

## [Help](#)

```
#include "
href../../../../mod/bs1d/bs1d_std/bs1d_std_h_src.pdfbs1d_std.h"

static int CallSpread_BlackScholes_73(double s, double k1, double k2, double t,
{
    double sigmasqrt, d1, d2, delta;

    sigmasqrt = sigma * sqrt(t);
    d1 = (log(s / k1) + (r - divid) * t) / sigmasqrt + sigmasqrt / 2.;
    d2 = d1 - sigmasqrt;
    delta = exp(-divid * t) * cdf_nor(d1);

    *ptprice = s * delta - exp(-r * t) * k1 * cdf_nor(d2);
    *ptdelta = delta;

    d1 = (log(s / k2) + (r - divid) * t) / sigmasqrt + sigmasqrt / 2.;
    d2 = d1 - sigmasqrt;
    delta = exp(-divid * t) * cdf_nor(d1);

    /*Price*/
    *ptprice -= s * delta - exp(-r * t) * k2 * cdf_nor(d2);

    /*Delta*/
    *ptdelta -= delta;

    return OK;
}

int CALC(CF_CallSpread)(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 CallSpread_BlackScholes_73(ptMod->S0.Val.V_PDOUBLE,
                                      (ptOpt->PayOff.Val.V_NUMFUNC_1)->Par[0].Val.
```

```

    ptOpt->Maturity.Val.V_DATE - ptMod->T.Val.V_
    &(Met->Res[0].Val.V_DOUBLE), &(Met->Res[1].V

}

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

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

    return OK;
}

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

```