

## Help

```

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

#ifndef TreeShortRate_H_INCLUDED
#define TreeShortRate_H_INCLUDED

#include "pnl/pnl_vector.h"
#include "math/read_market_zc/InitialYieldCurve.h"

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

    PnlVect *t;           // Time step grid, from t[0] to T[Ngrid].
    PnlVectInt *Jminimum; // Jminimum[i] : Minimal index at time i
    PnlVectInt *Jmaximum; // Jmaximum[i] : Maximal index at time i
    PnlVect *alpha;       // Translation from x to r. ( r_t = x_t + alpha_t)
} TreeShortRate;

//***** Datas specific to Hull and White *****//
typedef struct ModelParameters
{
    double MeanReversion;           /*Speed reversion of the SG model.*/
    double RateVolatility;          /*Volatility of the SG model.*/
} ModelParameters;

//***** Tree construction *****//
int SetTimeGrid(TreeShortRate *Meth, int n, double T); // Construction of the ti

int SetTimeGrid_Tenor(TreeShortRate *Meth, int NtY, double T0, double S0, double

void SetTreeShortRate(TreeShortRate *Meth, ModelParameters *ModelParam, ZCMarket

void BackwardIteration(TreeShortRate *Meth, ModelParameters *ModelParam, PnlVect

// Two functions used in the calibration to term structure
double PhiAlpha(double alpha_i, double delta_t_i, double delta_x_i, int jmin, in

```

```

double FindAlpha(double alpha_init, double delta_t_i, double delta_x_i, int jmin

int IndexTime(TreeShortRate *Meth, double s); // t[IndexTime(s)]< s <= t[IndexTi

double SpaceStep(double delta_t, double a, double sigma); // Return the space st

double ProbaUp(double x); // x : eta_ijk/SpaceStep(i+1) using the notation of th
double ProbaMiddle(double x);
double ProbaDown(double x);

int DeleteTimeGrid(struct TreeShortRate *Meth); // Delete the PnlVect t
int DeleteTreeShortRate(struct TreeShortRate *Meth); // Delete the PnlVect Jmini

///<***** Function that defines the model (HW=Hull&White, SG=Squared Ga

///<***** SG *****/
double func_model_sg1d(double x);

double func_model_der_sg1d(double x); // derivative

double func_model_inv_sg1d(double r); // inverse

///<***** HW *****/
double func_model_hw1d(double x);

double func_model_der_hw1d(double x);

double func_model_inv_hw1d(double r);

///<***** BK *****/
double func_model_bk1d(double x);

double func_model_der_bk1d(double x);

double func_model_inv_bk1d(double r);

#endif // HW2DTREE_H_INCLUDED
#endif //PremiaCurrentVersion

```