

## [Help](#)

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

#ifndef CIRpp1DTREE_H_INCLUDED
#define CIRpp1DTREE_H_INCLUDED

#include "pnl/pnl_vector.h"
#include "
href../../../../../common/math/read_market_zc/InitialYieldCurve_h_src.pdfmath/read_

//*****TreeCIRpp1D structure*****//
typedef struct TreeCIRpp1D
{
    double Tf;                // Final time of the tree, dt=Tf/Ngrid
    int Ngrid;                // Number of time step in the TreeCIRpp1D

    double delta_x;
    double bb;

    PnlVect *t;               // Time step grid, from t[0] to T[Ngrid].
    PnlVect *Xmax;
    PnlVect *Xmin;
    PnlVect *alpha;           // Translation from x to r. ( r_t = x_t + alpha_t)
} TreeCIRpp1D;

//***** Datas specific to Hull and White *****//
typedef struct ModelCIRpp1D
{
    double MeanReversion;     /*Speed reversion of the Hullwhite m
    double Volatility;         /*Volatility of the Hullwhite model.*/
    double LongTermMean;
    double Initialx0;
} ModelCIRpp1D;

//***** Fonctions relatives a la construction de l'arbre *****//

int SetTimegridCapCIRpp1D(TreeCIRpp1D *Meth, int NtY, double current_date, doubl
//Construction of the time grid
```

```

int SetTimegridZCbondCIRpp1D(TreeCIRpp1D *Meth, int n, double current_date, double T);
// Construction of the time grid

int SetTimegridCIRpp1D(TreeCIRpp1D *Meth, int n, double current_date, double T);

double x_value(int i, int h, TreeCIRpp1D *Meth);

double R(double x, double sigma);

double MiddleNode(TreeCIRpp1D *Meth, int i, double a, double b, double sigma, double T);

void SetTreeCIRpp1D(TreeCIRpp1D *Meth, ModelCIRpp1D *ModelParam, ZCMarketData *ZCMarketData);

int indiceTimeCIRpp1D(TreeCIRpp1D *Meth, double s); // t[indiceTimeCIRpp1D(s)] < T

int DeleteTimegridCIRpp1D(struct TreeCIRpp1D *Meth); // Delete the PnlVect t
int DeleteTreeCIRpp1D(struct TreeCIRpp1D *Meth); // Delete the PnlVect Jminimum,

#endif // HW2DTREE_H_INCLUDED
#endif //PremiaCurrentVersion

```