Binomial Method for pricing path-dependent options

Black-Scholes model

$$\frac{dS_t}{S_t} = r dt + \sigma dB_t, \quad S_0 = s_0,$$

- Asian option.
- Lookback option
- Barrier option.

Asian options

The price of an European Asian option is given by

$$P(0, s, s) = E\left[e^{-rT} f(S_T, A_T) | S_0 = s, A_0 = s\right].$$

where A_T is the integral mean

$$A_T = \frac{1}{T}\int_0^T S_t$$

Payoff examples

- Fixed Asian Call: the payoff is $(A_T - K)_+$.
- Fixed Asian Put: the payoff is $(K - A_T)_+$.
- Floating Asian Call: the payoff is $(S_T - A_T)_+$.
- Floating Asian Put: the payoff is $(A_T - S_T)_+$.

Discrete approximation

Idea: approximate the integral mean with the arithmetic average.

$$E\left[e^{-rT} f(S^N, A^N)\right].$$

ove

$$A^N = \frac{1}{N+1}\sum_{n=0}^{N} S^n$$

Pure Binomial method

The average process $(A_i)_{0\le i\le n}$ is recursively computed by

$$A_{i+1} = \frac{(i+1)A_i + S_{i+1}}{i+2}, A_0 = s_0.$$

The bidimensional transition matrix is given by

$$\begin{array}{llcll} \text{up} & (x,y) & \longrightarrow & (xu, \frac{(n+1)y+xu}{n+2}) & \text{with probability } q \\ \text{down} & (x,y) & \longrightarrow & (xd, \frac{(n+1)y+xd}{n+2}) & \text{with probability } 1-q \end{array}$$

Backward induction

$$\begin{cases} v(N,x,y) = f(x,y) \\ v(n,x,y) = e^{-r\Delta T}\Big[qv(n+1,xu,\dfrac{(n+1)y+xu}{n+2}) + (1-q)v(n+1,xd,\dfrac{(n+1)y+xd}{n+2})\Big], \end{cases}$$

Rem In the American case we have to take in account the early exercise $(y-k)_+$

Complexity

The obtained tree is not recombining so that the algorithm is of exponential complexity. The evaluation of $v(0, s_0, s_0)$ requires time computations and memory requirement of the order $O(2^n)$ and this fact shows that the algorithm is completely unfeasible from a practical point of view. Oss Se $n = 50$, $2^{50} = 1.12 \times 10^{15}$.

Implementation of the algorithm

- Computation of 2^N averages at maturity $v(N, x, y) = f(x, y)$. Binary representation.

$$vp[i] = (vm[i] - K)_+, \quad i = 0 \ldots\ldots (2^N - 1)$$

- For all $n = (N - 1) \ldots\ldots\ldots 0$

$$vp[i] = e^{-r\Delta T}\Big(q \quad vp[2i+1] + (1-q) \quad vp[2i]\Big), \quad i = 0 \ldots\ldots\ldots (2^n - 1)$$

Hull-White algoritm

Idea: The main idea of this procedure is to restrict the range of the possible arithmetic averages to a set of some representative values. These values are selected in order to span all the possible values of the averages reachable at each node of the tree. The price is then computed by a backward induction procedure where the prices associated to the averages not included in the set of representative values, are obtained by some suitable interpolation methods.

$$A^N_{min} = s_0 \frac{1}{N+1} \sum_{k=0}^{N} d^k = s_0 \frac{1}{N+1} \frac{1-d^{N+1}}{1-d}$$

$$A^N_{max} = s_0 \frac{1}{N+1} \sum_{k=0}^{N} u^k = s_0 \frac{1}{N+1} \frac{u^{N+1}-1}{u-1}$$

In particular for every node (n, j)

$$A^{n,j}_{min} = \frac{1}{n+1} s_0 (1 + d + \ldots\ldots + d^{j-1} + d^j + d^j u + d^j u^2 + \ldots\ldots + d^j u^{n-j}) =$$

$$\frac{1}{n+1} s_0 \left[\frac{1-d^{j+1}}{1-d}\right] + \frac{1}{n+1} s_0 d^j \left[\frac{u^{n-j+1}-1}{u-1} - 1\right]$$

$$A^{n,j}_{max} = \frac{1}{n+1} s_0 \left[\frac{u^{n-j+1}-1}{u-1}\right] + \frac{1}{n+1} s_0 u^{n-j} \left[\frac{1-d^{j+1}}{1-d} - 1\right]$$

Hull-White algorithm

Discretization mesh of type

$$A^{k,n} = s_0 e^{mh}$$

where for a given h, the range of m values is selected to span the possible average at timestep n. Hull and White suggest that, to ensure accuracy for the algorithm, the value $h = 0.005$ is sufficient. Linear interpolation should be performed Complexity of order N^3.

FSG Method Forward Shooting Grid Method of Barraquand-Pudet for both Fixed or Floating Strike cases.

$S_j^n = s_0 e^{j\sigma\sqrt{h}}$, $A_k^n = s_0 e^{k\sigma\sqrt{h}}$ $\quad j,k = -n,...,n$ where $n = N,..,0$.
If at time n the bidimensional process is at (S_j^n, A_k^n), at time n+1 the process can reach in the upward and downward transition cases

$$\begin{array}{llcll} \text{up} & (S_j^n, A_k^n) & \rightarrow & (S_{j+1}^{n+1}, A_{k+}^{n+1}) & \text{with probability } p_u \\ \text{down} & (S_j^n, A_k^n) & \rightarrow & (S_{j-1}^{n+1}, A_{k-}^{n+1}) & \text{with probability } p_d \end{array}$$

$$(1) \qquad \begin{cases} C_{j,k}^N = \psi(S_j^N, A_k^N) = (A_k^N - K)_+ \\ C_{j,k}^n = \max\left(\psi(S_j^n, A_k^n), e^{-r\Delta T}\left[p_u C_{j+1,k+}^{n+1} + p_d C_{j-1,k-}^{n+1}\right)\right] \end{cases}$$

Remark 1 Time complexity of FSG algorithm is $O(N^3)$ and the convergence is slow
Remark 2 However, these techniques have some drawbacks related both to the precision of the approximations and to the convergence to the continuous value, as observed by Forsyth et al in Review of Derivatives Research 2002. Forsyth et al proved that a procedure of order $O(n^{\frac{7}{2}})$ is necessary in order to assure the convergence of these algorithms.

Lookback options

The price of an European lookback option is given by

$$P(0, s, s) = E\left[e^{-rT} f(S_T, M_T) | S_0 = s, M_0 = s\right].$$

where M_T

$$M_T = \max_{0 \le t \le T} S_t$$

$$m_T = \min_{0 \le t \le T} S_t$$

Payoff example:

- Fixed Lookback Call: the payoff is $(M_T - K)_+$.
- Fixed Lookback Put: the payoff is $(K - m_T)_+$.
- Floating Lookback Call: the payoff is $(S_T - m_T)_+$.
- Floating Lookback Put: the payoff is $(M_T - S_T)_+$.

Binomial method

$$E\left[e^{-rT}f(S^N, M^N)\right].$$

where

$$M^N = \max_{0\le n\le N} S^n$$

Pure Binomial method

The maximum process $(M_i)_{0 \leq i \leq n}$ can be computed recursively by

$$M^{n+1} = \max(M^n, S^{n+1}), M^0 = s_0$$

The bidimensional transition matrix is given by

$$\begin{array}{llcll} \text{up} & (x,y) & \longrightarrow & (xu, \max(xu,y)) & \text{with probability } q \\ \text{down} & (x,y) & \longrightarrow & (xd, y) & \text{with probability } 1-q \end{array}$$

Backward induction

$$\begin{cases} v(N,x,y) = f(x,y) \\ v(n,x,y) = e^{-r\Delta T}\Big[qv(n+1, xu, \max(xu,y)) + (1-q)v(n+1, xd, y)\Big], \end{cases}$$

Rem In the American case we have to take in account the early exercise $(y-k)_+$

Complexity

The evaluation of $v(0, s_0, s_0)$ requires a number of computations of order n^3.

Implementation of the algorithm

Number of different maximum at every node (n, j)

$$\begin{cases} j+1 & j \leq \frac{n}{2} \\ n-j+1 & j > \frac{n}{2}, \end{cases}$$

FSG Method Forward Shooting Grid Method of Barraquand-Pudet for both Fixed or Floating Strike cases.

$S_j^n = s_0 e^{j\sigma\sqrt{h}}, M_k^n = s_0 e^{k\sigma\sqrt{h}} \qquad j, k = -n, \ldots, n$ where $n = N, .., 0$.

If at time n the bidimensional process is at (S_j^n, M_k^n), at time n+1 the process can reach in the upward and downward transition cases

$$\begin{array}{lllll} \text{up} & (S_j^n, M_k^n) & \longrightarrow & (S_{j+1}^{n+1}, M_{k+}^{n+1}) & \text{with probability } p_u \\ \text{down} & (S_j^n, M_k^n) & \longrightarrow & (S_{j-1}^{n+1}, M_{k-}^{n+1}) & \text{with probability } p_d \end{array}$$

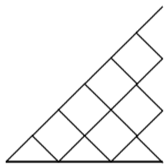
$$(2) \qquad \begin{cases} C_{j,k}^N = \psi(S_j^N, M_k^N) = (M_k^N - K)_+ \\ C_{j,k}^n = \max\left(\psi(S_j^n, M_k^n), e^{-r\Delta T}\left[p_u C_{j+1,k+}^{n+1} + p_d C_{j-1,k-}^{n+1}\right)\right] \end{cases}$$

Remark 1 Time complexity of FSG algorithm is $O(N^3)$ and the convergence is slow

Babbs method Babbs gives a very efficient and accurate solution to the problem with an one-dimensional tree method in the case of American floating strike Lookback options. The main idea is to use a change of "numeraire" approach using a reflected barrier.

$$Y_t = \frac{M_t}{S_t}$$

(3)
$$Y_{n+1} = \begin{cases} uY_n \quad with \quad p_u \\ \max(dY_n, 1) \quad with \quad p_d \end{cases}$$



Remark Time complexity of Babbs algorithm is $O(N^2)$ and the convergence with reflected barrier is very fast for the price

Barrier option

Te price of an down-out option

$$P(0,s) = E\left[e^{-rT} f(S_T) \mathbf{1}_{(\mathrm{m_T > L})} | \mathrm{S_0 = s}\right].$$

where m_T is

$$m_T = \min_{0 \le t \le T} S_t$$

Binomial method

$$E\left[e^{-rT}f(S^N)\mathbf{1}_{(\mathrm{S_k>L,k=0\ldots N})}\right].$$

Backward induction

$$\begin{cases} v(N,x) = f(x) & \text{if} \quad x > L \\ v(N,x) = 0 & \text{if} \quad x \leq L \\ v(n,x) = e^{-r\Delta T}\left[qv(n+1,xu) + (1-q)v(n+1,xd)\right] & \text{if} \quad x > L \\ v(n,x) = 0 & \text{if} \quad x \leq L \end{cases}$$

The classical CRR may be problematic when applied to barrier options because the convergence is very slow compared with that for standard vanilla options. The reason is clear: let n_L denote the index such that

$$S_0 d^{n_L} \geq L > S_0 d^{n_L+1}$$

Then the algorithm, N being fixed, yields the same result for any value of the barrier between $S_0 d^{n_L}$ and $S_0 d^{n_L+1}$.

Ritchken algorithm

Ritchken noted that the trinomial method, for the extra freedom in chosing the parameters λ, can be prefered to the binomial one. The main idea here is to choose the stretch parameter λ such that the barrier is hit exactly.

$$s_0 d^N = L$$

and then choose

$$\lambda = \frac{1}{N} \frac{\ln\left(\frac{S_0}{L}\right)}{\sigma\sqrt{\Delta T}}.$$