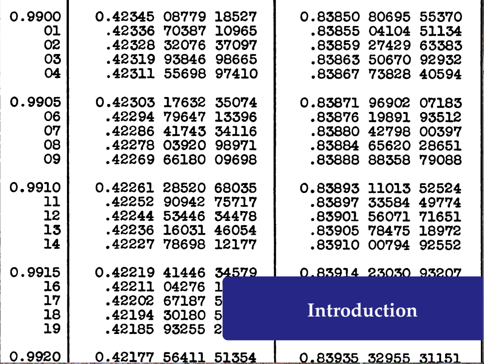
Around the Numerical Evaluation of D-Finite Functions

Marc Mezzarobba

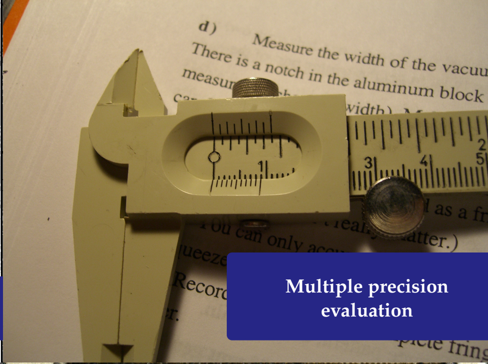
AriC Team – Inria

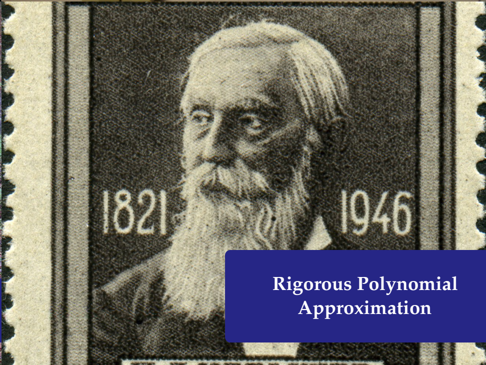
ENS DE LYON

CalSci department meeting, LIP6, Paris
February 19, 2013



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

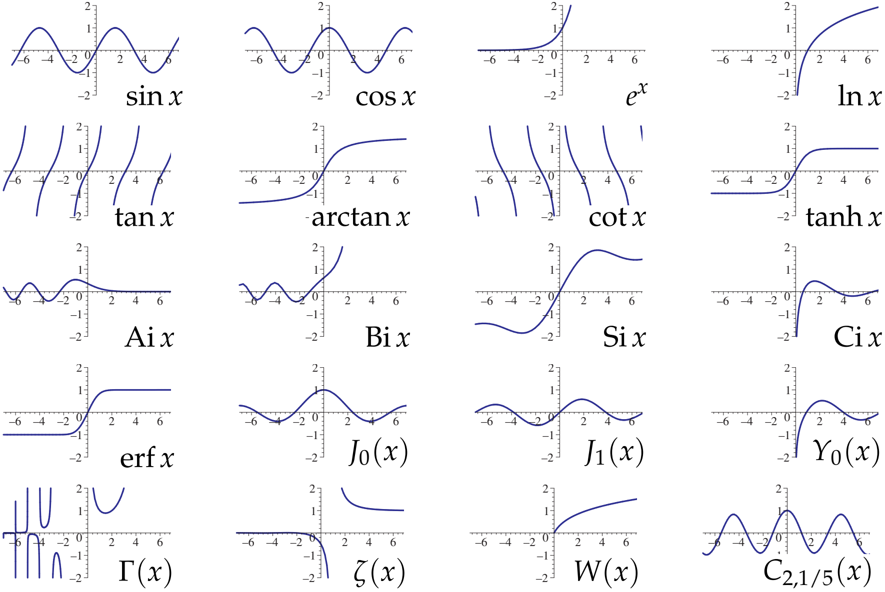




0.9900	0.42345 08779 18527	0.83850 80695 55370
01	.42336 70387 10965	.83855 04104 51134
02	.42328 32076 37097	.83859 27429 63383
03	.42319 93846 98665	.83863 50670 92932
04	.42311 55698 97410	.83867 73828 40594
0.9905	0.42303 17632 35074	0.83871 96902 07183
06	.42294 79647 13396	.83876 19891 93512
07	.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]
17	.42202 67187 5[illegible]	[illegible]
18	.42194 30180 5[illegible]	[illegible]
19	.42185 93255 2[illegible]	[illegible]
0.9920	0.42177 56411 51354	0.83935 32955 31151

Introduction

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

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

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:

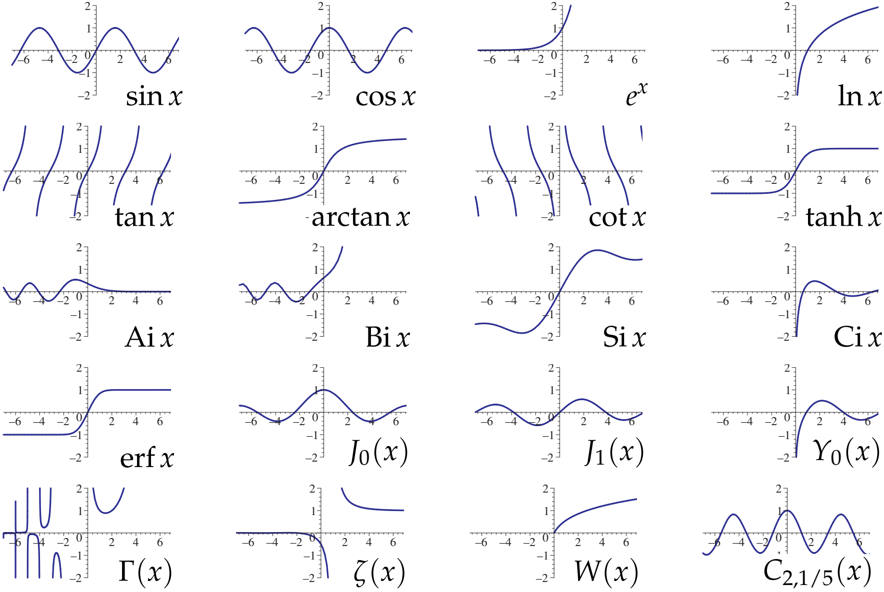
$$a_r(z)\, y^{(r)}(z) + \cdots + a_1(z)\, y'(z) + a_0(z)\, y(z) = 0, \quad a_j \in \mathbb{K}[z].$$

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

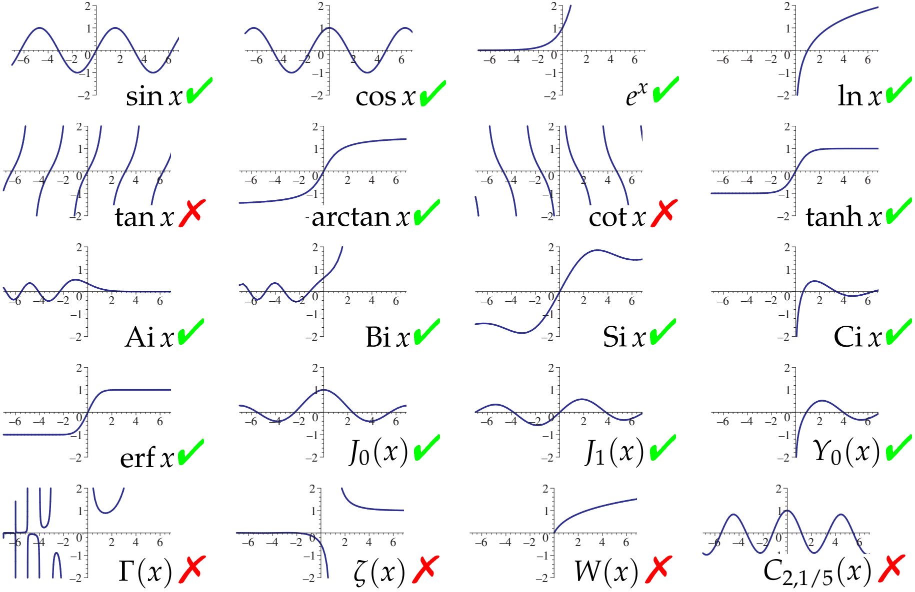
Example : $y(z) = K_0(z)$ (modified Bessel function)

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

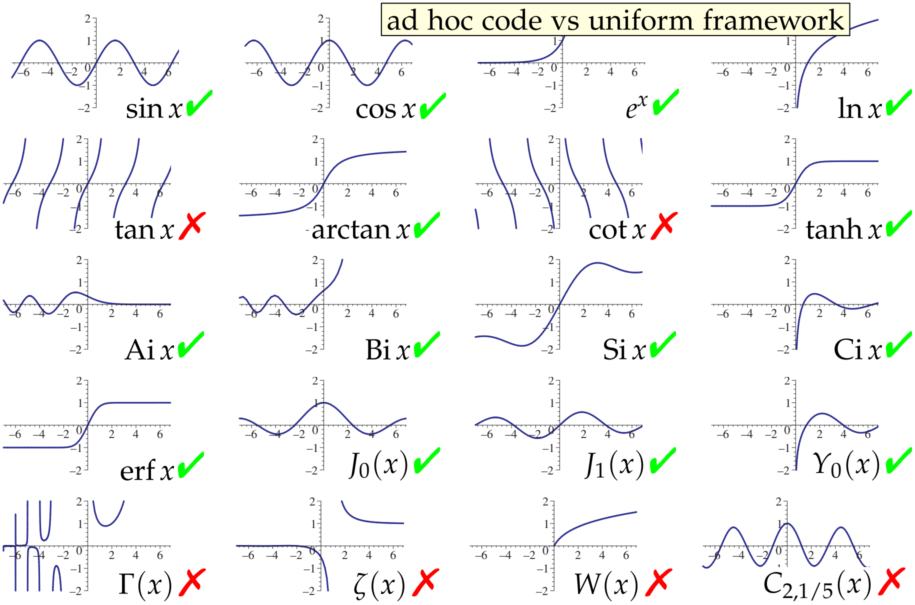
Elementary and Special Functions



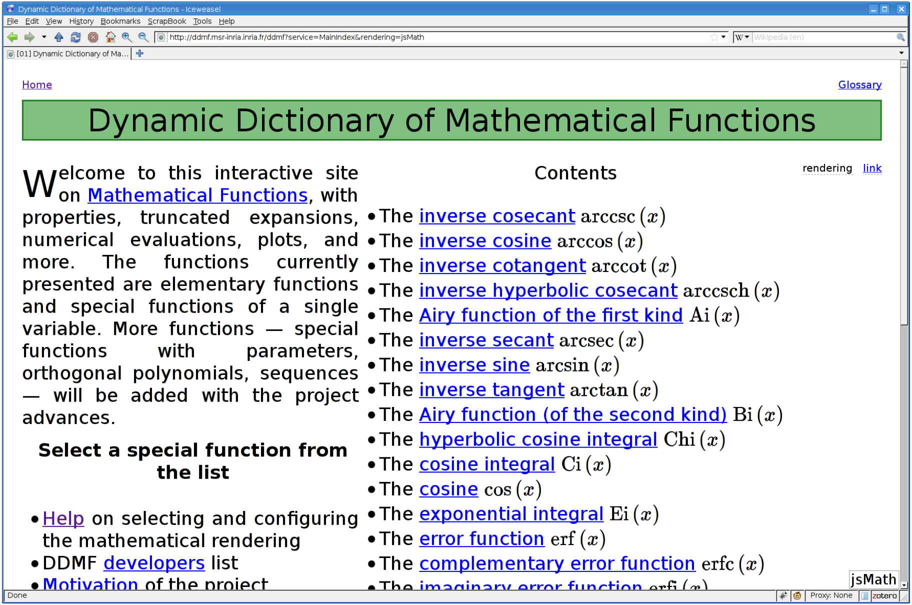
Elementary and Special Functions



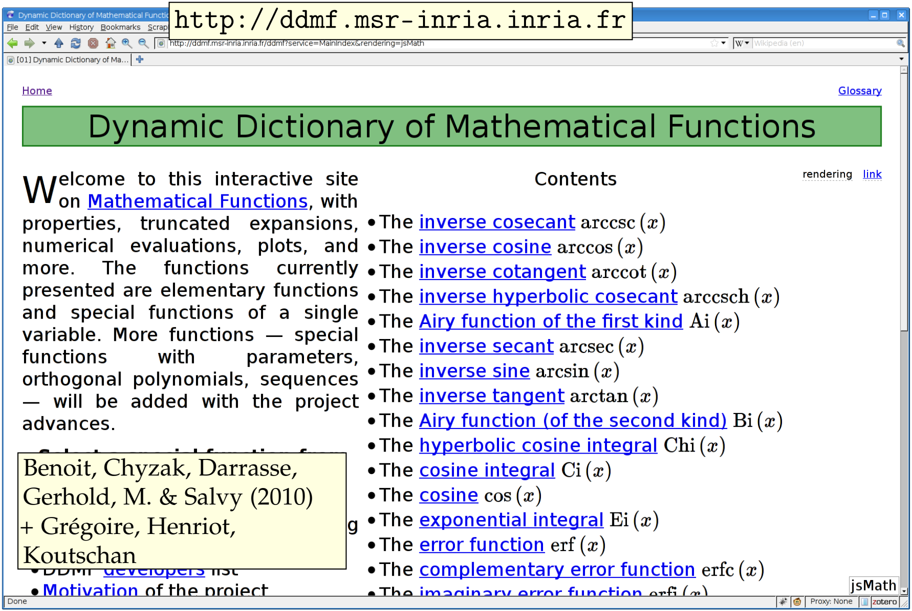
Elementary and Special Functions



A Dictionary of D-Finite Functions



A Dictionary of D-Finite Functions



A Dictionary of D-Finite Functions

Home Glossary

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

1. Differential equation

rendering link

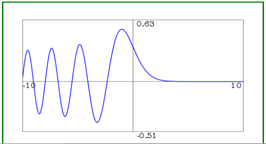
The function $\mathrm{Ai}(x)$ satisfies

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

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

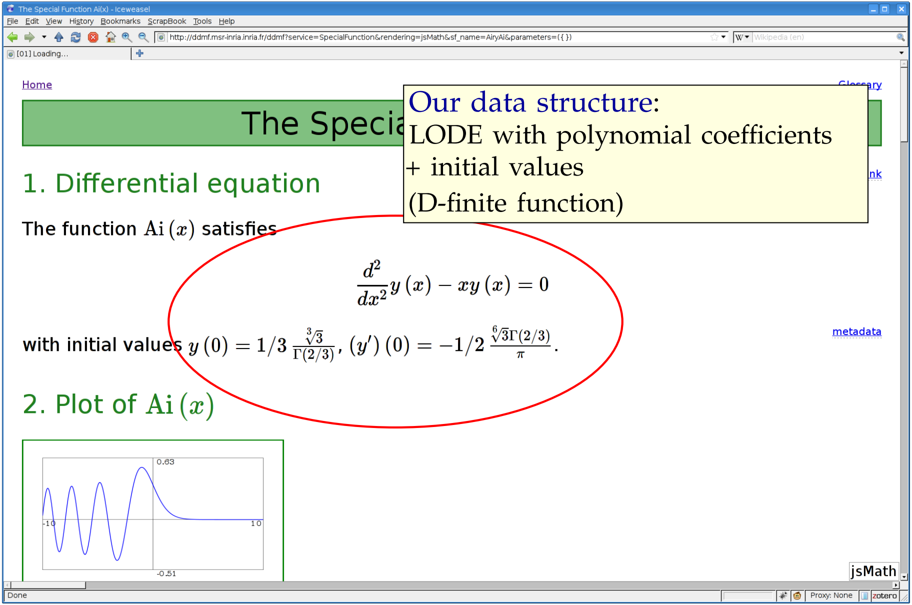
metadata

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



jsMath

A Dictionary of D-Finite Functions



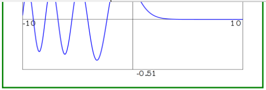
A Dictionary of D-Finite Functions

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

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

(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 = 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

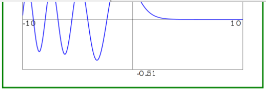
A Dictionary of D-Finite Functions

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

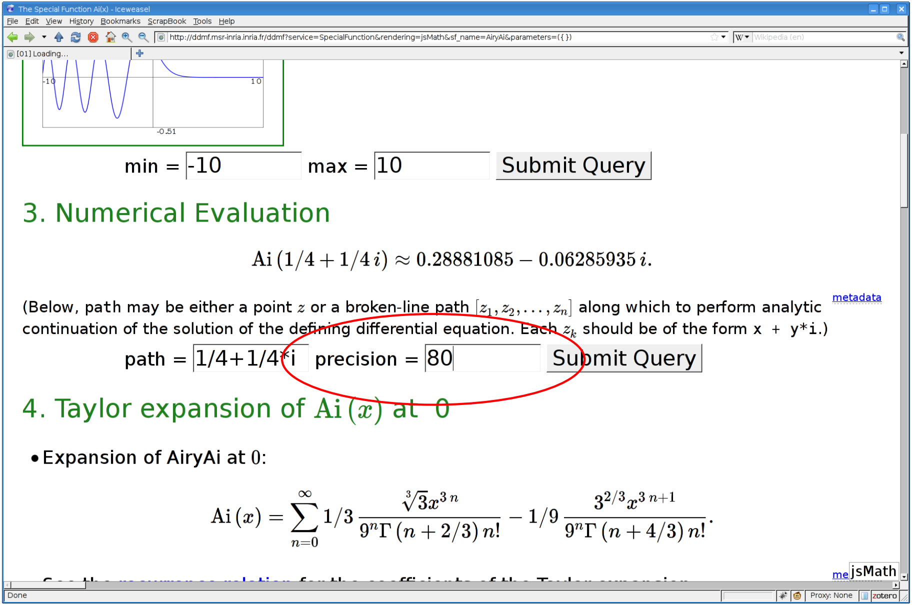
- Expansion of AiryAi at 0:

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

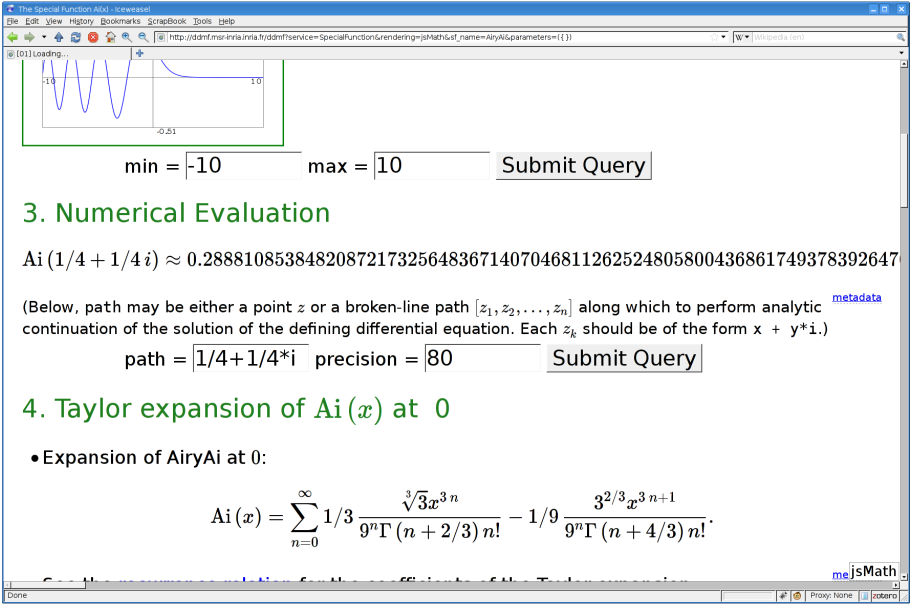
me jsMath

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...

-10 10 -0.51

min = -10 max = 10 Submit Query

3. Numerical Evaluation

$\mathrm{Ai}\,(1/4 + 1/4\,i) \approx 0.2888108538482087217325648367140704681126252480580043686174937839264 7$

(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:

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

See the recurrence relation for the coefficients of the Taylor expansion

me jsMath

Done Proxy: None zotero

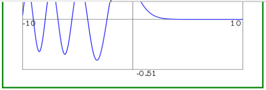
A Dictionary of D-Finite Functions

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

- computed from the ODE
- rigorous error bounds
- arbitrary precision

3. Numerical Evaluation

$$\mathrm{Ai}\,(1/4+1/4\,i) \approx 0.288810853848208721732564836714070468112625248058004368617493783926470$$

(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:

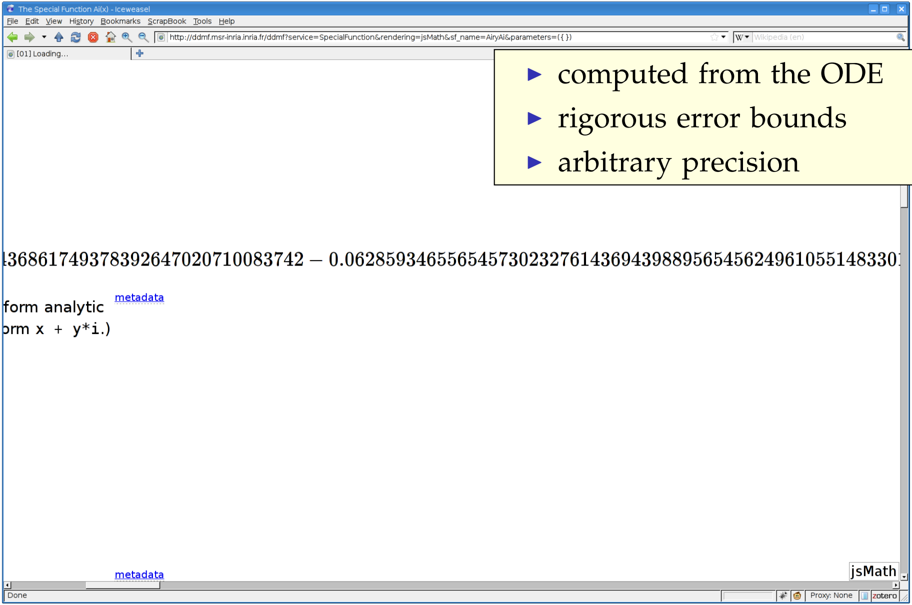
$$\mathrm{Ai}\,(x) = \sum_{n=0}^{\infty} 1/3\, \frac{\sqrt[3]{3}x^{3\,n}}{9^n\Gamma\,(n+2/3)\,n!} - 1/9\, \frac{3^{2/3}x^{3\,n+1}}{9^n\Gamma\,(n+4/3)\,n!}.$$

See the recurrence relation for the coefficients of the Taylor expansion

me jsMath

Done Proxy: None zotero

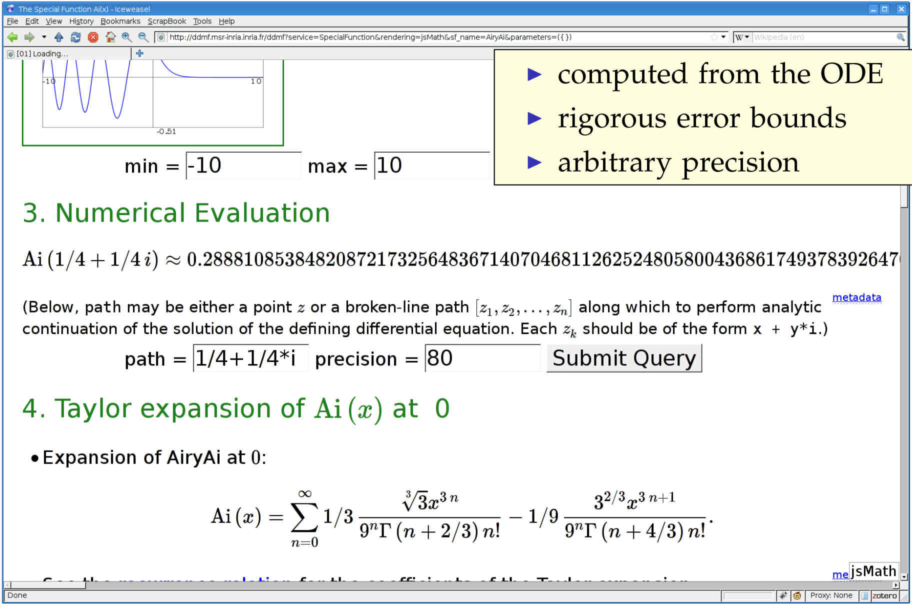
A Dictionary of D-Finite Functions



- computed from the ODE
- rigorous error bounds
- arbitrary precision

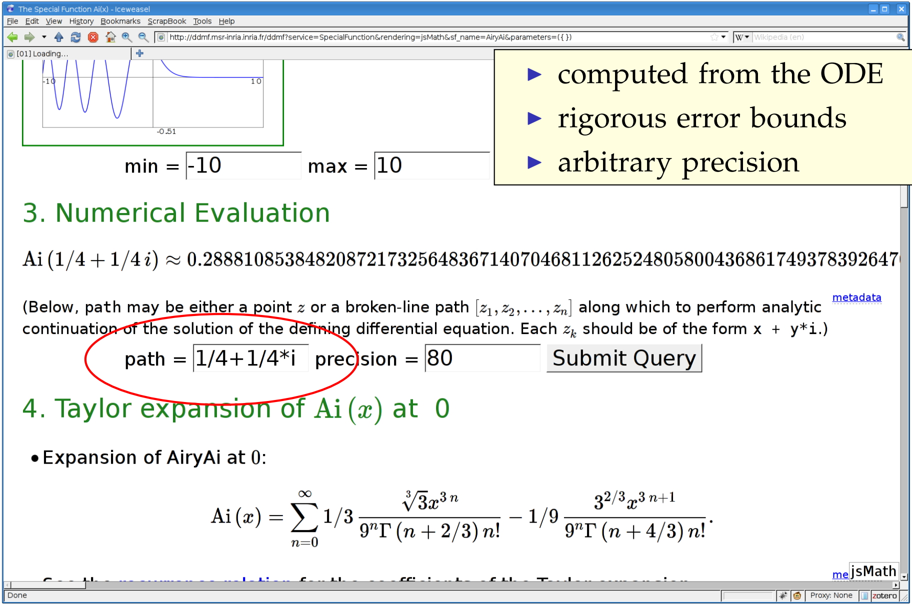
A Dictionary of D-Finite Functions

- computed from the ODE
- rigorous error bounds
- arbitrary precision



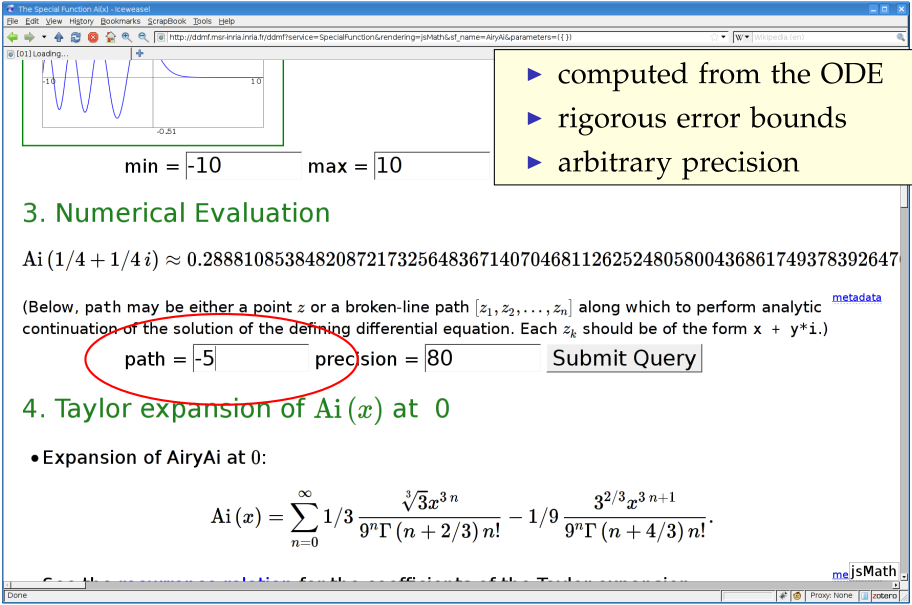
A Dictionary of D-Finite Functions

- computed from the ODE
- rigorous error bounds
- arbitrary precision



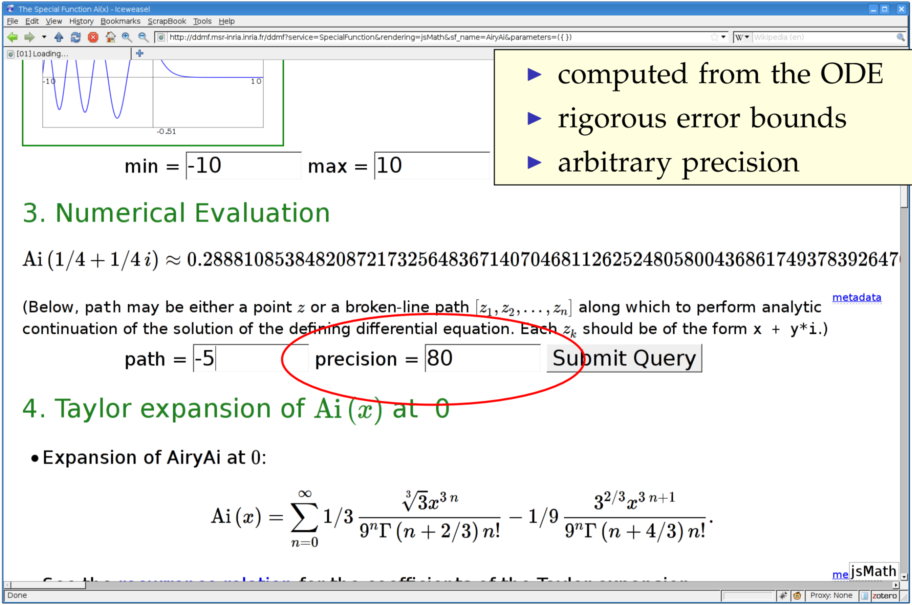
A Dictionary of D-Finite Functions

- computed from the ODE
- rigorous error bounds
- arbitrary precision



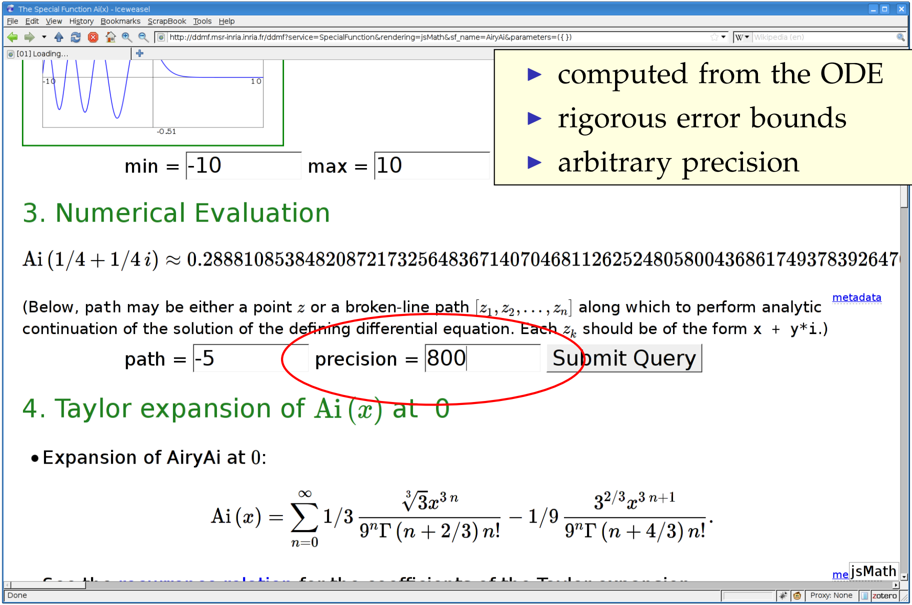
A Dictionary of D-Finite Functions

- computed from the ODE
- rigorous error bounds
- arbitrary precision



A Dictionary of D-Finite Functions

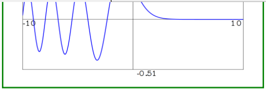
- computed from the ODE
- rigorous error bounds
- arbitrary precision



A Dictionary of D-Finite Functions

The Special Function Ai(x) - Iceweasel

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



min = -10 max = 10

- computed from the ODE
- rigorous error bounds
- arbitrary precision

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:

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

See the recurrence relation for the coefficients of the Taylor expansion

jsMath

Done Proxy: None zotero

NumGfun

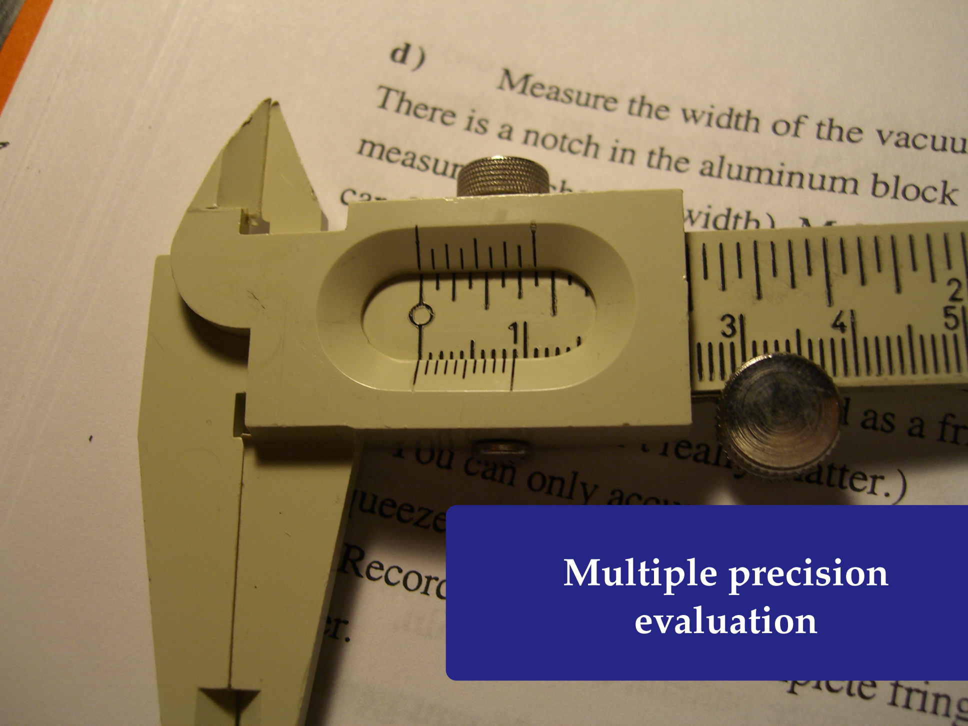
http://algo.inria.fr/libraries/ (GNU LGPL)

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.



Multiple precision evaluation

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};
```

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

```
>
```

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};
```

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

```
> 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};
```

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

```
> 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);
1.23715744756395253918007831405821000395447403052074724 9\
    773681223399104792726342791042603669170468682243266 93\
    220587400059578688690656372550637713781176348250035 48\
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    961701523808082462652309161587329644963237667773574 28\
    282143358101669038755863333203347465747570600605911 60\
    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);
1.237157447563952539180078314058210003954474030520747249\
    773681223399104792726342791042603669170468682243266693\
    22058740005957868869065637255063771378117634825003548\
    ...
    961701523808082462652309161587329644963237667773574428\
    282143358101669038755863333203347465747570600605911160\
    33361999970684428816250827723506800809
>
```

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);
4.67755852796689048164637161641413056565032356040992203\
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    89542201276207762696563032189351846152496641167932588\
    466046002397287307888 1
>
```

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);
4.6775585279668904816463716164141305656503235604099220037\
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    8954220127620776269656
    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);
```

0.04485557487767843131893308147593115486663

+ 0.01990489830212805305047897725810997882 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);
0.033253281257567506772459381920024394391065961347292863\
    1361178559307565437161078471985962090680571076277606\
    6599384479391829794197618862065053669108217914960590\
    3108048298855823993517550511176819489159174044677130\
    7473025189635972756153431009580734363927305651896233\
    9721759513884230988401642563243102957713043147210864\
    9548515476762402429734385158441412605623777191148968\
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    9793325825997236646657321960250165021813974778115734\
    7832262865574719581820528242814824080037691356145556\
    2959879449123182803958425643066993236588095610171972\
    3380613024394057453999112187785110527075237813842272\
    7617685959250804078177163720506043190222743767328690\
    7129257409846695090670592759003049446015009928821012\
    868701569
```

Some History

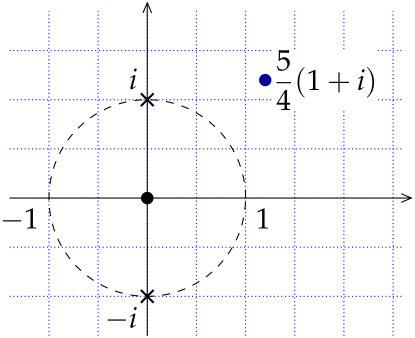
- Schroeppel (1972) – Special evaluation points
- Brent (1976) – Special case of exp (+ variants)
- Chudnovsky & Chudnovsky (1986-1988) – General method (incl. a sketch of the case of regular singular points)
- van der Hoeven (1999, 2001) – General algorithm with error bounds
- M. – Implementation, efficiency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

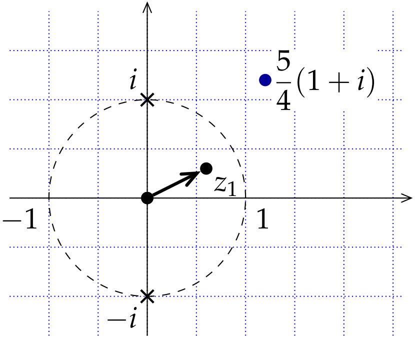
0 fast integer multiplication

1 binary splitting

2 analytic continuation

3 *bit burst*

2. Taylor series method for ODEs



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

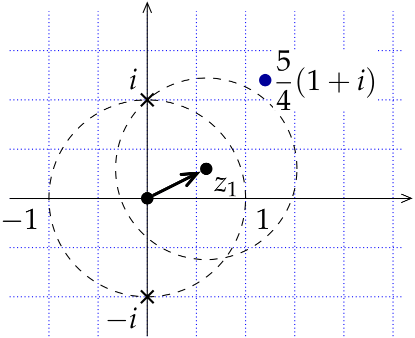
$$\begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix} = \begin{bmatrix} 1 & 0{,}570...+0{,}220...\,i \\ 0 & 0{,}728...-0{,}206...\,i \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

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



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

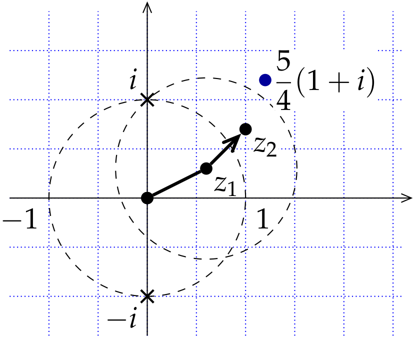
$$\begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix} = \begin{bmatrix} 1 & 0{,}570...+0{,}220...\,i \\ 0 & 0{,}728...-0{,}206...\,i \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

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



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

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

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

Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

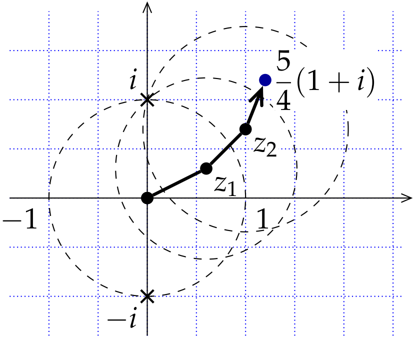
0 fast integer multiplication

1 binary splitting

2 analytic continuation

3 *bit burst*

2. Taylor series method for ODEs



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

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

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

$\ldots$

Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

0 fast integer multiplication

1 binary splitting

2 analytic continuation

3 *bit burst*

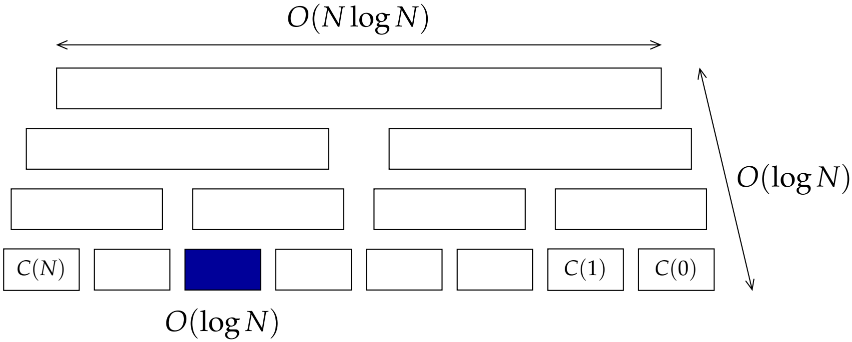
0. One can multiply two integers of $\leqslant n$ bits in $M(n) = O(n \log n \, 2^{O(\log^* n)})$ bit ops [Fürer 2007].

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!

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

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

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

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

bit operations.

Evaluation Algorithm [Chudnovsky & Chudnovsky 1988]

Main Ideas

0 fast integer multiplication

1 binary splitting

2 analytic continuation

3 *bit burst*

Theorem (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 (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 using $O(n)$ bits of memory.

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

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));
```

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

```
>
```

Regular Singular Points

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

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

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

Regular Singular Points

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

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

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

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

```
>
```

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));
```

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

$$+ (0.7853981634$$

```
>
```

Applications:

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

Bounds

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

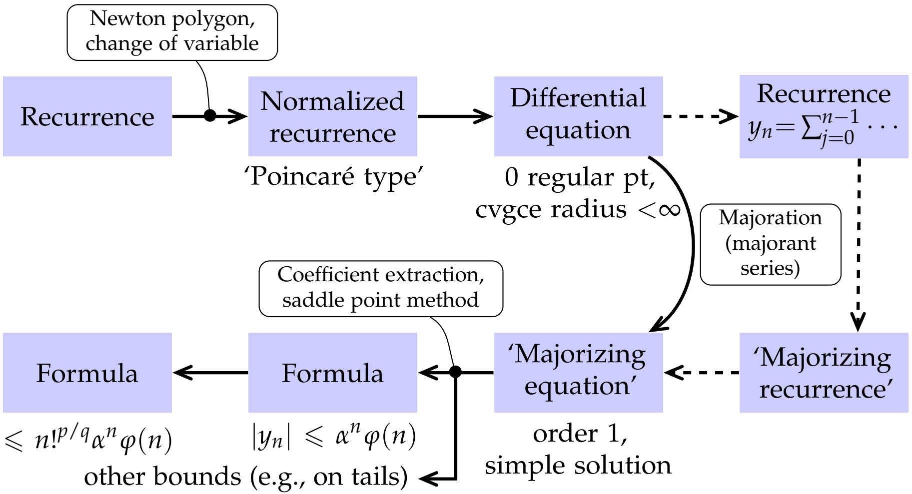
Symbolic Bounds

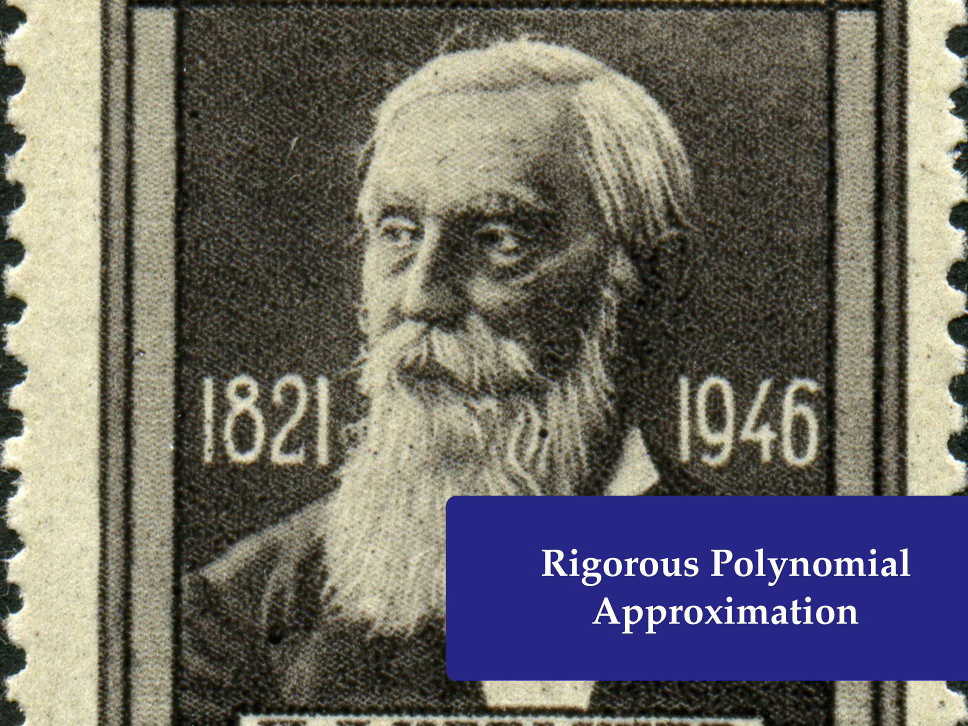
- Readable (as far as possible!)
- Asymptotically tight

Numeric Bounds

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

Outline of the Algorithm





Rigorous Polynomial Approximation

Repeated Evaluations

```
> deq := holexprtodiffeq(AiryAi(z), y(z)):
>
```

Repeated Evaluations

```
> deq := holexprtodiffeq(AiryAi(z), y(z)):
> myAi := diffeqtoproc(deq, y(z),
                        prec=12, disks=[[0,6]]):
```

Repeated Evaluations

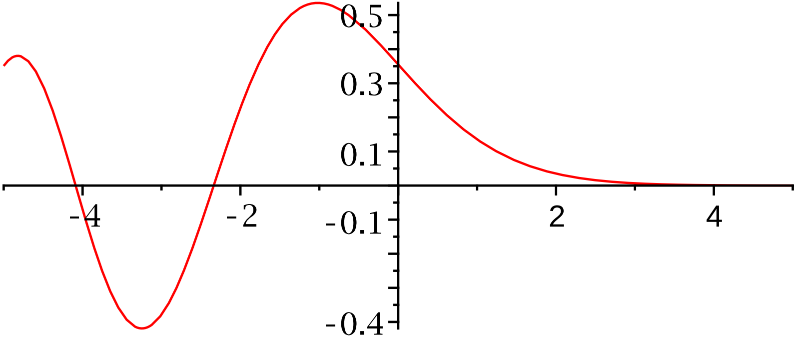
```
> deq := holexprtodiffeq(AiryAi(z), y(z)):
> myAi := diffeqtoproc(deq, y(z),
                      prec=12, disks=[[0,6]]):
>
```

Repeated Evaluations

```
> deq := holexprtodiffeq(AiryAi(z), y(z)):
> myAi := diffeqtoproc(deq, y(z),
                            prec=12, disks=[[0,6]]):
> plot(myAi, -5..5);
```

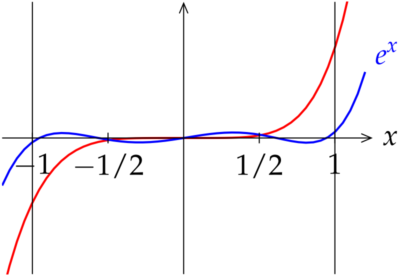
Repeated Evaluations

```
> deq := holexprtodiffeq(AiryAi(z), y(z)):
> myAi := diffeqtoproc(deq, y(z),
                          prec=12, disks=[[0,6]]):
> plot(myAi, -5..5);
```




```
>
```

Taylor Series vs. Chebyshev Series



$e^x - (1 + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4)$

$$e^x - (1{,}2660 T_0(x) + 1{,}1303 T_1(x) + 0{,}2714 T_2(x) + 0{,}0443 T_3(x) + 0{,}0054 T_4(x))$$

Quasi-Minimax Approximation

For any regular enough function f,

$$\|f - p_d\|_\infty \leqslant \left(\frac{4}{\pi^2}\log(d+1) + 4\right)\|f - p_d^*\|_\infty$$

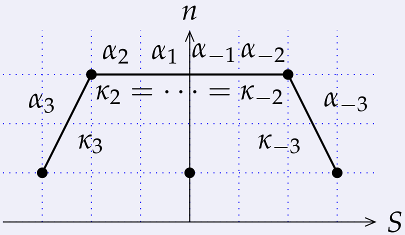
Previous Work

- Numerical computation of Chebyshev expansions
  - Lánczos (1938) – τ method
  - Clenshaw (1957) – backward iterative method à la Miller
- Recurrence relation
  - Fox & Parker (1968) – small orders, link with Clenshaw
  - Paszkowski (1975) – general case
  - Geddes (1977), Rebillard (1998), Benoit & Salvy (2009) – computer algebra
- Chebyshev expansions in Interval Analysis
  - Kaucher & Miranker (1984) – ultra-arithmetic
  - Brisebarre & Joldeş (2010) – ChebModels

D-finite Chebyshev Series

Obstacles

- Divergent solution sequences
- Initial values $\notin \mathbb{Q}$
- Error bounds



Our approach

1. Compute the coefficients by a variant of Clenshaw's method
2. Validate the output (enclosure + fixed-pt thm)

A. Benoit, M. Joldeş and M. Mezzarobba. Rigourous uniform approximation of D-finite functions using Chebyshev expansions. In preparation.

Computing the Coefficients

Example

$$y(x) = e^x = \sum_{n=-\infty}^{\infty} c_n\, T_n(x) \qquad\qquad c_{n+1} + 2n\, c_n - c_{n-1} = 0$$

$$\begin{aligned}
u_0 &\approx -4{,}40 \cdot 10^{81} \\
u_1 &\approx 1{,}96 \cdot 10^{81} \\
u_2 &\approx -4{,}72 \cdot 10^{80} \\
&\vdots \\
u_{50} &\approx 1{,}02 \cdot 10^{2} \\
u_{51} &= 1 \\
u_{52} &= 0
\end{aligned}
\quad \xrightarrow{\;c_n := u_n / S\;} \quad
\begin{aligned}
c_0 &\approx 1{,}27 \\
c_1 &\approx -5{,}65 \cdot 10^{-1} \\
c_2 &\approx 1{,}36 \cdot 10^{-1} \\
&\vdots \\
c_{50} &\approx -2{,}93 \cdot 10^{-80} \\
c_{51} &\approx -2{,}88 \cdot 10^{-82}
\end{aligned}$$

$$S = \sum_{n=-50}^{50} u_n\, T_n(0) \approx -3{,}48 \cdot 10^{81}$$

Computing the Coefficients

Linear complexity wrt starting index N.

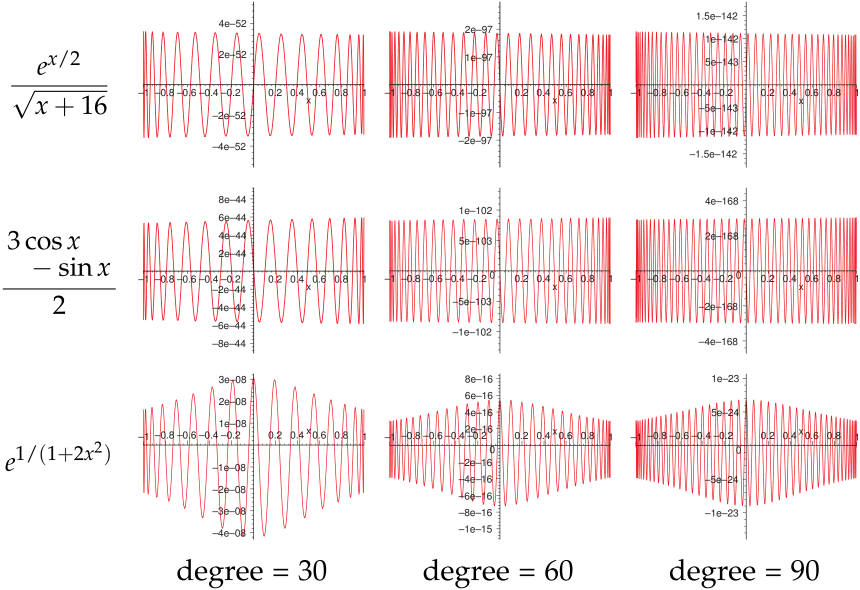
Theorem

The (method) error on the computed coefficients, i.e.,

$$\max_{n=0}^{N} \left| c_n^{[N]} - c_n \right|$$

decreases exponentially as $N \to \infty$.

Computed Polynomials



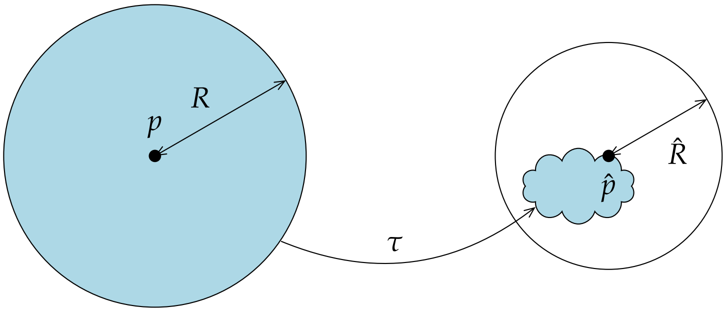
Validation Step

Input A differential operator, initial values, a polynomial p of degree d, a precision ε

Output R such that $\|y - p\|_\infty \leqslant R = O\big(\sqrt{d}\,(\|y - p\|_\infty + \varepsilon)\big)$

$$\tau(y) := \left(x \mapsto y_0 + \int_0^x \frac{a(t)}{b(t)}\, y(t)\,\mathrm{d}t\right)$$

$$\|\tau(f) - \tau(g)\|_\infty \leqslant \gamma\, \|f - g\|_\infty \qquad \gamma < 1$$



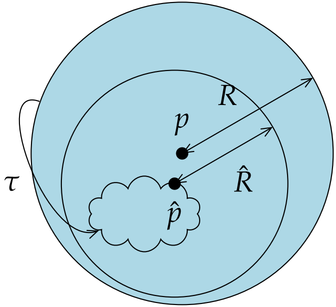
Validation Step

Input A differential operator, initial values, a polynomial p of degree d, a precision ε

Output R such that $\|y-p\|_\infty \leqslant R = O\big(\sqrt{d}\,(\|y-p\|_\infty + \varepsilon)\big)$

$$\tau(y) := \left(x \mapsto y_0 + \int_0^x \frac{a(t)}{b(t)}\, y(t)\,\mathrm{d}t\right)$$

$$\|\tau(f)-\tau(g)\|_\infty \leqslant \gamma\,\|f-g\|_\infty \qquad \gamma < 1$$



$$\|p-\hat{p}\|_\infty + \hat{R} \leqslant R$$

Validation Step

Input A differential operator, initial values,
a polynomial p of degree d, a precision ε

Output R such that $\|y - p\|_\infty \leqslant R = O\left(\sqrt{d}\,(\|y - p\|_\infty + \varepsilon)\right)$

$$\tau(y) := \left(x \mapsto y_0 + \int_0^x \frac{a(t)}{b(t)}\, y(t)\, \mathrm{d}t\right) \qquad \|\tau(f) - \tau(g)\|_\infty \leqslant \gamma \, \|f - g\|_\infty$$

$$\gamma < 1$$

Algorithm

- Choose i large enough
- Compute $p_i \approx \tau^i(p)$
- Return $R \geqslant \dfrac{\|p - p_i\|_\infty + (\text{errors})}{1 - \gamma_i}$

Validation Step

Input A differential operator, initial values, a polynomial p of degree d, a precision ε

Output R such that $\|y - p\|_\infty \leqslant R = O\big(\sqrt{d}\,(\|y - p\|_\infty + \varepsilon)\big)$

$$\tau(y) := \left(x \mapsto y_0 + \int_0^x \frac{a(t)}{b(t)}\, y(t)\, \mathrm{d}t\right) \qquad \|\tau(f) - \tau(g)\|_\infty \leqslant \gamma\, \|f - g\|_\infty$$

$$\gamma < 1$$

Algorithm

- Choose i large enough
- Compute $p_i \approx \tau^i(p)$ — O(d) ops
- Return $R \geqslant \dfrac{\|p - p_i\|_\infty + (\text{errors})}{1 - \gamma_i}$ — O(d) ops

Quality of the Resulting Bounds

$\dfrac{e^{x/2}}{\sqrt{x+16}}$	4,8	0,58	0,57
$\dfrac{3\cos x - \sin x}{2}$	3,1	3,7	4,1
$e^{1/(1+2x^2)}$	0,57	0,56	0,56
	degree = 30	degree = 60	degree = 90

Quality: $\log_{10} \dfrac{B}{\|y-p\|_\infty}$

Conclusion

This Talk

- D-finite functions, DDMF
  compute everything we can starting from LODE + ini. cond.
- Multiple precision analytic continuation
  general – rigorous – fully automatic – fast – code available
- Tight bounds
  symbolic + numeric – Cauchy majorant method
- Rigorous polynomial approximations
  backward recurrence à la Miller/Clenshaw + fixed point theorem

Other Work

- Roadmaps of real algebraic varieties
  internship with M. Safey El Din, 2005
- “Reconditioning” (cancellation reduction) of entire series
  new algo for $\mathrm{Ai}(x)$, with S. Chevillard, 2013
- Decimal floating-point arithmetic
  decimal64 $\lesseqgtr$? binary64, with N. Brisebarre, C. Lauter, and J.-M. Muller, 2013

Conclusion

This Talk

- D-finite functions, DDMF
  compute everything we can starting from LODE + ini. cond.
- Multiple precision analytic continuation
  general – rigorous – fully automatic – fast – code available
- Tight bounds
  symbolic + numeric – Cauchy majorant method
- Rigorous polynomial approximations
  backward recurrence à la Miller/Clenshaw + fixed point theorem

Other Work

- Roadmaps of real algebraic varieties
  internship with M. Safey El Din, 2005
- "Reconditioning" (cancellation reduction) of entire series
  new algo for $\mathrm{Ai}(x)$, with S. Chevillard, 2013
- Decimal floating-point arithmetic
  decimal64 $\lesseqgtr$? binary64, with N. Brisebarre, C. Lauter, and J.-M. Muller, 2013

Conclusion

This Talk

- D-finite functions, DDMF
  compute everything we can starting from LODE + ini. cond.
- Multiple precision analytic continuation
  general – rigorous – fully automatic – fast – code available
- Tight bounds
  symbolic + numeric – Cauchy majorant method
- Rigorous polynomial approximations
  backward recurrence à la Miller/Clenshaw + fixed point theorem

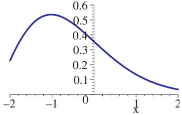
Other Work

- Roadmaps of real algebraic varieties
  internship with M. Safey El Din, 2005
- "Reconditioning" (cancellation reduction) of entire series
  new algo for $\mathrm{Ai}(x)$, with S. Chevillard, 2013
- Decimal floating-point arithmetic
  decimal64 $\lesseqgtr$? binary64, with N. Brisebarre, C. Lauter, and J.-M. Muller, 2013

Interests and Projects

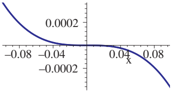
Code generation for special functions

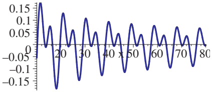
- Make the function part of the input
  e.g., via a differential equation
- Evaluation near singularities/∞
  non-polynomial approximations
- Towards formally proven implementations



Symbolic-numeric algorithms for ODEs

- "Beyond the DDMF"
  automatic study of D-finite functions
- Applications of eff. analytic continuation
  resummation of dvgt power series, diff. Galois
- Link with the above





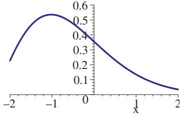
Mathematical software

- Sage, Mathemagix

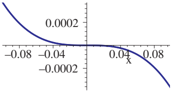
Interests and Projects

Code generation for special functions

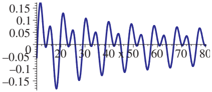
- Make the function part of the input
  e.g., via a differential equation
- Evaluation near singularities/∞
  non-polynomial approximations
- Towards formally proven implementations



Symbolic-numeric algorithms for ODEs



- "Beyond the DDMF"
  automatic study of D-finite functions
- Applications of eff. analytic continuation
  resummation of dvgt power series, diff. Galois
- Link with the above



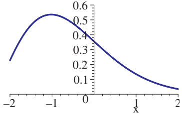
Mathematical software

- Sage, Mathemagix

Interests and Projects

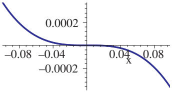
Code generation for special functions

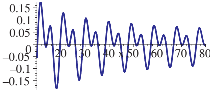
- Make the function part of the input
  e.g., via a differential equation
- Evaluation near singularities/∞
  non-polynomial approximations
- Towards formally proven implementations



Symbolic-numeric algorithms for ODEs

- "Beyond the DDMF"
  automatic study of D-finite functions
- Applications of eff. analytic continuation
  resummation of dvgt power series, diff. Galois
- Link with the above





Mathematical software

- Sage, Mathemagix

Credits

- *Tables of the error function and its derivative*. US National Bureau of Standards, 1954 (public domain)
- `http://commons.wikimedia.org/wiki/File:Calipers_in_physics_lab.jpg` (by User:Falcorian, Creative Commons Attribution-Share Alike)
- `http://commons.wikimedia.org/wiki/File:Paseky,_Havírna,_závora_u_domu_224.jpg` (by User:cs:ŠJů, Creative Commons Attribution-Share Alike)
- `http://en.wikipedia.org/wiki/File:Stamp_of_USSR_1047.jpg` (by User:Butko, taken from `http://kolekzioner.net/modules/smartsection/item.php?itemid=211`, 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.