Dupire Local Volatility Model
Version 1.0

1 Introduction

This plug-in implements the Dupire local volatility model. The main characteristic of the Dupire model is that is consistent with the surface of option prices (across strikes and maturities) given that its diffusion is governed by a state and time dependent volatility. For a general reference on local volatility models see [1], [2], [3].

2 How to use the plug-in

2.1 Dupire local volatility process

In the Fairmat user interface when you create a new stochastic process you will find the additional option "Dupire local volatility".

The stochastic process is defined by the parameters shown in table below:

Fairmat notation	**Documentation notation**
S0	S_0
Time Dependent Risk Free Rate (Zero Rate)	$r(t)$
Time Dependent Continuous Dividend Yield	$q(t)$
Local Volatility	$\sigma(t, s)$

S_0 is the starting values for the stock process $r(t)$ and $q(t)$ are one dimensional function while $\sigma(t, s)$ is a two dimensional function.

In order to use the plug-in function r and q must be defined in the Symbol lists and the stochastic process reference them: note that to specify a reference to a function defined in a Fairmat model (i.e `q`) the notation `@q` must be used.

3 Implementation Details

The Dupire local volatility model is used to describe the evolution of a stock price (or an index) with a volatility that is a function of time and index value. The process is defined by the following stochastic differential equation

$$dS(t) = (r(t) - q(t))S(t)dt + \sigma(t, S(t))S(t)dW(t) \tag{1}$$

where S represents the price process, and dW is a Wiener processes.

3.1 Simulation and discretization scheme

Applying straight Euler-Maruyama method to simulate a local volatility process, we obtain the following formula

$$S_{n+1} = S_n + [r(t_n) - q(t_n)]S_n\Delta t + \sigma(t_n, S_n)\sqrt{\Delta t}N(0,1) \tag{2}$$

where $\Delta t = t_{n+1} - t_n$ and $N(0,1)$ represents a realization of a standard normal random variable.

By using the discretization described above it is possible for S to reach negative values, therefore the best approach to overcome the problem is to simulate $log(S(t))$ process. Apply Ito's lemma, from equation (1) we can deduce

$$d\left[\log\left(S(t)\right)\right] = \left[r(t) - q(t) - \frac{1}{2}\sigma^2(t, S(t))\right]dt + \sigma(t, S(t))dW \tag{3}$$

and its discrete counterpart is

$$\begin{aligned}\log(S_{n+1}) = \log(S_n) + \left[r(t_n) - q(t_n) - \frac{1}{2}\sigma^2(t_n, S_n)\right]\Delta t \\ +\sigma(t_n, S_n)\sqrt{\Delta t}N_1(0,1)\end{aligned} \tag{4}$$

Simulating the stock price this way entails no discretization error and solve the problem of generating negative index value.

4 Calibration

It can be demonstrated that supposing to have a continuous surface of implied volatilities $\Sigma(t,S)$, the surface of local volatility is complitely determined by Dupire's formula

$$\sigma^2(t,S) = \frac{\Sigma^2 + 2\Sigma t\left[\frac{\partial\Sigma}{\partial t} + (r(t) - q(t))S\frac{\partial\Sigma}{\partial S}\right]}{\left(1 - \frac{yS}{\Sigma}\frac{\partial\Sigma}{\partial S}\right)^2 + tS\Sigma\left[\frac{\partial\Sigma}{\partial S} - \frac{1}{4}tS\Sigma\left(\frac{\partial\Sigma}{\partial S}\right)^2 + S\frac{\partial^2\Sigma}{\partial S^2}\right]} \tag{5}$$

where for simplicity we suppressed the (t,S) dependency of Σ and where

$$y(t,S) = \ln\left(\frac{S}{S_0}\right) + \int_0^t (q(s) - r(s))ds \tag{6}$$

Of course the main problem with this model is that the implied volatilitly surface Σ is not given, but the market provides us only an implied volatility matrix.

The first step is then to decide how to interpolate the implied volatility matrix to give a smooth surface on which calculate derivatives that appear in formula (5). This can be done in several ways, and the current implementation of the dupire plug-ins fits a quadratic model.

References

[1] Bruno Dupire. Pricing with a smile. *Risk Magazine*, 1994.

[2] Jim Gatheral. *The volatility surface: a practitioner's guide*. John Wiley & Sons, 2006.

[3] Roel van der Kamp. Local volatility modelling. Master's thesis, University of Twente, 2009.