Project discussion and essential techniques

Keith Law

Project requirement

- You are required to use the techniques learnt from the course to tackle the problem
- Each group is required to write a report on the methodology and results in less than 1000 words (which exclude all graphs / tables / codes), together with the python codes, Powerpoint (optional) all in a zipped folder
- All projects are popular structured notes trading in the market, you are required to solve the specific problem by Monte Carlo technique
- Grading will be based on a holistic approach: The effort of the whole team, how much you applied (the techniques you learnt from the course), the accuracy and proficiency of codes, the professional standard of the report and presentation
- Term sheet: Please pick a date (Trade date) as a reference point for all volatility surface, yield curve (discounting purpose), spot prices etc.
- You have freedom on how much effort to spend on the project (which will be reflected in your grade) and how to present the report. For example, you may try constant Black-Scholes vol/ local vol/ stochastic vol/ stochastic local vol, or different monte Carlo cycles and compare the difference in results etc.

Project requirement

- Teamwork:
  - Ideally, coding can be designed by modules (or 'Class' and each member can contribute)
  - For example, within a Monte Carlo framework, you can input whether to use BS vol / Local vol/ stochastic vol/ SLV etc. and they can be written in separate functions/ Classes to be called in the main function/ class
  - Supporting functions like correlation, Cholesky decomposition and yield curve interpolation can be stand-alone functions to be called
  - The team should have good division of labor. Suggestion: coding (at least 2 persons), writing and presenting the results in a professional report and presentation(1-2 persons)
  - Ensure good and concise English (make sure you proof-read)

Project requirement

- Presentation:
  - Will be held during the last class, each group has around 10mins, but no more than 15mins (including Q&A)
  - Powerpoint is optional, i.e. you may present slides or your project report directly
  - You only need to 'briefly' describe the product as all products are similar, this part should be as short and concise as possible
  - Focus should be on how you tackle the problem. Describe the structure of the code (NOT every line), briefly tell what functions do what…
  - Ensure the key results are presented in a clear way through tables / graphs etc.
  - You may also highlight the special effort (you believe that should deserve extra points)
  - **Competition: Each team is required to <u>'challenge'</u> the other team who works on the same problem by asking a meaningful and challenging question, i.e. STAY FOCUS on your competitor's presentation**

Some helpful tips

- Simulating multi-processes
  - You can use the ‘CORR’ function in Bloomberg which allows you to build correlation matrix, using whatever window and frequency of data
  - Cholesky decomposition: numpy.linalg.Cholesky
- Yield curve:
  - Risk-free rates: You can use the ‘GGR’ function in Bloomberg and pick the corresponding market
  - In case generic government rates not available, try ‘MMR’ which is the money market rate, not risk free
  - Usually short rates are only available up to one-year, for those longer rates, consider using the swap rates
- Dividend curve
  - For index, most liquid indexes have dividend swap trading. You can change the index points into yields for different maturities
  - For stocks, you may use dividend forecast from Bloomberg or banks

Some helpful tips

- Yield curve and dividend curve – How to interpolate?
  - Linear interpolation? 'Kinks' at the provided points not ideal
  - Polynomial interpolation? Erratic behavior can happen between nodes
  - Cubic spline: Most popular methods, fitting a local cubic polynomial for each interval
- Cubic spline (scipy.interpolate.CubicSpline)
  - Given $K = \{x_0, \dots, x_m\}$ be a set of given knots (maturities), and the rates observed $\{y_0, \dots, y_m\}$, we want to fit a local cubic polynomial s_i, where $i = 0, \dots, m-1$ and s_i is the polynomial for each interval $[x_i, x_{i+1}]$, we need $4m$ coefficients for the m equations (intervals)
    - $s_i(x_i) = y_i$ for $i = 0{:}\, m-1$ (m conditions for point fits)
    - $s_{m-1}(x_m) = y_m$ (1 condition for fitting the end point)
    - $s_i(x_{i+1}) = s_{i+1}(x_{i+1})$ for $i = 0{:}\, m-2$ (m-1 conditions for 'forced continuity')
    - $s'_i(x_{i+1}) = s'_{i+1}(x_{i+1})$ for $i = 0{:}\, m-2$ (m-1 conditions for 'forced 1st differentiability')
    - $s''_i(x_{i+1}) = s''_{i+1}(x_{i+1})$ for $i = 0{:}\, m-2$ (m-1 conditions for 'forced 2nd differentiability')
    - We need 2 more conditions to identify the system! One can use 'Natural spline' to define the 2 boundary conditions $s_0''(x_0) = 0$, $s_{m-1}''(x_m) = 0$.

Some helpful tips

- Volatility surface interpolation
  - Bloomberg has implied vol surface, but since the project is a test on your understanding of the course, you are encouraged to build your own (as a bank cannot use Bloomberg vol surface within their system)
  - You can use ‘OMON’ to check some quotes of the liquid listed options with lists of strikes and expiries
  - You may (optionally) use ‘BLOTTER’ to see what are actually traded within the InterDealer Board (IDB market)
  - You will get some scattered prices in the vol surface across strikes and expiries
  - You may either use the Dupire formula to get local vols of certain points and then parameterize the local vol surface using the calibration technique, which is preferred since the local vol surface can be directly used in Monte Carlo simulation
  - Alternatively, you may fit the implied vol surface using the calibration technique, but then you need to transform the implied vol into local vol using the Gatheral’s mapping we discussed

Some helpful tips

- Numerical differentiation
  - For handling the $\frac{\partial C}{\partial T}, \frac{\partial^2 C}{\partial K^2}$ etc. of the Dupire equation
  - If we use a forward difference scheme, i.e. $\frac{\partial V}{\partial S} = \frac{V(S_o+\Delta S)-V(S_0)}{\Delta S}$, we can show the approximation error is $O(\Delta S)$
  - If we use a central difference scheme, i.e. $\frac{\partial V}{\partial S} = \frac{V(S_o+\Delta S)-V(S_0-\Delta S)}{2\Delta S}$, we can show the approximation error is $O(\Delta S^2)$
  - We usually use the central difference scheme for the ‘strike’ and forward difference scheme for the ‘Time’
  - Using the same logic, we can derive $\frac{\partial^2 V}{\partial S^2} = \frac{V(S_0+\Delta S)-2V(S_0)+V(S_0-\Delta S)}{\Delta S^2}$

Once you get the local vol points using the above discretization methods, you can interpolate the whole local vol surface using the parametric technique for calibration