```
   Help
#include  "bs1d_pad.h"

static int Fixed_CallLookback_ConzeWiswanathan(
    double s, double s_max, double k, double t, double r,
              double divid, double sigma, double
    *ptprice, double *ptdelta)
{
  double  b,sigmasqrt,a1,a2,esp,disc;

  if (s_max < s)
    {
      *ptprice=0.;
      *ptdelta=0.;
    }
  else
    {
      b=r-divid;
      sigmasqrt=sigma*sqrt(t);
      esp=2.*b/SQR(sigma);
      disc=exp(-r*t);

      if (k>s_max)
    {
      a1=(log(s/k)+ (b+SQR(sigma)/2.)*t)/sigmasq
    rt;
      a2=a1-sigmasqrt;
      if (b == 0)
        {
          *ptprice = s*disc*(1.+SQR(sigma)*t/2.+
    log(s/k))*N(a1) +
                 s*disc*sigmasqrt*nd(a1) - k*dis
    c*N(a2);

          *ptdelta = disc*N(a1)*(2.+SQR(sigma)*t/
    2.+log(s/k)) +
                disc*nd(a1)*(1.+SQR(sigma)*t)/sig
    masqrt   -
                disc*(k/s)*nd(a2)/sigmasqrt;
        }
      else
```

```
    {
      *ptprice=s*exp(-divid*t)*N(a1)-k*exp(-
r*t)*N(a2)+
    s*exp(-r*t)*(SQR(sigma)/(2.*b))*
    (-pow(s/k,-esp)*N(a1-(2.*b/sigma)*sqrt(t)
)+exp(b*t)*N(a1));

      *ptdelta=exp(-divid*t)*N(a1)*(1.+SQR(si
gma)/(2.*b))+
    exp(-divid*t)*nd(a1)/(sigma*sqrt(t))-
    exp(-r*t)*(k/s)*nd(a2)/sigmasqrt+
    exp(-r*t)*pow(s/k,-esp)*N(a1-2.*(b/sigma)
*sqrt(t))*(1.-SQR(sigma)/(2*b));
    }
}
  else
{
  a1=(log(s/s_max)+ (b+SQR(sigma)/2.)*t)/sig
masqrt;
  a2=a1-sigmasqrt;
  if (b == 0)
    {
      *ptprice = disc*(s_max-k) + s*disc*(1.+
SQR(sigma)*t/2.+log(s/s_max))*N(a1) +
             s*disc*sigmasqrt*nd(a1) - s_max*
disc*N(a2) ;

      *ptdelta = disc*N(a1)*(2.+SQR(sigma)*t/
2.+log(s/s_max)) +
             disc*nd(a1)*(1.+SQR(sigma)*t)/si
gmasqrt   -
             disc*(s_max/s)*nd(a2)/sigmasqrt;
    }
  else
    {

      *ptprice=exp(-r*t)*(s_max-k)+s*exp(-div
id*t)*N(a1)-
    s_max*exp(-r*t)*N(a2)+
    s*exp(-r*t)*(SQR(sigma)/(2.*b))*
    (-pow(s/s_max,-esp)*N(a1-(2.*b/sigma)*sq
```

```
    rt(t))+exp(b*t)*N(a1));

          *ptdelta=exp(-divid*t)*N(a1)*(1.+SQR(si
    gma)/(2.*b))+
        exp(-divid*t)*nd(a1)/(sigma*sqrt(t))-
        exp(-r*t)*(s_max/s)*nd(a2)/sigmasqrt+
        exp(-r*t)*pow(s/s_max,-esp)*N(a1-2.*(b/si
    gma)*sqrt(t))*(1.-SQR(sigma)/(2*b));
        }
    }
    }


  return OK;
}

int CALC(CF_Fixed_CallLookBack)(void*Opt,void *
    Mod,PricingMethod *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 Fixed_CallLookback_ConzeWiswanathan(pt
    Mod->S0.Val.V_PDOUBLE,
                        (ptOpt->PathDep.Val.V_
    NUMFUNC_2)->Par[4].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,
                        divid,
                        ptMod->Sigma.Val.V_PDOU
    BLE,
                        &(Met->Res[0].Val.V_
    DOUBLE),
                        &(Met->Res[1].Val.V_
```

```
    DOUBLE));
}

int CHK_OPT(CF_Fixed_CallLookBack)(void *Opt, vo
    id *Mod)
{
  return strcmp( ((Option*)Opt)->Name,"
    LookBackCallFixedEuro");
}

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

PricingMethod MET(CF_Fixed_CallLookBack)=
{
  "CF_Fixed_CallLookBack",
  {{" ",END,0,FORBID}},
  CALC(CF_Fixed_CallLookBack),
  {{"Price",DOUBLE,100,FORBID},{"Delta",DOUBLE,10
    0,FORBID} ,{" ",END,0,FORBID}},
  CHK_OPT(CF_Fixed_CallLookBack),
  CHK_ok ,
  MET(Init)
} ;
```

References