Does Axiom Solve Systems of O.D.E.'s Like Mathematica? *

Nicolas Robidoux
Center for Nonlinear Studies
Los Alamos National Laboratory
and
Program in Mathematics
University of New Mexico

24 June 1993
Revised 28 July 1993

If I were demonstrating Axiom and were asked this question, my reply would be "No, but I am not sure that this is a bad thing." and I would illustrate this with the following example.

Consider the following system of O.D.E.'s:

$$\begin{aligned} \frac{dx_1}{dt} &= \left(1 + \frac{\cos t}{2 + \sin t}\right) x_1 \\ \frac{dx_2}{dt} &= x_1 - x_2 \end{aligned}$$

This is a very simple system: x_1 is actually uncoupled from x_2.

Solving with Mathematica

The obvious thing to do is to use Mathematica's `DSolve` (commands will appear in `typewriter` type while selected output will be shown using edited TEX produced by Mathematica or Axiom[1]; the command lines can be lifted from the LATEX file that generates this document and fed without modification to the symbolic manipulators).

*Work performed under the auspices of the US Department of Energy. This is report LA-UR-93-2235. It can be obtained by sending an e-mail message to infodesk@nag.com or luczak@nag.com requesting it in LATEX and/or PostScript form.

[1] Editing is necessary due to a bug in Mathematica's TEX facility: it does not convert "\$" (which is part of `Module`'s naming convention) to "\\$".

```
DSolve[
       {
        x1'[t] == (1+Cos[t]/(2+Sin[t])) x1[t]
        ,
        x2'[t] == x1[t]-x2[t]
       }
       ,
       {x1,x2}
       ,
       t
      ]

%[[1]]

{ x1 , x2 } = { x1/.% , x2/.% }

x1[t]

TeXForm[%]

x2[t]

In[-2]
```

This yields

$$\begin{aligned} x_1(t) &= e^{\frac{t\,(2+\cos(t)+\sin(t))}{2+\sin(t)}}\ \mathrm{C}(1) \\ x_2(t) &= \frac{\mathrm{C}(2)}{e^t} + \frac{\left(-1+e^{\frac{t\,(4+\cos(t)+2\ \sin(t))}{2+\sin(t)}}\right)\ \mathrm{C}(1)\ (2+\sin(t))}{e^t\ (4+\cos(t)+2\ \sin(t))} \end{aligned}$$

which is completely wrong: even x_1 is incorrectly given![2]

Because x_1 is uncoupled from x_2 the following alternative route to a solution can be tried: first integrate the equation which only involves x_1 (using `DSolve` or `Integrate`) and then plug the result into the other equation to get x_2. This is done here:

```
DSolve[
       x1'[t] == (1+Cos[t]/(2+Sin[t])) x1[t]
       , x1 , t ,
       DSolveConstants->(Module[{C}, C])  (* This prevents
```

[2] These results are obtained using Mathematica 2.1, which was the latest version when I first got them and is still the most current one available to me. Roger Germundsson points out that Mathematica 2.2 wisely returns the first command (`DSolve...`) unevaluated. He also mentions that similar results arise when one tries to solve matrix Riccati equations with time dependent coefficients.

```
            clashing constants of integration. *)
          ]

%[[1]][[1]]

x1=x1/.%

x1[t]

TeXForm[%]

DSolve[
            x2'[t] == x1[t]-x2[t]
            ,x2,t,
            DSolveConstants->(Module[{C}, C])
          ]

%[[1]][[1]]

x2=x2/.%

x2[t]

TeXForm[%]
```

This time the result is

$$\begin{aligned} x_1(t) &= e^t \, C\$\mathit{1}(1)\ (2+\sin(t)) \\ x_2(t) &= \frac{C\$\mathit{2}(1)}{e^t} + \frac{e^t \, C\$\mathit{1}(1)\ (5-\cos(t)+2\,\sin(t))}{5} \end{aligned}$$

which is correct.

Solving with Axiom

In the following the production of TEX by Axiom is toggled on and off through the Hyperdoc menu.

Because Axiom does not have a system of PDE's solver, I will follow the alternative route to a solution.[3]

[3] Another possibility, suggested by Richard Luczak, is to use `seriesSolve` which computes power series solutions of (possibly nonlinear) systems of O.D.E.'s:

```
x1 := operator 'x1
x2 := operator 'x2
deq1 := D(x1 t,t) = (1+ cos t /(2+sin t)) * x1 t
deq2 := D(x2 t,t) = x1 t - x2 t
seriesSolve( [deq1,deq2] , [x1,x2] , t=a , [ x1(a)=B , x2(a)=C ] )
```

```
X1 := operator 'X1

deq1 :=    D(X1 t,t) = (1+ cos t /(2+sin t)) * X1 t

solve( deq1 , X1 , t )

C1 * %.basis.1

function( % , 'x1 , 't )

x1

X2 := operator 'X2

deq2 :=    D(X2 t,t) = x1 t - X2 t

solve( deq2 , X2 , t )

%.particular

simplify %

% + C2 * %%(-3).basis.1

function( % , 'x2 , 't )

x2
```

This yields

$$\begin{aligned} x1(t) &= C1\ e^t\ \sqrt{4\ \sin(t) - \cos(t)^2 + 5} \\ x2(t) &= \frac{2\ C1\ e^t\ \sin(t) + (-C1\ \cos(t) + 5\ C1)\ e^t + 5\ C2\ e^{(-t)}}{5} \end{aligned}$$

which is correct, although one suspects (see Mathematica's solution!) that some simplification is possible, which gives me the opportunity to illustrate how one can go about getting at pieces of (maybe complex) expressions, manipulating them, here detecting a trigonometric polynomial that factors, and then putting the whole thing back together without having to retype any Axiom output.

```
x1 t

isTimes %
```

Getting a simple expression for the solution from the power series requires a keen eye for products of well known power series, although comparison with the other solutions is very easy.

```
% :: List EXPR INT

factors := %  --  This defines the list of factors of x1(t).
              --  I will work on the first factor, replace it by
              --  its simplified version and then put x1(t)
              --  back together from its revised list of factors.

%.1**2

removeCosSq %

eval( % , sin t = x )

% :: POLY INT

factor %

factors %  --  Note that even though I have previously defined a
           --  factors which is of type List Expression Integer,
           --  Axiom knows that I refer to the operation factors
           --  since the present factors has an argument.

%.1.factor  --  The above factor was an operation, while this
            --  one is a key.

eval( % , x = sin t )

factors.1 := %

reduce( * , factors )

function( % , 'x1 , 't )

x1 t
```

which gives me the simplified

$$x1(t) = C1\ e^t \sin(t) + 2\ C1\ e^t$$

With the aim of formatting mathematical expressions so that translation into other languages (C, for example) becomes straightforward, Larry Lambe (1993) has systematized the recursive process of breaking expressions into constituent parts, replacing operators by variables and processing the resulting rational functions, by implementing a new Axiom type called ExpressionTree in which expressions are decomposed (with optional factorization) into a natural tree structure. The new type is described in the report *On Generating Code in AXIOM* (submitted for publication) which is available from the University of Stockholm as a Mathematics Department Report.

Acknowledgments

I thank James M. Hyman, Larry Lambe, Richard Luczak and Stanly Steinberg for their advice and support.

Center for Nonlinear Studies, MS-B258, Los Alamos National Laboratory, Los Alamos, NM 87545
E-mail address: nicolas@goshawk.lanl.gov