

## Help

```

#ifndef _BS1D_LIMDISC_H
#define _BS1D_LIMDISC_H

#include "bs1d.h"
#include "limdisc.h"

#include "mathtools.h"
#include "random.h"
#include "numfunc.h"
#include "transopt.h"

static int formula(double s,double k,double r,
    double divid,double sigma,double t,double l, double re
    bate,int phi,int eta,
        double *A,double *B,double *C,
    double *D,double *E,double *F)
{
    double b,x1,x2,y1,y2,z,mu,lambda,sigmasqrt;

    sigmasqrt=sigma*sqrt(t);
    b=r-divid;
    mu=(b-SQR(sigma)/2.)/SQR(sigma);
    lambda=sqrt(SQR(mu)+2.*r/SQR(sigma));
    x1=log(s/k)/sigmasqrt + (1+mu)*sigmasqrt;
    x2=log(s/l)/sigmasqrt + (1+mu)*sigmasqrt;
    y1=log(SQR(l)/(s*k))/sigmasqrt+(1+mu)*sigmasq
    rt;
    y2=log(l/s)/sigmasqrt + (1+mu)*sigmasqrt;
    z=log(l/s)/sigmasqrt + lambda*sigmasqrt;
    *A=phi*s*exp((b-r)*t)*N(phi*x1)-phi*k*exp(-r*
    t)*N(phi*x1-phi*sigmasqrt);
    *B=phi*s*exp((b-r)*t)*N(phi*x2)-phi*k*exp(-r*
    t)*N(phi*x2-phi*sigmasqrt);
    *C=phi*s*exp((b-r)*t)*pow(l/s,2.*(1.+mu))*N(
    eta*y1)-
        phi*k*exp(-r*t)*pow(l/s,2.*mu)*N(eta*y1-
    eta*sigmasqrt);
    *D=phi*s*exp((b-r)*t)*pow(l/s,2.*(1.+mu))*N(
    eta*y2)-
        phi*k*exp(-r*t)*pow(l/s,2.*mu)*N(eta*y2-

```

```
    eta*sigmasqrt);
    *E=rebate*exp(-r*t)*(N(eta*x2-eta*sigmasqrt)-
    pow(1/s,2.*mu)*N(eta*y2-eta*sigmasqrt));
    *F=rebate*(pow(1/s,mu+lambda)*N(eta*z)+pow(1/
    s,mu-lambda)*N(eta*z-2.*eta*lambda*sigmasqrt));

    return OK;
}

#endif
```

## References