Source | Model | Option
| Model_Option | Help on mc methods | Archived Tests

# mc_fixedasian_privault

Input parameters:

- Number of iterations $N$
- Generator_Type
- Confidence Value
- Delta Method

Output parameters:

- Price $P$
- Error Price $\sigma_P$
- Delta $\delta$
- Error delta $\sigma_\delta$
- Price Confidence Interval: $IC_P$ =[Inf Price, Sup Price]
- Delta Confidence Interval: $IC_\delta$ =[Inf Delta, Sup Delta]

See DocPrivault

## Description:

Computation of the price and delta of a standard Asian option in a market with jumps.

We consider the following market model:

$$dS_t = rS_t dt + \sigma S_{t^-}(\beta_{N_{t^-}} dN_t - \nu dt),$$

where $(N_t)_{t\in\mathbb{R}_+}$ is a Poisson process with constant intensity $\lambda$, $(\beta_k)_{k\in\mathbb{N}}$ is a sequence of random variables independent of $(N_t)_{t\in\mathbb{R}_+}$, and $r$ represents the interest rate. For example, if $(\beta_k)_{k\in\mathbb{N}}$ is a sequence of independent indentically distributed random variables with: $P(\beta_k = b_i) = p_i$, $i = 1, \ldots, d$, $k \in \mathbb{N}$, we have

$$\beta_{N_{t^-}} dN_t = b_1 dN_t^1 + \cdots + b_d dN_t^d$$

where $N^1, \ldots, N^d$ are independent Poisson processes with intensities

$$(\lambda_i)_{i=1,\ldots,d} = (p_i\lambda)_{i=1,\ldots,d},$$

and $\nu = \lambda \sum_{i=1}^d b_i p_i$. The Delta of an Asian option is given by

$$\Delta = \frac{\partial C}{\partial x},$$

where

$$C(x) = E\left[f\left(\int_0^T S_u^x du\right)\right].$$

The derivative can be computed as

$$\frac{\partial}{\partial\zeta} E\left[f(F^\zeta)\right] = E\left[\frac{\partial}{\partial\zeta} f(F^\zeta)\right] = E\left[\frac{\partial}{\partial\zeta} F^\zeta f'(F^\zeta)\right],$$

however this formula requires a regularity property on $f$. Given a suitable function $w$ and a Poisson functional $F = f(T_1, \ldots, T_n)$, let $D_w F$ be the gradient of $F$ defined as

$$D_w F = -\sum_{k=1}^{k=n} w_{T_k} \partial_k f(T_1, \ldots, T_n).$$

The adjoint of D satisfies the integration by parts formula:

$$E\left[\delta(u)F\right] = E\left[D_u F\right].$$

As a consequence, a first derivative such as Delta can be computed as follows:

$$\frac{\partial}{\partial\zeta} E\left[f(F^\zeta)\right] = E\left[f(F^\zeta)\delta\left(w\frac{\partial_\zeta F^\zeta}{D_w F^\zeta}\right)\right], \tag{1}$$

with a weight given by

$$\delta\left(w\frac{\partial_\zeta F^\zeta}{D_w F^\zeta}\right) = \frac{\partial_\zeta F^\zeta}{D_w F^\zeta}\int_0^T \dot{w}_t dN_t - \frac{D_w \partial_\zeta F^\zeta}{D_w F^\zeta} + \frac{\partial_\zeta F^\zeta}{(D_w F^\zeta)^2} D_w D_w F^\zeta.$$

In the linear case $F^x = xF$ and $\partial_x F^x = F$, the above formula simplifies:

$$\delta\left(w\frac{\partial_x F^x}{D_w F^x}\right) = \frac{1}{x}\left(\frac{F}{D_w F}\int_0^T \dot{w}_t d\tilde{N}_t - 1 + \frac{F}{(D_w F)^2} D_w D_w F\right).$$

The solution of the equation

$$dS_t = \alpha S_t dt + \sigma S_{t^-}\beta_{N_{t^-}} dN_t, \quad S_0 = x,$$

driving $(S_t)_{t\in\mathrm{R}_+}$, with $\alpha = r - \nu\sigma$, can be written as

$$S_t = F(t, N_t),$$

where

$$F(t,k) = xe^{\int_0^t \alpha_s(N_s)ds}\prod_{i=0}^{i=k}(1+\beta_{i-1}\sigma).$$

The expression for the weight of Delta becomes:

$$\frac{1}{x\sigma}\left(\frac{\int_0^T S_t dt \int_0^T \dot{w}_t d\tilde{N}_t}{\int_0^T w_t S_{t^-}\beta_{N_{t^-}} dN_t} - 1 - \frac{\int_0^T S_t dt \int_0^T w_t\left(\dot{w}_t + \alpha w_t\right) S_{t^-}\beta_{N_{t^-}} dN_t}{\left(\int_0^T w_t S_{t^-}\beta_{N_{t^-}} dN_t\right)^2}\right).$$

## Algorithm:

/* Generation of Exponential Law. Interjump Times */

The interjump times are generated as exponential random variables.

/* Renormalization of sigma */

The value of the parameter $\sigma$ is divided by $\sqrt{\nu}$ in order to be consistent with the Black-Scholes model when the intensity $\nu$ becomes large.

/* Value to construct the confidence interval */

/* Initialization */

/*MC sampling*/

Initialization of the simulation: generator type, dimension, size $N$ of the sample, ... .

/* Test after initialization for the generator */

/* Begin N iterations */

/* Simulation of Poisson Jump Times */

We simulate Poisson jump times with intensity $\nu$ up to time $T$ and determine their number.

-/* Computation of Average and the Weight */

Two methods are used, first the classical finite differences and secondly the Malliavin method.

-/* Useful for computation of the weight */

We compute the terms needed for the calculation of weights.

/* Average */ Calculation of the average of the payoff.

/* Price */ The price of a standard Asian option is computed.

/* Delta */

/* Finite Differences */ In this method the computation of delta is performed with:

$$\text{Delta}_{FD} = C(s+\epsilon) - C(s-\epsilon)/(2*\epsilon)$$

/* Malliavin */ In this case we use the expression of the weight :

$$\frac{1}{x\sigma}\left(\frac{\int_0^T S_t dt \int_0^T \dot{w}_t d\tilde{N}_t}{\int_0^T w_t S_{t^-}\beta_{N_{t^-}} dN_t} - 1 - \frac{\int_0^T S_t dt \int_0^T w_t\left(\dot{w}_t + \alpha w_t\right) S_{t^-}\beta_{N_{t^-}} dN_t}{\left(\int_0^T w_t S_{t^-}\beta_{N_{t^-}} dN_t\right)^2}\right).$$

/*Sum*/

Computation of the sums $\sum P_i$ and $\sum \delta_i$ for the mean price and the mean delta where

$$P(i) = f\left(\frac{1}{T}\int_0^T S_t^x(i)dt, K\right) \quad \text{and} \quad \delta_i = f\left(\frac{1}{T}\int_0^T S_t^x(i)dt, K\right)W(i),$$

with

$$W(i) = \frac{1}{x\sigma}\left(\frac{\int_0^T S_t dt \int_0^T \dot{w}_t d\tilde{N}_t}{\int_0^T w_t S_{t^-}\beta_{N_{t^-}} dN_t} - 1 - \frac{\int_0^T S_t dt \int_0^T w_t\left(\dot{w}_t + \alpha w_t\right) S_{t^-}\beta_{N_{t^-}} dN_t}{\left(\int_0^T w_t S_{t^-}\beta_{N_{t^-}} dN_t\right)^2}\right).$$

/*Sum of squares*/ Computation of the sums $\sum P_i^2$ and $\sum \delta_i^2$ necessary for the variance price and the variance delta estimations. (finally only used for MC estimation)

/* End N iterations */

/*Price*/ The price estimator is:

$$P = \frac{1}{N} \exp(-rt) \sum_{i=1}^{N} P(i)$$

The error estimator is $\sigma_P$ with :

$$\sigma_P^2 = \frac{1}{N-1} \left( \frac{1}{N} \exp(-2rt) \sum_{i=1}^{N} P(i)^2 - P^2 \right).$$

/*Delta*/ The estimator of $\delta$ is:

$$\tilde{\delta} = \frac{1}{N} \exp(-rt) \sum_{i=1}^{N} f\left( \frac{1}{T} \int_0^T S_t^x(i) dt, K \right) W(i),$$

with

$$W(i) = \frac{1}{x\sigma} \left( \frac{\int_0^T S_t dt \int_0^T \dot{w}_t d\tilde{N}_t}{\int_0^T w_t S_{t^-} \beta_{N_{t^-}} dN_t} - 1 - \frac{\int_0^T S_t dt \int_0^T w_t \left(\dot{w}_t + \alpha w_t\right) S_{t^-} \beta_{N_{t^-}} dN_t}{\left(\int_0^T w_t S_{t^-} \beta_{N_{t^-}} dN_t\right)^2} \right).$$

The error estimator is $\sigma_\delta$ with:

$$\sigma_\delta^2 = \frac{1}{N-1} \left( \frac{1}{N} \exp(-2rt) \sum_{i=1}^{N} \delta(i)^2 - \delta^2 \right)$$

/* Price Confidence Interval */ The confidence interval is given as:

$$IC_P = [P - z_\alpha \sigma_P; P + z_\alpha \sigma_P]$$

with $z_\alpha$ computed from the confidence value.

/* Delta Confidence Interval */ The confidence interval is given as:

$$IC_\delta = [\delta - z_\alpha \sigma_\delta; \delta + z_\alpha \sigma_\delta]$$

with $z_\alpha$ computed from the confidence value.

Possible improvement: in this program the function jump_size yields a constant vector equal to 1 everywhere, but it can be implemented as a random variable.

# References