Source | Option Family

lim

1 The Limit Family

Premia 14

For this family, the payoff is given by:

- For an Out option:

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

  .$R\left(t\right)$, 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:

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

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

  - for an up barrier

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

    It is assumed that $S_{t_0} < U\left(t_0\right).$
  - for a down barrier

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

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

- For an In option:

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

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

2 Code Implementation

```
#ifndef  _LIM_H
#define _LIM_H

#include  "optype.h"
#include  "var.h"

#include  "chk.h"
#include  "numfunc.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 Limit->Par[3]
   * 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;

int OPT(Get)(int user,Planning *pt_plan,Option *opt, Model *mod);
int OPT(FGet)(char **InputFile,int user,Planning *pt_plan,Option *opt, Model *mo
int OPT(Show)(int user,Planning *pt_plan,Option *opt, Model *mod);
int OPT(Check)(int user,Planning *pt_plan,Option *opt);

#endif
```