REPRINTED FROM
Structured products
Risk publications
JULY 2015
risk.net

Structured products desks join the AAD revolution

Dealers in the structured products market have begun applying adjoint algorithmic differentiation to a variety of complex pricing and analytics problems, with a staggering increase in computation speeds as a result. But challenges remain before the technique can be utilised more widely. Clive Davidson reports

Dealers in the structured products market are used to facing down complex risk management challenges. But the continued need to do so quickly and accurately across a large, diverse book of products is throwing up a more mundane issue: a lack of processing capacity. Calculating a product's sensitivity to a huge variety of risk factors, commonly known as Greeks, is placing an ever-increasing strain on the modelling and computational demands of issuers' pricing and risk models.

Many dealers have sought clever ways to optimise their models, for instance by deploying computer grids, clouds and other forms of parallel processing to speed up calculations. But what if the Greeks could be calculated simply as a by-product of the original valuation? This may sound unlikely, but in fact it can be achieved by applying a mathematical technique known as adjoint algorithmic differentiation, or AAD. The technique has long been used in science and engineering and is now being applied to complex financial pricing and risk calculations.

Although still relatively new to the world of finance (see box AAD: a very brief history), some firms are already applying the technique to their structured products businesses. France's Natixis, for example, is using AAD with equity derivatives for calculating correlation sensitivity, or cega - said to be one of the most costly calculations in Monte Carlo pricing.

The results, says Adil Reghai, the bank's head quantitative analyst, have been little short of staggering. "We went from a calculation with 10,000 machines taking 24 hours, to 2,500 machines taking two hours," he says.

Societe Generale Corporate and Investment Banking (SG CIB), too, has developed tools that compute sensitivities of equity and forex structured products to market data - including computations it wouldn't previously have been able to perform, says Alexandre Charitopoulos, a member of the global markets quantitative research team at the bank.

"Getting the sensitivities to all market data, including to all the points of rate curves and implied volatility surfaces, is something that was not possible to do before AAD and it is very relevant for an exotics business," he says.

AAD is already being used in several other structured products applications. Nomura, for example, says it has successfully applied AAD to some of its most complex products, while Danske Bank has recently begun using AAD for making value-at-risk calculations in a more efficient way.

Vancouver-based vendor Fincad says some of its clients are using its own implementation of AAD, which it calls Universal Algorithmic Differentiation, for structured products, including a major Taiwanese bank that is using it for hedging target redemption forwards, an Austrian bank that is using it for managing equity basket exposures and a large European bank that uses it for CVA calculations on a portfolio covering 15 currencies.

Beware the Greeks

The traditional way of computing Greeks, such as delta, vega and theta, uses finite differences in a process commonly known as 'bumping'. In this approach, each input to an instrument valuation is increased - or bumped - by a small amount and the value recomputed, with the total time taken to compute the Greeks equal to the original valuation time multiplied by the number of inputs.

With the number of inputs for an instrument of any complexity typically at least 50 and often a lot more, there is clearly a high cost in terms of computation, which explains why the industry has been throwing high-performance computing solutions at it.

AAD exploits the fact that the calculation to find the value comprises a sequence of steps where basic arithmetic operations, such as additions or subtractions, or mathematical functions, such as log or sin, are carried out. By recording the sequence then applying the so-called chain rule of differentiation, the derivatives of each step can calculated with relatively few extra operations. This process is known as algorithmic - or sometimes automatic - differentiation.

But the clever part from a financial engineering point of view is that the recording of the valuation is run backwards, or adjoint. This means the time taken to calculate the derivatives is determined by the number of outputs - in this case one, the value - rather than the number of inputs, which can number 50 or more. Furthermore, the results are exact and without the rounding error that can be problematic with bumping.

"With algorithmic differentiation, the computation of all sensitivities takes roughly five

"With algorithmic differentiation, the computation of all sensitivities takes roughly five times longer than calculating one present value. With bump and recompute, each sensitivity takes roughly the same time as one present value"

Marc Henrard, OpenGamma

times longer than calculating one present value. With bump and recompute, each sensitivity takes roughly the same time as one present value," says Marc Henrard, head of quantitative research at London-based vendor OpenGamma.

Taking the example of computing sensitivities for an interest rate swap that depends on two curves, each with 25 data points, the bump and recompute approach will take roughly 50 times longer than simply calculating its present value, while the AAD approach will take only five times longer - a ratio of 10.

But for a more complex pricing model calibrated to a basket of vanilla instruments the ratio can be above 50, says Henrard. In other words, the standard bump and recompute approach would take an hour, whereas the algorithmic differentiation computation would take just one minute.

Memory trouble

Before AAD can be applied more widely across exotics desks, however, there are several barriers that must be overcome. The first problem is for quants to simply get their heads around the idea.

"It is so conceptually different from the way people normally use the computer - a complete upside-down way of thinking is required to implement algorithmic differentiation," says Jesper Fredborg Andreasen, head of quantitative research at Danske Bank in Copenhagen.

Incorporating AAD into the bank's credit valuation adjustment (CVA) calculations for counterparty credit risk has significantly reduced processing time and hardware requirements, says Andreasen.

The learning curve is steep and it can take considerable time to get comfortable with the approach even for experienced mathematicians, adds Mike Giles, professor of scientific computing at the Oxford-Man Institute of Quantitative Finance. "The maths involved in AAD is not difficult, but it's not very intuitive - so you have to have the mathematical self-confidence to proceed," he says. "After a few years, it gets better."

There are other barriers, too. One of the critical issues that quants face in implementing AAD is that, though the technique dramatically reduces processing time compared with bumping, it requires more memory, which can become problematic if not managed carefully.

The steps along the way to an initial valuation must be stored so the sequence can be replayed in reverse order to generate the Greeks. However, done naively, storing every element of the calculation can result in code that is "algorithmically efficient but explosive memory-wise," says Reghai.

Avoiding this requires a detailed understanding of the pricer that is being AAD-enabled, as well as prototyping and testing, and technology tricks for more efficient memory use - "which is why I say quants and IT have to work together to make AAD happen successfully," says Reghai.

Another issue is that, although AAD is in theory a universal approach, it cannot simply be tacked onto firms' existing pricing libraries. "The pricing library has to have been designed for analytics Greeks from its inception. Every single library function needs to have a Greeks version as well. This is simple to do if you are starting from scratch, but it could be very expensive to retrofit to an existing library," says Martin Baxter, head of global markets quantitative research at Nomura International. The bank has been using a version of the technique it calls analytic differentiation since 1998.

And apart from the problems of the underlying design, few existing libraries will have clean code that is not riddled with fixes, extensions and other inconsistent amendments, says Charitopoulos of SG CIB. "Implementing AAD on the scale of the pricing library of a large investment bank will likely require you to rewrite the entire library. This could be very costly," he says, adding that the most successful applications of AAD so far have generally been new developments, or judicious use of the approach on carefully selected problems.

Here, however, there are signs of progress. As Risk.net reported in January, a significant advance has been the development of a tool by Uwe Naumann, professor at RWTH Aachen University and a technical consultant at Numerical Algorithms Group, that will AAD-enable code written in the C++ language, which is widely used for pricing libraries.

Meanwhile, third-party software vendors have already started building the technique into their own libraries. Fincad can claim a first in this regard, having incorporated its own implementation of AAD into its F3 platform, which it launched in 2010. OpenGamma has also implemented AAD throughout its libraries, and other vendors, such as Calypso and Numerix, are exploring the technique.

Second-order problems

AAD has other limitations. While it is highly efficient at generating first-order Greeks - delta, vega and theta - it is less so for second-order Greeks, such as gamma, vanna and charm.

"Even if there were no computational challenges [in using AAD for second-order Greeks] - and there are - the information content is immense, because it scales as the square of the number of deltas. The cost of trying to interpret this complete set of information in a useful manner outweighs any benefit," says Russell Goyder, director of quantitative research and development at Fincad. One solution, he says, is to use the traditional bump method on subsets of the Greeks generated by AAD.

Another wrinkle is that, although AAD produces highly accurate Greeks, they are not as readily usable for hedging in the same way as Greeks produced by bumping.

"The Greeks produced by AAD are derivatives, as opposed to differences produced by traditional bumping methods, which allows them to be used for hedging

"Our estimate is that quant development time is doubled in order to write the AAD Greeks code as well as the pricing code"

Martin Baxter, Nomura International

purposes. One has to perform additional transformations to make the Greeks produced by AAD usable in the context of hedging," says Satyam Kancharla, chief strategy officer and senior vice-president of the client solutions group at Numerix.

That may not matter, however. Henrard of OpenGamma says he expects one of the significant drivers of interest in AAD among dealers with sizeable structured products businesses to be the International Swaps and Derivatives Association's search for an appropriately risk sensitive, easily replicable standard model to calculate margin requirements for non-cleared derivatives.

From September 2016, the computation and exchange of initial margin for non-cleared deals will be mandatory for major derivatives players, with the obligation extended to all market participants by 2019.

Market participants are broadly agreed that developing a Standard Internal Margin Model (or Simm) that only captured an instrument's sensitivity to first-order Greeks would be sufficient for risk management purposes. In fact, many add that a model tuned to capture second-order sensitivities could prove counter-productive, either by tying up more margin to cover increasingly less likely risks or by encouraging counterparty disputes.

"The current proposal by Isda would be to use a so-called Simm approach based on the trade sensitivities. It means that in a couple of years all derivatives participants will need to compute all sensitivities with respect to all risk factors - at least on a daily basis - for the computation of margins. This will probably be a significant change for a lot of structured products houses," says Henrard.

GPU, AAD, A-OK

The emergence of AAD follows the interest by many firms and software vendors in using graphics processing units (GPUs) for speeding up calculations. Like graphics rendering tasks, pricing and risk calculations can be broken down into many relatively simple operations that can be run simultaneously, exploiting GPUs' parallel processing architecture.

AAD and GPUs have been portrayed as competing solutions in some quarters, although this is only true in terms of a firm's resources. AAD is software and GPUs are hardware, while computing is the interaction of the two.

"The two sides [of scientific computing], mathematics and hardware, are generally orthogonal, not mutually exclusive, and AAD is no exception to this," says the Oxford-Man Institute of Quantitative Finance's Giles.

John Ashley, senior IBM software developer relations manager at GPU-maker Nvidia, adds: "In computational finance, there is no single silver bullet. AAD is an algorithmic advance in the computation of the sensitivities of a function. GPUs are parallel compute accelerators. The two are complementary."

What AAD and GPUs share is a requirement for new skills and a reworking of code to take advantage of the new methods. "Our estimate is that quant development time is doubled in order to write the AAD Greeks code as well as the pricing code," says Baxter.

Like AAD, GPUs require a conceptual shift, say vendors - from thinking in terms of a single thread of code to parallel programming - and a revision of library code. With most computational advances, they point out - whether software or hardware based - an initial period of hand-crafting and experiment is followed by the emergence of tools to aid custom implementation, followed by the incorporation of the advance into core in-house or standardised third-party systems. **SP**

AAD – A VERY BRIEF HISTORY

Every so often, there comes along an IT advancement that purports to be a game changer. Based on the results dealers have achieved when applying it to a variety of tasks thus far, AAD has a strong claim to this category. Experience and skills in its use are still rare, but already the technique is allowing quants to look at problems in a new way, and to contemplate issues that were once considered out of bounds.

"AAD is useful for anything in finance, including model types that we haven't even dared think about yet," says Jesper Andreasen, head of quantitative research at Danske Bank in Copenhagen.

Given this, and the fact that the technique has been around since at least the 1950s, it is somewhat surprising that AAD was not picked up earlier by the financial sector, and that it is not being more widely applied yet.

One reason suggested for the former is that many quants tend to come from a theoretical physics background, where the technique is less well known compared with other areas of science and engineering.

Although there are a relatively large number of tools available already developed for other industries, AAD pioneers in finance have mostly been hand-crafting, partly owing to the memory issues that result from naive application and the so-called dirty pricing code found in many banks, and the requirement for code transparency in model validation.

Another is sheer scepticism when the capabilities are presented. In 2010, an audience of quants heckled Russell Goyder, director of quantitative research and development at Fincad, out of disbelief that the calculations being shown in a demonstration of his company's implementation of AAD in its F3 valuation and risk platform were real.