

[Help](#)

```
#include "bs1d_std.h"

static int AllOrNothing_BlackScholes_73(double s,
    double k,double rebate,double t,double r,double divid,
    double sigma,double *ptprice,double *ptdelta)
{
    double sigmasqrt,d1;

    sigmasqrt=sigma*sqrt(t);
    d1=(log(s/k)+(r-divid)*t)/sigmasqrt-sigmasqrt
    /2.;

    /*Price*/
    *ptprice=exp(-r*t)*rebate*N(d1);

    /*Delta*/
    *ptdelta=exp(-r*t)*rebate*exp(-SQR(d1)/2.)/(
    sqrt(2.*PI*t)*sigma*s);

    return OK;
}

int CALC(CF_Digit)(void *Opt,void *Mod,Pricing
    Method *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 AllOrNothing_BlackScholes_73(ptMod->S0
    .Val.V_PDOUBLE,
        (ptOpt->PayOff.Val.V_NUMFUNC_1)->Par[0].
    Val.V_PDOUBLE,(ptOpt->PayOff.Val.V_NUMFUNC_1)->
    Par[1].Val.V_PDOUBLE,
        ptOpt->Maturity.Val.V_DATE-ptMod->T.Val.
    V_DATE,r,divid,ptMod->Sigma.Val.V_PDOUBLE,
        &(Met->Res[0].Val.V_DOUBLE),&(Met->Res[1])

```

```
.Val.V_DOUBLE));  
}  
  
int CHK_OPT(CF_Digit)(void *Opt, void *Mod)  
{  
    return strcmp( ((Option*)Opt)->Name,"  
    DigitEuro");  
}  
  
static int MET(Init)(PricingMethod *Met)  
{  
    return OK;  
}  
  
PricingMethod MET(CF_Digit)=  
{  
    "CF_Digit",  
    {{ " ",END,0,FORBID}},  
    CALC(CF_Digit),  
    {{ "Price",DOUBLE,100,FORBID},{ "Delta",DOUBLE,  
    100,FORBID} ,{ " ",END,0,FORBID}},  
    CHK_OPT(CF_Digit),  
    CHK_ok,  
    MET(Init)  
} ;
```

References