Fast Implied Volatilities using Chebyshev Interpolation

Edwin Hopkins (NAG)  Kathrin Glau (QMUL)  Linus Wunderlich (QMUL)

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

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

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 \( \sigma \) is the volatility.

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

Neutral volatilities

The graph below compares our optimized implementations.

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)).

It has been proposed that this should perform better due to increased vectorization.

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

