Computer Algebra and Differential Equations — An Overview

Werner M. Seiler
Institut für Algorithmen und Kognitive Systeme
Universität Karlsruhe, 76128 Karlsruhe, Germany
werner.seiler@informatik.uni-karlsruhe.de
http://iaks-www.ira.uka.de/iaks-calmet/werner/werner.html

We present an informal overview of a number of approaches to differential equations popular in computer algebra. This includes symmetry and completion theory, local analysis, differential ideal and Galois theory, dynamical systems and numerical analysis. A large bibliography is provided.

Introduction

Differential equations represent one of the largest fields within mathematics. Besides being an interesting subject of their own right one can hardly overestimate their importance for applications. They appear in natural and engineering sciences and increasingly often in economics and social sciences. Whenever a continuous process is modeled mathematically, chances are high that differential equations are used.

Thus it is not surprising that differential equations also play an important role in computer algebra and most general purpose computer algebra systems provide some kind of `solve` command. Many casual users believe that designing and improving such procedures is a central problem in computer algebra. But the real situation is somewhat different. Many computer algebra applications to differential equations work indirectly; they help to study and understand properties of the solution space.

The purpose of this article is to sketch in an informal way some of the main research directions in this field and to provide a starting point for more detailed studies by giving a large number of references. We omit all mathematical details (there is not a single formula in this article!) but describe briefly the central ideas. For the same reason we tried to cite introductory articles or books and not the historically first or the most "ground breaking" work.

The bibliography is of course far from being exhaustive. As a further source of references one should mention the survey [104] by Singer. It gives much more details, especially on the more algebraic approaches, and contains a large bibliography. The same holds for the more focused surveys by Hereman [50, 51] covering symmetry theory and related fields and the one by MacCallum [65] on the integration of ordinary differential equations. In addition there have been three conferences devoted exclusively to differential equations and computer algebra. Their proceedings [103, 112, 113] contain a number of useful introductory or review articles on more specialized topics.

We are taking a rather broad view and consider more or less any constructive approach to differential equations as "computer algebra". This also implies that we do not pay special attention to implementations. Among the many different approaches to differential equations which fall under this broad definition of computer algebra one can distinguish certain directions which have found most attention (at least measured in the number of articles devoted to them). We concentrate in this article on the following eight topics: (i) *solving* differential equations, (ii) *local analysis*, (iii) *symmetry analysis*, (iv) *completion*, (v) *differential ideal theory*, (vi) *differential Galois theory*, (vii) *dynamical systems theory*, and (viii) the relation between *numerical analysis* and computer algebra.

A comparison of the impact made by symmetry analysis and by differential Galois theory, respectively, demonstrates the importance of computer algebra tools. The latter one is a hardly known theory studied by a few pure mathematicians. The former one remained in the same state for many decades following Lie's original work. One reason was definitely the tedious determination of the symmetry algebra. As soon as computer algebra systems emerged, the first packages to set up at least the determining equations were written. Since then Lie methods belong to the standard tools for treating differential equations.

Solving Differential Equations

Most computer algebra systems can solve some differential equations.[1] They mainly apply some standard techniques like those in Zwillinger's handbook [127] or try some "pattern matching" in a list of solved equations like Kamke [58]. Heuristics often extend the applicability of these techniques, for example by finding a transformation such that a given equation can be handled by the implemented methods.

Although this approach solves more differential equations than one might expect (see e. g. the recent review by Postel and Zimmermann [86][2]), it has some drawbacks. A major one is that no information is obtained, if the computer algebra system does *not* return a solution. It could be that the given differential equation has indeed no solution (or at least none in closed form) or that simply the heuristics were not able to determine a suitable transformation.

For that reason researchers in computer algebra are more interested in *decision algorithms*. These either yield a solution in a specific class of functions or decide that no such solution exists. However, so far only for linear ordinary differential equations such algorithms are known. There it is possible to decide with the help of differential Galois theory whether or not Liouvillian solutions exist.

There exists a number of reasons for this perhaps disappointing situation. Computability theory yields principal limits to what can be solved. For example if one restricts to *computable functions* some classical existence theorems for differential equations fail [1, 87]. More precisely, one has constructed examples of differential equations where one can show that solutions exists but that it is not possible to compute them. Some further (positive and negative) results in this direction can be found in [23].

Ideally a solution algorithm should return the *general solution*. But for nonlinear equations it is surprisingly difficult even just to define this term. A rigorous resolution of this problem based on differential ideal theory was only recently presented [53]. Intuitively one would expect that the general solution depends on some arbitrary parameters (constants or functions) and that every solution of the differential equation can be obtained by a suitable specialization of these. This works fine for linear equations where the solution space has the structure of a vector space. But many nonlinear equations possess in addition *singular integrals* not contained in the general solution.

Similarly, defining the term "closed form solution" is notoriously difficult. Is a solution in terms of, say, Bessel functions in closed form or not? Up to now no generally accepted definition has emerged. The basic idea behind "closed form" is that of *finite constructibility* out of a set of "elementary functions". Note that this is an algebraic and not an analytic property! On the practical side one must see that even if a solution in closed form can be computed it may take very long and the result may be completely useless, as it is too large. Especially, if the main goal is to obtain an impression of the behavior of the solution, it is usually much more efficient to resort to numerical methods.

In any case one can state that a notable solution theory exists only for ordinary differential equations (see e. g. the survey [65]), mainly based on differential Galois theory. But the algorithms to compute the general solution suffer from a very high complexity and are in practice often rather useless, especially for higher order equations. One way out is to incorporate heuristics as mentioned above.

Another possibility that also addresses the problem of useless output is to aim for "simple" solutions [7]. Popular

[1] *MuPAD* provides the command `odesolve` for this task.

[2] An updated version can be found at `http://www.mupad.de/BIB/poszim96.html`.

variants are polynomial, rational [3] or exponential [11, 81] solutions. Because of their simple structure it is often possible to determine such solutions, if they exist, rather fast. But one should note that the classical methods for their computation are not always useful for computer algebra. It is still an active field of research to design effective algorithms being able to handle larger examples.

For partial differential equations the situation is much worse. In the last century mathematicians designed some solution methods (see e. g. the survey [118]). However, most of them are meanwhile almost forgotten; at least they are no longer found in textbooks on differential equations. It could be quite interesting to revive some of them for use in computer algebra systems. There exist a few implementations of standard techniques like characteristics, separation of variables or integral transforms, but they can usually handle only rather simple equations. Often they just reduce to a system of (nonlinear) ordinary differential equations and the question is whether this can be solved.

One can argue whether it really makes sense to speak of the general solution of a system of partial differential equations. For example one definition of a harmonic function is that it solves the Laplace equation (or more generally all holomorphic functions are solutions of the Cauchy-Riemann equation). Thus one might prefer to say that the Laplace equation defines a class of functions. In some simple cases like the wave equation one can give an explicit parameterization of this class in terms of some arbitrary functions which one may call the general solution. But usually no such parameterization exists. In order to get a really well-defined problem one must prescribe some initial or boundary conditions. In most applications such conditions arise automatically anyway.

Local Analysis

If it is not possible to construct a closed-form solution, one may go for an approximate solution describing the behavior of the solution in the neighborhood of a given point.[3] At *ordinary points* a Taylor series suffices; at *singular points* more general expansions must be used. In the case of linear differential equations singular points are only possible at singularities of the coefficients. Therefore one speaks of *fixed* singularities. Using the *Newton polygon* of the associated differential operator they can be further classified into *regular* and *irregular* ones [25].

Local analysis is a complex theory, even if one studies only real equations. Especially, if one wants to determine the radius of convergence of a formal solution, one must consider complex singularities. In the neighborhood of regular singular point one can represent the solution in form of a *Frobenius series*, a polynomial in $\log x$ with Taylor series coefficients multiplied by a factor $(x - x_0)^\alpha$ where α is a complex number. At irregular singular points the solution has usually an essential singularity and varies so rapidly that it makes no sense to construct an approximation; instead one tries to capture the asymptotic behavior which requires the addition of an exponential part.

There exist various algorithms for the construction of approximate or asymptotic solutions, partly dating back to Frobenius. Some are discussed together with implementations in [82, 115]. A main problem in the concrete application is that one cannot use an approximation of the location of the singularities. Thus one must not only solve polynomial equations but in general work with algebraic numbers which is quite expensive in any computer algebra system. However, with a careful analysis of the algorithms one can significantly reduce the necessary amount of computations with algebraic numbers.

Recent work concerns an extension to first order systems [6, 81]. In principle, one can transform any system into a single equation of higher order, e. g. using *cyclic vectors*. But this approach is rather inefficient, especially in higher dimensions. Hence one is interested in dealing directly with systems. *Moser's algorithm* performs here the classification into regular and irregular singularities; a rational version of it avoiding the use of algebraic extensions was presented by Barkatou [5].

For nonlinear differential equations the situation becomes much more complicated as *spontaneous* or *movable* singularities may occur, i. e. their location depends on initial or boundary data. One important direction is the *Painlevé*

[3] Especially in the linear case local solutions can also be very useful for the construction of closed-form solutions.

theory [54]. It was introduced by Painlevé while searching for new special functions (there still exists a strong connection between the local analysis of ordinary differential equations and special function theory). If all singularities are poles, no branch points appear in the (general) solution and it is single valued. A differential equation without movable branch points is said to possess the *Painlevé property* or to be integrable in the sense of Painlevé.

In general, it is not possible to check algorithmically whether or not a given differential equation has the Painlevé property. But there exist methods to check at least some necessary conditions; such methods are usually called *Painlevé test* [21]. One usually tries to construct a Laurent series around the singularity. Essentially, the test is passed, if this expansion has sufficiently many *resonances* or *Fuchsian indices* (free coefficients) to represent the general solution and if these occur at non-negative powers. In the case of negative resonances a perturbation approach [22] yields further information. Some references concerning implementations can be found in [95].

Weiss *et al*. [122] generalized the Painlevé theory to partial differential equations where a whole singularity manifold must be considered. This extension is much used in the theory of integrable systems, as the Painlevé test represents an important indicator for integrability and can be performed comparatively easily. The *Painlevé conjecture* states that every ordinary differential equation obtained by symmetry reduction of an integrable system is of Painlevé type; only weakened versions of it have been proven [2, 71]. Truncated series expansions are useful for constructing Bäcklund transformations, Lax pairs and much more [121]. There also exist relations to non-classical symmetry reductions [30].

Symmetry Analysis

Symmetry analysis [9, 77, 111] has made the strongest impact on computer algebra applications to differential equations. The most general definition of a symmetry is that of a transformation that maps solutions into solutions. Depending on the kind of transformations considered one obtains different kinds of symmetries. One possible application of symmetries is the construction of (special) solutions. Other goals are classifications, a proof of complete integrability, separation ansätze, conservation laws and much more.

Symmetry analysis goes back to the seminal work of Lie. He developed the concept of Lie groups in his quest for a Galois theory for differential equations. As we will see later, not much has remained of this original motivation. Symmetry and Galois theory have developed in very different directions. Even the relation between the Lie symmetry and the Galois group of a differential equation is rather unclear.

The most popular form of symmetry analysis deals with *point symmetries*. They are generated by vector fields acting on the space of independent and dependent variables. These vector fields span the Lie algebra of the Lie group of symmetries. The decisive observation of Lie was that it often suffices to work with the vector fields (or infinitesimal symmetries) instead of the symmetries themselves. This leads effectively to a linearization of the problem.

The symmetry generators arise as the solutions of a linear system of partial differential equations, the *determining system*. For ordinary differential equations it is unfortunately sometimes as difficult to solve this system as to solve the original one. This holds especially for first order equations where the original equation is just the characteristic equation of the determining equation. For partial differential equations the determining system is typically very over-determined and contains often some trivial equations allowing in many cases a rather straightforward solution.

For ordinary differential equations the existence of a sufficiently large, *solvable* symmetry algebra implies that its general solution can be constructed by quadratures only, as each symmetry allows us to reduce the order of the equation by one. In the case of partial differential equations symmetry reductions yield only special solutions, namely those being invariant under the symmetry group. Here each symmetry allows us to reduce the number of independent variables by one.

However, at intermediate steps of the reduction again linear partial differential equations must be solved. For in order to obtain the reduction, one must either perform a coordinate transformation such that the symmetry generator is rectified (so-called *canonical coordinates*) or the *differential invariants* of the symmetry must be determined. These are functions annihilated by the generator.

Thus the usefulness of Lie symmetries depends crucially on the ability to solve effectively all the arising linear partial differential equations. At first sight it might look, as if, especially for ordinary differential equations, we made the problem only worse. But in many cases of practical interest it turns out that is much simpler to solve these linear partial differential equations than the original equation.

There exist so many implementations of symmetry methods that it is rather difficult to keep an overview; we refer again to the surveys by Hereman [50, 51]. In almost any computer algebra system one can find a package for setting up the determining system. A few of the packages try furthermore some heuristics to solve it automatically. Again it is rather surprising how often this suffices to obtain the complete symmetry algebra. The symmetry package of MAPLE is somewhat unusual, as it uses the exterior systems approach of Harrison and Estabrook [47].

Although Lie point symmetries proved to be very useful in many applications, many differential equations of practical interest have no such symmetries. There are two basic approaches to generalize the theory. One can consider more general transformations; this leads to *generalized* or *Lie-Bäcklund symmetries* [4]. Alternatively, one weakens the requirement that every solution is mapped into a solution; this yields the so-called *non-classical methods*. In both cases the explicit construction of the symmetries becomes considerably more difficult.

Generalized symmetries are especially of interest for integrable systems [32, 123]. The existence of a *recursion operator* or a *master symmetry* generating an infinite hierarchy of symmetries is a strong indication that the considered system is integrable. Reduction with respect to generalized symmetries is an important tool for the construction of *soliton solutions*. It is also possible to classify nonlinear partial differential equations using these symmetries [73]. Some *MuPAD* packages for symmetries of integrable systems are described in [33].

Non-classical reductions can be understood within the general scheme of augmenting a given differential equation with *differential constraints* [78]. This corresponds to requiring that only some solutions are mapped into solutions, therefore one hopes to find more symmetries (these are sometimes called *weak symmetries*). In this approach the emphasis lies less on group theory but on the theory of over-determined systems of partial differential equations and thus on questions of completion (cf. [100]).

The first non-classical method was developed by Bluman and Cole [8] and uses the invariant surface condition as constraint. Although this leads for many differential equations to new reductions, the drawback is that the determining system becomes nonlinear. The *direct method* of Clarkson and Kruskal [20] tries to reduce a given partial differential equation to a system of ordinary differential equations by constructing a good ansatz; it corresponds to a special case of the method of Bluman and Cole.

The main problem in the method of differential constraints is to find compatible constraints leading to non-trivial reductions. Besides using the invariant surface condition no systematic way has been discovered so far and thus it remains essentially a game of “try and error”. For this reason differential constraints have not yet found much attention in applications.

Completion

Most textbooks on differential equations treat only *normal* systems (or systems in *Cauchy-Kowalevsky form*). For ordinary differential equations this implies that one assumes that the equations can be solved for the highest order derivatives. For partial differential equations one must furthermore assume the existence of a distinguished independent variable such that one can solve for its derivatives to obtain the Cauchy-Kowalevsky form. However, in many fields one encounters systems of differential equations which are not normal. A simple example are the determining systems appearing in symmetry analysis which are usually over-determined. Non-normal systems also occur naturally in differential geometry and in theoretical physics (gauge theories).

For a non-normal system it is a priori not clear whether it possesses any solutions. It may happen that the system is inconsistent. This can only be decided after the construction of all *integrability conditions*. These are differential equations satisfied by any solution of the system but which are nevertheless *algebraically* independent of it. While it makes no problem to construct one integrability condition (typically this requires only a cross-derivative), it is not so easy to decide when *all* have been found, as in principle an infinite number of conditions must be checked.

The process of finding all integrability conditions is called *completion* of the differential equation. It results in a *formally integrable system*, as after completion it is straightforward to construct order by order a formal power series solution. Under additional assumptions it is sometimes possible to show the convergence of the series. This leads for analytic equations to existence and uniqueness theorems like the Cartan-Kähler theorem (the well-known Cauchy-Kowalevsky theorem is a special case of it). For non-analytic equations solvability is a much more complicated question due to Lewy type effects [63].

The first systematic approach to the problem of completion was probably provided by the Janet-Riquier theory [55] with the introduction of *passive* systems. Their definition is based on a *ranking* of the derivatives which decides in what order the integrability conditions are constructed. The completion can be done completely automatically only for quasi-linear systems (if all arising integrability conditions are also quasi-linear), as it must be possible to solve for the leading derivative. In this case the resulting passive system is sometimes call a *standard form* [91].

In geometric theories the notion of a passive system is replaced by *involution*. It combines a geometric definition of formal integrability with an algebraic criterion for the termination of the completion. As an intrinsic concept involution requires no coordinate dependent ingredients like a ranking. Involution analysis based on the Cartan-Kähler theory [14] for exterior systems is discussed from an algorithmic point of view in [48, 49]. A completion algorithm for the jet bundle formalism based on the formal theory of Pommaret [84] was presented in [97].

Completion algorithms are very useful in the symmetry analysis of differential equations. Once a system is either passive or involutive, one can make statements about the size of the solution space [91, 98]. Thus it is possible to compute the size of the symmetry group without explicitly solving the determining system or to determine the loss of generality in a symmetry reduction [99]. One can even compute the abstract structure of the symmetry algebra without solving the determining system [64, 92].

These concepts are closely related to Gröbner bases in commutative algebra. This holds especially for the Janet-Riquier theory where rankings play a similar role as in the definition of a Gröbner basis. Therefore one sometimes find the term *differential Gröbner basis* for a passive system. Integrability conditions arising from cross-derivatives may be considered as “differential S-polynomials”. But these analogies acquire a precise meaning only in the context of differential algebra.

There is a one-to-one correspondence between linear systems of partial differential equations in one dependent variable and polynomial ideals. This has lead in commutative algebra to the new concept of an *involutive basis* of an ideal [42]. It is computed using algorithms coming from the completion theory of differential equations, but it is an ordinary (though not reduced) Gröbner basis. In some cases the new algorithms are considerably faster than the classical Buchberger algorithm.

Differential Ideal Theory

Like the differential Galois theory discussed in the next section, differential ideal theory belongs to the field of *differential algebra*. It can be informally described as an attempt “to write differential in front of everything in algebra”. Thus it deals with differential rings, differential fields etc. Of course, this requires an algebraic definition of differentiation. In differential algebra any mapping that is linear with respect to addition and satisfies the Leibniz or product rule is called a *derivation*. A differential ring is a commutative ring together with one (or more) derivation.

Differential polynomials arise by adjunction of differential indeterminates to a differential ring. But the ring of differential polynomials is not Noetherian. Adjoining a differential indeterminate corresponds to adjoining infinitely many algebraic indeterminates, as one must introduce all its derivatives as additional, algebraically independent variables. Thus Hilbert's Basis Theorem does not apply.

A *differential ideal* is an ideal which is in addition closed under the derivation of the differential ring. Many of the basic ideas in differential ideal theory can be traced back to Ritt [93]; the most advanced book is still the one by Kolchin [60]. Like in the purely algebraic theory one would like to introduce something like a Gröbner basis. As the ring of differential polynomials is not Noetherian, algorithms along the lines of the Buchberger algorithm do not terminate in general [18]. This is related to the fact that the *ideal membership problem* is undecidable for arbitrary differential ideals [35]. However, this result is more of theoretical interest, as for finitely generated ideals the decidability is still an open question. In any case one must say that so far no generally accepted definition of a differential Gröbner basis has emerged.

There exist two basic strategies to circumvent this principal problem. One can either restrict to special classes of ideals where a proof of termination is possible or one weakens the properties expected of a differential Gröbner basis. The completion algorithm of the Janet-Riquier theory can be considered as a simple example for the first strategy. An example for the second one are the bases introduced by Mansfield [70]. They are computed with pseudo-reductions and have thus weaker properties than their algebraic counterpart. Especially, it may happen that one leaves the ideal.

Recently, Boulier *et al.* [10] presented a *Rosenfeld-Gröbner algorithm* which computes a representation for the radical ideal of a finitely generated differential ideal in the following form. The radical is written as a finite intersection of saturations ideals; these are radical differential ideals defined by a system of differential polynomial equations and inequalities. This representation allows for an easy algorithmic test of radical ideal membership and for computing formal power series solutions.

Open problems are to obtain a minimal decomposition, i. e. to use only a minimal number of saturation ideals, and to find bases for these ideals (avoiding the inequalities). These questions are closely related to the inclusion problem for differential ideals which in turn can be seen as the problem of determining the relation between the singular and the general solutions of a differential equation. The principal obstacle in the construction of the bases is a very typical one in differential algebra. A theorem of Ritt asserts that by taking sufficiently many derivatives of the equations one can always get a basis but no bound for the number of derivatives needed is known.

Differential algebra is applied in *automatic theorem proving* in differential geometry [126]. This is similar to the use of algebraic ideal theory in theorem proving in elementary geometry. For this kind of applications *characteristic sets* seem to be more useful than Gröbner bases. A nice example for the possibilities here is the automatic derivation of Newton's law of gravity from the three Kepler laws [125].

Besides ideals of differential polynomials there has also been some work on ideals of linear differential operators or ideals of the Weil algebra [34]. However, here one is dealing with non-commutative rings. One could also consider the Cartan-Kähler theory as a kind of differential ideal theory, as it represents differential equations by closed ideals of differential forms.

Differential Galois Theory

Already Lie was looking for a differential analog of the (algebraic) Galois theory, when he introduced Lie groups. What is nowadays usually called differential Galois theory [66, 105] has however no connection to Lie symmetry theory. The latter one uses continuous transformation groups and can be applied to any differential equations. But as discussed above it is not completely algorithmic. The former one is based on linear algebraic groups. It considers exclusively linear ordinary differential equations and culminates in various algorithms for explicitly computing Liouvillian solutions.

Determining the solutions of linear differential equations is a very classical topic and many famous mathematicians like Liouville, Fuchs, Klein or Jordan studied it in the last century and their results are still very important for the design of algorithms. Differential Galois theory was essentially founded by Picard and Vessiot and given its modern form by Kolchin [60]. Pommaret [85] developed an alternative theory following more closely Lie's ideas and using the formal theory.

Liouvillian functions comprise essentially all expressions one can "easily write down". Allowed operations are the usual arithmetic ones, roots, exponentials, logarithms, integrals and algebraic functions. A more formal definition uses a tower of simple extensions of the field of rational functions. An important point is that for any Liouvillian function one needs only a finite number of extensions, thus it is algorithmically constructible. Most expressions one would call "closed-form" are in fact Liouvillian.

Most solution algorithms are based on the seminal work of Singer [102]. He showed that the logarithmic derivative of any Liouvillian solution is algebraic and determined an a priori bound for the degree of the minimal polynomial, namely the Jordan bound for the index of an Abelian normal subgroup of a finite linear group. In principle, this suffices to determine all Liouvillian solutions, but the bound grows rapidly with the order of the equations leading thus to a very high complexity of the algorithm.

Using the representation theory of finite groups Ulmer [114] could significantly improve the bound given by Singer, so that at least the treatment of equations up to third order seems feasible, but there does not yet exist an implementation. Group theory yields also a number of other interesting results like criteria for the existence (and number) of algebraic solutions (the solutions which are most expensive to determine belong to this class) and gives the basic case distinctions in the solution algorithms.

The original work of Singer covered only equations with rational coefficients. Later, he extended it to Liouvillian coefficients [12, 106]. For second order equations Kovačic [57, 28] developed independently a solution algorithm. Only much later it could be shown that the classification underlying this algorithm can also be derived within the Singer theory [108]. The Kovačic algorithm has been implemented in several computer algebra systems.

An alternative approach based on the *invariant ring* of the differential Galois group was presented by Fakler [31] following ideas going back to Fuchs (see also the work of Singer and Ulmer [108] and van Hoeij and Weil [116], respectively). For second order equations there exists an isomorphism between the invariant ring and the rational solutions of some symmetric power of the differential equation. This isomorphism allows one to derive explicit solution formulae and thus a rather efficient algorithm.[4]

The determination of the *differential Galois group* for a given equation is rather difficult. Some progress has recently been made for second and third order equations [107] where it could be reduced to the problem of finding solutions of some associated linear differential equations in the coefficient field and of factoring such equations. If there was an easy way to compute the group directly, one could probably design more efficient solution algorithms. But currently it is the other way round: the solution algorithms help finding the Galois group.

There has also been some work on the *inverse problem* of differential Galois theory. Here a linear algebraic group is given and the task is to determine a differential equation that has it as differential Galois group. One can prove that such a differential equation always exists [74]. Ramis [89] showed that it is often possible to reduce the inverse problem to the direct problem and then even give an explicit solution.

All the theory mentioned here works only for *irreducible* equations. Thus the (efficient) *factorization* of linear differential operators is an important problem in differential Galois theory. A solution of this problem based on the Newton polygon was recently presented by van Hoeij [115]. Factorization (although only of polynomials) is also an issue in differential ideal theory.

Differential Galois theory also gives algorithms for the construction of (Liouvillian) *first integrals* [69, 88, 120].

[4] A *MuPAD* implementation of this approach (and some related code) can be down-loaded from Fakler's WWW page with the URL `http://iaks-www.ira.uka.de/home/fakler/index.html`.

These can be used to construct explicit solutions [68]. Other applications appear in the theory of completely integrable systems. Ziglin has given an algebraic characterization of such systems based on their monodromy group. His criterion for integrability can be rephrased in terms of certain properties of the differential Galois group [19, 75].

Dynamical Systems

Applications in dynamical systems theory are not in the main stream of computer algebra. Conversely, numerical computations play a much more prominent role in dynamical systems theory than symbolic ones. Nevertheless, the use of computer algebra systems is becoming more and more popular in this field, especially for perturbation analysis [90]. The mainly used feature is the ability to expand functions into series and to manipulate these.

But also classical algebraic problems are of great importance in the study of dynamical systems. For example, before a fixed point can be analyzed it must be determined. This requires the solution of a nonlinear system of algebraic equations. If the vector field is rational, this can be done with Gröbner bases. Often the vector field depends on some parameters. At certain values of these parameters, the properties of the vector field may change, i. e. a *bifurcation* occurs. The determination of these values is a fundamental problem in dynamical systems theory.

Of special interest are here *equivariant systems*, i. e. systems for which the defining vector field is invariant under the action of a symmetry group [44, 45]. In this case one can use linear representation theory and polynomial invariant theory for determining the fundamental invariants and equivariants [38, 124]. Using normal forms they enable the local bifurcation analysis, i. e. the typical bifurcation diagram in the neighborhood of a critical point can be derived.

A typical problem in computer algebra is the determination of *normal forms*. For dynamical systems such normal forms have already been introduced by Poincaré, Birkhoff, Gustavson and many others, often in the context of celestial mechanics [13, 24]. They form the basis for the solution of many problems in dynamical systems theory like for example stability or bifurcation analysis. The main idea behind normal forms is to study the system in the neighborhood of a *fixed point* (or *equilibrium*) and to try to remove by a coordinate transformation as many nonlinear terms from the differential equation as possible. According to the Hartman-Grobman theorem all such terms can be eliminated near a *hyperbolic fixed point* where the Jacobian has no zero or purely imaginary eigenvalues. Thus at a hyperbolic point linear stability theory is sufficient.

Around other types of equilibria like centers the analysis is more involved. This concerns especially Hamiltonian systems where stable fixed points can never be hyperbolic. If there are *resonances* between the eigenvalues of the Jacobian, the normal form is necessarily non-linear. In order to determine a normal form one makes a power series ansatz for the coordinate transformation and determines the coefficients of the ansatz by requiring that besides the resonances all non-linear terms of the differential equations up to a certain order disappear. The resulting transformed differential equations are normal forms.

An algorithm for computing normal forms of a given system that is suitable for implementation in a computer algebra system was presented by Walcher [119]. It is closely related to *Lie transforms* [72]. This technique has its origin in Hamiltonian mechanics where it yields a canonical transformation. However, it can be extended to general dynamical systems. In contrast to this Birkhoff normal form Gatermann and Lauterbach [40] use normal forms from singularity theory in order to study bifurcation phenomena. For equivariant systems they automatically classify them using Gröbner bases.

Another application is the determination of *center manifolds* [17], a special form of invariant manifolds. If a system possesses a center manifold, it often suffices to study the behavior of the system on this manifold. For example, if the zero solution of the reduced system is stable, then solutions of the original system for initial data sufficiently close to the center manifold will approach this manifold exponentially fast. Thus the reduced system completely describes the asymptotic behavior of such solutions.

The main point in using center manifold theory is that it yields a reduction of the dimension and thus typically a considerable simplification of the analysis. Often it is possible to reduce an infinite-dimensional problem to a finite-dimensional one. There are two important computational steps in the application of the theory. First we need an approximation for the center manifold, then we must compute the reduced system. As in normal form theory, this is done step by step with a power series ansatz. Laure and Demay [61] showed for the Couette-Taylor problem how computer algebra and numerical analysis can interact to solve a complicated bifurcation problem for an infinite-dimensional problem using a reduction to a finite-dimensional center manifold.

A more theoretical application of computer algebra concerns *Hilbert's 16th problem* of bounding the number of limit cycles in a planar polynomial system. For quadratic systems a lot of results are known [96]; however already the cubic case becomes very complicated. An important subproblem is the *center problem*, namely to distinguish between a focus and a center. The derivation of sufficient and especially of necessary conditions for a center can be very involved and is sometimes hardly feasible without computer algebra [80]. In a recent study of cubic systems [29] a CRAY-J90 had to be used.

Numerical Analysis

It was already mentioned above that the capabilities of computer algebra systems to explicitly solve differential equations are limited. This holds especially for partial differential equations. Therefore numerical methods have lost nothing of their importance. Symbolic and numerical computations can interact in many ways and most computer algebra systems provide some numerical facilities.

The oldest and simplest approach consists of *interfacing* a computer algebra system and a numerical library. Typically the interaction is one-way: the computer algebra system is used to derive the differential equations (e. g. the equations of motion of a complicated physical system); the interface generates code in the language of the numerical library (perhaps including some optimization steps); finally, the differential equations are solved with some methods from the numerical library.

To some extend most of the common computer algebra systems can do this, as they all provide commands to convert an expression into C or FORTRAN. However, it is rather cumbersome to automatically generate whole programs that way. For such purposes one better uses a specialized package like GENTRAN [41] which is part of the REDUCE distribution. Another problem is the optimization of the generated code which is usually necessary. In REDUCE the package SCOPE [117] can do this.

MuPAD provides a very efficient form of interfacing: *dynamical modules* [110]. These are developed in a language like C or C++ and can be linked to *MuPAD* dynamically at run-time. Compared with approaches based on inter-process communication this leads to much less overhead. As the module has direct access to the internal data of the *MuPAD* session, much less data must be communicated. The procedures implemented in a dynamical module can be called within *MuPAD* like any other function. Provided a convenient interface exists this allows in principle to work within *MuPAD* interactively with a numerical library.[5]

Computer algebra systems can also help to *select* an appropriate method from a numerical library. Modern libraries have reached such a level of sophistication that for many users it is impossible to fully exploit their potential. They provide many different routines for the same task and their working can be further tuned by many input parameters whose meaning remains a secret for non-experts. A computer algebra system can analyze the given differential equation (e. g. estimate its stiffness) and then choose an appropriate method and determine reasonable values for its parameters. An example for this approach is the AXIOM package ANNA developed by Dupée [27].

Goldman *et al.* [43] go considerably further in their application of computer algebra in numerical analysis by using it as a software engineering tool. They automatically *generate* the full FORTRAN code for numerically solving the

[5] A prototypical implementation of an interface to some functions of the NAG C-library is discussed on the *MuPAD* web site at the URL `http://www.mupad.de/PAPERS/MODULES/NAGC/index.html`

Navier-Stokes equations. Their argument is that such programs are so long and complicated that their maintenance and adaption (new boundary conditions, different discretizations etc) is rather difficult and error-prone. They use instead a number of input files that contain all the relevant information about the problem in a format that is comparatively easy to read and let the computer algebra system then generate the source code.

One can also use computer algebra to *derive* numerical schemes. The Butcher theory of Runge-Kutta methods is here a typical example. For higher order methods the order conditions become rather large and complicated. Computer algebra packages have been developed that derive and solve them (using Gröbner bases) [46, 109].[6] For partial differential equations the construction of higher-order discretizations or finite elements can be rather involved and is sometimes only feasible with the help of a computer algebra system [76]. Computer algebra can also assist in proving the stability of finite difference schemes [36].

Another topic where computer algebra plays a certain role in numerical analysis are *differential algebraic equations*. The *index* of such a system comprising differential and algebraic equations measures in a certain sense, how far it is away from a pure differential equation [15]. This gives an indication of the difficulties one must expect in a numerical integration. The determination of the index is essentially equivalent to the completion procedures described above [62, 83, 101], as it can be defined as the number of steps needed for the completion. However, in practice numerical analysts often prefer the use of *automatic differentiation* to computer algebra [16].

None of the applications described so far represents really what one would call a *hybrid* algorithm combining symbolic and numerical elements, i. e. where computer algebra is an integral part of the solution process and not only used to determine either the problem or the numerical method for its solution. However, we are not aware of any such algorithm for differential equations, although some ansätze based on symmetry theory have been developed. Dorotnitsyn [26] showed how one may construct finite difference schemes inheriting the symmetries of a differential equation. Such schemes should very well preserve the associated conservation laws, but so far no numerical tests have been published.

In contrast, for solving algebraic equations several hybrid algorithms have been designed. One of them deals with nonlinear systems possessing symmetries [39], as they e. g. arise in equivariant systems. In the symbolic part it uses the linear representation theory of finite groups to transform the problem into an optimal form for the numerical part. This includes for example a block diagonalization of the Jacobian. The numerical algorithm is complicated due to the underlying group theory. Gatermann [37] showed how the numerical computations can be automated by first computing the necessary group theoretic data which is summarized in a bifurcation graph.

Conclusions and Outlook

The application of computer algebra to differential equations is a vast (and not very well defined) field. We could only briefly indicate some of the main research directions and had to omit many others. For example, one can extend the idea of transforming differential equations far beyond simple heuristics and is then lead to the *equivalence problem of Cartan* [59, 79]. Within the algebraic approaches we ignored the theory of $\mathcal{D}$-modules [67] which is of considerable importance in control theory.

The fields we have touched on are in rather different states. Some of them like symmetry theory are meanwhile fairly mature with the fundamentals well understood and they provide standard techniques for tackling differential equations implemented in many computer algebra systems. Others are still in an early stage of their development and essential questions are open. Such fields are usually known only to some experts and only prototypical implementations of algorithms exist.

One common feature shared by most of the fields is the complexity of the algorithms. If we take the various comple-

[6] Actually, the problem of automatically generating the order conditions for Runge-Kutta methods was mentioned by Jenks [56] in the problem section of the SIGSAM Bulletin already in 1976.

tion methods as example, it is obvious from their close relation to Gröbner bases that their complexity is at least as bad as that of the Buchberger algorithm. Although Gröbner bases solve *in principle* many problems in commutative algebra, it is well-known that one often fails to get a basis in reasonable time. One possible way out is the stronger use of heuristics and techniques from Artificial Intelligence, although this is an unpleasant thought for many pure mathematicians.

Some readers might be surprised that we discussed applications in numerical analysis as broadly as more traditional topics like symmetry theory. But we believe that in the future this direction will be among the most important ones. Despite all the successes of Lie symmetries, differential Galois theory etc. one must clearly see that these theories are of hardly any value for many of the problems an engineer, say, typically faces. A popular benchmark problem for numerical methods for differential algebraic equations comes from vehicle dynamics and models with five links a wheel suspension [52]. Its equations of motion must be generated by computer and consist of 7000 lines of FORTRAN code. It appears hardly realistic to solve such a system with Lie symmetries (if it possesses any!) or any other analytic technique.

This does not imply that there is no point in further studying symbolic methods, not at all! Toy models that can be solved analytically are important for obtaining a deeper understanding of underlying structures. One may hope that such understanding may lead to more efficient numerical algorithms for such large problems. And again we want to stress that the application of computer algebra to differential equations is not restricted to solving them!

Most of the current numerical methods for ordinary differential equations do not take into account any special properties of the equation (except its stiffness). It is a rather new trend in numerical analysis to try to use such properties for the design of better algorithms. One prominent example of this trend are *symplectic integrators* [94] which are superior to most conventional methods in the long term integration of Hamiltonian systems, as they preserve many *qualitative* features of such systems. One might expect that computer algebra will play an important role in this emerging field.

The combination of symbolic and numerical computation will become much more important in the future. In the form of simple interfaces it happens already now in many places and hopefully we can soon add powerful hybrid methods. For most *users* of computer algebra systems (this is a very different community than the participants of computer algebra conferences!) such possibilities are of much greater importance than many of the fancy algorithms developed by theorists.

Acknowledgments

I thank F. Postel for inviting me to write this article. J. Calmet, W. Fakler, K. Gatermann, V. Gerdt, E. Hubert and E. Pflügel read parts of various draft versions and gave valuable comments. This work was supported by Deutsche Forschungsgemeinschaft.

References

[1] O. Aberth. The failure in computable analysis of a classical existence theorem for differential equations. *Proc. Amer. Math. Soc.*, 30:151–156, 1971.

[2] M.J.D Ablowitz, A. Ramani, and H. Segur. Nonlinear evolution equations and ordinary differential equations of Painlevé type. *Lett. Nuovo Cim.*, 23:333–338, 1978.

[3] S.A. Abramov and K.Yu. Kvansenko. Fast algorithms to search for the rational solutions of linear differential equations with polynomial coefficients. In S.M. Watt, editor, *Proc. ISSAC '91*, pages 267–270. ACM Press, New York, 1991.

[4] R.L. Anderson and N.H. Ibragimov. *Lie-Bäcklund Transformations in Applications*. SIAM, Philadelphia, 1979.

[5] M.A. Barkatou. A rational version of Moser's algorithm. In A.H.M. Levelt, editor, *Proc. ISSAC '95*, pages 297–302. ACM Press, New York, 1995.

[6] M.A. Barkatou. An algorithm to compute the exponential part of a formal fundamental matrix solution of a linear differential system. *Appl. Alg. Eng. Commun. Comput.*, 8:1–23, 1997.

[7] O. Becken. Algorithmen zum Lösen einfacher Differentialgleichungen. *Rostocker Informatik Berichte*, 17:1–24, 1995.

[8] G.W. Bluman and J.D. Cole. The general similarity solution of the heat equation. *J. Math. Mech.*, 18:1025–1042, 1969.

[9] G.W. Bluman and S. Kumei. *Symmetries and Differential Equations*. Applied Mathematical Sciences 81. Springer-Verlag, New York, 1989.

[10] F. Boulier, D. Lazard, F. Ollivier, and M. Petitot. Representation for the radical of a finitely generated differential ideal. In A.H.M. Levelt, editor, *Proc. ISSAC '95*, pages 158–166. ACM Press, New York, 1995.

[11] M. Bronstein. Linear ordinary differential equations: breaking through the order 2 barrier. In B.M. Trager and D. Lazard, editors, *Proc. ISSAC '92*, pages 42–48. ACM Press, New York, 1992.

[12] M. Bronstein. On solutions of linear ordinary differential equations in their coefficient field. *J. Symb. Comp.*, 13:413–439, 1992.

[13] A.D. Bruno. *Local Methods in Nonlinear Differential Equations*. Springer-Verlag, Berlin, 1989.

[14] 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.

[15] S.L. Campbell and C.W. Gear. The index of general nonlinear DAEs. *Numer. Math.*, 72:173–196, 1995.

[16] S.L. Campbell and R. Hollenbeck. Automatic differentiation and implicit differential equations. In M. Berz, C. Bischof, G. Corliss, and A. Griewank, editors, *Proc. Second Int. Workshop Computational Differentiation*, pages 215–227. SIAM, Philadelphia, 1996.

[17] J. Carr. *Applications of Centre Manifold Theory*. Applied Mathematical Sciences 35. Springer-Verlag, New York, 1981.

[18] G. Carrà Ferro. Gröbner bases and differential algebra. In L. Huguet and A. Poli, editors, *Proc. AAECC-5*, Lecture Notes in Computer Science 350, pages 129–140. Springer-Verlag, Berlin, 1987.

[19] R.C. Churchill and D.L. Rod. On the determination of Ziglin monodromy groups. *SIAM J. Math. Anal.*, 22:1790–1802, 1991.

[20] P.A. Clarkson and M.D. Kruskal. New similarity reductions of the Boussinesq equation. *J. Math. Phys.*, 30:2201–2212, 1989.

[21] R. Conte. Singularities of differential equations and integrability. In D. Benest and C. Froeschlé, editors, *Introduction to Methods of Complex Analysis and Geometry for Classical Mechanics and Non-Linear Waves*, pages 49–143. Editions Frontières, Gif-sur-Yvette, 1994.

[22] R. Conte, A.P. Fordy, and A. Pickering. A perturbative Painlevé approach to nonlinear differential equations. *Physica D*, 69:33–58, 1993.

[23] J. Denef and L. Lipshitz. Power series solutions of algebraic differential equations. *Math. Ann.*, 267:213–238, 1984.

[24] J. Della Dora and L. Stolovitch. Normal forms of differential systems. In E. Tournier, editor, *Computer Algebra and Differential Equations*, London Mathematical Society Lecture Note Series 193, pages 143–184. Cambridge University Press, 1994.

[25] J. Della Dora and E. Tournier. Formal solutions of differential equations in the neighbourhood of singular points. In P.S. Wang, editor, *Proc. SYMSAC*, pages 25–29. ACM Press, New York, 1981.

[26] V.A. Dorodnitsyn. Finite difference models entirely inheriting symmetry of original differential equations. In N.H. Ibragimov, M. Torrisi, and A. Valenti, editors, *Modern Group Analysis: Advanced Analytical and Computational Methods in Mathematical Physics*, pages 191–201. Kluwer, Dordrecht, 1993.

[27] B.J. Dupée and J.H. Davenport. An intelligent interface to numerical routines. In J. Calmet and C. Limongelli, editors, *Proc. DISCO '96*, Lecture Notes in Computer Science 1128, pages 252–262. Springer, Heidelberg, 1996.

[28] A. Duval and M. Loday-Richaud. Kovačic's algorithm and its application to some families of special functions. *Appl. Alg. Eng. Commun. Comput.*, 3:211–246, 1992.

[29] V.F. Edneral. Computer evaluation of cyclicity in planar cubic system. In W.W. Küchlin, editor, *Proc. ISSAC '97*, pages 305–309. ACM Press, New York, 1997.

[30] P.G. Estévez. Non-classical symmetries and the singular manifold method: the Burgers and the Burgers-Huxley equations. *J. Phys. A*, 27:2113–2127, 1994.

[31] W. Fakler. On second order homogeneous linear ordinary differential equations with Liouvillian solutions. *Theor. Comp. Sci.*, to appear.

[32] A.S. Fokas. Symmetries and integrability. *Stud. Appl. Math.*, 77:253–299, 1987.

[33] B. Fuchssteiner, S. Ivanov, and W. Wiwianka. Algorithmic determination of infinite dimensional symmetry groups for integrable systems in 1+1 dimensions. *Math. Comp. Model.*, 25:91–100, 1997.

[34] A. Galligo. Some algorithmic questions on ideals of differential operators. In B.F. Caviness, editor, *Proc. EUROCAL '85, vol. 2*, Lecture Notes in Computer Science 204, pages 413–421. Springer-Verlag, Berlin, 1985.

[35] G. Gallo, B. Mishra, and F. Ollivier. Some constructions in rings of differential polynomials. In H.F. Mattson, T. Mora, and T.R.N. Rao, editors, *Proc. AAECC-9*, Lecture Notes in Computer Science 539, pages 171–182. Springer-Verlag, Heidelberg, 1991.

[36] V.G. Ganzha and R. Liska. Application of the Reduce computer algebra system to stability analysis of difference schemes. In E. Kaltofen and S.M. Watt, editors, *Computers and Mathematics*, pages 119–129. Springer, New York, 1989.

[37] K. Gatermann. Computation of bifurcation graphs. In E. Allgower, K. Georg, and R. Miranda, editors, *Exploiting Symmetry in Applied and Numerical Analysis*, AMS Lectures in Applied Mathematics 29, pages 187–201. AMS, Providence, 1993.

[38] K. Gatermann. Semi-invariants, equivariants and algorithms. *Appl. Alg. Eng. Commun. Comput.*, 7:105–124, 1996.

[39] K. Gatermann and A. Hohmann. Symbolic exploitation of symmetry in numerical pathfollowing. *Impact Comp. Sci. Eng.*, 3:330–365, 1991.

[40] K. Gatermann and R. Lauterbach. Automatic classification of normal forms. *Nonlin. Anal.*, to appear.

[41] B.L. Gates. A numerical code generation facility for REDUCE. In B.W. Char, editor, *Proc. SYMSAC '86*, pages 94–99. ACM Press, New York, 1986.

[42] V.P. Gerdt. Gröbner bases and involutive methods for algebraic and differential equations. In J. Fleischer, J. Grabmeier, F.W. Hehl, and W. Küchlin, editors, *Computer Algebra in Science and Engineering*, pages 117–137. World Scientific, Singapore, 1995.

[43] V.V. Goldman, J.A. van Hulzen, A.E. Mynett, A.S. Posthuma, and H.J. van Zuylen. The application of computer algebra for the discretization and coding of the Navier-Stokes equations. In A.M. Cohen, L. van Gastel, and S. Verduyn Lionel, editors, *Computer Algebra in Industry 2*, pages 131–150. John Wiley, New York, 1995.

[44] M. Golubitsky and D.G. Schaeffer. *Singularities and Groups in Bifurcation Theory I.* Applied Mathematical Sciences 51. Springer-Verlag, New York, 1985.

[45] M. Golubitsky, D.G. Schaeffer, and I. Stuart. *Singularities and Groups in Bifurcation Theory II.* Applied Mathematical Sciences 69. Springer-Verlag, New York, 1988.

[46] D. Gruntz. Symbolic computation of explicit Runge-Kutta formulas. In W. Gander and J. Hřebí ček, editors, *Solving Problems in Scientific Computing Using Maple and MATLAB*, chapter 19, pages 267–283. Springer-Verlag, Berlin, 1995.

[47] B.K. Harrison and F.B. Estabrook. Geometric approach to invariance groups and solutions of partial differential systems. *J. Math. Phys.*, 12:653–666, 1971.

[48] D.H. Hartley. Involution analysis for non-linear exterior differential systems. *Math. Comp. Model.*, 25:51–62, 1997.

[49] D.H. Hartley and R.W. Tucker. A constructive implementation of the Cartan-Kähler theory of exterior differential systems. *J. Symb. Comp.*, 12:655–667, 1991.

[50] W. Hereman. Review of symbolic software for the computation of Lie symmetries of differential equations. *Euromath Bull.*, 2:45–82, 1994.

[51] W. Hereman. Symbolic software for Lie symmetry analysis. In N.H. Ibragimov, editor, *CRC Handbook of Lie Group Analysis of Differential Equations, Volume 3: New Trends in Theoretical Development and Computational Methods*, chapter 13. CRC Press, Boca Raton, Florida, 1995.

[52] M. Hiller and S. Frik. Road vehicle benchmark 2: Five link suspension. In W. Kortüm, S. Sharp, and A. de Pater, editors, *Application of Multibody Computer Codes to Vehicle System Dynamics, Progress Report to the 12th IAVSD Symposium, Lyon 1991*, pages 198–203, 1992.

[53] E. Hubert. The general solution of an ordinary differential equation. In Y.N. Lakshman, editor, *Proc. ISSAC '96*, pages 189–195. ACM Press, New York, 1996.

[54] E.L. Ince. *Ordinary Differential Equations*. Dover, New York, 1956.

[55] M. Janet. Sur les Systèmes d'Équations aux Dérivées Partielles. *J. Math. Pure Appl.*, 3:65–151, 1920.

[56] R.J. Jenks. Problem #11: Generation of Runge-Kutta equations. *SIGSAM Bull.*, 10(1):6–7, 1976.

[57] J.J. Kovačic. An algorithm for solving second order linear homogeneous differential equations. *J. Symb. Comp.*, 2:3–43, 1986.

[58] E. Kamke. *Differentialgleichungen: Lösungsmethoden und Lösungen*, volume 1. Geest & Portig, Leipzig, 1967.

[59] N. Kamran. The equivalence problem of Elie Cartan, differential equations and computer algebra. In E. Tournier, editor, *Computer Algebra and Differential Equations*, pages 87–114. Academic Press, London, 1988.

[60] E.R. Kolchin. *Differential Algebra and Algebraic Groups*. Academic Press, New York, 1973.

[61] P. Laure and Y. Demay. Symbolic computation and equation on the center manifold: Application to the Couette-Taylor problem. *Comp. Fluids*, 16:229–238, 1988.

[62] G. Le Vey. Differential algebraic equations: A new look at the index. Rapport de Recherche 2239, INRIA Rennes, 1994.

[63] H. Lewy. An example of a smooth linear partial differential equation without solution. *Ann. Math.*, 66:155–158, 1957.

[64] I.G. Lisle, G.J. Reid, and A. Boulton. Algorithmic determination of structure of infinite Lie pseudogroups of symmetries of PDEs. In A.H.M. Levelt, editor, *Proc. ISSAC '95*, pages 1–6. ACM Press, New York, 1995.

[65] M.A.H. MacCallum. Using computer algebra to solve ordinary differential equations. In A.M. Cohen, L. van Gastel, and S.M. Verduyn Lunel, editors, *Computer Algebra in Industry 2*, pages 19–41. John Wiley, New York, 1995.

[66] A.R. Magid. *Lectures on Differential Galois Theory*. University Lecture Series 7. AMS, Providence, 1994.

[67] B. Malgrange. Motivation and introduction to the theory of d-modules. In E. Tournier, editor, *Computer Algebra and Differential Equations*, London Mathematical Society Lecture Note Series 193, pages 3–20. Cambridge University Press, 1994.

[68] Y.-K. Man. Computing closed form solutions of first order ODEs using the Prelle-Singer procedure. *J. Symb. Comp.*, 16:423–443, 1993.

[69] Y.-K. Man and M.A.H. MacCallum. A rational approach to the Prelle-Singer algorithm. *J. Symb. Comp.*, 24:31–43, 1997.

[70] E. Mansfield. *Differential Gröbner Bases*. PhD thesis, Macquarie University, Sydney, 1991.

[71] J.B. McLeod and P.J. Olver. The connection between partial differential equations soluble by inverse scattering and ordinary differential equations of Painlevé type. *SIAM J. Math. Anal.*, 14:488–506, 1983.

[72] K.R. Meyer. Lie transform tutorial II. In K.R. Meyer and D.S. Schmidt, editors, *Computer Aided Proofs in Analysis*, IMA Volumes in Mathematics and its Applications 28, pages 190–210. Springer-Verlag, New York, 1991.

[73] A.V. Mikhailov, A.B. Shabat, and R.I. Yamilov. The symmetry approach to the classification of non-linear equations. Complete lists of integrable systems. *Russ. Math. Surv.*, 42:3–53, 1987.

[74] C. Mitschi and M.F. Singer. The inverse problem in differential Galois theory. In B.L.J. Braaksma, G.K. Immink, and M. van der Put, editors, *Proc. The Stokes Phenomenon and Hilbert's 16th Problem*, pages 185–196. World Scientific, New York, 1996.

[75] J.J. Morales and C. Simo. Picard-Vessiot theory and Ziglin's theorem. *J. Diff. Eqs.*, 107:140–162, 1994.

[76] E.H. Mund. Computer algebra and finite element methods in engineering. In A.M. Cohen, L. van Gastel, and S. Verduyn Lionel, editors, *Computer Algebra in Industry 2*, pages 81–100. John Wiley, New York, 1995.

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

[78] P.J. Olver. Direct reduction and differential constraints. *Proc. Roy. Soc. Ser. A*, 444:509–523, 1994.

[79] P.J. Olver. *Equivalence, Invariants, and Symmetry*. Cambridge University Press, 1995.

[80] J.M. Pearson, N.G. Lloyd, and C.J. Christopher. Algorithmic derivation of centre conditions. *SIAM Rev.*, 38:619–636, 1996.

[81] E. Pflügel. An algorithm for computing exponential solutions of first order linear differential systems. In W.W. Küchlin, editor, *Proc. ISSAC '97*, pages 164–171. ACM Press, New York, 1997.

[82] E. Pflügel. On the latest version of DESIR. *Theor. Comp. Sci.*, to appear.

[83] O.-P. Piirilä and J. Tuomela. Differential-algebraic systems and formal integrability. Research Report A326, Helsinki University of Technology, Institute of Mathematics, 1993.

[84] J.F. Pommaret. *Systems of Partial Differential Equations and Lie Pseudogroups*. Gordon & Breach, London, 1978.

[85] J.F. Pommaret. *Differential Galois Theory*. Gordon & Breach, London, 1983.

[86] F. Postel and P. Zimmermann. A review of the ODE solvers of Maple, Mathematica, Macsyma and MuPAD. In A. Carriére and L.R. Oudin, editors, *Proc. 5th Rhine Workshop on Computer Algebra*, pages 2.1–2.10. Report PR 801/96, ISL, Saint-Louis (France), 1996.

[87] M.B. Pour-el and J.I. Richards. A computable ordinary differential equation which possesses no computable solution. *Ann. Math. Logic*, 17:61–90, 1979.

[88] M.J. Prelle and M.F. Singer. Elementary first integrals of differential equations. *Trans. Amer. Math. Soc.*, 139:167–189, 1969.

[89] J.P. Ramis. About the solution of some inverse problems in differential Galois theory by Hamburger equations. In K.D. Elworthy and L. Markus, editors, *Differential Equations, Dynamical Systems, and Control Science. A Festschrift in Honor of Lawrence Markus*, Lecture Notes in Pure and Applied Mathematics 152, pages 277–299. Marcel Dekker, New York, 1994.

[90] R.H. Rand and D. Armbruster. *Perturbation Methods, Bifurcation Theory and Computer Algebra*. Applied Mathematical Sciences 65. Springer-Verlag, New York, 1987.

[91] G.J. Reid. Algorithms for reducing a system of PDEs to standard form, determining the dimension of its solution space and calculating its Taylor series solution. *Eur. J. Appl. Math.*, 2:293–318, 1991.

[92] G.J. Reid, I.G. Lisle, A. Boulton, and A.D. Wittkopf. Algorithmic determination of commutation relations for Lie symmetry algebras of PDEs. In B.M. Trager and D. Lazard, editors, *Proc. ISSAC '92*, pages 63–68. ACM Press, New York, 1992.

[93] J.F. Ritt. *Differential Algebra*. Dover, New York, 1966. (Original: AMS Colloquium Publications, Vol. XXXIII, 1950).

[94] J.M. Sanz-Serna and M.P. Calvo. *Numerical Hamiltonian Problems*. Applied Mathematics and Mathematical Computation 7. Chapman & Hall, London, 1994.

[95] C. Scheen. Implementation of the Painlevé test for ordinary differential equations. *Theor. Comp. Sci.*, to appear.

[96] D. Schlomiuk, J. Guckenheimer, and R. Rand. Integrability of plane quadratic vector fields. *Expo. Math.*, 8:3–25, 1990.

[97] J. Schü, W.M. Seiler, and J. Calmet. Algorithmic methods for Lie pseudogroups. In N. Ibragimov, M. Torrisi, and A. Valenti, editors, *Proc. Modern Group Analysis: Advanced Analytical and Computational Methods in Mathematical Physics*, pages 337–344. Kluwer, Dordrecht, 1993.

[98] W.M. Seiler. On the arbitrariness of the general solution of an involutive partial differential equation. *J. Math. Phys.*, 35:486–498, 1994.

[99] W.M. Seiler. Arbitrariness of the general solution and symmetries. *Acta Appl. Math.*, 41:311–322, 1995.

[100] W.M. Seiler. Involution and symmetry reductions. *Math. Comp. Model.*, 25:63–73, 1997.

[101] W.M. Seiler. Numerical analysis of constrained Hamiltonian systems and the formal theory of differential equations. *Math. Comp. Simul.*, to appear, 1997.

[102] M.F. Singer. Liouvillian solutions of n-th order homogeneous linear differential equations. *Am. J. Math.*, 103:661–682, 1981.

[103] M.F. Singer, editor. *Differential Equations and Computer Algebra (CADE-90)*, Computational Mathematics and Applications, London, 1990. Academic Press.

[104] M.F. Singer. Formal solutions of differential equations. *J. Symb. Comp.*, 10:59–94, 1990.

[105] M.F. Singer. An outline of differential Galois theory. In E. Tournier, editor, *Computer Algebra and Differential Equations*. Academic Press, New York, 1990.

[106] M.F. Singer. Liouvillian solutions of linear differential equations with Liouvillian coefficients. *J. Symb. Comp.*, 11:251–273, 1991.

[107] M.F. Singer and F. Ulmer. Galois groups of second and third order linear differential equations. *J. Symb. Comp.*, 16:9–36, 1993.

[108] M.F. Singer and F. Ulmer. Liouvillian and algebraic solutions of second and third order linear differential equations. *J. Symb. Comp.*, 16:37–73, 1993.

[109] M. Sofroniou. Symbolic derivation of Runge-Kutta methods. *J. Symb. Comp.*, pages 265–296, 1994.

[110] A. Sorgatz. *Dynamische Module*. MuPAD Reports. Teubner, Stuttgart, 1996.

[111] H. Stephani. *Differential Equations: Their Solution Using Symmetries*. Cambridge University Press, Cambridge, 1989.

[112] E. Tournier, editor. *Computer Algebra and Differential Equations (CADE-88)*, Computational Mathematics and Applications, London, 1988. Academic Press.

[113] E. Tournier, editor. *Computer Algebra and Differential Equations (CADE-92)*, London Mathematical Society Lecture Note Series 193. Cambridge University Press, 1992.

[114] F. Ulmer. On Liouvillian solutions of linear differential equations. *Appl. Alg. Eng. Commun. Comput.*, 2:171–194, 1992.

[115] M. van Hoeij. Formal solutions and factorization of differential operators with power series coefficients. *J. Symb. Comp.*, 24:1–30, 1997.

[116] M. van Hoeij and J.A. Weil. An algorithm for computing invariants of differential Galois groups. *J. Pure Appl. Alg.*, 117&118:353–379, 1997.

[117] J.A. van Hulzen, B.J.A. Hulshof, B.L. Gates, and M.C. van Heerwaarden. A code optimization package for REDUCE. In G.H. Gonnet, editor, *Proc. ISSAC '89*, pages 163–170. ACM Press, New York, 1989.

[118] E. von Weber. Partielle Differentialgleichungen. In *Enzyklopädie der mathematischen Wissenschaften, Vol. II, Part 1.1*, chapter A5, pages 294–399. 1900.

[119] S. Walcher. On transformation into normal form. *J. Math. Anal. Appl.*, 180:617–632, 1993.

[120] J.A. Weil. First integrals and Darboux polynomials of homogeneous linear differential systems. In G. Cohen, M. Giusti, and T. Mora, editors, *Proc. AAECC-11*, Lecture Notes in Computer Science 948, pages 469–484. Springer-Verlag, Heidelberg, 1995.

[121] J. Weiss. The Painlevé property for partial differential equations II: Bäcklund transformations, Lax pairs, and the Schwarzian derivative. *J. Math. Phys.*, 24:1405–1413, 1983.

[122] J. Weiss, M. Tabor, and G. Carnevale. The Painlevé property for partial differential equations. *J. Math. Phys.*, 24:522–526, 1983.

[123] W. Wiwianka and B. Fuchssteiner. Algorithms to detect complete integrability in 1+1 dimensions. In S. Carillo and O. Ragnisco, editors, *Research Reports in Physics – Nonlinear Dynamics*, pages 131–135. Springer-Verlag, Berlin, 1990.

[124] P.A. Worfolk. Zeros of equivariant vector fields: Algorithms for an invariant approach. *J. Symb. Comp.*, 17:487–511, 1994.

[125] Wu W.-T. Automatic derivation of Newton's gravitational law from Kepler's laws. *Academica Sinica Mathematics-Mechanization Research Preprints*, 1:53, 1987.

[126] Wu W.-T. A constructive theory of differential algebraic geometry based on works of J.F. Ritt with particular applications to mechanical theorem proving of differential geometries. In G. Chaohao, M. Berger, and R.L. Bryant, editors, *Differential Geometry and Differential Equations*, Lecture Notes in Mathematics 1255, pages 173–189, Shanghai, 1987. Springer-Verlag, Berlin 1987.

[127] D. Zwillinger. *Handbook of Differential Equations*. Academic Press, San Diego, 1992.