GEOMETRIC-ALGEBRAIC COMPLETION OF LINEAR DIFFERENTIAL EQUATIONS

Marcus Hausdorf[1], Werner M. Seiler[2]
[1] *Institut für Algorithmen und kognitive Systeme, Universität Karlsruhe, 76128 Karlsruhe, Germany*
e-mail: hausdorf@ira.uka.de
[2] *Lehrstuhl I für Mathematik, Universität Mannheim, 68131 Mannheim, Germany*
e-mail: seiler@euler.math.uni-mannheim.de

We present in this a talk a new algorithm for completing systems of linear partial differential equations to involution. It combines ideas from the formal theory of differential equations and methods used in the computation of involutive bases of polynomial ideals and modules. In formal theory, the theorem of Cartan-Kuranishi provides a very useful tool for transforming systems of partial differential equations to formal integrable ones in a geometric framework; this is especially needed when looking for formal power series solution. Implementations of this algorithms to computer-algebra- systems exist, yet they are not very efficient; one has to handle rather large matrices. On the other hand there have been presented good algorithms for determining involutive bases for polynomial ideals and linear differential equations by V.Gerdt. We show how to develop an algorithm that fuses the advantages of both methods to a hybrid. After choosing Pommaret-division for the definition of multiplicative variables, we proceed as prescribed by the Cartan-Kuranishi-theorem: Prolongation until the symbol of the differential equation becomes involutive; afterwards projection and starting over with the new system, if necessary. However, we compute only the nonmultiplicative prolongations of equations and reduce them with respect to the multiplicative prolongations, thus performing essentially the same steps as in the computation of involutive bases. The most important quality of the algorithm is that we do not lose the connection to the differential geometric picture, i.e. we still regard the differential equations as fibered submanifolds of the appropriate jet bundles. For this aim, we introduce the notion of the "skeleton" of a system, which also supplies the suitable data type for implementation. We show how one can perform all tasks of the algorithm - prolongation, triangulation, testing for involutivity of the symbol and system - on the level of skeletons. Finally, we give a short overview of the implementation of these ideas in MuPAD (which has been done by the first author in the course of his diploma thesis) and propose a way how the algorithm could be transferred to polynomial equations, using pseudodivision.