

[Help](#)

```
#include "
href../../mod/bs1d/bs1d_pad/bs1d_pad_h_src.pdfbs1d_pad.h"

int Fixed_PutLookback_ConzeWiswanathan(double s, double s_min, double k, double
                                     double r, double divid, double sigma, dou
{
    double b, sigmasqrt, a1, a2, esp, disc;

    if (s_min > s)
    {
        *ptprice = 0.;
        *ptdelta = 0.;
    }
    else
    {
        b = r - divid;
        sigmasqrt = sigma * sqrt(t);
        esp = 2.*b / SQR(sigma);
        disc = exp(-r * t);

        if (k < s_min)
        {
            a1 = (log(s / k) + (b + SQR(sigma) / 2.) * t) / sigmasqrt;
            a2 = a1 - sigmasqrt;
            if (b == 0)
            {
                *ptprice = k * disc * cdf_nor(-a2) - s * disc * cdf_nor(-a1) +
                    s * disc * sigmasqrt * pnl_normal_density(a1) - s * dis

                *ptdelta = -(k / s) * disc * pnl_normal_density(a2) / sigmasqrt +
                    - disc * cdf_nor(-a1) * (2. + SQR(sigma) * t / 2. + log

            }
            else
            {
                *ptprice = k * exp(-r * t) * cdf_nor(-a2) - s * exp(-divid * t) *
                    s * exp(-r * t) * (SQR(sigma) / (2.*b)) *
                    (pow(s / k, -esp) * cdf_nor(-a1 + (2.*b / sigma) * sqrt

                *ptdelta = exp(-divid * t) * cdf_nor(a1) * (1. + SQR(sigma) / (2.*
```

```

        exp(-divid * t) * pnl_normal_density(a1) / (sigma * sqrt(s)) *
        exp(-r * t) * (k / s) * pnl_normal_density(a2) / sigmasqrt +
        exp(-r * t) * pow(s / k, -esp) * cdf_nor(-a1 + 2.*(b / sigma) *
        exp(-divid * t) * (SQR(sigma) / (2 * b) + 1.);
    }
}
else
{
    a1 = (log(s / s_min) + (b + SQR(sigma) / 2.) * t) / sigmasqrt;
    a2 = a1 - sigmasqrt;
    if (b == 0)
    {
        *ptprice = disc * (k - s_min) - s * disc * cdf_nor(-a1) + s_min *
            s * disc * sigmasqrt * pnl_normal_density(a1) - s * disc *
            sigmasqrt * cdf_nor(a2);

        *ptdelta = -(s_min / s) * disc * pnl_normal_density(a2) / sigmasqrt -
            disc * cdf_nor(-a1) * (2. + SQR(sigma) * t / 2. + log(s / s_min));
    }
    else
    {
        *ptprice = exp(-r * t) * (k - s_min) - s * exp(-divid * t) * cdf_nor(-a1) +
            s * exp(-r * t) * (SQR(sigma) / (2.*b)) *
            (pow(s / s_min, -esp) * cdf_nor(-a1 + (2.*b / sigma) * t) -
            exp(-divid * t) * (1. + SQR(sigma) / (2.*b)) * (cdf_nor(a2) -
            exp(-divid * t) * pnl_normal_density(a1) / (sigma * sqrt(s)) *
            exp(-r * t) * (s_min / s) * pnl_normal_density(a2) /
            exp(-r * t) * pow(s / s_min, -esp) * cdf_nor(-a1 + 2.*(b / sigma) * t)));
    }
}

return OK;
}

int CALC(CF_Fixed_PutLookBack)(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 Fixed_PutLookback_ConzeWiswanathan(ptMod->S0.Val.V_PDOUBLE, (ptOpt->PayOff.Val.V_NUMFUNC_2)->Par[0].Val.V_PDOUBLE,
      ptOpt->Maturity.Val.V_DATE - ptMod->T.Val.V_DATE,
      r, divid, ptMod->Sigma.Val.V_PDOUBLE, &(Met->Res[0].Val.V_DOUBLE), &(Met->Res[1].Val.V_DOUBLE));
}

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

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

    return OK;
}

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

```