

Help

```
#include "hes1d_std.h"

static double T,sigma,rho,k, v, r, divid, teta,
            lambda, S, K;

static double charact_funct1(double phi)
{
    double a,b,rs,rsp,sig,tau,tpf1,tpf2;
    fcomplex g,z,w,tp1,tp2,D,C,ans,d,expo;

    tau=T;
    a=k*teta;
    rs=rho*sigma;
    rsp=rs*phi;
    sig=sigma*sigma;

    b=k+lambda-rs;
    z=Complex(-b,rsp);
    z=Cmul(z,z);
    w=RCmul(sig,Complex(-phi*phi,phi));
    d=Csqrt(Csub(z,w));
    tp1=Complex(d.r+b,d.i-rsp);
    tp2=Complex(-d.r+b,-d.i-rsp);
    g=Cdiv(tp1,tp2);
    expo=Cexp(RCmul(tau,d));
    D=Csub(Complex(1,0),expo);
    D=Cdiv(D,Csub(Complex(1,0),Cmul(g,expo)));
    D=Cmul(D,RCmul(1.0/sig,tp1));
    C=Csub(Complex(1,0),Cmul(g,expo));
    C=Cdiv(C,Csub(Complex(1,0),g));
    tpf1=a*(tau*tp1.r-2.0*Clog(C).r)/sig;
    tpf2=a*(tau*tp1.i-2.0*Clog(C).i)/sig;
    tpf2+=(r-divid)*phi*tau;
    ans=Complex(tpf1+v*D.r,tpf2+v*D.i+phi*log(S));
    ans=Cmul(Cexp(ans),Cexp(Complex(0,-phi*log(K)))
    );
    ans=Cdiv(ans,Complex(0,phi));
```

```

    return ans.r;
}

static double charact_funct2(double phi)
{
    double a,b,rsp,sig,tau,tpf1,tpf2;
    fcomplex g,z,w,tp1,tp2,D,C,ans,d,expo;

    tau=T;
    a=k*teta;
    rsp=rho*sigma*phi;
    sig=sigma*sigma;

    b=k+lambda;
    z=Complex(-b,rsp);
    z=Cmul(z,z);
    w=RCmul(sig,Complex(-phi*phi,-phi));
    d=Csqrt(Csub(z,w));
    tp1=Complex(d.r+b,d.i-rsp);
    tp2=Complex(-d.r+b,-d.i-rsp);
    g=Cdiv(tp1,tp2);
    expo=Cexp(RCmul(tau,d));
    D=Csub(Complex(1,0),expo);
    D=Cdiv(D,Csub(Complex(1,0),Cmul(g,expo)));
    D=Cmul(D,RCmul(1.0/sig,tp1));
    C=Csub(Complex(1,0),Cmul(g,expo));
    C=Cdiv(C,Csub(Complex(1,0),g));
    tpf1=a*(tau*tp1.r-2.0*Clog(C).r)/sig;
    tpf2=a*(tau*tp1.i-2.0*Clog(C).i)/sig;
    tpf2+=(r-divid)*phi*tau;
    ans=Complex(tpf1+v*D.r,tpf2+v*D.i+phi*log(S));
    ans=Cmul(Cexp(ans),Cexp(Complex(0,-phi*log(K)))
    );
    ans=Cdiv(ans,Complex(0,phi));

    return ans.r;
}

static double probabilities(int n)
{

```

```

double tp;

if(n==1)
{
    tp=midpnt(charact_funct1,0.0,1.0,10);
    tp+=midinf(charact_funct1,1.0,1.0e30,5);
    tp=0.5+tp/PI;
    return tp;
}
else
{
    tp=midsql(charact_funct2,0.0,1.0,10);
    tp+=midinf(charact_funct2,1.0,1.0e30,5);
    tp=0.5+tp/PI;
    return tp;
}
}

static int CFCallHeston(double s, NumFunc_1 *p,
    double t, double ri, double dividi, double sigma0,
    double ka,double theta,double sigma2,double rhow,
    double *ptprice, double *ptdelta)
{
    double proba1,proba2,temp;

    K=p->Par[0].Val.V_DOUBLE;
    S=s;
    T=t;
    sigma=sigma2;
    v=sigma0;
    teta=theta;
    lambda=0.;
    r=ri;
    divid=dividi;
    rho=rhow;
    k=ka;

    proba1=probabilities(1);
    proba2=probabilities(2);

```

```

temp=s*proba1*exp(-divid*t);
temp-=K*exp(-r*t)*proba2;

/* Price*/
*ptprice=temp;

/* Delta */
*ptdelta=proba1*exp(-divid*t);

return OK;
}

int CALC(CF_CallHeston)(void *Opt, void *Mod, PricingMethod *Met)
{
    TYPEOPT* ptOpt=(TYPEOPT*)Opt;
    TYPEMOD* ptMod=(TYPEMOD*)Mod;
    double r,divid;

    if(ptMod->Sigma.Val.V_PDDOUBLE==0.0)
    {
        Fprintf(TOSCREEN,"BLACK-SHOLES MODEL{n{n{n");
        return WRONG;
    }
    else
    {
        r=log(1.+ptMod->R.Val.V_DOUBLE/100.);
        divid=log(1.+ptMod->Divid.Val.V_DOUBLE/100.);

        return CFCallHeston(ptMod->S0.Val.V_PDDOUBLE,
            ptOpt->PayOff.Val.V_NUMFUNC_1,
            ptOpt->Maturity.Val.V_DATE-ptMod->T.Val.V_DATE,
            r,
            divid, ptMod->Sigma0.Val.V_PDDOUBLE
            ,ptMod->MeanReversion.Val.V_PDDOUBLE,
            ptMod->LongRunVariance.Val.V_PDDOUBLE,
            ptMod->Sigma.Val.V_PDDOUBLE,
            ptMod->Rho.Val.V_PDDOUBLE,
            &(Met->Res[0].Val.V_DOUBLE),
            &(Met->Res[1].Val.V_DOUBLE)

```

```

    );
}

}

```

```

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

```

```

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

```

```

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

```

References