Help

```
#include "
href../../../common/math/kirkby/ap_european_proj_h_src.pdfap_european_proj.h"
#include "
href../../../common/math/kirkby/proj_integrand_h_src.pdfproj_integrand.h"

#include <pnl/pnl_fft.h>
#include <cmath>

int price_european_model(Model_proj* model, double S_0, double W, double T, int

long int N = (long int)pow(2, logN);    // grid roughly centered on[c1 - alph, c
double alph = model->get_truncation_alpha(L1, T);
double dx = 2 * alph / (N - 1.);
double a = 1 / dx;

double lws = log(W / S_0);
double lam = model->get_c1() - (N / 2 - 1.) * dx;
int nbar = floor(a * (lws - lam) + 1);
if (nbar >= N)
nbar = N - 1;

double xmin = lws - (nbar - 1.) * dx;
double dw = 2 * M_PI / (N * dx);

PnlVectComplex* grand = pnl_vect_complex_create_from_zero(N);
cubic_fourier_integrand(model, grand, T, N, dw, a, xmin);
pnl_fft_inplace(grand);

double cons1, cons2, cons3, cons4;

cons1 = (1./90) * (14. / 3 * (2 + cosh(dx))
+ .5 * (cosh(1.5 * dx) + 9 * cosh(1.25 * dx) + 23 * cosh(.5 * dx))
+ 1. / 6 * (cosh(7. / 4 * dx) + 121 * cosh(.75 * dx) + 235 * cosh(.25 * dx)));

cons2 = W * (23. / 24 - (exp(-dx) / 90) * ((28 + 7 * exp(-dx)) / 3.
+ (14 * exp(dx) + exp(-7. / 4 * dx) + 242 * cosh(.75 * dx) + 470 * cosh(.25 * dx
+ .25 * (exp(-1.5 * dx) + 9 * exp(-1.25 * dx) + 46 * cosh(.5 * dx))));

cons3 = W * (.5 - .05 * (28. / 27 + exp(-7. / 4 * dx) / 54. + exp(-1.5 * dx) / 1
```

```
+ exp(-1.25 * dx) / 2 + 14 * exp(-dx) / 27
+ 121. / 54 * exp(-.75 * dx) + 23. / 18 * exp(-.5 * dx) + 235. / 54 * exp(-.25 *

cons4 = W* (1. / 24 - 1. / 20 * exp(dx) *
(exp(-7. / 4 * dx) / 54 + exp(-1.5 * dx) / 18 + exp(-1.25 * dx) / 2 + 7 * exp(-d

double cons1S = S_0 * cons1;
double beta_i;
double val = 0.0;

for (int i = 0; i < nbar-2; i++) {
//LET(beta, i) = pnl_vect_complex_get_real(grand, i);
beta_i = pnl_vect_complex_get_real(grand, i);
val += beta_i * (W - exp(xmin + dx * i)*cons1S);
}
// Handle the remaining terms near nbar, ie near the strike discontinuity (they
val += cons2 * pnl_vect_complex_get_real(grand, nbar - 2);
val += cons3 * pnl_vect_complex_get_real(grand, nbar - 1);
val += cons4 * pnl_vect_complex_get_real(grand, nbar - 0);

double r = model->get_interest_rate();
val *= 32. * pow(a, 4) * exp(-r * T) / N;

if (call == 1){ // Price Call Option by Put-Call Parity
double q = model->get_div_yield();
val += S_0 * exp(-q * T) - W * exp(-r * T);
}

*price = val;
// TODO: need to add delta ... bump S_0 by a little, and compute another val

pnl_vect_complex_free(&grand);
return OK;
}

int price_european_cgmy(double C, double G, double M, double Y, double r, double
double T, int call, double* price, double* delta, int L1, int logN)
{
CGMY_Model_proj* model = new CGMY_Model_proj(r, q, C, G, M, Y);
int status = price_european_model(model, S_0, W, T, call, price, delta, L1, logN
delete model;
```

```
return status;
}

int Kirkby_PROJ_kou_european(int ifCall, double Spot, double sigma, double lambd
double r, double divid,
double T, double Strike,
int logN, int L1,
double *ptprice)
{
double ptdelta = 0;
Kou_Model_proj* model = new Kou_Model_proj(r, divid, sigma, lambda, P, lambdap,
int status = price_european_model(model, Spot, Strike, T, ifCall, ptprice, &ptde

delete model;
return status;
}
int Kirkby_PROJ_CGMY_european(int ifCall, double Spot, double C, double G, doubl
double r, double divid,
double T, double Strike,
int logN, int L1,
double *ptprice)
{
double ptdelta = 0;
CGMY_Model_proj* model = new CGMY_Model_proj(r, divid, C, G, M, Y);
int status = price_european_model(model, Spot, Strike, T, ifCall, ptprice, &ptde

delete model;
return status;
}
int Kirkby_PROJ_NIG_european(int ifCall, double Spot, double Sigma, double Theta
double r, double divid,
double T, double Strike,
int logN, int L1,
double *ptprice)
{
double ptdelta = 0;
double delta = Sigma / sqrt(Kappa);
double beta = Theta / SQR(Sigma);
double alpha = sqrt(1 / (Kappa*SQR(Sigma)) + beta*beta);

NIG_Model_proj* model = new NIG_Model_proj(r, divid, alpha, beta, delta);
```

```
int status = price_european_model(model, Spot, Strike, T, ifCall, ptprice, &ptde

delete model;
return status;
}
```