Report on guest professorship mini-course:

Computational Lie Theory

Henrik Winther
Masaryk University, Brno
Czech Republic

January & February 2021

The theory of Lie groups, Lie algebras, and their actions on smooth manifolds arises in a variety of geometric contexts. For example, principal bundles, holonomy representations and homogeneous spaces have Lie theory as their core.

Lie theory also provides highly effective ways to approach geometric problems from a computational perspective. When one is interested in constructing examples with certain properties, or in classifying of some type of geometric structure, it is natural to reach for these tools.

As a TFS Guest Professor, the visitor delivered a mini course providing an introduction to these techniques. In particular, we introduced techniques of symbolic algebra systems and computer implementations of algorithms related to geometric problems and related (Lie)-algebraic problems.

No familiarity with the use of computers in pure mathematics was assumed. Therefore, the lectures started with an introduction to computer algebra, and the delineating the classes of formal theorem provers, symbolic algebra systems, and numerics.

We discussed applications of symbolic algebra to problems in Lie theory and representation theory. In particular, we explained how to represent and solve problems in the two main frameworks for homogeneous geometries, that is the "symbolic calculus in coordinates" framework and the purely algebraic framework coming from the isotropy representation, and how to relate them. In the latter framework, geometrical problems are reduced to pure representation theory.

Thus, we discussed the most common computational problems in representation theory. This includes finding invariants, tensor product decomposition, branching problems, and plethysms. Our computer tools were the DifferentialGeometry package in the program Maple, and the representation theory calculator LiE.

Some representation theory was also covered, in particular regarding representations over real numbers and the real version of Schur's Lemma.

Towards the end of the course, we provided an introduction to the use of Maple as a programming language, in order to teach how to extend its capabilities whenever the built-in algorithms are insufficient to the task at hand. Here we introduced functions and procedures, as well as higher order functions and general programming ideas such as iteration constructs and flow control, variable assignment and variable scope.

The course was intended for faculty, post-docs and interested students. We hope that it left them prepared for future challenges.