

```
#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
 * 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  $t$ .  $L(\cdot,$ 
 * 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 meas
 * flag_numeraire=0 -> Terminal measure
 * flag_numeraire=1 -> Spot measure
 */

void Sim_Libor_Glasserman(int start_index, int end_index, Libor *ptLOld, Volatil

int Sim_Libor_Glasserman_TerminalMeasure(int start_index, int end_index, Libor *

double Swaption_Payoff_TerminalMeasure(Libor *ptL, Swaption *ptSwpt, NumFunc_1 *

int Sim_Libor_Glasserman_SpotMeasure(int start_index, int end_index, Libor *ptLO

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

double Swaption_Payoff_Discounted(Libor *ptL, Swaption *ptSwpt, NumFunc_1 *p, in

double european swaption ap rebonato(double valuation date, NumFunc 1 *p, Libor
```

```
double Numeraire(int i, Libor *ptLib_current, int flag_numeraire);
```

```
////////////////////////////////////
```

```
void MC_ExoticProduct_LongstaffSchwartz(char *CouponFlag, PnlVect *ContractParam  
#endif
```