

## Help

```

#ifndef _MC_LMM_GLASSERMAN_ZHAO_H
#define _MC_LMM_GLASSERMAN_ZHAO_H

#include "optype.h"
#include "numfunc.h"
#include "pnl/pnl_mathtools.h"
#include "pnl/pnl_random.h"
#include "pnl/pnl_cdf.h"

#include "math/lmm/lmm_libor.h"
#include "math/lmm/lmm_products.h"
#include "math/lmm/lmm_volatility.h"
#include "math/lmm/lmm_numerical.h"
#include "math/lmm/lmm_zero_bond.h"

/** "Arbitrage-Free Discretization Of Lognormal Forward
    Libor Model" by Glasserman and Zhao (2000)
* We consider a tenor structure  $0=T_0 < T_1 < \dots < T_N < T_{N+1}$ 
    equally spaced
* and Libor rates  $L(t, T_0), L(t, T_2), \dots, L(t, T_N)$  for a
    certain date  $t$ .  $L(\cdot, T_i)$  is set at  $T_i$  and payed at  $T_i + \text{tenor}$ 
.
* Convention: for  $t > T_i$   $L(t, T_i) = L(T_i, T_i)$ 
* Simulation can be done with the function "Sim_Libor_Glasserman"
    under two measure : Terminal measure and Spot measure.
* flag_numeraire=0 -> Terminal measure
* flag_numeraire=1 -> Spot measure
*/

void Sim_Libor_Glasserman(int start_index, int end_index,
    Libor *ptLOld, Volatility *ptVol, int generator, int NbrMCsimulation,
    int NbrStepPerTenor, int save_all_paths, PnlMat *
    LiborPathsMatrix, int save_brownien, PnlMat *BrownianMatrix
    Paths, int flag_numeraire);

int Sim_Libor_Glasserman_TerminalMeasure(int start_index,
    int end_index, Libor *ptLOld, Volatility *ptVol, int generator, int NbrM
    paths, PnlMat *LiborPathsMatrix, int save_brownien, PnlMat *Br

```

```

        ownianMatrixPaths);

double Swaption_Payoff_TerminalMeasure(Libor *ptL, Swaption
    *ptSwpt, NumFunc_1 *p);

int Sim_Libor_Glasserman_SpotMeasure(int start_index, int
    end_index, Libor *ptLOld, Volatility *ptVol, int generator,
    int NbrMCsimulation, int NbrStepPerTenor, int save_all_paths,
    PnlMat *LiborPathsMatrix, int save_brownien, PnlMat *BrownianMatrixPaths);

double Swaption_Payoff_SpotMeasure(Libor *ptL, Swaption *pt
    Swpt, NumFunc_1 *p);

double Swaption_Payoff_Discounted(Libor *ptL, Swaption *pt
    Swpt, NumFunc_1 *p, int flag_numeraire);

double european_swaption_ap_rebonato(double valuation_date,
    NumFunc_1 *p, Libor *ptLib, Volatility *ptVol, Swaption *
    ptSwpt);

double Numeraire(int i, Libor *ptLib_current, int flag_
    numeraire);

////////////////////////////////////

void MC_ExoticProduct_LongstaffSchwartz(char *CouponFlag,
    PnlVect *ContractParams, double *LS_Price, double first_exercise_date,
    double last_payment_date, double Nominal, int NbrMCsimulation, Libor *ptLib,
    Volatility *ptVol, int generator, int base_discount_rate, int flag_numeraire);
#endif

```

## References