

## Help

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <assert.h>

#include "pnl/pnl_integration.h"
#include "pnl/pnl_fft.h"
#include "pnl/pnl_finance.h"
#include "pnl/pnl_complex.h"
#include "levy_diffusion.h"
#include "carr.h"

#define EPSILON_DIFF 1.e-5
// ----- Var Swap price method -----

double Var_Swap_price_Levy(Levy_process *Model,
                           dcomplex(*psi)(dcomplex u, Levy_process *model))
{
    /*
        dcomplex Phip = (psi(Complex(EPSILON_DIFF,0.),Model));
        dcomplex Phi0 = (psi(Complex(0.,0.),Model));
        dcomplex Phim = (psi(Complex(-EPSILON_DIFF,0.),Model));
        dcomplex dPhi=Csub(Phip,Phim);
        Phi0=Csub(Cadd(Phip,Phim),RCmul(2.,Phi0));
        return 100.0*sqrt((Creal(Phi0)+0.25*Creal(dPhi)*Creal(dPhi))/(EPSILON_DIFF*EPS
        */
    // psi is hermitian :
    // psi(epsilon,0)-psi(-epsilon,0)= 2 Im(psi)(epsilon)
    // psi'(0) == Im(psi)(epsilon)/epsilon
    // psi(epsilon,0)+psi(-epsilon,0)= 2 Re(psi)(epsilon)
    // psi''(0,0)== 2 Re(psi)(epsilon)/epsilon^2
    dcomplex Phi = (psi(Complex(EPSILON_DIFF, 0.), Model));
    return 100.0 * sqrt(2.0 * Creal(Phi) / (EPSILON_DIFF * EPSILON_DIFF));
}

int Var_Swap_Price_option(Option_Eqd *opt,
```

```

                                Levy_process *Model)
{
    if ((opt->product_type != 6) && (opt->product != 3))
        PNL_ERROR(" Var swap method works only for var swap option !", "attari.c ");
    (opt->delta) = Var_Swap_price_Levy(Model, &Levy_process_characteristic_exponen
    opt->price = (opt->delta * opt->delta - opt->K * opt->K) * exp(-opt->rate * op
    return OK;
}

double Var_Swap_price(double T,
                        Levy_diffusion *Model,
                        dcomplex(*psi)(dcomplex u, double t, Levy_diffusion *model)
{
    // phi is hermitian :
    // phi(epsilon,0)-phi(-epsilon,0)= 2 Im(phi)(epsilon)
    // phi'(0) == Im(phi)(epsilon)/epsilon
    // phi(epsilon,0)+phi(-epsilon,0)= 2 Re(phi)(epsilon)
    // phi''(0,0)== 2 Re(phi)(epsilon)/epsilon^2
    dcomplex Phi = (psi(Complex(EPSILON_DIFF, 0.), T, Model));
    return 100.0 * sqrt(-2.0 * Creal(Phi) / (EPSILON_DIFF * EPSILON_DIFF * T));
}

int Var_Swap_Price_option_LD(Option_Eqd *opt,
                              Levy_diffusion *Model)
{
    if ((opt->product_type != 6) && (opt->product != 3))
        PNL_ERROR(" Var swap method works only for var swap option !", "attari.c ");
    //(opt->delta)=Var_Swap_price_Levy(opt->T,Model,&Levy_process_characteristic_e
    (opt->delta) = Var_Swap_price(opt->T, Model, &Levy_diffusion_ln_characteristic
    opt->price = (opt->delta * opt->delta - opt->K * opt->K) * exp(-opt->rate * op
    return OK;
}

#undef EPSILON_DIFF

```