

Help

```
#if defined(PremiaCurrentVersion) && PremiaCurrentVersion <
    (2010+2) //The "#else" part of the code will be freely av
    ailable after the (year of creation of this file + 2)
#else
```

```
#ifndef HW1DGCALIBRATION_H_INCLUDED
#define HW1DGCALIBRATION_H_INCLUDED
```

```
#include "pnl/pnl_vector.h"
#include "math/InterestRateModelTree/TreeHW1dGeneralized/
    TreeHW1dGeneralized.h"
#include "math/read_market_zc/InitialYieldCurve.h"
```

```
typedef struct MktATMCapletVolData
{
    double Periodicity;
    PnlVect* CapletMaturity;    // Vector of the maturities
    PnlVect* CapletVolatility;  // Vector of Caplet Volati
    lities for every maturity CapletMaturity[i]

    int NbrData;    // Number of values read in the file.
}MktATMCapletVolData;
```

```
// Compute price of caplet in Black Model
double black_caplet_price(ZCMarketData* ZCMarket, double    vol_impli, double c
```

```
// Compute the implied volatility for caplet in Black
    Model implied by caplet price.
double bk_caplet_vol_implied_newton(ZCMarketData* ZCMarket,
    double caplet_price, double caplet_strike, double perio
    dicity, double caplet_reset_date);
```

```
// Compute price of caplet in HW1dG Model
double hw1dg_caplet_price(ZCMarketData* ZCMarket, double    vol_avg, double cap
```

```
// Compute price of floorlet in HW1dG Model
double hw1dg_floorlet_price(ZCMarketData* ZCMarket, double    vol_avg, double c
```

```

// Compute the average volatility of forward ZC bond in HW1dG Model implied
double hwidg_fwd_zc_vol_implied_newton(ZCMarketData* ZCMarket, double caplet_price, double caplet_strike, double periodicity, double caplet_reset_date);

// From a vector of Black volatilities of caplets, read from market, we compute the corresponding average volatility of forward ZC bond in HW1dG Model
void From_Black_To_HW1dG_volatility(ZCMarketData* ZCMarket, MktATMCapletVolData* MktATMCapletVol, PnlVect* mkt_fwd_zc_mat, PnlVect* mkt_fwd_zc_vol);

// Compute the volatility function of HW1dG Model that makes the model prices of caplets fit those read in market.
// The volatility function of HW1dG Model is supposed to be piecewise constant
int hwidg_calibrate_volatility(ModelHW1dG* HW1dG_Parameters, ZCMarketData* ZCMarket, MktATMCapletVolData* MktATMCapletVol, double T);

// Price of ZC bond at time "t", maturing at time "T", knowing the yield curve at time "0" and short rate at "t" r_t.
double DiscountFactor(ZCMarketData* ZCMarket, ModelHW1dG* HW1dG_Parameters, double T, double t);

// Compute average volatility of forward ZC bond in HW1dG Model.
double hwidg_fwd_zc_average_vol(ModelHW1dG* HW1dG_Parameters, double T, double S);

// Compute price of put option on ZC bond in HW1dG Model.
double hwidg_zc_put_price(ZCMarketData* ZCMarket, ModelHW1dG* HW1dG_Parameters, double strike, double T, double S);

// Compute price of call option on ZC bond in HW1dG Model.
double hwidg_zc_call_price(ZCMarketData* ZCMarket, ModelHW1dG* HW1dG_Parameters, double strike, double T, double S);

///***** Read the caplet volatilities from file *****/

```

```
void ReadCapletMarketData(MktATMCapletVolData* MktATM    CapletVol, int CapletCu  
  
int DeleteMktATMCapletVolData(MktATMCapletVolData* MktATM    CapletVol);  
  
#endif // HW1DGCALIBRATION_H_INCLUDED  
#endif //PremiaCurrentVersion
```

References