nZetta Derivatives Pricing Toolkit: 2D-PDE

Jason Charlesworth (Zettamatics & NAG)

2D-PDE: What's the Problem?

The Feynman-Kac formula allows us to go from stochastic processes to partial differential equations (PDEs). In finance, many low-factor calibration and pricing problems are best treated by numerical PDEs, e.g., the Heston Local Stochastic Volatility model forms the backbone of many EQ and FX products. Classical numerical PDE methods are ill-suited to achieve the fastest performance on modern hardware suffering from loop-carried dependencies and cache misses. Designed from the outset for modern hardware, **the nZetta Toolkit achieves a 10x performance improvement** over reference 2D-PDE calculations on real finance problems.

Tridiagonal Solvers

PDE methods such as Craig-Sneyd and Hunsdorfer-Verwer rely on successive 1D PDE solutions of a tridiagonal problem in the $x-$ and $y-$ directions.

$$\begin{pmatrix} b_0 & c_0 & 0 & 0 & \dots & 0 \\ a_1 & b_1 & c_1 & 0 & \dots & 0 \\ 0 & a_2 & b_2 & c_2 & & 0 \\ 0 & 0 & a_3 & b_3 & \ddots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \dots\ a_{N-1} & b_{N-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-1} \end{pmatrix} = \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{N-1} \end{pmatrix}$$

Forward Pass

$$c'_0 = \frac{c_0}{b_0}, \quad c'_i = \frac{c_i}{b_i - a_i c'_{i-1}} \quad i = 1 \to n-1$$

$$d'_0 = \frac{d_0}{b_0}, \quad d'_i = \frac{d_i - a_i d'_{i-1}}{b_i - a_i c'_{i-1}} \quad i = 1 \to n-1$$

Backward Pass

$$x_{n-1} = d'_{n-1},$$
$$x_i = d'_i - c'_i x_{i+1} \quad i = n-2 \to 0$$

The standard Gaussian elimination-based Thomas algorithm involves a forward and backward pass, each with loop-carried dependencies; value i cannot be calculated until the value $i-1$ has been completed.

These inhibit the compiler's use of SIMD vectorization and degrade the x86's use of superscalar processing.

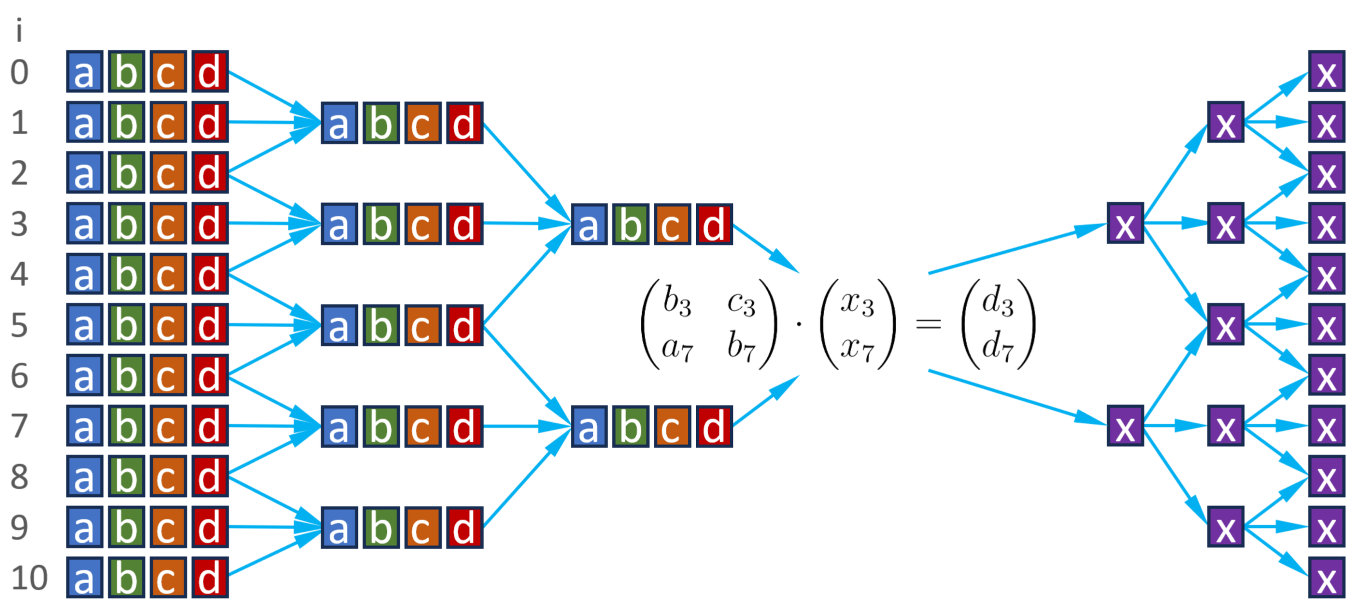
Cyclic Reduction: More is Less

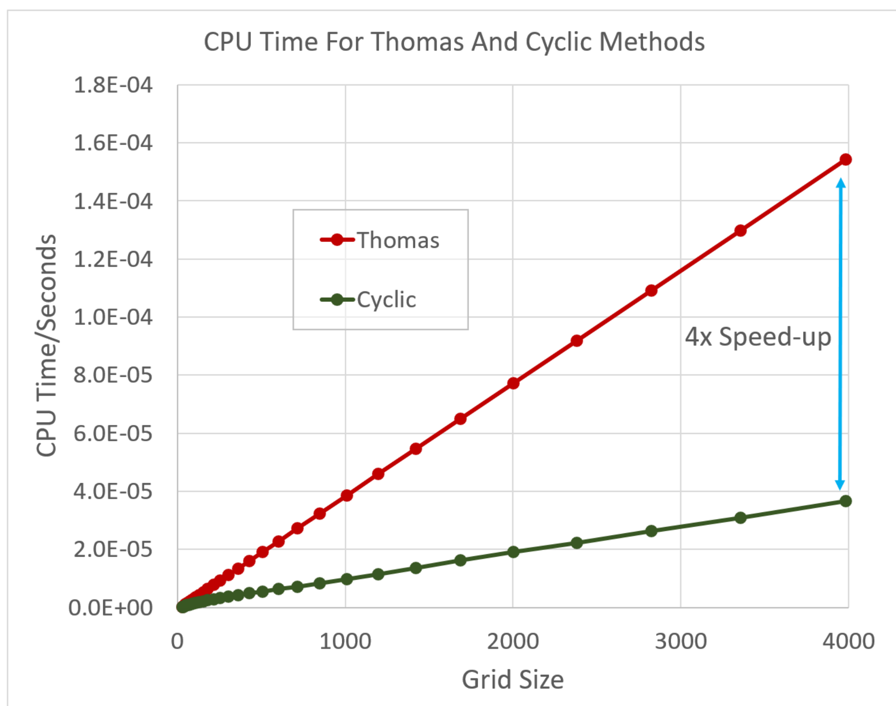
The Cyclic reduction method recursively splits the tridiagonal matrix problem into smaller problems until the smallest problem can be solved analytically. These are used to successively calculate the solution.

$$\begin{aligned} a_{i-1}x_{i-2} + b_{i-1}x_{i-1} + c_{i-1}x_i &= d_{i-1} \\ a_i x_{i-1} + b_i x_i + c_i x_{i+1} &= d_i \\ a_{i+1}x_i + b_{i+1}x_{i+1} + c_{i+1}x_{i+2} &= d_{i+1} \end{aligned}$$

$$a'_i x_{i-2} + b'_i x_i + c'_i x_{i+2} = d'_i$$

This method is more complex than the Thomas algorithm, involving extensive book-keeping for the general case and $\sim 2\times$ the number of calculations. However, at each recursion level, **all** calculations are independent, allowing **full** use of the SIMD and superscalar capabilities. Cyclic reduction is $\sim$**4**$\times$ **faster** than the standard Thomas algorithm.

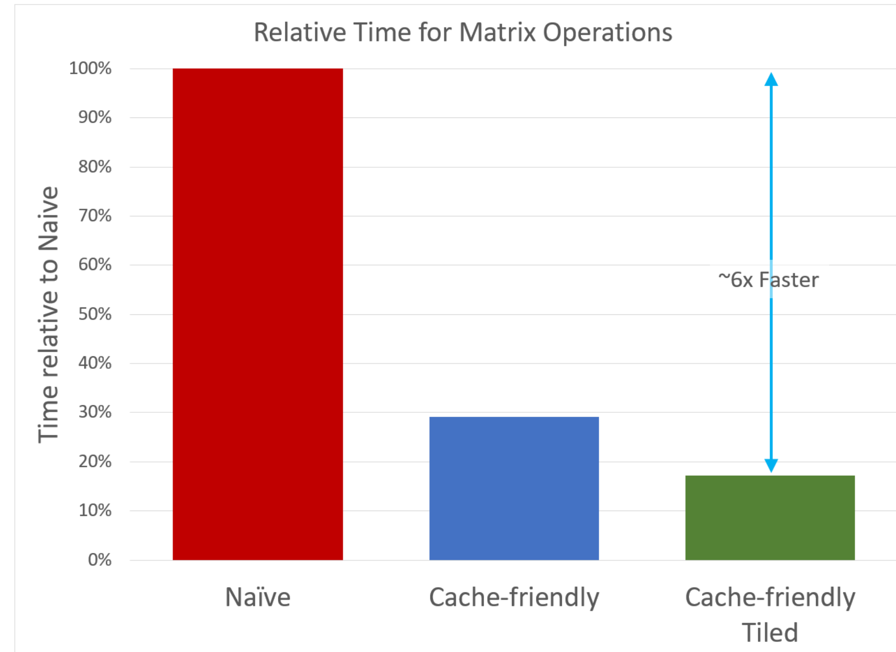




Cache Considerations

PDE performance is strongly affected by how fast the CPU can access data; A 400× 400 PDE matrix is ~ 1.2Mb. It will not fit into the L1 or L2 cache. Accessing the L3 cache is $10\times$ slower than accessing the L1 cache. 2D-PDEs are also affected by the strided traversal needed for solving $(\mathcal{I} + \theta\mathcal{L}_x).Y^{(2)} = Y^{(1)}$.

The nZetta Toolkit is designed to use cache-friendly data structures for optimal performance.

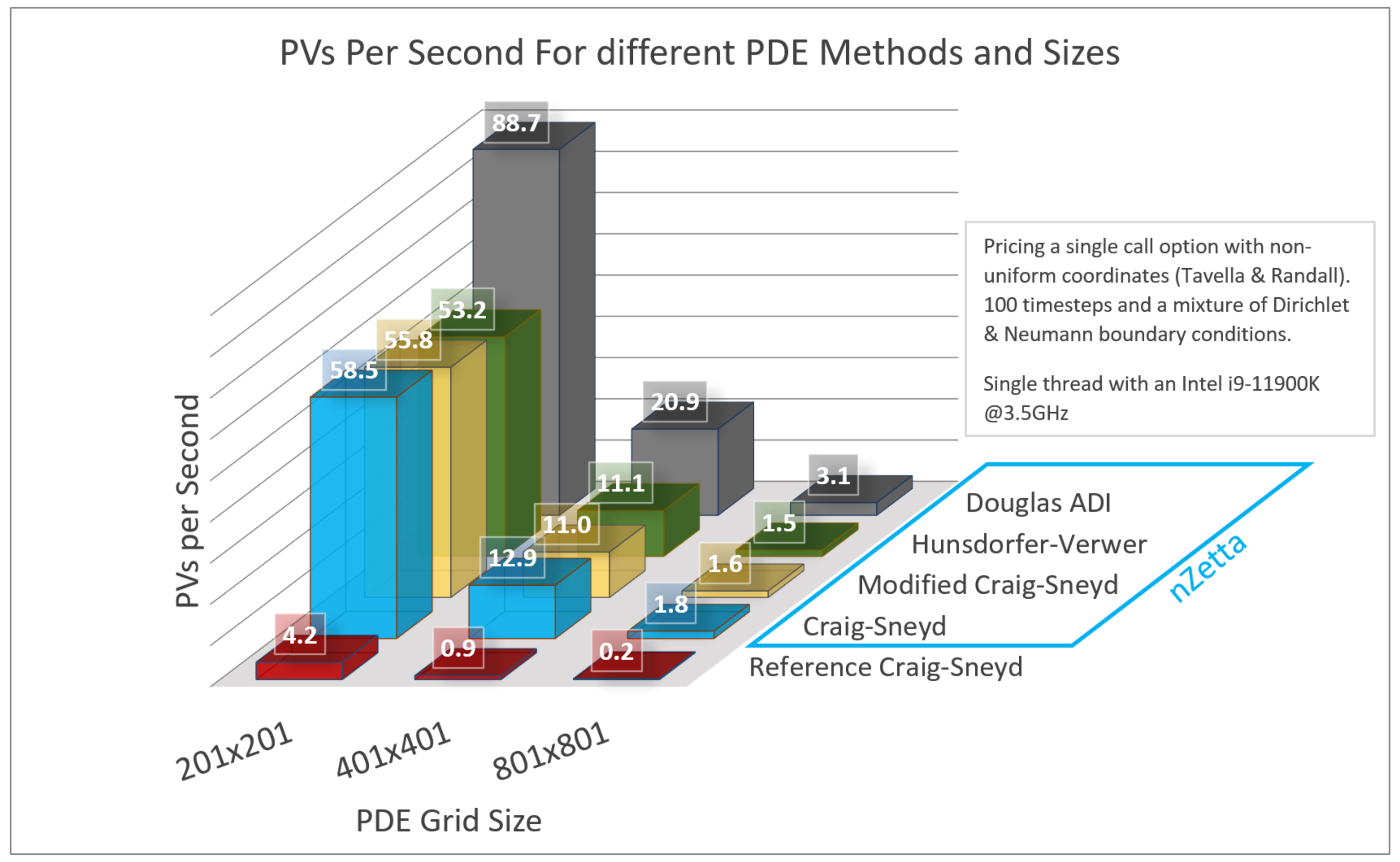


nZetta Toolkit 2D-PDE

The nZetta Toolkit 2D-PDE is a C-API drop-in replacement supporting Douglas-ADI, Craig-Sneyd, Modified Craig-Sneyd, and Hunsdorfer-Verwer methods along with necessary anciliary functions.

Example: Heston Local Stochastic Volatility Model

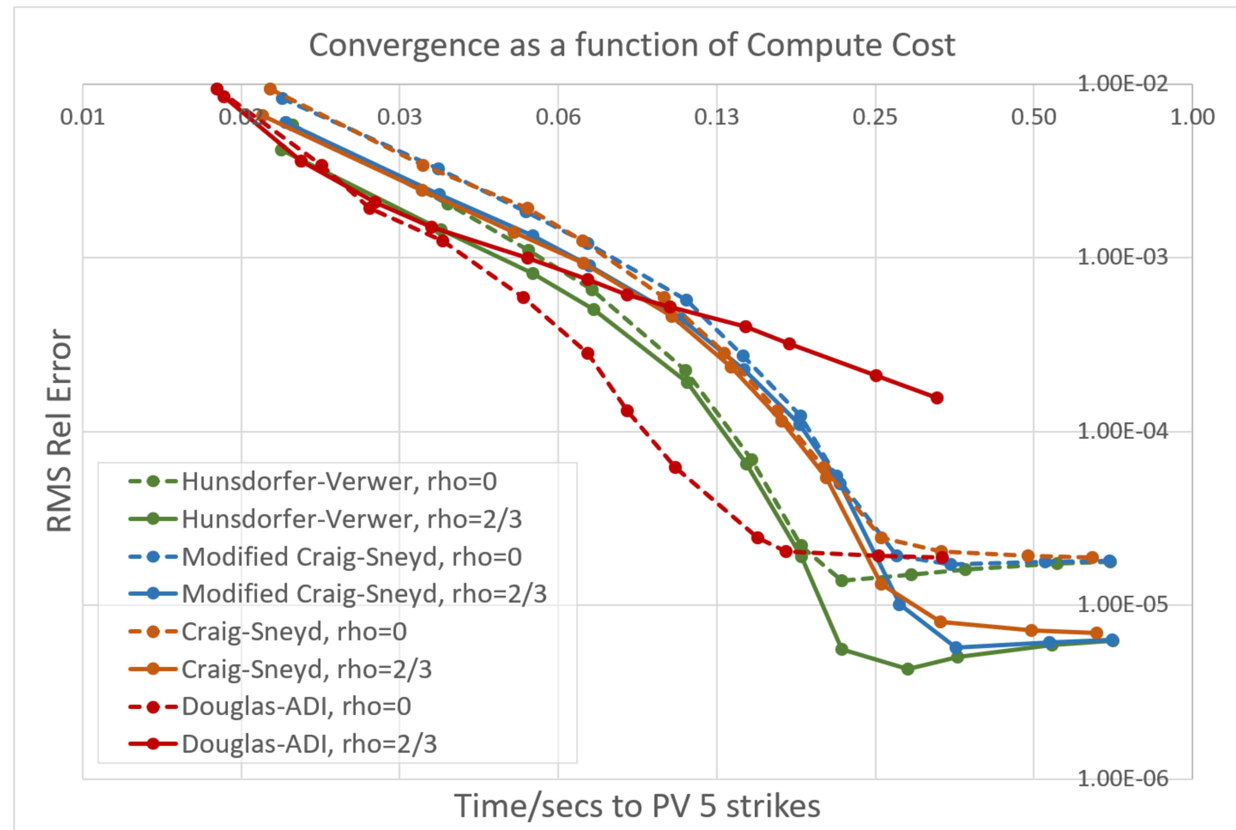
To illustrate performance, we consider European option pricing using the Heston Local Stochastic Volatility Model using 100 timesteps, no initial Rannacher stepping, mixed Dirichlet and Neumann boundary conditions, and abcissæ using an extended Tavella & Randall form.



$$\begin{aligned} dS_t &= \mu_t S_t dt + L(S_t, t)\sqrt{\nu_t} S_t dW_t^{(1)} \\ d\nu_t &= \kappa(\theta - \nu_t)dt + \eta\sigma\sqrt{\nu_t} dW_t^{(2)} \\ \rho dt &= E[dW_t^{(1)} dW_t^{(2)}] \end{aligned}$$

The nZetta Toolkit 2D-PDE is **over 10 times faster** than the reference implementation. Choosing the best method and abcissæ for your problem can give a further 50% speed-up for a given accuracy goal.

The nZetta Toolkit provides the performance you need for your most computationally costly and time-critical calculations in a simple C-API for the fastest integration into your existing codebase.



support@nag.co.uk   *www.nag.com*