Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

FWF
Der Wissenschaftsfonds.

DD-finite functions in Sage

Computing beyond holonomic functions

Antonio Jiménez-Pastor

LAND OBERÖSTERREICH

JKU
JOHANNES KEPLER
UNIVERSITY LINZ

FPSAC (Jul. 2019)

Outline

1. Installing the package
2. Using the package
3. Computing with the package
4. Proving with the package
5. Conclusions

How to install the package

Git repository

```
http://git.risc.jku.at/gitweb/?p=ajpastor/diff_defined_functions.git
```

Zip from webpage

```
https://www.dk-compmath.jku.at/Members/antonio/sage-package-dd_functions
```

PyPi (in process)

pip install dd_functions

- **Stable** version
- **Easy** to update

DD-finite Functions

Definition

Let $f \in K[[x]]$ and $R \subset K[[x]]$ a ring. We say that f is differentially definable over R if there exist $d \in \mathbb{N}$ and elements in R $r_0(x), \dots, r_d(x)$ such that:

$$r_d(x)f^{(d)}(x) + \dots + r_0(x)f(x) = 0.$$

D-finite examples

Elementary functions

Exponential (`Exp`), trigonometric (`Sin`, `Cos`),...

Special functions

Bessel functions (`BesselD`), hypergeometric functions (`HypergeometricFunction`),...

Combinatorial functions

Generating functions for holonomic sequences (Catalan numbers, Fibonacci sequence, etc)

DD-finite examples

Classic non-holonomic

Double exponential (e^{e^x-1}), tangent (`Tan`),...

Mathieu functions

DD-finite generalization of the sine and cosine

```
show(MathieuD(init=(1,1)))
```

$$\begin{cases} f''(x) + g_0(x)f(x) = 0 \\ \textit{where} \\ g_0(x) : \begin{cases} g_0^{(3)}(x) + (4)g_0'(x) = 0 \\ \hdashline g_0(0) = a - 2q, g_0'(0) = 0, g_0''(0) = 8q, g_0^{(3)}(0) = 0 \end{cases} \\ \hdashline f(0) = 1, f'(0) = 1, f''(0) = -a + 2q \end{cases}$$

Operations supported

Arithmetic operations

Addition (+, −), product (∗, /, ˆ)

Differential operations

Derivative (`derivative`), integration (`integrate`)

Composition

Using the standard `call` in Sage.

Extracting sequence

Ordinary generating functions

Method `getSequenceElement` allows to get the associated sequence.

Exponential generating functions

Method `getInitialValue` allows to get the associated exponential sequence.

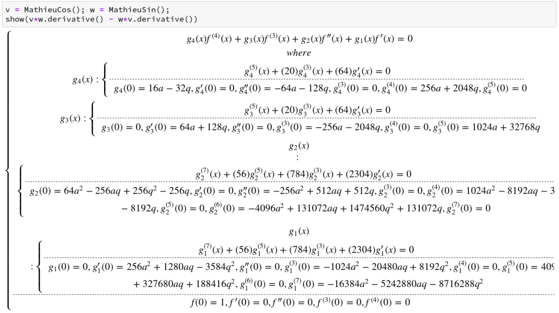
```
# Bell numbers
Exp(Exp(x)-1).getInitialValueList(10)
```

```
[1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147]
```

Proving DD-finite identities

Constant Wronskian

$$w''(x) - (a - 2q\cos(2x))w(x) = 0,$$
$$w_1'w_2 - w_2'w_1 = 1.$$



```
v = MathieuCos(); w = MathieuSin();
show(v*w.derivative() - w*v.derivative())
```

$$g_4(x)f^{(4)}(x) + g_3(x)f^{(3)}(x) + g_2(x)f''(x) + g_1(x)f'(x) = 0$$

where

$$g_4(x) : \left\{ \begin{array}{c} g_4^{(5)}(x) + (20)g_4^{(3)}(x) + (64)g_4'(x) = 0 \\ \hline g_4(0) = 16a - 32q, g_4'(0) = 0, g_4''(0) = -64a - 128q, g_4^{(3)}(0) = 0, g_4^{(4)}(0) = 256a + 2048q, g_4^{(5)}(0) = 0 \end{array} \right.$$

$$g_3(x) : \left\{ \begin{array}{c} g_3^{(5)}(x) + (20)g_3^{(3)}(x) + (64)g_3'(x) = 0 \\ \hline g_3(0) = 0, g_3'(0) = 64a + 128q, g_3''(0) = 0, g_3^{(3)}(0) = -256a - 2048q, g_3^{(4)}(0) = 0, g_3^{(5)}(0) = 1024a + 32768q \end{array} \right.$$

$$g_2(x)$$
$$:$$
$$\left\{ \begin{array}{c} g_2^{(7)}(x) + (56)g_2^{(5)}(x) + (784)g_2^{(3)}(x) + (2304)g_2'(x) = 0 \\ \hline g_2(0) = 64a^2 - 256aq + 256q^2 - 256q, g_2'(0) = 0, g_2''(0) = -256a^2 + 512aq + 512q, g_2^{(3)}(0) = 0, g_2^{(4)}(0) = 1024a^2 - 8192aq - 3 \\ - 8192q, g_2^{(5)}(0) = 0, g_2^{(6)}(0) = -4096a^2 + 131072aq + 1474560q^2 + 131072q, g_2^{(7)}(0) = 0 \end{array} \right.$$

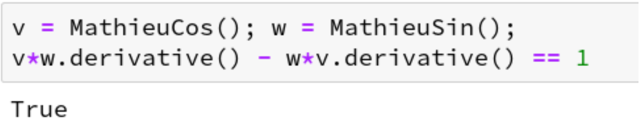
$$g_1(x)$$
$$: \left\{ \begin{array}{c} g_1^{(7)}(x) + (56)g_1^{(5)}(x) + (784)g_1^{(3)}(x) + (2304)g_1'(x) = 0 \\ \hline g_1(0) = 0, g_1'(0) = 256a^2 + 1280aq - 3584q^2, g_1''(0) = 0, g_1^{(3)}(0) = -1024a^2 - 20480aq + 8192q^2, g_1^{(4)}(0) = 0, g_1^{(5)}(0) = 409 \\ + 327680aq + 188416q^2, g_1^{(6)}(0) = 0, g_1^{(7)}(0) = -16384a^2 - 5242880aq - 8716288q^2 \end{array} \right.$$

$$f(0) = 1, f'(0) = 0, f''(0) = 0, f^{(3)}(0) = 0, f^{(4)}(0) = 0$$

Proving DD-finite identities

Constant Wronskian

$$w''(x) - (a - 2q\cos(2x))w(x) = 0,$$
$$w_1'w_2 - w_2'w_1 = 1.$$



```
v = MathieuCos(); w = MathieuSin();
v*w.derivative() - w*v.derivative() == 1
```

```
True
```

Conclusions

Features

- Sage package for DD-finite functions
- Arithmetic and differentil porperties implemented
- Composition implemented
- Zero recognition (equality) implemented

To be done

- Improve performance
- Getting more examples
- Polish the current implementation

Thank you!

Antonio Jiménez-Pastor

- https://www.dk-compmath.jku.at/people/antonio
- https://www.risc.jku.at/home/ajpastor

Sage package:

https://www.dk-compmath.jku.at/Members/antonio/
sage-package-dd_functions