Institution: University of Oxford		
Unit of Assessment: 10: Mathematical Sciences		
Title of case study: Efficient adjoint sensitivities in computational finance		
Period when the underpinning research was undertaken: 2006 - 2013		
Details of staff conducting the underpinning research from the submitting unit:		
Name(s):	**Role(s) (e.g. job title):**	**Period(s) employed by submitting HEI:**
Mike Giles	Professor	1992 – present
Period when the claimed impact occurred: 1 August 2013 – 31 December 2020		
Is this case study continued from a case study submitted in 2014? Y		

1. Summary of the impact

The large investment banks in London each have thousands of servers largely devoted to Monte Carlo simulations and, to quantify their risks and satisfy regulatory demands, they need to be able to calculate huge numbers of financial option sensitivities known collectively as "Greeks".

An adjoint technique developed by Professor Mike Giles at the University of Oxford in 2006 greatly reduced the computational cost of these calculations. The technique is used extensively now by most leading banks, enabling them to perform much more detailed calculations, in particular addressing new risk measurement and management methodologies required by international banking regulators. Without this mathematical approach, the banks would need to use many thousands of additional servers, with a very significant financial and energy cost.

The importance of the adjoint approach (also known as AAD – Adjoint Algorithmic Differentiation) led to a UK SME, NAG (the Numerical Algorithms Group), developing new software during 2009 – 2014 to support banks in implementing this new approach to computing sensitivities. This has generated significant revenue for them over the past 7 years, consolidating their pre-eminence in providing mathematical software to the banks.

2. Underpinning research

Adjoint techniques are a well-established set of mathematical methods that have been extensively used in engineering design optimisation to simultaneously and efficiently compute the sensitivity of a single output quantity with respect to a large number of input parameters. Professor Mike Giles has been a leading researcher in the use of adjoints in engineering design optimisation; his introductory article with Niles Pierce on the subject in 2000 has been cited almost 1,000 times according to Google Scholar.

When he switched research fields from computational fluid dynamics to computational finance, Giles recognised the opportunity to apply the adjoint technique to Monte Carlo option pricing in finance in order to much more efficiently compute option price sensitivities (known in the industry as "Greeks"). These Greeks are used to estimate, and thereby minimise, possible future losses due to changes in, for example, stock prices, interest rates, and exchange rates. In January 2006, together with Professor Paul Glasserman from Columbia University, he published the paper "Smoking adjoints: fast Monte Carlo Greeks" [1] in Risk, the leading publication for those working in quantitative finance within investment banks and other financial institutions. This is the key paper underpinning this Impact Case Study.

The adjoint approach can be applied at various levels, from a line-by-line treatment of a computer code which in computer science is referred to as Adjoint Algorithmic Differentiation

(AAD), up to the formulation of adjoints for significant mathematical operations. One piece of research by Giles within the latter category was on linear algebra operations relevant to key steps in Monte Carlo simulation and time-marching in financial PDE simulations [2]. An expanded technical report included the derivation of the adjoint sensitivities for eigenvalue and eigenvector calculations [3], and this is now cited in the source code of TensorFlow, PyTorch and Theano, three of the major open-source machine learning packages; in the context of neural networks, adjoint sensitivity analysis is known as “back propagation”.

A key technical limitation in the application of adjoints in computational finance was the fact that the adjoint approach requires differentiability, but many financial option payoffs are discontinuous. To address this issue, Giles invented the "vibrato" Monte Carlo method [4], which is a hybrid mix of the pathwise sensitivity method (which the adjoint treatment is based on) and the alternative, less efficient, Likelihood Ratio Method.

The specific requirements of correlation Greeks, that is computing the sensitivity of an option price to changes in any of the many elements in the correlation matrix, is addressed in [5], a Risk paper written by Giles and Dr Luca Capriotti, who is Managing Director, Global Head for Quantitative Strategies at Credit Suisse, and probably the leading proponent of adjoint techniques within the finance industry. In this paper, a new idea is introduced, again specific to the application of adjoints in Monte Carlo simulation, for batching samples in a way which minimises the computational cost (through requiring just one adjoint Cholesky factorisation per batch) while also providing multiple batch averages and their sensitivities from which a Monte Carlo confidence interval can be derived.

Manual implementation of discrete adjoint methods can be time-consuming and error-prone. Fortunately, much of the implementation can be automated using forward and reverse mode automatic differentiation methods developed in computer science. This was introduced in the finance context in [5], and further expanded on in [6], another Risk paper written by Giles and Capriotti, which was re-published in 2016, along with [1], in a book on ‘*Landmarks in XVA’,* edited by two leading finance industry experts.

3. References to the research

[1] M.B. Giles, P. Glasserman. ‘*Smoking adjoints: fast Monte Carlo Greeks*’, Risk, 19(1):88-92, January 2006. https://www0.gsb.columbia.edu/faculty/pglasserman/Other/RiskJan2006.pdf Re-published in 2016 in ‘Landmarks in XVA’, edited by Chris Kenyon and Andrew Green, Risk Books, ISBN: 9781782722939 (available on request).

[2] M.B. Giles. ‘*Collected matrix derivative results for forward and reverse mode algorithmic differentiation’*, pp. 35-44 in Advances in Automatic Differentiation, Springer, 2008. DOI: 10.1007/978-3-540-68942-3_4

[3] M.B. Giles. ‘*An extended collection of matrix derivative results for forward and reverse mode algorithmic differentiation’*, Oxford University Numerical Analysis Report 08/01, 2008. https://ora.ox.ac.uk/objects/uuid:8d0c0a29-c92b-4153-a1d2-38b276e93124

[4] M.B. Giles, ‘*Vibrato Monte Carlo sensitivities’*, pp.369-392 in Monte Carlo and Quasi Monte Carlo Methods 2008, Springer, 2009. DOI: 10.1007/978-3-642-04107-5_23

[5] L. Capriotti, M.B. Giles. ‘*Fast correlation Greeks by adjoint algorithmic differentiation’*, Risk, 23(4):77-83, 2010.https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1587822

[6] L. Capriotti, M.B. Giles. ‘*Algorithmic differentiation: adjoint Greeks made easy’*, Risk, 25(10), 2012. Re-published in 2016 in ‘Landmarks in XVA’, edited by Chris Kenyon and Andrew Green, Risk Books, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1801522

4. Details of the impact

The research by Giles has enabled the major "Tier 1" investment banks to much more efficiently compute the sensitivities ("Greeks") they require for risk management and satisfy increasingly stringent regulatory checks without greatly increasing their already huge computational facilities. This saves both money and energy, with corresponding environmental benefits.

The scale of the savings is indicated by a 2020 industry survey [A] which estimates the worldwide High Performance Computing (HPC) market at about USD39,100,000,000 per year in 2019, with the Banking, Financial Services and Insurance sector approximately 12% of the total. Servers largely devoted to Monte Carlo simulations comprised a significant fraction of this 12%.

In addition, a UK SME, the Numerical Algorithms Group (NAG), has gained financially from developing and selling mathematical software to the banks to support them in their implementation of the new adjoint approach.

From research to impact

The "Smoking adjoints" paper [1] was recognised immediately as a significant advance in the state-of-the-art, addressing a very important industry need. In January 2007 the paper was voted by the finance industry readers of Risk as the best paper of 2006, with Giles and Glasserman being jointly named "Quant of the Year" by the magazine [B]. The paper was re-published in 2016 as a chapter in a book on ‘Landmarks in XVA’, edited by two leading finance industry experts Kenyon and Green, indicating its continuing importance.

Another indication of the significance of paper [1] in stimulating work within banks is given by Capriotti, who writes [D]: “*As you know, when I joined the modeling team at Credit Suisse in 2005, coming from my background in theoretical physics, one of the first things I was asked to do ... was to read your [Risk] paper and understand the methodology. … I was then able to compute first order sensitivities at a greatly reduced computational cost for a range of applications far beyond what* [was] *initially presented in the paper, including products with complex and path dependent payoffs. This led to a decision to incorporate the methodology into our main pricing engine, and since this would be critical to our future software developments we protected our legal position by filing a US patent application in 2008 which was finally granted in 2015: (US Patent number US9058449B2). … Credit Suisse's pioneering role in the use of AAD was also mentioned in a number of editorials in Risk Magazine, including the one covering the 2013 Credit Derivative of the House Award.”*

Giles and Capriotti manually applied the Adjoint Algorithmic Differentiation (AAD) ideas to produce highly optimised adjoint code for the test cases presented in [1,4,5,6]. This manual process is error-prone when tackling very large legacy code-bases. To address this limitation and accelerate the take-up of adjoints within the finance industry, in 2009 NAG worked with computer scientist Professor Uwe Naumann from RWTH Aachen University to develop the software package for AAD called **dco**. This package largely automates the development of adjoint codes, although at the cost of producing less efficient code due to the technical details (operator overloading and “taping”). The 2018 online article [E] by Naumann refers to the *“seminal paper titled ‘Smoking Adjoints: fast Monte Carlo Greeks’ published in Risk Magazine in January 2006”* and discusses progress in AAD in the 10+ years since then, including his development of the **dco** software which is supported and marketed by NAG. Naumann’s work has further encouraged the take-up of AAD within the sector.

The transfer of adjoint ideas to the finance industry has also been facilitated by a one-day course on "Adjoint Methods for Option Pricing" given by Giles, Naumann and Capriotti at the leading international conference for quants and other researchers in the finance industry, QuantMinds (formerly Global Derivatives). The course has been delivered annually from 2013 to 2019, with 10-20 attendees each year with, at most, two from the same company.

Nature and extent of the impact

At the time of the REF2014 submission, the VP of Global Markets at NAG estimated that approximately 20% of Tier 1 banks had adopted the adjoint approach for computing sensitivities for certain classes of financial products. They now estimate [C] that 70% of the Tier 1 banks are using AAD, while Capriotti [D] believes that all of the Tier 1 banks are now using it. As explained in both [C] and [D], this growth has been partly driven by regulatory changes after the financial crisis of 2008. Capriotti [D] in particular writes:

"What has greatly increased the importance of adjoints has been the development of XVA, i.e., the stream of Valuation Adjustments (VAs), which started with CVA (Credit Valuation Adjustment) but now includes lots of other adjustments, such as those associated with own credit (DVA), funding (FVA) and capital costs (KVA). Some of these XVA calculations are a very important part of the Basel III regulatory environment introduced [in 2013] *by the international Basel Committee on Banking Supervision. They all require a portfolio level calculation involving the simulation of many, often, hundreds of risk factors. The calculation of the sensitivities of such VA, required for hedging, is often impractical without AAD or a very large amount of computational power; it would often require 100x more computation using traditional bumping methods* [which use a finite difference approximation to the sensitivity].

The current importance of adjoints to the finance industry can be seen in many ways: the continuing popularity of the course which the two of us teach with Uwe Naumann at the annual QuantMinds conference; the high proportion of talks in the main part of the conference which talk about AAD for XVA; the number of articles which discuss the use of AAD; the number of citations of my adjoint papers."

In a video interview [F], the Head of Quantitative Research at Danske Bank, which uses NAG's **dco** in developing their software, says: "*The purpose of AAD in finance … is real-time intra-day risk management for derivatives and XVA. Without AAD it couldn't be done."* Naumann also expands on the importance of XVA, saying in a second video interview [F] "*With developments like XVA and FRTB* [another new regulatory requirement from the Basel Committee] *the number of sensitivities is growing … so that AAD is becoming essential*".

The significance of Giles' research on adjoints to the banking industry is reinforced in a letter by a Managing Director at Scotiabank who says [G]:

"*I am writing to highlight the importance of adjoint algorithmic differentiation (AAD) to the calculation of Derivative Valuation Adjustments or XVAs, and the lasting influence of your paper on AAD, 'Smoking Adjoints: fast Monte Carlo Greeks'. All banks with derivative portfolios are required by accounting standards and banking regulation to perform XVA calculations, and in my experience most banks seek to actively hedge the market risk sensitivities of XVAs. As you know XVA calculations are amongst the most computationally demanding in modern finance, requiring the valuation of the banks derivative trades many thousands of times inside a Monte Carlo simulation just to obtain the basic XVA values. For those banks that seek to risk manage XVAs and for those that seek to use the forthcoming FRTB-CVA regulatory capital framework, sensitivities to market data inputs must also be calculated. XVA depends on thousands of market data inputs and traditional finite difference techniques would lead to thousands of recalculations of the XVAs to obtain sensitivities. In practice banks that do not have access to AAD are forced to either use a vast compute grid to perform these calculations or severely limit the accuracy or number of sensitivities that are calculated."*

NAG's webpages [H] detail its extensive AAD offerings, including consultancy services to assist banks in developing adjoint software. The VP Global Markets at NAG states [C]:

[text removed for publication]

The VP continues:

"*It seems the banks are still expanding their computing facilities, and they were already huge; it's hard to know how much bigger they would need to be if they didn't have adjoint capabilities.*

Ultimately, I think all of this can be traced back to your Smoking Adjoints paper, either directly through its publication in Risk and the talks you gave a few years ago at industry conferences, or indirectly through the follow-on work, publications and conference presentations of people like Luca Capriotti."

In the context of Machine Learning, adjoint methods are known as "back propagation". The source code files for the adjoints of the high level numerical linear algebra functions of the three major Machine Learning packages TensorFlow [I], PyTorch [J] and Theano [K] collectively cite just 13 papers and only reference [3] is cited by them all. These packages are used many millions of times each day and further demonstrate the wide reach of Giles' research on adjoints.

5. Sources to corroborate the impact

[A] HPC market analysis on the Grand View Research website, 2020, https://www.grandviewresearch.com/industry-analysis/high-performance-computing-market
[B] Risk website report on the Risk Quant of the Year award, 2007: http://www.risk.net/risk-magazine/feature/1498251/quants-paul-glasserman-michael-giles
[C] Letter from VP Global Markets, Numerical Algorithms Group (NAG), 22 Feb 2021,
[D] Letter from Luca Capriotti, Managing Director, Global Head for Quantitative Strategies, Credit Suisse, 10 Aug 2020.
[E] Informa Quant Finance website article by Uwe Naumann, 'Lessons from 10+ years of Algorithmic Differentiation in computational finance', 2018, https://knect365.com/quantminds/article/363a6ca9-1e89-4e17-be12-a58e7c160793/lessons-from-10-years-of-algorithmic-differentiation-in-computational-finance
[F] Informa Quant Finance website video interviews with industry experts on AAD, 2019, https://knect365.com/quantminds/article/c64d1720-16fe-4c8e-be08-394fbd62ddbd/demystifying-aad-what-is-the-role-of-adjoint-algorithmic-differentiation-in-quant-finance (videos available on request)
[G] Letter from Managing Director and lead XVA Quant at Scotiabank, 30 Sept 2020
[H] NAG webpages on the AAD tools and services it offers [accessed 31 Dec 2020]:
https://www.nag.co.uk/content/adjoint-algorithmic-differentiation
https://www.nag.co.uk/content/algorithmic-differentiation-software
https://www.nag.co.uk/content/nag-algorithmic-differentiation-services
[I] TensorFlow github page citing reference [3] in lines 23-27, 2020 [accessed 31 Dec 2020]: https://github.com/tensorflow/tensorflow/blob/master/tensorflow/python/ops/linalg_grad.py
[J] PyTorch github page citing reference [3] in line 1626, 2020 [accessed 31 Dec 2020]: https://github.com/pytorch/pytorch/blob/506142ac8aebe0b3794ba66d6d6c4019fc2182c8/tools/autograd/templates/Functions.cpp
[K] Theano github page citing reference [3] in lines 276-278, 2020 [accessed 31 Dec 2020]: https://github.com/Theano/Theano/blob/master/theano/tensor/slinalg.py