

# lim

## 1 The Limit Family

### Premia 22

For this family, the payoff is given by:

- For an Out option:

$\varphi(S_T)$  if  $S$  doesn't reach the upper boundary  $U(t)$  (resp. the lower boundary  $L(t)$ ) between the pricing date  $t_0$  and the maturity  $T$ .

$R(t)$ , which is paid at the time  $t$  when the upper (resp. lower) barrier is reached.

The payoff  $H_T$  in monetary unit of time of the maturity  $T$  may be written:

$$H_T = \varphi(S_T) \mathbf{1}(\tau^* > T) + e^{r(T-\tau^*)} R(\tau^*) \mathbf{1}(\tau^* \leq T)$$

under the assumption the instantaneous interest rate  $r$  is constant, where:

- for an up barrier

$$\tau^* = \inf \{u > t_0, S_u > U(u)\}$$

It is assumed that  $S_{t_0} < U(t_0)$ .

- for a down barrier

$$\tau^* = \inf \{u > t_0, S_u < L(u)\}$$

It is assumed that  $L(t_0) < S_{t_0}$ .

- For an In option:

The payoff  $H_T$  in monetary unit of time of the maturity  $T$  may be written:

$$H_T = \varphi(S_T) \mathbf{1}(\tau^* \leq T) + R(T) \mathbf{1}(\tau^* > T)$$

## 2 Code Implementation

```
#ifndef _LIM_H
#define _LIM_H

#include "optype.h"
#include "var.h"
#include "chk.h"
#include "numfunc.h"
#include "option.h"

#define TYPEOPT LIM

/*Limit Option// Single barrier*/

typedef struct TYPEOPT
{
    /* setable */
    VAR Maturity;
    VAR Limit; /*The Limit definition:
                * starting_date is in Limit->[0],
                * final_date is in Limit->Par[1],
                * frequency is in Limit->Par[2],
                * the value of the Limit in case of a constant limit is in Li
                * Parisian delay is in Limit->Par[4],
                * !!!!!WARNING!!!!!!
                * Wether the limit is backard/forward
                * should be tested in ChkOpt
                */
    VAR PayOff;
    VAR Rebate;
    /* non setable */
    VAR OutOrIn;
    VAR Parisian;
    VAR DownOrUp;
    VAR RebOrNo;
    VAR EuOrAm;
    VAR PartOrTot; /* Partial Or Total limit
                    * a partial limit is specified
                    * by starting_date, final_date
                    */
}
```

```
VAR ContOrDisc; /*Continuous or Discrete:
                * a discrete limit is specified
                * by frequency (regular sampling)
                */
VAR ConstLim; /*YES for constant limit*/

} TYPEOPT;

#endif
```