```
   Help
#include "lmm1d_exoi.h"
#include "pnl/pnl_basis.h"
#include "math/mc_lmm_glassermanzhao.h"
#include "enums.h"

#if defined(PremiaCurrentVersion) && PremiaCurrentVersion <
     (2011+2) //The "#else" part of the code will be freely av
    ailable after the (year of creation of this file + 2)
static int CHK_OPT(MC_LongstaffSchwartz_CallableRangeAccru
    al)(void *Opt, void *Mod)
{
  return NONACTIVE;
}
int CALC(MC_LongstaffSchwartz_CallableRangeAccrual)(void *
    Opt,void *Mod,PricingMethod *Met)
{
  return AVAILABLE_IN_FULL_PREMIA;
}
#else

static int MC_CIF_LongstaffSchwartz(double l0, double sigma
    _const, int nb_factors, double last_payement_date, double
    first_exercise_date,  double Nominal, double FixedRate,
    double LowerRangeBound, double UpperRangeBound, double tenor,
    long NbrMCsimulation, int generator, int basis_name, int Dim
    Approx, int NbrStepPerTenor, int flag_numeraire, double *swa
    ption_price)
{
  Volatility *ptVol;
  Libor *ptLib;
  int init_mc;
  int Nbr_Maturities;
  char* CouponFlag = "CallableRangeAccrual";
  PnlVect* ContractParams = pnl_vect_create(3);

  LET(ContractParams, 0) = FixedRate;
  LET(ContractParams, 1) = LowerRangeBound;
  LET(ContractParams, 2) = UpperRangeBound;

  Nbr_Maturities = intapprox(last_payement_date/tenor);
  mallocLibor(&ptLib , Nbr_Maturities, tenor,l0);
  mallocVolatility(&ptVol , nb_factors, sigma_const);

  init_mc = pnl_rand_init(generator, nb_factors, NbrMCsimu
    lation);
  if (init_mc != OK) return init_mc;

  MC_ExoticProduct_LongstaffSchwartz(CouponFlag, ContractP
    arams, swaption_price, first_exercise_date, last_payement_
    date, Nominal, NbrMCsimulation, ptLib, ptVol, generator,
    basis_name, DimApprox, NbrStepPerTenor, flag_numeraire);

  freeLibor(&ptLib);
  freeVolatility(&ptVol);
  pnl_vect_free(&ContractParams);

  return init_mc;
}

int CALC(MC_LongstaffSchwartz_CallableRangeAccrual)(void *
    Opt, void *Mod, PricingMethod *Met)
{
  TYPEOPT* ptOpt=(TYPEOPT*)Opt;
  TYPEMOD* ptMod=(TYPEMOD*)Mod;

  return MC_CIF_LongstaffSchwartz(   ptMod->l0.Val.V_PDOUB
    LE,
                                     ptMod->Sigma.Val.V_PDO
    UBLE,
                                     ptMod->NbFactors.Val.
    V_ENUM.value,
                                     ptOpt->LastPaymentDate
    .Val.V_DATE-ptMod->T.Val.V_DATE,
                                     ptOpt->FirstExerciseD
    ate.Val.V_DATE-ptMod->T.Val.V_DATE,
                                     ptOpt->Nominal.Val.V_
    PDOUBLE,
                                     ptOpt->FixedRate.Val.
    V_PDOUBLE,
                                     ptOpt->LowerRangeBound
    .Val.V_PDOUBLE,
                                     ptOpt->UpperRangeBound
    .Val.V_PDOUBLE,
                                     ptOpt->ResetPeriod.Val
    .V_DATE,
                                     Met->Par[0].Val.V_LON
    G,
                                     Met->Par[1].Val.V_
    ENUM.value,
                                     Met->Par[2].Val.V_
    ENUM.value,
                                     Met->Par[3].Val.V_INT,
                                     Met->Par[4].Val.V_INT,
                                     Met->Par[5].Val.V_
    ENUM.value,
                                     &(Met->Res[0].Val.V_
    DOUBLE));
}

static int CHK_OPT(MC_LongstaffSchwartz_CallableRangeAccru
    al)(void *Opt, void *Mod)
{
  if ((strcmp(((Option*)Opt)->Name,"CallableRangeAccrual")=
    =0))
    return OK;
  else
    return WRONG;
}

#endif //PremiaCurrentVersion

static int MET(Init)(PricingMethod *Met,Option *Opt)
{
  if ( Met->init == 0)
    {
      Met->init=1;

      Met->Par[0].Val.V_LONG=50000;
      Met->Par[1].Val.V_ENUM.value=0;
      Met->Par[1].Val.V_ENUM.members=&PremiaEnumRNGs;
      Met->Par[2].Val.V_ENUM.value=0;
      Met->Par[2].Val.V_ENUM.members=&PremiaEnumBasis;
      Met->Par[3].Val.V_INT=10;
      Met->Par[4].Val.V_INT=1;
      Met->Par[5].Val.V_ENUM.value=0;
      Met->Par[5].Val.V_ENUM.members=&PremiaEnumAfd;

    }

  return OK;
}


PricingMethod MET(MC_LongstaffSchwartz_CallableRangeAccru
    al)=
{
  "MC_LongstaffSchwartz_Callable_Range_Accrual",
  {
    {"N Simulation",LONG,{100},ALLOW},
    {"RandomGenerator",ENUM,{100},ALLOW},
    {"Basis",ENUM,{100},ALLOW},
    {"Dimension Approximation",INT,{100},ALLOW},
    {"Nbr discretisation step per periode",INT,{100},ALLOW}
    ,
    {"Martingale Measure",ENUM,{100},ALLOW},
    {" ",PREMIA_NULLTYPE,{0},FORBID}},
  CALC(MC_LongstaffSchwartz_CallableRangeAccrual),
  {   {"Price",DOUBLE,{100},FORBID},
      {" ",PREMIA_NULLTYPE,{0},FORBID}},
  CHK_OPT(MC_LongstaffSchwartz_CallableRangeAccrual),
  CHK_ok,
  MET(Init)
};
```

References