Fast Implied Volatilities using Chebyshev Interpolation

Edvin Hopkins (NAG) *Kathrin Glau (QMUL)* *Linus Wunderlich (QMUL)*

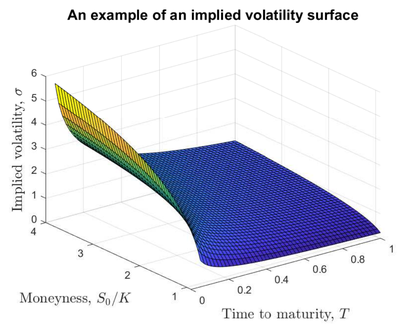
Background

The Black–Scholes formula for the price of a call option is

$$C = S_0\Phi\left(\frac{\ln\left(\frac{S_0}{K}\right) + \left[r + \frac{\sigma^2}{2}\right]T}{\sigma\sqrt{T}}\right) - Ke^{-rT}\Phi\left(\frac{\ln\left(\frac{S_0}{K}\right) + \left[r - \frac{\sigma^2}{2}\right]T}{\sigma\sqrt{T}}\right),$$

where T is the time to maturity, S_0 is the spot price of the underlying asset, K is the strike price, r is the interest rate and σ is the volatility.

An important problem in finance is to compute the ***implied volatility***, σ, given values for T, K, S_0, r and C. Typically volatilities are computed for large vectors of input data. An explicit formula for σ is not available, so numerical approximation is required.



As the above figure illustrates, the volatility surface can be highly curved. This makes approximating σ difficult.

In practice, most methods reduce the dimensionality of the problem by substituting

$$x = rT + \ln(S_0/K), \quad c = C/\sqrt{S_0e^{-rT}K}, \quad v = \sigma\sqrt{T},$$

so that

$$c(x, v) = e^{\frac{x}{2}}\Phi\left(\frac{x}{v} + \frac{v}{2}\right) + e^{-\frac{x}{2}}\Phi\left(\frac{x}{v} - \frac{v}{2}\right).$$

The algorithm of Jäckel (2015)

A modified Newton iteration is used to compute $v(c, x)$. The input domain is decomposed into four areas. Rational approximations are used to provide initial guesses and reduce the number of iterations.

The method is accurate to almost machine precision. However, branching prevents vectorization and impedes performance.

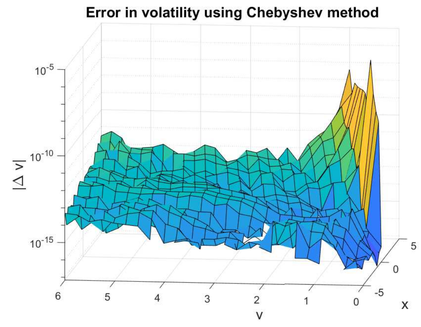
Can we improve performance without losing accuracy?

Computing volatilities using Chebyshev interpolation

Under suitable conditions ***the error in Chebyshev interpolation decays exponentially*** with the number of nodes (see e.g. Trefethen (2013)).

Glau *et. al.* (2018) exploit this by using a bivariate Chebyshev interpolation of $v(c, x)$, resulting in a vectorizable algorithm.

1. ***Offline phase:*** polynomial weights of a low-rank Chebyshev interpolation of the implied volatility surface are computed and stored for four different input domains, using the algorithm of Jäckel (2015). This step is only performed once, during code development.
2. ***Online phase:*** the input data is split into the four domains and the Chebyshev interpolation is applied to each domain, choosing precomputed nodes from Step 1 according to the desired accuracy.

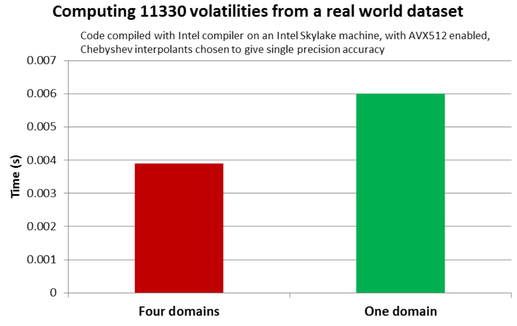


The error surface above shows that, for all but the most extreme input values, accuracy close to machine precision is attained.

Does the domain decomposition impede vectorization?

A single-domain version was developed. Matlab® prototypes suggest that this should perform better due to increased vectorization.

The graph below compares our optimized implementations.



In our production code, the single-domain version performs worse than the four-domain version. Why?

Performance analysis

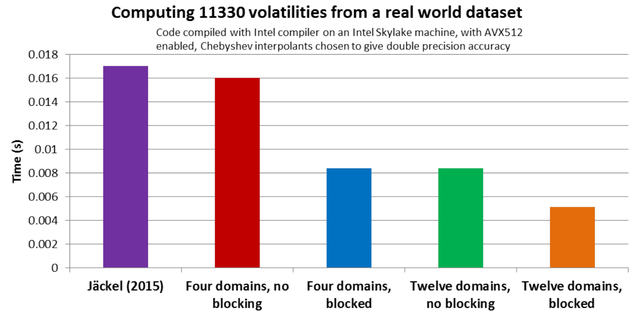
Profiling of the production code shows that:

- the domain decomposition and rearrangement of data only account for $\sim 3\%$ of runtime,
- the Chebyshev interpolation accounts for the remainder,
- the single-domain version's large domain size means more Chebyshev nodes are required to achieve a given accuracy, hence the longer runtime.

This demonstrates how important it is to ***always profile your code!***

Performance improvements were now sought by increasing the number of domains in the decomposition, and using a blocking scheme during the interpolation phase to improve cache use.

Performance results



The blocking scheme and the increased number of domains combine to give a $\sim 3.3\times$ ***speedup*** over Jäckel (2015).

Next steps and further improvements

There is scope for further performance improvements.

- What is the optimum number of domains?
- Can the blocking strategy be tuned further to decrease runtime?

To try a pre-release version of our code, contact NAG.

References

L. N. Trefethen (2013). *Approximation Theory and Approximation Practice*. SIAM books.

P. Jäckel (2015). Let's be rational. *Wilmott 2015*, 40-53.

K. Glau, P. Herold, D. B. Madan, C. Pötz (2018). The Chebyshev method for the implied volatility. Accepted for publication in the Journal of Computational Finance, preprint of former version available on arXiv:1710.01797