

[Help](#)

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

static int Fixed_CallLookback_ConzeWiswanathan(double s, double s_max, double k,
double divid, double sigma, double *ptprice, double *ptdelta)
{
    double b, sigmasqrt, a1, a2, esp, disc;

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

```

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

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

        *ptdelta = exp(-divid * t) * cdf_nor(a1) * (1. + SQR(sigma) / (2.*
            exp(-divid * t) * pnl_normal_density(a1) / (sigma * sqr
            exp(-r * t) * (s_max / s) * pnl_normal_density(a2) / si
            exp(-r * t) * pow(s / s_max, -esp) * cdf_nor(a1 - 2.*(b
    }
}

return OK;
}

int CALC(CF_Fixed_CallLookBack)(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_CallLookback_ConzeWiswanathan(ptMod->S0.Val.V_PDOUBLE,
      (ptOpt->PathDep.Val.V_NUMFUNC_2)->Par[4].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_CallLookBack)(void *Opt, void *Mod)
{
    return strcmp(((Option *)Opt)->Name, "LookBackCallFixedEuro");
}

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

    return OK;
}

PricingMethod MET(CF_Fixed_CallLookBack) =
{
    "CF_Fixed_CallLookBack",
    {" ", PREMIA_NULLTYPE, {0}, FORBID}},
    CALC(CF_Fixed_CallLookBack),
    {"Price", DOUBLE, {100}, FORBID}, {"Delta", DOUBLE, {100}, FORBID} , {" ", PR
    CHK_OPT(CF_Fixed_CallLookBack),
    CHK_ok ,

```

```
    MET(Init)
} ;
```