Using Interactive Theorem Prover for Scientific Computing

Submitter: Tomas Skrivan
Author: Tomas Skrivan

Modern physics relies heavily on advanced mathematical concepts such as vector calculus, differential geometry, and probability theory. However, translating these concepts into computer code can be a challenging and error-prone task. To simplify this process, we propose using an interactive theorem prover. Interactive theorem provers are capable of expressing any mathematical concept, allowing users to write code using advanced mathematics and then interactively translate it into executable code. This approach can reduce errors, document underlying theory, assist with derivations, and automatically detect errors that may arise when extending existing code by changing basic assumptions.

We are currently using the new interactive theorem prover, Lean 4, to develop scientific computing code. To begin, the user specifies their problem, such as obtaining the optimal solution of an ODE. They can then convert this into executable code by executing a series of commands, such as applying the Runge-Kutta method to solve the ODE, using Gaussian integration to compute an integral, formulating an adjoint problem, or employing symbolic/automatic differentiation.

Our primary focus has been on implementing symbolic and automatic differentiation, including the ability to differentiate higher-order functions. This has numerous applications, such as symbolic variational calculus or differentiating through the solution of differential equations, which leads to the formulation of the adjoint method. Our approach has the distinctive feature of being able to generate a formal proof to verify the correctness of the computed derivative.

The ultimate goal is to create a system where users can specify what they wish to compute, and with guidance from the system, convert their specifications into effective and accurate code.