Euromath Bulletin
Vol. 1, No. 2, 1994

Review of Symbolic Software for the Computation of Lie Symmetries of Differential Equations

W. Hereman
Department of Mathematics and Computer Sciences
Colorado School of Mines
Golden, CO 80401, USA

Abstract

A survey of symbolic programs for the determination of Lie symmetry groups of systems of differential equations is presented. The purpose, methods and algorithms of symmetry analysis are briefly outlined. Examples illustrate the use of the software. Directions for further research and development are indicated.

1 Introduction

Marius Sophus Lie (1842–1899) [210] founded the theory of infinitesimal transformations [103], [104], [105], [106], [107], [108] and Lie groups. His work brought many ad–hoc methods of solving special classes of differential equations under a common conceptual umbrella.

Lie's infinitesimal method provides the most widely applicable technique to find closed form solutions of ordinary differential equations (ODEs). Standard solution methods for first–order or linear ODEs, can be characterized in terms of symmetries. For nonlinear ODEs, Lie's method, when it succeeds, provides a means of reducing the solution to a series of quadratures and can be implemented in symbolic programs [23].

Applied to partial differential equations (PDEs), the method leads to group–invariant solutions, conservation laws, invariant center manifolds in bifurcation theory [64], [65], [154], [187], etc.. Lie's original ideas greatly influenced the study of physically important systems of differential equations in classical mechanics, fluid dynamics, elasticity and many other applied areas. Later on, the concept of 'symmetry' evolved into one of the most explosive developments of mathematics and physics throughout the twentieth century.

The application of Lie groups methods to concrete physical systems involves tedious, but rather mechanical computations. Even the calculation of the continuous symmetry group of a modest system of differential equations is prone to fail if done with pencil and paper. Symbolic

packages are extremely well suited for such computations. Indeed, symbolic packages can often find the defining (or determining) equations of the Lie symmetry group, solve them in closed form, and go on to calculate the infinitesimal generators that span the Lie algebra of symmetries. The group–invariant solutions obtained via Lie's approach then provide insight into the physical models themselves, or into the accuracy of numerical and approximate treatments.

We start with a brief discussion of the methods and algorithms used in the computation of Lie symmetries. Next, we give a detailed review of symbolic packages which aid in the investigation of Lie symmetries for systems of differential equations. Simple examples illustrate the type of results that can be obtained with the software packages. To conclude, we indicate directions for future research and further development of symbolic software for Lie symmetries.

A large body of old and new literature exists on this topic; here we just refer to older books [4], [5], [14], [19], [27], [28], [35], [42], [50], [78], [83], [92], [100], [103], [104], [105], [106], [107], [108], [115], [116], [117], [129], [131], [168], more recent books [7], [10], [11], [20], [26], [45], [47], [53], [63], [65], [86], [87], [88], [89], [93], [101], [113], [123], [130], [132], [134], [135], [147], [148], [154], [155], [172], [176], [191], [196], [199], [200], [210], Ph.D theses [32], [71], [95], [169], and papers presenting the key concepts [6], [8], [15], [98], [125], [153], [164], [181], [189], [190], [192], [197], [198]. We highly recommend the special issues of *Acta Applicandae Mathematicae* on 'Symmetries of Partial Differential Equations' [190], now available in book form [191], and the series of volumes on 'Lie Group Analysis of Differential Equations', edited by Ibragimov [89]. Readers interested in the differential geometrical foundation of Lie symmetry analysis, including topics like supersymmetry, graded differential geometry, and gauge theories, may want to consult [64], [98], [99], [113], [193], [194], [196], and [73], [74] for related REDUCE algorithms.

In this review, we will sparsely address the computation of conservation laws, which through Noether's famous theorem [20], [151], is intrinsically connected with the investigation of variational symmetries. Indeed, recall that variational symmetries admitted by a Lagrangian system ––these are symmetries that leave the action integral invariant–– can be obtained by computing the Lie–Bäcklund symmetries of the corresponding Euler–Lagrange equations. Once the Lie–Bäcklund symmetries are obtained and the variational symmetries are singled out, Boyer's formulation of Noether's theorem can be used to calculate the conservation laws.

2 Purpose, Methods and Algorithms

2.1 Purpose

The classical 'Lie symmetry group of a system of differential equations' is a local group of point transformations, meaning diffeomorphisms on the space of independent and dependent variables, which map solutions of the system to solutions. Various other types of local symmetries [8] and nonlocal symmetries [17], [19], [20], [191] have been studied. The software reviewed in this paper attempts to compute the classical Lie–point symmetries. Several packages go beyond that in computing contact (or dynamical) and generalized (or Lie–Bäcklund) symmetries, nonclassical (or conditional) symmetries. Several programs could be modified to compute Cartan's dynamical symmetries [28], [176] and hidden symmetries.

Loosely speaking, contact symmetries are generalized symmetries of order one, i.e. the coefficients in the vector field include only first derivatives of the dependent variables.

The name Lie–Bäcklund symmetries, commonly used for generalized symmetries, is somewhat misleading for its connection with Bäcklund transformations [118], which constitute a set of case–specific equations that allow to transform one solution into another, whereas generalized symmetries are infinitesimal symmetries involving higher–order derivatives of the dependent variables. Among the several ways of computing Lie–Bäcklund symmetries, this paper focuses on the method proposed by Sophus Lie. For a discussion of the difference with other symmetry approaches, including other computer algebra methods, we refer to [58], [60], [69], [70], [211] and [212].

Conditional symmetries are found by the 'nonclassical method of group–invariant solutions', as introduced in 1969 by Bluman and Cole[18]. Further generalizations of the nonclassical method lead to the less practical *weak symmetries* [127], [128], and the method of differential constraints [123], [124].

Nonlocal (potential) symmetries [17], [20], [130], [191] for a system of PDEs are computed as follows. First, one replaces (one or more of) the PDEs in the given system by equivalent conserved forms; second, one introduces auxiliary potential variables; finally, one determines the point symmetries of the resulting auxiliary system of PDEs. The form of an infinitesimal generator then determines whether or not it defines a nonlocal symmetry. The technique of potential symmetries leads to interesting linearizations involving non–invertible mappings.

'Hidden' symmetries [2] show up when, for example, a higher–order ODE is rewritten in an equivalent system of first order equations. Indeed, if one analyzes the system of first order equations, rather than the single equation of higher–order, the class of symmetries (and consequently solutions) can be substantially enlarged[129], [130], [153]. Precise definitions and a discussion of internal and external symmetries are given in [8], [125].

2.2 Methods for Computing Lie Symmetries

There are three major methods to compute Lie symmetries. The first one uses prolonged vector fields, the second utilizes differential forms (wedge products) due to Cartan [28]. The third one uses the notion of 'formal symmetry' [21], [22], [114]. Although restricted to evolution systems with 2 independent variables, the latter method provides a very quick way to compute canonical generalized symmetries. Due to its limited scope and lack of space, we will not elaborate on that technique.

2.3 Prolonged Vector Fields

The first method is used in the algorithm or our program SYMMGRP.MAX[30] and in most of the other Lie symmetry packages. Instead of looking for the Lie group G, one looks for its Lie algebra L, realized by the vector field. From the Lie algebra of symmetry generators one can obtain the Lie group of point transformations upon integration of a system of first order characteristic equations (see Section 6.1). For notational simplicity, let us consider the case of Lie–point symmetries[19], [20], [123]. We follow the notations, terminology and method for

symmetry analysis used in [123].

We start with a system of m differential equations,

$$\Delta^i(x, u^{(k)}) = 0, \qquad i = 1, 2, \ldots, m, \tag{1}$$

of order k, with p independent and q dependent real variables, denoted by $x = (x_1, x_2, \ldots, x_p) \in \mathbb{R}^p$, $u = (u^1, u^2, \ldots, u^q) \in \mathbb{R}^q$. We stress that m, k, p and q are *arbitrary* positive integers. The partial derivatives of u^l are represented using a multi index notation, for $J = (j_1, j_2, \ldots, j_p) \in \mathbb{N}^p$, we put

$$u^l_J = \frac{\partial^{|J|} u^l}{\partial x_1^{j_1} \partial x_2^{j_2} \ldots \partial x_p^{j_p}} \quad , \tag{2}$$

where $|J| = j_1 + j_2 + \ldots + j_p$. Finally, let $u^{(k)}$ denote a vector whose components are all the partial derivatives of order 0 up to k of all the u^l.

The group transformations have the form $\tilde{x} = \Lambda_g(x, u)$, $\tilde{u} = \Omega_g(x, u)$, where the functions Λ_g and Ω_g are to be determined. Note that the subscript g refers to the group parameters. Instead of looking for a Lie group G, we look for its Lie algebra L, realized by vector fields of the form

$$\alpha = \sum_{i=1}^{p} \eta^i(x, u) \frac{\partial}{\partial x_i} + \sum_{l=1}^{q} \varphi_l(x, u) \frac{\partial}{\partial u^l}. \tag{3}$$

The problem is now reduced to finding the coefficients $\eta^i(x, u)$ and $\varphi_l(x, u)$. In essence, the computer constructs the k^{th} prolongation $\text{pr}^{(k)}\alpha$ of the vector field α, applies it to the system (1), and requests that the resulting expression vanishes on the solution set of (1).

That sounds straightforward, but the method involves tedious calculations since the length and complexity of the expressions increase rapidly as p, q, m and especially k increase. To stress that point, here are the details and the steps to be performed.

1. Construct the k^{th} prolongation of the vector field α in (3) by means of the formula

$$\text{pr}^{(k)}\alpha = \alpha + \sum_{l=1}^{q} \sum_{J} \psi^J_l(x, u^{(k)}) \frac{\partial}{\partial u^l_J}, \qquad 1 \le |J| \le k, \tag{4}$$

where the coefficients ψ^J_l are defined as follows. The coefficients of the first prolongation are

$$\psi^{J_i}_l = D_i \varphi_l(x, u) - \sum_{j=1}^{p} u^l_{J_j} D_i \eta^j(x, u), \tag{5}$$

where J_i is a $p-$tuple with 1 on the i^{th} position and zeros elsewhere, and D_i is the total derivative operator

$$D_i = \frac{\partial}{\partial x_i} + \sum_{l=1}^{q} \sum_{J} u^l_{J+J_i} \frac{\partial}{\partial u^l_J}, \qquad 0 \le \#J < \infty. \tag{6}$$

The higher–order prolongations are defined recursively as

$$\psi^{J+J_i}_l = D_i \psi^J_l - \sum_{j=1}^{p} u^l_{J+J_j} D_i \xi^j(x,u), \qquad |J| \ge 1. \tag{7}$$

2. Apply the prolonged operator $\mathrm{pr}^{(k)}\alpha$ to each equation $\Delta^i(x, u^{(k)})$ and require that

$$\mathrm{pr}^{(k)}\alpha[\Delta^i]|_{\Delta^j=0} = 0 \qquad i, j = 1, \ldots, m. \tag{8}$$

The meaning of condition (8) is that $\mathrm{pr}^{(k)}\alpha$ vanishes on the solution set of the originally given system (1). Precisely, this condition assures that α is an infinitesimal symmetry generator of the group transformation $\tilde{x} = \Lambda_g(x,u)$, $\tilde{u} = \Omega_g(x,u)$, i.e. that $u(x)$ is a solution of (1) whenever $\tilde{u}(\tilde{x})$ is one.

3. Choose, if possible, m components of the vector $u^{(k)}$, say $v^1, \ldots, v^m$, such that:

   a. Each v^i is equal to a derivative of a u^l $(l = 1, \ldots, q)$ with respect to at least one variable x_i $(i = 1, \ldots, p)$.

   b. None of the v^i is the derivative of another one in the set.

   c. The system (1) can be solved algebraically for the v^i in terms of the remaining components of $u^{(k)}$, which we denoted by w. Hence, $v^i = S^i(x,w), \quad i = 1, \ldots, m.$

   d. The derivatives of v^i, $v^i_J = D_J S^i(x,w)$, where $D_J = D_{j_1} D_{j_2} \ldots D_{j_p}$, can all be expressed in terms of the components of w and their derivatives, without ever reintroducing the v^i or their derivatives.

   The requirements in step (3) put some restrictions on the system (1), but for many systems the choice of the appropriate v^i is quite obvious. For example, for a system of evolution equations

$$\frac{\partial u^i}{\partial t}(x_1, \ldots, x_{p-1}, t) = F^i(x_1, \ldots, x_{p-1}, t, u^{(k)}), \qquad i = 1, \ldots, m, \tag{9}$$

   where $u^{(k)}$ involves derivatives with respect to the variables x_i but not t, an appropriate choice is clearly $v^i = \frac{\partial u^i}{\partial t}$.

4. Use $v^i = S^i(x,w)$ to eliminate all v^i and their derivatives from the expression (8), so that all the remaining variables are now independent of each other. It is tacitly assumed that the resulting expression is now a polynomial in the u^l_J.

5. Obtain the determining equations for $\eta^i(x,u)$ and $\varphi_l(x,u)$ by equating to zero the coefficients of all functionally independent expressions (monomials) in the remaining derivatives u^l_J.

In the above algorithm the variables x_i, u^l and u^l_J are treated as independent; the dependent ones are η^i, φ_l and Δ^i.

In summary, the result of implementing (4) is a system of linear homogeneous PDEs for η^i and φ_p, in which x and u are independent variables. These are the so–called *determining equations* for the symmetries of the system. Solving these by hand, interactively or automatically with a symbolic package, will give the explicit forms of the $\eta^i(x,u)$ and $\varphi_l(x,u)$.

The procedure, for which a rigorous discussion is given in [123], thus consists of two major steps: *deriving* the determining equations and *solving* them explicitly.

We note that in [16], Bluman proves theorems about the forms of admitted infinitesimal transformations, which cover a large classes of scalar ODEs and PDEs. In essence the theorems [20] imply that the coefficients of the vector field are either free or depend linearly on the dependent variable. Use of these theorems can significantly simplify the tedious work involved in setting up and solving the determining equations.

2.4 Differential Forms

A differential geometric approach to invariance groups and solutions of partial differential equations was presented by Harrison and Estabrook [79] in 1971. They showed how to derive infinitesimal symmetries using Cartan's exterior differential calculus. The isovector fields, which are the generators of geometric transformations with suitable algebraic invariance properties, are then used to obtain invariant solutions of several partial differential equations, including the heat equation, the Korteweg–de Vries equation and the vacuum Maxwell equations.

Briefly, this method to find infinitesimal symmetries proceeds as follows. The system of differential equations is formulated as a closed exterior differential system of two–forms. The underlying manifold is prolonged by new coordinates, the so–called prolonged variables. The exterior differential system is then prolonged by special one–forms in these prolonged variables. The condition on these added one–forms is that the prolonged exterior differential systems remains closed. This condition leads to the determining equations for the coefficients of the manifold.

For the mathematical formulation and several worked examples, the reader is referred to [29], [79], [75], [96]. It should also be noted that routines [44], [156], [195] for Cartan's exterior calculus [80] are included as standard packages of modern symbolic manipulation programs. For instance, CARTAN in the MACSYMA Share Library, EXCALC in REDUCE, etc..

2.5 Methods for Solving the Determining Equations

There are no algorithms to solve an arbitrary system of determining system, which consists of linear homogeneous partial differential equations for the coefficients of the vector field. Most integration algorithms are based on a set of heuristic rules [81], [96], [157], [160], [170], [175]. Most commonly the following rules are used.

1. Integrate single term equations of the form

$$\frac{\partial^{|I|} f(x_1, x_2, \ldots, x_n)}{\partial x_1{}^{i_1} \partial x_2{}^{i_2} \ldots \partial x_n{}^{i_n}} = 0, \tag{10}$$

where $|I| = i_1 + i_2 + \ldots + i_n$, to obtain the solution

$$f(x_1, x_2, \ldots, x_n) = \sum_{k=1}^{n} \sum_{j=0}^{i_k = 1} h_{kj}(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n)(x_k)^j. \tag{11}$$

Thus introducing functions h_{kj} with fewer variables.

2. Replace equations of type

$$\sum_{j=0}^{n} f_j(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n)(x_k)^j = 0, \tag{12}$$

by $f_j = 0 \; (j = 0, 1, \ldots, n)$. More generally, this method of splitting equations (via polynomial decomposition) into a set of smaller equations is also allowed when f_j are differential equations themselves, provided the variable x_k is missing.

3. Integrate linear differential equations of first and second order with constant coefficients. Integrate first order equations with variable coefficients via the integrating factor technique, provided the resulting integrals can be computed in closed form.

4. Integrate higher−order equations of type

$$\frac{\partial^n f(x_1, x_2, \ldots, x_n)}{\partial x_k{}^n} = g(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n), \tag{13}$$

n successive times to obtain

$$\begin{aligned} f(x_1, x_2, \ldots, x_n) &= \frac{(x_k)^n}{n!} \; g(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n) \\ &+ \frac{x_k^{n-1}}{(n-1)!} \; h(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n) \\ &+ \ldots \; + r(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n), \end{aligned}$$

where $h, \ldots, r$ are arbitrary functions.

5. Solve any simple equation (without derivatives) for a function (or a derivative of a function) provided both (i) it occurs linearly and only once, (ii) it depends on all the variables which occur as arguments in the remaining terms.

6. Explicitly integrate exact equations.

7. Substitute the solutions obtained above in all the equations.

8. Add differences, sums or other linear combinations of equations (with similar terms) to the system, provided these combinations are shorter than the original equations.

With these simple rules, and perhaps a few more, the determining system can often be drastically simplified. Amazingly, in many cases nothing beyond the above heuristic rules is needed to solve the determining equations completely. If that is not possible, after

simplification, the program will return the remaining unsolved differential equations for further inspection. In most programs, the user can then interactively simplify and fully solve the determining equations on the computer, thereby minimizing human errors.

Two other important topics tie in with the integration of the determining equations. They are (i) the computation of the size of the symmetry group in Section 5, and (ii) the transformation of the determining equations into standard (passive) form, discussed in Sections 5 and 7.

2.6 Lie Symmetry Algorithms

Ideally, a fully automated software package for Lie symmetries should consist of effective, powerful algorithms and fast procedures for:

(i) the derivation of the determining equations for large or complicated systems of equations;

(ii) the reduction of determining equations into so−called standard form;

(iii) finding the size of the symmetry group;

(iv) determining obvious symmetry generators by fast analysis;

(v) if not all the generators have been found, attempt simplification and integration of the determining equations to compute the generators.

Then the program should be able to execute the following steps in the order relevant to the specific application:

(a) calculation of commutator tables, based on the results of (i), (ii) and (iii);

(b) calculation of group invariant solutions.

The program should be able to handle: calculation of nonlocal (potential) symmetries [17], [20], [130], [191], calculation of nonclassical reductions (conditional symmetries) and resulting solutions, calculation of generalized symmetries, calculation of equivalent conservation laws. Furthermore, it should be able to accept systems with free unknown (classification) functions [109], [140], [170] as input.

3 Beyond Lie−point Symmetries

3.1 Contact and Generalized Symmetries

For the computation of generalized symmetries or Lie−Bäcklund symmetries [7], [123] the use of symbolic programs is even more appropriate. The procedure to determine symmetries is essentially the same as that for point symmetries, although the calculations are lengthier and more time consuming. In a generalized vector field, which still takes the form of (3), the functions η^i and φ_l may now depend on a finite number of derivatives of u, i.e.,

$$\alpha\in \sum_{i=1}^{p} \eta^i(x, u^{(k)}) \frac{\partial}{\partial x_i} \in \sum_{l=1}^{q} \varphi_l(x, u^{(k)}) \frac{\partial}{\partial u^l}. \tag{14}$$

If $k=1$ the generalized symmetry determines a classical contact symmetry and vice versa, at least in the case of one dependent variable. The even simpler case $k=0$, with $u^{(0)}\equiv u$, leads to point symmetries. Olver [123] discusses various possibilities to simplify the calculations, e.g. by putting the symmetries in evolutionary form, or by fixing the order of derivation on which the ηs and φs may depend.

3.2 Nonclassical or Conditional Symmetries

Recently, it has been shown that the 'nonclassical method of group–invariant solutions' originally introduced in[18], can determine new solutions of various physically significant nonlinear PDEs. Examples include the nonlinear Schrödinger (NLS) [36], the Boussinesq [102], the Kadomtsev–Petviashvili [41], the Fitzhugh–Nagumo [52], [122] equations, shallow water wave equations [40], and a class of nonlinear heat equations [38]. For e.g. the Boussinesq equation, the new reductions that follow from application of the nonclassical method, were discovered earlier with a direct algorithmic method [36], [37]. Recently, Olver [126] proved that both methods are equivalent in the case of fiber preserving transformations.

In contrast to true Lie symmetries, the transformations corresponding to nonclassical (or conditional) symmetries do not leave the differential equation invariant, neither do they transform all the solutions into other solutions. They merely transform a subset of solutions into other solutions.

Accounting for 'nonclassical symmetries', the program should automatically add the q invariant surface conditions [18], [124],

$$Q^l(x,u^{(1)})\equiv \sum_{i=1}^{p} \eta^i(x,u)\,\frac{\partial u^l}{\partial x_i}\equiv \varphi_l(x,u)\equiv 0, \quad l=1,\ldots,q, \tag{15}$$

and their differential consequences, to the system (1). However, the inclusion of nonclassical symmetries, and perhaps other types of symmetries as discussed in [8], [124], [125], requires solving systems of determining equations which are no longer linear. Consequently, new integration algorithms must be designed.

4 Review of Symbolic Software

In this section we review the existing Lie symmetry programs, classified according to the underlying symbolic system. In Table 1 we brought together the most modern software packages, together with information about developers and distributors.

4.1 REDUCE Programs

The well–documented program SPDE, developed by Schwarz [157], [158], [160], [161], [162], [163], [164], automatically derives and often successfully solves the determining equations with minimal intervention by the user. The package SPDE is distributed with the current version of REDUCE for various types of computers, ranging from PCs to CRAYs.

Based on Cartan's exterior calculus [28], [57], [83], Edelen [46], [47], [48], [49], Gragert [71] and Gragert *et al.* [75] did some pioneering work in using computer algebra systems to calculate the classical Lie symmetries of differential equations. More recently, Gragert[72] added a package for more general Lie algebra computations. Kersten [95], [96] further perfected the software package for the calculation of the Lie algebra of infinitesimal symmetries (including Lie–Bäcklund symmetries) of exterior differential systems.

Eliseev *et al.* [51] wrote code in REDUCE–2 to generate (but not solve) the system of determining equations for point and contact symmetries. Their paper discusses the algorithm and shows three worked examples. Fedorova and Kornyak [54], [56] generalized the algorithm to include the case of Lie–Bäcklund symmetries.

The interactive REDUCE program NUSY by Nucci [120], [121] generates determining equations for (classical, nonclassical, Lie–Bäcklund and approximate) symmetries and provides interactive tools for the user to solve them. The manual [120] gives a clear description of the various routines, with their scope and limitations, and has several worked examples.

The package CRACK by Wolf and Brand [206], [207], [208] solves overdetermined systems of differential equations with polynomial terms. It also contains code for decoupling, separating and simplifying PDEs. Integration of exact PDEs and differential factorizations are also possible. Furthermore, the program CRACK has tools, e.g. the function LIEORD, that can aid in the investigation of Lie symmetries of ODEs. With CRACK one can also construct the Lagrangian for a given second order ODE, and find first integrals via the integrating factor method. In attempting to solve standard ODEs, the program makes use of the REDUCE package ODESOLVE written by MacCallum [110]. The functions available within CRACK allow simplification and integration of linear homogeneous PDEs, beyond those derivable via symmetry analysis.

Sarlet and Vanden Bonne [152] offer specific procedures to assist in the computation of adjoint symmetries of differential equations, where determining equations of the same nature occur.

The newly developed program DIMSYM by Sherring and Prince [170] was inspired by Head's symmetry program LIE [81], in turn influenced by SPDE [160], [164], but is much larger and grew independent of it during development. It is capable of finding various types of symmetries (including conditional symmetries), isolating special cases, bringing the determining equations in normal form, and aiding in the solution of group classification problems. It attempts to determine the generators and allows to check whether or not the generators are correct. It allows the user to specify the dependence of the symmetry vector field coefficients manually, which is particularly practical if one wants to compute Lie–Bäcklund symmetries. An attractive feature of the package is that the solver for systems of linear homogeneous differential equations works for equations that were not necessarily obtained from symmetry analysis.

The latest newcomer in REDUCE is LIEPDE by Wolf [205], that finds Lie–point and contact symmetries of PDEs by deriving and solving a few simple determining equations, before continuing with the computation of the more complicated determining equations. This idea, which makes the program highly efficient, was used in Wolf's old FORMAC program [202], [203], [204], and is also implemented in the design of the feedback mechanism of SYMMGRP.MAX [30]. For solving the determining equations, LIEPDE makes use of modules of the package CRACK discussed above. The difference is that within LIEPDE the steps are carried out automatically, without intervention by the user. This approach is

particularly useful when applied to large systems of PDEs, or in the computation of higher–order systems, where space and memory limitations come into play.

4.2 MACSYMA Programs

Apart from an earlier version [31] of our program [30] and the interactive programs by Char [32], [33] and Rosencrans [150], which could aid with specific examples, there are half a dozen other MACSYMA–based symmetry programs.

In 1979, Char and McNamara [34] developed LIEPROC, a program for finding adiabatic invariants of simple Hamiltionian systems via the Lie transform.

Schwarzmeier and Rosenau [149], [167] made a program that calculates the determining equations in a simple form, but does not solve them automatically.

The MACSYMA version of the program SYMCON [183], which was originally written in muMATH, tries to compute Lie–point and generalized symmetries and their conservation laws. Vafeades later produced PDELIE [184], [185], [186], which is a drastically improved version of SYMCON [183]. The package PDELIE attempts to produce similarity solutions of ODEs, analyze PDEs with a multiplicative or additive scalar parameter, produce the commutator table and the structure constants of the Lie algebra. PDELIE also allows to compute the Noether conservation laws of variational systems.

PDELIE consists of several subroutines. The function SYM, produces the determining equations and the generators of the Lie group. The function PDE2 tries to find the invariants of the symmetry group. Using these invariants, it then dimensionally reduces the given differential equation. In cases where the reduced equation is an ODE, it tries to integrate explicitly, thus arriving at special similarity solutions of the original equation. The functions COMTAB and STCON perform computations with elements in the commutator table of structure constants of the Lie algebra. The function CON computes some of the densities of the Noether conservation laws of systems of variational type.

Just as PDELIE, the program SYM_DE by Steinberg [173], [174], [175] was recently added to the out–of–core library of MACSYMA. Steinberg's program computes infinitesimal symmetry operators and the explicit form of the infinitesimal transformations for simple systems. In cases where the program cannot automatically finish the computation, the user can intervene and, for instance, ask for infinitesimal symmetries of polynomial form. The program solves some (or all) of the determining equations automatically and if needed the user can (interactively) add extra information. Currently, Steinberg is working at the extension of his program so that it would include the calculation of generalized symmetries.

The program SYMMGRP.MAX written by Champagne *et al.* [30], [82], is a modification of an earlier package [31] that has been extensively used over the last eight years at the University of Montréal and elsewhere. It has been tested on hundreds of systems of equations and has thus been solidly debugged.

The flexibility within SYMMGRP.MAX and the possibility of using it interactively, allow the user to find the symmetry group of in principal arbitrarily large and complicated systems of equations on relatively small computers. For example, whenever the prolongation can be applied successfully to the complete system, or a subset thereof, it produces a list of determining equations. This list is free of trivial factors, duplication and differential redundancies.

To make SYMMGRP.MAX work for *large* systems of differential equations, the designers followed the path that would be taken in manual calculations. That is, obtain in as

simple a manner as possible the simplest determining equations, solve them and feed the information back to the computer. Partial information can be extracted very rapidly. For instance, one can derive a subset of the determining equations, e.g. those that occur as coefficients in the highest derivatives in the independent variables. These are usually single term equations, which express that the coefficients of the vectorfield are independent of some variables or depend linearly on some of the other variables.

A feedback mechanism allows the determining equations to be solved step by step on the computer, hence avoiding human error in the algebraic simplifications. Typically, users will provide information about the ηs and φs, as it becomes available from solving the determining equations step by step. The amount of interaction by the user will depend on the complexity of the system of differential equations and on the capacity of the computer used. A worked example showing the use of the feedback mechanism is given in[30]. Needless to say, with the feedback mechanism, the program SYMMGRP.MAX can also be used to verify previously calculated solutions of the determining equations and hence detect errors in the literature on the subject.

Although not designed for that purpose, the program SYMMGRP.MAX can be easily adapted to compute the determining equations corresponding to nonclassical symmetries [38], [39], [40].

4.3 MAPLE Programs

In [29], Carminati *et al.* present their program LIESYMM for creating the determining equations via the Harrison–Estabrook procedure. Within LIESYMM various interactive tools are available for integrating the determining equations, and for working with Cartans differential forms. Their program is independent of Donsigs differential forms package *difforms*, also available in Maple.

Khai Vu (Monash University, Melbourne) has translated Head's muMATH program LIE[81], discussed below, into Maple syntax. The Maple version of LIE is currently under testing and is therefore not yet released.

Hickman [85] offers a collection of Maple routines that aid in the computation of Lie–point symmetries, non–local symmetries, and Wahlquist–Estabrook type prolongations. The 'tools' for symmetry analysis include user–friendly procedures to enter names of variables, to create total derivatives, to generate and prolong vector fields, and to derive and partially solve determining equations.

4.4 MATHEMATICA Programs

Herod [84] developed the program MathSym for deriving the determining equations corresponding to Lie–point symmetries, including nonclassical (or conditional) symmetries.

Most recently, the packages *Lie.m* and *Baecklund.m* have been added by Baumann [12], [13] to MathSource, the Mathematica Program Library. Baumann's program *Lie.m* [12] follows the structure of our MACSYMA program SYMMGRP.MAX [30] closely. Users familiar with SYMMGRP.MAX will not have much problems withh*Lie.mm*. In contrast to SYMMGRP.MAX, the program *Lie.m* can handle transcendental functions in the input equations, and solve the determining equations automatically under the assumption that the coefficients in the vectorfield are of polynomial type in the dependent and indepen

dent variables and with a user–specified degree. If this 'ansatz' turns out to be too restrictive and therefore does not lead to the complete symmetry group, the user has the option to solve the determining equations interactively. Once the solution of the determining equations is obtained, the program can continue with the computation of the vector basis, ideals, and commutator table of the Lie algebra, its structure constants, Casimir operators and its metric tensor.

Baumann's package *Baecklund.m* [13] contains functions that attempt to compute generalized symmetries and invariants of ODEs and PDEs. When applied to second order ODEs, the program attempts to verify if the computed symmetries are of variational type. If so, the program calculates the corresponding invariants (integrals of motion). For the explicit calculations to be successful, quite often one has to specify that the coefficients in the vectorfield are polynomials in the coordinates and momenta. With this 'ansatz' one may not be able to obtain all the generalized symmetries, but one may successfully obtain explicit forms of invariants.

Recently, Bérubé and de Montigny (Université de Montréal, Montréal) have contacted the author, and mentioned that they are working on Lie–symmetry code in Mathematica. Finally, Coult (Carleton College, Northfield, Minnesota) developed a Mathematica program, temporarily called *symmgroup.m*, for the computation of the determining equations corresponding to Lie–point symmetries of a large class of differential equations (with polynomial terms).

4.5 AXIOM Programs

Schwarz [163] rewrote SPDE [162], [164] for use with AXIOM (formerly SCRATCHPAD II), a symbolic manipulation program developed by IBM.

4.6 muMATH Programs

The program LIE by Head [81] is based on version 4.12 of muMATH, but is self contained and runs on IBM compatible PCs. As a matter of fact, the program comes bundled with a limited version of muMATH. Head's program calculates and solves the determining equations automatically for single equations and systems of differential equations. LIE also computes the Lie vectors and their commutators. Interventions by the user are possible but are rarely needed. The source code of the program is available, including the heuristic routines that attempt to solve the determining equations. Due to the limitations of muMATH, the program LIE is bounded by the 256 KB of memory for program and workspace. For a program of limited size, LIE is remarkable in its achievements. The current version (e.g. 4.1) of LIE is freely available by FTP from various public domain software archives such as SIMTEL and associated archives.

The SYMCON package written by Vafeades [183] also uses muMATH to calculate the determining equations (without solving them). The program is restricted to point symmetries. Furthermore, the program verifies whether the symmetry group is of variational or divergence type and computes the conservation laws associated with the symmetries. Unfortunately, these programs are confined to the 256 KB memory accessible by muMATH and can therefore presently not handle very large systems of equations. This limitation motivated Vafeades to

rewrite his SYMCON program in MACSYMA syntax [184], [185]. The MACSYMA version of SYMCON can handle generalized symmetries and their conservation laws.

For completeness, Mikhailov developed software in muMATH to verify the integrability of systems of PDEs by testing for the existence of higher symmetries. The program computes special symmetries, canonical conservation laws and carries out conformal transformations to bring PDEs into canonical form. With their PC program Mikhailov *et al.* [114] were able to produce an exhaustive list of integrable nonlinear Schrödinger type equations. In this context, integrable means that the equations have infinitely many conserved quantities and infinitely many local symmetries.

4.7 Programs for Other Systems

Kornyak and Fushchich [61], [62], [97] developed programs in Turbo C and AMP for the computation of Lie–Bäcklund symmetries. Their programs also classify equations with arbitrary parameters and functions with respect to such symmetries. Important to note is that their programs reduce the determining equations into passive form.

We should mention their two FORMAC programs. The first program, called *LB*, was written in the PL/1 language by Fedorova and Kornyak [54], [55]. The successor, called *LBF*, was developed by Fushchich and Kornyak [62]. Both programs create the system of determining equations for Lie–Bäcklund symmetries and attempt to solve these equations. The program*LBF*, with its 1362 lines of PL/1–FORMAC code, is completely automatic and consists of 37 subroutines, one of which brings the determining equations in passive (Riquier–Janet) form. The program*LB* [55] is available from the Computer Physics Communications Program Library in Belfast. The above programs were designed for low memory requirements so that they could run on PCs.

The PL/1–based FORMAC package CRACKSTAR developed by Wolf [202], [203], [204] allows investigation of Lie symmetries of systems of PDEs, besides dealing with dynamical symmetries of ODEs [133], and the like. A good overview of the capabilities of CRACKSTAR is given in[203]. For efficiency, CRACKSTAR generates and solves first order determining equations early on, and then continues with the higher–order determining equations. The successor of CRACKSTAR is the REDUCE package CRACK discussed earlier.

Gerdt and Zharkov [68] and Gerdt *et al.* [69], [70] used REDUCE and PL/1–FORMAC to investigate the integrability of nonlinear evolution equations. Their program FORMINT contains algorithms to calculate Lie–Bäcklund symmetries and conserved densities, but does not use the jet bundle formalism.

The calculation of the Lie group by computer was also proposed by Popov[136], who used the program SOPHUS for the calculation of conservation laws of evolution equations.

In [24], Bocharov and Bronstein present SCoLAr, a package written in standard PASCAL, for finding infinitesimal symmetries and conservation laws of arbitrary systems of differential equations.

The PC–package DELiA, written in Turbo PASCAL by Bocharov and his collaborators [21], [22], is a stand–alone computer algebra system for investigating differential equations. It performs various tasks based on Lie's approach, such as the computation of Lie–point and Lie–Bäcklund symmetries, canonical conserved densities and generalized conservation laws; simplification and partial integration of overdetermined systems of differential equations, etc..

In order to be able to handle large problems DELiA first generates and solves first order determining equations, and then continues to generate and solve the higher–order determining equations. The analyzer/integrator, which is available as a separate tool at the user–level, includes a general algorithm for passivization, together with a set of integration rules for linear and quasi linear systems of PDEs. Currently, a MS Windows version of DEliA, called MS Win DELiA, is under development. This new version, to be demonstrated during the ISSAC '93 meeting in Kiev, will facilitate user–level interface and allow the user to surpass the memory barrier of 640 KB under DOS.

Using the algorithmic language REFAL, Topunov [178] developed a software package for symmetry analysis that contains subroutines to reduce determining systems in passive form.

For completeness, we mention the pioneering work by C. Wulfman [209] and his students Davison [43] and Nagao [119]. Already in the early seventies, Davison [43] developed computer algorithms in SNOBOL, a now obsolete computer language, that could handle symbolic manipulations with differential operators. In 1980, Nagao [119] wrote the computer program DETERMININGEQS (in PASCAL) that could approximate Lie generators for dynamical systems.

5 Solving the Determining Equations and Computation of the Size of the Symmetry Group

Solving the determining equations can usually be done by inspection, using elementary results from the theory of linear PDEs. Solving them on a computer may be time consuming, since the simplest approach varies greatly from case to case. Furthermore, a computer program may accidentally not catch the most general result and therefore may return an incomplete symmetry group. The author is aware of this problem which occurred in testing some of the reviewed symmetry programs. Even worse, the computer algorithm may not be able to determine the solution of the determining equations in a finite number of steps.

The algorithm SYMSIZE [165], [166] is available with the computer algebra systems AXIOM and REDUCE, as part of the package SPDE, written by Schwarz. Use of SYMSIZE circumvents some of the shortcomings mentioned above. Indeed, if a differential equation has no other than trivial symmetries or if the symmetry group is small (because all generators are algebraic and of low degree), SYMSIZE will greatly help in completely solving the symmetry problem. In contrast to the *heuristic* algorithms for the explicit computation of the symmetry generators, the size of the symmetry group can always be determined with SYMSIZE in a finite number of steps. SYMSIZE accepts a systems of PDEs as input, and allows to compute *a priori* the number of free parameters if the group is finite and the number of unspecified functions if the group is infinite. In turn, SYMSIZE allows to test *a posteriori* if the solution of the determining equations is complete. In cases where some, perhaps all, symmetries are known by inspection or from the physics of the problem, the knowledge of the size of the symmetry group can evade an expensive search for more, perhaps non existent, symmetries.

The heart of SYMSIZE is the procedure *InvolutionSystem*, which transforms the determining system into an involutive system by means of a critical pair/completion algorithm. Similar algorithms are applied in computing Gröbner bases in polynomial ideal theory. The mathematical background of Schwarz' algorithms [159], [165], [166] goes back to the theory of differential equations due to Riquier [145] and Janet [90], [91]. Early efforts to implement

the above theories and Cartan's exterior forms approach to involutive systems, are due to Russian teams [9], [66], [178].

Although Schwarz did valuable pioneering work on solving systems of linear homogeneous PDEs [159], he is not the only one working on innovative ways of classifying, subsequently reducing, and finally solving overdetermined systems of linear homogeneous partial differential equations. Work by Bocharov and Bronstein[24], Ganzha and Yanenko [67], Gerdt and Zharkov [68], Reid [137], [138], [139], [140], Reid *et al.* [141], [142], [144], Topunov [178], and Wolf and Brand [204], [205], [206], [207], [208], amongst others, has lead to a partial implementation of reduction algorithms into sophisticated symbolic code. For instance, skillful use of the tools reducing systems of linear homogeneous PDEs, available within the package CRACK [206], [207], [208], can also greatly assist in the investigation of the size of the symmetry group.

The common idea is to bring the determining systems in passive form [26], [100], as defined early on by Janet [90], [91], Riquier [145] and Thomas [177]. Also, using the same completion method, one may attempt to decouple the systems of determining equations and thereby facilitate further integration[201].

Once systems are reduced into standard involutive form or decoupled, subsequent integration is more tractable and reliable. One could use separation of variables, standard techniques for linear differential equations, and specific heuristic rules as described in [81], [157], [158], [160], [175], [176]. The only determining equations left for manual handling, should be the 'constraint' equations or any other equations whose general solutions cannot be written explicitly in closed form.

Concurrently, yet independent of Schwarz, Reid [139], [140] realized that triangularization algorithms may allow bypassing the explicit solution of the determining equations and compute the size of the symmetry group and the commutators immediately. Reid developed the program SYMCAL[139], written originally in MACSYMA, but now converted by Reid and Wittkopf into Maple [144]. The program SYMCAL computes the dimension and the structure constants[137], [138] of the Lie symmetry algebra of any system of PDEs. An extension of the algorithm [137] also allows to classify differential equations (with variable coefficients) according to the structure of their symmetry groups. Furthermore, the approach advocated by Reid applies to the determination of symmetries of Lie–contact and Lie–Bäcklund types as well as potential symmetries.

The algorithm described in [140], provides information about the dimension and commutators of the Lie symmetry algebra. It is based on explicit Taylor expansions of the symmetry generators, and therefore computationally expensive and restricted to finite dimensional Lie algebras. The newest Maple algorithm [142] allows to compute the dimension and the commutation relations without Taylor expansions; hence it is applicable to infinite–dimensional Lie algebras.

Recently, Reid extended the underlying algorithm *standard_form* [139] to account for general, total degree orderings. The procedure *standard_form* can be viewed as a generalization to linear differential equations of the Gaussian reduction method for matrices or linear systems. The algorithm now takes as input the system of PDEs and a matrix which specifies a complete ordering on the derivatives appearing in the system. It then reduces the system of PDEs to an equivalent simplified ordered triangular system with all integrability conditions included and all redundancies (differential and algebraic) eliminated.

Once the system is in standard form, one can continue with the automatic determination of a Taylor series solution of the system to a specified finite degree. Reid and Wittkopf's

package [144] allows automated interfacing with major symmetry packages such as DIMSYM[170], LIESYMM [29], and SYMMGRP.MAX [30], and also with the differential Gröbner basis package DIFFGROB[111], [112]. Full details and many illustrative examples of the package, which includes the function *standard_form* and other powerful algorithms for symmetry analysis of PDEs, are given in[144].

Reid and McKinnon just finished the development of a recursive algorithm, called *Rsolve_Pdesys* [143], that builds on Reid's *standard_form* [139] and on algorithms of Abramov and Kvashenko [3] and Bronstein [25]. Reid and McKinnon's algorithm *RSolve_Pdesys* now finds particular solutions of linear systems of PDEs using only ODE solution techniques. Applied to symmetry problems, their algorithm will find all polynomial/rational solutions of the determining equations provided the symmetry group is finite–dimensional.

6 Examples

6.1 The Harry Dym Equation

Consider the Harry Dym equation[1],

$$u_t = u^3 u_{xxx}. \tag{16}$$

Clearly, this is one equation ($m=1$) with two independent variables ($p=2$) and one dependent variable ($q=1$). The assignments of the variables are as follows:

$$x \mapsto x[1] \quad , \quad t \mapsto x[2] \quad , \quad u \mapsto u[1] \tag{17}$$

This permits to rewrite the equation(16) in a form accepted by the program SYMMGRP.MAX, i.e.

```
e1  :  u[1,[0,1]]-u[1]^3*u[1,[3,0]].
```

For PDELIE and SYM_DE the input form would be

```
DIFF(U,T)-U^3*DIFF(U,X,3).
```

For SPDE and LIE the program accepts

```
U(1,2)-U(1)^3*U(1,1,1,1).
```

Next, one selects the variable u_t for elimination, e.g.

```
v1  :  u[1,[0,1]].
```

Then, the programs automatically compute the determining equations for the coefficients eta[1]$\Leftrightarrow \eta^x$, eta[2]$\Leftrightarrow \eta^t$ and phi[1]$\Leftrightarrow \varphi^u$ of the vector field

$$\alpha = \eta^x \frac{\partial}{\partial x} + \eta^t \frac{\partial}{\partial t} + \varphi^u \frac{\partial}{\partial u}. \tag{18}$$

There are only eight determining equations,

$$\frac{\partial \text{eta2}}{\partial u[1]} = 0,$$

$$\frac{\partial \text{eta2}}{\partial x[1]} = 0,$$

$$\frac{\partial \text{eta1}}{\partial u[1]} = 0,$$

$$\frac{\partial^2 \text{phi1}}{\partial u[1]^2} = 0,$$

$$\frac{\partial^2 \text{phi1}}{\partial u[1]\partial x[1]} - \frac{\partial^2 \text{eta1}}{\partial x[1]^2} = 0,$$

$$\frac{\partial \text{phi1}}{\partial x[2]} - 2u[1]^3 \frac{\partial^3 \text{phi1}}{\partial x[1]^3} = 0,$$

$$3u[1]^3 \frac{\partial^3 \text{phi1}}{\partial u[1]\partial x[1]^2} - \frac{\partial \text{eta1}}{\partial x[2]} - 2u[1]^3 \frac{\partial^3 \text{eta1}}{\partial x[1]^3} = 0,$$

$$u[1] \frac{\partial \text{eta2}}{\partial x[2]} - 3u[1] \frac{\partial \text{eta1}}{\partial x[1]} - 3\ \text{phi1} = 0.$$

These determining equations can easily be solved explicitly, either automatically with SPDE, LIE and PDELIE, or with the feedback mechanism within SYM_DE and SYMMGRP.MAX. The general solution, rewritten in the original variables, is

$$\eta^x = k_1 - k_3 t + k_5 t^2 ,$$

$$\eta^t = k_2 - 3k_4 t ,$$

$$\varphi^u = (k_3 - k_4 - 2k_5 x) u ,$$

where $k_1, \ldots, k_5$ are arbitrary constants. The five infinitesimal generators then are

$$G_1 = \partial_x \quad ,$$

$$G_2 = \partial_t \quad ,$$

$$G_3 = x\partial_x - u\partial_u \quad ,$$

$$G_4 = -3t\partial_t - u\partial_u \quad ,$$

$$G_5 = x^2\partial_x - 2xu\partial_u \quad .$$

Clearly, (16) is invariant under translations (G_1 and G_2) and scaling (G_3 and G_4). The flow corresponding to each of the infinitesimal generators can be obtained via simple integration. For example, let us compute the flow corresponding to G_5. This requires integration of the first order system

$$\begin{aligned} \frac{d\tilde{x}}{d\varepsilon} &= \tilde{x}^2 , & \tilde{x}(0) &= x , \\ \frac{d\tilde{t}}{d\varepsilon} &= 0 , & \tilde{t}(0) &= t , \\ \frac{d\tilde{u}}{d\varepsilon} &= 2\tilde{x}\tilde{u} , & \tilde{u}(0) &= u , \end{aligned} \tag{19}$$

where ε is the parameter of the transformation group. One readily obtains

$$\begin{aligned} \tilde{x}(\varepsilon) &= \frac{x}{(1-\varepsilon x)} , \\ \tilde{t}(\varepsilon) &= t , \\ \tilde{u}(\varepsilon) &= \frac{u}{(1-\varepsilon x)^2} . \end{aligned}$$

We therefore conclude that for any solution $u = f(x, t)$ of equation (16) the transformed solution

$$\tilde{u}(\tilde{x}, \tilde{t}) = (1+\varepsilon\tilde{x})^2 f\left(\frac{\tilde{x}}{1+\varepsilon\tilde{x}}, \tilde{t}\right) \tag{20}$$

will solve $\tilde{u}_{\tilde{t}} = \tilde{u}^3\, \tilde{u}_{\tilde{x}\tilde{x}\tilde{x}}$.

6.2 The Magneto–Hydro–Dynamics Equations

As an example of a large system of differential equations, we take the equations for Magneto–Hydro–Dynamics (MHD)[59] and carry out the symmetry analysis with SYMMGRP.MAX. If we neglect dissipative effects, and thus restrict the analysis to the *ideal* case, we have

$$\frac{\partial \rho}{\partial t} + (\vec{v}\cdot\nabla)\rho + \rho\nabla\cdot\vec{v} = 0 , \tag{21}$$

$$\rho\left(\frac{\partial \vec{v}}{\partial t} + (\vec{v}\cdot\nabla)\vec{v}\right) + \nabla\left(p + \frac{1}{2}\vec{H}^2\right) - (\vec{H}\cdot\nabla)\vec{H} = 0 , \tag{22}$$

$$\frac{\partial \vec{H}}{\partial t} + (\vec{v}\cdot\nabla)\vec{H} + \vec{H}\nabla\cdot\vec{v} - (\vec{H}\cdot\nabla)\vec{v} = 0 , \tag{23}$$

$$\nabla\cdot\vec{H} = 0 , \tag{24}$$

$$\frac{\partial}{\partial t}\left(\frac{p}{\rho^\kappa}\right) + (\vec{v}\cdot\nabla)\left(\frac{p}{\rho^\kappa}\right) = 0 , \tag{25}$$

with pressure p, mass density ρ, coefficient of viscosity κ, fluid velocity $\vec{v}$ and magnetic field $\vec{H}$. Using the first equation, we eliminate ρ from the last equation, and replace it by

$$\frac{\partial p}{\partial t} + \kappa p(\nabla \cdot \vec{v}) + (\vec{v} \cdot \nabla) p = 0. \tag{26}$$

If we split the vector equations in scalar equations for the vector components, we have a system of $m = 9$ equations, with $p = 4$ independent variables and $q = 8$ dependent variables. For convenience, we denote the components of the vector $\vec{v}$ by v_x, v_y and v_z, not to be confused with partial derivatives of v. The assignment of the variables is as follows:

$$\begin{aligned}
x &\mapsto x[1], & v_x &\mapsto u[3], \\
y &\mapsto x[2], & v_y &\mapsto u[4], \\
z &\mapsto x[3], & v_z &\mapsto u[5], \\
t &\mapsto x[4], & H_x &\mapsto u[6], \\
\rho &\mapsto u[1], & H_y &\mapsto u[7], \\
p &\mapsto u[2], & H_z &\mapsto u[8].
\end{aligned} \tag{27}$$

The variables to be eliminated are selected as follows: for the first 7 variables and the ninth variable we pick the partial derivatives with respect to t of ρ, v_x, v_y, v_z, H_x, H_y, H_z and p. From the eighth equation we select $\partial H_x/\partial x$ for elimination.

We will only consider the case where $k \neq 0$. Running this case on a Digital VAX 4500 with 64 MB of RAM, in about 50 minutes of CPU time, the program SYMMGRP.MAX creates the 222 determining equations for the coefficients of the vector field

$$\begin{aligned}
\alpha = {} & \eta^x \frac{\partial}{\partial x} + \eta^y \frac{\partial}{\partial y} + \eta^z \frac{\partial}{\partial z} + \eta^t \frac{\partial}{\partial t} + \varphi^\rho \frac{\partial}{\partial \rho} + \varphi^p \frac{\partial}{\partial p} \\
& + \varphi^{v_x} \frac{\partial}{\partial v_x} + \varphi^{v_y} \frac{\partial}{\partial v_y} + \varphi^{v_z} \frac{\partial}{\partial v_z} + \varphi^{H_x} \frac{\partial}{\partial H_x} + \varphi^{H_y} \frac{\partial}{\partial H_y} + \varphi^{H_z} \frac{\partial}{\partial H_z}.
\end{aligned}$$

Using SYMMGRP.MAX interactively, we then integrated the determining system and obtained the solution, expressed in the original variables,

$$\begin{aligned}
\eta^x &= k_2 + k_5 t - k_8 y - k_9 z + k_{11} x, \\
\eta^y &= k_3 + k_6 t + k_8 x - k_{10} z + k_{11} y, \\
\eta^z &= k_4 + k_7 t + k_9 x + k_{10} y + k_{11} z, \\
\eta^t &= k_1 + k_{12} t, \\
\varphi^\rho &= -2(k_{11} + k_{12} + k_{13})\rho, \\
\varphi^p &= -2(k_{13} + \gamma)p, \\
\varphi^{v_x} &= k_5 + k_8 v_y + k_9 v_z - (k_{11} + k_{12}) v_x, \\
\varphi^{v_y} &= k_6 + k_8 v_x + k_{10} v_z - (k_{11} + k_{12}) v_y, \\
\varphi^{v_z} &= k_7 + k_9 v_x + k_{10} v_y - (k_{11} + k_{12}) v_z,
\end{aligned}$$

$$\varphi^{H_x} = k_{13} H_x + k_8 H_y + k_9 H_z \ ,$$
$$\varphi^{H_y} = k_{13} H_y + k_8 H_x + k_{10} H_z \ ,$$
$$\varphi^{H_z} = k_{13} H_z + k_9 H_x + k_{10} H_y \ .$$

It is clear that there is a thirteen dimensional Lie algebra spanned by the generators:

$$G_1 = \partial_t \ ,$$
$$G_2 = \partial_x \ ,$$
$$G_3 = \partial_y \ ,$$
$$G_4 = \partial_z \ ,$$
$$G_5 = t\partial_x + \partial_{v_x} \ ,$$
$$G_6 = t\partial_y + \partial_{v_y} \ ,$$
$$G_7 = t\partial_z + \partial_{v_z} \ ,$$
$$G_8 = x\partial_y + y\partial x + v_x\partial_{v_y} + v_y\partial_{v_x} + H_x\partial_{H_y} + H_y\partial_{H_x} \ ,$$
$$G_9 = y\partial_z + z\partial y + v_y\partial_{v_z} + v_z\partial_{v_y} + H_y\partial_{H_z} + H_z\partial_{H_y} \ ,$$
$$G_{10} = z\partial_x + x\partial z + v_z\partial_{v_z} + v_x\partial_{v_z} + H_z\partial_{H_x} + H_x\partial_{H_z} \ ,$$
$$G_{11} = x\partial_x + y\partial_y + z\partial_z + 2\rho\partial_\rho + v_x\partial_{v_x} + v_y\partial_{v_y} + v_z\partial_{v_z} \ ,$$
$$G_{12} = t\partial_t + 2\rho\partial_\rho + (v_x\partial_{v_x} + v_y\partial_{v_y} + v_z\partial_{v_z}) \ ,$$
$$G_{13} = 2\rho\partial_\rho + 2p\partial_p + H_x\partial_{H_x} + H_y\partial_{H_y} + H_z\partial_{H_z} \ .$$

Thus, the MHD equations (21) – (25) are invariant under translations G_2 through G_4, Galilean boosts G_5 through G_7, rotations G_8 through G_{10}, and dilations G_{11} through G_{13}. In contrast to the results obtained for the 1+1 and the 2+1 dimensional versions of the MHD problem, the dimension of the Lie algebra for (21) in the full 3+1 dimensions (x, y, z and time t) is independent of the value of the coefficient of viscosity κ. Our results confirm those of [59], and Grundland and Lalague [76], [77], who computed the classical and some nonclassical symmetries of the MHD, and also classified all the subalgebras in conjugacy classes. The MHD system has been used by other investigators [29], [144] to test symmetry programs.

Another completely worked example of the calculation of Lie–point symmetries of a system of PDEs is given in [30]. It involves the Karpman equations [94], for which our symmetries were independently verified with REDUCE programs by Kersten and Gragert, Sherring and Prince, and Wolf [205].

7 Further Research and Development

The most challenging part of Lie symmetry analysis by computer, involves the design of an 'integrator' for the overdetermined systems of linear homogeneous partial differential equations. This topic is also of importance in the study of so–called adjoint symmetries of differential equations [152], and in many other areas where determining equations of the same type occur. Ideally, good integration code should be applicable to generic systems of linear differential equations, which do not necessarily come from symmetry analysis.

In the context of Lie symmetry analysis, one can aim at the design of faster and more powerful algorithms that work for *large* systems of determining equations, typically a few hundred, and that automatically reduce systems to where they can be handled interactively with the computer, or by hand.

Since the early developments [81], [157], [160], [174], [175] of semi–heuristic methods to solve determining equations, substantial progress has been made in understanding the mathematics of this problem and several algorithms are now available. These algorithms attempt to close the gap between solution techniques for ODEs and PDEs (consult [171] for an impressive review and large bibliography). Despite the innovative efforts of Reid, Schwarz, Wolf and Brand, and others, there is no general algorithm available to integrate an arbitrary (overdetermined) system of determining equations, which consists of linear homogeneous partial differential equations for the ηs and the φs.

After searching the relevant literature [24], [137], [138], [139], [143], [159], [165], [166], [178], [204], [205], [206], [207], [208] it is clear that many mathematical questions remain open. However, one thing is certain: to design a reliable and powerful integration algorithm for a system of determining equations, the system needs to be brought into a canonical form. Some authors call it the normal, involutive or passive form. The original theory of involutive systems goes back to Cartan [28], Janet [90], [91], Kähler [92], Riquier [145], Ritt [146], Thomas [177], and Tresse [179], [180].

A concise survey of the theory with complete proofs is given by Finikov[57].

The algorithms by Reid [137], [138], [144], Schwarz [165], and Topunov [178] are clones of the Riquier–Janet–Thomas method. Roughly speaking, the methods that reduce the systems of linear homogeneous PDEs into an equivalent, but much simpler standard form, can be viewed as a generalization of triangulation by Gauss–Jordan elimination but, applied here to systems of linear PDEs. First, the original system is appended by all its differential consequences. Second, highest derivatives are eliminated, and, if they occur, integrability conditions are added to the system. The procedure is then repeated until the new system is in involution.

In the full computer implementation of such a 'triangulation algorithm', one takes advantage of a 'differential' generalization of Buchberger's algorithm for Gröbner bases. Buchberger's algorithm, which is included as standard package with modern symbolic manipulation programs, provides a technique for canonically simplifying polynomial nonlinear systems of algebraic equations. The 'differential' generalization of that algorithm allows to reduce systems of nonlinear (and consequently also linear) PDEs into standard form. The Maple program DIFFGROB by Mansfield and Fackerell [111], [112] offers a differential Gröbner basis algorithm (DIFFGBASIS) applicable to nonlinear PDEs, not necessary resulting from symmetry analysis. Needless to say, the package DIFFGROB can be applied [38], [39], [40] to reduce linear and nonlinear determining equations, the latter arising in the case of nonclassical Lie symmetries.

Other ideas could be incorporated in the design of faster and more efficient symbolic software for Lie symmetry analysis. Let us give a couple of examples. Lengthy calculations should be broken up into smaller pieces. One should consistently take advantage of the 'linear algebra' structure of the Lie symmetry problem. For instance, prolongations should be applied to vector fields of single equations or subsets of equations. Furthermore, full expansions of the prolonged vector fields should be halted until the explicit forms are actually needed. Avoiding lengthy, redundant expansions will make the generation of the determining equations much more efficient, in particular for nonclassical and Lie–Bäcklund symmetries. Due to lack of efficiency, most of the existing Lie symmetry programs cannot generate the (classical) determining equations for e.g. Newton's three body problem in three dimensions, governed by nine second order equations.

8 Conclusion

The various programs that we reviewed need very little data and are straightforward to use provided the user has access to and knows the basics of the underlying symbolic manipulation programs, such as MACSYMA, Maple, Mathematica and REDUCE.

Apart from the theoretical study of the underlying mathematics, there is a need for further development and implementation of effective algorithms for generating, simplifying and fully solving the determining equations for (classical and nonclassical) Lie–point symmetries and generalized or Lie–Bäcklund symmetries.

The availability of sophisticated symbolic programs should accelerate the study of symmetries of physically important systems of differential equations in classical mechanics, fluid dynamics, elasticity and other applied areas.

9 Acknowledgements

The author likes to thank M. Ablowitz, G. Baumann, D. Bérubé, G. Bluman, A.V. Bocharov, B. Champagne, P. Clarkson, R. Conte, N. Coult, B. Fuchssteiner, M. Grundland, A.K. Head, B. Herbst, S. Herod, M. Hickman, D. Holm, D. Levi, M. MacCallum, E.L. Mansfield, L. Marchildon, A. Mikhailov, M.C. Nucci, P. Olver, J. Ondich, G. Prince, D. Rand, G. Reid, W. Sarlet, F. Schwarz, J. Sherring, W.–H. Steeb, S. Steinberg, M. Torrisi, P. Vafeades, T. Wilcox, P. Winternitz, T. Wolf and C. Wulfman.

Their suggestions, comments, corrections and criticisms greatly helped to improve the manuscript.

Special thanks goes to S. Collart for inviting me to write this paper for Euromath Bulletin.

The research for this paper was supported in part by Grant # CCR–9300978 of the National Science Foundation of the United States of America.

References

[1] M.J. Ablowitz and P.A. Clarkson, *Solitons, Nonlinear Evolution Equations and Inverse Scattering* (Cambridge University Press, Cambridge, 1991).

[2] B. Abraham–Shrauner and A. Guo, J. Phys. A.: Math. Gen. **25** (1992) 5597–5608.

[3] S.A. Abramov and K.Yu. Kvashenko, *Fast algorithms to search for the rational solutions of linear differential equations with polynomial coefficients*, in: Proc. ISSAC 91, Bonn, Germany, 1991, Ed.: S.M. Watt (ACM Press, New York, 1991) 267–270.

[4] W.F. Ames, *Nonlinear Ordinary Differential Equations in Transport Processes* (Academic Press, New York, 1968).

[5] W.F. Ames, *Nonlinear Partial Differential Equations in Engineering. Vol. I & II* (Academic Press, New York, 1965, 1972).

[6] W.F. Ames, *Symmetry in nonlinear mechanics*, in: Nonlinear Equations in the Applied Sciences, Eds.: W.F. Ames and C. Rogers (Academic Press, Boston, 1992) 33–78.

[7] R.L. Anderson and N.H. Ibragimov, *Lie–Bäcklund Transformations in Applications*, Studies in Applied Mathematics **1** (SIAM, Philadelphia, 1979).

[8] I.M. Anderson, N. Kamran and P.J. Olver, *Internal, external and generalized symmetries*, Adv. in Math. **100** (1993) 53–100.

[9] E.A. Arais, V.P. Shapeev and N.N. Janenko, DAN SSSR **214** (1974) p. 296, in Russian.

[10] R. Axford, *Lectures on Lie Groups and Systems of Ordinary and Partial Differential Equations* (Los Alamos, New Mexico, 1983).

[11] G.I. Barenblatt, *Similarity, Self–similarity and Intermediate Asymptotics* (Consultants Bureau, New York, 1979).

[12] G. Baumann, *Lie Symmetries of Differential Equations:*, A Mathematica program to determine Lie symmetries. Wolfram Research Inc., Champaign, Illinois, MathSource 0202–622, 1992.

[13] G. Baumann, *Generalized Symmetries:*, A Mathematica Program to Determine Lie–Bäcklund Symmetries. Wolfram Research Inc., Champaign, Illinois, MathSource 0204–680, 1993.

[14] J.G. Belinfante and B. Kolman, *A Survey of Lie Groups and Lie Algebras with Applications and Computational Methods*, Classics in Applied Mathematics **2** (SIAM, Philadelphia, 1972 & 1989).

[15] W.A. Beyer, *Lie–group theory for symbolic integration of first–order differential equations*, in: Proc. 1979 MACSYMA Users Conference, Washington, D.C., 1979, Ed.: V.E. Lewis (MIT Press, Boston, 1979) 362–384.

[16] G.W. Bluman, J. Math. Anal. Appl. **145** (1990) 52–62.

[17] G.W. Bluman, *Potential Symmetries and Linearization*. Preprint, Department of Mathematics (University of British Columbia, Vancouver, Canada, 1993).

[18] G.W. Bluman and J.D. Cole, J. Math. Mech. **18** (1969) 1025–1042.

[19] G.W. Bluman and J.D. Cole, *Similarity Methods for Differential Equations*, Applied Mathematical Sciences **13** (Springer Verlag, New York, 1974).

[20] G.W. Bluman and S. Kumei, *Symmetries and Differential Equations* (Springer Verlag, New York, 1989).

[21] A.V. Bocharov, *DEliA: A System of Exact Analysis of Differential Equations using S. Lie Approach*. Report by Joint Venture OWIMEX Program Systems Institute of the USSR (Academy of Sciences, Pereslavl–Zalessky, USSR, 1989).

[22] A.V. Bocharov, *DEliA: project presentation*, SIGSAM Bulletin **24** (1990) 37–38.

[23] A.V. Bocharov, The Mathematica Journal **3** (1993) 63–69.

[24] A.V. Bocharov and M.L. Bronstein, Acta Appl. Math. **16** (1989) 143–166.

[25] M. Bronstein, J. Symb. Comp. **13** (1992) 413–439.

[26] R.L. Bryant, S.S. Chern, R.B. Gardner, H.L. Goldschmidt and P.A. Griffiths, *Exterior Differential Systems*, Mathematical Sciences Research Institute Publications **18** (Springer Verlag, New York, 1991).

[27] J.E. Campbell, *Introductory Treatise on Lie's Theory of Finite Continuous Transformation Groups* (Chelsea Publishing Company, New York, 1903, 1966).

[28] E. Cartan, *Les Systèmes Différentiels Extérieur et leurs Applications Géometrique* (Hermann, Paris, 1946).

[29] J. Carminati, J.S. Devitt and G.J. Fee, J. Sym. Comp. **14** (1992) 103–120.

[30] B. Champagne, W. Hereman and P. Winternitz, Comp. Phys. Comm. **66** (1991) 319–340.

[31] B. Champagne and P. Winternitz, *A MACSYMA Program for Calculating the Symmetry Group of a System of Differential Equations.* Report CRM–1278 (Centre de Recherches Mathématiques, Montréal, Canada, 1985).

[32] B.W. Char, *Algorithms using Lie transformation groups to solve first order ordinary differential equations algebraically*. Ph.D Thesis, Department of Computer Science (University of California, Berkeley, California, 1980).

[33] B.W. Char, *Using Lie transformation groups to find closed form solutions to first order differential equations*, in: Proc. ACM Symposium on Symbolic and Algebraic Computation, Snowbird, Utah, 1981, Ed.: P.S. Wang (ACM Press, New York, 1981) 44–50.

[34] B.W. Char and B. McNamara, *LIEPROC: a MACSYMA program for finding adiabatic invariants of simple Hamiltionian systems via the Lie transform*, in: Proc. 1979 MACSYMA Users Conference, Washington, D.C., 1979, Ed.: V.E. Lewis (MIT Press, Boston, 1979) 237–255.

[35] C.C. Chevalley, *Theory of Lie Groups. Vol 1* (Princeton University Press, Princeton, New Jersey, 1946).

[36] P.A. Clarkson, Nonlinearity **5** (1992) 453–472.

[37] P.A. Clarkson and M.D. Kruskal, J. Math. Phys. **30** (1989) 2201–2213.

[38] P.A. Clarkson and E.L. Mansfield, *Symmetry Reductions and Exact Solutions of a Class of Nonlinear Heat Equations*, Physica D (1993) in press.

[39] P.A. Clarkson and E.L. Mansfield, *Algorithms for Nonclassical Symmetries*. Preprint, Department of Mathematics (University of Exeter, Exeter, United Kingdom, 1993).

[40] P.A. Clarkson and E.L. Mansfield, *On a Shallow Water Wave Equation*. Preprint, Department of Mathematics (University of Exeter, Exeter, United Kingdom, 1993).

[41] P.A. Clarkson and P. Winternitz, Physica D **49** (1991) 257–272.

[42] A. Cohen, *An Introduction to the Lie Theory of One−Parameter Groups* (Stechert and Co., New York, 1931).

[43] D.K. Davison, *MANDO: A Computer Program for Symbolic Manipulation of Differential Operators Generating Continuous Transformations*. M.Sc. Thesis (University of the Pacific, Stockton, California, 1973).

[44] A. Donsig, *The Differential Forms Package*, available in Maple 4.2.

[45] L. Dresner, *Similarity Solutions of Nonlinear Partial Differential Equations*, Research Notes in Mathematics **88** (Pitman, Boston, Massachusetts, 1983).

[46] D.G.B. Edelen, Comp. & II**6** (1980) 415–424.

[47] D.G.B. Edelen, *Isovector Methods for Equations of Balance* (Sijthoff & II

[48] D.G.B. Edelen, *Applied Exterior Calculus* (Wiley−Interscience, New York, 1985).

[49] D.G.B. Edelen, The Mathematica Journal **1**, no. 3 (1991) 92–95.

[50] L.P. Eisenhart, *Continuous Groups of Transformations* (Princeton University Press, Princeton, New Jersey, 1933).

[51] V.P. Eliseev, R.N. Fedorova and V.V. Kornyak, Comp. Phys. Comm. **36** (1985) 383–389.

[52] P.G. Estévez Phys. Lett. A **171** (1992) 259–261.

[53] N. Euler and W.−H. Steeb, *Continuous Symmetries, Lie Algebras and Differential Equations* (Bibliographisches Institut Wissenschaftsverlag, Mannheim, 1993).

[54] R.N. Fedorova and V.V. Kornyak, *Applying computer algebra systems for determination of the Lie and Lie−Bäcklund symmetries of differential equations*, in: Proc. Int. Conf. on Computer Algebra and its Applications in Theoretical Physics, Dubna, USSR, 1985 (Joint Institute for Nuclear Research, Dubna, USSR, 1985) 248–261, in Russian.

[55] R.N. Fedorova and V.V. Kornyak, Comp. Phys. Comm. **39** (1986) 93–103.

[56] R.N. Fedorova and V.V. Kornyak, *A REDUCE Program for Computing Determining Equations of Lie−Bäcklund Symmetries of Differential Equations*. Report R11–87–19 (Joint Institute for Nuclear Research, Dubna, USSR, 1987) in Russian.

[57] S.P. Finikov, *Cartan's Exterior Forms Method* (Gostechizdat, Moscow−Leningrad, USSR, 1948) p. 432, in Russian.

[58] A.S. Fokas, Stud. in Appl. Math. **44** (1987) 253–299.

[59] J.C. Fuchs, J. Math. Phys. **32** (1991) 1703–1708.

[60] B. Fuchssteiner, W. Oevel and W. Wiwianka, Comp. Phys. Comm. **44** (1987) 47–55.

[61] W.I. Fushchich, *Symmetry in the problems of mathematical physics*, in: Algebraic–Theoretical Studies in Mathematical Physics (Kiev Inst. of Math., Kiev, USSR, 1981) 6–28, in Russian.

[62] W.I. Fushchich and V.V. Kornyak, J. Symb. Comp. **7** (1989) 611–619.

[63] W.I. Fushchich and A.G. Nikitin, *Symmetries of Maxwell's Equations* (D. Reidel, Dordrecht, The Netherlands, 1987).

[64] G. Gaeta, Acta Appl. Math. **28** (1992) 43–68.

[65] G. Gaeta, *Geometrical Symmetries of Nonlinear Equations and Physics* (1993) in preparation.

[66] V.G. Ganja, S.V. Meleshko, F.A. Murzin, V.P. Shapeev and N.N. Janenko, DAN SSSR **261** (1981) p. 1044.

[67] V.G. Ganzha and N.N. Yanenko, *Compatibility Analysis for Systems of Differential Equations on a Computer*. ITPM Preprint # 20–82 (ITPM, Novosibirsk, USSR, 1982) in Russian.

[68] V.P. Gerdt and A.Yu. Zharkov, *Computer generation of necessary integrability conditions for polynomial–nonlinear evolution systems*, in: Proc. ISSAC 90, Tokyo, Japan, 1990, Eds.: S. Watanabe and M. Nagata (ACM Press, New York, 1990) 250–254.

[69] V.P. Gerdt, A.B. Shvachka and A.Yu. Zharkov, Comp. Phys. Comm. **34** (1985) 303–311.

[70] V.P. Gerdt, A.B. Shvachka and A.Yu. Zharkov, J. Symb. Comp. **1** (1985) 101–107.

[71] P.K.H. Gragert, *Symbolic Computations in Prolongation Theory*. Ph.D. Thesis (Twente University of Technology, Enschede, The Netherlands, 1981).

[72] P.K.H. Gragert, Acta Math. Appl. **16** (1989) 231–242.

[73] P.K.H. Gragert and P.H.M. Kersten, *Implementation of differential geometry objects and functions with an application to extended Maxwell equations*, in: Proc. EUROCAM 82, Marseille, France, 1982, Ed.: J. Calmet, Lecture Notes in Computer Science **144** (Springer Verlag, New York, 1982) 181–187.

[74] P.K.H. Gragert and P.H.M. Kersten, Acta Appl. Math. **24** (1991) 211–231.

[75] P.K.H. Gragert, P.H.M. Kersten and A. Martini, Acta Appl. Math. **1** (1983) 43–77.

[76] A.M. Grundland and L. Lalague, *Subgroupes de Lie des groupes de symmetry pour les équations de la dynamique de fluid I*. Preprint (Centre de Recherches Mathématiques, Montréal, Canada, 1993).

[77] A.M. Grundland and L. Lalague, *Subgroupes de Lie des groupes de symmetry pour les équations de magneto−hydro−dynamique.* Preprint (Centre de Recherches Mathématiques, Montréal, Canada, 1993).

[78] A.G. Hansen, *Similarity Analyses of Boundary Value Problems in Engineering* (Prentice Hall, Englewood Cliffs, 1964).

[79] B.K. Harrison and F.B. Estabrook, J. Math. Phys. **12** (1971) 653–666.

[80] D. Hartley and R. W. Tucker, J. Symb. Comp. **12** (1991) 655–667.

[81] A.K. Head, *LIE: A PC Program for Lie Analysis of Differential Equations*, Comp. Phys. Comm. (1993) in press.

[82] W. Hereman, *SYMMGRP.MAX: A symbolic program for symmetry analysis of partial differential equations*, in: Exploiting Symmetry in Applied and Numerical Analysis, Proc. AMS−SIAM Summer Seminar, Fort Collins, Colorado, 1992, Eds.: E. Allgower, K. Georg and R. Miranda, Lectures in Applied Mathematics (AMS, Providence, Rhode Island, 1993) in press.

[83] R. Hermann, *Differential Geometry and the Calculus of Variations* (Academic Press, New York, 1968).

[84] S. Herod, *MathSym: A Mathematica Program for Computing Lie Symmetries*. Preprint, Program in Applied Mathematics (The University of Colorado, Boulder, Colorado, 1992) submitted to the Mathematica Journal (1993).

[85] M. Hickman, *The Use of Maple in the Search for Symmetries*, Research Report no. 77, Department of Mathematics (University of Canterbury, Christchurch, New Zealand, 1993).

[86] J.M. Hill, *Solution of Differential Equations by Means of One−Parameter Groups* (Pitman, Boston, Massachusetts, 1982).

[87] J.M. Hill, *Differential Equations and Group Methods for Scientists and Engineers* (CRC Press, Boca Raton, 1992).

[88] N.H. Ibragimov, *Transformation Groups Applied to Mathematical Physics* (Reidel, Boston, Massachusetts, 1985).

[89] N.H. Ibragimov, Ed., *CRC Handbook of Lie Group Analysis of Differential Equations, Volume 1* (CRC Press, Boca Raton, Florida, 1993). Volumes 2 and 3, in preparation.

[90] M. Janet, J. Math. Pure et Appl. **3** (1920) 65–151.

[91] M. Janet, *Leçons sur les Systèmes d'Equations aux Dérivées* (Gauthier−Villars, Paris, 1929); *Equations aux Dérivées Partielles* (CNRS, Paris, 1956).

[92] E. Kähler, *Einführung in die Theorie der Systeme von Differentialgleichungen* (B.G. Teubner, Leipzig, 1934).

[93] E.G. Kalnins, *Separation of Variables for Riemannian Spaces of Constant Curvature* (Longman, Essex, 1986).

[94] V.I. Karpman, Phys. Lett. A **136** (1989) 216–220.

[95] P.H.M. Kersten, *Infinitesimal Symmetries: A Computational Approach*. Ph.D. Thesis (Twente University of Technology, Enschede, The Netherlands, 1985); CWI Tract 34 (Center for Mathematics and Computer Science, Amsterdam, 1987).

[96] P.H.M. Kersten, Acta Appl. Math. **16** (1989) 207–229.

[97] V.V. Kornyak and W.I. Fushchich, *A program for symmetry analysis of differential equations*, in: Proc. ISSAC 91, Bonn, Germany, 1991, Ed.: S.M. Watt (ACM Press, New York, 1991) 315–316.

[98] B. Kostant, *Graded manifolds, graded Lie theory and prequantization*, in: Proc. Symposium on Differential Geometrical Methods in Mathematical Physics, Bonn, Germany, 1975, Eds.: K. Bleuler and A. Reetz, Lecture Notes in Mathematics **570** (Springer Verlag, New York, 1977) 177–306.

[99] I.S. Krasil'shchik, V.V. Lychagin and A.M. Vinogradov, *Geometry of Jet Spaces and Nonlinear Partial Differential Equations*, Adv. Stud. Contemp. Math. **1** (Gordon & Breach, New York 1986).

[100] M. Kuranishi, *Lectures on Involutive Systems of Partial Differential Equations* (Publ. Soc. Math., Sao Paulo, Brasil, 1967).

[101] D. Levi and P. Winternitz, Eds., *Symmetries and Nonlinear Phenomena* (World Scientific, Singapore, 1988).

[102] D. Levi and P. Winternitz, J. Phys. A: Math. Gen. **22** (1989) 2915–2924.

[103] S. Lie, *Theorie der Transformationsgruppen*, Volumes I, II, III (B.G. Teubner, Leipzig, 1888, 1890, 1893).

[104] S. Lie, *Vorlesungen über Differentialgleichungen mit Bekannten Infinitesimalen Transformationen* (B.G. Teubner, Leipzig, 1891, 1912; Chelsea Publishing Company, New York, 1967).

[105] S. Lie, *Vorlesungen über Continuierliche Gruppen*, Abteilung III (Chelsea Publishing Company, New York, 1971).

[106] S. Lie, *Sophus Lie's 1880 Transformation Group Paper*, Translated by M. Ackerman, comments by R. Hermann, Lie Groups: History, Frontiers on Applications **1** (Mathematical Sciences Press, Brookline, 1975).

[107] S. Lie, *Sophus Lie's 1884 Differential Invariants Paper*, Translated by M. Ackerman, comments by R. Hermann, Lie Groups: History, Frontiers on Applications **3** (Mathematical Sciences Press, Brookline, 1975).

[108] S. Lie and F. Engel, *Theorie der Transformationsgruppen II*, (B.G. Teubner, Leipzig, 1890).

[109] I.G. Lisle, *Equivalence Transformations for Classes of DEs*. Ph.D. Thesis (Department of Mathematics, University of British Columbia, Vancouver, Canada, 1992).

[110] M.A.H. MacCallum, *An ordinary differential equation solver for REDUCE*, in: Proc. ISSAC 88, Rome, Italy, 1988, Ed.: P. Gianni, Lecture Notes in Computer Science **358** (Springer Verlag, Berlin, 1988) 196–205.

[111] E.L. Mansfield, *DIFFGROB2: A Symbolic Algebra Package for Analyzing Systems of PDE using Maple*, Users Manual for Release 2, School of Mathematics, Physics, Computer Science, and Electronics (Macquarie University, Sydney, Australia, 1993).

[112] E.L. Mansfield and E.D. Fackerell, *Differential Gröbner Bases*, Preprint 92–108 School of Mathematics, Physics, Computer Science, and Electronics (Macquarie University, Sydney, Australia, 1992) submitted to J. Symb. Comp. (1993).

[113] R. Martini, Ed., *Geometric Approaches to Differential Equations*, Lecture Notes in Mathematics **810** (Springer Verlag, New York, 1979).

[114] A.V. Mikhailov, A.B. Shabat and V.V. Sokolov, *The symmetry approach to classifciation of integrable equations*, in: What is Integrability?, Ed.: V.I. Zakharov (Springer Verlag, New York, 1990) 115–184.

[115] W. Miller, Jr., *Lie Theory and Special Functions* (Academic Press, New York, 1968).

[116] W. Miller, Jr., *Symmetry Groups and Their Applications*, Pure and Applied Mathematics **50** (Academic Press, New York, 1972).

[117] W. Miller, *Symmetry and Separation of Variables* (Addison–Wesley, Reading, Massachusetts, 1977).

[118] R.M. Miura, Ed., *Bäcklund Transformations, the Inverse Scattering Method, Solitons, and Their Applications* (Springer Verlag, New York, 1976).

[119] G.G. Nagao, *DETERMININGEQS: A Computer Program for Approximating Lie Generators Admitted by Dynamical Systems.* M.Sc. Thesis (University of the Pacific, Stockton, California, 1980).

[120] M.C. Nucci, *Interactive REDUCE Programs for Calculating, Classical, Non–classical, and Lie–Bäcklund Symmetries of Differential Equations*. Preprint GT Math. 062090–051, School of Mathematics (Georgia Institute of Technology, Atlanta, Georgia, 1990).

[121] M.C. Nucci, *Interactive REDUCE programs for calculating, classical, non–classical, and approximate symmetries of differential equations*, in: Proc. 13th IMACS World Congress, Dublin, Ireland, 1991. Eds.: R. Vichnevetsky and J.J.H. Miller (Criterion Press, Dublin, Ireland, 1991) 349–350.

[122] M.C. Nucci and P.A. Clarkson, Phys. Lett. A **164** (1992) 49–56.

[123] P.J. Olver, *Applications of Lie Groups to Differential Equations*, Graduate Texts in Mathematics **107** (Springer Verlag, New York, 1986 & II

[124] P.J. Olver, Appl. Num. Math. **10** (1992) 307–324.

[125] P.J. Olver, *Internal symmetries of differential equations*, in: Proc. of Modern Group Analysis: Advanced Analytical and Computational Methods in Mathematical Physics, Acireale, Catania, Italy, 1992, Eds.: N.H. Ibragimov, M. Torrisi, and A. Valenti (Kluwer Academic Publishers, Dordrecht, The Netherlands, 1993) in press.

[126] P.J. Olver, *Direct Reduction and Differential Constraints.* Preprint, Department of Mathematics (University of Maryland, College Park, Maryland, 1993).

[127] P.J. Olver and P. Rosenau, Phys. Lett. A **114** (1986) 107–112.

[128] P.J. Olver and P. Rosenau, SIAM J. Appl. Math. **47** (1987) 263–278.

[129] L.V. Ovsiannikov, *Group Properties of Differential Equations*, Siberian Section of the Academy of Sciences, USSR, Novosibirsk, 1962) in Russian, translated by G.W. Bluman (1967).

[130] L.V. Ovsiannikov, *Group Analysis of Differential Equations* (Academic Press, New York, 1982).

[131] R.S. Palais, *A Global Formulation of the Lie Theory of Transformation Groups*, Memoirs of the American Mathematical Society **22** (AMS, Providence, Rhode Island, 1957, 1989).

[132] F.A.E. Pirani, D.C. Robinson and W.F. Shadwick, *Local Jet Bundle Formulation of Bäcklund Transformations*, Mathematical Physics Studies **1** (Reidel, Boston, Massachusetts, 1979).

[133] H.J. Pohle and T. Wolf, Computing **41** (1989) 297–307.

[134] J.F. Pommaret, *Systems of Partial Differential Equations and Lie Pseudogroups* (Gordon & Breach, New York, 1978).

[135] J.F. Pommaret, *Lie Pseudogroups and Mechanics* (Gordon & Breach, New York, 1988).

[136] M.D. Popov, Izvestiya AN Belor. SSR, Ser. Phys.–Math. **2** (1985) 33–37.

[137] G.J. Reid, J. Phys. A: Math. Gen. **23** (1990) L853–L859.

[138] G.J. Reid, *Lie Theory, Differential Equations and Representation Theory*, Proc. Annual Seminar of the Canadian Math. Soc., Montréal, Canada, 1989, Ed.: V. Hussin, Les Publications de Centre de Recherches Mathématiques (Montréal, Québec, Canada, 1990) 363–372.

[139] G.J. Reid, Europ. J. Appl. Math. **2** (1991) 293–318.

[140] G.J. Reid, Europ. J. Appl. Math. **2** (1991) 319–340.

[141] G.J. Reid and A. Boulton, *Reduction of systems of differential equations to standard form and their integration using directed graphs*, in: Proc. ISSAC 91, Bonn, Germany, 1991, Ed.: S.M. Watt (ACM Press, New York, 1991) 308–312.

[142] G.J. Reid, I.G. Lisle, A. Boulton and A.D. Wittkopf, *Algorithmic determination of commutation relations for Lie symmetry algebras of PDEs*, in: Proc. ISSAC 92, Berkeley, California, 1992, Ed.: P.S. Wang (ACM Press, New York, 1992) 63–68.

[143] G.J. Reid and D.K. McKinnon, *Solving systems of linear PDEs in their coefficient field by recursively decoupling and solving ODEs*. Preprint, Department of Mathematics (University of British Columbia, Vancouver, Canada, 1993) submitted to J. Symb. Comp. (1993).

[144] G.J. Reid and A.D. Wittkopf, *Long Guide to the Standard Form Package*, Department of Mathematics (University of British Columbia, Vancouver, Canada, 1993).

[145] C. Riquier, *Les Systèmes d'Equations aux Dérivées Partielles* (Gauthier–Villars, Paris, 1910).

[146] J.F. Ritt, *Differential Algebra* (Dover, New York, 1966).

[147] C. Rogers and W.F. Ames, *Nonlinear Boundary Value Problems in Science and Engineering* (Academic Press, New York, 1989).

[148] C. Rogers and W.A. Shadwick, *Bäcklund Transformations and Their Applications* (Academic Press, New York, 1982).

[149] P. Rosenau and J.L. Schwarzmeier, *Similarity Solutions of Systems of Partial Differential Equations using MACSYMA*. Report COO–3077–160 MF–94, Magneto–Fluid Dynamics Division, Courant Institute of Mathematical Sciences (New York University, New York, New York, 1979).

[150] S.I. Rosencrans, Comp. Phys. Comm. **38** (1985) 347–356.

[151] W. Sarlet and F. Cantrijn, SIAM Review **23** (1981) 467–494.

[152] W. Sarlet and J. Vanden Bonne, J. Sym. Comp. **13** (1992) 683–693.

[153] C.C.A. Sastri, Contemp. Math. **54** (1986) 35–44.

[154] D.H. Sattinger, *Group Theoretic Methods in Bifurcation Theory* (Springer Verlag, New York, 1979).

[155] D.H. Sattinger and O.L. Weaver, *Lie Groups and Algebras with Applications to Physics, Geometry, and Mechanics*, Applied Mathematical Sciences **61** (Springer Verlag, New York, 1986).

[156] E. Schrüfer, *EXCALC, a system for doing calculations in the calculus of modern differential geometry−− User's Manual*, distributed with REDUCE 3.3 (1986).

[157] F. Schwarz, Comp. Phys. Comm. **27** (1982) 179–186.

[158] F. Schwarz, *Automatically determining symmetries of ordinary differential equations*, in: Proc. EUROCAL 83, Londen, England, 1983, Ed.: J.A. van Hulzen, Lecture Notes in Computer Science **162** (Springer Verlag, New York, 1982) 45–54.

[159] F. Schwarz, Physica D **11** (1984) 243–251.

[160] F. Schwarz, Computing **34** (1985) 91–106; Addendum: Computing **36** (1986) 279–280.

[161] F. Schwarz, *Symmetries and involution systems: some experiments in computer algebra*, in: Topics in Soliton Theory and Exactly Solvable Nonlinear Equations. Proc. Meeting on Nonlinear Evolution Equations, Oberwolfach, Germany, 1986, Eds.: M. Ablowitz, B. Fuchssteiner and M. Kruskal (World Scientific, Singapore, 1987) 290–299.

[162] F. Schwarz, *The Package SPDE for Determining Symmetries of Partial Differential Equations. User's Manual.* Distributed with REDUCE 3.3 (Rand Corporation, Santa Monica, California, 1987).

[163] F. Schwarz, *Trends in Computer Algebra*, in: Proc. Int. Symposium, Bad Neuenahr, Germany, 1987, Ed.: R. Janßen, Lecture Notes in Computer Science **296** (Springer Verlag, New York, 1988) 167–176.

[164] F. Schwarz, SIAM Review **30** (1988) 450–481.

[165] F. Schwarz, Computing **49** (1992) 95–115.

[166] F. Schwarz, *Reduction and completion algorithms for partial differential equations*, in: Proc. ISSAC 92, Berkeley, California, 1992, Ed.: P.S. Wang (ACM Press, New York, 1992) 49–56.

[167] J.L. Schwarzmeier and P. Rosenau, *Using MACSYMA to Calculate Similarity Transformations of Partial Differential Equations*. Report LA–UR 88–4157, Los Alamos National Laboratory (Los Alamos, New Mexico, 1988).

[168] L.I. Sedov, *Similarity and Dimensional Methods in Mechanics* (Academic Press, New York, 1959).

[169] W.A. Shadwick, *The Bäcklund Problem, Symmetries and Conservation Laws for Some Nonlinear Differential Equations*. Ph.D Thesis, Department of Mathematics (Kings College, London, 1979).

[170] J. Sherring and G. Prince, *DIMSYM – Symmetry Determination and Linear Differential Equations Package*. Preprint, Department of Mathematics (LaTrobe University, Bundoora, Australia, 1992).

[171] M.F. Singer, J. Symb. Comp. **10** (1991) 59–94.

[172] W.–H. Steeb, *Invertible Point Transformations and Nonlinear Differential Equations* (World Scientific, Singapore, 1993).

[173] S. Steinberg, *Symmetry operators*, in: Proc. 1979 MACSYMA Users Conference, Washington, D.C., 1979, Ed.: V.E. Lewis (MIT Press, Boston, Massachusetts, 1979) 408–444.

[174] S. Steinberg, *Lie series, Lie transformations, and their applications*, in: Lie Methods in Optics. Proc. CIFMO–CIO Workshop, Leon, Mexico, 1985, Eds.: J. Sánchez Mondragón and K.B. Wolf, Lecture Notes in Physics **250** (Springer Verlag, New York, 1986) 45–103.

[175] S. Steinberg, MACSYMA Newsletter **7** (1990) 3–7.

[176] H. Stephani, *Differential Equations: Their Solution using Symmetries* (Cambridge University Press, Cambridge, 1989).

[177] J. Thomas, *Differential Systems*, Colloquium Publications **21** (AMS, New York, New York, 1937).

[178] V.L. Topunov, Acta Appl. Math. **16** (1989) 191–206.

[179] A.M. Tresse, Acta Math. **18** (1894) 1–88.

[180] A.M. Tresse, *Détermination des Invariants Ponctuel de l'Equation Différentielle Ordinaire du Second Ordre* $y'' \Subset \omega(x, y, y')$ (S. Hirzel, Leipzig, 1896).

[181] T. Tsujishita, Osaka J. Math. **19** (1982) 311–363.

[182] P. Vafeades, *Computer algebraic determination of symmetries and conservation laws of PDEs*, in: Proc. ISMM Int. Symposium on Computer Applications in Design, Simulation and Analysis, New Orleans, Louisiana, 1990, Ed.: E.K. Park (Acta Press, Anaheim, California, 1990) 310–312.

[183] P. Vafeades, *SYMCON: A MACSYMA Package for the Determination of Symmetries and Conservation Laws of PDEs*. Preprint, Department of Engineering Sciences (Trinity University, San Antonio, Texas, 1990)

[184] P. Vafeades, MACSYMA Newsletter **9**, no. 1 (1992) 1–13.

[185] P. Vafeades, MACSYMA Newsletter **9**, no. 2–4 (1992) 5–20.

[186] P. Vafeades, MACSYMA Newsletter **10**, no. 1 (1993) to appear.

[187] A. Vanderbauwhede, *Local Bifurcation and Symmetry* (Pitman, London, 1982).

[188] A.M. Vinogradov, Acta Appl. Math. **2** (1984) 21–78.

[189] A.M. Vinogradov, Acta Appl. Math. **16** (1989) 3–21.

[190] A.M. Vinogradov, Ed., *Symmetries of Partial Differential Equations*, Part I, Acta Appl. Math. **15**, nos. 1 & II*ibid.* **16**, no. 1 (1989); Part III, *ibid.* **16**, no. 2 (1989).

[191] A.M. Vinogradov, Ed., *Symmetries of Partial Differential Equations. Conservation laws – Applications – Algorithms* (Kluwer Academic Publications, Dordrecht, The Netherlands, 1989).

[192] A.M. Vinogradov and I.S. Krasilshchik, Acta Appl. Math. **2** (1984) 79–96.

[193] E.M. Vorob'ev, Acta Appl. Math. **16** (1991) 1–24.

[194] E.M. Vorob'ev, Acta Appl. Math. **26** (1992) 61–86.

[195] H.D. Wahlquist, *Differential form analysis using MACSYMA*, in: Proc. 1977 MACSYMA Users Conference, Berkeley, California, 1977, Ed.: E.M. Andersen (NASA CP–2012, Washington, D.C., 1977) 71–74.

[196] F.W. Warner, *Foundations of Differential Manifolds and Lie Groups* (Springer Verlag, New York, 1983).

[197] P. Winternitz, *Lie groups and solutions of nonlinear differential equations*, in: Nonlinear Phenomena, Ed.: K.B. Wolf, Lecture Notes in Physics **189** (Springer Verlag, New York, 1983) 263–331.

[198] P. Winternitz, *Group theory and exact solutions of partially integrable differential systems*, in: Partially Integrable Nonlinear Evolution Equations and their Physical Applications. NATO ASI Series C: Math. and Phys. Scs. **310**, Eds.: R. Conte and N. Boccara (Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990) 515–567.

[199] J. Sánchez Mondragón and K.B. Wolf, Eds., *Lie Methods in Optics*, Proc. CIFMO–CIO Workshop, Léon, Guanajuato, México, 1985, Lecture Notes in Physics **250** (Springer Verlag, New York, 1986).

[200] K.B. Wolf, Ed., *Lie Methods in Optics II*, Proc. CIFMA–CIO Workshop, Cocoyoc, México, 1988, Lecture Notes in Physics **352** (Springer Verlag, New York, 1990).

[201] T. Wolf, J. Comp. Phys. **60** (1985) 437–446 .

[202] T. Wolf, *Zur analytischen Untersuching und exakten Lösung von Differentialgleichungen mit Computeralgebrasystemen*. Dissertation B (Friedrich Schiller Universität, Jena, Germany, 1989).

[203] T. Wolf, *A package for the analytic investigation and exact solutions of differential equations*, in: Proc. EUROCAL 87, Leipzig, GDR, 1987, Ed.: J.H. Davenport, Lecture Notes in Computer Science **378** (Springer Verlag, Berlin, 1989) 479–491.

[204] T. Wolf, *The symbolic integration of exact PDEs*, Preprint School of Mathematical Sciences (University of London, London, 1992) submitted to J. Symb. Comp. (1993).

[205] T. Wolf, *An efficiency improved program LIEPDE for determining Lie−symmetries of PDEs*, in: Proc. of Modern Group Analysis: Advanced Analytical and Computational Methods in Mathematical Physics, Acireale, Catania, Italy, 1992, Eds.: N.H. Ibragimov, M. Torrisi, and A. Valenti (Kluwer Academic Publishers, Dordrecht, The Netherlands, 1993) in press.

[206] T. Wolf and A. Brand, *Heuristics for overdetermined systems of PDEs*, in: Proc. Meeting on Artificial Intelligence in Mathematics, Glasgow, United Kingdom, 1991, Eds.: J.H. Johnson, S. McKee and A. Vella (Clarendon Press, Oxford, 1993) in press.

[207] T. Wolf and A. Brand, *The computer algebra package CRACK for investigating PDEs*, in: Proc. ERCIM Scool on Partial Differential Equations and Group Theory, Bonn, 1992, Ed.: J.F. Pommaret (Gesellschaft für Mathematik und Datenverarbeitung, Sankt Augustin, Germany, 1992); Also: Manual for CRACK added to the REDUCE Network Library, School of Mathematical Sciences (University of London, London, 1992) in press.

[208] T. Wolf and A. Brand, *Investigating symmetries and other analytical properties of ODEs with the computer algebra package CRACK*, submitted to: 'Exploiting Symmetry in Applied and Numerical Analysis, Proceedings of the AMS−SIAM Summer Seminar, Fort Collins, 1992, Eds.: E. Allgower, K. Georg and R. Miranda, Lectures in Applied Mathematics (AMS, Providence, Rhode Island, 1993) in press.

[209] C.E. Wulfman, *Systematic methods for determining the continuous transformation groups admitted by differential equations*, in: Symmetries in Science, Eds.: G. Gruber and R.S. Millman (Plenum, New York, 1980) 435–457.

[210] I.M. Yaglom, *Felix Klein and Sophus Lie* (Birkhäuser, Basel, 1988).

[211] H. Zhang, G. Tu, W. Oevel and B. Fuchssteiner J. Math. Phys. **32** (1991) 1908–1918.

[212] A. Yu. Zharkov, J. Symb. Comp. **15** (1993) 85–90.

Table of contents

1 Introduction 45

2 Purpose, Methods and Algorithms 46

2.1 Purpose 46

2.2 Methods for Computing Lie Symmetries 47

2.3 Prolonged Vector Fields 47

2.4 Differential Forms 50

2.5 Methods for Solving the Determining Equations 50

2.6 Lie Symmetry Algorithms 52

3 Beyond Lie–point Symmetries 52

3.1 Contact and Generalized Symmetries 52

3.2 Nonclassical or Conditional Symmetries 53

4 Review of Symbolic Software 53

4.1 REDUCE Programs 53

4.2 MACSYMA Programs 55

4.3 MAPLE Programs 56

4.4 MATHEMATICA Programs 56

4.5 AXIOM Programs 57

4.6 muMATH Programs 57

4.7 Programs for Other Systems 58

5 Solving the Determining Equations and Computation of the Size of the Symmetry Group 59

6 Examples 61

6.1 The Harry Dym Equation 61

6.2 The Magneto–Hydro–Dynamics Equations 63

7 Further Research and Development 66

8 Conclusion 67

9 Acknowledgements 67