Calibration of Heston Local Volatility Models

J. Göttker-Schnetmann, DZ BANK
K. Spanderen, E.ON Global Commodities SE

QuantLib User Meeting 2015
Düsseldorf
2015-11-30

- Model Overview
- Fokker-Planck Calibration
- Feyman-Kac Calibration
- Case Studies
- Summary

Model and Stochastic Differential Equations

- Add leverage function $L(S_t, t)$ and mixing factor η to the Heston Model:

$$\begin{aligned} d\ln S_t &= \left(r_t - q_t - \frac{1}{2}L(S_t,t)^2\nu_t\right)dt + L(S_t,t)\sqrt{\nu_t}dW_t^S \\ d\nu_t &= \kappa\left(\theta - \nu_t\right)dt + \eta\sigma\sqrt{\nu_t}dW_t^\nu \\ \rho dt &= dW_t^\nu dW_t^S \end{aligned}$$

- Leverage $L(x_t, t)$ is given by probability density $p(S_t, \nu, t)$ and

$$L(S_t,t) = \frac{\sigma_{LV}(S_t,t)}{\sqrt{\mathbb{E}[\nu_t | S = S_t]}} = \sigma_{LV}(S_t,t)\sqrt{\frac{\int_{\mathbb{R}^+} p(S_t,\nu,t)d\nu}{\int_{\mathbb{R}^+} \nu p(S_t,\nu,t)d\nu}}$$

- Mixing factor η tunes between stochastic and local volatility

Cheat Sheet: Link between SDE and PDE

Starting point is a multidimensional SDE of the form:

$$d\boldsymbol{x}_t = \boldsymbol{\mu}(\boldsymbol{x}_t, t)dt + \boldsymbol{\sigma}(\boldsymbol{x}_t, t)d\boldsymbol{W}_t$$

Feynman-Kac: price of a derivative $u(\boldsymbol{x}_t, t)$ with boundary condition $u(\boldsymbol{x}_T, T)$ at maturity T is given by:

$$\partial_t u + \sum_{k=1}^{n} \mu_i \partial_{x_k} u + \frac{1}{2} \sum_{k,l=1}^{n} \left(\boldsymbol{\sigma}\boldsymbol{\sigma}^T\right)_{kl} \partial_{x_k} \partial_{x_l} u - ru = 0$$

Fokker-Planck: time evolution of the probability density function $p(\boldsymbol{x}_t, t)$ with the initial condition $p(\boldsymbol{x}, t = 0) = \delta(\boldsymbol{x} - \boldsymbol{x_0})$ is given by:

$$\partial_t p = -\sum_{k=1}^{n} \partial_{x_k} \left[\mu_i p\right] + \frac{1}{2} \sum_{k,l=1}^{n} \partial_{x_k} \partial_{x_l} \left[\left(\boldsymbol{\sigma}\boldsymbol{\sigma}^T\right)_{kl} p\right]$$

Feynman-Kac Backward Equation

The SLV model leads to following Feynman-Kac equation for a function $u : \mathbb{R} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}, (x, \nu, t) \mapsto u(x, \nu, t)$:

$$\begin{aligned} 0 &= \partial_t u + \frac{1}{2} L^2 \nu \partial_x^2 u + \frac{1}{2} \eta^2 \sigma^2 \nu \partial_\nu^2 u + \eta \sigma \nu \rho L \partial_x \partial_\nu u \\ &\quad + \left(r - q - \frac{1}{2} L^2 \nu \right) \partial_x u + \kappa \left(\theta - \nu \right) \partial_\nu u - r u \end{aligned}$$

- PDE can be solved using either Implict scheme (slow) or more advanced operator splitting schemes like modified Craig-Sneyd or Hundsdorfer-Verwer in conjunction with damping steps (fast).
- Implementation is mostly harmless, extend `FdmHestonOp`.

Fokker-Planck Forward Equation

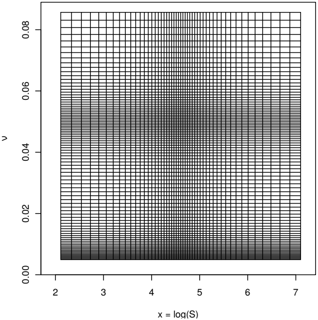
The corresponding Fokker-Planck equation for the probability density $p : \mathbb{R} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}, (x, \nu, t) \mapsto p(x, \nu, t)$ is:

$$\begin{aligned} \partial_t p &= \frac{1}{2}\partial_x^2 \left[L^2 \nu p\right] + \frac{1}{2}\eta^2\sigma^2\partial_\nu^2 \left[\nu p\right] + \eta\sigma\rho\partial_x\partial_\nu \left[L\nu p\right] \\ &\quad -\partial_x \left[\left(r - q - \frac{1}{2}L^2\nu\right) p\right] - \partial_\nu \left[\kappa \left(\theta - \nu\right) p\right] \end{aligned}$$

- Numerical solution of the PDE is cumbersome due to difficult boundary conditions and the Dirac delta distribution as the initial condition.
- PDE can be efficiently solved using operator splitting schemes, preferable the modified Craig-Sneyd scheme

Fokker-Planck Calibration: Last Year's Tool Set

- Coordinate transformation $z = \ln \nu$ to overcome divergent probability density at $\nu \to 0$
- Proper implementation of zero flux boundary condition for $\nu \to 0$ and $\nu \to \nu_{max}$
- Non-uniform meshes in two dimensions
- Semi-Analytical approximation of initial Dirac distribution for small t



- Three important improvements have been added since then

Fokker-Planck Calibration: Prediction-Correction Step

1. Use Fokker-Planck equation to get from

$$p(x,\nu,t) \to p(x,\nu,t+\Delta t)$$

   assuming a piecewise constant leverage function $L(x_t, t)$ in t

2. Calculate leverage function at $t+\Delta t$:

$$L(x,t+\Delta t) = \sigma_{LV}(x,t+\Delta t)\sqrt{\frac{\int_{\mathbb{R}+} p(x,\nu,t+\Delta t)d\nu}{\int_{\mathbb{R}+} \nu p(x,\nu,t+\Delta t)d\nu}}$$

3. Set $t := t+\Delta t$
4. If t is smaller than the final maturity goto 1

Fokker-Planck Calibration: Prediction-Correction Step

1. Set $L(x, t + \Delta t) = L(x, t)$
2. Use Fokker-Planck equation and $L(x_t, t + \Delta t)$ to evolve

$$p(x, \nu, t) \rightarrow p(x, \nu, t + \Delta t)$$

3. Calculate again the leverage function at $t + \Delta t$:

$$L(x, t + \Delta t) = \sigma_{LV}(x, t + \Delta t) \sqrt{\frac{\int_{\mathbb{R}^+} p(x, \nu, t + \Delta t) d\nu}{\int_{\mathbb{R}^+} \nu p(x, \nu, t + \Delta t) d\nu}}$$

4. For number of prediction-correction steps goto 2
5. Set $t := t + \Delta t$
6. If t is smaller than the final maturity goto 1

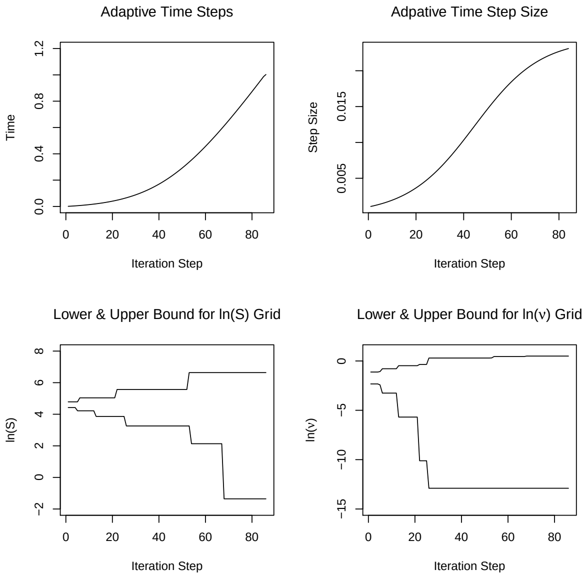
Fokker-Planck Calibration: Grid Optimization

- Observation: The shape of $p(x, \nu, t)$ changes rapidly for small t
- Adapt time step size with the evolution of the probability density

$$\Delta t(t) = \Delta t_{min} e^{-\beta t} + \Delta t_{max}(1 - e^{-\beta t})$$

- Adaptive grid boundaries to concentrate the grid at the singularity then spread out with the evolving density
- The local volatility surface can be used as a guide in x direction, since it generates the right density
- Distribution in ν_t direction is known and can be used to set the size.

Grid Optimization: Examples



Grid Optimization: Cruise Control

- The Calibration routine ends up with many grid related parameters
- Good news: Quality of the solution can be checked at any time

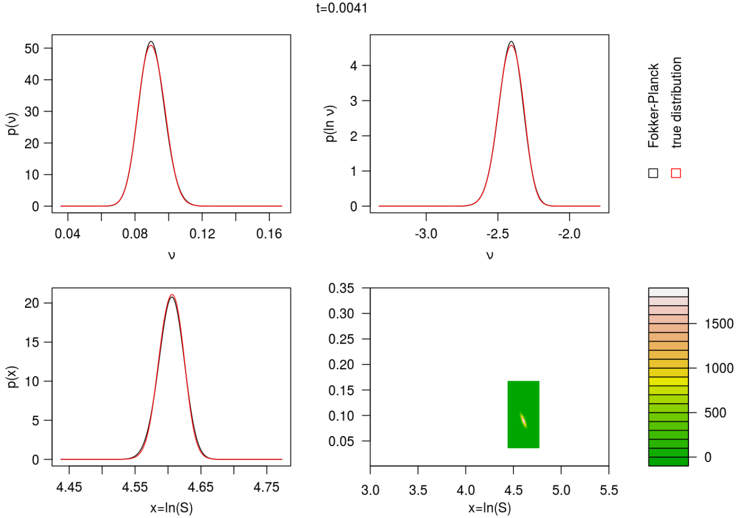
$$\int_{-\infty}^{\infty} p(x,\nu,t)dx \overset{!}{=} \frac{\eta^2\sigma^2\left(1-e^{-\kappa t}\right)}{4\kappa}\chi_d'^2\left(\frac{4\kappa e^{-\kappa t}}{\eta^2\sigma^2\left(1-e^{-\kappa t}\right)}\nu_0\right)$$

$$\int_{0}^{\infty} p(x,\nu,t)d\nu \overset{!}{=} p_{loc}(x,t)$$

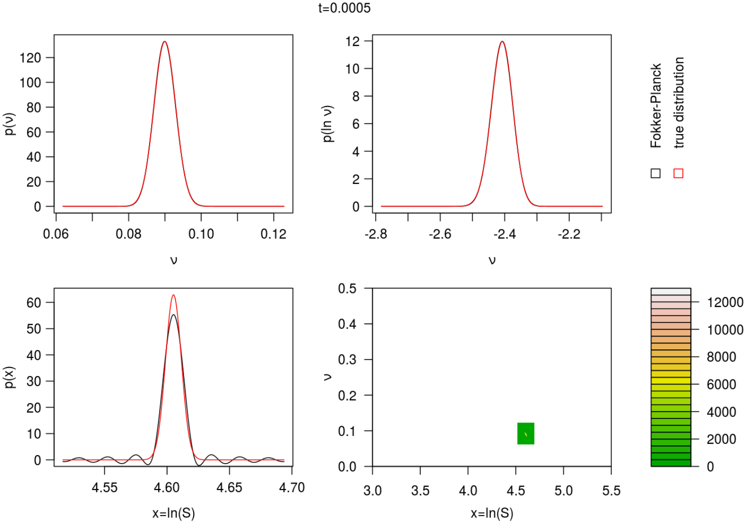
where p_{loc} is the solution of the corresponding Local Volatility Fokker-Planck equation [1]

[1] Bad news: We had to implement a Fokker-Planck solver for Local Volatility models

Cruise Control: Feller Constraint Fulfilled



Cruise Control: Feller Constraint Violated



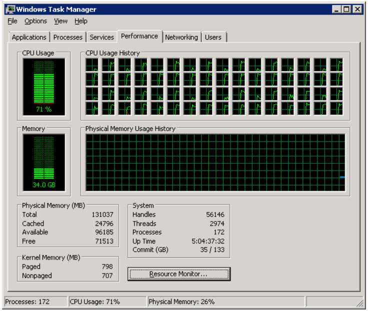
Feyman-Kac Calibration

- The backward Feyman-Kac equation is much simpler to solve than the Fokker-Planck forward equation
- Boundary condition is more well behaved and the initial start condition is not a Dirac delta distribution.

- Does brute force calibration via the Feynman-Kac equation work?
  - Define leverage function by a two dimensional interpolation on benchmark options
  - Value of the leverage function at each benchmark option is a parameter of the optimisation
  - Could add exotics to that, too..

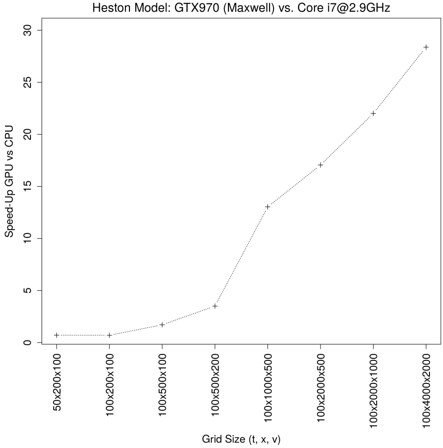
Feyman-Kac Calibration: Performance

- The Levenberg-Marquardt optimizer needs the partial derivatives against all parameters for an optimisation step
- Number of option valuations per Levenberg-Marquardt step grows with square of number of benchmark instruments
- Each option valuation translates into solving a two dimensional PDE
- Needs big machines and parallel computing

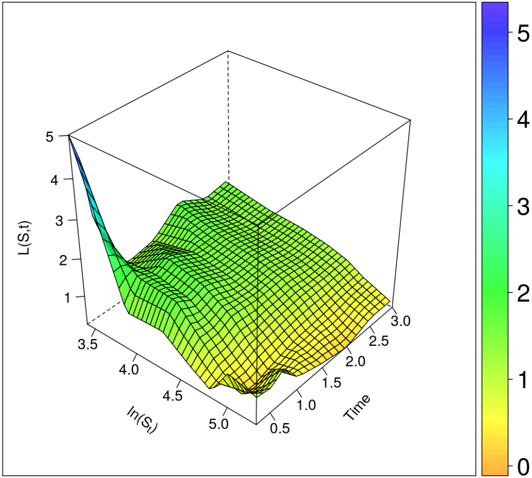


Feyman-Kac Calibration: GPU Computing

- Implementing Operator Splitting using recent CUDA tools has become pretty straight forward
- Construct operators on CPU and transfer to GPU via standard sparse matrix format



Feyman-Kac Calibration: Example



Leverage Function calibrated by brute force optimisation on a 7 $\otimes$ 11 option grid

Feyman-Kac Calibration: Result

- Resulting leverage function tends to oscillate
- Levenberg-Marquardt gets stuck in sub-minima
- Even on big machines (e.g. 64 nodes) calibration might take more than 30 minutes.
- Quadratic runtime scaling with number of benchmark instruments does not allow for a fine granular calibration
- The dual equation or AAD might help to mitigate some performance problems
- GPU will not help as PDE usually do not scale properly on GPUs

$\Longrightarrow$ Feyman-Kac calibration does not look promising

Given a calibrated Heston model and a calibrated local volatility model we can use the SLV model

$$
\begin{aligned}
d\ln S_t &= \left(r_t - q_t - \frac{1}{2}L(S_t,t)^2\nu_t\right)dt + L(S_t,t)\sqrt{\nu_t}\,dW_t^S \\
d\nu_t &= \kappa(\theta-\nu_t)\,dt + \eta\sigma\sqrt{\nu_t}\,dW_t^\nu \\
\rho dt &= dW_t^\nu dW_t^S
\end{aligned}
$$

for two things:

1. Remove calibration errors which the stiffer Heston model exhibits, especially skew for short-dated options
2. Match the volatility dynamics of the market. Interpolate between the two models by tuning η between 0 and 1.

From Heston via SLV to Local Volatility and Back

- Heston model: $\kappa = 2.0, \theta = 0.09, \rho = -0.75, \sigma = 0.4, \nu_0 = 0.09$
- Calibrate Local Volatility $\sigma_{loc}(S_t, t)$ to match Heston prices
- Define scaled leverage function

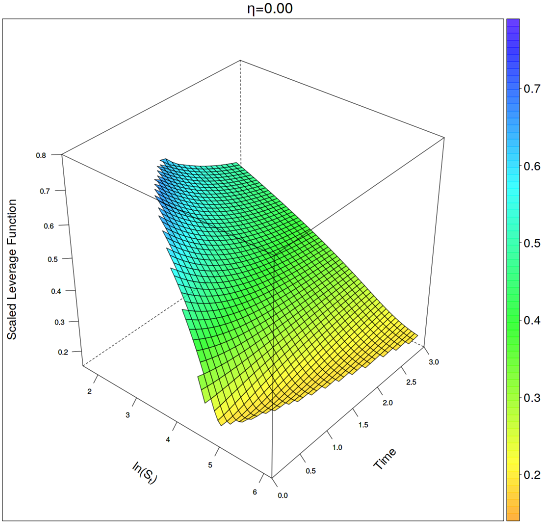
$$L^s(S_t, t) = L(S_t, t)\sqrt{\theta - e^{-\kappa t}(\theta - \nu_0)}$$

- For $\eta = 0$ we get

$$L^s(S_t, t) = \sigma_{loc}(S_t, t)$$

- Can be seen as the most complicated way to calibrate a local volatility surface

From Heston via SLV to Local Volatility and Back



SLV Pricing for Vanilla, Barrier and Express Options

Good news: Finite difference framework is already able to deal with SLV. Implementing a Barrier Option Pricer was literally only 5 lines of code

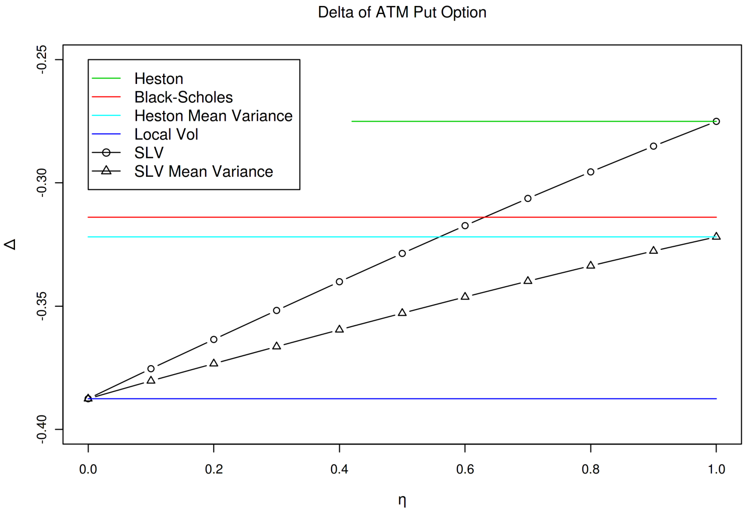
Modified Heston Solver

```
void FdmHestonSolver::performCalculations() const {
    boost::shared_ptr<FdmLinearOpComposite> op(
        new FdmHestonOp(
            solverDesc_.mesher, process_.currentLink(),
            (!quantoHelper_.empty()) ? quantoHelper_.currentLink()
                             : boost::shared_ptr<FdmQuantoHelper>(),
            leverageFct_));

    solver_ = boost::shared_ptr<Fdm2DimSolver>(
        new Fdm2DimSolver(solverDesc_, schemeDesc_, op));
}
```

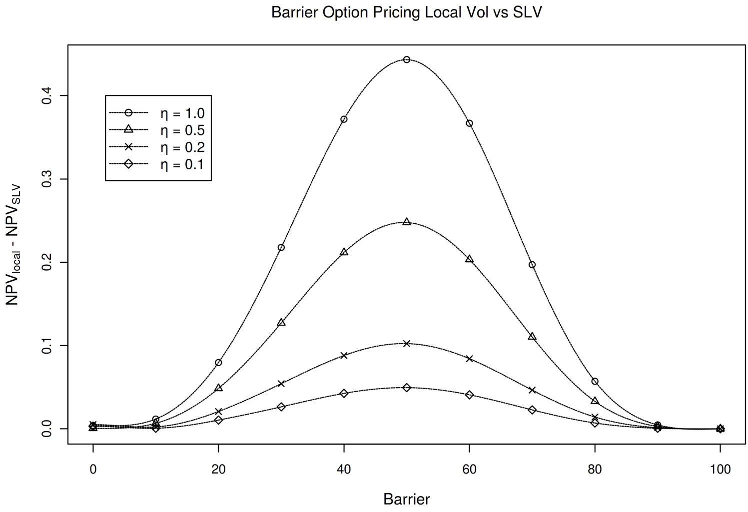
Case Study: Delta of Vanilla Option

Vanilla Put Option: 3y maturity, S_0=100, strike=100



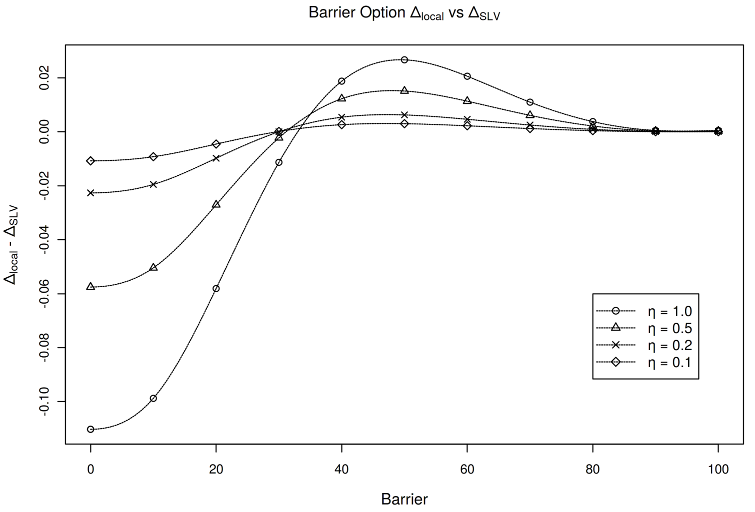
Case Study: Barrier Option Prices

DOP Barrier Option: 3y maturity, S_0=100, strike=100

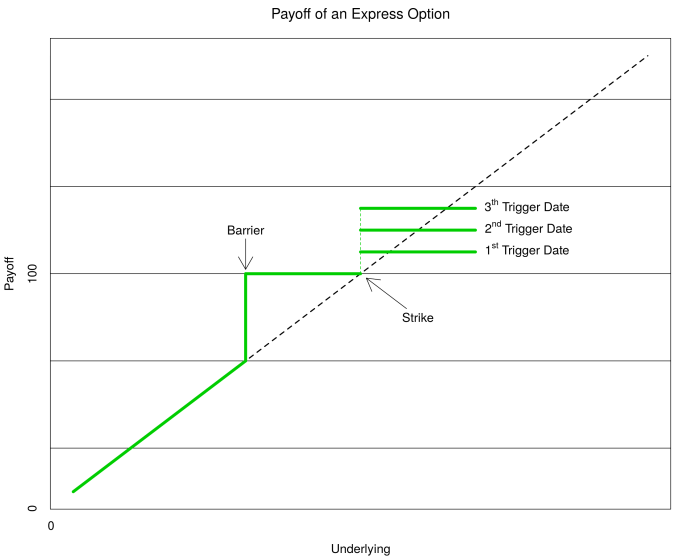


Case Study: Delta of Barrier Options

DOP Barrier Option: 3y maturity, S_0=100, strike=100

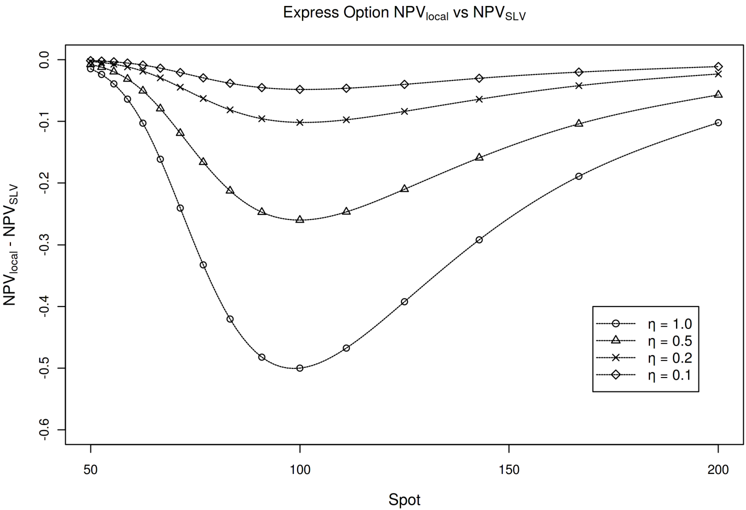


Case Study: Express Option



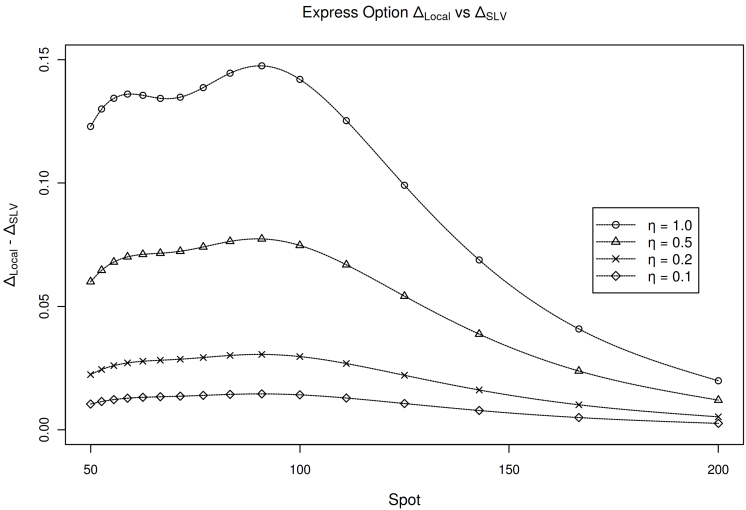
Case Study: Express Option Prices

Express Option: $S_0 = 100$, trigger=100, put strike=50, 3y maturity, coupon = (10%, 20%, 30%)



Case Study: Delta an Express Options

Express Option: $S_0 = 100$, trigger=100, put strike=50, 3y maturity, coupon = (10%, 20%, 30%)



Summary: Heston Stochastic Local Volatility

- Adaptive grid sizes speed-up calibration by concentration on important parameter regions
- Prediction-Correction steps have improved the calibration stability significantly
- Use Cruise Control to monitor solution accuracy.
- Calibration via Feynman-Kac backward equation was slow and inaccurate.
- Easy extension of finite difference framework to price Vanilla, Barrier and Express options
- Choice of η can have a significant impact on prices and greeks

Repository: Pull Request #320

`https://github.com/jschnetm/quantlib/tree/slv/QuantLib`

Literature

Johannes Göttker-Schnetmann and Klaus Spanderen.
Calibration of Heston Stochastic Local Volatility Model., 2015.
http://hpc-quantlib.de/src/slv.pdf.

K.J. in t'Hout and S. Foulon.
ADI Finite Difference Schemes for Option Pricing in the Heston Model with Correlation.
International Journal of Numerical Analysis and Modeling, 7(2):303–320, 2010.

Yu Tian, Zili Zhu, Geoffrey Lee, Fima Klebaner, and Kais Hamza.
Calibrating and Pricing with a Stochastic-Local Volatility Model., 2014.
http://ssrn.com/abstract=2182411.

Disclaimer

The views, opinions, positions or strategies expressed in this presentation are those of the authors and do not necessarily represent the views, opinions, positions or strategies of and should not be attributed to E.ON Global Commodities SE or DZ BANK.