



Model-free stochastic collocation for an arbitrage-free implied volatility: Part I

Fabien Le Floch¹ · Cornelis W. Oosterlee^{1,2}

Received: 23 July 2018 / Accepted: 8 February 2019 / Published online: 22 February 2019
© The Author(s) 2019

Abstract

This paper explains how to calibrate a stochastic collocation polynomial against market option prices directly. The method is first applied to the interpolation of short-maturity equity option prices in a fully arbitrage-free manner and then to the joint calibration of the constant maturity swap convexity adjustments with the interest rate swaptions smile. To conclude, we explore some limitations of the stochastic collocation technique.

Keywords Stochastic collocation · Implied volatility · Quantitative finance · Arbitrage-free · Risk-neutral density

JEL Classification C630 · G170 · G130

1 Introduction

The market provides option prices for a discrete set of strikes and maturities. In order to price over-the-counter vanilla options with different strikes, or to hedge more complex derivatives with vanilla options, it is useful to have a continuous arbitrage-free representation of the option prices, or equivalently of their implied volatilities. For example, the variance swap replication of Carr and Madan consists in integrating a specific function over a continuum of vanilla put and call option prices (Carr et al 1998; Carr and Lee 2008). An arbitrage-free representation is also particularly important for the Dupire local volatility model (Dupire 1994), where arbitrages will translate to a negative implied variance.

A rudimentary, but popular representation is to interpolate market implied volatilities with a cubic spline across option strikes. Unfortunately, it may not be arbitrage-free

✉ Fabien Le Floch
f.l.y.lefloch@tudelft.nl

¹ Delft Institute of Applied Mathematics, TU Delft, Delft, The Netherlands

² CWI-Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

as it does not preserve the convexity of option prices in general. Kahalé (2004) designs an arbitrage-free interpolation of the call prices. It, however, presents the following drawbacks: It requires convex input quotes and employs two embedded nonlinear minimizations, and it is not proven that the algorithm for the interpolation function of class C^2 converges. In reality, it is often not desirable to strictly interpolate option prices as those fluctuate within a bid–ask spread. Interpolation will lead to a very noisy estimate of the probability density (which corresponds to the second derivative of the undiscounted call option price).

More recently, Andreasen and Huge (2011) have proposed to calibrate the discrete piecewise constant local volatility of the single-step finite difference representation for the forward Dupire equation. In their representation, the authors use as many constants as the number of market option strikes for an optimal fit. It works well but often yields a noisy probability density estimate, as the prices are overfitted.

An alternative is to rely on a richer underlying stochastic model, which allows for some flexibility in the implied volatility smile, such as the Heston or SABR stochastic volatility models. While semi-analytic representations of the call option price exist for the Heston model (Heston 1993), the model itself does not allow to represent short-maturity smiles accurately. The SABR model is better suited for this, but has only closed-form approximations for the call option price, which can lead to arbitrage (Hagan et al. 2002, 2014).

Grzelak and Oosterlee (2017) use stochastic collocation to fix the Hagan SABR approximation formula defects and produce arbitrage-free option prices starting from the Hagan SABR formula. Here, we will explore how to calibrate the stochastic collocation polynomial directly to market prices, without going through an intermediate model.

This is of particular interest to the richer collocated local volatility (CLV) model, which allows to price exotic options through Monte Carlo or finite difference methods (Grzelak 2016). A collocation polynomial calibrated to the vanilla options market is key for the application of this model in practice.

Another application of our model-free stochastic collocation is to price constant maturity swaps (CMS) consistently with the swaption implied volatility smile. In the standard approximation of Hagan (2003), the CMS convexity adjustment consists in evaluating the second moment of the distribution of the forward swap rate. It can be computed in closed form with the stochastic collocation. This allows for an efficient method to calibrate the collocation method jointly to the swaptions market implied volatilities and to the CMS spread prices.

The outline of the paper is as follows: Section 2 presents the stochastic collocation technique in detail. Section 3 explains how to calibrate the stochastic collocation directly to market prices and how to ensure the arbitrage-free calibration transparently, through a specific parameterization of the collocation polynomial. Section 4 reviews some popular option implied volatilities interpolation methods and illustrates the various issues that may arise with those on a practical example. Section 5 applies the direct collocation technique on two different examples of equity index option prices. Section 6 introduces the joint calibration of CMS convexity adjustments and swaption prices in general. Section 7 applies the model-free stochastic collocation on the joint

calibration of CMS and swaption prices. Finally, Sect. 8 explores some limitations of the stochastic collocation technique along with possible remedies.

2 Overview of the stochastic collocation method

The stochastic collocation method (Mathelin and Hussaini 2003) consists in mapping a physical random variable Y to a point X of an artificial stochastic space. Collocation points x_i are used to approximate the function¹ mapping X to Y , $F_X^{-1} \circ F_Y$, where F_X, F_Y are, respectively, the cumulative distribution functions (CDF) of X and Y . Thus, only a small number of samples of Y (and evaluations of F_Y) are used. This allows the problem to be solved in the “cheaper” artificial space.

In the context of option price interpolation, the stochastic collocation will allow us to interpolate the market CDF in a better set of coordinates. In particular, we will follow (Grzelak and Oosterlee 2017) and use a Gaussian distribution for X .

In Grzelak and Oosterlee (2017), the stochastic collocation is applied to the survival density function G_Y , where $G_Y(y) = 1 - F_Y(y)$ with F_Y being the cumulative density function of the asset price process. When the survival distribution function is known for a range of strikes, their method can be summarized by the following steps:

1. Given a set of collocation strikes $y_i, i = 0, \dots, N$, compute the survival distribution values p_i at those points: $p_i = G_Y(y_i)$.
2. Project on the Gaussian distribution by transforming the p_i using the inverse cumulative normal distribution Φ^{-1} resulting in $x_i = \Phi^{-1}(1 - p_i)$.
3. Interpolate (x_i, y_i) with a Lagrange polynomial g_N .
4. Price by integrating on the density with the integration variable $x = \Phi^{-1}(1 - G_Y(y))$, using the Lagrange polynomial for the transform.

Let us now detail the last step. The undiscounted price of a call option of strike K is obtained by integrating over the probability density function f , with a change of variable:

$$\begin{aligned}
 C(K) &= \int_0^{+\infty} |y - K|^+ f(y) dy & (1) \\
 &= \int_{\Phi^{-1}(0)}^{\Phi^{-1}(1)} |G_Y^{-1}(1 - \Phi(x)) - K|^+ \phi(x) dx \\
 &\approx \int_{-\infty}^{\infty} |g_N(x) - K|^+ \phi(x) dx \\
 &= \int_{x_K}^{\infty} (g_N(x) - K) \phi(x) dx, & (2)
 \end{aligned}$$

where $\phi(x)$ is the Gaussian density function and

$$x_K = g_N^{-1}(K). \tag{3}$$

The change of variables is valid when the survival density is continuous and its derivative is integrable. In particular, it is not necessary for the derivative to be continuous.

¹ A polynomial is often used for the mapping.

As shown in Hunt and Kennedy (2004), a polynomial multiplied by a Gaussian can be integrated analytically as integration by parts leads to a recurrence relationship on $m_i(b) = \int_b^\infty x^i \phi(x) dx$. This idea is also the basis of the Sali tree method (Hu et al. 2006). The recurrence is

$$m_{i+2}(b) = (i + 1)m_i(b) + b^{i+1}\phi(b), \tag{4}$$

with $m_0(b) = \Phi(-b)$, $m_1(b) = \phi(b)$. We have then:

$$C(K) = \sum_{i=0}^N a_i m_i(x_K) - \Phi(-x_K)K, \tag{5}$$

where a_i are the coefficients of the polynomial in increasing powers.

The terms $m_i(K)$ involve only $\phi(x_K)$ and $\Phi(-x_K)$. The computational cost for pricing one vanilla option can be approximated by the cost of finding x_K and the cost of one normal density function evaluation plus one cumulative normal density function evaluation. For cubic polynomials, x_K can be found analytically through Cardano’s formula (Nonweiler 1968), and the cost is similar to the one of the Black–Scholes formulae. In the general case of a polynomial g_N of degree N , the roots can be computed in $O(N^3)$ as the eigenvalues of the associated Frobenius companion matrix M defined by

$$M(g_N) = \begin{pmatrix} 0 & 0 & \dots & 0 & -\frac{a_0}{a_N} \\ 1 & 0 & \dots & 0 & -\frac{a_1}{a_N} \\ 0 & 1 & & 0 & -\frac{a_2}{a_N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -\frac{a_{N-1}}{a_N} \end{pmatrix}.$$

We have indeed $\det(\lambda I - M) = g_N(\lambda)$. This is, for example, how the Octave or MATLAB `roots` function works (Moler 1991). Note that for a high degree N , the system can be very ill-conditioned. A remedy is to use a more robust polynomial basis such as the Chebyshev polynomials and compute the eigenvalues of the colleague matrix (Good 1961; Trefethen 2011). Jenkins and Traub solve directly the problem of finding the roots of a real polynomial in Jenkins (1975).

A simple alternative, particularly relevant in our case as the polynomial needs to be invertible and thus monotonic, is to use the third-order Halley’s method (Gander 1985) with a simple initial guess $x_K = -1$ if $K < F(0, T)$ or $x_K = 1$ if $K \geq F(0, T)$, with $F(0, T)$ the forward price to maturity T . In practice, not more than three iterations are necessary to achieve an accuracy around machine epsilon.

The put option price is calculated through the put-call parity relationship, namely

$$C(K) - P(K) = F(0, T) - K,$$

where $P(K)$ is the undiscounted price today of a put option of maturity T .

Table 1 Collocation polynomials at the Gauss–Hermite nodes for the Black–Scholes model with volatility 25%, and for the SVI parameters corresponding to a least-squares fit of SPX500 options of maturity 10 years

x_i	p_i	Black–Scholes 20 years		SVI 10 years	
		y_i	c_i	y_i	c_i
– 3.3242574335521193	0.9995567284080997	1.302	56.569	0.000608	88.616
– 1.8891758777537109	0.9705658670707293	6.475	60.296	1.196	90.181
– 0.6167065901925942	0.7312858631972767	26.861	24.385	39.392	12.040
0.6167065901925942	0.26871413680272327	106.662	11.119	146.921	– 8.065
1.8891758777537109	0.029434132929270655	442.452	6.348	255.167	– 0.267
3.3242574335521193	0.0004432715919002534	2201.299	1.210	378.042	0.457

The polynomial is expressed with the coefficients c_i as $g_N(x) = \sum_{i=0}^5 c_i x^i$

3 Calibration of the stochastic collocation to market option prices

A Lagrange polynomial g_N cannot always be used to interpolate directly on the collocation points implied by the market option strikes $(y_i)_{i=0,\dots,N}$, because on one side N might be too large for the method to be practical (there are typically more than hundred market option prices on the SPX500 equity index for a given maturity), and on the other side, there is no guarantee that the Lagrange polynomial will be monotonic, even for a small number of strikes. Grzelak and Oosterlee (2017) propose to rely on a set of collocation points $(x_i)_{i=0,\dots,N}$ determined in an optimal manner from the zeros of an orthogonal polynomial. It corresponds to the set of the Hermite quadrature points in the case of the Gaussian distribution. This presupposes that we know the survival distribution function values at strikes which do not correspond to any quoted market strike. In Grzelak and Oosterlee (2017), those values are given by the SABR model. Even with known survival distribution function values at the Hermite collocation points, the resulting polynomial is not guaranteed to be monotonic. For example, we consider options expiring in 20 years on an asset with spot $S = 100$ that follows the Black–Scholes model with a constant volatility $\sigma = 25\%$. The Lagrange collocation polynomial of degree $N = 3$ or $N = 5$ implied by the Gauss–Hermite nodes is not monotonically increasing; we have $g'_3(-2.34) = -15.2$ (see Table 1 for the polynomial details). Another simple example we encountered comes from fitting SPX500 options of maturity 10 years, with the Gatheral SVI parameterization (Gatheral and Lynch 2004). It corresponds to the following SVI parameters $a = 0.004$, $b = 0.027$, $s = 0.72$, $\rho = -0.99$ $m = 1.0$. The corresponding Lagrange quintic polynomial obtained at the Gauss–Hermite nodes decreases around $x = -2.36$ as we have $g'_5(-2.36) = -16.5$.

In this paper, we don't want to assume a prior model. Instead of using a Lagrange polynomial g_N to interpolate on well-chosen $(x_i)_{i=0,\dots,N}$ as in the step 3 of the collocation method described in Sect. 2, we will directly calibrate a *monotonic polynomial* g_N to the market option prices at strikes $(y_i)_{i=0,\dots,m}$, with m typically much larger than N . The monotonicity will be guaranteed through a specific isotonic parameterization. The proposed parameterization will also conserve the first moment of the distribution exactly.

In order to apply the stochastic collocation directly to market option prices, we thus need to:

- find an estimate of the survival density from the market option prices (corresponding to the step 1 of the collocation method described in Sect. 2),
- find a good initial guess for the monotonic polynomial g_N ,
- optimize the polynomial coefficients so that the collocation prices are closest to the market option prices.

We will detail each step.

3.1 A rough estimate of the market survival density

Kahalé (2004) proposes a straightforward estimate. Let $(y_i)_{i=0,\dots,m}$ be the market strikes and $(c_i)_{i=0,\dots,m}$ the market call option prices corresponding to each strike; the call price derivative c'_i toward the strike K_i can be estimated by

$$c'_i \approx \frac{l_i + l_{i+1}}{2} \text{ where } l_i = \frac{c_i - c_{i-1}}{y_i - y_{i-1}} \tag{6}$$

for $i = 1, \dots, m - 1$, and with $c'_0 = l_1, c'_m = l_m$.

If the market prices are arbitrage-free, that is when

$$-1 < \frac{c_i - c_{i-1}}{y_i - y_{i-1}} < \frac{c_{i+1} - c_i}{y_{i+1} - y_i} < 0, \text{ for } i = 1, \dots, m - 1 \tag{7}$$

it is guaranteed that $-1 < c'_i < 0$ and the c'_i are increasing. A more precise estimate consists in using the parabola that passes through the three points c_{i-1}, c_i, c_{i+1} to estimate the slopes:

$$c'_i \approx \frac{l_i(y_{i+1} - y_i) + l_{i+1}(y_i - y_{i-1})}{y_{i+1} - y_{i-1}} \text{ where } l_i = \frac{c_i - c_{i-1}}{y_i - y_{i-1}} \tag{8}$$

for $i = 1, \dots, m - 1$, and with $c'_0 = l_1, c'_m = l_m$. It will still lead to $-1 < c'_i < 0$ and increasing c'_i .

We can build a continuous representation of the survival density by interpolating the call prices $(y_i, c_i)_{i=0,\dots,m}$ with the C^1 quadratic spline interpolation of Schumaker (1983), where additional knots are inserted to preserve monotonicity and convexity.² By construction, at each market strike, the derivative will be equal to each c'_i .

The survival density corresponds to

$$G_Y(y) = -\frac{\partial C}{\partial y}(y), \tag{9}$$

² A C^1 polynomial spline on a fixed set of knots cannot preserve monotonicity and convexity in the general case (Passow and Roulier 1977).

or equivalently through the put option prices P :

$$G_Y(y) = 1 - \frac{\partial P}{\partial y}(y). \tag{10}$$

In practice, one will use out-of-the-money options to compute the survival density using alternative Eqs. (9) and (10). While in the case of the SABR model, it is important to integrate the probability density (or the second derivative of the call price) from y to ∞ (Grzelak and Oosterlee 2017), here we are only interested in a rough guess.³

From the survival density at the strikes y_i , it is then trivial to compute the normal coordinates x_i .

3.2 Filtering out the market call prices quotes

In reality, it is not guaranteed that the market prices are convex, because of the bid–ask spread. While the collocation calibration method we propose in this paper will still work well on a nonconvex set of call prices, starting the optimization from a convex set has two advantages: a better initial estimate of the survival density and thus a better initial guess, and the use of a monotonic interpolation of the survival density.

The general problem of extracting a “good”, representative convex subset of a nearly convex set is not simple. By “good”, we mean, for example, that the frontier defined by joining each point of the set with a line minimizes the least-squares error on the full set, along with possibly some criteria to reduce the total variation. In Appendix C, we propose a quadratic programming approach to build a convex set that closely approximates the initial set of market prices. It can, however, be relatively slow when the number of market quotes is large. The algorithm takes 4.8 s on a Core i7 7700U, for the 174 SPX500 option prices as of January 23, 2018, from our example in Sect. 5.

A much simpler approach is to merely filter out problematic quotes, i.e., quotes that will lead to a call price derivative estimate lower than -1 or positive. We assume that the strikes $(y_i)_{0 \leq i \leq m}$ are sorted in ascending order. The algorithm starts from a specific index $k \in \{0, \dots, m\}$. We will use $k = 0$, but we let the algorithm to be more generic. The forward sweep to filter out problematic quotes consists then in:

- (i) Start from strike y_k . Let the filtered set be $\mathcal{S} = \{(y_k, x_K)\}$. Let $j^* = k$,
- (ii) Search for the next lowest index j , such that $-1 + \epsilon < \frac{c_j - c_{j^*}}{y_j - y_{j^*}} < -\epsilon$ and $j^* < j \leq m$. Replace j^* by j ,
- (iii) Add (y_{j^*}, c_{j^*}) to the filtered set \mathcal{S} . Repeat steps (ii) and (iii).

In our examples, we set the tolerance $\epsilon = 10^{-7}$ to avoid machine epsilon accuracy issues close to -1 . A small error in the derivative estimate near -1 or near 0 will lead to a disproportionately large difference in the coordinate x .

While the above algorithm will not produce a convex set, we will see that it can be surprisingly effective to compute a good initial guess for the collocation polynomial.

³ Integration is still possible with the quadratic spline interpolation approach.

We could also derive a similar backward sweep algorithm and combine the two algorithms to start at the strike y_k close to the forward price $F(0, T)$. On our examples, this was not necessary.

3.3 An initial guess for the collocation polynomial

In order to obtain an arbitrage-free price, it is not only important that the density (zeroth moment) sums up to 1, which the collocation method will obey by default, but it is also key to preserve the martingale property (the first moment), that is

$$\int_{-\infty}^{\infty} g_N(x)\phi(x) = F(t, T). \tag{11}$$

Using the recurrence relation [Eq. (4)], this translates to

$$a_0 + \sum_{i=1}^{\frac{N-1}{2}} a_{2i}(2i - 1)!! = F(t, T). \tag{12}$$

Instead of trying to find directly good collocation points, a simple idea for an initial guess is to consider the polynomial $h_N(x) = \sum_{k=0}^N b_k x^k$ corresponding to the least-squares fit of x_i, y_i :

$$b_0, \dots, b_N = \min_{(a_0, \dots, a_N) \in \mathbb{R}^{N+1}} \left\{ \sum_{i=0}^m \left[\left(\sum_{k=0}^N a_k x_i^k \right) - y_i \right]^2 \right\}, \tag{13}$$

with the additional martingality constraint. This is a linear problem and is very fast to solve, for example, by QR decomposition. Unfortunately, the resulting polynomial might not be monotonic.

As we want to impose the monotonicity constraint by a clever parameterization of the problem, we will only consider the least-squares (with additional martingality constraint) cubic polynomial as starting guess. The following lemma helps us determine whether it is monotonic.

Lemma 1 *A cubic polynomial $a_0 + a_1x + a_2x^2 + a_3x^3$ is strictly monotonic and increasing on \mathbb{R} if and only if $a_2^2 - 3a_1a_3 < 0$.*

Proof The derivative has no roots if and only if the discriminant $a_2^2 - 3a_1a_3 < 0$ \square

If our first attempt for a cubic initial guess is not monotonic, we follow the idea of Murray et al. (2016) and fit a cubic polynomial of the form $A + Bx + Cx^3$. For this specific case, the linear system to solve is then given by

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \sum_{i=0}^m x_i^2 & \sum_{i=0}^m x_i^4 \\ 0 & \sum_{i=0}^m x_i^4 & \sum_{i=0}^m x_i^6 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} F(t, T) \\ \sum_{i=0}^m x_i y_i \\ \sum_{i=0}^m x_i^3 y_i \end{pmatrix}. \tag{14}$$

In our case, Lemma 1 reduces to $B > 0$ and $C > 0$. As the initial guess, we thus use the cubic polynomial with coefficients $a_0 = A = F(t, T)$, $a_1 = |B|$, $a_2 = 0$, $a_3 = |C|$.

3.4 The measure

The goal is to minimize the error between specific model implied volatilities and the market implied volatilities, taking into account the bid–ask spread. The implied volatility error measure corresponds then to the weighted root-mean-square error of implied volatilities:

$$M_\sigma = \frac{\sqrt{\sum_{i=0}^m \mu_i^2 (\sigma(\xi, K_i) - \sigma_i)^2}}{\sqrt{\sum_{i=0}^m \mu_i^2}}, \tag{15}$$

where $\sigma(\xi, K_i)$ is the Black implied volatility⁴ obtained from the specific model considered, with parameters ξ , σ_i is the market implied volatility and μ_i is the weight associated with the implied volatility σ_i . In our numerical examples, we will choose $\mu_i = 1$. In practice, it is typically set as the inverse of the bid–ask spread.

An alternative is to use the root-mean-square error of prices:

$$M_V = \frac{\sqrt{\sum_{i=0}^m w_i^2 (C(\xi, K_i) - c_i)^2}}{\sqrt{\sum_{i=0}^m w_i^2}}, \tag{16}$$

where $C(\xi, K_i)$ is the model⁵ option price and c_i is the market option price at strike K_i . We can find a weight w_i that makes the solution similar to the one under the measure M_σ by matching the gradients of each problem. We compare

$$\sum_{i=0}^m 2w_i^2 \frac{\partial C}{\partial \xi}(\xi, K_i) (c_i - C(\xi, K_i)),$$

with

$$\sum_{i=0}^m 2\mu_i^2 \frac{\partial \sigma}{\partial \xi}(\xi, K_i) (\sigma_i - \sigma(\xi, K_i)).$$

As we know that $\frac{\partial C}{\partial \xi} = \frac{\partial \sigma}{\partial \xi} \frac{\partial C}{\partial \sigma}$, we approximate $\frac{\partial C}{\partial \sigma}$ by the market Black–Scholes Vega, the term $(c_i - C(\xi, K_i))$ by $\frac{\partial C}{\partial \xi}(\xi_{opt} - \xi)$, and $(\sigma_i - \sigma(\xi, K_i))$ by $\frac{\partial \sigma}{\partial \xi}(\xi_{opt} - \xi)$ to obtain

$$w_i \approx \frac{1}{\frac{\partial c_i}{\partial \sigma_i}} \mu_i. \tag{17}$$

⁴ Fast and robust algorithms to obtain the implied volatility from an option price are given in Jäckel (2015) and Li and Lee (2011). When no implied volatility corresponds to the model option price, which can happen because of numerical error, we just fix the implied volatility to zero.

⁵ In the case of the stochastic collocation, ξ corresponds to the coefficients of the collocation polynomial.

In practice, the inverse Vega needs to be capped to avoid taking into account too far out-of-the-money prices, which won't be all that reliable numerically and we take

$$w_i = \min\left(\frac{1}{v_i}, \frac{10^6}{F}\right) \mu_i, \tag{18}$$

where $v_i = \frac{\partial c_i}{\partial \sigma}$ is the Black–Scholes Vega corresponding to the market option price c_i .

3.5 Optimization under monotonicity constraints

We wish to minimize the error measure M_V while taking into account the martingality and the monotonicity constraints (Lemma 1) at the same time. The polynomial g_N is strictly monotonically increasing if its derivative polynomial is strictly positive. We follow the central idea of Murray et al. (2016) and express g_N in an isotonic parameterization:

$$g_N(x) = a_0 + \int_0^x p(x)dx, \tag{19}$$

where $p(x)$ is a strictly positive polynomial of degree $N - 1 = 2Q$. It can thus be expressed as a sum of two squared polynomials of respective degrees at most Q and at most $Q - 1$ (Reznick 2000):

$$p(x) = p_1(x)^2 + p_2(x)^2. \tag{20}$$

As in the case of the cubic polynomial, we can refine the initial guess by first finding the optimal positive least-squares polynomial with the sum of squares parameterization. Let $(\beta_{1,0}, \dots, \beta_{1,q}) \in \mathbb{R}^{q+1}$ be the coefficients of the polynomial p_1 and $(\beta_{2,0}, \dots, \beta_{2,q-1}) \in \mathbb{R}^q$ be the coefficients of the polynomial p_2 . The coefficients $(\gamma_k)_{k=0, \dots, N-1}$ of p can be computed by adding the convolution of β_1 with itself to the convolution of β_2 with itself, that is

$$\gamma_k = \sum_{l=0}^k \beta_{1,l} \beta_{1,k-l} + \sum_{l=0}^k \beta_{2,l} \beta_{2,k-l}, \tag{21}$$

with $\beta_{1,l} = 0$ for $l > q$ and $\beta_{2,l} = 0$ for $l > q - 1$. The martingality condition leads to

$$g_N(x) = F(t, T) - \sum_{k=1}^{\frac{N-1}{2}} \frac{\gamma_{2k-1}}{2k} (2k - 1)!! + \sum_{k=1}^N \frac{\gamma_{k-1}}{k} x^k. \tag{22}$$

Lemma 2 *The gradient of g_N toward $(\beta_{1,0}, \dots, \beta_{1,q}, \beta_{2,0}, \dots, \beta_{2,q-1})$ can be computed analytically, and we have*

$$\frac{\partial g_N}{\partial \beta_{l,j}}(x_i) = 2 \sum_{k=0}^q \frac{\beta_{l,k} x_i^{j+k+1}}{k+j+1} - \sum_{k=1}^q \frac{(2k-1)!!}{k} \beta_{l,2k-j-1}, \tag{23}$$

with $\beta_{l,k} = 0$ for $k < 0$ and $\beta_{1,k} = 0$ for $k > q$ and $\beta_{2,k} = 0$ for $k > q - 1$.

Proof

$$\begin{aligned} g_N(x) &= a_0 + \int_0^x \sum_{k=0}^{2q} \sum_{l=0}^k \beta_{1,l} \beta_{1,k-l} x^k + \sum_{k=0}^{2q-2} \sum_{l=0}^k \beta_{2,l} \beta_{2,k-l} x^k dx \\ &= a_0 + \sum_{k=0}^{2q} \frac{1}{k+1} \sum_{l=0}^k \beta_{1,l} \beta_{1,k-l} x^{k+1} + \sum_{k=0}^{2q-2} \frac{1}{k+1} \sum_{l=0}^k \beta_{2,l} \beta_{2,k-l} x^{k+1}. \end{aligned}$$

We thus have

$$a_{k+1} = \begin{cases} \frac{1}{k+1} \left(\sum_{l=0}^k \beta_{1,l} \beta_{1,k-l} + \sum_{l=0}^k \beta_{2,l} \beta_{2,k-l} \right) & \text{for } 0 \leq k \leq 2q - 2, \\ \frac{1}{k+1} \sum_{l=0}^k \beta_{1,l} \beta_{1,k-l} & \text{for } k = 2q - 1, 2q. \end{cases}$$

We recall that the martingality condition implies

$$a_0 = F(t, T) - \sum_{k=1}^q a_{2k} (2k - 1)!!.$$

We have

$$\frac{\partial a_0}{\partial \beta_{l,j}} = - \sum_{k=1}^q (2k - 1)!! \frac{\partial a_{2k}}{\partial \beta_{l,j}},$$

and

$$\begin{aligned} \frac{\partial a_{k+1}}{\partial \beta_{1,j}} &= \frac{1}{k+1} 2\beta_{1,k-j} \text{ for } j \leq k \leq 2q, \\ \frac{\partial a_{k+1}}{\partial \beta_{2,j}} &= \frac{1}{k+1} 2\beta_{2,k-j} \text{ for } j \leq k \leq 2q - 2, \\ \frac{\partial a_{k+1}}{\partial \beta_{l,j}} &= 0 \text{ for } k < j \text{ and } l = 1, 2. \end{aligned}$$

Thus,

$$\frac{\partial a_0}{\partial \beta_{l,j}} = - \sum_{k=1}^q \frac{(2k-1)!!}{k} \beta_{l,2k-j-1},$$

with $\beta_{l,k} = 0$ for $k < 0$ and $\beta_{1,k} = 0$ for $k > q$ and $\beta_{2,k} = 0$ for $k > q - 1$. □

In particular, for a cubic polynomial, we have

$$\begin{aligned} \frac{\partial g_3}{\partial \beta_{1,0}}(x_i) &= 2\beta_{1,0}x_i + \beta_{1,1}x_i^2 - \beta_{1,1}, & \frac{\partial g_3}{\partial \beta_{2,0}}(x_i) &= 2\beta_{2,0}x_i, \\ \frac{\partial g_3}{\partial \beta_{1,1}}(x_i) &= \beta_{1,0}x_i^2 + \frac{2\beta_{1,1}}{3}x_i^3 - \beta_{1,0}, \end{aligned}$$

and for a quintic polynomial,

$$\begin{aligned} \frac{\partial g_5}{\partial \beta_{1,0}}(x_i) &= 2\beta_{1,0}x_i + \beta_{1,1}x_i^2 + \frac{2\beta_{1,2}}{3}x_i^3 - \beta_{1,1}, \\ \frac{\partial g_5}{\partial \beta_{2,0}}(x_i) &= 2\beta_{2,0}x_i + \beta_{2,1}x_i^2 - \beta_{2,1}, \\ \frac{\partial g_5}{\partial \beta_{1,1}}(x_i) &= \beta_{1,0}x_i^2 + \frac{2\beta_{1,1}}{3}x_i^3 + \frac{\beta_{1,2}}{2}x_i^4 - \beta_{1,0} - \frac{3}{2}\beta_{1,2}, \\ \frac{\partial g_5}{\partial \beta_{2,1}}(x_i) &= \beta_{2,0}x_i^2 + \frac{2\beta_{2,1}}{3}x_i^3 - \beta_{2,0}, \\ \frac{\partial g_5}{\partial \beta_{1,2}}(x_i) &= \frac{2\beta_{1,0}}{3}x_i^3 + \frac{\beta_{1,1}}{2}x_i^4 + \frac{2\beta_{1,2}}{5}x_i^5 - \frac{3}{2}\beta_{1,1}. \end{aligned}$$

The cubic polynomial initial guess can be rewritten in the isotonic form as follows,

$$\begin{aligned} a_0 + a_1x + a_2x^2 + a_3x^3 &= a_0 + \int_0^x (a_1 + 2a_2t + 3a_3t^2) dt \\ &= a_0 + \int_0^x \left(\sqrt{3a_3t} + \frac{a_2}{\sqrt{3a_3}} \right)^2 + \left(\sqrt{a_1 - \frac{a_2^2}{3a_3}} \right)^2 dt. \end{aligned} \tag{24}$$

Based on the initial guess (refined or cubic), we can use a standard unconstrained Levenberg–Marquardt algorithm to minimize the measure M_V , based on the isotonic parameterization. This results in the optimal coefficients $(\beta_{1,0}, \dots, \beta_{1,q})$ and $(\beta_{2,0}, \dots, \beta_{2,q-1})$, which we then convert back to a standard polynomial representation, as described above.

The gradient of the call prices toward the isotonic parameters can also be computed analytically from Eq. (2), as we have

$$\frac{\partial C}{\partial \beta_{l,j}}(K) = \frac{\partial x_K}{\partial \beta_{l,j}}(K)C(K) + \int_{x_K}^\infty \frac{\partial g_N}{\partial \beta_{l,j}}(x)\phi(x)dx,$$

where x_K is the integration cutoff point defined by Eq. (3). As $\frac{\partial g_N}{\partial x}(\boldsymbol{\beta}, x_K) > 0$, we can use the implicit function theorem to compute the partial derivatives $\frac{\partial x_K}{\partial \beta_{i,j}}(K)$:

$$\nabla x_K(\boldsymbol{\beta}) = -\frac{1}{\frac{\partial g_N}{\partial x}(\boldsymbol{\beta}, x_K)} \nabla g_N(\boldsymbol{\beta}, x_K)$$

where $\nabla x_K = \left(\frac{\partial x_K}{\partial \beta_{1,0}}, \dots, \frac{\partial x_K}{\partial \beta_{2,q-1}} \right)$ and $\nabla g_N = \left(\frac{\partial g_N}{\partial \beta_{1,0}}, \dots, \frac{\partial g_N}{\partial \beta_{2,q-1}} \right)$.

4 Examples of equity index smiles

We consider a set of vanilla option prices on the same underlying asset, with the same maturity date. As an illustrating example, we will use SPX500 option quotes expiring on March 7, 2018, as of February 5, 2018, from Appendix D. The options' maturity is thus nearly one month. The day before this specific valuation date, a big jump in volatility across the whole stock market occurred. One consequence is a slightly more extreme (but not exceptional) volatility smile.

4.1 A short review of implied volatility interpolations

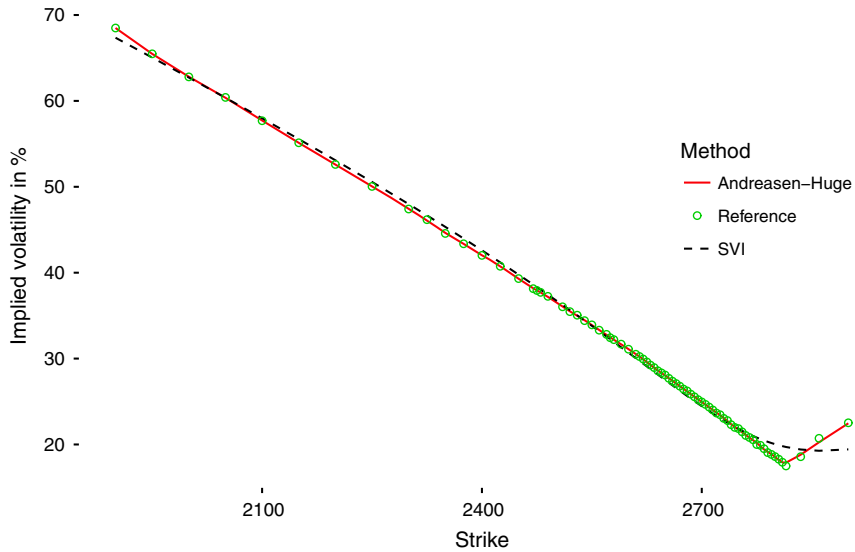
Let us recall shortly some of the different approaches to build an arbitrage-free implied volatility interpolation, or equivalently, to extract the risk-neutral probability density.

We can choose to represent the asset dynamics by a stochastic volatility model such as Heston (1993), Bates (1996), Double-Heston (Christoffersen et al. 2009). This implies a relatively high computational cost to obtain vanilla option prices and thus to calibrate the model, especially when time-dependent parameters are allowed. Furthermore, those models are known to not fit adequately the market of vanilla options with short maturities. Their implied volatility smile is typically too flat.

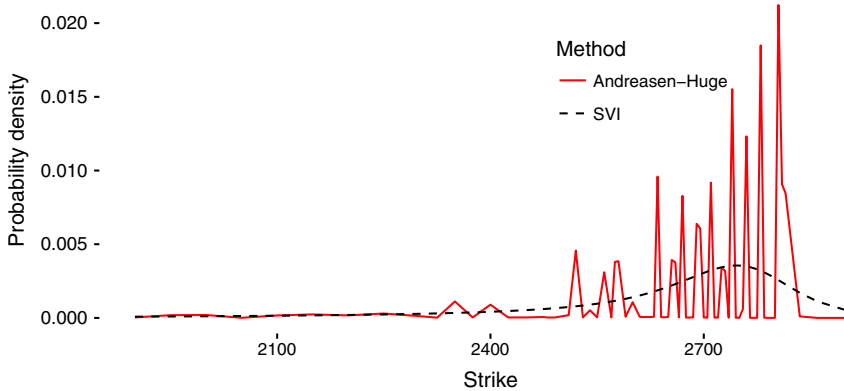
Many practitioners revert to a parameterization-based or inspired from a stochastic volatility model, such as the Hagan SABR expansion (Hagan et al. 2002), or the Gatheral SVI model (Gatheral and Lynch 2004; Flint and Maré 2017). These are much faster to calibrate. SVI is one of the most popular parameterizations to represent the equity option volatility smile, because of its simplicity, its relation to stochastic volatility models asymptotically, and its almost arbitrage-free property. However, as we shall see, the fit for options on equities can still be poor (Fig. 1a). SVI manages to fit only a part of the left wing and fails to represent well the market quotes in the region of high implied volatility curvature. SVI and SABR are usually much better at fitting longer option maturities.

Another approach is not to assume any underlying model and use an exact interpolation. A cubic spline interpolation of the implied volatilities is not arbitrage-free, although it is advocated⁶ by M. Malz (2014). Kahalé (2004) proposes an arbitrage-free spline interpolation of the option prices. Unfortunately, it is not guaranteed that

⁶ Malz precises that the challenge of his approach is to find a good filter for the quotes, which he does not describe at all.



(a) Implied volatility.



(b) Probability density.

Fig. 1 SVI and Andraesen–Huge calibrations on 1m SPX500 options as of February 05, 2018

his algorithm for C^2 interpolation, necessary for a continuous probability density, converges. Furthermore, it assumes that the input call option quotes are convex and decreasing by strike. But the market quotes are not convex in general, mainly because of the bid–ask spread. While we propose a quadratic programming-based algorithm to build a convex set that closely approximates the market prices in Appendix C, it can be relatively slow when the number of quotes is large. Finally, the resulting implied probability density will be noisy, as evidenced by Syrdal (2002).

A smoothing spline or a least-squares cubic spline will allow to avoid overfitting the market quotes. For example, Syrdal (2002) and Bliss and Panigirtzoglou (2004)

use a smoothing spline on the implied volatilities as a function of the option deltas,⁷ with flat extrapolation. Smoothing is ensured by adding a penalty multiplied by the integral of the second derivative of the spline function to the objective function. The smoothing parameter is challenging to pick. In fact, Syrdal reverts to a manual ad hoc choice of this parameter. Furthermore, the interpolation is a priori not arbitrage-free. In order to make it arbitrage-free, an additional nonlinear penalty term against butterfly spread arbitrages needs to be added to the objective function. Instead of a smoothing parameter, the least-squares spline requires to choose the number of knots and their locations, which can also seem arbitrary. A slightly different approach is taken by Wystup (2015, p. 47) for the foreign exchange options market, where a Gaussian kernel smoother is applied to the market volatilities as a function of the option delta. The kernel bandwidth is fixed and the number of kernel points (specific deltas) is typically lower than the number of market option strikes. Wystup recommends to use up to 7 kernel points. While a larger number of points leads to a better fit on our example, it may also lead to a negative density. With the Gaussian kernel smoothing, the shape of the implied volatility looks unnatural⁸ around the point of high curvature (Fig. 2a), and the density can become negative (Fig. 2b). It is thus not always arbitrage-free.

In order to guarantee the arbitrage-free property by construction and still stay model-free, Andreasen and Huge (2011) use a specific one-step implicit finite difference where a discrete piecewise constant local volatility function is calibrated against market prices. While it is simple and fast, it leads to a noisy implied density, even if we replace the piecewise constant parameters by a cubic spline (Fig. 1b). This is because, by design, similarly to a spline interpolation, the method overfits the quotes as the number of parameters is the same as the number of market option quotes.

Finally, we can model directly the risk-neutral probability density (RND). Many papers use the double lognormal mixture of Bahra (Bahra 2007; Arneric et al. 2015) to represent the RND. The double lognormal mixture is not flexible enough to capture our example of short-maturity smile (Fig. 3a). This is extended in Cheng (2010) to a mixture of multiple lognormal distributions. With a mixture of 6 lognormal distributions, the root-mean-square error of the model volatilities against market volatilities is nearly as low as with Andreasen–Huge (Table 2), and the RND is very smooth. Furthermore, the model is also fully arbitrage-free by construction. But, in similar fashion as the Gaussian kernel smoother (Wystup 2015), the mixture of lognormal distributions tends to create artificial peaks in the RND (Figs. 2b, 3b), just to fit the input quotes better on our example.

Compared to the Gaussian kernel smoothing, the mixture of lognormal distributions results in fewer peaks as the volatility of each distribution is optimized, but there are still clearly multiple modes in the density. In reality, as we will see with the collocation method, the density has no particular reasons to have multiple modes. Mixture of normal or lognormal distributions will, by their nature, tend to create multimodal densities.

⁷ Especially for options on a foreign exchange rate, the implied volatility may be parameterized as a function of the option delta, instead of the option strike. Note that the delta is itself a function of the implied volatility, see Appendix A.

⁸ This could be remedied by a kernel smoothing on the strikes instead of the deltas, but then the probability density goes negative in more places. We thus preferred to stay close to Wystup's original idea.

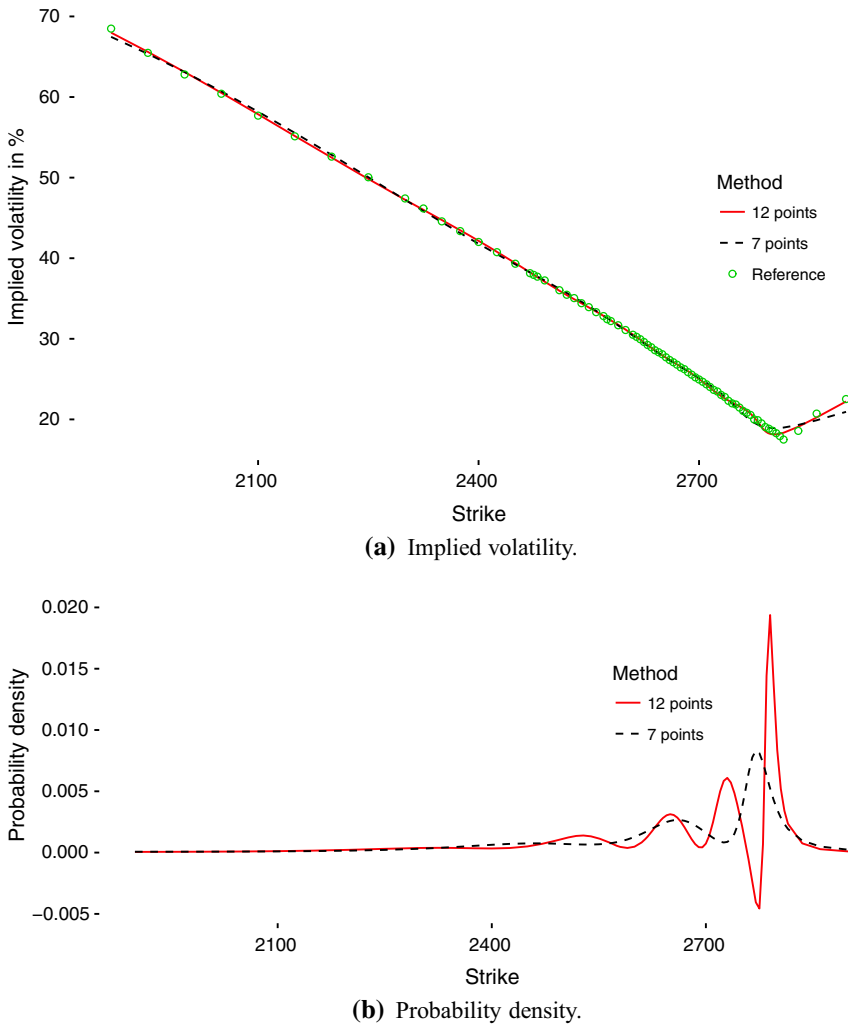


Fig. 2 Gaussian kernel smoothing calibrations on 1m SPX500 options as of February 05, 2018

Let us give more details about the setup of each technique on our example. For the SVI parameterization, we use the quasi-explicit calibration method described in Zeliade Systems (2009), which leads to the parameters of Table 3. For the Andreasen–Huge method, we use a dense log-uniform grid composed of 800 points and solve the probability density (the Fokker–Planck equation) instead of the call prices. We then interpolate in between grid points by integrating the density to obtain the call option prices to preserve the arbitrage-free property everywhere in a similar spirit as (Hagan et al. 2014; Le Floc'h and Kennedy 2017). We minimize the error measure M_V with the Levenberg–Marquardt algorithm. The Gaussian kernel smoother calibration is described in Appendix A. We use, respectively, 7 and 12 kernel points, with a bandwidth of 0.5 and 0.3. When the bandwidth is too large for the number of kernel

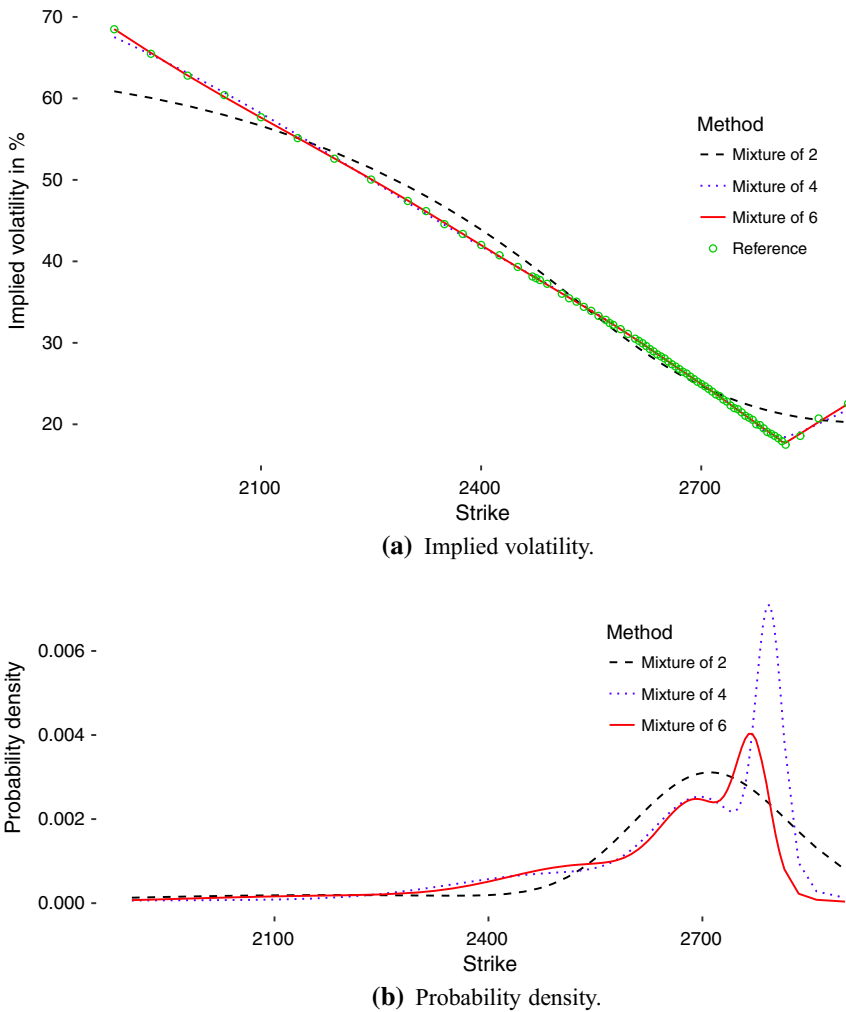


Fig. 3 Mixture of lognormal distributions calibration on 1m SPX500 options as of February 05, 2018

points considered, the problem becomes ill-conditioned numerically as the optimal kernel weights α become very large numbers (Table 4), while the result of the kernel is a volatility of the order of 10%. Our multiple lognormal mixture optimization is described in Appendix B and the calibrated parameters in Table 10. Note that the optimization problem becomes very challenging numerically for a mixture of more than 7 lognormal distributions, because, on the one side, the number of free parameters is relatively high ($3n - 2$ free parameters for n lognormal distributions) and on the other side, the highly nonlinear structure of the problem creates multiple local minima.

Table 2 Root-mean-square error (RMSE) of the calibrated model volatilities against the market volatilities

Model	Free parameters	RMSE
SVI	5	0.00757
Andreasen–Huge	75	0.00088
Gaussian smoothing kernel	7	0.00400
Gaussian smoothing kernel	12	0.00175
Mixture of 2 lognormals	4	0.01807
Mixture of 4 lognormals	10	0.00252
Mixture of 6 lognormals	16	0.00094

Table 3 Parameters resulting of the calibration of the SVI model $\sigma^2(k) = a + b\{\rho(k - m) + \sqrt{(k - m)^2 + s^2}\}$ against 1m SPX500 options as of February 05, 2018, with $k = \ln \frac{K}{F}$

a	b	ρ	s	m
0.000	0.794	-0.492	0.0537	0.0554

Table 4 Optimal kernel observations α_i resulting of the Gaussian kernel smoothing against 1m SPX500 options as of February 05, 2018, for different bandwidths λ

Number of points	λ	$\min \alpha_i $	$\max \alpha_i $
7	0.5	4.8E03	1.1E05
12	0.292	2.5E05	1.3E08
12	0.5	1.6E10	1.2E13

5 Polynomial collocation of SPX500 options

Previous literature has explored the calibration of stochastic collocation against market quotes for interest rates swaptions, in the case of the SABR model in Grzelak and Oosterlee (2017) as well as for FX options. In both cases, the set of quotes is relatively small (usually less than 10) and the risk of arbitrage in the quotes, related to the bid–ask spread size, is very low. In the world of equity options, the quotes are denser (it is not unusual to have 50 quotes for liquid equity indices), or less liquid, and thus have a higher probability of containing small theoretical arbitrages.

Here, we consider the quotes of vanilla options on the index SPX500, expiring on February 23, 2018, as of January 23, 2018, for 174 distinct strikes. We first take a look at the cubic polynomial guess, least-squares quintic polynomial guess, and optimal quintic polynomial that are calibrated to the market quotes. Figure 4 shows that without any preprocessing, the cubic and quintic polynomial guesses are of relatively poor quality, because of a few outliers. Filtering out the problematic quotes by imposing hard boundaries on the resulting slope estimates is enough to fix this (see Fig. 5b). The preprocessing to produce a convex set of quotes through quadratic programming results in a better quintic polynomial guess but a worse cubic polynomial initial guess (Fig. 5a, b). This is because the cubic polynomial guess has not been refined: It has not been optimized with the monotonicity constraint. Otherwise, it would fit better than the cubic polynomial optimized against our simple filtered quotes. The difference in

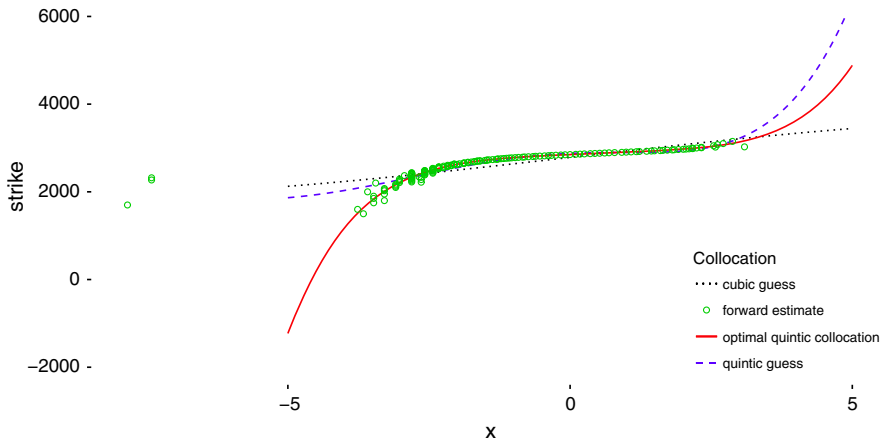


Fig. 4 Strike as a function of $x = \Phi^{-1}(p)$ where p is the cumulative density for SPX500 options expiring on February 23, 2018, without applying any convexity filter

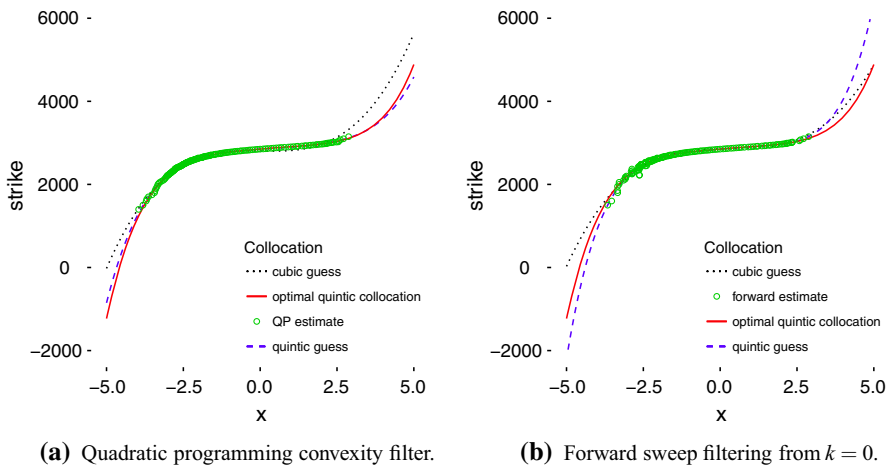


Fig. 5 Strike as a function of $x = \Phi^{-1}(p)$ where p is the cumulative density for SPX500 options expiring on February 23, 2018, using a filter on the quotes

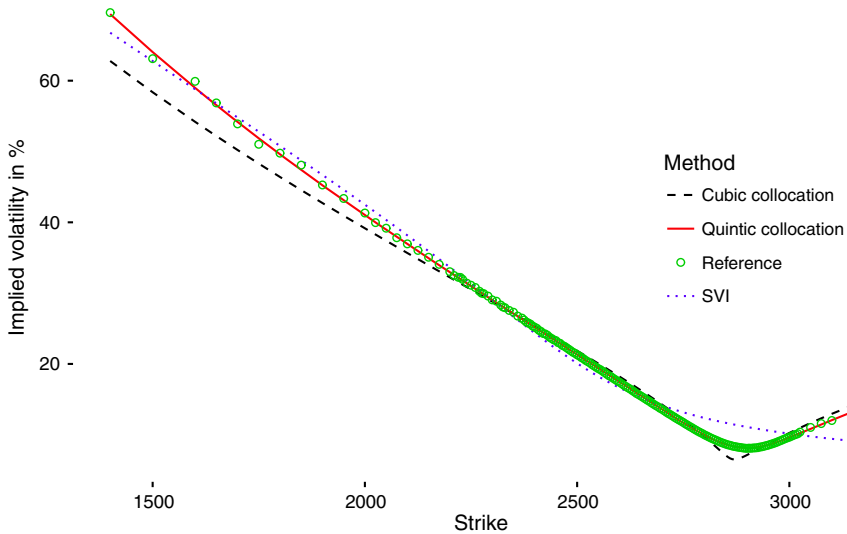
the polynomial guesses between the two methods is, however, not large. Table 5 shows that the first three moments⁹ of the quintic refined guess with the forward sweep filter or with the convexity filter are close to the moments corresponding to the optimal quintic polynomial collocation. On this example, the convexity filter improves the estimate of the kurtosis significantly.

Let us take a look at the quality of the fit in terms of implied volatilities. In Figs. 6 and 7, the reference implied volatilities include all market options, i.e., they are unfiltered, even when afterward, we process those in order to apply the collocation method.

⁹ Here, we calculate the statistics of the underlying asset price distribution, as implied from the option prices. We are not interested in the statistics of the market options prices themselves.

Table 5 Mean, variance, skew and kurtosis corresponding to quintic polynomial collocation of SPX500 1m options as of February 5, 2018, for different filtering of market quotes

Collocation polynomial	Mean	Variance	Skew	Kurtosis
Refined quintic guess on raw quotes	2839.00	102.59	-1.59	14.72
Refined quintic guess with forward sweep	2839.00	93.36	-2.74	58.97
Refined quintic guess on convex quotes	2839.00	88.06	-2.63	38.87
Optimal quintic polynomial collocation	2839.00	88.92	-2.71	38.52

**Fig. 6** SPX500 smile

On our SPX500 option quotes from January 23, 2018, Fig. 6 shows that Gatheral SVI parameterization does not fit very well. While SVI is generally quite good at fitting medium and long maturities, it is often not very well suited for short maturities such as the 1-month maturity we consider here. The cubic collocation, which has fewer parameters fits better than SVI, and the quintic collocation provides a nearly perfect fit on this example, which is impressive since it has the same number of free parameters as SVI.

We now consider the same SPX500 option quotes expiring on March 7, 2018, as of February 5, 2018, as in Sect. 4.1, and is listed in Appendix D. The option maturities are still one month, but the valuation day corresponds to one day after a big jump in volatility across the whole stock market. The smile is more complex. Figure 7 shows that the curvature is much higher at the lowest point and that the left wing is slightly concave. SVI manages to fit only a part of the left wing and fails to represent well the market quotes in the region of high curvature. The quintic polynomial achieves a reasonably good fit. In order to illustrate that our calibration technique still works well with a higher degree polynomial, we also calibrate a nonic polynomial. It results in a nearly perfect fit, despite the very strong curvature.

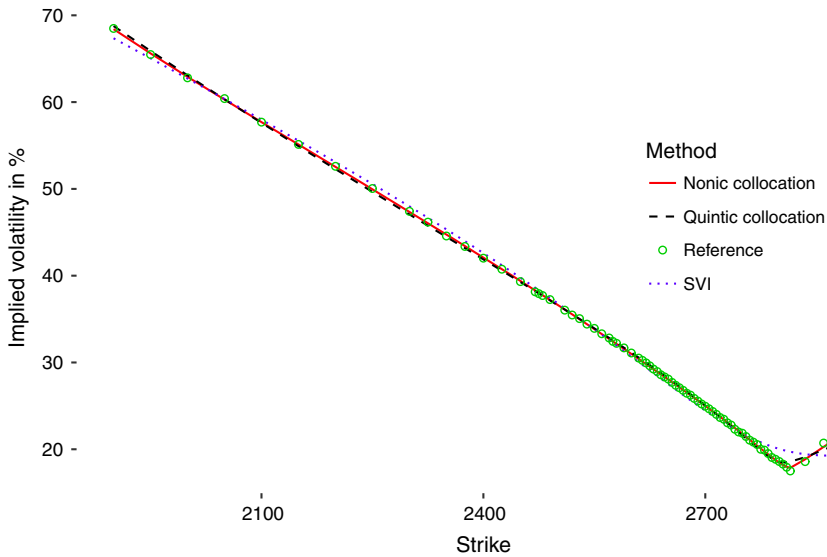


Fig. 7 Implied volatility smile of SPX500 1m options as of February 5, 2018

We show the corresponding probability density in Fig. 8 and observe a high and narrow spike in the region where the implied volatility has a strong curvature. Contrary to the stochastic collocation method, SVI does not allow to capture this spike properly. The density is also markedly different from the one obtained by the mixture of lognormal distributions in Fig. 3b. In particular, it does not exhibit multiple peaks. This stays true for collocations on a polynomial of higher degree. The root-mean-square error (RMSE) of implied volatilities with a cubic polynomial collocation is smaller than with SVI which has two more free parameters (Table 6 against Table 2). The nonic polynomial collocation has a RMSE similar to the results from the Andreasen–Huge method and the mixture of 6 lognormal distributions model, while the collocation has again significantly fewer free parameters.

So far, we have considered individual smiles corresponding to a specific option maturity. In order to build a full volatility surface, a linear interpolation of the collocating polynomial in between maturities will preserve the arbitrage-free property across strikes as the intermediate polynomial will still be monotonic. Such an interpolation could, however, introduce arbitrage in time, as it does not guarantee, a priori, that the call prices will increase with the maturity for a fixed log-moneyness. The alternative is to fit each expiry independently and interpolate linearly in total variance across a fixed log-moneyness.

6 Calibration of CMS convexity adjustments

Here, we will evaluate the quality of the stochastic collocation method on the problem of calibrating the constant maturity swaps jointly with the interest rate swaptions.

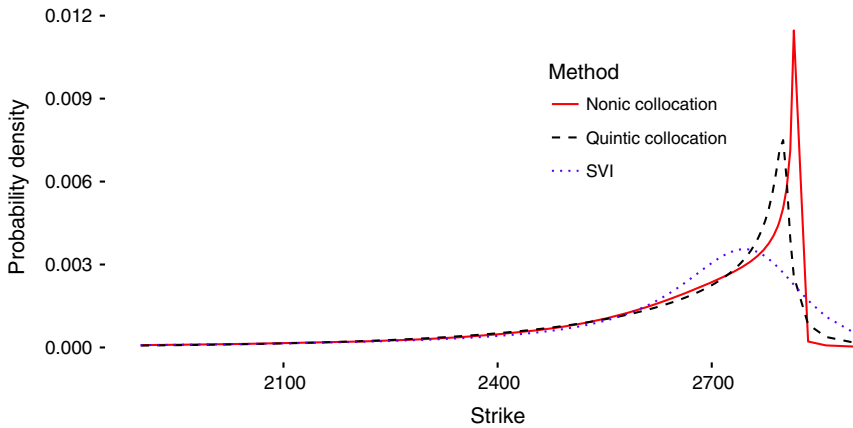


Fig. 8 Implied probability density of SPX500 1m options as of February 5, 2018

Table 6 Root-mean-square error (RMSE) of the collocation method implied volatilities against the market implied volatilities of the SPX500 1m options as of February 5, 2018

Model	Free parameters	RMSE
Cubic collocation	3	0.00538
Quintic collocation	5	0.00280
Nonic collocation	9	0.00110
11th degree collocation	11	0.00099

A constant maturity swap (CMS) is a swap where one leg refers to a reference swap rate which fixes in the future (for example, the 10 year swap rate), rather than to an interest rate index such as the LIBOR. The other leg is a standard fixed or floating interest rate leg.

For a maturity T_a , the forward swap rate $S_{a,b}$ at time t with payments at T_{a+1}, \dots, T_b is

$$S_{a,b}(t) = \frac{P(t, T_a) - P(t, T_b)}{\tau \sum_{j=a+1}^b P(t, T_j)}, \tag{25}$$

where τ is the year fraction for a period ($\tau = 1$ in the 30/360 daycount convention in our examples) and $P(t, T)$ is the discount factor from t to T .

In the $T_a + \delta$ forward measure, the convexity adjustment for the swap rate $S_{a,b}$ is

$$CA(S_{a,b}, \delta) = \mathbb{E}^{T_a+\delta} [S_{a,b}(T_a)] - S_{a,b}(0), \tag{26}$$

where δ is the accrual period of the swap rate. It depends on the entire evolution of the yield curve. Following Hagan’s standard model (Hagan 2003), Mercurio and Pallavicini (2005, 2006) approximate the convexity adjustment by using a linear function of the underlying swap rate for the Radon–Nikodym derivative in order to express

the value in the forward swap measure associated with $S_{a,b}$. This leads to

$$CA(S_{a,b}, \delta) \approx S_{a,b}(0)\theta(\delta) \left(\frac{\mathbb{E}^{a,b} \left[S_{a,b}^2(T_a) \right]}{S_{a,b}^2(0)} - 1 \right), \tag{27}$$

where

$$\theta(\delta) = 1 - \frac{\tau S_{a,b}(0)}{1 + \tau S_{a,b}(0)} \left(\frac{\delta}{\tau} + \frac{b - a}{(1 + \tau S_{a,b}(0))^{b-a} - 1} \right). \tag{28}$$

Now $\mathbb{E}^{a,b} \left[S_{a,b}^2(T_a) \right]$ can be derived from the market swaption prices. In Hagan (2003) and Mercurio and Pallavicini (2005, 2006), the replication method is used. We will see in Sect. 7 that this expectation has a very simple closed-form expression with the collocation method. This will simplify and speed up the calibration.

As explained in Mercurio and Pallavicini (2006), the market quotes the spreads $X_{m,c}$ which sets to zero the no-arbitrage value of CMS swaps starting today and paying the c -year swap rate $S'_{i,c}$ from t_{i-1} to $t_{i-1} + c$ with $t_0 = 0$. We have

$$X_{m,c} = \frac{\sum_{i=1}^m \left(S'_{i,c}(0) + CA(S'_{i,c}, \delta) \right) P(0, t_i)}{\sum_{i=1}^m P(0, t_i)} - \frac{1 - P(0, t_m)}{\delta \sum_{i=1}^m P(0, t_i)}. \tag{29}$$

We will consider the example market data from Mercurio and Pallavicini (2006) as of February 3, 2006, where δ corresponds to a quarter year and the CMS leg is expressed in ACT/360 while the floating leg and the spread are in 30/360 daycount convention.

In order to compute the CMS spread $X_{m,c}$, the convexity adjustments are needed for many dates not belonging to the market swaption expiries. We follow Mercurio and Pallavicini (2006) and interpolate the convexity adjustments at the swaption expiries by a cubic spline (with an adjustment of zero at $t = 0$).

The market swaptions are expressed in Black volatility and can be priced through the standard Black formula on the forward swap rate.

Mercurio and Pallavicini (2005, 2006) describe a global calibration, where the swaption volatility errors for each market swaption expiry and strike, plus a penalty factor multiplied by the CMS spread error for each quoted market CMS spread, are minimized in a least-squares fashion. This is a single high-dimensional minimization. We will see how to apply this methodology to the collocation method, as well as a new alternative calibration.

7 Joint calibration of swaptions and CMS convexity adjustments with the stochastic collocation

In Sect. 6, we have described the calibration of swaptions and CMS convexity adjustments from Mercurio and Pallavicini (2006). A key estimate for their method is an approximation of the expectation $\mathbb{E}^{a,b} \left[S_{a,b}^2(T_a) \right]$ from Eq. (27). Instead of using the

replication method, when a collocation polynomial is calibrated to the market swaption prices, this expectation can be computed by a direct integration:

$$\mathbb{E}^{a,b} \left[S_{a,b}^2(T_a) \right] = \int_{-\infty}^{\infty} g_N^2(x) \phi(x) dx. \quad (30)$$

The coefficients (b_0, \dots, b_{2N}) of g_N^2 correspond to the self-convolution of the coefficients (a_0, \dots, a_N) of g_N . Similarly to the calculation of the first moment, the recurrence relation [Eq. (4)] leads to the closed-form expression

$$\mathbb{E}^{a,b} \left[S_{a,b}^2(T_a) \right] = b_0 + \sum_{i=1}^N (2i-1)!! b_{2i}. \quad (31)$$

There are two ways to include the CMS spread in the calibration of the smile to the swaption quotes, which we will label as the global and the decoupled approach. The collocation method can be used in the global approach from Mercurio and Pallavicini (2006), by first computing an initial guess in the form of a list of isotonic representations, out-of-the-market swaption quotes for each expiry according to Sect. 3. Then, the least-squares minimization updates the isotonic representations iteratively.

Tables 7 and 8 show that the error with a quintic collocation polynomial is as low as with the SABR interpolation.¹⁰ Compared to SABR interpolation, however, the advantages of stochastic collocation are: it is arbitrage-free by construction, while the SABR approximation formula has known issues with low or negative rates (Hagan et al. 2014); the accuracy of the fit can be improved by simply increasing the collocating polynomial degree; the calculation of the CMS convexity adjustment is much faster as it does not involve an explicit replication.

The decoupled calibration procedure consists of the following two steps:

- (A) Find the optimal convexity adjustment for each market swaption expiry
 - Compute the initial guess for each convexity adjustment by fitting the swaption smile at each expiry as described in Sect. 3, without taking into account any CMS spread price.
 - Compute the cubic spline which interpolates the convexity adjustments across the expiries. Minimize the square root of CMS spread errors, by adjusting the convexity adjustments and recomputing the cubic spline, for example, with the Levenberg–Marquardt algorithm.
- (B) Minimize the square of the swaption volatility error plus a penalty factor multiplied by the convexity adjustment error, against the optimal convexity adjustment for each expiry independently.

The penalty factor allows to balance the swaption volatilities fit with the CMS spread fit. The decoupled calibration involves $n + 1$ independent low-dimensional minimizations, with n being the number of swaptions expiries.

¹⁰ The SABR numbers come from Mercurio and Pallavicini (2006). For the Black model, our numbers differ slightly from Mercurio and Pallavicini, likely because of the handling of holidays.

Table 7 Absolute differences (in bp) between market and model swaptions implied volatilities

Strike	-200	-100	-50	-25	0	25	50	100	200
SABR									
5 × 10	1.1	0.2	0.3	0.5	0.2	0.2	0.8	0.2	0.4
10 × 10	0.2	0.1	0.4	0.3	0.1	0.2	0.4	0.5	0.3
20 × 10	0.8	0.6	0.4	0.4	0	0.3	0.3	0.5	0.5
Quintic global									
5 × 10	0.1	-0.4	0.3	0.4	0.2	-0.5	-0.3	0.3	0.0
10 × 10	0.4	-1.8	0.4	1.0	1.3	0.2	-0.5	-1.7	1.0
20 × 10	0.7	-1.7	0.8	1.2	1.1	0.5	-0.7	-1.8	1.3
Quintic decoupled									
5 × 10	0.0	-0.3	0.2	0.3	0.2	-0.5	-0.3	0.4	-0.1
10 × 10	0.1	-0.6	0.2	0.3	0.6	-0.2	-0.3	-0.4	0.3
20 × 10	0.4	-2.0	0.6	1.0	1.0	0.5	-0.6	-1.6	0.9

Strikes are expressed as absolute differences in basis points w.r.t the at-the-money values

Table 8 Absolute differences (in bp) between market CMS swap spreads and those induced by the SABR functional form, the Black model and the collocation of a quintic polynomial for the 10-year tenor

Maturity (years)	SABR	Black	Quintic global	Quintic decoupled
5	0.8	0.0	0.3	0.3
10	1.7	2.3	1.2	1.4
15	1.8	3.2	1.3	1.7
20	1.3	4.0	1.3	1.8
30	2.1	6.3	2.3	1.9

Table 9 CMS convexity adjustments (in bp) for different expiries for the 10-year tenor

Expiry	1 years	5 years	10 years	20 years	30 years
Optimal CA (step A)	1.75	10.62	20.24	35.02	49.41
Decoupled CA (step B)	1.75	10.63	20.52	37.14	60.41

Table 9 presents the optimal convexity adjustments for CMS swaps of tenor 10 years using the market data of Mercurio and Pallavicini resulting from the first step of the decoupled calibration method. With a cubic spline interpolation on these adjustments, the error in market CMS spreads is then essentially zero. The adjustments resulting from the second step are also displayed for indication.

As evidenced in Tables 7 and 8, the error in the swaptions volatilities and in the CMS spreads is as small as, or smaller than the decoupled calibration when compared to the global calibration. In Fig. 9, we look at the smile generated by the quintic collocation calibrated with a penalty of 1 (which corresponds to a balanced fit of

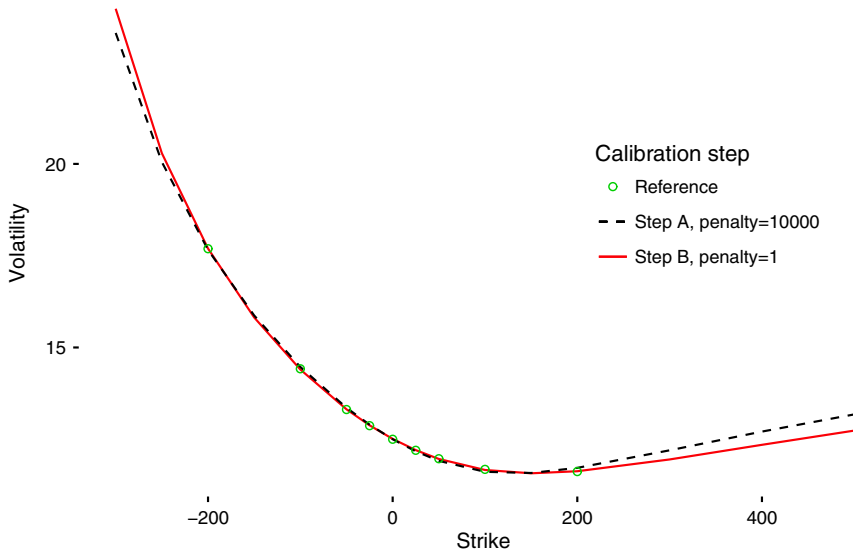


Fig. 9 20×10 swaption smile, calibrated with a penalty factor of 10,000 (exact CMS spread prices) and a penalty factor of 1 (balanced fit corresponding to Table 8)

market CMS spreads vs. swaption volatilities as in Table 8) and a penalty of 10,000 (which corresponds to a nearly exact fit to the CMS spreads).

Instead of using a monotonic quintic polynomial, we could have used a monotonic cubic polynomial with quadratic left and right C^1 -extrapolation. Two parameters of each extrapolation would be set by the value and slope continuity conditions, and the remaining extra parameter could be used to calibrate the tail against the CMS prices. Overall, it would involve the same number of parameters to calibrate and would likely be more flexible. The calibration technique would, however, remain the same: All the parameters would need to be recalibrated as changes in the extrapolation result in changes in the first two moments of the distribution as well.

8 Limitations of the stochastic collocation

A question which remains is whether the stochastic collocation method can also fit multimodal distributions well?

Theorem 1 *For any continuous survival distribution function G_Y , there exists a stochastic collocation polynomial g_N which can approximate the survival distribution to any given accuracy $\epsilon > 0$ across an interval $[a, b]$ of \mathbb{R} .*

Proof The function $g(x) = G_Y^{-1}(1 - \Phi(x))$ is continuous and monotone on \mathbb{R} . Wolibner (1951) and Lorentz and Zeller (1968) have shown that for any $\eta > 0$, there exists a monotone polynomial $g_{N,\eta}$ such that $\sup_{x \in [a,b]} \|g(x) - g_{N,\eta}(x)\| \leq \eta$ on any interval $[a, b]$ of \mathbb{R} . From Eq. (2), the approximate survival distribution corresponding to the

collocation polynomial $g_{N,\eta}$ is $G_{N,\eta} = 1 - \Phi \circ g_{N,\eta}^{-1}$ where the symbol \circ denotes the function composition.

Let $I = [g_{N,\eta}(a), g_{N,\eta}(b)]$, $\Phi \circ g_{N,\eta}^{-1}$ and thus $G_{N,\eta}$ are also monotone and continuous on I . Let $J = [1 - \Phi(b), 1 - \Phi(a)]$, and $h_{N,\eta} = g_{N,\eta} \circ \Phi^{-1} \circ (1 - id)$ where id is the identity function. We have $G_Y^{-1} = g \circ \Phi^{-1} \circ (1 - id)$. As $\Phi^{-1} \circ (1 - id)$ is continuous and monotone, we have

$$\sup_{y \in J} \|G_Y^{-1}(y) - h_{N,\eta}(y)\| \leq \eta.$$

As h_N^{-1} is continuous and monotone on I , we also have

$$\sup_{x \in I} \|G_Y^{-1} \circ h_{N,\eta}^{-1}(x) - x\| \leq \eta. \tag{32}$$

The uniform continuity of G_Y implies that for each $\epsilon > 0$, we can find an $\eta > 0$ such that, if Eq. (32) holds, then

$$\sup_{x \in I} \|h_{N,\eta}^{-1}(x) - G_Y(x)\| \leq \epsilon.$$

□

In reality, simple multimodal distributions can be challenging to approximate in practice, as they might require a very high degree of the collocation polynomial for an accurate representation. In order to illustrate this, we consider an equally weighted mixture of two Gaussian distributions of standard deviation 0.1 centered, respectively, at $f_1 = 0.8$ and $f_2 = 1.2$. We can price vanilla options based on this density, simply by summing the prices of two options using a forward at, respectively, f_1 and f_2 under the Bachelier model. We set the forward $F(0, T) = 1$ and consider 20 options of maturity $T = 1$ and equidistant strikes between 0.5 and 1.5.

The Black volatility smile implied by this model is absolutely not realistic (Fig. 10), but it is perfectly valid and arbitrage-free theoretically, and we can still calibrate our models to it.

Figure 11a shows that the cubic and quintic polynomial collocations do not allow to capture the bimodality at all. The nonic polynomial does better in this respect, but the implied distribution is still quite poor.

One solution, here, is to collocate on a monotonic cubic spline (Fig. 11b). The simple algorithms (Dougherty et al. 1989; Huynh 1993) to produce monotone cubic splines are only guaranteed to be of class C^1 . While the stochastic collocation can be applied to C^1 functions, the probability density will then only be of class C^0 . The more difficult issue is the proper choice of the knots of the spline. How many knots should be used? If we place a knot at each market strike, we will overfit and end up with many wiggles in the probability density, like with the Andreasen–Huge method (Fig. 1b).

Furthermore, the problem of optimizing the spline is numerically more challenging, given that the parameterization does not enforce automatically monotonicity or

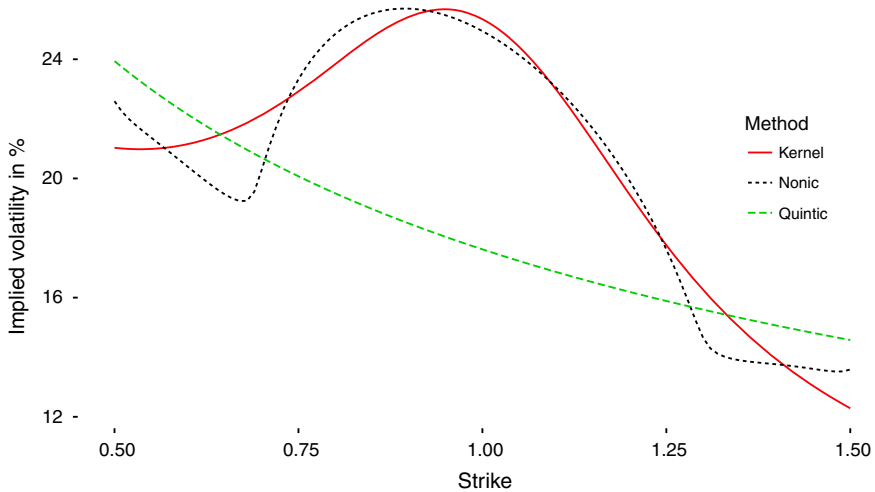


Fig. 10 Black implied volatility smile for a bimodal distribution and the calibrated stochastic collocation smiles with a quintic and a nonic polynomial

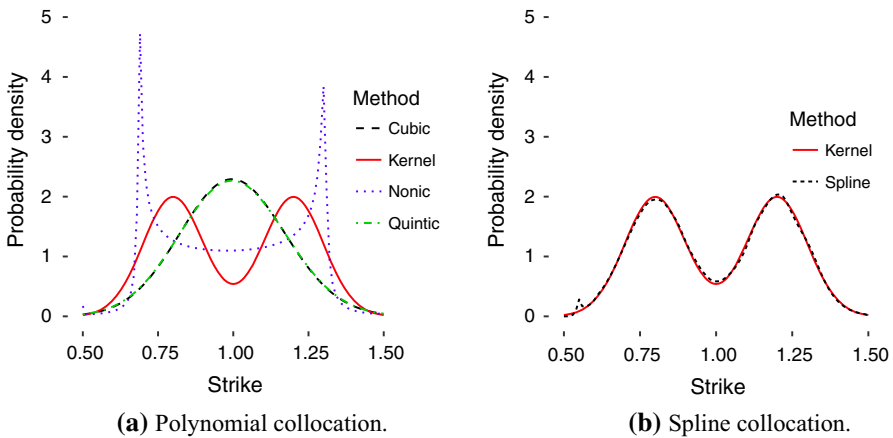


Fig. 11 Probability density obtained by the stochastic collocation of a reference bimodal distribution

martingality. Regarding monotonicity, in the case of the algorithm from Hyman (1983) or Dougherty et al. (1989), a nonlinear filter is applied, which could make the optimizer get stuck in a local minimum. Finally, there is the issue of which boundary conditions and which extrapolation to choose. A linear extrapolation, combined with the so-called natural boundary conditions, would result in a smooth density,¹¹ but the linear extrapolation still has to be of positive slope in order to guarantee the monotonicity over \mathbb{R} . A priori, this is not guaranteed. An alternative is to rely on the forward

¹¹ Although this makes the model relatively rigid toward the wings representation.

difference estimate as the slope, along with clamped boundary conditions, at the cost of losing the continuity of the probability density at the boundaries.

9 Conclusion

We have shown how to apply the stochastic collocation method directly to market options quotes in order to produce a smooth and accurate interpolation and extrapolation of the option prices, even on challenging equity options examples. A specific isotonic parameterization was used to ensure the monotonicity of the collocation polynomial as well as the conservation of the zeroth and first moments transparently during the optimization, guaranteeing the absence of arbitrages.

The polynomial stochastic collocation leads to a smooth implied probability density, without any artificial peak, even with high degrees of the collocation polynomial. We also applied the technique to interest rate derivatives. This results in closed-form formula for CMS convexity adjustments, which can thus be easily calibrated jointly with interest rate swaptions.

Finally, we illustrated, on a theoretical example, how the polynomial stochastic collocation had difficulties in capturing multimodal distributions.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

A Gaussian kernel smoothing

Let $(x_i, y_i)_{i=0, \dots, m}$ be $m + 1$ given points. Using the points $(\hat{x}_k)_{k=0, \dots, n}$, which are typically within the interval (x_0, x_m) , a smooth interpolation is given by the function g defined by

$$g(x) = \frac{1}{\sum_{k=0}^n \psi_\lambda(x - \hat{x}_k)} \sum_{k=0}^n \alpha_k \psi_\lambda(x - \hat{x}_k) \tag{33}$$

for $x \in \mathbb{R}$, $n \leq m$ and where ψ_λ is the kernel. The Gaussian kernel is defined by

$$\psi_\lambda(u) = \frac{1}{\lambda \sqrt{2\pi}} e^{-\frac{u^2}{2\lambda^2}}. \tag{34}$$

The calibration consists in finding the α_i that minimizes the given error measure on the input $(x_i, y_i)_{i=0, \dots, m}$. The solution can be found by QR decomposition of the linear problem

$$\begin{pmatrix} \psi_\lambda(x_0 - \hat{x}_0) & \psi_\lambda(x_0 - \hat{x}_1) & \dots & \psi_\lambda(x_0 - \hat{x}_n) \\ \vdots & \vdots & \dots & \vdots \\ \psi_\lambda(x_m - \hat{x}_0) & \psi_\lambda(x_m - \hat{x}_1) & \dots & \psi_\lambda(x_m - \hat{x}_n) \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_n \end{pmatrix}$$

$$= \begin{pmatrix} y_0 \sum_{k=0}^n \psi_{\lambda}(x_0 - \hat{x}_k) \\ \vdots \\ y_m \sum_{k=0}^n \psi_{\lambda}(x_m - \hat{x}_k) \end{pmatrix}. \tag{35}$$

The x_i are typically option strikes, log-moneynesses or deltas. The y_i are the corresponding option implied volatilities. Wystup (2015) applies the method in terms of option deltas. There are multiple delta definitions in the context of volatility interpolation; we use the undiscounted call option forward delta:

$$\Delta = \Phi \left(\frac{1}{\sigma \sqrt{T}} \ln \frac{F(0, T)}{K} + \frac{1}{2} \sigma \sqrt{T} \right) \tag{36}$$

where Φ is the cumulative normal distribution function, $F(0, T)$ is the forward to maturity T , K is the option strike and σ is the corresponding option volatility.

When the points are defined in delta, we need to find the delta for a given option strike in order to compute the option implied volatility or the option price for a given strike. But the delta is also a function of the implied volatility. This is a nonlinear problem. Equation 36, with $\sigma = g(\Delta)$, is solved by a numerical solver such as Toms348 (Alefeld et al. 1995) or a simple bisection, starting at $\Delta = 0.5$, in the interval $[0, 1]$.

Wystup recommends a bandwidth of 0.25; we find that a bandwidth of 0.5 minimizes the root-mean-square error of implied volatilities on our example.

B Mixture of lognormal distributions

Let $(x_i, y_i)_{i=0, \dots, m}$ be $m + 1$ given points. Using the points $(\hat{x}_k)_{k=0, \dots, n}$, which are typically within the interval (x_0, x_m) , a kernel estimate of the density on the points \hat{x}_k is

$$g(x) = \sum_{k=0}^n \alpha_k \psi_{\lambda_k}(x, \hat{x}_k), \tag{37}$$

where ψ_{λ_k} is a kernel of bandwidth λ_k , along with the condition $\sum_{k=0}^n \alpha_k = 1$ and $\alpha_k \geq 0$ for $k = 0, \dots, n$.

In our case, $(y_i)_{i=0, \dots, m}$ are option prices, and we want to estimate the underlying distribution. We then consider that $\psi_{\lambda_k}(x)$ is a Gaussian and x is a logarithmic function of the option strike K , $x = \ln(K)$. In order to map the kernel exactly to the Black–Scholes probability density, we use

$$\psi_{\lambda_k}(x, \hat{x}_k) = \frac{1}{\lambda_k \sqrt{2\pi}} e^{-\frac{(x - \hat{x}_k + \frac{1}{2} \lambda_k^2)^2}{2\lambda_k^2}}. \tag{38}$$

The option price of strike $K = e^x$ is then given by

$$V(x) = \sum_{k=0}^n \alpha_k V_{\text{Black}}(e^x, e^{\hat{x}_k}, \lambda_k, 1), \tag{39}$$

where $V_{\text{Black}}(K, f, \lambda, T)$ is the price of a vanilla option of maturity T and strike K given by the Black-76 formula for an asset of forward f and volatility λ .

The calibration consists then finding the parameters $\alpha_k, \hat{x}_k, \lambda_k$ that minimize the measure M_V . In the calibration, we will also add the martingality constraint, which will ensure that the put-call parity holds exactly in the mixture model. This translates to the additional constraint

$$\sum_{k=0}^n \alpha_k e^{\hat{x}_k} = F(0, T), \tag{40}$$

where $F(0, T)$ is the underlying asset forward to maturity.

We can enforce the constraints by a variable transformation. The $\sqrt{\alpha_k}$ are located on the hypersphere of radius $R = 1$ in \mathbb{R}^{n+1} . Let $\theta \in \mathbb{R}^n$, a point $\mathbf{u} \in \mathbb{R}^{n+1}$ is on the hypersphere if and only if

$$u_k = R \cos(\theta_k) \prod_{j=0}^{k-1} \sin(\theta_j) \tag{41}$$

for $k = 0, \dots, n - 1$ and $u_n = \prod_{j=0}^n \sin(\theta_j)$. We thus use the transform $\alpha_k = u_k^2$. It is invertible and the inverse is

$$\theta_k = \frac{\pi}{2} - \arctan \left(\frac{u_k}{\sqrt{\sum_{j=k+1}^n u_j^2}} \right), \tag{42}$$

or directly in terms of α_k :

$$\theta_k = \frac{\pi}{2} - \arctan \left(\sqrt{\frac{\alpha_k}{\sum_{j=k+1}^n \alpha_j}} \right). \tag{43}$$

The martingality condition can be enforced the same way, using the intermediate variable $z_k = \alpha_k e^{\hat{x}_k}$. Indeed, Eq. (40) means that z_k is located on the hypersphere of radius $R = \sqrt{F(0, T)}$ in \mathbb{R}^{n+1} .

This allows the use of an unconstrained algorithm such as Levenberg–Marquardt to minimize the error measure M_V on \mathbb{R}^{3n+1} . As an initial guess, we use $\alpha_k = \frac{1}{n+1}$, $\hat{x}_k = F(0, T)$, $\lambda_k = \sigma_{\text{ATM}} \sqrt{T}$ where σ_{ATM} is the implied volatility of the option whose strike is closest to the forward price.

If negative strikes are allowed, we can replace lognormal distributions by normal distributions, and the Black–Scholes formula by the Bachelier one. Note that if the bandwidth (λ_k) and the mean (\hat{x}_k) are fixed in advance, the problem becomes a quadratic programming problem (Table 10).

Table 10 Calibrated mixture of lognormal distributions against 1m SPX500 options as of February 05, 2018, for different number of distributions

Mixture of 2 lognormals						
α	0.1664	0.8336				
$e^{\hat{x}}$	2208.46	2713.91				
λ	0.1561	0.04033				
Mixture of 4 lognormals						
α	0.09789	0.2739	0.3489	0.2793		
$e^{\hat{x}}$	2203.35	2523.00	2701.93	2793.91		
λ	0.2467	0.06519	0.02442	0.006543		
Mixture of 6 lognormals						
α	0.02164	0.1121	0.2480	0.2593	0.1794	0.1795
$e^{\hat{x}}$	2045.69	2242.37	2551.67	2696.07	2812.76	2771.53
λ	0.5711	0.1129	0.04712	0.01909	1.576e-05	0.008333

C An algorithm to build a good convex set from the market call option prices

Let $(y_i)_{i=0,\dots,m}$ be the market option strikes and $(c_i)_{i=0,\dots,m}$ the corresponding call option prices. Boyd and Vandenberghe (2004, p. 338) show that the least-squares convex piecewise-linear function $f(y) = \max_{i=0,\dots,m} \hat{c}_i - g_i(y - y_i)$ fitting the (y_i, c_i) is the solution of a quadratic programming problem:

$$\begin{aligned} &\text{minimize } \sum_{i=0}^m (c_i - \hat{c}_i)^2 \\ &\text{subject to } \hat{c}_i + g_i(y_j - y_i) \leq \hat{c}_j, \quad i, j = 0, \dots, m, \\ &\quad \text{and } -1 + \epsilon < g_i < -\epsilon, \quad i = 0, \dots, m. \end{aligned}$$

where \hat{c}_i, g_i are $2(m + 1)$ free parameters. The constraints on g_i make sure that our piecewise-linear function respects the call prices no-arbitrage bounds. Most of those constraints are redundant with the convexity constraints: they can be reduced to $-1 + \epsilon < g_0$ and $g_m < -\epsilon$. A robust quadratic solver such as CVXOPT can be used, and we have then a good initial convex estimate of the call prices by evaluating this piecewise-linear function at each market strike. The estimate is thus given by $(y_i, \hat{c}_i)_{i=0,\dots,m}$.

For 75 option quotes, the constraints consist in a sparse matrix of 5775 rows and 150 columns. The python implementation of CVXOPT takes 0.38 s to solve the quadratic programming problem on an Intel Core i7 7700U processor.

D Implied volatility quotes for vanilla options on SPX500 expiring on March 7, 2018, as of February 5, 2018

The implied volatilities were obtained by using the mid-price of call and put options. The forward price is implied from the put-call parity relationship (Table 11).

E Example code

The following Matlab/Octave code will calibrate the collocated polynomial to market implied volatilities, without any filtering of the market quotes. The code is meant only as an illustration of the method and is not particularly robust or fast.

Listing 1 makeIsotonicCollocation.m - Matlab/Octave code to compute the optimal collocation polynomial for a given set of option strikes and vols.

```
function C = makeIsotonicCollocation(strikes, vols, tte, forward, degree)
    m = length(strikes);
    [prices, vegas] = blackCall(strikes, forward, vols, tte);
    dx = diff(strikes,1); dz = diff(prices,1)./dx;
    s = (dx(1:m-2) .* dz(2:m-1) + dx(2:m-1) .* dz(1:m-2)) ./ (dx(1:m-2)+dx(2:m-1));
    x = -stdnormal_inv(-s);
    binfilter = !isnan(x); x = x(binfilter); y = strikes(2:m-1)(binfilter);
    cubic = polyfit(x,y,3); cubic(4)=forward-cubic(2);
    if cubic(2)^2 - 3*cubic(3)*cubic(1) >= 0
        M = [1, 0, 0; 0, sum(x.^2), sum(x.^4); 0, sum(x.^4), sum(x.^6)];
        R = [forward; sum(x.*y); sum(x.^3 .* y)];
        C = M \ R;
        cubic = [abs(C(3)), 0, abs(C(2)), forward];
    end
    C1 = [sqrt(3*cubic(1)), cubic(2)/sqrt(3*cubic(1))];
    C2 = [sqrt(cubic(3)-cubic(2)^2/(3*cubic(1)))];
    C1p = zeros(1,(degree+1)/2); C2p = zeros(1, (degree-1)/2);
    C1p(length(C1p)-length(C1)+1:length(C1p)) = C1;
    C2p(length(C2p)-length(C2)+1:length(C2p)) = C2;
    guess0 = [C1p C2p];
    guess1 = lsqnonlin(@(guess) isopolyObjective(guess,x,y,forward,degree), guess0);
    weights = min(1.0./vegas,1e6/forward);
    sol = lsqnonlin(@(guess) isocolloObjective(guess1,strikes,prices,weights,forward,degree), guess1);
    [C1,C2] = guessToIso(sol,degree);
    C = isoToPoly(C1, C2, forward);
end

function C = isoToPoly(C1, C2, forward)
    C1sq = conv(C1,C1); C2sq = conv(C2,C2);
    sq = zeros(1,length(C1sq));
    sq(length(C1sq)-length(C2sq)+1:length(C1sq))=C2sq;
    C = polyint(sq+C1sq);
    polyForward = polyHermiteIntegral(C);
    C(length(C)) = forward - polyForward;
end

function [C1,C2] = guessToIso(guess,degree)
    C1=guess'(1:(degree+1)/2); C2=guess'((degree+1)/2+1:length(guess));
end

function value = isopolyObjective(guess,x,y,forward,degree)
    [C1,C2] = guessToIso(guess,degree);
    C = isoToPoly(C1,C2,forward);
    value = polyval(C,x)-y;
end

function value = isocolloObjective(guess,strikes,prices,weights,forward,degree)
    [C1,C2] = guessToIso(guess,degree);
    C = isoToPoly(C1,C2,forward);
    value = (priceEuropeanCall(C, strikes, forward)-prices) .* weights ;
end

function [price, vega] = blackCall(strike, forward, vol, tte)
    vsqrtt = vol*sqrt(tte);
    d1 = log(forward ./ strike) ./ vsqrtt + 0.5*vsqrtt; d2 = d1 - vsqrtt;
    price = forward * stdnormal_cdf(d1) - strike .* stdnormal_cdf(d2);
    vega = forward*stdnormal_pdf(d1)*sqrt(tte);
end
```

Table 11 Implied volatility quotes for SPX500 options expiring on March 7, 2018, as of February 5, 2018. In ACT/365, the option maturity is $T = 0.082192$

Strike	Log-moneyness	Implied vol.	Strike	Log-moneyness	Implied vol.
1900	-0.325055	0.684883	2650	0.007651	0.280853
1950	-0.299079	0.6548	2655	0.009536	0.277035
2000	-0.273762	0.627972	2660	0.011417	0.273715
2050	-0.249069	0.604067	2665	0.013295	0.270891
2100	-0.224971	0.576923	2670	0.01517	0.267889
2150	-0.201441	0.551253	2675	0.017041	0.264533
2200	-0.178451	0.526025	2680	0.018908	0.262344
2250	-0.155979	0.500435	2685	0.020772	0.258598
2300	-0.134	0.474137	2690	0.022632	0.2555
2325	-0.123189	0.461716	2695	0.024489	0.25219
2350	-0.112493	0.445709	2700	0.026343	0.249534
2375	-0.101911	0.433661	2705	0.028193	0.246659
2400	-0.09144	0.42016	2710	0.03004	0.243553
2425	-0.081077	0.407463	2715	0.031883	0.240202
2450	-0.070821	0.393168	2720	0.033723	0.236588
2470	-0.062691	0.381405	2725	0.03556	0.234574
2475	-0.060668	0.3793	2730	0.037393	0.230407
2480	-0.05865	0.377109	2735	0.039223	0.227866
2490	-0.054626	0.372471	2740	0.041049	0.223049
2510	-0.046626	0.360294	2745	0.042872	0.219888
2520	-0.04265	0.354671	2750	0.044692	0.218498
2530	-0.03869	0.350533	2755	0.046509	0.214702
2540	-0.034745	0.34419	2760	0.048322	0.210506
2550	-0.030815	0.339273	2765	0.050132	0.208175
2560	-0.026902	0.333069	2770	0.051939	0.205508
2570	-0.023003	0.328206	2775	0.053742	0.199967
2575	-0.021059	0.324314	2780	0.055542	0.199007
2580	-0.019119	0.322041	2785	0.057339	0.195062
2590	-0.015251	0.3168	2790	0.059133	0.190547
2600	-0.011397	0.310914	2795	0.060923	0.188427
2610	-0.007559	0.305042	2800	0.062711	0.185893
2615	-0.005645	0.302416	2805	0.064495	0.182878
2620	-0.003734	0.299488	2810	0.066276	0.179292
2625	-0.001828	0.29609	2815	0.068053	0.175001
2630	7.5E-05	0.292378	2835	0.075133	0.185751
2635	0.001974	0.289516	2860	0.083913	0.207173
2640	0.00387	0.28584	2900	0.097802	0.225248
2645	0.005762	0.283342			

The implied forward price is $F(0, T) = 2629.80$. The interest rate is $r = 0.97\%$

Listing 2 priceEuropeanCall.m - Matlab/Octave code to price a European call option using a collocation polynomial for a given set of option strikes.

```
function value = priceEuropeanCall(C, strike, forward)
    ck = solveStrike(C, strike);
    value = polyHermiteIntegralBounded(C, ck) - stdnormal_cdf(-ck).*strike;
end

function ck = solveStrike(C, strike)
    ck = zeros(1, length(strike));
    for i=1: length(strike)
        Cp = C; Cp(length(Cp)) -= strike(i); Cp /= C(1);
        r = roots(Cp); r = r(imag(r)==0);
        ck(i) = r;
    end
end

function value = polyHermiteIntegral(C)
    n = length(C); value = C(n); m0=1; m1 = 0;
    for i=2:(n-1)
        m2 = m0 * (i-1); value += C(n-i)*m2; m0=m1; m1=m2;
    end
end

function value = polyHermiteIntegralBounded(C, x0)
    n = length(C); m0 = stdnormal_cdf(-x0); nx0 = stdnormal_pdf(x0); m1 = nx0;
    value = C(n)*m0 + C(n-1)*m1;
    for i=2:(n-1)
        m2 = m0 * (i-1) + nx0 .* (x0 .^ (i-1));
        value += C(n-i)*m2; m0=m1; m1=m2;
    end
end
```

References

- Alefeld, G., Potra, F.A., Shi, Y.: Algorithm 748: enclosing zeros of continuous functions. *ACM Trans. Math. Softw. (TOMS)* **21**(3), 327–344 (1995)
- Andreasen, J., Høge, B.: Volatility interpolation. *Risk* **24**(3), 76 (2011)
- Arneric, J., Aljinović, Z., Poklepović, T.: Extraction of market expectations from risk-neutral density. *J. Econ. Bus.* **33**, 235–256 (2015)
- Bahra, B.: Implied risk-neutral probability density functions from option prices: a central bank perspective. In: Satchell, S., Knight, J. (eds.) *Forecasting Volatility in the Financial Markets*, 3rd edn, pp. 201–226. Elsevier, Amsterdam (2007)
- Bates, D.S.: Jumps and stochastic volatility: exchange rate processes implicit in Deutsche mark options. *Rev. Financ. Stud.* **9**(1), 69–107 (1996)
- Bliss, R.R., Panigirtzoglou, N.: Option-implied risk aversion estimates. *J. Finance* **59**(1), 407–446 (2004)
- Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, Cambridge (2004)
- Carr, P., Lee, R.: Robust replication of volatility derivatives. Preprint (2008)
- Carr, P., Stanley, M., Madan, D.: Towards a theory of volatility trading. In: Reprinted in *Option Pricing, Interest Rates, and Risk Management*, Musiella, Jouini, Cvitanic, pp. 417–427. University Press (1998)
- Cheng, M.K.C.: A new framework to estimate the risk-neutral probability density functions embedded in options prices (10-181) (2010)
- Christoffersen, P., Heston, S., Jacobs, K.: The shape and term structure of the index option smirk: why multifactor stochastic volatility models work so well. *Manag. Sci.* **55**(12), 1914–1932 (2009)
- Dougherty, R.L., Edelman, A.S., Hyman, J.M.: Nonnegativity-, monotonicity-, or convexity-preserving cubic and quintic hermite interpolation. *Math. Comput.* **52**(186), 471–494 (1989)
- Dupire, B.: Pricing with a smile. *Risk* **7**(1), 18–20 (1994)
- Flint, E., Maré, E.: Estimating option-implied distributions in illiquid markets and implementing the ross recovery theorem. *S. Afr. Actuar. J.* **17**(1), 1–28 (2017)
- Gander, W.: On Halley's iteration method. *Am. Math. Month.* **92**, 131–134 (1985)
- Gatheral, J., Lynch, M.: Lecture 1: Stochastic volatility and local volatility. *Case Studies in Financial Modeling Notes*, Courant Institute of Mathematical Sciences (2004)
- Good, I.: The colleague matrix, a Chebyshev analogue of the companion matrix. *Q. J. Math.* **12**(1), 61–68 (1961)
- Grzelak, L.A.: The CLV framework—a fresh look at efficient pricing with smile. [Browser Download This Paper](#) (2016)

- Grzelak, L.A., Oosterlee, C.W.: From arbitrage to arbitrage-free implied volatilities. *J. Comput. Finance* **20**(3), 31–49 (2017)
- Hagan, P.S.: Convexity conundrums: pricing CMS swaps, caps, and floors. *Wilmott* **2003**(2), 38–45 (2003). <https://doi.org/10.1002/wilm.42820030211>
- Hagan, P.S., Kumar, D., Lesniewski, A.S., Woodward, D.E.: Managing smile risk. *Wilmott magazine* (2002)
- Hagan, P.S., Kumar, D., Lesniewski, A., Woodward, D.: Arbitrage-free SABR. *Wilmott* **2014**(69), 60–75 (2014)
- Heston, S.L.: A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Financ. Stud.* **6**(2), 327–343 (1993)
- Hu, Z., Kerkhof, J., McCloud, P., Wackertapp, J.: Cutting edges using domain integration. *Risk* **19**(11), 95 (2006)
- Hunt, P., Kennedy, J.: *Financial Derivatives in Theory and Practice*. Wiley, New York (2004)
- Huynh, H.T.: Accurate monotone cubic interpolation. *SIAM J. Numer. Anal.* **30**(1), 57–100 (1993)
- Hyman, J.M.: Accurate monotonicity preserving cubic interpolation. *SIAM J. Sci. Stat. Comput.* **4**(4), 645–654 (1983)
- Jäckel, P.: Let's be rational. *Wilmott* **2015**, 40–53 (2015). <https://doi.org/10.1002/wilm.10395>
- Jenkins, M.A.: Algorithm 493: zeros of a real polynomial [c2]. *ACM Trans. Math. Softw. (TOMS)* **1**(2), 178–189 (1975)
- Kahalé, N.: An arbitrage-free interpolation of volatilities. *Risk* **17**(5), 102–106 (2004)
- Le Floc'h, F., Kennedy, G.: Finite difference techniques for arbitrage-free SABR. *J. Comput. Finance* **20**(3), 51–79 (2017)
- Li, M., Lee, K.: An adaptive successive over-relaxation method for computing the Black–Scholes implied volatility. *Quant. Finance* **11**(8), 1245–1269 (2011)
- Lorentz, G., Zeller, K.: Degree of approximation by monotone polynomials I. *J. Approx. Theory* **1**(4), 501–504 (1968)
- Malz, A.M.: A simple and reliable way to compute option-based risk-neutral distributions. FRB of New York Staff Report No. 677 (2014)
- Mathelin, L., Hussaini, M.Y.: A stochastic collocation algorithm for uncertainty analysis. NASA Contractor Report CR-2003-212153 (2004)
- Mercurio, F., Pallavicini, A.: Swaption skews and convexity adjustments. Banca IMI (2005)
- Mercurio, F., Pallavicini, A.: Smiling at convexity: bridging swaption skews and cms adjustments. *Risk* August, pp. 64–69 (2006)
- Moler, C.: Cleve's corner: roots-of polynomials, that is. *MathWorks Newsl.* **5**(1), 8–9 (1991)
- Murray, K., Müller, S., Turlach, B.: Fast and flexible methods for monotone polynomial fitting. *J. Stat. Comput. Simul.* **86**(15), 2946–2966 (2016)
- Nonweiler, T.R.: Algorithms: Algorithm 326: roots of low-order polynomial equations. *Commun. ACM* **11**(4), 269–270 (1968)
- Passow, E., Roulier, J.A.: Monotone and convex spline interpolation. *SIAM J. Numer. Anal.* **14**(5), 904–909 (1977)
- Reznick, B.: Some concrete aspects of Hilbert's 17th problem. *Contemp. Math.* **253**, 251–272 (2000)
- Schumaker, L.I.: On shape preserving quadratic spline interpolation. *SIAM J. Numer. Anal.* **20**(4), 854–864 (1983)
- Syrdal, S.A.: A study of implied risk-neutral density functions in the Norwegian option market. Norges Bank working paper ANO 2002/13 (2002)
- Trefethen, L.N.: Six myths of polynomial interpolation and quadrature. Technical report, Mathematics Today (2011)
- Wolibner, W.: Sur un polynôme d'interpolation. *Colloquium Mathematicae* **2**(2), 136–137 (1951). <http://eudml.org/doc/209940>
- Wystup, U.: *FX Options and Structured Products*. Wiley, New York (2015)
- Zeliade Systems: Quasi-explicit calibration of Gatheral's SVI model. Technical report, Zeliade White Paper, Zeliade Systems (2009)