A Maple-based Package for Computation of Conservation Laws, Symmetries and Invariant forms of DEs

Alexei F. Cheviakov

University of Saskatchewan, Saskatoon, Canada

ICIAM 2011
July 22, 2011

Outline

Outline

Consider a general DE system

$$R^\sigma[u] = R^\sigma(x, u, \partial u, \ldots, \partial^k u) = 0, \quad \sigma = 1, \ldots, N$$

with variables $x = (x^1, ..., x^n), \quad u = u(x) = (u^1, ..., u^m)$.

GeM software for Maple

Consider a general DE system

$$R^\sigma[u] = R^\sigma(x, u, \partial u, \ldots, \partial^k u) = 0, \quad \sigma = 1, \ldots, N$$

with variables $x = (x^1, ..., x^n), \quad u = u(x) = (u^1, ..., u^m)$.

Generates determining equations for:

- Symmetries (linear and nonlinear equations);
  - Point
  - Local
  - Approximate (Fushchich)
  - Approximate (Baikov-Ibragimov)
- Equivalence transformations (point)
- Conservation law multipliers (local, PDE)
- Integrating factors (local, ODE)

GeM software for Maple

Consider a general DE system

$$R^\sigma[u] = R^\sigma(x, u, \partial u, \dots, \partial^k u) = 0, \quad \sigma = 1, \dots, N$$

with variables $x = (x^1, ..., x^n)$, $\quad u = u(x) = (u^1, ..., u^m)$.

Generates determining equations for:

- Symmetries (linear and nonlinear equations);
  - Point
  - Local
  - Approximate (Fushchich)
  - Approximate (Baikov-Ibragimov)
- Equivalence transformations (point)
- Conservation law multipliers (local, PDE)
- Integrating factors (local, ODE)

Is used to directly find:

- Fluxes of a conservation law
- Global forms of a symmetry group
- Reduced form of a DE system with respect to a given point symmetry

Main idea for symmetry-like computations:

- Use symbols to replace
  - all variables,
  - derivatives.

$$\frac{\partial^2 U}{\partial x \partial t} \to \texttt{Utx}.$$

- Carry out subsequent differentiations internally. Preserve alphabetic order.

$$\frac{\partial}{\partial y}\texttt{Utx} = \texttt{Utxy}.$$

- Generate determining equations.
- Split determining equations with respect to higher derivatives not entering functions sought.
- Simplify split determining equations using `Maple rifsimp`.
- Solve using `Maple pdsolve` or by other means.

- **Main goals:** practical usability, flexibility, transparency.

Initial Program Sequence

Consider a sample PDE

$$u_t + a\, u u_{xx} + f(u_x) = 0, \quad u = u(x,t), \quad a = \mathrm{const}.$$

Example of input of an equation

- Use the module: `read("d:\gem31_8.txt"):`
- Declare variables:

```
gem_decl_vars(indeps=[x,t], deps=[U(x,t)],
    freeconst=[a],freefunc=[f(diff(U(x,t),x))]);
```

- Declare equation(s):

```
gem_decl_eqs([diff(U(x,t),t) + a*U(x,t)*diff(U(x,t),x,x)
    +f(diff(U(x,t),x))]=0,
    solve_for=[diff(U(x,t),t)]);
```

Outline

Consider **incompressible Euler equations**

$$E_1 = \mathrm{div}\, \mathbf{u} = 0, \qquad (E_2, E_3, E_4) = \mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla P = 0.$$

Point symmetries:

$$\begin{aligned}
t^* &= t + a\xi^t(\mathbf{x}, \mathbf{u}, P) + O(a^2), \\
\mathbf{x}^* &= \mathbf{x} + a\xi^{\mathbf{x}}(\mathbf{x}, \mathbf{u}, P) + O(a^2), \quad \mathbf{x} = (x, y, z), \\
\mathbf{u}^* &= \mathbf{u} + a\eta^{\mathbf{u}}(\mathbf{x}, \mathbf{u}, P) + O(a^2), \quad \mathbf{u} = (u, v, w), \\
P^* &= P + a\eta^P(\mathbf{x}, \mathbf{u}, P) + O(a^2).
\end{aligned}$$

Infinitesimal generator:

$$\mathrm{X} = \xi^t \frac{\partial}{\partial t} + \xi^{x^i} \frac{\partial}{\partial x^i} + \eta^{u^j} \frac{\partial}{\partial u^j} + \eta^P \frac{\partial}{\partial P}.$$

Determining equations:

$$\mathrm{X} E_i = 0 \quad \text{when} \quad E_i = 0, \quad i = 1, \ldots, 4.$$

Point and local symmetries

Consider incompressible Euler equations

$$E_1 = \mathrm{div}\, u = 0, \qquad (E_2, E_3, E_4) = u_t + (u \cdot \nabla)u + \nabla P = 0.$$

Commands for point symmetries (local: same way)

- Generate and split determining equations:

```
det_eqs:=gem_symm_det_eqs([ind,u(ind),v(ind),w(ind),P(ind)]):
```

- Get names of symmetry components: `sym_components:=gem_symm_components();`
- Simplify determining equations:

```
simplified_eqs:=DEtools[rifsimp](det_eqs1, sym_components,mindim=1);
```

- Solve for symmetry components:

```
symm_sol:=pdsolve(simplified_eqs[Solved]);
```

- Output symmetries: `gem_output_symm(symm_sol);`

- **Maple example**

Outline

Point transformation:

- Infinitesimal generator:

$$\mathrm{X} = \xi^i(x,u)\frac{\partial}{\partial x^i} + \eta^\mu(x,u)\frac{\partial}{\partial u^\mu}.$$

- Point transformation:

$$\begin{aligned} x^* &= f(x,u;\varepsilon) = e^{\varepsilon \mathrm{X}} x = x + \varepsilon \xi(x,u) + O(\varepsilon^2), \\ u^* &= g(x,u;\varepsilon) = e^{\varepsilon \mathrm{X}} u = u + \varepsilon \eta(x,u) + O(\varepsilon^2). \end{aligned}$$

- f, g are found from ξ, η through Lie's First Fundamental Theorem.

- **Maple example**

Outline

Symmetry reductions

Given:

- A PDE system with m dependent and n independent variables: $u(x)$, x.
- A symmetry X involving independent variables.

Symmetry reductions

Given:

- A PDE system with m dependent and n independent variables: $u(x)$, x.
- A symmetry X involving independent variables.

Symmetry reduction:

- Find m invariants $h^1(x,u),\ldots,h^m(x,u)$ (new dependent variables) that essentially involve u.
- Find $n-1$ other invariants $z^1(x,u),\ldots,z^{n-1}(x,u)$ (new independent variables).
- Find a "translation coordinate" z^n: $\quad \mathrm{X}\, z^n = 1$.
- Rewrite the given system in terms of z, $u(z)$ and drop dependence on z^n.

Given:

- A PDE system with m dependent and n independent variables: $u(x)$, x.
- A symmetry X involving independent variables.

Symmetry reduction:

- Find m invariants $h^1(x,u),\ldots,h^m(x,u)$ (new dependent variables) that essentially involve u.
- Find $n-1$ other invariants $z^1(x,u),\ldots,z^{n-1}(x,u)$ (new independent variables).
- Find a "translation coordinate" z^n: $\quad \mathrm{X}\, z^n = 1$.
- Rewrite the given system in terms of z, $u(z)$ and drop dependence on z^n.

Implementation:

- An additional Maple package IRT, with Andrey Olinov (M.Sc. student).
- Flexible, with much control on the choice of invariants.

- **Maple example**

Outline

Point symmetries of linear PDEs

Linear PDEs:

- Infinite number of point symmetries.
- Can add any solution of linear homogeneous equation.

Theorem (Bluman (1990))

- *Given: a linear scalar PDE* $L(x, u, \partial u, \ldots, \partial^k u) = F(x); \;\; x = (x^1, ..., x^n)$.
- *Symmetry generator:*

$$\mathrm{X} = \xi^i(x, u)\frac{\partial}{\partial x^i} + \eta(x, u)\frac{\partial}{\partial u}.$$

- *Then*

$$\frac{\partial \xi}{\partial u} = \frac{\partial^2 \eta}{\partial u^2} = 0, \quad i.e., \quad \mathrm{X} = \xi^i(x)\frac{\partial}{\partial x^i} + [f(x)u + g(x)]\,\frac{\partial}{\partial u}.$$

- *To get nontrivial symmetries: set* $g(x) = 0$.

Conjecture (cf. Ovsiannikov (1959))

- Given: a linear PDE system

$$L^{\sigma}(x, u, \partial u, \ldots, \partial^k u) = F^{\sigma}(x), \quad \sigma = 1, \ldots, N,$$

- Symmetry generator:

$$\mathrm{X} = \xi^i(x,u)\frac{\partial}{\partial x^i} + \eta^{\mu}(x,u)\frac{\partial}{\partial u^{\mu}}.$$

- Then

$$\frac{\partial \xi^i}{\partial u^{\nu}} = 0, \quad \frac{\partial^2 \eta^{\mu}}{\partial u^{\nu} u^{\lambda}} = 0, \quad i = 1, \ldots, n, \quad \mu, \nu, \lambda = 1, \ldots, m.$$

or

$$\mathrm{X} = \xi^i(x)\frac{\partial}{\partial x^i} + \left[f^{\mu}_{\nu}(x)u^{\nu} + g^{\mu}(x)\right]\frac{\partial}{\partial u^{\mu}}.$$

Conjecture (cf. Ovsiannikov (1959))

- Given: a linear PDE system

$$L^\sigma(x, u, \partial u, \ldots, \partial^k u) = F^\sigma(x), \quad \sigma = 1, \ldots, N,$$

- Symmetry generator:

$$\mathrm{X} = \xi^i(x, u)\frac{\partial}{\partial x^i} + \eta^\mu(x, u)\frac{\partial}{\partial u^\mu}.$$

- Then

$$\frac{\partial \xi^i}{\partial u^\nu} = 0, \quad \frac{\partial^2 \eta^\mu}{\partial u^\nu u^\lambda} = 0, \quad i = 1, \ldots, n, \quad \mu, \nu, \lambda = 1, \ldots, m.$$

or

$$\mathrm{X} = \xi^i(x)\frac{\partial}{\partial x^i} + \left[f^\mu_\nu(x)u^\nu + g^\mu(x)\right]\frac{\partial}{\partial u^\mu}.$$

- The conjecture is true in the "vast majority of cases".
- There are counterexamples (e.g. Cheviakov (2010)).
- The conjecture can be symbolically tested using rifsimp.
- Implemented in GeM.

- **Maple example**

Outline

Example: a Grad-Shafranov equation of Plasma Physics

$$\Phi_{rr} - \frac{1}{r}\Phi_r + \Phi_{zz} = r^2 A(\Phi) + B(\Phi),$$

A, B - arbitrary functions.

- Find all point transformations that preserve GS but may change forms of A, B.

Method:

- Treat functions A, B as additional dependent variables:

$$\begin{aligned} r^* &= r + a\xi^r(r, z, \Phi) + O(a^2), \\ z^* &= z + a\xi^z(r, z, \Phi) + O(a^2), \\ \Phi^* &= \Phi + a\eta^\Phi(r, z, \Phi) + O(a^2), \end{aligned}$$

$$\begin{aligned} A^*(\Phi^*) &= A(\Phi) + a\eta^A(A, B, \Phi) + O(a^2), \\ B^*(\Phi^*) &= B(\Phi) + a\eta^B(A, B, \Phi) + O(a^2). \end{aligned}$$

- Enforce correct dependence of $\eta^A(\Phi)$, $\eta^B(\Phi)$. Implemented in GeM.
- **Maple example**

Outline

Consider a general DE system

$$R^\sigma[u] = R^\sigma(x, u, \partial u, \ldots, \partial^k u) = 0, \quad \sigma = 1, \ldots, N$$

with variables $x = (x^1, ..., x^n), \quad u = u(x) = (u^1, ..., u^m)$.

A local conservation law:

A divergence expression equal to zero on solutions: $\mathrm{D}_i \Phi^i[u] = 0$.
[Fluxes/densities depend on local quantities.]

The Direct Construction Method (Anco, Bluman (1997, 2002ab)):

For arbitrary set of U's, seek local multipliers $\Lambda_\sigma[U]$, such that

$$\Lambda_\sigma[U] R^\sigma[U] \equiv \mathrm{D}_i \Phi^i[U].$$

Then on solutions $U = u$, one has a local conservation law

$$\mathrm{D}_i \Phi^i[u] \equiv \Lambda_\sigma[u] R^\sigma[u] = 0.$$

Euler Operators

An Euler Operator (with respect to each variable U^j):

$$\mathrm{E}_{U^j} = \frac{\partial}{\partial U^j} - \mathrm{D}_i \frac{\partial}{\partial U^j_i} + \cdots + (-1)^s \mathrm{D}_{i_1} \ldots \mathrm{D}_{i_s} \frac{\partial}{\partial U^j_{i_1 \ldots i_s}} + \cdots , \quad j = 1, ..., m.$$

Theorem

The equations

$$\mathrm{E}_{U^j} F(x, U, \partial U, \ldots) \equiv 0, \quad j = 1, \ldots, m$$

hold for arbitrary $U(x)$ if and only if

$$F(x, U, \partial U, \ldots) \equiv \mathrm{D}_i \Psi^i(x, U, \partial U, \ldots)$$

for some functions $\Psi^i(x, U, \partial U, \ldots)$, $i = 1, \ldots, n$.
[I.e., Euler Operators annihilates any divergence expression, and the converse also holds.]

Determining Equations for Multipliers:

$$\mathrm{E}_{U^j}(\Lambda_\sigma(x, U, \partial U, \ldots, \partial^l U) R^\sigma(x, U, \partial U, \ldots, \partial^k U)) \equiv 0, \quad j = 1, \ldots, m, \tag{1}$$

- Generate in Maple:

```
det_eqs:=gem_conslaw_det_eqs([x,t, U(x,t), diff(U(x,t),x),
    diff(U(x,t),x,x), diff(U(x,t),x,x,x)]):
```

- Reduce the overdetermined system using `rifsimp`.

Fluxes of conservation laws

Example of use of the GeM package for Maple for the KdV.

- Solve determining equations:

```
multipliers_sol:=pdsolve(simplified_eqs[Solved]);
```

- Obtain corresponding conservation law fluxes/densities:

```
gem_get_CL_fluxes(multipliers_sol, method=*****);
```

Methods for flux computation implemented in GeM:

- **Direct:** obtain fluxes from solving PDEs $\Lambda_\sigma[U]R^\sigma[U] \equiv \mathrm{D}_i\Phi^i[U]$.
- **Homotopy** integral formulas.
- Use of scaling or another **local symmetry**.

Example 1: KdV

$$u_t + uu_x + u_{xxx} = 0, \quad u = u(x,t).$$

Example 2: Surfactant dynamics equations

u: velocity; Φ: level set function, c: surfactant concentration.

$$\frac{\partial u^i}{\partial x^i} = 0,$$

$$\Phi_t + \frac{\partial (u^i \Phi)}{\partial x^i} = 0,$$

$$c_t + u^i \frac{\partial c}{\partial x^i} - c(\delta_{ij} - n_i n_j)\frac{\partial u^i}{\partial x^j} - \alpha(\delta_{ij} - n_i n_j)\frac{\partial}{\partial x^j}\left[(\delta_{ij} - n_i n_k)\frac{\partial c}{\partial x^k}\right] = 0,$$

$$\mathbf{n} = -\frac{\nabla \Phi}{\|\nabla \Phi\|}.$$

Outline

Conclusions

Results

- Symmetry-related computations can be automated using Maple:
  - Symbolic generation of determining equations.
  - Subsequent use of Rif and integrators.
- Various other related routines are contained in GeM.
- User can access all structures that are computed, e.g.,
  - split and unsplit determining equations;
  - components of prolonged symmetry generators;
  - routines for symbolic differentiation, divergence, infinitesimal generator application, etc.
- Software can handle rather complicated cases.

Conclusions

Results

- Symmetry-related computations can be automated using Maple:
    - Symbolic generation of determining equations.
    - Subsequent use of Rif and integrators.
- Various other related routines are contained in GeM.
- User can access all structures that are computed, e.g.,
    - split and unsplit determining equations;
    - components of prolonged symmetry generators;
    - routines for symbolic differentiation, divergence, infinitesimal generator application, etc.
- Software can handle rather complicated cases.

Future work:

- Development of GeM: mostly application- and request-driven.
- A bullet-proof general procedure for conservation law fluxes?
- Parallelization?

Bluman, G.W., Cheviakov, A.F., and Anco, S.C. (2010).
Applications of Symmetry Methods to Partial Differential Equations.
Springer: Applied Mathematical Sciences, Vol. 168.

Anco, S.C. and Bluman, G.W. (1997).
Direct construction of conservation laws from field equations. *Phys. Rev. Lett.* **78**, 2869–2873.

Anco, S.C. and Bluman, G.W. (2002a,b).
Direct construction method for conservation laws of partial differential equations. Parts I, II. *Eur. J. Appl. Math.* **13**, 545–566.

Cheviakov, A.F. (2007).
GeM software package for computation of symmetries and conservation laws of differential equations. *Comput. Phys. Commun.* **176**, 48–61.

Cheviakov, A.F. (2010).
Symbolic Computation of Local Symmetries of Nonlinear and Linear Partial and Ordinary Differential Equations. *Math.Comput.Sci.* **4**, 203-222.