1 Option Calibration

The problem of calibrating option prices to market values (the "inverse problem") is non-trivial especially with complex pricing models with many parameters. A naive approach is to perform optimization by minimizing a distance between the prices provided by the market and the modeled prices by varying the input parameters. However, this can be computationally intensive. The problem is not convex and there may be a plethora of local minima. The parameter surface may have many "flat" areas leading to unstable parameter solutions.

In our study, we focus on calibrating models defined in the Option Calculation paper. We use a Heston model, a jump-diffusion a la Merton (1976), and a CGMY model.

The code which runs the results shown below is available at the following Github repo: Option Price Functions.

2 Calibration

Calibration has traditionally taken the following form:

$$\min_{\theta} \sum_{k} w_k \left(C_k - C(k;\theta)\right)^2$$

Where w_k is a weight, θ are the parameters describing the (risk-neutral) asset process, C_k is the observed option prices at log-strike k, and $C(k,\theta)$ is the modeled price. Following Chen Bin, we choose $w_k = w = \frac{50}{\sum_i C_i^2)}$.

As mentioned in the introduction, this optimization problem is not trivial. See Cont and Tankov (2006) for details. We use the Nelder-Mead algorithm and random starting points for the parameters to find a global minimum. This works well in practice, as the following optimization attests. Code to generate these tables is here

2.1 Results

2.1.1 Heston

Table 1: Heston Comparison

	η	speed	σ	ρ
Actual	0.5751	1.5768	0.1995	-0.5711
Estimate	0.5701	1.6157	0.1917	-0.5697

Table 2: Merton Comparison

	λ	μ_l	σ_l	σ
Actual	1	-0.025	0.2236	0.2236
Estimate	1	-0.025	0.2236	0.2236

2.1.2 Merton

2.1.3 CGMY

Table 3: CGMY Comparison

	c	g	m	y
Actual	1	5	5	1.5
Estimate	1.0457	5.0704	5.0706	1.4894