AADC by MatLogica

What is AADC?

Alternatives to AADC

What is AADC?

MatLogica developed a breakthrough technology that enables more efficient use of modern CPUs for mathematically intensive tasks. We designed a compiler that quickly executes complex mathematical functions, and developed an easy-to-use library that speeds up computations and calculates risks automatically.

Mathematically complex tasks in finance and insurance (such as risk management and pricing) and various mathematical physics applications (such as weather modeling, and various types of tomography) are often solved using popular programming languages like C++ or Python. Their object-oriented nature makes it possible to write a readable code that is easy to modify and maintain.

But from a business perspective, computational speed is critical. In financial applications, one often needs to recalculate a portfolio of hundreds of thousands of exotic trades instantly but the system takes minutes to handle them and often requires batch execution. Calculating risks such as delta, gamma, vega, as well as second-order derivatives or sensitivities to model parameters is especially costly and might be impossible. As a result, the business becomes less efficient, decision-making is delayed, and risk management deteriorates.

Traditional methods used to speed up a program include using more computing resources (such as Cloud), rewriting the code for new technologies (such as CUDA and GPU), and adding support for low-level concepts. All these options are expensive, time-consuming, and require a thorough analysis.

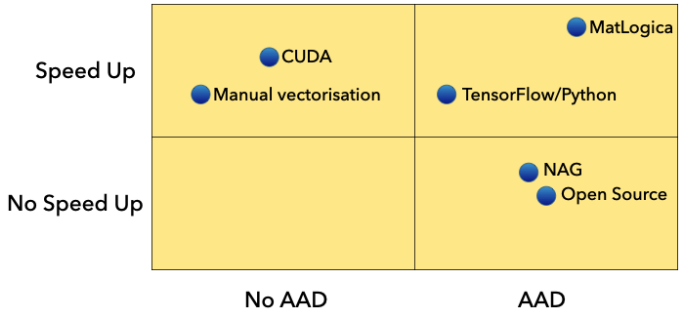
With AADC, a software developer can write in C++ or Python focusing on business problems and the library will take care of performance automatically. We can get an acceleration of 10-100x on a single CPU core using built-in vectorization by making simple changes to the original code using the AADC library. Automatic multi-threading additionally speeds up the program almost directly proportional to the number of cores. Risk calculation is then automatic using AAD and is significantly faster than traditional methods.

Alternatives to AADC

MatLogica's AADC was designed to quickly calculate AAD risks but it can also speed up any “forward” repetitive calculations such as Monte Carlo simulations, “what if” scenarios, historical analysis, and more. Both "forward" and “adjoint” calculations are very important for the finance and insurance industries and are widely used in industrial mathematics and physics, artificial intelligence, etc.

Our competitors can only solve one problem at a time. Some can speed up the "forward" task by extensive efforts in low-level programming or switching to CUDA and GPU. Both approaches require significant investments and usually take years. Libraries such as NAG and some Open Source libraries (Adept, etc.) can be used to solve inverse problems, but they do not speed up the calculation and also take a long time to integrate. TensorFlow/PyTorch works well for machine learning but applying them to other financial modeling tasks is problematic. This approach requires completely rewriting the existing code and memory allocation is not optimal.

MatLogica’s integration approach delivers performance with minimal changes to the existing code and is optimized for tackling complex financial mathematics problems that use multiple points in the valuation graph.



MatLogica’s library is the only technical solution that accelerates a variety of tasks in the finance and insurance sectors and automatically calculates risks using AAD. It is also the only library that can be used to speed up scripting languages.

AADC by MatLogica

How to use AADC?

How does AADC work?

How to use AADC?

Financial institutions

AADC can significantly improve risk management efficiency by accelerating market and credit risk reporting and calculating risks in real-time. The library also speeds up numerical models, including curve calibration, and simplifies parallel computations. This approach is flexible enough to be used for Monte Carlo pricing and an automatic calculation of derivatives.

Industrial mathematics and science

The library can be used to solve problems in mathematical physics such as weather modeling, various types of tomography, and (more generally) restoration of surfaces of unknown parameters in arbitrary models.

Software developers

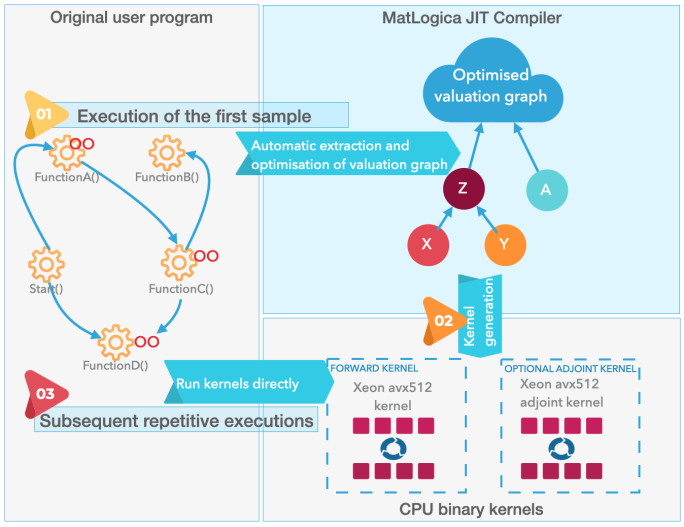
Parallel programming is hard to develop and support in the long run. One of the stumbling blocks is the multi-thread safety requirement that is especially difficult to achieve for industrial software projects that are in constant flux. The unique (patent pending) approach of AADC enables developers to achieve multi-thread safety by converting their object-oriented single-threaded code into binary kernels ready to run in multi-thread mode.

Infrastructure

Using AADC, one can get the best performance for multiple generations of vectorized Intel/AMD processors (supporting avx2 and avx512 architectures) leveraging the existing infrastructure for longer. Our library is cross-platform for supported processor architectures. We are also developing support for graphics card computing (GPU), which will allow customers to use graphics cards without needing to resort to low-level interfaces such as Cuda or OneAPI.

How does AADC work?

While the user program executes the first set of data, our library generates vectorized CPU kernels designed to run in multi-thread mode. The first kernel replicates original calculations and the second (optional) computes the backward pass of automatic differentiation. Tasks such as Monte-Carlo simulations, historical analysis, derivative calculations, model calibration, and curve construction are accelerated by a factor of 6-100, enabling real-time calculations instead of the traditionally used overnight batches.



The library uses an unorthodox integration approach using the Operator Overloading pattern. That allows minimizing changes to the code and completing integration in a matter of weeks, with significant acceleration on modern CPUs. Our approach to integration into industrial projects was tested on well-known libraries (such as Open Risk Engine and QuantLib) and presented to the wide academic and business communities at various finance and financial mathematics conferences.[1, 2]

[1] SIAM Conference on Financial Mathematics and Engineering, 2021 https://meetings.siam.org/sess/dsp_talk.cfm?p=112373

[2] WBS Quant Insights, 2021, spring edition https://www.youtube.com/watch?v=BGBVXmpNi0U