From CFD to computational finance (and back again?)

Mike Giles

mike.giles@maths.ox.ac.uk

Oxford University Mathematical Institute

Oxford-Man Institute of Quantitative Finance

Bill Morton's 80th Birthday Conference

Oxford, May 29, 2010

Computational Finance

Options pricing – investment banks

- Monte Carlo methods (60%)
- PDEs / finite difference methods (30%)
- other semi-analytic methods (10%)

High-frequency algorithmic trading – hedge funds

Computational Finance

Might seem a bad time to be in this business, but as an academic it's fine:

- clear need for better models
- regulators (and internal risk management) are demanding more simulation
- computational finance accounts for 10% of Top500 supercomputers
- still plenty of MSc students willing / able to fund themselves
- only problem is lack of research funding

Computational Finance

Computational finance is where CFD was 20-25 years ago

- not many academics working on numerical methods
- codes are small – my biggest is probably 1000 lines
- still lots of low-hanging fruit, but maybe more on Monte Carlo than on PDE side
- Olivier Pironneau, Peter Forsyth and others moved earlier from CFD to finance, but kept to PDEs
- Monte Carlo researchers have mainly come from theoretical physics and statistics
- in banks, each product group often has its own codes; consolidation into a single corporate Monte Carlo system for both London and New York is underway

SDEs in Finance

In computational finance, stochastic differential equations are used to model the behaviour of

- stocks
- interest rates
- exchange rates
- weather
- electricity/gas demand
- crude oil prices
- …

The stochastic term accounts for the uncertainty of unpredictable day-to-day events.

SDEs in Finance

Examples:

- Geometric Brownian motion (Black-Scholes model for stock prices)

$$\mathrm{d}S = r\,S\,\mathrm{d}t + \sigma\,S\,\mathrm{d}W$$

- Cox-Ingersoll-Ross model (interest rates)

$$\mathrm{d}r = \alpha(b - r)\,\mathrm{d}t + \sigma\,\sqrt{r}\,\mathrm{d}W$$

- Heston stochastic volatility model (stock prices)

$$\begin{aligned} \mathrm{d}S &= r\,S\,\mathrm{d}t + \sqrt{V}\,S\,\mathrm{d}W_1 \\ \mathrm{d}V &= \lambda\,(\sigma^2 - V)\,\mathrm{d}t + \xi\,\sqrt{V}\,\mathrm{d}W_2 \end{aligned}$$

with correlation ρ between $\mathrm{d}W_1$ and $\mathrm{d}W_2$

Generic Problem

Stochastic differential equation with general drift and volatility terms:

$$\mathrm{d}S(t) = a(S,t)\,\mathrm{d}t + b(S,t)\,\mathrm{d}W(t)$$

$W(t)$ is a Wiener variable with the properties that for any $q<r<s<t$, $W(t)-W(s)$ is Normally distributed with mean 0 and variance $t-s$, independent of $W(r)-W(q)$.

In many finance applications, we want to compute the expected value of an option dependent on the terminal state

$$P \equiv f(S(T))$$

Standard MC Approach

Euler-Maruyama discretisation with timestep h:

$$\widehat{S}_{n+1} = \widehat{S}_n + a(\widehat{S}_n, t_n)\, h + b(\widehat{S}_n, t_n)\, \Delta W_n$$

In the scalar case, each ΔW_n is a Normal random variable with mean 0 and variance h.

Simplest estimator for expected payoff $\mathbb{E}[P]$ is an average from N independent path simulations:

$$\widehat{Y} = N^{-1} \sum_{i=1}^{N} \widehat{P}^{(i)}$$

May seem very simple-minded but it's hard to improve on the Euler discretisation, and many codes are this simple.

The "Greeks"

As well as estimating the value $V = \mathbb{E}[P]$, also important to estimate various first and second derivatives for hedging and risk management:

$$\Delta = \frac{\partial V}{\partial S_0}, \qquad \Gamma = \frac{\partial^2 V}{\partial S_0^2}, \qquad \text{Vega} = \frac{\partial V}{\partial \sigma}$$

In some cases, can need 100 or more first order derivatives, so use of adjoints is natural

“Smoking Adjoints”

First finance paper in 2006 was with Paul Glasserman from Columbia Business School:

- “Smoking Adjoints: fast Monte Carlo Greeks” in Risk, a monthly publication for the finance industry
- explains how to use discrete adjoints for an important application which requires lots of Greeks
- Yves Achdou and Olivier Pironneau had previously used adjoints for finance PDEs, but the technique hadn’t been transferred over to the Monte Carlo side
- absolutely nothing novel from an academic point of view, but has had an impact in the industry – I think a number of banks now use it

“Smoking Adjoints”

The adjoint implementation is based on pathwise sensitivity analysis which relies on the identity

$$\frac{\partial}{\partial \theta}\,\mathbb{E}[P] = \mathbb{E}\left[\frac{\partial P}{\partial \theta}\right]$$

but this breaks down if P is discontinuous.

There are some other ways of treating this case, but they don’t have efficient adjoint implementations.

I’ve recently developed a new way of handling the discontinuity (a hybrid combination of two old methods) which does retain an efficient adjoint implementation.

… and next?

Coming from CFD, the use of adjoints was quite natural

What else is there? Multigrid!

But there's no iterative solver here – instead just keep the central ideas of

- a nested sequence of grids
- fine grid accuracy at coarse grid cost

Multilevel Monte Carlo

Consider multiple levels of simulations with different timesteps $h_l = 2^{-l}\,T,\ l = 0, 1, \ldots, L$, and payoff $\widehat{P}_l$

The expected value on the finest level, which is what we want, can be expressed as a telescoping sum:

$$\mathbb{E}[\widehat{P}_L] = \mathbb{E}[\widehat{P}_0] + \sum_{l=1}^{L} \mathbb{E}[\widehat{P}_l - \widehat{P}_{l-1}]$$

The aim is to estimate the quantity on the left by independently estimating each of the expectations on the right, and do so in a way which minimises the overall variance for a fixed computational cost.

Multilevel Monte Carlo

Key idea: approximate $\mathbb{E}[\widehat{P}_l - \widehat{P}_{l-1}]$ using N_l simulations with $\widehat{P}_l$ and $\widehat{P}_{l-1}$ obtained using same Brownian path.

$$\widehat{Y}_l = N_l^{-1} \sum_{i=1}^{N_l} \left(\widehat{P}_l^{(i)} - \widehat{P}_{l-1}^{(i)} \right)$$

Why is this helpful?

- $\widehat{P}_l \approx \widehat{P}_{l-1}$ since both approximate P
- $V_l \equiv \mathbb{V}[\widehat{P}_l - \widehat{P}_{l-1}]$ is small, especially on finer levels
- fewer samples needed for to estimate $\mathbb{E}[\widehat{P}_l - \widehat{P}_{l-1}]$
- end up using many (cheap) samples on coarse levels, and few (expensive) samples on fine levels

Multilevel Monte Carlo

This has led to a number of papers, covering both applications and numerical analysis. Main point is a big reduction in computational cost for many problems.

To achieve a root-mean-square accuracy of ε:

- cost of standard approach is $O(\varepsilon^{-3})$
- cost of multilevel approach is $O(\varepsilon^{-2})$
- cost is further reduced using quasi-random numbers

Back to CFD?

Have recently started a new project with Rob Scheichl (Bath) and Andrew Cliffe (Nottingham) applying these ideas to oil reservoir and nuclear waste repository simulation,

Here we have an elliptic SPDE coming from Darcy's law:

$$\nabla \cdot \big(\kappa(x) \nabla p\big) = 0$$

where $\log \kappa(x)$ is Normally distributed with a spatial covariance such as

$$\mathsf{cov}(\log \kappa(x_1), \log \kappa(x_2)) = \sigma^2 \exp(-\|x_1 - x_2\|/\lambda)$$

Back to CFD?

In oil reservoir simulation, we're mainly interested in average behaviour, such as the homogenisation of fine-scale structure.

In nuclear waste repository simulation we are interested in the (hopefully very low) probability of contamination exceeding some threshold.

In both cases we can formulate the problem as needing to determine an expected value

Back to CFD?

When λ is large, so there is strong spatial correlation, it is a relatively low-dimensional problem, and “polynomial chaos” or Karhunen-Loeve expansions work well.

However, when λ is small, these become prohibitively expensive and so Monte Carlo methods are used, despite requiring huge numbers of simulations.

Since the computational cost goes up rapidly with resolution, we think multilevel methods have a lot to offer, with most simulations being carried out on very coarse levels.

Back to CFD?

Preliminary numerical analysis suggests we can obtain the following asymptotic costs for ε r.m.s. accuracy

dim	MC	MLMC
1	ε^{-3}	ε^{-2}
2	ε^{-4}	$\varepsilon^{-2}(\log \varepsilon)^2$
3	ε^{-5}	ε^{-3}

and preliminary 1D / 2D numerical results by Rob Scheichl's student Aretha Teckentrup are very encouraging.

Back to CFD?

Another project concerns the use of GPUs for HPC

- latest NVIDIA GPUs have up to 448 cores
- 1 GPU is 10–20$\times$ faster than 2 CPUs, with similar cost and power consumption
- programmed in C with some extensions
- ideal for trivially-parallel Monte Carlo simulations
- also very effective for finite difference applications
- new project with Rolls-Royce, BAE Systems and Paul Kelly at Imperial College addresses the needs of unstructured grid applications through a general purpose open-source library and program transformation tools

Conclusions

- Having great fun with Monte Carlo simulation – very refreshing to do something completely different, and yet still be able to exploit old techniques from CFD
- The differences between the engineering industry and the big investment banks are fascinating
- Some of the Monte Carlo developments are now feeding back into CFD for quantifying the consequences of uncertainty
- Also having great fun with GPU computing – I think GPUs are having a big impact on scientific computing

For more, see: `www.maths.ox.ac.uk/~gilesm/`