Convex Volatility Interpolation*

Fabrice Deschâtres fabrice.deschatres@volptima.com

February 27, 2026

Over the last decade or so, convex optimization software has made tremendous progress, particularly in the open-source ecosystem. For example, in 2014, Stanford University's release of CVXPY, one of the first domain-specific modeling languages, marked a milestone in convex optimization, enabling non-specialists to define complex problems easily in a human-readable way. Another significant breakthrough came in 2022, when the University of Oxford introduced Clarabel, an interior-point numerical solver that uses a novel homogeneous embedding. As reported by Goulart and Chen (2024), this is the fastest and most robust solver to date for problems with a quadratic objective.

Option markets have also evolved significantly over the last few decades. The days when *Stochastic Volatility Inspired* (SVI) (Gatheral 2004), with its five parameters, sufficed to calibrate options on the most liquid assets are long gone.

Despite the increasingly detailed volatility smiles observed in the market, fitting volatility surfaces has become highly efficient thanks to today's advanced convex optimization solvers. Moreover, the large number of listed options makes volatility fitting a heavily constrained problem, favoring purely optimization-based, model-free approaches.

Analysis of the Problem

Successfully framing volatility fitting as a convex optimization problem depends on identifying an appropriate parameterization in the correct mathematical space. To guarantee the absence of static arbitrage, the volatility surface must be free of both calendar spread and butterfly arbitrage. Both conditions are linear in price space, but the latter becomes nonlinear in variance space. Fengler (2009) proposes constrained natural cubic spline smoothing of call prices, motivated primarily by local volatility calibration. Lucic (2019) formulates the fitting problem with B-splines and bid-ask bounds, solved via quadratic programming. Le Floc'h and Oosterlee (2019) take a different route, using stochastic collocation on monotonic splines to produce arbitrage-free interpolations in probability space. However, price-space approaches suffer from wing instability: out-of-the-money option prices approach zero in the far wings and small absolute errors translate into large implied volatility discrepancies. Furthermore, defining intuitive parameters outside of variance space is challenging. This paper adopts a variance-based approach, drawing on Fengler's second-derivative-penalized spline smoothing and the bid-ask-aware objective of Lucic, while accepting the caveat that no-butterfly-arbitrage conditions are nonlinear in variance space.

*Author version of the article published in *Cutting Edge*, *Risk*, February 2026.

A least squares fit to market quotes yields a quadratic objective. To keep the problem quadratic, the fitted function $v(K,T)$ should be expressed as a linear combination of the parameters:

$$v(K,T) = \sum_i p_{i,T} f_{i,T}(K)$$

where $p_{i,T}$ are the parameters for the time to expiry T, and $f_{i,T}$ are their associated functions.

The parameterization is therefore restricted to a space of basis functions. However, a potential drawback is that the parameters may lack interpretability; therefore, we need to map these parameters to another parameter space with a more intuitive meaning.

After presenting the dual parameterization in cubic spline and B-spline spaces, we shall detail the different building blocks of the optimization, namely the objective function and the constraints. Notably, this paper derives and linearizes the no-butterfly-arbitrage constraints within the chosen cubic spline parameter space.

Notation We introduce here the key notation used throughout this paper. We denote by $k \coloneqq \log(K/F)$ the log-forward moneyness, where F is the forward price. We define the normalized log-moneyness as $z \coloneqq \frac{k}{\sigma_\star \sqrt{T}}$, where $\sigma_\star$ is the anchor at-the-money (ATM) volatility, an estimate of the ATM volatility determined *prior* to the volatility surface fit. In contrast, the calibration fits the variance $v(z,T) \coloneqq \sigma^2(z,T)$. The product vT is referred to as the total variance.

We define the shape function as $z \mapsto \frac{v(z)}{v_\star}$, which represents the variance normalized by the anchor ATM variance $v_\star \coloneqq \sigma_\star^2$. We denote $s \coloneqq \frac{1}{v_\star}\frac{\partial v}{\partial z}$ and $c \coloneqq \frac{1}{v_\star}\frac{\partial^2 v}{\partial z^2}$ as its first and second derivatives. s and c are, respectively, dimensionless measures of skew and convexity. While v, s, and c are functions of z and T, and $v_\star$ and F are functions of T, we often omit these dependencies in the notation for simplicity.

There are m expiries labeled $T_1, \ldots, T_m$.

The CVI Parameterization

While various basis functions could be employed, cubic B-splines are a natural choice since they form the basis of cubic splines, which are widely used for volatility interpolation.

Let us discretize the z space and introduce $z_0, \ldots, z_{n-1}$, the n anchored normalized log-moneyness points (knots of the spline). Furthermore, we assume that zero is one of these knots, meaning there exists an index l such that $z_l = 0$ (ATM knot). The range of knots should be sufficiently wide to include all the listed strikes targeted for fitting. While evenly spaced knots provide a simple initial

approach, increasing knot density near-the-money better captures variations in variance convexity.

CVI Cubic Spline The cubic spline has the following $n+2$ parameters:

- $v(z=0)$: the ATM variance
- $\left.\frac{\partial v}{\partial z}\right|_{z=0}$: the ATM skew
- $\left\{\left.\frac{\partial^2 v}{\partial z^2}\right|_{z_i}\right\}_{0\le i\le n-1}$: the n anchored convexities

In a nutshell, the variance v is represented as a function of z, specified by its value and first derivative at the origin ($z=0$), and by its second derivative, which is a piecewise linear function over a fixed set of knots. This is illustrated in Figure 1.

The cubic spline consists of $n+1$ third-order polynomial segments, which have $4(n+1)$ degrees of freedom. These are found by solving a $4(n+1)\times 4(n+1)$ linear system using the following $4(n+1)$ relations:

- The second derivative of the cubic spline should match the convexities $\left\{\left.\frac{\partial^2 v}{\partial z^2}\right|_{z_i}\right\}_{0\le i\le n-1}$ for each z_i ($\to 2n$ relations).
- At $z=0$, its value should match $v(z=0)$ ($\to 2$ relations).
- At $z=0$, its first derivative should match $\left.\frac{\partial v}{\partial z}\right|_{z=0}$ ($\to 2$ relations).
- For $z\neq 0$, the value of the cubic spline and its first derivative should be continuous ($\to 2(n-1)$ relations).
- The cubic coefficient is zero for z_0 and z_{n-1} ($\to 2$ relations).

Note that, additionally, $\left.\frac{\partial^2 v}{\partial z^2}\right|_{z_0}$ and $\left.\frac{\partial^2 v}{\partial z^2}\right|_{z_{n-1}}$ are set to zero to ensure that the variance is linearly extrapolated in log-strike in the tails.

CVI B-Spline The variance $z\mapsto v(z)$ represented by a cubic spline can also be formulated as the weighted sum of the $n+2$ B-splines associated with the n knots. Outside of $[z_0, z_{n-1}]$, both representations are linearly extrapolated. These two sets of parameters (the $n+2$ cubic spline inputs and the $n+2$ B-spline weights) are linked by a known linear transformation and can thus be used interchangeably in the convex optimization.

Objective Function

As the m expiries are fitted simultaneously, the optimization solves for $m(n+2)$ variables. It could also be expressed with two fewer variables per expiry since, as noted earlier, $\left.\frac{\partial^2 v}{\partial z^2}\right|_{z_0}$ and $\left.\frac{\partial^2 v}{\partial z^2}\right|_{z_{n-1}}$ are set to zero. Thus, there are n free parameters per expiry.

The objective function is the sum of the four penalties detailed below. Each term requires careful consideration of its weight relative to the others to ensure consistent behavior regardless of the number of expiries, options per expiry, and the level of volatility. Each penalty is the sum of dimensionless numbers across expiries. We do not differentiate between puts and calls; bids and asks refer to the tightest quotes across the two.

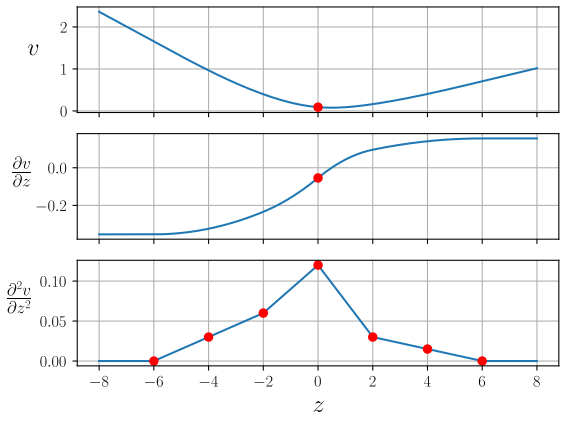


Figure 1: The CVI cubic spline (top), its first derivative (middle), and its second derivative (bottom). The parameters of the cubic spline are displayed as red circles.

Least Squares Penalty We define the least squares penalty as:

$$\sum_{j=1}^{m}\frac{1}{N_{T_j,\text{mid}}}\sum_{i\in\mathcal{S}_{T_j,\text{mid}}}\left(v(K_i,T_j)-v_{\text{mid}}(K_i,T_j)\right)^2 w_{i,j}$$

where:

- v_{mid} is the variance associated with the mid price $\frac{P_{\text{ask}}+P_{\text{bid}}}{2}$.
- $w_{i,j}$ is the weight associated with the option of strike K_i and expiry T_j.
- $N_{T,\text{mid}}$ is the number of options in $\mathcal{S}_{T,\text{mid}} \coloneqq \mathcal{S}_{T,\text{ask}}\cap\mathcal{S}_{T,\text{bid}}$, the set of options expiring at T with both a bid and an ask.

In order to make the penalty exactly equivalent to a chi-squared statistic, the weights $w_{i,j}$ are inversely proportional to the squared difference of ask and bid variances:

$$w_{i,j}=\frac{1}{\left(v_{\text{ask}}(K_i,T_j)-v_{\text{bid}}(K_i,T_j)\right)^2}$$

With that penalty, very wide quotes have a negligible impact on the fit, as can be seen in Figure 2.

Above Ask Penalty The above-ask penalty is applied to options whose fitted variance exceeds the quoted ask variance. Unlike the Least Squares penalty, it also applies to options for which only an ask quote is available. The weights are proportional to the vega derived from the ask price. The penalty can be expressed as:

$$\sum_{j=1}^{m}\frac{1}{N_{T_j,\text{ask}}}\sum_{i\in\mathcal{S}_{T_j,\text{ask}}} w_{i,j,\text{ask}}\max(v(K_i,T_j)-v_{\text{ask}}(K_i,T_j),0)^2$$

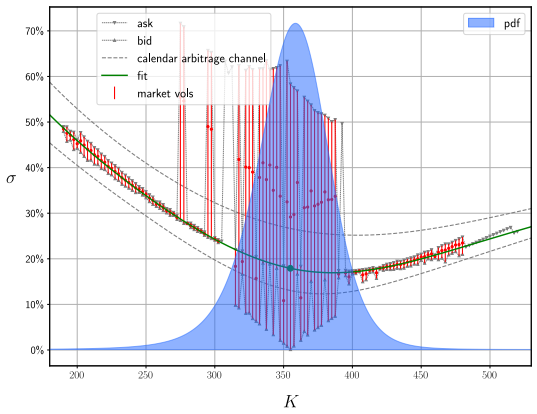


Figure 2: Volatility fit and probability density function (KOSPI 200).

where the weight $w_{i,j,\text{ask}}$ is defined as:

$$w_{i,j,\text{ask}} = q_j \frac{\text{Vega}_{\text{ask}}(K_i, T_j)}{\sum_{l \in \mathcal{S}_{T_j,\text{ask}}} \text{Vega}_{\text{ask}}(K_l, T_j)}$$

The factor q_j is given by:

$$q_j = \sum_{i \in \mathcal{S}_{T_j,\text{mid}}} \frac{1}{\Big(v_{\text{ask}}(K_i, T_j) - v_{\text{bid}}(K_i, T_j)\Big)^2}$$

Here, q_j plays the role of a normalization constant, ensuring that the penalty for expiry T_j scales in a manner consistent with a chi-squared statistic—analogous to the Least Squares penalty.

Below Bid Penalty The below-bid penalty, which mirrors the above-ask penalty, is applied to options whose variance is below the bid variance.

The Above Ask and Below Bid terms are important because the Least Squares penalty does not account for all the market quotes. Removing these two terms from the objective function leads to some loss of information, particularly in the tails where very OTM options are typically offer-only. However, they require additional auxiliary variables in the optimization canonicalization process, which increases the problem size. This impacts the solver performance and can increase the calibration times severalfold. Applying these penalties for strikes that have a bid or an ask, but not both, is a good compromise, as the objective function still uses all the market quotes while minimizing the impact on speed. The calibration times reported in this paper are based on this choice.

Options whose normalized log-moneyness falls outside the range $[z_0, z_{n-1}]$ should be excluded from the Least Squares and Below Bid penalties.

Strike Regularization Penalty This penalty encourages a smooth smile at each expiry and prevents overfitting. For expiry T, let:

$$\mathbf{c}_T := \big(c(z_0, T),\, c(z_1, T),\, \ldots,\, c(z_{n-1}, T)\big) \in \mathbb{R}^n,$$

be the vector of normalized convexities sampled at the knots. We penalize the ℓ_1 norm of its first-order differences (a total-variation penalty):

$$\lambda \sum_{j=1}^{m} \sum_{i=0}^{n-2} \big|\, \mathbf{c}_{T_j}[i+1] - \mathbf{c}_{T_j}[i] \,\big|$$

where λ is the strike regularization factor. A higher λ yields a smoother cubic spline and less overfitting. For example, $\lambda = 0.05$ is a sensible choice across underlyings. Since the other penalties scale as a chi-squared statistic, a practical interpretation is that we are willing to increase the chi-squared by about 0.05 to reduce the total variation by 1, and vice versa.

The c curve is not necessarily bell-shaped and can exhibit well-defined features, such as peaks and troughs, reflecting significant changes in the curvature of the volatility smile. This is illustrated in Figure 5b.

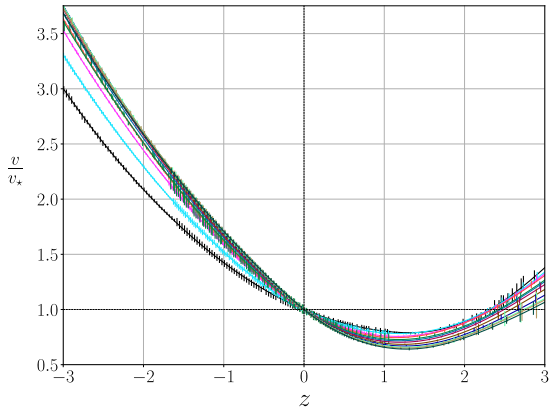


Figure 3: Shape function $\frac{v(z)}{v_\star}$ across expiries (SPX).

Non-Arbitrage-Related Constraints

After defining the objective function, we need to review the constraints that are part of the convex optimization. We first review the constraints that are not directly related to the absence of arbitrage.

Linear Extrapolation As previously stated, the variance is linearly extrapolated in log-moneyness beyond the last cubic spline knots. Therefore, convexity is set to zero at the edge knots z_0 and z_{n-1}: $\frac{\partial^2 v}{\partial z^2}(z_0) = 0$ and $\frac{\partial^2 v}{\partial z^2}(z_{n-1}) = 0$.

Positivity of the Variance The variance, v, must be positive for each point z. To ensure this, we add the linear constraint $v(z) \geq 0$ on a set of points. Applying this constraint only to the first expiry is sufficient, owing to the bounds set by no-calendar-spread-arbitrage constraints (see the section below) for subsequent expiries. Moreover, it only needs to be enforced between z_0 and z_{n-1}, as the next constraint addresses the tails. An alternative is to enforce the positivity of the B-spline coefficients directly.

Positivity of the Variance in the Tails Since the variance cannot be negative and is linearly extrapolated beyond the edge knots, it must be non-increasing in the left tail and non-decreasing in the right tail: $\frac{\partial v}{\partial z}(z_0) \leq 0$ and $\frac{\partial v}{\partial z}(z_{n-1}) \geq 0$.

No-Calendar-Spread-Arbitrage Constraints

The absence of calendar spread arbitrage is straightforward to implement, as it is linear in variance space. The total variance, defined for a fixed strike-to-forward ratio, must increase monotonically with the time to expiry T. Graphically, this condition implies that the total variance curves for various maturities, plotted against the strike-to-forward ratio K/F_T, must not intersect. This is illustrated in Figure 4. For a given expiry, this condition implies there are upper and lower bounds on the implied variance at each strike, derived from the total variances of the adjacent expiries. These bounds form a calendar arbitrage channel within which the volatility fit must lie, as shown in Figures 2 and 5a.

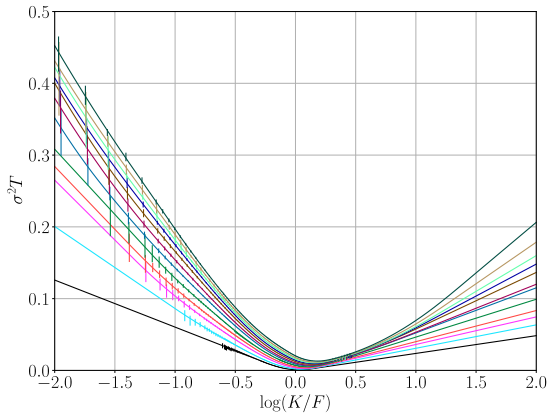


Figure 4: Total variance curves (SPX).

Generic No-Calendar-Spread-Arbitrage Constraints Assuming the condition is applied to r strikes, we end up with $(m-1)r$ linear constraints. For $1 \leq j < m$, $0 \leq i < r$, these are expressed as:

$$v(K_i, T_j)T_j \leq v\left(K_i \frac{F_{T_{j+1}}}{F_{T_j}}, T_{j+1}\right) T_{j+1} \tag{1}$$

No-Calendar-Spread-Arbitrage Constraints in the Tails The relation (1) is applied to a discrete set of strikes, which, for instance, could be linearly spaced in z space between the edge knots z_0 and z_{n-1}. It does not need to be applied beyond the edge knots, as no-calendar-spread-arbitrage constraints can be implemented more efficiently in the tails given the linear extrapolation. After differentiating (1) with respect to $\log(K)$, we can enforce no-calendar-spread-arbitrage constraints in the tails by applying them at z_0 and z_{n-1}:

$$\begin{cases} s(z_0, T_j)\sqrt{v_\star(T_j)T_j} & \geq s(z_0, T_{j+1})\sqrt{v_\star(T_{j+1})T_{j+1}} \\ s(z_{n-1}, T_j)\sqrt{v_\star(T_j)T_j} & \leq s(z_{n-1}, T_{j+1})\sqrt{v_\star(T_{j+1})T_{j+1}} \end{cases}$$

Equivalently, the slope of the total variance $T\frac{dv}{dk}$ is a non-increasing function of T at z_0 and a non-decreasing function of T at z_{n-1}.

No-Strike-Arbitrage Constraints

The absence of strike arbitrage is guaranteed by:

- the absence of large- (or small-) strike arbitrage
- the absence of butterfly arbitrage

Lee's Tail Slope Bounds We control the large-/small-strike (tail) behavior via the method of Lee (2004). In our setting, variance is linearly extrapolated in log-strike, so in the right tail we impose:

$$T\frac{\partial v}{\partial \log K} < 2$$

We use a strict inequality to exclude the boundary case, where the tail slope equals 2.

Expressed in terms of s, this becomes:

$$s < \frac{2}{\sqrt{v_\star T}}$$

By symmetry, for small strikes (i.e., in the left tail), the condition becomes:

$$s > -\frac{2}{\sqrt{v_\star T}}$$

These constraints are applied respectively at z_{n-1} and z_0. Because the constraints are strict, we enforce them numerically with a small margin.

Large-/Small-Strike No-Arbitrage Limits The following tail limits must hold:

$$\begin{cases} \lim_{k \to +\infty} d_1(k) = -\infty & \text{(large-strike no-arbitrage)} \\ \lim_{k \to -\infty} d_2(k) = +\infty & \text{(no mass at zero)} \end{cases}$$

where $d_{1,2} = \frac{-k \pm \frac{vT}{2}}{\sqrt{vT}}$.

Under linear wings and the strict Lee bounds in each wing, these limits follow immediately.

No-Butterfly-Arbitrage Constraints We begin with the no-arbitrage formulation from Martini and Mingone (2022):

$$\sigma(k)d_1'(k)d_2'(k) + \sigma''(k) \geq 0$$

Performing a change of variables from log-forward moneyness k to normalized log-moneyness z, we rewrite the above condition explicitly in terms of the cubic spline parameters:

$$c \geq \frac{v_\star}{2v}s^2 + \sqrt{v_\star T}s - 2\left(1 + d_1\sqrt{\frac{v_\star}{v}}s + d_1 d_2 \frac{v_\star}{4v}s^2\right) \tag{2}$$

PDF $\geq$ 0 between z_0 and z_{n-1} The no-butterfly-arbitrage condition (2) is inherently nonlinear and non-convex in variance space, as it is linear in price space. Therefore, as we cannot enforce it directly, the workaround is to linearize it using the solution found at the previous iteration, denoted "ref". In practice, a small number of iterations is enough.

The linearized form is:

$$c \geq \beta_0 + \beta_1(s - s_{\text{ref}}) + \beta_2(v - v_{\text{ref}})$$

with:

$$\begin{cases} \beta_0 = & \frac{v_\star}{2v_{\text{ref}}} s_{\text{ref}}^2 + \sqrt{v_\star T}\, s_{\text{ref}} \\ & -2\left(1 + d_{1,\text{ref}} \sqrt{\frac{v_\star}{v_{\text{ref}}}}\, s_{\text{ref}} + d_{1,\text{ref}} d_{2,\text{ref}} s_{\text{ref}}^2 \frac{v_\star}{4v_{\text{ref}}}\right) \\ \beta_1 = & \frac{v_\star}{v_{\text{ref}}} s_{\text{ref}} + \sqrt{v_\star T} \\ & -2\left(d_{1,\text{ref}} \sqrt{\frac{v_\star}{v_{\text{ref}}}} + d_{1,\text{ref}} d_{2,\text{ref}}\, s_{\text{ref}} \frac{v_\star}{2v_{\text{ref}}}\right) \\ \beta_2 = & \frac{s_{\text{ref}}}{v_{\text{ref}}} \left(-\frac{v_\star}{2v_{\text{ref}}} s_{\text{ref}} - 2\frac{k}{v_{\text{ref}} T}\left(\sqrt{v_\star T} - \frac{k}{2} s_{\text{ref}} \frac{v_\star}{v_{\text{ref}}}\right)\right) \end{cases}$$

where $d_{1,2,\text{ref}} = \frac{-k \pm \frac{v_{\text{ref}} T}{2}}{\sqrt{v_{\text{ref}} T}}$.

PDF $\geq$ 0 at the edge knots and beyond Starting from (2), the no-butterfly-arbitrage condition simplifies to a quadratic inequality in terms of the skew s at and beyond the edge knots, where $c = 0$:

$$v_\star \left(\frac{d_1 d_2 - 1}{2v}\right) s^2 + \sqrt{v_\star}\left(\frac{2d_1}{\sqrt{v}} - \sqrt{T}\right) s + 2 \geq 0.$$

The discriminant of this quadratic is:

$$\Delta = v_\star \left[\left(\frac{2d_1}{\sqrt{v}} - \sqrt{T}\right)^2 - \frac{4(d_1 d_2 - 1)}{v}\right]$$

This discriminant simplifies neatly to:

$$\Delta = v_\star \left(T + \frac{4}{v}\right)$$

Hence, the discriminant is positive, and the equation admits two roots:

$$s = \frac{2v\sqrt{T}\left(k \pm \sqrt{vT}\sqrt{1 + \frac{vT}{4}}\right)}{\sqrt{v_\star}\left(k^2 - \frac{v^2T^2}{4} - vT\right)}$$

We assume that z_0 and z_{n-1} have been chosen far enough into the tails, so that $d_1 d_2 > 1$ or, equivalently, that $|k| > \sqrt{vT}\sqrt{1 + \frac{1}{4}vT}$. This holds for sufficiently large $|k|$ as we have assumed a strict inequality for Lee's tail slope.

The relation $d_1 d_2 > 1$ results in a positive quadratic coefficient. Consequently, the quadratic polynomial will be negative (indicating the presence of arbitrage) between the two roots and positive (indicating no arbitrage) for strikes below the lower root and for strikes above the higher root. Therefore, to avoid arbitrage, it is crucial to ensure that s does not fall between the two roots.

At z_{n-1}, we enforce the skew s to be lower than the lower root (which is positive), as the lower root—at the limit of arbitrage—already implies a very steep skew:

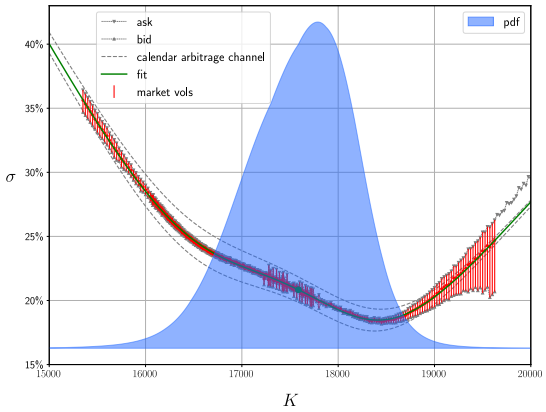
$$s \leq \frac{2v\sqrt{T}\left(k - \sqrt{vT}\sqrt{1 + \frac{vT}{4}}\right)}{\sqrt{v_\star}\left(k^2 - \frac{v^2T^2}{4} - vT\right)} \tag{3}$$

Similarly, for z_0, the two roots are negative and the skew needs to be greater than the higher root:

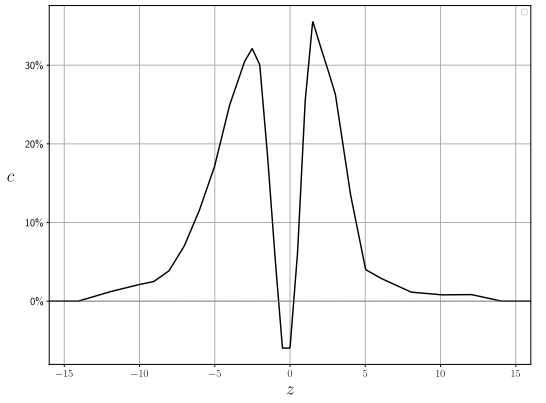
$$s \geq \frac{2v\sqrt{T}\left(k + \sqrt{vT}\sqrt{1 + \frac{vT}{4}}\right)}{\sqrt{v_\star}\left(k^2 - \frac{v^2T^2}{4} - vT\right)} \tag{4}$$

Like (2), the two equations above must be linearized.

We only need to enforce these relations at the edge knots z_0 and z_{n-1}, and not beyond. Indeed, if the edge knots are placed sufficiently far into the tails, the constraints are automatically satisfied for more extreme strikes. Given that we enforce Lee's tail slope bounds, the right-hand side of (3) reaches a positive constant asymptotically from *below* for $k \to \infty$, whereas the right-hand side of (4) reaches a negative constant asymptotically from *above* for $k \to -\infty$.



(a) Volatility fit and probability density function.



(b) c as a function of z. The Strike Regularization penalty is proportional to the total variation of that curve.

Figure 5: Nasdaq-100 one-week expiry. c is negative around the money, which is a common feature of tech stocks before earnings. See Glasserman and Pirjol (2023) for a study of W-shaped smiles.

Putting It All Together[1]

CVI fits all the expiries simultaneously by solving a quadratic programming (QP) problem with linear constraints. We have presented the quadratic objective consisting of four terms and eight types of constraints related to linear extrapolation, positivity of the variance, and absence of static arbitrage. The no-butterfly-arbitrage constraints must be linearized, which implies a two-step calibration process:

- A first optimization without the constraints that need to be linearized.
- One or several subsequent iterations with the no-butterfly-arbitrage constraints linearized using the previous iteration.

One or a few iterations with the no-butterfly-arbitrage constraints are sufficient in practice.

Constraint	Linearized	Discretized
Linear Extrapolation	–	–
Positivity of the Variance	–	✓
Positivity of the Variance in the Tails	–	–
No Calendar Spread	–	✓
No Calendar Spread in the Tails	–	–
Lee's Tail Slope Bounds	–	–
PDF ≥ 0 between z_0 and z_{n-1}	✓	✓
PDF ≥ 0 at the edge knots and beyond	✓	–

Table 1: Overview of CVI Constraints

Some constraints are by nature discrete and need to be applied to a limited set of strikes. There is a trade-off between the level of discretization and computational speed: finer discretization can help eliminate smaller arbitrages but at the cost of a slower calibration.

Canonical QP Problem In its canonical form, a convex quadratic program with n_v variables and n_c constraints can be formulated as follows:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} x^T P x + q^T x \\ \text{subject to} \quad & l \leq Ax \leq u \end{aligned}$$

where $x \in \mathbb{R}^{n_v}$ is the optimization variable. The objective function is defined by the positive semidefinite matrix $P \in \mathbb{S}_+^{n_v}$ and vector $q \in \mathbb{R}^{n_v}$. The linear constraints are defined by the matrix $A \in \mathbb{R}^{n_c \times n_v}$ and vectors l and u. l (respectively, u) is a vector of size n_c whose elements are in $\mathbb{R} \cup \{-\infty\}$ (respectively, $\mathbb{R} \cup \{+\infty\}$).

Numerical Results

We present the calibration times for SPX and SX5E volatility surfaces using Clarabel under the following assumptions:

- Discretized constraints are applied across 20 strikes.

[1]This section was not included in the published version due to space constraints.

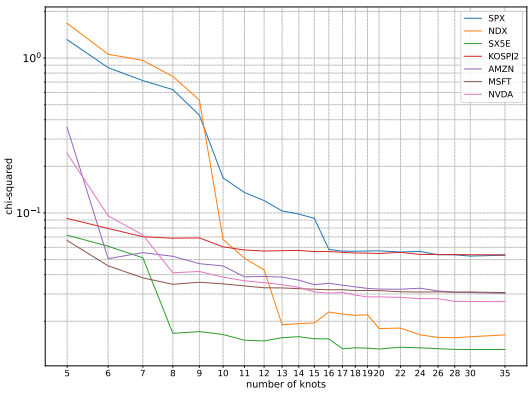


Figure 6: Chi-squared of a fit (average across expiries) as a function of the number of knots per expiry. For single-name equities, the implied volatility surfaces are de-Americanized from American option quotes.

- Two optimization iterations are performed: the first without linearized constraints and the second with all the constraints. The number of constraints shown in Table 2 corresponds to the second iteration, while the time in Table 3 represents the total time taken by the solver across both iterations.

The number of volatility quotes indicated represents the total number of bid and ask quotes used in the calibration, selected by choosing the tightest quotes in volatility terms from both calls and puts.

	$n_{\text{vol quotes}}$	n_{expiries}	$n_{\text{constraints}}$
SX5E	3 153	19	948
SPX	14 538	46	2 298

Table 2: Market data and number of constraints

Solving time is shown for a number of knots per expiry—also the number of free parameters—ranging from 10 to 40, to illustrate how calibration time scales. The number of knots, n, must be sufficiently large—typically up to the mid-to-high teens for highly liquid assets—to achieve a good fit, while 10 parameters can suffice for less liquid assets. As illustrated in Figure 6, the fit converges when the chi-squared statistic plateaus, indicating that additional knots no longer materially improve the calibration quality. Having too few knots acts similarly to applying stronger regularization.

As shown in Tables 2 and 3, it takes approximately 0.15 s to fit the SPX volatility surface with 20 knots across 46 expiries (920 parameters), calibrated with 14,538 quotes. The calibration times presented here were obtained using an AMD Ryzen 9 5900HX processor. These timings include the two Clarabel optimization steps: the first without linearized constraints, and the second with these constraints enforced. While a modeling language such as CVXPY enables rapid prototyping, bypassing the modeling layer to interface directly with the solver demands considerable additional engineering effort. With a low-level language, however, the overhead involved in setting

up the problem from market data is negligible. Furthermore, the reported timings assume Clarabel's default tolerance. Less strict settings can be used to reduce calibration time without materially impacting fit quality.

Scaling The number of non-zero coefficients in the sparse matrices P and A is also indicated. The number of non-zero coefficients in A increases linearly with the number of parameters per expiry, whereas for P, the growth is quadratic. Calibration time has been observed to grow quadratically with the number of parameters when the number is very high, but in typical use cases, the scaling is much closer to linear.

asset	n_knots	P Non-Zeros	A Non-Zeros	calc_time (s)
SX5E	10	1 046	14 424	0.034
	20	1 766	22 465	0.061
	40	5 836	51 710	0.155
SPX	10	2 905	32 158	0.078
	20	6 706	52 683	0.152
	40	30 717	122 416	0.467

Table 3: Calibration time using Clarabel solver

Conclusion

We presented CVI, an advanced volatility-fitting framework based on convex optimization, which introduces an intuitive parameterization in cubic spline space and its equivalent B-spline representation, framing the volatility fitting as a quadratic programming problem with linear constraints. The objective function combines least squares, quadratic penalties for deviations beyond the bid-ask, and a regularization term to smooth the volatility surface across strikes. Absence of arbitrage is enforced via linear constraints: no-calendar-spread constraints are naturally linear in variance, while we derived and linearized the no-butterfly-arbitrage constraints within the CVI parameter space. As illustrated with SPX and SX5E, the approach calibrates volatility surfaces in hundredths to tenths of a second using modern open-source QP solvers.

On the practical side, enhanced wing stability for demanding applications such as systematic trading or market making has been the subject of further development. Higher-order B-splines would yield smoother Greeks, particularly for exotic derivatives. On the theoretical side, a formal analysis of the linearization convergence properties would be a valuable contribution, and developing a sensible time-interpolation scheme between listed expiries with provable no-arbitrage guarantees would complement the framework.

References

[1] **Diamond S and Boyd S, 2016**
CVXPY: A Python-embedded modeling language for convex optimization
Journal of Machine Learning Research 17(83), pages 1–5

[2] **Fengler M, 2009**
Arbitrage-Free Smoothing of the Implied Volatility Surface
Quantitative Finance 9(4), pages 417–428

[3] **Gatheral J, 2004**
A parsimonious arbitrage-free implied volatility parameterization with application to the valuation of volatility derivatives
Paper presented at Global Derivatives & Risk Management Conference, Madrid.

[4] **Glasserman P and Pirjol D, 2023**
W-Shaped Implied Volatility Curves and the Gaussian Mixture Model
Quantitative Finance 23(4), pages 557–577

[5] **Goulart P and Chen Y, 2024**
Clarabel: An interior-point solver for conic programs with quadratic objectives
Preprint, arXiv

[6] **Le Floc'h F and Oosterlee C, 2019**
Model-free stochastic collocation for an arbitrage-free implied volatility, Part II
Risks 7(1), article 30

[7] **Lee R, 2004**
The Moment Formula for Implied Volatility at Extreme Strikes
Mathematical Finance 14(3), pages 469–480

[8] **Lucic V, 2019**
Volatility Notes
SSRN preprint

[9] **Martini C and Mingone A, 2022**
No arbitrage SVI
SIAM Journal on Financial Mathematics 13(1), pages 227–261