Proceedings of 10th International Conference in MOdern GRoup ANalysis 2005, 64–70

SYM: A New Symmetry — Finding Package for *Mathematica*

Stelios DIMAS and Dimitri TSOUBELIS

Department of Mathematics, University of Patras, Greece
E-mail: *spawn@math.upatras.gr, tsoubeli@math.upatras.gr*

A new package for computing the symmetries of systems of differential equations using *Mathematica* is presented. Armed with adaptive equation solving capability and pattern matching techniques, this package is able to handle systems of differential equations of arbitrary order and number of variables with the least memory cost possible. By harnessing the capabilities of Mathematica's front end, all the intermediate mathematical expressions, as well as the final results apear in familiar form. This renders the package a very useful tool for introducing the symmetry solving method to students and non-mathematicians.

1 Introduction

The effectiveness of the method of symmetry analysis of differential equations first introduced by Sophus Lie is well established. The success of Lie's method is partly due to the fact that it allows one to find the symmetries of a given (system of) equation(s) algorithmically. However, as the number of variables and/or equations increases, the pertinent calculations become unmanageable. On the other hand, complex systems involving a large number of independent variables are frequently met in practice, e.g. in all areas of theoretical and applied physics. The Einstein field equations of general relativity and the Navier–Stokes equations of hydrodynamics could be cited as representative examples. In all such cases, the huge amount of calculations involved in applying the symmetry method render the use of computer algebra programs imperative.

In recent years, several symmetry–finding packages have been developed [1]. Most of them are based on the widely used computer algebra systems (CAS), such as REDUCE [2], MACSYMA, Maple and Mathematica [3]. The functionality of the above packages varies greatly. Some of them are effective only for differential equations of polynomial form. Others give only the determining equations in a reduced form and, then, the user must solve the latter interactively, at best. In any case, most of the packages developed so far fare well in practice only for determining the Lie point symmetries of scalar equations.

The purpose of this presentation is to introduce SYM, a new package for computing symmetries using Mathematica. SYM's main advantage over its predecessors is twofold. First, it provides the user with an easy to comprehend interface. This has been made possible by hiding effectively the cumbersome and awkward way the CAS itself represents mathematical expressions. In particular, all the expressions appearing in both the input and output of the package are represented in the familiar form encountered in the mathematical literature. Moreover, SYM is distinguished by its ability to handle and calculate the symmetries of complex systems of differential equations efficiently and without much intervention from the user. This has been achieved by making use of the powerful programming language of Mathematica. As a result the package is fast, reliable, and consumes less memory.

2 The *Sym* Package

The fundamental characteristic of SYM is its modularity. This means that it is based on a specific set of functions which are employed in the symmetry analysis of a given equation. They are functions defined using the well known algorithms of symmetry analysis which stems from Sophus Lie's theory ([5–8]). In this section we give some further details regarding the features of the program and a few examples that illustrate its effectiveness.

2.1 Main Features

The basic functions that any symmetry finding package has to perform are [4, 5]:

(i) To obtain the determining equations,

(ii) To reduce and simplify the system of determining equations, and

(iii) To integrate this overdetermined system

Besides complying to the above guidelines, SYM carries the following features.

- Every infinitesimal generator and its prolongation are defined and used as operators. Hence, the action of an infinitesimal generator on any algebraic or differential equation can be easily manipulated. This is accomplished using the command `X[n,{x,y},{u}]` which turns the n-th extension of the infinitesimal operator $\xi_1(x,y,u)\partial_x+\xi_2(x,y,u)\partial_y+\eta(x,y,u)\partial_u$ into a "a pure function". Examples where this feature can be exploited are the analysis of the invariant surface condition and the supplementary equations involved in the symmetry analysis of an initial–boundary problem. Likewise, the command `X[n,{x,y},{u,h},2]` defines as a pure function the n-th extension of the infinitesimal generator in characteristic form:

$$\mathcal{Q}_1(x,y,u,h,u_x,u_y,u_{xy},u_{xx},u_{yy},h_y,h_{xy},h_{xx},h_{yy})\partial_u$$

$$+\mathcal{Q}_2(x,y,u,h,u_x,u_y,u_{xy},u_{xx},u_{yy},h_y,h_{xy},h_{xx},h_{yy})\partial_h,$$

  which is needed in the investigation of generalized symmetries.

- The structural elements of the equations to be analyzed are automatically pinpointed and characterized. This is attained by making the program look at a differential equation in a human like fashion, using commands like `CharacterizeEq[*]`. The latter produces automatically several features of equation, such as its order, the independent and dependent variables, etc. This feature minimizes the input required, restricting it to the differential equation or the system of such equations under study, only. The above commands not only facilitate the substitutions needed in the process of automatically solving the linearized symmetry condition, but they render these substitutions easy to materialize in the case the user has to solve the above system interactively.

- An intelligent integrator of the system of overdetermined equations, which is incorporated in the fundamental command `SolveOverdeterminedEqs`. Enhancing Mathematica's internal one, SYM's differential solver mimics the human behavior by following a novel algorithm we call "Seek&Solve": it locates the appropriate equation to solve, substitutes the solution of the latter to the remaining equations and, after making the necessery simplifications, it repeats the previous cycle. Thanks to the various rules and tactics incorporated in the solver, the program will adapt its solving strategy to the system at hand. It terminates only when the complete solution is achieved, or when the remaining equations are not solvable. In this connection, we stress that the solver can deal with systems which include equations of non polynomial type. All possible differential constraints on arbitrary functions contained in the solutions are given explicitly. In addition, the package provides the option of printing all the steps followed in obtaining the solution. This feature allows the program's user to check all the intermediate steps at any time.

- Additional functions for manipulating the system's symmetries are included. SYM gives all the generators of the one–parameter subgroups, their commutator table and the structure constants of the corresponding algebra.

- All intermediate and final expressions are presented in a compact and elegant fashion. More specifically, by taking advandage of the expression masking capabilities of Mathematica, SYM presents both the equations to be solved as well as the intermediate and final results in the familiar form that one encounters in the mathematical literature. Moreover, these familiar expressions can be manipulated freely by the user himself.

2.2 Illustrative Examples

The package has been tested against a variety of differential equations, especially systems, from various sources [5–8]. It has also been tested by the interactive derivation of conditional symmetries — both point and generalized, of several equations of research interest. The following are characteristic examples of the equations against which SYM has been tested. The last one has been considered, up to now, as the benchmark for symmetry–finding packages.

(i) The modified Kadomtsev–Petviashvili equation

$$3u_{yy} - 4u_{xt} - 6u_y u_{xx} - 6u_x^2 u_{xx} + u_{xxxx} = 0 \tag{1}$$

(ii) The generalization of the Ernst equation derived in [9]

$$\begin{aligned}
&\partial_u \left(A_v + \frac{A^2}{\rho} U_v(u,v) + m\frac{A}{\rho} \right) + \partial_v \left(A_u + \frac{A^2}{\rho} U_u(u,v) - n\frac{A}{\rho} \right) = 0,\\
&\rho = \frac{1}{2}(v-u), \quad A = \frac{1}{2}\left(2\rho \frac{U_{uv}}{U_u U_v} + \frac{n}{U_u} - \frac{m}{U_v} \right).
\end{aligned} \tag{2}$$

(iii) The Einstein vacuum equations for the Bondi metric [10]

$$\begin{aligned}
\beta_r &= \frac{r\gamma_r^2}{2},\\
U_{rr} &= \frac{2e^{-2\gamma(u,r,\theta)}}{r^3} \Big(-2e^{2\beta(u,r,\theta)}\beta_\theta - 2e^{2\gamma}r^2U_r + e^{2\gamma}r^3U_r\beta_r\\
&\quad - 2re^{2\beta}\gamma_r \cot\theta + 2e^{2\beta}r\gamma_r\gamma_\theta - e^{2\gamma}r^3U_r\gamma_r + e^{2\beta}r\beta_{r\theta} - e^{2\beta}r\gamma_{r\theta} \Big),\\
\beta_{\theta\theta} &= -\frac{1}{4}e^{-4\beta} \Big(-4e^{4\beta} - 8e^{2(\beta+\gamma)}rU(u,r,\theta)\cot\theta - 8e^{2(\beta+\gamma)}rU_\theta +\\
&\quad 4e^{4\beta}\beta_\theta\cot\theta + 4e^{4\beta}\beta_\theta^2 - 12e^{4\beta}\gamma_\theta\cot\theta - 8e^{4\beta}\beta_\theta\gamma_\theta + 8e^{4\beta}\gamma_\theta^2 -\\
&\quad 4e^{4\beta}\gamma_{\theta\theta} - 2e^{2(\beta+\gamma)}r^2U_r\cot\theta + e^{4\gamma}r^4U_r^2 + 4e^{2(\beta+\gamma)}V_r -\\
&\quad 2e^{2(\beta+\gamma)}r^2U_{r\theta} \Big),\\
\gamma_{\theta\theta} &= -e^{-2\beta} \Big(3e^{2\gamma}rU\cot\theta + e^{2\gamma}rU_\theta - 2e^{2\beta}\beta_\theta\cot\theta + 3e^{2\beta}\gamma_\theta\cot\theta\\
&\quad + e^{2\beta} - 2e^{2\gamma}rU\gamma_\theta + 2e^{2\beta}\beta_\theta\gamma_\theta - 2e^{2\beta}\gamma_\theta^2 + e^{2\gamma}r^2U_r\cot\theta - e^{2\gamma}V_r\\
&\quad - e^{2\gamma}r^2\gamma_\theta U_r - e^{2\gamma}r^2U\cot\theta\gamma_r + e^{2\gamma}V(u,r,\theta)\gamma_r - e^{2\gamma}r^2U_\theta\gamma_r\\
&\quad + e^{2\gamma}rV_r\gamma_r - 2e^{2\gamma}r^2U\gamma_{r\theta} + e^{2\gamma}rV\gamma_{rr} - 2e^{2\gamma}r\gamma_u - 2e^{2\gamma}r^2\gamma_{ur} \Big)
\end{aligned} \tag{3}$$

(iv) The Magneto–Hydro–Dynamics equations

$$\begin{aligned}
&\rho_t = -\nabla\cdot(\rho(x,y,z,t)\vec{v}),\\
&\vec{v}_t = -(\vec{v}\cdot\nabla)\vec{v} - \frac{1}{\rho}\left(\nabla\left(p(x,y,z,t) + \frac{1}{2}\vec{H}^2(x,y,z,t) \right) - (\vec{H}\cdot\nabla)\vec{H} \right),\\
&\vec{H}_t = (\vec{H}\cdot\nabla)\vec{v} - (\vec{v}\cdot\nabla)\vec{H} - \vec{H}\nabla\cdot\vec{v},\\
&\nabla\cdot\vec{H} = 0, \quad p_t = -kp(\nabla\cdot\vec{v}) - (\vec{v}\cdot\nabla)p.
\end{aligned} \tag{4}$$

The Lie point symmetries of the above equations were obtained using SYM's `ClassicalSymmetries[]` function. In the table below we present the time and the amount of physical memory needed for the calculation. The PC used in the test was a Pentium IV laptop at 3.2GHz with 1GB of physical memory.

equation	time	physical memory
1	6.8 sec	6 MB
2	95.8 sec	374 MB
3	25.1 min	269 MB
4	19.4 min	32 MB

In way of comparison, we first mention that MathLie, the symmetry–finding package for Mathematica developed by G. Baumann [11], wasn't able to give *non–interactively* even the determining equations for examples (ii)–(iv). On the other hand, the MACSYMA based package SYMMGRP.MAX took 50 minutes of CPU time on a Digital VAX 4500 with 64MB of RAM for deriving only the (222) determining equations of example (iv).

The Lie point symmetries of the equations in examples (i) and (iv) are well documented [4]. Therefore we restrict ourselves to presenting the symmetry generators of the equations in examples (ii) and (iii). They are given by

$$X_1 = \partial_u + \partial_\upsilon, \quad X_2 = u\partial_\upsilon + \upsilon\partial_\upsilon, \quad X_3 = \partial_U, \quad X_4 = U\partial_U, \quad X_5 = U^2\partial_U$$

and

$$\begin{aligned}
X_1 &= -r\partial_r - 2V\partial_V + \frac{1}{2}\partial_\beta + \partial_\gamma, \quad X_2 = 2r\partial_r + 4V\partial_V + \partial_\beta, \\
X_{f_1} &= f_1(u)\partial_u - Uf_{1u}\partial_U - Vf_{1u}\partial_V - \frac{f_{1u}}{2}\partial_\beta, \\
X_{f_2} &= -\frac{r}{2}(f_2\cot\theta x + f_{2\theta})\partial_r + f_2(u,\theta)\partial_\theta + \\
&\quad \left(Uf_{2\theta} + f_{2u} + \frac{e^{2(\beta-\gamma)}}{2r}(f_{2\theta\theta} + f_{2\theta}\cot\theta - f_2\csc^2\theta)\right)\partial_U + \\
&\quad \big(r^2(Uf_{2\theta\theta} + f_{2u}\cot\theta + f_{2u\theta}) + (r^2U\cot\theta - V)f_{2\theta} - \\
&\quad (r^2U\csc^2\theta + V\cot\theta)f_2\big)\,\partial_V + \frac{f_2\cot\theta + f_{2\theta}}{4}\partial_\beta + \frac{f_2\cot\theta - f_{2\theta}}{2}\partial_\gamma,
\end{aligned}$$

respectively.

3 Applications in Education

Because of the familiar way it represents mathematical expressions, its easy to use interface and modular structure, SYM can be used effectively in courses on the symmetry analysis of differential equations. By using it, students can become familiar with the fundamental notions of symmetry analysis much more easily.

Because it presents the symmetry construction process in a step by step fashion and allows students to experiment on their own. In addition, the package can be exploited in the context of webMathematica. More specifically, everyone with an internet access can use SYM for getting introduced to modern group analysis of differential equations, without having to own the actual CAS.

4 Future Additions

The symmetry–finding package presented in this talk needs to be further developed and completed. The following are among the additions that would make SYM even more effective:

- High–level comands that would make it able to automatically calculate conditional, non–local and discrete symmetries,
- Tools for the construction of recursion operators and master symmetries,
- Functions concerning various aspects of the corresponding Lie algebras, such as their solvability, the optimal system etc., and the group classification of solutions.
- Differential algebra algorithms which determine the system of determining equations and specify its solution space [12, 13]

Acknowledgements

We thank European Social Fund (ESF), Operational Program for Educational and Vocational Training II (EPEAEK II) and particularly the program *Irakleitos*, for funding the above work. We would also like to thank the University of Patras and particularly the program *Karatheodory* 2001 for funding the above work during its initial stage of development.

[1] W. Hereman, Symbolic software for Lie symmetry analysis. CRC Handbook of Lie Group Analysis of Differential Equations. Vol 3: New Trends in Theoretical Developments and Computational Methods, ed. N.H. Ibragimov, 367–413, CRC Press, Boca Raton, 1996.

[2] A.C. Hearn, *REDUCE User's Manual Version 3.5*, RAND publication CP78(Rev. 10/93), Santa Monica California, 1993.

[3] Wolfram Reasearch, Inc., *Mathematica, Version 5.0*, Wolfram Reasearch, Inc., Illinois, 2003.

[4] N.H. Ibragimov (ed), *CRC Handbook of Lie Group Analysis of Differential Equations. Vol 3: New Trends in Theoretical Developments and Computational Methods*, CRC Press, Boca Raton, 1996.

[5] P.E. Hydon, *Symmetry Methods for Differential Equations: A Beginners's Guide*, Cambridge University Press, U.S.A. , 2000.

[6] N.H. Ibragimov (ed), *CRC Handbook of Lie Group Analysis of Differential Equations. Vol 1: Symmetries, Exact Solutions, and Conservation Laws*, CRC Press, Boca Raton, 1994.

[7] N.H. Ibragimov (ed), *CRC Handbook of Lie Group Analysis of Differential Equations. Vol 2: Applications in Engineering and Physical Sciences*, CRC Press, Boca Raton, 1995.

[8] P.J. Olver, *Applications of Lie Groups to Differential Equations 2nd Ed.*, Springer–Verlag, New York, 1993.

[9] A. Tongas, D. Tsoubelis, P. Xenitidis, A family of integrable nonlinear equations of hyperbolic type., *J. Math. Phys.* **42** (2001), 5762–5784.

[10] H. Bondi , M.G.J. van der Burg and A.W.K. Metzner, Gravitational waves in general relativity VII. Waves from axi–symmetric isolated systems. *Proc. Roy. Soc. A* **229** (1962), 21–52.

[11] G. Baumann, *Symmetry Analysis of Differential Equations with Mathematica*, Telos/ Springer, New York, 2000.

[12] E.L. Mansfield and P.A. Clarkson, Applications of the differential algebra package diffgrob2 to classical symmetries of differential equations. *J. Symb. Comp.* **23** (1997), 517–533.

[13] G. Carra'Ferro and V.P.Gerdt, Improved Kolchin-Ritt Algorithm, *Program. Comput. Software* **29** (2003), 84–87.