Applying AXIOM to Partial Differential Equations

W.M. Seiler
Institut für Algorithmen und Kognitive Systeme
Universität Karlsruhe
76128 Karlsruhe, Germany
Email: seilerw@ira.uka.de

Abstract

We present an AXIOM environment called JET for geometric computations with partial differential equations within the framework of the jet bundle formalism. This comprises especially the completion of a given differential equation to an involutive one according to the Cartan-Kuranishi Theorem and the setting up of the determining system for the generators of classical and non-classical Lie symmetries. Details of the implementation are described and examples of applications are given. An appendix contains tables of all exported functions.

1 Computer Algebra and Differential Equations

Most casual users of computer algebra systems think that computer algebra and differential
equations concerns basically the design of solution algorithms. But the real situation is
fairly different. Although most general purpose systems provide a kind of `solve` command
ifferential equations, they actually employ mainly well-known techniques and some
heuristics to choose and apply them. Especially for partial differential
hmtreating reasonably general and complicated systems is
do not treat them at all.
fferential equations tend to work
f the solution space.
algebra

so-called Differential Gröbner Bases [29] can also be seen as an extension of this approach.

In geometric theories the notion of a passive system is replaced by involution. Hartley
l Tucker [18] implemented the Cartan-Kähler approach [6] using exterior systems. An
our AXIOM implementation of the formal approach was published in

ideal theory. Here one tries to find a differential extension of alge-
theory. Many of the ideas can already be found in the book of Ritt [39]. The
öbner bases or characteristic sets. Since the ring of differential
Noetherian, this generalization runs into problems, for
algorithm do not terminate in general [8]. As
mplemented) so-called Differ-
aker properties

g for a differential analog of
s Theory resem
and

differences to more standard systems. The following four section describe in some detail
the implementation of JET, whereas Sections 10 and 11 give examples of its application.
ally, some conclusions are given. An appendix contains tables of the exported functions
nce.

ution

uses a geometric approach to differential equations based on the jet bundle
e scope of this paper to give a detailed introduction into the
er is referred to the literature [31, 47].
te system, although the whole theory can
the space of the independent
u^m be fiber coordinates for
are written in multi-index notation
$\cdots + \mu_n$ is the length of the multi-index
p^α_μ up to order q defines a local coordinate sys-
ndle $J_q\mathcal{E}$. A system of differential equation $\mathcal{R}_q$ of order q can
lly by

$$\mathcal{R}_q : \left\{ \Phi^\tau\left(x^i, u^\alpha, p^\alpha_\mu\right) = 0\,, \quad \tau = 1, \dots, p; \ |\mu| \le q. \right. \qquad (1)$$

his represents a fibered submanifold of $J_q\mathcal{E}$
ast some of the ideas behind the concept of involution can be understood best
by considering the order by order construction of a formal power series solution. For
this purpose, we introduce the *symbol* $\mathcal{M}_q$ of a differential equation $\mathcal{R}_q$. If $\mathcal{R}_q$ is locally
described by the system (1), then its symbol is the solution space of the foll
system of (algebraic!) equations in the unknowns v^α_μ

$$\mathcal{M}_q : \left\{ \sum_{\alpha, |\mu| = q} \left(\frac{\partial \Phi^\tau}{\partial p^\alpha_\mu}\right) v^\alpha_\mu = 0\,. \right.$$

(By abuse of language, we will refer to both the linear syst
the symbol).
The v^α_μ provide coordinates of the finite-dim
introduce one coordinate for each d
considering a quasi-l
For such

The remaining coefficients can be computed by linear algebra only. For the coefficients
of order $q+r$ we use the *prolonged* systems $\mathcal{R}_{q+r}$ which are obtained by differenti
each equation in $\mathcal{R}_q$ r times formally with respect to all independent var
formal derivative is defined by

$$D_i\,\Phi^\tau = \frac{\partial\Phi^\tau}{\partial x^i} + \sum_{\alpha}\frac{\partial\Phi^\tau}{\partial u^\alpha}p^\alpha_i + \sum_{\alpha,\,\mu|}\frac{\partial\Phi^\tau}{\partial p^\alpha_\mu}p^\alpha_{\mu+1_i}\,.$$

Hence all prolonged equations are quasi-linear. If we subst
into $\mathcal{R}_{q+r}$ and evaluate at x^0, we get an inhomoge
of order $q+r$. Its homogeneous part is
the symbol of $\mathcal{R}_{q+r}$.
The Taylor coefficients a^α_μ o
side of this linear s
the coeff

The above definition of the $\beta_q^{(k)}$ is obviously coordinate dependent. Thus it seems,
as if the involution of a symbol depends on the chosen coordinate system, too
can, however, show that almost every coordinate system leads to the
the $\beta_q^{(k)}$. These values are characterized by the property
$k = 1, \ldots, n$, are maximal.[2] A coordinate system
δ-regular. Definition 2 assumes that
There exist ways to ci
a generic line

systemuntil its symbol becomes involutive. The outer loop checks then for integrability
conditions and adds them. The difficult part of the proof is to showthe terminat
inner loop. The termination of the outer one follows froma simple
Involution of a symbol can be checked easily using D
coordinate systemis δ-regular what we will d
Whether or not integrability
a dimensional a
Denote the
Its di
wh

are of course not possible. The only possibility for integrability conditions is the prolo
gation of lower order equations. For partial differential equations we rec
integrability conditions can always be found by considerin
to non-multiplicative variables.

To conclude this section we brief
arbitrariness of the ge
but thei

4 Symmetry Theory

The most general definition of a symmetry simply states that it is a transformation th
maps solutions into solutions. We will consider here diffeomorphisms ϕ :
them a *Lie point symmetry* of the differential equation
usually impossible to find all such symmetries.
infinitesimal transformation, i.e.

$\bar{v}$

Using the chain rule it is s
acting on $J_q\mathcal{E}$,

where the coeffic

If now a lo

holds

solution $u^\alpha(x^i)$ satisfies not only the considered differential equation $\mathcal{R}_q$ but in addition the *invariant surface condition*

$$\sum_{i=1}^{n} \zeta^i(x,u) p_i^\alpha = \eta^\alpha(x,u)\,, \qquad \alpha=1,\dots,m$$

These are quasi-linear, first-order equations. In the case of a hi
algebra we must add one such set of conditions fo
the general solution of (19), one subst
tion. Or, if we assume w
derivatives

mark AXIOM. We are currently using Version 1.2. It is rather different compared with

other general purpose computer algebra systems. Systems like REDUCE, Maple, M

matica etc. differ not much in their principal structure: they have one ba

symbolic expressions and the core of the systemconsists of

ences between the systems lay in the simplific

packages (e.g. factorization, i

user interface

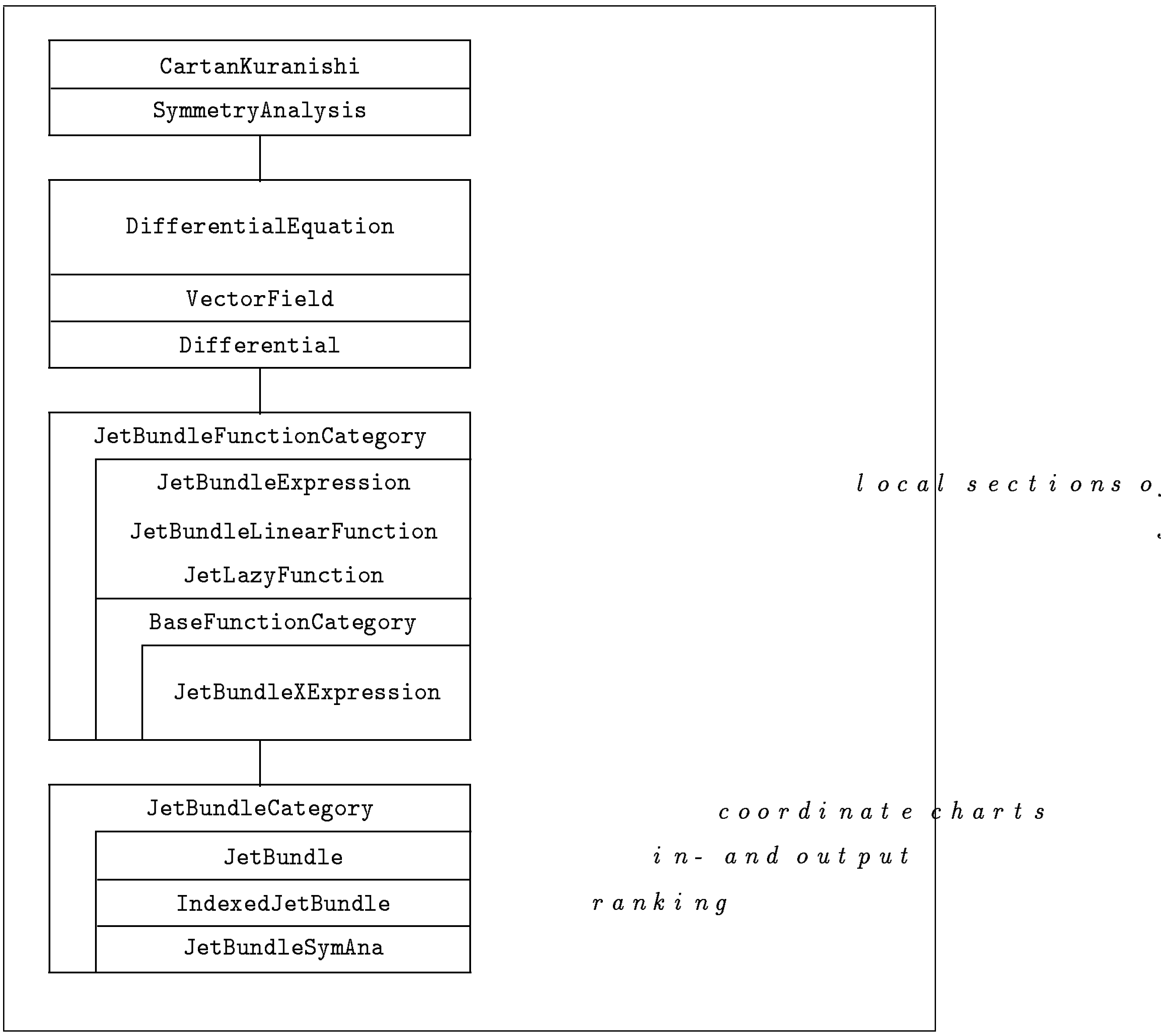
AXIOMh

to ask whether a given domain has a certain attribute. This is useful, when domains or
categories are passed as arguments (see below).

AXIOM's approach to symbolic computation has several advantages. It
grammer to ensure the correctness of his computations. Since i
every object must have a type, i.e. it belongs t
of operations can be performed with i
sound, only well-defined op
ring elemen

whi ch t ri es t o deri ve t he types of t he obj ect s i n t he i nput l i ne. Onl y i f t here are ambi gui t i e
or t he st ruct ure of t he i nput i s t oo compl i cat ed, t he user must decl are t he t
some obj ect s expl i ci t l y. The onl y i nformat i on t hi s mechani
(here oft en cal l ed mode maps) of al l used operat
Ref. [56]).
Because of t he huge si ze of t
at t he st art of a
"expo

This problem could be avoided by implementing a general purpose environment for
geometric computations. Such an environment should comprise basic data structures
procedures for jet bundles and differential equations as they are typical
this environment it would then be possible to implement p
like completion to an involutive system, const
We have started with the develop
within the jet bundle fo
such a proje


CartanKuranishi
SymmetryAnalysis
DifferentialEquation
VectorField
Differential
JetBundleFunctionCategory
JetBundleExpression
JetBundleLinearFunction
JetLazyFunction
BaseFunctionCategory
JetBundleXExpression
JetBundleCategory
JetBundle
IndexedJetBundle
JetBundleSymAna
local sections o
coordinate charts
in- and output
ranking

echelon forms, whereas LUDecomposition implements the LU decomposition method for the standard matrix type of AXIOM. JetCoordinateTransformation prolongs coordinate transformations in the base space $\mathcal{E}$ into the jet bundle.

JetBundleCategory	JBC
JetBundle	JB
IndexedJetBundle	IJB
JetBundleSymAna	JBSA
JetBundleFunctionCategory	JBFC
BaseFunctionCategory	BFC
JetBundleExpression	JBE
JetBundleXExpression	JBX
JetBundleLinearFunction	JBLF
JetLazyFunction	JLF
DifferentialEquation	DE
VectorField	VF
Differential	DIFF
CartanKuranishi	CK
SymmetryAnalysis	SYMANA
SparseEchelonMatrix	SEM
JetCoordinateTransformation	JCT
LUDecomposition	LUD

Figure 3: Abbreviations of the new categories, domains, and p

Examples of the application of this environment in concre in Sections 10 and 11. They contain complete e sessions. Tables with most of the expo also contain short descripti

7 Imple

index is needed for derivatives. IJB recognizes two different notations for the lower index.
Internally always standard multi-index notation is used. For in- and output
choose between this and repeated index notation which is the def
convenient for derivatives of low order. The nota
`setNotation`.
Jet variables are generated with t
one independent or only
of the

`subst` substitutes a given expression for a jet variable in another given expression.

`order` yields the order as differential equation of a function; it is computed

of the leading derivative obtained from `leadingDer`. The defaul

latter one in turn calls `jetVariables` and determines

Whereas `subst` operates purely algebraic,

given jet variable. This is

For partial der

obje

As we will see in the next section the power and the efficiency of the simplification rou-
tines and here especially of simplify are crucial for the performance of our env
in almost any calculation. Thus special care should be applied to t
tation of any domain belonging to JBFC. It is e.g. ver
system in such a form that its symbol is al
This does not only make the
that the simplifi
The

The implementation of `simplify` in JBE tries therefore to avoid the use of the AXIOM
`groebner` procedure as much as possible. The main strategy rests on the observa
that differential equations are usually sparse, i.e. not every jet vari
equation. A large part of the simplification can ofte
equations according to their leading deriv
same leading derivative, si
this case this eq

compute a partial derivative but only stores a pointer to the function to be differentiated
and the variable with respect to which it is differentiated. Only if later the
is needed, the differentiation is actually performed.
Such lazy evaluation schemes have been successf
of infinite objects like series [7]. Th
momentarily necessary, but
case the idea i
der

evaluation was performed. Otherwise it starts using the procedure eval1 to evaluate as
many of the lazy terms as necessary to obtain a sharp bound.
We have already mentioned that zero? (and similarly one?) is based on
But many procedures implemented categorically in JBFC use zero?
to avoid vanishing entries. This would lead to many un
introduced the attribute lazyRep to distinguish
mechanism. In the case of such a dom
might cause evaluation as
discussed in Sect
As explai
ex

Because of this list we try to keep track of the equations during simplification. If an
equation is a combination of several other equations, then its value in Derivi
by the minimum of the values of the other equations. This strate
but it is the best one which can be realized with reaso
The central operations in DE are prolong,
ones for symbol and tableau. simplify
in JBFC and assumes that i
starting wit

can thus be used to compute k tableaux. To enter one-forms the domain Differential
must be used. It represents together with the domain VectorField the remainder of th
third layer. The implementation of both is somewhat rudimentary, as we hope
day AXIOM will contain a reasonable environment for differential
and then it should be used instead of some special do
Both use an identical representation co
ficients, the other one the
The one for th
belon

one procedure detSys to set up the determining systemfor symmetry generators. There
exist different mode maps for this procedure. One can e.g. provide a special ansa
the symmetry generator or a list of derivatives for which the equatio
be solved. The default is the most general ansatz and eac
leading derivative.
If the general ansatz is chosen, d
data type, namely as functi
variables. In thi
and fu

for matrices with polynomial entries is, however, not correct and essentially due to his
incorrect implementation.

Most of the matrices studied by Berchtold were still fairly dense co
matrices typically appearing as symbols. For such matrices G
essentially to sorting the rows according to the
that two rows have their pivots i
Another result of Ber
the Barei

The package `JetCoordinateTransformation` provides two procedures `transform` to
prolong coordinate transformations of the base bundle $\mathcal{E}$ into higher order jet bundles.
parameters are two jet bundles and two vectors (one for the independent a
dependent variables) containing expressions for the old coor
ones. One procedure computes the transformation
one transforms an expression in the o
In the current implemen
for this restri

and of all found integrability conditions. The output is directly in TEX. The computa-
tion follows exactly the steps of the treatment in Ref. [33]. It is, however,
fact that the program produces the integrability conditions i
Fig. 4. Actually many more integrability conditi
a lot of non-multiplicative variabl
independent. complet
equations

In- and output are shown in Fig. 5 on page 33. The input is very similar to the
previous example. The main difference is that we use another domain for the equati
namely JBE. They can now be arbitrary expressions in the independent
variables. The integrability condition can be obtained by taki
D equations and subtracting the t-derivative of t

$$\sum_{i,j=1}^{D} \partial_{x^i} u^j$$

Due to the simplification procedures t
appear different in the outp
tries to solve eq
c

ample. Consider the system

$$\mathcal{R}_2 : \quad \begin{cases} u_{zz} + v u_{xx} = 0, \\ u_{yy} = 0, \\ v_z = 0, \\ v_y - v_x = 0. \end{cases} \tag{25}$$

The completion runs completely analogously to the classical Janet example, i.e. th
jections occur at the same places and $\mathcal{R}_3^{(2)}$ is involutive. The only differ
second projection two further integrability conditions[3]

$$2v_x u_{yxx} + v_{xx} u_{xx}$$

$$4v_x^4 u_{xxxx} + v_{xx}(4v_x^2 - v v_{xx})(2v_x u_{xxx}$$

$$2v_x v_{xxx} - 3v_{xx}^2 = 0.$$

It is obvious that here the simplification routines
time for this example is slightly more
simplification modulo lo
10 hours!
We showed
wit

later in more detail. Here we just want to point out that in several equations the con-
straint was not used for simplification; in the third projection we even obtain t
equations. Both effects are mainly due to current restrictions i
of JBE.

11 Examples II —Other Ap

One of the classical examples
SYMANA to calculate
further

The timing of the first transformation also contains the time needed for the precom-
putation of an inverse Jacobian. Since its result is stored, subsequent transfor
be faster. In this example this effect is neclectable, as the i
unchanged and the mentioned Jacobian measures th
variables. JCT further keeps as hash
second call with the sam

```
jb:=JB(['x,'y,'z],['u])
jbx:=JBX jb
jbl:=JBLF(jb,jbx)
de:=DE(jb,jbl)
ck:=CK(jb,jbl)

eq1 := D('u,['z,'z])$jb::jbl + 'y::jb::jbl * D('u,['x,'x])$jb::jbl
eq2 := D('u,['y,'y])$jb::jbl
janet:de := generate [eq1,eq2]

setOutMode(14)$ck
setRedMode(1)$ck
complete janet
```

Symbol M_2 not involutive! Dimension: 4

Symbol M_3 involutive! Dimension: 4

Equation R_3 not involutive! Dimension: 12

══════1. Projection ══════

Integrability condition(s)

$u_{y,x,x} = 0$

════════════════════

involutive! Dimension: 3

e! Dimension: 2

Dimension: 13

```
jb:=IJB('x,'u,'p,4,3)
jbe:=JBE jb
de:=DE(jb,jbe)
ck:=CK(jb,jbe)

eq1:jbe := P(1,[4]) + U(1)*P(1,[1]) + U(2)*P(1,[2]) + U(3)*P(1,[3])
eq2:jbe := P(2,[4]) + U(1)*P(2,[1]) + U(2)*P(2,[2]) + U(3)*P(2,[3])
eq3:jbe := P(3,[4]) + U(1)*P(3,[1]) + U(2)*P(3,[2]) + U(3)*P(3,[3])
eq4:jbe := P(1,[1]) + P(2,[2]) + P(3,[3])
euler:de := generate [eq1,eq2,eq3,eq4]

setOutMode(14)$ck
setSimpMode(1)$ck
complete euler
```

Symbol $\mathcal{M}_1$ involutive! Dimension: 8

Equation $\mathcal{R}_1$ not involutive! Dimension: 11

═══1. Projection ═══

Integrability condition(s)

$$2\,p^3_2\,p^2_3 + 2\,p^3_1\,p^1_3 + 2\,{p^2_2}^2 + 2\,p^1_1\,p^2_2 + 2\,p^2_1\,p^1_2 + 2\,{p^1_1}^2 = 0$$

$_1)$ involutive! Dimension: 7

Result ***************

lutive!

mension: 14

$=0,$

$- u^3\,p^2_1\,p^1_2 - u^3\,{p^1_1}^2 = 0,$

$^2 = 0$

uation (23).

Equation R_2 not involutive! Dimension: 8

═════1. Projection ═════

Integrability condition(s)

$2\,z\,z_t + 2\,y\,y_t + 2\,x\,x_t = 0$

══════════════════════

$R_2^{(1)}$ not involutive! Dimension: 7

rojection ═════

on(s)

$^{2} + 4\,L\,z^4 + (\;\;4\,Ly^2 + 4\,Lx^2\,)\;\;\;z^2 = 0$

```
v:vf := ansatz()$sym

   tau D  + xi D  + eta D
        t       x        u

ds := detSys([eq])$sym

   [- 2tau , - 2tau , - xi   , - tau   , eta    - 2xi   , - 2tau    - 2xi ,
          u        x      u,u       u,u     u,u      u,x        u,x      u
    2eta    - xi    + xi , - tau    - 2xi  + tau , eta    - eta ]
        u,x     x,x     t       x,x      x      t     x,x      t

lds:List jbl2 := [retract(eq)  for eq in ds]
r2:de := generate lds
setOutMode(4)$ck
setRedMode(1)$ck
complete r2

   *************** Final Result ***************

             (4)
   Equation R     involutive!
             3
   System without prolonged equations. Dimension: 13

       tau     = 0
          t,t,t
        eta   = 0
           u,u
             1
    eta    + - xi = 0
       u,x   2   t
     eta    - eta = 0
        x,x      t
            1
   eta    + - tau   = 0
      u,t   4    t,t
         xi   = 0
           t,t
          xi = 0
            u
         tau = 0
            u
           1
     xi  - - tau = 0
       x   2    t
         tau = 0
            x

   Cartan characters: 2,0,0
```

Figure 7: Determining system for the Heat Equation.

```
burgers:jbe1 := P [2] - P [1,1] - P([1])**2

                        2
   (8)  - u     + u   - u
           x,x     t     x
                           Time: 0.33 (IN) + 0.23 (EV) + 0.80 (OT) = 1.36 sec
transform(burgers)$jct

        - v     + v
           y,y     s
   (9)  -----------
             v
                           Time: 0.22 (IN) + 2.02 (EV) + 1.54 (OT) = 3.78 sec
transform(burgers)$jct

         - v     + v
            y,y     s
   (10)  -----------
              v
                           Time: 0.01 (IN) + 0.35 (EV) + 0.05 (OT) = 0.41 sec
```

Figure 8: Cole-Hopf transformation of the Burgers Equation.

12 Outlook and Discussion

It should be clear from the discussion so far, that this environment is by far

There are permanent changes and improvements. Most changes a

for non-linear equations. The simplification and redu

are still too inefficient for more complica

as a substitute for a spe

are mainly d

c

system Otherwise it will prolong and prolong and prolong without ever finding an invo-
lutive symbol. In principle one could implement the method presented in Ref. [5
k tableaux. We have refrained from this approach, because it become
tionally very demanding.

Computing row echelon forms of symbolic matrices i
computer algebra a matrix is considered as
columns (of course this depend
could partially rem
handle

mathématiques in Montréal and at the School of Physics and Materials in Lancaster. I
am grateful to P. Winternitz and R.W. Tucker, respectively, for their hospitali
work was partially supported by grants of Studienstiftung des deuts
Forschungsgemeinschaft and School of Physics and Mater

References

[1] Th. Becker and V. Weispfenning.

[2] I. Berchtold. Sp
Züri

[3] C

[15] K.O. Geddes, S.R. Czapor, and G. Labahn. *Algorithms for Computer Algebra*
Academic Publishers, Dordrecht, 1992.

[16] K. Gottheil. Axioms, categories and domains. *mathPAD*,

[17] D. Gruntz and M.B. Monagan. Introduction to

[18] D. Hartley and R.W. Tucker. A con
theory of exterior differe

[19] A.K. He

[34] W.H. Preuss, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recip*
in C. Cambridge University Press, Cambridge, 2nd edition, 1992.

[35] D.W. Rand and P. Winternitz. ODEPAINLEVE—a Macsyma package
analysis of ordinary differential equations. *Comp. Phy*

[36] G.J. Reid. Algorithms for reducing a sy
the dimension of its solution
Appl. M

[37]

[49] WM Seiler. Arbitrariness of the general solution and symmetries. *Acta Appl. Ma*
to appear, 1995. (Special Issue Proc. Algebraic and Geometric Str
ential Equations, Twente 1993, J. Krasilshik and P.H.M

[50] WM Seiler. Generalized tableaux and fo
Preprint Lancaster Universit

[51] WM Seiler and R.W T
approach. Prep

[52] M.F. S

A Exported Procedures

The purpose of this appendix is to provide tables of most procedures currently im
in our environment for geometric computations with partial differe
also contain brief descriptions of the tasks of the di
not comment on the implementation or th
text. The same holds for exampl
The order of the tabl
start with

The simplification procedures can be divided into two classes. simplify, simpMod
and simpOne use only algebraic operations. One of the main tasks of simplify i
exhibit integrability conditions, if any are present. Otherwise no mix
different order happens. reduceMod and autoReduce also use
correspond to algorithms used in differential a
There are currently three instance
JetBundleExpression (JBE),
(JLF). JetBundleXExpressi
the sub-categor
space X

Highe

`allRepeated`	`L NNI -> L L PI`	Computes all possible realizations of a given multi-index as repeated index.
`class`	`L NNI -> NNI`	Returns the class of a multi-index.
`class`	`$ -> NNI`	Returns the class of a jet variable.
`coerce`	`$ -> EI`	Coerces a jet variable into an expression.
`coerce`	`$ -> S`	Coerces a jet variable into a symbol.
`rativeOf?`	`($, $) -> L NNI`	Checks whether the first argument is a derivative of the second one. Returns either the difference of their multi-indices, if positive, or an empty list.
	`PI) ->` `n($,"0")`	Differentiates a jet variables with respect to the independent variable labeled by the second argument.
		Returns the (fiber) dimension of the jet bundle of the order.
		the dimension of $S_qT^*X \otimes V\mathcal{E}$ for the given
		tly used notation.
		x of a jet variable.
		h respect to an indepen- ot possible.
		o the indepen-

autoReduce	L $ -> L $	Reduces a system with respect to itself.
class	$ -> NNI	Class of an expression.
const?	$ -> B	Checks whether an expression depends on jet variables.
coerce	JB -> $	Transforms a jet variable into a function.
denominator	$ -> $	Denominator of an expression.
erentiate	($, JB) -> $	Differentiation with respect to a jet variable.
n	(L $, SEM, NNI) -> NNI	Dimension of a system with given Jacobian in the jet bundle of given order.
, JB, $) -> $		Like subst but takes also derivatives into account.
SEM		Extracts symbol from the Jacobian.
		Formal differentiation with respect to an independent variable given by its label. There exist further aps for systems and with more detailed out- ions it is also possible to pass the rgument.
		pends on a given jet

analyseSymbol	SEM -> MVREC	Computes a row echelon form of a given symbol. Counts the multiplicative variables and determines the rank of the matrix.
copy	$ -> $	Returns a copy of a differential equation.
nsion	($, NNI) -> NNI	Computes the dimension of a given differential equation considered as submanifold of a jet bundle of given order.
	-> OUT	Prints most of the information stored about a differential equation: equations ordered by their order, for each order the Jacobian, whether the system is lready simplified and so on.
		ts the symbol from the Jacobian of the highest e equation. If the second argument form is computed.
		ation from a given
		differential

`coefficient`	`($, JB) -> D`	Returns the coefficient in a given direction.
`coefficients`	`$ -> L D`	Returns the coefficients of a vector field.
`commutator`	`($, $) -> $`	Computes the commutator of two given vector fields.
`copy`	`$ -> $`	Returns a copy of a given vector field.
	`JB -> $`	Generates base vector field with given direction.
	`(PI, L NNI) -> $`	Generates base vector field in direction of a derivative.
	`PI -> $`	Generates base vector field in direction of a dependent variable.
		Generates base vector field in direction of an independent variable.
		Returns a list of the directions of the base vectors where the vector fields has non-vanishing coefficients.
		vector field to a function.
		e derivative of a given vector field her vector field.
		on $\mathcal{E}$ to a field on the
		n list of

alpha	(NNI, L NNI) -> L NNI	Computes the Cartan characters for a differential equation of given order from the $\beta^{(k)}_{q}$.
alphaHilbert	UP("r",FI) -> L NNI	Compute the Cartan characters for a given Hilbert polynomial.
arbFunctions	(NNI, I, L NNI) -> L I	Uses the Cartan characters to compute the number of arbitrary functions of a fixed differentiation order for a differential equation of given order.
	NNI, NNI, NNI) > NNI	Computes a bound $\hat{q}(n,m,q)$ for the number of prolongations needed to render a symbol involutive.
		Completes a given differential equation to an involu- one. No result is returned, but information on n process is displayed. The amount of ing setOutput.
		eturns a record containing ng set of equations, e number of on

*	(M D, $) -> $	Left multiplication of a sparse matrix with a usual matrix.
*	(M F D, $) -> $	Left multiplication of a sparse matrix with a usual matrix over the quotient field of D. Available only if D belongs to IntegralDomain.
llIndices	$ -> L C	Yields a list of all indices used to label columns.
ndRow!	($, ROWREC) -> Void	Adds a new row as last row
ols!	$ -> Void	Removes columns containing only zeros. This effects, however, basically only the value of allIndices.
M D		Coerces a matrix from SEM to the usual matrix type.
->		Adds a new row as first row. Argument is changed destructively.
		elds a copy of a matrix.
		the indicated row
		entry in the given row and the column ndex.
		sisting of the indicated
		ces. It is as- e smaller

ansatz	() -> VF	Yields the most general ansatz for a symmetry generator.
detSys	(L JBE1, L JB, VF) -> L JBE1	Computes the determining system for a given system. The second argument contains a list of derivatives for which the equations can be solved. If it is omitted the leading derivatives are used. The third argument contains an ansatz for the generators. It can also be omitted. ansatz() is then used.

-> L JBL2 | Retracts equations to linear ones, if possible.

JB, | Computes the determining system for conditional

Symmetries. The meaning of the arguments is as

rs.

a between the different jet bundles

mmetryAnalysis.

et variable in the old coordinates into

new ones.

the old coordinates into

rmation.

a given matrix.

he two