

[Help](#)

```
#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2007+2) //The "#els
#else

#ifndef LEVYFD_H
#define LEVYFD_H

#include <iostream>
#include <fstream>
#include <
href../../common/math/highdim_solver/highdim_vector_h_src.pdfvector>
#include "
href../../common/math/progonka_h_src.pdfprogonka.h"
#include "
href../../common/math/numerics_h_src.pdfnumerics.h"
#include "
href../../common/math/levy_h_src.pdflevy.h"

class Grid
{
    double A1;
    double Ar;
    double dx;
    int N;

public :

    Grid(const double dA1, const double dAr, const int dN);
    inline double x(double i) const
    {
        return A1 + i * dx;
    }
};

double init_cond(const double x, const double S0,
                 const double K, const int product);

double bound_cond(const double x, const double S0, const double K, const double
                 const double ttm, const double r,
                 const int product, const int product_type);
```

```

/*Explicit-implicit finite difference scheme*/
vector<double> price2(int am, const Levy_measure &measure, int product,
                    int product_type, double r, double divid, double S0,
                    double K, double rebate, double A1, double Ar,
                    int Nspace, double T, int Ntime, double &price0, double &delta0)
/*Meaning of arguments:
- product: Call(1), Put(2), or forward(3);
- product_type: European vanilla (1), Up-and-Out(2), Down-and-Out(3), or double barrier (4);
- rebate: constant rebate in the barrier case;
- price0, delta0: output variables*/

/*Centered version of the explicit-implicit scheme*/
vector<double> price2c(int am, const Levy_measure &measure, int product,
                    int product_type, double r, double divid, double S0,
                    double K, double rebate, double A1, double Ar,
                    int Nspace, double T, int Ntime, double &price0, double &delta0)
#endif

#endif //PremiaCurrentVersion

```