

[Help](#)

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

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

    if (s_max < s)
    {
        *ptprice = 0.;
        *ptdelta = 0.;
    }
    else
    {
        b = r - divid;
        sigmasqrt = sigma * sqrt(t);
        a1 = (log(s / s_max) + (b + SQR(sigma) / 2.) * t) / sigmasqrt;
        a2 = a1 - sigmasqrt;
        esp = 2.*b / SQR(sigma);
        discount = exp(-r * t);

        if (b == 0)
        {
            *ptprice = s_max * discount * cdf_nor(-a2) - s * discount * cdf_nor(-a
            s * discount * ((SQR(sigma) * t / 2. + log(s / s_max)) * cd

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

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

```

        (1. - SQR(sigma) / (2.*b)) - exp(-r * t) * (s_max / s) * pn
        exp(-divid * t) * (pnl_normal_density(a1) / sigmasqrt - 1.)
    }
}

return OK;
}

int CALC(CF_Floating_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 Floating_PutLookback_GoldmanSosinGatto(ptMod->S0.Val.V_PDOUBLE,
        (ptOpt->PathDep.Val.V_NUMFUNC_2)->Par[4].Val.V_PDOUBLE, ptOpt->Maturity
        r, divid, ptMod->Sigma.Val.V_PDOUBLE, &(Met->Res[0].Val.V_DOUBLE), &(Me

}

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

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

    return OK;
}

PricingMethod MET(CF_Floating_PutLookBack) =
{
    "CF_Floating_PutLookBack",

```

```

    {" ", PREMIA_NULLTYPE, {0}, FORBID}},
    CALC(CF_Floating_PutLookBack),
    {"Price", DOUBLE, {100}, FORBID}, {"Delta", DOUBLE, {100}, FORBID} , {" ", PR
    CHK_OPT(CF_Floating_PutLookBack),
    CHK_ok,
    MET(Init)
};

```