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

ap_luba

Output parameters:

- Price
- Delta

This routine is disigned to give either the put price or the call price. The put price is obtained from the call price of a symmetric option, by invertion of $K \leftrightarrow x$ and $r \leftrightarrow \delta$. This is the reason why almost all the functions are designed to compute the call option price.

Broadie and Detemple [1] have developped approximations for pricing standard american options. The lower and upper bound approximation is obtained in three steps :

The lower bound : They consider a european up and out call option with strike K, barrier L and rebate (L-K). They maximise over L the price of this option. Since the call up and out with rebate $(L-K)$ corresponds to exercise at the minimum of the hitting time of the boundary L and the matutity T, its price is smaller than the price of the american call option. Therefore, $C^l(x) = \max_L C(x, L)$ provides a lower bownd for the price of the american call.

The upper bound : To obtain their upperbound of the american call option price, Broadie and Detemple first calculate a lower bound of the optimal exercise boundary : L^*. They derive the upperbound $C^u(x)$ by replacing the optimal exercise boundary B by this lower bound L^* in the early exercise premium formula.

The approximation : From those two bounds, broadie and Detemple obtain the lower and upperbound approximation (luba) by applying a coefficient λ :

$$C_{luba}(x) = \lambda C^l(x) + (1-\lambda)C^u(x)$$

Computation of the lower bound

/*assign_var_temp*/

This function fixes some temporary variables widely used in this program.

/*assign_var_temp_L*/

It fixes temporary variables depending on L.

/*call_up_out*/

Returns $C(x, L)$, the price of an up and out european option with strike K, barrier L and rebate (L-K).

$$\begin{array}{rcl} C(x,L) & = & (L-K)[\lambda^{\frac{2\phi}{\sigma^2}} N(d_0) + \lambda^{\frac{2\phi}{\sigma^2}} N(d_0 + 2f\frac{\sqrt{T}}{\sigma})] \\ & & +x.e^{-\delta T}[N(d_1^-(L) - \sigma\sqrt{T}) - N(d_1^-(K) - \sigma\sqrt{T})] \\ & & -\lambda^{-2\frac{r-\delta}{\sigma^2}} L.e^{-\delta T}[N(d_1^+(L) - \sigma\sqrt{T}) - N(d_1^+(K) - \sigma\sqrt{T})] \\ & & -K.e^{-rT} \quad [N(d_1^-(L)) - N(d_1^-(K)) \\ & & \qquad -\lambda^{1-2\frac{r-\delta}{\sigma^2}}[N(d_1^+(L)) - N(d_1^+(K))]] \end{array}$$

Where :

$b = \delta - r + \frac{1}{2}\sigma^2$

$f = \sqrt{b^2 + 2r.\sigma^2}$

$\phi = \frac{1}{2}(b - f)$

$\alpha = \frac{1}{2}(b + f)$

$\lambda = \frac{x}{L}$

$d_0 = \frac{log(\lambda) - f(T)}{\sigma\sqrt{T}}$

$d_1^+(x) = \frac{log(\lambda) - log(L) + log(x) + b.T}{\sigma\sqrt{T}}$

$d_1^-(x) = \frac{-log(\lambda) - log(L) + log(x) + b.T}{\sigma\sqrt{T}}$

/*dCdL*/

Returns $\frac{\partial C(x,L)}{\partial L}$. This derivative value is necessary for the maximisation. this result is computed using a closed formula.

/*maximise_C*/

Return the value L_{max} for which $C(x, L)$ is a maximum. This result is obtained by a dichotomy research started on the interval $[x, 1000(x + K)]$.

/*call_lower_bound*/

Calculates the lower bound C^l applying the L_{max} value to the /*call_up_out*/ function :

$$C^l(x) = C(x, L_{max})$$

computation of the upperbound

/*D*/

$D(L,t) = \lim_{x \nearrow L} \frac{\partial C_t(x,L)}{\partial L}$ This function is necessary for the computation of L^*. $C_t(x,L)$ is the price of an up and out call option at current time t. This value is computed using the closed formula of $D(L,t)$.

/*zero_de_D*/

This function returns the L_t value for which $D(L,t) = 0$. The zero value is computed by a dichotomy search started on the interval $[K, 1000K]$.

/*Ls*/

Returns the value of the lower bound of the optimal exercise boundary L^* at time t.

/*d2*/

Secondary function necessary for the early exercise premium formula.

$$d_2(x, B_s, s) = \frac{log(\frac{x}{B_s}) + (r - \delta + \frac{1}{2}\sigma^2)(s)}{\sigma\sqrt{s}}$$

/*d3*/

Secondary function necessary for the early exercise premium formula.

$$d_3(x, B_s, s) = d_2(x, B_s, s) - \sigma\sqrt{s}$$

/*integr*/

This function evaluates the second member of the early exercise premium :

$$\int_{s=0}^{T} \begin{array}{l} [\delta.x.e^{-\delta.s}N(d_2(x, Ls, s)) \\ -r.K.e^{-r.s}N(d_3(x, Ls, s))]ds \end{array}$$

This integration is computed using a 10 points Gauss Legendre integration.

/*call_upper_bound*/

Returns the upper bound on the american option price. The price is computed with the early exercise premium formula :

$$C^u(x) = V(x, L^*) = c(x) + \int_{s=0}^{T} [\delta.x.e^{-\delta.s} N(d_2(x, L_s^*, s)) - r.K.e^{-r.s} N(d_3(x, L_s^*, s))] ds$$

With $c(x)$ the european call option price.

The λ coefficient

/*dCdx*/

Returns $\frac{\partial C(x,L)}{\partial x}$. This derivative value is necessary for the calculation of the coefficient. It is obtained by a numeric approximation :

$$\frac{\partial C(x,L)}{\partial x} = \frac{C(x + 10^{-4}, L) - C(x, L)}{10^{-4}}$$

/*coeff_upper*/

Return the λ coefficient as defined in Broadie and Detemple's formula.

/*call_low_up_approx*/

Returns the Lower and upper bound approximation :

$$C_{luba}(x) = \lambda C^l(x) + (1 - \lambda) C^u(x)$$

/*call_low_up_delta*/

Calculates the delta for the call option : $\frac{C_{luba}(x+10^{-5}) - C_{luba}(x)}{10^{-5}}$

/*put_low_up_delta*/

Calculates the delta : $\frac{P_{luba}(x+10^{-5}) - P_{luba}(x)}{10^{-5}}$

P_{luba} is the put price obtained from the price of the symmetric call option.

References

[1] M.BROADIE J.DETEMPLE. American option valuation : new bounds, approximations and a comparison of existing methods. *Review of financial studies, to appear*, 1995. 1