

Help

```

#if defined(PremiaCurrentVersion) && PremiaCurrentVersion <
    (2007+2) //The "#else" part of the code will be freely available
    after the (year of creation of this file + 2)
#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/Ng
    int rid;
    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 time grid

int SetTimeGrid_Tenor(TreeShortRate *Meth, int NtY, double
    T0, double S0, double periodicity);

void SetTreeShortRate(TreeShortRate *Meth, ModelParameters
    *ModelParam, ZCMarketData *ZCMarket, double (*func_model)(
    double), double (*func_model_der)(double), double (*func_model_i
    nv)(double));

void BackwardIteration(TreeShortRate *Meth, ModelParameters
    *ModelParam, PnlVect *OptionPriceVect1, PnlVect *OptionP
    riceVect2, int index_last, int index_first, double (*func_
    model)(double));

// Two functions used in the calibration to term structure
double PhiAlpha(double alpha_i, double delta_t_i, double de
    lta_x_i, int jmin, int jmax, PnlVect *Q, double ZCbondprice
    , double (*func_model)(double), double (*func_model_der)(
    double));

double FindAlpha(double alpha_init, double delta_t_i,
    double delta_x_i, int jmin, int jmax, PnlVect *Q, double ZCbondprice, do
    er)(double));

int IndexTime(TreeShortRate *Meth, double s); // t[IndexT
    ime(s)] < s <= t[IndexTime(s) + 1]

double SpaceStep(double delta_t, double a, double sigma); /
    / Return the space step at time t(i) with delta_t=t(i)-t(
    i-1) : Delta_x(i)

double ProbaUp(double x); // x : eta_ijk/SpaceStep(i+1) us
    ing the notation of the book Brigo&Mercurio
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 Jminimum, Jmaximum, alpha

///<***** Function that defines the model (HW=Hull&White, SG=Squared Gaussian, BK=Black&Karasinski)*****
*****///

///<***** 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

```

References