```
   Help
#include  "libor_affine_cir1d_stdi.h"
#include "math/libor_affine_model/libor_affine_framework.h"
#include "math/libor_affine_model/libor_affine_pricing.h"
#include "math/libor_affine_model/libor_affine_models.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(CF_LibAffCir1d_Fourier_Swaption)(void *
    Opt, void *Mod)
{
    return NONACTIVE;
}
int CALC(CF_LibAffCir1d_Fourier_Swaption)(void *Opt,void *
    Mod,PricingMethod *Met)
{
    return AVAILABLE_IN_FULL_PREMIA;
}
#else


static int cf_swaption_fourier_libaff_cir1d(int InitYield
    Curve_flag, double R_flat, double x0, double lambda, double
    theta, double eta, double swaption_start, double swaption_
    end, double swaption_period, double swaption_strike, double
    swaption_nominal, int swaption_payer_receiver, double *swapt
    ion_price)
{
    StructLiborAffine LiborAffine;
    ZCMarketData ZCMarket;
    PnlVect *ModelParams=pnl_vect_create(4);

    LET(ModelParams, 0) = x0;
    LET(ModelParams, 1) = lambda;
    LET(ModelParams, 2) = theta;
    LET(ModelParams, 3) = eta;

    SetInitYieldCurve(InitYieldCurve_flag, R_flat, &ZCMarke
    t);
```

```
    CreateStructLiborAffine(&LiborAffine, &ZCMarket, swapt
    ion_start, swaption_end, swaption_period, ModelParams, &ph
    i_psi_cir1d, &MaxMgfArg_cir1d);

    *swaption_price = cf_swaption_fourier_libaff(&LiborAffi
    ne, swaption_start, swaption_end, swaption_period, swaptio
    n_strike,  swaption_nominal, swaption_payer_receiver);

    FreeStructLiborAffine(&LiborAffine);

    return OK;
}


///********************************************** PREMIA
    FUNCTIONS ***********************************************///

int CALC(CF_LibAffCir1d_Fourier_Swaption)(void *Opt,void *
    Mod,PricingMethod *Met)
{
    TYPEOPT* ptOpt=(TYPEOPT*)Opt;
    TYPEMOD* ptMod=(TYPEMOD*)Mod;

    int swaption_payer_receiver = (((ptOpt->PayOff.Val.V_
    NUMFUNC_1)->Compute)==&Call);

    return  cf_swaption_fourier_libaff_cir1d(   ptMod->fla
    t_flag.Val.V_INT,
                                                MOD(GetYi
    eld)(ptMod),
                                                ptMod->x0.
    Val.V_DOUBLE,
                                                ptMod->lam
    bda.Val.V_PDOUBLE,
                                                ptMod->thet
    a.Val.V_DOUBLE,
                                                ptMod->eta.
    Val.V_PDOUBLE,
                                                ptOpt->OM
    aturity.Val.V_DATE-ptMod->T.Val.V_DATE,
                                                ptOpt->BM
```

```
    aturity.Val.V_DATE-ptMod->T.Val.V_DATE,
                                                ptOpt->Res
    etPeriod.Val.V_DATE,
                                                ptOpt->Fix
    edRate.Val.V_PDOUBLE,
                                                ptOpt->Nom
    inal.Val.V_PDOUBLE,
                                                swaption_
    payer_receiver,
                                                &(Met->Res[
    0].Val.V_DOUBLE));
}
static int CHK_OPT(CF_LibAffCir1d_Fourier_Swaption)(void *
    Opt, void *Mod)
{
    if ((strcmp(((Option*)Opt)->Name,"PayerSwaption")==0) |
    | (strcmp(((Option*)Opt)->Name,"ReceiverSwaption")==0))
        return OK;
    else
        return WRONG;
}
#endif //PremiaCurrentVersion

static int MET(Init)(PricingMethod *Met,Option *Opt)
{
    if ( Met->init == 0)
    {
      Met->init=1;
       Met->HelpFilenameHint = "    cf_libor_affine_cir1d_swaption_fourier";
    }
    return OK;
}


PricingMethod MET(CF_LibAffCir1d_Fourier_Swaption)=
{
    "CF_LibAffCir1d_Fourier_Swaption",
    {{" ",PREMIA_NULLTYPE,{0},FORBID}},
    CALC(CF_LibAffCir1d_Fourier_Swaption),
    {{"Price",DOUBLE,{100},FORBID},{" ",PREMIA_NULLTYPE,{0}
    ,FORBID}},
```

```
    CHK_OPT(CF_LibAffCir1d_Fourier_Swaption),
    CHK_ok,
    MET(Init)
} ;
```

References