A Numerical Exploration of the Local Volatility Model for Option Pricing

Francisco Rivera Kevin Chen

December 17, 2017

Abstract

In Nobel-prize winning work, Black, Merton, and Scholes developed a model to price options. The tractability of the model revolutionized options pricing and allowed for the derivatives to become widely traded. However, the predictions of the model run counterfactual to empirically observed prices. In this paper, we consider a generalization to the Black-Scholes model, the local-volatility model. This generalization gives us more degrees of freedom to fit the prices we observe. With the additional expressiveness, however, come numerical complications. Black-Scholes requires fitting only a single number known as the *implied volatility*, but the local volatility model requires fitting an entire multi-dimensional function. We explore these complications and arrive at solutions that are assessed for numerical stability and accuracy to price out-of-sample options.

Contents

1 Background

1.1 Options Terminology

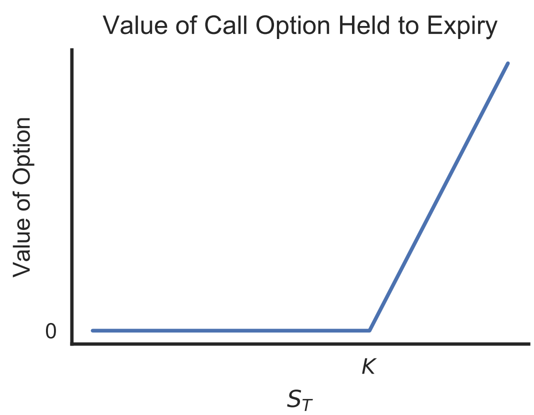
Before delving into the theoretical and numerical results, we briefly summarize options terminology. An option is a *derivative* on some asset, henceforth called the *underlying*—i.e. it's value is *derived* from the value of the underlying. The owner of a call (resp. put) option has the right but not the obligation to buy (resp. sell) the underlying asset at a given price at some date in the future.

The price at which the holder of the option can buy/sell is called the *strike price*, denoted K. Invoking the right to buy or sell is called *exercising* the option. The last time at which the holder can exercise is called the *expiry*, denoted as time T. The value of the underlying asset at time t is denoted as S_t (and sometimes the subscript is dropped whenever it is implied).

The payoff of a call option at expiry is thus given by,

$$\max(S_T - K, 0) \tag{1.1}$$

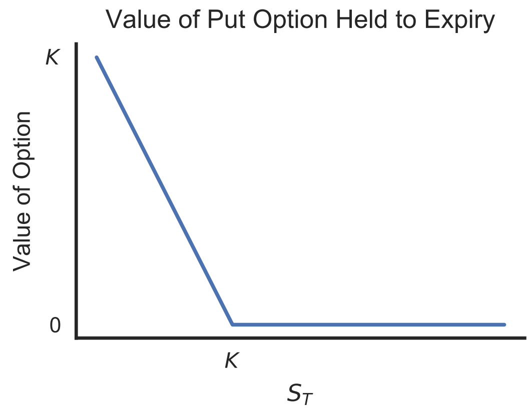
because the owner will only exercise it if the underlying is worth more than the strike price. This function gives rise the characteristic "hockey-stick" payoff diagram of an option at expiry,



The payoff of a put option is very similar to equation 1.1,

$$\max(K - S_T, 0) \tag{1.2}$$

because the owner of the option will only exercise it if the underlying is worth less than the strike price, and the payoff curve looks like its mirror image,



1.2 Risk-Neutral Pricing

The value of an option at expiry is easily observable and displayed in the figures in Section 1.1. However, before time T, the value of S_T is a random variable. Asset pricing (change of measure) theorems allow us to write the value of a call option as the discounted expectation of the payoff under what we call the *risk-neutral distribution*,

$$\frac{C(S_0, K, T)}{e^{-rT}} = \int_K^\infty (S - K) \underbrace{\phi(S, T; S_0)}_{\text{risk-neutral PDF}} dS \tag{1.3}$$

However, in order to make any progress beyond this, we need to make a further assumption about what this risk-neutral distribution looks like.

1.3 Black-Scholes Model

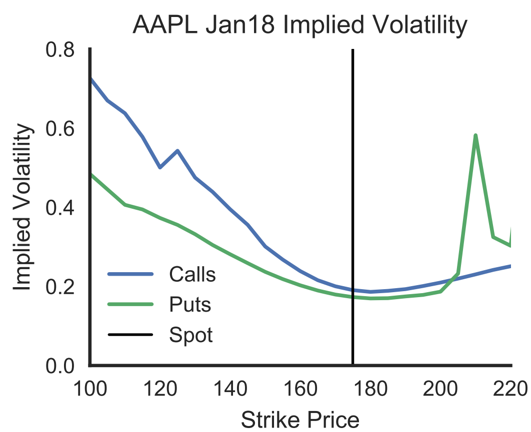
The Black-Scholes model makes one such assumption by writing how the asset diffuses over time. In particular, the Black-Scholes model treates the asset price as a geometric Brownian motion represented by the following stochastic difference equation:

$$\frac{dS}{S} = rdt + \sigma dW \tag{1.4}$$

where W is Brownian motion and σ is a constant value called the *implied volatility*.

This forces the risk-neutral distribution to be log-normal and makes finding the price analytically tractable. In particular, since the only pricing input into this model that we do not observe is the implied volatility, we can quote price as a function of implied volatility (e.g. an option is said to cost $\sigma = 16\%$ if its price is consistent with the price the Black-Scholes model would predict if $\sigma = 16\%$).

Thus, Black-Scholes predicts that if σ is indeed a constant, then for any option written on the same underlying (regardless of strike price or expiry), that the implied volatility will be (approximately) constant. However, this is directly counterfactual to the options prices that we observe. For example, looking at the prices of AAPL options that expire on January 2018 retrieved on November 20$^{\text{th}}$, 2017 from Yahoo Finance, we get the following curve



While the data is noisy, particularly for deep-in-the money[1] puts which are hardly ever traded, the graph is unmistakably not constant. Moreover, AAPL is not an exception: most graphs of implied volatility versus strike have a similar shape. These deviations from Black-Scholes also correspond to intuitive qualitative ideas. If the price of AAPL has just plunged 50%, it is palatable to think there is a lot of investor uncertainty, and that the future price of AAPL will diffuse with higher volatility than if it is just up 5%. Thus, the assumption of constant σ is suspect.

2 Local Volatility Theory

The *local-volatility model* directly responds this shortcoming of the Black-Scholes model by rewriting the diffusion equation as,

$$\frac{dS}{S} = rdt + \sigma(S,t)dW \tag{2.1}$$

such that the volatility is no longer a global constant, but rather a function of spot-price and time. We refer to this function as the *local volatility function*, and the value of the function at any given point as the *local volatility*. Note of course that in the special case when σ is a constant function, we have simply recovered the Black-Scholes model. We will make use of this fact when we need to confirm the numerical results of the local volatility model against theoretical closed-forms.

[1]An option is said to be *in-the-money* when its payoff is positive if $S_T = S_0$, *out-of-the-money* when its payoff is negative if $S_T = S_0$, and *at-the-money* otherwise.

In the general case, though, we care about what the local volatility surface says about prices and vice-versa. To this end, we can invoke Dupire's[2] work. The first thing to note is that we can differentiate our risk-neutral pricing formula with respect to K twice to get that,

$$e^{-rT}\phi(K,T;S_0) = \frac{\partial^2 C}{\partial K^2}. \tag{2.2}$$

Moreover, because we have a diffusion rule for the underlying S_t, the probability distributions must satisfy the *Fokker-Planck equation*. (We display this without drift. In practice, we can do this for the underlying itself if is driftless, but if we are worried about drift, we can simply use the forward price which is by construction a martingale),

$$\frac{\partial C}{\partial T} = \frac{1}{2}e^{rT}\sigma^2 K^2 \frac{\partial^2 C}{\partial K^2} \tag{2.3}$$

Rearranging gives us Dupire's equation,

$$\sigma^2(K,T,S_0) = \frac{\frac{\partial C}{\partial T}e^{-rT}}{\frac{1}{2}K^2\frac{\partial^2 C}{\partial K^2}}. \tag{2.4}$$

We can also rewrite this assuming 0 interest rates, which we will assume going forward for simplicity,

$$\sigma^2(K,T,S_0) = \frac{\frac{\partial C}{\partial T}}{\frac{1}{2}K^2\frac{\partial^2 C}{\partial K^2}}. \tag{2.5}$$

There are a couple notable takeaways from this equation. First of all, if we fully and perfectly observed a continuum of option prices for all strikes and expiries, we would be able to uniquely determine the local volatility surface. Moreover, we can do this *regardless* of what the option prices are[3]. This means that unlike Black-Scholes, the local volatility model can perfectly fit what we see, and we should be apprehensive of overfitting.

3 Pricing with Local Volatility

In this section, we will concern ourselves with what we can do once we have the local volatility surface. In particular, our main objective will be to price options. Note that the prices may exist in the market (e.g. as a sanity check for our model), or they may not (e.g. to price options with expiries that are not quoted, giving our model predictive power).

3.1 Monte-Carlo Methods

One way to do this is by realizing that the price is an expectation under the risk-neutral distribution of the pay-off random variable (as in Equation 1.3). Thus, if we can sample

[2] TODO: citation needed

[3] We get negative local volatility if arbitrage conditions are violated, but we assume this does not happen.

from this distribution, we can invoke the law of large numbers and get an estimate for its average with sufficient samples. To this end, first we build a function that allows us to sample from the risk-neutral distribution of ending spot prices in Listing 1.

In this implementation, we have two types of numerical error to reason about. The first is simply sampling error. Because we are only taking a finitely many number of draws from the distribution, our sample mean will differ from the true mean by a quantity that decreases with $\sqrt{n}$. Harder to reason about is the discretization error that we introduce.

3.2 Discretization errors in Monte Carlo pricing

In Monte Carlo pricing, we draw from the distribution of S_T and compute option prices. We start with the stochastic difference equation[4]

$$dS_t = S_t \sigma(S_t, t)\, dW_t,$$

and approximate with finite differences

$$\tilde{S}_{t_{k+1}} - \tilde{S}_{t_k} = \tilde{S}_{t_k}\sigma(\tilde{S}_{t_k}, t_k)\,(W_{t_{k+1}} - W_{t_k}) = \tilde{S}_{t_k}\sigma(\tilde{S}_{t_k}, t_k)\sqrt{t_{k+1} - t_k} Z_k, \tag{3.1}$$

for some $Z_k = \frac{W_{t_{k+1}} - W_{t_k}}{\sqrt{t_{k+1} - t_k}} \sim \mathcal{N}(0, 1)$. This is the well-known *Euler-Maruyama method*, an extension of the forward Euler method in stochastic calculus. It has been shown that the recursively-computed sequence $\tilde{S}_{t_1}, \ldots, \tilde{S}_{t_N}$ converges to a draw from the true stochastic process S_t as the mesh of the partition $\{t_1, \ldots, t_N\}$ tends to zero (**?**).

Moreover, we can bound the rate of convergence. Let $\Delta t = t_{k+1} - t_k$ and suppose that S_{t_k} is a draw from the true distribution. Let $S_{t_{k+1}}$ be drawn from the true distribution conditional on S_{t_k}, and let $\tilde{S}_{t_{k+1}}$ be the corresponding finite difference approximation via (3.1) with respect to the same draw from the Brownian motion W_t. We have the following bounds on convergence (**?**):

$$E\left|S_{t_{k+1}} - \tilde{S}_{t_{k+1}}\right| \le C_1(\Delta t)^{1/2} \quad \text{(Strong convergence)}$$

$$\left|E(g(\tilde{S}_{t_{k+1}})) - E(g(S_{t_{k+1}}))\right| \le C_2 \Delta t \quad \text{(Weak convergence)}$$

for a g that satisfies certain regularity conditions. Applying an equally-spaced partition to $[0, T]$ with step-size Δt, we immediately observe that the draws from the distribution at expiry must obey the same bounds of convergence:

$$E\left|S_T - \tilde{S}_T\right| = O\left((\Delta t)^{1/2}\right)$$

$$\left|E(g(\tilde{S}_T)) - E(g(S_T))\right| = O(\Delta t).$$

Since we apply Euler-Murayama method to option pricing, we are more interested in the error $\left|E(g(\tilde{S}_T)) - E(g(S_T))\right|$, where g is a function of the form $g(S_T) = \max(S_T - K, 0)$

[4] We assume interest rates are zero, and so there is no drift term for simplicity

(for call options). The nondifferentiability of g may pose some concern, as **?** notes that the weak convergence bound works for g smooth. However, since there exists a sequence of smooth functions $\tilde{g}$ that uniformly converges to $g(S_T) = \max(S_T - K, 0)$, we have

$$\left|E(g(\tilde{S}_T)) - E(g(S_T))\right| \leq 2||g - \tilde{g}||_\infty + \left|E(\tilde{g}(\tilde{S}_T)) - E(\tilde{g}(S_T))\right|,$$

where $||g - \tilde{g}||_\infty$ can be arbitrarily small.

4 Fitting Local Volatility

5 Theoretical analysis of numerical methods

Our approach involves two separate applications of numerical techniques. First, we observe existing option prices $C(K, T)$ and compute the local volatility function via Dupire's equation

$$\sigma^2 = \frac{\frac{\partial C}{\partial T}}{\frac{1}{2}K^2\frac{\partial^2 C}{\partial K^2}}.$$

Second, given an estimate of the local volatility σ^2, we numerically predict option prices. In this section, we discuss some theoretical difficulties in deriving analytical bounds on our approximations. In particular, we proceed in reverse order. We first discuss the numerical errors of Monte Carlo pricing, assuming the local volatility σ^2 is known.

Numerical experiment of first-order convergence of discretization error Let $S_0 = 1, K = 1.1$, and

$$\sigma(S, t) = \min\left(0.1 + (S-1)^2, 0.5\right).$$

Consider a call option at expiry $T = 1$ with strike K, whose payoff is $\max(S_T, K) - K$. Assume zero interest rate. We approximate the expectation with sample mean with sample size 10^6. We approximate the true value of the option by computing the Euler-Murayama approximation with step-size equalling $1/200$. We then plot the pricing errors of Euler-Murayama approximations with step size $1/n$ for $n = 1, \ldots, 40$ in Figure 1. It is difficult to control the Monte Carlo sampling error, due to computational resource constraints, but we do observe approximately a first-order convergence pattern.

Thus, for a known σ^2, we can limit the size of the discretization error at rate $O(\Delta t)$, and the size of the sampling error at rate $O(n^{-1/2})$ where n is the sample size. The theory and the numerical experimentation suggests that the Monte Carlo pricing method is numerically robust.

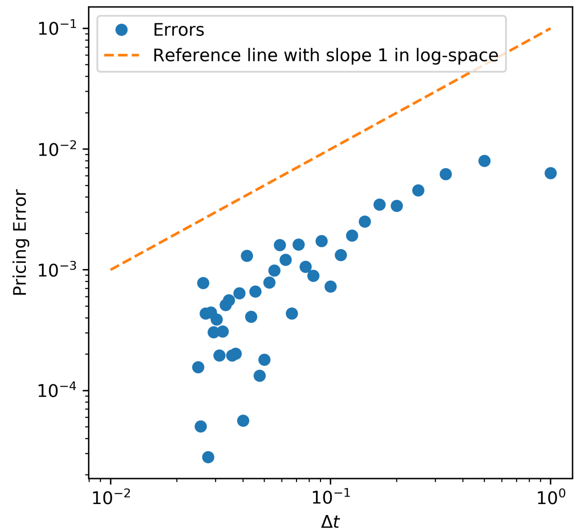


Figure 1: Log-log plot of pricing error against Δt

5.1 Approximation of Dupire's equation

However, computing σ^2 from observed data accurately is a much more difficult task. We illustrate the main difficulty below.

Let $A = \frac{\partial C}{\partial T} + e_1$ and $B = \frac{\partial^2 C}{\partial K^2} + e_2$ be two approximations. Then

$$\frac{A}{\frac{1}{2}K^2 B} = \sigma^2 \frac{\frac{\partial^2 C}{\partial K^2}}{\frac{\partial^2 C}{\partial K^2} + e_2} + \frac{e_1}{\frac{1}{2}K^2 B}.$$

Thus the absolute error

$$\left| \frac{A}{\frac{1}{2}K^2 B} - \sigma^2 \right| \leq \sigma^2 \left| \frac{e_2}{\frac{\partial^2 C}{\partial K^2} + e_2} \right| + \left| \frac{e_1}{\frac{1}{2}K^2 B} \right|.$$

Since $\frac{\partial^2 C}{\partial K^2} > 0$, it should be unsurprising that the error vanishes as $e_1, e_2 \to 0$. However, note that deep-in-the-money options are virtually indistinguishable from the underlying asset, whereas deep-out-of-the-money options are virtually worthless. Thus C is almost linear as a function of K when K is away from S_0. Even though $\frac{\partial^2 C}{\partial K^2} > 0$, the infimum of this second derivative is zero. This presents a first difficulty when attempting to bound the error of the approximations. A second challenge arises from the discreteness of empirical prices. We cannot evaluate the function $C(K, T)$ anywhere we wish, but rather we only observe its values on a grid whose fineness is capped at the finest intervals that prices are quoted in. To make matters worse, the prices quoted are accurate to \$0.01, and so it is as if we are working in a world where machine precision is nontrivially large.

The usual centered difference approximations have

$$|e_1| \leq \frac{h_1^2}{6} \sup \left| \frac{\partial^3 C}{\partial T^3} \right| + \frac{\epsilon}{2h_1} \qquad |e_2| \leq \frac{h_2^2}{12} \sup \left| \frac{\partial^4 C}{\partial K^4} \right| + \frac{2\epsilon}{h_2^2}.$$

For values of $\frac{\partial^2 C}{\partial K^2}$ sufficiently large, it is plausible that these approximations are sufficiently accurate. However, the approximation is certainly not accurate for values of $\frac{\partial^2 C}{\partial K^2}$ close to zero. Yet for these values, which correspond to deep-money options, the option price is virtually known—the large errors in σ^2 for K far away from S_0 is weighted by the extremely small probability that such errors matter. Such weighting is analytically intractable, and we mainly focus on numerical experimentation.

6 Conclusions

Appendices

A Code

Listing 1: Sampling from Risk-Neutral S_T Distribution

```
def sample_end_price(S0, local_vol_f, duration, n_intervals, n_samples):
    """
    Inputs
    ------
    S0 : float
        The initial spot price of the asset at time t=0

    local_vol_f : float -> float (vectorized)
        The local volatility at a given spot price (assumed constant over time)

    duration : float
        The time to expiry, i.e. T.

    n_intervals : float
        Number of intervals into which to break up the numerical simulation

    n_samples : int
        Number of simulations to run

    Output
    -----
    S : NumPy float vector of length n_samples
        The ending spot prices of the asset diffusion for each simulation
    """
    scaling_factor = duration / np.sqrt(n_intervals)
    S = np.zeros((n_samples, n_intervals+1))
    S[:,0] = S0
    for i in range(1,n_intervals+1):
        local_vols = local_vol_f(S[:,i-1])
        growth_factor = np.exp(local_vols * np.random.randn(n_samples) * scaling_factor)
        S[:,i] = S[:,i-1] * growth_factor
    return S[:,-1]
```