

---

# Practical implementation of adjoint algorithmic differentiation (AAD)

Sébastien Geeraert\*<sup>1</sup>

<sup>1</sup>Murex Analytics – Murex – France

## Abstract

Adjoint algorithmic differentiation (AAD) is a method producing the exact derivatives of a computation, in an automatic and efficient way. By using pathwise differentiation with AAD, we can compute Monte Carlo sensitivities of a price, in a time that does not depend on the number of sensitivities. We first show a naive implementation in C++, working on classical processors (CPU) or on graphic cards (GPU). We are confronted with a major drawback of AAD: every intermediate result must be saved, which leads to a large memory consumption. We then present a few possible optimizations to increase parallelization and limit memory consumption, and their practical effects on an example of CVA computation. Thanks to these optimizations, we achieve interesting gains in comparison to finite differences. Lastly, we mention possible ways to handle non-smooth payoffs, such as Vibrato Monte Carlo.

---

\*Speaker