

[Help](#)

```
#include "bs1d_pad.h"

static int Floating_PutLookback_GoldmanSosinGat
to(double s, double s_max, double t, double r,
    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)/sig
masqrt;
        a2=a1-sigmasqrt;
        esp=2.*b/SQR(sigma);
        discount=exp(-r*t);

        if (b == 0)
        {
            *ptprice = s_max*discount*N(-a2) - s*dis
count*N(-a1) +
                s*discount*( (SQR(sigma)*t/2.+log(s/s_
max))*N(a1) + sigmasqrt*nd(a1) );

            *ptdelta = discount*N(a1)*(2.+SQR(sig
ma)*t/2.+log(s/s_max)) - discount +
                    discount*nd(a1)*(1.+SQR(sigma)*
t)/sigmasqrt -
                    discount*(s_max/s)*nd(a2)/sig
masqrt;
        }
        else
        {
```

```

        *ptprice=s_max*exp(-r*t)*N(-a2)-s*exp(-div
id*t)*N(-a1)+
s*exp(-r*t)*(SQR(sigma)/(2.*b))*
(-pow(s/s_max,-esp)*N(a1-(2.*b/sigma)*sqrt(t)
)+exp(b*t)*N(a1));

        *ptdelta=exp(-divid*t)*N(a1)*(1.+SQR(sigma)
/(2.*b))+
exp(-r*t)*pow(s/s_max,-esp)*N(a1-(2.*b/sigma)
*sqrt(t))*
(1.-SQR(sigma)/(2.*b))-exp(-r*t)*(s_max/s)*nd
(a2)/sigmasqrt+
exp(-divid*t)*(nd(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.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));
}

int CHK_OPT(CF_Floating_PutLookBack)(void *Opt,
void *Mod)

```

```
{
    return strcmp( ((Option*)Opt)->Name, "
        LookBackPutFloatingEuro");
}

static int MET(Init)(PricingMethod *Met)
{
    return OK;
}

PricingMethod MET(CF_Floating_PutLookBack)=
{
    "CF_Floating_PutLookBack",
    {{ " ", END, 0, FORBID}},
    CALC(CF_Floating_PutLookBack),
    {{ "Price", DOUBLE, 100, FORBID}, {"Delta", DOUBLE, 10
        0, FORBID} , {" ", END, 0, FORBID}},
    CHK_OPT(CF_Floating_PutLookBack),
    CHK_ok,
    MET(Init)
};
```

## References