NumGfun

A Package for Numerical and Analytic Computation with D-finite Functions

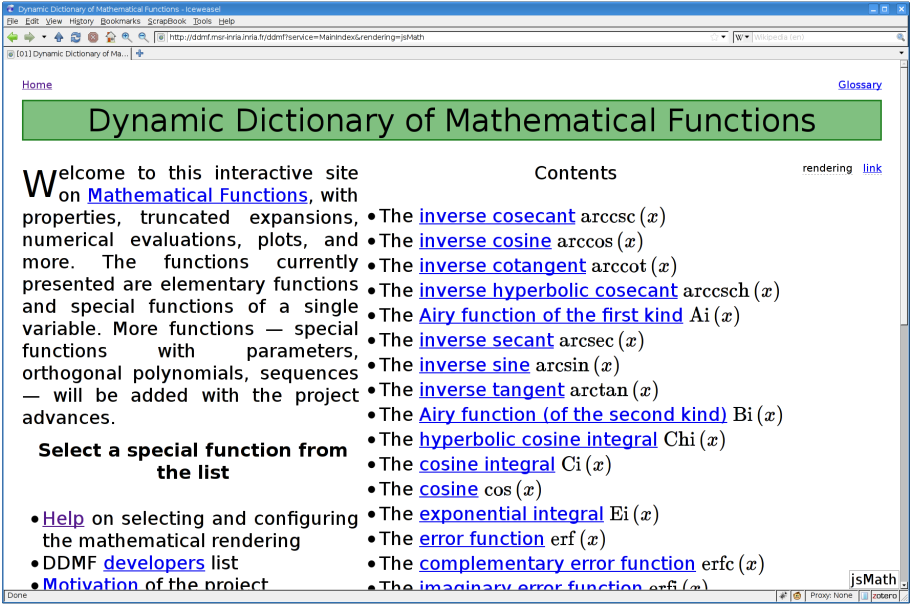
Marc Mezzarobba

Project-Team Algorithms
INRIA Paris-Rocquencourt, France

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA
centre de recherche PARIS - ROCQUENCOURT

ISSAC 2010

Motivation



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

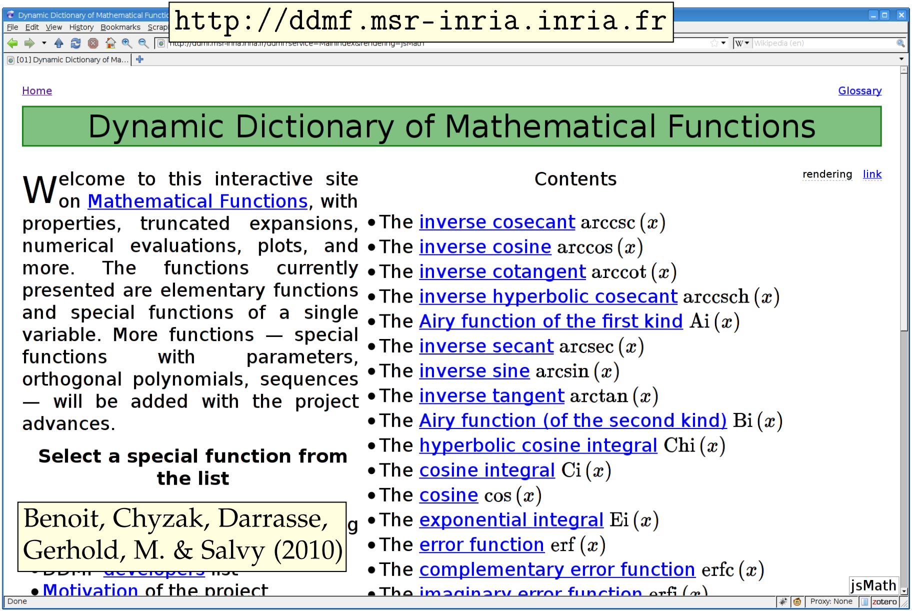
- The inverse cosecant $\operatorname{arccsc}(x)$
- The inverse cosine $\arccos(x)$
- The inverse cotangent $\operatorname{arccot}(x)$
- The inverse hyperbolic cosecant $\operatorname{arccsch}(x)$
- The Airy function of the first kind $\operatorname{Ai}(x)$
- The inverse secant $\operatorname{arcsec}(x)$
- 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

Motivation

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



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

Motivation

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 $\mathrm{Ai}\,(x)$

rendering link

1. Differential equation

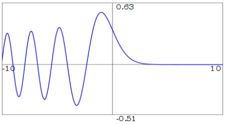
The function $\mathrm{Ai}\,(x)$ satisfies

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

metadata

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

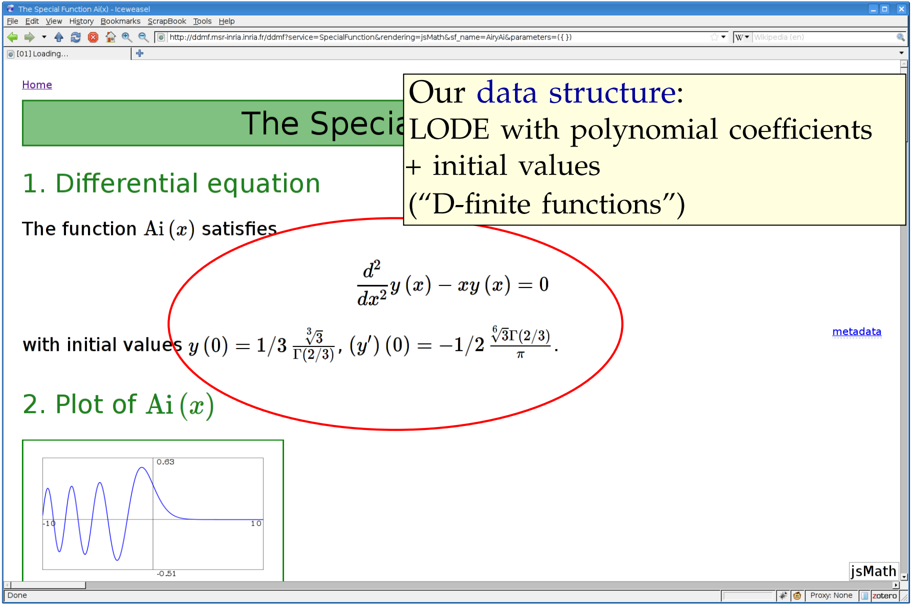
2. Plot of $\mathrm{Ai}\,(x)$



jsMath

Done | Proxy: None | zotero

Motivation



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

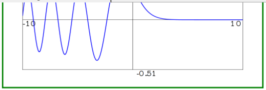
Motivation

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

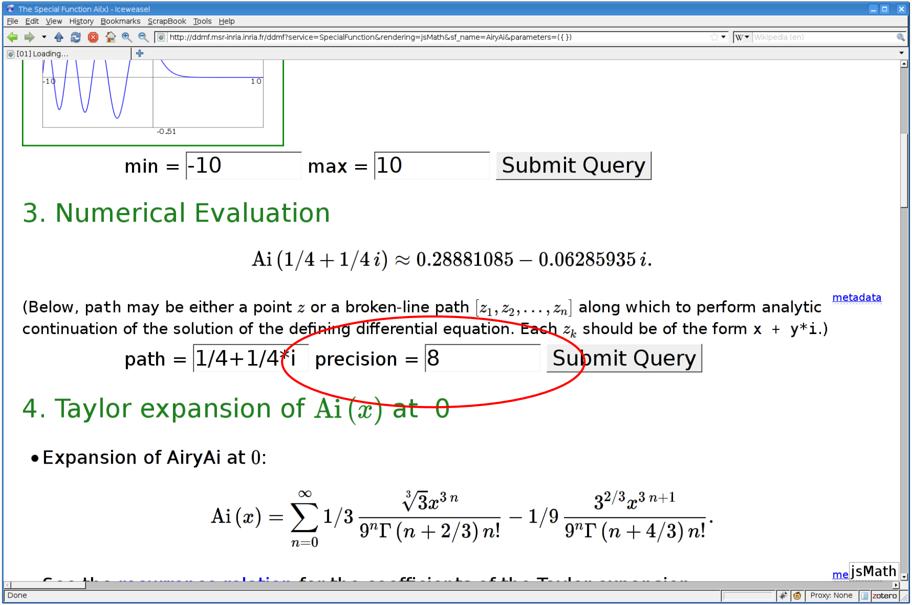
- Expansion of AiryAi at 0:

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

me jsMath

Done Proxy: None zotero

Motivation


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:

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

jsMath

Done Proxy: None zotero

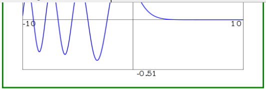
Motivation

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 = 80 Submit Query

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

- Expansion of AiryAi at 0:

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

jsMath

Done

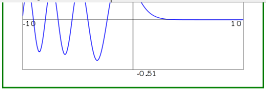
Proxy: None zotero

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:

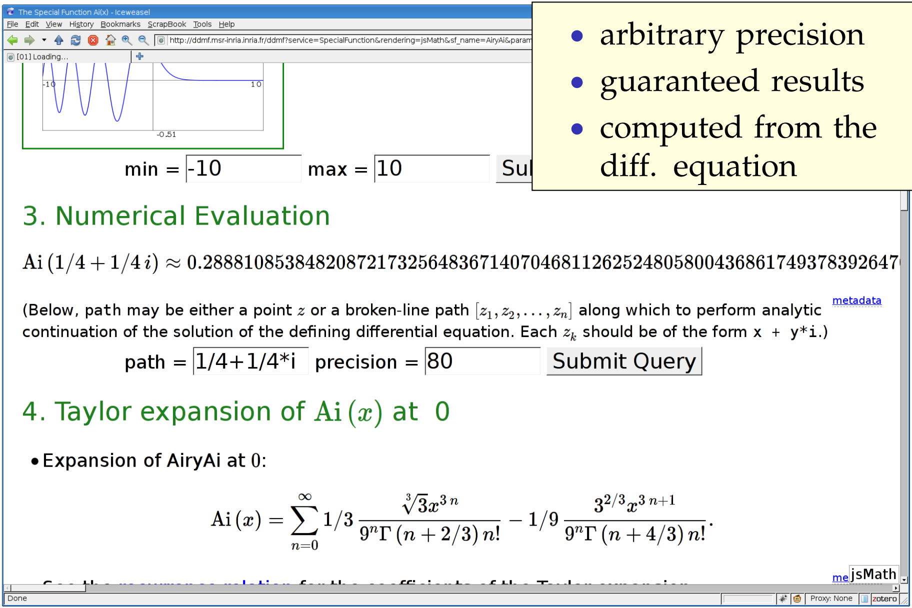
$$\mathrm{Ai}\,(x) = \sum_{n=0}^{\infty} 1/3\,\frac{\sqrt[3]{3}x^{3\,n}}{9^n\Gamma\,(n+2/3)\,n!} - 1/9\,\frac{3^{2/3}x^{3\,n+1}}{9^n\Gamma\,(n+4/3)\,n!}.$$

me jsMath

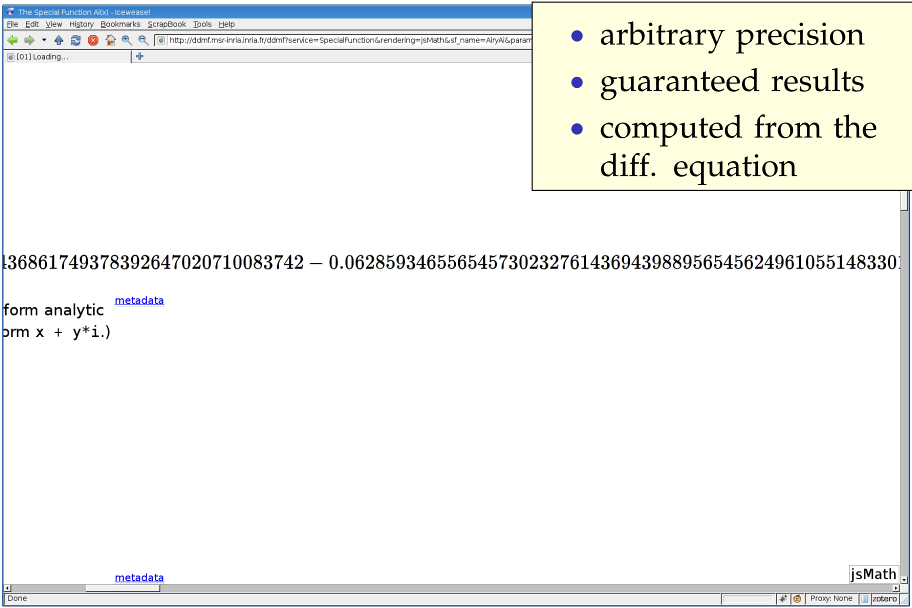
Done Proxy: None zotero

Motivation

- arbitrary precision
- guaranteed results
- computed from the diff. equation



Motivation



- arbitrary precision
- guaranteed results
- computed from the diff. equation

Motivation

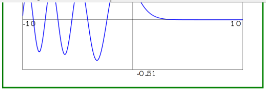
- arbitrary precision
- guaranteed results
- computed from the diff. equation

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&param

[01] Loading...



min = -10 max = 10 Su

3. Numerical Evaluation

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

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 = 80 Submit Query

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

- Expansion of AiryAi at 0:

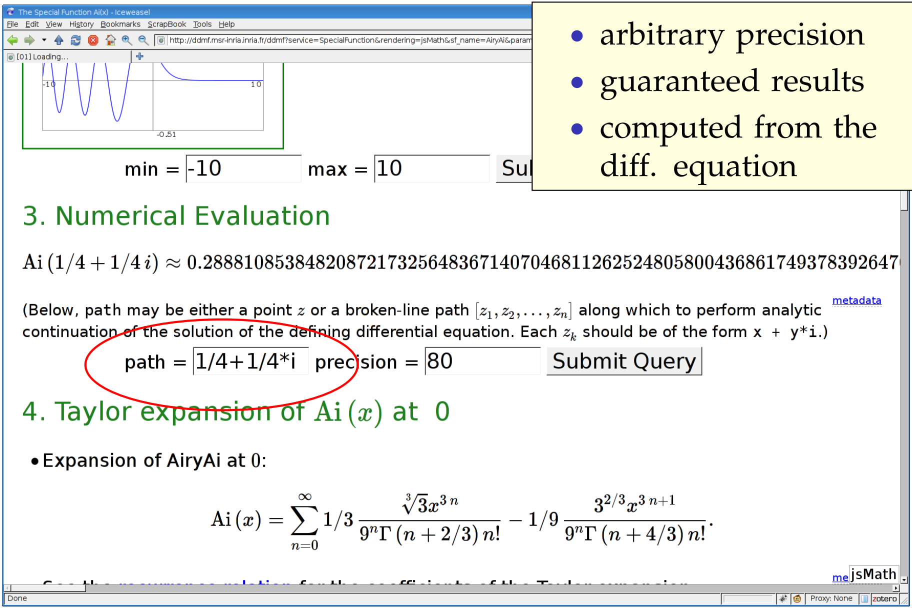
$$\mathrm{Ai}\,(x) = \sum_{n=0}^{\infty} 1/3\,\frac{\sqrt[3]{3}x^{3\,n}}{9^n\Gamma\,(n+2/3)\,n!} - 1/9\,\frac{3^{2/3}x^{3\,n+1}}{9^n\Gamma\,(n+4/3)\,n!}.$$

me jsMath

Done Proxy: None zotero

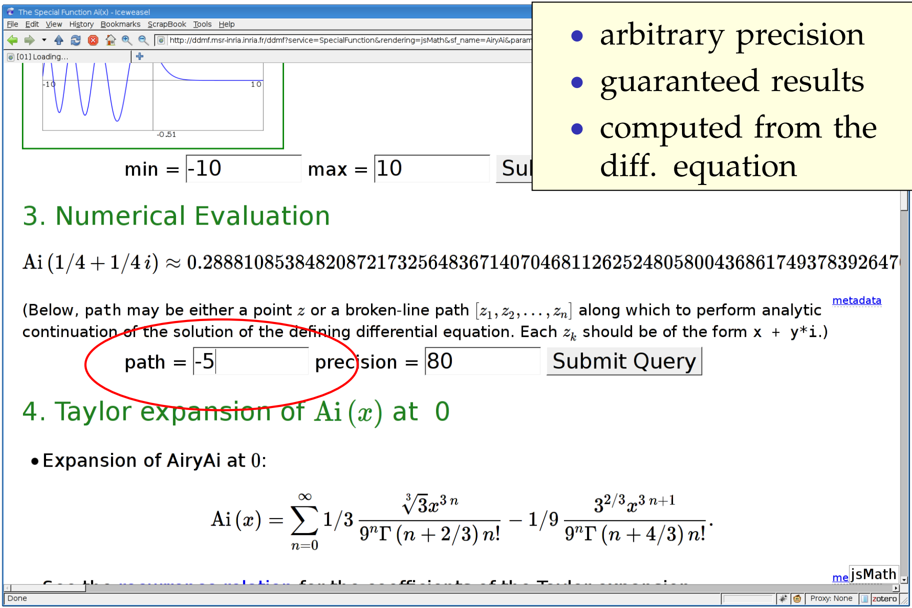
Motivation

- arbitrary precision
- guaranteed results
- computed from the diff. equation

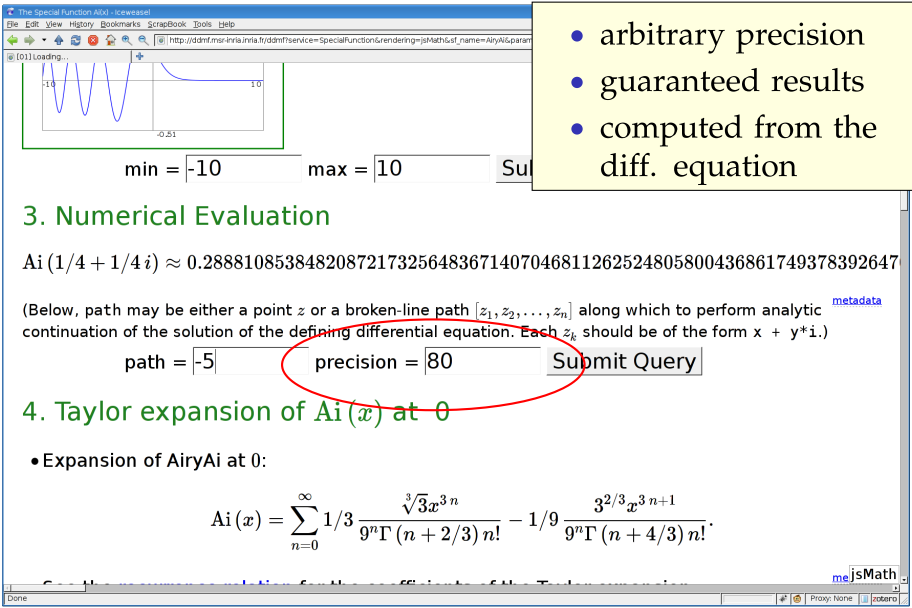


Motivation

- arbitrary precision
- guaranteed results
- computed from the diff. equation

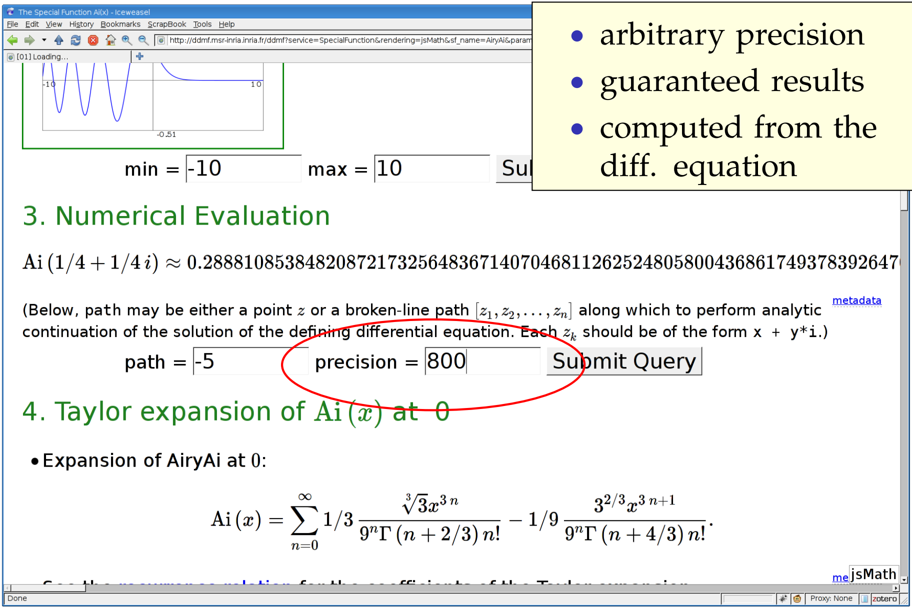


Motivation



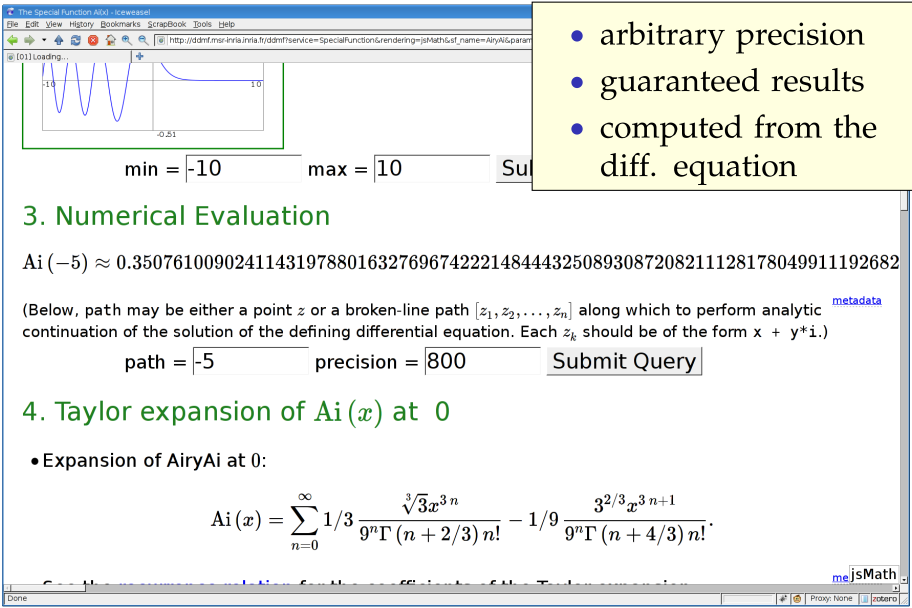
- arbitrary precision
- guaranteed results
- computed from the diff. equation

Motivation



- arbitrary precision
- guaranteed results
- computed from the diff. equation

Motivation



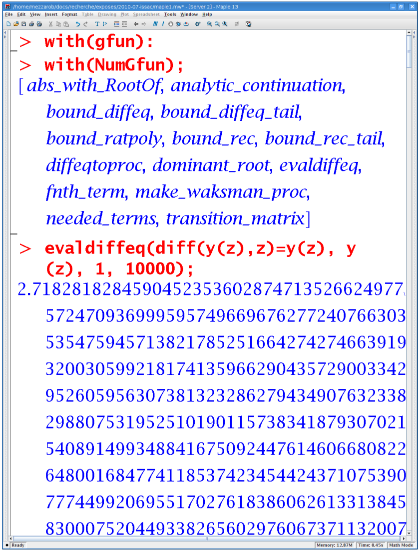
- arbitrary precision
- guaranteed results
- computed from the diff. equation

Previous Work

- Schroeppel (1972) – Evaluation at special points
- Brent (1976) – Narrow class of functions (at any point)
- Chudnovsky & Chudnovsky (1988) – General method
- van der Hoeven (1999) – Complete algorithm with error bounds
- M. & Salvy (2009) – Improved bounds

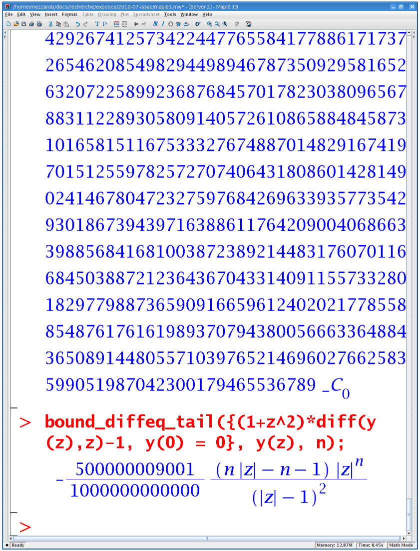
No general implementation so far

This Talk



- NumGfun ≙ Maple package behind the "analytic" features of DDMF
- Numerical evaluation
  First general implementation of the Chudnovsky's algorithm
  (with various improvements)
- Symbolic bounds

This Talk



- NumGfun ≙ Maple package behind the "analytic" features of DDMF
- Numerical evaluation
  First general implementation of the Chudnovsky's algorithm
  (with various improvements)
- Symbolic bounds

A Less Common Special Function

The Double Confluent Heun Function

```
> diffeq := collect({diff(y(z),z,z)-(-2*
  z^5+4*z^3+z^4*a-2*z-a)*(diff(y(z),z))/(
  (z-1)^3*(z+1)^3)-(-z^2*b+(-c-2*a)*z-d)*y
  (z)/((z-1)^3*(z+1)^3),y(0)=1,D(y)(0)=0},
  diff,factor);
```

A Less Common Special Function

The Double Confluent Heun Function

```
> diffeq := collect({diff(y(z),z,z)-(-2*
z^5+4*z^3+z^4*a-2*z-a)*(diff(y(z),z))/(
(z-1)^3*(z+1)^3)-(-z^2*b+(-c-2*a)*z-d)*y
(z)/((z-1)^3*(z+1)^3),y(0)=1,D(y)(0)=0},
diff,factor);
```

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

```
>
```

A Less Common Special Function

The Double Confluent Heun Function

```
> diffeq := collect({diff(y(z),z,z)-(-2*
z^5+4*z^3+z^4*a-2*z-a)*(diff(y(z),z))/(
(z-1)^3*(z+1)^3)-(-z^2*b+(-c-2*a)*z-d)*y
(z)/((z-1)^3*(z+1)^3),y(0)=1,D(y)(0)=0},
diff,factor);
```

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

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

```
> diffeq := collect({diff(y(z),z,z)-(-2*
z^5+4*z^3+z^4*a-2*z-a)*(diff(y(z),z))/(
(z-1)^3*(z+1)^3)-(-z^2*b+(-c-2*a)*z-d)*y
(z)/((z-1)^3*(z+1)^3),y(0)=1,D(y)(0)=0},
diff,factor);
```

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

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

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

```
>
```

Accuracy Issues

More General Code Often Means Less Bugs

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

Accuracy Issues

More General Code Often Means Less Bugs

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

Accuracy Issues

More General Code Often Means Less Bugs

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

Accuracy Issues

More General Code Often Means Less Bugs

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

Accuracy Issues

More General Code Often Means Less Bugs

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

Accuracy Issues

More General Code Often Means Less Bugs

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

Accuracy Issues

More General Code Often Means Less Bugs

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

Accuracy Issues

More General Code Often Means Less Bugs

```
> 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

More General Code Often Means Less Bugs

```
> 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

More General Code Often Means Less Bugs

```
> 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);
1.2371574475639525391800783140582100039544740305207472 49\
    773681223399104792726342791042603669170468682243266 93\
    220587400059578688690656372550637713781176348250035 48\
. . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    961701523808082462652309161587329644963237667773574 28\
    282143358101669038755863333203347465747570600605911 60\
    33361999970684428816250827723506800809
>
```

Accuracy Issues

Approaching Singularities

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

Accuracy Issues

Approaching Singularities

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

2.695836763

```
>
```

Accuracy Issues

Approaching Singularities

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

Accuracy Issues

Approaching Singularities

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

Accuracy Issues

Approaching Singularities

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

Accuracy Issues

Approaching Singularities

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

Accuracy Issues

Approaching Singularities

```
> 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
>
```

Accuracy Issues

Approaching Singularities

```
> 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);
```

Accuracy Issues

Approaching Singularities

```
> 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
>
```

Accuracy Issues

Approaching Singularities

```
> 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);
```

Accuracy Issues

Approaching Singularities

```
> 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. . .)

Accuracy Issues

Approaching Singularities

```
> 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);
4.6775585279668904816463716164141305656503235604099220 37\
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    8954220127620776269656303218935184615249664116793 2588\
    4660460023972873078881
>
```

Accuracy Issues

Approaching Singularities

```
> 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\
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    895422012762077626965630321893518461524966411679325 88\
    466046002397287307888 1
>
```

No numerical instability issues

A “Random” Example

```
> random_diffeq := proc(ord, d)
    uses RandomTools; local myrat;
    myrat := 'rational'('denominator'=60);
    { add(Generate('polynom'(myrat, z,
        'degree'=d)) * diff(y(z), [z$k]),
        k=0..ord),
      seq((D@@k)(y)(0) = Generate(myrat),
        k=0..ord-1) };
  end proc:
```

A "Random" Example

```
> random_diffeq := proc(ord, d)
    uses RandomTools; local myrat;
    myrat := 'rational'('denominator'=60);
    { add(Generate('polynom'(myrat, z,
        'degree'=d)) * diff(y(z), [z$k]),
        k=0..ord),
      seq((D@@k)(y)(0) = Generate(myrat),
        k=0..ord-1) };
  end proc:
> diffeq := random_diffeq(3, 2);
```

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);
```

$$0.04485557487767843131893308147593115486663 + 0.01990489830212805305047897725810997882 82 \, I$$

High Precision

It Scales!

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

High Precision

It Scales!

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

(29 min later…)

High Precision

It Scales!

```
> evaldiffeq(diffeq, y(z), 1/5, 1000000);
0.033253281257567506772459381920024394391065961347292863\
    13611785593075654371610784719859620906805710762776061\
    65993844793918297941976188620650536691082179149605904\
    31080482988558239935175505111768194891591740446771304\
    74730251896359727561534310095807343639273056518962333\
    97217595138842309884016425632431029577130431472108646\
    95485154767624024297343851584414126056237771911489680\
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    97933258259972366466573219602501650218139747781157348\
    78322628655747195818205282428148240800376913561455564\
    29598794491231828039584256430669932365880956101719727\
    33806130243940574539991121877851105270752378138422728\
    76176859592508040781771637205060431902227437673286901\
    71292574098466950906705927590030494460150099288210121\
    868701569
```

Complexity

Theorem (Chudnovsky^2)

Let y be a D-finite function. Let z be a point on the Riemann surface of y.

The value $y(z)$ may be computed with error bounded by 2^{-n} in

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

bit operations.

Note: this is for fixed y and z. See the paper for some results about the dependency on y.

Complexity

Theorem (Chudnovsky^2, van der Hoeven)

Let y be a D-finite function. Let z be a point on the Riemann surface of y.

The value $y(z)$ may be computed 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.

Note: this is for fixed y and z. See the paper for some results about the dependency on y.

Complexity

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

Let y be a D-finite function. Let z be a point on the Riemann surface of y.

The value $y(z)$ may be computed 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.

Note: this is for fixed y and z. See the paper for some results about the dependency on y.

What NumGfun Really Computes

```
> deq := (1+z^2)*diff(y(z),z,z)
            + 2*z*diff(y(z),z);
```

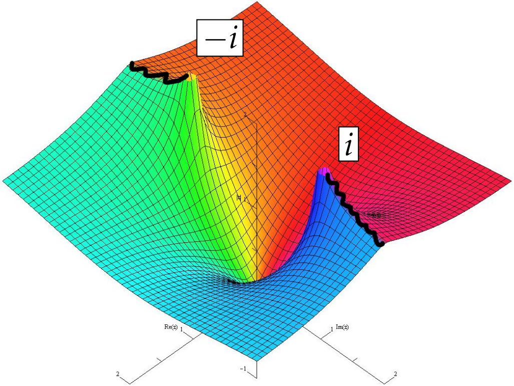
What NumGfun Really Computes

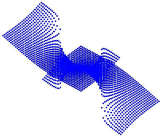
```
> deq := (1+z^2)*diff(y(z),z,z)
         + 2*z*diff(y(z),z);
```

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

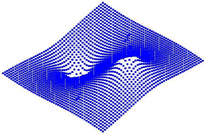
```
>
```

arctan z

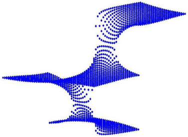




(Re)



(Im)



(arg)

$$(1+z^2)\,y''(z) + 2z\,y'(z) = 0,$$
$$y(0) = 0, \quad y'(0) = 1$$

What NumGfun Really Computes

```
> deq := (1+z^2)*diff(y(z),z,z)
          + 2*z*diff(y(z),z);
```

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

```
>
```

What NumGfun Really Computes

```
> deq := (1+z^2)*diff(y(z),z,z)
          + 2*z*diff(y(z),z);
```

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

```
> evaldiffeq(deq, y(z), 1/2, 20);
```

What NumGfun Really Computes

```
> deq := (1+z^2)*diff(y(z),z,z)
          + 2*z*diff(y(z),z);
```

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

```
> evaldiffeq(deq, y(z), 1/2, 20);
```

$$1.0000000000000000000\ \_C_0 + 0.46364760900080611621\ \_C_1$$

```
>
```

What NumGfun Really Computes

```
> deq := (1+z^2)*diff(y(z),z,z)
          + 2*z*diff(y(z),z);
```

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

```
> evaldiffeq(deq, y(z), 1/2, 20);
```

$$1.0000000000000000000\ \_C_0 + 0.46364760900080611621\ \_C_1$$

```
> transition_matrix(deq, y(z), 1/2, 20);
```

What NumGfun Really Computes

```
> deq := (1+z^2)*diff(y(z),z,z)
         + 2*z*diff(y(z),z);
```

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

```
> evaldiffeq(deq, y(z), 1/2, 20);
```

$$1.0000000000000000000\,\_C_0 + 0.46364760900080611621\,\_C_1$$

```
> transition_matrix(deq, y(z), 1/2, 20);
```

$$\begin{bmatrix} 1.0000000000000000000 & 0.46364760900080611621 \\ 0. & 0.80000000000000000000 \end{bmatrix}$$

```
>
```

What NumGfun Really Computes

```
> deq := (1+z^2)*diff(y(z),z,z)
             + 2*z*diff(y(z),z);
```

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

```
> evaldiffeq(deq, y(z), 1/2, 20);
```

$$1.0000000000000000000\ \_C_0 + 0.46364760900080611621\ \_C_1$$

```
> transition_matrix(deq, y(z), 1/2, 20);
```

$$\begin{bmatrix} 1.0000000000000000000 & 0.46364760900080611621 \\ 0. & 0.80000000000000000000 \end{bmatrix}$$

```
> evaldiffeq(deq, y(z), 5/4*(1+I), 20);
```

What NumGfun Really Computes

```
> deq := (1+z^2)*diff(y(z),z,z)
          + 2*z*diff(y(z),z);
```

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

```
> evaldiffeq(deq, y(z), 1/2, 20);
```

$$1.0000000000000000000\ \_C_0 + 0.46364760900080611621\ \_C_1$$

```
> transition_matrix(deq, y(z), 1/2, 20);
```

$$\begin{bmatrix} 1.0000000000000000000 & 0.46364760900080611621 \\ 0. & 0.80000000000000000000 \end{bmatrix}$$

```
> evaldiffeq(deq, y(z), 5/4*(1+I), 20);
Error, (in rewrite path) evaluation point
outside the disk of convergence of the
differential equation (try specifying an
analytic continuation path such as [0, 5/4+
(5/4)*I])
```

What NumGfun Really Computes

Numerical Analytic Continuation

```
> evaldiffeq(deq, y(z), [0,5/4*(1+I)], 20);
```

What NumGfun Really Computes

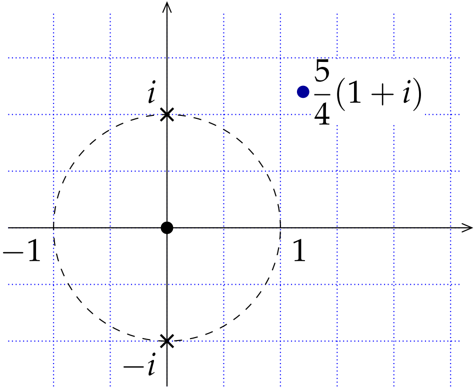
Numerical Analytic Continuation

```
> evaldiffeq(deq, y(z), [0,5/4*(1+I)], 20);
```

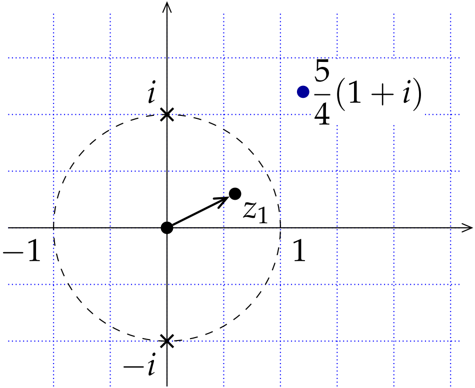
$$1.0000000000000000000\ \_C_0 + (1.1376451955185571679 + 0.35133563902264627452\,\mathrm{I})\ \_C_1$$

```
>
```

Numerical Analytic Continuation

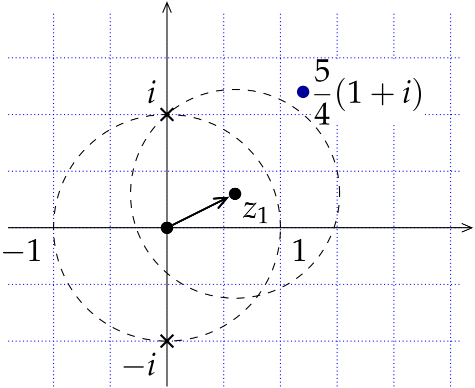


Numerical Analytic Continuation



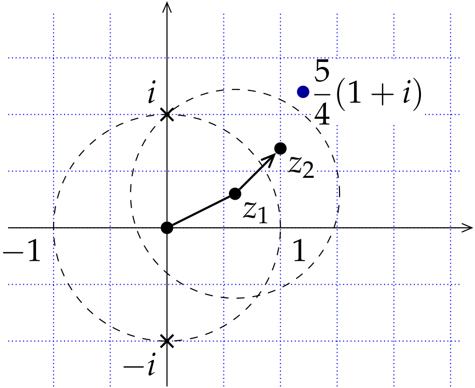
$$\begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix} = \begin{bmatrix} 1 & 0.5705170238\cdots + 0.2200896807\ldots i \\ 0 & 0.7288378766\cdots - 0.2065997130\ldots i \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

Numerical Analytic Continuation



$$\begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix} = \begin{bmatrix} 1 & 0.5705170238\cdots + 0.2200896807\ldots i \\ 0 & 0.7288378766\cdots - 0.2065997130\ldots i \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

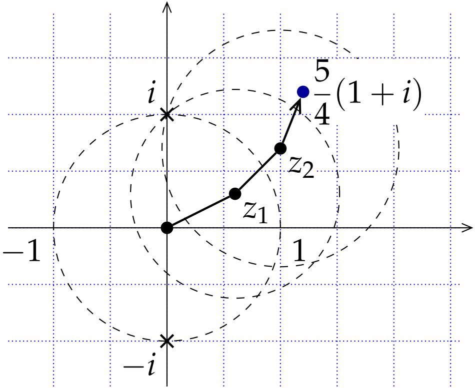
Numerical Analytic Continuation



$$\begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix} = \begin{bmatrix} 1 & 0.5705170238\cdots + 0.2200896807\ldots i \\ 0 & 0.7288378766\cdots - 0.2065997130\ldots i \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

$$\begin{bmatrix} y(z_2) \\ y'(z_2) \end{bmatrix} = \begin{bmatrix} 1 & 0.3656231471\cdots + 0.3290407483\ldots i \\ 0 & 0.7515011402\cdots - 0.0792619810\ldots i \end{bmatrix} \begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix}$$

Numerical Analytic Continuation



$$\begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix} = \begin{bmatrix} 1 & 0.5705170238\cdots + 0.2200896807\ldots i \\ 0 & 0.7288378766\cdots - 0.2065997130\ldots i \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

$$\begin{bmatrix} y(z_2) \\ y'(z_2) \end{bmatrix} = \begin{bmatrix} 1 & 0.3656231471\cdots + 0.3290407483\ldots i \\ 0 & 0.7515011402\cdots - 0.0792619810\ldots i \end{bmatrix} \begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix}$$

What NumGfun Really Computes

Numerical Analytic Continuation

```
> evaldiffeq(deq, y(z), [0,5/4*(1+I)], 20);
```

$$1.0000000000000000000 \_C_0 + (1.1376451955185571679 4 + 0.35133563902264627452\,\mathrm{I})\ \_C_1$$

```
>
```

What NumGfun Really Computes

Numerical Analytic Continuation

```
> evaldiffeq(deq, y(z), [0,5/4*(1+I)], 20);
```

$1.0000000000000000000\ \_C_0 + (1.1376451955185571679 + 0.35133563902264627452\,I)\ \_C_1$

```
> infolevel[gfun] := 2:
  evaldiffeq(deq, y(z), [0,11/10], 20);
```

```
> evaldiffeq(deq, y(z), [0,5/4*(1+I)], 20);
```

$$1.0000000000000000000\ \_C_0 + (1.1376451955185571679 + 0.35133563902264627452\ I)\ \_C_1$$

```
> infolevel[gfun] := 2:
  evaldiffeq(deq, y(z), [0,11/10], 20);
step_transition_matrix: 1/2 --> 11/10 (1
derivative[s]), prec~=.33333e-21, #terms=96
step_transition_matrix: 0 --> 1/2 (2
derivative[s]), prec~=.30264e-21, #terms=86
absolute_precision_warning: Recall that gfun:-
NumGfun works with *absolute* error.
```

$$1.0000000000000000000\ \_C_0 + 0.83298126667443170542\ \_C_1$$

```
>
```

What NumGfun Really Computes

Numerical Analytic Continuation

```
> evaldiffeq(deq, y(z), [0,5/4*(1+I)], 20);
```

$1.0000000000000000000\ \_C_0 + (1.1376451955185571679 4 + 0.3513356390226462745 2\,I)\ \_C_1$

```
> infolevel[gfun] := 2:
  evaldiffeq(deq, y(z), [0,11/10], 20);
step_transition_matrix: 1/2 --> 11/10 (1
derivative[s]), prec~=.33333e-21, #terms=96
step_transition_matrix: 0 --> 1/2 (2
derivative[s]), prec~=.30264e-21, #terms=86
absolute_precision_warning: Recall that gfun:-
NumGfun works with *absolute* error.
```

$1.0000000000000000000\ \_C_0 + 0.83298126667443170542\ \_C_1$

```
>
```

Bounds and Error Control

Core idea: Replace y by a simpler function that "dominates" it

Bound parameters

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

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

Tools: Cauchy-Kovalevskaya majorant series method + basic asymptotic analysis (M. & Salvy 2009)

Symbolic bounds

- Human-readable
- Asymptotically tight

Numeric bounds

- Conservative approximations of parameters
- Faster (no algebraic numbers)

Recap: Numerical evaluation

general	whole class of D-finite functions
guaranteed	rigorous bounds
automatic	input = diff. eq. + ini. val. (no “hints”)
asymptotically fast	quasi-linear complexity w.r.t. precision

Get NumGfun at

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

See the paper for

- other features: recurrence unrolling, more bounds, ...
- new techniques for regular singular points
- a new faster way to compute the transition matrices
- details that matter in practice
- ...

Recap: Numerical evaluation

general	whole class of D-finite functions
guaranteed	rigorous bounds
automatic	input = diff. eq. + ini. val. (no “hints”)
asymptotically fast	quasi-linear complexity w.r.t. precision

Get NumGfun at

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

See the paper for

- other features: recurrence unrolling, more bounds, …
- new techniques for regular singular points
- a n[illegible] pute the transition matrices
- details that matter in practice
- …

Thank you for your attention!

Recap: Numerical evaluation

general	whole class of D-finite functions
guaranteed	rigorous bounds
automatic	input = diff. eq. + ini. val. (no “hints”)
asymptotically fast	quasi-linear complexity w.r.t. precision

Get NumGfun at

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

See the paper for

- other features: recurrence unrolling, more bounds, …
- new techniques for regular singular points
- a new faster way to compute the transition matrices
- details that matter in practice
- …