Distributed Computing in Finance: Case Model Calibration

Juho Kanniainen

Global Derivatives Trading & Risk Management
19 May 2010

Techila Technologies,
Tampere University of Technology

juho.kanniainen@techila.fi
juho.kanniainen@tut.fi

Outline

- Simple example
- Monte Carlo simulations using Techila Grid
- The calibration of GARCH models with a large set of empirical option prices

Example: Prime numbers

- Calculate the prime numbers from the interval
$$\left[10^7+1, 10^7+n10^7\right] = \left[10^7+1, 10^7(n+1)\right],$$
where n is a positive integer.
- For $n = 1$, it takes more than 250 seconds, and hence, e.g. for $n = 10$, it takes a loooong time.
- MATLAB code:

```
%% Calculate the result locally
integers = round((10^7+1):1:(n*10^7+10^7));
% The primality of the integers is done by trial division
% See also http://en.wikipedia.org/wiki/Prime_number
primeNumbers = integers(isprime(integers));
```

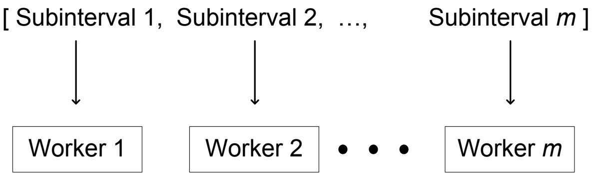
Example: Prime numbers

- We can divide the original interval into several, say m, subintervals:
  1. $[10^7 + 1, 10^7(n/m + 1)]$
  2. $[10^7(n/m + 1) + 1, 10^7(2 \times n/m + 1)]$
  3. $[10^7(2 \times n/m + 1) + 1, 10^7(3 \times n/m + 1)]$
  4. $[10^7(3 \times n/m + 1) + 1, 10^7(4 \times n/m + 1)]$
  5. $\ldots$
- That is, the kth, $k = 1, 2, \ldots, m$, subinterval is

$$\left[\left(10^7\left((k-1)\frac{n}{m}+1\right)+1\right), \left(10^7\left(k\frac{n}{m}+1\right)\right)\right].$$

Example: Prime numbers

- We can calculate the primes of the each subinterval separately, and then collect the results.
- That is, we can divide the problem into m independent subtasks and time-consuming calculations can be run on different computational units ("cores"), with no need for communication between the subtasks.
- Ideally the speedup is nearly linear, i.e. proportional to the number of cores, m.



Example: Prime numbers

```
%% Add grid toolbox functions to the path
gmkRoot='c:\gmk\';
addpath([gmkRoot '/grid/Matlab']);

%% Calculate the result using the TECHILA GRID
results = peach('findPrimes', {n, m,'<param>'},{}, ...
                 1:m, 'gmkRoot', gmkRoot);
primeNumbers = cell2mat(results);
```

```
function result = findPrimes(n, m, jobIdx)
    % Define the interval
    integers = round(10^7*((jobIdx-1)*n/m + 1) + 1):...
                    1:(10^7*(jobIdx*n/m + 1));
    % Find the prime numbers within the interval
    result = integers(isprime(integers));
end
```

Example: Prime numbers

Results with $n = 10$ and $m = 400$:

```
Created parameter bundle in 1.159609 seconds.
Project created with id: 103840
Created project in 2.459716 seconds.
Completed: 100%
Downloading results....
Downloaded in 7.848910 seconds.
Extracting results....
Extracted results in 3.675095 seconds.
Loaded results in 0.432372 seconds.
CPU time used 0 d 2 h 11 m 51 s.
Real time used 0 d 0 h 0 m 57 s.
Total time used 0 d 0 h 1 m 21 s.
```

Monte Carlo Methods and Distributed Computing

- Monte Carlo simulation has become an essential tool in financial engineering
  - GARCH models and other volatility models without (semi-)closed-form expressions for options
  - Exotic options
  - LIBOR models
  - Risk measures
- Variance-reduction techniques can reduce the number of Monte Carlo simulations needed to achieve a given accuracy and computing speed for a certaint point, but not necessarily enough.
- On the other hand, Monte Carlo computations can naturally be divided into independent subtasks, and thus the distributed computing is applicable.

Our Research Related on Distributed Computing

- Stochastic Volatility, Dividends, and Stock Market Equilibrium
- Calibration of GARCH Models using Particle Filter
- Calibration Strategies of Stochastic Volatility Models
- Calibrated Stochastic Volatility Models and Exotic Options
- Volatility Feedback in Option Data: Empirical GARCH Analysis

Introduction to GARCH Option Pricing

- A very important feature of stock returns is conditional heteroscedasticity, which means that the variability (volatility) of the returns changes in time. That is, there are more and less uncertain times in the financial markets.
- The family of GARCH (generalized autoregressive conditional heteroscedasticity) volatility models capture the empirical properties well, which makes them important tools in empirical asset pricing and risk management.
- The goal of this presentation is to show how the performance of GARCH models can be assesed using distributed computing.

Introduction to GARCH Option Pricing

- Lately, GARCH models have been applied to option pricing with a good success.
- The main advantage of GARCH models is that the volatility dynamics can be characterized in imaginative ways. In fact, the family of GARCH models amounts to an infinite state space setups.
- However, there is a drawback: In most GARCH option pricing models, no closed-form analytical solution for the option price is available and the price is available only through Monte Carlo simulation.
- This makes the model calibration computationally expensive, even if we use variance reduction techniques (EMS, control variates etc.)

Some related articles:

- Christoffersen P. and K. Jacobs, 2004, "Which GARCH Model for Option Valuation," *Management Science*, 50, 9
- Barone-Adesi, G., R. Engle, , L. Mancini, 2008, "A GARCH Option Pricing Model with Filtered Historical Simulation", *Review of Financial Studies*, 21, 1223-1258
- Mercuri, L, 2008, "Option pricing in GARCH model with tempered stable innovations," *Finance Research Letters*, 5, 172-182.

GARCH Models

- The stock log-return process is assumed to have a form

$$\ln\left(\frac{S_t}{S_{t-1}}\right) = r + \lambda\sqrt{h_t} - \frac{1}{2}h_t + \sqrt{h_t}z_t,$$

where $r, \lambda > 0$ and $z_t \sim N(0,1), \mathbb{E}\left[z_{t_j} z_{t_k}\right] = 0$ for $j \neq k$. Here h_t denotes the conditional squared volatility.

- The volatility process evolves as

$$h_t = \beta_0 + \sum_{i=1}^{p} \beta_i h_{t-1} + \sum_{i=p+1}^{p+q} \beta_i f\left(z_{t-1}, h_{t-1}\right),$$

where $p, q \geq 1$.

GARCH Models under Risk-Neutral Probability Measure

- Under the risk-neutral probability measure, the corresponding return and volatility processes are

$$\ln\left(\frac{S_t}{S_{t-1}}\right) = r - \frac{1}{2}h_t + \sqrt{h_t}z_t^*,$$

and

$$h_t = \beta_0 + \sum_{i=1}^{p}\beta_i h_{t-1} + \sum_{i=p+1}^{p+q}\beta_i f\left(z_{t-1}^* - \lambda, h_{t-1}\right),$$

where $z^* \sim N(0,1)$ under the risk-neutral probability measure.

Some Specific GARCH Models (under the risk-neutral measure)

- Leverage model:

$$\begin{aligned} h_t &= \beta_0 + \beta_1 h_{t-1} + \beta_2 h_{t-1} \left(z^*_{t-1} - \lambda - \theta \right)^2 \\ &= \beta_0 + \beta_1 h_{t-1} + \beta_2 h_{t-1} \left(z^*_{t-1} - \beta^*_3 \right)^2, \end{aligned}$$

  where $\beta^*_3 = \beta_3 + \lambda$.

- HGARCH (news&power):

$$h_t = \beta_0 + \beta_1 h_{t-1} + \beta_2 h_{t-1} \left(\left| z^*_{t-1} - \beta^*_3 \right| - \kappa \left(z^*_{t-1} - \beta^*_3 \right) \right)^{2\gamma}.$$

Calibration of Option Pricing Models

- For the purpose of option pricing, it may be preferable to estimate parameters of the underlying model directly using the empirical observations of option prices. This procedure is called calibration.
- The task is to find GARCH parameters $\boldsymbol{\theta}$ that minimizes the empirical pricing error (the objective function of our optimization problem):

$$\min_{\boldsymbol{\theta}} \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{C}_i - C_i(\boldsymbol{\theta})\right)^2},$$

where n is the number of observations, $\hat{C}_i$ an observed market price, and $C_i(\boldsymbol{\theta})$ the corresponding model price with GARCH parameter vector.

Calibration of Option Pricing Models

- For example, in the case of the leverage model there are four parameters to estimate:

$$\boldsymbol{\theta} = (\beta_0, \beta_1, \beta_2, \beta_3^*)^{\mathsf{T}}$$

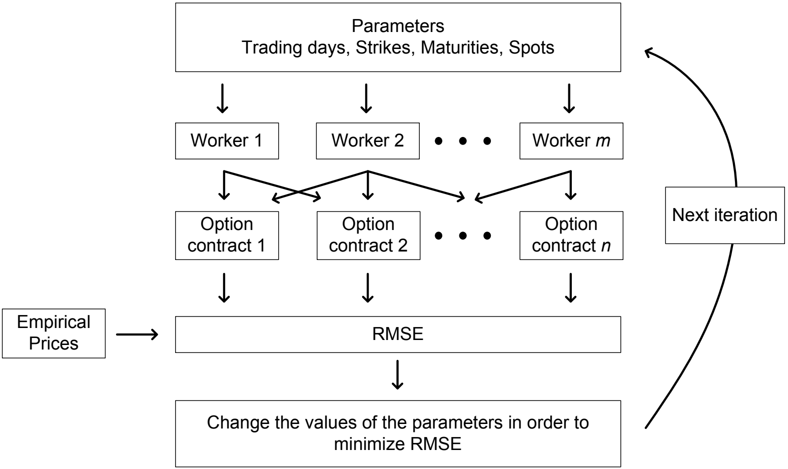
- HGARCH model has six unknown parameters:

$$\boldsymbol{\theta} = (\beta_0, \beta_1, \beta_2, \beta_3^*, \kappa, \gamma)^{\mathsf{T}}.$$

Data

- We have option data for Wednesdays from 1990 to 1995 (the data sets were graciously provided to us by Prof. Peter Christoffersen and Prof. Gurdip Bakshi).
- The three year period between January 3, 1990, and December 31, 1992, is used for in-sample calibration. This data set consists of **9,176** data points.
- We use multi-day option prices to calibrate the parameters, and thus we 'update' the volatilities on different dates using the time series of stock returns, as Christoffersen and Jacobs (2004) do.
- One year period between January 6, 1993, and December 31, 1994, is used for out-of-the-sample testing. This data set consists of **3,578** data points.

Optimization Algorithms on Grid: Nelder-Mead



Calibration with the Distributing Computing

- We use 50,000 paths, antithetic variables, and also the empirical martingale simulation procedure proposed by Duan and Simonato (1998) to increase numerical efficiency. See also Duan and Simonato (2001).
- The use of Nelder-Mead algorithm (MATLAB: fminsearch) requires 300 – 1,000 iterations (depending on the specified tolerances) to minimize the pricing error sufficiently.
- Therefore, with our accuracy, the calibration of the leverage model could take about 800 - 1,200 hours of CPU time, but with Techila Grid, the total time is 3-6 hours.

Option Price Surfaces, Leverage, In-Sample

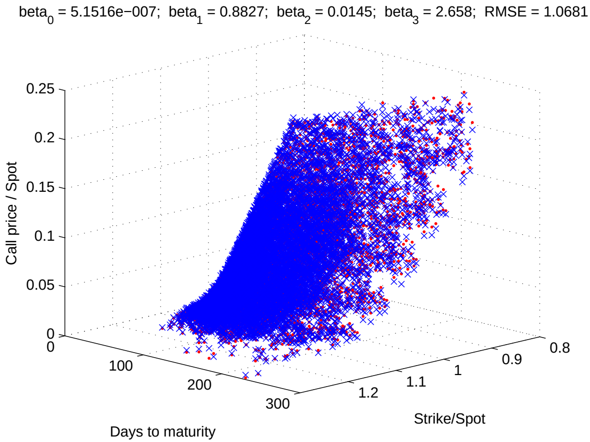


Figure: Finished calibration. Option price observations and model prices

Option Price Surfaces, Leverage, Out-of-Sample

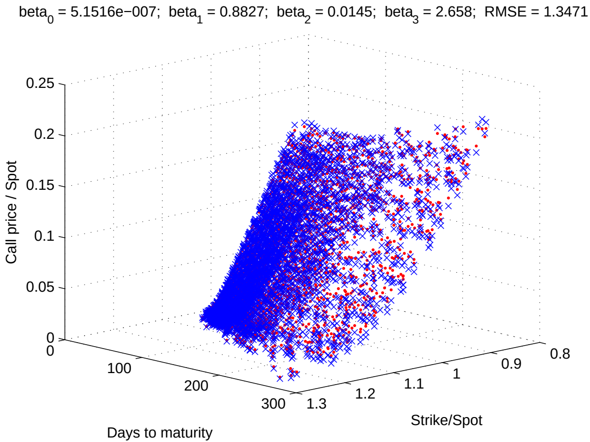


Figure: Out-of-Sample Fit. Option price observations and model prices

Some Results

Parameters:	Leverage	HGARCH
β_0	5.1517e-007	3.30E-07
β_1	0.88268	0.8590
β_2	0.014546	0.1690
β_3^*	2.6581	2.2606
γ		0.7260
κ		-0.6270
RMSE in sample	1.0496	0.9835
RMSE out-of sample	1.34705	1.1884

Thank you for your attention!
Question?

Email: juho.kanniainen@tut.fi, juho.kanniainen@techila.fi
Mobile: +358-40-7074532