Help

```
#ifndef _MC_LMM_GLASSERMAN_ZHAO_H
#define _MC_LMM_GLASSERMAN_ZHAO_H

#include "
href../../common/optype_h_src.pdfoptype.h"
#include "
href../../common/numfunc_h_src.pdfnumfunc.h"
#include "pnl/pnl_mathtools.h"
#include "pnl/pnl_random.h"
#include "pnl/pnl_cdf.h"

#include "
href../../common/math/lmm/lmm_libor_h_src.pdfmath/lmm/lmm_libor.h"
#include "
href../../common/math/lmm/lmm_products_h_src.pdfmath/lmm/lmm_products.h"
#include "
href../../common/math/lmm/lmm_volatility_h_src.pdfmath/lmm/lmm_volatility.h"
#include "
href../../common/math/lmm/lmm_numerical_h_src.pdfmath/lmm/lmm_numerical.h"
#include "
href../../common/math/lmm/lmm_zero_bond_h_src.pdfmath/lmm/lmm_zero_bond.h"


/** "Arbitrage-Free Discretization Of Lognormal Forward Libor Model" by Glasserm
* We consider a tenor structure 0=T_0 < T_1 < ... < T_N < T_N+1 equaly spaced
* and Libor rates L(t, T_0), L(t,T_2),..., L(t, T_N) for a certain date t. L(.,
* 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
```