

## [Help](#)

```
#ifndef _OPT_H
#define _OPT_H

#include "pnl/pnl_matrix.h"
#include "
href../../../../common/math/ImportanceSampling_jl/src/Model_h_src.pdfmath/Import
#include "
href../../../../common/math/jlparser/include/jlparser/parser_h_src.pdfjlparser/p

#include <cmath>
#include <string>

class BaseOption
{
public:
    std::string label;
    double maturity; /*!< maturity of the option */
    /* duplicates BaseModel::size, no other solution found sofar */
    int size;
    /* duplicates BaseModel::nTimeSteps, no other solution found sofar */
    int nTimeSteps;

    BaseOption();
    BaseOption(const Param &ParamTab);
    virtual ~BaseOption()
    {
    };

    void dvar_girsanov(PnlVect *res, const PnlVect *g, const PnlVect *x, double pa

    /** Matrix path has size columns and
     * (nTimeSteps + 1) rows be carefull this not
     * natural.
     */
    virtual double payoff(const PnlMat *) = 0;
    virtual void print() const;

    static double bound(int number_projection)
    {
```

```

    return std::log((double) number_projection + 1);
}

static bool projection(const PnlVect *value, int number_projection)
{
    if (pnl_vect_norm_two(value) < bound(number_projection))
        return false;

    return true;
}

};

extern BaseOption *instantiate_option(const Param &ParamMap);

#endif

```