Contribution ID: **25** Type: **not specified**

Numba-Enzyme: A Python Compiler for Differentiable Simulations and Beyond

We present Numba-Enzyme, a gradient-providing Just-in-time (JIT) compiler for simulations in Python providing rewrite-free access to gradients for Numba, a popular LLVM-based Python compiler for simulations. In recent years a number of simulation areas have started to expand beyond efficient simulations, and began to utilize gradients for gradient-based optimization, differentiable simulation hybrids, scientific machine learning, and physics-inspired machine learning architectures built with simulation blocks. Machine learning frameworks like JAX and PyTorch provide automatic-differentiation capabilities as well as just-in-time compilation for acceleration through tracing or taping, approaches optimized for array-programming, non-mutable operations, and static computational graphs that are typical in machine learning. In contrast, our compiler-based approach can efficiently handle dynamic or unstructured programs that make frequent use of indirections, branches, loops, array mutation, and other patterns commonly used in scientific computing applications without the need for extensive rewrites. Operating at the compiler-level, Numba-Enzyme is furthermore able to seamlessly interoperate with common machine learning frameworks such as PyTorch, and JAX

We will demonstrate the effectiveness of Numba-Enzyme as a differentiable framework across a set of established automatic differentiation benchmarks. Its utility, and performance on modern-day scientific machine learning approaches will be demonstrated with examples from neural ordinary differential equations, and Bayesian uncertainty quantification.

Author: PAEHLER, Ludger (Technical University of Munich)

Presenter: PAEHLER, Ludger (Technical University of Munich)

Session Classification: Computer Science

Track Classification: Computer Science