On the Calibration of Stochastic Volatility Models: A Comparison Study

Jia Zhai

Department of Accounting, Finance and Economics,
University of Ulster
Jordanstown, United Kingdom
j.zhai@ulster.ac.uk

Yi Cao

School of Computing and Intelligent Systems
University of Ulster
Londonderry, United Kingdom
cao-y1@email.ulster.ac.uk

Abstract—We studied the application of gradient based optimization methods for calibrating stochastic volatility models. In this study, the algorithmic differentiation is proposed as a novel approach for Greeks computation. The “payoff function independent” feature of algorithmic differentiation offers a unique solution cross distinct models. To this end, we derived, analysed and compared Monte Carlo estimators for computing the gradient of a certain payoff function using four different methods: algorithmic differentiation, Pathwise delta, likelihood ratio and finite differencing. We assessed the accuracy and efficiency of the four methods and their impacts into the optimisation algorithm. Numerical results are presented and discussed.

I. INTRODUCTION

In the financial industry, traders use parameter-dependent models, to price derivatives. Such models can also be used by investors to measure the potential financial risk related to a portfolio of derivatives and to make important investment decisions. Ideally, we would like the data depicted by the model to be as close as possible to the observed data. Since the future is uncertain, we would like the model to be at least consistent with historical market data. This is the aim of model calibration.

In this paper, we studied the problem of calibrating an option pricing model in a risk neutral world. We considered the family of stochastic volatility models known as Vasicek model and European style option to be calibrated to market data. Calibrating such a model amounts to optimising an objective function representing the misfit between the predicted and observed data. Solving the optimization problem by gradient methods requires the calculation of gradient of the objective function with respect to the model parameters to be optimized. Using a Monte Carlo simulation of the option pricing model, we have derived estimators of the gradient by four different methods: Finite Difference Method, Pathwise Method, Likelihood Ratio Method (LRM) and Algorithmic Differentiation (AD).

Our approach to the calibration process is as follows. We considered the Vasicek model, 2-dimensional stochastic differential equations (SDEs), which are discretized using the Euler scheme. We found the analytical solution of the 2-dimensional SDEs. Subsequently, the option price is estimated within a Monte Carlo simulation by either the Euler scheme of the SDE or the analytical solution. Afterwards, we formed the objective function, which is minimised by four different

methods to evaluating the gradient. Numerical results have shown a nearly linear convergence of the minimiser using the Pathwise method or Algorithmic Differentiation, therefore matching the theoretical convergence of the minimiser. In the process of evaluating the gradient, we have shown how to derive the Pathwise and LRM derivative (sensitivity of an output with respect to an input parameter) estimators on a 2-dimensional SDE model, an involved process in terms of calculation.

Related work includes that of [1] wherein stochastic volatility models were calibrated using the Malliavin gradient method, which combines gradient method with Malliavin calculus. Calibration for the Heston's model by an Excel solver using Monte-Carlo simulation was solved in [2] with finite difference method and closed-form solution of the model. A collateral incorporation of a Market Swaption Formula and a rule-of-thumb formula in the calibration routine was proposed in [3]. His algorithm was implemented using C++ in [4]. While each one of those articles discussed one calibration method, our paper has evaluated and compared four different methods to obtain the gradient and discussed their implications to the minimiser.

The rest of the paper is organized as follows. Section II reviews stochastic volatility models in general to focus on the Vasicek. In Section III, we derive and analyse Monte Carlo estimators for evaluating sensitivities; in particular we have derived derivative estimators using Pathwise method and LRM for the 2-dimensional Vasicek model. Section IV discusses the optimization process of model parameters as well as its convergence rate. Section V gives some concluding remarks and an outline of future work.

II. STOCHASTIC VOLATILITY MODELS

One of the stochastic volatility models, widely studied in the computational finance area, is the mean-reverting stochastic volatility model. The most basic form of the mean-reverting process is the Ornstein Uhlenbeck model [5], $dx_t = \theta(\mu - x_t)dt + \sigma dW_t$, where μ represents the long-run equilibrium (mean value level), σ is the degree of volatility around it caused by shocks, θ is the rate by which these shocks dissipate and the variable reverts towards the mean and W_t is a Wiener process under the risk neutral.

The mean-reverting stochastic volatility model (SV model) treats the underlying's volatility as a random process as well, governed by state variables such as the price level of the

underlying security, the tendency of volatility to revert to some long-run mean value, and the variance of the volatility process itself, among others. The popular Heston model is a commonly used mean-reverting SV model, in which the randomness of the variance process varies as the square root of variance [6]. Heston SV model assumes that volatility is a random process that exhibits a tendency to revert towards a long-term mean volatility at a specific rate. Heston also derived a closed-form solution for the price of a European call using Fourier inversion methods. But his model allows any degree of correlation between stochastic volatility and spot asset returns. He found that correlation between volatility and the spot price is important for explaining return skewness and strike-price biases in the Black-Scholes model.

In 1996, Bates [7] further extended Bates's model (developed by himself in 1991) [8] and Heston's model [6] for option pricing on combined stochastic volatility/jump-diffusion (SVJD) processes under systematic jump and volatility risk. He found that stochastic volatility alone cannot explain the "volatility smile" of implied excess kurtosis except under implausible parameters of stochastic volatility, but jump fears can explain the smile. In 2000, Bates refined his 1996's model [9] by incorporating multi-factor specification in stochastic volatilities and time-varying jump risk to explain the negative skewness in S&P 500 future option prices. A closed-form European option pricing model that admits stochastic volatility, stochastic interest rates, and jump-diffusion process was developed in [10]. Besides, we also have non-mean-reverting models, see for example Hull-White model [11] [12], Rendleman and Bartter model [13], Ho-Lee model [14] and Black-Karasinski model [15].

Among all the models, we concentrated on the Vasicek model, a typical example of the mean reverting stochastic volatility model. We have chosen this model because it is one of the simplest stochastic volatility models and it is widely used in financial industry.

A. Risk Neutral Pricing

The Vasicek model we have considered is given by the stochastic differential equations:

$$dS_t = S_t \left(\sqrt{1 - \gamma_1^2} \sigma_t dW_t^1 + \gamma_1 \sigma_t dW_t^2 \right)$$

$$d\sigma_t = \kappa(\theta - \sigma_t)dt + \gamma_2 dW_t^2$$

where dW_t^1 and dW_t^2 are correlated Brownian motion processes with constant correlation ρ . The Euler approximation S_{t_i} and σ_{t_i} are defined as:

$$S_{t_{i+1}} = S_{t_i} \left(1 + \sqrt{1 - \gamma_1^2} \sigma_{t_i} \Delta t Z_{t_i}^1 + \gamma_1 \sigma_{t_i} \Delta t Z_{t_i}^2 \right) \quad (1)$$

$$\sigma_{t_{i+1}} = \sigma_{t_i} + \kappa(\theta - \sigma_{t_i})\Delta t + \gamma_2 \Delta t Z_{t_i}^2$$

where Z_i is independent standard normal random variable for $i = 0, \dots, n-1$ and $\Delta t = t_{i+1} - t_i$ is the fixed time step.

Normally, we can find out the closed form expression of the random variable S_T and estimate the price at the maturity.

We rewrote Euler approximation of the Vasicek model in a standard form:

$$\frac{dS_t}{S_t} = \alpha \sigma_t dW_t^1 + \beta \sigma_t dW_t^2,$$

where $\alpha = \sqrt{1 - \gamma_1^2}$ and $\beta = \gamma_1$. Hence, we get the integral of the left hand side of Vasicek model as:

$$\int_0^T \frac{dS_t}{S_t} = \alpha \sum_{s=0}^T \sigma_s dW_s^1 + \beta \sum_{q=0}^T \sigma_q dW_q^2.$$

Utilizing $It\hat{o}$ formula, we could easily derive the closed-form analytical solution for S_T :

$$S_T = S_0 e^{\left(\alpha \sum_{s=0}^T \sigma_s dW_s^1 + \beta \sum_{q=0}^T \sigma_q dW_q^2 - \frac{1+2\alpha\beta^2}{2} \sum_{r=0}^T \sigma_r^2 dr \right)}. \quad (2)$$

Meanwhile, we can estimate the expected European option prices using Monte Carlo simulation by the closed form analytical solution in Equation (2) or by the Euler approximation Equation (1) over the time interval $[0, T]$.

B. Option Pricing for the Vasicek Model

Consider the case of a family of Vasicek model and the discretized scheme in Equation (1). For the model parameters $\Theta_0[\kappa, \theta, \gamma_1, \gamma_2] = [0.5, 0.2, -0.2, 0.1]$, the initial price of the stock $S_0 = 1$, and the initial volatility $\sigma_0 = 0.2$; we have considered three European call options with strike prices 0.8, 1.0 and 1.2 respectively, and maturity time $T=1$. Prices for the European call options are estimated by Monte Carlo simulation. The results of the simulation, where we use 5×10^4 simulation paths, are shown in Table 1.

Table 1 European call option prices estimated by Euler scheme and closed form solution

Strike	Euler Scheme	Closed form solution
0.8	0.2163	0.2191
1.0	0.0811	0.0831
1.2	0.0215	0.0225

We have also considered the digital call options with the strike price 0.8, 1.0 and 1.2 respectively, and maturity time $T = 1$, i.e.

$$h_1(S_T) = 1_{(S_T \geq 0.8)},$$

$$h_2(S_T) = 1_{(S_T \geq 1.0)}, \quad (3)$$

$$h_3(S_T) = 1_{(S_T \geq 1.2)}.$$

With the same simulation paths, the simulation results are shown in Table 2.

Table 2 Digital call option prices estimated by Euler scheme and closed form solution

Strike	Euler Scheme	Closed form solution
0.8	0.8454	0.8466
1.0	0.4790	0.4858
1.2	0.1528	0.1570

The option prices in two tables show us the closed form solution achieving identical pricing results as the Euler scheme.

The further sensitivity analysis is dependent on the correctness of the closed form formula.

III. EVALUATING OPTION PRICE SENSITIVITIES

To calibrate an option price model to market data, one of the primary methods is to evaluate price sensitivities through the Gradient optimization approach. Consider an option price given by the function $v_\theta = E(\phi(S_\theta))$, where S_θ is a family of random variables defined in a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ such that $S_\theta \in \mathbb{R}^N$ for a parameter θ , $\phi: \mathbb{R}^N \rightarrow \mathbb{R}$ is a kind of payoff function that is Borel measurable. The expectation v_θ can be estimated using Monte Carlo by

$$\hat{v}_{m,\theta} = \frac{\sum_{i=1}^m \phi(S_{i,\theta})}{m},$$

for a sequence $(S_{i,\theta})_{i \geq 1}$ of independent random variables using the distribution law of $S_{i,\theta}$ in the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and m being the number of independent paths. The question is how to evaluate the sensitivity $\frac{dv_\theta}{d\theta} = \frac{dE(\phi(S_\theta))}{d\theta}$.

A. Pathwise Derivative Estimates

The Pathwise method consists in interchanging the derivative and expectation $\frac{dv_\theta}{d\theta} = \frac{dE(\phi(S_\theta))}{d\theta} = E\left(\frac{\partial \phi(S_\theta)}{\partial \theta}\right)$ if the following assumptions hold,

- S_θ is differentiable with respect to θ ;
- $\frac{S_\theta}{\theta}$ is uniformly bounded by a integrable function P ;
- ϕ is Lipschitz continuous and differentiable;

In this case, we derived an unbiased estimator of the derivative $\frac{dv_\theta}{d\theta}$ as

$$\frac{dv_{m,\theta}}{d\theta} = \frac{1}{m} \sum_{i=1}^m \frac{\partial \phi(S_{i,\theta})}{\partial \theta} = \frac{1}{m} \sum_{i=1}^m \frac{\partial \phi(S_i)}{\partial S_i} \frac{S_i}{\theta}.$$

This estimator has the attractive property of a lower variance, e.g., $O\left(\frac{1}{m}\right)$ but relies on the assumption of a differentiable payoff function, which is hardly the case in real applications such as the digital option and the barrier options, with the discontinuous payoff function, in which the Pathwise derivative exists but is entirely uninformative [16]. Even when the payoff function is continuous and piecewise differentiable, it can be very difficult in practice to evaluate the derivative of very complex financial products, which is always the case in modern financial industry. The application of this method to the Vasicek model illustrates this difficulty.

1) Application to the Vasicek Model

To illustrate the derivation of Pathwise derivative estimators, using the discretized Euler scheme of Vasicek in Equation (1), let us consider a European Call Option

$$P = E[e^{-rT}(S_T - K)^+]$$

for which we aim to calculate $\frac{\partial P}{\partial K}$, $\frac{\partial P}{\partial \theta}$, $\frac{\partial P}{\partial \gamma_1}$ and $\frac{\partial P}{\partial \gamma_2}$.

Under the assumption given in Section A, we can derive the Pathwise estimators of the sensitivity based on chain rule

and necessary conversions and simplifications as the following equations:

$$\frac{\partial P}{\partial K} = e^{-rT} 1_{S_T \geq K} S_T \sum_{i=1}^{T-1} \left\{ \frac{S_{i+1} - S_i}{\sigma_i S_{i+1}} [\sum_{l=0}^{i-3} (1 - kdt)^l + (\theta - \sigma_{T-2-l})] \right\},$$

$$\frac{\partial P}{\partial \theta} = e^{-rT} 1_{S_T \geq K} S_T \sum_{i=1}^{T-1} \left\{ \frac{S_{i+1} - S_i}{\sigma_i S_{i+1}} [\sum_{j=0}^{i-1} (1 - kdt)^j kdt] \right\},$$

$$\frac{\partial P}{\partial \gamma_2} = e^{-rT} 1_{S_T \geq K} S_T \sum_{i=1}^{T-1} \left\{ \frac{S_{i+1} - S_i}{\sigma_i S_{i+1}} [\sum_{j=0}^{i-1} (1 - kdt)^{i-1-j} dW_j^2] \right\},$$

$$\frac{\partial P}{\partial \gamma_1} = e^{-rT} 1_{S_T \geq K} \sum_{i=1}^{T-1} \left\{ \sigma_i \left[dW_i^2 - dW_i^1 \frac{\gamma_1}{\sqrt{1 - \gamma_1^2 \sigma_1}} \right] \frac{S_i}{S_{i+1}} S_T \right\}.$$

B. The Likelihood Ratio Derivative Estimate

In this method, the value of the expectation is calculated as

$$v_\theta = E(\phi(S_\theta)) = \int \phi(S) f_\theta(S) dS,$$

where $f_\theta(S)$ is the probability density function for the random variable S_θ . Differentiating this equation gives

$$\frac{dv_\theta}{d\theta} = E\left(\phi(S_\theta) \frac{\partial \log f_\theta(S)}{\partial \theta}\right).$$

The sensitivity can then be estimated by

$$\hat{v}'_{m,\theta} = \frac{1}{m} \sum_{i=1}^m \phi(S_\theta) \frac{\partial \log f_\theta(S)}{\partial \theta}.$$

This method has the advantages of being simple from an implementation viewpoint and applicable even if the payoff function is not continuous however its estimator can have higher variance for certain class of payoff functions [16]. It also requires the existence and knowledge of the probability density function of the random variable S_θ . In most financial engineering literatures, the LRM is illustrated through examples with one-dimensional stochastic differential equation in equation. However, the complication is related to how to get the probability density function of S_θ especially in the case of a 2-dimensional stochastic differential equation pricing model such as Heston's model, GARCH model or the Vasicek model in Equation (2).

1) Application to the Vasicek Model

To use LRM, we started by finding the probability density function $f_\theta(S)$ describing the distribution S_T where θ is the parameter of interest. Based on the closed-form analytical formula for S_T , the probability density function of S_T can be derived as

$$P(S_T) = \frac{1}{\sqrt{2\pi\Delta t \sum_{s=0}^T \sigma_s^2}} \exp\left(-\frac{\ln \frac{S}{S_0} + \frac{1+\alpha\beta^2}{2} \sum_{s=0}^T \sigma_s^2 ds}{2\Delta t \sum_{s=0}^T \sigma_s^2}\right).$$

With the knowledge of the probability density function of S_T , the partial derivatives can be further derived as:

$$\partial \ln P(S_T) / \partial K =$$

$$\frac{\ln \frac{S_T}{S_0}}{2\Delta \sum_{s=0}^T \sigma_s^2} \left(2 \sum_{r=0}^T \frac{\partial \sigma_r}{\partial K} \sigma_r\right) + \frac{1+2\alpha\beta^2}{2} \left(2 \sum_{s=0}^T \sigma_s ds \frac{\partial \sigma_s}{\partial K}\right),$$

$$\begin{aligned}
\frac{\partial \ln P(S_T)}{\partial \theta} &= \frac{-\ln \frac{S_T}{S_0}}{2\Delta \sum_{s=0}^T \sigma_s^2} \left(2 \sum_{r=0}^T \frac{\partial \sigma_r}{\partial \theta} \sigma_r \right) + \frac{1+2\alpha\beta^2}{2} \left(2 \sum_{s=0}^T \sigma_s ds \frac{\partial \sigma_s}{\partial \theta} \right), \\
\frac{\partial \ln P(S_T)}{\partial \gamma_2} &= \frac{-\ln \frac{S_T}{S_0}}{2\Delta \sum_{s=0}^T \sigma_s^2} \left(2 \sum_{r=0}^T \frac{\partial \sigma_r}{\partial \gamma_2} \sigma_r \right) + \frac{1+2\alpha\beta^2}{2} \left(2 \sum_{s=0}^T \sigma_s ds \frac{\partial \sigma_s}{\partial \gamma_2} \right), \\
\frac{\partial \ln P(S_T)}{\partial \gamma_1} &= \frac{\sum_{s=0}^T \sigma_s^2 ds}{2\Delta t \sum_{r=0}^T \sigma_r^2} \left(\frac{1-2\beta^2}{\alpha} \right).
\end{aligned}$$

Note that the LRM estimators for the sensitivities require a complete derivative process, which make use of the Pathwise method. This discourages the use of LRM in high dimensional stochastic differential equation models.

C. Finite Differencing (FD)

Finite differencing allows us to easily estimate the sensitivity $\frac{\partial v_\theta}{\partial \theta}$ even when the payoff function is not regular by using the approximation

$$\frac{\partial v_\theta}{\partial \theta} \approx \frac{v_\theta(\theta+\Delta h) - v_\theta(\theta-\Delta h)}{2\Delta h}.$$

In the method, the initial simulation is run to determine a base price, then the parameter of interest such as θ is perturbed by $\theta \pm \Delta h$ and another two simulations are run to determine the perturbed prices.

D. Algorithmic Differentiation (AD)

Algorithmic Differentiation [17] is a semantics augmentation framework based on the idea that a source program P representing $\mathbf{f}: R^N \rightarrow R^M$ (N, M are number of inputs and outputs respectively), $x \rightarrow y$ can be viewed as a sequence of instructions; each representing a function ϕ_i that has a continuous first order derivative. In particular, the function $\mathbf{f}$ represented by the program P can be that of a price option. This assumes that the program P is piecewise differentiable and therefore we can conceptually fix the program's control flow to view P as a sequence of q assignments. An assignment is defined as given $v_i = \phi(\{v_j\} | j < i)$, $i = 1, \dots, q$ wherein in $j < i$ means v_i depends on v_j , computing the value of a variable v_i in terms of previously defined v_j . Thus P represents a composition of functions $\phi_q \circ \phi_{q-1} \circ \dots \circ \phi_1$ and can be differentiated using the chain rule. Differentiating $\mathbf{f}$ yields the following chain of matrix multiplications that compute the derivative of the function $\mathbf{f}$ represented by the program P .

$$\mathbf{f}'(x) = \phi'_q(v_{q-1})\phi'_{q-1}(v_{q-2}) \dots \phi'_1(x)$$

There are two main AD algorithms both with predictable complexities: the *forward mode* and the *reverse mode* [18]. Denoting $\dot{x}$ as an input directional derivative, the derivative $\dot{y}$ can be computed by the *forward mode* AD as follows:

$$\dot{y} = \mathbf{f}'(x)\dot{x} = \phi'_q(v_{q-1})\phi'_{q-1}(v_{q-2}) \dots \phi'_1(x)\dot{x}.$$

The cost (in terms of floating-point operations) of computing $\nabla \mathbf{f}$ is about $3N$ times the cost of computing $\mathbf{f}$. Denoting $\bar{y}$ the adjoint of the output y , the adjoint $\bar{x}$ of the input x can be computed by *reverse mode* AD as follows:

$$\dot{x} = \mathbf{f}'(x)^T \bar{y} = \phi'_1(x)^T \phi'_2(x_1)^T \dots \phi'_q(x_{q-1})^T \bar{y}.$$

The cost of computing $\nabla \mathbf{f}$ is about $3M$ times the cost of computing $\mathbf{f}$ [19] but the memory requirement may be excessive without the use of sophisticated checkpointing or recalculation strategies. It follows that gradients, with $M = 1$ uses fewer floating-point operations with reverse mode AD.

An AD tool can be implemented using operator overloading or source transformation. The source transformation approach of AD relies on compiler construction technology. It parses the original code into an abstract syntax tree, as in the front-end of a compiler [20]. Certain constructs in the abstract syntax tree may be transformed into a semantically equivalent one suitable to applying the AD technique. This is termed *canonicalization*. Then, the code's statements that calculate real valued variables are augmented with additional statements to calculate their derivatives. Data flow analyses can be performed in order to improve the performance of the AD transformed code, which can be compiled and ran for numerical simulations. The operator overloading approach in overloading arithmetic operations supports derivative calculation as well. MAD [21] is an example of AD tool, which implements the forward mode AD using operator overloading techniques to differentiate MATLAB codes.

E. Numerical Results for Price Sensitivities

The numerical estimation for the sensitivities is carried out using the toolbox MAD [21] as the implementation of the algorithmic differentiation. Under this platform, we evaluated the partial derivatives of option price (sensitivities) with respect to the parameters using the estimators derived in Section III.A-D. This is in fact the gradient evaluation as the expression

$$\frac{\partial}{\partial(\theta, \kappa, \gamma_1, \gamma_2)} P_{\theta, \kappa, \gamma_1, \gamma_2}(S_T).$$

The evaluation results as well as the runtime performance are shown in Table 3.

Table 3 The Deviation and Runtime Ratio of Simulation Results for Sensitivities

methods	Deviation	Runtime Ratio
Forward mode AD	--	18.78
Finite Difference	1.6241×10^{-5}	7.04
Pathwise Method	2.5475×10^{-11}	10.79
LRM	1.3536×10^{-3}	14.79

In the "Deviation" column of Table 3, we chose the gradient results by AD as the standard value and calculate the deviations between it and the Pathwise, Finite Difference and LRM respectively. The "Runtime Ratio" is estimated as

$$RunTimeRatio = \frac{Avg.CpuT(\nabla P_{\theta,\kappa,\gamma_1,\gamma_2}(S_T))}{Avg.CpuT(P_{\theta,\kappa,\gamma_1,\gamma_2}(S_T))},$$

where the *Avg.CpuT* of a function is calculated by MATLAB command “tic-toc pairs”. As a reference, the average elapsed CPU time for the function $P_{\theta,\kappa,\gamma_1,\gamma_2}(S_T)$ is about 1.15946 milliseconds. Several points are noteworthy from Table 3.

- The Pathwise estimator gives results that are almost identical to those from MAD's forward mode AD. Because the Pathwise method is unbiased [16] and theoretically similar to the forward mode AD, we argue that the results from the forward mode AD and the Pathwise method are closer to the true value. Because of the equivalence of the results from the Pathwise and MAD, the computational speed advantage of the Pathwise method is a strong argument in its favour. By overloading the elementary operations in the object function, the forward mode AD calculates the gradients as well as the object function itself. This implies a runtime overhead for MAD's forward mode AD. Nonetheless, an advantage of AD compared with Pathwise method is that it involves no mathematics/programming effort beyond the pricing simulation. This justification is not weak compared with the advantages of the Pathwise method since on the case of a model of more than 1-dimension or different models alternation, the efforts for Pathwise estimator derivation will increase significantly whereas AD minimizes the effort required to evaluate sensitivities at least from the user's viewpoint.
- The FD method gave results with large deviation compared to the Pathwise method. This is because the variance of the estimated derivatives is inversely proportional with the parameter increment Δh and the bias linearly depends on Δh . If Δh is too small, then a large variance can be caused and if Δh is too big, a large bias in the derivative estimate will occur. Furthermore, the finite difference method (using the central difference formula), in the case of the Vasicek Model, showed a runtime ratio of 7.04, almost in line with its theoretical complexity of 8 function evaluations (re-simulations). FD is the fastest of the four methods in this case.
- The deviation of LRM is about 10^6 times greater than that of the Pathwise method and 10^2 times than that of the FD method. The larger deviation is likely due to the likelihood ratio estimators not depending on the form of the payoff function. Its runtime is higher than that of the Pathwise method. This can be expected given that our LRM estimators made use of some of the Pathwise formulae.

IV. OPTION PRICE CALIBRATION

Given observed market data, option price calibration aims at finding some parameters of the model so that the misfit between market data and model data is minimised. In here, the main difficulty lies in finding “market data generators”

satisfying constraints in modern finance: arbitrage-free, risk-neutrality, volatility smile, etc. To illustrate, is it possible to calibrate the S&P index to its model with a maximum error of less than 0.05% for call options?

In our work, we assumed that we are on a virtual/artificial market wherein European style options are traded using the Vasicek stochastic volatility model given in Equation (1). This model depends on four parameters $\kappa, \theta, \gamma_1, \gamma_2$. For a given payoff function, the model is numerically solved using Monte Carlo methods in order to determine the option price. The calibration problem consists then in finding the four parameters such that the distance between a given set of observed option prices and the prices predicted by our model is minimized. This is important as errors on model parameters combined with errors due to the numerical solver (e.g., Monte Carlo simulation) can lead an investor to some financial lose.

A. Calibration Scheme

In our experiment, we considered European call options using the Vasicek model. To measure the distance or misfit between model and market prices, we defined a lost function $\mathbf{L}$ as follows. For an initial vector of parameters $\Theta_0 = [\kappa, \theta, \gamma_1, \gamma_2]$ and a European call option with payoff function ϕ_i , we compute corresponding option prices $P^m(\phi_i(S_{T_i}))$, which are considered to be the observed market prices. We denote P^s the option price calculated with the stochastic volatility model dependent on the four model parameters. The misfit function $\mathbf{L}$ is then defined as the following least squared error

$$\mathbf{L}(\kappa, \theta, \gamma_1, \gamma_2) = \sum_{i=1}^N \left[P_{\kappa,\theta,\gamma_1,\gamma_2}^s(\phi_i(S_{T_i})) - P^m(\phi_i(S_{T_i})) \right]^2,$$

where N is the number of call options. Our aim is then at finding the values of $\Theta^* = [\kappa^*, \theta^*, \gamma_1^*, \gamma_2^*]$ that minimizes the quantity of $\mathbf{L}$. To this end, we used a standard gradient-based optimisation algorithm.

B. Gradient Method for calibration

Minimizing the objective function $\mathbf{L}(\kappa, \theta, \gamma_1, \gamma_2)$ is clearly a nonlinear programming problem. We chose the gradient method to find the minimizers for the function above. Within the algorithm we had to choose an initial guess for the parameter vector Θ_0^* and calculated the partial derivatives with respect to the parameters to be optimized. The algorithm then determines the optimal direction and the step size and is moving downhill on the parameter manifold to the minimum of the objective function. The partial derivatives of $\mathbf{L}$ with respect to Θ are as follows:

$$\begin{aligned} \frac{\partial}{\partial \kappa} \mathbf{L}(\kappa, \theta, \gamma_1, \gamma_2) &= 2 \sum_{i=1}^M \Psi_i[S_{T_i}(S_{0_i}, k_i)] \left[\frac{\partial}{\partial \kappa} P_{\kappa,\theta,\gamma_1,\gamma_2}^s[S_{T_i}(S_{0_i}, k_i)] \right], \\ \frac{\partial}{\partial \theta} \mathbf{L}(\kappa, \theta, \gamma_1, \gamma_2) &= 2 \sum_{i=1}^M \Psi_i[S_{T_i}(S_{0_i}, k_i)] \left[\frac{\partial}{\partial \theta} P_{\kappa,\theta,\gamma_1,\gamma_2}^s[S_{T_i}(S_{0_i}, k_i)] \right], \end{aligned}$$

$$\frac{\partial}{\partial \gamma_j} \mathbf{L}(\kappa, \theta, \gamma_1, \gamma_2) = 2 \sum_{i=1}^M \Psi_i[S_{T_i}(S_{0_i}, k_i)] \left[\frac{\partial}{\partial \gamma_j} P_{\kappa, \theta, \gamma_1, \gamma_2}^s[S_{T_i}(S_{0_i}, k_i)] \right],$$

where

$\Psi_i[S_{T_i}(S_{0_i}, k_i)] = P_{\kappa, \theta, \gamma_1, \gamma_2}^s[S_{T_i}(S_{0_i}, k_i)] - P^m[S_{T_i}(S_{0_i}, k_i)]$, and $j = 1, 2$. The computation of $\Psi_i[S_{T_i}(S_{0_i}, k_i)]$ which mainly involves the option price evaluation in the corresponding model can be done by Monte Carlo through the Vasicek model's Euler scheme or the closed-form solution.

We use the formulas above for partial derivatives evaluations. The experiments setup is as follows. We consider three digital call options with strike prices 0.8, 1.0 and 1.2, respectively and maturity time $T=1$ as Equation (3). We assume that the real market parameters are given as

$$[\kappa, \theta, \gamma_1, \gamma_2] = [0.5, 0.2, -0.2, 0.1]$$

and that the initial price of the stock is $S_0 = 1$ whereas the initial guess of the parameters are $\hat{\Theta}_0 = [0.8, 0.5, -0.5, 0.3]$.

C. Numerical Results for Calibration

This section presents the calibration computation results. We have used the optimization toolbox in MATLAB to minimise the objective function L . The partial derivatives of L with respect to the parameters Θ are evaluated using the Monte Carlo simulation. This gave us the gradients of L . We then monitored the convergence rate of the solver for different iteration numbers. The simulation results for this gradient-based optimisation approach are shown in Fig. 1 and Fig. 2 using logarithmic scale. In the figures, we have plotted the maximum absolute error from the four optimized parameter $\kappa, \theta, \gamma_1, \gamma_2$ against the number of iterations by the optimisation solver. Theoretically, in classical standard gradient method, the number of iterations required for $\hat{\Theta}_k - \Theta \leq \xi$ is $O\left(\frac{1}{\sqrt{k}}\right)$. From this, we have added the theoretical convergence curve for a typical gradient method in the Fig. 1. From Fig. 1 and Fig. 2, we can make the following remarks:

- Fig. 1 shows that for a small number of iterations, the Pathwise method and the forward mode AD reached more precise results than expected by the theoretical curve. As the number of iterations increases, the gradient method using Pathwise derivatives matched the expected convergence, i.e., it has converged linearly. The optimization using AD-obtained derivatives is slightly lagging behind but its convergence is nearly linear.
- In Fig. 2, we have omitted the expected convergence curve since its inclusion makes the curves by FD and LRM nearly invisible from the figure. Nonetheless, those two curves showed a sub-linear converge, i.e., the rate of convergence approximates to 1 as $\lim_{k \rightarrow 10^3} \frac{\hat{\Theta}_{k+1} - \Theta}{\hat{\Theta}_k - \Theta} \approx 1$, where Θ is the true value and $\hat{\Theta}$ is the optimized value after k iterations. Thus,

using the Pathwise or AD supplied derivatives showed a better convergence rate than using FD or LRM derivatives because of the accuracy their evaluations.

- We do not see the use of the forward mode AD as inherently better or worse than the use of the Pathwise method. Theoretically, they are the same computation methods. The difference is that forward mode AD overloads the operators and evaluates the original function as well whereas the Pathwise is focus on deriving the derivative by manually algebraic manipulations.
- Note that an efficient AD algorithm to evaluate gradients is the reverse mode AD. In this particular numerical simulation, we could get a theoretical runtime ratio of less than 5. This can further improve the performance of AD over the remaining methods.

Fig. 1 Convergence by Pathwith, AD and Standard Gradient.

Fig. 2 Convergence by Finite Difference and LRM

V. CONCLUDING REMARKS

In this paper, we have introduced the idea of AD-based calibration method, which can be viewed as an application of AD into a gradient optimisation approach for stochastic volatility models arising in Finance. Thereby, we have calibrated the Vasicek model parameters by minimizing a misfit function for European style options. We focussed on various methods for evaluating the gradient of the objective function in Monte Carlo simulation: the Pathwise method, the likelihood ratio method (LRM), finite differencing and AD. In particular, we derived Monte Carlo sensitivity estimators for the Vasicek model represented as 2-dimensional stochastic differential equations. Numerical results showed that accurate derivatives by the Pathwise and forward mode AD yielded high precision calibration results while finite differencing and LRM displayed a low convergence rate in the calibration process. In finance, there is a need for complex stochastic models in order to exactly predict future market prices. As a result, financial engineers have to recalibrate model parameters to intraday market data every day. The methods we have presented aim to partially respond to the financial industry needs.

One direction for future research is the use of Quasi-Monte Carlo simulation for the AD-based calibration method. Quasi-Monte Carlo offers low discrepancy estimation results and has the potential to accelerate the convergence rate over the ordinary Monte Carlo method. Another direction for future work is the use of AD-based methods in estimating Value at Risk for stochastic volatility models.

ACKNOWLEDGEMENT

The authors thank the Business and Management Research Institute of University of Ulster for financial support for this research and conference attendance. Additionally, the authors gratefully acknowledge the Department of Accounting, Finance and Economics, University of Ulster for a rich and stimulating working environment and Professor Gillian Armstrong and Professor Paul Humphreys for fruitful discussions and advices.

REFERENCE

- [1] C. O. Ewalda and A. Zhang, "A new technique for calibrating stochastic volatility models: the malliavin gradient method," *Quantitative Finance*, pp. 147--158, 2006.
- [2] M. Sergei and U. Nögel, "Heston's stochastic volatility model implementation, calibration and some extensions," *Wilmott magazine*, pp. 74-79, 2004.
- [3] J. G. M. Schoenmakers, "Calibration of libor models to caps and swaptions: a way around intrinsic instabilities via parsimonious structures and a collateral market criterion," in *Risk Europe Conference*, 2002.
- [4] N. PRIVAULT, T. C. Avenue and X. WEI, "Calibration of the libor market model-implementation in premia".
- [5] G.E.Uhlenbeck and L.S.Ornstein, "On the theory of the brownian motion," *Physical Review*, vol. 36, pp. 823--841, 1930.
- [6] S. Heston, "A closed-form solution for options with stochastic volatility with applications to bond and currency options," *The Review of Financial Studies*, vol. 6, pp. 327--343, 1993.
- [7] D. Bates, "Jumps and stochastic volatility: Exchange rate processes implicit in deutsche mark options," *The Review of Financial Studies*, vol. 9, pp. 69--107, 1996.
- [8] D. Bates, "Was it expected? the evidence from options markets," *Journal of Finance*, vol. 46, pp. 1009-1044, 1991.
- [9] D. Bates, "Post 87 crash fears in the s&p 500 futures option market," *Journal of Econometrics*, vol. 94, pp. 181--238, 2000.
- [10] G. Bakshi, C. Cao and Z. Chen, "Empirical performance of alternative option pricing models," *Journal of Finance*, vol. 52, pp. 2003--2049, 1997.
- [11] J. Hull and A. White, "Pricing interest rate derivatives securities," *Review of financial Studies*, vol. 3, pp. 573--592, 1990.
- [12] J. Hull and A. White, "Numerical procedures for implementing term structure modelsii: Two-factor models," *Journal of Derivatives*, vol. 2, pp. 37--48, 1994.
- [13] R. Rendleman and B. Bartter, "The pricing of options on debt securities," *Journal of Financial and Quantitative Analysis*, vol. 15, pp. 11--24, 1980.
- [14] T. S. Ho and S. LEE, "Term structure movements and pricing interest rate contingent claims," *Journal of Finance*, vol. 41, pp. 1011--1029, 1986.
- [15] F. Black and P. Karasinski, "Bond and option pricing when short rates are lognormal," *Financial Analysis Journal*, pp. 52--59, 1991.
- [16] P. Glasserman, Monte carlo methods in financial engineering, New York: Springer-Verlag, 2004.
- [17] E. M. Tadjouddine, "On Formal Certification of AD Transformations," *Advances in Automatic Differentiation - Lecture Notes in Computational*, pp. 23--34, 2008.
- [18] S. A. Forth, M. Tadjouddine, J. D. Pryce and J. K., "Jacobian code generated by source transformation and vertex elimination can be as efficient as hand-coding," *ACM Transactions on Mathematical Software*, vol. 30, pp. 266--299, 2004.
- [19] A. Griewank and A. Walther, Evaluating derivatives: Principles and techniques of algorithmic differentiation, Philadelphia: SIAM, 2000.
- [20] J. Ullman, R. Sethi, A. Aho and M. S. Lam, Compilers: principles, techniques, and tools, Boston: Addison-Wesley Publishing Company, 2006.
- [21] S. A. Forth, "An efficient overloaded implementation of forward mode automatic differentiation in MATLAB," *ACM Transactions on Mathematical*, vol. 32, pp. 195--222, 2006.