

## Help

```
extern "C" {
#include "
href../../mod/twohypergeometric1d/twohypergeometric1d_std/twohypergeometric1d
#include "pnl/pnl_cdf.h"
}

static int AP_PrivaultCallHyper(double s, double k, double t, double v, double a
{
    double y=sqrt(1.0/c*log(1.0+c*exp(2.0*v)*(exp(2.0*a*t)-1.0)/2.0/a));
    double d1=1.0/y*(log(s/k)+y*y/2.0);
    double d2=1.0/y*(log(s/k)-y*y/2.0);
    double A1=pow(y,8.0)/2.0/c+5.0*pow(y,6.0)/4.0/c/c+2.0*pow(y,4.0)/c/c/c+9.0/4.0
    double A2=-pow(y,8.0)/c-5.0*pow(y,6.0)/c/c-16.0*pow(y,4.0)/c/c/c-24.0*y*y/c/c/
    double A3=-pow(y,8.0)/10.0/c+exp(-2.0*c*y*y)*A1+A2*exp(-c*y*y)+93.0/4.0/c/c/c/
    double B1=-pow(y,6.0)/4.0/c/c-pow(y,4.0)/2.0/c/c/c-3.0*y*y/4.0/c/c/c/c-3.0/4.0
    double B2=pow(y,6.0)/c/c+4.0*pow(y,4.0)/c/c/c+12*y*y/c/c/c/c+24.0/c/c/c/c/c+24
    double B3=pow(y,8.0)/10.0/c+exp(-2*c*y*y)*B1+B2*exp(-c*y*y)-189.0/8.0/c/c/c/c/
    double C1=-pow(y,5.0)/c/c+pow(y,3.0)/2.0/c/c/c+3*y*y/c/c/c/c+4*y/c/c/c/c+4.5/c
    double C2=-3.0*pow(y,5.0)/c/c-9.0*y*y*y/c/c/c-6.0*y*y/c/c/c/c-64.0*y/c/c/c/c-3
    double C3=-7.0/30.0/c*pow(y,7.0)-2*y/pow(c,4.0)+C1*exp(-2.0*c*y*y)+C2*exp(-c*y
    double D=exp(-2.0*c*y*y)*(exp(c*y*y)*(c*c*c*pow(y,6.0)-3.0)+3.0*c*y*y*(0.5*c*y
    double E1=pow(y,10.0)/c/c+pow(y,8.0)/c/c/c-15.0*pow(y,6.0)/pow(c,4.0)-27.0*pow
    double E2=2.0*pow(y,10.0)/c/c+2.0*pow(y,8.0)/c/c/c+17.0*pow(y,6.0)/pow(c,4.0)+
    double E3=pow(y,12.0)/15.0/c+4*pow(y,6.0)/pow(c,4.0)-189.0*y*y/2/pow(c,6.0)+E1
    double c0=s*pnl_cdfnor(d1)-k*pnl_cdfnor(d2);
    double c1=-eta*rho*k*d2/y/y/c/c/c*pnl_normal_density(d2)*(exp(-c*y*y)*(c*c*y*y
    double c2=eta*eta/c*k*pnl_normal_density(d2)*(A3/y+d2*B3+d2*d2/y*B3)+eta*eta*r
    double dc1=(-eta*rho*k/c/c/c/y/y/s/y*pnl_normal_density(d2)-eta*rho*k/c/c/c*d2
    double dc2=eta*eta/c*k*pnl_normal_density(d2)*d2/s/y*(A3/y+d2*B3+d2*d2/y*B3)+e
    (A3/y+B3/s/y+2.0*d2/s/y/y*B3)+eta*eta*rho*rho*k*pnl_normal_density(d2)*d2/s/y*(C
    eta*eta*rho*rho*k*pnl_normal_density(d2)*(C3+2.0*D/3/pow(c,7.0)/pow(y,4.0)*
    (4.0*pow(d2,3.0)/3.0/y/y/s-1.0/s/y+d2*d2/s/y)+2.0*d2*E3/s/pow(y,5.0));

    *ptprice=c0+epsilon*c1+epsilon*epsilon*c2;
    *ptdelta=pnl_cdfnor(d1)+pnl_normal_density(d1)/y-k*pnl_normal_density(d2)/s/y+

    double sigma0=y/sqrt(t);
    double sigma1=c1/k/t/pnl_normal_density(d2);
    double sigma2=c2/k/t/pnl_normal_density(d2)-d1*d2*sigma1*sigma1/2/sigma0;
```

```

    *ptIMPLIEDvolatility=sigma0+epsilon*sigma1+epsilon*epsilon*sigma2;
    return OK;
}

extern "C" {
    int CALC(AP_CALLPRIVAULTSHE)(void *Opt, void *Mod, PricingMethod *Met)
    {
        TYPEOPT *ptOpt = (TYPEOPT *)Opt;
        TYPEMOD *ptMod = (TYPEMOD *)Mod;
        double strike;
        NumFunc_1 *p;

        p = ptOpt->PayOff.Val.V_NUMFUNC_1;
        strike = p->Par[0].Val.V_DOUBLE;

        return AP_PrivaultCallHyper(ptMod->S0.Val.V_PDOUBLE,
            strike/*ptOpt->PayOff.Val.V_NUMFUNC_1*/,
            ptOpt->Maturity.Val.V_DATE - ptMod->T.Val.V_DATE,
            ptMod->V0.Val.V_PDOUBLE
            , ptMod->a.Val.V_PDOUBLE,
            ptMod->c.Val.V_PDOUBLE,
            ptMod->epsilon.Val.V_PDOUBLE,
            ptMod->eta.Val.V_PDOUBLE,
            ptMod->Rho.Val.V_PDOUBLE,
            &(Met->Res[0].Val.V_DOUBLE),
            &(Met->Res[1].Val.V_DOUBLE), &(Met->Res[2].Val.V_DOUBLE)
        );
    }

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

    static int MET(Init)(PricingMethod *Met, Option *Opt)

```

```

{
    if (Met->init == 0)
    {
        Met->init = 1;
Met->HelpFilenameHint = "ap_callprivaultshe_2hypergeom";
    }

    return OK;
}

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

```