

[Help](#)

```
#include "
href../../../../mod/bs2d/bs2d_std2dg/bs2d_std2dg_h_src.pdfbs2d_std2dg.h"
#include "pnl/pnl_cdf.h"

#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2013+2) //The "#els
static int CHK_OPT(AP_Spread_Bjerk Sund)(void *Opt, void *Mod)
{
    return NONACTIVE;
}
int CALC(AP_Spread_Bjerk Sund)(void *Opt, void *Mod, PricingMethod *Met)
{
    return AVAILABLE_IN_FULL_PREMIA;
}
#else

static int Spread_Bjerk SundAn(double s01, double s02, double K, double t, double
                                double sigma1, double sigma2, double rho, double *
{

    //forward prices
    double xx1 = s01 * exp((r - divid1) * t);
    double xx2 = s02 * exp((r - divid2) * t);

    double a = xx2 + K;
    double b = xx2 / a;

    //intermediary variables
    double ss1 = sigma1 * sigma1 / 2;
    double ss2 = sigma2 * sigma2 / 2;
    double ss12 = rho * sigma1 * sigma2;
    double logxa = log(xx1 / a);

    //quotient volatility S1/S2
    double s = sqrt(2 * (ss1 - b * ss12 + b * b * ss2));

    double st = s * sqrt(t);

    double d1 = (logxa + (ss1 - b * ss12 + b * b * ss2) * t) / st;
```

```

double d2 = (logxa + (-ss1 + ss12 + b * (b - 2) * ss2) * t) / st;
double d3 = (logxa + (-ss1 + b * b * ss2) * t) / st;

double res = exp(-r * t) * (xx1 * pnl_cdfnor(d1) - xx2 * pnl_cdfnor(d2) - K *

if (res < 0) res = 0.;
*ptprice = res;

*ptdelta1 = exp(-divid1 * t) * pnl_cdfnor(d1);
*ptdelta2 = -exp(-divid2 * t) * pnl_cdfnor(d2);

return OK;
}

int CALC(AP_Spread_Bjerk Sund)(void *Opt, void *Mod, PricingMethod *Met)
{
    TYPEOPT *ptOpt = (TYPEOPT *)Opt;
    TYPEMOD *ptMod = (TYPEMOD *)Mod;
    double r, divid1, divid2;

    r = log(1. + ptMod->R.Val.V_DOUBLE / 100.);
    divid1 = log(1. + ptMod->Divid1.Val.V_DOUBLE / 100.);
    divid2 = log(1. + ptMod->Divid2.Val.V_DOUBLE / 100.);
    //CallSpread
    if ((ptOpt->PayOff.Val.V_NUMFUNC_2)->Compute == CallSpread2d)
        return Spread_Bjerk SundAn(ptMod->S01.Val.V_PDOUBLE, ptMod->S02.Val.V_PDOUBLE
    else//PutSpread with -strike
        return Spread_Bjerk SundAn(ptMod->S01.Val.V_PDOUBLE, ptMod->S02.Val.V_PDOUBLE
}

static int CHK_OPT(AP_Spread_Bjerk Sund)(void *Opt, void *Mod)
{
    if ((strcmp(((Option *)Opt)->Name, "CallSpread2dEuro") == 0) || (strcmp(((Opti
        return OK;

    return WRONG;
}
#endif //PremiaCurrentVersion
static int MET(Init)(PricingMethod *Met, Option *Opt)
{
    if (Met->init == 0)

```

```

    {
        Met->init = 1;
    }

    return OK;
}

PricingMethod MET(AP_Spread_Bjersund) =
{
    "AP_Spread_Bjersund",
    {{" ", PREMIA_NULLTYPE, {0}, FORBID}},
    CALC(AP_Spread_Bjersund),
    { {"Price", DOUBLE, {100}, FORBID}, {"Delta1", DOUBLE, {100}, FORBID} , {"Delt
        {" ", PREMIA_NULLTYPE, {0}, FORBID}
    },
    CHK_OPT(AP_Spread_Bjersund),
    CHK_ok,
    MET(Init)
} ;

```