Software Demo: DifferentialAlgebra

François Boulier

March 7th, 2024

What the Package is All About

Ritt and Kolchin *Differential Algebra* is an algebraic theory for systems of polynomial differential equations

- Joseph Fels Ritt. *Differential Algebra*. 1950
- Ellis Robert Kolchin. *Differential Algebra and Algebraic Groups*. 1973

The `integrate` command

It rewrites a *differential fraction* F as a sum

$$F = F_0 + \frac{\mathrm{d}}{\mathrm{d}x} F_1 + \frac{\mathrm{d}^2}{\mathrm{d}x^2} F_2 + \cdots$$

This form permits to easily integrate F

$$\int\int F = \int\int F_0 + \int F_1 + F_2 + \cdots$$

- F. Boulier, J. Lallemand, F. Lemaire, G. Regensburger, M. Rosenkranz. *Additive Normal Forms and Integration of Differential Fractions*. 2016

Software Internals

In Linux, Mac OS and (hopefully) soon Windows:

`python3 -m pip install DifferentialAlgebra`

Source code (LGPL license) available at

`codeberg.org/francois.boulier/DifferentialAlgebra`

Made of

- The BMI Cython / Maple code for Python / Maple packages
- The BMI C interface library (12000 lines of code)
- The BLAD C computer algebra library (73000 lines, 1260 functions, 240 data types, 320 tests, 220 pages of documentation, still counting . . .)
- a rather small software which gets compiled in a few minutes

Software Internals

In Linux, Mac OS and (hopefully) soon Windows:

`python3 -m pip install DifferentialAlgebra`

Source code (LGPL license) available at

`codeberg.org/francois.boulier/DifferentialAlgebra`

Install (should be!) easy

- Thanks to Python wheels, the library is precompiled for each pair (Python release $\times$ supported platform/OS)
- Users choose to install precompiled version or compile the source code
- I would appreciate some help for Mac OS, Windows platforms

Software Internals

In Linux, Mac OS and (hopefully) soon Windows:

`python3 -m pip install DifferentialAlgebra`

Source code (LGPL license) available at

`codeberg.org/francois.boulier/DifferentialAlgebra`

Long term goal

- Make the theory easier to spread by providing introductory theoretical texts illustrated by computations
- Make the software interesting for nonexpert scientific computing users by focusing on portability, reliability and connecting differential algebra internals to more widely known scientific computing tools

Numerical Integration of ODE

An important application domain: existence and uniqueness of formal power series solutions allows numerical integration

A nontrivial problem: the next ODE has formal power series solutions at $x = 0$ if and only if $y_0 \neq -1/n$ for any nonnegative integer n

$$x\,y\,\ddot{y} + \dot{y} + y^2 + 1 \quad = \quad 0$$

Many theoretical results in

- Jan Denef, Leonard Lipshitz. *Formal Power Series Solutions of Algebraic Differential Equations*. 1984

Software is work in progress: internal BLAD demo
A more achieved presentation at CASC 2024 / Rennes?

From Differential to Integral Equations

March 8, 2024

From Differential to Integral Equations

This worksheet works with DifferentialAlgebra4.0

[1]:
```
from sympy import *
from DifferentialAlgebra import *
init_printing ()
```

[2]:
```
t, k12, k21, ke, Ve = var('t, k_12, k_21, k_e, V_e')
x1, x2, u, y = function ('x_1, x_2, u, y')
params = [k12, k21, ke, Ve]
```

The DifferentialRing function defines a differential polynomial ring endowed with a ranking

[3]:
```
R = DifferentialRing (derivations = [t], blocks = [[x1,x2],[u,y],params],␣
↪parameters = params)
R
```

[3]:
```
differential_ring
```

A ranking is a total ordering on the set of the derivatives of the differential indeterminates

This one eliminates the state variables x1(t), x2(t) and their derivatives

[4]:
```
L = [x1(t), x2(t), y(t), u(t)]
L = L + R.differentiate (L, t) + R.differentiate (L, t**2)
L
```

[4]:
$$\left[x_1(t),\ x_2(t),\ y(t),\ u(t),\ \frac{d}{dt}x_1(t),\ \frac{d}{dt}x_2(t),\ \frac{d}{dt}y(t),\ \frac{d}{dt}u(t),\ \frac{d^2}{dt^2}x_1(t),\ \frac{d^2}{dt^2}x_2(t),\ \frac{d^2}{dt^2}y(t),\ \frac{d^2}{dt^2}u(t)\right]$$

[5]:
```
R.sort (L, 'descending')
```

[5]:
$$\left[\frac{d^2}{dt^2}x_1(t),\ \frac{d^2}{dt^2}x_2(t),\ \frac{d}{dt}x_1(t),\ \frac{d}{dt}x_2(t),\ x_1(t),\ x_2(t),\ \frac{d^2}{dt^2}u(t),\ \frac{d^2}{dt^2}y(t),\ \frac{d}{dt}u(t),\ \frac{d}{dt}y(t),\ u(t),\ y(t)\right]$$

A nonlinear ODE model: one command u(t), one output y(t)

[6]:
```
syst = [ Eq (Derivative (x1(t),t), - k12*x1(t) + k21*x2(t) - (Ve*x1(t))/(ke +␣
↪x1(t)) + u(t)),
         Eq (Derivative (x2(t),t), k12*x1(t) - k21*x2(t)),
         Eq (y(t), x1(t)) ]
```

```
syst
```

[6]:
$$\left[\frac{d}{dt}x_1(t) = -\frac{V_e x_1(t)}{k_e + x_1(t)} - k_{12}x_1(t) + k_{21}x_2(t) + u(t),\ \frac{d}{dt}x_2(t) = k_{12}x_1(t) - k_{21}x_2(t),\ y(t) = x_1(t)\right]$$

The RosenfeldGroebner method performs an elimination process over the ODE model

Input: a system + a ranking

Output: finitely many regular differential chains

The ranking aims at producing a differential equation in y(t) and u(t)

[7]:
```
ideal = R.RosenfeldGroebner (syst)
ideal
```

[7]:
```
[regular_differential_chain, regular_differential_chain]
```

[8]:
```
[ C.equations (solved=False) for C in ideal ]
```

[8]:
$$\left[\left[V_e k_{21} k_e y(t) + V_e k_{21} y^2(t) + V_e k_e \frac{d}{dt}y(t) + k_{12}k_e^2\frac{d}{dt}y(t) + 2k_{12}k_e y(t)\frac{d}{dt}y(t) + k_{12}y^2(t)\frac{d}{dt}y(t) - k_{21}k_e^2 u(t) + k_{21}k_e^2 \frac{d}{d}\right.\right.$$

To avoid discussing the possible vanishing of parameters, consider them as base field elements.

Perform again the elimination process

[9]:
```
K = BaseFieldExtension (generators = params)
K
```

[9]:
```
differential_field
```

[10]:
```
ideal = R.RosenfeldGroebner (syst, basefield = K)
ideal
```

[10]:
```
[regular_differential_chain]
```

The sought ODE as a fraction of two differential polynomials

[11]:
```
ode = ideal[0].equations ()[0]
ode = ode / R.initial (ode)
ode
```

[11]:
$$V_e k_{21} k_e y(t) + V_e k_{21} y^2(t) + V_e k_e \tfrac{d}{dt}y(t) + k_{12}k_e^2\tfrac{d}{dt}y(t) + 2k_{12}k_e y(t)\tfrac{d}{dt}y(t) + k_{12}y^2(t)\tfrac{d}{dt}y(t) - k_{21}k_e^2 u(t) + k_{21}k_e^2 \tfrac{d}{dt}y(t)$$

The integrate method expresses its argument as a sum of derivatives of fractions.

The result can be used to transform a differential fraction in integral form

[12]:
```
L = R.integrate (ode, t)
L
```

[12]:
$$\left[\frac{V_e k_{21} y(t) - k_{21}k_e u(t) - k_{21}u(t)y(t)}{k_e + y(t)},\ \frac{-V_e k_e - k_{12}k_e^2 + k_{12}y^2(t) - k_{21}k_e^2 + k_{21}y^2(t) - k_e u(t) - u(t)y(t)}{k_e + y(t)},\ y(t)\right]$$

[13]:
```
ode_in_integral_form = L[2] + Integral (L[1], (t, 0, t)) + Integral (Integral
↪(L[0], (t, 0, t)), (t, 0, t))
ode_in_integral_form
```

[13]:

$$y(t) + \int_0^t \frac{-V_e k_e - k_{12} k_e^2 + k_{12} y^2(t) - k_{21} k_e^2 + k_{21} y^2(t) - k_e u(t) - u(t) y(t)}{k_e + y(t)}\, dt + \int_0^t \int_0^t \frac{V_e k_{21} y(t) - k_{21} k_e u(t) - k_{21} u(t) y(t)}{k_e + y(t)}\, dt\, dt$$

[14]:
```
simplify (Derivative (ode_in_integral_form, t, t).doit () - ode)
```

[14]:

$$0$$

[ ]:
```

```