

## [Help](#)

```
#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2008+2) //The "#els
#else

#ifndef TreeHW2D_H_INCLUDED
#define TreeHW2D_H_INCLUDED
#include "
href../../../../../common/math/read_market_zc/InitialYieldCurve_h_src.pdfmath/read_

//*****TreeHW2D structure*****//
typedef struct TreeHW2D
{
    double Tf;                // Final time of the tree, dt=Tf/Ngrid
    int Ngrid;                // Number of time step in the TreeHW2D

    PnlVect *t;               // Time step grid, from t[0] to T[Ngrid].
    PnlVectInt *uIndexMin;    // Jminimum[i] : Minimal index of u at time i
    PnlVectInt *uIndexMax;    // Jmaximum[i] : Maximal index of u at time i

    PnlVectInt *yIndexMin;    // Jminimum[i] : Minimal index of y at time i
    PnlVectInt *yIndexMax;    // Jmaximum[i] : Maximal index of y at time i

    PnlMat *ProbasMatrix;     // Matrix 3x3 of probabilities
    PnlVect *alpha;           // Translation from x to r. ( r_t = y_t - u/(b-a) + alp
} TreeHW2D;

//***** Datas specific to Hull and White *****//
typedef struct ModelHW2D
{
    double rMeanReversion;    /*Speed reversion of r */
    double rVolatility;        /*Volatility of r */
    double uVolatility;        /*Speed reversion of u */
    double uMeanReversion;     /*Volatility of u */
    double correlation;        /*Correlation between r and u */
} ModelHW2D;

//***** Functions specifics to the construction of the tree *****//

int SetTimegridHW2D(TreeHW2D *Meth, int n, double T);
```

```

int SetTimegridHW2D_Cap(TreeHW2D *Meth, int NtY, double T_intermediate, double T

// Construction of the tree (uIndexMin, uIndexMax, yIndexMin, yIndexMax and alph
void SetTreeHW2D(TreeHW2D *Meth, ModelHW2D *ModelParam, ZCMarketData *ZCMarket)

void BackwardIterationHW2D(TreeHW2D *Meth, ModelHW2D *ModelParam, ZCMarketData *

int indiceTimeHW2D(TreeHW2D *Meth, double s); // t[indiceTimeHW2D(s)] < s <= t[in

double delta_xHW2D(double delta_t, double a, double sigma); // Return the step (

double ProbaUpHW2D(double x);          // x : eta_ijk/delta_xHW2D(i+1) avec les not
double ProbaMiddleHW2D(double x);      // x : eta_ijk/delta_xHW2D(i+1) avec les not
double ProbaDownHW2D(double x);        // x : eta_ijk/delta_xHW2D(i+1) avec les not

// Build the matrix 3x3 of probabilities
void BuildProbasMatrixHW2D(TreeHW2D *Meth, double eta_over_delta_t, double eta_ov

int DeleteTimegridHW2D(TreeHW2D *Meth); // Delete the PnlVect t
int DeleteTreeHW2D(TreeHW2D *Meth);    // Delete the PnlVects uIndexMin, uInde

#endif // TreeHW2D_H_INCLUDED
#endif //PremiaCurrentVersion

```