

[Help](#)

```
#include "bs1d_std.h"

int Call_BlackScholes_73(double s,double k,
    double t,double r,double divid,double sigma,double *pt
    price,double *ptdelta){
    double sigmasqrt,d1,d2,delta;

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

    /*Price*/
    *ptprice=s*delta-exp(-r*t)*k*N(d2);

    /*Delta*/
    *ptdelta=delta;

    return OK;
}

int CALC(CF_Call)(void *Opt,void *Mod,Pricing
    Method *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 Call_BlackScholes_73(ptMod->S0.Val.V_
    PDOUBLE,(ptOpt->PayOff.Val.V_NUMFUNC_1)->Par[0].
    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_Call)(void *Opt, void *Mod)
{
    return strcmp( ((Option*)Opt)->Name, "
    CallEuro");
}

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

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

References