

[Help](#)

```
#include "
href../../mod/hullwhite2d/hullwhite2d_std/hullwhite2d_std_h_src.pdfhullwhit
#include "
href../../mod/hullwhite2d/hullwhite2d_std/hullwhite2d_includes_h_src.pdfhull
#include "pnl/pnl_cdf.h"

//The "#else" part of the code will be freely available after the (year of creat
#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2009+2)
int CALC(CF_ZBPUTHW2D)(void *Opt, void *Mod, PricingMethod *Met)
{
    return AVAILABLE_IN_FULL_PREMIA;
}
static int CHK_OPT(CF_ZBPUTHW2D)(void *Opt, void *Mod)
{
    return NONACTIVE;
}
#else

// Volatility of an european option on a ZC bond P(T,S)
static double cf_ZB0volatility2d(double a, double sigma1, double b, double sigma2)
{
    double sigma_p;
    double exp_atT, exp_btT, exp_aTS, exp_bTS;
    double sigma3, eta, rhoG2;

    sigma3 = sqrt(SQR(sigma1) + SQR(sigma2) / ((b - a) * (b - a)) + 2 * rho * sigma1 * sigma2);
    eta = sigma2 / (a - b);
    rhoG2 = (sigma1 * rho - eta) / sigma3 ;

    exp_atT = exp(-a * (T - t));
    exp_btT = exp(-b * (T - t));

    exp_aTS = exp(-a * (S - T));
    exp_bTS = exp(-b * (S - T));

    //B_TS = (1 - exp_aTS) / a;
    //U = (exp_aTS - 1) * exp_atT / (a * (a - b)); // (1/exp_aS - 1/exp_aT) / (a * (a - b));
    //V = (exp_bTS - 1) * exp_btT / (b * (a - b)); // (1/exp_bS - 1/exp_bT) / (b * (a - b));
```

```

sigma_p = SQR(sigma3) * SQR(1 - exp_aTS) * (1 - SQR(exp_atT)) / (2 * CUB(a))

sigma_p += SQR(eta) * SQR(1 - exp_bTS) * (1 - SQR(exp_btT)) / (2 * CUB(b));

sigma_p += 2 * rhoG2 * sigma3 * eta * (1 - exp_aTS) * (1 - exp_bTS) * (1 - exp

sigma_p = sqrt(sigma_p);

return sigma_p;
}

```

```

static int cf_zbput2d(int flat_flag, double r, char *curve, double u, double a,
{
    double PtS, PtT, X;
    double h, sigma_p;

    ZCMarketData ZCMarket;
    /* Flag to decide to read or not ZC bond datas in "initialyields.dat" */
    /* If P(0,T) not read then P(0,T)=exp(-r0*T) */
    if (flat_flag == 0)
    {
        ZCMarket.FlatOrMarket = 0;
        ZCMarket.Rate = r;
    }

    else
    {
        ZCMarket.FlatOrMarket = 1;
        ZCMarket.filename = curve;
        ReadMarketData(&ZCMarket);

        if (S > GET(ZCMarket.tm, ZCMarket.Nvalue - 1))
        {
            printf("\ nError : time bigger than the last time value entered in ini
            exit(EXIT_FAILURE);
        }
    }

    sigma_p = cf_ZB0volatility2d(a, sigma1, b, sigma2, rho, 0, T, S);

```

```

X = p->Par[0].Val.V_DOUBLE; // Strike

PtT = cf_hw2d_zcb(&ZCMarket, a, sigma1, b, sigma2, rho, 0, r, u, T);

PtS = cf_hw2d_zcb(&ZCMarket, a, sigma1, b, sigma2, rho, 0, r, u, S);

h = log(PtS / (PtT * X)) / sigma_p + 0.5 * sigma_p ;

*price = PtS * (cdf_nor(h) - 1) - X * PtT * (cdf_nor(h - sigma_p) - 1);

DeleteZCMarketData(&ZCMarket);

return OK;
}

int CALC(CF_ZBPUTHW2D)(void *Opt, void *Mod, PricingMethod *Met)
{
    TYPEOPT *ptOpt = (TYPEOPT *)Opt;
    TYPEMOD *ptMod = (TYPEMOD *)Mod;

    return cf_zbput2d(ptMod->flat_flag.Val.V_INT,
                      MOD(GetYield)(ptMod),
                      MOD(GetCurve)(ptMod),
                      ptMod->InitialYieldsu.Val.V_PDOUBLE,
                      ptMod->aR.Val.V_DOUBLE,
                      ptMod->SigmaR.Val.V_PDOUBLE,
                      ptMod->bu.Val.V_DOUBLE,
                      ptMod->Sigmau.Val.V_PDOUBLE,
                      ptMod->Rho.Val.V_PDOUBLE,
                      ptOpt->BMaturity.Val.V_DATE - ptMod->T.Val.V_DATE,
                      ptOpt->OMaturity.Val.V_DATE - ptMod->T.Val.V_DATE,
                      ptOpt->PayOff.Val.V_NUMFUNC_1,
                      &(Met->Res[0].Val.V_DOUBLE));
}

static int CHK_OPT(CF_ZBPUTHW2D)(void *Opt, void *Mod)
{
    return strcmp(((Option *)Opt)->Name, "ZeroCouponPutBondEuro");
}

#endif //PremiaCurrentVersion

```

```

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

    return OK;
}

PricingMethod MET(CF_ZBPUTHW2D) =
{
    "CF_ZBPutEuroHW2D",
    {{ " ", PREMIA_NULLTYPE, {0}, FORBID}},
    CALC(CF_ZBPUTHW2D),
    {{ "Price", DOUBLE, {100}, FORBID} /*, {"Delta", DOUBLE, {100}, FORBID} */ , { " ", PR
    CHK_OPT(CF_ZBPUTHW2D),
    CHK_ok,
    MET(Init)
} ;

```