```
   Help
#if defined(PremiaCurrentVersion) && PremiaCurrentVersion <
     (2007+2) //The "#else" part of the code will be freely av
    ailable after the (year of creation of this file + 2)
#else

#include <stdio.h>
#include <stdlib.h>
#include "premia_obj.h"
#include "pnl/pnl_mathtools.h"
#include "pnl/pnl_vector.h"
#include "InitialYieldCurve.h"

/*Name of the file where to read P(0, T) of the market.*/
static char init[]="initialyield.dat";

///******************* Initial yield curve ***************
    *****///
void SetInitYieldCurve(int InitYieldCurve_flag, double R_
    flat, ZCMarketData* ZCMarket)
{
    if (InitYieldCurve_flag==0)
    {
        /* Flag to decide to read or not ZC bond datas in "
    initialyields.dat" */
        ZCMarket->FlatOrMarket = 0;
        ZCMarket->Rate = R_flat;
    }

    else
    {
        /* If P(0,T) not read then P(0,T)=exp(-R_flat*T) */
        ZCMarket->FlatOrMarket = 1;
        ReadMarketData(ZCMarket);
    }
}

// Read the ZC price from the file "initialyield.dat" and
    put it in the structure "ZCMarket".
void ReadMarketData(ZCMarketData* ZCMarket)
{
```

```
FILE* Entrees;                    /*File variable of th
e code*/

int i;
char ligne[20];
char* pligne;
double p, tt;
char data[MAX_PATH_LEN];

sprintf(data, "%s%s%s", premia_data_dir, path_sep, ini
t);
Entrees=fopen(data, "r");

if(Entrees==NULL)
{
  printf("Le FICHIER N'A PU ETRE OUVERT. VERIFIER LE
CHEMIN{n"); abort();
}

i=0; // i represents the number of value read in the
file
pligne=ligne;

ZCMarket->Pm = pnl_vect_create(100);
ZCMarket->tm = pnl_vect_create_from_double(100, 0);

while(1)
{
    pligne=fgets(ligne, sizeof(ligne), Entrees);
    if(pligne==NULL)
    {
        break;
    }
    else
    {
        sscanf(ligne, "%lf t=%lf", &p, &tt);
        /* La ligne lue dans le fichier doit etre de
la forme "0.943290 t=0.5" ou 0.943290 est un double pour le
prix de B(0,t=0.5)*/
        LET(ZCMarket->Pm,i) = p;   /*enregistre le prix
 du zero coupon*/
```

```
            LET(ZCMarket->tm,i) = tt;  /*enreristre le
    temps correspondant*/
            i++;
        }
    }

    fclose(Entrees);

    ZCMarket->Nvalue = i;
    pnl_vect_resize(ZCMarket->Pm, i);
    pnl_vect_resize(ZCMarket->tm, i);
}

// Compute f(0, T) the forward rate, known at 0, maturing
    at T.
double ForwardRate(double T, ZCMarketData* ZCMarket)
{
    return -(log(BondPrice(T + INC,ZCMarket))-log(Bond
    Price(T,ZCMarket)))/(INC);
}

double ATMSwaptionStrike(double T_start, double T_end,
    double period, ZCMarketData* ZCMarket)
{
    int i, n=intapprox((T_end-T_start)/period);
    double sum=0., T_i=T_start;

    for(i=0; i<n; i++)
    {
        T_i += period;
        sum += BondPrice(T_i, ZCMarket);
    }
    sum *= period;


    return (BondPrice(T_start, ZCMarket)-BondPrice(T_end,
    ZCMarket))/sum;
}

// Compute the ZC price P(0,T) by interpolating the initia
    l yield curve contained in ZCMarket.
```

```
double BondPrice(double T, ZCMarketData* ZCMarket)
{
    double P0T;
    int i;

    if(T>0)
    {
        if(ZCMarket->FlatOrMarket==0) // If there is no
    curve to read. ie : the initial yield curve is flat.
        {
            P0T = exp(-ZCMarket->Rate * T);
        }
        else
        {
            for(i=0; i<ZCMarket->Nvalue; i++)
            {
                if(T<=GET(ZCMarket->tm,i)) break;
            }

            if(i < ZCMarket->Nvalue)
            {
                P0T = GET(ZCMarket->Pm,i-1)*(GET(ZCMarket->
    tm,i)-T)/(GET(ZCMarket->tm,i)-GET(ZCMarket->tm,i-1)) + GET(
    ZCMarket->Pm,i)*(T-GET(ZCMarket->tm,i-1))/(GET(ZCMarket->tm,
    i)-GET(ZCMarket->tm,i-1));
            }
            else
            {
                P0T=GET(ZCMarket->Pm,i-1)+(T-GET(ZCMarket->
    tm,i-1))*(GET(ZCMarket->Pm,i-1)-GET(ZCMarket->Pm,i-2))/(GET
    (ZCMarket->tm,i-1)-GET(ZCMarket->tm,i-2));
            }
        }
    }

    else // P(0,0) = 1
    {
        P0T=1;
    }

    return P0T;
```

```
}


int DeleteZCMarketData(ZCMarketData* ZCMarket)
{
    if(ZCMarket->FlatOrMarket!=0)
    {
        pnl_vect_free(&(ZCMarket->tm));
        pnl_vect_free(&(ZCMarket->Pm));
    }

    return 1;
}
#endif //PremiaCurrentVersion
```

References