Financial Applications of HPC: Comparative Analysis of Asian/European Option Pricing and Sensitivity

Aidan Coleman (colemaa3@tcd.ie)

April 2025

1 Calculating the Δ and $\mathcal{V}$ of an Asian Option

We use a straightforward "bump and revalue" approach to calculate the $\Delta = \frac{\partial C}{\partial S}$ (C is the call option price, S is the underlying initial asset price to which the Asian option pertains). We slightly increase and decrease the underlying asset's initial spot price, calculate the prices in either case, get their difference and divide this difference by the difference between initial spot prices (this is a central differencing approach rather than forward difference as the error term is only of order $\mathcal{O}(h^2)$, and we choose h to be sufficiently small, which is smaller than using forward differencing which has an error term that is linear in h):

$$\Delta \approx \frac{\pi(S_0 + h, ...) - \pi(S_0 - h, ...)}{2h}$$

Where $\pi((S_0, ...)$ is the discounted net present value of the option payoff (option price) given parameters like initial spot price S_0 and other parameters ... like annual volatility σ etc. To reduce noise, both up- and down-valuations share the same RNG seed. The formula for $\mathcal{V}$ is given by:

$$\mathcal{V} \approx \frac{\pi(S_0, \sigma + h, ...) - \pi(S_0, \sigma - h, ...)}{2h}$$

2 Asian Option Monte Carlo Pricer

We simulate n independent asset paths, each with daily steps over d trading days. Let

$$\Delta t = \frac{1}{255}, \quad T = d\,\Delta t,$$

and $S_{t+\Delta t} = S_t \exp\left[(r - \frac{1}{2}\sigma^2)\Delta t + \sigma\sqrt{\Delta t}\, Z\right]$, where $Z \sim N(0,1)$. On each path we record the arithmetic average

$$A = \frac{1}{d}\sum_{k=0}^{d-1} S_{k\Delta t}$$

and payoff $\max(A - K, 0)$. The Monte Carlo estimator is discounted to today:

$$\hat{C}_{\text{Asian}} = e^{-rT}\,\frac{1}{n}\sum_{i=1}^{n} \max(A^{(i)} - K, 0).$$

3 European Call via Black–Scholes

The standard Black–Scholes formulas for a European call are

$$C_{\text{BS}} = S_0\,\Phi(d_1) - Ke^{-rT}\,\Phi(d_2), \quad \Delta_{\text{BS}} = \Phi(d_1),$$

where

$$d_{1,2} = \frac{\ln(S_0/K) + (r \pm \frac{1}{2}\sigma^2)\,T}{\sigma\sqrt{T}},$$

and Φ is the standard normal CDF.

4 Side-by-Side Comparison

After implementing both methods in a single C++ class, we run with

$$S_0 = 100,\ K = 103,\ \sigma = 0.10,\ r = 0.01,\ d = 255,\ n = 10^6.$$

Outputs:

```
    Asian Option (MC) Results:
  NPV:   1.285067
  Delta: 0.341856
  Vega:  21.229058
European Delta: 0.442123 (European is +0.100267 relative)
European Option Price: 3.116293
```

The path-averaging inherent in the Asian reduces both price and Delta relative to the plain-vanilla European call.

5 C++ Compilation

I compiled on my macOS with:

```
    g++ asian_option_mc_pricer.cc -o asian_option_mc_pricer \
>     -Wall -Wextra \
>     -lgsl -lgslcblas -lm
```

Ensure you pass GSL's libraries to the linker like the above compilation code does. Run the program without passing any command line arguments and edit source file `asian_option_mc_pricer.cc` if you wish to play around with parameters.