October 2018

MVA Using Algorithmic Differentiation (AD)

BY: QUATERNION RISK MANAGEMENT

At the end of 2018, the financial industry has still not yet established a consensus methodology for calculation of Margin Value Adjustment (MVA) on non-centrally cleared derivatives. MVA represents the expected funding cost of initial margin over the lifetime of a trade/portfolio, and is particularly relevant today due to BCBS-IOSCO 261 – more commonly referred to as the "Swaps Margin Rules", it requires financial entities to exchange sufficient collateral to cover potential losses over a 10-day period with 99% confidence, and is complementary to the margin typically used to settle daily mark-to-market changes (variation margin), phasing-in to cover most derivatives market participants by September 2020. MVA is the most recent valuation adjustment ("xVA"), joining similar calculations for counterparty credit risk (CVA), the funding cost of variation margin (FVA), and the cost of capital (KVA), among others. MVA is particularly difficult to calculate due to the requirements for trade sensitivities along each Monte Carlo simulation path as inputs to ISDA's Standard Initial Margin Model (SIMM). AD provides the most efficient and robust calculation of these in-simulation sensitivities.

What is Algorithmic Differentiation (AD)?

Simply put, AD calculates derivatives in computer programs using the chain rule by exploiting simple arithmetic operations and functions that have known analytical derivatives. It uses a small, constant factor of these operations more than the original program (generally less than 4) to recursively apply the chain rule by fixing either the dependent or independent variables (colloquially, sweeping either "out-to-in" or "in-to-out"; also "backward mode" or "forward mode", respectively). Since the required trade sensitivities are just partial derivatives of the price with respect to each underlying risk factor, AD can be used to more efficiently produce the required trade sensitivities than the more common alternative "bump and revalue" approach, which requires two pricing steps as well as perturbation of the underlying risk factors. "Bump and revalue" is frequently used in risk systems for spot sensitivities but can be very inefficient to apply within Monte Carlo simulations for large portfolios.

ALTERNATE APPROACHES

Quaternion is aware of a wide array of practical approaches to calculate MVA, though each make trade-offs on performance, speed, and accuracy compared to AD…

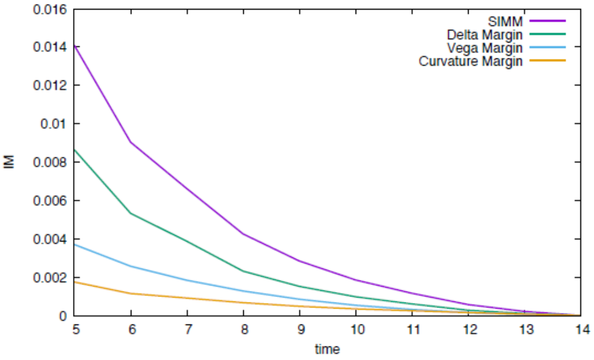
1. Brute force bump & revalue
2. Regression of ΔNPV over MPOR, see "Forecasting IM Requirements - A Model Evaluation" by Caspers, et al. (2017) & "DIM Estimation Based on Quantiles of Johnson Distributions" by McWalter, et al. (2018)
3. Pre-computing sensitivity grids / look-up tables, see "DIM via Chebyshev Spectral Decomposition" by Zeron, et al. (2018)
4. Probability Matrix Method, see "Coherent Global Market Simulations by Albanese, et. Al (2010)

Performance – Single Bermudan Swaption

Quaternion embedded AD within an American Monte Carlo pricing method and benchmarked both accuracy and performance against classic "bump and revalue" sensitivities on a lattice engine. Results below show sensitivities and SIMM for a "5 into 10" Bermudan swaption, calculated along 5,000 paths on a single core MacBook Pro with no parallelization.

Single Trade Conclusions

- ✓ AD produces accurate in-simulation sensitivities
- ✓ Timing increases linearly on # simulation paths; "forward mode" increases w/ # of risk factors, while "backward mode" increases w/ # of future grid points
- ✓ Memory consumption remains limited without the need for "tape compression" (i.e. memory does not increase with the number of simulation paths)



Setup: 5K sim. paths, 10 future grid points, single-core (no parallelization)

Step	Task	Time (ms)
T_0 Evaluation	Pricing	33
	CG Setup + Initialize Simulation Values	*3*
	Trade Evaluation	*30*
Future Sensitivities	10 Sweeps (1 per future grid point)	274
TOTAL	BACKWARD MODE	307

Bermudan Swaptions on Realistic Computation Grid

Compared to the previous single-trade test, Quaternion also tested a more granular computation grid (300 forward points for risk factor evolution, 30 forward points for sensitivity calculation) with five currencies (resulting in 9 total risk factors, i.e. by counting each currency interest rate and FX pair). The steps below isolate the effects of only these assumption/set-up changes on performance compared to the previous example:

	Single Bermudan Example (Above)	**More Granular CG** (Trade Independent)		**Increased # of Currencies**	
T_0 Evaluation	33ms	99ms	~3X	595ms	~6X
Future Sensis (Backward)	274ms	1225ms	~4.5X	1250ms	
SUBTOTAL	**307ms**	**1324ms**		**1845ms**	

Performance – Large IR/FX Portfolio Tests

To examine how AD performance scaled to realistic portfolios, Quaternion randomized large interest rate and FX portfolios consisting of Swaps, Bermudan Swaptions, FX Forwards, and FX Options across five currencies.

Large Portfolio Conclusions

- ✓ Computation time is linear in the # of trades & paths
  - ○ Can be easily/efficiently parallelized
  - ○ Performance scales linearly on available hardware
- ✓ Memory consumption is independent of # trades & paths
  - ○ Peak memory remains manageable for large portfolios
- ✓ Efficiency gains depend on # risk factors & trade maturity
  - ○ More risk factors ~linearly increases time & memory
  - ○ More granular CG ~linearly increases time & memory
- ✓ For a Bermudan Swaption, compare AD at 1.3s/trade to "B&R" on an American Monte Carlo engine at 9.9s/trade (i.e. using Longstaff-Schwartz, 2001) and "B&R" on a naïve lattice engine at 77.5min/trade

Setup: 5K paths, random IR/FX portfolio, single-core (no parallelization)

Step	Task	Time	Peak Mem.
T_0 Evaluation	Pricing		
	CG Setup + Initialize Sim. Values	*0.5s*	*0.5 GB*
	Trade Evaluation	*17.0s*	*2.0 GB*
Future Sensis	Averaged over full portfolio		
	Backward Mode (30 Sweeps)	*2.4s / trade*	*1.4 GB*
	Forward Mode (9 Sweeps)	*1.9s / trade*	*4.0 GB*

HARDWARE VS. SOFTWARE INVESTMENT?

Banks must decide whether to invest finite resources in software enhancements, such as analytical methodologies and/or AD, or hardware investments in GPUs or multi-core cloud computing.

Front-office research teams may be able to avoid the enormous computational resources required for brute-force MVA by pushing calculations to overnight batch runs and/or pre-calculating certain trade sensitivities, but for true real-time intra-day measures or marginal "what-if" pre-trade analytics they must implement more clever analytical approaches or invest heavily in multi-core processors. Graphics Processing Units (GPUs) seem to have the most promising applications for MVA, with some firms even specializing in the sale or lease of expensive GPUs specifically for quantitative finance. Hardware innovation will continue to have seemingly limitless growth potential – for example, just this year Google went public with the industry's first Tensor Processing Unit, reducing dependency on the market-leading chipmaker NVIDIA and providing a more-than-formidable challenger to GPUs – but rather than attempt to keep pace with constantly outdated hardware, Quaternion strongly recommends that firms invest in the most efficient software methodology and then attempt to scale performance from there based on available hardware. AD provides the most efficient combination of accuracy and speed, at the expense of moderate additional coding effort to make pricing models compatible.

> *"There is a perfect match with [AD and] brute force, but there is enormous acceleration with respect to brute force – several hundred times." - Alexandre Antonov, Standard Chartered, on the benefits of AD for MVA, Risk.net 4 Sep 2018*

Quaternion
Risk Management

OPEN
SOURCE RISK

Quaternion Risk Management Ltd.
54 Fitzwilliam Square
Dublin 2 | D02 X308 | Ireland

Office: +353 1 678 7922
Email: info@quaternion.com
www.quaternion.com