```
   Help
#include "pnl/pnl_integration.h"

#include "math/lmm_stochvol_piterbarg/lmm_stochvol_piterbarg.h
    "
#include "math/lmm_stochvol_piterbarg/ap_averagingtech_lmmpit.h
    "
#include  "lmm_stochvol_piterbarg_stdi.h"
#include "enums.h"

#if defined(PremiaCurrentVersion) && PremiaCurrentVersion <
     (2010+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(AP_CaplFloor_LmmPit)(void *Opt, void *
    Mod)
{
  return NONACTIVE;
}
int CALC(AP_CaplFloor_LmmPit)(void *Opt,void *Mod,Pricing
    Method *Met)
{
  return AVAILABLE_IN_FULL_PREMIA;
}
#else

int func_premia_capfloor(int InitYieldCurve_flag, double R_
    flat, double Var_SpeedMeanReversion, double Var_Volatility,
                         double SkewsParams_a, double Skew
    sParams_b, double SkewsParams_c, double SkewsParams_d,
                         double VolsParams_a, double Vols
    Params_b, double VolsParams_c, double VolsParams_d,
                         double Tn, double Tm, double perio
    d, double cap_strike, double Nominal, int flag_capfloor,
    int FlagClosedFormula_in, double *price)
{
  int NbrVolFactors=1;
  StructLmmPiterbarg *LmmPiterbarg;
  PnlMat *SkewsParams = pnl_mat_create_from_list(4, 1, Skew
    sParams_a, SkewsParams_b, SkewsParams_c, SkewsParams_d);
  PnlMat *VolsParams = pnl_mat_create_from_list(4, 1, Vols
    Params_a, VolsParams_b, VolsParams_c, VolsParams_d);
  LmmPiterbarg = SetLmmPiterbarg(InitYieldCurve_flag, R_fla
    t, period, Tm, Var_SpeedMeanReversion, Var_Volatility, NbrV
    olFactors, SkewsParams, VolsParams);


  *price = cf_lmm_stochvol_piterbarg_capfloor(LmmPiterbarg,
     Tn, Tm, period, cap_strike, Nominal, flag_capfloor, FlagC
    losedFormula_in);

  FreeLmmPiterbarg(&LmmPiterbarg);
  pnl_mat_free(&SkewsParams);
  pnl_mat_free(&VolsParams);

  return OK;
}


int CALC(AP_CaplFloor_LmmPit)(void *Opt,void *Mod,Pricing
    Method *Met)
{
  TYPEOPT* ptOpt=(TYPEOPT*)Opt;
  TYPEMOD* ptMod=(TYPEMOD*)Mod;

  int flag_capfloor = 1;
  if (((ptOpt->PayOff.Val.V_NUMFUNC_1)->Compute)==&Call)
    {
      flag_capfloor = -1;
    }

  return func_premia_capfloor(    ptMod->Flag_InitialYield
    Curve.Val.V_INT,

                                  MOD(GetYield)(ptMod),
                                  ptMod->Var_SpeedMeanReversion.V
    al.V_PDOUBLE,
                                  ptMod->Var_Volatility.Val
    .V_PDOUBLE,
                                  ptMod->SkewsParams_a.Val.
    V_PDOUBLE,
                                  ptMod->SkewsParams_b.Val.
    V_PDOUBLE,
                                  ptMod->SkewsParams_c.Val.
    V_PDOUBLE,
                                  ptMod->SkewsParams_d.Val.
    V_PDOUBLE,
                                  ptMod->VolsParams_a.Val.
    V_PDOUBLE,
                                  ptMod->VolsParams_b.Val.
    V_PDOUBLE,
                                  ptMod->VolsParams_c.Val.
    V_PDOUBLE,
                                  ptMod->VolsParams_d.Val.
    V_PDOUBLE,

                                  ptOpt->FirstResetDate.Val
    .V_DATE-ptMod->T.Val.V_DATE,
                                  ptOpt->BMaturity.Val.V_DA
    TE-ptMod->T.Val.V_DATE,
                                  ptOpt->ResetPeriod.Val.V_
    DATE,
                                  ptOpt->FixedRate.Val.V_
    PDOUBLE,
                                  ptOpt->Nominal.Val.V_PDO
    UBLE,
                                  flag_capfloor,
                                  Met->Par[0].Val.V_ENUM.
    value,
                                  &(Met->Res[0].Val.V_
    DOUBLE));

}

static int CHK_OPT(AP_CaplFloor_LmmPit)(void *Opt, void *
    Mod)
{
  if ((strcmp(((Option*)Opt)->Name,"Cap")==0) || (strcmp(((
    Option*)Opt)->Name,"Floor")==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_ENUM.value=0;
      Met->Par[0].Val.V_ENUM.members=&PremiaEnumAveraging;
    }

  return OK;
}


PricingMethod MET(AP_CaplFloor_LmmPit)=
{
  "AP_CapFloor_LmmPit",
  {
    {"Averaging Vol",ENUM,{100},ALLOW},
    {" ",PREMIA_NULLTYPE,{0},FORBID}},
  CALC(AP_CaplFloor_LmmPit),
  {{"Price",DOUBLE,{100},FORBID},{" ",PREMIA_NULLTYPE,{0},
    FORBID}},
  CHK_OPT(AP_CaplFloor_LmmPit),
  CHK_ok,
  MET(Init)
} ;
```

References