Help

```
#include  "bs2d_std2d.h"

static int CallMaxAn(double s1,double s2,double k
    ,double t,
              double r,double divid1,double divid
    2,
              double sigma1,double sigma2,double
    rho,
              double *ptprice,double *ptdelta1,
    double *ptdelta2)
    {
    double b1,b2,sigma,rho1,rho2,d,d1,d2,norm1,no
    rm2,norm3;
  int dummy=0;

    b1=r-divid1;
    b2=r-divid2;
    sigma=sqrt(SQR(sigma1)+SQR(sigma2)-2.0*rho*si
    gma1*sigma2);
  if (((sigma-PRECISION)<=0)&&((rho+PRECISION)>=
    1))
    {if ((s1*exp(-divid1*t))>=(s2*exp(-divid2*
    t)))
      {
      dummy=Call_BlackScholes_73(s1,k,t,r,div
    id1,sigma1,ptprice,ptdelta1);
      *ptdelta2=0.;
      }
    else
      {
      dummy=Call_BlackScholes_73(s2,k,t,r,div
    id2,sigma2,ptprice,ptdelta2);
      *ptdelta1=0.;
      }
    }
  else
  {
    rho1=(sigma1-rho*sigma2)/sigma;
    rho2=(sigma2-rho*sigma1)/sigma;
    d=(log(s1/s2)+(b1-b2+SQR(sigma)/2.0)*t)/(sig
```

```
    ma*sqrt(t));
    d1=(log(s1/k)+(b1+SQR(sigma1)/2.0)*t)/(sigma1
    *sqrt(t));
    d2=(log(s2/k)+(b2+SQR(sigma2)/2.0)*t)/(sigma2
    *sqrt(t));


    norm1=NN(d1,d,rho1);
    norm2=NN(d2,-d+sigma*sqrt(t),rho2);
    norm3=NN(-d1+sigma1*sqrt(t),-d2+sigma2*sqrt(
    t),rho);

    /*Price*/
    *ptprice=s1*exp((b1-r)*t)*norm1+s2*exp((b2-r)
    *t)*norm2-k*exp(-r*t)*(1.0-norm3);

    /*Deltas*/
    *ptdelta1=exp((b1-r)*t)*norm1;
    *ptdelta2=exp((b2-r)*t)*norm2;
  }
    return 0;
    }

int CALC(CF_CallMax)(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 CallMaxAn(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_DA
    TE-ptMod->T.Val.V_DATE,
```

```
                        r,divid1,divid2,
                        ptMod->Sigma1.Val.V_PDOUBL
    E,ptMod->Sigma2.Val.V_PDOUBLE,ptMod->Rho.Val.V_RG
    DOUBLE,
                        &(Met->Res[0].Val.V_
    DOUBLE),&(Met->Res[1].Val.V_DOUBLE),&(Met->Res[2].Val.
    V_DOUBLE) );
    }


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

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

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

References