

Help

```

#include "libor_affine_cir1d_std.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 available after the (year of creation of this file + 2)
static int CHK_OPT(CF_LibAffCir1d_Fourier_CapFloor)(void *
    Opt, void *Mod)
{
    return NONACTIVE;
}
int CALC(CF_LibAffCir1d_Fourier_CapFloor)(void *Opt,void *
    Mod,PricingMethod *Met)
{
    return AVAILABLE_IN_FULL_PREMIA;
}
#else

static int cf_capfloor_fourier_libaff_cir1d(int InitYield
    Curve_flag, double R_flat, double x0, double lambda, double
    theta, double eta, double cap_start, double cap_end,
    double cap_period, double cap_strike, double cap_nominal, int cap_floor,
{
    double caplet_Tk1, caplet_Tk2, caplet_price;
    int i, nb_payment = intapprox((cap_end-cap_start)/cap_
    period);
    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, cap_
start, cap_end, cap_period, ModelParams, &phi_psi_cir1d, &Max
MgfArg_cir1d);

caplet_Tk1 = cap_start;
caplet_Tk2 = caplet_Tk1+cap_period;
*cap_price = 0.;

for(i=0; i<nb_payement; i++)
{
    caplet_price = cf_swaption_fourier_libaff(&LiborA
ffine, caplet_Tk1, caplet_Tk2, cap_period, cap_strike, cap_
nominal, cap_floor);
    *cap_price += caplet_price;

    caplet_Tk1 += cap_period;
    caplet_Tk2 += cap_period;
}

FreeStructLiborAffine(&LiborAffine);

return OK;
}

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

int CALC(CF_LibAffCir1d_Fourier_CapFloor)(void *Opt,void *
Mod,PricingMethod *Met)
{
    TYPEOPT* ptOpt=(TYPEOPT*)Opt;
    TYPEMOD* ptMod=(TYPEMOD*)Mod;

    int cap_floor = (((ptOpt->PayOff.Val.V_NUMFUNC_1)->
Compute)==&Call);

    return cf_capfloor_fourier_libaff_cir1d( ptMod->fla
t_flag.Val.V_INT,
MOD(GetYi
eld)(ptMod),

```

```

Val.V_DOUBLE,
bda.Val.V_PDOUBLE,
a.Val.V_DOUBLE,
Val.V_PDOUBLE,
stResetDate.Val.V_DATE-ptMod->T.Val.V_DATE,
aturity.Val.V_DATE-ptMod->T.Val.V_DATE,
etPeriod.Val.V_DATE,
edRate.Val.V_PDOUBLE,
inal.Val.V_PDOUBLE,

ptMod->x0.
ptMod->lam
ptMod->thet
ptMod->eta.
ptOpt->Fir
ptOpt->BM
ptOpt->Res
ptOpt->Fix
ptOpt->Nom
cap_floor,
&(Met->Res[

0].Val.V_DOUBLE));
}
static int CHK_OPT(CF_LibAffCir1d_Fourier_CapFloor)(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->HelpFilenameHint = "    cf_libor_affine_cir1d_capfloor_fourier";
    }
}

```

```
    return OK;
}
```

```
PricingMethod MET(CF_LibAffCir1d_Fourier_CapFloor)=
{
    "CF_LibAffCir1d_Fourier_CapFloor",
    {{ " ",PREMIA_NULLTYPE,{0},FORBID}},
    CALC(CF_LibAffCir1d_Fourier_CapFloor),
    {{ "Price",DOUBLE,{100},FORBID},{ " ",PREMIA_NULLTYPE,{0}
    ,FORBID}},
    CHK_OPT(CF_LibAffCir1d_Fourier_CapFloor),
    CHK_ok,
    MET(Init)
} ;
```

References