0.7853981634 -$$

```
>
```

Applications:

- Bessel Functions
- Analytic Combinatorics
- Resummation, diff. Galois

CALÇUL INTÉGRAL. — *Mémoire sur l'emploi du nouveau calcul appelé calcul des limites, dans l'intégration d'un système d'équations différentielles; par* M. A. CAUCHY.

§ I^er. *Considérations générales.*

« Soit donné, entre la variable indépendante t et les inconnues

$$x,\ y,\ z,\ldots,$$

un système d'équations différentielles de la forme

$$(1)\qquad D_t x = X,\quad D_t y = Y,\quad D_t z = Z,\ldots,$$

X, Y, Z,... désignant des fonctions connues de

x, y, [redacted]

Soient d'ailleurs Bounds

ξ, [redacted]

les valeurs nouvelles, qu'acquièrent les in

Motivation (I): Numerical Evaluation

$$\sum_{n=0}^{\infty} y_n z^n = \sum_{n=0}^{N-1} y_n z^n + \underbrace{\sum_{n=N}^{\infty} y_n z^n}_{?}$$

Compute suitable truncation orders (and other bounds)?
A priori bounds tend to be easier to use in fast algorithms.

- Chudnovsky & Chudnovsky – Orders of magnitude only
- van der Hoeven (1999, 2001, 2003) – Cauchy-like bounds

We look for asymptotically optimal bounds

Motivation (II): Symbolic Bounds

Baxter Permutations (OEIS A001181)

- $(n+2)(n+3)B_n = (7n^2+7n-2)B_{n-1} + 8(n-1)(n-2)B_{n-2}$,
  $B_0 = B_1 = 1$
- $B_n \leqslant 2{,}9 \cdot 8^n$

Chudnovsky Formula for π

- $$\frac{1}{\pi} = \frac{12}{640320^{3/2}} \sum_{k=0}^{\infty} t_k$$

  $$\text{where} \quad t_k = \frac{(-1)^k (6k)!(13591409 + 545140134k)}{(3k)!(k!)^3 640320^{3k}}$$
- $$\left| \frac{640320^{3/2}}{12\pi} - \sum_{k=0}^{n-1} t_k \right| \leqslant 10^6(2{,}3n^3 + 13{,}6n^2 + 25n + 13{,}6)\alpha^n$$

  $$\text{where } \alpha = \frac{1}{151931373056000} \simeq 0,66 \cdot 10^{-14}$$

“Tight” Bounds

Input Recurrence + Initial terms
$$\{p_s(n)\, y_{n+s} + \cdots + p_0(n)\, y_n = 0, \quad y_0 = \ldots, y_1 = \ldots\}$$

Output $|y_n| \leqslant n!^{p/q}\, \alpha^n\, \varphi(n)$

φ subexponential, i.e. $\varphi(n) = e^{o(n)}$

- rigorous bound in all cases
- for generic initial values:
  optimal p/q and α (or even $\varphi(n) = n^{O(1)}$)

Theorem
One may compute p/q, α, φ fulfilling these conditions.

M. Mezzarobba and B. Salvy. Effective bounds for P-recursive sequences. Journal of Symbolic Computation, 2010.

Symbolic and Numeric Bounds

Bound Parameters

$\kappa, \alpha, \ldots \in \mathbb{Q}$ or $\bar{\mathbb{Q}}$ s.t.

$$|y_n| \leqslant n!^{\kappa} \cdot \alpha^n \cdot \varphi(n)$$

Main Tools:
Cauchy majorants
+ elementary asymptotic analysis

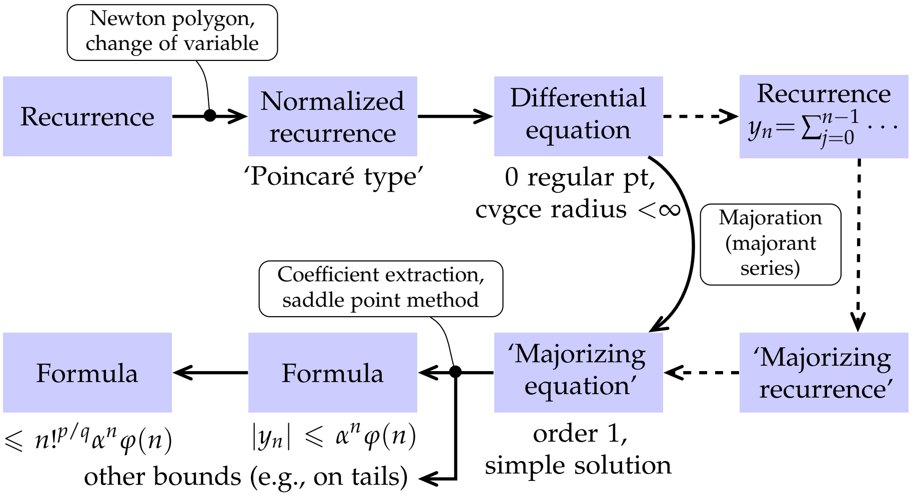
Symbolic Bounds

- Readable (as far as possible!)
- Asymptotically tight

Numeric Bounds

- Safe approx. of parameters
- Faster (no alg. numbers)

Outline of the Algorithm



Impact on Numerical Evaluation

	$\frac{\operatorname{arccot}(z)}{(z^2-1)(z^2+5)}$ @$\frac{1}{2}$	$\psi(1/2)$	$\arctan\frac{9}{10}$	$\arctan\frac{99}{100}$
10^{-10}	64/27	40/23	336/164	4238/1496
10^{-100}	380/321	342/313	2338/2108	25210/21848
10^{-1000}	3392/3307	3336/3293	22050/21754	231844/227810

	$\frac{\exp(1/(1-z))}{(1-z)}$ @$\frac{1}{2}$	$\operatorname{Bi}\left(\frac{1}{1-z}\right)$@$\frac{1}{2}$	$\operatorname{Ai}\left(\frac{1}{1-z}\right)$@$\frac{3}{4}$	$\operatorname{Ai}\left(\frac{1}{1-z}\right)$@$\frac{7}{8}$
10^{-10}	70/54	148/56	1558/77	23818/215
10^{-100}	418/387	664/416	3430/879	29258/2025
10^{-1000}	3568/3490	4700/3645	16284/8372	69594/18529

	e^{-100}	$\operatorname{erf}^2(1)$	$\operatorname{erf}(10)$	$\operatorname{erf}(100)$
10^{-10}	298/291	60/33	628/574	54492/54388
10^{-100}	456/450	190/163	936/894	54904/54800
10^{-1000}	1406/1402	1036/1011	2828/2800	58870/58772

terms: computed / optimal

Summary

- D-finite functions, DDMF
  compute everything we can starting from LODE + ini. cond.
- Multiple Precision Analytic Continuation
  general – rigorous – fully automatic – fast
- Tight bounds
  symbolic + numeric – Cauchy majorant method

Code available from

`http://algo.inria.fr/libraries/papers/gfun.html`

Current & future work

- NumGfun 1.0 and beyond
- Moderate precision, Rigorous polynomial approximations
- Majorants vs enclosures + fixed point thms
- D-Finite functions in Computer Arithmetic, code generation

Summary

- D-finite functions, DDMF
  compute everything we can starting from LODE + ini. cond.
- Multiple Precision Analytic Continuation
  general – rigorous – fully automatic – fast
- Tight bounds
  symbolic + numeric – Cauchy majorant method

Code available from

`http://algo.inria.fr/libraries/papers/gfun.html`

Current & future work

- NumGfun 1.0 and beyond
- Moderate precision, Rigorous polynomial approximations
- Majorants vs enclosures + fixed point thms
- D-Finite functions in Cor

Tack för er uppmärksamhet!

Summary

- D-finite functions, DDMF
  compute everything we can starting from LODE + ini. cond.
- Multiple Precision Analytic Continuation
  general – rigorous – fully automatic – fast
- Tight bounds
  symbolic + numeric – Cauchy majorant method

Code available from

`http://algo.inria.fr/libraries/papers/gfun.html`

Current & future work

- NumGfun 1.0 and beyond
- Moderate precision, Rigorous polynomial approximations
- Majorants vs enclosures + fixed point thms
- D-Finite functions in Computer Arithmetic, code generation

Credits

- *Tables of the error function and its derivative*. US National Bureau of Standards, 1954 (public domain)
- http://commons.wikimedia.org/wiki/File:Acer_saccharum_leaves.jpg (par Bruce Marlin, http://www.cirrusimage.com/tree_maple_sugar.htm, Creative Commons Attribution-Share Alike 2.5 Generic)
- http://commons.wikimedia.org/wiki/File:Calipers_in_physics_lab.jpg (by User:Falcorian, Creative Commons Attribution-Share Alike)
- Augustin-Louis Cauchy; Mémoire sur l'emploi du nouveau calcul, appelé calcul des limites, dans l'intégration d'un système d'équations différentielles; *Comptes-rendus de l'Académie des Sciences* 15, 1842 (public domain)
- Some icons used in this document are from the Oxygen icon set (http://oxygen-icons.org), licensed under the GNU Lesser General Public Licence v. 3.