W.I. Fushchych, Scientific Works 2001, Vol. 3, 497–505.

Computer algebra application for determining Lie and Lie–Bäcklund symmetries of differential equations

W.I. FUSHCHYCH, V.V. KORNYAK

The application of computer algebra for determining Lie and Lie–Bäcklund (LB) symmetries of differential equations is considered. Algorithms for calculating the symmetries are developed and implemented on the basis of computer algebra systems REDUCE, AMP and FORMAC. The most effective and advanced program is written in FORMAC. It finds LB symmetries completely automatically. In many cases the program yields the full algebra of symmetries. If the program fails in full integration of the determining system, it reduces the remaining determining equations to the system in involution.

1. Introduction

The determination of point and contact Lie symmetries and Lie–Bäcklund symmetries of differential equations is one of the central problems in applied mathematics and mathematical physics. The mathematical theory is rather well developed [10, 8], but computing of the symmetries of certain systems of differential equations requires extremely tedious symbolic manipulations. In many cases the only possibility to cope with the task is the application of computer algebra. There are several programs and packages for solving the problems in this field. The REDUCE package for obtaining determining systems of point symmetries was suggested by Schwarz [11]. This package includes several programs for different kinds of differential equations and systems. This work was developed by adding programs for solving the determining systems [12], and the resulting package was successfully applied to many problems in mathematical physics. In [1] the universal REDUCE program was suggested for computing determining systems of point and contact symmetries of arbitrary systems of differential equations. To obtain determining equations of LB symmetries the FORMAC and REDUCE programs have been developed [2, 3]. We have also written an analogous AMP program. It is very difficult or impossible to handle the LB determining systems by hand, because generally they contain several hundreds of equations (though linear and overdetermined). We should mention, also, a recent FORMAC package CRACKSTAR [13], which is closely related with the subject considered. This package is intended for investigation of Lie-symmetries of pde's and dynamical symmetries of ode's as well as other analytic properties.

Comparing different computer algebra systems (REDUCE 2, REDUCE 3.∅, AMP 6.4 and PL/I-FORMAC) we came to a conclusion that FORMAC is the most suitable system for our purposes. Of course, it is out of date to some extent but much more effective than REDUCE and AMP. We developed the FORMAC program LBF for determining Lie–Bäcklund symmetries of arbitrary systems of differential equations. The program creates the LB determining system, integrates it as far as possible and,

J. Symbolic Computation, 1989, **7**, 611–619.

if some part of the determining system remains, it reduces this part to the system in involution, i.e. to system with all integrability conditions being explicit [4].

In this paper our consideration is given mostly to the general case of LB symmetries, because point and contact symmetries are only special cases of LB ones. Readers interested in the details of algorithms and programs concerning point and contact symmetries are referred to papers mentioned above.

2. Mathematical background

We shall consider a system of s partial differential equations of kth order for m functions u^α in the n independent variables x^i

$$\begin{aligned} &F^\nu(x^i, u^\alpha, u^\alpha_{i_1\cdots i_l}) = 0, \\ &\nu = 1,\ldots,s; \quad \alpha = 1,\ldots,m; \quad i, i_1,\ldots,i_l = 1,\ldots,n; \quad l = 1\ldots,k, \end{aligned} \tag{1}$$

where $u^\alpha_{i_1\cdots i_l}$ are jet bundle coordinates corresponding to the partial derivatives of u^α with respect to $x^{i_1},\ldots,x^{i_l}$.

The LB group is defined as the tangent transformation group of infinite order. In the terms of infinitesimal generators it means that coordinates of Lie-algebra depend on the unlimited number of derivatives. The Lie-algebra vector called the LB operator has the form

$$X = \xi^i \frac{\partial}{\partial x^i} + \eta^\alpha \frac{\partial}{\partial u^\alpha} + \sum_{l\geq 1} \zeta^\alpha_{i_1\cdots i_l} \frac{\partial}{\partial u^\alpha_{i_1\cdots i_l}}. \tag{2}$$

Summation over twice occurring indices is always understood, ξ^i, η^α, $\zeta^\alpha_{i_1\cdots i_l}$ depend on variables x^i, u^α, $u^\alpha_{i_1\cdots i_l}$, and $\zeta^\alpha_{i_1\cdots i_l}$ are generated recursively by

$$\zeta^\alpha_i = D_i(\eta^\alpha) - u^\alpha_j D_i(\xi^j), \quad \zeta^\alpha_{i_1\cdots i_l} = D_{i_l}(\zeta^\alpha_{i_1\cdots i_{l-1}}) - u^\alpha_{ji_1\cdots i_l} D_{i_l}(\xi^j). \tag{3}$$

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

$$D_i = \frac{\partial}{\partial x^i} + u^\alpha_i \frac{\partial}{\partial u^\alpha} + \sum_{l\geq 1} u^\alpha_{ii_1\cdots u_l} \frac{\partial}{\partial u^\alpha_{i_1\cdots i_l}}. \tag{4}$$

System (1) with all the differential consequences is called a differential manifold

$$[F]: \quad F^\nu = 0, \quad D_i(F^\nu) = 0, \quad \ldots, \quad D_{i_1} D_{i_2} \cdots D_{i_l}(F^\nu) = 0, \quad \ldots . \tag{5}$$

According to the definition systems, (1) is invariant with respect to LB group, if the differential manifold $[F]$ is invariant, i.e.

$$X[F]\big|_{[F]} = 0. \tag{6}$$

There is a theorem staling that condition (6) is equivalent to

$$XF|_{[F]} = 0, \tag{7}$$

i.e. it is sufficient to apply the X operator only to initial equations (1), but to consider the differential consequences when transferring to the manifold.

It is easy to check that LB operators of the form

$$X_* = \xi^i_* D_i, \tag{8}$$

where ξ^i_* are arbitrary functions of the x^i, u^α, $u^\alpha_{i_1\cdots i_l}$ variables, and to leave the artibrary differential manifold invariant, i.e. they do not contribute to the invariance condition. These operators form the ideal in the Lie-algebra of all the LB operators. Therefore, it is possible to consider, without loss of generality, the factor-algebra of the complete Lie-algebra with respect to the above ideal. Each operator (2) is equivalent in the factor-algebra to some operator with vanished ξ^i, viz.

$$X \sim Y = X - \xi^i D_i = (\eta^\alpha - \xi^i u^\alpha_i)\frac{\partial}{\partial u^\alpha} + \cdots.$$

Thus the elements of factor-algebra may be represented in the form

$$X = \eta^\alpha \frac{\partial}{\partial u^\alpha} + \sum_{l\geq 1} \zeta^\alpha_{i_1\cdots i_l} \frac{\partial}{\partial u^\alpha_{i_1\cdots i_l}}. \tag{9}$$

Operators (9) are called "canonical operators". Transition to canonical operators essentially simplifies the calculations, since now it is sufficient to consider m functions η^α instead of $n+m$ functions ξ^i and η^α. Moreover, extension formulae (3) take a simple form

$$\zeta^\alpha_i = D_i(\eta^\alpha), \quad \zeta^\alpha_{i_1\cdots i_l} = D_{i_l}(\zeta^\alpha_{i_1\cdots i_{l-1}}). \tag{10}$$

In terms of canonical operators the invariance conditions, i.e. the determining equations, take the form

$$\left(\eta^\alpha \frac{\partial F^\nu}{\partial u^\alpha} + D_i(\eta^\alpha)\frac{\partial F^\nu}{\partial u^\alpha_i} + D_{i_2}(D_{i_1}(\eta^\alpha))\frac{\partial F^\nu}{\partial u^\alpha_{i_1 i_2}} + \cdots\right)\Big|_{[F]} = 0. \tag{11}$$

This is a system of equations with respect to η^α. The solutions of the determining equations depending on the derivatives of no more than kth order are called kth order solutions. This definition allows the overdetermined system to be obtained, because we can split the left part of (11) with respect to "free derivatives", i.e. $u^\alpha_{i_1\cdots i_l}$ for $l > k$. If the 1st order derivatives are not expressed by derivatives of a higher order in the process of transition to manifold $[F]$, then the 1st order solutions contain point and contact Lie symmetries. In particular, point symmetries are corresponding to the 1st order solutions of the form

$$\eta^\alpha_{\text{point}} = \eta^\alpha(x^i, u^\beta) - \xi^j(x^i, u^\beta)u^\alpha_j, \tag{12}$$

where $\alpha, \beta = 1, \ldots, m$; $i, j = 1, \ldots, n$; ξ^j and η^α are usual coordinates of the Lie-algebra of point transformations. Note that the point transformation groups obtained from the solutions of LB determining equations may be wider than the classical ones. This may occur if some equations of the system under consideration are of an order less than the maximum one of the system, because, when transferring to the manifold, the LB symmetries imply the use of differential consequences, i.e. relations

$$F^\nu = 0, \quad D_i(F^\nu) = 0, \quad \ldots, \quad D_{i_1}\cdots D_{i_l}(F^\nu), \quad \ldots,$$

whereas the classical definition of point symmetries uses relations $F^\nu = 0$ only [8]. There are non-trivial LB symmetries of the 1st order, i.e. non-equivalent to point or contact ones. It occurs, for example, for Dirac equations [14].

3. Description of the program and algorithms

The LBF program is a single program acting completely automatically. Its text contains 37 internal procedures besides the main one. The total length of the program is now 1362 lines of PL/1-FORMAC code. We shall describe the program and its algorithms illustrating some steps by the simple example of determining 1st order LB symmetries for one-dimensional linear heat equation. To be more concrete, let us give the full input and output for this task with the comments on the right side.

Input:

'HEAT EQUATION'	— comment	
1	— symmetry order	
'X, T; U'	— independent and dependent variables	(13)
'$UT - UXX$'	— heat equation $U_T - U_{XX} = 0$.	

To simplify the input and output expressions, the derivatives are represented by concatenation of dependent and independent variables. Thus, if one uses multicharacter symbols for variables one has to choose them so as to avoid ambiguities. If one considers the system of equations, the corresponding character strings separated by blanks must be added.

Output:

HEAT EQUATION
INDEPENDENT AND DEPENDENT VARIABLES:
X, T; U
SYSTEM OF EQUATIONS:
1). $UT - UXX = 0$
GENERATORS OF LIE–BÄCKLUND SYMMETRIES OF THE 1ST ORDER:
1).
$U \neq= C1\,UX + C2\,(1/2\,UX + T\,UX) + C3\,(UX\,X + 2\,T\,UT) + C4$
$(T\,UX\,X + T^2\,UT + U\,(1/2\,T + 1/4\,X^2)) + C5\,UT + C6\,U + F1$
DEPENDENCES OF FUNCTIONS:
$F1 = F1(X, T)$
REMAINING EQUATIONS:
1).
$0 = F1.(T) - F1.(X, X)$ — equation $F^1_T - F^1_{XX} = 0$ (14)

The designations Ci and Fi mean constants C^i and functions F^i, $U\#$ means $\overset{*}{U} \equiv \eta^1$. Taking into account formula (12) we see that the 1st order LB symmetries appear to be point ones. Considering the coefficients at C^i and using (12) it is easy to obtain the symmetry operators

$$e_1 = \partial_X, \quad e_2 = \frac{1}{2}XU\partial_U - T\partial_X, \quad e_3 = X\partial_X + 2T\partial_T,$$
$$e_4 = U\left(\frac{1}{2}T + \frac{1}{4}X^2\right)\partial_U - XT\partial_X - T^2\partial_T, \quad e_5 = \partial_T, \quad e_6 = U\partial_U.$$

As is the case for every linear partial differential equation, there is also infinite-dimensional subalgebra $e_\infty = F^1(X,T)\partial_U$, where F^1 is an arbitrary solution of equation (14). This example takes 47 seconds of CPU time and 260 Kbytes memory

on the ES 1045 computer (under the OS/VS) with the internal performance of about 700 000 op/sec.

The LBF program consists of two main parts; the computation of determining system, and the solution of that system. The 1st part is the improved and generalised version of the program described in [2].

We use the sequencing of $u^{\alpha}_{i_1\cdots i_l}$ and derivatives of functions F^i, included in symmetry generators, in the following way: sets of lower indices are ordered lexicographically; after that the items are ordered in accordance with their upper indices. The position of the item in such a row we shall call "ordinal". This ordering permits to perform the considerable part of calculations using the fast and non-wasting memory of PL/I integer arithmetic only.

The program executes sequentially the following steps.

(1) Reading the input data and transforming them into internal form introducing the ordering mentioned above.

(2) Computation of the differential consequences up to the required order and elimination of dependences thereof. The program tries to solve the system together with differential consequences with respect to derivatives of the highest ordinals which are included linearly in relations. To do this, the program uses the Gauss excluding method. Note that differential consequences depend on the highest order derivatives only linearly. Some equations may not contain linearly included derivatives. The program marks such equations.

If the problem of classification is considered, then the system of equations contains arbitrary parameters or functions. Some combinations of these items may be used as denominators during the excluding process. Everywhere in the program in similar situations such different denominators are memorised to be printed at the end of program execution. It is necessary to consider separately the cases when such combinations are zeros.

(3) The determination of symmetry variables, i.e. variables which the generators depend on. (The generators do not depend on derivatives of an order higher than the symmetry order and on derivatives excluded in the previous step.)

(4) Computation of canonical LB operator (9) and the result of its action on system (1), i.e. computation of invariance conditions.

(5) Transition to manifold. The program eliminates the derivatives, excluded in step (2), in the invariance conditions. If there are equations unresolved in step (2) the program adds them to invariance conditions (11) having preliminarily multiplied them by indefinite factors.

(6) Separation of the determining system with respect to free variables, i.e. derivatives of an order higher than the symmetry order. The program separates the equations not only with respect to different powers of the free variables but also with respect to arbitrary different independent functions of such variables. It is possible that the functions independent in general may be dependent in some particular cases. For example, if u_{xx} is a free variable, then from $A\sin(u_{xx}) + B\cos(u_{xx}) = 0$ it follows that $A = 0$ and $B = 0$, but from $Au^2_{xx} + Bu^k_{xx} = 0$ it follows that $A = 0$ and $B = 0$ if $k \neq 0$, but $A + B = 0$ otherwise. Another obvious example $Af(u_{xx}) + Bf'(u_{xx}) = 0$ requires to consider the particular case

$f = e^{u_{xx}}$. For subsequent consideration of similar particular cases the program memorises all different separation factors containing arbitrary functions and parameters. For the sake of economy of memory, during the separation process, zeros are deleted immediately, i.e. when a one-term determining equation arises, it and its differential consequences are substituted at once into all remaining expressions.

(7) Exclusion of indefinite factors. Using the Gauss method the program excludes indefinite factors if they were introduced at step (5).

At the end of the first part of the program the state of the determining system (i.e. expressions for generators, dependences of functions, equations) considered is for the example:

$$\overset{*}{u} = F^1; \quad F^1 = F^1(x,t,u,u_x,u_t);$$
$$F^1_t - F^1_{xx} - 2u_t u_x F^1_{uu_x} - 2u_t F^1_{xu_x} - 2u_x F^1_{xu} - u_t^2 F^1_{u_x u_x} - u_x^2 F^1_{uu} = 0,$$
$$F^1_{xu_t} + u_t F^1_{u_x u_t} + u_x F^1_{uu_t} = 0, \quad F^1_{u_t u_t} = 0.$$

(8) Simplification of the determining system. The fragment of the program responsible for this step is

```
BC = '1'B;
DO WHILE(BC);
  BB = '1'B;
DO WHILE(BB);
    BA = '1'B;
    DO WHILE (BA);
      CALL REDSYS(1); CALL ORTSYS(∅); BA = SMONINT;
    END;
    BB = INVOL;
  END;
  BC = RESTINT;
END;
```

Here, BA, BB, BC are the control variables. The very inner loop contains the calls of the most efficient procedures. The REDSYS procedure is auxiliary. It reduces the determining system to some canonical form. Argument "1" means that the reduction begins from the first equation. (There are calls of REDSYS from other procedures with different values of argument.) The call of the ORTSYS(K) procedure reduces the determining system to the orthonomic form [4], i.e. the derivatives of the highest ordinals (leading derivatives) are singled out and substituted (with theirs differential consequences of the order up to K), equations are ordered in accordance with the increase of ordinals of leading derivatives. ORTSYS deletes the equations not containing derivatives after having performed the corresponding substitutions. The SMONINT procedure integrates the systems of monomial equations, i.e. systems of the form $\{F^j_{i_1\cdots i_k} = 0\}$. Here,

the lower index i_l means the differentiation with respect to the i_lth symmetry variable. The procedure yields the expression for F^j simultaneously considering all monomials for the given j. It allows the number of iterations to be reduced. SMONINT substitutes also F^j and its derivatives in other expressions and separates the determining system after that. If there are no monomial equations in the determining system, then BA takes the value '∅'B and the loop is completed.

After this loop the state of the determining system for the heat equation is

$$\overset{*}{u} = F^3 + F^4 u + F^2 u_x + F^1 u_t;$$
$$F^1 = F^1(t), \quad F^2 = F^2(x,t), \quad F^3 = F^3(x,t), \quad F^4 = F^4(x,t);$$
$$2F^2_x - F^1_t = 0, \quad 2F^4_x - F^2_t + F^2_{xx} = 0, \quad F^3_t - F^3_{xx} = 0, \quad F^4_t - F^4_{xx} = 0.$$

The INVOL procedure performs the further integrations or reduces the remaining part of the determining system to the system in involution by Riquier–Janet method [4]. The method consists in calculating differential consequences of equations and in excluding dependences out of them. INVOL tries to separate and integrate the relations arising during this process. If it fails in integration, then the system is reduced to the one in involution, and BB takes the value '∅'B; if otherwise, BB = '1'B, and the whole process is repeated. The INVOL procedure integrates the monomials and often arising equations of the form $F^j_{i_1 \cdots i_k} = P$, where P is the polynomial of the variables of differentiations.

As a rule, the combination of the above-mentioned operations is sufficient to reveal the major part of dependences of the symmetry generators, because these dependences are often polynomial. For instance, our example is completed by the INVOL procedure. In some cases it is possible to go further in integration of the remaining part of the determining system. This is effected by the RESTINT procedure. In gaining experience, it is possible to add in this procedure some particular and rare methods of integration. Now RESTINT contains the procedure for solving the ordinary differential equations or systems with constant (with respect to differentiation variable) coefficients up to the 4th order, and the procedure for solving differential equations of the 1st order with variable coefficients. The last procedure yields in non-polynomial cases the formal expressions for indefinite integrals. BC takes the value '1'B if further simplification after RESTINT is possible, and '∅'B, if otherwise.

(9) The last step of the LBF program is output. The internal designations are replaced by more expressive ones. The expressions for generators are reduced to some form simplifying the extraction of different one-dimensional subalgebras, as in the example above. The program removes also superfluous functions or constants if they arise during integration. For example, if $F^1(X,Y)$ and $F^2(X)$ are arbitrary functions, then $F^1(X,Y) + F^2(X)$ is equivalent to $F^1(X,Y)$. In general, the output may contain two more items, in addition to those presented in the example: the list of different denominators containing arbitrary functions or parameters, and the analogous list of separation factors.

Let us demonstrate the result of the program application in obtaining the 1st order symmetries of a more complicated non-linear equation [5]

$$\frac{\partial u}{\partial \tau} - \Box u - \lambda u \frac{\partial u}{\partial x^\mu}\frac{\partial u}{\partial x_\mu} = 0, \tag{15}$$

where

$$\Box = \frac{\partial^2}{\partial \varphi_t^2} - \frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} - \frac{\partial^2}{\partial z^2}, \quad \frac{\partial u}{\partial x^\mu}\frac{\partial u}{\partial x_\mu} = u_{\varphi_t}^2 - u_x^2 - u_y^2 - u_z^2.$$

The LBF program gives the general solution having in terms of operators the form:

$$e_1 = \partial_x, \quad e_2 = \partial_y, \quad e_3 = \partial_z, \quad e_4 = \partial_{\varphi_t}, \quad e_5 = \partial_x + x\partial_{\varphi_t},$$
$$e_6 = \varphi_t\partial_y + y\partial_{\varphi_t}, \quad e_7 = \varphi_t\partial_z + z\partial_{\varphi_t}, \quad e_8 = y\partial_x - x\partial_y, \quad e_9 = x\partial_z - z\partial_x,$$
$$e_{10} = z\partial_y - y\partial_z, \quad e_{11} = \varphi(u)\partial_u, \quad e_{12} = \partial_\tau,$$
$$e_{13} = x\partial_x + y\partial_y + z\partial_z + t\partial_t + 2\tau\partial_\tau,$$
$$e_{14} = x\tau\partial_x + y\tau\partial_y + z\tau\partial_z + t\tau\partial_{\varphi_t} + \tau^2\partial_\tau + \left(\frac{x^2+y^2+z^2-t^2}{4} - 2\tau\right)\varphi(u)\partial_u,$$
$$e_{15} = \tau\partial_x + \frac{x}{2}\varphi(u)\partial_u, \quad e_{16} = \tau\partial_y + \frac{y}{2}\varphi(u)\partial_u, \quad e_{17} = \tau\partial_z + \frac{z}{2}\varphi(u)\partial_u,$$
$$e_{18} = \tau\partial_{\varphi_t} - \frac{t}{2}\varphi(u)\partial_u, \quad e_\infty = \psi(x,y,z,t,\tau)\exp\left(-\frac{\lambda u^2}{2}\right)\partial_u,$$

where

$$\varphi(u) = \exp\left(-\frac{\lambda u^2}{2}\right)\int \exp\left(\frac{\lambda u^2}{2}\right) du,$$

$\psi(x,y,z,t,\tau)$ is an arbitrary solution of equation

$$\frac{\partial \psi}{\partial \tau} - \Box u = 0. \tag{16}$$

This example takes 5 min 18 s and 320 Kbytes on ES 1045. Generators e_1–e_{10} create the Poincaré algebra. As it follows from the above operators, equation (15) turned out to be automorphic, i.e. all its solutions lie on one group orbit. It allows to reduce non-linear equation (15) to linear one (16) using standard techniques of symmetry analysis.

4. Conclusion

There are some problems in connection with the considered one where the computer algebra may be successfully applied. For example, to complete the symmetry analysis of the system of differential equations it is important to learn the subgroup structure of the symmetry group, i.e. to classify the subalgebras of Lie-algebra of symmetries into conjugacy classes. This problem is also important in many other fields. In [9] the algorithm for classification of subalgebras of finite-dimensional Lie-algebras and its computer implementation were described.

Modern development of symmetry analysis includes several approaches considering non-local symmetries, i.e. symmetries depending on integrals or even more general operators acting in the functional space of the dependent functions [7]. Some class of such symmetries and corresponding algorithms and programs were considered in [6].

We are grateful to Professors B. Buchberger, B. F. Caviness and J. A. van Hulzen for their interest in this work.

1. Eliseev V.P., Fedorova R.N., Kornyak V.V., A REDUCE program for determining point and contact Lie symmetries of differential equations, *Comput. Phys. Commun.*, 1985, **36**, 383.
2. Fedorova R.N., Kornyak V.V., Determination of Lie–Bäcklund symmetries of differential equations using FORMAC, *Comput. Phys. Commun.*, 1986, **39**, 93.
3. Fedorova R.N., Kornyak V.V., A REDUCE program for computing determining equations of Lie–Bäcklund symmetries of differential equations, Dubna, JINR, 1987, R11-87-19 (in Russian).
4. Finikov S.P., Carton's exterior forms method, Moscow–Leningrad, Gostechizdat, 1948, 432 p. (in Russian).
5. Fushchych W.I., Symmetry in the problems of mathematical physics, in Algebraic-Theoretical Studies in Mathematical Physics, Kyiv, Inst. of Math., 1981, 6–28 (in Russian).
6. Fushchych W.I., Kornyak V.V., Computation on a computer of non-local symmetries of linear systems in mathematical physics, In International Conference on Computer Algebra and its Applications in Theoretical Physics, Dubna, JINR, 1987, D11-85-791, 345–350 (in Russian).
7. Fushchych W.I., Nikitin A.G., Symmetries of Maxwell's equations, Dordrecht, D. Reidel, 1987, 217 p.
8. Ibragimov N.H., Transformation groups in mathematical physics, Moscow, Nauka, 1983, 286 p. (in Russian).
9. Kornyak V.V., Classification of subalgebras for finite-dimensional Lie-algebra using computer, In international Conference on Computer Algebra and its Applications in Theoretical Physics, Dubna, JINR, D11-85-791, 339–344 (in Russian).
10. Ovsiannikov L.V., Group analysis of differential equations, Moscow, Nauka, 400 p. (in Russian).
11. Schwarz F.A., A REDUCE package for determining Lie symmetries of ordinary and partial differential equations, *Comput. Phys. Commun.*, 1982, **27**, 179.
12. Schwarz F.A., Automatically determining symmetries of partial differential equations, *Computing*, 1985, **34**, 91.
13. Wolf T., Analytic solutions of differential equations with computer algebra systems, Preprint N 87/5, Friedrich Schiller Universitat, Jena, 1987.
14. Zhdanov R.Z., On application of Lie–Bäcklund method to study of symmetries of the Dirac equations, in Group-Theoretical Studies of Mathematical Physics Equations, Kyiv, Inst. of Math., 1985, 70–73 (in Russian).