Help

```
#ifndef _WIENERHOPF_H
#define _WIENERHOPF_H

#include <stdlib.h>
#include <
href../../common/math/cdo/cdo_math_h_src.pdfmath.h>

#include "pnl/pnl_vector.h"
#include "pnl/pnl_fft.h"
#include "pnl/pnl_matvect.h"
#include "pnl/pnl_random.h"
#include "pnl/pnl_mathtools.h"

/*////////////////////////////////////////////////*/
void fft_real(PnlVect *a, int fft_size, int sign);

int findcoefnew(int model, double mu, double sigma, double lm1, double lp1,
                double num, double nup,
                double cnum, double cnup, double q, double r1,
                double T, double h, long int kmax,
                double er, long int Nt,
                PnlVect *al1);
int findcoef_mc(int model, double mu, double sigma, double lm1, double lp1,
double num, double nup,
double cnum, double cnup, double q, double r1,
double T, double h, long int kmax,
double er, long int Nt,
PnlVect * al1);
int findfactor(double np, double nm, double h, long int kmax,  double lp1, doubl

int expectation(long int kmax, PnlVect *vv1, PnlVect *bl1);

int fastwienerhopf(int model, double mu, double qu, double om, int am, int upord
                   double num, double nup, double cnum, double cnup,
                   double r, double divid,
                   double T, double h, double Strike1,
                   double bar, double rebate,
                   double er, long int step,
                   double *ptprice, double *ptdelta);
```

```
int fastwienerhopfamerican(int model, double mu, double qu, double om,
                           int ifCall, double Spot, double lm1, double lp1,
                           double num, double nup, double cnum, double cnup,
                           double r, double divid,
                           double T, double h, double Strike1,
                           double er, long int step,
                           double *ptprice, double *ptdelta);
//Code for Backward Fourier
int bi_american(double mu, double qu, double om,
                int ifCall, double Spot, double lm1, double lp1,
                double num, double nup, double cnum, double cnup,
                double r, double divid,
                double T, double h, double Strike1,
                double er, long int step,
                double *ptprice, double *ptdelta);
int bi_barr(double mu, double qu, double om, int upordown, int ifCall, double Sp
            double num, double nup, double cnum, double cnup,
            double r, double divid,
            double T, double h, double Strike1,
            double bar, double rebate,
            double er, long int step,
            double *ptprice, double *ptdelta);
// Code for lookback floating strike
int lookback_fls(int model, double mu, double qu, double om, int ifCall, double
                 double num, double nup, double cnum, double cnup,
                 double r, double divid,
                 double T, double h,
                 double er, double *ptprice, double *ptdelta);
// Code for lookback fixed strike
int lookback_fxs(int model, double mu, double qu, double om, int ifCall, double
                 double num, double nup, double cnum, double cnup,
                 double r, double divid, double Strike,
                 double T, double h,
                 double er, double *ptprice, double *ptdelta);
// Code for swing
int swing(int model, double mu, double qu, double om,
          int ifCall, double Spot, double lm1, double lp1,
          double num, double nup, double cnum, double cnup,
          double r, double divid,
          double T, double h, double Strike1, double del, int Nd,
```

```
          double er, long int step,
          double *ptprice, double *ptdelta);
// Code for var
int var_fft(int model, double mu,
            double Spot, double lm1, double lp1,
            double num, double nup, double cnum, double cnup,
            double T, double h, double Strike1, double er, double alpha,
            double *ptprice, double *ptdelta);
// MC for lookback
int froot(long int kmax, PnlVect * F, double Fx, long int *k0, long int *k1);
int frootm(long int kmax, PnlVect * F, double Fx, long int *k0, long int *k1);
int TSL_lookbackfixed_WHMC(int ifCall, double S0, double K, double s_maxmin, dou
T, double r, double divid, double lm1, double lp1, double num, double nup, doubl
int generator, int n_paths, double *ptPrice, double *priceError);
int TSL_lookbackfloat_WHMC(int ifCall, double S0, double s_maxmin, double
T, double r, double divid, double lm1, double lp1, double num, double nup, doubl
int generator, int n_paths, double *ptPrice, double *priceError);

#endif
```