

Convex Volatility Interpolation

Fabrice Deschâtres

September 16, 2025

Abstract

CVI (Convex Volatility Interpolation) is a novel approach to constructing and fitting implied volatility surfaces from observed market option prices. Its key innovation lies in reformulating the problem of fitting arbitrage-free volatility surfaces as an approximately convex optimization task. This allows the fitting process to leverage efficient convex solvers, offering both speed and robustness, and bridging the gap between “flow” and “exotics” fitters.

Specifically, CVI uses a parameterization of the volatility surface in variance space, calibrated using quadratic programming (QP) with linear constraints. Its dual parameterization in cubic spline and B-spline spaces maps a set of intuitive parameters to the weights of basis functions.

As CVI has no restrictions on the number of parameters, it can fit any volatility surface. The method works consistently across all underlyings without the need for hyperparameter tuning, relying on dimensionless numbers for the parameterization and for the relative weights of the different terms in the objective function. The objective function includes various terms that penalize deviations from the mid-price or those beyond the bid-ask, as well as a regularization term in the strike dimension.

Since CVI is model-free, it relies entirely on the optimization process to enforce the absence of static arbitrage. CVI eliminates butterfly and calendar spread arbitrage across all strikes and expiries, including in the tails. While no-calendar-spread-arbitrage constraints are linear in variance space, no-butterfly-arbitrage constraints must be linearized in that space. This paper notably details the derivation and linearization of no-butterfly-arbitrage constraints within the CVI cubic spline parameter space.

This study also highlights the effectiveness of Clarabel, a state-of-the-art, open-source convex optimization solver, in handling the CVI optimization problem. For most underlyings, the calibration time is measured in hundredths of a second, and in tenths of a second for the most liquid ones. As an example, Clarabel can fit the S&P 500 volatility surface, calibrated from 14,500 volatility bids and asks across 46 expiries, involving two iterations of the CVI’s QP problem with 20 parameters per expiry, in just 0.15 seconds.

The calibration of implied volatility surfaces is a central piece of derivative pricing and a much studied problem in the quantitative finance literature. The complexity of this task has attracted considerable attention within the field. It has also captured the imagination of quants with its elegant curves that emerge from the market forces shaping option prices.

Since the very beginning with Black-Scholes, people have attempted to model the processes describing the market dynamics behind these option prices. From the Black-Scholes lognormal process of the 1970s to the stochastic volatility models of the 1990s and the development of the rough volatility approach over the last decade, each generation of models has captured new stylized facts of market behavior.

Yet, the models that best capture market dynamics are not necessarily the most suitable computationally for calibrating volatility surfaces. This paper focuses solely on the practical aspect of calibrating volatility surfaces at a given time from market option bid-asks. By making no assumptions about market dynamics, this approach concentrates on the numerical aspect using a model-free methodology. This paper is intended for practitioners seeking a reliable, high performance numerical recipe for fitting volatility surfaces.

Accurately calibrating these surfaces is challenging and complex, as they must be flexible enough to fit market data well while preventing arbitrage opportunities. Traditional methods for calibrating volatility surfaces tend to have some limitations, such as too few parameters, the presence of arbitrage, numerical problems that affect both stability and speed, or parameters with no intuitive meaning.

A new approach to calibrating volatility surfaces called CVI (*Convex Volatility Interpolation*) overcomes these problems. *Convex* refers to the optimization technique used—convex optimization—rather than to any convexity property of the volatility or variance. Indeed, it can fit any type of volatility smile, including W-shaped ones.

In this paper, we demonstrate how to use this method to fit volatility surfaces and discuss the different building blocks of the methodology, from parameterization to calibration.

The paper is organized as follows:

- In Part 1, we present the properties of a good volatility fitter and discuss their implications to show that CVI arises as a natural solution for volatility fitting.
- Part 2 presents the parameterization based on a dual representation with B-splines and cubic splines. That dual parameterization, which relies on a linear transformation between the two sets of parameters, is a central element of CVI. It serves as a bridge between the mathematical parameter space and the physical one.
- In Part 3, we introduce the different components of the convex optimization used for the calibration, namely the objective function and the constraints. A number of analytical formulas, mainly concerning the absence of butterfly arbitrage, are derived in Appendix A and B. A numerical example of no-butterfly-arbitrage constraints is shown in Appendix C.

- Finally, in Part 4, after some consideration regarding the canonicalization, we highlight the speed and robustness of Clarabel, a cutting-edge convex optimization solver, in handling CVI's quadratic programming problem. We also provide examples of calibration times for various underlyings across a range of numbers of parameters.

In this paper, we define the variance v as the square of the volatility: $v := \sigma^2$. In contrast, vT is referred to as the total variance.

Keywords: volatility surface, volatility interpolation, option pricing, convex optimization, quadratic programming

Contents

1 Problem Formulation	3
1.1 Volatility Surface Usages	3
1.2 The Four Pillars of Volatility Fitting	3
1.3 Analysis of the Problem	4
2 The CVI Parameterization	5
2.1 The CVI Cubic Spline	5
2.2 The CVI B-Spline	7
2.3 Dual Parameterization	8
3 The CVI Optimization	8
3.1 Variables	8
3.2 Objective Function	9
3.3 Constraints	13
3.3.1 Positivity of the variance	13
3.3.2 Boundary Conditions	13
3.3.3 No-Calendar-Spread-Arbitrage Constraints	14
3.3.4 Linearized No-Butterfly-Arbitrage Constraints	14
4 QP solver	19
4.1 Canonical QP Problem	19
4.1.1 Canonicalization with CVXPY	19
4.2 Clarabel	19
4.2.1 Calibration Time	19
5 Conclusion	22
A Non-Arbitrage Conditions	25
A.1 No Strike Arbitrage	25
A.2 No Calendar Spread Arbitrage	25
B No-Butterfly-Arbitrage Constraints	25
B.1 PDF ≥ 0 between z_0 and z_{n-1}	26
B.1.1 Special Case Studies	26
B.2 PDF ≥ 0 at the edge knots and beyond	27
B.2.1 Minimum s in the left tail	27
B.2.2 Maximum s in the right tail	28
B.2.3 Beyond the edge knots	28
C Example of Linearized No-Butterfly-Arbitrage Constraints	29

1 Problem Formulation

In this paper, we assume that the forward and discount factor for each expiry are known or have already been estimated.

Additionally, we assume that the options under consideration are European. If the listed options are American rather than European (for example, single stock options), it is necessary to perform a process known as de-Americanization [5] beforehand. De-Americanization transforms American option prices into European option prices, enabling consistent analysis and calibration within the European option framework.

1.1 Volatility Surface Usages

Volatility surfaces tend to serve several purposes:

- At banks' vanilla desks and among option market makers, the models often are not accurate enough to exactly fit the market prices. Traders frequently make adjustments by applying various offsets to their models, which can be a time-consuming process. Furthermore, for this application, the volatility surfaces only need to be arbitrage free within the core region (listed strikes). However, it is paramount for the parameters to be intuitive, and they should be closely related to the risks that traders manage.
- Banks' exotic desks. Exotic options are path-dependent options typically priced with Monte Carlo or PDE using a stochastic local volatility model. This model is calibrated to match the volatility surface. It is critical for the volatility surface to be arbitrage free across the surface (including in the tails), as the stochastic local volatility model calibration would fail otherwise. For this purpose, models often employ interpolation in price space, not in volatility space, and do not require intuitive parameters (see the work from Matthias Fengler [7] and Nabil Kahalé [10]). That tends to be inappropriate outside of that specific usage. Another popular method involves the Andreasen-Huge interpolation [1], an efficient algorithm that employs a parameterization of the local volatility surface.
- Other users, including hedge funds and market data vendors. Their requirements vary, and since these users generally do not have traders marking the volatility surface, the methods employed by banks' vanilla desks may not be suitable. Furthermore, the models used by banks' exotic desks are not necessarily relevant either. This could result in the use of models that either do not fit the market well or are not arbitrage free.

While these three types of users are essentially trying to solve the common problem of volatility surface calibration, they may employ different methods and make trade-offs based on their specific needs, in the absence of a better model.

1.2 The Four Pillars of Volatility Fitting

In this paper, the requirements of a good volatility surface fitter are assumed to be the following:

- It should accurately fit the market bids and asks in a consistent manner to prevent overfitting.
- It should be arbitrage free in strike and time, without assuming that the market bids and asks are themselves arbitrage free. There should be no calendar spread or butterfly arbitrage, including outside of the listed strikes, in the tails.
- The parameterization should be intuitive. The relationship between risks managed by traders, such as sensitivity to ATM vol and skew, and the parameters should be straightforward.
- The calibration should be stable and fast.

The first two requirements imply that the volatility surface should be calibrated in a single process rather than expiry per expiry. Indeed, the alternative—a sequential calibration, expiry per expiry, similar to the method used by Matthias Fengler [7]—would necessitate prioritizing either short-dated or long-dated expiries to enforce the absence of calendar spread arbitrage, depending on whether it is done forward or backward. However, this method is suboptimal and does not guarantee the best fit to option prices.

Can Fit Any Shape of Vol Surface	Arbitrage Free in Strike (Incl. Tails) And Time
Intuitive Parameters	Numerically Fast And Stable

Figure 1: The Four Pillars of volatility fitting

1.3 Analysis of the Problem

At first glance, the problem may seem daunting. However, we can leverage the complexity of the problem to guide us in the right direction. Let us first break down the problem into several key questions:

- Which parameterization should we use? How many parameters are needed?
- What numerical method should be employed for the calibration?
- How do we enforce non-arbitrage conditions? Should this be built into the parameterization, or should it be enforced by the calibration process?

Number of parameters and the need for convex optimization To the question *How many parameters are needed?*, it is fair to say that there is no definitive answer. One can undoubtedly fit some very illiquid option markets with three parameters per expiry, but liquid equity indexes, which have a well-defined shape, require many more parameters. Moreover, as market participants become more sophisticated and markets more liquid, the number of required parameters can also increase over time. A parameterization with a fixed number of parameters that works today may become less effective tomorrow.

Let us reflect on that. The parameterization needs to accommodate an *arbitrary* number of parameters. What does this imply? Consider a well-known parameterization of the volatility surface: SVI [8]. With only five parameters, a naive calibration is fraught with difficulties, and even with smart initial guesses, its calibration remains challenging. Zeliade developed a particularly robust method [15]. It involves a two-step calibration process where the inner calibration fits three of the parameters via convex optimization, leaving only two parameters to be calibrated with a generic solver.

Non-convex problems typically take exponential time in the number of variables to find the ‘global optimum’. However, many convex optimizations can be solved in polynomial time. As outlined above, the experience with SVI has taught us that with as few as five parameters, non-convex optimization can be hard. And it is not just about speed but also about stability. To have any chance of fitting a volatility surface with an arbitrary number of parameters, we must therefore restrict ourselves to using convex optimization.

The implication of using convex optimization and the need for basis functions Now that we have established the necessity of convex optimization, let us explore the consequences. To fit the market quotes, we expect to utilize quadratic terms derived from a least squares regression. We must also determine the appropriate space for the parameterization—whether it be volatility, variance (i.e., volatility squared), price, etc. To ensure the problem remains convex, the safest approach is to maintain it quadratic, meaning the fitted function, in whichever space chosen, must be a linear function of the parameters. For example, if we choose to parameterize the variance, the variance $v(K, T)$ 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.

Thus, we have determined that our parameterization must be restricted to a function space defined by certain basis functions.

One potential drawback of using basis functions is the lack of interpretability of the parameters themselves; therefore, it is foreseeable that we will need to map these parameters to a secondary parameter space that has a more intuitive meaning.

Non-arbitrage constraints and the need to parameterize the variance Having established the necessity of (1) employing convex optimization and (2) relying on basis functions for parameterization, we are compelled to abandon the notion of an intrinsically arbitrage-free parameterization, which would prove challenging to reconcile with the other requirements. Consequently, the absence of arbitrage will have to be enforced through calibration (i.e., convex optimization) rather than being inherently embedded in the parameterization itself. In other words, *non-arbitrage conditions become non-arbitrage constraints* within the convex optimization process.

The non-arbitrage conditions are documented in Appendix A. Our review of these conditions indicates:

- The no-butterfly-arbitrage condition, which enforces convexity of the price as a function of the strike, is more naturally applied in price space. If we parameterize the *price*, implementation should be relatively simple.
- The no-calendar-spread-arbitrage condition, which ensures that the total variance curves do not intersect, is more naturally addressed in variance space. Parameterizing the *variance* facilitates straightforward implementation of this condition.

This leads us to a dilemma: should we parameterize the volatility surface in *price* or *variance* space? Should we prioritize addressing butterfly arbitrage at the cost of a potentially more challenging approach to calendar arbitrage, or vice versa?

Before making a decision, let us make a few more observations:

- For extreme low/high strikes, assuming that the variance is a linear function of the log strike (like in SVI) simplifies the modeling. Lee’s tail slope bounds, which addresses butterfly arbitrage in the far wings, is naturally expressed in variance space, making it easy to handle butterfly arbitrage at extreme strikes in this space.

- How could we develop intuitive parameters for the *volatility* surface in *price* space?

Given these considerations, parameterizing in variance space emerges as the natural choice. The consequences of this choice include:

- No-calendar-spread-arbitrage conditions are linear constraints.
- No-butterfly-arbitrage conditions are linear constraints for extreme strikes, but they are non-linear (and non-convex) for non-extreme strikes. Iterating and linearizing these constraints offers a practical solution to remain within the scope of convex optimization.

B-spline Any basis functions could be employed; however, it is natural to opt for order 3 B-splines, which are the basis functions of cubic splines and are widely utilized for interpolating volatility. In the rest of the paper, we will use order 3 B-splines, but it could easily be adapted to higher order B-splines or other basis functions.

This entails a dual representation of the parameterization of the variance:

- The B-spline parameters (mathematical space): These do not carry any special intuitive meaning; they are merely the weights of the basis functions.
- The cubic spline parameters (physical space): Various parameterizations are possible in terms of either the value of the cubic spline or its derivatives at each knot, provided there are as many degrees of freedom as there are constraints. An intuitive parameterization is presented in Section 2.1.

2 The CVI Parameterization

After analyzing the volatility fitting problem, we have established the following criteria:

- The parameterization must accommodate an arbitrary number of parameters.
- The calibration should be based on convex optimization.
- The variance $v(K)$ should be represented using a set of basis functions, specifically order 3 B-splines, and is assumed to be a linear function of $\log(K)$ for extreme strikes.

This section introduces a dual parameterization of the variance using cubic splines (see Section 2.1) and its equivalent B-spline representation (see Section 2.2).

2.1 The CVI Cubic Spline

We call $k := \log(K/F)$ the log-forward moneyness. z is the normalized log-moneyness:

$$z := \frac{k}{\sigma_* \sqrt{T}}$$

where σ_* is an estimate of the ATM (At-The-Money-Forward) volatility, which we will simply note ATM. σ_* is called the anchor ATM volatility and is determined *prior* to the convex optimization. It is approximately equal to $\sigma(z = 0)$ and does not have to be perfectly accurate; its purpose is to scale the log strikes into the dimensionless number z . We rely on σ_* instead of directly using $\sigma(z = 0)$ to keep the optimization convex. Consequently, the relationship between K and z is established before optimization, while the ‘true’ ATM volatility $\sigma(z = 0)$ is the result of the convex optimization.

Next, let us discretize the z space and introduce z_0, z_1, \dots, 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.

Parameters The cubic spline has the following $n + 2$ inputs:

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

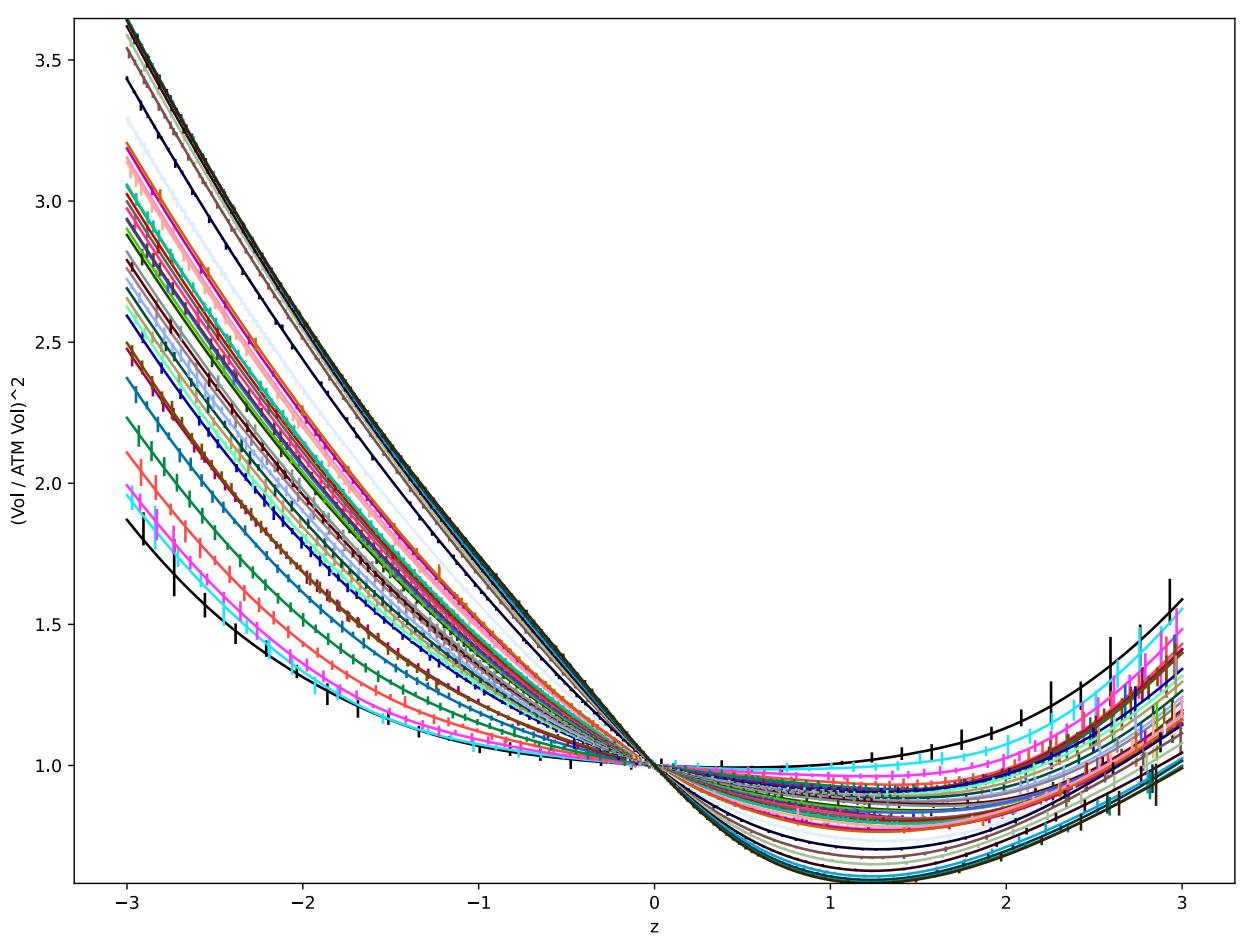


Figure 2: Shape of the S&P 500 Vol Surface as of December 5th, 2023. For each expiry, the shape function $f(z) = \frac{v(z)}{v_*}$ is plotted. s and c represent its first and second derivatives, respectively.

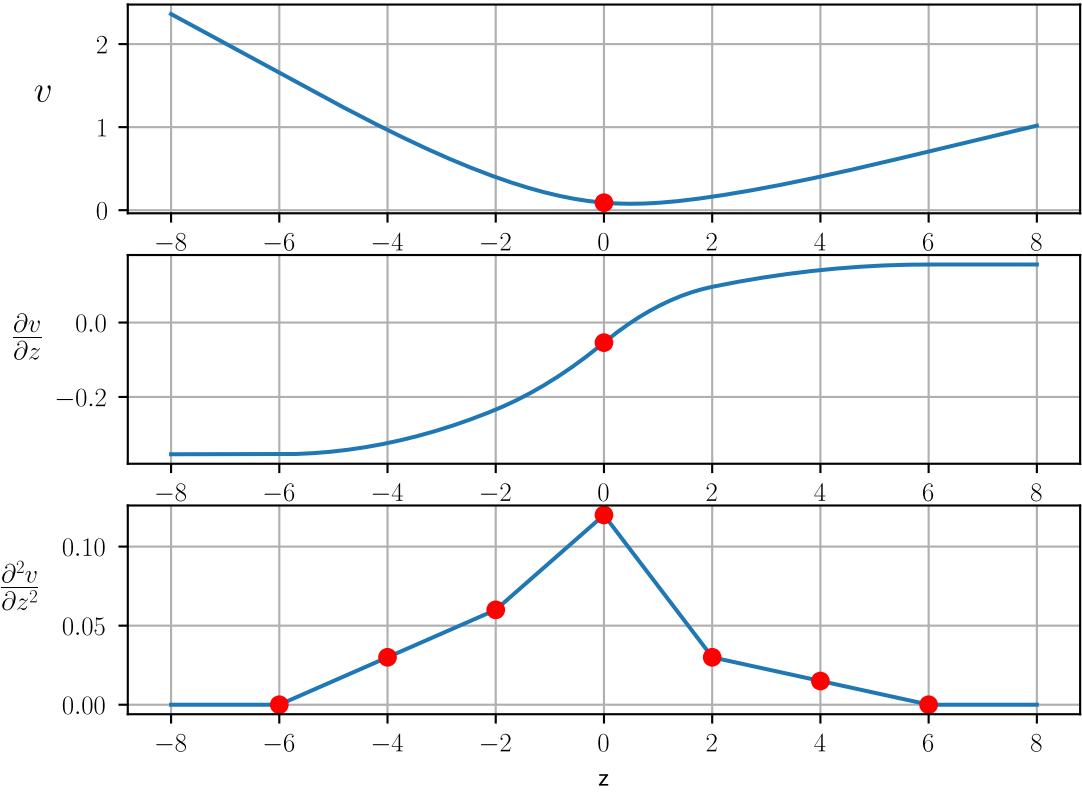


Figure 3: The CVI cubic spline (top), its first derivative (middle), and second derivative (bottom). The red dots indicate the values that the cubic spline and its derivatives are required to match. The second derivative on the edge knots (-6 and 6 in that example) is zero by design (linear extrapolation of the variance as a function of the log strike outside the knots).

s and c It is convenient to introduce the following notations:

$$s := \frac{1}{v_*} \frac{\partial v}{\partial z}$$

$$c := \frac{1}{v_*} \frac{\partial^2 v}{\partial z^2}$$

s and c are, respectively, the first and second derivatives of the variance with respect to z , normalized by the anchor ATM variance $v_* := \sigma_*^2$. They are dimensionless measures of the skew and the convexity. As noted earlier, the normalization factor $\frac{1}{v_*}$ is determined prior to the convex optimization.

These notations are analogous to s_2 and c_2 used by Tim Klassen (see [11], p. 9), although in this paper, s and c are considered functions of z rather than being evaluated at the money.

Cubic Spline Linear System The cubic spline consists of $n+1$ third-order polynomials, which collectively have $4(n+1)$ degrees of freedom. These are found by solving a $4(n+1)$ by $4(n+1)$ linear system using the following $4(n+1)$ relations:

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

In practice, the convex optimization constraints enforce that, $\frac{\partial^2 v}{\partial z^2} \Big|_{z_0} = 0$ and $\frac{\partial^2 v}{\partial z^2} \Big|_{z_{n-1}} = 0$. This ensures that the variance is linear in log strike for extreme strikes (i.e., for $z \leq z_0$ and $z \geq z_{n-1}$).

2.2 The CVI B-Spline

As described before, the variance $z \rightarrow v_T(z)$ is represented by a cubic spline, a function of $n+2$ parameters: the ATM variance, the ATM skew, and the convexities at n knots.

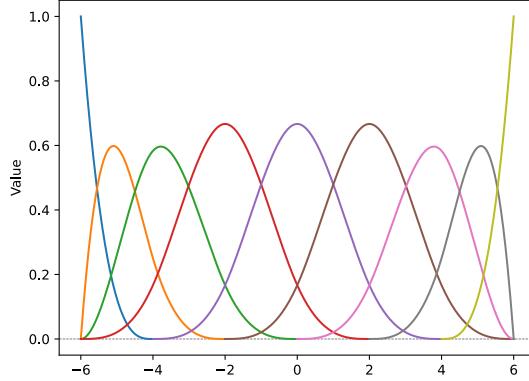


Figure 4: The 9 B-spline functions associated with the 7 knots $[-6, -4, -2, 0, 2, 4, 6]$.

$$v_T(z) = F_{\text{cubic spline}} \left(v_T(z=0), \frac{\partial v_T(z)}{\partial z} \Big|_{z=0}, \frac{\partial^2 v_T(z)}{\partial z^2} \Big|_{z_0}, \dots, \frac{\partial^2 v_T(z)}{\partial z^2} \Big|_{z_{n-1}} \right)$$

That same function can also be represented by a basis spline, a function of $n+2$ parameters.

$$v_T(z) = F_{\text{basis spline}}(\alpha_{0,T}, \alpha_{1,T}, \dots, \alpha_{n+1,T}) \\ = \begin{cases} \sum_{i=0}^{n+1} \alpha_{i,T} f_i(z) & \text{if } z_0 \leq z \leq z_{n-1} \\ \text{linear extrapolation} & \text{if } z \leq z_0 \text{ or } z \geq z_{n-1} \end{cases}$$

If $z \geq z_0$ and $z \leq z_{n-1}$, $\{\alpha_{i,T}\}_{\{0 \leq i \leq n+1\}} \rightarrow v_T(z)$ is by design a linear function (i.e., the variance is a linear function of the basis spline weights). This is also true for $z < z_0$ and $z > z_{n-1}$, as the extrapolation is linear. Furthermore, if we reason in strike space, rather than z space, this is also true, i.e., $\{\alpha_{i,T}\}_{\{0 \leq i \leq n+1\}} \rightarrow v_T(K)$ is linear given that $z = \frac{\log(K/F)}{\sigma_* \sqrt{T}}$, with σ_* estimated beforehand. Thus, the least squares on $v(K)$ are a quadratic function of $\{\alpha_{i,T}\}_{0 \leq i \leq n+1}$ inside and outside the support of the B-spline.

2.3 Dual Parameterization

We have shown that the volatility surface $z \rightarrow v(z)$ can be effectively represented either by cubic splines or B-splines.

These two parameterization methods are linked by a known linear transformation, allowing parameters from one form to be converted into the other seamlessly.

Given this linear relationship, either set of parameters can be interchangeably utilized in the convex optimization discussed in Section 3.

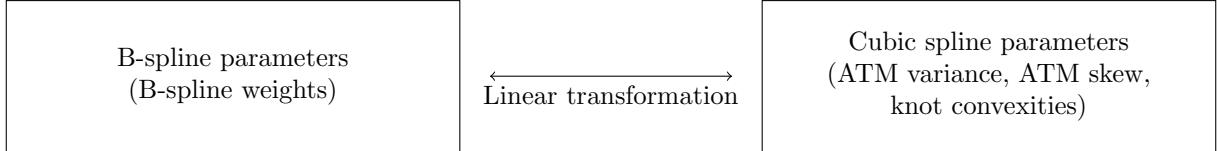


Figure 5: The volatility surface can be represented via B-spline parameters or cubic spline parameters.

3 The CVI Optimization

In this section, we present the different building blocks of the CVI convex optimization: the variables (Section 3.1), the objective function (Section 3.2), and the constraints (Section 3.3).

3.1 Variables

Assuming m expiries T_1, T_2, \dots, T_m , the convex optimization solves for $m(n+2)$ variables, denoted by $\{\alpha_{i,T_j}\}_{0 \leq i \leq n+1, 1 \leq j \leq m}$. All expiries are fitted simultaneously in one single optimization to ensure that no-calendar-spread-arbitrage constraints are properly enforced.

Each expiry involves $n + 2$ parameters¹, corresponding to the $n + 2$ basis spline coefficients. These coefficients can be converted through a linear transformation into the $n + 2$ cubic spline parameters: $v_T(z = 0)$, $\frac{\partial v_T(z)}{\partial z} \Big|_{z=0}$, and $\left\{ \frac{\partial^2 v_T(z)}{\partial z^2} \Big|_{z_i} \right\}_{0 \leq i \leq n-1}$. Both parameter sets are essential in defining the constraints and objectives of the convex optimization.

3.2 Objective Function

In this section, when employing the word ‘option’, we do not differentiate between calls and puts, and the bids and asks refer to the tightest vols across calls and puts.

Let us define \mathcal{S} as the set of all options listed on the asset. For any given expiry T , \mathcal{S}_T denotes the subset of options expiring at T . Additionally, we define $\mathcal{S}_{T,\text{ask}}$ (respectively, $\mathcal{S}_{T,\text{bid}}$) as the subset of options expiring at T with a valid ask price (respectively, bid), and $\mathcal{S}_{T,\text{mid}} := \mathcal{S}_{T,\text{ask}} \cap \mathcal{S}_{T,\text{bid}}$ the subset of options expiring at T with both a bid and an ask. We note $N_{T,\text{ask}}$, $N_{T,\text{bid}}$ and $N_{T,\text{mid}}$ as the numbers of options in $\mathcal{S}_{T,\text{ask}}$, $\mathcal{S}_{T,\text{bid}}$ and $\mathcal{S}_{T,\text{mid}}$, respectively.

The objective function is the sum of the following terms:

Least Squares Penalty + Above Ask Penalty + Below Bid Penalty + Strike Regularization Penalty

Each term, while straightforward, requires careful consideration of its weight relatively to the others to ensure consistent behavior regardless of the number of expiries, options per expiry and the level of the volatility. As the ‘Strike Regularization Penalty’ is dimensionless, the other terms are weighted to scale like a chi-square (sum of chi-squares across expiries). This scaling allows for the use of the same regularization factor λ across all underlyings.

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. However, the range $[z_0, z_{n-1}]$ should ideally be chosen to include most, if not all, options in the first place.

Least Squares Penalty

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

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 option of strike K_i and expiry T_j .

The double summation in the least squares penalty is simply summing over expiries and options of each expiry for options with both a bid and an ask.

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

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

With that penalty, very wide quotes have a negligible impact on the fit, as can be seen in Figure 6.

Above Ask Penalty This penalty is imposed on options whose variance exceeds the ask variance. Contrary to the Least Squares Penalty, this also applies to options that are offered only. For instance, typically, very OTM options are offered only. The weights differ from the least squares penalty as this applies to all options with an ask price (with or without a bid price). In addition, the weights are proportional to the vega rather than inversely proportional to the bid-ask vol spread (which does not exist if there is no bid).

Let us call $\mathcal{V}_{\text{ask}}(K, T)$ the vega associated with the ask price of option with strike K and expiry T .

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

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

$$w_{i,j,\text{ask}} = q_j \frac{\mathcal{V}_{\text{ask}}(K_i, T_j)}{\sum_{k \in \mathcal{S}_{T_j,\text{ask}}} \mathcal{V}_{\text{ask}}(K_k, T_j)}$$

with

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

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-square statistic—analogous to the Least Squares penalty.

¹As the edge convexities $\frac{\partial^2 v_T(z)}{\partial z^2} \Big|_{z_0}$ and $\frac{\partial^2 v_T(z)}{\partial z^2} \Big|_{z_{n-1}}$ are set to zero, there are only n free parameters per expiry.

Below Bid Penalty This penalty is imposed on options whose variance falls below the bid variance, effectively mirroring the Above Ask Penalty.

We denote $\mathcal{V}_{\text{bid}}(K, T)$ as the vega associated with the bid price of option with strike K and maturity T .

$$\text{Below Bid Penalty} = \sum_{j=1}^m \frac{1}{N_{T_j, \text{bid}}} \sum_{i \in \mathcal{S}_{T_j, \text{bid}}} w_{i,j, \text{bid}} \max(\mathcal{V}_{\text{bid}}(K_i, T_j) - \mathcal{V}(K_i, T_j), 0)^2$$

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

$$w_{i,j, \text{bid}} = q_j \frac{\mathcal{V}_{\text{bid}}(K_i, T_j)}{\sum_{k \in \mathcal{S}_{T_j, \text{bid}}} \mathcal{V}_{\text{bid}}(K_k, T_j)}$$

Tip: The ‘Above Ask’ and ‘Below Bid’ terms are important as the least squares penalty does not account for all the market quotes. Removing these two terms in the objective function leads to some loss of information, particularly in the tails where typically very out-of-the-money (OTM) options are offered only. Without these terms, the optimization can generate a fit outside of the bid-offer for options offered only (or bid only) without incurring a penalty. However, the ‘Above Ask’ and ‘Below Bid’ terms necessitate the addition of a number of auxiliary variables in the canonicalization process (see 4.1), which increases the size of the P matrix. This impacts the solver’s performance and can, in some cases, increase the calibration times severalfold.

Applying these penalties for strikes that have a bid or an offer but not both is a good compromise, as it does not lead to any loss of information while minimizing impact on speed. The calibration times reported in 4.2.1 are based on that assumption.

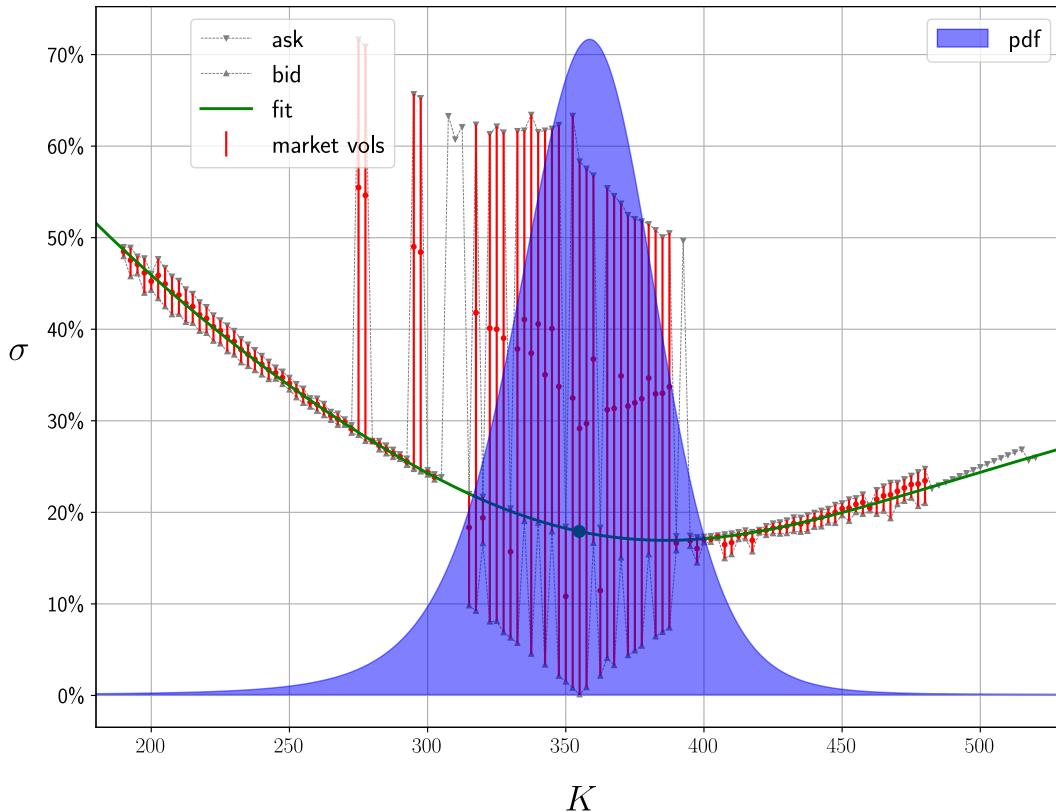
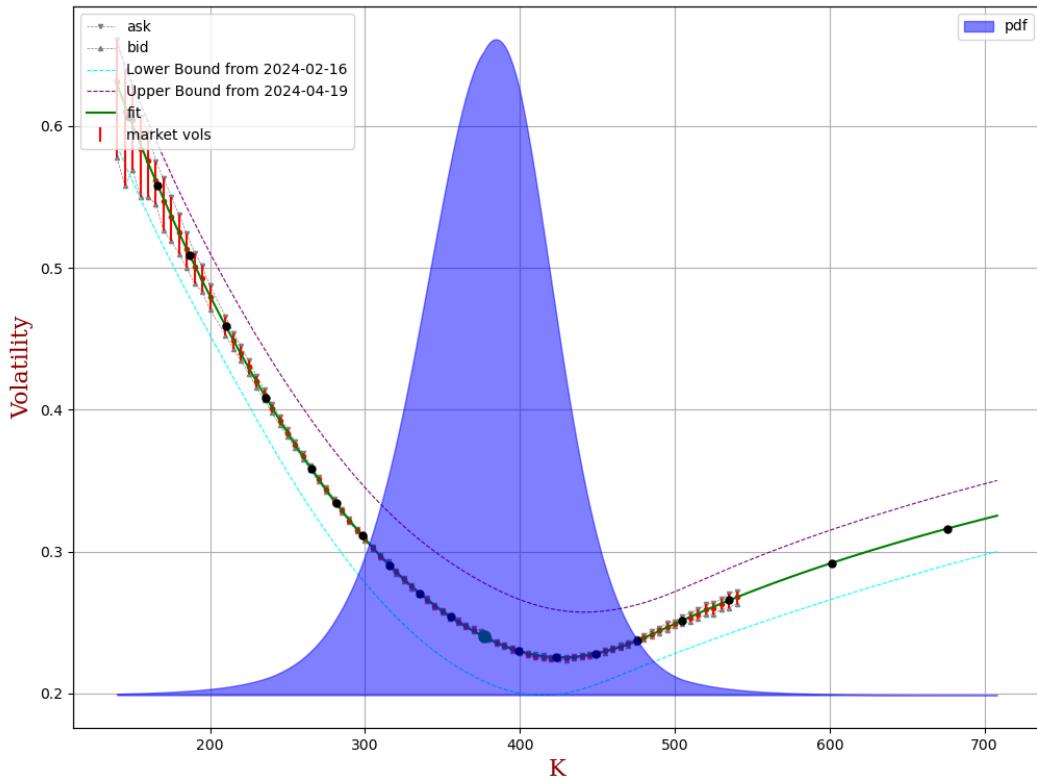


Figure 6: Volatility fit and probability density function (KOSPI 200). With a Least Square Penalty equivalent to a chi-square in variance space, very wide quotes have little impact on the fit.

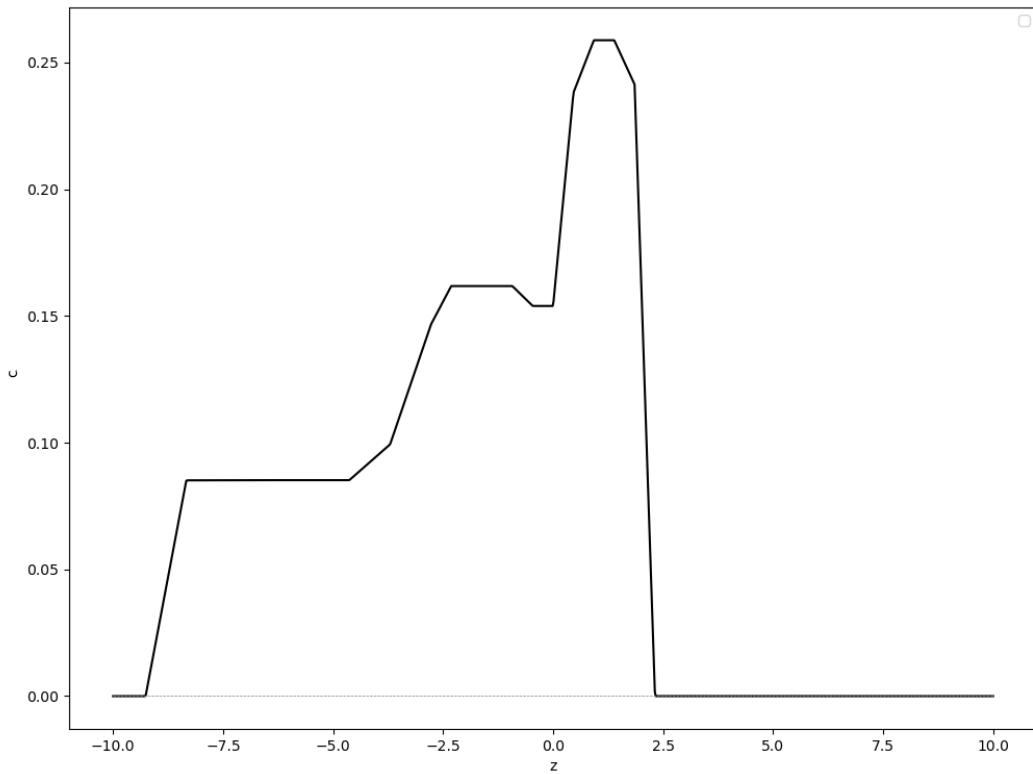
Strike Regularization Penalty This penalty is applied to each expiry T to maintain a smooth smile curve and prevent overfitting. It is proportional to the L_1 norm of the dimensionless second derivative coefficients c , thus focusing on the differences in curvature between adjacent knots to enhance smoothness:

$$\text{Strike Regularization Penalty} = \lambda \sum_T \sum_{i=0}^{n-2} |c_T[i] - c_T[i+1]|$$

where λ is the regularization factor. A higher λ ensures a smoother cubic spline and less overfitting. For example, using $\lambda = 0.05$ has proven effective in practice for all underlyings. As the other penalty terms scale like a chi-square, an intuitive interpretation is that we are ready to deteriorate the chi-square by approximately 0.05 in order to reduce the L_1 norm of c by 1, and vice versa.

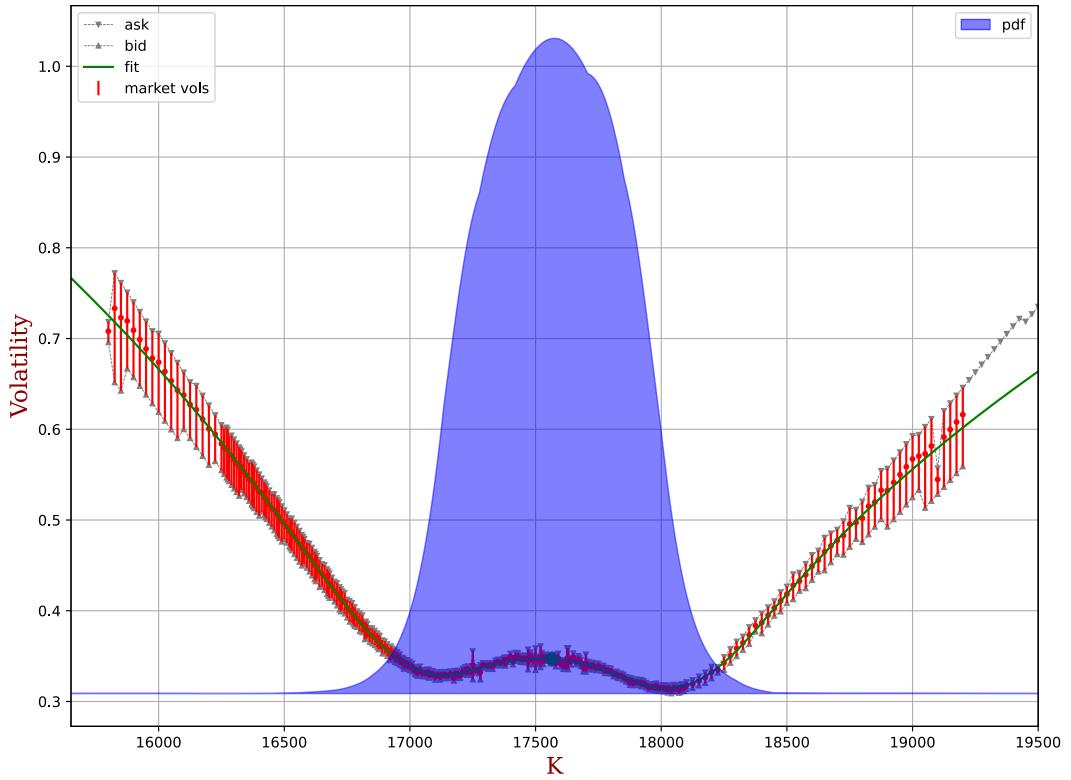


(a) Volatility fit and probability density function

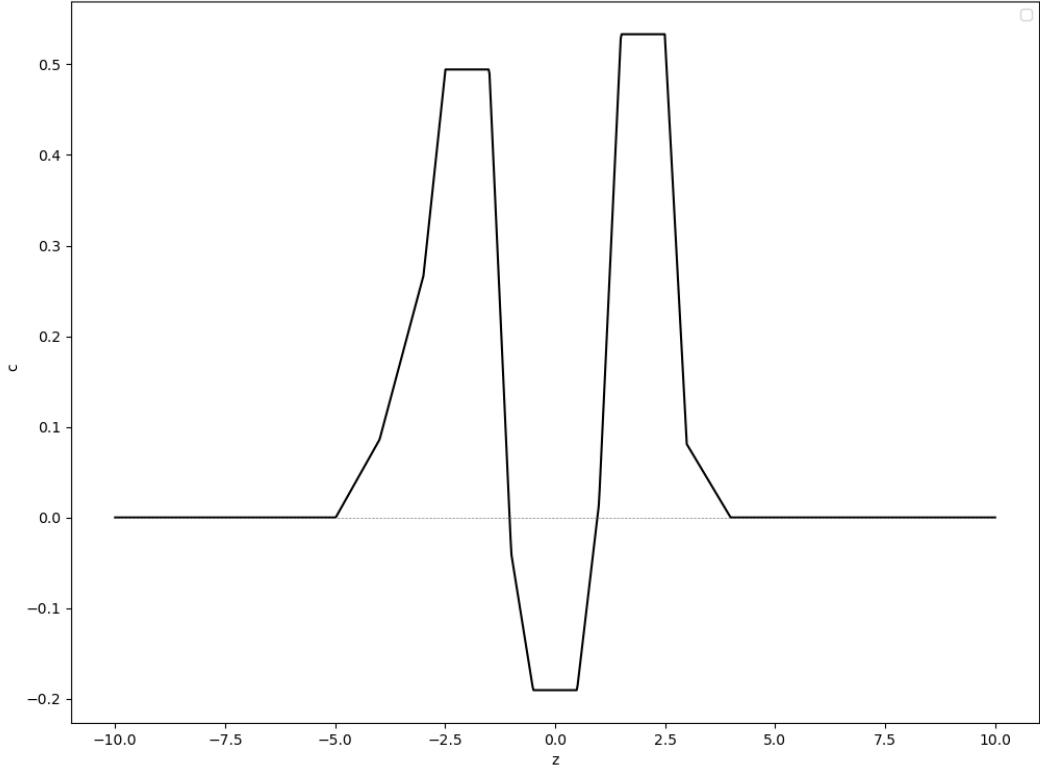


(b) c as a function of z

Figure 7: Microsoft March 15th, 2024 expiry as of December 5th, 2023. The ‘Strike Regularization Penalty’ is proportional to the L_1 norm of the c curve.



(a) Volatility fit and probability density function. It is worth noting that for strike 19,100, the ask volatility at 55.7% is much lower than for nearby strikes. The CVI fit (green curve) is outside of the bid-offer for that strike. CVI would require a much lower regularization factor for the fit to be within that bid-ask.



(b) c as a function of z

Figure 8: NASDAQ-100 (NDX) one day option as of April 30th, 2024, before Amazon quarterly earnings. For tech stock/indices, it is not unusual to see a negative c around the ATM strike on short dated expiries before the earnings. Known as W-shaped smile, or more colloquially as ‘moustache’ [2], this effect, when strong enough, can give rise to bimodal probability density functions.

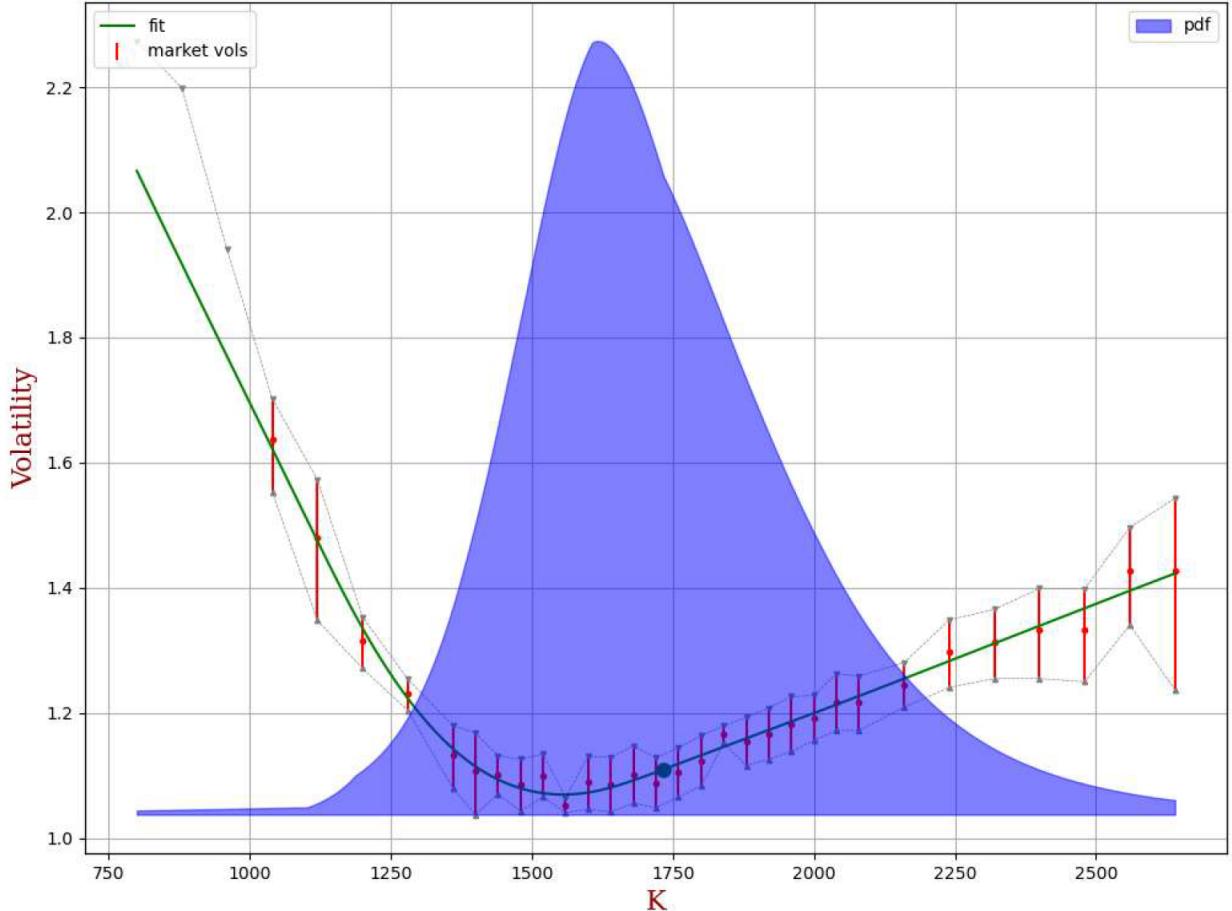


Figure 9: The Strike Regularization Penalty prevents overfitting. A higher regularization factor λ allows the fit to deviate further from the market prices in order to produce a smoother curve. In this example, a couple of options are at the limit of having a fair value outside of the bid-ask spread.

3.3 Constraints

The convex optimization employs the following detailed linear constraints.

3.3.1 Positivity of the variance

The variance, v , must be positive for each point z . To ensure this, we add the linear constraint $v(z) > 0$ on a set of points. Applying this constraint to the first expiry suffices, as subsequent expiries automatically meet the lower bounds set by the no-calendar-spread-arbitrage constraints. And this only needs to be enforced between z_0 and z_{-1} as the tails are taken care of by Equation 1.

3.3.2 Boundary Conditions

Linear extrapolation Consistent with the boundary conditions, in order to linearly extrapolate v beyond the cubic spline knots, convexity is zero at the edge knots z_0 and z_{n-1} .

$$\text{Linear variance in the wings} \left\{ \begin{array}{l} \frac{\partial^2 v}{\partial z^2}(z_0) = 0 \\ \frac{\partial^2 v}{\partial z^2}(z_{n-1}) = 0 \end{array} \right.$$

Upward sloping wings As the variance cannot get negative and as it is linearly extrapolated beyond the cubic spline knots, the variance has to increase when going further in the wings:

$$\text{Increasing variance in the wings} \left\{ \begin{array}{l} \frac{\partial v}{\partial z}(z_0) \leq 0 \\ \frac{\partial v}{\partial z}(z_{n-1}) \geq 0 \end{array} \right. \quad (1)$$

Lee's tail slope bounds Beyond the first knot z_0 and the last knot z_{n-1} , the variance is linearly extrapolated in z space, and its slope cannot be too steep to satisfy Lee's tail slope bounds.

Equations 7 and 8 can be rewritten as:

$$\text{Lee's maximum tail skew} \left\{ \begin{array}{l} \frac{\sqrt{T}}{2\sigma_*} \frac{\partial v}{\partial z}(z_0) > -1 \\ \frac{\sqrt{T}}{2\sigma_*} \frac{\partial v}{\partial z}(z_{n-1}) < 1 \end{array} \right. \quad (2)$$

These equations guarantee the absence of large- (or small-) strike arbitrage.

3.3.3 No-Calendar-Spread-Arbitrage Constraints

The no-calendar-spread-arbitrage constraints are detailed in Appendix A.2. For any fixed strike-to-forward ratio, the total variance must be an increasing function of time. In practice, this is enforced on a finite set of strikes. If we consider r strikes, we thus end up with $(m - 1)r$ linear constraints:

$$\left\{ v(K_i, T_j)T_j \leq v \left(K_i \frac{F_{T_{j+1}}}{F_{T_j}}, T_{j+1} \right) T_{j+1} \right\} \Big|_{1 \leq j < m, 0 \leq i < r} \quad (3)$$

Depending on the tolerance for small arbitrage, it could be applied to a number of strikes ranging from 10 to 100. If there are, say, 20 expiries, the number of constraints would then range from 190 to 1900. These strikes could, for instance, be linearly spaced in z space between the edge knots z_0 and z_{n-1} .

No-calendar-spread-arbitrage constraints in the tails In the tails (i.e. $z \leq z_0$ or $z \geq z_{n-1}$), the variance is assumed to be an affine function of z . Equation 3 (after deriving it by $\log(K)$) implies that at the edge knots z_0 and z_{n-1} , the function $T \rightarrow s(T)$ (which is negative for z_0 and positive for z_{n-1}) cannot decay faster than the standard deviation $\sigma_*(T)\sqrt{T}$.

For $T_1 < T_2$, we can write:

$$\begin{cases} s(z_0, T_1)\sigma_*(T_1)\sqrt{T_1} & \geq s(z_0, T_2)\sigma_*(T_2)\sqrt{T_2} \\ s(z_{n-1}, T_1)\sigma_*(T_1)\sqrt{T_1} & \leq s(z_{n-1}, T_2)\sigma_*(T_2)\sqrt{T_2} \end{cases} \quad (4)$$

Hence in the tails, there is no need to enforce Equation 3 directly. Instead, it is more efficient to apply Equation 4 as this only needs to be implemented at z_0 and z_{n-1} .

3.3.4 Linearized No-Butterfly-Arbitrage Constraints

The no-butterfly-arbitrage constraints (Equations 13, 18, 19) are inherently non-linear and non-convex. Therefore, rather than applying these constraints directly, the workaround is to linearize them using the solution found at the previous iteration.

Iteration

- The first run of the convex optimization is done without the no-butterfly-arbitrage constraints.
- Subsequent runs are performed with the constraints linearized at the previous solution.

In pseudo-code, this looks like this:

```
# First iteration: run without no-butterfly-arbitrage constraints
solution = optimize(no_butterfly_arb_constraints=[])
i = 1

while solution.has_butterfly_arbitrage and i < n_itermax:
    # Subsequent iterations: use no-butterfly-arbitrage constraints
    # linearized at the previous solution
    solution = optimize(no_butterfly_arb_constraints=
        solution.butterfly_arb_constraints)
    i += 1
```

In practice, using $n_{itermax} = 2$ often suffices, i.e., this involves solving twice a convex optimization problem (or only once if there is no butterfly arbitrage at the first iteration).

Notation Let us define $z \rightarrow \sigma_{ref}^2(z)$ as the solution from the previous iteration. The normalized skew s of that solution is noted: $z \rightarrow s_{ref}(z) := \frac{1}{v_*} \frac{\partial v_{ref}}{\partial z}$.

PDF ≥ 0 between z_0 and z_{n-1} Let us linearize Equation 13 at the solution of the previous iteration, which is denoted as "ref". The linearized form is:

$$c \geq \beta_0 + \beta_1(s - s_{ref}) + \beta_2(v - v_{ref}) \quad (5)$$

with

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

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

In Equation 5, the optimization variables are v , s , and c , while β_0 , β_1 , and β_2 are known coefficients.

That constraint should be applied to a set of z strictly between z_0 and z_{n-1} . Depending on the tolerance for small arbitrage, you may consider applying it at 10 to 100 strikes per expiry.

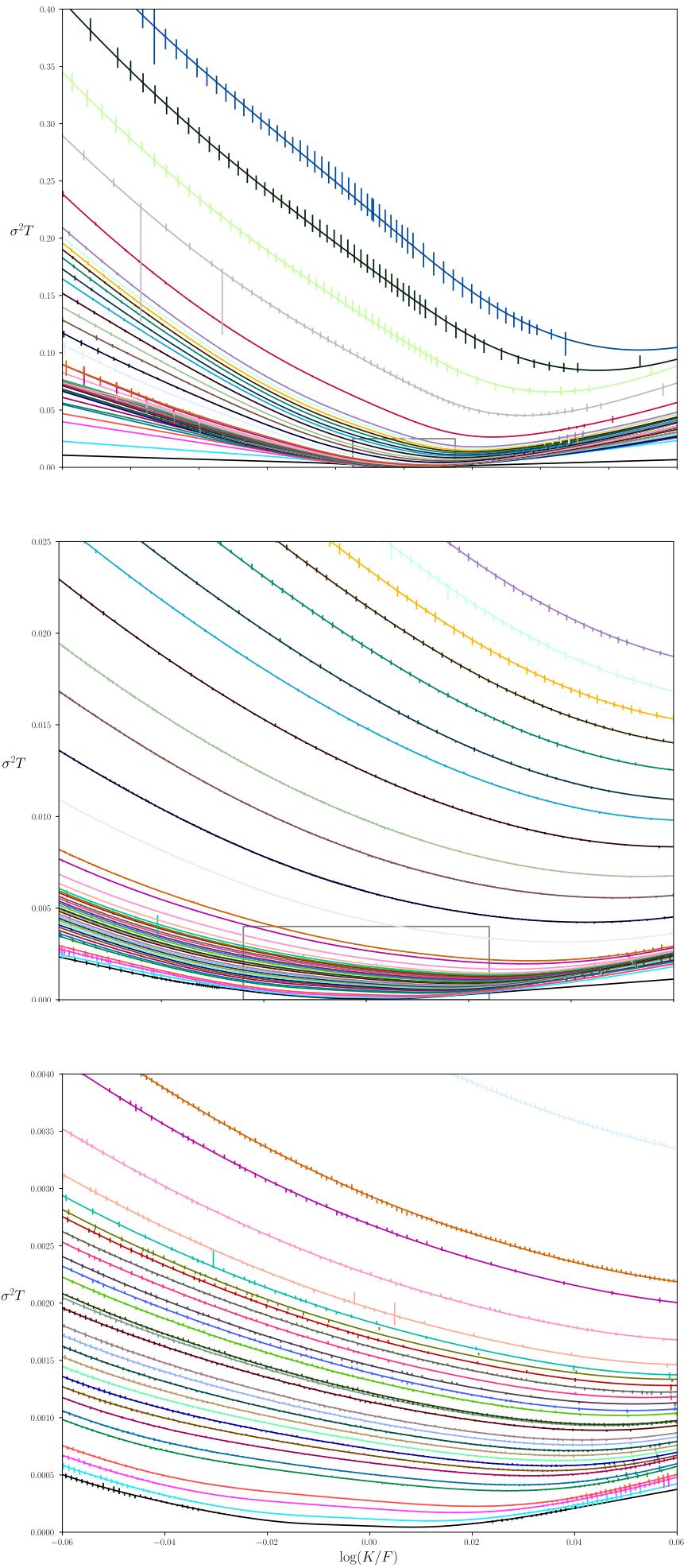


Figure 10: Total variance vT as a function of the log-moneyness $\log K/F$ for the S&P 500. The three figures represent the same plot with different levels of zoom. Each curve represents an expiry. The curves do not intersect to ensure the absence of calendar spread arbitrage. This is enforced by Equations 3 and 4.

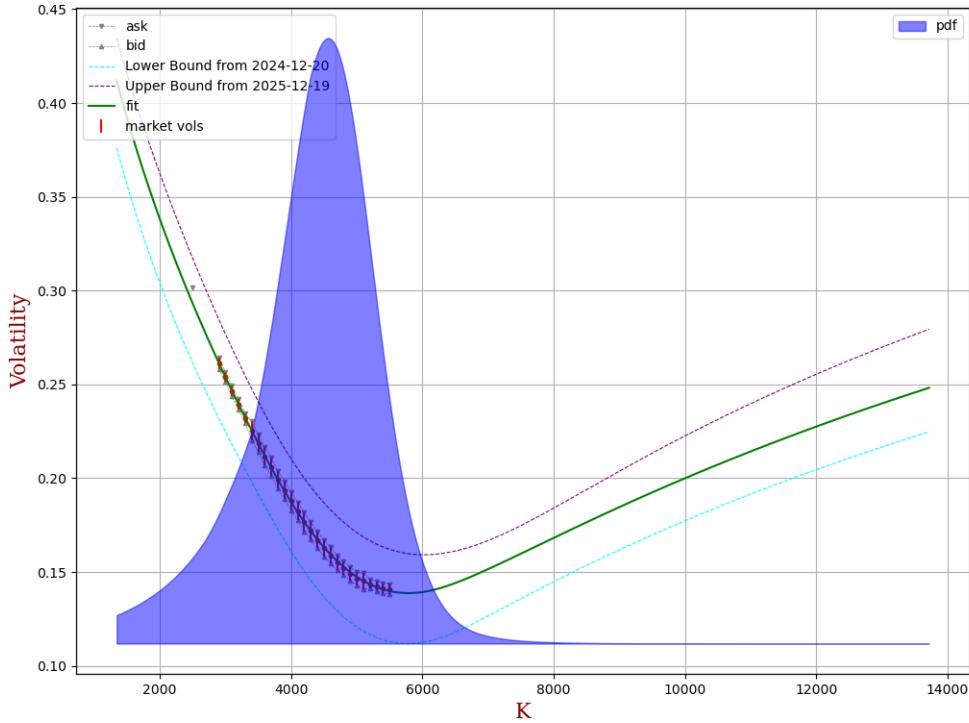


Figure 11: Despite having no quotes beyond the strike with minimum volatility, that expiry has a sensible right wing. The no-calendar spread constraints propagate information about the shape of the volatility surface across expiries (the no-calendar spread constraints are materialized by the channel delimited by the cyan and purple curves).

PDF ≥ 0 at the edge knots and beyond The Equation 18 (resp. 19) needs to be linearized and applied at z_0 (resp z_{n-1}).

For z_0 ,

$$s \geq s_{\min}(v_{\text{ref}}) + \frac{\partial s_{\min}}{\partial v} \Bigg|_{v_{\text{ref}}} (v - v_{\text{ref}})$$

with

$$s_{\min} = -\sigma\sqrt{T} \frac{-k - \sqrt{1 + \frac{vT}{4}}\sigma\sqrt{T}}{k^2 - \frac{v^2T^2}{4} - vT} \frac{2\sigma}{\sigma_*}$$

For z_{n-1} ,

$$s \leq s_{\max}(v_{\text{ref}}) + \frac{\partial s_{\max}}{\partial v} \Bigg|_{v_{\text{ref}}} (v - v_{\text{ref}})$$

with

$$s_{\max} = \sigma\sqrt{T} \frac{k - \sqrt{1 + \frac{vT}{4}}\sigma\sqrt{T}}{k^2 - \frac{v^2T^2}{4} - vT} \frac{2\sigma}{\sigma_*}$$

We leave it to the reader to calculate the derivative of s_{\min} and s_{\max} with respect to v .

These two equations only need to be enforced at the edge knots, and not beyond as detailed in Appendix B.2.

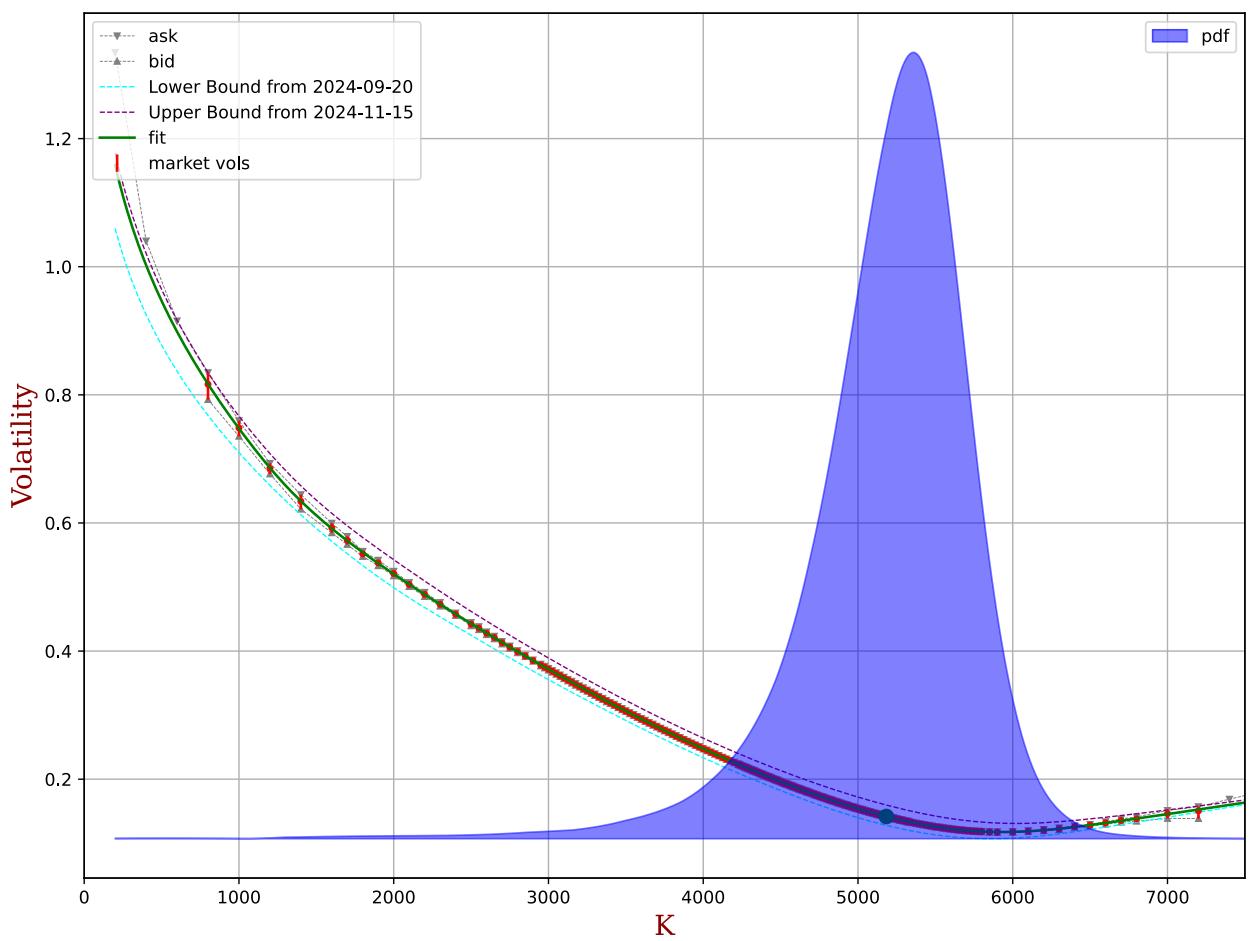
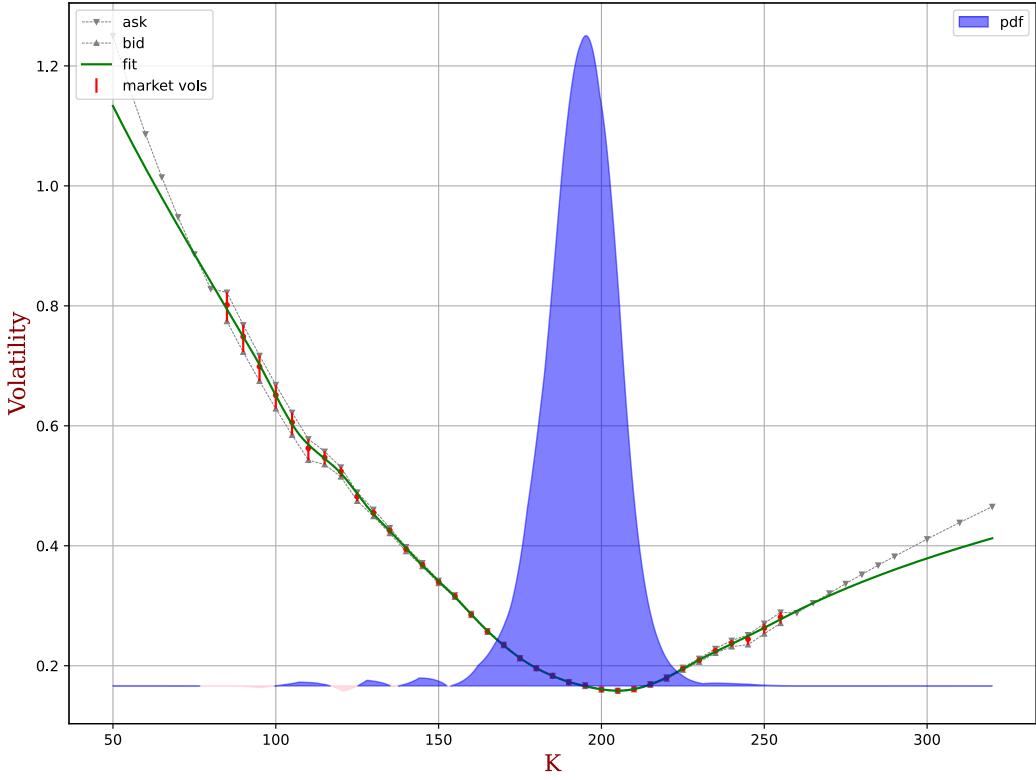
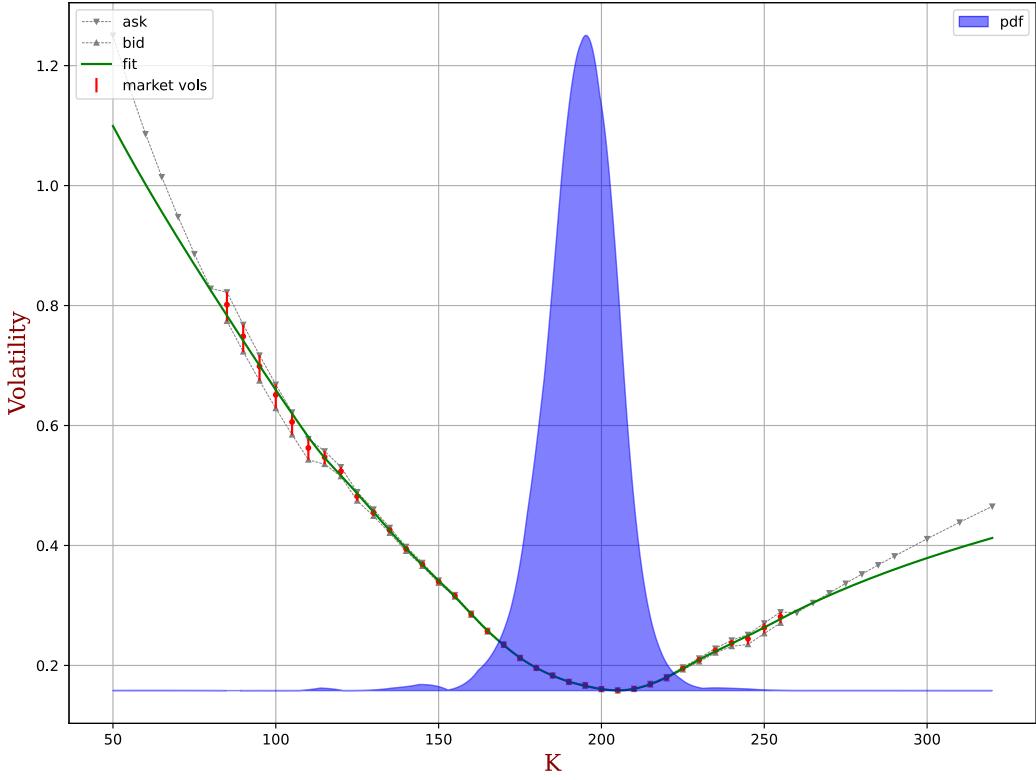


Figure 12: S&P 500 (SPX) October 18th, 2024 expiry as of April 23th, 2024. The bid asks are in red and in grey (red bars are only available for strikes with both a bid and an ask). The fit is shown in green. The blue-shaded curve is the probability density function (implied by the fitted volatility) on the right axis. Its positivity is enforced by the linearized butterfly constraints discussed in Section 3.3.4.



(a) Without no-butterfly-arbitrage constraints, the probability density function can be negative (pink shaded area).



(b) With no-butterfly-arbitrage constraints, the probability density function stays positive

Figure 13: AAPL January 19th, 2024 expiry as of December 6th 2023. In the first plot (a), the linearized no-butterfly-arbitrage constraints have not been applied. The probability density function is negative (pink zone around strike 80-100 as the skew is too steep and around strike 120 as the vol is too concave). In the second plot (b), with the no-butterfly-arbitrage constraints, the volatility fit moves further from the mid for those strikes to prevent arbitrage.

4 QP solver

In this section, we review the canonical form of a quadratic programming (QP) problem and specifically examine the Clarabel solver. We also present results regarding the time taken to solve these convex optimization problems.

4.1 Canonical QP Problem

CVI is based on quadratic programming with linear constraints. In its canonical form, a convex quadratic program with n_v variables and n_c constraints can be formulated as follows:

$$\begin{aligned} & \text{minimize} && \frac{1}{2}x^T Px + q^T x \\ & \text{subject to} && l \leq Ax \leq u \end{aligned} \tag{6}$$

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, $\mathbb{R} \cup \{+\infty\}$).

4.1.1 Canonicalization with CVXPY

To solve the QP problem, it must be expressed in its canonical form. This can be easily accomplished using a modeling language designed for convex optimization, such as CVXPY [6], an initiative from the Department of Electrical Engineering at Stanford University.

For readers interested in code examples of CVXPY applications in finance, the following papers may be referenced:

- *Markowitz Portfolio Construction at Seventy* ([4]): A review of portfolio optimization, the most widespread usage of convex optimization in finance.
- *Convex optimization over risk-neutral probabilities* ([3]): This approach can be utilized, for example, to determine the minimum and maximum values of futures compatible with option prices. Unlike the classical call-put parity regression used to determine future prices, this method provides a range of values, quantifying uncertainty.

CVXPY allows for quick and easy prototyping of CVI. However, for enhanced performance, consider bypassing CVXPY and interact directly with the QP solver's API. Although the canonicalization process in CVXPY is efficient, it introduces a non-negligible time overhead. After formulating the problem in CVXPY, the canonicalization process can be reverse-engineered to tailor the specific CVI problem, enabling direct formulation using lower-level programming languages such as C/C++ or Rust.

4.2 Clarabel

Alongside Stanford, the University of Oxford has made a strong impact in the advancement of open-source convex optimization software. Both universities were behind the introduction of OSQP [14] in 2017, a quadratic programming solver particularly efficient for large problems.

In 2022, the Oxford Control Group released Clarabel, an interior point solver capable of handling a variety of problems, including quadratic programming problems with linear constraints. Clarabel is the state-of-the-art solver for a wide range of problems including sparse QPs. Clarabel is faster and more robust than competing commercial and open-source solvers. See [9] for the methodology and benchmark results. Clarabel quickly gained widespread adoption and became the default solver in CVXPY in May 2024.

For the CVI problem, we found that Clarabel is several times faster than OSQP and also exhibits better convergence. While OSQP may not fully converge in rare instances, Clarabel consistently reaches the optimal solution.

4.2.1 Calibration Time

In this section, we will examine, with some practical examples, how long it takes to perform a CVI calibration with Clarabel. The calibration times presented below were obtained using an AMD Ryzen 9 5900HX CPU.

We remind the reader that a CVI calibration consists of several convex optimizations (typically two are sufficient):

- The first convex optimization without no-butterfly-arbitrage constraints.
- One (or more) optimization(s) with no-butterfly-arbitrage constraints linearized at the previous iteration.

The first optimization is generally 10%-30% faster than the subsequent ones as it involves fewer constraints, hence a smaller A matrix.

Table 1 covers three underlyings—Bitcoin (BTC), EURO STOXX 50 (SX5E), and S&P 500 (SPX)—across various numbers of parameters per expiry (5, 10, 15, 20, 30, 40, 80). These results were generated with the following assumption regarding the discretization of the constraints:

- The positivity of the variance (Section 3.3.1), the no-calendar-spread-arbitrage constraints (Equation 3), and the no-butterfly-arbitrage linearized constraints (Section 3.3.4) are applied across twenty 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 slower calibration.

We can read in Table 1 the following results:

- With 20 knots per expiry, the two Clarabel optimizations take 0.03 seconds for the BTC volatility surface calibration (calibrated from 950 vol bids or asks, across 12 expiries).
- With 20 knots per expiry, the two Clarabel optimizations take 0.06 seconds for the EURO-STOXX 50 volatility surface calibration (calibrated from 3,153 vol bids or asks, across 19 expiries).
- With 20 knots per expiry, the two Clarabel optimizations take 0.15 seconds for the S&P 500 volatility surface calibration (calibrated from 14,538 vol bids or asks, across 19 expiries).

Most underlyings require fewer than 10 parameters; however, the most liquid underlyings may require 15 or more. Calibration times are also reported for much larger numbers of parameters (up to 80), to illustrate how calibration time scales with the number of parameters.

While using a higher number of knots is not detrimental to the quality of the fit, it increases calibration time. Conversely, using too few knots effectively acts as a stronger regularization, potentially smoothing out genuine market features.

Scaling The number of non-zero coefficients in the sparse matrix A increases linearly with the number of parameters per expiry, whereas the growth is quadratic for the number of non-zero coefficients in P . 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.

Scaling times have not been documented for Clarabel, but in the case of OSQP, the execution time has been reported to scale with the total number of non-zero elements in matrices P and A (see [13]).

Table 1: Volatility surface calibration time with Clarabel

underlying_name	n_knots	P_non_zero_coeffs	A_non_zero_coeffs			calc_time		n_constraints	n_price_bid_or_ask	n_vol_bid_or_ask	n_expiries
		1st Iter.	2nd Iter.	1st Iter.	2nd Iter.	Total	1st Iter.	2nd Iter.	-	-	-
BTC	5	262	3,741	5,037	0.003	0.004	0.007	334	598	1,770	950
	10	396	5,175	6,843	0.006	0.008	0.014	-	-	-	-
	15	559	7,032	9,204	0.010	0.012	0.022	-	-	-	-
	20	779	8,741	11,345	0.012	0.016	0.028	-	-	-	-
	30	1,748	15,566	20,690	0.021	0.031	0.052	-	-	-	-
	40	2,725	20,273	26,886	0.028	0.041	0.069	-	-	-	-
	80	8,688	38,762	51,279	0.063	0.092	0.155	-	-	-	-
	SX5E	769	9,168	11,220	0.011	0.015	0.026	530	948	7,797	3,153
SPX	10	1,046	11,783	14,424	0.015	0.019	0.034	-	-	-	-
	15	1,361	15,267	18,706	0.021	0.025	0.046	-	-	-	-
	20	1,766	18,342	22,465	0.029	0.032	0.061	-	-	-	-
	30	3,799	32,077	40,190	0.046	0.067	0.113	-	-	-	-
	40	5,836	41,237	51,710	0.064	0.091	0.155	-	-	-	-
	80	18,601	76,957	96,778	0.163	0.208	0.371	-	-	-	-
	SPX	1,802	19,053	24,021	0.027	0.036	0.063	1,286	2,298	30,365	14,538
	10	2,905	25,764	32,158	0.035	0.043	0.078	-	-	-	-
21	15	4,709	34,584	42,910	0.050	0.063	0.113	-	-	-	-
	20	6,706	42,701	52,683	0.070	0.082	0.152	-	-	-	-
	30	18,547	74,754	94,396	0.123	0.168	0.291	-	-	-	-
	40	30,717	97,055	122,416	0.197	0.271	0.467	-	-	-	-
	80	110,733	184,263	232,256	0.523	0.678	1.200	-	-	-	-

n_{knots}: Number of cubic spline knots; this is also the number of free parameters per expiry.

P_non_zero_coefficients: Number of non-zero coefficients in the *P* sparse matrix (quadratic objective).

A_non_zero_coefficients: Number of non-zero coefficients in the *A* sparse matrix (linear constraints).

calc_time: Solving time (time spent inside the Clarabel solver) in seconds.

n_constraints: Number of linear constraints (including, notably, no-calendar-spread-arbitrage constraints, and in the case of the second iteration, the no-butterfly-arbitrage constraints).

n_price_bid_or_ask: Number of bid and ask prices, for calls and puts. Note that the convex optimization only takes into account vol bids and asks, meaning that it is only given the best bid and best ask vol across calls and puts. For example, in the calibration of the SPX volatility surface, CVI is provided with a total of 14,538 market volatility bids or asks.

n.vol_bid_and_ask: Number of volatility quotes (bid or ask).

n_expiries: Number of listed expiries.

5 Conclusion

This paper introduced CVI, a new volatility fitter designed to overcome the limitations of traditional models. CVI emerges as a solution to the volatility surface fitting problem when focusing on a numerically sound calibration. Framing the volatility surface calibration problem as a convex optimization guarantees speed and robustness.

After reviewing the non-arbitrage conditions, we chose to parameterize the volatility surface in variance space. Dual parameterization is one of the key ideas of CVI, linearly mapping basis function weights to more intuitive parameters that characterize the shape of the volatility smile. This approach allows the optimization to seamlessly handle variables within both cubic spline and B-spline spaces. Beyond the at-the-money variance $v(z = 0)$, the smile is represented by dimensionless parameters: the at-the-money skew $s(z = 0)$ and the convexities $\{c(z_i)\}_{0 \leq i \leq n-1}$.

QP Problem CVI expresses the volatility surface calibration as a quadratic programming problem with linear constraints. CVI goes beyond the traditional mid-price fitting approach by taking into account bids and asks, even when a mid does not exist. The objective function includes several terms: a least squares component, other quadratic terms to penalize a fit beyond the market quotes, and a regularization term to balance overfitting and underfitting. We carefully weight the different terms to ensure consistent scaling across all underlyings, regardless of the number of listed strikes and expiries, the level of the volatility, the particular shape of the smile or the number of knots. The only hyperparameter is the regularization factor λ , which should require minimal tuning, if any. The discretization of the problem involves selecting the knots of the cubic spline and the normalized log-moneyness points at which to apply non-arbitrage constraints.

While CVI ensures an arbitrage-free calibration in strike and time, it does not assume that the market quotes themselves are free from arbitrage. Applying no calendar spread arbitrage is straightforward in variance space. Enforcing no butterfly arbitrage is not trivial because it is not convex in that space, but linearizing the constraints is a practical workaround. This paper derives the no-butterfly-arbitrage conditions and their linearization in the CVI parameters' space. We also provide a specific condition for the maximum skew at the edge cubic spline knots z_0 and z_{n-1} , where the convexity vanishes. Lee's tail slope bounds also offers a relation in the tails where the variance is assumed to be a linear function of the log strike.

Visualization The CVI calibration is easy to comprehend when represented visually. In this paper, we have strived to illustrate this visual aspect by plotting several key elements:

- the volatility fit along its calendar spread channel, bid and ask error bars (with the volatility implied from the mid-market price), and bids and asks (useful in case there is no error bars, i.e. bid or ask is not available), alongside the probability density function on the right axis. This provides a compact view to assess the fit quality and the presence of arbitrage both in strike and time.
- the convexity curve $z \rightarrow c(z)$, a simple way to evaluate the smoothness of the cubic spline. The CVI calibration balances the L_1 norm of this curve against a chi-square of the fit.
- The total variance curves, a classical way of visualizing calendar spread arbitrage across the surface at a glance.

Plotting these curves provides a valuable way to analyze any volatility fitter, whether CVI or another, by showing the quality of the fit, the eventual presence of arbitrage, and the extent of overfitting or underfitting.

CVXPY One of the greatest strengths of convex optimization is its ecosystem. Thanks to high-level domain-specific languages (DSLs), such as CVXPY, practitioners can implement the CVI calibration without specific knowledge of the mathematics underlying convex optimization.

The CVI calibration can be written in a human-readable way as a problem whose variables are the cubic spline parameters. The DSL then transforms the problem into a lower level form suitable for the solver. The canonicalization involves, notably, two sparse matrices, P and A , which typically contain thousands or tens of thousands of non-zero coefficients. These encode, respectively, the market prices and the no-arbitrage constraints.

While a DSL facilitates quick prototyping of CVI, it can be bypassed at a later stage to improve performance by eliminating the DSL's overhead.

Clarabel Open-source solvers have reached an impressive level of stability and speed, even for large problems. Developed by the Oxford Control Group, Clarabel can solve CVI's quadratic programming problems remarkably fast. For illiquid underlyings, such as Bitcoin, Clarabel can fit an arbitrage-free volatility surface in just a few hundredths of a second. For liquid underlyings, like the S&P 500, the process takes a few tenths of a second.

In the S&P 500 volatility surface example provided, the calibration time merely doubles from 0.078 seconds to 0.152 seconds when increasing the number of parameters per expiry from 10 to 20, a linear relationship. As the number of parameters gets really large, the calibration time increases quadratically. This is in sharp contrast with non-convex optimization problems, which typically become exponentially more challenging as the number of parameters increases.

Future research could pursue several promising directions. Although the linearization process has shown remarkable effectiveness, further investigation is warranted, particularly to establish rigorous conditions for its convergence and robustness across diverse market scenarios. Additionally, using higher-order B-splines would enhance the accuracy of Greeks calculations, notably for exotic derivatives. Finally, the methodology

could be extended to incorporate multiple snapshots and add a temporal regularization of the volatility surface shape.

References

- [1] Andreassen J., Huge B., Volatility Interpolation
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1694972
- [2] Baker M., Gillberg T., Thomas S., Trading events: Smiles, frowns and moustaches—the many faces of the options market
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3263368
- [3] Barratt S., Tuck J., Boyd S., Convex optimization over risk-neutral probabilities
https://stanford.edu/~boyd/papers/pdf/cvx_opt_risk_neutral.pdf
- [4] Boyd S., Johansson K., Kahn R., Schiele P., Schmelzer T., Markowitz Portfolio Construction at Seventy
<https://web.stanford.edu/~boyd/papers/pdf/markowitz.pdf>
- [5] Burkowska O., Gaf M., Glau K., Mahlstedt M., Schoutens W., Wohlmuth B., Calibration to American Options: Numerical Investigation of the de-Americanization
<https://arxiv.org/abs/1611.06181>
- [6] Diamond S., Boyd S. CVXPY: A Python-embedded modeling language for convex optimization
https://web.stanford.edu/~boyd/papers/pdf/cvxpy_paper.pdf
- [7] Fengler M., Arbitrage-Free Smoothing of the Implied Volatility Surface
<https://core.ac.uk/download/pdf/6978470.pdf>
- [8] Gatheral J., A parsimonious arbitrage-free implied volatility parameterization with application to the valuation of volatility derivatives
- [9] Goulart P., Chen Y., Clarabel: An interior-point solver for conic programs with quadratic objectives
<https://arxiv.org/pdf/2405.12762>
- [10] Kahalé N., An arbitrage-free interpolation of volatilities
https://www.researchgate.net/publication/228872089_An_Arbitrage-free_Interpolation_of_Volatilities
- [11] Klassen T., Arbitrage-Free Parametric Volatility Surfaces and Real-Time Fitting
https://voladynamics.com/pdf/Klassen_GD_Chicago_2017.pdf
- [12] Lee R., The Moment Formula for Implied Volatility at Extreme Strikes
<http://math.uchicago.edu/~rl/moment.pdf>
- [13] Schubiger M., Banjac G., Lygeros J. GPU acceleration of ADMM for large-scale quadratic programming
<https://www.sciencedirect.com/science/article/pii/S0743731520303063?via%3Dihub>
- [14] Stellato B., Banjac G., Goulart P., Bemporad A., Boyd S. OSQP: an operator splitting solver for quadratic programs
<https://web.stanford.edu/~boyd/papers/pdf/osqp.pdf>
- [15] Zeliade Systems, Quasi-explicit calibration of Gatheral's SVI model
<https://www.zeliade.com/wp-content/uploads/whitepapers/zwp-0005-SVICalibration.pdf>

A Non-Arbitrage Conditions

To prevent static arbitrage, we should enforce the absence of strike and calendar spread arbitrage. Let us examine these conditions.

A.1 No Strike Arbitrage

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) behaviour via Lee (2004) [12]. In our setting, variance is linearly extrapolated in log-strike, so in the right tail we impose:

$$\boxed{\frac{\partial v}{\partial k} < \frac{2}{T}} \quad (7)$$

We use a *strict* inequality to stay away from the boundary case where the tail slope equals 2.

Expressed in terms of s , this becomes:

$$s < \frac{2}{\sigma_* \sqrt{T}}$$

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

$$\boxed{\frac{\partial v}{\partial k} > -\frac{2}{T}} \quad (8)$$

The previous equation can be written in terms of s :

$$s > -\frac{2}{\sigma_* \sqrt{T}}$$

Butterfly arbitrage The probability density function can be expressed as $e^{rT} \frac{d^2 C}{dK^2}$ (Breeden-Litzenberger formula) and can be rewritten as:

$$\phi(d_2) \left(\frac{1}{K\sigma\sqrt{T}} + 2 \frac{d_1}{\sigma} \frac{\partial \sigma}{\partial K} + \frac{d_1 d_2 K \sqrt{T}}{\sigma} \left(\frac{\partial \sigma}{\partial K} \right)^2 + K \sqrt{T} \frac{\partial^2 \sigma}{\partial K^2} \right)$$

where ϕ is the standard normal probability density function and where $d_{1,2} = \frac{-k \pm \frac{\sigma^2 T}{2}}{\sigma\sqrt{T}}$.

Butterfly arbitrage arises when the probability density function (pdf) becomes negative. To prevent such arbitrage, the pdf must be non-negative. This requirement can be expressed as follows:

$$\boxed{\phi(d_2) \left(\frac{1}{K\sigma\sqrt{T}} + 2 \frac{d_1}{\sigma} \frac{\partial \sigma}{\partial K} + \frac{d_1 d_2 K \sqrt{T}}{\sigma} \left(\frac{\partial \sigma}{\partial K} \right)^2 + K \sqrt{T} \frac{\partial^2 \sigma}{\partial K^2} \right) \geq 0} \quad (9)$$

A.2 No Calendar Spread Arbitrage

Assuming deterministic interest rates, to prevent calendar spread arbitrage, the total variance, defined for a fixed strike-to-forward ratio, must be a monotonically increasing function of the time to expiry T :

$$\boxed{v(K, T)T \leq v(K \frac{F_{T+dt}}{F_T}, T + dt)(T + dt)}$$

Another way to express this condition is through graphical representation. If we plot the total variance curves² for various maturities as a function of the strike-to-forward ratio K/F , these curves should not intersect. This requirement ensures that for any given strike-to-forward ratio, the total variance increases with the maturity of the option, thereby preventing potential calendar spread arbitrage opportunities. This concept is visually illustrated by the total variance curves (see Figure 10) and by the calendar spread channel (see Figures 7, 11, 12, and 14).

B No-Butterfly-Arbitrage Constraints

In Appendix A.1, we have derived the equation for the probability density function. Here, we consider how to use this equation in practice by considering two different cases:

- For $z_0 < z < z_{n-1}$, we express the positivity of the pdf using s and c in Appendix B.1.
- The cases z_0 and z_{n-1} represent boundary conditions of the cubic spline, where the second derivative, $\frac{\partial^2 v}{\partial z^2}$, is set to zero. This specific boundary scenario is explored in more detail in Appendix B.2.

Note that we do not need to directly enforce any constraints for $z < z_0$ and $z > z_{n-1}$ (see B.2.3).

²Here, by total variance, we refer to vT .

B.1 PDF ≥ 0 between z_0 and z_{n-1}

For $z_0 < z < z_{n-1}$, the convexity is free to move. To ensure a positive pdf, the convexity must exceed a certain threshold that depends on both the volatility and the skew.

Using Equation 9, we can rewrite the positivity of the pdf as:

$$\frac{\partial^2 \sigma}{\partial K^2} \geq -\frac{1}{K\sqrt{T}} \left(\frac{1}{K\sigma\sqrt{T}} + 2\frac{d_1}{\sigma} \frac{\partial \sigma}{\partial K} + \frac{d_1 d_2 K \sqrt{T}}{\sigma} \left(\frac{\partial \sigma}{\partial K} \right)^2 \right) \quad (10)$$

Let us introduce the normalized convexity c :

$$\begin{aligned} c &:= \frac{1}{\sigma_*^2} \frac{\partial^2 v}{\partial z^2} \\ &= \frac{2}{\sigma_*^2} \left(\left(\frac{\partial \sigma}{\partial z} \right)^2 + \sigma \frac{\partial^2 \sigma}{\partial z^2} \right) \end{aligned}$$

where σ_* is the anchor ATM vol and $z = \frac{\log K/F}{\sigma_* \sqrt{T}}$

Let us introduce the normalized skew s :

$$\begin{aligned} s &:= \frac{1}{\sigma_*^2} \frac{\partial v}{\partial z} \\ &= \frac{2}{\sigma_*^2} \sigma \frac{\partial \sigma}{\partial z} \end{aligned}$$

It is helpful to consider the following change of variables:

$$\begin{aligned} \frac{\partial^2 \sigma}{\partial z^2} &= \sigma_*^2 T K \left(\frac{\partial \sigma}{\partial K} + K \frac{\partial^2 \sigma}{\partial K^2} \right) \\ &= \frac{\sigma_*^3 \sqrt{T}}{2\sigma} s + \sigma_*^2 T K^2 \frac{\partial^2 \sigma}{\partial K^2} \\ \frac{\partial \sigma}{\partial z} &= \sigma_* \sqrt{T} K \frac{\partial \sigma}{\partial K} \\ &= \frac{\sigma_*^2}{2\sigma} s \end{aligned}$$

$\frac{\partial \sigma}{\partial K}$ can be rewritten as a function of s :

$$\frac{\partial \sigma}{\partial K} = \frac{\sigma_*}{2\sigma \sqrt{T} K} s \quad (11)$$

$\frac{\partial^2 \sigma}{\partial K^2}$ can be rewritten as of function of s and c :

$$\begin{aligned} \frac{\partial^2 \sigma}{\partial K^2} &= \left(\frac{\partial^2 \sigma}{\partial z^2} - \frac{\sigma_*^3 \sqrt{T}}{2\sigma} s \right) \frac{1}{\sigma_*^2 T K^2} \\ &= \frac{c}{2\sigma T K^2} - \frac{\sigma_*^2}{4\sigma^3} \frac{s^2}{T K^2} - \frac{\sigma_*}{2\sigma \sqrt{T} K^2} s \end{aligned} \quad (12)$$

Equations 10, 11, and 12 can be combined to express a relation between σ , s , and c :

$$c \geq \frac{\sigma_*^2}{2v} s^2 + \sigma_* \sqrt{T} s - 2 \left(1 + d_1 \frac{\sigma_*}{\sigma} s + d_1 d_2 \left(\frac{\sigma_*}{2\sigma} s \right)^2 \right)$$

The right-hand side of that equation represents c_{\min} , the minimum convexity c required to prevent a negative probability density function.

If we expand d_1 and d_2 , which are functions of σ , we obtain:

$$c \geq \frac{\sigma_*^2}{2\sigma^2} s^2 + \sigma_* \sqrt{T} s - 2 \left(1 + \frac{-k + \frac{\sigma^2 T}{2}}{\sigma \sqrt{T}} \frac{\sigma_*}{\sigma} s + \left(-k + \frac{\sigma^2 T}{2} \right) \left(-k - \frac{\sigma^2 T}{2} \right) s^2 \frac{\sigma_*^2}{4\sigma^4 T} \right) \quad (13)$$

B.1.1 Special Case Studies

As a side note, we look at how Equation 13 can be rewritten in some special cases.

At-The-Money Case We observe that at-the-money ($k = 0$), Equation 13 simplifies neatly to:

$$c \geq -2 + \frac{\sigma_*^2 s^2}{8} \left(T + \frac{4}{\sigma^2} \right) \quad (14)$$

From Equation 14, several insights emerge:

- the at-the-money skew increases the at-the-money bound on convexity
- The at-the-money convexity has a theoretical lower bound of -2 , which is attained only in the absence of ATM skew ($s = 0$).

$\sigma^2 T \ll |k|$ In the limit $\sigma^2 T \ll k$, Equation 13 simplifies to:

$$c \geq -2 + \left(\sigma_* \sqrt{T} + \frac{2k\sigma_*}{\sigma^2 \sqrt{T}} \right) s - \frac{k^2 \sigma_*^2}{2\sigma^4 T} s^2 \quad (15)$$

We can observe that the RHS of Equation 15 is a quadratic function of the dimensionless number $\frac{k\sigma_* s}{\sigma^2 \sqrt{T}}$.

B.2 PDF ≥ 0 at the edge knots and beyond

At the edge knots (z_0 and z_{n-1}) and beyond, the convexity is set to zero, i.e., $\frac{\partial^2 v}{\partial z^2} = 0$. Using Equation 12, that is equivalent to:

$$\frac{\partial^2 \sigma}{\partial K^2} = -\frac{1}{\sigma} \left(\frac{\partial \sigma}{\partial K} \right)^2 - \frac{1}{K} \frac{\partial \sigma}{\partial K} \quad (16)$$

By combining Equations 10 and 16, the absence of butterfly arbitrage is:

$$\frac{(d_1 d_2 - 1) K \sqrt{T}}{\sigma} \left(\frac{\partial \sigma}{\partial K} \right)^2 + \left(\frac{2d_1}{\sigma} - \sqrt{T} \right) \frac{\partial \sigma}{\partial K} + \frac{1}{K \sigma \sqrt{T}} \geq 0$$

This is a quadratic equation in $\frac{\partial \sigma}{\partial K}$. Its discriminant Δ is given by:

$$\Delta = \frac{-4d_1 d_2 + (2d_1 - \sqrt{T})^2 + 4}{v}$$

Using the relationship $d_2 = d_1 - \sigma \sqrt{T}$, that can be rewritten as:

$$\Delta = T + \frac{4}{v}$$

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

$$\frac{-d_1 + \frac{\sqrt{vT}}{2} \pm \sqrt{1 + \frac{vT}{4}}}{(d_1 d_2 - 1) K \sqrt{T}} \quad (17)$$

We are assuming that z_0 and z_{n-1} have been chosen far enough in 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, it is crucial to ensure that the skew $\frac{\partial \sigma}{\partial K}$ does not fall between the two roots to avoid arbitrage.

Furthermore, referring to Equation 17, we observe that the two roots are negative for z_0 because the sign of the roots follows that of $-d_1$, as this term dominates the numerator. For the same reason, the roots are positive for z_{n-1} .

B.2.1 Minimum s in the left tail

For z_0 , the two roots are negative. We have established that the skew has to be below the lower root or above the higher root:

$$\frac{\partial \sigma}{\partial K} \leq \frac{-d_1 + \frac{\sqrt{vT}}{2} - \sqrt{1 + \frac{vT}{4}}}{(d_1 d_2 - 1) K \sqrt{T}} \text{ or } \frac{\partial \sigma}{\partial K} \geq \frac{-d_1 + \frac{\sqrt{vT}}{2} + \sqrt{1 + \frac{vT}{4}}}{(d_1 d_2 - 1) K \sqrt{T}}$$

The higher root – at the limit of arbitrage – already implies a very steep skew. Thus, $\frac{\partial \sigma}{\partial K}$ should be higher than the higher root:

$$\frac{\partial \sigma}{\partial K} \geq \frac{-d_1 + \frac{\sqrt{vT}}{2} + \sqrt{1 + \frac{vT}{4}}}{(d_1 d_2 - 1) K \sqrt{T}}$$

In terms of s , this can be rewritten as:

$$s \geq \frac{2v\sqrt{T} \left(k + \sqrt{vT} \sqrt{1 + \frac{vT}{4}} \right)}{\sigma_* \left(k^2 - \frac{v^2 T^2}{4} - vT \right)}. \quad (18)$$

B.2.2 Maximum s in the right tail

For z_{n-1} , the two roots are positive. The high root implies an unrealistically high skew - given that it is higher than the lower root, which is already at the limit of arbitrage. Hence, we need to have a skew lower than the lower root, i.e., the skew cannot be too steep:

$$\frac{\partial \sigma}{\partial K} \leq \frac{-d_1 + \frac{\sqrt{vT}}{2} - \sqrt{1 + \frac{vT}{4}}}{(d_1 d_2 - 1) K \sqrt{T}}$$

That can be written in term of s :

$$s \leq \frac{2v\sqrt{T}\left(k - \sqrt{vT} \sqrt{1 + \frac{vT}{4}}\right)}{\sigma_\star \left(k^2 - \frac{v^2 T^2}{4} - vT\right)} \quad (19)$$

B.2.3 Beyond the edge knots

We only need to enforce Equation 18 (resp. 19) at z_0 (resp. 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 RHS of Equation 19 reaches a positive constant asymptotically from *below*, monotonically increasing as $k \rightarrow \infty$, whereas the RHS of Equation 18 reaches a negative constant asymptotically from *above*, monotonically decreasing as $k \rightarrow -\infty$.

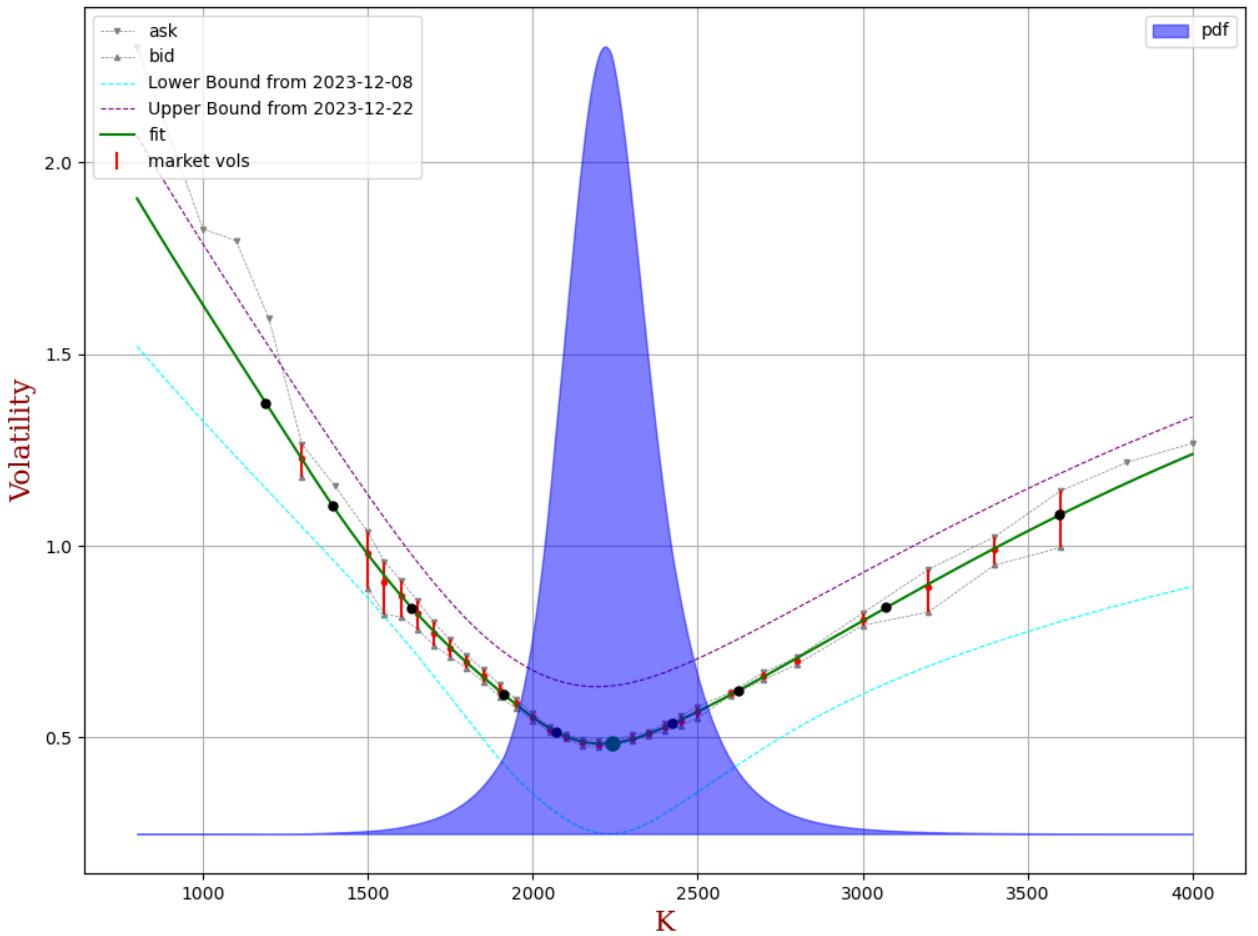


Figure 14: ETH December 15th, 2023 expiry as of December 5th, 2023. The bid-asks are indicated in red and grey (red bars are only available for strikes with both a bid and an ask). The fit is shown in green. The cyan and purple curves represent the lower and upper boundaries, respectively, from the nearest expiries via the no-calendar-spread-arbitrage constraints. The blue-shaded curve on the right axis is the probability density function (implied by the fitted volatility). Its positivity is enforced by the linearized no-butterfly-arbitrage constraints shown in Table 2.

C Example of Linearized No-Butterfly-Arbitrage Constraints

Here we give a numerical example of the linearized no-butterfly-arbitrage constraints using ETH December 15th 2023 expiry as of December 5th 2023. On that example $T = 0.0265$, $\sigma_* = 0.4853$ and we have chosen $z_0 = -8$ and $z_{n-1} = 8$.

	strike	$\sigma = \sqrt{v}$	s	c	no-butterfly-arbitrage PDF constraint
$z=-8.0$	1190.37	1.3725	-1.4752	0	$s \geq -1.48709 - 0.67787 \cdot (v - 1.90134)$
$z=-6.0$	1394.10	1.1036	-1.2914	0.183802	$c \geq 0.0543 - 0.80983 \cdot (s + 1.31035) - 0.73423 \cdot (v - 1.21501)$
$z=-4.0$	1632.69	0.8386	-0.8794	0.228180	$c \geq -0.21948 - 1.41118 \cdot (s + 0.86944) - 1.56605 \cdot (v - 0.70172)$
$z=-2.0$	1912.12	0.6124	-0.5600	0.091217	$c \geq -0.74223 - 1.97955 \cdot (s + 0.5604) - 2.6916 \cdot (v - 0.37537)$
$z=-1.0$	2069.29	0.5153	-0.3212	0.386380	$c \geq -1.42302 - 1.80625 \cdot (s + 0.32237) - 2.01845 \cdot (v - 0.26557)$
$z=0.0$	2239.38	0.4851	0.0652	0.386381	$c \geq -1.99788 + 0.06527 \cdot (s - 0.06511) - 0.00902 \cdot (v - 0.23528)$
$z=1.0$	2423.44	0.5362	0.3410	0.165272	$c \geq -1.43207 + 1.68919 \cdot (s - 0.34138) - 1.83909 \cdot (v - 0.28751)$
$z=2.0$	2622.64	0.6223	0.5063	0.165268	$c \geq -0.87986 + 1.99175 \cdot (s - 0.50644) - 2.40194 \cdot (v - 0.38733)$
$z=4.0$	3071.49	0.8388	0.8368	0.165267	$c \geq -0.26959 + 1.45946 \cdot (s - 0.83656) - 1.56731 \cdot (v - 0.70358)$
$z=6.0$	3597.17	1.0800	1.1089	0.106831	$c \geq -0.09056 + 1.02123 \cdot (s - 1.10877) - 0.86286 \cdot (v - 1.16621)$
$z=8.0$	4212.81	1.3123	1.2158	0	$s \leq 1.3644 + 0.69068 \cdot (v - 1.72203)$

Table 2: No butterfly arb PDF constraints (linearization of the Equations 18 ($z = -8$), 19 ($z = 8$) and 13 ($-8 < z < 8$). The linearization of Equation 13 is given by Equation 5.