Serdica J. Computing **1** (2007), 505–518

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

A COMPUTER ALGEBRA APPLICATION TO DETERMINATION OF LIE SYMMETRIES OF PARTIAL DIFFERENTIAL EQUATIONS*

Vladimir I. Pulov, Edy J. Chacarov, Ivan M. Uzunov

Abstract. A *MATHEMATICA* package for finding Lie symmetries of partial differential equations is presented. The package is designed to create and solve the associated determining system of equations, the full set of solutions of which generates the widest permissible local Lie group of point symmetry transformations. Examples illustrating the functionality of the package's tools are given. The results of the package application to performing a full Lie group analysis of coupled nonlinear Schrödinger equations from nonlinear fiber optics are presented. Comparisons with earlier published computer algebra implementations of the Lie group method are discussed.

1. Introduction. It is well known that the Lie groups of symmetry transformations leaving a system of partial differential equations (PDEs) invariant can be very useful in a wide range of mathematical and physical applications

ACM Computing Classification System (1998): G.4, I.1.2, I.1.4, J.2.
Key words: MATHEMATICA package, Lie symmetries, partial differential equations.

*The paper has been presented at the 12th International Conference on Applications of Computer Algebra, Varna, Bulgaria, June, 2006.

[8]. We will only mention a few of them, those with the possibility to obtain new solutions from known ones, the symmetry reduction technique leading to a simplification of the original system, explicit determination of group invariant solutions and their classification, derivation of conservation laws reflecting the underlined physical phenomenon. The key to finding the symmetry group is the infinitesimal generator of the group—a vector field on the space of the independent and the dependent variables. In order to provide a basis of group generators one has to create and solve the so-called determining system of equations (DSEs). Although the method is algorithmically straightforward, it often appears to be formidably difficult to follow, which is primarily due to the great number of symbolic calculations that must be performed. The operations are routine, but nonetheless very tedious and time-consuming to do by hand. It often occurs that hundreds of equations are manipulated while creating and solving the DSEs, which makes essential the use of a certain computer algebra system (CAS), such as Reduce, *MATHEMATICA*, Maple, etc.

The goal of this paper is to present a computer algebra implementation of the Lie method—the *MATHEMATICA* package *LieSymm-PDE* that we have developed in order to overcome the computational difficulties. The package is designed to create and solve the associated DSEs. It covers the most general case of an arbitrary system of PDEs without any restrictions on the number of the equations, on the number of the independent and dependent variables, and as well as on the highest order of the derivatives that may be involved. To the authors' knowledge, other programs related to Lie symmetries have been developed in Reduce [14], *MATHEMATICA* [5], Maple[1]. The algorithm of *LieSymm-PDE* for solving the DSEs is closely related to the solving technique of [14]. In comparison with the "liesymm" package of Maple the package we present can be used not only for creating the DSEs but also for solving it. *LieSymm-PDE* works with less external advice to solve the DSEs than it is needed by the *MATHEMATICA* program in [4].

The paper is organized as follows. In Section 2 we give a short description of the problem. In Section 3 we explain the algorithm of *LieSymm-PDE*. In Section 4 we illustrate how the tools of *LieSymm-PDE* work. In Section 5 we give the results of the package application to equations from nonlinear fiber optics. In Section 6 we discuss the general features of *LieSymm-PDE* in comparison with other Lie packages.

2. Lie symmetries of PDEs: theoretical background. We are going to give a brief outline of the Lie method using the terms and notations in [8].

[1] The package "liesymm" included in the standard release of Maple 10.

Let F be a given system of l PDEs of order n for q functions $u = (u^1, \ldots, u^q) \in U \equiv R^q$ in p independent variables $x = (x^1, \ldots, x^p) \in X \equiv R^p$:

$$F(x, u^{(n)}) = 0, \tag{1}$$

where $u^{(n)}$ denotes a point in the Euclidean space $U^{(n)}$ having as coordinates the dependent variables u^α and the derivatives $u^\alpha_{j_1 \cdots j_s} \equiv \partial u^\alpha / \partial x^{j_1} \ldots \partial x^{j_s}$ of order $s = 1, \ldots, n$, $\alpha = 1, \ldots, q$, $j_\nu = 1, \ldots, p$, $\nu = 1, \ldots, s$; $F = (F_1, \ldots, F_l)$. Note that no limitations for the nonlinear properties of the left-hand sides are demanded, i.e., $F(x, u^{(n)})$ can be a nonlinear function of all its arguments. It is said that the system (1) admits a one-parameter local Lie group of point transformations $G = \{T_a | a \in \Delta \subset R,\ 0 \in \Delta\}$ with

$$T_a = \begin{cases} x' = f(a, x, u) \\ u' = \varphi(a, x, u) \end{cases} \tag{2}$$

(a is the group parameter) if it has the property of being unaltered after the transformations of the group. The admitted group is also called a group of Lie symmetries of the considered PDEs in the sense that each solution of (1) after the transformations of the group remains a solution. Finding the admitted Lie groups of a system of PDEs is based on the fundamental correspondence between the Lie groups and their Lie algebras of infinitesimal generators. The infinitesimal generator is a first-order linear differential operator

$$V = \sum_{i=1}^{p} \xi^i(x, u) \frac{\partial}{\partial x^i} + \sum_{\alpha=1}^{q} \eta^\alpha(x, u) \frac{\partial}{\partial u^\alpha}, \tag{3}$$

where $\xi^i(x, u) = \partial f^i(0, x, u)/\partial a$, $\eta^\alpha(x, u) = \partial \varphi^\alpha(0, x, u)/\partial a$; $f = (f^1, \ldots, f^p)$, $\varphi = (\varphi^1, \ldots, \varphi^q)$. From a geometrical point of view, V is a tangent vector field on $Z = X \times U$, which flow coincides with a one-parameter group of transformations, that is, if the coordinates $\xi = (\xi^1, \ldots, \xi^p)$, $\eta = (\eta^1, \ldots, \eta^q)$ of the vector field are given, then the group (2) can be obtained by solving the equations [Lie equation]

$$\begin{aligned} \frac{df}{da} &= \xi(f, \varphi), \quad f_{|a=0} = x, \\ \frac{d\varphi}{da} &= \eta(f, \varphi), \quad \varphi_{|a=0} = u. \end{aligned} \tag{4}$$

The infinitesimal generators of all one-parameter groups admitted by the considered differential equations constitute a Lie algebra—a vector space supplied with a Lie bracket operation $[V_1, V_2] = V_1 V_2 - V_2 V_1$.

The milestone of the Lie method is the infinitesimal criterion that works on the base of a special technique for prolongation of the groups and their infinitesimal generators. Assume that the rank of the Jacobi matrix of $F(x, u^{(n)})$ is l whenever the point $(x, u^{(n)})$ belongs to the sub-manifold $\Delta_F \subset X \times U^{(n)}$ defined by (1). Then the system of PDEs (1) admits a one-parameter group G with the infinitesimal generator V if and only if the following infinitesimal condition holds

$$\text{pr}^{(n)}V\left[F(x, u^{(n)})\right] = 0 \text{ for } (x, u^{(n)}) \in \Delta_F, \tag{5}$$

where $\text{pr}^{(n)}V$ is the n^{th} prolongation of V

$$\text{pr}^{(n)}V = V + \sum_{i=1}^{p}\sum_{\alpha=1}^{q}\zeta_i^\alpha \frac{\partial}{\partial u_i^\alpha} + \cdots + \sum_{j_1=1}^{p}\cdots\sum_{j_n=1}^{p}\sum_{\alpha=1}^{q}\zeta_{j_1\dots j_n}^\alpha \frac{\partial}{\partial u_{j_1\dots j_n}^\alpha}. \tag{6}$$

The coefficients $\zeta_{j_1\dots j_k}^\alpha$, $k = 1, \dots, n$ depend on the functions $\xi(x, u)$ and $\eta(x, u)$ by the recursive formulae

$$\begin{aligned} \zeta_i^\alpha &= D_i(\eta^\alpha) - \sum_{s=1}^{p} u_s^\alpha D_i(\xi^s), \\ \zeta_{j_1\dots j_k}^\alpha &= D_{j_k}(\zeta_{j_1\dots j_{k-1}}^\alpha) - \sum_{s=1}^{p} u_{j_1\dots j_{k-1}s}^\alpha D_{j_k}(\xi^s), \end{aligned} \tag{7}$$

where D_i is the operator of total differentiation with respect to the variable x^i

$$\begin{aligned} D_i = \frac{\partial}{\partial x^i} + \sum_{\alpha=1}^{q} u_i^\alpha \frac{\partial}{\partial u^\alpha} &+ \sum_{j=1}^{p}\sum_{\alpha=1}^{q} u_{ji}^\alpha \frac{\partial}{\partial u_j^\alpha} + \cdots \\ &+ \sum_{j_1=1}^{p}\cdots\sum_{j_{n-1}=1}^{p}\sum_{\alpha=1}^{q} u_{j_1\dots j_{n-1}i}^\alpha \frac{\partial}{\partial u_{j_1\dots j_{n-1}}^\alpha}. \end{aligned} \tag{8}$$

The equation (5) serves to determine the coefficients $\xi(x, u)$ and $\eta(x, u)$ of the infinitesimal generators that constitute the widest Lie algebra admitted by the system (1). If this algebra has a finite dimension r, then the corresponding Lie symmetry group is an r-parameter group of transformations of the space Z. In order for this group to be found one has to solve the equation (5), and then to take some basis $(V_1, V_2, \dots, V_r)$ of its solutions. By solving the Lie equation (4) for each one of these basic infinitesimal generators, the corresponding one-parameter groups of symmetry transformations $T_{a_1}, T_{a_2}, \dots, T_{a_r}$ are obtained.

Their composition performs the general r-parameter group of transformations T_a with the vector parameter $a = (a_1, a_2, \dots, a_r)$.

3. Algorithm of the package *LieSymm-PDE*. In order to find the unknown coefficient functions $\xi^i(x,u)$, $\eta^\alpha(x,u)$, one has to solve the equation (5). Since the variables $x^i, u^\alpha, u^\alpha_{j_1 \dots j_s}$ are assumed to be independent, the equation (5) can be facilitated by equating to zero all the coefficients of the monomials in the partial derivatives $u^\alpha_{j_1 \dots j_s}$. Thus, a large number of linear homogeneous partial differential equations are obtained, which are known as the DSEs of the symmetry group admitted by (1). When systems of PDEs of order higher than two are considered and the independent variables are more than about two, the DSEs may consist of hundreds of equations. In situations like this, it is essential to use a contemporary CAS to tackle the great number of symbolic manipulations.

The package *LieSymm-PDE* presented here utilizes the tools of the CAS *MATHEMATICA*. It is designed to create the DSEs and to provide automatic assistance for solving it. The algorithm flow (Fig. 1) of the package goes strictly through the theoretical formulae in the preceding section. It involves the following steps.

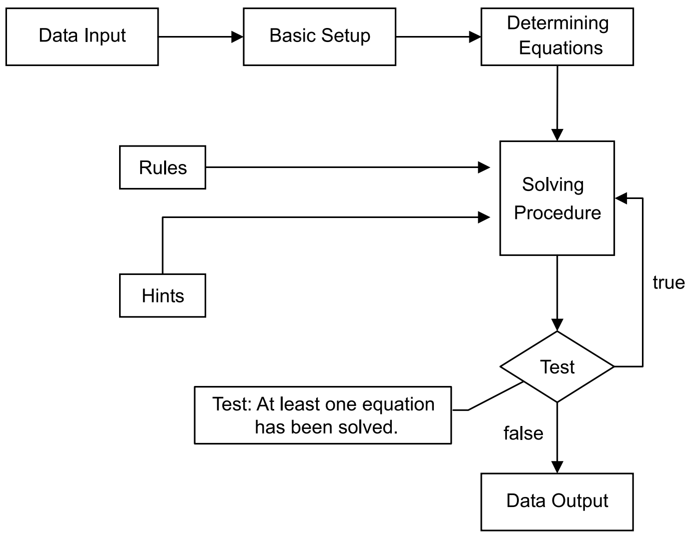

Fig. 1. Flowchart of the algorithm

Step 1. *Data Input*. The initially given data provide all the information that is needed to create the DSEs. These data consist of the differential equations (1) in their full explicit form and a detailed description of the types of the variables used in them.

Step 2. *Basic Setup.* At this step, some basic symbolic expressions are generated. They compose a setup of symbols, rules and operators prepared on the basis of the data input and satisfying the specific requirements of the algorithm's logic. We should point out two of them, those of the submanifold Δ_F represented by a list of rules, and of the prolonged group generator $\mathrm{pr}^{(n)}V$ defined as an operator. They are both accomplished in full accordance with the definitions – formulae (1), (6–8). Two other lists of variables stand for the unknown coefficients $\xi^i(x,u)$ and $\eta^\alpha(x,u)$ of the infinitesimal generator (3). At this stage of the program they are merely $p+q$ arbitrary functions.

Step 3. *Determining Equations.* The infinitesimal criterion of the Lie group method is carried out, i.e., the prolonged infinitesimal generator $\mathrm{pr}^{(n)}V$ is applied to the functions $F_1, F_2, \ldots, F_l$ with the resultant expressions being recalculated on the submanifold Δ_F. Then, after equating to zero the coefficients of the monomials in $u^\alpha_{j_1 \ldots j_s}$, a list of the determining equations is obtained.

Step 4. *Solving procedure.* At the fourth step of the algorithm a solving procedure is started up. It consists of a repetition of special programming modules for solving differential and algebraic equations. Each module searches for determining equations that belong to some specific class of equations with known solutions. If such an equation has been found, its solution is substituted for the respective variable in the remainder of the equations. As a result, the functions $\xi^i(x,u)$ and $\eta^\alpha(x,u)$ change, getting closer to the exact explicit solution and the number of the equations in the DSEs diminishes by those of them that have been already solved. The modules available by the package cope with the following five types of equations

$$
\begin{gathered}
C_1 x + C_2 = 0, \quad C_1 x + C_2 y = 0, \\
C_1 y_x + C_2 = 0, \quad C_1 y_{xx} + C_2 = 0, \quad C_1 y_{xxx} + C_2 = 0
\end{gathered} \tag{9}
$$

($C_1, C_2 =$ const. in regard to x and y; $y_x \equiv \partial y/\partial x$, etc.). The solving process is completed when either the number of the determining equations has been reduced to zero, or all of the remaining equations have become unsolvable by the existing modules.

Note that there is a possibility that the DSEs can be simplified at the beginning of the solving procedure. This is achieved with the aid of two programming tools: *Rules* and *Hints* (Fig. 1). *Rules* is a collection of modules for making transformations such as for adding, subtracting, and differentiating equations. One special module is designated to carry out a search for functionally independent parts of the equations that, after being equated to zero, are added to the list of the DSEs. *Hints* is a list of substitutions provided by the user in order to specify the functions being sought.

Step 5. *Data Output*. Data that is generated at the output consist of the solution $(\xi^1, \dots, \xi^p, \eta^1, \dots, \eta^q)$ that is possibly expressed by some unknown functions and a list of equations that these functions must satisfy.

4. MATHEMATICA tools of *LieSymm-PDE*. The program works in partly interactive mode allowing the user to effectively participate in the solving process by applying pre-determined transformation rules and by giving hints to the solutions. This is needed in view of the fact that no general solution scheme of the DSEs is known to date. There are also user-level commands used to display any current state of the DSEs and its solution helping the user to decide which rules and hints to apply before each intermediate run of the package. A usage message explains all that is needed to execute the program. For instance, each of the original equations must be solved in regard to any variable involved – independent, dependent, or any of the derivatives – and this single variable must be typed for the left-hand side of the equation. The program also needs to know the independent and the dependent variables. Firstly, based on the data input the determining equations are created, and secondly, a solving procedure that aims at finding the general solution of the DSEs is carried out. Note that the package contains private context specification, which protects the objects from getting confused with other objects defined outside the package and having the same names.

In order to illustrate the basic tools of the package we consider, as a first and simplest example, the heat equation

$$u_t - u_{xx} = 0, \tag{10}$$

whose full symmetry group is well known. The function that creates the determining system is named **CreateDSE[ ]**. All the initially given data are provided by this command. Another command **DetSysEqs** is used to exhibit the DSEs. In *MATHEMATICA* notation, for equation (10), these read (with $a[1]^{(1,0,2)}[x,t,u] \equiv \partial^3 a[1]/\partial x \partial u^2$, etc.):

CreateDSE[{u[t]},{u[x, x]},{x, t},{u}]; DetSysEqs

Eqn[1]: $a[2]^{(0,0,1)}[x,t,u] == 0$

Eqn[2]: $a[1]^{(0,0,2)}[x,t,u] == 0$

Eqn[3]: $a[2]^{(0,0,2)}[x,t,u] == 0$

Eqn[4]: $a[2]^{(1,0,0)}[x,t,u] == 0$

Eqn[5]: $-a[3]^{(0,0,2)}[x,t,u] + 2a[1]^{(1,0,1)}[x,t,u] == 0$

Eqn[6]: $2a[1]^{(0,0,1)}[x,t,u] + 2a[2]^{(1,0,1)}[x,t,u] == 0$

Eqn[7]: $-a[1]^{(0,1,0)}[x,t,u] - 2a[3]^{(1,0,1)}[x,t,u] + a[1]^{(2,0,0)}[x,t,u] == 0$

Eqn[8]: $-a[2]^{(0,1,0)}[x,t,u] + 2a[1]^{(1,0,0)}[x,t,u] + a[2]^{(2,0,0)}[x,t,u] == 0$

Eqn[9]: $a[3]^{(0,1,0)}[x,t,u] - a[3]^{(2,0,0)}[x,t,u] == 0$

The result is a system of nine differential equations for three functions that coincide with the coefficients ξ_1, ξ_2 and η_1 of the symmetry generator as is shown by:

LieInfGen

$\xi_1 = a[1][x,t,u], \quad \xi_2 = a[2][x,t,u], \quad \eta_1 = a[3][x,t,u]$

By the use of the *LieSymm-PDE* iterative function **SolveDSE**[ ], special solving modules are applied repeatedly in sequence to the determining equations in order to identify and solve those of them that match any of the pre-determined types of equations (9) (in *MATHEMATICA* output a prime means differentiation):

SolveDSE[DetSysEqs]; LieInfGen

$\xi_1 = a[6][x,t], \quad \xi_2 = a[7][t], \quad \eta_1 = ua[8][x,t] + a[9][x,t]$

DetSysEqs

Eqn[1]: $-a[7]'[t] + 2a[6]^{(1,0)}[x,t] == 0$

Eqn[2]: $-a[6]^{(0,1)}[x,t] - 2a[8]^{(1,0)}[x,t] + a[6]^{(2,0)}[x,t] == 0$

Eqn[3]: $a[8]^{(0,1)}[x,t] - a[8]^{(2,0)}[x,t] == 0$

Eqn[4]: $a[9]^{(0,1)}[x,t] - a[9]^{(2,0)}[x,t] == 0$

As a result, the coefficients of the infinitesimal generator are expressed by four new functions that satisfy a smaller number of equations. Instead of trying to solve them by hand the user can take advantage of the additional tools of *LieSymm-PDE*—the commands **SplitDSE**[ ], **DiffDSE**[ ], **AddDSE**[ ]. They provide automatic equivalent transformations of the DSEs that are, respectively, for splitting up of polynomials to functionally independent terms, for differentiating of equations, for adding and subtracting pairs of equations. By trial and error the user decides which particular set of commands shall be applied so that more equations can be simplified. For the considered case it is most efficient to use the series of operators:

SplitDSE[DiffDSE[DiffDSE[DetSysEqs]]];
SolveDSE[DetSysEqs]; DetSysEqs
Eqn[1]: $a[9]^{(0,1)}[x,t] - a[9]^{(2,0)}[x,t] == 0$
LieInfGen
$\xi_1 = ta[19][\,] + a[20][\,] + (1/2)xta[23][\,] + (1/2)xa[24][\,]$
$\xi_2 = (1/2)t^2a[23][\,] + ta[24][\,] + a[25][\,]$
$\eta_1 = a[9][x,t] - (1/2)xua[19][\,] + ua[22][\,] - (1/8)x^2ua[23][\,]$
$\quad -(1/4)tua[23][\,]$

The result obtained exactly corresponds to the full explicit solution given elsewhere, e.g., [8]. It includes the function $a[9][x,t]$ that satisfies the heat equation and six other constant functions $a[19][\,], a[20][\,], \ldots$. For the present example, the final solution can be also obtained by a single run of the full function **LieInfGen[{u[t]}, {u[x, x]}, {x, t}, {u}]**.

In order to demonstrate the usefulness of another user-level tool of *Lie-Symm-PDE*—the function named **Hints**[ ], we consider as a second example Burgers' equation

$$u_t - u_{xx} - u_x^2 = 0. \tag{11}$$

It differs from the heat equation by the nonlinear term u_x^2. The symmetries of Burgers' equation are very similar to the symmetries of the heat equation. Despite that, the solving modules and the transformation rules available by the package are not enough to solve all of the determining equations—there still remain four unsolved equations for four unknown functions:

Eqn[1]: $2a[7][t] - a[6]'[t] == 0$
Eqn[2]: $2a[7][t] - a[6]'[t] - a[3]^{(0,0,1)}[x,t,u] - a[3]^{(0,0,2)}[x,t,u] == 0$
Eqn[3]: $-xa[7]'[t] - a[8]'[t] - 2a[3]^{(1,0,0)}[x,t,u] - 2a[3]^{(1,0,1)}[x,t,u] == 0$
Eqn[4]: $a[3]^{(0,1,0)}[x,t,u] - a[3]^{(2,0,0)}[x,t,u] == 0$

In cases like this, it suffices that the user could derive some additional information from the returned equations that are fed back as hints to the solving modules. This is achieved by the special command **Hints**[ ], whose input consists of a list of substitutions. For the equation at hand (11) the desired information is obtained by applying the command **SplitDSE[AddDSE [DetSysEqs]]**. One of the sixteen equations that are retrieved is suitable for producing a hint:

$$a[3]^{(0,0,1)}[x,t,u] + a[3]^{(0,0,2)}[x,t,u] == 0$$

The hint reads **Hints**$[\{a[3] = f1[x,t]e^{-u} + f2[x,t]\}, \{f1[x,t], f2[x,t]\}]$. The variables put in the second curly brackets are considered by *LieSymm-PDE* as new unknown functions. The determining system is then simplified by *LieSymm-PDE* to:

$$\begin{aligned}
&\text{Eqn[1]:} && 2\mathrm{e}^u a[7][t] - \mathrm{e}^u a[6]'[t] == 0\\
&\text{Eqn[2]:} && -\mathrm{e}^u x a[7]'[t] - \mathrm{e}^u a[8]'[t] - 2\mathrm{e}^u a[10]^{(1,0)}[x,t] == 0\\
&\text{Eqn[3]:} && a[9]^{(0,1)}[x,t] + \mathrm{e}^u a[10]^{(0,1)}[x,t] - a[9]^{(2,0)}[x,t] -\\
& && \mathrm{e}^u a[10]^{(2,0)}[x,t] == 0
\end{aligned}$$

These three equations are easily solved in a fully automatic way by the help of the existing transformation rules ($a[9][x,t]$ is an arbitrary solution of the heat equation):

$$\begin{aligned}
\xi_1 &= ta[18][\,] + a[19][\,] + (1/2)xta[22][\,] + (1/2)xa[23][\,]\\
\xi_2 &= (1/2)t^2 a[22][\,] + ta[23][\,] + a[24][\,]\\
\eta_1 &= \mathrm{e}^{-u} a[9][x,t] - (1/2)xa[18][\,] + a[21][\,] -\\
&\quad (1/8)x^2 a[22][\,] - (1/4)ta[22][\,]
\end{aligned}$$

The result is equivalent to the coefficients of the general infinitesimal generator admitted by Burgers' equation given in [8].

In both examples the obtained algebras are with infinite dimension, which is due to the inclusion of the function $a[9][x,t]$ in the final solution. This is a clear manifestation of the package's capabilities to work also in infinite dimensional cases. In the next section we present the results of the package application to a system of nonlinear equations from fiber optics.

5. Coupled nonlinear Schrödinger equations. The package *LieSymm-PDE* has been applied to basic models of nonlinear fiber optics [10]–[13]. As a consequence of this, a full Lie group analysis of the respective systems of PDEs has been prepared and the optimal set of reduced equations obtained. Here we present the results of the package application to coupled nonlinear Schrödinger equations (CNSEs)

$$
(12)\qquad
\begin{aligned}
&iA_x + \frac{\nu_1}{2}A_{tt} + (|A|^2 + h|B|^2)A = 0,\\
&iB_x + \frac{\nu_2}{2}B_{tt} + (|B|^2 + h|A|^2)B = 0
\end{aligned}
$$

describing pulse propagation in optical fibers [1]. The functions $A(t, x)$ and $B(t, x)$ are the complex electric field amplitudes (normalized) depending on the dimensionless time t and space coordinate x. The parameter h has the following physically relevant values: $h = 2/3$ for strong birefringent fibers and $h = 2$ for two waves at different carrier wavelengths in two-mode fibers. The coefficients ν_i, $i = 1, 2$ determine the dispersion regime of the two modes: normal for $\nu_i = -1$ and anomalous for $\nu_i = 1$. By the help of the *LieSymm-PDE* tools for making equivalent transformations, without giving hints, 139 determining equations were solved. The result consists of eight basic infinitesimal generators ($A = u + iv$, $B = w + is$; $\partial_t \equiv \partial/\partial t$, $\partial_x \equiv \partial/\partial x, \ldots$):

$$
\begin{aligned}
&V_1 = \partial_t,\ V_2 = \partial_x,\ V_3 = u\partial_v - v\partial_u,\ V_4 = w\partial_s - s\partial_w,\\
&V_5 = x\partial_t + \nu_1 t(u\partial_v - v\partial_u) + \nu_2 t(w\partial_s - s\partial_w),\\
&V_6 = -t\partial_t - 2x\partial_x + u\partial_u + v\partial_v + w\partial_w + s\partial_s,\\
&V_7 = \nu_1\nu_2 w\partial_u - u\partial_w + s\partial_v - \nu_1\nu_2 v\partial_s,\\
&V_8 = s\partial_u - \nu_1\nu_2 u\partial_s + v\partial_w - \nu_1\nu_2 w\partial_v.
\end{aligned}
$$

They compose an 8-dimensional Lie algebra admitted by the CNSEs (12) for the case when $h\nu_1\nu_2 = 1$. The sub-algebra built upon the first six generators applies for arbitrary h, $h = 2$, $2/3$ and for $h\nu_1\nu_2 = -1$. In addition, as a consequence of the application of the package, we give the optimal set of one-dimensional Lie sub-algebras valid for $\nu_1 = 1$, $h = \nu_2 = -1$, which we present in a compact form of eighteen unified cases:

(A1) $V_1 + \epsilon V_3$, $\epsilon = 0, \pm 1$, (A2) $V_1 + \epsilon V_7$, $\epsilon = \pm 1$,
(A3) $\epsilon V_1 + 2V_3 + \delta V_7$, $\epsilon, \delta = \pm 1$,
(B1) $\epsilon V_4 + V_5$, $\epsilon = 0, \pm 1$, (B2) $V_5 + \epsilon V_7$, $\epsilon = \pm 1$,
(B3) $2V_4 + \epsilon V_5 + \delta V_7$, $\epsilon, \delta = \pm 1$,
(C1) $V_2 + \delta V_3 + \epsilon V_4$, $\epsilon = 0, \delta = 0, \pm 1$ or $\epsilon = \pm 1, \delta \in R$,
(C2) $V_2 + \epsilon(V_3 - V_4) + \delta V_7$, $\epsilon = \pm 1, \delta \in R\backslash\{0\}$ or $\epsilon = 0, \delta = \pm 1$,

(C3) $\epsilon V_2 + 2V_3 + \delta V_7$, $\epsilon V_2 + 2V_4 + \delta V_7$, $\epsilon V_2 + V_3 + V_4 + \delta V_7, \epsilon, \delta = \pm 1$,
(D1) $\epsilon V_2 + \delta V_4 + V_5$, $\epsilon = \pm 1, \delta \in R$,
(D2) $\epsilon V_2 + V_5 + \delta V_7$, $\epsilon = \pm 1, \delta \in R\backslash\{0\}$,
(D3) $\epsilon V_2 + 2V_4 + \delta V_5 + \sigma V_7$, $\epsilon, \sigma = \pm 1, \delta \in R\backslash\{0\}$,
(E1) $\epsilon V_3 + \delta V_4 + V_6$, $\epsilon, \delta \in R$,
(E2) $\epsilon(V_3 - V_4) + V_6 + \delta V_7$, $\epsilon \in R, \delta \in R\backslash\{0\}$,
(E3) $2V_3 + \delta V_6 + \epsilon V_7$, $2V_4 + \delta V_6 + \epsilon V_7$, $V_3 + V_4 + \delta V_6 + \epsilon V_7$,
$\epsilon = \pm 1, \delta \in R\backslash\{0\}$,
(F1) $\epsilon V_3 + \delta V_4$, $\epsilon = 1, \delta = 0$ or $\epsilon \in R, \delta = 1$,
(F2) $\epsilon(V_3 - V_4) + \delta V_7$, $\epsilon = 0, \delta = 1$ or $\epsilon = 1, \delta \in R\backslash\{0\}$,
(F3) $2V_3 + \epsilon V_7$, $2V_4 + \epsilon V_7$, $V_3 + V_4 + \epsilon V_7$, $\epsilon = \pm 1$.

By setting various possible values to the parameters ϵ, δ and σ, different elements of the optimal set are obtained. Note that the optimal set found in [10] for arbitrary h comprises only (A1), (B1), ..., (F1) cases. The optimal set of subalgebras can be used for explicit determination of group invariant solutions and their classification.

6. Discussions and conclusion. We presented the *MATHEMATICA* package *LieSymm-PDE* that we have developed for automatic determination of Lie point symmetries of PDEs. The functions of *LieSymm-PDE* provide a possibility to find the infinitesimal generator, either directly in one step, or by taking advantage of an elaborate interactive mode. The usefulness of *LieSymm-PDE* functions for making equivalent transformations and for giving hints during the solving process has been illustrated in Section 4. After applying *LieSymm-PDE* to the equations considered in [4], we come to the conclusion that in comparison with the *MATHEMATICA* program described in [4] the package *LieSymm-PDE* does not require a polynomial ansatz for the infinitesimals and needs less external advice (hints) to fulfill the task. We compared the functions of *LieSymm-PDE* with those available by the package "liesymm" of Maple. We note that the *LieSymm-PDE* function **CreateDSE** for creating of the DSEs can be used as an alternative of the Maple command "liesymm[determine]()". We found also that Maple does not provide tools for solving of the DSEs as the *LieSymm-PDE* special functions **SolveDSE**, **SplitDSE**, **DiffDSE** and **AddDSE** do.

The method of *LieSymm-PDE* for solving DSEs is based on several programming modules for dealing with some distinct types of equations. This method is generally allied with the approach applied in the Reduce package [14]. Finally, it should be noted that *LieSymm-PDE* is open to adding new solving modules and transformation rules so that its capabilities can be constantly enhanced. This leads to reducing the needs of user's hints and makes the program flexible and self-contained. As a result new larger and more complicated systems of PDEs become manageable.

The package has been successfully tested by a large number of PDEs. The obtained symmetries are in full agreement with those found in literature: in [8], for the heat equation, the Burgers' equation and the Korteweg-de Vries equation, in [2], for the nonlinear Schrödinger equation, in [6], for the nonlinear Schrödinger equation with a perturbation term, in [3], for the CNSEs (12) with additional linear terms. We used the package to perform a Lie group analysis of different models of nonlinear fiber optics: strong [10] and weak [13] birefringent fibers, two waves at different carrier wavelengths in two-mode fibers [10], fibers with birefringence and stimulated Raman scattering [11], light pulses propagation at zero-dispersion wavelength [12], and nonlinear directional couplers [13]. A careful comparison with the earlier results in [3, 6, 7, 9] proves the effectiveness of the package *LieSymm-PDE* in solving practical problems and justifies this presentation.

REFERENCES

[1] Agrawal G. Nonlinear fiber optics. Academic, Boston, 1995.

[2] Akhmediev N. N., A. Ankiewicz. Solitons, nonlinear pulses and beams. Chapman & Hall, London, 1997.

[3] Alfinito E., M. Leo, R. A. Leo, G. Soliani, L. Solombrino. Symmetry properties and exact patterns in birefringent optical fibers. *Phys. Rev. E* **52** (1995), 3159–3165.

[4] Baumann G. Lie symmetries of differential equations: A *MATHEMATICA* program to determine Lie symmetries. `www.library.wolfram.com/infocenter /MathSource/431`, 1993.

[5] Baumann G. *MathLie* a program of doing symmetry analysis. *Math. Comp. Simulation* **48** (1998), 205–223.

[6] Gagnon L., P. A. Bélanger. Soliton self-frequency shift versus Galilean-like symmetry. *Opt. Lett.* **15** (1990), 466–468.

[7] Manganaro N., D. F. Parker. Similarity reductions for variable-coefficient coupled nonlinear Schrödinger equations. *J. Phys. A: Math. Gen.* **26** (1993), 4093–4105.

[8] Olver P. Applications of Lie groups to differential equations. Springer, Berlin, 1986.

[9] Parker D. F. Coupled cubic Schrödinger equations for axially symmetric waveguides. Proc. of Fourth Meeting "Waves and stability in continuous media", 1988, 260–280.

[10] Pulov V. I., I. M. Uzunov, E. J. Chacarov. Solutions and laws of conservation for coupled nonlinear Schrödinger equations: Lie group analysis. *Phys. Rev E* **57** (1998), 3468–3477.

[11] Pulov V. I., I. M. Uzunov, E. J. Chacarov. Symmetry properties of nonlinear pulse propagation in birefringent optical fibers with stimulated Raman scattering. *Proc. of SPIE* **5830** (2005), 105–109.

[12] Pulov V. I., I. M. Uzunov, E. J. Chacarov. Finding Lie symmetries of partial differential equations with *MATHEMATICA*: Applications to nonlinear fiber optics. *Geometry, Integrability and Quantization*, 2008 (to appear).

[13] Pulov V. I., I. M. Uzunov, E. J. Chacarov, V.L. Lyutskanov. Lie group symmetry classification of solutions to coupled nonlinear Schrödinger equations. *Proc. of SPIE* **6604** (2007), 66041K.

[14] Schwarz F. Automatically determining symmetries of partial differential equations. *Computing* **34** (1985), 91–106.

Vladimir I. Pulov
Dept. of Physics
Technical University – Varna
1, Studentska Str.
9010 Varna, Bulgaria
e-mail: vpulov@hotmail.com

Edy J. Chacarov
Dept. of Mathematics and Computer Studies
Varna Free University, Chaika Resort
9007 Varna, Bulgaria
e-mail: eddy@abv.bg

Ivan M. Uzunov
Dept. of Applied Physics
Technical University – Sofia
8, St. K. Ohridski Str.
1000 Sofia, Bulgaria
e-mail: ivan_uzunov@hotmail.com

Received February 16, 2007
Final Accepted October 18, 2007