

[Help](#)

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

#include <stdio.h>
#include <stdlib.h>
#include "premia_obj.h"
#include "pnl/pnl_mathtools.h"
#include "pnl/pnl_vector.h"
#include "
href../../common/math/read_market_zc/InitialYieldCurve_h_src.pdfInitialYieldC

//***** Initial yield curve *****//
void SetInitYieldCurve(int InitYieldCurve_flag, double R_flat, ZCMarketData *ZCM
{
    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 structur
void ReadMarketData(ZCMarketData *ZCMarket)
{
    FILE *Entrees;          /*File variable of the code*/

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

    Entrees = fopen(ZCMarket->filename, "r");
```

```

if (Entrees == NULL)
{
    printf("Le FICHER 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(pligne, "%lf t=%lf", &p, &tt);
        /* La ligne lue dans le fichier doit etre de la forme "0.943290 t=0.5"
        LET(ZCMarket->Pm, i) = p; /*enregistre le prix du zero coupon*/
        LET(ZCMarket->tm, i) = tt; /*enregistre 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(BondPrice(T, ZCMarket))) / (I

```

```

}

double ATMSwaptionStrike(double T_start, double T_end, double period, ZCMarketData *ZCMarketData)
{
    int i, n = pnl_iround((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, ZCMarketData);
    }
    sum *= period;

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

// Compute the ZC price P(0,T) by interpolating the initial yield curve contained in ZCMarketData
double BondPrice(double T, ZCMarketData *ZCMarketData)
{
    double POT;
    int i;

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

            if (i < ZCMarketData->Nvalue)
            {
                POT = GET(ZCMarketData->Pm, i - 1) * (GET(ZCMarketData->tm, i) - T) / (GET(ZCMarketData->tm, i) - GET(ZCMarketData->tm, i - 1)) +
                    GET(ZCMarketData->Pm, i - 1);
            }
        }
    }
    else
    {
        POT = 1.0;
    }
}

```

```

        else
        {
            POT = GET(ZCMarket->Pm, i - 1) + (T - GET(ZCMarket->tm, i - 1)) *
        }
    }

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

return POT;
}

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

    return 1;
}
#endif //PremiaCurrentVersion

```