[SADE] A Maple package for the Symmetry Analysis of Differential Equations

Description of package commands

Tarcísio M. Rocha Filho* & Annibal Figueiredo
Mathematical Physics Group
Instituto de Física &
International Center for Condensed Matter Physics
Universidade de Brasília, CP: 04455, 70919-970 - Brasília, Brazil
*marciano@fis.unb.br

Command name: `liesymmetries`
Feature: Obtains Lie symmetry generators.
Calling sequence: `liesymmetries(equations,unknowns,options)`
Parameters:

`equations` - a set of differential equations.

`unknowns` - list of unknown functions in `equations`.

`options` - optional arguments: `determining` - returns only the determining system. `involutive` - reduces the determining system to the involutive form. `parameter = paramset` - computes the generators with conditions on the free parameters specified in the set `paramset`.

Command name: `ncsymmetries`
Feature: Obtains nonclassical symmetry generators.
Calling sequence: `ncsymmetries(equations,unknowns,options)`
Parameters:

`equations` - a set of differential equations.

`unknowns` - list of unknown functions in `equations`.

`options` - optional arguments: `determining` - returns the determining systems. `involutive` - the determining system is first reduced to involutive form. `case=`n, with n integer, only the case with $\theta_n = 1$, $\theta_i = 0$ $(i < n)$ is considered. If case=0 then the full set of determining equations is given, with no special values for θ_n or θ_i. `builtin` - solves the determining system using the MAPLE builtin command `pdsolve`. `default_parameters` - the determining system is solved using default parameters reducing CPU time, although the system may not be completely solved.

Command name: `LBsymmetries`
Feature: Obtains Lie-Bäcklund symmetry generators.
Calling sequence: `LBsymmetries(equations,unknowns,options)`
Parameters:

`equations` - a set of differential equations.

`unknowns` - list of the unknown functions in `equations`.

`option` - optional arguments: `determining` - returns the determining system. `involutive` - the determining system is first reduced to the involutive form.

Command name: `lindsolve`
Feature: Solves a linear overdetermined system of PDE's.
Calling sequence: `lindsolve(equations,unknowns)`
Parameters:
`equations` - a set of linear partial differential equations.
`unknowns` - list of the unknown functions in `equations`.

Command name: `nonlindsolve`
Feature: Solves a nonlinear overdetermined system of PDE's.
Calling sequence: `nonlindsolve(equations,unknowns)`
Parameters:
`equations` - a set of linear partial differential equations.
`unknowns` - list of the unknown functions in `equations`.

Command name: `casimir_invariant`
Feature: Computes the Casimir invariants of a set of generators.
Calling sequence: `casimir_invariant(generators,depvars,indepvars,order)`
Parameters:
`generators` - a set of generators.
`depvars` - list of the dependent variables.
`indepvars` - list of independent variables.
`order` - list defining the order in the Casimir invariants of the derivatives of dependent variables in the following format: $[[n_{11}, n_{12}, \ldots, n_{1K}], \ldots, [n_{M1}, n_{M2}, \ldots, n_{MK}]]$, where n_{ij} is the order of the highest derivative of the i-th dependent variable in `depvars`, with respect to the j-independent variable in `indepvar`.

Command name: `ansatz`
Feature: Applies a rule (ansatz) to the determining system and solves the resulting equations.
Calling sequence: `ansatz(substitutions,functions)`
Parameters:
`substitutions` - a set of substitutions to replace in the determining equations.
`functions` - set of new unknown functions in the ansatz.

Command name: `noether`
Feature: Computes Nöther conserved quantities from a lagrangian.
Calling sequence: `noether(lagrangian,functions,generator)`
Parameters:

`lagrangian` - the lagrangian function.
`functions` - list of variables in the lagrangian.
`generator` - optional argument: a single generator - returns the associated first-integral or conserved current.

Command name: `equivalence`
Feature: Obtains the most generic form of a class of equations admitting a symmetry algebra.
Calling sequence: `equivalence(equations,generators,functions)`
Parameters:

`equations` - a generic form for a class of equations.
`generators` - a set of symmetry generators.
`functions` - the set of undetermined functions in `equations`.

Command name: `comm`
Feature: Commutator of two linear operators (generators).
Calling sequence: `comm(generator1,generator2,variables)`
Parameters:

`generator1` - a generator written in SADE notation.
`generator2` - a generator written in SADE notation.
`variables` - a set or list of variables used in the generators.

Command name: `com_table`
Feature: Commutation table of a set of infinitesimal generators.
Calling sequence: `com_table(generators,variables,name)`
Parameters:

`generators` - a list of symmetry generators.
`variables` - a set with the dependent and independent variables.
`name` - a name to be used to represent each generator in the table.

Command name: `AdjointRep`
Feature: Computes the table with the action of adjoint maps on each generator of a Lie Algebra.
Calling sequence: `AdjointRep(generators,variables,name,parameter)`
Parameters:

`generators`	-	a list of symmetry generators.
`variables`	-	a set of dependent and independent variables.
`name`	-	a name to be used to represent each generator in the table.
`parameter`	-	a variable name for the adjoint Lie group parameter.

Command name: `StructConst`
Feature: Computes the array with the structure constants of a Lie algebra.
Calling sequence: `StructConst(generators,variables)`
Parameters:

`generators`	-	a list of symmetry generators.
`variables`	-	a set of dependent and independent variables.

Command name: `linear_rep`
Feature: Determines a linear operator defining a DE admitting a symmetry algebra.
Calling sequence: `linear_rep(operator,generators,variables,name)`
Parameters:

`operator`	-	a linear differential operator.
`generators`	-	a list of symmetry generators.
`variables`	-	a set of dependent and independent variables.
`name`	-	a name to be used for arbitrary constants or functions in the output.

Command name: `PDEreduction`
Feature: Obtains the reduced form of a PDE or a PDE system from a set of symmetry generators. If the original system has M independent variables and K symmetry generators are given, then the the reduced system depends on $M - K$ (transformed) independent variables.
Calling sequence: `PDEreduction(equations,unknowns,generators)`
Parameters:

`equations`	-	a set of differential equations.
`unknowns`	-	the set of unknown functions in `equations`.
`generators`	-	a set of symmetry generators.

Command name: `invariant_sol`
Feature: Obtains invariant solutions of a PDE or a system of PDE's using symmetry generators.
Calling sequence: `invariant_sol(equations,unknowns,generators)`
Parameters:

`equations` - a set of differential equations.
`variables` - the set of unknowns in `equations`.
`generators` - a set of symmetry generators.
`option` - optional arguments: if `pde_reduction`, returns only the transformation reducing the original system to a system of ODE's'. if `characteristic_equations`, returns the characteristic equations satisfying the given symmetries.

Command name: `issolvable`
Feature: Tests if a Lie algebra is solvable.
Calling sequence: `issolvable(generators,variables)`
Parameters:

`generators` - a set of generators.
`variables` - the set of dependent and independent variables in `generators`.

Command name: `canonical_basis`
Feature: Computes the canonical basis of a Lie algebra.
Calling sequence: `canonical_basis(generators,variables)`
Parameters:

`generators` - a set of generators.
`variables` - the set of dependent and independent variables.

Command name: `derived_subalg`
Feature: Computes the generators of the derived subalgebra of a Lie algebra.
Calling sequence: `derived_subalg(generators,variables)`
Parameters:

`generators` - a set generators.
`variables` - the set of dependent and independent variables.

Command name: `odesolver`
Feature: Solves an ODE by successive reductions using a solvable Lie algebra.
Calling sequence: `odesolver(equations,generators,unknowns,option)`
Parameters:

`equations`	-	a set of differential equations.
`generators`	-	a set symmetry generators.
`variables`	-	the set of unknowns in `equations`.
`option`	-	optional argument: `transformation` - returns only the transformation of variables that solves the system.

Command name: `cancoord`
Feature: Determines the canonical coordinates for a given generator.
Calling sequence: `cancoord(generator,vars1,vars2)`
Parameters:

`generator`	-	an infinitesimal symmetry generator.
`vars1`	-	list of variables in `generator`.
`vars2`	-	list of variables names for the canonical variables.

Command name: `reduce_ode_sist`
Feature: Reduces by one the dimension of a system of first order ODE's using a symmetry generator.
Calling sequence:
```
reduce_ode_sist(equations,generator,depvars1,
                indepvar1,depvars2,indepvar2)
```
Parameters:

`equations`	-	set of first order ODE's.
`generator`	-	a symmetry generator.
`depvars1`	-	set of the dependent variables in `equations`.
`indepvar1`	-	the independent variable in `equations`.
`depvars2`	-	set of names to be used as dependent variables in the reduced system.
`indepvar2`	-	name to be used as independent variable in the reduced system.

Command name: `ode_reduce_order1`
Feature: Reduces by one the order a a single ODE using a symmetry generator.
Calling sequence: `ode_reduce_order1(equation,generator,depvar1,`
`indepvar1,depvar2,indepvar2)`
Parameters:

`equation`	-	a first order ODE.
`generator`	-	a symmetry generator.
`depvar1`	-	dependent variable in `equation`.
`indepvar1`	-	independent variable in `equation`.
`depvar2`	-	name to be use as dependent variable in the reduced equation.
`indepvar2`	-	name to be use as independent variable in the reduced equation.

Command name: `ode_invsolution`
Feature: Obtains invariant solutions for a single ODE.
Calling sequence: `ode_invsolution(equation,function,generator):`
Parameters:

`equation`	-	a single first order ODE.
`function`	-	unknown in `equation`.
`generator`	-	a symmetry generator.
`option`	-	optional argument: if `reduction` returns the transformation for the solvable form.

Command name: `conserved`
Feature: Obtains the QP-invariants of a QP first order system.
Calling sequence: `conserved(equations,functions,parameters,order,options)`
Parameters:

`equations`	-	set of first order QP ODE's .
`functions`	-	list of unknowns in `equations`.
`parameters`	-	set of free parameters in `equations`.
`order`	-	order of the semi-invariants computed.
`options`	-	optional arguments: `Groebner` - a Gröbner basis computation is used to solve the polynomial system of determining equations. `positive` - the results are simplified to the positive orthant. `surfaces` - returns the defining equations for invariant hyper-surfaces.

Command name: `QPsymmetries`
Feature: Determines QP symmetry generators.
Calling sequence: `QPsymmetries(equations,functions,parameters,order)`
Parameters:
`equations` - set of first order QP ODE's .
`functions` - list of unknowns in `equations`.
`parameters` - set of free parameters in `equations`.
`order` - order of the semi-invariants computed.

Command name: `verif_if_inv`
Feature: Determines parameter values such that non-trivial QP first-integrals (i. e. with non-integer exponents) may exist.
Calling sequence: `verif_if_inv(equations,functions,parameters):`
Parameters:
`equations` - set of first order QP ODE's .
`functions` - list of unknowns in `equations`.
`parameters` - set of free parameters in `equations`.

Command name: `constlog`
Feature: Computes first-integrals of the form $P_1(x)+\log(x^\xi)$ and $P_2(x, \ln(x))$, with P_1 and P_2 polynomials and x representing all dependent variables.
Calling sequence: `constlog(equations,functions,parameters,order)`
Parameters:
`equations` - set of first order QP ODE's .
`functions` - list of unknowns in `equations`.
`parameters` - set of free parameters in `equations`.
`order` - order of the semi-invariants computed.

Table 1: Summary of package commands

Command	Purposes
`liesymmetries`	Computes Lie symmetry generators for a system of DE.
`ncsymmetries`	Computes the nonclassical symmetry generators of DE's.
`LBsymmetries`	Computes Lie-Bäcklund and contact symmetries
`lindsolve`	Solves a linear overdetermined system of PDE's.
`nonlindsolve`	Solves a non-linear overdetermined system of PDE's.
`casimir_invariant`	Computes the Casimir invariants of a set of generators.
`ansatz`	Applies an ansatz to the determining equations.
`noether`	Computes the Nöther conserved currents or first-integrals.
`equivalence`	Classify a class of equation from a symmetry algebra.
`comm`	Commutator of two operators.
`com_table`	Commutation table of a set of generators.
`AdjointRep`	Computes the table defining the action of adjoint maps on each generator.
`StructConst`	Determines the structure constants array for a Lie algebra.
`linear_rep`	Linear equation operator from a symmetry algebra.
`cancoord`	Determines the caconical coordinates for a given generator.
`PDEreduction`	Reduction of a PDE or a system of PDE's.
`invariant_sol`	Obtains the invariant solutions from symmetry generators.
`issolvable`	Tests if a Lie algebra defined by a set of generators is solvable.
`canonical_basis`	Computes the canonical basis of a Lie algebra.
`derived_subalg`	Computes the derived sub-algebra of a Lie algebra.
`odesolver`	Solution an ODE by successive reductions.
`reduce_ode_sist`	Reduction of dimension of an ODE system.
`ode_reduce_order1`	Order reduction of a single ODE.
`ode_invsolution`	Obtains invariant solutions for a single ODE.
`conserved`	QP-invariants and invariant hyper-surfaces of a QP system.
`QPsymmetries`	QP symmetry generators for a system of QP ODE's.
`verif_if_inv`	Determines parameter values for QP first-integrals.
`constlog`	First-integrals of QP systems of the form $P_1(x)+\log(x^\xi)$ and $P_2(x, \ln(x))$, with P_1 and P_2 polynomials.