Source | Model | Option Family
| Model_Option | Archived Dyn Tests

bs1d_std_test

/* Variables needed for exercice time of american options */

n_us: exercice time, it's an integer corresponding to the number of hedges to be performed by the market maker before the exercice of the american option.

sigma_us: square deviation for the simulation of n_us in the case of a gaussian random(see further).

m_us: mean deviation for the simulation of n_us in the case of a gaussian random(see further).

/* Variables needed for Brownian Bridge */

Bridge: the brownian bridge is a brownian process which trajectory passes by a wanted point at a wanted instant. (For further details on Brownian Bridge, see the documentation of Dynamic Tests)

d_Bridge: variation of the bridge between two periods of time.

StockT1: value of the stock at time *T1*, this value is given by the user as "Spot Target".

BridgeT1: value of the bridge at time *T1*, this *T1* is given by the user as "Time Target" and *BridgeT1* is deduced by the knowledge of *StockT1*.

/* Variables needed for graphic outputs */

stock_array and *pl_array* are arrays of double which contain the values of the stock and the P&L time after time.

/****** Initialization of the test's parameters ********/**

path_number: number of the different simulated stock's trajectories.

hedge_number: number of hedging acts by the market maker.

step_hedge: $h = \frac{Maturity-current\_date}{hedge\_number}$, period of time between two hedging acts.

cash_rate=e^{rh}, interest rate yielded by the bank account over the period of time h.

stock_rate=$e^{divid.h}-1$, dividends rate yielded by owning the stock account.

sigmaxsqrth=$\sigma\sqrt{h}$.

exp_trendxh=$e^{\left(\mu h-\frac{\sigma^2}{2}h\right)}$.

/* Determining exercice time for american option */

Here, we have chosen to simulate an exercice time for american option as following :

1. : If the option is european (the boolean equals 0), we set *n_us*=*hedge_number* (exercice time is the maturity).
2. : If the option is american and if the boolean equals 1, we simulate the exercice time *n_us* as an integer uniform random in [0 ; hedge_number].
3. : If the option is american and if the boolean equals 2, we simulate *n_us* as an integer gaussian random $N\left(\frac{hedge\_number}{2};\frac{hedge\_number}{6}\right)$.

/* Some initialization for Brownian Bridge */

Here we initialize the parameters needed to simulate the brownian bridge.

/* Graphic outputs initializations and dynamical memory allocutions */

We allocate dynamically some arrays to keep in the values needed for graphic outputs : stock's and P&L's trajectories.

/****** Trajectories of the stock *******/**

In this loop, we simulate path_number different stock's trajectories and for each we calculate the corresponding P&L.

/* Calculating selling_price and delta */

We send informations like the current date and the option's type to the chosen method, and this last gives us the corresponding selling price and delta at initial time.

/* Calculating cash_account and stock_account */

With the selling price and the delta given before, we determine the first cash account : *cash_account=selling_price-delta*stock*. And the stock account equals *delta*stock*, in fact delta is the quantity of stock owned by the Market Maker.

/* Brownian Bridge initialization */
We set the initial value of the Brownian Bridge at zero, and H is an intermediate parameter which depends of the current and the target time, it's needed to calculate the Bridge's value.

/****** Dynamic Hedge *******/**
This loop calculates the amount of money at current time out of a cash amount *selling_price* and a sequence of buying/selling (hedging) of the underlying asset between time *initial_time* and current time, with no option deals any longer between these two dates.

/* Capitalization of cash_account and yielding dividends */
The cash_account is capitalized at the rate *cash_rate* defined before and dividends are yielded with the rate *stock_rate* defined before.

/* Calculating the new stock's value */
At each step of the loop we simulate the stock's value given by the Black&Scholes model.

1. The first mode is to calculate the new stock's value is to use the expression of the solution of the E.D.S. $dS_t = S_t\left(\mu dt + \sigma dB_t\right)$, it gives us : $S_{t+h} = S_t \times e^{\left(\mu h - \frac{\sigma^2}{2}h + \sigma\sqrt{h}G\right)}$ where $G \hookrightarrow N\left(0,1\right)$.
2. The second consists in the decision to set the spot's value at S_{T1} at time $T1$, using the same formula as before but with a brownian bridge, the new stock's value is also calculated as follow : $S_{t+h} = S_t \times e^{\left(\left(\mu - \frac{\sigma^2}{2}\right)h\right)} \times e^{\sigma\left(\left(B_{T1}-B_t\right)H + \sqrt{h(1-H)}\times G\right)}$,
   where B_{T1} is the value of the bridge at time $T1$, $H = \frac{h}{T1 - current\_time}$, $h = step\_hedge$ and $G \hookrightarrow N\left(0,1\right)$. For further informations, please consult the document on Dynamic Tests.

/* Calculating the new selling_price and the new delta */
The same as before.

/* Calculating the new cash_account and the new stock_account */

new_*cash_account*=old_*cash_account*-(new_*delta*-old_*delta*)*stock.

/****** Last Hedge ********/**

That's the last step in the hedge process. Here, rather than using the selling_price, we simply use the pay-off known since the beginning : so we avoid all numerical problem of the last step of the loop for the reason that we are, at this time, nearby the maturity date and some numerical methods may cause a problem here.

/* **Capitalization of cash_account and yielding dividends */**
The same as before.

/* Calculating the last stock's value */
The same as before.

/* Capitalization of cash_account and calculating the P&L using the PayOff */
The capitalization is about the same as before (capitalization for a time step), the difference is that is capitalized to the maturity time in the case where the exercice time is before maturity (american option).

The P&L is finally calculated as follow : $P\&L = selling\_price - \sum(previous\_delta - delta) * stock - pay\_off$. (For replication formula, please refer to the document on Dynamic Tests)

/* Selection of trajectories (Spot and P&L) for graphic outputs */
Here we select different noteworthy spot's trajectories : we keep stock's trajectories generating the minimal, the maximal and the average P&L. The aim is to display them to observe the behavior of the used pricing method in extrem situations.