Calibration & Optimization

Goals:

- Review the process for extracting implied volatilites from a set of market prices.
- Introduce optimization methods
- Discuss common calibration techniques
- Identify the challenges in calibrating volatility surfaces in real-world problems.

Hirsa, Chapter 7 is a very good reference for this material.

Review: Black-Scholes Formula

- Recall that the **Black-Scholes formula** states that the price of a European call can be written as:

$$c_0 = \tilde{\mathbb{E}}\left[e^{-rT}\left(S_T - K\right)_+\right] = \Phi(d_1)S_0 - \Phi(d_2)Ke^{-rT} \quad (1)$$

- Here, Φ is the CDF of the standard normal distribution and

$$\begin{aligned} d_1 &= \frac{1}{\sigma\sqrt{T}}\left(\ln\frac{S_0}{K} + \left(r + \frac{\sigma^2}{2}\right)T\right), \\ d_2 &= d_1 - \sigma\sqrt{T} \end{aligned} \quad (2)$$

- Recall also that this formula is the result of an underlying asset that is governed by **Geometric Brownian Motion**.
- In reality, few markets trade in accordance with these dynamics.
- To more accurately reflect market dynamics, we need to rely on more complex models, many of which you have already seen.

Review: Uses of Black-Scholes Formula

- Despite its simplicity and inability to produce realistic market dynamics, the Black-Scholes model is widely used throughout the industry as a quoting convention among traders.
- In particular, comparing options with different strikes and expiries in terms of price is not particularly meaningful. Discussing them in terms of their underlying volatility parameter, σ, however, normalizes them and provides a more intuitive comparison.
- Additionally, Black-Scholes Greeks are relied upon by options traders in order to provide an approximation of their books' exposures.

Review: Implied Volatility

- If we know the parameters of the Black-Scholes model r and σ, then we can apply the Black-Scholes formula to obtain the price of a European call or put.
- Of course, markets don't provide us with the σ but instead provides us with a bid and ask price.
- We generally replace this with a mid price in most modelling exercises, although this is certainly not required

$$\tilde{c_0} = \frac{c_0(\text{bid}) + c_0(\text{ask})}{2} \tag{3}$$

- Given this price $\tilde{c_0}$, we need to find the corresponding volatility.

Review: Implied Volatility

- Specifically, for each strike we look for the $\sigma_{\text{impl}}(K)$ that solves the following equation:

$$\left(\hat{c}(\tau, K, \sigma) - c_{\tau,K}\right)^2 < \epsilon \tag{4}$$

  where c is the market price, $\hat{c}$ is the model price and ϵ is the tolerance for our solution.

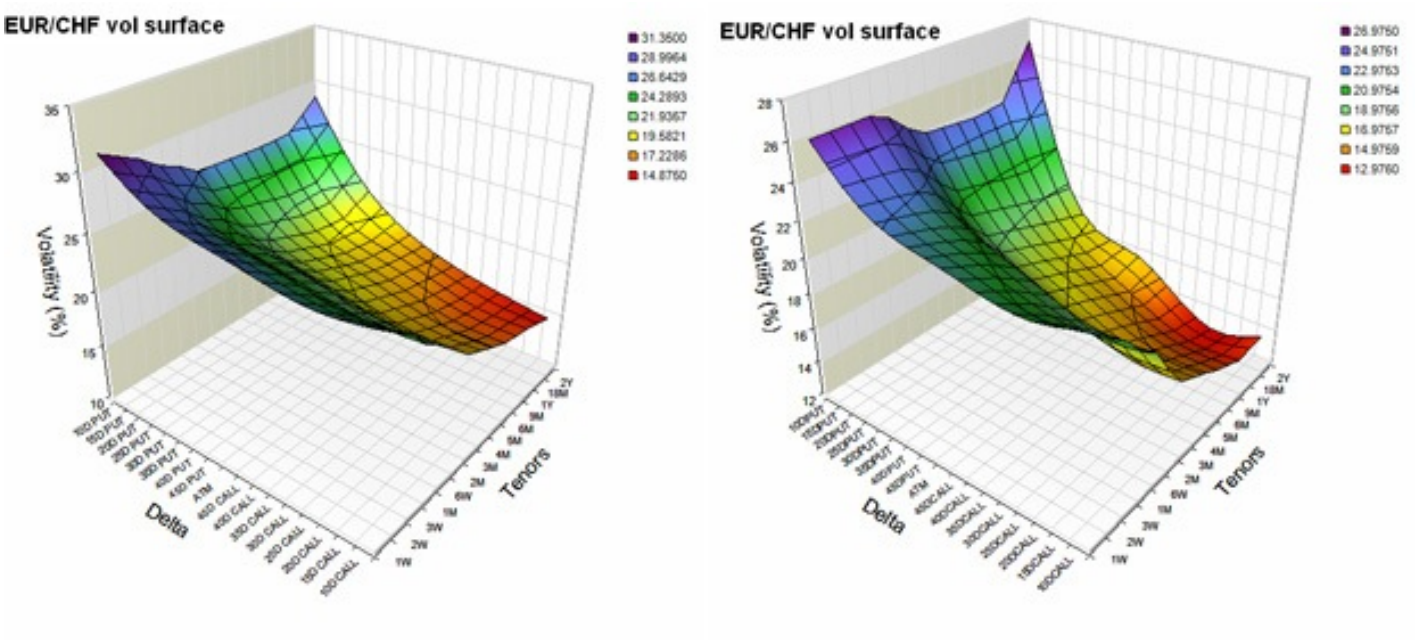
- To extract an **implied volatility**, we need to solve a one-dimensional optimization problem.
- In R, we can use the *uniroot* function to do this.
- This is the simplest **calibration** problem that we face as quants.

Review: Implied Volatility Skew

- In practice we often take a set of prices and extract a corresponding set of implied volatilities.
- The inputs for this process may be market or model prices.
- Note that this does not provide us with a single consistent set of dynamics for the underlying asset.

Review: Implied Volatility Surface

- In reality the implied volatility function for a given asset is a function of both strike and expiry, and can be respresented via a 3D plot.



The left chart shows the implied vol calibrated on August 22, 2011. The right chart shows the implied vol calibrated on September 6, 2011. *Source:* Reuters

Calibration of Volatility Surfaces

- We are often interested in finding a set of model parameters that provide the best fit to observed market data.
- Note that although an implied volatility surface is a useful visual tool, and helps us compare options across strike and expiry, *it does not provide a consistent set of dynamics for the underlying asset.*
- We can define our set of best model parameters by minimizing the squared pricing error:

$$\vec{p}_{\min} = \underset{\vec{p}}{\operatorname{argmin}} \left\{ \sum_{\tau,K} \left(\hat{c}(\tau, K, \vec{p}) - c_{\tau,K} \right)^2 \right\} \tag{5}$$

  where c and $\hat{c}$ are the market and model price for a given option.
- For example, if we were to use the Heston model, the parameters would be: κ, θ, ρ, σ and ν_0.

Calibration of Volatility Surfaces

- In order to solve equations of the form of (5) requires us to be able to solve **optimization problems**.
- In equation (5) we have defined the best fit to be the one with minimal squared pricing error, however, we could choose any function we like.
- The dimensionality of the optimization depends on the stochastic process that we choose. In the case of Heston, we have a 5D optimization problem.
- Today we will learn how to formulate and solve these types of optimization problems.

Where do optimization problems arise in finance?

- Calibration: Finding an optimal set of parameters to fit a volatility surface
- Estimation: Finding the best fit line to forecast an asset price (e.g., via regression)
- Asset Allocation: Constructing an optimal portfolio given a set of constraints
- Yield Curve Construction
- Risk Neutral Density Extraction

Optimization: Background & Terminology

- Optimization problems that we face will have the following form:

$$\max_x \left\{ f(x) \,\middle|\, g(x) \leq c \right\} \tag{6}$$

- We refer to $f(x)$ as the **objective function** in our optimization.
- We refer to $g(x) \leq c$ as the **constraints** in our optimization.
  - We generally differentiate between *equality* and *inequality* constraints
- Remember that maximizing a function $f(x)$ is the same as minimizing the function $-f(x)$.
- We refer to x^* as a maximizer of the function $f(x)$ if:
  $f(x) \leq f(x^*) \quad \forall x$

First & Second Order Conditions: Single Variable

- Necessary, but not sufficient conditions for x^* to be a global maximizer of $f(x)$ are:

$$\frac{\partial f(x^*)}{\partial x^*} = 0 \tag{7}$$

$$\frac{\partial^2 f(x^*)}{\partial x^{*2}} < 0 \tag{8}$$

- Similarly, necessary but not sufficient conditions for x^* to be a global minimizer of $f(x)$ are:

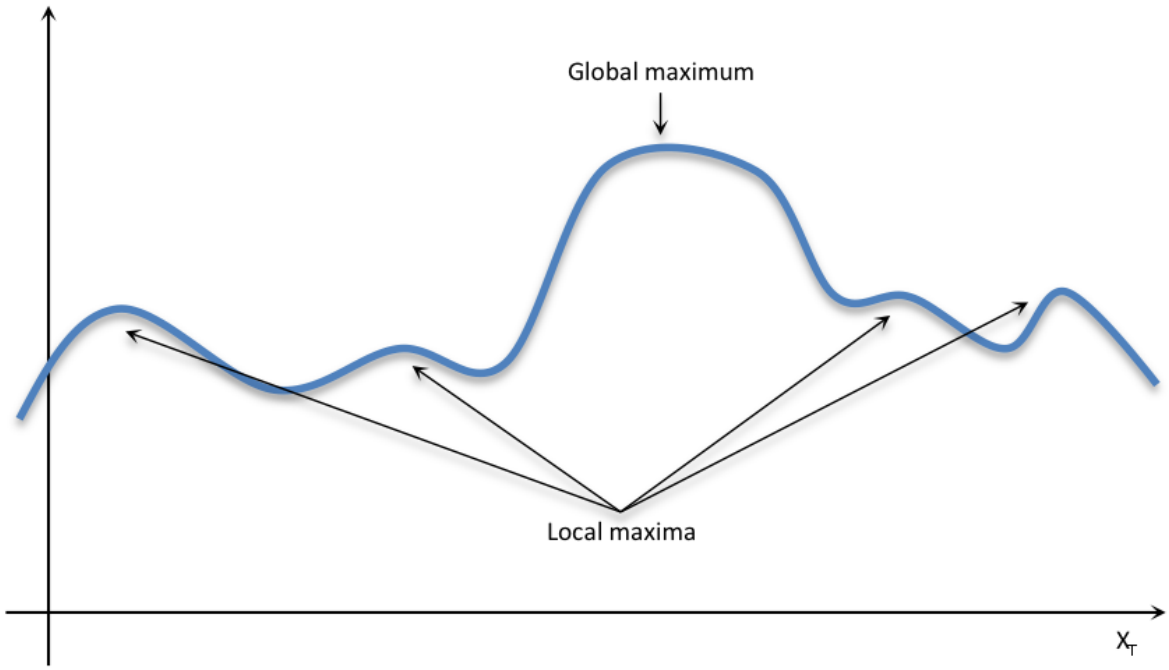
$$\frac{\partial f(x^*)}{\partial x^*} = 0 \tag{9}$$

$$\frac{\partial^2 f(x^*)}{\partial x^{*2}} > 0 \tag{10}$$

First & Second Order Conditions: Many Variables

- We refer to the first derivative of the objective function as the **Gradient** and the second derivative as the **Hessian**.
- In multi-dimensional problems with N parameters:
  - The Gradient will be a vector of length N.
  - The F.O.C. is: $\nabla f(x^*) = 0$
  - The Hessian will be an $N \times N$ matrix.
  - The S.O.C. is that the Hessian is *negative definite* for a maximum and that it is *positive definite* for a minimum.

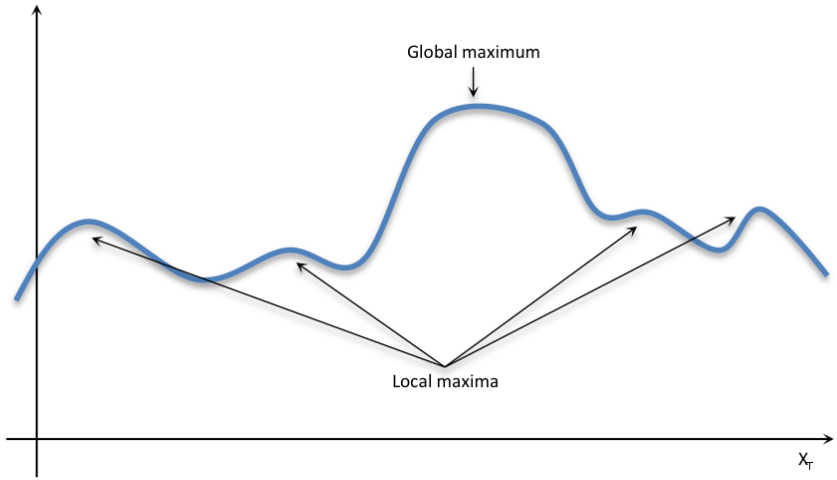
Local vs. Global Maxima & Minima



- We are interested in global maxima (minima) rather than local maxima (minima).
- Global maxima and minima ares inherently much more difficult to find than local maxima and minima.

Global or Local Maximum?

- A solution of the first and second-order conditions is only guaranteed to be a local maximum
- In order to make sure that we have actually reached a global maximum, we need to compare the value of objective function at all local optima
- The global maximum is the local maximum with the highest value



Convex Functions

- Under certain conditions, we can guarantee that a function will have one and only one local minimum or maximum.
- In particular this is true for **convex functions**, which have a positive second derivative at all points.
- For convex functions, if we find a local minimum then we know it is also a global minimum.
- As this greatly simplifies our optimization problem, it is recommended that we use **convex objective functions** in our minimization problems whenever possible.
- The same is true of concave functions and global maxima.

Unconstrained Optimization

- Let's start by walking through a very simple unconstrained minimization example.
- Consider the following problem:

$$\min_x \quad f(x) \tag{11}$$

$$f(x) = x^2 + 3x + 12 \tag{12}$$

- The F.O.C is: $\frac{\partial f(x)}{\partial x} = 2x + 3 = 0$.
- The S.O.C. is: $\frac{\partial^2 f(x)}{\partial x^2} = 2 > 0$.
- $x^* = -1.5$ is a minimum of f(x).
- Note that since the function is convex we can say that this local minimum is also a global minimum.

Lagrange Multipliers: Two Variables

- For constrained optimization problems we use Lagrange Multipliers.
- The key insight behind Lagrange Multipliers is that the function $f(x, y)$ is **tangent** to the constraint $g(x, y) = c$ at optimal points.
- Recall that two functions are tangent when their gradients are parallel. Therefore, we have:

$$\nabla f(x, y) = \lambda \nabla g(x, y) \tag{13}$$

when the constraint is $g(x, y) = c$.

Lagrange Multipliers: Two Variables

Lagrange re-wrote this as a single function, referred to as a Lagrangian, which incorporates the constraint. Note the extra variable λ:

$$\begin{aligned} \mathcal{L}(x, y, \lambda) &= f(x, y) - \lambda(g(x, y) - c) \\ \nabla\mathcal{L}(x, y, \lambda) &= \nabla f(x, y) - \lambda\nabla g(x, y) \\ \partial_\lambda\mathcal{L}(x, y, \lambda) &= -(g(x, y) - c) \end{aligned} \tag{14}$$

Constrained Optimization Example

- To see this, let's consider a 2D objective function with a single equality constraint:

$$\max_{x,y} \left\{ f(x,y) \,\middle|\, g(x,y) = c \right\} \tag{15}$$

where

$$\begin{aligned} f(x,y) &= xy^2 \\ g(x,y) &= 3x^2 + 2y^2 = 6 \end{aligned} \tag{16}$$

- To solve this problem, we can use **Lagrange Multipliers**. That is, we can re-write (16) in terms of the following Lagrangian:

$$\begin{aligned} \max_{x,y,\lambda} \quad & \mathcal{L}(x,y,\lambda) \\ \mathcal{L}(x,y,\lambda) &= xy^2 - \lambda\left(3x^2 + 2y^2 - 6\right) \end{aligned} \tag{17}$$

Constrained Optimization with Lagrange Multipliers

- Notice that we have transformed our problem into an unconstrained optimization.
- As a result, we can use the same procedure for finding a local maximum as we did in the unconstrained case.
- The First Order Conditions are:

$$\frac{\partial \mathcal{L}(x,y,\lambda)}{\partial x} = y^2 - 6x\lambda = 0 \quad (18)$$

$$\frac{\partial \mathcal{L}(x,y,\lambda)}{\partial y} = 2xy - 4y\lambda = 0 \quad (19)$$

$$\frac{\partial \mathcal{L}(x,y,\lambda)}{\partial \lambda} = 3x^2 + 2y^2 - 6 = 0 \quad (20)$$

- We could similarly make the Hessian from the derivatives computed above.

Interpretation of Lagrange Multipliers

- Lagrange Multipliers measure the *rate of change in the objective function as we relax the constraint*.
- To see this, recall that the F.O.C. of the Lagrangian is:

$$\nabla f(\vec{x}) - \lambda \nabla g(\vec{x}) = 0, \quad g(\vec{x}) = c \tag{21}$$

$$\frac{\partial f(\vec{x})}{\partial x_i} = \lambda \frac{\partial g(\vec{x})}{\partial x_i} \quad \forall i \tag{22}$$

- Since $\vec{x}$ depends on c, if we take derivative wrt c of $g(\vec{x}(c)) = c$, we get

$$\sum_{i=1}^{N} \frac{\partial g(\vec{x})}{\partial x_i} \frac{dx_i}{dc} = 1 \tag{23}$$

Interpretation of Lagrange Multipliers

- Next, let's calculate the change in $f(\vec{x})$ for a corresponding change in c:

$$\begin{aligned} \frac{df(\vec{x})}{dc} &= \sum_{i=1}^{N} \left\{ \frac{\partial f(\vec{x})}{\partial x_i} \frac{\partial x_i}{\partial c} \right\} \\ &= \sum_{i=1}^{N} \lambda \left\{ \frac{dg(\vec{x})}{\partial x_i} \frac{dx_i}{dc} \right\} \\ &= \lambda \end{aligned} \tag{24}$$

because of (23).

Interpretation of Lagrange Multipliers

- Therefore, the Lagrange Multipliers, or the λ's, have an important economic interpretation.
- The λ's are often referred to as shadow prices of inventory.
- In the context of portfolio optimization, the λ's tell us the additional utility that we could get by relaxing a given constraint.
- In the context of calibration formulated with option prices as constraints, the λ's will tell us the increase in the objective function that we could get by changing the option price.
  - We saw an example of this in our Risk Neutral Density lecture, where we minimized the curvature of a density.
  - In this example, the λ would tell us how much curvature is being introduced by each option price constraint.

Optimization Methods in Practice

- As we incorporate inequality constraints, it will become impossible to find even a local maximum or minimum analytically. Instead we must rely on iterative methods.
- This is also true as we increase the complexity of our objective function.
  - For example, we may be working with functions with unknown derivatives.
- Problems that quants face in the real-world generally require iterative algorithms and cannot be solved analytically.
- It is generally recommended to use pre-built optimization packages rather than trying to implement your own.

Root-Finding

- The simplest form of optimization problem that we will face is:

$$g(\theta) = 0 \tag{25}$$

- A solution to (25) is called a **root** of g.
- We assume we are dealing with cases where the roots cannot be computed analytically.
  - Perhaps the most commonly used example of this is extracting an implied volatility.
- Many different algorithms exist for solving these problems. In class today we will consider **Newton's Method**, but there are many others.

Root-Finding Algorithm

- We can find a root by repeatedly evaluating the function g as follows:

  1. Start with some initial θ

  2(a) If $g(\theta) > 0$, then choose a new θ' such that $g(\theta) > g(\theta')$

  (b) If $g(\theta) < 0$, then choose a new θ' such that $g(\theta') > g(\theta)$

  (c) If $g(\theta) = 0$, stop

  3. Set $\theta = \theta'$ and repeat step 2

- Outstanding Questions
  - How do we choose the new parameter θ' in each step?
  - How do we know that the method converges?
  - How do we know that we have reached a global maximum?

Root Finding: Newton's Method

- Newton's method relies on Taylor Series expansion to choose θ' in each step:

$$g(\theta') \approx g(\theta) + g'(\theta)(\theta' - \theta) \tag{26}$$

  where g' is the Jacobian of g.

- The idea, since finding the root of g itself is impossible, is to find a root of the linear approximation to g and use it to refine the previous guess.
- Thus, we can solve (26) for θ' to obtain a first order approximation for the root of the function, $g(x)$.

$$\theta' = \theta - (g'(\theta))^{-1} g(\theta)$$

- This choice of θ' at each step is known as **Newton's method**
- Note that we use information about the function $g(\theta)$ as well as its first derivative in each iteration.

Calibration of Stochastic Processes

- Returning to the problem of calibrating a volatility surface, as discussed in earlier lectures there are two general methods:
  - Parametric: Begin with a model or stochastic process and find the parameters that lead to the best fit to option prices
  - Non-Parametric: Attempt to extract a set of probabilities directly
- In both cases we begin with a set of market prices that we would ideally like our model to match and we formulate our optimization.
- We have previously considered least squares formulations to find a set of optimal parameters:

$$\vec{p}_{\min} = \underset{\vec{p}}{\operatorname{argmin}} \left\{ \sum_{\tau,K} \left(\hat{c}(\tau, K, \vec{p}) - c_{\tau,K} \right)^2 \right\} \tag{27}$$

Calibration of Stochastic Processes: Objective Functions

- Of course, equation (27) is one of many objective functions that we could consider in our calibration.
- In particular we could:
  - Place different weight on each option, based on moneyness or our confidence in the quote.
  - Calculate the error as a function of price, implied volatility or logarithm of price
  - Use absolute or relative pricing error
  - Raise the absolute pricing error to a different power rather than use the squared pricing error.

Calibration of Stochastic Processes: Regularization Functions

- We can use a **regularization** term in our calibration:
  - This may help us to enforce stability of the parameters in time.
  - It might direct us to the "right" local minimum.
  - Additionally, it can reduce sensitivity of our calibration to our initial guess, choice of objective or weighting function, or input data.
- The regularization and pricing error functions combine to make an augmented objective function.
- To the extent that we are able, it would be wise to formulate the regularization term as a convex function. (Why?)
- One example would be to minimize the squared distance from our initial guess.

Calibration of Stochastic Processes: Formulation

- Putting this together, we have the following form of optimization function:

$$\vec{p}_{\min} = \underset{\vec{p}}{\operatorname{argmin}} \left\{ \sum_{\tau,K} \omega_{\tau,K} \left(|\hat{c}(\tau, K, \vec{p}) - c_{\tau,K}| \right)^m \right\} \tag{28}$$

- Equation (28) assumes absolute error, but we could just as easily replace with relative error.
- Additionally, although we have formulated our problem in terms of pricing constraints, there are often **linear constraints** related to the values that the model parameters can take.
- We can write these linear constraints as:

$$l \leq \vec{p} \leq u \tag{29}$$

Gradient vs. Gradient Free Optimization

- Optimization algorithms can be classified into two main groups: gradient based and gradient free.
  - Gradient based methods utilize information about the function and its derivative in order to determine each iterative step.
  - Gradient free methods utilize only information about the function in order to determine the next step.
- In general each step will be more expensive in a gradient based method because each step requires evaluation of the function and its derivatives.
- However, gradient based methods will generally require fewer iterations as they are able to determine the optimal step size.

Gradient vs. Gradient Free Optimization

- Gradient Free methods are often preferable in the financial applications that you will face in your career
- If you choose to apply Gradient methods in your applications you will often have to compute the required derivatives numerically as they will not be available analytically.
- Many commercial optimization packages already exist and it is unlikely you will ever have to implement your own.
- One particularly useful optimization package is **IPOPT**, which is available in R and most other programming languages.

Gradient Based Methods with Linear Constraints

- Gradient Based Optimization require evaluation of the **objective function**, the **constraints**, as well as the **gradient of the objective function and of the constraints**.
- For large dimension problems, this means many evaluations are being computed per step in the optimization routine.
- Linear constraints are of particular interest in this context because the have a constant derivative.
  - This means they only need to be evaluated once rather than once per step.

Nelder-Mead Simplex Optimization Method

- Nelder-Mead is a gradient free optimization algorithm that allows us to solve unconstrained optimization problems.
- The idea is to search for an optimum in a grid of parameters, and then make the grid finer in areas in which the optimum is most likely to be in
- By searching in many different areas, the Nelder-Mead can find global optima without evaluating the gradient of f
- R and MatLab both have pre-built packages that implement the Nelder-Mead method.

Working with Options Data

- Calibration algorithms tend to be very sensitive to input data.
- Ideally, we would like small changes in market data not to cause significant changes in our calibration results; however, this is not always the case.
- Generally speaking, the models that we use assume that the market is arbitrage free. If the data violates this assumption then the calibration may not yield meaningful results.
- Additionally, how much weight is given to strikes that are close to near at-the-money strikes vs. out-of-the-money strikes is subjective and can have a significant impact on the calibration results.

Data: Arbitrage Requirements

- In order to prevent arbitrage in our data, we must have:
  - Call (put) prices that are monotonically decreasing (increasing) in strike.
  - Call (put) prices whose rate of change is greater than 0 (-1) and less than 1 (0)
  - Call and put prices that are convex with respect to changes in strike.
- As a general rule, I also tend to discard one-sided quotes (that is, with 0 bid or offer).
- See **Hirsa, page 331** for a detailed set of recommended data requirements.

Challenges of Calibrating Stochastic Processes in Practice

- We are working with options data that is dirty, even for the most liquid underlyings. How we clean this data may have a non-trivial impact on our results.
- In many cases, the parameters in the SDE's that we are trying to solve are highly correlated.
- The calibration problems that we are trying to solve are often poorly posed.
  - Even when we are able to guarantee the objective function is convex, we may find that derivatives are not well behaved.
  - We often find that these models have a flat bottom with respect to multiple parameters.
- The end result is that our results are very sensitive to inputs.