

[Help](#)

```

#include "cirpp1d_std.h"

#ifdef PREMIA_CURRENT_VERSION
static int CHK_OPT(CF_ZCBond)(void *Opt, void *Mod)
{
    return NONACTIVE;
}
int CALC(CF_ZCBond)(void *Opt, void *Mod, PricingMethod *Met)
{
    return AVAILABLE_IN_FULL_PREMIA;
}
#else

/*Zero Coupon Bond*/
static double zcbond(double rcc, double a, double b, double sigma, double t, double T)
{
    if (t == 0)
    {
        return BondPrice(T, ZCMarket);
    }
    else
    {
        double h, A, B, At, AT, shift, c;
        double f0_t, P0_t, P0_T, P0_t_plus, P0_t_minus;

        P0_t = BondPrice(t, ZCMarket);
        P0_T = BondPrice(T, ZCMarket);

        /*Computation of Forward rate*/
        P0_t_plus = BondPrice(t * (1. + INC), ZCMarket);
        P0_t_minus = BondPrice(t * (1. - INC), ZCMarket);
        f0_t = -(log(P0_t_plus) - log(P0_t_minus)) / (2.*t * INC);

        /*A,B coefficient*/
        h = sqrt(SQR(a) + 2.*SQR(sigma));
        B = 2.*(exp(h * (T - t)) - 1.) / (2.*h + (a + h) * (exp(h * (T - t)) - 1.));
        A = pow(h * exp(0.5 * (a + h) * (T - t)) / (h + 0.5 * (a + h) * (exp(h * (T - t)) - 1.)));
        At = pow(h * exp(0.5 * (a + h) * (t)) / (h + 0.5 * (a + h) * (exp(h * (t)) - 1.)));
        AT = pow(h * exp(0.5 * (a + h) * (T)) / (h + 0.5 * (a + h) * (exp(h * (T)) - 1.)));
    }
}

```

```

        c = sqrt(a * a + 2 * sigma * sigma);

        shift = (f0_t - 2 * a * b * (exp(t * c) - 1) / (2 * c + (a + c) * (exp(t * c) - 1)));

        A = A * (P0_T * At) / (AT * P0_t) * exp(B * shift);

        /*Price*/
        return A * exp(-B * rcc);
    }
}

static int zcb_cirpp1d(double flat_flag, double a, double b, double t, double sigma, double rcc)
{
    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 = rcc;
    }

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

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

    /*Price*/
    *price = zcbond(rcc, a, b, sigma, t, T, &ZCMarket);

    return OK;
}

```

```

}

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

    return zcb_cirpp1d(ptMod->flat_flag.Val.V_INT, ptMod->a.Val.V_DOUBLE, ptMod->b,
        ptMod->Sigma.Val.V_PDOUBLE, MOD(GetYield)(ptMod),
        MOD(GetCurve)(ptMod), ptOpt->BMaturity.Val.V_DATE,
        &(Met->Res[0].Val.V_DOUBLE)/*,&(Met->Res[1].Val.V_DOUBLE)*/);
}

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

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

    return OK;
}

PricingMethod MET(CF_ZCBond) =
{
    "CF_Cirpp1d_ZCBond",
    {" ", PREMIA_NULLTYPE, {0}, FORBID}},
    CALC(CF_ZCBond),
    {"Price", DOUBLE, {100}, FORBID}, {" ", PREMIA_NULLTYPE, {0}, FORBID}},
    CHK_OPT(CF_ZCBond),
    CHK_ok,
    MET(Init)
} ;

```

