

# Local stochastic volatility models: calibration and pricing

Cristian Homescu \*

This version: July 14, 2014

We analyze in detail calibration and pricing performed within the framework of local stochastic volatility *LSV* models, which have become the industry market standard for FX and equity markets. We present the main arguments for the need of having such models, and address the question whether jumps have to be included. We include a comprehensive literature overview, and focus our exposition on important details related to calibration procedures and option pricing using PDEs or PIDEs derived from *LSV* models.

We describe calibration procedures, with special attention given to usage and solution of corresponding forward Kolmogorov PDE/PIDE, and outline powerful algorithms for estimation of model parameters. Emphasis is placed on presenting practical details regarding the setup and the numerical solution of both forward and backward PDEs/PIDEs obtained from the *LSV* models. Consequently we discuss specifics (based on our experience and best practices from literature) regarding choice of boundary conditions, construction of nonuniform spatial grids and adaptive temporal grids, selection of efficient and appropriate finite difference schemes (with possible enhancements), etc. We also show how to practically integrate specific features of various types of financial instruments within calibration and pricing settings.

We consider all questions and topics identified as most relevant during the selection, calibration and pricing procedures associated with local stochastic volatility models, providing answers (to the best of our knowledge), and present references for deeper understanding and for additional perspectives. In a nutshell, it is our intention to present here an effective roadmap for a successful *LSV* journey.

---

\*Email: [cristian.homescu@gmail.com](mailto:cristian.homescu@gmail.com)

# Contents

<b>Contents</b>	<b>2</b>
<b>1 Introduction</b>	<b>4</b>
<b>2 Why use <i>LSV</i> models?</b>	<b>4</b>
2.1 Advantages and disadvantages of <i>LV</i> and <i>SV</i> models	5
2.2 Motivating example	6
2.3 Do we really need to incorporate jumps in a <i>LSV</i> model?	6
2.4 Regime switching as a more parsimonious alternative	7
<b>3 <i>LSV</i> models</b>	<b>7</b>
3.1 Generic specifications for a <i>LSV</i> model	7
3.2 Lipton model	8
3.3 Model with Heston-like dynamics	8
3.4 Model with SABR-like dynamics	8
3.5 Tremor model	9
3.6 Hyp-hyp model	9
3.7 VolMaster model	10
3.8 Exponential Levy model	10
3.9 Tataru and Fisher model	11
3.10 Sepp model	11
<b>4 Calibration of <i>LSV</i> model</b>	<b>11</b>
4.1 Common calibration procedure	12
4.1.1 Calibrate the purely stochastic volatility model	12
4.1.2 Calibrate the leverage function and mixing ratio parameter	12
4.2 Calibration of jump parameters	14
4.3 Calibration using Markovian projection	14
4.4 Other specific calibration approaches	14
4.5 Remarks on numerical calibration	15
<b>5 Pricing using <i>LSV</i> model</b>	<b>16</b>
5.1 PDE approach	16
5.1.1 Example: Pricing PDE for Heston <i>LSV</i> model	17
5.1.2 Example: Pricing PDE for Lipton <i>LSV</i> model	17
5.1.3 Examples of efficient solvers for 2D PDEs in finance	18
5.2 PIDE approach	19
5.2.1 Example: Pricing PIDE for Sepp model	19
5.2.2 Example: Pricing PIDE for exponential Levy model	19
5.2.3 Efficient finite difference solvers for 2D PIDEs	19
5.2.4 PIDEs corresponding to regime switching models	19
5.3 Efficient Monte Carlo approach	20
<b>6 Pricing financial options using PDE/PIDE approach</b>	<b>20</b>
6.1 Pricing path independent options	20
6.2 Pricing strong path dependent options	21
6.3 Pricing weak path dependent options	21
6.4 Pricing early exercise options	22

<b>7</b>	<b>Numerical solution of PDEs and PIDEs</b>	<b>24</b>
7.1	Major types of numerical methods . . . . .	24
7.2	Finite difference schemes for PDEs . . . . .	25
7.2.1	Alternating Direction Implicit scheme . . . . .	26
7.2.2	Enhanced Yanenko scheme . . . . .	26
7.2.3	Predictor corrector scheme . . . . .	26
7.3	Galerkin-Ritz method . . . . .	27
7.4	Finite difference scheme for PIDEs . . . . .	28
7.4.1	Predictor corrector scheme . . . . .	28
7.4.2	Itkin scheme . . . . .	29
7.5	Boundary conditions . . . . .	29
7.5.1	Explicit or implicit boundary conditions? . . . . .	29
7.5.2	Boundary conditions for forward Kolmogorov PDE . . . . .	30
7.6	(Semi-) infinite spatial domain . . . . .	31
7.6.1	Domain truncation . . . . .	31
7.6.2	Transformation of variables . . . . .	31
7.7	Spatial grid . . . . .	32
7.8	Temporal grid . . . . .	32
<b>8</b>	<b>AAD for speed and more speed</b>	<b>32</b>
8.1	AAD for calibration . . . . .	33
8.2	AAD for risk calculation . . . . .	34
<b>9</b>	<b>Conclusion</b>	<b>34</b>
	<b>Bibliography</b>	<b>35</b>
	<b>Appendix A: Connecting LSV leverage function and local volatility</b>	<b>42</b>
	<b>Appendix B: Local volatility function</b>	<b>43</b>
	<b>Appendix C: Fokker-Planck equation for the LSV model</b>	<b>45</b>
	<b>Appendix D: Constructing a non-uniform spatial grid</b>	<b>46</b>
	<b>Appendix E: Enhancements of finite difference approach</b>	<b>48</b>
	<b>Appendix F: Incorporating special features of financial instruments</b>	<b>49</b>
	<b>Appendix G: Spatial discretization formulas</b>	<b>51</b>
	<b>Appendix H: Advanced time discretization schemes</b>	<b>55</b>
	<b>Appendix I: Parameter estimation using Kim filter</b>	<b>56</b>

# 1 Introduction

The selection of an appropriate model to successfully price and hedge financial instruments is based on a careful study of the characteristics for both the financial structure to be considered, and the market in which we have to risk manage the position. The quantitative finance literature has initially considered local volatility  $LV$  and stochastic volatility  $SV$  models as means of explaining the observed market smile. Besides their differing qualities of fitting the market data, these two approaches are fundamentally different in terms of their assumptions regarding the behavior of underlying, reflected for example in the different types of forward smile generated over future times, and usually lead to different attributes of risk sensitivities generated.

However, the need for better models became apparent eventually, given the observed drawbacks for both  $LV$  models and  $SV$  models in conjunction with various asset classes (such as FX, equities or equities) and/or various financial instruments (e.g., barrier and touch options). As described in Clark [2011], a purely stochastic volatility model generates the same smile irrespective of the initial level of spot, and therefore is a “sticky-delta” model – the smile stays anchored at points corresponding to the specified deltas, while a local volatility model parametrized by a local volatility function clearly depends on the spot level (and its initial level), and is therefore “sticky-strike”. Consequently local stochastic volatility  $LSV$  models were introduced in the literature to combine the best characteristics of both  $LV$  and  $SV$  models, while minimizing their downsides.

The  $LSV$  literature contains different viewpoints of modeling and calibration approaches: relying on trinomial tree method Jex et al. [1999], universal volatility model combining  $LV$ ,  $SV$  and jump diffusion models Lipton [2002, 2001], Lipton and McGhee [2002], Lipton et al. [2014], log-normal model for spot process and volatility process with zero correlation Ren et al. [2007], term-structure model with log-normal process for volatility Tataru and Fisher [2010], Monte Carlo based approaches for calibration Henry-Labordere [2009], van der Stoep et al. [2013], Guyon and Henry-Labordere [2013], calibration based on McKean’s particle method Guyon and Henry-Labordere [2011, 2013], hyperbolic-local hyperbolic model Jackel and Kahl [2010], a stochastic volatility following a mean-reverting Ornstein-Uhlenbeck process Choi et al. [2012], adding stochastic interest rate to price long-dated FX options Deelstra and Rayee [2012], incorporating jumps Pagliarani and Pascucci [2012], Sepp [2011a,b], Lipton and McGhee [2002], Numerix [2013], Itkin [2014a], Heston  $LSV$  with constant parameters Clark [2011], Engelmann et al. [2012] or time dependent parameters Tian et al. [2013], Tian [2013], relying on a parametric approach with a forward process for the construction of the leverage function Murex [2011], Wystup [2011], adding local volatility component to an unspanned volatility term structure model Halperin and Itkin [2013], incorporating a mixing weight parameter describing the correlation between spot and slope of the smile Reghai et al. [2012b].

The literature also includes references to approximation formulae useful for faster calibration: based on a heat kernel expansion on a Riemann manifold Henry-Labordere [2005], explicit implied volatilities for multifactor  $LSV$  under five different model dynamics: CEV local volatility, quadratic local volatility, Heston stochastic volatility, 3/2 stochastic volatility, and SABR local-stochastic volatility Lorig et al. [2014], pricing basket options in a local-stochastic volatility model with jumps Shiraya and Takahashi [2013], analytical approximations of vanilla prices based on appropriate regularization of the payoff and a suitably perturbed model Bompis [2013], analytical expressions for vanilla option prices and implied Black volatilities for any parametric  $LSV$  model, using perturbative expansion techniques Jackel and Kahl [2010], closed-form series expansion in powers of correlation Lipton et al. [2014], etc.

## 2 Why use $LSV$ models?

We mention the most important requirements that should be satisfied by any selection and implementation of a model in order to provide competitive pricing and hedging results:

- Consistency with the observed market dynamics
- Consistency with vanilla option prices
- Consistency with market prices of liquid exotic options
- Stable, robust and fast enough calibration of model parameters
- Efficient and stable calculation of prices and Greeks

While all these provisions have to be accounted for, the first three requirements are especially relevant when selecting the model, while the last two provide motivation for selection of numerical methods employed in calibration and/or pricing.

## 2.1 Advantages and disadvantages of *LV* and *SV* models

To understand why there was a need for *LSV* models, let us start by describing the advantages and disadvantages of both *LV* models and *SV* models, following ideas from Sepp [2011a,b]:

### 1. *LV* MODELS

#### a) Advantages

- i. consistent with today's market prices by construction
- ii. calibration may not require numerical optimization

#### b) Disadvantages

- i. tend to replicate rather poorly some characteristics of market dynamics for spot and volatility (implied volatility tends to move too much given a change in the spot, no mean-reversion effect)
- ii. impossible to tune-up the volatility of the implied volatility, as there is simply no parameter for that
- iii. forward volatility implied by *LV* model is not realistic, since it flattens out
- iv. changes in the underlying imply a general parallel shift of the smile (approximately) under *LV* models, while market experience indicates that often smiles are "sticky" and remain invariant under many types of changes

### 2. *SV* MODELS

#### a) Advantages

- i. tend to be more in line with the market dynamics
- ii. equipped to model the term-structure (through mean-reversion parameters) and the volatility of the variance (through vol-of-vol parameters)
- iii. forward volatility implied by *SV* model has a much more realistic behavior

#### b) Disadvantages

- i. calibration is done by least squares optimization, and requires special attention to ensure stability of parameters
- ii. any change in either mean reversion or vol-of-vol requires re-calibration of other parameters
- iii. usually does not fit well short term market skew/smile

The *LSV MODEL* aims to incorporate the advantages (and eliminate the disadvantages) of both *LV* models and *SV* models, by allowing some of the smile to come from stochastic volatility and some to come from a local volatility contribution.

For additional perspectives we encourage the reader to consult practitioner references Crosby [2013], Sepp [2011a,b], Tan [2012], Clark [2011], Lipton [2002], Lipton et al. [2014].

## 2.2 Motivating example

Let us consider the pricing of Double No Touch (DNT) options with *LV* and *SV* models, similar to Crosby [2013], Lipton and McGhee [2002].

DNT options are the most actively traded barrier options in FX markets. These options pay one unit of domestic currency at expiry if the spot FX rate (quoted as the number of units of domestic currency per unit of foreign currency) never trades equal to or outside a lower barrier level nor an upper barrier level. If either the lower barrier level or the upper barrier level are touched or breached prior to expiry, the option expires worthless. The market prices of DNT options on major currency pairs are widely available to market-making banks, and thus a needed requirement for any model in FX markets is to provide prices that match well enough these market prices.

Let us consider that a *LV* model was calibrated to the market prices of vanilla options and then used to price DNT options. It was observed that the resulting model DNT option prices are less than the market prices, i.e., a *LV* model tends to under-price DNT options relative to the market prices. On the other hand, if a *SV* model was calibrated to the market prices of vanilla options and then used to price DNT options, it was noticed that the resulting model DNT option prices are greater than the market prices, i.e., a stochastic volatility model tends to over-price DNT options relative to the market prices. In either case, the degree of mispricing was found to be well in excess of the bid-offer spread.

The above observations were described as fairly robust regardless which type of stochastic volatility model is chosen and which currency pair is chosen (at least for major currency pairs). Adding a compound Poisson jump process may give a more realistic fit to the short-term skew/smile but it does not change the broad conclusion regarding DNT option prices.

To put it another way, a local volatility model tends to under-price DNT options relative to the market prices, a stochastic volatility model tends to overprice DNT options relative to the market prices, so we want to “mix” the two models into a *LSV* model, with a “mixing” parameter calibrated such that we match the market prices of DNT options.

## 2.3 Do we really need to incorporate jumps in a *LSV* model?

While the enhancement of adding jumps is often invoked as a mechanism to increase the smile in the short end over that of a pure stochastic or local-stochastic volatility model, and thus to improve the fitting to market data, this improvement comes with various costs, such as:

- calibration procedure has to be augmented to include jump parameters
- pricing is done by a partial integro-differential equation (PIDE) rather than a PDE.

Thus the main question is whether *LSV* model can accurately describe the market smile at all expiries without paying the price of incorporating jumps. The literature is divided on this topic, with some practitioners Numerix [2013], Lipton and McGhee [2002], Sepp [2011a,b], VolMaster [2014] have considered necessary to add jumps to *SLV* models for both FX and equities markets, while others have found that a *LSV* model without jumps may be sufficient Murex [2011], Tataru and Fisher [2010], Clark [2011]. It is clear, though, that jumps are necessary in some situations, as explained next.

It is mentioned in Sepp [2011a,b] that market prices of exotic equity options depending on forward vol (e.g., cliquet options) imply very steep forward skews, so the value of vol-of-vol parameter must be

very large. However, *LSV* model with a large vol-of-vol parameter cannot be calibrated consistently to given local volatility. In such situations one way for the calibration to provide prices which match well both exotics and vanilla options is to add infrequent but large negative jumps.

In Sepp [2011a] it is shown that *LV* models, *SV* models and jump diffusion *JD* models are not consistent with the implied volatility skew observed in options on the VIX index. Moreover, while only the *SV* model with appropriately chosen jumps can fit the implied VIX skew, results indicate that solely the *LSV* model with jumps can fit both Equity and VIX option skews.

It is argued in Henrotte [2012] that any mixture of local and stochastic volatility cannot fully describe all relevant phenomena observed in equity derivatives space, with ample evidence that jumps add critical features to the dynamics of the underlying that cannot be tackled under a smooth diffusion setting. Short-dated options are all about jumps and, for the popular weekly options, even small jumps can make a huge difference. Rare, but catastrophic, events cannot be ignored either, and can be described as very large jumps, not only on the underlying price, but also on the volatility.

## 2.4 Regime switching as a more parsimonious alternative

If jumps are needed to be added to *LSV* model, a more parsimonious approach is to rely on regime switching concepts instead of incorporating jump diffusion components into the model. This would reduce the complexity of the model, both from theoretical (e.g., Henrotte [2012]) and computational (e.g., Henrotte [2012], Andreasen and Dahlgren [2006], Andersen [2010]) points of view. For a regime switching model with  $M$  regimes and  $N$  factors, the reduction in computational effort is achieved because we end up solving  $M$  coupled PIDEs of dimension  $N - 1$  instead of a PIDE in  $N$  dimensions. Although done in the context of pricing electricity derivatives and respectively commodity derivatives, and not relying on a *LSV* model, the references Andreasen and Dahlgren [2006], Andersen [2010] provide very good examples of computational efficiency obtained when using regime switching models.

## 3 *LSV* models

We start by outlining generic specifications that we need to consider for a good *LSV* model, and then describe various *LSV* models presented in the literature.

### 3.1 Generic specifications for a *LSV* model

We follow Sepp [2011a,b] to describe such specifications:

1. **GLOBAL FACTORS** - specify factors relevant for product risk: stochastic volatility, stochastic interest rate, jumps, default risk, etc
  - a) Estimate or calibrate model parameters for the dynamics of these factors using either historical or market data
  - b) Parameters are updated infrequently
2. **LOCAL FACTORS** - specify local factors for either parametric or non parametric local volatility or local drift (for quantos)
  - a) Parameters of local factors are updated frequently (on the run) to fit the risk-neutral distributions implied by market prices of vanilla options
3. **MIXING WEIGHT** - specify mixing weight  $\gamma$  between stochastic and local volatilities
  - a)  $\gamma$  multiplies vol-of-vol
  - b)  $\gamma$  multiplies both vol-of-vol and correlation

### 3.2 Lipton model

Described in [Lipton \[2002\]](#), [Lipton and McGhee \[2002\]](#), [Lipton et al. \[2014\]](#), this model is representative for a class of *LSV* models which combines a mean reverting process for the volatility or the variance (like in Heston model), with a general local volatility function acting as multiplication factor for the stochastic volatility. The combination is done through a mixing ratio  $\gamma$  which multiplies the vol of vol parameter. It also incorporates jumps, and is formulated as:

$$\begin{aligned} \frac{dS(t)}{S(t)} &= (r_d - r_f - \lambda\varpi) dt + \sigma_L(t, S(t)) \sqrt{V(t)} dW_S(t) + (\exp[J] - 1) dN(t) \quad (3.1) \\ dV(t) &= \kappa(\theta - V(t)) dt + \gamma \sqrt{V(t)} dW_V(t) \\ \langle dW_S, dW_V \rangle &= \rho \end{aligned}$$

with  $S(t)$  the spot FX rate,  $r_d$  and  $r_f$  the domestic and foreign interest rates,  $\lambda$  the intensity rate of Poisson process  $N(t)$ ,  $W_S$  and  $W_V$  Brownian processes correlated by  $\rho$ ,  $V(t)$  stochastic variance,  $\varpi \triangleq \mathbb{E}[\exp[J] - 1]$  and  $\sigma_L(t, S(t))$  denotes a local volatility function.

We note that the model is the same as Heston *SV* plus jump when  $\sigma_L(t, S(t)) \equiv 1$ , and the same as Dupire *LV* plus jumps when  $\gamma \equiv 0$ .

This model was introduced in [Lipton \[2002\]](#); since then it has become popular among both practitioners and academics, e.g., [Andersen and Hutchings \[2009\]](#) and references therein. A version without jumps was implemented by a well-known software provider Murex under the name of Tremor.

### 3.3 Model with Heston-like dynamics

It combines a mean reverting process for the variance (like in Heston model) and a general local volatility function, using a mixing ratio parameter  $\gamma$  to multiply the vol of vol parameter

$$\begin{aligned} \frac{dS(t)}{S(t)} &= (r_d(t) - r_f(t)) dt + L(t, S(t)) \sqrt{V(t)} dW^{(1)}(t) \quad (3.2) \\ dV(t) &= \kappa(t)(\theta(t) - V(t)) dt + \gamma \cdot \xi(t) \sqrt{V(t)} dW^{(2)}(t) \\ Cov(dW^{(1)}(t), dW^{(2)}(t)) &= \rho(t) dt \end{aligned}$$

Models belonging to this class are described in [Tian et al. \[2013\]](#), with Heston parameters  $\kappa, \theta, \xi, \rho$  assumed to have term structure, or in [Clark \[2011\]](#), where the Heston parameters are constant.

### 3.4 Model with SABR-like dynamics

It was described in [Reghai et al. \[2012b\]](#). We note that the mixing ratio parameter  $\gamma$  multiplies both the vol of vol parameter and the correlation

$$\begin{aligned} \frac{dS(t)}{S(t)} &= L(t, S(t)) \cdot f(t, S(t), \sigma(t)) \cdot dW^{(1)}(t) \quad (3.3) \\ d\sigma(t) &= \gamma \cdot \xi(t) \cdot dW^{(2)}(t) \\ Cov(dW^{(1)}(t), dW^{(2)}(t)) &= \rho(t) dt \end{aligned}$$

### 3.5 Tremor model

It models [Wystup \[2013\]](#) the forward as

$$\begin{aligned}
 df(t, T) &= \sqrt{v(t)}g(f(t, T))dW^{(1)}(t) \\
 dv(t) &= \kappa(\theta - v(t))dt + \gamma \cdot \xi \sqrt{v(t)}dW^{(2)}(t) \\
 g(f) &= \min[a + b \cdot f + c \cdot f^2, cap \cdot f] \\
 Cov(dW^{(1)}(t), dW^{(2)}(t)) &= \rho dt
 \end{aligned} \tag{3.4}$$

with  $f(t, T)$  the forward to maturity  $T$  as seen from  $t$ ,  $v(t)$  the variance process,  $\theta$  long term variance set to  $v(0)$  the initial instantaneous variance,  $\kappa$  the rate of mean reversion and  $\xi$  vol of vol. We also have the mixing ratio  $\gamma$  to multiply the vol of vol parameter.

The model is done along the same lines as Universal volatility model described in [Lipton \[2002\]](#), which included an additional jump component in the spot dynamics, and is given in [Eq. 3.1](#).

### 3.6 Hyp-hyp model

It was argued in [Jackel and Kahl \[2010, 2007\]](#) that the main reason for Heston model (and possibly its extensions with CEV for local volatility) to emerge as the most popular model at the time was not that it matches the market dynamics in a particularly realistic way, but rather the existence of analytical prices which can be used in calibration. Moreover, according to references given in [Jackel and Kahl \[2010\]](#), it turns out that Heston model is not as analytically solvable as first thought, nor that its numerical implementation by means of Monte Carlo simulations or finite difference solving is as trivial as one might hope. Research into the scaling of volatility of volatility as a function of the level of volatility seemed to suggest that the stochasticity of volatility observed in the market is probably closer to the SABR. However, SABR model has its own drawbacks, with some of them detailed in [Jackel and Kahl \[2010\]](#).

Thus it was argued that there is a need for a new stochastic volatility model designed to have the same desirable properties as all the above, but fewer, or ideally none, of the undesirable ones. Such a model, introduced in [Jackel and Kahl \[2010, 2007\]](#), combined hyperbolic parametric local vol and hyperbolic stochastic vol

$$\begin{aligned}
 dx &= \sigma_0 \cdot f(x) \cdot g(y) dW^{(1)}(t) \\
 dy &= -\kappa y dt + \alpha \sqrt{2\kappa} dW^{(2)}(t) \\
 f(x) &= \frac{(1 - \beta + \beta^2) \cdot x + (\beta - 1) \cdot [-\beta + \sqrt{x^2 + \beta^2(1 - x^2)}]}{\beta} \\
 g(y) &= y + \sqrt{y^2 + 1} \\
 Cov(dW^{(1)}(t), dW^{(2)}(t)) &= \rho dt
 \end{aligned} \tag{3.5}$$

where  $x$  is the financial observable that underlies the given derivatives pricing problem and  $y$  is the driver of volatility, with  $\beta > 0$  a parameter.

Since both  $f(\cdot)$  and  $g(\cdot)$  are hyperbolic versions of conic sections this model was referred to as the hyperbolic-local hyperbolic-stochastic volatility model, or the Hyp-Hyp model.

The local volatility functional  $f(x)$  was designed to resemble the CEV functional form  $x^\beta$  of local volatility at the forward, up to second order. Unlike the CEV or the displaced diffusion functional form of local volatility, the hyperbolic functional  $f(x)$  not only converges to zero for small  $x$ , but also has finite slope for  $x \rightarrow 0$ , as well as positive slope for  $x \rightarrow \infty$ . When no stochasticity of volatility is present, the local volatility functional gives rise to finite positive implied volatilities for options for

very high and very low strikes, as a consequence of its zero value at zero, finite slope at zero, and finite positive slope for large  $x$ . Unlike the CEV and displaced diffusion models, it does not give rise to the underlying stochastic process attaining or even crossing zero, which is of considerable convenience for both numerical implementations as well as for pricing of forward options.

The stochastic volatility component of the new model was designed to balance the ideal: to be as close as possible to the case of absolute volatility of volatility scaling like  $\sigma^p$  with  $p \approx 1$ , while avoiding the fat tails of a log-normal distribution for volatility in order to circumvent any moment explosions. The chosen hyperbolic functional  $g(y)$  shares level, slope, and curvature with the exponential function in  $y = 0$ , but differs as of the third derivative. It can be seen that while the hyperbolic form gives rise to a density that resembles closely the log-normal distribution reasonably near the bulk of the distribution, it has much thinner tails for very low and very high values of volatility, which is precisely what was desired to be achieved with the selection of this particular functional form.

### 3.7 VolMaster model

It is described in [Wystup \[2013\]](#)

$$\begin{aligned}
 dS(t) &= \left( r^d(t) - r^f(t) \right) S(t)dt + \sigma(t)S(t)dW^{(1)}(t) & (3.6) \\
 dV(t) &= \kappa(t) (\ln(z(t)\theta(t)) - \ln V(t)) V(t)dt + \xi(t)dW^{(2)}(t) \\
 L(t) &= f(S(t), \chi(t), \rho(t), t) \\
 \sigma(t) &= \min[cap, z(t) (\omega(t)V(t) + (1 - \omega(t)) L(t))] \\
 Cov(dW^{(1)}(t), dW^{(2)}(t)) &= \rho(t)dt
 \end{aligned}$$

with  $S(t)$  the spot price,  $\sigma(t)$  volatility,  $V(t)$  stochastic volatility,  $\kappa(t)$  mean reversion speed of stochastic volatility,  $\theta(t)$  the equilibrium level of stochastic volatility,  $\xi(t)$  vol of vol,  $z(t)$  volatility drift factor,  $L(t)$  local volatility function,  $\omega(t)$  stochastic volatility weight.

### 3.8 Exponential Levy model

This model assumes the underlying asset price to be driven by an exponential Levy process

$$\begin{aligned}
 S(t) &= S(0) \exp(L(t)) & (3.7) \\
 L(t) &= \chi t + \sigma W^{(1)}(t) + Z(t)
 \end{aligned}$$

with Levy triplet  $(\chi, \sigma, \nu)$  and  $\chi$  expressed as

$$\chi = r - q - 0.5\sigma^2 - \int_{\mathbb{R}} (e^x - 1 - x \cdot 1_{|x| < 1}) \nu(dx)$$

with  $\nu(dx)$  a Levy measure. Moreover, it assumes that  $\sigma(t) \triangleq L(S, t)\sqrt{v(t)}$  is a combination between a local volatility function  $L(S, t)$  and square root of stochastic variance  $v(t)$ , as described in [Itkin \[2014a\]](#), with  $v(t)$  having dynamics given by

$$\begin{aligned}
 dv(t) &= \kappa(v_\infty - v(t)) dt + \gamma \xi v(t) dW^{(2)}(t) & (3.8) \\
 Cov(dW^{(1)}(t), dW^{(2)}(t)) &= \gamma \rho(t) dt
 \end{aligned}$$

with parameter  $\beta$  determining a mean-reverting CEV process for  $V(t)$  and assumed to be a calibrated parameter of the model.

A version of the model was given in [Pagliarani and Pascucci \[2012\]](#) as

$$\begin{aligned}
dX(t) &= (r - \mu_1 - 0.5\sigma^2(t, X(t-)) \cdot v(t)) dt + \sigma(t, X(t-)) dW^{(1)}(t) + Z(t) \\
dv(t) &= \kappa(\theta - v(t)) dt + \gamma \cdot \xi \sqrt{v(t)} \left( \rho dW^{(1)}(t) + \sqrt{1 - \rho^2} dW^{(2)}(t) \right)
\end{aligned} \tag{3.9}$$

where  $Z(t)$  is a pure jump Levy process with Levy triplet  $(\mu_1, 0, \nu)$  and  $\sigma(t, X)$  is local volatility function.

### 3.9 Tataru and Fisher model

It was described in [Tataru and Fisher \[2010\]](#)

$$\begin{aligned}
\frac{dS(t)}{S(t)} &= (r_d(t) - r_f(t)) dt + L(t, S(t)) v(t) dW^{(1)}(t) \\
dv(t) &= \kappa(\theta - v(t)) dt + \xi v(t) dW^{(2)}(t) \\
Cov(dW^{(1)}(t), dW^{(2)}(t)) &= \rho(t) dt
\end{aligned} \tag{3.10}$$

where the local volatility  $L$  and stochastic volatility  $V$  are mixed using  $\gamma \in [0, 1]$  for

$$\begin{aligned}
\xi &= \gamma \xi_{MAX} \\
\rho &= \gamma \rho
\end{aligned}$$

### 3.10 Sepp model

It was described in [Sepp \[2011b,a\]](#)

$$\begin{aligned}
dS(t) &= \mu(t) S(t_-) dt + L(t_-, S(t_-)) \cdot \vartheta(t_-, Y(t_-)) \cdot S(t_-) \cdot dW^{(1)}(t) \\
&\quad + ((e^{-\nu} - 1) dN(t) + \lambda \nu dt) \cdot S(t_-) \\
dY(t) &= -\kappa \cdot Y(t) \cdot dt + \gamma \cdot \xi \cdot dW^{(2)}(t) + \eta \cdot dN(t) \\
Cov(dW^{(1)}(t), dW^{(2)}(t)) &= \gamma \cdot \rho(t) \cdot dt
\end{aligned} \tag{3.11}$$

with  $S(t)$  the underlying and  $Y(t)$  the factor for instantaneous stochastic volatility,  $L(t, S)$  the local volatility function,  $\kappa$  mean reversion,  $\xi$  vol of vol,  $N(t)$  a Poisson process with intensity  $\lambda$ .

We note that the mixing ratio  $\gamma$  multiplies both the vol of vol parameter and the correlation.

The jumps in  $S(t)$  and  $Y(t)$  are assumed to be simultaneous and discrete, with magnitudes  $-\nu < 0$  and  $\eta > 0$ , while the volatility mapping  $\vartheta(t, Y(t))$  is defined as

$$\vartheta(t, Y(t)) = \exp[Y(t) - \mathbb{V}[Y(t)]]$$

with  $\mathbb{V}[Y(t)]$  the variance of  $Y(t)$ . For  $\rho = 0$ , we have  $\mathbb{E}[\vartheta(t, Y(t)) | Y(0) = 0] = 1$ , so that  $\vartheta(t, Y(t))$  introduces "volatility-of-volatility" effect without affecting the local volatility close to the spot

## 4 Calibration of LSV model

While calibration of each model has specific characteristics and "finer points", there are common aspects of an *LSV* calibration which will be presented in this chapter. We describe these common characteristics for a *LSV* model which incorporates a leverage function to mix the *LV* and *SV* components, and provide details on enhancing the calibration procedure when *LSV* is extended to include jumps, represented via jump diffusion processes, Levy processes or regime switching.

## 4.1 Common calibration procedure

A two-stage calibration procedure is most common Clark [2011], Tan [2012], Tian et al. [2013]:

1. Calibrate the purely stochastic volatility model to market data
2. Calibrate the local volatility correction (the “leverage function”) through optimal selection of mixing ratio parameter  $\gamma$

When we calibrate the purely stochastic volatility model, the resulting model can match well the market data around ATM and in intermediate regions, but has troubles matching the behavior for the wings, especially for far ITM and OTM strikes. This behavior is improved using the local volatility correction (the “leverage function”), which pushes the pure stochastic volatility model in the right direction towards market implied volatilities. Thus we would have 2 sets of parameters to be calibrated:

- parameters of  $SV$  dynamics
- parameters of leverage function

Without significant loss of generality we exemplify the procedure for a Heston-like  $LSV$  model, as described by Eqs. (3.2).

### 4.1.1 Calibrate the purely stochastic volatility model

The purely stochastic volatility component of  $LSV$  model is obtained by setting the leverage function to unity function  $L(t, S) \equiv 1$  and the mixing ratio parameter to 1:  $\gamma = 1$ .

Calibrating parameters of stochastic volatility models is a topic that have been extensively studied in the literature. While such a calibration has to usually overcome issues of theoretical, numerical and computational nature, many of these challenges were already dealt with in the literature, and will not be addressed here.

Regarding calibration of Heston model, the reader may consult Zeliade [2011], Rouah [2013], Benhamou et al. [2010] for efficient approaches for models with constant and, respectively, time dependent parameters.

### 4.1.2 Calibrate the leverage function and mixing ratio parameter

The leverage function pushes the implied volatilities generated by the stochastic volatility model towards market implied volatilities. This calibration is based on solving a 2D Fokker Planck (forward Kolmogorov) PDE, with the value of the mixing ratio parameter  $\gamma$  being determined through matching the price of some exotic options or by historical estimation.

As shown in Appendix A, the leverage function can be found using following formula

$$L(x, t) = \sigma_{LV}(x, t) \sqrt{\frac{\int_0^\infty p_{LSV}(t, x, y) dy}{\int_0^\infty t \cdot p_{LSV}(t, x, y) dy}} \quad (4.1)$$

with  $\sigma_{LV}(x, t)$  local volatility function and  $p_{LSV}(t, S(t), V(t))$  transition probability of  $LSV$  model. Thus we see that in order to calculate the leverage function we need to determine the local volatility function and the transition probability density function.

Details on how to obtain the local volatility function are given in Appendix B.

We also know that the Fokker-Planck equation describes the evolution of the transition probability density (see Appendix C). To exemplify, we consider the Heston  $LSV$  model given in (3.2), after being

enhanced with following change of variables to avoid issues with Feller condition being violated [Tian et al. \[2013\]](#), [Clark \[2011\]](#)

$$\begin{aligned}x(t) &= \ln \frac{S(t)}{S(0)} \\z(t) &= \ln \frac{v(t)}{v(0)}\end{aligned}\tag{4.2}$$

Then we obtain the following Fokker-Planck (forward Kolmogorov) PDE:

$$\begin{aligned}\frac{\partial P}{\partial t} &= -\frac{\partial}{\partial x} [(r(t) - 0.5L^2(x, t) \exp(z) v_0) P] + \frac{\partial^2}{\partial z \partial x} [\gamma^2 \xi \rho L(x, t) P] \\&\quad - \frac{\partial}{\partial z} \left[ \left( (\kappa \theta - 0.5\gamma^2 \xi^2) \frac{1}{\exp(z) v_0} - \kappa \right) P \right] \\&\quad + 0.5 \frac{\partial^2}{\partial x^2} [L^2(x, t) \exp(z) v_0 P] + 0.5 \frac{\partial^2}{\partial z^2} \left[ \frac{\gamma^2 \xi^2}{\exp(z) v_0} P \right] \\P(x, z, 0) &= \delta(x - x_0) \delta(z - z_0)\end{aligned}\tag{4.3}$$

where  $\delta(\cdot)$  is Dirac function and  $x_0 = z_0 = 0$ .

In conclusion, if we solve the Fokker-Planck equation numerically, we can evaluate the leverage function via integrals of probability density function.

Finally we have all the ingredients to calibrate the leverage function and mixing ratio parameter. We follow the approach described in section 6.8.3 of [Clark \[2011\]](#), which is along the same lines as bootstrapping an yield curve, however attempting to infer a surface  $L(x, t)$  rather than a curve.

Suppose that we have market data for a sequence of  $N$  timepoints  $\{0, t_1, t_2, \dots, t_N\}$ . The approach is to solve Eq. (4.3) numerically, marching forward timestep by timestep, and after each iteration in the solution of (4.3) to refine the function  $L(x, t)$  by use of (4.1). Thus we obtain the following steps:

1. Start at time  $t = 0$  with an initial local volatility correction  $L(x, 0) = 1$  for all  $x$
2. Apply a forward time stepping scheme for (4.3) to go to next time point
3. Refine the function  $L(x, t)$  by use of (4.1), at each required level of  $X$ , taking numerical integrals in the variance dimension. Then update the diffusion, convection and force terms in (4.3).

We apply the mixing ratio parameter  $\gamma$  to the vol of vol (and possibly to correlation) for each maturity, with the market data input given by vanilla and perhaps exotic option prices. The local volatility data and the Fokker-Planck equation are used to generate probability density function, while corresponding leverage function is utilized to price the input market vanillas and exotics. We perform one dimensional optimization to identify the value of the mixing ratio parameter  $\gamma$  that gives the smallest overall error, and the procedure is repeated for the next maturity.

When solving Fokker-Planck PDE, potential numerical issues may appear [Clark \[2011\]](#):

- The initial condition for the Fokker-Planck equation is highly singular. This issue can be addressed by
  - having an aggressively nonuniform mesh to pack more mesh points in around  $(x_0, v_0)$  in the spatial dimension, and  $t = 0$  in the time dimension.
  - use bivariate normal distribution density function with a very small time step to approximate the Dirac delta function, as in [Tian et al. \[2013\]](#)
- Certain markets (e.g., FX) will require a nontrivial correlation, which can be problematic for some finite difference schemes

- If the Feller condition is violated, then probability mass will tend to pile up against the  $v = 0$  boundary. A nonuniform mesh in variance with a concentration of points in a boundary layer adjoining  $v = 0$  helps considerably in such a situation.
- The calibration works best for shorter dated maturities for currency pairs with relatively symmetric smiles. For longer dated calibrations to skewed currency pairs, such as USDJPY and EURJPY, we find that the numerics are much more challenging

## 4.2 Calibration of jump parameters

If jump processes are incorporated into the model (via jump diffusion, Levy processes, or regime switching), the corresponding jump parameters have to be estimated using historical data, e.g., [Sepp \[2011a,b\]](#), and not part of the regular calibration. Since we expect jumps to be rare, extreme events, an optimal value provided by optimizer of 100 jumps per year would not match the observed data, although it may be perfectly valid answer from a purely numerical point of view.

Estimation can also be done through filtering, which combines maximum likelihood method with a filter or other algorithms, such as

- EM (expectation maximization) algorithm [Erlwein \[2008\]](#), [Weron \[2009, 2006\]](#)
- Kim Filter [Weron \[2006\]](#), [Bloechlinger \[2008\]](#)
- Baum Welch algorithm [Mittra \[2009\]](#)
- Hamilton filter [Burger et al. \[2008\]](#)

Our recommendation is to use the Kim filter, which combines the Kalman filter, the Hamilton filter and a collapsing procedure. It is an optimal estimator in the sense that no other estimator based on a linear function of the information set yields a smaller mean square error. The reader is referred to Appendix I, to chapter 5 of [Erlwein \[2008\]](#) and to [Kim and Nelson \[1999\]](#) for additional details.

## 4.3 Calibration using Markovian projection

This approach [Piterbarg. \[2007\]](#) moves away from the direct solution of the *LSV* model and derives closed-form approximations, via the Markovian projection, to prices of European options on various underlyings. Work on Markovian projections in the context of the *SLV* models has also been presented in [Henry-Labordere \[2009\]](#), where a so-called “effective local volatility” was derived.

The Markovian projections can be widely applied but require a number of conditional expectations to be determined. Very often these expectations are not available analytically and brute-force assumptions need to be imposed so that approximations can be defined. Although mathematically appealing the Markovian projection technique preserves only marginal densities and does not keep marginal distributions of orders higher than one intact. Due to this, prices of securities depending on stock values at multiple times, such as American options and barriers, may significantly differ between the original model and the projected model.

## 4.4 Other specific calibration approaches

[Ren et al. \[2007\]](#) proposed a stochastic volatility model driven by a lognormal volatility process and developed a tailor-made algorithm for solving the corresponding Kolmogorov forward PDE. Their approach is based on an algorithm described in [Lipton \[2002\]](#). An extension of this technique to the Heston *LSV* was presented in [Engelmann et al. \[2012\]](#), which employed a finite volume scheme for the model evaluation.

Levenberg-Marquardt optimization technique for a non-linear Fokker-Planck equation was applied in [Tataru and Fisher \[2010\]](#). Another approach for simulation was proposed in [Deelstra and Rayee \[2012\]](#). By assuming zero correlation between the volatility process and the underlying asset it is possible to efficiently simulate the extended Schöbel-Zhu model.

#### 4.5 Remarks on numerical calibration

Having a successful calibration relies on more than just selecting a model to accurately capture the behavior and characteristics of a specific market. There are various aspects of the calibration procedure that one needs to pay special attention to, such as:

- which instruments are included in calibration procedure
- formulation of the cost functional which has to be minimized
- stability and efficiency of numerical optimizer

It is an industry preferred practice to use primarily out-of-the-money options for calibration, due to their greater liquidity and model sensitivity compared to their in-the-money counterparts. The cost functional is constructed to incorporate either weighted differences between prices given by the model and prices seen in the market, or weighted differences of corresponding implied volatilities. The cost functional is usually based on a  $L^2$ -norm, although other norms have been considered, while the choice of using either prices or implied volatilities is determined by the quantity (price or vol) which is provided directly by the model.

Practitioners usually compute the calibration weights as inverse proportional to

- the square of the bid-ask spreads, to give more importance to the most liquid options
- the square of the Black-Scholes vegas
- a combination of the 2 approaches.

In some papers it is argued that it is statistically optimal (minimal variance of the estimates) to choose the weights as the inverse of the variance of the residuals, which is then considered to be proportional to the inverse of squared bid-ask spread.

The optimization problem to be solved is not well-posed in vast majority of practical calibrations encountered in quantitative finance. In other words, there may be many local minima, one of which is located by the "solver", as opposed to the global minimum, the results may be highly sensitive to the starting point of the "solver" algorithm, and small changes in calibration inputs may result in large changes in values of parameters provided by the "solver".

This issue can be addressed in various ways, such as

1. adding a regularization parameter to the cost functional, possibly of Tikhonov type, to increase convexity of the augmented cost functional
2. employing a better optimizer
3. incorporating historical data into calibration.

Let us consider an example of calibrating a model which incorporates jumps. Since we expect jumps to be rare, extreme events, an optimal value provided by optimizer of 100 jumps per year would not match the observed data, although it may be acceptable from a purely numerical point of view. In fact, it is likely we would expect a few (e.g., less than 5) jumps per year on the average. This gives a very rough estimate of the real-world physical measure jump intensity rate, with more refined estimates

obtained from filtering of time-series data (which is roughly counting the number of jumps per year bigger than a specified number of standard deviations). One could also use time-series data to estimate the mean jump size, and we may probably expect the estimates of the mean jump size to be equivalent (in magnitude) to, say, at least 2 standard deviation daily moves (the sign can be determined from the slope of the implied volatility as a function of strike).

Other examples of incorporating historical data into calibration include

- relying on previous day's calibrated parameters (or an average over  $N$  previous days) as initial guesses for the numerical calibration
- estimating mean reversion levels, "vol-of-vol" parameters, or correlation values.

Such parameter estimates are very rough and approximate but, nonetheless, it should be possible to "guesstimate" many (if not all) parameter values (at least to the right order of magnitude) before the calibration takes place. For example, we might expect mean-reversion rates to reflect characteristic time-scales for specific markets. Let us assume such time-scales to be 3 months to 3 years for FX markets and 3 years to 40 years for interest-rate markets. This might translate into mean-reversion rates (at least for single-factor models with a single mean-reversion rate - we might have to modify our reasoning if there are two mean-reversion rates) of the order of magnitude of 0.33 to 4.0 for FX options models with mean-reverting stochastic volatility and of the order of magnitude of 0.025 to 0.33 for interest-rate derivatives models.

The choice of the optimizer is also extremely important, for both stability and efficiency reasons. Gradient based optimizers, for example, are likely to get stuck in a local minimum and are in addition dependent on the initial parameter guess. To avoid this issue, the calibration may be done using some stochastic global optimization routine, such as Differential Evolution, which is considered the most robust, and utilized for financial calibrations Bloch and Coello [2010], Vollrath and Wendland [2009], Gilli and Schumann [2010], Schatz [2011], Ardia et al. [2011], Le Floch [2014b], FINCAD [2007].

However, relying solely on global optimization is much more computationally demanding. Thus the recommended approach is to use a hybrid optimizer, which combines a global optimizer with a local optimizer.. First we run a global optimizer such as Differential Evolution for a few iterations. In the second stage we run a local optimizer, using as initial guess the output from the global optimizer.

The reader is also referred to chapter 7 of Homescu [2011b] for additional discussion and references related to this topic.

## 5 Pricing using $LSV$ model

Depending whether the  $LSV$  model includes jumps or not, we obtain a pricing PIDE or respectively PDE. We discuss each type next.

### 5.1 PDE approach

The generic pricing PIDE for value of option  $U(t, S, Y)$  is given by

$$\begin{aligned} \frac{\partial U}{\partial t} &= \mathcal{L}(t, S, Y) \\ U(t, S, Y) &= \text{Payoff}(S) \end{aligned}$$

### 5.1.1 Example: Pricing PDE for Heston LSV model

The resulting pricing PIDE, Tian [2013], Tian et al. [2013], using transformations shown in (4.2):

$$\begin{aligned} \frac{\partial U}{\partial t} &= 0.5 [L(t, x)]^2 V(t) \frac{\partial^2 U}{\partial x^2} + \left( r_d(t) - r_f(t) - 0.5 [L(t, x)]^2 V \right) \frac{\partial U}{\partial x} \\ &+ 0.5 \gamma^2 \xi^2 \frac{1}{V(t)} \frac{\partial^2 U}{\partial z^2} + \left[ (\kappa \theta - 0.5 \xi^2) \frac{1}{V(t)} - \kappa \right] \frac{\partial U}{\partial z} \\ &+ \rho \cdot \gamma \cdot \xi \cdot L(t, x) \frac{\partial^2 U}{\partial x \partial z} - r_d(t) U \end{aligned}$$

### 5.1.2 Example: Pricing PDE for Lipton LSV model

We consider the Lipton model(3.1) as given in Lipton et al. [2014], namely without jumps and with forward price as underlying:

$$\begin{aligned} dF(t) &= \sigma_L(F(t)) \sqrt{v(t)} dW_F(t) \\ dv(t) &= \kappa(\theta - v(t)) dt + \gamma \sqrt{v(t)} dW_v(t) \\ dW_F dW_v &= \rho dt \end{aligned} \tag{5.1}$$

with initial conditions

$$\begin{aligned} F(0) &= F_0 \\ v(0) &= v_0 \end{aligned}$$

A properly normalized system of SDEs can be written as follows:

$$\begin{aligned} d\tilde{F}(\bar{t}) &= \tilde{\sigma}_L(\tilde{F}(\bar{t})) \sqrt{\tilde{v}(\bar{t})} dW_F(\bar{t}) \\ d\tilde{v}(\bar{t}) &= \tilde{\kappa}(1 - \tilde{v}(\bar{t})) d\bar{t} + \tilde{\gamma} \sqrt{\tilde{v}(\bar{t})} dW_v(\bar{t}) \\ dW_F dW_v &= \rho d\bar{t} \end{aligned} \tag{5.2}$$

with corresponding initial conditions

$$\begin{aligned} \tilde{F}(0) &= 1 \\ \tilde{v}(0) &= \bar{v}_0 \end{aligned}$$

where normalized (dimensionless) quantities have following formulas:

$$\begin{aligned} \bar{t} &= \Sigma^2 t \\ dW_F(\bar{t}) &= \Sigma dW_F(t) \\ dW_V(\bar{t}) &= \Sigma dW_V(t) \\ \tilde{F}(\bar{t}) &= \frac{F(t)}{F_0} \\ \tilde{v}(\bar{t}) &= \frac{v(t)}{v_0} \\ \tilde{\sigma}_L(\tilde{F}(\bar{t})) &= \frac{\sigma_L(\tilde{F}(\bar{t}) \cdot F_0)}{\sigma_L(F_0)} \end{aligned}$$

$$\begin{aligned}\tilde{\kappa} &= \frac{\kappa}{\Sigma^2} \\ \tilde{\gamma} &= \frac{\gamma}{\Sigma\sqrt{\theta}} \\ \tilde{v}(\tilde{t}) &= \frac{v(t)}{\theta}\end{aligned}$$

with

$$\Sigma \triangleq \frac{\sigma_L(F_0)\sqrt{\theta}}{F_0}$$

Then the corresponding normalized PDE has time independent coefficients, and after introducing  $\tau = T - t$  we obtain the following forward PDE (for simplicity we have omitted bars and  $\sim$  from the notation of variables) :

$$\frac{\partial V}{\partial \tau} = \frac{v\sigma_L^2(F)}{2} \frac{\partial^2 V}{\partial F^2} + \rho\gamma v\sigma_L(F) \frac{\partial^2 V}{\partial F \partial v} + \frac{\gamma^2 v}{2} \frac{\partial^2 V}{\partial v^2} + \kappa(1-v) \frac{\partial V}{\partial v} \quad (5.3)$$

To simplify Eq.(5.3) we use Liouville transformation (see [Lipton \[2001\]](#), [Lipton et al. \[2014\]](#))

$$\begin{aligned}(F, V) &\implies (X, U) \\ \frac{dF}{\sigma_L(F)} &= dX \\ X &= \int_1^F \frac{dF}{\sigma_L(F)} \\ V &= U\sqrt{\sigma_L}\end{aligned}$$

and we obtain

$$\begin{aligned}\frac{\partial U}{\partial \tau} &= \frac{vU}{2} \frac{\partial^2 U}{\partial X^2} + \frac{\gamma^2 v}{2} \frac{\partial^2 U}{\partial v^2} + \left[ \kappa - \left( \kappa - \frac{\rho\gamma\sigma_L'}{2} \right) v \right] \frac{\partial U}{\partial v} \\ &+ \rho\gamma v \frac{\partial^2 U}{\partial X \partial v} + \frac{1}{8} \left( 2\sigma_L\sigma_L'' - (\sigma_L')^2 \right) U\end{aligned} \quad (5.4)$$

where  $' \triangleq d/dF$

### 5.1.3 Examples of efficient solvers for 2D PDEs in finance

There is a large body of research (from both academics and practitioners) on topic of efficient solvers for 2D PDEs arising in computational finance. Some representative papers are mentioned in the list below:

- Heston: [de Graaf \[2012\]](#)
- SABR: [Sheppard \[2007\]](#), [de Graaf \[2012\]](#)
- Hull-White: [Sheppard \[2007\]](#)
- Heston *LSV*: [Tian et al. \[2013\]](#), [Ait-Haddou \[2013\]](#), [Tian \[2013\]](#)
- quadratic *LSV*: [Lipton et al. \[2014\]](#)

## 5.2 PIDE approach

The generic pricing PIDE for value of option  $U(t, S, Y)$  is given by

$$\begin{aligned}\frac{\partial U}{\partial t} &= \mathcal{L}(t, S, Y) + \mathcal{J}(t, S, Y) \\ U(t, S, Y) &= \text{Payoff}(S)\end{aligned}$$

### 5.2.1 Example: Pricing PIDE for Sepp model

The resulting pricing PIDE is

$$\begin{aligned}\frac{\partial U}{\partial t} &= 0.5 [L(t, S)\vartheta(t, Y)S]^2 \frac{\partial^2 U}{\partial S^2} + \mu(t)S \frac{\partial U}{\partial S} + 0.5\gamma^2\xi^2(Y) \frac{\partial^2 U}{\partial Y^2} \\ &+ \kappa \frac{\partial U}{\partial Y} + \rho\gamma\xi L(t, S)\vartheta(t, Y)S \frac{\partial^2 U}{\partial S \partial Y} \\ &+ \lambda \left[ \int_{\mathbb{R}} \int_{\mathbb{R}} U(Se^J, Y + \mathcal{Y}) \varpi(J) \varsigma(\mathcal{Y}) dJ d\mathcal{Y} - \nu S \frac{\partial U}{\partial S} - U \right]\end{aligned}$$

where jump sizes in  $S$  and  $Y$  have pdf's  $\varpi(J)$  and  $\varsigma(\mathcal{Y})$ , while  $\nu$  is jump compensator

### 5.2.2 Example: Pricing PIDE for exponential Levy model

With notation  $x = \ln \frac{S(t)}{S(0)}$ , the resulting pricing PIDE for an European option (see [Itkin \[2014a\]](#)) of the model given by Eqs. (3.7) and (3.8) :

$$\begin{aligned}\frac{\partial U}{\partial t} &= 0.5v \frac{\partial^2 U}{\partial x^2} + (r - 0.5v) \frac{\partial U}{\partial x} + 0.5\gamma^2\xi^2v^{2\beta} \frac{\partial^2 U}{\partial v^2} \\ &+ \kappa(v_\infty - v) \frac{\partial U}{\partial v} + \gamma\rho\xi v^{\beta+1} L(x, t) \frac{\partial^2 U}{\partial x \partial v} - rC \\ &+ \int_{\mathbb{R}} \left[ U(x + y, v, t) - U(x, v, t) - (e^y - 1) \frac{\partial U}{\partial x} \right] \nu(dy) \\ U(x, v, 0) &= \text{Payoff}(x) \\ v(0) &= v_0\end{aligned}$$

### 5.2.3 Efficient finite difference solvers for 2D PIDEs

Efficient finite difference solvers for 2D PIDEs arising in computational finance were obtained for following models:

- Exponential Levy *LSV*: [Itkin \[2014a\]](#)
- Jump diffusion *LSV*: [Sepp \[2011b,a\]](#)
- Jump-diffusion models with Inverse Normal Gaussian, Hyperbolic and Meixner jumps: [Itkin \[2014b\]](#)

### 5.2.4 PIDEs corresponding to regime switching models

We would like to explicitly mention the pricing PIDEs corresponding to regime switching models which are increasingly used in quant finance [Henrotte \[2012\]](#), [Andersen \[2010\]](#), [Andreasen and Dahlgren \[2006\]](#). Such models reduce complexity by having a finite number of jumps, yet provide a rich enough framework because each regime is a jump-diffusion process with its own level of volatility, jump structure, jump

intensity, etc. They also provide large computational savings, given that for a regime switching model with  $N$  factors and  $M$  regimes we end up solving  $M$  separate PDEs (or PIDEs) of dimension  $N - 1$  instead of one PIDE of dimension  $N$ . Examples of efficient finite difference solvers for 2D PIDEs regime switching include [Andreasen and Dahlgren \[2006\]](#), [Andersen \[2010\]](#).

### 5.3 Efficient Monte Carlo approach

An efficient simulation of a general stochastic-local volatility model is described in [van der Stoep et al. \[2013\]](#). The method is based on approximating a conditional expectation in a non-parametric way which is intuitive and easy to implement. This approximation is embedded in a simulation scheme based on the QE scheme [Andersen \[2007\]](#).

The main difference for Monte Carlo simulation between the pure Heston and the Heston *LSV* models lies in the fact that the variance of the latter is not only driven by the stochastic volatility, but also by the local volatility component, which is state-dependent. This requires an additional "freezing approximation", which is not present in the derivation of the original QE scheme. Numerical experiments on vanilla options and forward starting options show that the additional approximation still yields an accurate simulation scheme.

## 6 Pricing financial options using PDE/PIDE approach

We can divide the financial options into 4 major categories [Wilmott \[2006\]](#):

1. **PATH INDEPENDENT**, e.g., European vanilla or digital options
2. **STRONG PATH DEPENDENT**, such as Asian, lookback or cliquet options. They have payoffs that depend on some property of the asset price path in addition to the value of the underlying at the present moment in time, and their pricing needs at least one more independent variable
3. **WEAK PATH DEPENDENT**, e.g., single and double barrier and touch options; no new independent variable is needed for pricing
4. **EARLY EXERCISE**, such as American and Bermudan options

For all these types of options the pricing PDE/PIDE will have the initial condition given by the payoff of the option, and boundary conditions appropriately given:

- on variance/volatility boundaries due to asymptotic model behavior
- on the boundaries corresponding to underlying due to
  - characteristics of the financial instrument
  - to asymptotic model behavior

### 6.1 Pricing path independent options

Pricing setup is straightforward for European vanilla options. However, we need to pay attention to digital options, due to non-smooth payoff. The reader is referred to Appendix F for details on treatment of these types of payoffs.

## 6.2 Pricing strong path dependent options

In many cases their pricing is done through introduction of additional state variables [Tavella and Randall \[2000\]](#), [Andersen and Piterbarg \[2010\]](#), [Lipton \[2001\]](#).

For example, if we consider a continuously sampled Asian option, its payoff depends on

$$I(t) = \int_0^t h(x(s)) ds$$

Then the value of option  $u = u(t, x(t), I(t))$  satisfies PDE

$$\begin{aligned} \frac{\partial u}{\partial t} + a(x, t) \frac{\partial^2 u}{\partial x^2} + b(x, t) \frac{\partial u}{\partial x} + h(x) \frac{\partial u}{\partial I} &= r(t, x)u \\ u(0, x, I) &= g(x, I) \end{aligned}$$

This PDE is convection dominated in  $I$ -direction, and thus numerical schemes have to be selected accordingly. Additional complications may arise when  $h(x)$  is of different order of magnitude compared to other coefficients.

Discretely sampled Asian option is much more commonly traded instrument, with payoff depending on

$$I(t) = \sum_{i=1}^{Nav} h(x(T_i)) (T_{i+1} - T_i)$$

We incorporate this feature as described in Appendix F, through a jump condition:

$$V(T_{k+1}, X_i, I) = V(T_k, X_i, i + h(X_i)(T_{i+1} - T_i)) \quad (6.1)$$

In-between schedule dates we have  $dI(t) = 0$  and thus we can solve PDE without term involving  $I$

$$\frac{\partial u}{\partial t} + a(x, t) \frac{\partial^2 u}{\partial x^2} + b(x, t) \frac{\partial u}{\partial x} = r(t, x)u$$

In other words, if we assume that  $I$ -direction is discretized with  $N_I$  points, then we need to solve  $N_I$  different 1D PDEs, with information exchanged at  $\{T_k\}$  dates through (6.1), making sure to use a higher order interpolator.

## 6.3 Pricing weak path dependent options

For barrier options which are continuously monitored, the boundaries of spatial grid given by barrier level(s) and min/max attainable value for underlying on  $[0, T]$ , while rebates (i.e., payment if barrier is hit) are implemented as Dirichlet BCs. We can use barrier option parity to price Knock-out/Knock-In.

The pricing approach can be extended [Tavella and Randall \[2000\]](#), [Randall \[2010\]](#), [Andersen and Piterbarg \[2010\]](#) in various ways to cover step-up/step-down barriers

- use different spatial grids over  $[0, T^*]$  and  $[T^*, T]$ , with interpolation (at least third order accurate) from one spatial grid to another
- use same spatial grid, with both  $H$  and  $H^*$  on grid; solve smaller system of equations after time loop reaches  $T^*$

For barrier options discretely monitored we need to allow the option price  $V$  to “diffuse” outside interval given by barrier level(s) [Tavella and Randall \[2000\]](#), [Randall \[2010\]](#), [Andersen and Piterbarg \[2010\]](#).

Barrier jump condition (BJC) have to be imposed at each observation time  $T_k$ . Here is an example for Up-And-Out, with barrier  $H$ :

$$V_i(T_{k+1}) = \begin{cases} V_i(T_k), & \text{if } X_i < H \\ 0, & \text{otherwise} \end{cases}$$

Since BJC likely to produce discontinuity in  $V$  around  $H$ , we need to apply Rannacher time stepping plus grid shifting past each  $T_k$

We can also extend the approach to cover step-up/step-down barriers and rebates. As an example, for Double Knock-Out with barriers  $H_L, H_U$  we have:

$$V_i(T_{k+1}) = \begin{cases} V_i(T_k) + R(T_k, X_i), & \text{if } H_L < X_i < H_U \\ R(T_k, X_i), & \text{otherwise} \end{cases}$$

Touch options have discontinuous payoffs, and thus have a special treatment. The reader is referred to Appendix F for more details.

## 6.4 Pricing early exercise options

Depending on the type of monitoring, the early exercise option can be split into 2 main classes:

- Discrete monitoring, represented by Bermudan options
- Continuous monitoring, represented by American options

If exercise values determined by a deterministic function  $h(t, x)$ , then early exercise options can be easily priced in a PDE/PIDE setting [Toivanen \[2010\]](#), [Salmi and Toivanen \[2012\]](#), [Ikonen and Toivanen \[2009\]](#), [Tavella and Randall \[2000\]](#), [Forsyth and Vetzal \[2002\]](#), [Oosterlee et al. \[2005\]](#), [d' Halluin et al. \[2004\]](#), [Duffy \[2006\]](#), [Andersen and Piterbarg \[2010\]](#). It relies on solving a linear complementarity problem

$$\begin{cases} \frac{\partial u}{\partial t} - \mathcal{L}[u] & \leq 0 \\ u(x, t) & \geq h(t, x) \\ u(T, x) & = g(x) \\ \mathcal{L}[u] \cdot (u(x, t) - h(t, x)) & = 0 \end{cases} \quad (6.2)$$

There are 2 distinct regions when pricing an early exercise option:

1. **CONTINUATION** region: option price satisfies the pricing PDE, and its value is greater than the intrinsic value of the option
2. **EARLY EXERCISE** region: option value is simply equal to the intrinsic value.

This suggests a straightforward way of adapting FD solver to price American options:

- At each time step FD solver produces a solution
- check each value against option intrinsic value; take larger value at each node

This approach is most common in the literature and was found to approximate relatively well the optimal early-exercise behavior.

However, it is not as accurate as the one based on operator splitting Ikonen and Toivanen [2009], Toivanen [2010], which reformulates the problem (6.2) using Lagrange variable  $\lambda$ :

$$\begin{cases} \frac{\partial u}{\partial t} - \mathcal{L}[u] & = \lambda \\ u(x, t) & \geq g(x) \\ u(T, x) & = g(x) \\ \lambda \cdot (u(x, t) - h(t, x)) & = 0 \\ \lambda & \geq 0 \end{cases}$$

Its discrete version becomes

$$\begin{cases} B \cdot U^{(m+1)} & = C \cdot U^{(m)} + \Delta t \lambda^{(m+1)} \\ \lambda^{(m+1)} & \geq 0 \\ U^{(m+1)} & \geq g \\ \lambda^{(m+1)} \cdot (U^{(m+1)} - h) & = 0 \end{cases} \quad (6.3)$$

The operator splitting method has two fractional time steps:

1. predictor-corrector to find  $U^{(m+1)}$  through intermediate variable  $V$
2.  $U^{(m+1)}$  and  $\lambda^{(m+1)}$  are updated to satisfy constraints

Formulas in (6.3) may be interpreted in component-wise manner

$$\begin{cases} U_i^{(m+1)} - V_i - \Delta t (\lambda_i^{(m+1)} - \lambda_i^{(m)}) & = 0 \\ \lambda_i^{(m+1)} & \geq 0 \\ U_i^{(m+1)} & \geq g_i \\ \lambda_i^{(m+1)} (U_i^{(m+1)} - g_i) & = 0 \end{cases}$$

We set the initial value of  $\lambda$  to zero:  $\lambda^{(0)} = 0$ , since  $U_i^{(0)} = g_i$   
After combining the equations we obtain

$$\lambda_i^{(m+1)} (V_i - \Delta t (\lambda_i^{(m+1)} - \lambda_i^{(m)}) - g_i) = 0$$

After some calculations, we end up with 2 cases

- If  $V_i - \lambda_i^{(m)} \Delta t < g_i$

$$\begin{cases} \lambda_i^{(m+1)} = \lambda_i^{(m)} + \frac{g_i - V_i}{\Delta t} \\ U_i^{(m+1)} = 0 \end{cases}$$

- If  $V_i - \lambda_i^{(m)} \Delta t > g_i$

$$\begin{cases} \lambda_i^{(m+1)} = 0 \\ U_i^{(m+1)} = V_i - \lambda_i^{(m)} \Delta t \end{cases}$$

Pricing Bermudan options is similar, with only difference that it has to incorporate discrete monitoring (as discussed in Appendix F).

## 7 Numerical solution of PDEs and PIDEs

For main references of PDE/PIDE in quantitative finance we refer the reader to [Tavella \[2002\]](#), [Tavella and Randall \[2000\]](#), [Duffy \[2006\]](#), [Lipton \[2001\]](#), [Andersen and Piterbarg \[2010\]](#), [Achdou and Pironneau \[2005\]](#), [Cont and Tankov \[2004\]](#), [Itkin \[2014a\]](#), and references herein.

For main references of numerical solution of PDE/PIDE in computational science and engineering we refer the reader to [Strikwerda \[2004\]](#), [Thomas \[2010, 1999\]](#), [Kopriva \[2009\]](#), [LeVeque \[2005, 2007\]](#), [Trefethen \[2001\]](#), [Langtangen \[2003\]](#), [Roache \[1998\]](#), [Li and Chen \[2008\]](#), [Zienkiewicz et al. \[2013\]](#), and references herein.

A general 2D PDE with mixed derivative term could be formulated as

$$\begin{cases} \frac{\partial U}{\partial t} = c_x \frac{\partial U}{\partial x} + c_{xx} \frac{\partial^2 U}{\partial x^2} + c_y \frac{\partial U}{\partial y} + c_{yy} \frac{\partial^2 U}{\partial y^2} + c_{xy} \frac{\partial^2 U}{\partial x \partial y} + c_1 U + c_0 = 0, & \forall 0 \leq t \leq T \\ U(x, y; 0) = U_0(x, y) \end{cases} \quad (7.1)$$

### 7.1 Major types of numerical methods

We describe the major numerical approaches considered in the literature to solve PDEs/PIDEs:

- **FINITE DIFFERENCE METHODS (FDM)**, with major references [Duffy \[2006\]](#), [LeVeque \[2007\]](#), [Tavella and Randall \[2000\]](#), [Thomas \[2010\]](#)
  - local Taylor expansion to approximate the derivatives
  - functions represented by their values at grid points
- **FINITE ELEMENT METHODS (FEM)**, with major references [Topper \[2005\]](#), [Achdou and Pironneau \[2010\]](#), [Zienkiewicz et al. \[2013\]](#)
  - differential equation solved in its integral (weak) form
  - functions are represented in terms of basis functions
  - at first glance unnecessarily complex, but very flexible
- **FINITE VOLUME METHODS (FVM)**, with major references [Versteeg and Malalasekera \[2007\]](#), [Engelmann et al. \[2012\]](#)
  - divides domain into “volumes”, employs “flow rate” across surfaces of “volume”; better suited for hyperbolic PDEs
- **SPECTRAL METHODS (SM)**, with major references [Trefethen \[2001\]](#), [Kopriva \[2009\]](#)
  - functions represented as a sum of basis functions
- **METHOD OF LINES (MoL)**, with major references [Schiesser and Griffiths \[2009\]](#)
  - all but one variable is discretized; solve system of ODEs
- **MULTIGRID**, with major references [Trottenberg et al. \[2000\]](#)
  - solve using a hierarchy of discretizations

## 7.2 Finite difference schemes for PDEs

While we need to discuss primarily 2D PDEs, we would like to start by mentioning 1D schemes regularly used in quantitative finance, depending on the PDE type:

- Diffusion dominated PDE:
  - $\theta$ -scheme: includes explicit, implicit, Crank-Nicolson
  - Alternating Direction Explicit (ADE)
  - Yanenko splitting
  - predictor corrector
- Convection dominated PDE
  - upwinding
  - exponentially fitted
  - flux limiting

We should note that our recommendation does not include the regular  $\theta$ -scheme, although this scheme may be the most commonly described in courses of quantitative finance or in the literature. That is due to the drawbacks of this scheme, in terms of Greeks behavior, requiring the original scheme needs to be enhanced in order to provide satisfactory Greeks (see for example [Giles and Carter \[2006\]](#)). However, the corresponding enhancement may increase the computational cost to a level where other schemes (which already have better stability properties) become quite competitive, and thus there is no compelling reason to use  $\theta$ -scheme in such situations.

Regarding 2D PDEs, they can be divided into 2 main classes:

- without mixed derivatives  $c_{xy} = 0$ :
- with mixed derivatives  $c_{xy} \neq 0$

We note that special attention needs to be paid to the discretization of boundary conditions which incorporate spatial or time derivatives, and that particular treatment choices will greatly affect the performance of the finite difference solver.

The main schemes considered for 2D PDEs are:

- Alternating Direction Implicit (ADI)
- Enhanced Yanenko
- Predictor corrector (which includes Craig Sneyd)

Additional advanced time discretization schemes are described in Appendix H.

We note that the most commonly used schemes in literature and by practitioners are various types of ADI schemes as well as Craig-Sneyd scheme, which belongs to class of predictor-corrector methods. While they work fine for PDE without mixed derivative, for the case of PDE with mixed derivative the Craig-Sneyd scheme is usually considered, given that ADI scheme may not be completely suitable (it requires commutativity between partial derivative operators).

However, it is our experience that Craig-Sneyd cannot handle to full satisfaction some situations, especially for various FX markets with pronounced skew (e.g., USD/JPY or USD/developing world currencies), or for Heston models with parameters which violate Feller condition. In these cases alternative methods, such as enhanced Yanenko or more specialized predictor corrector methods, may need to be considered.

### 7.2.1 Alternating Direction Implicit scheme

We rewrite PDE as

$$\frac{\partial u}{\partial t} + (\mathcal{L}_1 + \mathcal{L}_2) u = f$$

The main idea of Alternating Direction Implicit (ADI) scheme is to split the simultaneous application of two operators (here  $\mathcal{L}_1, \mathcal{L}_2$ ) into 2 sequential operator applications. However, it requires commutativity between operators. The computational advantage of the decomposition is as follows: only 1 operator in LHS  $\implies$  tridiagonal systems of equations  $\implies$  great efficiency gain.

All ADI schemes will alternate the “implicit” directions:

1. first step:  $x$ -direction is fully implicit while  $y$ -direction is fully explicit
2. second step:  $y$ -direction is fully implicit while  $x$ -direction is fully explicit

The order of convergence is  $O(\Delta t^2 + \Delta x^2 + \Delta y^2)$

### 7.2.2 Enhanced Yanenko scheme

The original Yanenko scheme can be described as, with  $\bar{t}_m \triangleq \frac{t_m + t_{m+1}}{2}$

$$\begin{aligned} \frac{V_{i,j} - U_{i,j}^{(m)}}{\Delta t} &= a_1(\bar{t}_m, X_i, Y_j) \delta_X^2 V_{i,j} + b_1(\bar{t}_m, X_i, Y_j) \delta_X V_{i,j} \\ &\quad + a_{12}(\bar{t}_m, X_i, Y_j) \delta_{XY} U_{i,j}^{(m)} \\ \frac{U_{i,j}^{(m+1)} - V_{i,j}}{\Delta t} &= a_2(t_{m+1}, X_i, Y_j) \delta_Y^2 U_{i,j}^{(m+1)} \\ &\quad + b_2(t_{m+1}, X_i, Y_j) \delta_Y U_{i,j}^{(m+1)} + a_{12}(\bar{t}_n, X_i, Y_j) \delta_{XY} V_{i,j} \end{aligned}$$

An enhanced Yanenko scheme (incorporating exponential fitting) was proposed in [Sheppard \[2007\]](#)

$$\begin{aligned} \frac{V_{i,j} - U_{i,j}^{(m)}}{\Delta t} &= \Psi_{i,j}[\bar{t}_m, a_1, b_1, \max(h_X^-, h_X^+)] \delta_X^2 V_{i,j} \\ &\quad + b_1(\bar{t}_m, X_i, Y_j) \delta_X V_{i,j} + a_{12}(\bar{t}_m, X_i, Y_j) \delta_{XY} U_{i,j}^{(m)} \\ \frac{U_{i,j}^{(m+1)} - V_{i,j}}{\Delta t} &= \Psi_{i,j}[t_{m+1}, a_2, b_2, \max(h_Y^-, h_Y^+)] \delta_Y^2 U_{i,j}^{(m+1)} \\ &\quad + b_2(t_{m+1}, X_i, Y_j) \delta_Y U_{i,j}^{(m+1)} + a_{12}(\bar{t}_n, X_i, Y_j) \delta_{XY} V_{i,j} \end{aligned}$$

with

$$\Psi_{i,j}[t, a, b, h] \triangleq \frac{b(t, X_i, Y_j) \cdot h}{2} \coth\left(\frac{b(t, X_i, Y_j) \cdot h}{2 \cdot a(t, X_i, Y_j)}\right)$$

### 7.2.3 Predictor corrector scheme

This scheme incorporates the mixed derivative operator  $\widehat{\mathcal{L}}_{1,2}$  into a regular ADI scheme. The most popular example used by finance practitioners is Enhanced Craig-Sneyd [Craig and Sneyd \[1988\]](#) scheme, with  $\lambda \in [0, 1]$ .

It has order of convergence:  $O\left((\Delta x + \Delta y)^2 + 1_{\{\theta \neq \frac{1}{2}\}} \Delta t + 1_{\{\lambda \neq \frac{1}{2}\}} \Delta t + \Delta t^2\right)$ , and is stable if  $\theta \geq \frac{1}{2}$  and  $\frac{1}{2} \leq \lambda \leq \theta$ . If properly implemented, this scheme was reported to be only 30%-40% slower than Douglas-Rachford ADI scheme [Andersen and Piterbarg \[2010\]](#).

The scheme employs two intermediate variables:  $V$  in predictor step and  $Z$  in corrector step.

The predictor step consists of solving iteratively for  $V$  using  $U^{(m)}$

$$\begin{aligned} (1 - \theta \Delta t \widehat{\mathcal{L}}_1(\tau_m^\theta, y = Y_j)) V_{i,j}^{(1\#)} &= ((1 - \theta) \Delta t \widehat{\mathcal{L}}_1(\tau_m^\theta, y = Y_j)) U_{i,j}^{(m)} \\ &\quad + (1 + \Delta t \cdot \widehat{\mathcal{L}}_2(\tau_m^\theta, x = X_i) + \Delta t \cdot \widehat{\mathcal{L}}_{12}(\tau_m^\theta)) U_{i,j}^{(m)} \\ (1 - \theta \Delta t \widehat{\mathcal{L}}_2(\tau_m^\theta, x = X_i)) V_{i,j}^{(2\#)} &= V_{i,j}^{(1\#)} - \theta \Delta t \widehat{\mathcal{L}}_2(\tau_m^\theta, x = X_i) U_{i,j}^{(m)} \end{aligned}$$

During the corrector step we solve for  $U^{(m+1)}$  using  $V$

$$\begin{aligned} (1 - \theta \Delta t \widehat{\mathcal{L}}_1(\tau_m^\theta, y = Y_j)) Z_{i,j} &= ((1 - \theta) \Delta t \widehat{\mathcal{L}}_1(\tau_m^\theta, y = Y_j)) U_{i,j}^{(m)} \\ &\quad + (1 + (1 - \lambda) \Delta t \cdot \widehat{\mathcal{L}}_{12}(\tau_m^\theta)) U_{i,j}^{(m)} + \lambda \Delta t \cdot \widehat{\mathcal{L}}_{12}(\tau_m^\theta) V_{i,j}^{(2\#)} \\ (1 - \theta \Delta t \widehat{\mathcal{L}}_2(\tau_m^\theta, x = X_i)) U_{i,j}^{(m+1)} &= Z_{i,j} - \theta \Delta t \widehat{\mathcal{L}}_2(\tau_m^\theta, x = X_i) U_{i,j}^{(m)} \end{aligned}$$

We note that the iteration in predictor step may contain more than one step, if necessary.

### 7.3 Galerkin-Ritz method

The Galerkin-Ritz method is one of the most fundamental tools of modern computing. Its origins lie in Hilbert's "direct" approach to the variational calculus of Euler Lagrange and in the thesis of Walther Ritz, who died 100 years ago at the age of 31 after a long battle with tuberculosis [Gander and Wanner \[2012\]](#). It is one of the main fundamental blocks for the class of finite element methods.

This method allowed [Lipton et al. \[2014\]](#) to obtain a good representation of the mixed derivative term without the time-averaging step (as done in the finite different approach) when solving a PDE derived from *LSV* model.

The method relies on choosing a convenient basis  $\{e_k\}$  in, say,  $x$  direction and representing the solution of the PDE through elements of the basis:

$$U(t, x, y) = \sum_{k=1}^{\infty} U_k(t, y) e_k(x)$$

although in practice one does not consider an infinite series, but a truncated one:

$$U(t, x, y) = \sum_{k=1}^M U_k(t, y) e_k(x)$$

A possible choice for the basis is given by the set of eigenfunctions of operator  $\frac{\partial^2}{\partial x \partial y}$  supplied with homogenous boundary conditions at  $x = X_L$  and  $x = X_U$ :

$$\begin{aligned} e_k(x) &= \sin[\xi_k(x - X_L)] \\ \xi_k &\triangleq \frac{\pi k}{X_U - X_L} \end{aligned}$$

We can now think of  $U(t, x, y)$  as a vector function of two variables  $(t, y)$  with vector components parametrized by the index  $k$ .

As a result, the two-factor parabolic PDE is replaced with a coupled system of one-factor parabolic PDEs. This system is solved by treating the cross term fully explicitly, allowing usage the standard technique for solving scalar one-factor parabolic PDEs with non-zero source terms.

If the number of grid point in the  $x$  direction is  $N_X$  and the number of terms in the truncated series is  $M$ , then in general computational savings are of order  $\frac{N_X}{M}$ , since the Galerkin-Ritz method requires solving  $M$  equations, while a typical ADI method requires solving  $N_X$  equations. Moreover, if maturity of the problem under consideration is long, one can safely choose  $M$  to be small given that higher order modes decay very rapidly, and add very little to the overall solution.

For more details the reader is referred to [Lipton et al. \[2014\]](#).

## 7.4 Finite difference scheme for PIDEs

Let us consider a 2D PIDE for  $u = u(t, x, y)$

$$\frac{\partial u}{\partial t} + D_X [u] + Integral_Y [u] = 0$$

Main references for numerical schemes to solve pricing PIDEs include [Tavella and Randall \[2000\]](#), [Voltchkova \[2010\]](#), [Cont and Tankov \[2004\]](#), [Tankov \[2009\]](#), [Andersen and Andreasen \[2000\]](#), [Itkin and Carr \[2009\]](#), [d' Halluin et al. \[2005\]](#), [Cont and Voltchkova \[2005\]](#), [Matache et al. \[2005\]](#)

Schemes depend on how  $J [U^{(m)}]$  is computed, and two main approaches have been considered:

1. obtain  $J$  as solution of separate PDE (e.g., applied to Merton model)
2. obtain  $J$  as numerical approximation (e.g., applied to Kou, VG, CGMY models)

[Itkin \[2014a,b\]](#) and references herein provide a comprehensive overview of methods considered in the literature:

- discretization of the PIDE that is implicit in differential terms and explicit in integral term
- Picard iterations for computing the integral equation
- a second-order accurate, unconditionally-stable operator splitting (ADI) method that does not require an iterative solution of an algebraic equation at each time step
- schemes based on operator splitting

### 7.4.1 Predictor corrector scheme

As the name implies, the scheme relies on two steps

- Predictor step: solve for intermediate  $W$  using  $U^{(m)}$ :

$$\left[ \frac{1}{\Delta t} + \frac{1}{2} D_X^{(m+1)} \right] W = \left[ \frac{1}{\Delta t} + \frac{1}{2} D_X^{(m)} \right] U^{(m)} + J [U^{(m)}]$$

- Corrector step: solve for  $U^{(m+1)}$  using  $W$ :

$$\begin{aligned} \left[ \frac{1}{\Delta t} + \frac{1}{2} D_X^{(m+1)} \right] U^{(m+1)} &= \left[ \frac{1}{\Delta t} + \frac{1}{2} D_X^{(m)} \right] U^{(m)} \\ &+ J \left[ \frac{1}{2} U^{(m)} + \frac{1}{2} W \right] \end{aligned}$$

Additional predictor corrector schemes are presented in [Sepp \[2011b,a\]](#), and include extensions of Craig Sneyd, Hundsdorfer-Verwer and in't Hout-Foulon schemes.

### 7.4.2 Itkin scheme

Second order schemes in both space and time are constructed in Itkin [2014a,b] and applied to models with jumps and discrete dividend. This method has 3 steps:

1. Transform a linear non-local integro-differential operator (jump operator) into a local nonlinear (fractional) differential operator. Thus the jump-diffusion operator is represented as a sum of the linear and non-linear parts.
2. Splitting a space operator into diffusion and jumps parts, providing a second-order approximation in time.
3. Apply finite-difference approximations to the non-linear jump operator.

### 7.5 Boundary conditions

If PDE/PIDE is considered on a spatial domain  $\Omega$ , let us denote the boundary by  $\partial\Omega$ . The boundary conditions for  $x^B \in \partial\Omega$  may be of various types:

- Dirichlet:  $u(x^B, t) = g^B(x^B, t)$
- Neumann:  $\frac{\partial u}{\partial X}(x^B, t) = g^B(x^B, t)$
- Robin:  $\alpha^B(t, X) \cdot u(x^B, t) + \beta^B(x^B, t) \cdot \frac{\partial u}{\partial X}(x^B, t) = g^B(x^B, t)$
- May involve derivatives of order  $\geq 2$ , e.g.  $\frac{\partial^2 u}{\partial X^2}(x, t) = g^B(x^B, t)$

We use Fichera theory to identify subsets of boundary  $\partial\Omega$  where it is necessary or not to explicitly specify BCs. If no BCs are allowed use the degenerated pricing PDE on that boundary.

We also address the special case of specifying boundary conditions for Fokker-Planck (forward Kolmogorov) PDE, which may be quite more challenging compared to identifying boundary conditions for backward PDEs.

#### 7.5.1 Explicit or implicit boundary conditions?

We use Fichera theory for BCs Duffy [2009], Duffy and Kienitz [2009]. We describe it in the general case of a  $N$ -dimensional PDE

The characteristic form  $\Psi$  of the PDE

$$L[u] \triangleq \sum_{i=1}^N \sum_{j=1}^N a_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^N b_i \frac{\partial u}{\partial x_i} + c = f$$

is defined as  $\Psi \triangleq \sum_{i=1}^N \sum_{j=1}^N a_{ij} \xi_i \xi_j$ ,  $\forall \xi = (\xi_1, \dots, \xi_N) \in \mathbb{R}^N$

We are interested in subsets  $\Sigma_{\Psi=0} \subset \Sigma$  on which  $\Psi = 0$

Fichera function is defined as

$$\varphi \triangleq \sum_{i=1}^N \left( b_i - \sum_{k=1}^N \frac{\partial a_{ik}}{\partial x_k} \right) \nu_i \tag{7.2}$$

where  $\nu_i$  is  $i$ -th component of inward normal  $\nu$  on  $\Sigma$ .

Then we have 2 situations:

- On  $\Sigma_{\varphi=0}$  no explicitly BCs are allowed when  $\varphi \geq 0$ ; In this case we use degenerate PDE
- We explicitly give BC when  $\varphi < 0$

For a parabolic PDE we focus on elliptic part given by  $L[u]$  and use same reasoning  
 Here is an example from [Duffy \[2009\]](#) which relies on pricing PDE in CIR model

$$\frac{\partial B}{\partial t} + \frac{1}{2}\sigma^2 r \frac{\partial^2 B}{\partial r^2} + (a - cr) \frac{\partial B}{\partial r} - rB = 0 \quad (7.3)$$

The Fichera function given by (7.2) has the formula

$$\varphi = ((a - cr) - 0.5\sigma^2) \nu$$

Then  $\nu$  is inward unit normal at  $r = 0$  ( $\nu = 1$ ) and at  $r = 1$  ( $\nu = -1$ )

We only consider the characteristic boundary at  $r = 0$

$$\begin{aligned} \Sigma_2 : \varphi < 0 &\rightarrow \sigma > \sqrt{2a} \text{ (explicitly given BC)} \\ \Sigma_0 : \varphi \leq 0 &\rightarrow \sigma = \sqrt{2a} \text{ (Degenerate PDE)} \\ \Sigma_1 : \varphi > 0 &\rightarrow \sigma < \sqrt{2a} \text{ (Degenerate PDE)} \end{aligned}$$

The degenerate PDE is obtained by setting  $r = 0$  in (7.3):

$$\frac{\partial B}{\partial t} + a \frac{\partial B}{\partial r} = 0$$

[Ekstrom et al. \[2009\]](#), [Ekstrom and Tysk \[2011\]](#) provide additional examples of such endogenous boundary conditions, directly obtained from PDE.

### 7.5.2 Boundary conditions for forward Kolmogorov PDE

For some models, especially ones based on CIR processes, resulting PDEs may have numerical difficulties related to implementation of reflecting boundary at zero. In the absence of an absorbing mass of zero, it is typically acceptable (although sometimes inaccurate) to impose a condition of zero second derivatives at the boundary. However, a better method is described in [Lucic \[2008\]](#), based on application of a zero flux procedure.

For the backward PDEs arising in finance this problem is not as severe, and is typically handled by dropping out some of the terms of the original PDE at the boundary. The situation changes, however, for Fokker-Planck (forward) PDEs, where additional care is needed, due to potentially explosive nature of the density of the stochastic process in the vicinity of the origin.

[Lucic \[2008\]](#) extends the Feller's result from one dimension to two dimensions. The original result was applied to a 1D PDE of the form

$$\begin{aligned} \frac{\partial p}{\partial t} &= 0.5\eta^2 \frac{\partial^2}{\partial v^2} (vp) - \frac{\partial}{\partial v} (\lambda(\bar{v} - v)p) \\ p(0, v) &= \delta(v - v_0) \end{aligned} \quad (7.4)$$

which arises as the Fokker-Planck equation for the well-known CIR stochastic process

$$dv(t) = \lambda(\bar{v} - v(t)) dt + \eta\sqrt{v(t)}dW(t)$$

Then the origin is (instantaneously) reflecting, regular, and attainable if  $\eta^2 > 2\lambda\bar{v}$ , and unattainable, non-attracting boundary otherwise. This is the well-known Feller condition.

It was shown that Eq. (7.4) has the density of  $v(t)$  as the unique solution when so-called "zero flux" boundary condition  $f(t) = 0$  is imposed, where flux  $f(t)$  at boundary  $v = 0$  is defined as

$$f(t) \triangleq \left[ 0.5\eta^2 \frac{\partial}{\partial v} (vp) - \lambda(\bar{v} - v)p \right] \Big|_{v=0}$$

This relationship is derived by taking the time derivative (and changing the order of integration and differentiation) of the formula  $\int_0^\infty p(t, y) dy = 1$ , and relying on the fact that

$$\lim_{y \rightarrow \infty} p(y) = \lim_{y \rightarrow \infty} \frac{\partial p}{\partial y} = 0$$

This reasoning can be extended by analogy to obtain the following zero flux boundary condition for the forward PDE of *LSV* model (3.2):

$$0.5\eta^2 \frac{\partial}{\partial V} (V \cdot p) - \kappa p (\theta(t) - V) + \gamma \rho \xi V \frac{\partial}{\partial S} (L(t, S) \cdot p) = 0$$

at  $p(x, V, t) |_{V=0}$ , as given in chapter 4 of [Lucic \[2008\]](#).

## 7.6 (Semi-) infinite spatial domain

Two main approaches are considered in the literature [Tavella and Randall \[2000\]](#), [Tavella \[2002\]](#), [Duffy \[2006\]](#), [Andersen and Piterbarg \[2010\]](#), [Lipton \[2001\]](#):

- domain truncation
- transformation of variables.

We describe next details of each approach.

### 7.6.1 Domain truncation

The values of  $\pm\infty$  are replaced in computational grid by probabilistic lower/upper limits, defined as lowest/highest attainable value of underlying  $S(t)$  on time interval given by option expiry. We use exact confidence interval if available, otherwise approximate confidence interval based on “average” values for drift and diffusion components of SDE.

We give an example from [Andersen and Piterbarg \[2010\]](#), for the Black-Scholes-Merton model defined by  $r, \sigma$ . The infinite interval is transformed into the following numerical domain:

$$(-\infty, \infty) \Rightarrow \left[ \bar{S} - \alpha\sigma\sqrt{T}, \bar{S} + \alpha\sigma\sqrt{T} \right], \alpha \in [3, 5]$$

We have a probability of  $2\Phi(-\alpha)$  for underlying falling outside interval  $\left[ \bar{S} - \alpha\sigma\sqrt{T}, \bar{S} + \alpha\sigma\sqrt{T} \right]$ , where  $\Phi(\cdot)$  is the normal distribution function.

For the general PDE we find an approximate confidence interval. One possibility is to rely on “average” values for drift  $\mu(X, t)$  and diffusion  $\sigma(X, t)$  functions of SDE describing underlying dynamics:

$$dX(t) = \mu(X, t)dt + \sigma(X, t)dW$$

### 7.6.2 Transformation of variables

Reference [Duffy \[2009\]](#) transforms PDE on semi-infinite interval into a PDE on unit interval  $[0, 1]$ , arguing that it allows more clarity for imposing BCs at transformed near and far fields. Here are some examples of transformations

$$\begin{aligned} y &= \frac{x}{x + \alpha} \\ y &= \tanh x \\ y &= \frac{1}{1 + \alpha x} \\ y &= e^{-\alpha x} \end{aligned}$$

## 7.7 Spatial grid

We consider the spatial grid  $x_L = X_1 < X_2 < \dots < X_{N-1} < X_N = x_U$ . We concentrate on the nonuniform grid (since uniform grid is a special case), which is needed:

- to align spatial grid to particular values (barriers, strike...), for increased stability for Greeks
- to concentrate grid points in regions important for PDE
- to reduce computational cost

Construction of non-uniform spatial grid is described in Appendix D.

## 7.8 Temporal grid

We partition the time interval  $[0, T]$  into  $M$  grid points  $t_1 = 0 < t_1 < \dots < t_M = T$ , and denote

$$\Delta t_m \triangleq t_{m+1} - t_m$$

The simplest type is based on having equally spaced time points. Alternatively we can use adaptive time stepping, to concentrate timesteps around each of important trade dates  $\{0, T_{Trade}^{(1)}, \dots, T_{Trade}^{(NT)}\}$ . We exemplify with a formula from section 2.8.1 of ?

$$T_j = \frac{T}{\chi^{K-j}}, \quad j = 1 \dots K$$

with typical values  $K = 3, \chi = 4$ . Then the time grid for interval  $[T_j, T_{j+1}]$  is given as

$$t_i^j = T_j + i \frac{T_{j+1} - T_j}{\lfloor M/K \rfloor}$$

while important trade dates falling into  $[T_j, T_{j+1}]$  were also added.

Time stepsize is controlled in Forsyth and Vetzal [2002] as:

$$\Delta t_m = C \left( \min_i \left[ \frac{\max \left\{ \|U_j^{m-1}\|, \|U_j^m\|, D \right\}}{\|U_j^m - U_j^{m-1}\|} \right] \right) \Delta t_{m-1}$$

with  $C$  target relative change,  $D$  scale of option value.

## 8 AAD for speed and more speed

*AAD* stands for Adjoint Automatic Differentiation or for Adjoint Algorithmic Differentiation. It is a very powerful computer science technique that was applied to various areas of computational science in the last 25 years, and recently became one of the main tools employed by quant finance practitioners to greatly reduce the computational speed (even by several orders of magnitude) for both calibration and calculation of risk. We include a list of references related to *AAD* in finance Capriotti and Giles [2012, 2011, 2010], Capriotti and Lee [2013], Homescu [2011a], Henrard [2013], Reghai [2014], Savine [2014], Andreasen [2014], Denson and Joshi [2010], Joshi and Yang [2010].

**AUTOMATIC DIFFERENTIATION (AD)**, also known as Algorithmic Differentiation, is a chain-rule-based technique for evaluating the derivatives with respect to the input variables of functions defined by a high-level language computer program. **AD** relies on the fact that all computer programs, no matter how complicated, use a finite set of elementary (unary or binary, e.g.  $\sin(\cdot)$ ,  $\text{sqrt}(\cdot)$ ) operations as defined by the programming language. The value or function computed by the program is

simply a composition of these elementary functions. The partial derivatives of the elementary functions are known, and the overall derivatives can be computed using the chain rule.

**AD** has two basic modes of operations, the forward mode and the reverse mode. In the forward mode the derivatives are propagated throughout the computation using the chain rule, while the reverse mode computes the derivatives for all intermediate variables backwards (i.e., in the reverse order) through the computation. In the literature, **AD** forward mode is sometimes referred to as *tangent linear* mode, while **AD** reverse mode is denoted as *adjoint* mode.

The adjoint method is advantageous for calculating the sensitivities of a small number of outputs with respect to a large number of input parameters. The forward method is advantageous in the opposite case, when the number of outputs (for which we need sensitivities) is larger compared to the number of inputs.

When compared to regular methods (such as finite differencing) for computing sensitivities, **AD** has 2 main advantages: reduction in computational time and computing the results up to machine precision. The reduction in computational time is assured by a theoretical result [Griewank and Walther \[2008\]](#) that states that the cost of the reverse mode is smaller than five times the computational cost of a regular run. The computational cost of the adjoint approach is independent of the number of inputs for which we want to obtain the sensitivities with respect to, whereas the cost of the tangent linear approach increases linearly with the number of inputs. Regarding accuracy, **AD** computes the derivatives exactly (up to machine precision) while finite differences incur truncation errors. The size of the step  $h$  needed for finite difference varies with the current value of input parameters, making the problem of choosing  $h$ , such that it balances accuracy and stability, a challenging one [Homescu \[2011c\]](#). **AD** on the other hand, is automatic and time need not be spent in choosing step-size parameters, etc.

The **AD** approach is generic, in the sense that it can be applied in conjunction with any analytical or numerical method (finite difference, Monte Carlo, etc) used for pricing, preserving the numerical properties of the original method (stability, convergence, accuracy etc). While its development requires some effort, its main advantages (accuracy up to machine precision and, respectively, speedup by one or more orders of magnitude) justify the development cost in our point of view.

The 2 main areas that *AAD* is deployed in quantitative finance are:

1. calibration
2. risk calculation

## 8.1 *AAD* for calibration

Since *AAD* is extremely powerful especially when very many sensitivities have to be calculated, one can imagine how *AAD* can be extremely useful towards reducing the computational cost when the calibration procedure relies on optimization algorithms which are gradient (and possibly Hessian) based.

An even more powerful application of *AAD* within the calibration framework was presented in [Henrard \[2011, 2013\]](#), and relies on combining the adjoint approach and the **IMPLICIT FUNCTION THEOREM**. It is applicable for the 2 main categories of calibration procedures:

- solve a nonlinear system of equations
- minimize a cost functional

For details the reader is referred to [Henrard \[2013\]](#).

## 8.2 AAD for risk calculation

*AAD* can be employed to compute both first order and second order Greeks, such as deltas, vegas, correlation vegas (cegas), rhos, gammas, volgas, credit sensitivities, counterparty credit risk etc. Assuming we have to compute risk sensitivities with respect to  $N$  quantities, *AAD* can compute the corresponding  $N$  first order Greeks with a computational cost that is no more than  $10 \cdot \mathfrak{S}$ , where  $\mathfrak{S}$  is the computer time needed for a single price calculation, while the  $N^2$  second order Greeks are obtained in no more than  $10 \cdot N \cdot \mathfrak{S}$ .

It might be easier to visualize the computational savings by exemplifying with some numbers. Let us assume that we calibrate a *LSV* model to a matrix of market data volatilities with 40 expiries and 9 strikes and that the computational time for a single price calculation with calibrated *LSV* model is 0.12 seconds. If we compute the corresponding vegas using central finite difference approximation, the total computational cost will be  $40 \times 9 \times 0.12 = 43.2$  seconds. If we compute the corresponding vegas with *AAD*, total computational cost will not exceed  $10 \times 0.12 = 1.2$  seconds.

## 9 Conclusion

We have analyzed in detail calibration and pricing performed within the framework of local stochastic volatility *LSV* models, which have become the industry market standard for FX and equity markets. We have presented the main arguments for the need of having such models, and have addressed the question whether jumps have to be included. We include a comprehensive literature overview, and focus our exposition on important details related to calibration procedures and option pricing using PDEs or PIDEs derived from *LSV* models.

We have described in detail calibration procedures, with special attention given to usage and solution of corresponding forward Kolmogorov PDE/PIDE, and have outlined powerful algorithms for estimation of model parameters. Emphasis was placed on practical details regarding the setup and the numerical solution of both forward and backward PDEs/PIDEs obtained from the *LSV* models. Consequently we have discussed specifics (based on our experience and best practices from literature) regarding choice of boundary conditions, construction of nonuniform spatial grids and adaptive temporal grids, selection of efficient and appropriate finite difference schemes (and possible enhancements), etc. We have also shown how to practically integrate specific features of various types of financial instruments within calibration and pricing settings.

We have considered all questions and topics identified as most relevant during the selection, calibration and pricing procedures associated with local stochastic volatility models, providing answers (to the best of our knowledge), and have presented references for deeper understanding and for additional perspectives. In a nutshell, it was our intention to present here an effective roadmap for a successful *LSV* journey.

## Bibliography

- Y. Achdou and O. Pironneau. *Computational methods for option pricing*. SIAM, 2005.
- Y. Achdou and O. Pironneau. Finite element methods. In *Encyclopedia of Quantitative Finance*. Wiley, 2010.
- X. Ait-Haddou. Stochastic local volatility and high performance computing. Master's thesis, University of Manchester, 2013.
- L. Andersen. Markov models for commodity futures: theory and practice. *Quantitative Finance*, 10(8): 831–854, 2010.
- L. B. G. Andersen and J. Andreasen. Jump-diffusion processes: Volatility smile fitting and numerical methods for pricing. *Review of Derivatives Research*, 4:231–262, 2000.
- L.B.G. Andersen. Efficient Simulation of the Heston Stochastic Volatility Model. Available at SSRN: <http://ssrn.com/abstract=946405>, 2007.
- L.B.G. Andersen and N.A. Hutchings. Parameter Averaging of Quadratic SDEs With Stochastic Volatility. Available at SSRN: <http://ssrn.com/abstract=1339971>, 2009.
- L.B.G. Andersen and V. V. Piterbarg. *Interest Rate Modeling Volume I: Foundations and Vanilla Models*. Atlantic Financial Press, 2010.
- J. Andreasen. How To Do CVA Calculations On Your iPad Mini. In *Global Derivatives Conference, Amsterdam*, 2014.
- J. Andreasen and M. Dahlgren. At the flick of a switch. *Energy Risk*, February:71–75, 2006.
- J. Andreasen and B. Hugel. Volatility interpolation. *Risk magazine*, pages 76–79, March 2011.
- D. Ardia, J. David, Os. Arango, and N.D. Giraldo Gomez. Jump-diffusion calibration using differential evolution. *Wilmott Magazine*, pages 76–79, 2011.
- E. Benhamou, E. Gobet, and M. Miri. Time dependent Heston model. *SIAM Journal of Financial Mathematics*, 1:289–325, 2010.
- D. A. Bloch and C. A. Coello Coello. Smiling at evolution. Technical report, 2010.
- L. Bloechlinger. *Power prices– A regime switching spot/forward model with Kim filter estimation*. PhD thesis, Univ of St Gallen, 2008.
- R. Bompis. *Stochastic expansion for the diffusion processes and applications to option pricing*. PhD thesis, Ecole Polytechnique, 2013.
- M Burger, B. Graeber, and G. Schindlmayr. *Managing energy risk: an integrated view on power and other energy markets*. Wiley, 2008.
- L. Capriotti and M. Giles. Fast correlation greeks by adjoint algorithmic differentiation. *Risk Magazine*, April, 2010.
- L. Capriotti and M.B. Giles. Algorithmic differentiation: Adjoint Greeks made easy. Available at SSRN: <http://ssrn.com/paper=1801522>, 2011.
- L. Capriotti and M.B. Giles. Adjoint Greeks made easy. *Risk Magazine*, 2012.

- L. Capriotti and S. Lee. Adjoint credit risk management. Available at SSRN: <http://ssrn.com/abstract=2342573>, 2013.
- J.H. Chan and M. Joshi. Fast Monte-Carlo Greeks for Financial Products with Discontinuous Pay-Offs. *Mathematical Finance*, 23(3), 2013.
- S.-Y. Choi, J.-P. Fouque, and J.-H. Kim. Option pricing under hybrid stochastic and local volatility. Available at [http://www.pstat.ucsb.edu/faculty/fouque/PubliFM/CFK\(QF12\\_06\).pdf](http://www.pstat.ucsb.edu/faculty/fouque/PubliFM/CFK(QF12_06).pdf), 2012.
- I. Clark. *Foreign Exchange Option Pricing: A Practitioners Guide*. Wiley, 2011.
- R. Cont and P. Tankov. *Financial modeling with jump processes*. Chapman and Hall, 2004.
- R. Cont and E. Voltchkova. A finite difference scheme for option pricing in jump diffusion and exponential Levy models. *SIAM J. Numer. Anal.*, 43:1596–1626, 2005.
- J. Craig and A. Sneyd. An Alternating Direction Implicit Scheme for Parabolic Equations with Mixed Derivatives. *Computers and Mathematics with Applications*, 16:341–350, 1988.
- J. Crosby. Practicalities of pricing exotic derivatives. Available at [http://www.john-crosby.co.uk/pdfs/JCrosby\\_OxfordJune2013\\_Exotics.pdf](http://www.john-crosby.co.uk/pdfs/JCrosby_OxfordJune2013_Exotics.pdf), 2013.
- Y. d’Halluin, P. Forsyth, and G. Labahn. A penalty method for American options with jump diffusion processes. *Numer. Math.*, 97:321–352, 2004.
- Y. d’Halluin, P. Forsyth, and K. Vetzal. Robust numerical methods for contingent claims under jump diffusion processes. *IMA J. Numer. Anal.*, 25:87–112, 2005.
- C.S.L. de Graaf. Finite difference methods in derivatives pricing under stochastic volatility models. Master’s thesis, Leiden University, 2012.
- G. Deelstra and G. Rayee. Local volatility pricing models for long-dated FX derivatives. Available at Arxiv: <http://arxiv.org/abs/1204.0633>, 2012.
- N. Denson and M. Joshi. Fast Greeks for Markov-functional models using adjoint PDE methods. Available at SSRN: <http://ssrn.com/paper=1618026>, 2010.
- R. T. Des Combes. *Non-Parametric Model Calibration in Finance*. PhD thesis, Ecole Polytechnique, 2011.
- D. Duffy and J. Kienitz. *Monte Carlo Frameworks: Building Customisable High-performance C++ Applications*. Wiley, 2009.
- D. J. Duffy. *Finite Difference Methods in Financial Engineering: A Partial Differential Equation Approach*. Wiley Finance, 2006.
- D. J. Duffy. Unconditionally Stable and Second-Order Accurate Explicit Finite Difference Schemes Using Domain Transformation: Part I One-Factor Equity Problems. Available at SSRN: <http://ssrn.com/abstract=1552926>, 2009.
- B. Durrain and M. Fournie. High-order compact finite difference scheme for option pricing in stochastic volatility models. *J. Comput. Appl. Math.*, 2012.
- E. Ekstrom and J. Tysk. Boundary conditions for the single-factor term structure equation. *Ann. Appl. Prob.*, 21:332–350, 2011.

- E. Ekstrom, P. Lotstedt, and J. Tysk. Boundary values and finite difference methods for the single factor term structure equation. *Appl. Math. Finance*, 16:253–259, 2009.
- B. Engelmann, F. Koster, and D. Oeltz. Calibration of the Heston local stochastic volatility model: A finite volume scheme. Available at SSRN: <http://ssrn.com/abstract=1823769>, 2012.
- C. Erlwein. *Applications of hidden Markov models in financial modeling*. PhD thesis, Brunel University, 2008.
- FINCAD. Calibrating financial models using differential evolution. Available at <http://www.fincad.com/derivatives-resources/articles/financial-model-calibration.aspx>, 2007.
- P. Forsyth and K. Vetzal. Quadratic convergence of a penalty method for valuing American options. *SIAM J. Sci. Comp.*, 23:2095–2122, 2002.
- M. J. Gander and G. Wanner. From Euler, Ritz, and Galerkin to Modern Computing. *SIAM Review*, 54(4):627–666, 2012.
- J. Gatheral. *The Volatility Surface: A practitioner's guide*. Wiley Finance, 2006.
- M.B. Giles. Vibrato Monte Carlo sensitivities. In *Monte Carlo and Quasi Monte Carlo Methods 2008*, pages 369–382. Springer, 2008. URL <http://people.maths.ox.ac.uk/gilesm/files/mcqm08.pdf>.
- M.B. Giles and R. Carter. Convergence analysis of Crank-Nicolson and Rannacher time-marching. *Journal of Computational Finance*, 9(4):89–112, 2006.
- M. Gilli and E. Schumann. Calibrating the Heston Model with Differential Evolution. In *Applications of Evolutionary Computation*, pages 242–250. 2010.
- A. Griewank and A. Walther. *Evaluating derivatives : principles and techniques of algorithmic differentiation, 2nd edition*. SIAM, 2008.
- J. Guyon and P. Henry-Labordere. The Smile Calibration Problem Solved, 2011.
- J. Guyon and P. Henry-Labordere. *Nonlinear Option Pricing*. CRC Press, 2013.
- I. Halperin and A. Itkin. USLV: Unspanned Stochastic Local Volatility Model. Available at Arxiv: <http://arxiv.org/abs/1301.4442>, 2013.
- M. Henrard. Calibration in finance: Very fast greeks through algorithmic differentiation and implicit function. In *International Conference on Computational Science*, 2013.
- M.P.A. Henrard. Adjoint algorithmic differentiation: Calibration and implicit function theorem. Available at SSRN: <http://ssrn.com/abstract=1896329>, 2011.
- P. Henrotte. Which model for equity derivatives? *Risk Magazine*, 2012.
- P. Henry-Labordere. A general asymptotic implied volatility for stochastic volatility models. <http://dx.doi.org/10.2139/ssrn.698601>, April 2005. URL <http://ssrn.com/abstract=698601>.
- P. Henry-Labordere. Calibration of local stochastic volatility models to market smiles: A Monte-Carlo approach. *Risk Magazine*, September, 2009.
- C. Homescu. Adjoints and automatic (algorithmic) differentiation in computational finance. [papers.ssrn.com/sol3/papers.cfm?abstract\\_id=1828503](http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1828503), 2011a.

- C. Homescu. Implied volatility surface: Construction methodologies and characteristics. Technical report, Wells Fargo Securities, July 2011b. URL <http://ssrn.com/abstract=1882567>.
- C. Homescu. Generic computing alternatives for better greeks. Available at SSRN: <http://ssrn.com/abstract=1921085>, 2011c.
- S. Ikonen and J. Toivanen. Operator splitting methods for pricing american options under stochastic volatility. *Numerische Mathematik*, 113(2):299–324, 2009.
- A. Itkin. High-Order Splitting Methods for Forward PDEs and PIDEs. Available at Arxiv: <http://arxiv.org/abs/1403.1804>, 2014a.
- A. Itkin. Splitting and Matrix Exponential approach for jump-diffusion models with Inverse Normal Gaussian, Hyperbolic and Meixner jumps. Available at Arxiv: <http://arxiv.org/abs/1405.6111>, 2014b.
- A. Itkin and P. Carr. Using pseudo-parabolic equations for option pricing in jump diffusion models. In *Global Derivatives Conference, Rome*, 2009.
- P. Jackel and C. Kahl. Hyp hyp hooray! Available at <http://www.mth.kcl.ac.uk/research/finmath/semfiles/jaeckel.pdf>, 2007.
- P. Jackel and C. Kahl. Hyp hyp hooray. Available at <http://www.pjaeckel.webspace.virginmedia.com/HypHypHooray.pdf>, 2010.
- M. Jex, R. Henderson, and D. Wang. Pricing exotics under the smile. *Risk*, (November):72–75, 1999.
- M. Joshi and C. Yang. Algorithmic Hessians and the fast computation of cross-Gamma risk. Available at SSRN: <http://ssrn.com/paper=1626547>, 2010.
- Joerg Kienitz and Daniel Wetterau. *Financial Modelling: Theory, Implementation and Practice with MATLAB Source*. Wiley, 2012.
- C. Kim and C. Nelson. *State space models with regime switching*. MIT Press, 1999.
- D. Kopriva. *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*. Springer, 2009.
- H.P. Langtangen. *Computational Partial Differential Equations*. Springer, 2003.
- F. Le Flo'h. Tr-bdf2 for fast stable american option pricing. *Journal of Computational Finance*, 17(3):31–56, 2014a.
- F. Le Flo'h. Fourier integration and stochastic volatility calibration. Available at SSRN: <http://ssrn.com/abstract=2362968>, 2014b.
- F. Le Flo'h and G.J. Kennedy. *Finite Difference Techniques for Arbitrage Free SABR*, 2014.
- S.K. Lele. Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103:16–42, 1992.
- R. LeVeque. *Numerical Methods for Conservation Laws*. Birkhauser, 2005.
- R. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM, 2007.

- J. Li and Y.-T. Chen. *Computational Partial Differential Equations Using MATLAB*. CRC Press, 2008.
- A. Lipton. *Mathematical Models for Foreign Exchange*. World Scientific, 2001.
- A. Lipton. The vol smile problem. *Risk Magazine*, (February):61–65, 2002.
- A. Lipton and W. McGhee. Universal barriers. *Risk magazine*, 2002.
- A. Lipton and A. Sepp. Filling the gaps. *Risk*, pages 78–83, October 2011.
- A. Lipton, A. Gal, and A. Lasis. Pricing of vanilla and first generation exotic options in the local stochastic volatility framework: survey and new results. *Quantitative Finance*, 2014.
- M. Lorig, S. Pagliarani, and A. Pascucci. Explicit implied vols for multifactor local-stochastic vol models. Available at SSRN: <http://ssrn.com/abstract=2283874>, 2014.
- V. Lucic. Boundary Conditions for Computing Densities in Hybrid Models via PDE Methods. Available at SSRN: <http://ssrn.com/abstract=1191962>, 2008.
- A. Matache, C Schwab, and T. Wihler. Fast numerical solution of parabolic integrodifferential equations with applications to finance. *SIAM J. Sci Comput.*, 27:369–363, 2005.
- S. Mitra. Regime switching volatility calibration by the Baum-Welch method. Available at Arxiv: <http://arxiv.org/abs/0904.1500>, 2009.
- Murex. Trust is good, control is better: Complex model validation. *Risk*, (October), 2011.
- Numerix. Numerix keeps focus on cutting-edge models for dominant pricing in latest crossasset release. Available at [http://www.numerix.com/Press-Detail/r/PRESS\\_RELEASES-1120](http://www.numerix.com/Press-Detail/r/PRESS_RELEASES-1120), 2013.
- C Oosterlee, C Leentvaar, and X. Huang. Accurate American option pricing by grid stretching and high order finite differences. Available at [http://ta.twi.tudelft.nl/users/vuik/numanal/huang\\_paper.pdf](http://ta.twi.tudelft.nl/users/vuik/numanal/huang_paper.pdf), 2005.
- S. Pagliarani and A Pascucci. Local stochastic volatility with jumps. Available at [http://www.dm.unibo.it/~pascucci/web/Ricerca/PDF/42\\_PP\\_SLV.pdf](http://www.dm.unibo.it/~pascucci/web/Ricerca/PDF/42_PP_SLV.pdf), 2012.
- V. V. Piterbarg. Markovian projection for volatility calibration. *Risk Magazine*, 20:84–89, 2007.
- D. M. Pooley, K. R. Vetzal, and P. A. Forsyth. Convergence remedies for non-smooth payoffs in option pricing. *J. Comp. Fin*, 6(4):25–40, 2003.
- C. Randall. Finite difference methods for barrier options. In *Encyclopedia of Quantitative Finance*. 2010.
- A. Reghai. AAD applications for pricing and hedging Applications : Cega & American and CVA. In *Financial Risk Conference, Paris*, 2014.
- A. Reghai, G. Boya, and G. Vong. Local volatility: Smooth calibration and fast usage. Available at SSRN: <http://ssrn.com/abstract=2008215>, 2012a.
- A. Reghai, V. Klaeyle, and A. Boukhaffa. LSV Models with a Mixing Weight. Available at SSRN: <http://ssrn.com/abstract=2008207>, 2012b.
- Y. Ren, D. Madan, and M. Qian Qian. Calibrating and pricing with embedded local volatility models. *Risk Magazine*, September:138–143, 2007.

- P.J. Roache. *Fundamentals of Computational Fluid Dynamics*. Hermosa, 1998.
- F.D. Rouah. *The Heston Model and its Extensions in Matlab and C#*. Wiley, 2013.
- S. Salmi and J. Toivanen. Comparison and Survey of Finite Difference Methods for Pricing American Options under Finite Activity Jump-diffusion Models. *International Journal of Computer Mathematics*, 89(9):1112–1134, 2012.
- A. Savine. Algorithmic Differentiation And Its Applications In Finance. In *Global Derivatives Conference, Amsterdam*, 2014.
- D. Schatz. *Robust Calibration of the Libor Market Model and Pricing of Derivative Products*. PhD thesis, University of Ulm, 2011.
- W. E. Schiesser and G. W. Griffiths. *A Compendium of Partial Differential Equation Models: Method of Lines Analysis with Matlab*. Cambridge University Press, 2009.
- A. Sepp. Parametric and non-parametric local volatility models: Achieving consistent modeling of vix and equities derivatives. In *Quant Congress Europe*, 2011a.
- A. Sepp. Efficient Numerical PDE Methods to Solve Calibration and Pricing Problems in Local Stochastic Volatility Models. In *Global Derivatives*, 2011b.
- R. Sheppard. Pricing equity derivatives under stochastic volatility : A partial differential equation approach. Master’s thesis, University of the Witwatersrand, 2007.
- K. Shiraya and A. Takahashi. Pricing Basket Options Under Local Stochastic Volatility with Jumps. Available at SSRN: <http://ssrn.com/abstract=2372460>, 2013.
- R.K. Shukla and X. Zhong. Derivation of high-order compact finite difference schemes for non-uniform grid using polynomial interpolation. *Journal of Computational Physics*, 204:404–429, 2005.
- J. Strikwerda. *Finite Difference Schemes and Partial Differential Equations, 2nd edition*. SIAM, 2004.
- C.C. Tan. *Market Practice in Financial Modelling*. World Scientific, 2012.
- P. Tankov. Pricing and hedging in exponential levy models: review of recent results. [www.math.jussieu.fr/~tankov/explevy\\_pplnmf.pdf](http://www.math.jussieu.fr/~tankov/explevy_pplnmf.pdf), 2009.
- G. Tataru and T. Fisher. Stochastic local volatility. Technical report, Bloomberg, 2010.
- D. Tavella. *Quantitative Methods in Derivatives Pricing: An Introduction to Computational Finance*. Wiley, 2002.
- D. Tavella and C. Randall. *Pricing Financial Instruments: The Finite Difference Method*. John Wiley & Sons, New York, 2000.
- J.W. Thomas. *Numerical Partial Differential Equations: Conservation Laws and Elliptic Equations*. Springer, 1999.
- J.W. Thomas. *Numerical Partial Differential Equations: Finite Difference Methods*. Springer, 2010.
- Y. Tian. *Hybrid Stochastic-Local Volatility Model with Applications in Pricing FX Options*. PhD thesis, Monash University, 2013.
- Y. Tian, Z. Zhu, F. Klebaner, and K. Hamza. Calibrating and pricing with stochastic-local volatility model. Available at SSRN: <http://ssrn.com/abstract=2182411>, 2013.

- J. Toivanen. Finite difference methods for early exercise options. In *Encyclopedia of Quantitative Finance*. Wiley, 2010.
- J. Topper. *Financial Engineering with Finite Elements*. Wiley, 2005.
- L. Trefethen. *Spectral Methods in MATLAB (Software, Environments, Tools)*. SIAM, 2001.
- U. Trottenberg, C.W. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, 2000.
- A. W. van der Stoep, L. A. Grzelak, and C. W. Oosterlee. The Heston stochastic-local volatility model: Efficient Monte Carlo simulation. Available at SSRN: <http://ssrn.com/abstract=2278122>, 2013.
- H. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Prentice Hall, 2007.
- I. Vollrath and J. Wendland. Calibration of interest rate and option models using differential evolution. 2009.
- VolMaster. Volmaster pricing models. Available at <http://www.volmaster.com/Models.aspx>, 2014.
- E. Voltchkova. Partial integro-differential equations. In *Encyclopedia of Quantitative Finance*. Wiley, 2010.
- R. Weron. *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach*. Wiley, 2006.
- R. Weron. Heavy-tails and regime-switching in electricity prices. *Mathematical Methods of Operations Research*, 69(3), 2009.
- P. Wilmott. *Paul Wilmott on Quantitative Finance*. Wiley, 2006.
- U. Wystup. Quanto options. Available at [http://www.mathfinance2.com/mf\\_website/useranonymous/company/papers/wystup\\_quanto\\_eqf.pdf](http://www.mathfinance2.com/mf_website/useranonymous/company/papers/wystup_quanto_eqf.pdf), 2011.
- U. Wystup. FX Options models trends. Available at <http://www.slideshare.net/wystup/fxmodels-slidesenglpublic>, 2013.
- Zeliade. Heston 2010. Available at <http://www.zeliade.com/whitepapers/zwp-0004.pdf>, 2011.
- O. C. Zienkiewicz, R. L. Taylor, and J.Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals, Seventh Edition*. Butterworth-Heinemann, 2013.

## Appendix A: Connecting LSV leverage function and local volatility

We exemplify the connection formula for the *LSV* model with Heston-like dynamics; the same procedure can be easily applied for *LSV* model constructed with other *SV* models.

If we want the price process of the *LSV* model to mimic that of the local volatility model and hence they generate the same pricing results for European options, we should match the diffusion terms of the two models. Noting that the local volatility  $\sigma_{LV}(K, T)$  is the square root of the expectation of the future instantaneous variance at time  $T$ , conditional on spot  $S(T) = K$ , we can connect the *LSV* leverage function and the local volatility by a mimicking theorem Des Combes [2011], Clark [2011], Tian et al. [2013], given the transition probability  $p_{LSV}(t, S(t), V(t))$  of the *LSV* model and the transition probability  $p_{LV}(t, S(t))$  of the *LV* model.

Essentially, to mimic the local volatility model, the diffusion term in the *LSV* model follows

$$\sigma_{LV}^2(x, t) = \mathbb{E} [L^2(t, S(t)) V(t) | S(t) = x] = L^2(x, t) \mathbb{E} [V(t) | S(t) = x]$$

Furthermore, the probability distribution of the *LV* model is the same as the marginal probability distribution of the *LSV* model and we have the relation between the transition probability densities

$$p_{LV}(t, x) = \int_0^\infty p_{LSV}(t, x, y) dy$$

We refer to Des Combes [2011] for a detailed proof.

Thus we obtain

$$L(x, t) = \frac{\sigma_{LV}(x, t)}{\sqrt{\mathbb{E} [V(t) | S(t) = x]}} = \sigma_{LV}(x, t) \sqrt{\frac{\int_0^\infty p_{LSV}(t, x, y) dy}{\int_0^\infty t \cdot p_{LSV}(t, x, y) dy}}$$

We also refer the reader to the pioneering work of Lipton [2002].

## Appendix B: Local volatility function

According to Sepp [2011b], the local volatility function can have:

- a parametric form
  - specify a functional form, such as CEV, shifted lognormal, quadratic, etc
  - calibrated by bootstrap and least squares
  - for markets with less liquid quotes
- a non-parametric form
  - obtained either from Dupire equation or from implied volatility surface
  - for markets with liquid options

As an example of a parametric form we consider the one presented in Sepp [2011b]:

$$\sigma_{LV}(t, S) = \sigma_{ATM}(t) \cdot \sigma_{SKEW}(t, S) \cdot \sigma_{SMILE}(t, S)$$

with  $\sigma_{ATM}$  the ATM forward volatility,  $\sigma_{SKEW}$  specified as ratio of 2 CEVs using a skew parameter  $\beta(t)$  and a weight parameter  $q \in (0, 1)$

$$\sigma_{SKEW}(t, S) = \frac{1}{2} \left[ (1+q) + (1-q) \tanh \left( \frac{1+q}{1-q} (\beta(t) - 1) \ln \frac{S}{S_0} \right) \right]$$

and the smile function is assumed quadratic, with  $\alpha(t)$  a smile parameter

$$\sigma_{SMILE}(t, S) = \sqrt{1 + \left[ \alpha(t) \ln \frac{S}{S_0} \right]^2}$$

As for non-parametric form, the local volatility function can be derived from market prices of the call options using Dupire formula

$$\sigma_{LV}(t, S) = \sqrt{\frac{\frac{\partial C}{\partial T} + (r_d - r_f) K \frac{\partial C}{\partial K} + r_f C}{\frac{K^2}{2} \frac{\partial^2 C}{\partial K^2}}} \Big|_{K=S, T=t}$$

or from implied volatility surface Gatheral [2006]

$$\sigma_{LV}(t, S) = \sqrt{\frac{\sigma_{IV}^2 + 2\sigma_{IV}T \frac{\partial \sigma_{IV}}{\partial T} + 2(r_d - r_f) \sigma_{IV}K \frac{\partial \sigma_{IV}}{\partial K}}{\left(1 + d_1(S, K) K \sqrt{T} \frac{\partial \sigma_{IV}}{\partial K}\right)^2 + \sigma_{IV}K^2T \left[\frac{\partial^2 \sigma_{IV}}{\partial K^2} - d_1(S, K) \sqrt{T} \left(\frac{\partial \sigma_{IV}}{\partial K}\right)^2\right]}} \Big|_{K=S, T=t}$$

with

$$d_1(S, K) = \frac{\ln \frac{S}{K} + [r + \sigma_{IV}^2] T}{\sigma_{IV} \sqrt{T}}$$

For a good overview of procedures to construct an implied volatility surface we refer the reader to Homescu [2011b].

Although local volatility model was introduced 20 years ago, there are new developments (see Reghai et al. [2012a] and references herein) that the reader should be made aware of.

A common approach is to start by designing an interpolation method to complete the discrete market prices and obtain a continuous implied volatility surface. It is then differentiated in the Dupire formula to obtain the local volatility. At first sight it provides a simple and explicit method to calibrate a single-factor diffusion model to equity vanilla options. However it requires the availability of a continuum of option prices across strikes and maturities. One is therefore lead to solve the problem of completing the discrete set of market prices in a non-arbitrage free manner. However, the problem may not be trivial. Practical implementation of the Dupire model therefore fails to be robust, especially when applied on a large population of underlyings and under stressed markets.

Very robust approaches are proposed in [Andreasen and Huge \[2011\]](#), [Lipton and Sepp \[2011\]](#), [Reghai et al. \[2012a\]](#):

- [Andreasen and Huge \[2011\]](#) proposes a methodology that uses an implicit finite difference scheme in a novel way to construct an arbitrage free surface from a discrete set of option prices observed on the market. The standard Dupire formula can then be safely applied.
- [Lipton and Sepp \[2011\]](#) directly starts from specifying a process model and uses transform methods to quickly calibrate a tiled local volatility surface to a particular set of sparse market data.
- [Reghai et al. \[2012a\]](#) propose to calibrate the local volatility on a discrete set of benchmark prices using a fixed point algorithm. Besides its general applicability, the algorithm also has the property that the calibration inputs are not required to be well-posed, which is useful in highly volatile markets; large and brutal intra-day moves of the spot often create occurrences of arbitrage in the implied volatility surface leading to breaks in the Dupire local volatility model.

## Appendix C: Fokker-Planck equation for the LSV model

To exemplify, we consider the Heston *LSV* model given in (3.2) modified with following change of variables to avoid issues with Feller condition being violated Tian et al. [2013]

$$\begin{aligned}X(t) &= \ln \frac{S(t)}{S(0)} \\Z(t) &= \ln \frac{V(t)}{V(0)}\end{aligned}$$

After various manipulations shown in section 6.8.2 of Clark [2011], we obtain the following Fokker-Planck (forward Kolmogorov) PDE Tian et al. [2013]:

$$\begin{aligned}\frac{\partial P}{\partial t} &= -\frac{\partial}{\partial X} [(r(t) - 0.5L^2(X, t) \exp(Z) V_0) P] - \frac{\partial}{\partial Z} \left[ \left( (\kappa\theta - 0.5\gamma^2\nu^2) \frac{1}{\exp(Z) V_0} - \kappa \right) P \right] \\&\quad + 0.5 \frac{\partial^2}{\partial X^2} [L^2(X, t) \exp(Z) V_0 P] + 0.5 \frac{\partial^2}{\partial Z^2} \left[ \frac{\gamma^2\nu^2}{\exp(Z) V_0} P \right] + \frac{\partial^2}{\partial Z \partial X} [\gamma^2\nu\rho L(X, t) P] \\P(X, Z, 0) &= \delta(X - X_0) \delta(Z - Z_0)\end{aligned}$$

where  $\delta(\cdot)$  is Dirac function and  $X_0 = Z_0 = 0$ .

## Appendix D: Constructing a non-uniform spatial grid

There are various approaches in the literature to construct nonuniform grids. We describe below several of them, with  $\{P_i\}$  denoting clustering points

- solving ODE from [Tavella and Randall \[2000\]](#), [Tavella \[2002\]](#), with  $\alpha, \beta$  given constants

$$\begin{aligned} \frac{dS}{d\xi} &= \frac{A}{\left[ \sum_{k=1}^{NP} \frac{1}{\alpha_k^2 + (S(\xi) - P_k)^2} \right]^{\frac{1}{2}}} \\ S(0) &= X_1 \\ S(1) &= X_N \end{aligned} \tag{D.1}$$

- solving ODE from [Randall \[2010\]](#), with  $\alpha, \beta$  given constants

$$\begin{aligned} \frac{dS}{d\xi} &= \frac{A}{\left[ \sum_{k=1}^{NP} \frac{\alpha^2 + (S(\xi) - P_k)^2}{\alpha^2 \beta^2 + (S(\xi) - P_k)^2} \right]^{\frac{1}{2}}} \\ S(0) &= X_1 \\ S(1) &= X_N \end{aligned} \tag{D.2}$$

The above ODEs may be solved using an algorithm described in [Tian et al. \[2013\]](#), which uses Runge-Kutta 4th-order method.

A secondary transformation is then applied, to place specified points either exactly at grid points or at grid midpoints (especially useful for digital trades). It follows the algorithm:

- Given  $N + 1$  points  $S_j = S(\xi_j)$ ,  $0 \leq \xi_j \leq 1$ , from (D.1), define

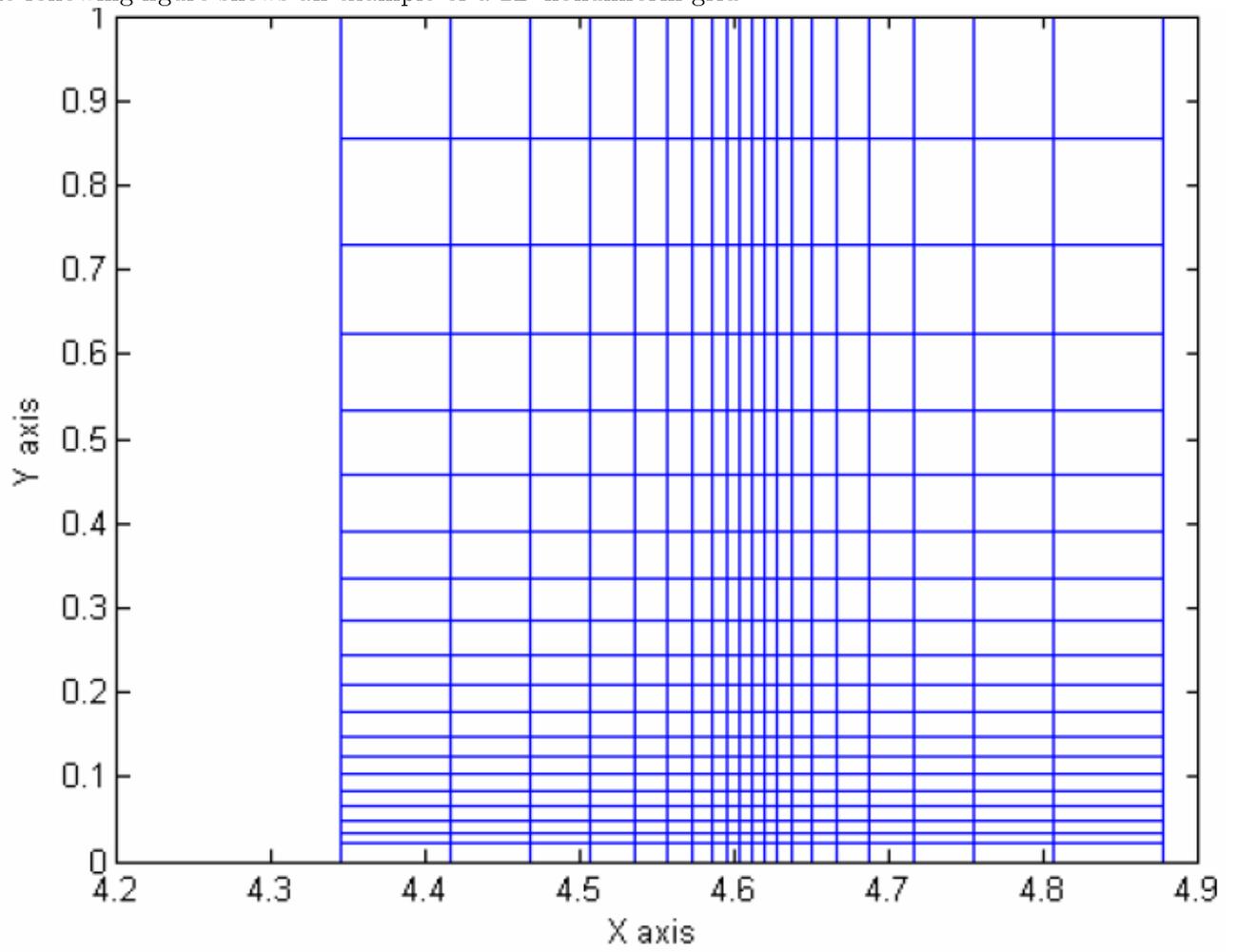
$$\begin{aligned} \xi_k^* &= \xi_{j(k)} + (\xi_{j(k)+1} - \xi_{j(k)}) \frac{P_k - S_{j(k)}}{S_{j(k)+1} - S_{j(k)}} \\ \tilde{\xi}_k &= \frac{\lfloor \xi_k^* \cdot N \rfloor}{N - 1} + \frac{\beta_k}{2 \cdot (N - 1)} \end{aligned} \tag{D.3}$$

where the index  $j(k)$  is defined as

$$j(k) = \min_j \{ \|S_{j(k)} - P_k\|, S_{j(k)} \leq P_k \}$$

- Then  $P_k$  lies either
  - ON the grid (if  $\beta_k = 0$ )
  - MIDWAY between grid points (if  $\beta_k = 1$ )
- We fit the spline interpolator  $\psi(\xi)$  to the set of points  $\{(0, 0), (\xi_1^*, \tilde{\xi}_1), \dots, (\xi_N^*, \tilde{\xi}_N), (1, 1)\}$
- Then we let  $S_j = S(\psi(\xi_j))$ , obtaining the desired grid point concentration and correct relationship with set of critical points

The following figure shows an example of a 2D nonuniform grid



## Appendix E: Enhancements of finite difference approach

We describe various enhancements of finite difference approach to solve PDEs/PIDEs.

### Compact finite difference schemes

The compact finite difference schemes [Lele \[1992\]](#), [During and Fournie \[2012\]](#), [Shukla and Zhong \[2005\]](#) provide much higher order of convergence with relatively similar computational effort compared to regular schemes using the same spatial stencil.

For example a compact scheme for Heston model is described in [During and Fournie \[2012\]](#) that is fourth order accurate in space and second order accurate in time. To show how such a scheme can be obtained, let us start with central FD approx (stepsize  $h$ ) to  $\frac{d^2y}{dx^2} = g$ , namely

$$\frac{y_{j+1} - 2y_j + y_{j-1}}{h^2} - \frac{h^2}{12}y_j^{(4)} + O(h^4) = g_j \quad (\text{E.1})$$

Another way to write error term is to use original equation

$$\frac{h^2}{12}y_j^{(4)} = \frac{h^2}{12}g_j^{(2)} = \frac{h^2}{12} \left[ \frac{g_{j+1} - 2g_j + g_{j-1}}{h^2} - \frac{h^2}{12}g_j^{(4)} + O(h^4) \right]$$

Inserting into original finite difference equation ([E.1](#))

$$\frac{y_{j+1} - 2y_j + y_{j-1}}{h^2} = g_j + \frac{g_{j+1} - 2g_j + g_{j-1}}{h^2} + O(h^4)$$

### Time extrapolation

This approach can increase the order of time convergence in a straightforward way, with little additional development effort. To exemplify, let us consider a scheme with time convergence order of  $O(\Delta t)$

$$U^{(m+1)} = L_{\Delta t} [U^{(m)}] + \Theta^{(m)}$$

The main idea to obtain an order of convergence of  $O(\Delta t^3)$  is to obtain  $U^{(m+3)}$  in different ways, then match expansions

$$\begin{aligned} U_{(1)}^{(m+3)} &= L_{\Delta t} L_{\Delta t} L_{\Delta t} [U^{(m)}] + \Theta_{(1)}^{(m)} \\ U_{(2)}^{(m+3)} &= L_{2\Delta t} L_{\Delta t} [U^{(m)}] + \Theta_{(2)}^{(m)} \\ U_{(13)}^{(m+3)} &= L_{\Delta t} L_{2\Delta t} [U^{(m)}] + \Theta_{(3)}^{(m)} \\ U_{(4)}^{(m+3)} &= L_{3\Delta t} [U^{(m)}] + \Theta_{(4)}^{(m)} \end{aligned}$$

By matching expansions to  $2^{nd}, 3^{rd}$  order, we obtain the solution with  $O(\Delta t^3)$  order of time convergence

$$U^{(m+3)} = \frac{9}{2}U_{(1)}^{(m+3)} - \frac{9}{4}U_{(2)}^{(m+3)} - \frac{9}{4}U_{(13)}^{(m+3)} + U_{(4)}^{(m+3)}$$

## Appendix F: Incorporating special features of financial instruments

We describe how to deal in the implementation with special characteristics of financial instruments.

### Discrete monitoring

Most of the literature considers options with continuous monitoring. However, most traded financial instruments have discretely monitored features, such as barrier, touch, Asian, Bermudan options. To handle these features the numerical solution needs to be allowed to “diffuse” outside of region determined by trade-specific spatial points (e.g., barriers for barrier option), as described in Andersen and Piterbarg [2010].

The trade-specific conditions enforced only at monitoring times, and thus all trade observation dates  $\{T_k\}$  have to be contained in the time grid, with multiple time steps between  $[T_k, T_{k+1}]$

A “jump condition” may be imposed at each observation time  $T_k$ , linking values at  $T_k$  with values at  $T_{k-1}$  Tavella and Randall [2000], Andersen and Piterbarg [2010].

### Coupon-paying securities and dividends

Their corresponding characteristics are integrated into the solver through a jump condition ?. Let payment at time  $T_k$  be  $p(T_k, x)$ . Then

$$V_i(T_{k+1}) = V_i(T_k) + p(T_k, X_i)$$

It is a little more complicated for case of instrument which does not pay coupons, but is written on a security that does. If we consider a stock paying dividend, say  $d(T_k, x)$ , the underlying  $X$  will have a discontinuity at  $T_k$ :  $x(T_{k+1}) = x(T_k) - d(T_k, x)$ . Then the jump condition becomes

$$V(T_{k+1}, X_i) = V(T_k, X_i - d(T_k, X_i))$$

Values  $V(T_k, X_i - d(T_k, X_i))$  are found by interpolation in  $X$ -direction of  $\{V_i(T_k)\}$ , using higher order (not linear) interpolator.

### Treatment of nonsmooth payoffs

Discontinuities in the payoff function (or its derivatives) may cause inaccuracies for numerical schemes, as well as poor estimates of Greeks (e.g., Delta and Gamma). To minimize these issues various enhancements were considered in the literature Pooley et al. [2003], Giles and Carter [2006], Tavella and Randall [2000], Tavella [2002], Duffy [2006], Andersen and Piterbarg [2010], Kienitz and Wetterau [2012], Chan and Joshi [2013], Giles [2008], such as:

- Rannacher time marching
- Averaging of initial conditions (continuity correction)
- Grid shifting
- Vibrato

## Continuity correction and Grid shifting

If spectrum of  $g(x)$  contains frequencies higher than Nyquist frequency  $\frac{1}{2\Delta x}$ , then the information is lost when  $g(x)$  sampled on mesh  $\{x_j\}$ . Important features of payoff are lost between grid points and erratic behavior when location of critical points changes relative to spatial grid (“odd-even” effect).

The continuity correction at  $x_j$  uses average value of function over interval centered in  $x_j$

$$\frac{2}{x_{j+1} - x_{j-1}} \int_{0.5(x_j+x_{j-1})}^{0.5(x_j+x_{j+1})} g(x) dx$$

Grid shifting approach is useful for options with discontinuous payoff (e.g., digital options). This approach relies on spatial grid constructed such that the  $x$ -values where payoff (or its derivatives) are discontinuous are exactly midway between grid nodes.

## Rannacher time marching

Stability of Greeks may require very small timesteps. Due to smoothing property of parabolic PDE, the solution is usually quite smooth shortly before expiry. This observation is exploited in Rannacher time marching enhancement, which replaces a few timesteps at the start by implicit Euler steps, as mentioned in [Pooley et al. \[2003\]](#).

An improvement described in [Giles and Carter \[2006\]](#) replaces each of first 2 timesteps by 2 implicit Euler timesteps of half the stepsize. This enhancement produces sufficient damping to high-frequency errors to give stable deltas and gammas.

Plots from [Giles and Carter \[2006\]](#) are illustrative of results before and after the procedure:

FIGURE 1  $V$ ,  $\Delta$  and  $\Gamma$  for a European call option.

FIGURE 2  $V$ ,  $\Delta$  and  $\Gamma$  for a digital call option.

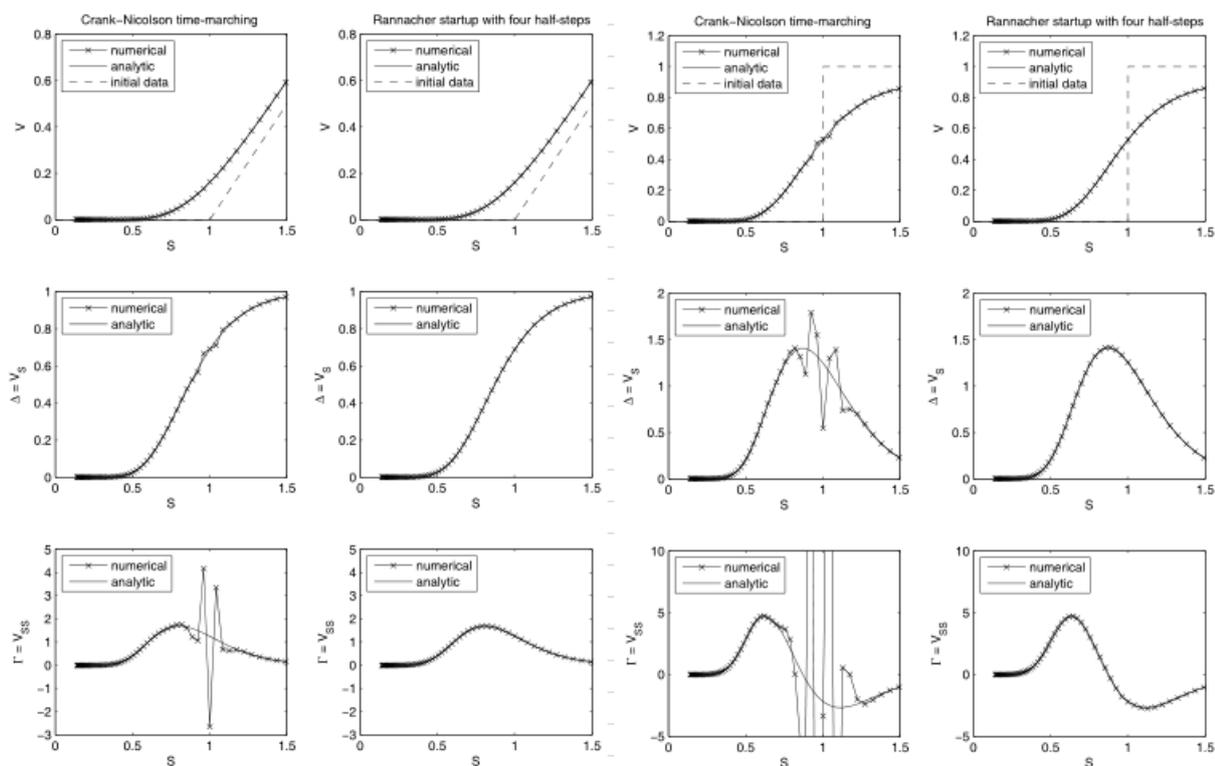


Figure .1: Results before (left column) and after (right column)

## Appendix G: Spatial discretization formulas

We start with formulas for 1D operators, and then show formulas for 2D operators.

### Formulas for 1D operators

For the 1D domain  $[x_L, x_U]$  we consider the spatial grid given by  $\{X_j\}$ , with  $x_L = X_1 < X_2 < \dots < X_{N-1} < X_N = x_U$ .

We introduce the following notations

$$\begin{aligned} h_j^+ &\triangleq X_{j+1} - X_j \\ h_j^- &\triangleq X_j - X_{j-1} \\ \varphi_j^- &\triangleq h_j^- (h_j^- + h_j^+) \\ \varphi_j^0 &\triangleq h_j^- h_j^+ \\ \varphi_j^+ &\triangleq h_j^+ (h_j^- + h_j^+) \end{aligned}$$

and

$$\begin{aligned} \left[ \frac{\partial U_j^m}{\partial x} \right]^+ &\triangleq \frac{U_{j+1}^m - U_j^m}{h_j^+} \\ \left[ \frac{\partial U_j^m}{\partial x} \right]^- &\triangleq \frac{U_j^m - U_{j-1}^m}{h_j^-} \end{aligned}$$

Various types of spatial discretization methods can be considered

### Second order central difference discretization

We discretize the spatial derivatives as follows

$$\begin{aligned} \frac{\partial u}{\partial x}(X_j, t_m) &\approx \frac{h_j^- \left[ \frac{\partial U_j^m}{\partial x} \right]^+ + h_j^+ \left[ \frac{\partial U_j^m}{\partial x} \right]^-}{h_j^+ + h_j^-} \\ &= -\frac{h_j^+}{\varphi_j^-} U_{j-1}^m + \frac{h_j^+ - h_j^-}{\varphi_j^0} U_j^m + \frac{h_j^-}{\varphi_j^+} U_{j+1}^m \\ \frac{\partial^2 u}{\partial x^2}(X_j, t_m) &\approx \frac{\left[ \frac{\partial U_j^m}{\partial x} \right]^+ - \left[ \frac{\partial U_j^m}{\partial x} \right]^-}{\frac{1}{2}(h_j^+ + h_j^-)} \\ &= \frac{2}{\varphi_j^-} U_{j-1}^m - \frac{2}{\varphi_j^0} U_j^m + \frac{2}{\varphi_j^+} U_{j+1}^m \end{aligned}$$

### First order (to the right) difference discretization

We discretize the spatial derivatives as follows

$$\begin{aligned} \frac{\partial u}{\partial x}(X_j, t_m) &\approx -\frac{1}{h_j^+} U_j^m + \frac{1}{h_j^+} U_{j+1}^m \\ \frac{\partial^2 u}{\partial x^2}(X_j, t_m) &\approx -\frac{2}{h_j^+ (h_j^+ + h_{j+1}^+)} U_j^m - \frac{2}{h_j^+ h_{j+1}^+} U_{j+1}^m + \frac{2}{h_{j+1}^+ (h_j^+ + h_{j+1}^+)} U_{j+2}^m \end{aligned}$$

## First order (to the left) difference discretization

We discretize the spatial derivatives as follows

$$\begin{aligned}\frac{\partial u}{\partial x}(X_j, t_m) &\approx -\frac{1}{h_j^-}U_{j-1}^m + \frac{1}{h_j^-}U_j^m \\ \frac{\partial^2 u}{\partial x^2}(X_j, t_m) &\approx -\frac{2}{h_{j-1}^-(h_{j-1}^- + h_j^-)}U_{j-2}^m - \frac{2}{h_{j-1}^-h_j^-}U_{j-1}^m + \frac{2}{h_j^-(h_{j-1}^- + h_j^-)}U_j^m\end{aligned}$$

## Spatial discretization operator

Let us consider that the spatial operator is defined as

$$L[u] = a(x, t)\frac{\partial^2 u}{\partial x^2} + b(x, t)\frac{\partial u}{\partial x} + c(x, t)u + f(x, t)$$

Assuming spatial derivatives are discretized using central finite difference, the expression of the spatial discretization operator  $\tilde{L}_x$  becomes

$$\begin{aligned}\tilde{L}_x[U_j^m] &= a_j^m\left(\frac{2}{\varphi_j^-}U_{j-1}^m - \frac{2}{\varphi_j^0}U_j^m + \frac{2}{\varphi_j^+}U_{j+1}^m\right) \\ &+ b_j^m\left(-\frac{h_j^+}{\varphi_j^-}U_{j-1}^m + \frac{h_j^+ - h_j^-}{\varphi_j^0}U_j^m + \frac{h_j^-}{\varphi_j^+}U_{j+1}^m\right) \\ &+ c_j^mU_j^m + f_j^m\end{aligned}$$

For the special case of uniform grid (with  $\Delta X \triangleq X_{j+1} - X_j$ ) we obtain the familiar discretization

$$\tilde{L}_x[U_j^m] = a_j^m\frac{U_{j+1}^m - 2U_j^m + U_{j-1}^m}{\Delta X^2} + b_j^m\frac{U_{j+1}^m - U_{j-1}^m}{2\Delta X} + c_j^mU_j^m + f_j^m$$

## Formulas for 2D operators

For the 2D domain  $[x_L, x_R] \times [y_D, y_U]$  we consider the spatial grid given by  $\{(X_j, Y_k)\}$ , with  $x_L = X_1 < X_2 < \dots < X_{N-1} < X_N = x_R$  and  $y_D = Y_1 < Y_2 < \dots < Y_{P-1} < Y_P = y_U$ .

We introduce the following notations

$$\begin{aligned}\Delta X_j^+ &\triangleq X_{j+1} - X_j \\ \Delta X_j^- &\triangleq X_j - X_{j-1} \\ \Delta Y_k^+ &\triangleq Y_{k+1} - Y_k \\ \Delta Y_k^- &\triangleq Y_k - Y_{k-1} \\ \varphi_j^- &\triangleq \Delta X_j^- (\Delta X_j^- + \Delta X_j^+) \\ \varphi_j^0 &\triangleq \Delta X_j^- \Delta X_j^+ \\ \varphi_j^+ &\triangleq \Delta X_j^+ (\Delta X_j^- + \Delta X_j^+) \\ \chi_j^- &\triangleq \Delta Y_k^- (\Delta Y_k^- + \Delta Y_k^+) \\ \chi_j^0 &\triangleq \Delta Y_k^- \Delta Y_k^+ \\ \chi_j^+ &\triangleq \Delta Y_k^- (\Delta Y_k^- + \Delta Y_k^+)\end{aligned}$$

## Second order central difference discretization

We discretize the spatial derivatives as follows

$$\frac{\partial u}{\partial x}(X_j, Y_k, t_m) \approx -\frac{\Delta X_j^+}{\varphi_j^-} U_{j-1,k}^m + \frac{\Delta X_j^+ - \Delta X_j^-}{\varphi_j^0} U_{j,k}^m + \frac{\Delta X_j^-}{\varphi_j^+} U_{j+1,k}^m$$

$$\frac{\partial^2 u}{\partial x^2}(X_j, Y_k, t_m) \approx \frac{2}{\varphi_j^-} U_{j-1,k}^m - \frac{2}{\varphi_j^0} U_{j,k}^m + \frac{2}{\varphi_j^+} U_{j+1,k}^m$$

$$\frac{\partial u}{\partial y}(X_j, Y_k, t_m) \approx -\frac{\Delta Y_k^+}{\chi_k^-} U_{j,k-1}^m + \frac{\Delta Y_k^+ - \Delta Y_k^-}{\chi_k^0} U_{j,k}^m + \frac{\Delta Y_k^-}{\chi_k^+} U_{j,k+1}^m$$

$$\frac{\partial^2 u}{\partial y^2}(X_j, Y_k, t_m) \approx \frac{2}{\chi_k^-} U_{j,k-1}^m - \frac{2}{\chi_k^0} U_{j,k}^m + \frac{2}{\chi_k^+} U_{j,k+1}^m$$

$$\begin{aligned} \frac{\partial^2 u}{\partial y \partial x}(X_j, Y_k, t_m) &\approx \frac{\Delta X_j^+ \Delta Y_k^+}{\varphi_j^- \chi_k^-} U_{j-1,k-1}^m - \frac{(\Delta X_j^+ - \Delta X_j^-) \Delta Y_k^+}{\varphi_j^- \chi_k^0} U_{j,k-1}^m - \frac{\Delta X_j^- \Delta Y_k^+}{\varphi_j^- \chi_k^+} U_{j+1,k-1}^m \\ &\quad - \frac{\Delta X_j^+ (\Delta Y_k^+ - \Delta Y_k^-)}{\varphi_j^0 \chi_k^-} U_{j-1,k}^m + \frac{(\Delta X_j^+ - \Delta X_j^-) (\Delta Y_k^+ - \Delta Y_k^-)}{\varphi_j^0 \chi_k^0} U_{j,k}^m \\ &\quad + \frac{\Delta X_j^- (\Delta Y_k^+ - \Delta Y_k^-)}{\varphi_j^0 \chi_k^+} U_{j+1,k}^m \\ &\quad - \frac{\Delta X_j^+ \Delta Y_k^-}{\varphi_j^+ \chi_k^-} U_{j-1,k+1}^m + \frac{(\Delta Y_k^+ - \Delta Y_k^-) \Delta Y_k^-}{\varphi_j^+ \chi_k^0} U_{j,k+1}^m + \frac{\Delta X_j^- \Delta Y_k^-}{\varphi_j^+ \chi_k^+} U_{j+1,k+1}^m \end{aligned}$$

## First order (to the right) difference discretization

We discretize the spatial derivatives as follows

$$\frac{\partial u}{\partial x}(X_j, Y_k, t_m) \approx -\frac{1}{\Delta X_j^+} U_{j,k}^m + \frac{1}{\Delta X_j^+} U_{j+1,k}^m$$

$$\frac{\partial^2 u}{\partial x^2}(X_j, Y_k, t_m) \approx -\frac{2}{\Delta X_j^+ (\Delta X_j^+ + \Delta X_{j+1}^+)} U_{j,k}^m - \frac{2}{\Delta X_j^+ \Delta X_{j+1}^+} U_{j+1,k}^m + \frac{2}{\Delta X_j^+ (\Delta X_j^+ + \Delta X_{j+1}^+)} U_{j+2,k}^m$$

$$\frac{\partial u}{\partial y}(X_j, Y_k, t_m) \approx -\frac{1}{\Delta Y_k^+} U_{j,k}^m + \frac{1}{\Delta Y_k^+} U_{j,k+1}^m$$

$$\frac{\partial^2 u}{\partial y^2}(X_j, Y_k, t_m) \approx -\frac{2}{\Delta Y_k^+ (\Delta Y_k^+ + \Delta Y_{k+1}^+)} U_{j,k}^m - \frac{2}{\Delta Y_k^+ \Delta Y_{k+1}^+} U_{j,k+1}^m + \frac{2}{\Delta Y_k^+ (\Delta Y_k^+ + \Delta Y_{k+1}^+)} U_{j,k+2}^m$$

## First order (to the left) difference discretization

We discretize the spatial derivatives as follows

$$\frac{\partial u}{\partial x}(X_j, Y_k, t_m) \approx -\frac{1}{\Delta X_j^-} U_{j-1,k}^m + \frac{1}{\Delta X_j^-} U_{j,k}^m$$

$$\frac{\partial^2 u}{\partial x^2}(X_j, Y_k, t_m) \approx -\frac{2}{\Delta X_j^- (\Delta X_j^- + \Delta X_{j-1}^-)} U_{j-2,k}^m - \frac{2}{\Delta X_j^- \Delta X_{j-1}^-} U_{j-1,k}^m + \frac{2}{\Delta X_j^- (\Delta X_j^- + \Delta X_{j-1}^-)} U_{j,k}^m$$

$$\begin{aligned}\frac{\partial u}{\partial y}(X_j, Y_k, t_m) &\approx -\frac{1}{\Delta Y_k^-} U_{j,k-1}^m + \frac{1}{\Delta Y_k^-} U_{j,k}^m \\ \frac{\partial^2 u}{\partial y^2}(X_j, Y_k, t_m) &\approx -\frac{2}{\Delta Y_k^- (\Delta Y_k^- + \Delta Y_{k-1}^-)} U_{j,k-2}^m - \frac{2}{\Delta Y_k^- \Delta Y_{k-1}^-} U_{j,k-1}^m + \frac{2}{\Delta Y_k^- (\Delta Y_k^- + \Delta Y_{k-1}^-)} U_{j,k}^m\end{aligned}$$

### Spatial discretization operator

Let us consider that the spatial operator is defined as

$$\begin{aligned}L[u] &= a(x, y, t) \frac{\partial^2 u}{\partial x^2} + b(x, y, t) \frac{\partial u}{\partial x} \\ &\quad + c(x, y, t) \frac{\partial^2 u}{\partial y^2} + d(x, y, t) \frac{\partial u}{\partial y} + e(x, y, t) \frac{\partial^2 u}{\partial x \partial y} \\ &\quad + g(x, y, t)u + f(x, y, t)\end{aligned}$$

Assuming spatial derivatives are discretized using central finite difference, the expression of the spatial discretization operator  $\tilde{L}$  becomes

$$\begin{aligned}\tilde{L}[U_j^m] &= a_{j,k}^m \left( \frac{2}{\varphi_j^-} U_{j-1,k}^m - \frac{2}{\varphi_j^0} U_{j,k}^m + \frac{2}{\varphi_j^+} U_{j+1,k}^m \right) \\ &\quad + b_{j,k}^m \left( -\frac{\Delta X_j^+}{\varphi_j^-} U_{j-1,k}^m + \frac{\Delta X_j^+ - \Delta X_j^-}{\varphi_j^0} U_{j,k}^m + \frac{\Delta X_j^-}{\varphi_j^+} U_{j+1,k}^m \right) \\ &\quad + c_{j,k}^m \left( \frac{2}{\chi_k^-} U_{j,k-1}^m - \frac{2}{\chi_k^0} U_{j,k}^m + \frac{2}{\chi_k^+} U_{j,k+1}^m \right) \\ &\quad + d_{j,k}^m \left( -\frac{\Delta Y_k^+}{\chi_k^-} U_{j,k-1}^m + \frac{\Delta Y_k^+ - \Delta Y_k^-}{\chi_k^0} U_{j,k}^m + \frac{\Delta Y_k^-}{\chi_k^+} U_{j,k+1}^m \right) \\ &\quad + e_{j,k}^m \times \left( \frac{\Delta X_j^+ \Delta Y_k^+}{\varphi_j^- \chi_k^-} U_{j-1,k-1}^m - \frac{(\Delta X_j^+ - \Delta X_j^-) \Delta Y_k^+}{\varphi_j^- \chi_k^0} U_{j,k-1}^m - \frac{\Delta X_j^- \Delta Y_k^+}{\varphi_j^- \chi_k^+} U_{j+1,k-1}^m \right. \\ &\quad \left. - \frac{\Delta X_j^+ (\Delta Y_k^+ - \Delta Y_k^-)}{\varphi_j^0 \chi_k^-} U_{j-1,k}^m + \frac{(\Delta X_j^+ - \Delta X_j^-) (\Delta Y_k^+ - \Delta Y_k^-)}{\varphi_j^0 \chi_k^0} U_{j,k}^m + \frac{\Delta X_j^- (\Delta Y_k^+ - \Delta Y_k^-)}{\varphi_j^0 \chi_k^+} U_{j+1,k}^m \right. \\ &\quad \left. - \frac{\Delta X_j^+ \Delta Y_k^-}{\varphi_j^+ \chi_k^-} U_{j-1,k+1}^m + \frac{(\Delta Y_k^+ - \Delta Y_k^-) \Delta Y_k^-}{\varphi_j^+ \chi_k^0} U_{j,k+1}^m + \frac{\Delta X_j^- \Delta Y_k^-}{\varphi_j^+ \chi_k^+} U_{j+1,k+1}^m \right) \\ &\quad + g_{j,k}^m U_{j,k}^m + f_{j,k}^m\end{aligned}$$

## Appendix H: Advanced time discretization schemes

While  $\theta$ -scheme is the most commonly deployed time discretization scheme in quantitative finance, it needs to be enhanced [Pooley et al. \[2003\]](#), [Giles and Carter \[2006\]](#), [Tavella and Randall \[2000\]](#), [Tavella \[2002\]](#) to address drawbacks when used for pricing some financial instruments. However, even those enhancements are not enough in some cases, such as when pricing American or Bermudan options [Le Floc'h \[2014a\]](#). Thus there is a need for considering better time discretization schemes, and we present below some of them.

The **LAWSON-MORRIS-GOURLAY** scheme was applied to solving SABR PDE in [Le Floc'h and Kennedy \[2014\]](#). It is a local Richardson extrapolation in time of second and third order, and it is a faster alternative to the standard Richardson extrapolation because the tridiagonal matrix stemming out of the finite difference discretization can be reused, while keeping L-stability and thus strong damping properties.

The **TR-BDF2** scheme has been applied to finance in the context of American option pricing [Le Floc'h \[2014a\]](#). It is second order accurate in time, L-stable, and fully implicit Runge–Kutta method, with 2 stages:

1. apply the (weighted) trapezoidal rule (TR)
2. apply the second order backward difference scheme (BDF2)

The **LAWSON-SWAYNE** scheme was considered for SABR PDE in [Le Floc'h and Kennedy \[2014\]](#) as a slightly faster second order unconditionally stable scheme. The scheme consists of applying two implicit Euler steps followed by extrapolation on the values at those two steps.

The following plots (from [Le Floc'h and Kennedy \[2014\]](#)) attest to better efficiency of Lawson-Swayne scheme (labeled LS) and TR-BDF2 scheme (labeled TRBDF2).

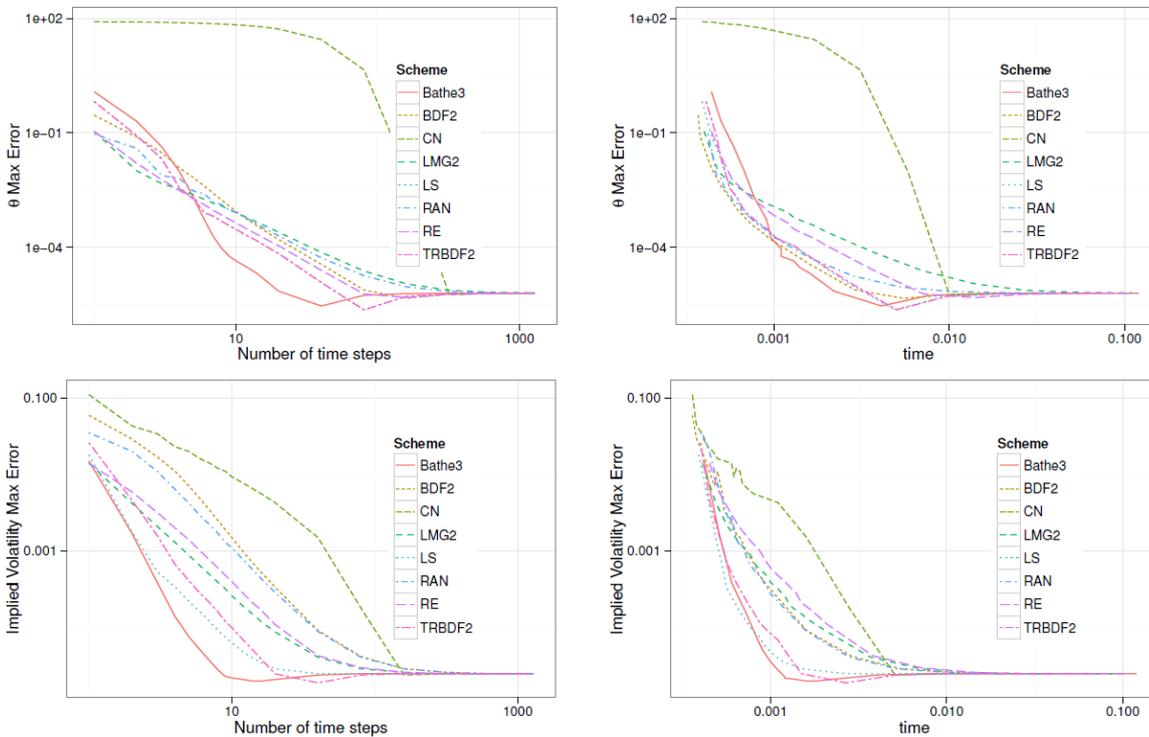


Figure .2: Results from [Le Floc'h \[2014a\]](#)

## Appendix I: Parameter estimation using Kim filter

Since the Kalman filter is the main ingredient of the Kim filter, we start with its description.

The Kalman filter is an iterative process. To use it, the model has to be expressed in a state-space form characterized by:

- transition equation: describes dynamics of state variables for which there are no empirical data
- measurement equation: represents the relationship linking the observable variables with the non-observable variables

Thus, the Kalman filter makes it possible to evaluate the non-observable variables, and it updates their values at each step using the new information. The Kalman filter is an efficient estimator for a state vector (of a linear dynamic system perturbed by Gaussian white noise, using measurements that are linear functions of the system state but corrupted by additive Gaussian white noise).

It is essentially a set of equations that implements a predictor-corrector type estimator for the unobserved state variables. During the prediction step, the Kalman filter forms an optimal predictor of the unobserved state variable, given all the information available up to previous time point, to compute **PRIOR ESTIMATES**. During the updating step, new information becomes available and is used to update the prior estimates. The resulting values are called **POSTERIOR ESTIMATES**.

In a similar manner we construct for Kim filter prior and posterior estimates for the unobserved state vector and its associated mean square error, based on historical observations and conditional on the regime at current and previous time.

The Kim filter has the following 5 major steps:

1. The prior estimates are calculated using the regime-specific transition equations
2. Compute prediction errors
3. Compute the Kalman gains corresponding to each error term
4. Update the prior estimates
5. The posterior estimates are collapsed by applying a Hamilton filter

Collapsing the posterior estimates enable us to handle the curse of dimensionality, since they do not depend anymore on the regime at time (they incorporate that information through a weighted average) at a previous observation time point.

The whole algorithm incorporates maximum likelihood in the following way. We start with a feasible choice of the parameters and initial state vector, with given covariance matrix. The prediction equations are used to give the prior estimates. Given the historical market information on the observable variables, the prediction error and its corresponding mean square error are then determined and the prior estimates are updated. When the last observation is reached, the log-likelihood is evaluated and a new parameter set is chosen such that the log-likelihood value is maximized (using a numerical optimizer). The algorithm stops when the difference between values of log-likelihood function at two consecutive iterations is smaller than specified tolerance.

The next figure illustrates the whole algorithm. While it is true that the figure (shown on page 93 in Bloechlinger [2008]) presents the flowchart of an algorithm that includes not the Kim filter but the regular Kalman filter (which has only 4 steps), we hope that it provides sufficient visual aid for the reader. The main difference is given by the additional collapsing step, but otherwise the algorithm is similar.

