Today’s Topics

- Volatility smile and model calibration
- Solving nonlinear least squares problems
- Challenges in option model calibration
Calibrating Option Models

Given a model (e.g., LVF, SV, jump), we can (using FD and MC)

- price options
- hedge options
- quantitatively evaluate risk, e.g., reporting VaR and CVaR

How do we obtain a good model?

⇒ Choose a model which is consistent with the option market!
Marking a Model to Market

Assume that a set of today’s liquid option prices
\[\{V^\text{mkt}_0(K_l, T_l), l = 1, 2, \cdots, m\}\] on an underlying are observed from the option market.

Assume that \(\Omega\) denotes the set of possible models of interest.

Assume that \(V_0(K_l, T_l; \mathcal{H}), \mathcal{H} \in \Omega\), denotes the model initial value.
A model can be calibrated by solving

\[ V_0(K_l, T_l; \mathcal{H}) - V_0^{\text{mkt}}(K_l, T_l) = 0, \quad l = 1, 2, \cdots, m \]  

(1)

or

\[ \min_{\mathcal{H} \in \Omega} \sum_{l=1}^{m} \left( V_0(K_l, T_l; \mathcal{H}) - V_0^{\text{mkt}}(K_l, T_l) \right)^2 \]  

(2)

Typically calibration problem is formulated as in (2).

Additional constraints, e.g., bound constraints on the parameters, can also be imposed on \( \mathcal{H} \).

Is a solution to (2) a solution to (1)?
Model Calibration Examples

For a Merton’s jump model, $\mathcal{H} = \{\sigma, \lambda_J, \mu_J, \sigma_J\}$

Similarly, stochastic volatility model is described by a few parameters.

For a local volatility function model, different parametric forms have been propose, e.g., a quadratic function.

A machine learning approach?

Calibration becomes determining the model parameters.
Note: \( V_0(K_l, T_i; \mathcal{H}) \) depends \textit{nonlinearly} on model parameters \( \mathcal{H} \).

Solving this nonlinear least squares problem is often not straightforward.
Marking to Market

Calibrating model $\leftrightarrow$ marking to market.

Calibration ensures that the resulting model is consistent with the current market option price information.

If the calibrated model is used to price illiquid options, they are priced consistently with the current market information.

If the calibrated model is used to hedge options, the model consistent with the current market information is used to determine hedging positions.
A local volatility function \( \sigma(S, t; x) \) proposed in the literature is a quadratic, e.g., \( x \in \mathbb{R}^3 \):

\[
\sigma(S, t; x) = \max(x_1 + x_2(S - S_0) + x_3(S - S_0)^2, 0)
\]

or

\[
\sigma(S, t; x) = \max(x_1 + x_2S + x_3t + x_4S^2 + x_5t^2 + x_6St, 0) \quad (3)
\]

The model which best fits the market option prices can be estimated by solving the following nonlinear least squares problem

\[
\min_{x \in \mathbb{R}^n} \frac{1}{2} \sum_{j=1}^{m} \left( V_0(K_j, T_j; \sigma(S, t; x)) - V_0^{mkt}(K_j, T_j) \right)^2.
\]
Let $F(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ denote the residual vector

\[
F(x) \overset{\text{def}}{=} \begin{bmatrix}
V_0(K_1, T_1; \sigma(S, t; x)) - V_0^\text{mkt}(K_1, T_1) \\
\vdots \\
V_0(K_m, T_m; \sigma(S, t; x)) - V_0^\text{mkt}(K_m, T_m)
\end{bmatrix}
\]
Each initial model value $V_0(K_j, T_j; \sigma(S, t; x))$ is a complicated nonlinear function of $x \in \mathbb{R}^n$.

Calibration problem is a *nonlinear least squares* problem

$$\min_{x \in \mathbb{R}^n} f(x) \overset{\text{def}}{=} \frac{1}{2} \|F(x)\|_2^2$$
Solving a Nonlinear Least Squares Problem

A nonlinear least squares problem is a special unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

where $f(x)$ is continuously differentiable.

Local vs Global:

- **local minimizer** $x_*$: there exists $\delta > 0$ such that
  $$f(x_*) \leq f(x), \quad \forall \|x - x_*\| \leq \delta$$

- **global minimizer** $x_*$: $f(x_*) \leq f(x), \quad \forall x$

**Warm Start**: Try to start from a good initial point, ideally close to the global solution. (More later on convex vs nonconvex optimization)
Computing a solution to a nonlinear optimization problem requires calculating function/gradient (and maybe Hessian).

For least squares function \( f(x) = \frac{1}{2} \|F\|_2^2 = \frac{1}{2} F^T F \), the gradient of \( f(x) \) is

\[
\nabla f = \begin{pmatrix}
\frac{\partial f}{\partial x_1} \\
\frac{\partial f}{\partial x_2} \\
\vdots \\
\frac{\partial f}{\partial x_n}
\end{pmatrix} = J(x)^T F(x)
\]
The Jacobian matrix $m$-by-$n$ of $F(x)$ is

$$J(x) = \begin{pmatrix}
\frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \ldots & \frac{\partial F_1}{\partial x_n} \\
\frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \ldots & \frac{\partial F_2}{\partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial F_m}{\partial x_1} & \frac{\partial F_m}{\partial x_2} & \ldots & \frac{\partial F_m}{\partial x_n}
\end{pmatrix}$$
Hessian of any function $f(x)$

$$\nabla^2 f(x) = \begin{pmatrix}
\frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\
\frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2}
\end{pmatrix}$$

The Hessian of the least squares objective function $f(x) = \frac{1}{2} \| F \|^2_2$ is

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{i=1}^m \nabla^2 F_i(x) F_i(x)$$

where $\nabla^2 F_i(x)$ is the Hessian matrix of $F_i(x)$.

Proof: Exercise.
A Newton Step

Assume that $x_c$ is the current approximation.

Newton step for $\nabla f = 0$ is

$$x_+ = x_c - (J(x_c)^T J(x_c) + S(x_c))^{-1} J(x_c)^T F(x_c)$$

where $S(x) = \sum_{i=1}^m \nabla^2 F_i(x) F_i(x)$. 
Jacobian: Finite Difference

Consider a single function $g(z)$ of one variable $z \in \mathbb{R}$.

$$g'(z) = \frac{g(z + \Delta z) - g(z)}{\Delta z} + O(\Delta z)$$

Suppose that the function is already evaluated at $z = z_c$.

The derivative can be approximated by one additional function evaluation $g(z + \Delta z)$.

How do we choose $\Delta z$?

What about choosing machine epsilon ($\text{eps}$ in Matlab) as $\Delta z$?
With no additional information, choose $\Delta z = \sqrt{\epsilon}$. 

Note: complex finite difference approximation can be another alternative
Approximate Jacobian Using FD

Assume $F(x_c)$ has been computed. Let $\Delta \approx \sqrt{\epsilon}$

The $m$-by-$n$ Jacobian matrix can be approximated by:

for $j = 1 : m$
  for $i = 1 : n$
    $\frac{\partial F_j}{\partial x_i} \approx \frac{F_j(x_c + \Delta \cdot e_i) - F_j(x_c)}{\Delta}$
  end
end

e_i = (0, 0, \cdots, 0, 1, 0, \cdots, 0): zero everywhere except the $i$th component
A Gauss Newton Step

Hessian matrices $\nabla^2 F_i(x)$, $i = 1, \ldots, m$, are too expensive to compute.

Assume that $x_c \in \mathbb{R}^n$ is the current approximation.

Consider an affine model for $F(x)$ around $x_c$

$$M_c(x) = F(x_c) + J(x_c)(x-x_c)$$

The system $M_c(x) = 0$ is an over-determined (if $m \geq n$) linear least squares problem.
It is natural to set $x_+$ as the least squares solution, i.e.,

$$
\min_x \frac{1}{2} \| F(x_c) + J(x_c)(x-x_c) \|_2^2
$$

$$
x_+ = x_c - (J(x_c)^T J(x_c))^{-1} J(x_c)^T F(x_c)
$$

This is a Gauss Newton step.

In general, the least squares residual, when $m \geq n$, is not zero.
Newton and Gauss Newton Methods

**Difference:** the second order derivatives $S(x_c)$ is not computed in the Gauss Newton step and $\nabla^2 f(x_c)$ is approximated by $J(x_c)^T J(x_c)$.

Recall that $\nabla^2 f(x) = J(x)^T J(x) + S(x)$ where

$$S(x) = \sum_{i=1}^{m} \nabla^2 F_i(x) F_i(x).$$
If, at a minimizer $x_*$, $S(x_*) = 0$, the Gauss Newton method can be shown to converge locally quadratically. This includes:

- $F(x_*) = 0$ for all $i$ or
- $\nabla^2 F_i(x_*) = 0$ for all $i$
Advantages of the Gauss Newton Method

• Only function $F(x)$ and Jacobian $\nabla F$ evaluations are required.
• Solve a linear least squares problem in one iteration
• Quickly locally linearly convergent on problems that are not too nonlinear and have reasonably small residuals
Difficulties:

1. Problems that are sufficiently nonlinear or have reasonably large residuals: slow linear convergence

2. Very nonlinear or have large residuals: not locally convergent.

3. Not well defined if $J(x_c)$ does not have full column rank.

4. Not necessarily globally convergent, i.e., convergent from any starting point.