Help

```
#include <stdlib.h>
#include <
href../../../common/math/cdo/cdo_math_h_src.pdfmath.h>
#include "pnl/pnl_vector.h"
#include "pnl/pnl_fft.h"
#include "
href../../../common/math/wienerhopf_h_src.pdfmath/wienerhopf.h"
#include "
href../../../mod/cgmy1d/cgmy1d_pad/cgmy1d_pad_h_src.pdfcgmy1d_pad.h"
#include "
href../../../common/enums_h_src.pdfenums.h"
#include "pnl/pnl_cdf.h"
#include"pnl/pnl_random.h"
#include"pnl/pnl_specfun.h"

#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2020+2) //The "#els
static int CHK_OPT(MC_WH_Lookback)(void *Opt, void *Mod)
{
  return NONACTIVE;
}
int CALC(MC_WH_Lookback)(void *Opt, void *Mod, PricingMethod *Met)
{
  return AVAILABLE_IN_FULL_PREMIA;
}
#else
/*////////////////////////////////////////////////////*/

//=============================================================================


static int mc_wienerhopf_lookback(double s_maxmin, NumFunc_2 *P, double Spot, do
                                        double r, double divid, double C, double
   int generator, int n_paths, double *ptprice, double *priceError)


{
  double lp1, lm1, ptprice1, priceError1;
  int ifCall;
  /* if (M < 2 || G <= 1 || Y >= 2 || Y == 0) */
```

```
  /*   { */
  /*     fprintf(stderr, "Invalid parameters. We must have M>=2, G>1, 0<Y<2\ n")
  /*   } */

//CALL
  if ((P->Compute) == &Call_OverSpot2 || (P->Compute) == &Call_StrikeSpot2)
    {
      ifCall = 1;
    }
//PUT
  if ((P->Compute) == &Put_OverSpot2 || (P->Compute) == &Put_StrikeSpot2)
    {
      ifCall = 0;
    }

  lm1 = -M;
  lp1 = G;


  if ((P->Compute) == &Call_StrikeSpot2 || (P->Compute) == &Put_StrikeSpot2)
   TSL_lookbackfloat_WHMC(ifCall, Spot, s_maxmin,
T, r, divid,  lm1,  lp1, Y, Y, C, C, h, er, generator, n_paths, &ptprice1, &pric
  else
   TSL_lookbackfixed_WHMC(ifCall, Spot, Strike,s_maxmin,
T, r, divid,  lm1,  lp1, Y, Y, C, C, h, er, generator, n_paths, &ptprice1, &pric

 //Price
  *ptprice = ptprice1;
  //MC Price Error
  *priceError = priceError1;

  return OK;
}

//=============================================================================
int CALC(MC_WH_Lookback)(void *Opt, void *Mod, PricingMethod *Met)
{
  TYPEOPT *ptOpt = (TYPEOPT *)Opt;
  TYPEMOD *ptMod = (TYPEMOD *)Mod;
  double r, divid;
```

```
  r = log(1. + ptMod->R.Val.V_DOUBLE / 100.);
  divid = log(1. + ptMod->Divid.Val.V_DOUBLE / 100.);

  return  mc_wienerhopf_lookback((ptOpt->PathDep.Val.V_NUMFUNC_2)->Par[4].Val.V_
                                      ptOpt->PayOff.Val.V_NUMFUNC_2, ptMod->S0.V
                                      r, divid, ptMod->C.Val.V_PDOUBLE, ptMod->G
                                      Met->Par[1].Val.V_SPDOUBLE,Met->Par[0].Val
  Met->Par[2].Val.V_PINT, Met->Par[3].Val.V_PINT,///check int parameters
                                  &(Met->Res[0].Val.V_DOUBLE), &(Met->Res[1].Val
}

static int CHK_OPT(MC_WH_Lookback)(void *Opt, void *Mod)
{
  if ((strcmp(((Option *)Opt)->Name, "LookBackCallFixedEuro") == 0) || (strcmp((
    return OK;
  return WRONG;
}

#endif //PremiaCurrentVersion
static int MET(Init)(PricingMethod *Met, Option *Opt)
{
   if (Met->init == 0)
    {
      Met->HelpFilenameHint = "MC_WH_cgmy_Lookback";
      Met->Par[0].Val.V_PDOUBLE=2.0;
      Met->Par[1].Val.V_PDOUBLE=0.001;
  Met->Par[2].Val.V_ENUM.value = 0;
      Met->Par[2].Val.V_ENUM.members = &PremiaEnumMCRNGs;
      Met->Par[3].Val.V_PINT=10000;  ///check int parameters

    }
  return OK;
}

PricingMethod MET(MC_WH_Lookback) =
{
  "MC_WH_CGMY",
  { {"Scale of logprice range", DOUBLE, {100}, ALLOW},
    {"Space Discretization Step", DOUBLE, {500}, ALLOW},
{"RandomGenerator", ENUM, {100}, ALLOW},
    {"N iterations",INT,{100},ALLOW},        ///check int parameters
```

```
    {" ", PREMIA_NULLTYPE, {0}, FORBID}
  },
  CALC(MC_WH_Lookback),
  { {"Price", DOUBLE, {100}, FORBID},
    {"Price Error", DOUBLE, {100}, FORBID},
    {" ", PREMIA_NULLTYPE, {0}, FORBID}
  },
  CHK_OPT(MC_WH_Lookback),
  CHK_split,
  MET(Init)
};
```