

[Help](#)

```
#include "bs2d_std2d.h"

static int ExchangeAn(double s1,double s2,double
    ratio,double t,
                    double r,double divid1,double divid2,
                    double sigma1,double sigma2,
    double rho,
                    double *ptprice,double *ptdelta1,
    double *ptdelta2)
{
    double b1,b2,sigma,d1,d2;

    b1=r-divid1;
    b2=r-divid2;
    sigma=sqrt(SQR(sigma1)+SQR(sigma2)-2.0*rho*sigma1*sigma2);
    d1=(log(s1/(s2*ratio) )+ (b1-b2+SQR(sigma)/2.0)*t)/(sigma*sqrt(t));
    d2=d1-sigma*sqrt(t);

    /*Price*/
    *ptprice=s1*exp((b1-r)*t)*N(d1)-ratio*s2*exp((b2-r)*t)*N(d2);

    /*Deltas*/
    *ptdelta1=exp((b1-r)*t)*N(d1);
    *ptdelta2=-ratio*exp((b2-r)*t)*N(d2);

    return OK;
}

int CALC(CF_Exchange)(void *Opt,void *Mod,Pricing
    Method *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.
    );

    return ExchangeAn(ptMod->S01.Val.V_PDOUBLE,pt
    Mod->S02.Val.V_PDOUBLE,(ptOpt->PayOff.Val.V_
    NUMFUNC_2)->Par[0].Val.V_PDOUBLE,
        ptOpt->Maturity.Val.V_DATE-ptMod->T.Val.
    V_DATE,
        r,divid1,divid2,
        ptMod->Sigma1.Val.V_PDOUBLE,ptMod->Sigma2
    .Val.V_PDOUBLE,ptMod->Rho.Val.V_RGDOUBLE,
        &(Met->Res[0].Val.V_DOUBLE),&(Met->Res[1]
    .Val.V_DOUBLE),&(Met->Res[2].Val.V_DOUBLE) );
}

int CHK_OPT(CF_Exchange)(void *Opt, void *Mod)
{
    return strcmp( ((Option*)Opt)->Name,"
    ExchangeEuro");
}

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

PricingMethod MET(CF_Exchange)=
{
    "CF_Exchange",
    {{ " ",END,0,FORBID}},
    CALC(CF_Exchange),
    {{ "Price",DOUBLE,100,FORBID},{ "Delta1",
    DOUBLE,100,FORBID} ,{"Delta2",DOUBLE,100,FORBID} ,
    { " ",END,0,FORBID}},
    CHK_OPT(CF_Exchange),
    CHK_ok,
    MET(Init)
} ;

```

References