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

mc_floatingasian_standard

Input parameters:

- Time StepNumber M
- Generator_Type
- Number of iterations N
- Scheme
- Confidence Value α

Output parameters:

- Price P
- Error Price σ_P
- Delta δ
- Error delta σ_δ
- Price Confidence Interval: IC_P =[Inf Price, Sup Price]
- Delta Confidence Interval: IC_δ =[Inf Delta, Sup Delta]

Description:

Computation for a Floating Asian Call or Put European Option of its Price and its Delta with the Standard Monte Carlo or Quasi-Monte Carlo. In the case of Monte Carlo simulation, the method also provides an estimation for the integration error and a confidence interval.

For a best understanding of Asian option and a detailed description of the notations, we refer the reader to the general part about options ?????????. Simulation of an Asian option is not obvious because we need to generate

the mean of the underlying asset over a given period. Explanations about this point are described in the next points. You can read the part on simulation of random variables????????/ for a more complete presentation about simulation of Brownian trajectory.
Quasi Monte Carlo simulation is available for this options, but some restrictions appear: we need multidimensional low-discrepancy sequences and for some of them (like Sobol for instance) we are limited in practice with their dimension. See the implemented part for low-discrepancy sequences.

The underlying asset price evolves according to the Black and Scholes model, that is:

$$dS_u = S_u((r-d)du + \sigma dB_u), \quad S_{T_0} = s$$

then

$$S_T = s \exp\left((r - d - \frac{\sigma^2}{2})(T - T_0)\right) \exp(\sigma B_{T-T_0})$$

where S_T denotes the spot at maturity T, s is the initial spot, T_0 is the initial time.

The Price of an asian option at t is:

$$P_t = e^{-r(T-t)} E\left[f(S_T, A(t_0,T))\right]$$

where f denotes the payoff of the option, K the strike and $A(t_0,T)$ the mean of the price of the underlying asset over a given period $[t_0,T]$.
We have

$$A(t_0,T) = \frac{1}{T-t_0}\int_{t_0}^{T} S_u du$$

The Delta is given by:

$$\delta = e^{-r(T-t)}\frac{\partial}{\partial s}E[f(S_T, A(t_0,T))]$$

The estimators are expressed as:

$$\widetilde{P} = \frac{1}{N}e^{-r(T-t)}\sum_{i=1}^{N} P(i)$$

$$\widetilde{\delta} = \frac{1}{N}e^{-r(T-t)}\sum_{i=1}^{N}\frac{\partial}{\partial s}P(i) = \frac{1}{N}e^{-r(T-t)}\sum_{i=1}^{N}\delta(i)$$

Values for $P(i)$ and $\delta(i)$ are detailed for each option.

- **Asian Call Floating** : The payoff is $(S_T - A(t_0,T))^+$.
  - Case $t_0 \leq T_0$:
  We decompose $A(t_0,T)$ over $[t_0,T_0]$ and $[T_0,T]$. Then we have:

$$\begin{aligned} E\left[(S_T - A(t_0,T))^+\right] &= E\left[\left(S_T - K' - \tfrac{T-T_0}{T-t_0}\tfrac{1}{T-T_0}\int_{T_0}^{T} S_u du\right)^+\right] \\ &= E\left[(S_T - K' - A'(T_0,T))^+\right] \end{aligned}$$

  with $K' = \frac{T_0-t_0}{T-t_0}A(t_0,T_0)$
  and $A'(T_0,T) = \frac{1}{T-T_0}\int_{T_0}^{T} S'_u du$ with $S'_u = \frac{T-T_0}{T-t_0}S_u$.
  K' is named the pseudo strike and S' the pseudo spot.

  Hence we obtain the following expressions:

$$P(i) = \left(S_T(i) - K' - A'(T_0,T)(i)\right)^+$$

$$\delta(i) = \begin{cases} \frac{\partial S_T(i)}{\partial s} - \frac{\partial A'(T_0,T)(i)}{\partial s} = \frac{S_T(i)}{s} - \frac{A'(T_0,T)(i)}{s} & \text{if} \quad P(i) > 0 \\ 0 & \text{otherwise} \end{cases}$$

  - Case $T_0 < t_0$:

- **Asian Put Floating** : The payoff is $(A(t_0,T) - S_T)^+$.
  - Case $t_0 \leq T_0$:
  With the same decomposition as for a call, we find the following expressions:

$$P(i) = \left(K' + A'(T_0,T)(i) - S_T(i)\right)^+$$

$$\delta(i) = \begin{cases} \frac{\partial A'(T_0,T)(i)}{\partial s} - \frac{\partial S_T(i)}{\partial s} = \frac{A'(T_0,T)(i)}{s} - \frac{S_T(i)}{s} & \text{if} \quad P(i) > 0 \\ 0 & \text{otherwise} \end{cases}$$

  - Case $T_0 < t_0$:

Simulation of the mean A_T

We note

$$Y_t = \int_0^t S_u du$$

We propose 3 different schemes to estimate Y_t. The interval $[0,t]$ is divided into M steps and we note the step size $h = t/M$. We define the times $t_k = kt/M$.
A more detailed description for these schemes is given in [1].

1. **Scheme 1: Rieman sums**
   The estimation for Y_t is given by:

$$\widetilde{Y}_t^M = h \sum_{k=0}^{M-1} S_{t_k}$$

2. **Scheme 2: Trapezoidal method**
   The approximation for Y_t is obtained by considering the conditional expectation $E\left[\int_0^t S_u du \backslash \mathcal{B}_h\right]$ where $\mathcal{B}_h$ is the σ-field generated by the $S_{t_k}, k = 0, \ldots, M-1$.
   The conditional law of $(B_u \backslash B_{t_k}, B_{t_{k+1}})$ for $u \in [t_k, t_{k+1}]$ is given by:

$$\mathcal{L}(B_u \backslash B_{t_k} = x, B_{t_{k+1}} = y) = \mathcal{N}\left(\frac{t_{k+1} - u}{h} x + \frac{u - t_k}{h} y, \frac{(t_{k+1} - u)(u - t_k)}{h}\right)$$

   Then we have:

$$E\left[\int_0^t S_u du \backslash \mathcal{B}_h\right] = \sum_{k=0}^{M-1} S_{t_k} \int_{t_k}^{t_{k+1}} e^{\frac{u-t_k}{h}((r-d)h + \sigma(B_{t_{k+1}} - B_{t_k}) - \frac{\sigma^2}{2}(u - t_k))} du$$

   for which we give the following approximation from a Taylor expansion:

$$\widetilde{Y}_t^M = h \sum_{k=0}^{M-1} S_{t_k} \left(1 + \frac{(r-d)h}{2} + \sigma \frac{B_{t_{k+1}} - B_{t_k}}{2}\right)$$

3. **Scheme 3: Brownian bridge method**
   We express Y_t as:

$$Y_t = \sum_{k=0}^{M-1} \int_{t_k}^{t_{k+1}} S_{t_k} exp\left((r - d - \frac{\sigma^2}{2})(u - t_k) + \sigma(B_u - B_{t_k})\right) du$$

   With a Taylor expansion, we obtain:

$$\widetilde{Y}_t^M = h \sum_{k=0}^{M-1} S_{t_k} \left(1 + \frac{h}{2}(r - d - \frac{\sigma^2}{2}) - \sigma B_{t_k} + \frac{\sigma}{h} \int_{t_k}^{t_{k+1}} B_u du\right)$$

   For the simulation, we will use that:

$$\mathcal{L}\left(\int_{t_k}^{t_{k+1}} B_u du \backslash B_{t_k}, B_{t_{k+1}}\right) = \mathcal{N}\left(\frac{h}{2}(B_{t_k} + B_{t_{k+1}}); \frac{h^3}{6}\right)$$

Remarks:
For each scheme, we need to simulate M independent gaussian variables ($2M$ for the third scheme) to obtain the values of S_{t_k} and B_{t_k}.
- In the case of Monte Carlo simulation, pseudo-random generators generate successive terms that are independent so there is no particular problem with this point.
- However, in the case of Quasi-Monte Carlo simulation, successive terms generated by a low discrepancy sequence are not independent. Thus we can not consider a one-dimensional simulation. We need to use M (or $2M$)-dimensional sequences. See the implementation part for Quasi-Monte Carlo to verify maximum allowed dimension for each low discrepancy sequence.

Algorithm:

♣ /* Function SimulStockAndAverage */
Computation of $S_T(i)$ and the average $A(t_0,T)(i)$ according to the selected scheme for each step of the Monte Carlo simulation.
/*Initialisation*/
- /* Average and Stock Computation */
For the 3 schemes, we need values of B_{t_k} for each of the M times t_k defined on $[T_0,T]$.
We compute $B_{t_{k+1}}(i) = B_{t_k}(i) + \sqrt{h} g_k(i)$, where $g_k(i)$ are independant standard gaussian variables.
We have $S_{t_k}(i) = s \exp\left((r-d-\frac{\sigma^2}{2})(t_k - T_0)\right)\exp(\sigma B_{t_k}(i))$
And then we use the specific formula for each scheme to estimate the average $A(t_0,T)(i)$.
/* Scheme 1 : Rieman sums */
/* Scheme 2 : Trapezoidal method */
/* Simulation of M gaussian variables according to the generator type, that is Monte Carlo or Quasi Monte Carlo. */
For the two first schemes, we need to generate M independent gaussian variables. We keep them in a table.
Call to the appropriate function to generate independent standard gaussian variables. See the part about simulation of random variables for explanations on this point. We just recall that for a MC simulation, we use the Gauss-Abramovitz algorithm, and for a QMC simulation we use an inverse method and a M-dimensional low-discrepancy sequence.
/* Gaussian value from the table Gaussians */
At each step, we take the next gaussian value in the table.

/* Scheme 3 : Brownian Bridge method */

/* Simulation of $2M$ gaussian variables according to the generator type, that is Monte Carlo or Quasi Monte Carlo. */
Call to the appropriate function to generate independent standard gaussian variables. See the part about simulation of random variables for explanations on this point. We just recall that for a MC simulation, we use the Gauss-Abramovitz algorithm, and for a QMC simulation we use an inverse method and a $2M$-dimensional low-discrepancy sequence.
/* Gaussian value from the table Gaussians */
For the third scheme, we need to generate $2M$ independent gaussian variables.
In fact at each step, a second standard gaussian variable $g'_k(i)$ is required to simulate $\left(\frac{1}{h}\int_{t_k}^{t_{k+1}} B_u du \backslash B_{t_k}, B_{t_{k+1}}\right)$ as $\left(\frac{B_{t_k}+B_{t_{k+1}}}{2} + \sqrt{\frac{h}{6}} g'_k(i)\right)$.

- /*Stock*/
Final value $S_T(i)$.
- /*Average*/
Final value $A(t_0,T)(i)$.

♣ /* Function FloatingAsianStandardMC */
Main function to realize the Standard Monte Carlo simulation for an Asian option.
/* Value to construct the confidence interval */
For example if the confidence value is equal to 95% then the value z_α used to construct the confidence interval is 1.96. This parameter is taken into account only for MC simulation and not for QMC simulation.
/*Initialisation*/
/* Size of the random vector we need in the simulation */
For each of the three schemes, we need a vector of size M (or $2M$ for the third scheme) of independent gaussian variables to simulate the Brownian trajectory. In case of QMC simulation, it involves that we need a M or $2M$-dimensional low-discrepancy sequence.

- /*MC sampling*/
/* Test after initialization for the generator */
Test if the dimension of the simulation is compatible with the selected generator. In this case, we need a vector of size M or $2M$. Some low discrepancy sequences are not necessary adapted to a so large dimension.
/* Begin N iterations */

- /*Price*/

At the iteration i, we obtain $S_T(i)$ and the average $A(t_0,T)(i)$ from the function 'SimulStockAndAverage'. And we compute:

$$P(i) = \text{Payoff}(S_T(i), A'(t_0,T)(i) + K')$$

- /*Delta*/
Calculation of Delta δ_i with formula for a Call: if $P(i) > 0$

$$\delta(i) = \frac{S_T(i)}{s} - \frac{A'(T_0,T)(i)}{s}$$

/*Sum*/
Computation of the sums $\sum P_i$ and $\sum \delta_i$ for the mean price and the mean delta.
/*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:

$$P = \frac{1}{N} e^{-r(T-t)} \sum_{i=1}^{N} P(i)$$

The error estimator is σ_P with :

$$\sigma_P^2 = \frac{1}{N-1} \left(\frac{1}{N} e^{-2r(T-t)} \sum_{i=1}^{N} P(i)^2 - P^2 \right)$$

- /* Price Confidence Interval */

The confidence interval is given as:

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

with z_α computed from the confidence value.

- /*Delta*/

$$\delta = \frac{1}{N} e^{-r(T-t)} \sum_{i=1}^{N} \delta(i)$$

The error estimator is σ_δ with:

$$\sigma_\delta^2 = \frac{1}{N-1}\left(\frac{1}{N}e^{-2r(T-t)}\sum_{i=1}^{N}\delta(i)^2 - \delta^2\right)$$

- /* Delta Confidence Interval */

The confidence interval is given as:

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

with z_α computed from the confidence value.

Confidence intervals are always computed, but for a QMC simulation they don't work, thus they don't appear in the results.

References

[1] E.TEMAM. Monte carlo methods for asian options. *preprint*, 98-144 CERMICS, 1998. 3