Short course on Algorithmic Differentiation: A Practical Approach for Engineers

Course Instructor: Pavanakumar Mohanamuraly, Senior Researcher, ALGO Team, CERFACS, Toulouse, France

Algorithmic differentiation (AD), also known as Automatic Differentiation or AutoDiff, is a crucial algorithmic technique in mathematics that has played a significant role in the success of fields such as Machine Learning (ML). AD enables efficient computation of adjoint gradients that are essential for training complex ML models like Large Language Models (LLMs) with billions of parameters. It also plays a pivotal role in multi-disciplinary design optimisation involving very large design spaces and PDE constraints in Aerospace and Mechanical Engineering.

Despite its broad utility and applicability, AD is mostly taught to a niche community of researchers in universities. Access to AD tools and practical application knowledge is limited for engineering students. Specifically, the reverse mode of AD requires adherence to a subset of a given programming language due to specific algorithmic constraints. Domain Specific Languages (DSLs) with AD support like TensorFlow also impose severe restrictions on the programming style. These constraints demand meticulous programming discipline to develop efficient AD-enabled scientific software.

Fortunately, numerous practical solutions to AD-related challenges have emerged and the AD community has been actively developing tools to tackle them. This course aims to democratise AD knowledge and bring it to a wider audience in engineering. Understanding AD is critical for students pursuing problems in gradient-based optimisation, and this course will serve as a useful primer. Given the popularity and prevalence of ML frameworks, we cover aspects of AD applied to ML. We present example problems from engineering and mathematics to demonstrate how AD can be leveraged to solve them efficiently using open-source AD tools in programming languages such as Fortran and Julia.

Key topics addressed in this course

1. AD history and introduction to terminologies
2. Three types of evaluating derivatives : Finite-difference, Symbolic and Algorithmic
3. Forward and Reverse Mode of AD (Back propagation in ML)
4. Adjoint-gradients in design optimisation and ML
5. Programming Paradigms in AD : Source Transformation and Operator Overloading
6. Application to practical problems in Fortran, Julia and Python languages

References

1. Numerical Optimization, Jorge Nocedal and Stephen Wright, Springer, 2006.
2. Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, 2nd Edition, Andreas Griewank, Andrea Walther, SIAM, 2008.
3. The Art of Differentiating Computer Programs: An Introduction to Algorithmic Differentiation, Uwe Naumann, SIAM, 2011.
4. Algorithms for Optimization, Mykel J. Kochenderfer and Tim A. Wheeler, MIT Press, 2022.