MÄLARDALENS HÖGSKOLA

Code: MdH-IMa-2005:016

MASTER THESIS IN MATHEMATICS /APPLIED MATHEMATICS

A web-based mini system for fixed income analysis

by

Takoeta Fred and Hamadou Hamaounde

Magisterarbete i Matematik/tillämpad Matematik

DEPARTMENT OF MATHEMATICS AND PHYSICS
MÄLARDALEN UNIVERSITY
SE-721 23 VÄSTERÅS, SWEDEN

MÄLARDALENS HÖGSKOLA

DEPARTEMENT OF MATHEMATICS AND PHYSICS

Master thesis in mathematics / applied mathematics

Date:
2005-11-24

Projectname:
A web-based mini system for fixed income analysis

Author:
Takoeta Fred and Hamadou Hamaounde

Supervisor:
Jan Röman

Examiner:
Anatoliy Malyarenko

Comprising:
20 points

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENT

We would like to thank our supervisor Jan Röman for his academic and professional assistance and helping us bring our work to a higher academic level. Our special thanks go to all the lecturers in the MIMA analytical finance program whose inspiration has been of great support to us.

We specially want to thank our respective families for their overall support; this thesis couldn’t have been accomplished without them. We would extend our appreciation to all our friends especially Armand Fotsing, Paula Mbunga, Abange Arah, Susan Mengot and Aminur Roshid for all their help.

Västerås, November 2005.

Fred Tambong Takoeta Hamadou Hamaounde

ABSTRACT

The purpose of this project entitled “A web-based mini system for fixed income analysis” is to build a mini web system that is able to:

- Extract the daily risk-free yield curve from observed Treasury bond market data in Sweden, United States of America, United Kingdom and Germany.
- Value a corporate bond (risky bonds) by adding some credit spread above the risk-free yield curves.
- Compute a range of bond risk factors measures.

In addition to this, we present and review the mathematical theories and techniques for fixed income analysis.

1. INTRODUCTION

This chapter presents our motivation for choosing fixed income as our area of interest; this is presented under the background. Next we specify the purpose of our study and close the chapter with details on scope and limitations of our study.

In the last decade, the financial world has witnessed a boom in the trading of derivative securities. Beside the growth in volume, derivatives markets have also seen a growth in complexity. In the derivative trading, the fixed-income market has been one of the fastest growing**.**

The word Fixed income refers to any type of investment that yields a regular (fixed) payment. The Fixed Income world is well known for variety and is best suited for capital which does not face imminent liquidation and for which the need of preservation is important. Designed to bridge the gap between no real return and principal speculation, fixed income investors will have the return of principal on a specified maturity date, and interest payments of a defined amount and frequency. This fact attracted a lot of research into modelling the interest rates, the pricing and risk characteristics of interest rates derivatives.

Zero-coupon interest rates (spot rates) are the basic building block of fixed-income mathematics. They are widely used in Quantitative finance. Their field of applications includes bond pricing, construction and pricing of derivative products, generation of forward rates curves, estimations of premiums and stochastic interest rates modelling. Therefore for any application that requires the use of risk-free interest rates, it is necessary to use a yield curve extracted from the Treasury bonds (including bills and notes).
The ability to react instantly to changes in market situation requires on-line access to the data at any given time - presented in a way that can be interpreted simply and fast. This project entitled a web-based mini system for fixed income analysis is mainly about building a system which, from a given market data of treasury bonds extracts the zero yield curve. Among other tasks, the mini system can compute bonds yield to maturity, price risky and corporate bonds and compute the risk sensitiveness of those bonds.
To reach these goals, in this project we

- First describe and review the major concepts, mathematical theories and techniques in analyzing fixed-income. We will particularly focus our review on the concepts of Cash flows instruments, Term structure Theory and Interest rates risk sensitiveness.
- Observe risk-free markets, such as the market of treasury bonds (including bills and notes).
- And finally, we will expose the results in a web-based Graphical User Interface (GUI). The codes will be written in C/C++ using cygwin (a free "operating system with compilers etc.) and in php/html. A database, MySQL (a free database) will be used for the data storages. JpGraph (a free package written in php) will be used for the graphical output. To use php on we have to use an apache web server.

2. THE BOND INSTRUMENT

In this chapter we give a brief overview about the time Value of money, basic features and definitions, present Value and discounting, discount Factors, day-count convention, Bond Pricing and Yield: The Traditional Approach, bond pricing, bond Yield, accrued interest, clean and dirty bond prices.

To value any security, one looks at its cash flows. Since we are mainly interested in the present value of the expected cash flows, three features stand out: the magnitude and direction of the cash flows, times when each cash flow occurs and an appropriate factor to discount future cash flows. We thus consider elementary financial mathematics essential for any financial calculation.

2.1. Cash Flow Instruments

Cash flow instruments are instruments that are only treated as cash flow sequences. These instruments have different cash flow types that are described by various parameters.

Parameters
Parameters common to all cash flow are:

- **Pay Date**: the calendar day when the cash flow is paid
- **Currency**: the currency of the cash flow.
- **Zero coupon curve**: used for discounting all cash flow from payout date to valuation date.

The simplest cash flow type is a single fixed payment, Fixed Amount - the fixed amount to be paid. All other cash flows relate to interest rates payments in some way. They have the common attributes:

- **Day count**: the day count convention used for this period
- **Start day**: the date on which the interest rate period starts
- **End day**: the date on which the interest rate period ends

The simplest interest payment is the fixed coupon rate, using the attribute, Fixed Rate - the simple fixed interest rate that applies in the period.

2.2. Present and Future Value

To calculate the payoff on the equity market we use the simple rate *r* on annual basis and the invested amount *P* to calculate the amount at maturity:

$$F = P\left(1 + \frac{r}{100}\right) \qquad (2.1)$$

In the money market we usually use the following measure for the yield:

$$F = P \cdot \left(1 + \frac{r}{100} \cdot \frac{d}{360}\right) \qquad (2.2)$$

where *d* is the number of days to maturity. Using an effective annual rate r_c we get:

$$F = P \cdot \left(1 + \frac{r_c}{100}\right)^{\frac{d}{360}} \qquad (2.3)$$

For treasury bills and certificates we calculate the discounted present value with the expression:

$$P = \frac{N}{1 + \frac{r}{100} \cdot \frac{d}{360}} \qquad (2.4)$$

Here, *N* is the nominal amount and r the simple rate on yearly basis.
In general, if interest is compounded m-times per annum, the future value is

$$F = P \cdot \left(1 + \frac{r}{m}\right)^{nm} \qquad (2.5)$$

Where *n* is the number of years from now.

2.3. Discount Function

The discount function describes the present value at time t_0 of a unit cash flow at time *t*. It is usually denoted as $P(t_0, t)$ where t_0 is the current date and t is the time between t_0 and *t*.
The discount function is a monotonic decreasing function with value 1 at *t* = 0 confirming the fact that interest rates are always positive.

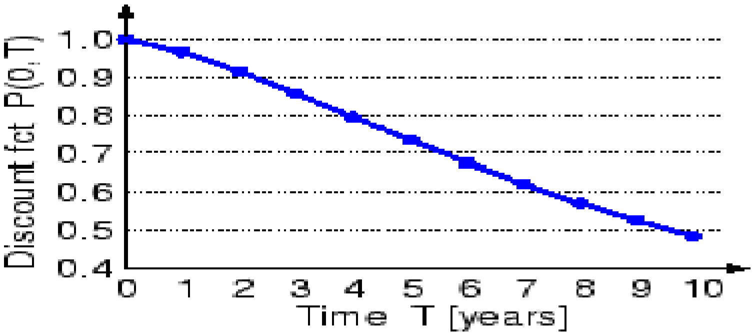


Figure 1 Graph of discount function

The relationship between the discount function and the annually compounded yield curve, using a day count convention that reflects the actual time between time t_0 and *t* measured in years, can be written as:

$$P(t)=\frac{1}{\left(1+r_1(t)\right)^t} \qquad (2.6)$$

The semi-annual is related to the discount function as:

$$p(t)=\frac{1}{\left(1+\frac{r_2(t)}{2}\right)^{2t}} \qquad (2.7)$$

The n-annual rate is related to the discount function as:

$$P(t)=\frac{1}{\left(1+\frac{r_n(t)}{n}\right)^{nt}} \qquad (2.8)$$

The continuous compounding rate is related to the discount function as:

$$p(t)=\exp\left\{-r_c(t)*t\right\} \qquad (2.9)$$

2.3.1. Conversion between compounding methods

Suppose r_1 is the interest rate with continuous compounding and r_2 is the equivalent rate compounded m times per annum. Then

$$\left(1+\frac{r_2}{m}\right)^m=e^{r_1} \qquad (2.10)$$

We get

$$r_1=m\ln\left(1+\frac{r_2}{m}\right) \qquad (2.11)$$

$$r_2=m\left(e^{r_1/m}-1\right) \qquad (2.12)$$

It is therefore important to use the correct interest rate in calculating the value of any security. Figure2 below shows a graph plotted for different compounding with the same interest rate.

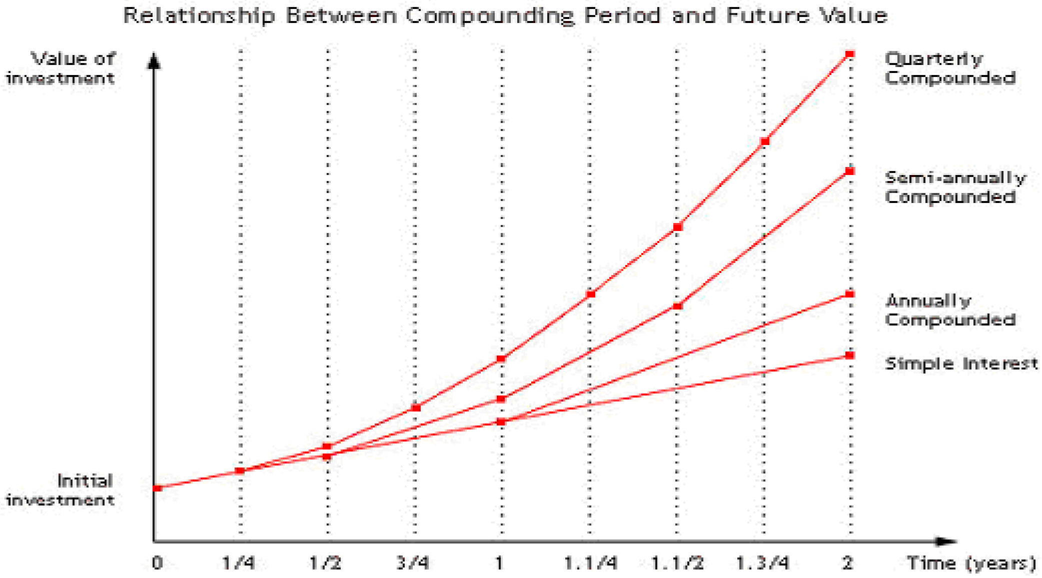


Figure 2 Relationship between compounding period and future value[1]

2.4. Day Count Conventions

When using the discount function to express yield or interest rates, the day count convention must be taken into consideration. The day count convention is a user-defined, instrument-specific parameter and is used when valuing that particular instrument.
Some alternative day count conventions are:

- 30/360
- 30E/360
- Act/360
- Act/365
- NL/365
- Act/Act

The meaning of the abbreviations used in the naming of the above conventions is as follows:

- **Act**: Actual number of calendar days.
- **NL**: Actual number of calendar days. Exception: If the year is a leap year then February is considered to have 28 days (instead of 29).
- **30**: Each month is considered to have 30 days.
  Exception 1: If the later date is the last day of February, that month is considered to have its actual number of days.
  Exception 2: When the later date of the period is the 31st and the first day is not the 30th or the 31st, the month that includes the later date is considered to have its actual number of days.
- **30E**: Each month has 30 days.
  Exception: If the later date is the last day of the month of February, that month is considered to have its actual number of days.

[1] Lecture notes in Analytical Finance II by Jan Röman

Interest rates are typically expressed for annual periods. The time period measured in years between two dates, t, is described as the fraction of the number of days between two dates, t_d, and the number of days in a year, t_y:

$$t = \frac{t_d}{t_y} \qquad (2.13)$$

t_d and t_y are determined according to the specified day count convention.

2.5. Types of Rates

There exist many different types of interest rates frequently quoted in the market and these interest rates are applicable to different situations depending on the credit risk. For the purpose of this study, we give a brief discussion of the interest rates applicable to this study and then enumerate other rates existing in the market.

● **Treasury Rates**
These are interest rates applicable to borrowing by a government in its own currency. These rates are often called risk-free rates since it is assumed that there is no chance a government will default on an obligation dominated in its own currency.

● **Benchmark rate, Base rate (UK), Prime rate (US)**
This is the lowest interest rate an investor is willing to take to make an investment in a riskless security.

● **Discount rate**
The rate used to discount a given cash flow in the future to a present value. This rate reflects the time-value of money. Sometimes discount rate = zero coupon rate.

● **Inter Bank Rates**
Rates used by large international banks trading with each other. It is given as an average they will charge each other for borrowings. Examples are: LIBOR – London InterBank Offer Rate, STIBOR – Stockholm InterBank Offer Rate, LIBID- London Interbank Bid Rate etc.

● **Repo Rate**
This is the interest rate for a repurchase agreement. This is the rate you have to pay by selling a security only to buy it back after a short period. The period could be O/N (over-night-repo), T/N (tomorrow-next) or C/W (Corporate-Week)

● **Spot rate (Zero rates)**
The spot rate is defined as the theoretical profit given by a zero coupon bond.

● **Forward rate**
The forward rate is the rate of interest implied by current zero rates for periods of time in the future.

● **Yield to Maturity (YTM)**
The rate an investor will earn if he or she keeps an interest paying security until maturity.

The list is too long but other interest rates include: mortgage rates, deposit rates, swap rates, dividend rate, accrued interest rate, credit rate etc.

2.6. Bond Pricing and Yield

The number of cash flow instruments available in the market is so many, to this effect, we limit ourselves to cash flow instruments which are applicable to this study, that is our discussion will centre around bills, notes and bonds which are the most common cash flow instruments and the starting point in financial theory. To give a clear understanding of these cash flow instruments, begin by defining and giving the basic structure of these instruments.

2.6.1. Definitions

● **Bills**
A bill is an instrument where only the nominal amount is paid or received at expiry. If the instrument is traded before expiry, it is traded at a "discount" of the nominal amount and is a true zero-coupon instrument. The lifetime of a bill is normally a year or less.

● **Notes**
A note is similar to a bond, but with a shorter lifetime normally a between one and ten years.

● **Bonds**
A bond is a debt instrument that obligates the **issuer** to pay to the **bondholder** the principal (the original amount of the loan) plus interest. Thus, a private or governmental corporation essentially issues a bond. The issuer is the legal entity, which undertakes to comply with all the terms and conditions of the bonds. The issuer is also known as: borrower, obligor or debtor. The corporation "borrows" the face amount of the bond from its buyer, pays interest on that debt while it is outstanding, and then "redeems" the bond by paying back the debt.

The most important features of a bond are:

● **Principal**
The principal is the face value of the bond. Also known as the nominal value, par value, face value or redemption amount, it is the amount, which the borrower will repay at maturity. The principal is the basis on which coupon interest is calculated. The bond certificate also states the currency in which payment will be made.

● **Coupon**
The coupon is the rate of interest on a debt security that the borrower promises to pay the holder. It is expressed as an annual percentage of the face value.

$$\text{Annual coupon amount} = \frac{\text{coupon}}{100} \times \text{Principal amount}$$

Generally, the coupon is always expressed as an annual rate, the actual coupon payment may be made in one or more instalments throughout the year, depending on the market in which a bond is issued, coupons may be payable annually, semi-annually, quarterly or monthly.

$$\text{Coupon amount} = \frac{\text{Coupon}}{100} \times \frac{\text{Principal amount}}{\text{No of coupans per year}}$$

● **Maturity Date**

A bond's maturity or redemption date tells when investors should expect to get the principal back and how long they can expect to receive interest payments. The maturity of a bond can be any length of time, although typical bond maturities range from one year to 30 years. There are three groups of bond maturities:

○ Short-term (Bills): Maturities less than 1 year
○ Medium-term (Notes): Maturities between 1 and 10 years
○ Long-term bonds (Bonds): Maturities greater than 10 years. Most bonds are 30 years or less, but bonds have been issued with maturities of up to 100 years. Some bonds never mature (perpetuities).

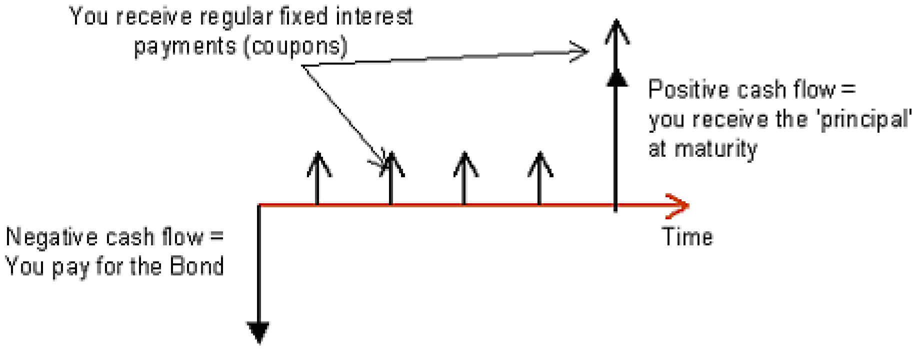

You receive regular fixed interest payments (coupons)
Positive cash flow = you receive the 'principal' at maturity
Negative cash flow = You pay for the Bond
Time

Figure 3 General representation of a bond[2]

The features described above are the structure of cash flows that the investor expects to receive if the bond was held to maturity.
Other features of a bond include:

1. Credit quality: This represents our beliefs about the issuer's capacity to repay the principal and interest. US Government bonds are considered to be the most creditworthy, since they are backed by the "full faith and power" of the government.

2. Whether the interest rate is fixed or floating.

3. Whether the lender can force the borrower to buy back the bond before maturity (which is often called an embedded put option).

4. Whether the borrower can pay back the bond at any time before the maturity date (which is called prepayment or an embedded call option).

[2] Lecture notes Analytical Finance II by Jan Röman

2.6.2. Types of bonds

Bonds may be of the following types:

● **Callable bond** is a bond where the issuer has the right, but not the obligation to call back/repurchase the bond at one or more specified points over the bond's life. If called, the issuer pays the investor the pre-specified call price, the strike. The call price is usually higher than the bond's par value. The difference between the call price and par value is called the call premium. For the investor, this means that there is uncertainty as to the true maturity of the loan.

● **Putable bonds** are bonds where the holder has the right, but not the obligation to put back the bond to the issuer at one or more specified points over the bond's life. If putted, the investor pays the issuer pre-specified put price. The difference between the put price and par value is called the put premium.
Convertible bonds (usually issued by corporations) can be converted into stock at a given price (the "conversion ratio"). Conversion events complicate the pricing because the investor is uncertain about the cash flows that will be received.

● **Zero-coupon bond** (also called a "zero" or a "pure discount bond") is a bond that has a single payment of principal at maturity, without any intermediate interest payments.

● **Bullet bond** is a conventional bond paying a fixed periodic coupon and having no embedded optionality. Such bonds are non-amortizing, i.e., the principal remains the same throughout the life of the bond and is repaid in its entirety at maturity. Bullet bonds are also called straight bonds.

● **Benchmark bullet bond** is a bullet bond issued by a sovereign government and assumed to have no credit risk (e.g., Treasury bond).

● **Non-benchmark bullet bond** is a bullet bond issued by an entity other than a sovereign government and therefore has some credit risk.

2.7. Bond Valuation

The process of determining the fair price of a bond is known as bond valuation. The fair value of a bond is the present value of the stream of cash flows it is expected to generate. Thus, the price or value of a bond is determined by discounting the bond's expected cash flows to the present using the appropriate discount rate.
The price of a bond in the market is always given as the dirty price, which is the bond’s clean price plus accrued interest since last coupon payment. The fair price of a straight bond (a bond with no embedded option) is determined by:

$$Bond\ \text{Price} = \frac{c/n}{\left(1+r/n\right)} + \frac{c/n}{\left(1+r/n\right)^2} + \frac{c/n}{\left(1+r/n\right)^3} + \ldots. + \frac{principal + c/n}{\left(1+r/n\right)^{n\times t}}$$

Where

c = Coupon rate and $C = r \times F$
r = Return on the investment (the yield),
n = Number of coupon payments per year (= compounding period)
t = Number of years
F = Nominal or Principal
T = maturity of bond

Alternatively we could use the formula

$$P = \frac{F}{(1+r)^T} + \sum_{t=1}^{T} \frac{c}{(1+r)^t} \qquad (2.14)$$

The market price of the bond is calculated using the yield to maturity (*YTM*) as the discount rate. The market price of the bond is given as:

$$Market\ \text{Price} = \frac{F}{(1+YTM)^T} + \sum_{t=1}^{T} \frac{c}{(1+YTM)^t} \qquad (2.15)$$

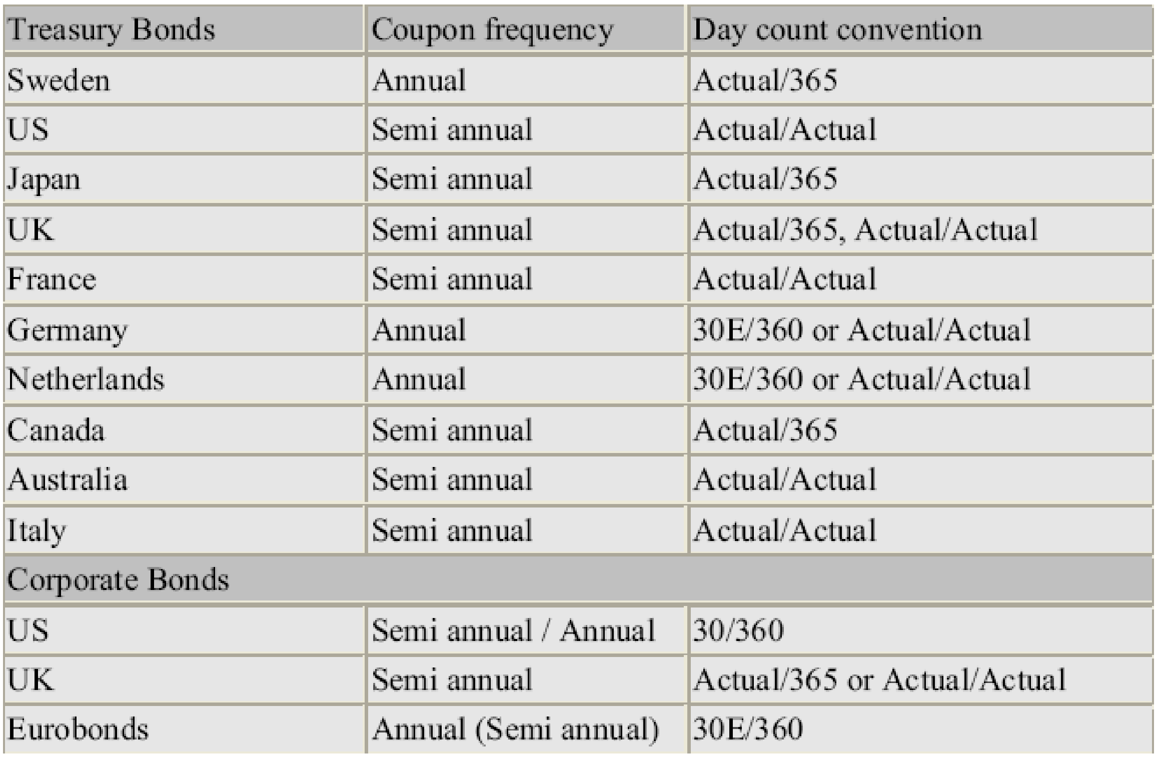
The industry-standard formula for calculating the present value of a bond is given by:

$$\text{Bond Price} = \frac{c/n}{\left(1+r/n\right)^a} + \frac{c/n}{\left(1+r/n\right)^{a+1}} + \frac{c/n}{\left(1+r/n\right)^{a+2}} + \ldots + \frac{\text{Pr}\,incipal + c/n}{\left(1+r/n\right)^{a+n}}$$

All symbols are as describe above and:

$$a = \frac{\text{Number of days to next coupon}}{\text{Number of days in current coupon period}}$$

The table below shows the coupon frequencies and the day count convention for typical bonds in some countries:



Treasury Bonds	Coupon frequency	Day count convention
Sweden	Annual	Actual/365
US	Semi annual	Actual/Actual
Japan	Semi annual	Actual/365
UK	Semi annual	Actual/365, Actual/Actual
France	Semi annual	Actual/Actual
Germany	Annual	30E/360 or Actual/Actual
Netherlands	Annual	30E/360 or Actual/Actual
Canada	Semi annual	Actual/365
Australia	Semi annual	Actual/Actual
Italy	Semi annual	Actual/Actual
Corporate Bonds		
US	Semi annual / Annual	30/360
UK	Semi annual	Actual/365 or Actual/Actual
Eurobonds	Annual (Semi annual)	30E/360

Table 1.0 Coupon frequencies of some countries

2.7.1. Accrued Interest rate

Accrued interest is the fraction of the coupon payment the bond seller earns for holding the bond for a period of time between bond payments. That is t he interest that has accumulated on a bond since the last interest payment up to but not including the settlement date. The bond price's inclusion of any interest accrued since the last payment period determines whether the bond's price is “dirty” or “clean.”

It should be noted that the market price of bonds quoted in the newspapers are clean prices. That is, they are quoted without any accrued interest. The relationship between the dirty price and clean price is given by:

$$\textit{Clean}\text{ price} = \text{Dirty price - Accrued interest}$$

Where

$$\text{Accrued interest} = \frac{c}{n} \times \text{Fractional coupon period}$$

$$\text{Fractional coupon period} = \frac{\text{Number of days since last coupon}}{\text{Number of days in current coupon period}}$$

c = Coupon rate
n = Number of coupon payments per year

The diagram below shows the relation between the dirty price and the clean price.

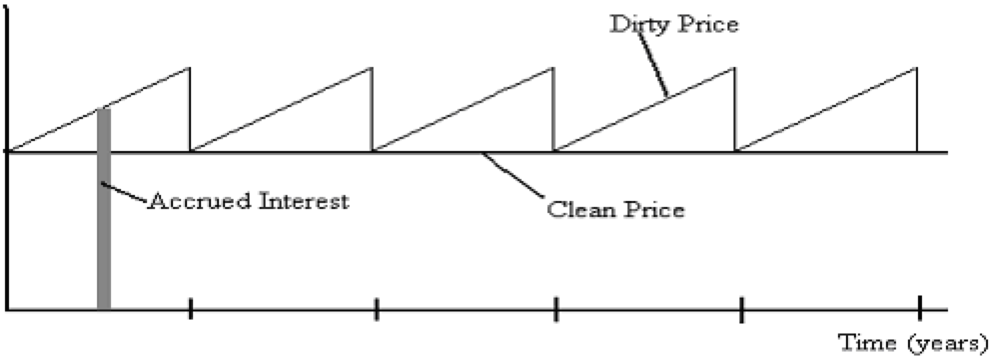


Figure 4 Relation between clean price, dirty price and accrued interest[3]

2.8. Yields

The term yield refers to the return of an investment and it is usually expressed as a percentage. It is quoted in many ways and care must be taken as to which yield is meant. On The *Wall Street Journal* on august 26, 1997, for instance, a corporate bond is quoted as follows.

Company	*Cur.Yld*	*Vol*	*Close*	*NetChg*
$ATT8\frac{5}{8}31$	8.1	162	$106\frac{1}{2}$	$-\frac{3}{8}$

This is a bond issued by AT&T (American Telephone & Telegram), maturity 2031 with a nominal yield of $8\frac{5}{8}$%, which is the same as the coupon of the bond and is part of the identification of the bond.

- **Current yield**

The current yield of a bond is the coupon rate of interest divided by the current market price.

$$\text{Current Yield} = \frac{\text{coupon rate}}{\text{clean price}} \times 100 \qquad (2.16)$$

The adjusted current yield is given as

$$\text{Adjusted current yield} = \frac{\text{coupon rate} + (100 \times \text{clean price})/t}{\text{clean price}} \times 100$$

where t is the number of years to maturity with the use of day count convention. It should be noted that the current and adjusted yields have the flaw they ignore any interest earned from the reinvestment of coupons.

[3] Lecture notes in Analytical Finance II by Jan Röman

• **Bond Yield**
The yield of a coupon-bearing bond is the discount rate that equates the cash flows on the bond to its market value.

• **Par Yield**
The par yield of certain maturity is the coupon rate that causes the bond price to equal its face value.

• **Yield to Maturity (YTM)**
This is the rate of return expected on a bond if it is held until the maturity date. The calculation of YTM takes into account the current market price, par value, coupon rate and time to maturity and it is also assumed that all coupons are reinvested at the same rate. This is the most widely used measure of return in the bond markets, and in fact when market participants speak of yield they typically mean yield to maturity (YTM).
The calculation of a bond's YTM is complex and involves trial and error; using some numerical methods such as the Newton-Raphson method and the Bisection method usually does it.

2.9. Numerical Methods for finding yields

The yield problem amount to solving the equation

$$f(y) - p = 0 \qquad (2.17)$$

where

$$f(y) = \sum_{t=1}^{T} \frac{c_t}{(1+y)^t} \qquad (2.18)$$

It is assumed that $y > -1$. The function *f(y)* is monotone in y if the c_t's are all positive. In this case, a simple geometric argument in Figure 5 shows that a unique solution exists.

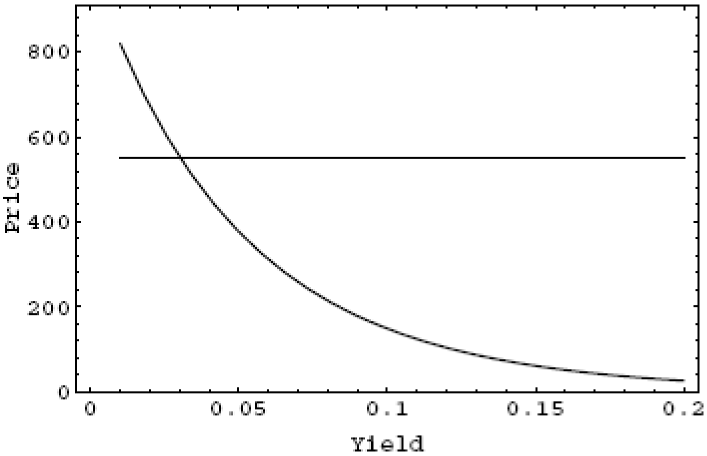


Figure 5 Computing Yields

The yield is the number at which the current market price represented by the horizontal line, equals the present value of future cash flows, represented by the downward-sloping curve. It is the value on the x-axis at which the two curves intersect.
In this project, we approximated the yield with algorithmic approach that is with the two methods listed above.

2.9.1. Bisection Method

This method is the simplest to solve equations of the form $f(y) - p = 0$.
We observe that if $f(a)f(b) < 0 \Rightarrow f(\xi) = 0$ for some $\xi \in [a \quad b]$
If we evaluate f at the mid-point c between a and b then either

1. $f(c) = 0$
2. $f(a)f(c) < 0$
3. $f(c)f(b) < 0$

In the first case we are done, in the second case, we continue the process with the new brackets [a, c] and in the third case with [c, b]. It should be noted that the bracket is halved and after n steps we will have confirmed ξ within a bracket of length $(b\text{-}a)/2^n$.

Algorithm for the Bisection method

Input: ε, a and b ($b > a$ and $f(a)f(b) < 0$);

```
real    length, c;
length = b-a;

While [length > ε] {
  c = (b-a)/2;
  if [f(c) = 0 ] return c;
  else if [f(a)f(c) < 0] b = c;
  else a = c;
}
return c;
```

The number ε is an upper bound on the absolute error of the returned value $c := f(\xi) = 0$ for some ξ such that $|\xi\text{-}c| \leq \varepsilon$. The interval [$a$, b] guarantees the existence of a root with the condition that $f(a)f(b) < 0$

2.9.2. Newton-Raphson Method

The Newton-Raphson method is an iterative method that converges faster than the Bisection method. In this method, we start with a first approximation x_0 to a root. Successive approximations are computed as

$$x_0, f(x_0), f(f(x_0)), \ldots\ldots\ldots$$

for some function f.

The necessary condition for convergence of such a procedure to a root ξ is

$$\left|f^{'}(\xi)\right| \leq 1$$

where $f^{'}$ denotes the derivative of f. The Newton-Raphson method picks

$$x_{k+1} = x_k + \frac{f(x_k)}{f'(x_k)} \qquad (2.19)$$

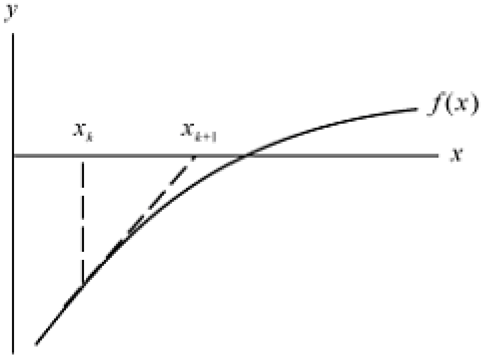


Figure 6 Newton-Raphson Method

Algorithm for the Newton-Raphson method

```
Input:  $\varepsilon$, $x_{initial}$
real    $x_{new}$, $x_{old}$;
$x_{old}$ = $x_{initial}$;
$x_{new}$ = $\infty$;

While [|$x_{old}$ - $x_{new}$| > $\varepsilon$]
  $x_{new}$ = $x_{old}$ - $f(x_{old})/f'(x_{old})$;
return $x_{new}$;
```

3. The Term Structure of Interest Rates

This section we examine zero-coupon bonds, fundamental concepts of term structure of interest rates zero-coupon bonds the spot and forward yield curve, calculating spot rates,

3.1. Term structure theories

Yield curve is a graphical representation of yield plotted against time or maturity for a particular class of bonds. For example one can construct a Treasury yield curve, an AAA corporate yield curve, a BBB yield curve, and so on.

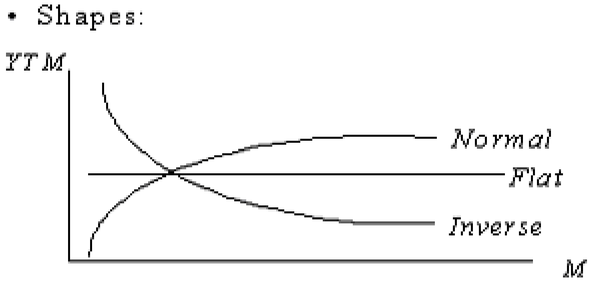


Figure 7 Yield Curves shape

Whether they are derived from current, spot, or averages rates, empirically generated yield curves have tended to take on one of the three shapes shown in figure 7. Yield curves can be positively sloped with long-term rates being greater than shorter-term ones. Such yield curves are called ***normal*** or ***upward sloping curves***. They are usually convex from below, with the YTMs flattening out at higher maturities. Yield curves can also be negatively sloped, with short-term rates greater than long-term ones. These curves are known as ***inverted*** or ***downward sloping yield curves***. Like normal curves, these curves also tend to be convex, with the yields flattening out at the higher maturities. Finally, yield curves can be relatively flat, with YTMs being invariant to maturity, or they can have both positively-sloped and negatively sloped portions, known as a ***humped yield curve.***

It is natural to ask what determines the shape of the zero curves? Why is it sometimes upward sloping, sometimes downward sloping and sometimes partly upward sloping, partly downward sloping?. A number of different theories have been proposed.
The **Expectations theory:** Although this theory is a widely accepted explanation of the term structure, it is rejected by most empirical studies. The theory also implies that all bonds, regardless of their maturities, are expected to earn the same riskless return. This is not credible either, because it means investors are indifferent to risk.

The Market Segmentation Theory (MST) posits that investors and borrowers have strong maturity preferences that they try to attain when they invest in or issue fixed income securities.

The theory that is in some ways most appealing is **Liquidity preference theory**, which argues that forward rates should always be higher than expected future zero rates.
The ***Liquidity Preference Theory,*** also referred to as the ***Risk Premium Theory (RPT)***, posits that there is a liquidity premium for long-term bonds over short-term bonds.

The basic assumption underlying the theory is that investors prefer to preserve their liquidity and invest funds for short periods of time. Borrowers on the other hand usually prefer to borrow at fixed rate for long period of time. If the interest rates offered by banks and other financial intermediaries corresponded to expectation theory, long term long term interest rates would equal the average of expected future short-term interest rates. In the absence of any incentive to do other wise, investors would tend to deposit their funds for short time periods, and borrowers would tend to choose to borrow for long time periods. Financial intermediaries would then find themselves financing substantial amounts of long-term fixed-rate loans with short- term deposits. Excessive interest rate risk would result. In practice, in order to match depositors with borrowers and avoid interest rate risk, financial intermediaries raise long- term interest rate relative to expected future short-term interest rates. This strategy reduces the demand for long-term fixed-rate borrowing and encourages investors to deposit their funds for long term. Thus, if the yield curve were initially flat, but had no risk premium factored in to compensate investors for the additional volatility they assumed from buying long-term bonds, then the demand for long-term bonds would decrease and their rates increase until risk-averse investors were compensated.
Liquidity preference theory leads to a situation in which forward rates are greater than expected future zero rates. It is also consistent with the empirical result that yield curves tend to be upward sloping more often than they are downward sloping.

As we can see, each of these theories by itself is usually not sufficient to explain the shape of a yield curve; rather, the full explanation underlying the structure of interest rates depends on elements of all theories.

3.2. Spot rates

The spot rates, also known, as the zero rates are the key element in the fixed- income pricing. This because Yield to maturity can give a misleading impression of return since it implicitly assumes that all intermediate payments are reinvested and earn the same rate of return. The term structure of interest rates allows a more precise presentation and analysis of expectations in the bond market and ensures enhanced international comparability of the estimation results. Moreover, (implied) forward rates can be calculated directly from the term structure of spot rates.

The *n*-year spot rate (zero rates) is the rate of interest earned on an investment that starts today and lasts for *n*-years. All the interest rates and principal is realized at the end of n- years.
In other words The *t*-period spot rate $S(t)$ is the yield to maturity of a *t*-period zero-coupon bonds .Hence, $(1+S(t))^{-t}$ is the present value one dollars *t* -periods from now. In particular one-period spot rate, called the short rate, plays an important role in modelling interest dynamics. It is also worth to mention that the stochastic term structure models rely on zero-coupon rates, but never on yield to maturity. A spot rate curve is a plot of spots rate against maturity. Its other name includes spot yield curve and zero-coupon yield curve.
A major shortcoming of the yield to maturity concept can be seen clearly with the spot rate idea. If recall the bond price formula,

$$Bond\ \text{price} = \frac{c/n}{\left(1+r/n\right)} + \frac{c/n}{\left(1+r/n\right)^2} + \frac{c/n}{\left(1+r/n\right)^3} + \ldots + \frac{\Pr incipal + c/n}{\left(1+r/n\right)^{n\times t}} \qquad (3.1)$$

A single interest is used to discount all future cash flows. Specifically, every cash flow is discounted at the same rate *r*. To see the inconsistency of this methodology, consider two riskless bonds of equal quality but different yield s to maturity because of their different cash flow patterns. The above methodology would discount their cash flows with different rates, and this will obviously lead to an arbitrage opportunity. We can conclude that the correct way to value bonds is to discount their cash flows by spot rates. If the market does not value bonds this way, then arbitrage opportunities would exist by buying the bond and stripping it, or buying stripped securities and rebonding them. We remark then that the yield to maturity is roughly a weighted sum of spot with each weigh being proportional to the dollar duration of the cash flow.

In the past, many analysts constructed yield curves from yield data on Treasury securities or different corporate bonds. The problem with generating yield curves in this way, is finding a sample of bonds that are identical in all respects except maturity (i.e., the same coupon rates, risk, option features, and the like). Today, the convention is to generate a spot yield curve showing the relation between the spot rates and maturity.

3.3. Extracting yield curves from bond prices

As we argued before, the clearest picture of the term structure of interest rates is obtained by looking at the yields of zero-coupon bonds of different maturities. However, most traded bonds are coupon bonds, not zero-coupon bonds. Although Treasury STRIPS can be used to represent these theoretical risk-free spot rates, there are some problems with this approach. The main one is that the Treasury STRIPS market is less liquid than the treasury coupon market. It is thus necessary to extract spot rates from yields of coupon bonds of different maturities.

This section discusses methods to extract or estimate a zero-coupon yield curve from the prices of coupon bonds at a given point in time. The standard methods that are used to estimate spot rates are the bootstrapping and statistical techniques

3.3.1. Bootstrapping

Here we consider the so-called bootstrapping technique. It is sometimes possible to construct zero-coupon bonds by forming certain portfolios of coupon bonds. If so, we can deduce an arbitrage-free price of the zero-coupon bond and transform it into a zero-coupon yield. This is the basic idea in the bootstrapping approach.

Any coupon bond can be viewed as a package of zero-coupon bonds. In other words, a bond with cash flow C_i at time *i* and a face value *F* at maturity n for $1 \le i \le n$ is equivalent to a package of zero-coupon bonds where the i^{th} bond pays C_i dollars at time *i*. A bond can be therefore priced by using the formula of the present value (*PV*)

$$PV = \sum_{i=1}^{n} d(i)c_i + Fd(n) \qquad (3.2)$$

Let's recall that *d(i)* here are the discount factors and are obtained by the following relationship

$$d(i) = \frac{1}{(1+S(i))^i}, \quad \text{for } i = 1, 2........n, \qquad (3.3)$$

where $S(i)$ is the i-period spot rate(zero rate).

Suppose we have a market where two bonds selling at P_1 and P_2 and, maturing respectively at period one and period 2 respectively are traded. The second bond pays a coupon of c per period. The first discount factor $d(1)$ is easily derived (short term treasury are pure discount securities). $d(2)$ is therefore discounted using the following relationship

$$P_2 = Cd(1) + (C+F)d(2).$$

The example above can easily be generalized to more periods. Suppose we have M bonds with maturities of 1, 2, . . .,M periods, respectively, one payment date each period and identical payment date and with market prices B_i and cash flows Let Y_{in} denote the payment of bond i in n periods. Then we can construct successively zero-coupon bonds for each of these maturities and hence compute the market discount factors $d(1)$, $d(2)$,... d(n) by solving the following equation:

$$B_i = \sum_{n=1}^{M} d(n) Y_n \qquad (3.4)$$

First, $d(1)$ is computed using the shortest bond. Then $d(2)$ is computed using the next-to-shortest bond and the already computed value of $d(1)$.The following discount factors are recursively computed using the bond price equation and all the previous discount factors, etc. Finally, given the discount factors $d(1)$, $d(2)$,... d(n) , we can compute the zero-coupon interest rates and hence the zero-coupon yield curve up to time M (for the M selected maturities). This approach is called **bootstrapping** or **yields curve stripping**.

In reality, computing the spot rates is not as clean-cut as the above bootstrapping procedure might suggest .In practice, some maturities might be missing from the data points, for instance in case of a market where bonds pays semi-annual coupons, we may not encounter exact maturities like 0.5 years,1 years,1.5 years etc. The approach often used by analysts is to interpolate between bond cash flows i.e. bond prices and coupons. If some maturities are missing, the requisite coupon rate, $c(n)$, is replaced by interpolation between $c(n_1)$ and $c(n_2)$, where the adjacent maturities n_1 and n_2 satisfy $n_1 < \text{n} < n_2$. In general, linear interpolation is used. The missing values are replaced by $c(n)$, and the recursive bootstrap computations are performed as before.

Let's notice that only in bond markets with sufficiently many coupon bonds with regular payment dates and maturities can the bootstrapping approach deliver a decent estimate of the whole zero-coupon yield curve. In other markets, alternative methods are called for. We will therefore examine in the following two sections the so-called statistical techniques.
It is worth to mention that this method is mainly used on the swap market and one important feature of the interest rate swaps is the availability of data for a wide range of maturities, except for some few maturities dates, where therefore linear interpolation of swaps data maybe be used and as stated by many researches this application generates good results. However an attempt to use the same procedure to the Treasury bonds data failed. We encountered unwilled results such as negative spot rates, and fluctuations of the yield curve,

which is due to the limited number of available bonds. For these reasons we abandoned this initiative.

One of the most important questions to be answered in determining an empirical term structure from Treasury bond data is: What shape is the yield curve allowed to take? To answer this question, we need to find a balance between two criteria: smoothness and flexibility. There must not be over-smoothing as this might cause the elimination of valuable market information. Though the decision on this balance is subjective, certain properties are essential; in particular, an estimated discount function should be both positive and monotonic non-increasing (to avoid negative forward rates) and equate unity at time zero (The present value of a unit today is a unit)!

To overcome these problems in the last years a number of methods were proposed by different researchers for yield curve construction and among the most appealing, the splines interpolation that is will discuss further.

3.3.2. Bootstrapping with Splines Interpolation

What needs to be stated here is the specificity of the interpolations schemes used in the yield curve extraction: the interpolation methods are intimately connected to the bootstrapping process. This therefore makes the process more complicated than the usual interpolation between given data points.

The approach of using piecewise polynomials functions or splines to approximate the discount function from observed bond data was first suggested by McCulloch (1971). The word spline indicates that the maturity axis is divided into subintervals and that the separate functions (of the same type) are used to describe the discount function in the different subintervals. The reasoning for doing this is that it can be quite hard to fit a relatively simple functional form to prices of a large number of bonds with very different maturities.
According to Rice (1969) 'spline functions are the most successful approximating functions for practical applications so far discovered'.A polynomial spline is a number of separate polynomial functions (usually of same degree and structure), joined at a number of so-called knot points. To ensure a continuous and smooth term structure of interest rates, one must impose certain conditions for the maturities separating the subintervals. The condition is that the spline functions are continuous and differentiable and ensure a smooth transition in the knot points. One the main advantages of splines is their flexibility in for modelling special curvature in different region of the term structure.

3.3.3. Cubic splines

The approach is quite similar to the form of cubic spline described by Mastronikola (1991). The difference being that his method is fitted to the redemption yield rather than to the discount function.

To simplify the notation in what follows, let $d(\tau)$ denote the discount factor for the next τ periods . Hence, the function $d(\tau)$ for τ represents the time t market discount function. In particular, $d(0) = 1$. We will use a similar notation for zero-coupon rates and forward rates: $y(\tau)$ and $f(\tau)$. The methods studied in this and the following sections are both based on the assumption that the discount function $d(\tau)$ can be described by some functional form

involving some unknown parameters. The parameter values are chosen to get a close match between the observed bond prices and the theoretical bond prices computed using the assumed discount function. Given prices for *M* bonds with time-to-maturities of $T_1 \leq T_2 \ldots \leq T_M$. Divide the maturity axis into subintervals defined by the "knot points"
$0 = \tau_0 < \tau_1 < \ldots \tau_k = T_M$. Let's consider a maturity point τ, where $\tau_j < \tau < \tau_{j+1}$ spline approximation of the discount function $d(\tau)$ is based on an expression like

$$d(\tau) = \alpha_j + \beta_j(\tau - \tau_j) + \gamma_j(\tau - \tau_j)^2 + \delta_j(\tau - \tau_j)^3 \qquad (3.5)$$

and α_j, β_j, γ_j, and δ_j are constants and $j = 1,2,\ldots,k$.

For $\tau \in [0, \tau_1)$, we have

$$d(\tau) = \alpha_0 + \beta_0\tau + \gamma_0\tau^2 + \delta_0\tau^3 \qquad (3.6)$$

Since $d(0) = 1$, we must have $\alpha_0 = 1$. For $\tau \in [\tau_1, \tau_2)$, we have

$$d(\tau) = \alpha_1 + \beta_1(\tau - \tau_1) + \gamma_1(\tau - \tau_1)^2 + \delta_1(\tau - \tau_1)^3 \qquad (3.7)$$

To get a smooth transition between (3.6) and (3.7) in the point $\tau = \tau_1$, we demand that

$$d(\tau_1 -) = d(\tau_1 +), \qquad (3.8)$$

$$d'(\tau_1 -) = d'(\tau_1 +), \qquad (3.9)$$

$$d''(\tau_1 \text{-}) = d''(\tau_1 +), \qquad (3.10)$$

The condition (3.8) ensures that the discount function is continuous in the knot point τ_1. The condition (3.9) ensures that graph of the discount function has no kink at τ_1 by restricting the first-order derivative discount function to approach the same value whether t approaches τ_1 from below or from above. The condition (3.10) requires the same to be true for the second-order derivative, which ensures an even smoother behaviour of the graph around the knot point τ_1.We obtain this by setting both the first and second derivatives at each knot points to be continuous. Solving for the parameters α_j, β_j, γ_j, and $\delta_{j\text{ of}}$ the above cubic spline leads to a system of equations.

Let $t_1, t_2 \ldots t_N$ denote the time distance from today (date *t*) to the each of the payment dates in the set of all payment dates of the bonds in the data set. Let Y_{in} denote the payment of bond *i* in t_n periods. From the no-arbitrage pricing relation, we should have that

$$P_i = \sum_{n=1}^{N} Y_{in} d(t_n) \qquad (3.11)$$

Where P_i is the current market price of bond *i*. Here too, the results were rather disappointing: After some good results in the first intervals (due probably to the fact that the first bonds are zero coupon bonds). Besides the complexity of the algorithm we encountered difficulties with frequent oscillations in the discount function. Shea (1984) considers some of the pitfalls to model the term structure: He demonstrates that the constraints used in cubic spline scheme do not in fact restrict the discount function to its desired negative slope. This leads us to decide for a simpler method, which we will introduce in the next sub section.

3.3.4. Simple Interpolation Method: Linear splines

The principles in constructing the yield curve here is similar to the cubic spline. The main difference is that a polynomial of degree one is rather used. We fixed each Treasury bond maturity as a knot point.
Let's consider a maturity point τ, where $\tau_j < \tau < \tau_{j+1}$ spline approximation of the discount function $d(\tau)$ is based on an expression like

$$d(\tau) = b_j + a_j(\tau), \tag{3.12}$$

and a_j ,b_j, are constants and j=0, 1……..k.

For τ ϵ [0, τ$_1$), we have

$$d(\tau) = b_0 + a_0\tau \tag{3.13}$$

Since d (0) = 1, we must have b_0= 1. For τ [τ_1, τ_2), we have

$$d(\tau) = b_1 + a_1(\tau) \tag{3.14}$$

Unlike the completed form of cubic splines, here only one condition is required:

$$d(\tau_1 \text{ -}) = d(\tau_1 +), \tag{3.15}$$

The other favourable issue is that on the market the shortest maturity bond is always a zero coupon bond, finding a$_0$ is therefore just to solve an equation with one unknown.
Using the formula

$$P_i = \sum_{n=1}^{N} Y_{in} d(t_n) \tag{3.16}$$

and from the condition (3.15) we obtain a linear system of two(2) equations with two(2) unknowns at each knot point starting from the second available bond.

3.4. Forward rates

The yield curve contains not only the prevailing interest rate structure but also information regarding future interest rates currently expected by the market known as forward rates. The concept of forward rates in fixed-income pricing is as much important as that of spot rates. Forward rates are, defined, as the rates of interest implied by current zero rates for periods of time in the future.

Since *s(i) is* the *i*-period spot rate from now *we* define \$1 today will yield (1+ *s(i))*^{i} *at time i* . Suppose somebody wants to invest \$1 in riskless securities for *i* periods and, at time *for* two years. Two strategies open to him.

Strategy 1 is to put the money in a *j (j<i)* time *spot* rates *s(j)*. And then at the end of the *j*-time, he takes his money and invest at *spo*t rates *s(j, i)* for another *i-j* period. Let us call the rate this second interest is, the spot rate of interest at time *j* on a zero coupon bond maturing at time *i*. The final payoff to this strategy is

$$(1+ s(j))^{j}\,(1+ s(j,i))^{i\text{-}j}$$

Instead of making two separate investments of 1 year each, he could invest his money today in a zero-coupon bond that pays off in time *i*. If the i-time spot rate is *s(i)*, the final payoff is

$$(1+ s(i))^{i}$$

The first strategy can be interpreted as investing for *j*-time at spot rate is *s(j)*, and then for the *i-j* time at a forward rate *f(j,i)*. This implicit forward rate guaranteed by the arbitrage condition and can be expressed as

$$(1+ s(j))^{j}\,(1+ s(j,i))^{i\text{-}j} =(1+ s(i))^{i} \;\Rightarrow f(j,i)=\left[\frac{(1+S(i))^{i}}{(1+S(j))^{j}}\right]^{1/(i-j)}-1$$

The formula for implied forward rates is based on an arbitrage argument, where the rate for a specific nominal amount between two future dates can be locked in by borrowing and lending at the current rates to the future dates.

In the case of continuous compounding, if r_1 is the spot rate of interest applying for maturity T_1 and r_2 is the spot rate of interest applying for maturity T_2 where $T_2 > T_1$, the forward interest rate for the period of time between T_2 and T_1 is given by

$$f(T_1,T_2)=\frac{r_2T_2-r_1T_1}{T_2-T_1} \qquad (3.17)$$

Equation can be also written

$$f(T_1,T_2)=r_2+(r_2-r_1)\frac{T_1}{T_2-T_1}$$

This shows that if $r_2 > r_1$ (upward sloping), then $f > r_2 > r_1$ so that forward rates are higher than zero-coupon yields. On the other hand, if $r_2 < r_1$ (downward sloping), then $f < r_2$ so that zero coupon yields are higher than forward rates. This is shown in Figure8.

Taking limits as T_2 approaches T_1 in the previous equation; we obtain the instantaneous forward rate for a maturity *T*:

$$f=r+T\frac{\partial r}{\partial T}$$

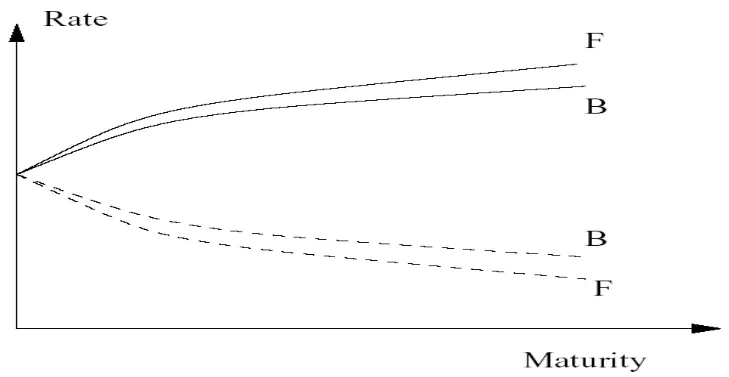


Figure 8 Forward rate curve (F) for upward sloping and downward sloping zero rate curve (B)

4. Risk Measures

In this section we talk about some key measures frequently used by Financial companies to hedge themselves against interest rate risk. We first begin by the reviewing the notion of Static spread Which is fundamental in corporate bonds pricing.

4.1. Static spread

Bond traders have developed procedures for taking credit risk into account when pricing corporate bonds. This is usually done by adding a constant credit spread to the Treasury spot rate curve to reflect the risk premium.

Consider a risky bond with cash flows C_1, C_2 … C_n and selling for P. The equivalent benchmark price $P*$ using spot rates from riskless Treasury security would be:

$$P* = \sum_{i=1}^{n} C_i d(i) = \sum_{i=1}^{n} C_i (1+S(i))^{-i} \qquad (4.1)$$

where $P^* > P$. We therefore, look for the rate s, when added to the spot rate curve, gives the bond's market price, that is,

$$P = \sum_{i=1}^{n} C_i (1+s+S(i))^{-i} \qquad (4.2)$$

This s is called the static spread. Static spread is hence the spread that a risky bond would realize over the entire Zero curve (Treasury) if the bond is held to maturity; it signifies that the amount the whole spot rate curve should shift in parallel in order to price the bond correctly.

Supposing that the spot rate curve is flat at r (that is of course unrealistic!). If a bond is selling for P with cash flows $C, \ldots C$, $C + F$, then the static spread is such that

$$P = \sum_{t=1}^{n} C(1+r+s)^{-t} + F(1+r+s)^{-t} \qquad (4.3)$$

Traditionally, spread refers to the difference between the yield to maturity of the bond under consideration and that of a Treasury security of comparable maturity. This spread is called Nominal spread. And we should mention that the static spread is inherently more accurate.

4.2. Duration

The duration of a bond, as its name implies, is a measure of how long in average the holder of the bond has to wait before receiving cash payments. A zero-coupon bond that matures in n years has duration of n years. However a coupon- bearing bond maturing in n years has a duration of less than n years, because the holders receives some of the cash payments prior to year *n*.

Suppose that a bond provides the holder with cash flows c_i at time $t_i (1 \le i \le n)$. The price, *B*, and yield, *y* (continuously) compounded, are related by

$$B = \sum_{i=1}^{n} c_i e^{-yt_i} \tag{4.4}$$

The duration, *D*, of the bond is defined as

$$D = \frac{\sum_{i=1}^{n} t_i c_i e^{-yt_i}}{B} \tag{4.5}$$

This can be written as

$$D = \sum_{i=1}^{n} t_i \left[\frac{c_i e^{-yt_i}}{B} \right] \tag{4.6}$$

The term in square brackets is the ratio of present value of the cash flow at time t_i to the bond price. Given the bond price is the present value of all payments, the duration can therefore be interpreted as a weighted average of the times when payments are made. From equation (4.4),

$$\frac{\partial B}{\partial y} = \sum_{i=1}^{n} - c_i e^{-yt_i} \tag{4.7}$$

This can be rewritten as

$$\frac{\partial B}{\partial y} = -BD \tag{4.8}$$

If we make a small parallel shift to the yield curve, increasing all interest rates by a small amount, δy, the yields on all bonds also increase by δy. The previous equation shows that the bond's price increases δB, where

$$\frac{\delta B}{\delta y} = -BD$$

or

$$\frac{\delta B}{B} = -D\delta y \tag{4.9}$$

Equation (4.9) is the key relationship. It shows that the percentage change in the bond price is, (approximately) equal to the duration multiplied by the size of the parallel shift in the yield curve. The preceding analysis is based on the assumption that the yield to maturity is expressed in continuous compounding. In case of discretely compounding, if *y* is expressed with a compounding frequency of *m* times a year,

$$\delta B = -\frac{BD\delta y}{1 + y/m}$$

A variable $D*$ defined by

$$D* = -\frac{D}{1 + y/m} \qquad (4.10)$$

is called the ***modified duration***. The duration, with y expressed with a compounding frequency of m times a year is therefore simplified to

$$\delta B = -BD * \delta y \qquad (4.11)$$

4.3. Convexity

The convexity is defined as the derivative of the duration with respect to the yield or as the second order derivative of the bond price with respect to time. This is the corresponding measure to gamma in option theory. The formula is given by:

$$\frac{1}{B}\frac{\partial B}{\partial y} = \frac{1}{B}\sum_{i=1}^{n} t_i^2 c_i e^{-yt_i} \Rightarrow \frac{\Delta B}{B} = -D.\Delta y + \frac{1}{2}C.(\Delta y)^2 \qquad (4.12)$$

The duration relationship applies only to small changes in yields. This is illustrated in figure 9 below, which shows the relationship between the percentage in price and change in yield for two bond portfolios having the same duration. The gradients of the two curves are the same at the origin. This means that both portfolios change in value by the same percentage for a small yield changes. This is consistent with equation (4.9). The difference in Convexity (curvature) can become important with large changes in the yield.

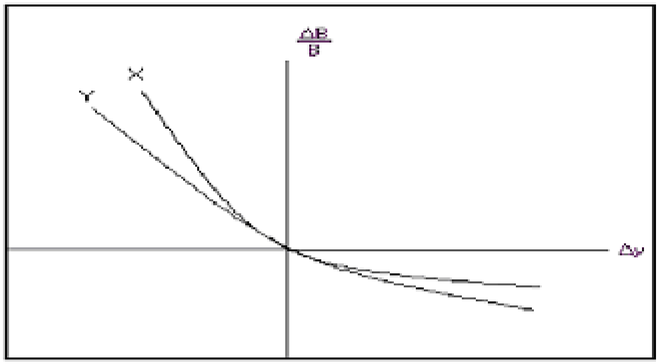


Figure 9 Two bond portfolios with the same duration

4.4. Other Risk Measures

4.4.1. Delta yield

The yield curve delta shows the change in the present value, given a shift of one basis point in all yield curves used. The shift is applied to the annually compounded zero coupon curve, using the day count fraction Act/365.
The yield delta can either refer to an upward or a downward shift of yields. The general **Delta Yield** formula is

$$\Delta_{yield} = \left[PV(r+h) - PV(r)\right] * scale \qquad (4.13)$$

where $h = \pm 0.00001$ and $scale = 1000$.

Normally, the shift step used in the calculations is thus actually 1/1000 of a basis point, but the result is scaled to a one basis point shift. The delta can also be broken down according to different time buckets, to illustrate the sensitivity to a particular shift in a given time bucket.

4.4.2. Gamma yield

The *Gamma yield* formula can be represented as

$$\Gamma_{yield} = \left[\partial(y+h) - \partial(y)\right] = \left[PV(y+2h) - 2PV(y+h) + PV(y)\right] * scale \qquad (4.14)$$

The yield curve gamma can, like the delta, be broken down into different time buckets. If this is the case, the gamma value shows the change in delta for the corresponding time bucket given a one basis point change in the yield curve as a whole and not just in the individual bucket.

4.4.3. Theta

The *Theta* value shows the change in present value (*PV*) from the valuation date until the next calendar date, given unchanged market conditions. Unchanged market conditions here imply that the yield curve will look the same on both days. For generic periods, the zero coupon rates are the same on both days, while all rates for fixed dates will be rolled down the curve by one day. Forward rates for fixed periods will also be affected when pushing the zero coupon yield curve.

Volatility values used for option pricing are also affected by the shift in the valuation date. This is only significant when a volatility landscape with a slope in the option expiry dimension is used. When shifting the valuation date, the time to expiration of the option will be one day shorter and volatility will be fetched. When the underlying market price is used in the calculations, the underlying price is not affected by the one-day forward shift.

4.4.4. Val01

This is the basis point value calculated from the *YTM* formula. It is calculated as

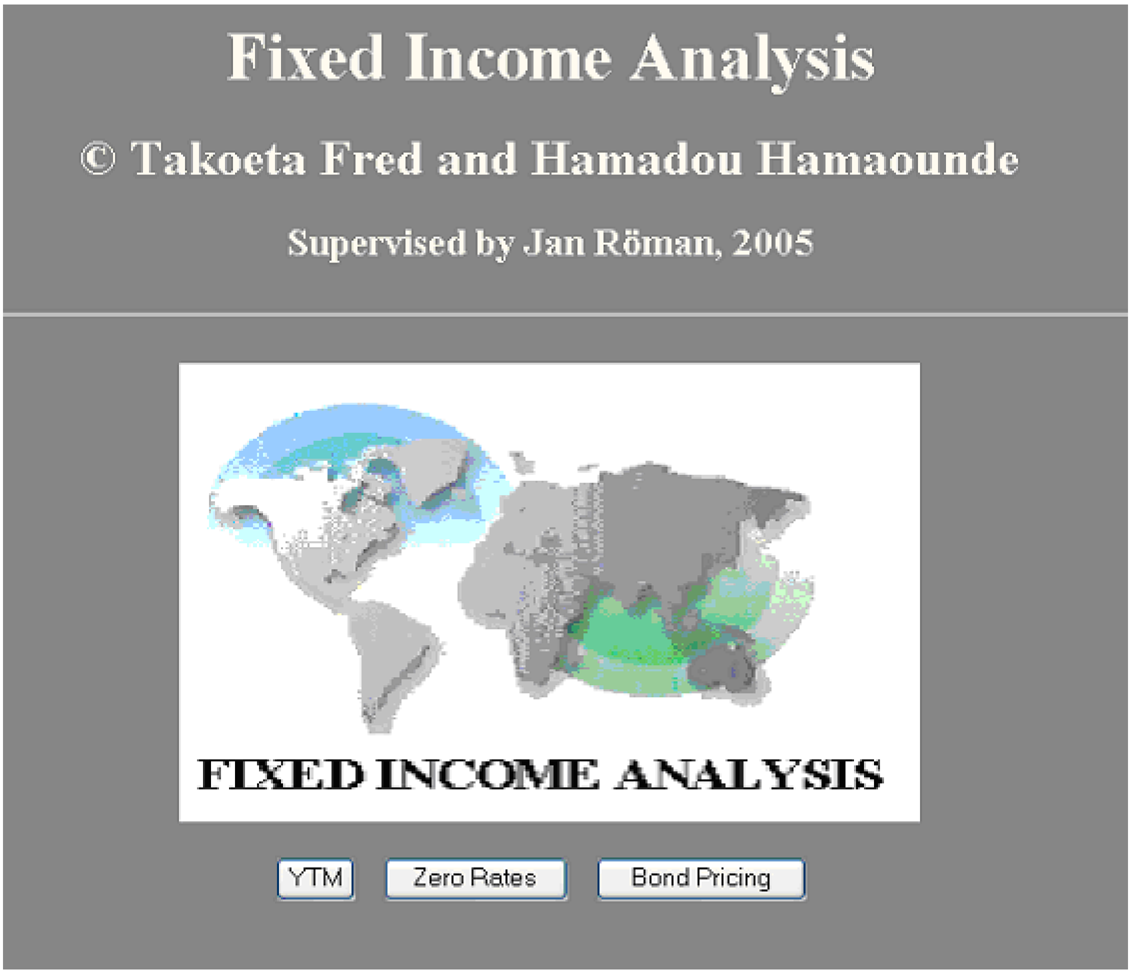
$$Val01 = P(YTM - 0.5bp) - P(YTM + 0.5bp) \tag{4.15}$$

Where *P* represents the dirty price and *bp* a basis point. The shifts are added to the yield compounded according to the rolling period of the bond.

5. THE SYSTEM

In this chapter, we present the components of the system, how the system works and how to use the system.

When the system is lunched from the browser, the system is seen as in Figure 10 shown below.



Fixed Income Analysis

© Takoeta Fred and Hamadou Hamaounde

Supervised by Jan Röman, 2005

Figure 10 Initial view of the system when lunched

The system is made up of three buttons namely the `YTM`, `Zero Rates` and `Bond Pricing` buttons. Each of this buttons when clicked opens a new window which gives analysis of the different sections of fixed income (bond) as the names of the buttons implies.

5.1. The `YTM` Button

The `YTM` button when clicked opens a new window for the analysis of the yield to maturity of bonds. Figure 11 shows the `YTM` button.

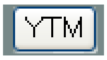
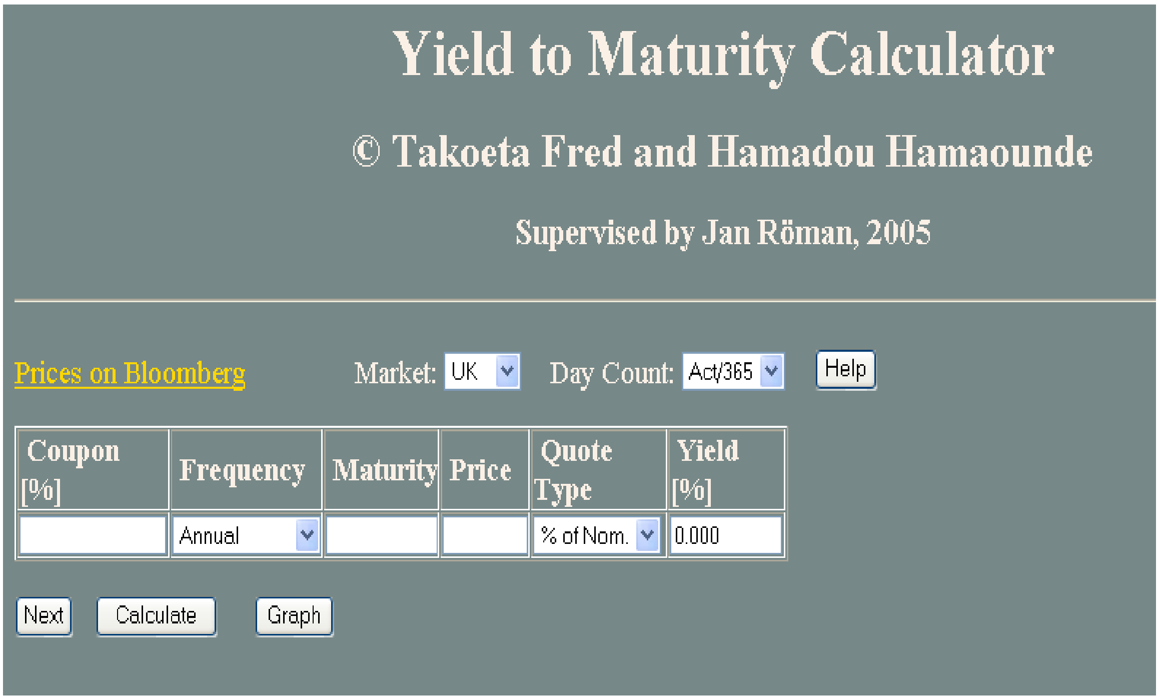


Figure 11 YTM button

A click of the YTM button opens a new window, the yield to maturity calculator as seen in figure 12 below. This calculator is used to calculate the yields to maturity of various treasuries in different markets.



Yield to Maturity Calculator

© Takoeta Fred and Hamadou Hamaounde

Supervised by Jan Röman, 2005

Prices on Bloomberg Market: UK Day Count: Act/365

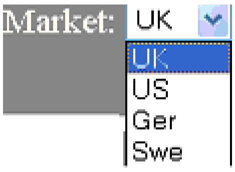
Coupon [%]	Frequency	Maturity	Price	Quote Type	Yield [%]
	Annual			% of Nom.	0.000

Figure 12 YTM Calculator

The YTM calculator has eight fields, three command buttons, four select buttons and a link for prices of different markets from Bloomberg.

The fields are market, day count, coupon, frequency, maturity, price, quote type and yield.

- The Market field is made up of a select button, which gives the user the option to select the market he or she wants to consider. In this study we limit our list of markets to four that is United Kingdom (UK), United States (US), Germany (Ger) and Sweden (Swe). Figure 13 below shows the select button of the market field.



Market: UK

Figure 13 Select button for the markets

- The Day Count field is made up of a select button, which gives the user the option to select the day count convention, he or she wants to consider depending on the day count used in the market in consideration. Figure 14 shows the select button of day count.

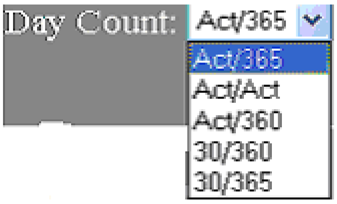

Day Count: Act/365

Figure 14 Select button for Day Count

● The Frequency field is made up of a select button, which gives the user the option to select the frequency, he or she wants to consider depending on the market in consideration. Table 1.0 in chapter three gives a list of the frequency used by the various markets. Figure 15 shows the select button for the frequency.

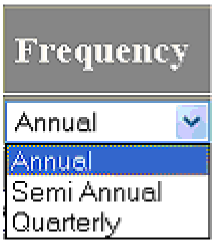

Frequency
Annual

Figure 15 Select button for frequency

● The Quote Type field is made up of a select button which gives the user the option to select the quote type he or she wants to consider. Figure 16 shows the select button for the Quote Type.

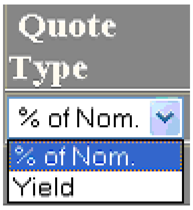
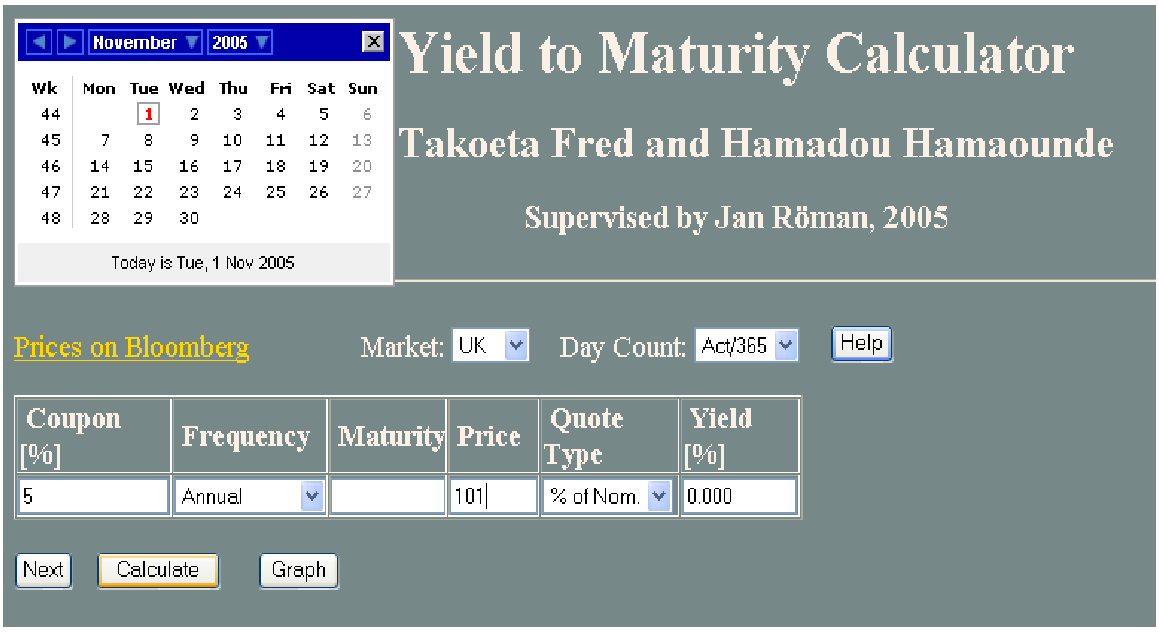

Quote Type
% of Nom.

Figure 16 Select button for Quote Type

The user of the calculator first selects a market and an appropriate day count of the market. After this selection, the user then input the coupon and price, select a frequency and a quote type. On clicking the maturity text field, a calendar pops up, see figure 17 and the user selects the month, day and year of maturity of the bond.



◀ ▶ November 2005 ✕

Wk	Mon	Tue	Wed	Thu	Fri	Sat	Sun
44		1	2	3	4	5	6
45	7	8	9	10	11	12	13
46	14	15	16	17	18	19	20
47	21	22	23	24	25	26	27
48	28	29	30				

Today is Tue, 1 Nov 2005

Yield to Maturity Calculator

Takoeta Fred and Hamadou Hamaounde

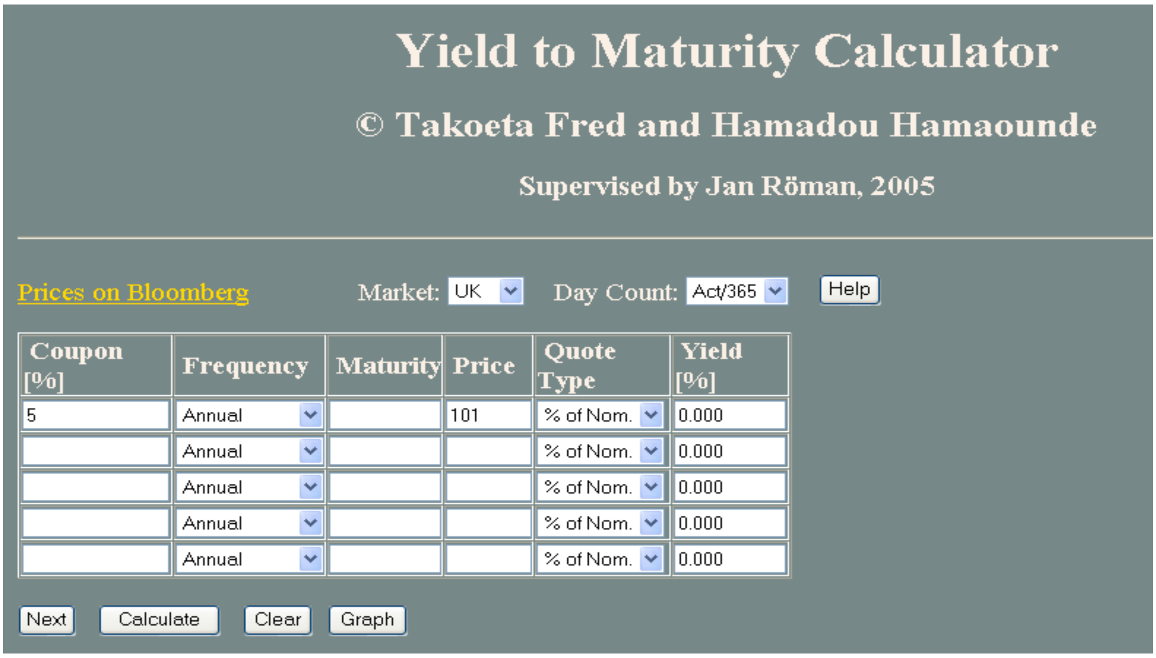
Supervised by Jan Röman, 2005

Prices on Bloomberg Market: UK Day Count: Act/365 Help

Coupon [%]	Frequency	Maturity	Price	Quote Type	Yield [%]
5	Annual		101	% of Nom.	0.000

Figure 17 Calendar pops up when maturity field is clicked

The user clicks on the Next button as many times as he or she wishes depending on the number of treasuries for new line input text fields as shown in figure 18 below.



Yield to Maturity Calculator

© Takoeta Fred and Hamadou Hamaounde

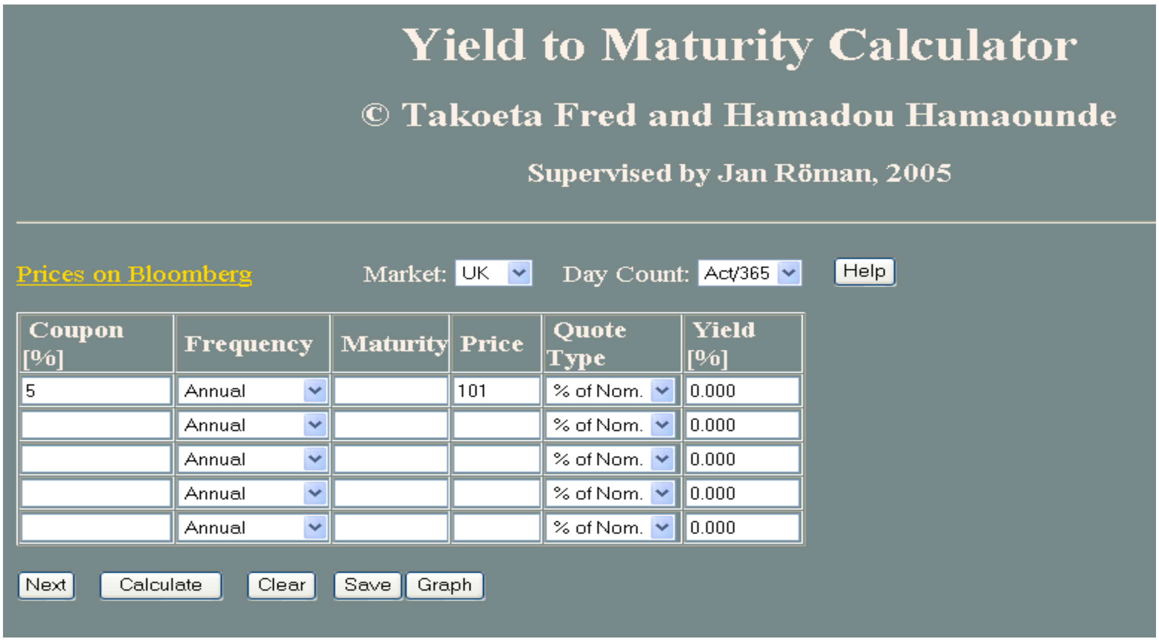
Supervised by Jan Röman, 2005

Prices on Bloomberg Market: UK Day Count: Act/365 Help

Coupon [%]	Frequency	Maturity	Price	Quote Type	Yield [%]
5	Annual		101	% of Nom.	0.000
	Annual			% of Nom.	0.000
	Annual			% of Nom.	0.000
	Annual			% of Nom.	0.000
	Annual			% of Nom.	0.000

Figure 18 more input field when Next button is clicked

When the required number of input lines is acquired, repeats all the steps discussed above and press the Calculate button. The yields to maturity of each bond will appear in the yield text field of each line and a new button, save button appears as seen below.



Yield to Maturity Calculator

© Takoeta Fred and Hamadou Hamaounde

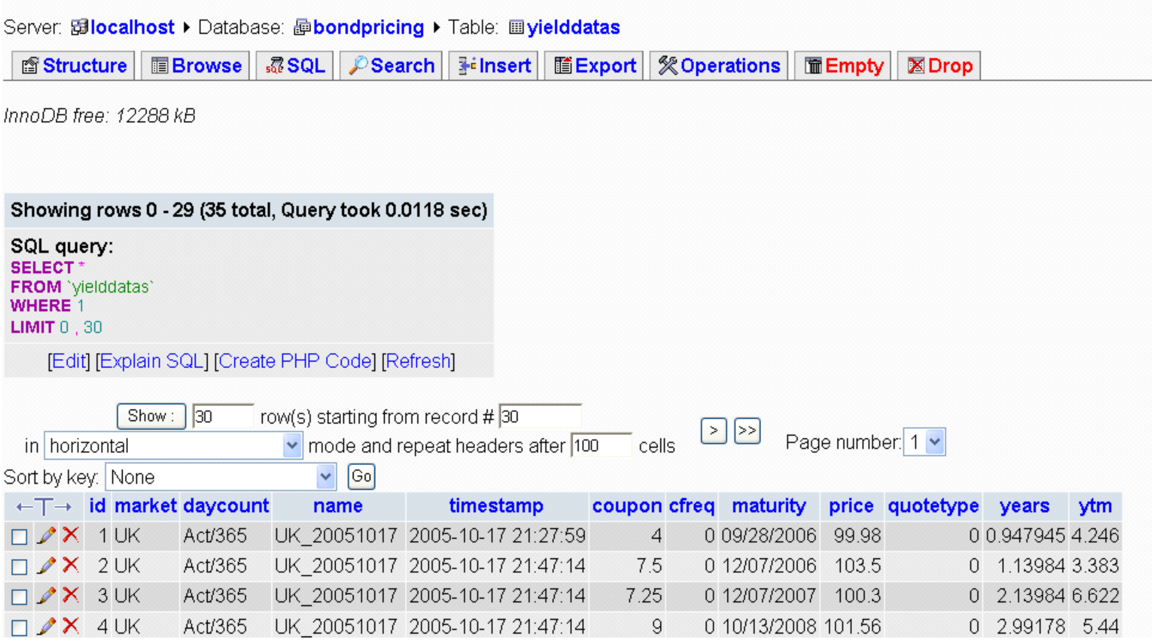
Supervised by Jan Röman, 2005

Prices on Bloomberg Market: UK Day Count: Act/365

Coupon [%]	Frequency	Maturity	Price	Quote Type	Yield [%]
5	Annual		101	% of Nom.	0.000
	Annual			% of Nom.	0.000
	Annual			% of Nom.	0.000
	Annual			% of Nom.	0.000
	Annual			% of Nom.	0.000

Figure 19 save button

Clicking the Save button, data in the table is saved in the database. An example of saved data in the database is shown below.



Server: localhost ▸ Database: bondpricing ▸ Table: yielddatas

InnoDB free: 12288 kB

Showing rows 0 - 29 (35 total, Query took 0.0118 sec)

SQL query:

```
SELECT *
FROM `yielddatas`
WHERE 1
LIMIT 0 , 30
```

[Edit] [Explain SQL] [Create PHP Code] [Refresh]

Show : 30 row(s) starting from record # 30
in horizontal mode and repeat headers after 100 cells
Page number: 1
Sort by key: None

←T→	id	market	daycount	name	timestamp	coupon	cfreq	maturity	price	quotetype	years	ytm
□	1	UK	Act/365	UK_20051017	2005-10-17 21:27:59	4	0	09/28/2006	99.98	0	0.947945	4.246
□	2	UK	Act/365	UK_20051017	2005-10-17 21:47:14	7.5	0	12/07/2006	103.5	0	1.13984	3.383
□	3	UK	Act/365	UK_20051017	2005-10-17 21:47:14	7.25	0	12/07/2007	100.3	0	2.13984	6.622
□	4	UK	Act/365	UK_20051017	2005-10-17 21:47:14	9	0	10/13/2008	101.56	0	2.99178	5.44

Figure 20 Example of save data in the database

Clicking the Graph button, a graph of the calculated yield to maturity is shown on a different window. It should be noted that the save button must be clicked first to save the data in the database because data is fetched from the database to draw the corresponding graph with the help of JpGraph. Figure 21 shows an example of a graph of the yield to maturity of the German market.

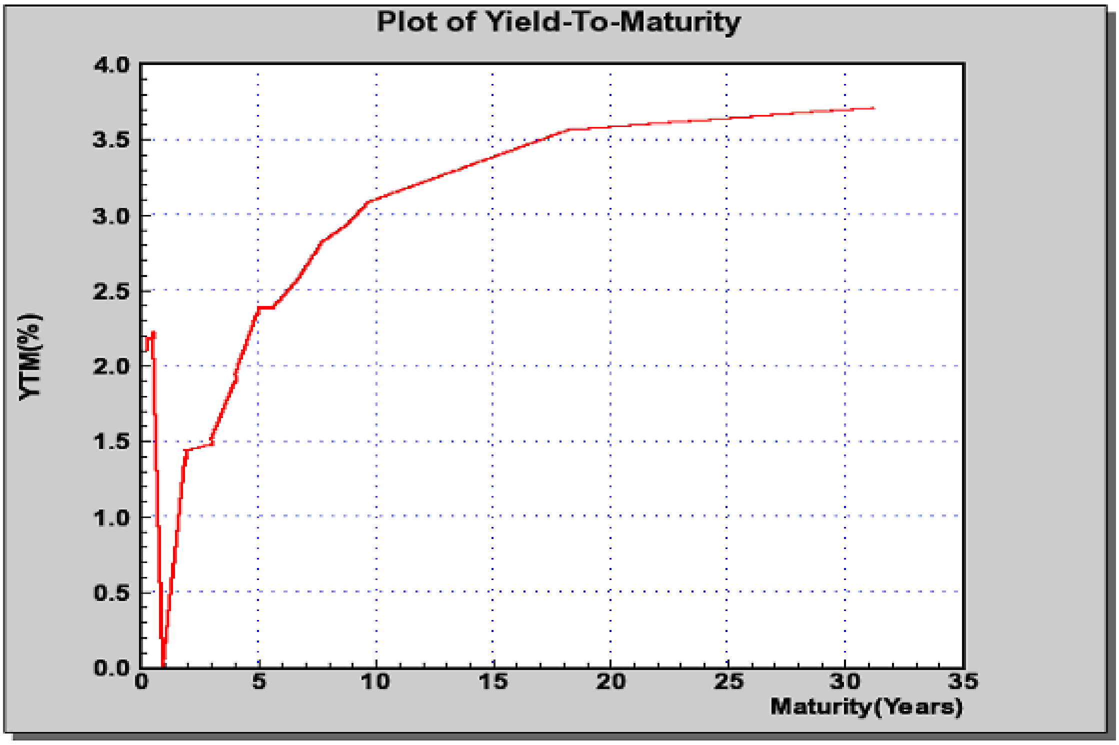


Figure 21 Graph of YTM of German market.

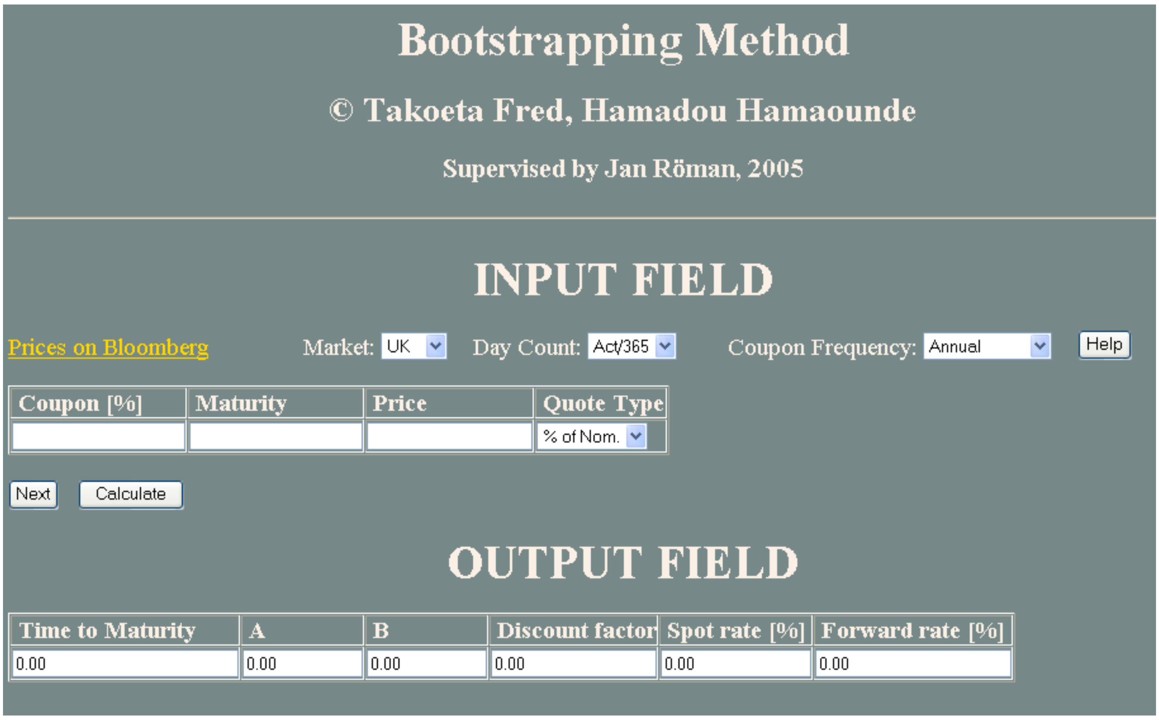
To clear the results, click on the Clear button to return to the default in Figure 12.

5.2. The `Zero Rates` Button

When the `Zero rate` button is clicked, a new window opens for the analysis of the spot rates. Figure 22 shows the `Zero rate` button

Figure 22 Zero Rate button

The user of the calculator first selects a market, appropriate day count of the market and the coupon frequency. After this selection, the user then input the coupon, maturity, price and selects a quote type. On clicking the calculate button, the result is shown in the output field of the calculator that is the years to maturity, the discount rate, the spot and forward rates. The result is also saved automatically into the database. Figure 23 below show the zero rate calculators. The Next button gives the user the opportunity to input as many lines as he or she wishes (as described in section 6.1).



Bootstrapping Method

© Takoeta Fred, Hamadou Hamaounde

Supervised by Jan Röman, 2005

INPUT FIELD

Prices on Bloomberg

Market: UK

Day Count: Act/365

Coupon Frequency: Annual

Coupon [%]	Maturity	Price	Quote Type
			% of Nom.

OUTPUT FIELD

Time to Maturity	A	B	Discount factor	Spot rate [%]	Forward rate [%]
0.00	0.00	0.00	0.00	0.00	0.00

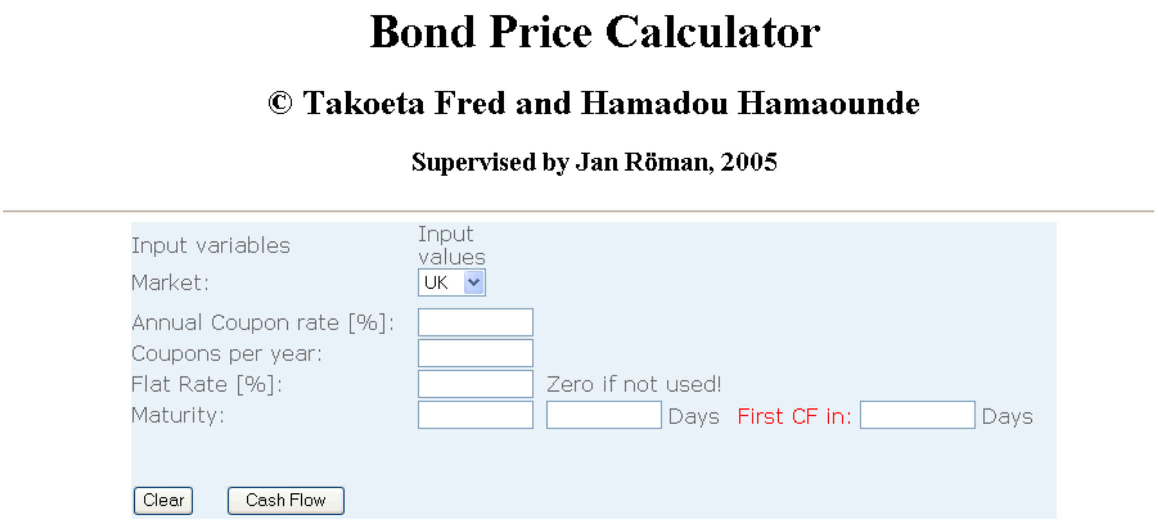
Figure 23 Zero rate calculator.

5.3. The `Bond Pricing` Button

The `Bond Pricing` button when clicked opens a new window for calculating prices of bonds and risk factors of corporate bonds.
Figure 24 shows the `Bond Pricing` button.

Figure 24 Bond Pricing button

Clicking the `Bond Pricing` button opens a new window, the bond pricing calculator as seen in figure 25 below. This calculator is used to calculate the prices and risk measures of bonds in different markets.



Bond Price Calculator

© Takoeta Fred and Hamadou Hamaounde

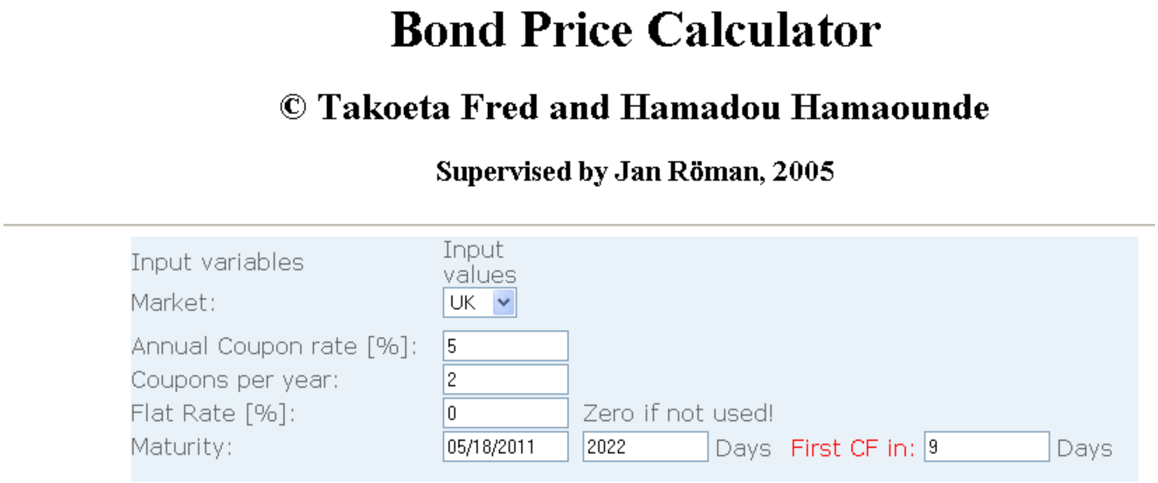
Supervised by Jan Röman, 2005

Input variables	Input values					
Market:	UK					
Annual Coupon rate [%]:						
Coupons per year:						
Flat Rate [%]:		Zero if not used!				
Maturity:			Days	First CF in:		Days

Figure 25 Bond pricing calculator.

In the Bond pricing calculator, the user uses a select button to select the market and has four input fields to input data that is coupon rate, number of coupons per year, fate rate and maturity which has the same properties as that of the maturity field described in the YTM calculator.

When the user inputs all the required information of the bond and clicks the `Cash Flow` button, the cash flow of the bond, the price of the bond and the risk measures are calculated. Figure 26 and figure 27 shows an example of the cash flow of a bond.



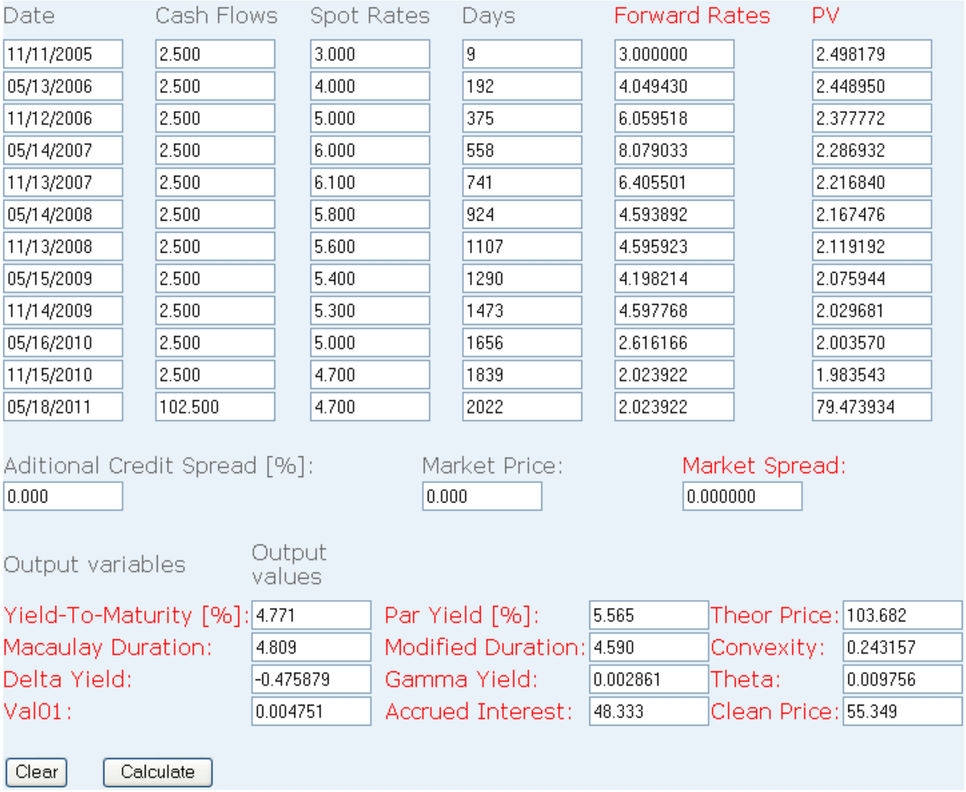
Bond Price Calculator

© Takoeta Fred and Hamadou Hamaounde

Supervised by Jan Röman, 2005

Input variables	Input values					
Market:	UK					
Annual Coupon rate [%]:	5					
Coupons per year:	2					
Flat Rate [%]:	0	Zero if not used!				
Maturity:	05/18/2011	2022	Days	First CF in:	9	Days

Figure 26 shows number of days to maturity and first cash flow.



Date	Cash Flows	Spot Rates	Days	Forward Rates	PV
11/11/2005	2.500	3.000	9	3.000000	2.498179
05/13/2006	2.500	4.000	192	4.049430	2.448950
11/12/2006	2.500	5.000	375	6.059518	2.377772
05/14/2007	2.500	6.000	558	8.079033	2.286932
11/13/2007	2.500	6.100	741	6.405501	2.216840
05/14/2008	2.500	5.800	924	4.593892	2.167476
11/13/2008	2.500	5.600	1107	4.595923	2.119192
05/15/2009	2.500	5.400	1290	4.198214	2.075944
11/14/2009	2.500	5.300	1473	4.597768	2.029681
05/16/2010	2.500	5.000	1656	2.616166	2.003570
11/15/2010	2.500	4.700	1839	2.023922	1.983543
05/18/2011	102.500	4.700	2022	2.023922	79.473934

Aditional Credit Spread [%]:	Market Price:	Market Spread:
0.000	0.000	0.000000

Output variables	Output values				
Yield-To-Maturity [%]:	4.771	Par Yield [%]:	5.565	Theor Price:	103.682
Macaulay Duration:	4.809	Modified Duration:	4.590	Convexity:	0.243157
Delta Yield:	-0.475879	Gamma Yield:	0.002861	Theta:	0.009756
Val01:	0.004751	Accrued Interest:	48.333	Clean Price:	55.349

Figure 27 show cash flow of bond, bond price and risk measures.

For calculation of corporate bonds, the user input the additional spread and clicks the calculate button or the user input the desired market price and press calculate to get the theoretical price of the bond and the risk measures depending on his or her additional spread and desired market price

6. CONCLUSION

We have reviewed the major concepts, mathematical theories and techniques used in the pricing of fixed-income instruments. We have tackled the 'problematic' theory of yield curve extraction and settle our choice on the bootstrapping method using a simple interpolation method: the linear interpolation on discount-factors.

We developed different types of algorithms based on the above enumerated concepts and fulfil our goal of building the Web based mini system for fixed-income pricing.
Concluding this part we shall notice that 'there is no single correct way to complete the term structure of yield curve'[4]. One should only observe the desired criteria of the discount function and particularly avoid an over-smoothing of the curve which may result in a lost of valuable market information. Let's notice that only in bond markets with sufficiently many coupon bonds with regular payment dates and maturities can the bootstrapping approach deliver a decent estimate of the whole zero-coupon yield curve. In other markets, alternative methods are called for: the so-called statistical techniques, based on the assumption that the discount function is of a given functional form with some unknown parameters.

The findings of this study could be used by all, including but not limited to financial engineering students, practitioners and investors, especially investors who are interested in risky bonds. Generally, this thesis intends to make a valid contribution to the advancement of knowledge in the area of fixed income analysis.

[4] Patrick S. Hagan and Graeme West (2005)

7. REFERENCES

1) Julie C. Meloni, PHP, MySQL and Apache, All in One, Sams Teach Yourself (2005).
2) Jan Röman, Lecture Notes in Analytical Finance I&II, Mälardalen University (2005).
3) Haug, E., The Complete Guide to Option Pricing Formulas. McGraw-Hill (1998).
4) Hull J. Option Futures and other Derivatives, Prentice-Hall, New Jersey, fifth edition (2002).
5) Clewlow and Strickland. "Implementing Derivatives models, Wiley (2000).
6) David J. Bolder, Grahame Johnson, and Adam Metzler(2004), "An Empirical Analysis of the Canadian Term Structure of Zero-Coupon Interest Rates", Banque of Canada Working Paper 2004-48
7) Anderson, N., Breedon, F., Deacon, M., Derry, A., & Murphy, G. (1996). "Estimating and interpreting the yield curve." Chichester: John Wiley Series in Financial Economics and Quantitative Analysis.
8) Patrick s. Hagan and Graeme West: Interpolation methods for curve construction, July 2005
9) Shoichiro Nakamura (1993), "Applied Numerical Methods in C", PTR PRENTICE HALL.
10) http://mathworld.wolfram.com/cubicspline.html

8. APPENDIX

```
/*
Name: project.php
Authors: Takoeta Fred & Hamadou Hamaounde
Supervisor: Jan Röman
Description: This program creates the main window of the project
*/

<html>
<head>
 <META NAME="GENERATOR" CONTENT="Emacs 20.6.1">
<Title>Fixed Income Analysis</Title>
<script language="javascript">
</script>
</head>
<BODY BGCOLOR="#778888" TEXT="#faf0e6" LINK="#ffd700" VLINK="#00ffff">
<?php

switch ($action) {
  case "YTM":

  $wnd = mt_rand(0, 100000000);
  echo "<script LANGUAGE=\"JavaScript\">";echo
    "window.open('http://localhost/tambong.php', $wnd,
    'toolbar=yes,location=no,directories=no,status=no,menubar=no,
    scrollbars=yes,resizable=yes,height=600,width=600')";
  echo "</script>";
  break;

case "Zero Rates":
  $wnd = mt_rand(0, 100000000);
  echo "<script LANGUAGE=\"JavaScript\">";echo
    "window.open('http://localhost/boot2.php',
    $wnd,'toolbar=yes,location=no,directories=no,status=no,menubar=no,
    scrollbars=yes,resizable=yes,height=600,width=600')";
  echo "</script>";
  break;

case "Bond Pricing":
  $wnd = mt_rand(0, 100000000);
  echo "<script LANGUAGE=\"JavaScript\">";echo
    "window.open('http://localhost/bond.php',
    $wnd,'toolbar=yes,location=no,directories=no,status=no,menubar=no,
    scrollbars=yes,resizable=yes,height=600,width=600')";
  echo "</script>";
  break;
}

$action="";
?>
<FORM name = "rateform" ACTION = "<? $SCRIPT_NAME ?>" METHOD = "POST">
<p><b><h1 align = "center">Fixed Income Analysis</h1></b></p>
<p><b><h2 align = "center">&copy Takoeta Fred and Hamadou Hamaounde
</h2></b></p>
<p><b><h3 align = "center">Supervised by Jan Röman, 2005</h></b></p>
<hr align="left">
<p>
<img src="cash.gif"
align = "center" width="400" height="250">
</p>
```

```
<INPUT TYPE=submit NAME=action  VALUE="YTM">&nbsp;&nbsp
<INPUT TYPE=submit NAME=action  VALUE="Zero Rates">&nbsp;&nbsp
<INPUT TYPE = submit  NAME = action  VALUE ="Bond Pricing">&nbsp;&nbsp
</FORM>
</html>

/*
Name: tambong.php
Authors: Takoeta Fred & Hamadou Hamaounde
Supervisor: Jan Röman
Description: Window for YTM calculator
*/
<html>
<head>
 <Title>Yield to Maturity Calculator</Title>
  function count(date1)
  {
    var startdate = new Date();
    alert(startdate);
    alert(date1);
    var enddate   = new Date(date1);
    alert(enddate);
    var diff      = Math.round((enddate.getTime() -
startdate.getTime())/86400000);
    alert("It is "+diff+" days between!");
return diff;
  }
</script>
</head>


<?php
include ("./include/Calendar.html");
include ("YTM2.php");

DEFINE (UK,  "0");
DEFINE (US,  "1");
DEFINE (GER, "2");
DEFINE (SWE, "3");

$B = array();

$action = strtolower(trim($action));
$str    = "";
$C      = array();
$M      = array();
$Y      = array();
$P      = array();
$S      = array();
$T      = array();
$wx     = 690;         // X-Window size
$wy     = 540;         // Y-Window size
$db     = "BondPricing";

if ($item == "") {
  $item    = 0;
  $NumBond = 0;
}

switch ($action) {
  case "clear":
    $save  = 0;
    $item  = 0;
    $MType = 0;

     for ($i = 0; $i < 16; $i++) {
      $C[$i]=0.0;
      $M[$i]="";
      $P[$i]="";
      $Y[$i]="";
      $CType[$i] = 0;
      $QType[$i] = 0;
    }
    break;
  case "calculate":
  case "next":
  case "save":
  case "graph":
    $save = 1;
    $C[0]=  $C0;  $M[0]=  $M0;  $P[0]=  $P0;  $Y[0]=  $Y0;  $S[0]=  $S0;
    $C[1]=  $C1;  $M[1]=  $M1;  $P[1]=  $P1;  $Y[1]=  $Y1;  $S[1]=  $S1;
    $C[2]=  $C2;  $M[2]=  $M2;  $P[2]=  $P2;  $Y[2]=  $Y2;  $S[2]=  $S2;
    $C[3]=  $C3;  $M[3]=  $M3;  $P[3]=  $P3;  $Y[3]=  $Y3;  $S[3]=  $S3;
    $C[4]=  $C4;  $M[4]=  $M4;  $P[4]=  $P4;  $Y[4]=  $Y4;  $S[4]=  $S4;
    $C[5]=  $C5;  $M[5]=  $M5;  $P[5]=  $P5;  $Y[5]=  $Y5;  $S[5]=  $S5;
    $C[6]=  $C6;  $M[6]=  $M6;  $P[6]=  $P6;  $Y[6]=  $Y6;  $S[6]=  $S6;
    $C[7]=  $C7;  $M[7]=  $M7;  $P[7]=  $P7;  $Y[7]=  $Y7;  $S[7]=  $S7;
    $C[8]=  $C8;  $M[8]=  $M8;  $P[8]=  $P8;  $Y[8]=  $Y8;  $S[8]=  $S8;
    $C[9]=  $C9;  $M[9]=  $M9;  $P[9]=  $P9;  $Y[9]=  $Y9;  $S[9]=  $S9;
    $C[10]= $C10; $M[10]= $M10; $P[10]= $P10; $Y[10]= $Y10; $S[10]= $S10;
    $C[11]= $C11; $M[11]= $M11; $P[11]= $P11; $Y[11]= $Y11; $S[11]= $S11;
    $C[12]= $C12; $M[12]= $M12; $P[12]= $P12; $Y[12]= $Y12; $S[12]= $S12;
    $C[13]= $C13; $M[13]= $M13; $P[13]= $P13; $Y[13]= $Y13; $S[13]= $S13;
    $C[14]= $C14; $M[14]= $M14; $P[14]= $P14; $Y[14]= $Y14; $S[14]= $S14;
    $C[15]= $C15; $M[15]= $M15; $P[15]= $P15; $Y[15]= $Y15; $S[15]= $S15;

    if ($action == "next") {
      $item++;
    }

   if ($action == "save") {
      $link = mysql_connect("localhost");
      if (! $link)
        die("Couldn't connect to MySQL");

      mysql_select_db($db , $link)
        or die("Couldn't open $db: ".mysql_error());

      for ($i = 0; $i <= $item; $i++) {
        $dat = date("Ymd");
        $coupon   = $C[$i];
        $freq     = $CType[$i];
        $maturity = $M[$i];
        $price    = $P[$i];
        $quote    = $QType[$i];
        $ytm      = $Y[$i];
        $days     = (strtotime($M[$i]) - strtotime(date("Y-m-d"))) / (60 *
60 * 24);
        $T[$i]    =$days/365;
        $years    = $T[$i];

        switch ($MType[$i]) {
          case '0':
            $market   = "'UK'";
            $name     = "'UK_".$dat."'";
            break;

          case '1':
            $market   = "'US'";
            $name     = "'US_".$dat."'";
            break;

          case '2':
            $market   = "'Ger'";
            $name     = "'Ger_".$dat."'";
            break;

          case '3':
            $market   = "'Swe'";
            $name     = "'Swe_".$dat."'";
            break;
        }

        switch ($DType[$i]) {
          case "0":
            $daycount   = "'Act/365'";
            break;

          case "1":
            $daycount   = "'Act/Act'";
            break;

          case "2":
            $daycount   = "'Act/360'";
            break;

          case "3":
            $daycount   = "'30/360'";
            break;

          case "4":
            $daycount   = "'30/365'";
            break;
        }

        mysql_query ("INSERT INTO YieldDatas  (market,
                                               daycount,
                                               name,
                                               coupon,
                                               cfreq,
                                               maturity,
                                               price,
                                               quotetype,
                                               years,
                                               ytm )
               VALUES ($market, $daycount ,$name, $coupon, $freq,
'$maturity', $price, $quote, $years, $ytm)");
      }
   }

   if ($action == "graph") {
    $wnd = mt_rand(0, 100000000);
    echo "    echo "window.open('yieldplot.php', $wnd, 'width=$wx,
height=$wy, scrollbars=yes,resize=yes')";
    echo "</script>";
   }

   if ($action == "calculate") {
     $save  =2 ;
     for ($i = 0; $i <=$item; $i++){

       // day counting from present day to maturity of bond
       $days =(strtotime($M[$i]) - strtotime(date("Y-m-d"))) / (60 * 60 *
24);
       $T[$i]=$days/365;

       //loop for changing frequency

       if($CType[$i] == 0 )
         $F = 1;
       else
             if($CType[$i] == 1)
               $F = 2;
             else
               if($CType[$i] == 2 )
                 $F = 4;

       // Calculating yield to maturity
       $Y[$i]=100*roots(0.0, 0.25, 100, $C[$i], $F, $T[$i], $P[$i]);

     }
   }
   break;

  case "help":
    $wnd = mt_rand(0, 100000000);
echo "    echo "</script>";
  break;
}
$action="";
?>
<FORM name = "rateform" ACTION = "<? $SCRIPT_NAME ?>" METHOD = "POST">
<p><b><h1 align = "center">Yield to Maturity Calculator</h1></b></p>
<p><b><h2 align="center">&copy Takoeta Fred and Hamadou
Hamaounde</h2></b></p>
<p><b><h3 align = "center">Supervised by Jan Röman, 2005</h3></b></p>

<hr align="left">
<p>
<AHREF="http://www.bloomberg.com/markets/rates/index.html"
TARGET=_blank>Prices on Bloomberg</A>
&nbsp;&nbsp&nbsp;&nbsp&nbsp;&nbsp&nbsp;&nbsp&nbsp;&nbsp&nbsp;&nbsp&nbsp;
Market:
      <SELECT NAME="MType">
<?
    if ($MType == 0)
       echo "<OPTION Value=0 SELECTED>UK";
     else
       echo "<OPTION Value=0 >UK";
     if ($MType == 1)
        echo "<OPTION value=1 SELECTED>US";
     else
        echo "<OPTION value=1 >US";
     if ($MType == 2)
        echo "<OPTION Value=2 SELECTED>Ger";
     else
        echo "<OPTION Value=2 >Ger";
     if ($MType == 3)
        echo "<OPTION Value=3 SELECTED>Swe";
      else
        echo "<OPTION Value=3 >Swe";
?>
</SELECT>&nbsp;&nbsp;&nbsp;
Day Count:
      <SELECT NAME="DType">
<?
    if ($DType == 0)
       echo "<OPTION Value=0 SELECTED>Act/365";
     else
       echo "<OPTION Value=0 >Act/365";
     if ($DType == 1)
        echo "<OPTION value=1 SELECTED>Act/Act";
     else
        echo "<OPTION value=1 >Act/Act";
     if ($DType == 2)
        echo "<OPTION Value=2 SELECTED>Act/360";
     else
        echo "<OPTION Value=2 >Act/360";
     if ($DType == 3)
        echo "<OPTION Value=3 SELECTED>30/360";
      else
        echo "<OPTION Value=3 >30/360";
     if ($DType == 4)
       echo "<OPTION Value=3 SELECTED>30/365";
else
        echo "<OPTION Value=3 >30/365";
?>
</SELECT>&nbsp;&nbsp;&nbsp;
<INPUT TYPE = "submit" NAME = "action" VALUE = "Help"></p>
<p><TABLE>
<TABLE WIDTH="530" BORDER="1" CELLSPACING="1" CELLPADDING="0">
   <tr>
      <td>&nbsp;<b>Coupon [%]</b></td>
      <td>&nbsp;<b>Frequency</b></td>
      <td>&nbsp;<b>Maturity</b></td>
      <td>&nbsp;<b>Price</b></td>
      <td>&nbsp;<b>Quote Type</b></td>
      <td>&nbsp;<b>Yield [%]</b></td>
   </tr>
   <tr>
      <td><INPUT TYPE="text" NAME="C0" SIZE="13" VALUE="<? echo $C[0]
?>"></td>
      <td><SELECT NAME="CType[0]">
      <?
        if ($CType[0] == 0)
          echo "<OPTION Value=0 SELECTED>Annual";
        else
          echo "<OPTION Value=0 >Annual";
        if ($CType[0] == 1)
          echo "<OPTION value=1 SELECTED>Semi Annual";
        else
          echo "<OPTION value=1 >Semi Annual";
        if ($CType[0] == 2)
          echo "<OPTION Value=2 SELECTED>Quarterly";
        else
          echo "<OPTION Value=2 >Quarterly";
      ?>
      </select></td>
      <td><INPUT TYPE="text" NAME="M0"
           onfiltered="popUpCalendar(this, rateform.M0, 'mm/dd/yyyy');"
           SIZE="9" VALUE="<? echo $M[0] ?>"></td>
      <td><INPUT TYPE="text" NAME="P0" SIZE="6" VALUE="<?echo $P[0]
?>"></td>
      <td><SELECT NAME="QType[0]">
      <?
        if ($QType[0] == 0)
          echo "<OPTION Value=0 SELECTED>% of Nom.";
        else
          echo "<OPTION Value=0 >% of Nom.";
        if ($QType[0] == 1)
          echo "<OPTION value=1 SELECTED>Yield";
        else
          echo "<OPTION value=1 >Yield";
      ?>
      </select></td>
      <td><INPUT TYPE="text" NAME="Y0" SIZE="9" VALUE="<? printf("%.3f",
$Y[0]); ?>"></td>

  </tr>
       <INPUT TYPE="hidden" NAME="item"  VALUE="<? echo $item ?>">
<?
   for ($i = 1; $i <= $item; $i++) {
     echo("<td><INPUT TYPE='text' NAME='C".$i."' SIZE='13' VALUE=".$C[$i]."
></td>\n");
     echo("<td><SELECT NAME='CType[".$i."]'>\n");
      if ($CType[$i] == 0)
        echo "<OPTION Value=0 SELECTED>Annual\n";
      else
        echo "<OPTION Value=0 >Annual\n";
      if ($CType[$i] == 1)
        echo "<OPTION value=1 SELECTED>Semi Annual\n";
      else
        echo "<OPTION value=1 >Semi Annual\n";
      if ($CType[$i] == 2)
        echo "<OPTION Value=2 SELECTED>Quarterly\n";
      else
        echo "<OPTION Value=2 >Quarterly\n";
     echo("</select></td>");
     echo("<td><INPUT TYPE='text' NAME='M".$i."'");

echo("onfiltered=\"popUpCalendar(this,rateform.M".$i.",'mm/dd/yyyy');\"");
     echo("SIZE='9' VALUE=".$M[$i]."></td>\n");
     echo("<td><INPUT TYPE='text' NAME='P".$i."' SIZE='6'
VALUE=".$P[$i]."></td>\n");
     echo("<td><SELECT NAME='QType[".$i."]'>\n");
      if ($QType[$i] == 0)
        echo "<OPTION Value=0 SELECTED>% of Nom.\n";
      else
        echo "<OPTION Value=0>% of Nom.\n";
      if ($QType[$i] == 1)
        echo "<OPTION value=1 SELECTED>Yield\n";
      else
        echo "<OPTION value=1>Yield\n";
     echo("</select></td>");
         echo("<td><INPUT TYPE='text' NAME='Y".$i."' SIZE='9' VALUE=");
     printf("%.3f", $Y[$i]);
      echo("></td></tr>");
     echo("<INPUT TYPE='hidden' NAME= 'S".($i-1)."' VALUE=".$S[$i-
1].">\n");
   }
?>
</TABLE></p>
<INPUT TYPE="submit" NAME="action"  VALUE="Next">&nbsp;&nbsp
 <INPUT TYPE = submit  NAME = action  VALUE = Calculate>&nbsp;&nbsp;
<?
if ($save == 1||$save == 2){
   echo ("<INPUT TYPE = submit  NAME = action  VALUE=Clear>");
}
echo("&nbsp;&nbsp;");
if ($save == 2){
   echo ("<INPUT TYPE = submit  NAME = action  VALUE=Save>");
}
echo ("<INPUT TYPE = submit  NAME = action  VALUE=Graph>");
?>

</FORM>
</html>

/* Name: ytm2.php
 Authors: Takoeta Fred & Hamadou Hamaounde
 Supervisor: Jan Röman
 Description: several functions including Bisection and Newton-Raphson
methods for calculation of YTM.
*/

<html>
<head>

<?php

/*          variables definition
*F- Bond face value
*C-Coupon rate
*M-coupon frequency for instance M=1 if annual and M=2 for semiannual
*R-interest rate
*P-Bond price
*T-Time to maturity(in years)
*N.B in the while loop -$i should never be multiplied by $m since its range
will go from 1 to $t*$m-1)
*we do $t*$m-1) because since at the maturity is unknown we will receive
the last coupon + Bond face value
*/

/* --------------------------------------------------------------------- */
// Returns $f  = function value. Ex: PV - market price (shuld be zero)
//         $df = function derivative value = {f(x + h) - f(x - h)}/2h

function f_vald($r, $f, $df, $p, $c, $m, $t, $price)
{
  $DH = 1.0e-5;

  $h = $DH*MAX(1.0, abs($r));
  $r2  = $r - $h;
  $r3  = $r + $h;
  $x1  = pv($p, $c, $m, $t, $r)  - $price;
  $x2  = pv($p, $c, $m, $t, $r2) - $price;
  $x3  = pv($p, $c, $m, $t, $r3) - $price;
  $f   = $x1;
  $df  = ($x3 - $x2)/(2*$h);
}

/*---------------------------------------------------------------------- */

function roots($xmin, $xmax, $f, $c, $m, $t, $p)
{
  $xacc  = 1.0e-8; // The root accuracy
  $maxit = 30;     // Maximum number of iterations
  $fp    = pv($f, $c, $m, $t, $xmin) - $p;
  $fc    = pv($f, $c, $m, $t, $xmax) - $p;
  $str   = "";

  if ($fc*$fp < 0.0 || $fc == 0.0) { /* If sign change occurs then record
them. */
    $root = rtsafe($xmin, $xmax, $xacc, $maxit, $f, $c, $m, $t, $p);
    $str = $str . "Root: " . 100*$root . " %<br>\n";
      $str = "PV = " . pv($f, $c, $m, $t, $root);
    $rt++;
  }
  $fp = $fc;
  return $root;
}

// ------------------------------------------------------------------------

function rtsafe($x1, $x2, $xacc, $maxit, $f, $c, $m, $t, $p)
{
  $ok = 0;
  f_vald($x1, &$fl, &$df, $f, $c, $m, $t, $p);
  f_vald($x2, &$fh, &$df, $f, $c, $m, $t, $p);

  if ($fl*$fh >= 0.0) return 0.0;
  if ($fl < 0.0) {  /* orient the search so that f(x) <
0.................. */
    $xl = $x1;
    $xh = $x2;
  }
  else {
    $xh   = $x1;
    $xl   = $x2;
    $swap = $fl;
    $fl   = $fh;
    $fh   = $swap;
  }
  $rts   = 0.5*($x1 + $x2);    /* initial guess of the root............ */
  $dxold = abs($x2 - $x1);     /* the "step-size before last.".......... */
  $dx    = $dxold;             /* and the last step.................... */
  f_vald($rts, &$ff, &$df, $f, $c, $m, $t, $p);

  for ($j = 0; $j < $maxit; $j++) {
    if (((($rts - $xh)*$df - $ff)*(($rts - $xl)*$df - $ff) >=0.0) || /*
Bisect if N-R.*/
        (abs(2.0*$ff) > abs($dxold*$df))){/* out of range or not decreasing
*/
      $dxold = $dx;                 /* fast enough.................. */
      $dx    = 0.5*($xh - $xl);
      $rts   = $xl + $dx;
      if ($xl == $rts) {
        $ok = 1;
        return $rts;                   /* change in root is negligible.. */
      }
    }
    else {                           /* Newton step accepable. Take it */
      $dxold = $dx;
      $dx    = $ff/$df;
      $temp  = $rts;
      $rts  -= $dx;
      if (temp == $rts) {
        $ok = 1;
        return $rts;
      }
    }

    if (abs($dx) < $xacc) {
      $ok = 1;
      return $rts;                    /* Convergence criterion.......... */
    }
    f_vald($rts, &$ff, &$df, $f, $c, $m, $t, $p); /* The one new function
evaluation per iteration. */

    if ($ff < 0.0) {        /* Maintain the bracket on the
root.............. */
      $xl = $rts;
      $fl = $ff;
    }
    else {
      $xh = $rts;
      $fh = $ff;
     }
  }
  return 0.0;
}

// ------------------------------------------------------------------------

function pv($f, $c, $m, $t, $r)
{
  //present value of cashflows

  $i=1;
  $result=0;

  while($i<$t*$m-1){
    $result+=($c/$m)*pow((1+($r/$m)),-$i);
    $i++;
  }

  $pv=$result+(($c/$m)+$f)*pow((1+($r/$m)),-$m*$t);
  return $pv;

}
?>
</p>
<?php

?>

</html>
/*
```

```
Name: yieldplot.php
Authors: Takoeta Fred & Hamadou Hamaounde
 Supervisor: Jan Röman
 Description: the functions used to plot the YTM with the aid of JPGraph
*/
<?php
include ("./include/jpgraph.php");
include ("./include/jpgraph_log.php");
include ("./include/jpgraph_line.php");
include ("./options/global_data.php");

$db     = "BondPricing";
$plotx  = array();
$ploty  = array();

$fp = fopen('C:/tmp/test.txt', 'w') or die("can't open file");
$link = mysql_connect("localhost");
if (! $link)
  die("Couldn't connect to MySQL");

mysql_select_db($db , $link)
  or die("Couldn't open $db: ".mysql_error());

switch ($MType[$i]) {
  case '0':
    $market   = "'UK'";
    $name     = "'UK_".$dat."'";
    break;

  case '1':
    $market   = "'US'";
    $name     = "'US_".$dat."'";
    break;

  case '2':
    $market   = "'Ger'";
    $name     = "'Ger_".$dat."'";
    break;

  case '3':
    $market   = "'Swe'";
    $name     = "'Swe_".$dat."'";
    break;
}

$result = mysql_query ("select  market, years, ytm from yielddatas where
market= 'Ger' order by years");
$i = 0;
while ($line = mysql_fetch_array($result)) {
  $ploty[$i] = $line['ytm'];
  $plotx[$i] = $line['years'];
  fwrite($fp, $plotx[$i].'   '.$ploty[$i]."\n");
  $i++;
}
fclose($fp);
$color  = "red";
$title  = "Plot of Yield-To-Maturity";
$xlabel = "Maturity(Years)";    // X-Axis label title
$ylabel = "YTM(%)"; // Y-Axis label title
$graph = new Graph($winx, $winy);
$graph->SetScale(linlin);
```

```
$graph->SetTickDensity(3, 3);
$graph->img->SetMargin($xm1+20, $xm2, $ym1, $ym2);
$graph->title->SetFont($tfn, $titfns, $titfnz);
$graph->SetBox        (true, $axcol,  $axwth);

$graph->xaxis->SetColor      ($axcol);
$graph->xaxis->SetWeight     ($axwth);
$graph->xaxis->SetTitle      ($xlabel, $xftype);
$graph->xaxis->SetFont       ($tfn,    $titfns, $xfnz);
$graph->xaxis->title->SetFont($tfn,    $titfns, $xfnz);
$graph->xaxis->SetPos("min");

$graph->yaxis->SetColor      ($axcol);
$graph->yaxis->SetWeight     ($axwth);
$graph->yaxis->SetFont       ($tfn, $titfns, $xfnz);
$graph->yaxis->SetTitle      ($ylabel, $yftype);
$graph->yaxis->SetTitleMargin(60);
$graph->yaxis->title->SetFont($tfn, $titfns, $xfnz);

$graph->xgrid->SetColor    ($grcol);
$graph->ygrid->SetColor    ($grcol);
$graph->xgrid->SetWeight   ($grwth);
$graph->ygrid->SetWeight   ($grwth);
$graph->xgrid->SetLineStyle($grtype);
$graph->ygrid->SetLineStyle($grtype);

$graph->SetShadow();
$graph->title->Set($title);
$graph->ygrid->Show(true, false);
$graph->xgrid->Show(true, false);

//$lineplot=new LinePlot($ploty);
$lineplot=new LinePlot($ploty, $plotx);
$lineplot->SetColor($color);
$lineplot->SetStyle($type);
$lineplot->SetWeight($weight);
$graph->Add($lineplot);       // Add the plot to the graph

// Display the graph
$graph->Stroke();
?>

/*
Name: newdiscount.php
Authors: Takoeta Fred & Hamadou Hamaounde
 Supervisor: Jan Röman
 Description: Main window  and function for the calculations of spot rates,
forward rate and discont functions.
*/
<html>
<head>
 <META NAME="GENERATOR" CONTENT="Emacs 20.6.1">
<Title>Spot Rates</Title>
</head>
<BODY BGCOLOR="#778888" TEXT="#faf0e6" LINK="#ffd700" VLINK="#00ffff">

<?php
//include ("a2.php");
include ("./include/Calendar.html");
//include ("./include/YTM2.php");
//include ("spotrate.php");
```

```
DEFINE (UK,  0);
DEFINE (US,  1);
DEFINE (GER, 2);
DEFINE (SWE, 3);

// global $NumBond;
// global $B = array();
$B = array();

$action = strtolower(trim($action));
$str    = "";
$C      = array();
$M      = array();
$P      = array();
$S      = array();
$Fw      = array();
$T      = array();
$Ncoupons = array();
$AT      = array();
$AI      = array();
$DP    = array();
$A    = array();
$B    = array();
$d    = array();
 $Ctime   = array();

$wx     = 760;      // X-window
$wy     = 600;      // Y-window
if ($item == "") {
  $item    = 0;
  $NumBond = 0;
}

switch ($action) {
  case "clear":
    $save  = 0;
    $item  = 0;
    $k= 0;
    $MType = 0;


     for ($i = 0; $i <= 50; $i++) {
      $C[$i]="";
      $M[$i]="";
      $P[$i]="";
      $Y[$i]="";
      $CType[$i] = 0;
      $QType[$i] = 0;
      $T[$i]="";
      $S[$i]="";
      $Fw[$i]="";
      $d[$i]="";
     $d[$i+1]="";
      $A[$i]="";
     $B[$i]="";
    }
    break;
  case "calculate":
  case "next":
case "graph":
    $save = 1;

    $C[0]=  $C0;  $M[0]=  $M0;  $P[0]=  $P0;  $Y[0]=  $Y0;  $S[0]=  $S0;
    $C[1]=  $C1;  $M[1]=  $M1;  $P[1]=  $P1;  $Y[1]=  $Y1;  $S[1]=  $S1;
    $C[2]=  $C2;  $M[2]=  $M2;  $P[2]=  $P2;  $Y[2]=  $Y2;  $S[2]=  $S2;
    $C[3]=  $C3;  $M[3]=  $M3;  $P[3]=  $P3;  $Y[3]=  $Y3;  $S[3]=  $S3;
    $C[4]=  $C4;  $M[4]=  $M4;  $P[4]=  $P4;  $Y[4]=  $Y4;  $S[4]=  $S4;
    $C[5]=  $C5;  $M[5]=  $M5;  $P[5]=  $P5;  $Y[5]=  $Y5;  $S[5]=  $S5;
    $C[6]=  $C6;  $M[6]=  $M6;  $P[6]=  $P6;  $Y[6]=  $Y6;  $S[6]=  $S6;
    $C[7]=  $C7;  $M[7]=  $M7;  $P[7]=  $P7;  $Y[7]=  $Y7;  $S[7]=  $S7;
    $C[8]=  $C8;  $M[8]=  $M8;  $P[8]=  $P8;  $Y[8]=  $Y8;  $S[8]=  $S8;
    $C[9]=  $C9;  $M[9]=  $M9;  $P[9]=  $P9;  $Y[9]=  $Y9;  $S[9]=  $S9;
    $C[10]= $C10; $M[10]= $M10; $P[10]= $P10; $Y[10]= $Y10; $S[10]= $S10;
    $C[11]= $C11; $M[11]= $M11; $P[11]= $P11; $Y[11]= $Y11; $S[11]= $S11;
    $C[12]= $C12; $M[12]= $M12; $P[12]= $P12; $Y[12]= $Y12; $S[12]= $S12;
    $C[13]= $C13; $M[13]= $M13; $P[13]= $P13; $Y[13]= $Y13; $S[13]= $S13;
    $C[14]= $C14; $M[14]= $M14; $P[14]= $P14; $Y[14]= $Y14; $S[14]= $S14;
    $C[15]= $C15; $M[15]= $M15; $P[15]= $P15; $Y[15]= $Y15; $S[15]= $S15;

 $T[0]=  $T0;  $A[0]=  $A0;  $B[0]=  $B0;  $d[0]=  $d0;  $Fw[0]=  $Fw0;
    $T[1]=  $T1;  $A[1]=  $A1;  $B[1]=  $B1;  $d[1]=  $d1;  $Fw[1]=  $Fw1;
    $T[2]=  $T2;  $A[2]=  $A2;  $B[2]=  $B2;  $d[2]=  $d2;  $Fw[2]=  $Fw2;
    $T[3]=  $T3;  $A[3]=  $A3;  $B[3]=  $B3;  $d[3]=  $d3;  $Fw[3]=  $Fw3;
    $T[4]=  $T4;  $A[4]=  $A4;  $B[4]=  $B4;  $d[4]=  $d4;  $Fw[4]=  $Fw4;
    $T[5]=  $T5;  $A[5]=  $A5;  $B[5]=  $B5;  $d[5]=  $d5;  $Fw[5]=  $Fw5;
    $T[6]=  $T6;  $A[6]=  $A6;  $B[6]=  $B6;  $d[6]=  $d6;  $Fw[6]=  $Fw6;
    $T[7]=  $T7;  $A[7]=  $A7;  $B[7]=  $B7;  $d[7]=  $d7;  $Fw[7]=  $Fw7;
    $T[8]=  $T8;  $A[8]=  $A8;  $B[8]=  $B8;  $d[8]=  $d8;  $Fw[8]=  $Fw8;
    $T[9]=  $T9;  $A[9]=  $A9;  $B[9]=  $B9;  $d[9]=  $d9;  $Fw[9]=  $Fw9;
    $T[10]= $T10; $A[10]= $A10; $B[10]= $B10; $d[10]= $d10; $Fw[10]= $Fw10;
    $T[11]= $T11; $A[11]= $A11; $B[11]= $B11; $d[11]= $d11; $Fw[11]= $Fw11;
    $T[12]= $T12; $A[12]= $A12; $B[12]= $B12; $d[12]= $d12; $Fw[12]= $Fw12;
    $T[13]= $T13; $A[13]= $A13; $B[13]= $B13; $d[13]= $d13; $Fw[13]= $Fw13;
    $T[14]= $C14; $A[14]= $A14; $B[14]= $B14; $d[14]= $d14; $Fw[14]= $Fw14;
    $C[15]= $C15; $A[15]= $A15; $B[15]= $B15; $d[15]= $d15; $Fw[15]= $Fw15;

if ($action == "next"){
 $item++;
}
      if ($action == "calculate") {



//loop for changing coupon frequency


          if($CType == 0 )
            $F = 1;
            else
             if($CType == 1)
               $F = 2;
           else
             if($CType == 2 )
               $F = 4;
//
/***********************Time to maturity************************/
$l=0;

  //Maturity array day counting from present day to maturity of bond
 for($i=0;$i<=$item;$i++){
$days = (strtotime($M[$i]) - strtotime(date("Y-m-d"))) / (60 * 60 * 24);
```

```
          $T[$i]=$days/360;
// Number of coupon payments for each bond
$Ncoupons[$i]=(int)($T[$i]/(1/$F));



/****************************Acrrued Interest ********************/
// Acrrued Time in years for each bond
$AT[$i]=$T[$i]-$Ncoupons[$i]*(1/$F);
// Accrued Interest rates
$AI[$i]=$AT[$i]*$C[$i]/360 ;
//Dirty Price
$DP[$i]=$P[$i] + $AI[$i];
}

/******************* First Zero coupon bonds**************************/
// Face value
$R=100;
$d[0]=$DP[0]/$R;
$B[0]=1;$A[0]=($d[0]-1)/$T[0];
$A[2]=$A[0]*$T[0]+1;
$S[0]= 100*$F*(pow(1/$d[0],1/($F*$T[0 ]))-1);
$Fw[0]=$S[0];

/**************************Other Bonds***********************************/
// $Ct- coupons time array
for($i=1;$i<=$item;$i++){
//
$coef_a[$i] = $R*$T[$i];
//$coef_a[$i] = $R;
$coef_b[$i] =$R;

for($j=0;$j <= $Ncoupons[$i];$j++){
// $Ct- coupons time array
$Ct[$i][$j]=$AT[$i] + $j*(1/$F);
// initial position
$Z[$i][$j]=0;
for($k=0;$k<$i;$k++){
// Checking the position of a coupon date
if($Ct[$i][$j]>$T[$k]){$Z[$i][$j]++;}
}
//Checking if a cash flow date is in the current interval
if($Ct[$i][$j]>$T[$i-1]){
$coef_a[$i] +=($C[$i]/$F)*$Ct[$i][$j];
$coef_b[$i] +=$C[$i]/$F;
}
// PV of cash flows to be received before the current time  interval
// since the a and b of the current interval are not yet known, their value
are zero>>the discounted cash //flows are also zero
$W[$i]+=$Ct[$i][$j]*$A[$Z[$i][$j]] +$B[$Z[$i][$j]] ;
}
// Coefficients of a  and  b
// Slope and Intercept of each discount line
$B[$i]=($DP[$i]-$W[$i]-($d[$i-1]/$T[$i-1])*$coef_a[$i])/($coef_b[$i]-
$coef_a[$i]/$T[$i-1]);
$A[$i]=($d[$i-1]-$B[$i])/$T[$i-1];
$d[$i]= $A[$i]*$T[$i ]+ $B[$i];

if($d[$i]>$d[$i-1]){$A[$i]=$A[$i-1]; $B[$i]=$B[$i-1];$d[$i]= $A[$i]*$T[$i
]+ $B[$i];}
//Spot Rates
```

```
//$zRate[$i] =100*$plotz[$Z[$i]]*(pow(1/$Disc[$i] ,
1/($plotz[$Z[$i]]*$years3[$i])) -1);
$S[$i]= 100*$F*(pow(1/$d[$i],1/($F*$T[$i ]))-1);
// Forward rates
$Fw[$i]  = pow(1.0 + $S[$i]/100.0,   $T[$i]);
        $Fw[$i]  /= pow(1.0 + $S[$i-1]/100.0,   $T[$i-1]);
        $Fw[$i]   = 100.0*(pow($Fw[$i] , $F) - 1.0);

}
/*****************Data base*****************************************/
 $link = mysql_connect("localhost","","");
      if (! $link)
        die("Couldn't connect to MySQL");
   $db="Bondpricing";
      mysql_select_db($db , $link)
        or die("Couldn't open $db: ".mysql_error());

      //      $fp = fopen('C:/tmp/test.txt', 'w') or die("can't open
file");
      for ($i = 0; $i <= $item; $i++) {
        $dat = date("Ymd");
        $coupon   = $C[$i];
        $freq     = $F;
        $maturity = $M[$i];
        $price    = $P[$i];
        $quote    = $QType[$i];
       $time   = $T[$i];
      $disc=$d[$i];
     $slope=$A[$i];
       $intercept=$B[$i];
      //   $market   = $MType[$item];
      //      $daycount = $DType[$item];

        switch ($MType[$i]) {
          case '0':
            $market   = "'UK'";
            $name     = "'UK_".$dat."'";
            break;

          case '1':
            $market   = "'US'";
            $name     = "'US_".$dat."'";
            break;

          case '2':
            $market   = "'Ger'";
            $name     = "'Ger_".$dat."'";
            break;

          case '3':
            $market   = "'Swe'";
            $name     = "'Swe_".$dat."'";
            break;
        }

        switch ($DType[$i]) {
          case "0":
            $daycount   = "'Act/365'";
            break;

          case "1":
            $daycount   = "'Act/Act'";
            break;

          case "2":
            $daycount   = "'Act/360'";
            break;

          case "3":
            $daycount   = "'30/360'";
            break;

          case "4":
            $daycount   = "'30/365'";
            break;
        }
        //        fwrite($fp, $market.'  '.$daycount.'  '.$name.'
'.$coupon.'  '.$freq.'  '.$maturity.'  '.$price.'  '.$quote.'
'.$ytm."\n");

        mysql_query ("INSERT INTO discountdata  (market
, daycount ,name, coupon, cfreq, maturity,
                                                price,
                                                quotetype,
                                              time, discount,A,B


)
               VALUES ($market, $daycount,$name,$coupon, $freq,'$maturity',
$price, $quote, $time, $disc,$slope, $intercept) ");
      }
      //      fclose($fp);
   $fp = fopen('t.php', 'w') or die("can't open file");

           fwrite($fp, '<?php $name='.$name.'; ?>');
  fclose($fp);
}
  if ($action == "graph") {

    $wnd = mt_rand(0, 100000000);
    echo "<script LANGUAGE=\"JavaScript\">";
    echo "window.open('discountplot.php', $wnd, 'width=$wx, height=$wy,
scrollbars=yes,resize=yes')";
    echo "</script>";
   }
break;
/**************** Help Button***************************************/
 case "help":
    $wnd = mt_rand(0, 100000000);
echo "<script LANGUAGE=\"JavaScript\">";echo
"window.open('include/Bootstrapping.htm', $wnd,
'toolbar=yes,location=no,directories=no,status=no,menubar=no,scrollbars=yes
,resizable=yes,height=600,width=600')";
    echo "</script>";
  break;
}
$action="";
?>



<FORM name = "rateform" ACTION = "<? $SCRIPT_NAME ?>" METHOD = "POST">
```

```
<p><b><h1 align = "center">Bootstrapping Method</h1></b></p>
<p><b><h2 align = "center">&copy Takoeta Fred, Hamadou Hamaounde
</b></p>Supervised by Jan Röman </b></p>2005</h2></b></p>
<hr align="left">

<p>
<A HREF="http://www.bloomberg.com/markets/rates/index.html"
TARGET=_blank>Prices on Bloomberg</A>
&nbsp;&nbsp&nbsp;&nbsp&nbsp;&nbsp&nbsp;&nbsp&nbsp;&nbsp&nbsp;&nbsp&nbsp;
Market:
      <SELECT NAME="MType">
<?
    if ($MType == 0)
       echo "<OPTION Value=0 SELECTED>UK";
     else
       echo "<OPTION Value=0 >UK";
     if ($MType == 1)
        echo "<OPTION value=1 SELECTED>US";
     else
        echo "<OPTION value=1 >US";
     if ($MType == 2)
        echo "<OPTION Value=2 SELECTED>Ger";
     else
        echo "<OPTION Value=2 >Ger";
     if ($MType == 3)
        echo "<OPTION Value=3 SELECTED>Swe";
      else
        echo "<OPTION Value=3 >Swe";
?>
</SELECT>&nbsp;&nbsp;&nbsp;
Day Count:
      <SELECT NAME="DType">
<?
    if ($DType == 0)
       echo "<OPTION Value=0 SELECTED>Act/365";
     else
       echo "<OPTION Value=0 >Act/365";
     if ($DType == 1)
        echo "<OPTION value=1 SELECTED>Act/Act";
     else
        echo "<OPTION value=1 >Act/Act";
     if ($DType == 2)
        echo "<OPTION Value=2 SELECTED>Act/360";
     else
        echo "<OPTION Value=2 >Act/360";
     if ($DType == 3)
        echo "<OPTION Value=3 SELECTED>30/360";
      else
        echo "<OPTION Value=3 >30/360";
     if ($DType == 4)
       echo "<OPTION Value=3 SELECTED>30/365";
else
        echo "<OPTION Value=3 >30/365";
?>
</SELECT>&nbsp;&nbsp;&nbsp;

</SELECT>&nbsp;&nbsp;&nbsp;
Coupon Frequency:
      <SELECT NAME="CType">
 <?
```

```
        if ($CType == 0)
          echo "<OPTION Value=0 SELECTED>Annual";
        else
         echo "<OPTION Value=0 >Annual";
        if ($CType == 1)
          echo "<OPTION value=1 SELECTED>Semi Annual";
        else
          echo "<OPTION value=1 >Semi Annual";
        if ($CType == 2)
          echo "<OPTION Value=2 SELECTED>Quarterly";
        else
          echo "<OPTION Value=2 >Quarterly";



      ?>

</SELECT>&nbsp;&nbsp;&nbsp;
<INPUT TYPE=submit NAME=action  VALUE=Help>&nbsp;&nbsp</p>


<P><B><h1 align="CENTER">INPUT FIELD</h1></B>


<TABLE WIDTH="530" BORDER="1" CELLSPACING="1" CELLPADDING="0">
   <tr>
      <td>&nbsp;<b>Coupon [%]</b></td>
      <td>&nbsp;<b>Maturity</b></td>
      <td>&nbsp;<b>Price</b></td>
      <td>&nbsp;<b>Quote Type</b></td>
         </tr>
   <tr>
      <td><INPUT TYPE="text" NAME="C0" SIZE="11" VALUE="<? echo $C[0]
?>"></td>


      <td><INPUT TYPE="text" NAME="M0"
          onClick="popUpCalendar(this, rateform.M0, 'mm/dd/yyyy');"
          SIZE="9" VALUE="<? echo $M[0] ?>"></td>
      <td><INPUT TYPE="text" NAME="P0" SIZE="6" VALUE="<? echo $P[0]
?>"></td>
      <td><SELECT NAME="QType[0]">
      <?
        if ($QType[0] == 0)
          echo "<OPTION Value=0 SELECTED>% of Nom.";
        else
          echo "<OPTION Value=0 >% of Nom.";
        if ($QType[0] == 1)
          echo "<OPTION value=1 SELECTED>Yield";
        else
          echo "<OPTION value=1 >Yield";
      ?>
      </select></td>

  </tr>
       <INPUT TYPE="hidden" NAME="item"  VALUE="<? echo $item ?>">
<?
   for ($i = 1; $i <= $item; $i++) {
     echo("<td><INPUT TYPE='text' NAME='C".$i."' SIZE='11'
VALUE=".$C[$i]."></td>\n");
     echo("<td><INPUT TYPE='text' NAME='M".$i."'");
     echo("onClick=\"popUpCalendar(this, rateform.M".$i.",
'mm/dd/yyyy');\"");
     echo("SIZE='9' VALUE=".$M[$i].."></td>\n");
     echo("<td><INPUT TYPE='text' NAME='P".$i."' SIZE='6'
VALUE=".$P[$i].."></td>\n");
     echo("<td><SELECT NAME='QType[".$i."]'>\n");
      if ($QType[$i] == 0)
        echo "<OPTION Value=0 SELECTED>% of Nom.\n";
      else
        echo "<OPTION Value=0>% of Nom.\n";
      if ($QType[$i] == 1)
        echo "<OPTION value=1 SELECTED>Yield\n";
      else
        echo "<OPTION value=1>Yield\n";
     echo("</select></td>");

     echo("</td></tr>");
     echo("<INPUT TYPE='hidden' NAME= 'S".($i-1)."' VALUE=".$S[$i-
1].">\n");

   }
?>
</TABLE></p>
 <INPUT TYPE = submit  NAME = action  VALUE = Next>&nbsp;&nbsp;
 <INPUT TYPE = submit  NAME = action  VALUE = Calculate>&nbsp;&nbsp;
<?

if ($save == 1){
   echo ("<INPUT TYPE = submit  NAME = action  VALUE=Graph>&nbsp;&nbsp;");
  echo ("<INPUT TYPE = submit  NAME = action  VALUE=Clear>");
}


?>
<P><B><h1 align="CENTER">OUTPUT FIELD</h1></B>
<p><TABLE>
<TABLE WIDTH="830" BORDER="1" CELLSPACING="1" CELLPADDING="0">
   <tr>
      <td>&nbsp;<b>Time to Maturity</b></td>
       <td>&nbsp<b>A</b></td>
      <td>&nbsp<b>B</b></td>
      <td>&nbsp<b>Discount factor </b></td>
      <td>&nbsp<b>Spot rate [%]</b></td>
      <td>&nbsp<b>Forward rate [%]</b></td>
   </tr>
   <tr>
<td><INPUT TYPE="text" NAME="T0" SIZE="26" VALUE="<? printf("%.2f",
$T[0]);?>"></td>
       <td><INPUT TYPE="text" NAME="A0" SIZE="12" VALUE="<? printf("%.7f",
$A[0]); ?>"></td>
     <td><INPUT TYPE="text" NAME="B0" SIZE="12" VALUE="<? printf("%.7f",
$B[0]); ?>"></td>
     <td><INPUT TYPE="text" NAME="d0" SIZE="12" VALUE="<? printf("%.2f",
$d[0]); ?>"></td>
     <td><INPUT TYPE="text" NAME="S0" SIZE="12" VALUE="<? printf("%.2f",
$S[0]); ?>"></td>
     <td><INPUT TYPE="text" NAME="Fw0" SIZE="12" VALUE="<? printf("%.2f",
$Fw[0]); ?>"></td>
  </tr>
<?
  for ($i = 1; $i <= $item; $i++) {
     echo("<td><INPUT TYPE='text' NAME='T".$i."' SIZE='12' VALUE=");
     printf("%.2f", $T[$i]);
     echo("></td>\n");
        echo("<td><INPUT TYPE='text' NAME='A".$i."' SIZE='12' VALUE=");
     printf("%.2f", $A[$i] );
     echo("></td>\n");
    echo("<td><INPUT TYPE='text' NAME='B".$i."' SIZE='12' VALUE=");
     printf("%.2f", $B[$i] );
     echo("></td>\n");
   echo("<td><INPUT TYPE='text' NAME='d".$i."' SIZE='12' VALUE=");
     printf("%2f", $d[$i]);
     echo("></td>\n");
     echo("<td><INPUT TYPE='text' NAME='S".$i."' SIZE='12' VALUE=");
     printf("%2f", $S[$i]);
     echo("></td>\n");
     echo("<td><INPUT TYPE='text' NAME='Fw".$i."' SIZE='12' VALUE=");
     printf("%.2f", $Fw[$i]);
  echo("></td></tr>");

}

?>
</TABLE></p>
</FORM>
</BODY>
</html>
/*
Name: discountplot .php
Authors: Takoeta Fred & Hamadou Hamaounde
 Supervisor: Jan Röman
 Description: the functions used to plot the spot rates with the aid of
JPGraph
*/
<?php
include ("./include/jpgraph.php");
include ("./include/jpgraph_log.php");
include ("./include/jpgraph_line.php");
include ("./options/global_data.php");
include ("./t.php");
$db     = "BondPricing";
$plotx  = array();
$ploty  = array();
$link = mysql_connect("localhost","","");
if (! $link)
  die("Couldn't connect to MySQL");

mysql_select_db($db , $link)
  or die("Couldn't open $db: ".mysql_error());

$result = mysql_query ("select name, time, discount from discountdata where
name ='".$name."' order by  time");
$i = 0;
while ($line = mysql_fetch_array($result)) {
  $ploty[$i] = $line['discount'];
  $plotx[$i] = $line['time'];
  fwrite($fp, $plotx[$i].'   '.$ploty[$i]."\n");
  $i++;
}
$color  = "red";
$title  = "Plot of  Discount Function";
$xlabel = "Maturity(Years)";    // X-Axis label title
```

```
$ylabel = "Discount"; // Y-Axis label title
$graph = new Graph($winx, $winy);
$graph->SetScale(linlin);
$graph->SetTickDensity(3, 3);
$graph->img->SetMargin($xm1+20, $xm2, $ym1, $ym2);
$graph->title->SetFont($tfn, $titfns, $titfnz);
$graph->SetBox        (true, $axcol,  $axwth);

$graph->xaxis->SetColor      ($axcol);
$graph->xaxis->SetWeight     ($axwth);
$graph->xaxis->SetTitle      ($xlabel, $xftype);
$graph->xaxis->SetFont       ($tfn,    $titfns, $xfnz);
$graph->xaxis->title->SetFont($tfn,    $titfns, $xfnz);
$graph->xaxis->SetPos("min");

$graph->yaxis->SetColor      ($axcol);
$graph->yaxis->SetWeight     ($axwth);
$graph->yaxis->SetFont       ($tfn, $titfns, $xfnz);
$graph->yaxis->SetTitle      ($ylabel, $yftype);
$graph->yaxis->SetTitleMargin(60);
$graph->yaxis->title->SetFont($tfn, $titfns, $xfnz);

$graph->xgrid->SetColor    ($grcol);
$graph->ygrid->SetColor    ($grcol);
$graph->xgrid->SetWeight   ($grwth);
$graph->ygrid->SetWeight   ($grwth);
$graph->xgrid->SetLineStyle($grtype);
$graph->ygrid->SetLineStyle($grtype);

$graph->SetShadow();
$graph->title->Set($title);
$graph->ygrid->Show(true, false);
$graph->xgrid->Show(true, false);

//$lineplot=new LinePlot($ploty);
$lineplot=new LinePlot($ploty, $plotx);
$lineplot->SetColor($color);
$lineplot->SetStyle($type);
$lineplot->SetWeight($weight);
$graph->Add($lineplot);       // Add the plot to the graph

// Display the graph
$graph->Stroke();
?>
/*
Name: newbond.php
Authors: Takoeta Fred & Hamadou Hamaounde
 Supervisor: Jan Röman
 Description: main window and function for pricing bonds.
*/
<HTML>
<HEAD>
  <META NAME="GENERATOR" CONTENT="Emacs 20.6.1">
  <TITLE>Bond Price Calculator</TITLE>
  <style type="text/css">
  <!--
  a{text-decoration:none}
  a:active { color: #666666}
  a:hover { color: #666666}
  a:link { color: #666666}
  a:visited { color: #666666}
  -->
  </style>
  <style type="text/css">
  <!--
  #logo {position:absolute; left:0px; top:0px;}
  -->
  </style>
</HEAD>

<BODY>

<?php
include ("./include/Calendar.html");
include ("bonds[1].php");
DEFINE (UK,  0);
DEFINE (US,  1);
DEFINE (GER, 2);
DEFINE (SWE, 3);
Global $CouponRate, $FlatRate, $CFPerYear, $Price, $action,
$Days,$Days3,$years3, $Maturity, $forts, $tmp2;$Z;
Global $Coupon, $YTM, $diff, $num_cf, $tim, $HaveRates, $Dur, $MDur,
$Days2, $Cnvx, $i;
Global $zRate, $cCF, $par, $tmp, $CSpread, $fRate, $DeltaYield,
$GammaYield, $Theta, $Val01;
Global $MPrice, $MSpread, $Clean; $Accrued;

if ($CouponRate == "") $CouponRate = 5.0;
if ($FlatRate   == "") $FlatRate   = 4.0;
if ($CFPerYear  == "") $CFPerYear  = 2;
//if ( $Maturity  == "") $Maturity   = "05/18/2011";
if ($YTM        == "") $YTM        = 0.0;
if ($par        == "") $par        = 0.0;
if ($CSpread    == "") $CSpread    = 0.0;
if ($DeltaYield == "") $DeltaYield = 0.0;
if ($GammaYield == "") $GammaYield = 0.0;
if ($Theta      == "") $Theta      = 0.0;
if ($Val01      == "") $Val01      = 0.0;
if ($HaveRates  == "") $HaveRates  = 0;

$zRate = array();
$fRate = array();
$cCF   = array();
$Z =array();
$action = strtolower(trim($action));
if ($FlatRate != 0.0)
  $YTM = $FlatRate;
else
  $YTM = 0.0;

switch ($action) {
/*****************************Clear Button********************************/
  case "clear":
    $CouponRate = "";
    $Coupon     = "";
    $FlatRate   = "";
    $CFPerYear  = "";
    $Maturity   = "";
    $Days       = "";
    $Days2      = "";
    $YTM        = "";
    $par        = "";
    $forts      = 0;
    break;

/****************Calculate Button*********************************/
  case "calculate":
    $HaveRates = 1;
    $forts  = 1;
    $Price  = 0;
    $Days   = Round((strtotime($Maturity) - strtotime(now))/86400);
    $Days2  = $Days;
    $tim    = 365.0/$CFPerYear;
    $tim    = Round(365.0/$CFPerYear);
    $Coupon = $CouponRate/$CFPerYear;;

    do {
      $Days2 -= $tim;
    } while ($Days2 > 0);
    $Days2 += $tim;
    $Days2 = Floor($Days2);

    if ($FlatRate != 0.0) {
      $Price = BondPrice($Coupon, 1, $FlatRate/100.0, $Days2, $tim, $Days);
      $Dur   = Duration ($Coupon, 1, $FlatRate/100.0, $Days2, $tim, $Days,
$Price);
      $MDur  = ModDur($Dur, $FlatRate/100.0);
      $Cnvx  = Duration ($Coupon, 1, ($FlatRate - 0.00001)/100.0, $Days2,
$tim, $Days, $Price);
      $tmp   = Duration ($Coupon, 1, ($FlatRate + 0.00001)/100.0, $Days2,
$tim, $Days, $Price);
      $Cnvx  = ($Cnvx - $tmp)/0.00002;
      $tmp   = BondPrice($Coupon, 1, ($FlatRate + 0.0001)/100.0, $Days2,
$tim, $Days);
      $tmp2  = BondPrice($Coupon, 1, ($FlatRate + 0.0002)/100.0, $Days2,
$tim, $Days);
      $DeltaYield = ($tmp  - $Price)*1000.0;
      $GammaYield = ($tmp2 - 2.0*$tmp + $Price)*1000.0*1000.0;
      $Theta = $Price - BondPrice($Coupon, 1, $FlatRate/100.0, $Days2,
$tim, $Days+1);
      $Val01 = BondPrice($Coupon, 1, ($FlatRate - 0.0005)/100.0, $Days2,
$tim, $Days) -
               BondPrice($Coupon, 1, ($FlatRate + 0.0005)/100.0, $Days2,
$tim, $Days);
    }
    else {
      $num_cf = Floor($CFPerYear*$Days/365 + 1 );

      $zRate[0]  = $zRate0;   $zRate[1]  = $zRate1;   $zRate[2]  = $zRate2;
$zRate[3]  = $zRate3;
      $zRate[4]  = $zRate4;   $zRate[5]  = $zRate5;   $zRate[6]  = $zRate6;
$zRate[7]  = $zRate7;
      $zRate[8]  = $zRate8;   $zRate[9]  = $zRate9;   $zRate[10] =
$zRate10;  $zRate[11] = $zRate11;
      $zRate[12] = $zRate12;  $zRate[13] = $zRate13;  $zRate[14] =
$zRate14;  $zRate[15] = $zRate15;
      $zRate[16] = $zRate16;  $zRate[17] = $zRate17;  $zRate[18] =
$zRate18;  $zRate[19] = $zRate19;
      $zRate[20] = $zRate20;  $zRate[21] = $zRate21;  $zRate[22] =
$zRate22;  $zRate[23] = $zRate23;
      $zRate[24] = $zRate24;  $zRate[25] = $zRate25;  $zRate[26] =
$zRate26;  $zRate[27] = $zRate27;
```

```
      $zRate[28] = $zRate28;  $zRate[29] = $zRate29;  $zRate[30] =
$zRate30;  $zRate[31] = $zRate31;
      $zRate[32] = $zRate32;  $zRate[33] = $zRate33;  $zRate[34] =
$zRate34;  $zRate[35] = $zRate35;
      $zRate[36] = $zRate36;  $zRate[37] = $zRate37;  $zRate[38] =
$zRate38;  $zRate[39] = $zRate39;
      $zRate[40] = $zRate40;  $zRate[41] = $zRate41;  $zRate[42] =
$zRate42;  $zRate[43] = $zRate43;
      $zRate[44] = $zRate44;  $zRate[45] = $zRate45;  $zRate[46] =
$zRate46;  $zRate[47] = $zRate47;
      $zRate[48] = $zRate48;  $zRate[49] = $zRate49;  $zRate[50] =
$zRate50;  $zRate[51] = $zRate51;
      $zRate[52] = $zRate52;  $zRate[53] = $zRate53;  $zRate[54] =
$zRate54;  $zRate[55] = $zRate55;
      $zRate[56] = $zRate56;  $zRate[57] = $zRate57;  $zRate[58] =
$zRate58;  $zRate[59] = $zRate59;
      $zRate[50] = $zRate60;
      $Price     = 0.0;

      for ($i = 0; $i < $num_cf-1; $i++) {
        if ($zRate[$i] > 0) {
          $cCF[$i] = $Coupon/pow(1.0 + $zRate[$i]/100.0, ($Days2 +
$i*$tim)/365.0);
        }
        if ($i == 0) {
          $fRate[$i] = $zRate[$i];
        }
        else {
          $fRate[$i]  = pow(1.0 + $zRate[$i]/100.0,   ($Days2 +
$i*$tim)/365.0);
          $fRate[$i] /= pow(1.0 + $zRate[$i-1]/100.0, ($Days2 + ($i-
1)*$tim)/365.0);
          $fRate[$i]  = 100.0*(pow($fRate[$i], 365.0/$tim) - 1.0);
        }
      }
      $zRate[$num_cf-1] = $zRate[$num_cf-1];
      if ($zRate[$num_cf-1] > 0) {
        $cCF[$num_cf-1]   = (100.0 + $Coupon)/pow(1.0 + $zRate[$num_cf-
1]/100.0, $Days/365.0);
       $fRate[$num_cf-1] = $fRate[$num_cf-2];
        $HaveRates        = 1;
      }
      $Price = BondPrice2($Coupon, $zRate, $num_cf, 1, $Days2, $tim, $Days,
$CSpread);
      $Dur   = Duration2 ($Coupon, $zRate, $num_cf, 1, $Days2, $tim, $Days,
$Price, $CSpread);
      $YTM   = 100.0*ytm(0, 1.0, $Coupon, 1, $Days2, $tim, $Days, $Price);
      $par   = 100.0*ytm(0, 1.0, $Coupon, 1, $Days2, $tim, $Days, 100.0);
      $MDur  = ModDur($Dur, $YTM/100.0);
      $Cnvx  = Duration2($Coupon, $zRate, $num_cf, 1, $Days2, $tim, $Days,
$Price, $CSpread-0.00001);
      $tmp   = Duration2($Coupon, $zRate, $num_cf, 1, $Days2, $tim, $Days,
$Price, $CSpread+0.00001);
      $tmp2  = Duration2($Coupon, $zRate, $num_cf, 1, $Days2, $tim, $Days,
$Price, $CSpread+0.00002);
      $Cnvx  = ($Cnvx - $tmp)/0.00002;
      $tmp   = BondPrice2($Coupon, $zRate, $num_cf, 1, $Days2, $tim, $Days,
$CSpread+0.0001);
      $tmp2  = BondPrice2($Coupon, $zRate, $num_cf, 1, $Days2, $tim, $Days,
$CSpread+0.0002);
      $DeltaYield = ($tmp  - $Price)*1000.0;
      $GammaYield = ($tmp2 - 2.0*$tmp + $Price)*1000.0*1000.0;
      $Theta = $Price - BondPrice2($Coupon, $zRate, $num_cf, 1, $Days2,
$tim, $Days+1, $CSpread);
      $Val01 = BondPrice($Coupon, 1, ($YTM - 0.0005)/100.0, $Days2, $tim,
$Days) -
               BondPrice($Coupon, 1, ($YTM + 0.0005)/100.0, $Days2, $tim,
$Days);
      $MSpread = spread(-10.0, 10.0, $Coupon, $zRate, $num_cf, 1, $Days2,
$tim, $Days, 0.0, $MPrice);
      $Accrued = $Coupon*($tim-$Days2)/$Days2;
      $Clean   = $Price - $Accrued;
    }
    break;



/***********************Cash flow Button******************************/
  case "cash flow":
    $HaveRates  = 1;
    $forts      = 1;
  if ($CouponRate == "" || $CFPerYear == "" || $FlatRate == "" || $Maturity
== "") {
      echo("<script language='JavaScript'>\n");
      echo("alert('Missing input data!!!')\n");
      echo("</script>\n");
     $Maturity   = "05/18/2011";
       $FlatRate   = 0;
}
   $Days       = Round((strtotime($Maturity) - strtotime(now))/86400);
    $Days2      = $Days;
    $tim        = Round(365.0/$CFPerYear);
    $Coupon     = $CouponRate/$CFPerYear;;
    $num_cf     = Floor($CFPerYear*$Days/365 + 1 );

    do {
      $Days2 -= $tim;
    } while ($Days2 > 0);
    $Days2 += $tim;
    $Days2 = Floor($Days2);
/*************************market type******************************/
    $dat = date("Ymd");
switch ($MType) {
          case '0':
            $market   = "'UK'";
            $name     = "'UK_".$dat."'";
            break;

          case '1':
            $market   = "'US'";
            $name     = "'US_".$dat."'";
            break;

          case '2':
            $market   = "'Ger'";
            $name     = "'Ger_".$dat."'";
            break;

          case '3':
            $market   = "'Swe'";
            $name     = "'Swe_".$dat."'";
            break;
```

```
        }

    /***************Fetch discount data from data base******************/
$db     = "BondPricing";
$plotx  = array();
$ploty  = array();
//$fp = fopen('C:/tmp/test.txt', 'w') or die("can't open file");
$link = mysql_connect("localhost","","");
if (! $link)
  die("Couldn't connect to MySQL");

mysql_select_db($db , $link)
  or die("Couldn't open $db: ".mysql_error());

$result = mysql_query ("select name, time, discount, A, B, cfreq from
discountdata where name =".$name." order by  time");

$i = 0;
while ($line = mysql_fetch_array($result)) {
  //  $plotx[$i] = $line['maturity'];
  $ploty[$i] = $line['discount'];
  $plotx[$i] = $line['time'];
$plotz[$i] = $line['cfreq'];
  $plotA[$i] = $line['A'];
  $plotB[$i] = $line['B'];
$ex='&&&&       ';
  //$plotx[$i] = (strtotime($line['maturity']) - strtotime(date("Y-m-
d")))/(60*60*24/360);
//echo $plotx[$i] ;
//echo $ex ;
  //echo $plotB[$i] ;
//echo $ex ;
 // fwrite($fp, $plotx[$i].'   '.$ploty[$i]."\n");
  $i++;$k=$i;
}
//fclose($fp);

 /***locate corporate bond cash flow time and find the corresponding
discount factor *********/

// k=length of $plotx array
//$k=2;
//$plotx=array(1,2,3);

 $years3[$num_cf-1]= $Days/365;
 for ($i = 0; $i <=$num_cf-1; $i++) {
$Z[$i]=0;
  $Days3= $Days2 + $i*$tim;
    $years3[$i]= $Days3/365;

   for ($j=0;$j<$k;$j++){
if($years3[$i] > $plotx[$j]){$Z[$i]++;}
}


 /******Compute corresponding   discount factor ************************/

$Disc[$i]=$plotA[$Z[$i]]*$years3[$i] + $plotB[$Z[$i]];

 /******Convert discount data to Spot rates data ***********************/
```

```
$zRate[$i] =100*$plotz[$Z[$i]]*(pow(1/$Disc[$i] ,
1/($plotz[$Z[$i]]*$years3[$i])) -1);

}

    for ($i = 0; $i < $num_cf-1; $i++) {

     $zRate[$i]   =$zRate[$i];

      if ($zRate[$i] > 0) {
        $cCF[$i] = $Coupon/pow(1.0 + $zRate[$i]/100.0, ($Days2 +
$i*$tim)/365.0);
      }
      if ($i == 0) {
        $fRate[$i] = $zRate[$i];
      }
      else {
        $fRate[$i]  = pow(1.0 + $zRate[$i]/100.0,   ($Days2 +
$i*$tim)/365.0);
        $fRate[$i] /= pow(1.0 + $zRate[$i-1]/100.0, ($Days2 + ($i-
1)*$tim)/365.0);
        $fRate[$i]  = 100.0*(pow($fRate[$i], 365.0/$tim) - 1.0);
      }
    }
    $zRate[$num_cf-1] = $zRate[$num_cf-1];

    if ($zRate[$num_cf-1] > 0) {
      $cCF[$num_cf-1]   = (100.0 + $Coupon)/pow(1.0 + $zRate[$num_cf-
1]/100.0, $Days/365.0);
      $fRate[$num_cf-1] = $fRate[$num_cf-2];
      $HaveRates        = 1;
    }
    $Price = BondPrice2($Coupon, $zRate, $num_cf, 1, $Days2, $tim, $Days,
$CSpread);
    $Dur   = Duration2 ($Coupon, $zRate, $num_cf, 1, $Days2, $tim, $Days,
$Price, $CSpread);
    $YTM   = 100.0*ytm(0, 1.0, $Coupon, 1, $Days2, $tim, $Days, $Price);
    $par   = 100.0*ytm(0, 1.0, $Coupon, 1, $Days2, $tim, $Days, 100.0);
    $MDur  = ModDur($Dur, $YTM/100.0);
    $Cnvx  = Duration2($Coupon, $zRate, $num_cf, 1, $Days2, $tim, $Days,
$Price, $CSpread-0.00001);
    $tmp   = Duration2($Coupon, $zRate, $num_cf, 1, $Days2, $tim, $Days,
$Price, $CSpread+0.00001);
    $Cnvx  = ($Cnvx - $tmp)/0.00002;
    $tmp   = BondPrice2($Coupon, $zRate, $num_cf, 1, $Days2, $tim, $Days,
$CSpread+0.0001);
    $tmp2  = BondPrice2($Coupon, $zRate, $num_cf, 1, $Days2, $tim, $Days,
$CSpread+0.0002);
    $DeltaYield = ($tmp  - $Price)*1000.0;
    $GammaYield = ($tmp2 - 2.0*$tmp + $Price)*1000.0*1000.0;
    $Theta = $Price - BondPrice2($Coupon, $zRate, $num_cf, 1, $Days2, $tim,
$Days+1, $CSpread);
    $Val01 = BondPrice($Coupon, 1, ($YTM - 0.0005)/100.0, $Days2, $tim,
$Days) -
             BondPrice($Coupon, 1, ($YTM + 0.0005)/100.0, $Days2, $tim,
$Days);
    $MSpread = spread(-1.0, 1.0, $Coupon, $zRate, $num_cf, 1, $Days2, $tim,
$Days, 0.0, $MPrice);
    $Accrued = $Coupon*($tim-$Days2)/$Days2;
    $Clean   = $Price - $Accrued;
    break;
}
$action="";
?>

<? /***************HTML FORM******************************************/ ?>

<form name="bondform" ACTION="" METHOD="POST">
<p><b><h1 align = "center">Bond Price Calculator</h1></b></p>
<p><b><h2 align = "center">&copy Takoeta Fred and Hamadou Hamaounde
</h2></b></p>
<p><b><h3 align = "center">Supervised by Jan Röman, 2005</h></b></p>
<hr align="left">
<table border=0 width=720 cellPadding=0 cellSpacing=0 bgcolor="#E9F2F8">
  <tr>
    <TD width=170><font color="#666666" face="Verdana, Arial, Helvetica,
                   sans-serif" size=2>Input variables</font></TD>
    <TD width=100><font color="#666666" face="Verdana, Arial, Helvetica,
                   sans-serif" size=2>Input values</font></TD>
  </tr>
  <tr>
    <td><img src="../images/pixel.jpg" width=7></td>
  </tr>
<tr>
    <TD><font color="#666666" face="Verdana, Arial, Helvetica,
         sans-serif" size=2>Market:</font></TD>
    <TD><SELECT NAME="MType">
<?
    if ($MType == 0)
       echo "<OPTION Value=0 SELECTED>UK";
     else
       echo "<OPTION Value=0 >UK";
     if ($MType == 1)
        echo "<OPTION value=1 SELECTED>US";
     else
        echo "<OPTION value=1 >US";
     if ($MType == 2)
        echo "<OPTION Value=2 SELECTED>Ger";
     else
        echo "<OPTION Value=2 >Ger";
     if ($MType == 3)
        echo "<OPTION Value=3 SELECTED>Swe";
      else
        echo "<OPTION Value=3 >Swe";
?>
</tr>
  <tr>
    <TD><font color="#666666" face="Verdana, Arial, Helvetica,
         sans-serif" size=2>Annual Coupon rate [%]:</font></TD>
    <TD><INPUT size=12 value="<? echo $CouponRate ?>" name=CouponRate
style="width=90"></TD>
  </tr>
  <tr>
    <TD><font color="#666666" face="Verdana, Arial, Helvetica,
         sans-serif" size=2>Coupons per year:</font></TD>
    <TD><INPUT size=12 value="<? echo $CFPerYear ?>" name=CFPerYear
style="width=90"></TD>
  </tr>
  <tr>
    <TD><font color="#666666" face="Verdana, Arial, Helvetica,
         sans-serif" size=2>Flat Rate [%]:</font></TD>
    <TD><INPUT size=12 value="<? echo $FlatRate ?>" name=FlatRate
style="width=90"></TD>
    <TD><font color="#666666" face="Verdana, Arial, Helvetica,
         sans-serif" size=2> Zero if not used!</font></TD>
  </tr>
  <tr>
    <TD><font color="#666666" face="Verdana, Arial, Helvetica,
         sans-serif" size=2>Maturity:</font></TD>
    <TD><INPUT TYPE="text" size=12 onClick="popUpCalendar(this,
bondform.Maturity, 'mm/dd/yyyy');"
         value="<? echo $Maturity ?>" name=Maturity style="width=90"></TD>
    <TD><INPUT size=12 value="<? echo $Days ?>" name=Days style="width=90">
        <font color="#666666" face="Verdana, Arial, Helvetica,
         sans-serif" size=2> Days</font></TD>
    <TD><font color="#FF0000" face="Verdana, Arial, Helvetica,
         sans-serif" size=2>First CF in:</font>
        <INPUT size=12 value="<? echo $Days2 ?>" name=Days2
style="width=90">
        <font color="#666666" face="Verdana, Arial, Helvetica,
         sans-serif" size=2> Days</font></TD>
  </tr>
  <tr>
    <TD>&nbsp;</TD>
    <TD>&nbsp;</TD>
  </tr>


</table>
<?
/****************if ($FlatRate == 0 && $forts)***************************/

if ($FlatRate == 0 && $forts) {

  printf("<table border=0 width=720 cellPadding=0 cellSpacing=0
bgcolor='#E9F2F8'>\n");
  printf("<tr>\n");
  printf("<TD><font color='#666666' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Date</font></TD>\n");
  printf("<TD><font color='#666666' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Cash Flows</font></TD>\n");
  printf("<TD><font color='#666666' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Spot Rates</font></TD>\n");
  printf("<TD><font color='#666666' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Days</font></TD>\n");
  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Forward Rates</font></TD>\n");
  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>PV</font></TD>\n");
  printf("</tr>\n");
  printf("<tr>\n");
  printf("<td><img src='../images/pixel.jpg' width=7></td>\n");
  printf("</tr>\n");


/*****************************  cash flow table***************************/
  for ($i = 0; $i < $num_cf-1; $i++) {

  $Days3= $Days2 + $i*$tim;
    $years3[$i]= $Days3/365;
   $years3[$num_cf-1]= $Days/365;
    printf("<tr>\n");
    printf("<TD><INPUT type='text' name='cDate".$i."' size=12 value='\n");
    printf(date("m/d/Y", mktime(0, 0, 0, date("m"), date("d") + $Days2 +
$i*$tim , date("Y"))));
    printf("' style='width=90'></TD>\n");

    printf("<TD><INPUT size=12 value='\n");
    printf('%.3f', $Coupon);
    printf("' style='width=90'></TD>\n");

    printf("<TD><INPUT type='text' name='zRate".$i."' size=12 value='");
    printf('%.3f', $zRate[$i]);
    printf("' style='width=90'></TD>\n");

    printf("<TD><INPUT size=12 value='\n");
    printf('%d', $Days3);
    printf("' style='width=90'></TD>\n");

    printf("<TD><INPUT type='text' name='fRate".$i."' size=12 value='");
    printf('%.6f', $fRate[$i]);
    printf("' style='width=90'></TD>\n");

    printf("<TD><INPUT type='text' name='cCF".$i."' size=12 value='");
    printf('%.6f', $cCF[$i]);
    printf("' style='width=90'></TD>\n");
    printf("</tr>\n");
  }
  printf("<tr>\n");
  printf("<TD><INPUT type='text' name='cDate".($num_cf-1)."' size=12
value='\n");
  printf($Maturity);
//printf(date("m/d/Y", mktime(0, 0, 0, date("m"), date("d") + $Days,
date("Y"))));
  printf("' style='width=90'></TD>\n");

  printf("<TD><INPUT size=12 value='\n");
  printf('%.3f', 100.0 + $Coupon);
  printf("' style='width=90'></TD>\n");

  printf("<TD><INPUT type='text' name='zRate".($num_cf-1)."' size=12
value='");
  printf('%.3f', $zRate[$num_cf-1]);
  printf("' style='width=90'></TD>\n");

  printf("<TD><INPUT size=12 value='\n");
  printf('%d', $Days);
  printf("' style='width=90'></TD>\n");

  printf("<TD><INPUT type='text' name='fRate".($num_cf-1)."' size=12
value='");
  printf('%.6f', $fRate[$num_cf-1]);
  printf("' style='width=90'></TD>\n");

  printf("<TD><INPUT type='text' name='cCF".($num_cf-1)."' size=12
value='");
  printf('%.6f', $cCF[$num_cf-1]);
  printf("' style='width=90'></TD>\n");
  printf("</tr>\n");
  printf("<tr>\n");
  printf("<TD>&nbsp;</TD>\n");
  printf("</tr>\n");
```

```
  printf("</table>\n");

/***************  Spread Table***********************************/

  printf("<table border=0 width=720 cellPadding=0 cellSpacing=0
bgcolor='#E9F2F8'>\n");
  printf("<tr>\n");
  printf("</tr>\n");
  printf("<tr>\n");
  printf("<TD><font color='#666666' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Aditional Credit Spread [%%]:</font>\n");
  printf("<INPUT size=12 value='\n");
  printf('%.3f', $CSpread);
  printf("'name=CSpread style='width=90'></TD>\n");
  printf("<TD><font color='#666666' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Market Price:</font>\n");
  printf("<INPUT size=12 value='\n");
  printf('%.3f', $MPrice);
  printf("'name=MPrice style='width=90'></TD>\n");
  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Market Spread:</font>\n");
  printf("<INPUT size=12 value='\n");
  printf('%.6f', $MSpread);
  printf("'style='width=90'></TD>\n");
  printf("</tr>\n");
  printf("</table>\n");
}


/*************if ($HaveRates)*****************************************/

if ($HaveRates) {

/**************************Risk Measures*******************************/

  printf("<table border=0 width=720 cellPadding=0 cellSpacing=0
bgcolor='#E9F2F8'>\n");

  printf("<tr>\n");
  printf("<TD>&nbsp;</TD>\n");
  printf("</tr>\n");

  printf("<tr>\n");
  printf("<TD width=170><font color='#666666' face='Verdana, Arial,
Helvetica, sans-serif' size=2>Output variables</font></TD>\n");
  printf("<TD width=100><font color='#666666' face='Verdana, Arial,
Helvetica, sans-serif' size=2>Output values</font></TD>\n");
  printf("</tr>\n");

  printf("<tr>\n");
  printf("<td><img src='../images/pixel.jpg' width=7></td>\n");
  printf("</tr>\n");

  printf("<tr>\n");
  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Yield-To-Maturity [%%]:</font></TD>\n");
  printf("<TD><INPUT size=12 value='\n");
  printf('%.3f', $YTM);
  printf("'name=YTM style='width=90'></TD>\n");
  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Par Yield [%%]:</font></TD>\n");
  printf("<TD><INPUT size=12 value='\n");
  printf('%.3f', $par);
  printf("'name=par style='width=90'></TD>\n");

  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Theor Price:</font></TD>\n");
  printf("<TD><INPUT size=12 value='\n");
  printf('%.3f', $Price);
  printf("' style='width=90'></TD>\n");
  printf("</tr>\n");

  printf("<tr>\n");
  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Macaulay Duration:</font></TD>\n");
  printf("<TD><INPUT size=12 value='\n");
  printf('%.3f', $Dur);
  printf("' style='width=90'></TD>\n");

  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Modified Duration:</font></TD>\n");
  printf("<TD><INPUT size=12 value='\n");
  printf('%.3f', $MDur);
  printf("' style='width=90'></TD>\n");

  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Convexity:</font></TD>\n");
  printf("<TD><INPUT size=12 value='\n");
  printf('%.6f', $Cnvx);
  printf("' style='width=90'></TD>\n");
  printf("</tr>\n");

  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Delta Yield:</font></TD>\n");
  printf("<TD><INPUT size=12 value='\n");
  printf('%.6f', $DeltaYield);
  printf("' style='width=90'></TD>\n");

  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Gamma Yield:</font></TD>\n");
  printf("<TD><INPUT size=12 value='\n");
  printf('%.6f', $GammaYield);
  printf("' style='width=90'></TD>\n");

  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Theta:</font></TD>\n");
  printf("<TD><INPUT size=12 value='\n");
  printf('%.6f', $Theta);
  printf("' style='width=90'></TD>\n");

  printf("<tr>\n");
  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Val01:</font></TD>\n");
  printf("<TD><INPUT size=12 value='\n");
  printf('%.6f', $Val01);
  printf("' style='width=90'></TD>\n");

  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Accrued Interest:</font></TD>\n");
  printf("<TD><INPUT size=12 value='\n");
  printf('%.3f', $Accrued);
  printf("' style='width=90'></TD>\n");

  printf("<TD><font color='#FF0000' face='Verdana, Arial, Helvetica, sans-
serif' size=2>Clean Price:</font></TD>\n");
  printf("<TD><INPUT size=12 value='\n");
  printf('%.3f', $Clean);
  printf("' style='width=90'></TD>\n");
  printf("</tr>\n");

  printf("</table>\n");
}
?>

<table cellSpacing=1 cellPadding=0 width=720 border=0 bgcolor="#E9F2F8">
  <tr>
    <td>
    <br>
<?
if ($HaveRates) {
  printf("<input type='submit' name='action'
value='Clear'>&nbsp;&nbsp;&nbsp;&nbsp;\n");
  printf("<input type='submit' name='action' value='Calculate'>\n");
}
else {
  printf("<input type='submit' name='action'
value='Clear'>&nbsp;&nbsp;&nbsp;&nbsp;\n");
  //printf("<input type='submit' name='action'
value='Continue'>&nbsp;&nbsp;&nbsp;&nbsp;\n");
  //printf("<input type='submit' name='action' value='Example
1'>&nbsp;&nbsp;&nbsp;&nbsp;\n");
  printf("<input type='submit' name='action' value='Cash
Flow'>&nbsp;&nbsp;&nbsp;&nbsp;\n");
}
?>
  </TD>
  </tr>
</table>



</form>
</body>
</html>


/*
Name: bonds[1].php
Authors: Takoeta Fred & Hamadou Hamaounde
 Supervisor: Jan Röman
 Description:  functions for bond pricing and risk measures.
*/


<?php
DEFINE ("EPS0",   "1.0e-7");

function BondPrice($Coupon, $DayCount, $YTM, $Day1, $DeltaDays, $Maturity)
{
  $Price = 0.0;
  for ($day = $Day1; $day <= $Maturity; $day += $DeltaDays) {
    $Price += $Coupon/pow(1.0 + $YTM, $day/365.0);
  }
  $Price += 100.0/pow(1.0 + $YTM, $Maturity/365.0);
  return $Price;
}

function BondPrice2($Coupon, $SpotRate, $N, $DayCount, $Day1, $DeltaDays,
$Maturity, $s)
{
  $Price = 0.0;
  $i     = 0;

  for ($day = $Day1; $day <= $Maturity; $day += $DeltaDays) {
    $Price += $Coupon/pow(1.0 + ($SpotRate[$i] + $s)/100.0, $day/365.0);
    $i++;
  }
  $Price += 100.0/pow(1.0 + ($SpotRate[$N-1] + $s)/100.0, $Maturity/365.0);
  return $Price;
}

function Duration($Coupon, $DayCount, $YTM, $Day1, $DeltaDays, $Maturity,
$Price)
{
  $MDur = 0.0;

  for ($day = $Day1; $day <= $Maturity; $day += $DeltaDays) {
    $MDur += $day*$Coupon/pow(1.0 + $YTM, $day/365.0);
  }
  $MDur += $Maturity*100.0/pow(1.0 + $YTM, $Maturity/365.0);
  $MDur /= $Price*365;
  return $MDur;
}

function Duration2($Coupon, $SpotRate, $N, $DayCount, $Day1, $DeltaDays,
                   $Maturity, $Price, $spread)
{
  $MDur = 0.0;
  $i    = 0;

  for ($day = $Day1; $day <= $Maturity; $day += $DeltaDays) {
    $MDur += $day*$Coupon/pow(1.0 + ($SpotRate[$i] + $spread)/100.0,
$day/365.0);
    $i++;
  }
  $MDur += $Maturity*100.0/pow(1.0 + ($SpotRate[$N-1] + $spread)/100.0,
$Maturity/365.0);
  $MDur /= $Price*365;
  return $MDur;
}

function ModDur($Dur, $YTM)
{
  return $Dur/(1.0 + $YTM);
}

function BondPrice_debug($Coupon, $DayCount, $YTM, $Day1, $DeltaDays,
$Maturity)
{
  $Price = 0.0;
  $fp    = fopen("./result.html", "w");
  fwrite($fp, "<HTML>\n");
  fwrite($fp, "<HEAD>\n");
  fwrite($fp, "    fwrite($fp, "  <TITLE>Bonds Dubeg</TITLE>\n");
  fwrite($fp, "</HEAD>\n");
  fwrite($fp, "

  for ($day = $Day1; $day <= $Maturity; $day += $DeltaDays) {
    fwrite($fp, "Price += " . $Coupon. "/(pow(1.0 + " . $YTM . ", " . $day
. "/365) = ");
    fwrite($fp, $Coupon/pow(1.0 + $YTM, $day/365.0) . "\n");
    $Price = $Price + $Coupon/pow(1.0 + $YTM, $day/365.0);
    fwrite($fp, "Price  = " . $Price . "\n");
  }
  fwrite($fp, "Price += 100/(pow(1 + " . $YTM . ", " . $Maturity . "/365) =
" . 100.0/pow(1.0 + $YTM, $Maturity/365.0) . "\n");
$Price = $Price + 100.0/pow(1.0 + $YTM, $Maturity/365.0);
  fwrite($fp, "Price  = " . $Price . "\n");
  fwrite($fp, "<BR>\n");
  fwrite($fp, "  fwrite($fp, "</HTML>\n");
  fclose($fp);
  $wnd = mt_rand(0, 100000000);
  echo "  echo "window.open('result.html', $wnd,
'toolbar=yes,location=no,directories=yes,status=tes,menubar=tes,scrollbars=
yes,resizable=yes,height=800,width=600')";
  echo "</script>";
  return $Price;
}

/* ---------------------------------------------------------------------*/

function f_vald($x1, $f, $df, $Coupon, $DayCount, $Day1, $DeltaDays,
                $Maturity, $Price)
{
  $DH = 1.0e-5;

  $h = $DH*MAX(1.0, abs($x1));
  $x2  = $x1 - $h;
  $x3  = $x1 + $h;
  $x1  = BondPrice($Coupon, $DayCount, $x1, $Day1, $DeltaDays, $Maturity)
       - $Price;
  $x2  = BondPrice($Coupon, $DayCount, $x2, $Day1, $DeltaDays, $Maturity)
       - $Price;
  $x3  = BondPrice($Coupon, $DayCount, $x3, $Day1, $DeltaDays, $Maturity)
       - $Price;
  $f   = $x1;
  $df  = ($x3 - $x2)/(2*$h);
}

function f_vald2($x1, $f, $df, $Coupon, $SpotRate, $N,
                 $DayCount, $Day1, $DeltaDays, $Maturity, $spread, $Price)
{
  $DH = 1.0e-5;

  $h = $DH*MAX(1.0, abs($x1));
  $x2  = $x1 - $h;
  $x3  = $x1 + $h;
  $x1  = BondPrice2($Coupon, $SpotRate, $N, $DayCount, $Day1, $DeltaDays,
$Maturity, $x1)
       - $Price;
  $x2  = BondPrice2($Coupon, $SpotRate, $N, $DayCount, $Day1, $DeltaDays,
$Maturity, $x2)
       - $Price;
  $x3  = BondPrice2($Coupon, $SpotRate, $N, $DayCount, $Day1, $DeltaDays,
$Maturity, $x3)
       - $Price;
  $f   = $x1;
  $df  = ($x3 - $x2)/(2*$h);
}

// --------------------------------------------------------------------------

function ytm($xmin, $xmax, $Coupon, $DayCount, $Day1, $DeltaDays,
             $Maturity, $Price)
{
  $xacc  = 1.0e-7;
  $maxit = 30;
  $x     = $xmin;
  $fp    = BondPrice($Coupon, $DayCount, $xmin, $Day1, $DeltaDays,
$Maturity)
         - $Price;
  $fc    = BondPrice($Coupon, $DayCount, $xmax, $Day1, $DeltaDays,
$Maturity)
         - $Price;

  if ($fc*$fp < 0.0 || $fc == 0.0) {
    $root = rtsafe($xmin, $xmax, $xacc, $maxit, $Coupon, $DayCount, $Day1,
                   $DeltaDays, $Maturity, $Price, 1);
  }
  return $root;
}

// --------------------------------------------------------------------------

function spread($xmin, $xmax, $Coupon, $SpotRate, $N, $DayCount, $Day1,
                $DeltaDays, $Maturity, $spread, $Price)
{
  $xacc  = 1.0e-7;
  $maxit = 30;
  $x     = $xmin;
  $fp    = BondPrice2($Coupon, $SpotRate, $N, $DayCount, $Day1, $DeltaDays,
                      $Maturity, $xmin) - $Price;
  $fc    = BondPrice2($Coupon, $SpotRate, $N, $DayCount, $Day1, $DeltaDays,
                      $Maturity, $xmax) - $Price;

  if ($fc*$fp < 0.0 || $fc == 0.0) {
    $root = rtsafe2($xmin, $xmax, $xacc, $maxit, $Coupon, $SpotRate, $N,
                    $DayCount, $Day1, $DeltaDays, $Maturity, $spread,
$Price);
  }
  return $root;
}

// --------------------------------------------------------------------------

function rtsafe($x1, $x2, $xacc, $maxit, $Coupon, $DayCount, $Day1,
                $DeltaDays, $Maturity, $Price)
{
  $ok = 0;
  f_vald($x1, &$fl, &$df, $Coupon, $DayCount, $Day1, $DeltaDays,
         $Maturity, $Price);
  f_vald($x2, &$fh, &$df, $Coupon, $DayCount, $Day1, $DeltaDays,
         $Maturity, $Price);

  if ($fl*$fh >= 0.0) return 0.0;
  if ($fl < 0.0) {  /* orient the search so that f(x) <
0.................. */
    $xl = $x1;
    $xh = $x2;
  }
  else {
    $xh   = $x1;
    $xl   = $x2;
    $swap = $fl;
    $fl   = $fh;
    $fh   = $swap;
  }
  $rts   = 0.5*($x1 + $x2);          /* initial guess of the
root............ */
  $dxold = abs($x2 - $x1);           /* the "step-size before
last.".......... */
  $dx    = $dxold;                   /* and the last
step.................... */
  f_vald($rts, &$f, &$df, $Coupon, $DayCount, $Day1, $DeltaDays,
         $Maturity, $Price);

  for ($j = 0; $j < $maxit; $j++) {
    if (((($rts - $xh)*$df - $f)*(($rts - $xl)*$df - $f) >=0.0) || /*
Bisect if N-R.*/
        (abs(2.0*$f) > abs($dxold*$df))){/* out of range or not decreasing
*/
      $dxold = $dx;                         /* fast
enough.................. */
      $dx    = 0.5*($xh - $xl);
      $rts   = $xl + $dx;
      if ($xl == $rts) {
        $ok = 1;
        return $rts;                      /* change in root is negligible..
*/
      }
    }
    else {                               /* Newton step accepable. Take it */
      $dxold = $dx;
      $dx    = $f/$df;
      $temp  = $rts;
      $rts  -= $dx;
      if (temp == $rts) {
        $ok = 1;
        return $rts;
      }
    }

    if (abs($dx) < $xacc) {
      $ok = 1;
      return $rts;                  /* Convergence criterion.......... */
    }
    f_vald($rts, &$f, &$df, $Coupon, $DayCount, $Day1, $DeltaDays,
           $Maturity, $Price);

    if ($f < 0.0) {        /* Maintain the bracket on the
root............. */
      $xl = $rts;
      $fl = $f;
    }
    else {
      $xh = $rts;
      $fh = $f;
    }
  }
  return 0.0;
}

// ------------------------------------------------------------------------

function rtsafe2($x1, $x2, $xacc, $maxit, $Coupon, $SpotRate, $N,
                 $DayCount, $Day1, $DeltaDays, $Maturity, $spread, $Price)
{
  $ok = 0;
  f_vald2($x1, &$fl, &$df, $Coupon, $SpotRate, $N,
          $DayCount, $Day1, $DeltaDays, $Maturity, $spread, $Price);
  f_vald2($x2, &$fh, &$df, $Coupon, $SpotRate, $N,
          $DayCount, $Day1, $DeltaDays, $Maturity, $spread, $Price);

  if ($fl*$fh >= 0.0) return 0.0;
  if ($fl < 0.0) {  /* orient the search so that f(x) <
0.................. */
    $xl = $x1;
    $xh = $x2;
  }
  else {
    $xh   = $x1;
    $xl   = $x2;
    $swap = $fl;
    $fl   = $fh;
    $fh   = $swap;
  }
  $rts   = 0.5*($x1 + $x2);          /* initial guess of the
root............ */
  $dxold = abs($x2 - $x1);           /* the "step-size before
last."......... */
  $dx    = $dxold;                   /* and the last
step.................... */
  f_vald2($rts, &$f, &$df, $Coupon, $SpotRate, $N,
          $DayCount, $Day1, $DeltaDays, $Maturity, $spread, $Price);

  for ($j = 0; $j < $maxit; $j++) {
    if (((($rts - $xh)*$df - $f)*(($rts - $xl)*$df - $f) >=0.0) || /*
Bisect if N-R.*/
        (abs(2.0*$f) > abs($dxold*$df))){/* out of range or not decreasing
*/
      $dxold = $dx;                         /* fast
enough.................. */
      $dx    = 0.5*($xh - $xl);
      $rts   = $xl + $dx;
      if ($xl == $rts) {
        $ok = 1;
        return $rts;                   /* change in root is negligible.. */
      }
    }
    else {                             /* Newton step accepable. Take it */
      $dxold = $dx;
      $dx    = $f/$df;
      $temp  = $rts;
      $rts  -= $dx;
      if (temp == $rts) {
        $ok = 1;
        return $rts;
      }
    }

    if (abs($dx) < $xacc) {
      $ok = 1;
      return $rts;                  /* Convergence criterion.......... */
    }
    f_vald2($rts, &$f, &$df, $Coupon, $SpotRate, $N,
            $DayCount, $Day1, $DeltaDays, $Maturity, $spread, $Price);

    if ($f < 0.0) {        /* Maintain the bracket on the
root.............. */
      $xl = $rts;
      $fl = $f;
    }
    else {
      $xh = $rts;
      $fh = $f;
    }
  }
  return 0.0;
}

?>

/*
Authors: Takoeta Fred & Hamadou Hamaounde
 Supervisor: Jan Röman
 Description: Creation of database and tables in the database
*/

// creating the database
<?php
$link = mysql_connect("localhost");
if (! $link)
  die("Couldn't connect to MySQL");

// create database

mysql_query("CREATE DATABASE BondPricing");
mysql_close($link);
?>

// Adding the table for yielddata

<?
$db="BondPricing";
$link = mysql_connect("localhost");
if (! $link)
  die("Couldn't connect to MySQL");

mysql_select_db($db , $link)
or die("Select DB Error: ".mysql_error());

//create table
mysql_query("CREATE TABLE YieldDatas( id        INT NOT NULL
AUTO_INCREMENT, PRIMARY KEY(id),
                                    market    VARCHAR(4),
                                    daycount  VARCHAR(8),
                                    name      VARCHAR(40),
                                    timestamp TIMESTAMP,
                                    coupon    FLOAT(10),
                                    cfreq     INT,
                                    maturity  VARCHAR(11),
                                    price     FLOAT(10),
                                    quotetype INT,
                                    years     FLOAT(5),
                                    ytm       FLOAT(24))")
or die("Create table Error: ".mysql_error());

mysql_close($link);
?>

// Adding the table for discountdata

<?
$db="BondPricing";
$link = mysql_connect("localhost");
if (! $link)
  die("Couldn't connect to MySQL");

mysql_select_db($db , $link)
or die("Select DB Error: ".mysql_error());

//create table
mysql_query("CREATE TABLE DiscountData( id      INT NOT NULL
AUTO_INCREMENT, PRIMARY KEY(id),

                                    market     VARCHAR(4),
                                    daycount   VARCHAR(8),
                                    name       VARCHAR(40),
                                    timestamp  TIMESTAMP,
                                    coupon     FLOAT(10),
                                    cfreq      INT,
                                    maturity   VARCHAR(11),
                                    price      FLOAT(10),
                                    quotetype  INT,
                                    time       FLOAT(10),
                                    discount   FLOAT(10),
                                    A          FLOAT(10),
                                    B          FLOAT(10))")
or die("Create table Error: ".mysql_error());

mysql_close($link);
?>
```