Option Pricing, Hedging, and Risk Analysis Under Rough Volatility Models

Cameron Sahirad

Mathematics in Finance, Courant Institute of Mathematical Sciences, NYU
cbs476@nyu.edu

May 12, 2021

Problem Statement

- Recent research has shown that log-volatility is well modelled using a fractional Brownian motion process and has led to the development of so called rough volatility models.
- The purpose of this project is to develop a suite of tools that will enable the analysis of rough volatility models, in particular the rough Bergomi model.

The Rough Bergomi Model

- Build a simulation framework in Python with the ability to generate sample paths of stock price and volatility for a given forward variance curve

$$dS_t = \sqrt{v_t} S_t dB_t$$

$$dB_t = \rho W_t^1 + \sqrt{1-\rho^2} W_t^2$$

$$W_t^\alpha = \sqrt{2\alpha+1} \int_0^t (t-u)^\alpha dW_u^1$$

$$v_t = \xi_0(t) \exp\left(\eta W_t^\alpha - \frac{\eta^2}{2} t^{2\alpha+1}\right)$$

$$\xi_0(t) = \mathbb{E}[v_t | \mathscr{F}_0]$$

Exotic Payoffs

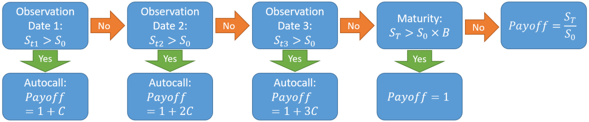
- Code payoffs for vanilla and exotic options including cliquet, barrier, asian, and autocallable

$$\text{n-Period cliquet option payoff} = \max\left[\sum_{i=1}^{n} \min\left(\text{cap}, \frac{S_{i/n}}{S_{(i-1)/n}} - 1\right), 0\right]$$

$$\text{Up-and-out call payoff} = \begin{cases} \max(S_T - K, 0) & S_t < H \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Asian call payoff} = \max\left(\frac{1}{n}\sum_{i=1}^{n} S_{t_i} - K, 0\right)$$

Autocallable payoff:



Model Calibration and Risk Analysis

- Train a neural network to represent the pricing function that can be used for calibration

$$\hat{\theta} = \underset{\theta \in \Theta}{\arg\min}\, \delta\left(\tilde{P}(\mathscr{M}(\theta), \zeta), \mathscr{P}^{MKT}(\zeta)\right)$$

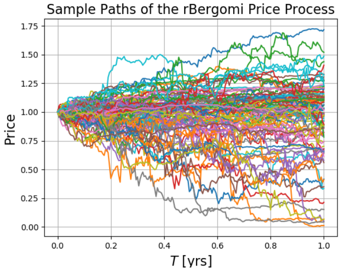
- Project the risk of implied volatility dynamics on the vega risk profile of each product through a static hedge of vanilla instruments. Where E is a vector of exotics under different perturbations to the vol surface, H is the matrix of hedge instruments, and w is a vector of weights

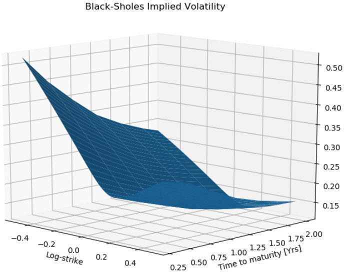
$$E = Hw$$

$$w = \underset{w}{\arg\min} \left\{ ||E - Hw||^2 + \lambda ||w||^2 \right\}$$

Monte Carlo Simulation

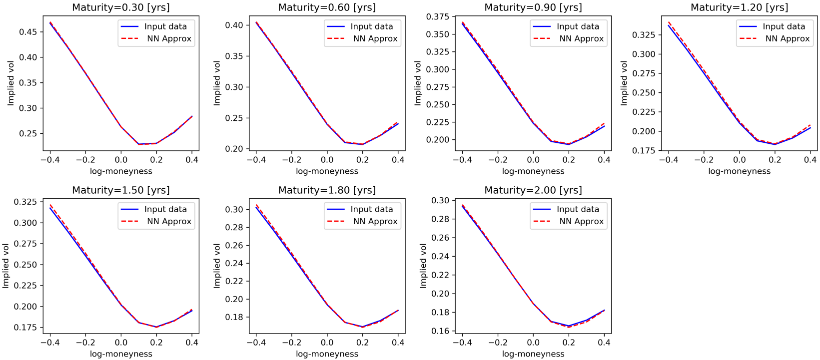
- rBergomi Python framework is able to generate sample paths of the stock and volatility process and execute Monte Carlo simulation to price vanilla and exotic options
- Efficient hybrid scheme of convolution and Riemann sums implemented to generated fBM



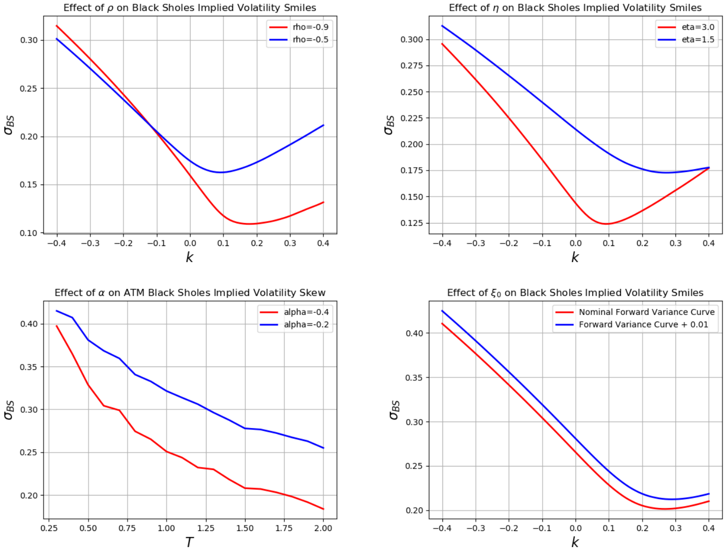


Calibration

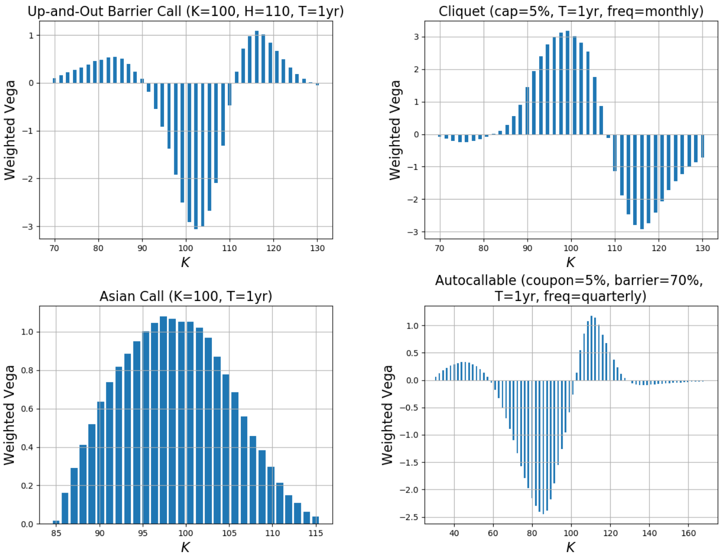
- A large set of randomly selected model parameters is then used to generate many volatility surfaces in order to produce the training data for the neural network


Maturity=0.30 [yrs]
Maturity=0.60 [yrs]
Maturity=0.90 [yrs]
Maturity=1.20 [yrs]
Maturity=1.50 [yrs]
Maturity=1.80 [yrs]
Maturity=2.00 [yrs]
Input data
NN Approx
Implied vol
log-moneyness

Sensitivity Analysis



Projected Risk of Implied Volatility Dynamics on Vega Profile



Thank you

&

Questions