Eureka!

Not the one in California, the one that Ryan Ferguson yelled out when he had the idea for the AI foundations of Riskfuel. Dan Tudball pans for gold with Ferguson.

"This truly is a 'Eureka!' sort of story. It wasn't something where I was playing around with it for years and then finally thought it was time, it was like; 'This is a great idea!'"

Ryan Ferguson is in ebullient form as he chats with *Wilmott* about the roots of Riskfuel, an AI-based accelerator for valuation and risk workloads he founded in 2019, and who would begrudge him his enthusiasm? The firm is picking up awards and great word-of-mouth all over the quantosphere, although *Wilmott* suspects that ebullience is likely Ferguson's default mode rather than a transitory state.

Ferguson gained his doctorate in Physics from Imperial College, London, in 2002, held a variety of quanty roles and rose over 15 years at Scotiabank Global Banking and Markets to the position of Managing Director and Head of Securitization, Credit Derivatives, and XVA before leaving to set up Riskfuel two years ago.

It's in the hallowed XVA halls of Scotiabank that the Archimedean moment occurred, although not a drop of bathwater was spilt.

It began with a conversation between Ferguson and Andrew Green about how to tackle the models that Green was building for the new XVA system. The particular issue was how to deal with the large number of low-materiality models that they wanted to run in the system. This led to the throwing around of ideas that were popular at the time – different types of approximations happened to be one of them. This triggered a memory for Ferguson, which very swiftly led to a rather big "Aha!"

In that moment, Ferguson was transported back to his days at the University of Waterloo, where he did his Master's degree in Electrical Engineering between 1995 and 1997. "…my project was actually an application of neural networks," Ferguson shares, in Total Recall fashion. "In that application, I used a neural network to learn a simulator [that] had nothing to do with finance, but the idea that occurred to me is that 'Hey! What I did back in 1995 ... that's going to work for this problem!'

"The biggest surprise to me in all of this was how I could learn something 25 years ago, then go off to industry and deal with a frustrating problem as a quant and then as a trader – namely, slow models – for 15 or more years, and then finally, after 15 years, one day realize that I've had a potential solution to the problem all along! I mean, what's going on there!?"

But first, this

Consider valuation, just bog-standard day-to-day trading book valuation. For each trade, maybe it takes a second to run a model. Of course, trad-

ers don't just want a single valuation, they want all of the risk sensitivities as well, which means typically a bump and revaluation for each of, say, 50 risk sensitivities. A bank with a million trades in its book will have to do one million valuations and 50 million risk sensitivities.

Most existing banks probably have some sort of valuation system that runs on CPUs and have thought about how to make it performant enough to do XVAs.

XVA sees your trading book valuation problem and raises it by whatever magnitude induces hyperventilation.

XVAs require the calculation of potential future valuations under a wide range of scenarios. Someone with 20,000 scenarios who wants to know the value along each scenario path at 50 different time points would need a million valuations for one instrument. If a bank has a million trades and needs to value them a million times each, that's a trillion valuations up from the 51 million.

If you insist on torturing your CPUs and you're running something that takes a second with a certain level of accuracy using the traditional models, you can turn down the accuracy and make it significantly faster. A fairly typical approach with XVA involves turning down the accuracy until the numerical noise just gets too much to bear.

If you don't want your treatment of CPUs to turn up on Amnesty International's social media feed, you'd look into GPUs. Implementing the XVA valuation on GPU means that a quant development team will have to write that code or port that code from your library that runs on CPU. They will have to write a lot of code.

"There's an upfront cost there, [in] that the quant developers need to do a lot of programming of the new models so that they run on the GPU," Ferguson says. "The thought is, [for] some of these models that might have only a handful of trades, the improvement isn't going to be worth it. But we want it to be in the system that is being developed; we don't want to have [two] separate systems. But it's painful to do a lot of work for something that doesn't have a lot of materiality."

Ryan Ferguson

XVA sees your trading book valuation problem and raises it by whatever magnitude induces hyperventilation

In the Riskfuel approach, quants don't have to write these new models. It's just a case of running the existing model to create a big training set which enables a neural network to imitate the existing CPU-based model. You are not working with a true copy but an excellent approximation of the original model which, on the inside, does things completely differently. Ferguson says, "Essentially, it's a neural network and it's using neural network inferencing, which is really a form of parametric nonlinear regression analytics. It's no longer using things like Monte Carlo, so, as a result, it's a million times faster but you get the same answers."

Back to that

The electrical engineering problem that Ferguson was solving back in 1995 was to do with transistor design. His supervisor had a simulator which took the physical characteristics of a transistor and then, through the simulation, produced the electrical characteristics. Ferguson explains, "You would specify the size, the doping concentrations, and various material parameters, and it would [tell you] how fast it was going to be and how much energy it was going to use, and all these other properties.

"I was looking at it as a mapping from one space into another. You have a description of something in the physical space and there is a mapping to it in another space – namely, the electrical properties space. That's just basically a fancy [way of] saying it's a function. I was learning a function."

When Ferguson had that conversation with Green, he thought: "'I know how to learn a function.' I [can] use a neural network to learn this function.

"Instead of [the] physical properties of a transistor, why don't we throw in things like the state of the market or the characteristics of the trade as the inputs and then its value as the output?" Working from that premise, Ferguson very quickly saw that it worked well. "The thing that was the biggest surprise to me was that it was way, way faster than the way we normally do things, which is to do a Monte Carlo simulation or [use] a finite differences method."

The fast runtime performance is another way of saying cheaper: less compute power is needed to get the same results. "If you have a class of calculations that you solve millions of times each day, it makes sense to pay the cost of training a neural network up front and amortize the cost every time that you use it. That's the Riskfuel approach," says Ferguson, before questioning quants' personal hygiene: "Like buying a washing machine as opposed to going to the laundromat all the time – if you need to wash your clothes once, you're not going to buy a washing machine, but if you're doing that on a regular basis, and I hope we all are, eventually it makes sense to."

A few months after the epiphany, the ideas that underpin Riskfuel made their debut in the form of a preprint that Ferguson published with Green on SSRN and arXiv. The paper conceptually showed the possibilities, but that paper didn't demonstrate its applicability to what Ferguson would call 'a production level model.'

"We used Quantlib – it had a moderate level of complexity. I think it was the first model that really demonstrated the potential performance gains. I've found since writing that paper, people had written papers

about using neural networks to learn the Black–Scholes equation – but there isn't a problem with Black–Scholes; it's analytic, it's fast."

But, as you know, there's more to this world than Black–Scholes. "I wouldn't have quit my job, which I rather liked, if it were a solution to only one particular model!" Ferguson exclaims. "I saw this as a generally applicable technique that can be applied to pretty much any of the valuation models that quants use today. Some don't need it. Some are already analytic and quite fast. But there's an awful lot that use Monte Carlo or finite differences or trees."

Real-time sensitivities were a pipe dream as long as the cost of compute remained exorbitant

What was on the table represented a breakthrough for anyone running a book.

Ferguson recalls his time as a trader. The official PNL and risk sensitivities came with breakfast. The day's activity would be assessed via those risk sensitivities, which were the result of an overnight compute run. The resulting PNL submitted at the end of the day was, especially in volatile times, crying out for something that just wasn't on offer. Real-time sensitivities were a pipe dream as long as the cost of compute remained exorbitant.

Ferguson says, "I started thinking, what would we do if risk compute were a million times cheaper? My thought was 'Wow, you know what? This is going to change how we do things.'"

In quick order, Ferguson announced his retirement from Scotiabank shortly after the paper went public. Staying on until March 2019 to ensure a smooth transition, he then threw himself into Riskfuel.

Riskfuel

The publication of the original paper, collaborations with the likes of Microsoft, and further published research induced the expected: a lot of people wondering about the accuracy of Riskfuel's approach.

"When people ask me about the accuracy of a Riskfuel model, what they are really asking is how well does it mimic the original because we're talking about a huge multidimensional space potentially filled with discontinuities. And the answer is that with good training, a neural network model can be made virtually indistinguishable from the original.

"But I'll give you a more provocative answer to the accuracy question," says Ferguson. Riskfuel's approach can produce production models that are more accurate than the original. I think people forget that these numerical approaches, like finite difference grids and Monte Carlo, are themselves approximations. They approximate the answer as you increase the number of scenarios or the density of the grid that's being solved on, and with that, compute time goes up. A lot of [the time] models at the production settings have a fair amount of numerical error in them. There's a trade-off that's being consciously made between compute time and accuracy."

With the Riskfuel approach, the accuracy–performance trade-off is moved to a one-off upfront training step. Ferguson says: "You need to think about the problem upfront, so that during training you can dial in the accuracy you want. In terms of accuracy, though, nothing is out of the question if you put in the effort to generate the necessary training set.

"I love the technical stuff." Ferguson shares. "I was a quant and then I was a trader. I loved being a trader because even if you were trading the same product, the market changes, and the people and the technology that people are using, it's in a constant state of reinvention. It really suited my desire to keep learning."

For Ferguson, Riskfuel has kicked that excitement up a notch. "I was responsible for a narrow area of expertise inside the bank and trading complicated derivatives – which was right up my alley. Obviously, with Riskfuel, I am still involved in solving those problems, but I'm also trying to figure out how to hire people, how to sell the product, how to do all this stuff. [There are] great short-term problems that I can get sucked in[to] and contribute on, but then there's the big story arc as well, which is kind of fun."

The entrepreneurial experience has been an eye-opener. Ferguson says: "The demonstration of the concept came quickly, but we have spent most of our time just ironing out all the little issues that might stand in the way of getting these sorts of models, [not only] into production at a client, but also for us to be able to produce them at an economic scale."

Ferguson sees this as a particularly exciting time to be a quant, with a proliferation of innovative potential solutions to problems that the industry has faced for years. A practitioner's response to the tumult will be telling, he suggests. "Some people just dive right in and like the fact that their toolbox is getting new and powerful tools. There is a flip side of that coin; some people are exceptionally adept at using the current tools in that toolbox, and they'd like things to stay the same. I think those people are conflicted on whether to support the direction that things are taking or to resist it.

"In the end, machine learning is a tool!" Ferguson exclaims. "If people use it properly, it's just going to make them more effective practitioners. It's a real opportunity for quants to build more realism into their models, and more accuracy, without having to compromise for performance. Ultimately, machine learning liberates quants so they can build the models they've always wanted."

Riskfuel

For more information about Riskfuel, visit www.riskfuel.com.

FURTHER READING

Finder, I. Azure GPUS with Riskfuel's technology offer 20 million times faster valuation of derivatives. Available at https://azure.microsoft.com/en-ca/blog/azure-gpus-with-riskfuels-technology-offer-20-million-times-faster-valuation-of-derivatives/.
Kudus. K. Ultra-fast and accurate derivatives pricing with deep learning. Available at https://medium.com/riskfuel/ultra-fast-and-accuratederivatives-pricing-with-deep-learning-2976a84bbc25.