

## Help

```
extern "C" {
#include "
href../../mod/doublehes1d/doublehes1d_std/doublehes1d_std_h_src.pdfhes1d_std.
#include "pnl/pnl_specfun.h"

#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2017+2) //The "#els
static int CHK_OPT(AP_ZhangCall_Heston)(void *Opt, void *Mod)
{
    return NONACTIVE;
}
int CALC(AP_ZhangCall_Heston)(void *Opt, void *Mod, PricingMethod *Met)
{
    return AVAILABLE_IN_FULL_PREMIA;
}
#else
////////////////////////////////////
}

int ApZhangCallHeston(double s0,NumFunc_1 *p, double t,double v0, double kappa,
{
    double strike;

    strike = p->Par[0].Val.V_PDOUBLE;

    double z=theta*t + 1/kappa*(1-exp(-kappa*t))*(v0-theta);
double d1=(log(s0/strike)+z/2)/sqrt(z);
double d2=(log(s0/strike)-z/2)/sqrt(z);
double A=1 - kappa*t*exp(-kappa*t)/(1-exp(-kappa*t));
double B=kappa*t*t*exp(-kappa*t)/(1-exp(-kappa*t))-1/kappa*(1-exp(-kappa*t));
double C=1-exp(-kappa*t)/(1-exp(-kappa*t))*(kappa*t+0.5*kappa*t*kappa*t);
double D=t*exp(-kappa*t)-2/kappa*(1-exp(-kappa*t))+t*exp(-kappa*t)/(1-exp(-kappa
double E=1-exp(-kappa*t)/(1-exp(-kappa*t))*(2*kappa*t-(1-exp(-kappa*t)));
double F=0.5/kappa*(1-exp(-2*kappa*t))-2*kappa*(1-exp(-kappa*t))+t*exp(-kappa*t)
double g1= A*z+theta*B;
double g2=0.5*(A*z+theta*B)*(A*z+theta*B);
double h2=C*z+theta*D;
double m2=E*z+theta*F;
double G1=-0.5*rho*sigma/kappa*d2*1/z*strike*1/sqrt(2*M_PI)*exp(-0.5*d2*d2);
double G2=0.25*rho*rho*sigma*sigma/kappa/kappa*strike*1/sqrt(2*M_PI)*exp(-0.5*d2
```

```

        double H2=-0.5*rho*rho*sigma*sigma/kappa/kappa*strike*1/sqrt(2*M_PI)*exp(
double M2=1/8*sigma*sigma/kappa/kappa*strike*1/sqrt(2*M_PI)*exp(-0.5*d2*d2
double c0=s0*0.5*pnl_sf_erfc(-(d1)*M_SQRT1_2)-strike*0.5*pnl_sf_erfc(-d2*M
double c1=g1*G1;
double c2=g2*G2+h2*H2+m2*M2;

*ptprice=c0+c1+c2;

double dG1=0.5*rho*sigma/kappa*d2*1/z*strike*1/sqrt(2*M_PI)*exp(-0.5*d2*d2
double dG2=-0.25*rho*rho*sigma*sigma/kappa/kappa*strike*d2*1/sqrt(2*M_PI)*
double dH2= 0.5*rho*rho*sigma*sigma/kappa/kappa*strike*d2*1/sqrt(2*M_PI)*e
double dM2=-1/8*sigma*sigma/kappa/kappa*strike*d2*1/sqrt(2*M_PI)*exp(-0.5*

*ptdelta=0.5*pnl_sf_erfc(-d1*M_SQRT1_2)+1/sqrt(2*M_PI)*exp(-0.5*d1*d1)/sqrt(

double sigma0=sqrt(z/t);
double sigma1=-0.5*rho*sigma/kappa*g1*d2/z/t;
double sigma2=c2/strike/pnl_normal_density(d2)/t-0.5*sigma1*sigma1*d1*d2/s

*ptimpliedvolatility=sigma0+sigma1+sigma2;

return OK;
}
extern "C" {
int CALC(AP_ZhangCall_Heston)(void *Opt, void *Mod, PricingMethod *Met)
{
    TYPEOPT *ptOpt = (TYPEOPT *)Opt;
    TYPEMOD *ptMod = (TYPEMOD *)Mod;

    if (fabs(ptMod->R.Val.V_PDDOUBLE)>0.00001||fabs(ptMod->Divid.Val.V_PDDOUBLE)>0.0
    {
        fprintf(stderr, "Untreated case : r and dividend need be different from ze
        return WRONG;
    }
    else
    {

        return ApZhangCallHeston(ptMod->S0.Val.V_PDDOUBLE,
                                ptOpt->PayOff.Val.V_NUMFUNC_1,
                                ptOpt->Maturity.Val.V_DATE - ptMod->T.Val.V_DATE,

```

```

        ptMod->Sigma0.Val.V_PDOUBLE
        , ptMod->MeanReversion.Val.V_PDOUBLE,
        ptMod->LongRunVariance.Val.V_PDOUBLE,
        ptMod->Sigma.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_ZhangCall_Heston)(void *Opt, void *Mod)
{
    // As the method returns an error, we deactivate it
    return NONACTIVE;
    // if (strcmp(((Option *)Opt)->Name, "CallEuro") == 0)
    //     return OK;
    // return WRONG;
}

#endif //PremiaCurrentVersion
static int MET(Init)(PricingMethod *Met, Option *Opt)
{
    if (Met->init == 0)
    {
        Met->HelpFilenameHint = "AP_zhang_call";
        Met->init = 1;
    }

    return OK;
}

PricingMethod MET(AP_ZhangCall_Heston) =
{
    "AP_ZhangCall_Heston",
    {" ", PREMIA_NULLTYPE, {0}, FORBID}},
    CALC(AP_ZhangCall_Heston),
    { {"Price", DOUBLE, {100}, FORBID}, {"Implied Volatility", DOUBLE, {100}, FORBID},
      {"Delta", DOUBLE, {100}, FORBID},
      {" ", PREMIA_NULLTYPE, {0}, FORBID}
    },

```

```
    CHK_OPT(AP_ZhangCall_Heston),  
    CHK_ok,  
    MET(Init)  
};  
}
```