

[Help](#)

```
#include "bs1d_pad.h"

static int Floating_CallLookback_GoldmanSosinGatto(double s, double s_min, double t,
double sigma, double *ptprice, double *ptdelta)
{
    double b, sigmasqrt, a1, a2, esp, discount;

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

        if (b == 0.)
        {
            *ptprice = discount * (s * cdf_nor(a1) - s_min * cdf_nor(a2)) +
                s * discount * (sigmasqrt * pnl_normal_density(a1) - cdf_nor(a2));

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

            *ptdelta = exp(-divid * t) * cdf_nor(a1) * (1. + SQR(sigma) / (2.*b)) +
                exp(-divid * t) * pnl_normal_density(a1) / (sigma * sqrt(t)) -
                s_min * exp(-r * t) * cdf_nor(a2) * (1. + SQR(sigma) / (2.*b)) +
                s_min * exp(-r * t) * pnl_normal_density(a2) / (sigma * sqrt(t));
        }
    }
}
```

```

        - exp(-divid * t) * SQR(sigma) / (2.*b) +
        exp(-r * t) * pow(s / s_min, -esp) * cdf_nor(-a1 + 2.*(b /
    }
}
return OK;
}

int CALC(CF_Floating_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 Floating_CallLookback_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_CallLookBack)(void *Opt, void *Mod)
{
    return strcmp(((Option *)Opt)->Name, "LookBackCallFloatingEuro");
}

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

    return OK;
}

PricingMethod MET(CF_Floating_CallLookBack) =
{
    "CF_Floating_CallLookBack",
    {" ", PREMIA_NULLTYPE, {0}, FORBID}},

```

```
CALC(CF_Floating_CallLookBack),  
{{"Price", DOUBLE, {100}, FORBID}, {"Delta", DOUBLE, {100}, FORBID} , {" " , PR  
CHK_OPT(CF_Floating_CallLookBack),  
CHK_ok,  
MET(Init)  
};
```