Yield Curve Fitting

Chia-Shen Chiang
Department of Computer Science and Information Engineering
National Taiwan University

Contents

List of Figures

List of Tables

Abstract

This thesis considers the basic yield curve fitting problem: when derive yields of current bonds with different maturity date, how can we choose(fit) a reasonable curve to indicate the yield-maturity relation.

There are a lot of curve fitting methods available. In this thesis we concern three kind of them, including least squares polynomial, cubic spline, and least squares with cubic spline constraints. The first two methods are frequently used in scientific filed and are easily understood. The third method combines the properties of the first two and is the most important part of this thesis.

When doing least squares we consider a polynomial with specified degree which formed least distance to original data points. When doing cubic spline it considers a set of polynomials each indicating the yield curve between two subsequential data points. Furthermore, every two subsequence polynomials agree on the breakpoint with the slope and curvature. In the third method of this thesis, it finds out the least distance to original data points within every cubic spline interval, which are separated by user specified breakpoints.

All these curve fitting methods use a lot of matrix calculation, which will be briefly discussed in this thesis. Although the last method may be a little more complex than the other two, it increases little calculation time. The empirical result is given in this thesis to compare the difference between these methods.

In the cubic spline, the fitting curves pass exactly through the input data points while in the other two cases the fitting curves formed a least squares distance to the input data points. So we give a discussion on how these two methods making estimating errors.

Chapter 1

Introduction

1.1 An Overview of Yield Curve

For any security, its yield to maturity can be derived from current market information such as price and coupon payment. In the other hand, if both the coupon payment and yield to maturity are specified, it is possible to find out the corresponding price inversely. The purpose of such calculation is important since there might be profit by buying low priced securities and selling high priced ones on the market. Therefore, there is always an interest to understand how these yields go with time. When plotting this relationship graphically, it is called a yield curve.

In general case this yield curve is not always available by current market information in following reasons. First, there doesn't exist all kind of securities with different time to maturity in usual, this makes the yield curve leaky of original data points. Second, while dealing with certain maturity date, there may exist more than one price of the same security. Third, the market's original information include "noises" such as the expectation of the firm's future, the coupon payment and its payment schedule, the tax effect, etc. To avoid these noise and to figure out a yield curve for further analysis usage, it is important to choose a suitable fitting method for the yield curve.

1.2 Organization of This Thesis

The main idea of this thesis is to implement three curve fitting methods, including least squares polynomial, cubic spline, and least squares polynomial with cubic spline constraint.

Since the fitted data here are bond yields, there is a brief survey on how yield and bond are defined in chapter 2.

The curve fitting methods in this thesis use a lot of matrix calculation, hence it is necessary to give a general matrix description of basic usage and further decomposition processes. This part is used by all three fitting methods in common, which comes in chapter 3. And the fitting methods mentioned above are discussed in chapter 4, including some theorems needed for understanding how these methods work.

The implementation of all these methods is in chapter 5 with some other details of how to transform original input data points as needed. The parameters of user input data are also concerned in this chapter. In the last fitting method, the decision of choosing breakpoints is left to user, so different breakpoints choosing strategy on the same data is given. The conclusion of all these fitting methods is described in chapter 6.

Chapter 2

Yield Curve Basis

2.1 Yield

For any investments, the term yield refers to the rate of return on it. There has many definition of yield, each with slightly different name and formula on how to calculate it. In this thesis, yield always refers to the internal rate of return(IRR), which is discussed below.

A bond's yield is the interest rate implied by the payment structure. Specifically, it is the interest rate at which the present value of the stream of payments(consisting of the coupon payments and the final face-value redemption payment) is exactly equal to the current price(present value, PV). This value is termed more properly the yield to maturity to distinguish it from other yield numbers that are sometimes used. Besides, yield is always quoted on an annual or semi-annual basis when using in the market.

In particular, it may said that a bond has a yield of 8.5% means this bond will return 1 plus 8.5%(totally 108.5%) percent of its principle to the investor.

Definition 2.1.1 (Internal Rate of Return(IRR)) *Suppose that a security with face value F makes m coupon payments of C/m dollars each year(in other words, the coupon payments sum to C within a year). There are n periods remaining and the face value of this security is F. Suppose also that the present value of the security is PV. Then the internal rate of return(the yield) is the value of λ such that satisfies* *the following equation:*

$$PV = \frac{F}{[1 + (\lambda/m)]^n} + \sum_{k=1}^{n} \frac{C/m}{[1 + (\lambda/m)]^k} \tag{2.1}$$

This value of λ indicates the interest rate implied by the security when interest is compounded m times per year. Note that the first term in the right of 2.1 is the present value of the face-value payment. The kth term in the summation is the present value of the kth coupon payment C/m. The sum of the present values, based on a nominal interest rate of λ, is set equal to the bond's present value.

2.2 Bond

A bond is an obligation by the bond issuer to pay money to the bond holder according to rules specified when the bond is issued. Generally, a bond pays a specific amount called face value(or par value) at the maturity date(or redemption date). It is usually $1,000 in the US. In addition, most bonds pay periodic *coupon* payments. The last coupon date corresponds to the maturity date, so the last payment is equal to the face value plus the coupon value.

A bond's price is calculated by

$$P = \frac{F}{[1 + (r/m)]^n} + \sum_{k=1}^{n} \frac{Fc/m}{[1 + (r/m)]^k}, \tag{2.2}$$

where c is the coupon rate, m is the number of payments per year, n is the total periods of coupon payment, and r is the annual interest rate compounded m times per annum. Compare with equation 2.1 , when p, c, m, n are known, the annual interest rate r is simply the yield of this bond.

There are several ways to redeem or retire a bond. A bond is redeemed at maturity if the principal is repaid at maturity. Most corporate bond are *callable*, meaning the issuer can redeem some or all of the bonds before the stated maturity, usually with a price above the par value. Since this provision gives the issuer the advantage of calling a bond when the prevailing interest is much lower than the coupon rate, the bondholders usually demand premium for them. A callable bond may also have call protection so that it is not callable for the first few years.

Although bonds offer a supposedly fixed cash in-stream, they are subject to default if the issuer has financial difficulties or falls into bankruptcy. To characterize the nature of this risk, bonds are classified by rating organizations like Moody's. It is important to choose the same quality bonds when measuring term structure. Though there exist two bonds with totally the same terms when they are issued, the price(as well as their yield to maturity) would differ with differing rating.

A pure discount bond(also zero-coupon bond or simply *zero*) promises a single payment in the future and is sold at a discount from its par value. No interest is paid with such kind of bond. The price of a zero-coupon bond that pays F in n periods is simply: PV $= \frac{f}{(1+r)^n}$, where r is the market interest rate of the required yield per period. The required yield is base on the yields on comparable investments in the market. Such bonds are important in practice since they can be used to meet future obligations without worrying the reinvestment. They are also important for their usage in the analysis of coupon-bearing bonds as they can be thought of a series of zero-coupon bonds.

Day Count Convention

When valuing bond's price(or the yield to maturity), the consideration of day count convention is needed. In the so called "actual/actual" day count convention, the first "actual" refers to the actual number of days in a month, and the second refers to the actual number of days in a coupon period. A convention popular with money market is "30/360." Here, each month is assumed to have 30 days and each year have 360 days. The number of days between two given dates under the "30/360" convention can be computed by $360 \times (y_2 - y_1) + 30 \times (m_2 - m_1) + (d_2 - d_1)$, where y_i denote the years, m_i denote the months, d_i denote the days.

In bond market, it always takes "actual/365" day count convention, which means there has 365 days per year. Whenever there is day interval shorter than one year, it converts to year measuring. For example, 48 days converted to $48/365 = 13.282$(year), and is used to calculate as before.

2.3 Yield Curve and Term Structure

Term structure is a set of yields to maturity for different bonds with the same quality. This term refers to the yields of zero-coupon bonds sometimes. Yield curve is the plotting of the yield to maturity against maturity, what represents the prevailing interest rates for various maturity dates.

Term structure of interest rates is the relationship between yield and maturity on securities differing only in length of time to maturity. When sketched graphically as yield-maturity figure, it is known as a yield curve. This curve is important and fundamental in finance market because of the ability to predict interest rate tendency and to measure other derivative securities' prices.

A normal yield curve is upward sloping , an inverted yield curve is downward sloping, and a flat yield curve is flat. In a humped yield curve, the yield is upward sloping at first but then turns downward sloping. A simple example is shown in figure

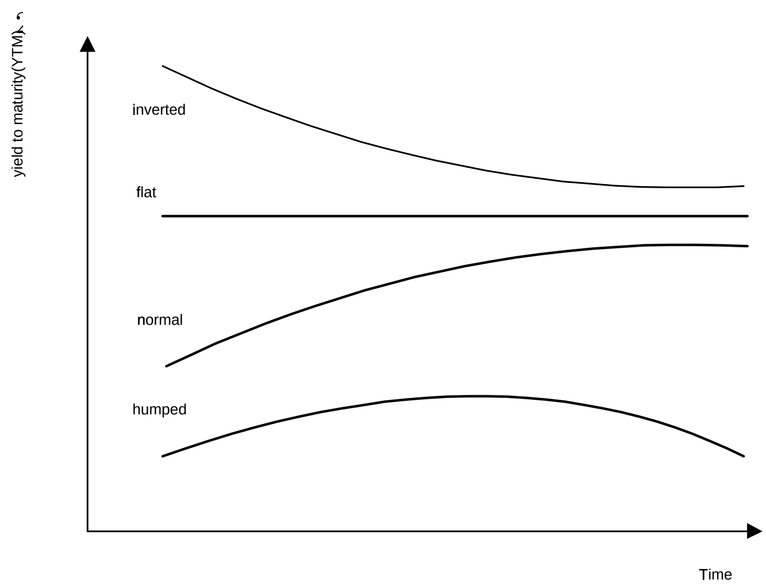

Figure 2.1: TERM STRUCTURE EXAMPLE.

Chapter 3

Matrix Algebra

Matrices provide convenient tools for scientific calculations by involving a compact notation for data storing and describing complicated relationships. This chapter reviews basic matrix notation and computations, which are heavily used in this thesis.

3.1 Basic Definition and Results

A m $\times$ n matrix A is a rectangular array of m $\times$ n numbers(though it may be real or complex numbers, we only care about the case of real numbers in this thesis) enclosed in square brackets. It can be presented as A$\in$ $R^{m\times n}$. The numbers in the matrix are called entries. We use $\langle A\rangle_{ij}$ to indicate the entry located at the ith row of the jth column. A matrix may also be presented as $[a_1, a_2, ..., a_n]$ where $a_i \in R^m$ are (column)vectors. A general m $\times$ n matrix is written as:

$$A_{m\times n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

For a m $\times$ n matrix A, the transpose(denoted by A^T) of A is the n $\times$ m matrix obtained by interchanging the rows and columns of A–the first row becomes the first column, and so on. That is, $\langle A^T\rangle_{ij} = \langle A\rangle_{ji}$, for $1 \leq i \leq m$ and $1 \leq j \leq n$. A square matrix Q is said to be symmetric if $QT =$ Q.

A matrix is lower-triangular if it satisfies $A_{ij} = 0$ if i <j, for $1 \leq i \leq m$ and $1 \leq j \leq n$. Alternatively, it is upper-triangular if it satisfies A_{ij}= 0 if i >j, for

$1 \leq i \leq m$ and $1 \leq j \leq n$.

A diagonal matrix $D_{m \times n}$ is denoted by diag($d_1, ..., d_k$), where k = min(m, n), D_{ii} = d_i, and all other entries in D are 0. The n × n identity(or unit) matrix is the diagonal matrix I_n=diag$\overbrace{(1, ..., 1)}^{n}$.

A matrix is said to be partitioned when there has dashed vertical lines of the full height of the matrix drawn between selected columns or dashed horizontal lines of the full width of the matrix drawn between selected rows. The smaller matrices formed by the original matrix is called submatrices. A leading principal submatrix of an n × n matrix is a submatrix consisting of the first k rows and the first k columns for some 1≤ k ≤ n.

For any matrix A, if there exists a matrix L that satisfies LA = I, then L is called a left-inverse of A. Alternatively, if there exists a matrix R that satisfies AR = I, then R is called a right-inverse of A. For any matrix X which is both left- and right-inverse of A is called an inverse of A.

A nonsingular matrix is a (necessarily square) matrix A that possesses an inverse X: For nonsingular A, there exists an X with AX = XA = I. This inverse is denoted by A^{-1}. A singular matrix is a square matrix that does *not* possess an inverse.

Let A be n × n. The (i,j)-minor of A, denoted M_{ij}, is the determinant of the (n-1) × (n-1) matrix formed by deleting the ith row and jth column from A. The (i,j)-cofactor of A, denoted A_{ij}, is $(-1)^{i+j} M_{ij}$. Note that the signs$(-1)^{i+j}$ in the definition of cofactor form a checkerboard pattern:

$$\begin{bmatrix} + & - & + & \cdots \\ - & + & - & \cdots \\ + & - & + & \cdots \\ \vdots & & & \vdots \end{bmatrix},$$

Now we can define the determinants:

(a) The determinant of the 1 × 1 matrix $[\alpha]$ is defined as det$[\alpha] = \alpha$.

(b) The determinant of the n × n matrix A is defined as

$$detA = \sum_{j=1}^{n} \langle A \rangle_{1j} A_{1j}. \tag{3.1}$$

In words: The determinant of A is the sum of the products of the entries of the first row and the cofactors of the first row.

Theorem 3.1.1 (determinants and nonsingularity)

(a) *A is nonsingular if and only if det A $\neq$ 0; equivalently, A is singular if and only if det A = 0.*

(b) *If A is nonsingular, then* $det(A^{-1}) = 1/det(A)$.

Given two vectors u and v, the inner product denoted by (u,v) is defined as $u^T v = v^T u$. If (u,v) = 0, these two vectors are called orthogonal to each others. A set of vectors are said to be orthogonal if and only if every two vectors from the set are orthogonal. A set of vectors are said to be orthonormal if and only if the set is orthogonal and every vector v_i in this set satisfies (v,v) = 1.

A real square matrix Q is said to be orthogonal if $Q^T Q =$ I. An immediate consequence is $Q^{-1} = Q^T$ and $QQ^T =$ I. For any matrix that has the property $A^T = A^{-1}$ is said to be a unitary matrix.

Theorem 3.1.2 (Matrix Norm) *Let A be $m \times n$. Then:*

(a) *The 1-norm of matrix A is*

$$\| A \|_1 = \max_j \sum_{i=1}^{m} | a_{ij} | \text{ (maximum absolute column sum)}. \tag{3.2}$$

(b) *The ∞-norm of matrix A is*

$$\| A \|_\infty = \max_i \sum_{j=1}^{n} | a_{ij} | \text{ (maximum absolute row sum)}. \tag{3.3}$$

(c) *The 2-norm of matrix A is*

$$\| A \|_1 = \text{(maximum eigenvalue of } A^T A)^{1/2} = \text{maximum singular value of } A. \tag{3.4}$$

3.2 Gauss Elimination and LU Decomposition

3.2.1 Gauss Elimination

Consider the systems of linear equations:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

where x_i are unknowns. This can be written in matrix form as: Ax = b. Where

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}.$$

And the partitioned matrix [A b] contained A and b is called *augmentedmatrix*.

Gauss Elimination

Gauss elimination proceeds to eliminate the augmented matrix in the columns starting with column 1, then column 2, and so on; we use row r_j to eliminate in column j, and may well interchange a lower row with the r_jth row before elimination. We can describe the process using successive columns as follows:

Gauss elimination with interchanges on the m × (n+1) augmented matrix [A b] proceeds as follows:

1. Set j=1 and $r_1 = 1$ and use row r$_j$ to eliminate in column j of the present augmented matrix as prescribed in steps 2 through 6.

2. Select one of the rows from among those numbered r_j,$r_j + 1$,..., m for use in eliminating in column j; call this row i, so that the (i,j)-entry–called the pivot–in the present augmented matrix must be nonzero. If there has no nonzero entries in this lower portion of column j, then no elimination is necessary; set $r_{j+1} = r_j$ to use the same row, and skip to step 6.

3. Interchange the ith and r_jth rows.

4. Replace this new r_jth row by itself divided by the pivot(its nonzero entry in its jth column).

5. Use this newest r_jth row to eliminate the entries in the jth column in the rows $r_j + 1$, $r_j + 2$,...,m. Set $r_{j+1} = r_j$+1 to use the next row.

6. (a) If j$\leq$ q and $r_{j+1} \leq$ m, then further elimination is still possible: increase j by 1 and return to step 2. (b)Otherwise, Gauss elimination has been completed: go to step 7.

7. Interpret the final reduced matrix as the augmented matrix for a system of equations and proceed to find the solutions, if any, by back-substitution.

The number of nonzero rows in any Gauss-reduced form of a matrix A is called the rank of A.

3.2.2 LU Decomposition

Suppose that A is n $\times$ n. Then:

(a) A can be reduced by Gauss elimination with interchanges in the usual way to a reduced unit-upper-triangular U if and only if A is nonsingular. When such U is produced, there exists a permutation matrix P and a nonsingular lower-triangular matrix L such that PA = LU, and A = $P^T LU$,

(b) There exist a permutation matrix P, an unit-lower-triangular matrix L_0, and an upper-triangular matrix U_0 such that PA = $L_0 U_0$, and A = $P^T L_0 U_0$

A is nonsingular if and only if U is nonsingular; for nonsingular A, this L_0 and U_0 relate to the L and U of (a) by $U = D_0^{-1} U_0$ and L = $L_0 D_0$, where D_0 = diag(α_{11},..., α_{pp}) and $\alpha_{ii} = < U_0 >_{ii} = < L >_{ii}$.

When solving equation Ax = b, of course, requires that b be processed also and that back-substitution be performed; this is easily expressed in terms of the LU-decomposition A = P^TLU as described above. equation Ax = b means that P^TLUx

= b, so LUx = Pb; we write this as Ly = Pb and Ux = y, each of which is simply a triangular system that can be solved immediately by forward- or back-substitution.

To solve Ax = b, given the LU-decomposition A = P^TLU:

- Permute the entries of b to obtain b′ = Pb.
- Solve the lower-triangular system Ly = b′ for y by forward-substitution.
- Solve the unit-triangular system Ux = y for x by back-substitution.

3.3 Creating Orthogonal Set

When solving least squares problem, it has to approximate a given vector as a linear combination of other given vectors. If the vectors from which one forms linear combinations are mutually orthogonal, then the solution of this problem could be solved easily.

Theorem 3.3.1 *Let $\bigvee$ be a vector space with an inner product, and let $\bigvee_0$ be the subspace spanned by the orthogonal set of nonzero vectors v_1,..., v_q. Then, for any v, P_0v is the unique closest point in $\bigvee_0$ to v and $\|v - P_0v\|$ is the distance from v to $\bigvee_0$, in the sense that P_0v is in $\bigvee_0$ and $\|v - P_0v\| < \|v - v_0\|$ for all $v_0 \neq P_0vin\bigvee_0$.*

Such orthogonal set can be created by different ways. In this thesis it uses Gram-Schmidt and Householder processes described below.

3.3.1 Gram-Schmidt Process

Giving a vector set v_1,...,v_m span the vector space $\bigvee$ with inner product:

1. Define u_1 =v_1.
2. For 2≤ i ≤ n, define $u_i = v_i - \alpha_{1i}u_1 - ... - \alpha_{i-1,i}u_{i-1}$, where $\alpha_{ij} = (\frac{u_jv_i}{u_ju_j})$ if $u_j \neq 0$ and $\alpha_{ji} = 0$ if $u_j = 0$.

The following holds for the vectors produced by this process:

1. B = $\{u_1, ..., u_m\}$ is an orthogonal set.

2. $u_i = 0$ if and only if v_i is linearly dependent on the vectors $v_1, \ldots, v_{i-1}$.

3. An orthogonal basis for V can be obtained from the orthogonal spanning set B by omitting the zero u_i, if any.

3.3.2 Householder Reflection

For nonzero w in R^n , the n $\times$ n Householder matrix H_w is defined as $H_w = I_n - (\frac{2}{w^T w}) w w^T$. Then:

(a) H_w is symmetric and orthogonal.

(b) For each x, $H_w x$ equals the reflection of x about the subspace of all v orthogonal to w.

(c) $\det(H_w) =$ -1.

(d) For any nonzero x and nonzero y in R^p with x $\neq$ y, there exists an H_w such that $H_w x = \alpha$ y for some real number α. More precisely:

1. Unless x equals a positive multiple of y, w can be taken as $w = x - (\frac{||x||_2}{||y||_2})y$, and then $H_w x = (\frac{||x||_2}{||y||_2})y$
2. Unless x equals a negative multiple of y, w can be taken as $w = x + (\frac{||x||_2}{||y||_2})y$, and then $H_w x = -(\frac{||x||_2}{||y||_2})y$

The term (d) above is heavily used in the application of Householder matrices, especially to transform a given x to a multiple of e_1:$H_w x = ||x||_2 e_1$. The Householder matrix in effect "zeros out" the entries below the first in x.

3.4 QR Decomposition

After creating Householder reflection of a matrix, the matrix can be decomposed by another method called QR decomposition.

(a) Suppose that A is m $\times$ n and real. A sequence $H_1, ..., H_n$ of at most n Householder matrices can be easily computed, so that $H_n H_{n-1} ... H_1 A = R$, where R is upper-triangular and has nonnegative entries on the main diagonal; equivalently,

$$A = QR, \tag{3.5}$$

where Q $= H_1 H_2 ... H_n$ is m $\times$ n and orthogonal.

(b) Every m $\times$ m orthogonal matrix Q can be written as the product Q $= H_1 H_2 ... H_m$ of at most m Householder matrices.

(c) A can be written in its normalized QR decomposition as A $=$ Q_0R_0, where Q_0 is a orthonormal matrix such that $Q_0^{-1} = Q_0^T$ and R_0 a upper-triangular matrix.

This decomposition is analogous to the LU decomposition which can be used to find out the solutions of a set of equations.

Pseudoinverse

Suppose that A $=$ QR is a normalized QR decomposition of the m $\times$ n matrix A. Then the term $A^+ = R^T(RR^T)^{-1}Q^T$ is called the pseudoinverse of A.

Note that when solving the least squares problem Ax $\cong$ b with the normalized QR decomposition, the solution $\tilde{x}$ can be written as $\tilde{x}$ $=$ A^+b.

Chapter 4

Curve Fitting Methods

Interpolation, the computing of values for a tabulated function at points not in the table, is historically a most important task. Many famous mathematicians have their names associated with procedures for interpolation, but in this chapter we only survey the three topics used in this thesis.

4.1 Polynomial Interpolation with Least Squares Problem

Suppose we have the following set of m data points(need not be equally spaced):

(x_i, y_i), i = 1, 2,..., m.

For the purpose of fitting a curve to an approximate set of data, such as the yield-maturity pairs in this thesis. Suppose this curve is a polynomial of degree n. We want to suitably determine the constants a_i in the equation relating yield y and maturity x such that

$$y = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0, \tag{4.1}$$

with n+1 unknowns, so that this curve performs minimize distance to each data pair. This criterion comes from the mathematically way of minimizing the sum of the error squares(SES).

Obviously, the relationship m $\geq$ n must be valid to promise the solution polynomial is unique. While m = n + 1, the solution polynomial passes through each input data.

In matrix form, the least squares problem is written as $min\|Ax - b\|$, where A$\in$R$^{m\times n}$, b$\in$R^m, and x$\in$R^n are shown below

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & & & \ddots & \vdots \\ 1 & a_m & x_m^2 & \cdots & x_m^n \end{bmatrix}, x = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}, b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

The least squares problem can be approached geometrically. Observe that Ax is a linear combination of A's columns with coefficients x$_1$,...,x$_n$. The least squares problem finds the minimum distance between b and A's column space. Furthermore, the solution x$_{LS}$ identifies the point Ax$_{LS}$ which is at least as close to b as any other point in the column space. The error vector Ax$_{LS}$ - b must therefore be perpendicular to that space, that is

$$(Ay)^T(Ax_{LS} - b) = y^T(A^TAx_{LS} - A^Tb) = \mathbf{0}$$

for all y. Therefore, the solution must satisfy the normal equations

$$A^TAx = A^Tb.$$

Consider the QR decomposition discussed in last chapter, suppose A = QR, then Ax = b becomes Rx = bQ^T, which is solvable by straight substitution.

4.2 Cubic Spline

Suppose we have the following set of m data points(need not be equally spaced):

(x_i, y_i), i = 1, 2,..., m. So there exists totally m-1 intervals separated by the data points.

A spline curve can be of varying degrees when doing curve fitting. In general, we fit a set of nth-degree polynomials g$_i$(x) between intervals of [x$_i$, x$_{i+1}$], for i = 1, 2,..., m-1. Each x$_i$ here is called a breakpoint or knot. If the degree of the spline is one(just straight lines between the points), the "curve" would appear in accompanying figure 4.1. The problem with this linear spline is that the slope is discontinuous at the breakpoints. Spline of degree greater than one can avoid such problem. If both slope and curvature are asked to be continuous at breakpoints , the spline curve must use polynomials of degree three(or more) to match this behavior. Such spline interpolation is called cubic spline and is used in this thesis.

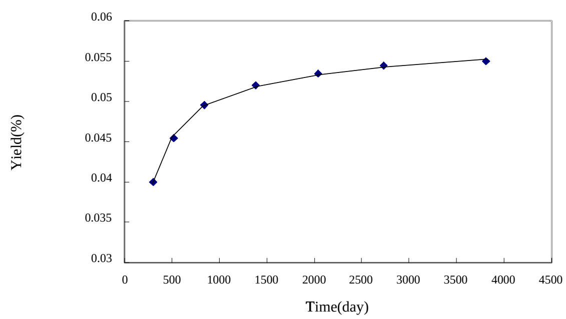

Figure 4.1: Spline interpolation with degree of one

When writing down the equation for a cubic spline of the ith interval, which lies between the points (x_i,y_i) and (x_{i+1},y_{i+1}). It looks like this:

$$g_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i. \tag{4.2}$$

Thus the cubic spline function wanted is of the form

$g(x) = g_i(x)$ on the interval $[x_i, x_{i+1}]$, for i = 1, 2,..., m-1 and meets the following conditions:

$$g_i(x_i) = y_i,\ \text{i = 1, 2,..., m-1 and } g_{n-1}(x_n) = y_n;\ \text{fits to input data} \tag{4.3}$$

$$g_i(x_{i+1}) = g_{i+1}(x_{i+1}),\ \text{for i = 1, 2,..., m-1 ; continuous at breakpoint} \tag{4.4}$$

$$g_i'(x_{i+1}) = g_{i+1}'(x_{i+1}),\ \text{for i = 1, 2,..., m-1 ; slope is continuous at breakpoint} \tag{4.5}$$

$$g_i''(x_{i+1}) = g_{i+1}''(x_{i+1}),\ \text{for i = 1, 2,..., m-1 ; curvature is continuous at breakpoint} \tag{4.6}$$

Equation 4.6 above immediately gives

$d_i = y_i,$ i = 1, 2,..., m-1.

To relate the slopes and curvatures of the joining splines, we differentiate equation 4.2.

$$g_i'(x) = 3a_i h_i^2 + 2b_i h_i + c_i, \tag{4.7}$$

$$g_i''(x) = 6a_i h_i + 2b_i, \text{ for i = 1, 2,..., m-1.} \tag{4.8}$$

where $h_i = x_{i+1} - x_i$.

Development is simplified by writing the equations in terms of the second derivatives– that is, set $S_i = g_i''(x_i)$ for i = 1, 2,..., m-1 and $S_m = g_{m-1}''(x_m)$. From equation above we have

$S_i = 6a_i(x_i - x_i) + 2b_i = 2b_i;$
$S_{i+1} = 6a_i(x_{i+1} - x_i) + 2b_i = 6a_i h_i + 2b_i.$

hence

$a_i = \frac{S_{i+1} - S_i}{6h_i}, \quad b_i = \frac{S_i}{2}.$

Then substitute the relations for a_i, b_i, d_i given above and solve for c_i,

$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6}.$

Now invoke the condition that the slopes of the two cubics that join at(x_i, y_i) are the same. For the equation in the ith interval, equation 4.5 becomes, with $x=x_i$,

$y_i' = 3a_i(x_i - x_i)^2 + 2b_i(x_i - x_i) + c_i = c_i.$

In the previous interval, from $x_{i=1}$ to x_i, the slope at its right end will be

$y_i' = 3a_{i-1}(x_i - x_{i-1})^2 + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1} = 3a_{i-1}h_{i-1}^2 + 2b_{i-1}h_{i-1} + c_i.$

Equating these, and substituting for a, b, c, d of their relationships in terms of S and y, we get

$$\begin{aligned} y_i' &= \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6} \\ &= 3(\frac{S_i - S_{i-1}}{6h_{i-1}})h_{i-1}^2 + 2(\frac{S_{i-1}}{2})h_{i-1} + \frac{y_i - y_{i-1}}{h_{i-1}} - \frac{2h_{i-1}S_{i-1} + h_{i-1}S_i}{6}. \end{aligned}$$

Simplifying this equation, we get

$$h_{i-1}S_{i-1} + (2h_{i-1} + 2h_i)S_i + h_iS_{i+1} = 6(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}). \tag{4.9}$$

Consider all equations in every interval, this lead to 4(m-1) unknowns, which are the (a_i, b_i, c_i, d_i) for i = 1, 2,..., m-1. But there exists only 4(m-1)-2 equations, so additional two constraints are needed. Many alternative choices are discussed, in this thesis only the natural spline condition is taken which assigns S_1 and S''_m both equaling 0.

When writing equations above in matrix form, by looking them as Ax = b, the corresponding A, x, and B are listed below. Remember that the unknowns here are not (x_i, y_i), but S_i.

$$A = \begin{bmatrix} 2(h_1+h_2) & h_2 & & & \\ h_2 & 2(h_2+h_3) & h_3 & & \mathbf{0} \\ & h_3 & 2(h_3+h_4) & h_4 & \\ \mathbf{0} & & & \ddots & \\ & & & h_{m-2} & 2(h_{m-2}+h_{m-1}) \end{bmatrix},$$

$$x = \begin{bmatrix} S_2 \\ S_3 \\ \vdots \\ S_{m-1} \end{bmatrix}, b = 6\begin{bmatrix} \frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_1} \\ \frac{y_4-y_3}{h_3} - \frac{y_3-y_2}{h_2} \\ \vdots \\ \frac{y_m-y_{m-1}}{h_{m-1}} - \frac{y_{m-1}-y_{m-2}}{h_{m-2}} \end{bmatrix}.$$

After the S_i values are obtained, we get the coefficients a_i, b_i, c_i, and d_i for the cubic spline functions in each interval as

$a_i = \frac{S_{i+1}-S_i}{6h_i}$;

$b_i = \frac{S_i}{2}$;

$c_i = \frac{y_{i+1}-y_i}{h_i} - \frac{2h_iS_i+h_iS_{i+1}}{6}$;

$d_i = y_i$.

From these we can compute points on each of the interpolating interval.

4.3 Least Square Polynomial with Cubic Spline Constraints

This thesis consults a new method on yield curve fitting. Consider that there are more data points than breakpoints(which are assigned by user), we may be asked to find a cubic spline with breakpoints that minimizes the distance to the data. This involves a new problem that solving a set of linear equations with constraints given by another set of linear equations.

Problem Description

Suppose we have the following set of m data points(need not be equally spaced) as previous sections

(x_i, y_i), i = 1, 2,..., m.

Unlike the condition in last section, the breakpoints β are not the same as input data points. Instead, they are assigned by user, say β_1, β_2,... , β_{n-1}(and the interval number is n).

In applications where there are more data points than breakpoints, it may be asked to find a set of cubic spline with breakpoints β_i that minimizes the distance to the data. Let the cubic polynomial for the interval [b_{i-1}, b_i] be written as

$g_i(x) = a_i + b_i x + c_i x^2 + d_i x^3$ for i = 1, 2,..., n-1.

and

for i = 0, g_0 in interval [x_1, β_1],

for i = n, g_n in interval [β_{n-1}, x_m].

Since cubic spline is defined twice continuously differentiable, so

$g_i(x_i) = g_{i+1}(x_i)$,

$g_i'(x_i) = g_{i+1}'(x_i)$,

$g_i''(x_i) = g_{i+1}''(x_i)$, for i = 0, 1,..., n-1.

which imply

$a_i + b_i\beta_i + c_i\beta_i^2 + d_i\beta_i^3 = a_{i+1} + b_{i+1}\beta_i + c_{i+1}\beta_i^2 + d_{i+1}\beta_i^3$

$b_i + 2c_i\beta_i + 3d_i\beta_i^2 = b_{i+1} + 2c_{i+1}\beta_i + 3d_{i+1}\beta_i^2$

$2c_i + 6d_i\beta_i = 2c_{i+1} + 6d_{i+1}x_i$

The above equations can be written as Bx = d for some B$\in \mathbf{R}^{3n\times 4(n+1)}$ and d$\in \mathbf{R}^{3n\times 1}$. Assuming B's rank is 3n. Indeed the constraint on B's full rank property is not necessary, but in this thesis it always holds. The matrix B and d are given below,

$B_{3(n-1)\times 4n}$ =

$$\begin{bmatrix}
1 & \beta_0 & \beta_0^2 & \beta_0^3 & -1 & -\beta_0 & -\beta_0^2 & -\beta_0^3 & & & & & & & \\
0 & 1 & 2\beta_0 & 3\beta_0^2 & 0 & -1 & -2\beta_0 & -3\beta_0^2 & & & & & & \mathbf{0} & \\
0 & 0 & 2 & 6\beta_0 & 0 & 0 & -2 & -6\beta_0 & & & & & & & \\
 & & & & 1 & \beta_1 & \beta_1^2 & \beta_1^3 & -1 & -\beta_1 & -\beta_1^2 & -\beta_1^3 & & & \\
 & & & & 0 & 1 & 2\beta_1 & 3\beta_1^2 & 0 & -1 & -2\beta_1 & -3\beta_1^2 & & & \\
 & & & & 0 & 0 & 2 & 6\beta_1 & 0 & 0 & -2 & -6\beta_1 & & & \\
 & & \mathbf{0} & & & & & \vdots & & & & & & & \\
 & & & & & & & 1 & \beta_{n-1} & \beta_{n-1}^2 & \beta_{n-1}^3 & -1 & -\beta_{n-1} & -\beta_{n-1}^2 & -\beta_{n-1}^3 \\
 & & & & & & 0 & 1 & 2\beta_{n-1} & 3\beta_{n-1}^2 & 0 & -1 & -2\beta_{n-1} & -3\beta_{n-1}^2 & \\
 & & & & & & 0 & 0 & 2 & 6\beta_{n-1} & 0 & 0 & -2 & -6\beta_{n-1} &
\end{bmatrix},$$

$$d_{3(n-1)\times 1} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix},$$

Consider the data set (x_1, y_1), (x_2, y_2),...,(x_m, y_m). Suppose x_j falls within interval [β_{i-1}, β_i]. Then the least squares formula is

$g_i(x_j) = a_i + b_i x_j + c_i x_j^2 + d_i x_j^3 \cong y_j.$

Writing in matrix form becomes

$$A_{m\times 4n)} = \begin{bmatrix}
1 & x_1 & x_1^2 & x_1^3 & & & & & & & & & \\
 & & & & \ddots & & & & & & \mathbf{0} & & \\
 & & & & 1 & x_i & x_i^2 & x_i^3 & & & & & \\
 & \mathbf{0} & & & & & & & \ddots & & & & \\
 & & & & & & & & 1 & x_m & x_m^2 & x_m^3 &
\end{bmatrix}.$$

for the jth row of A, there exists 4i 0s before non-zero numbers if this input data falls in the ith interval. That is,

The jth row of A $= [\underbrace{0, \cdots, 0}_{4i}, 1, x_j, x_j^2, x_j^3, 0, \cdots, 0].$

and

$$b_{m\times 1} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

So there has the constrained least squares problem

$min_{Bx=d}\|Ax - b\|.$

Theoretical Discussion

Consider the matrices given above, there exists unique solution if and only if the augmented matrix $\begin{bmatrix} A \\ B \end{bmatrix}$ is of rank 4n. Remember the assumption that matrix B has rank of 3(n-1), means $m + 3(n-1) \geq 4n$, so that $m \geq n+3$.

Algorithms for such problems are generally transforming to problems into unconstrained least squares ones. The basic idea is due to Lawson and Hanson(1974). This method can be interpreted as the following three stages:

1. Derive a lower-dimensional unconstrained least squares problem from the given problem.

2. Solve the derived problem

3. Transform the solution of the derived problem to obtain the solution of the original constrained problem.

<Theorem 4.1>

For problem min $\|Ax = b\|$, suppose that A is an m $\times$ n matrix of rank k and that $A = HRK^T$ where R is a m $\times$ n matrix of the form

$$R = \begin{bmatrix} R_{11} & 0 \\ 0 & 0 \end{bmatrix},$$

with R_{11} a k $\times$ k matrix of rank k, H is a m$\times$ m orthogonal matrix, and K is a n$\times$ n orthogonal matrix.
Define the vector

$$H^T b = g \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \begin{matrix} \}k \\ \}n-k \end{matrix} \quad and \quad K^T x = y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \begin{matrix} \}k \\ \}n-k \end{matrix}$$

Define $\tilde{y}_1$ to be the unique solution of $R_{11}y_1 = g_1$. Then all solutions to the problem of minimizing $\|Ax - b\|$ are of the form

$$\hat{x} = K \begin{bmatrix} \tilde{y}_1 \\ y_2 \end{bmatrix},$$

where x_2 is arbitrary.

proof:

An important property of orthogonal matrices is the preservation of euclidean length under multiplication. From this, and replace A with HRK^T, the problem becomes

$\|Ax - b\|^2 = \|HRK^T x - b\|^2 = \|RK^T x - H^T b\|^2$

From the matrices defined above we have

$\|Ax - b\|^2 = \|R_{11}y_1 - g_1\|^2 + \|g_2\|^2$

So this problem has minimum length when
$R_{11}y_1 = g_1$, with residual $\|g_2\|^2 = H\begin{bmatrix} 0 \\ g_2 \end{bmatrix}$

<Theorem 4.2>
Consider problem of min$\|Ax - b\|$ with constraints $Bx = d$.
Suppose $B^T = HRK^T$ with rank k, and partition K as

$$K = [\underbrace{K_1}_{k}, \underbrace{K_2}_{(n-k)}]\}n.$$

The problem has a unique minimal length solution given by
$\hat{x} = B^+d + K_2(AK_2)^+(b - AB^+d)$ if and only if the augmented matrix $\begin{bmatrix} A \\ B \end{bmatrix}$ is of rank n.

proof:
From theorem I, the solution set X may be represented as
$X = \{x: x = \tilde{x} + K_2y_2\}$,
where $\tilde{x} = C^+d$.

So the solution is equivalent to finding a (n-k)-vector y_2 that minimizes
$\|A(\tilde{x} + K_2y_2) - b\|$
or equivalently
$\|(AK_2)y_2 - (b - A\tilde{x})\|$

Then the unique minimal length solution vector for this problem is given by
$\hat{y_2} = (AK_2^T)^+(b - A\tilde{x})$

Thus the solution vector for this problem is given by
$\hat{x} = \tilde{x} + K_2\hat{y}_2$
which is equivalent to
$\hat{x} = B^+d + K_2(AK_2)^+(b - AB^+d)$

QR Implementation

In practice we use QR decomposition to implement this method. Before that, we must make sure that the first 3(n-1) columns in matrix B are linear independent. This is done by Gram-Schmidt procedure, where the algorithm is shown in the previous chapter. After finding the linear independent columns, the are swapped into the first 3(n-1) columns. Remember swapping the corresponding rows in the x matrix. Consider the matrix below, if the 2nd, 3rd, and 5th column in A are swapped, the corresponding rows in x must be swapped, too. And after solving x, remember to re-swap the rows of x into original sequence.

$$\begin{bmatrix} & \downarrow & \downarrow & & \downarrow \\ a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \end{bmatrix} \begin{bmatrix} x_1 & \\ x_2 & \leftarrow \\ x_3 & \leftarrow \\ x_4 & \\ x_5 & \leftarrow \end{bmatrix}$$

For the purpose of constructing an orthogonal decomposition as in theorem 4.1, we take normalized QR decomposition of B^T, thus

$B^T = QR.$

so that

$B = (B^T)^T = (QR)^T = R^T Q^T,$

where R^T is a lower-triangular matrix, Q_T is orthonormal and can be partitioned as

$Q = [\underbrace{Q_1}_{3(n-1)}, \underbrace{Q_2}_{4n-3(n-1)}]\}4n.$

So that matrix B can be decomposed as in theorem 4.1 as

$B = HRK^T = IR^TQ^T$, where I is an identity matrix.

According to the theorem above, the augmented matrix can be written as

$$\begin{bmatrix} B \\ A \end{bmatrix} Q = \begin{bmatrix} R^T & 0 \\ \underbrace{\tilde{A}_1}_{3(n-1)} & \underbrace{\tilde{A}_2}_{4n-3(n-1)} \end{bmatrix} \begin{matrix} \}3(n-1) \\ \}m \end{matrix}$$

where R^T is a 3(n-1)× 3(n-1) lower triangular and nonsingular matrix. Then

1. Solve the lower triangular system $R^T y_1 = d$. Since $d = \mathbf{0}$ in the case here, the solution $\widehat{y_1}$ must be $\mathbf{0}$.

2. Solve $\tilde{A}_2 y_2 \cong b - \tilde{A}_1 \widehat{y_1} = b$ for the (4(n+1)-3n)-vector $\widehat{y_2}$.

3. compute the solution vector

$$\tilde{x} = Q \begin{bmatrix} \widehat{y_1} \\ \widehat{y_2} \end{bmatrix}$$

The numerical result is given in the next chapter.

Chapter 5

Implementation and Empirical Result

There are three kinds of fitting methods implemented in this thesis, which are least squares polynomial, cubic spline, and least squares polynomial with cubic spline constraint. By inviting the formatted input row data, each method tries to find out a mathematical reasonable yield curve. The basic concepts and calculation methods are presented in previous chapters, here the implementation procedures are discussed.

5.1 On Data Input

This thesis focus on curve fitting method, in which the input data pairs are not necessary to be huge. Consider Taiwan's bonds, the price is quoted on yield, in that the input conversion from price and coupon payment to yield is no longer existent. The original data in this thesis are part of Taiwan's government bond price on 1999/4/15 from the Grand Cathay Securities Corp., which are listed in table 5.1.

For the sake of easily assigning breakpoints and observing, the input maturity date are converted in day measurement, which is different with market convention(in

Table 5.1: The Input Data Yield

Bond ID	Maturity Date	Yield	Bond ID	Maturity Date	Yield
000824	19/2/2000	4.950	00842	16/12/2005	5.400
00064	25/3/2001	5.000	00751	19/1/2006	5.850
00341	17/3/2002	5.100	86101	24/9/2006	5.900
00853	22/12/2002	5.350	87102	21/11/2007	5.285
00831	22/9/2003	5.550	88101	25/9/2008	5.690
00833	18/1/2004	5.550	00551	21/7/2010	5.920
00841	18/11/2004	5.450	00955	22/3/2011	5.920

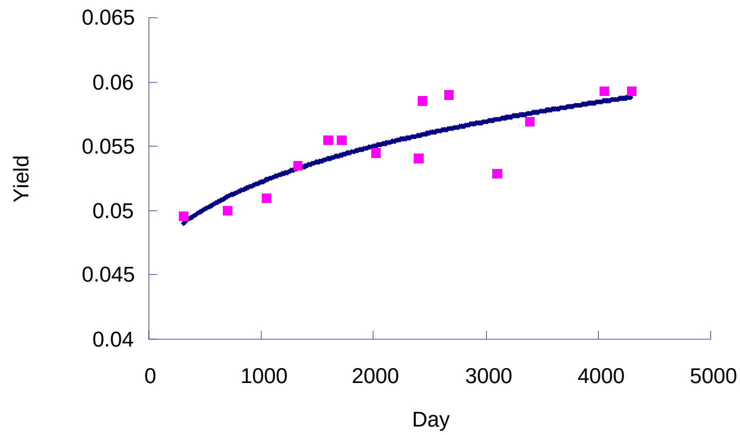

Figure 5.1: LEAST SQUARE WITH N = 2.

year measurement).

In the curve fitting methods of least squares polynomial and least squares polynomial with cubic spline constraint, there exists error between the original data points and the fitting curve. According to statistic, the error in this thesis is noted by sum of error square(SES). For every input data pair (x, y), consider the estimated value $\hat{y}$, the error square between y and $\hat{y}$ is given by (y - $\hat{y}$)^2, and the sum of error square(SES) is therefore:

$$SES = \sum_i (y - \hat{y})^2, \text{ for every input data point i.} \tag{5.1}$$

The output curve given in this chapter is plotted from the fitted function in 10 days a point. If there exists errors between estimated values and original input points, the corresponding SES is attached under the plot, too.

5.2 Least Squares Polynomial

By computing the LU decomposition of matrix A given in section 4.1 and solve the vector X by back-substitution, the arguments of least squares polynomial could be found. Use this function to plot the fitting curve and calculate the SES on that.

The different degree n of the least squares polynomial makes different fitting curve as well as the SES on it. With n = 2, 4, 6, the fitting curve are listed in figure 5.1 to figure 5.3. The small squares in the figures indicate original input data points.

Consider the SES against the degree of fitted curve, the relationship is shown in fig 5.4. It is obviously that the SES decreased with the increasing of degree n.

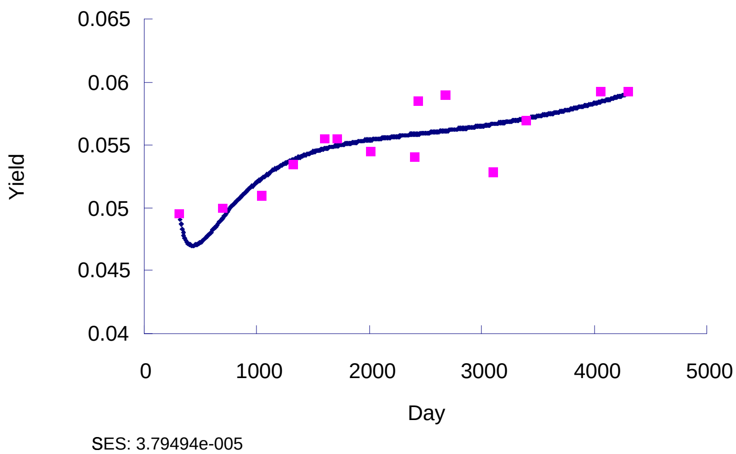

Figure 5.2: LEAST SQUARE WITH N = 4.

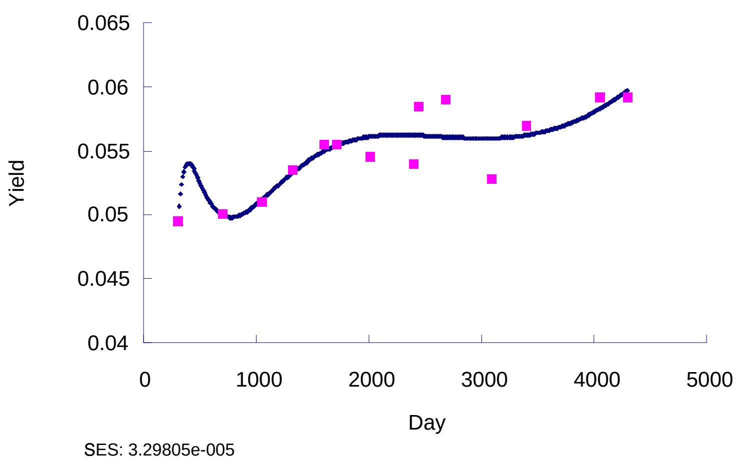

Figure 5.3: LEAST SQUARE WITH N = 6.

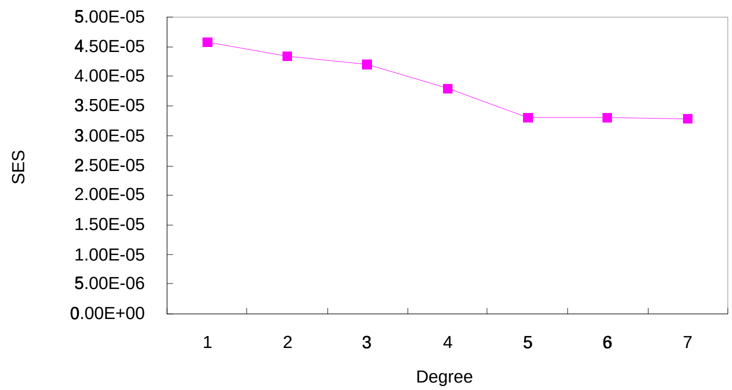

Figure 5.4: SES AGAINST N.

5.3 Cubic Spline

According to the concepts listed in chapter 4, with nature spline constraints, the fitting curve of the input data is shown in figure 5.5.

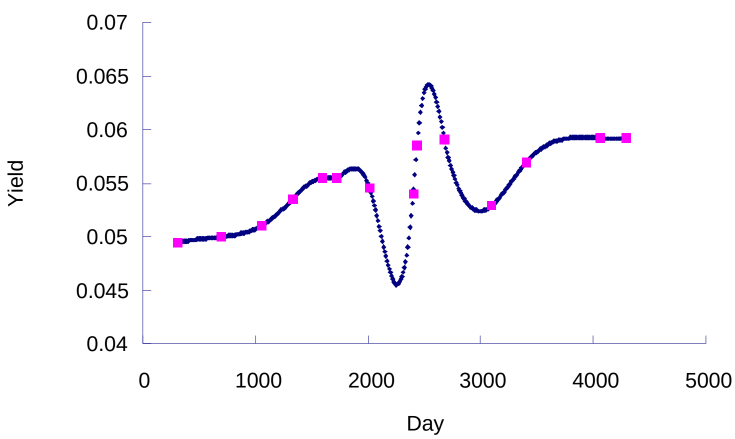

Figure 5.5: NATURE CUBIC SPLINE.

5.4 Least Squares Polynomial with Cubic Spline Constraints

The most important part of this thesis is to combine the previous methods together, in that the output fitting curve has their properties both. When using least squares polynomial with cubic spline constraints, the breakpoints must be assigned by user. The SES analysis is more complex than before since it differs with different breakpoints.

According to McCulloch(1971), where he recommended to have equal maturity dates in every interval. Since the input data here are distributed nearly uniform, the breakpoints here are taken equally spaced.

Consider the equally spaced breakpoint case, with number of intervals n = 2, 4, 6, 8, the corresponding output curve are shown in figure 5.6 to figure 5.9. The fitting curve is separated by breakpoints, and the breakpoints and SES are listed below the plot. Remember the interval number is one more than the breakpoints number, with the first interval starts at the first input data point and the last interval ends on the last input data point.
And the SES against number of intervals are listed in figure 5.10. While the number of intervals(as well as breakpoints) increases, the corresponding SES decreases. The n here is at most 11 because of the constraint $m \geq (n+3)$ described in previous chapter to keep unique solution. In the case of n = 11, the SES should be 0 theoretically(the curve pass through every data point), the nonzero SES here comes from the truncation error of calculation.

It is reasonable to expect a smaller SES with carefully chosen breakpoints, but according to the case here only little improvement can it get. For the sake of finding how the breakpoints' choosing influence SES, the next section gives another input data for further discussion.

5.5 The Effect of Breakpoints: Another Example

In the previous case where the breakpoints' position makes SES only little difference. Considering the fitting method's properties, because it is consisted by separated cubic polynomial, choosing breakpoints by hand may take advantage on jumping data. This section gives an example of how jumping data looked like and how to choose breakpoints on it.

First, we generate a different input data made by picking part of previous one, and change the yield of bond 88101 from 5.690% to 4.100%. The result jumping data are listed in table 5.2.

The least square polynomial with n = 2 is given in figure 5.11. And the SES comparing to degree n is shown in figure 5.12. Because of the far-away yield of bond 88101, the SES is larger than previous case. At n = 7, the output curve should pass

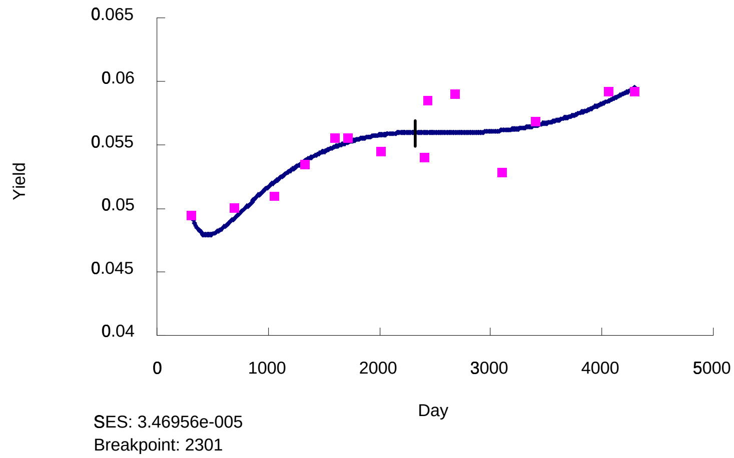

Figure 5.6: Least Squares Polynomial with Cubic Constraint n = 2.

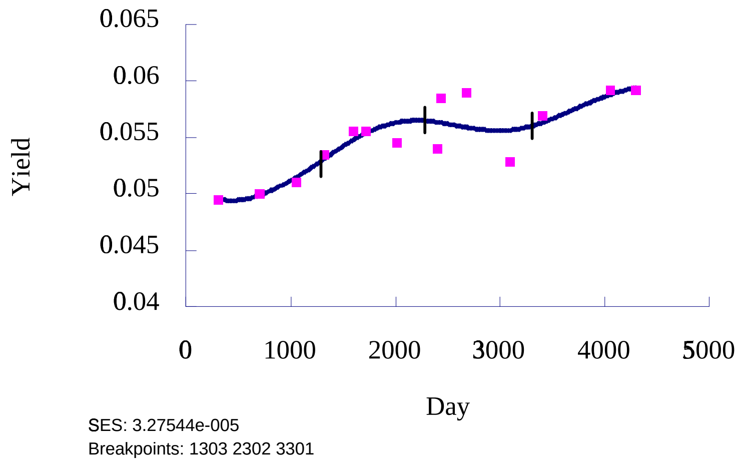

Figure 5.7: Least Squares Polynomial with Cubic Constraint n = 4.

Table 5.2: The Jumping Input Data Yield

Bond ID	Maturity Date	Yield	Bond ID	Maturity Date	Yield
000824	19/2/2000	4.950	00833	18/1/2004	5.550
00064	25/3/2001	5.000	00841	18/11/2004	5.450
00341	17/3/2002	5.100	86101	24/9/2006	4.100
00853	22/12/2002	5.350	88101	25/9/2008	5.690
00831	22/9/2003	5.550	00955	22/3/2011	5.920

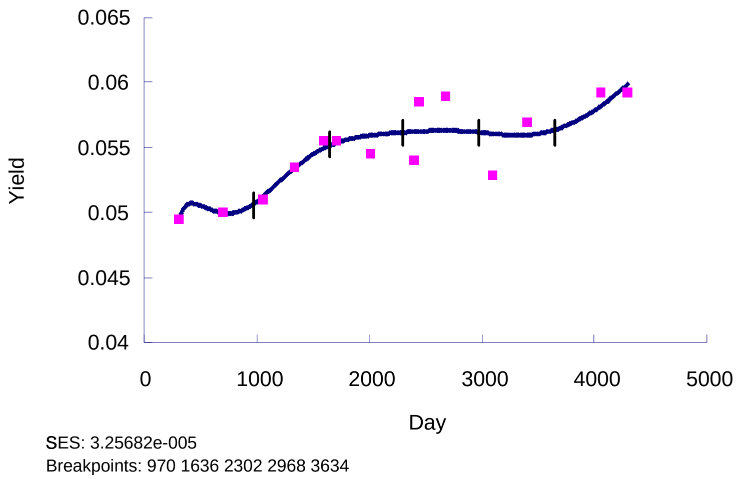

Figure 5.8: Least Squares Polynomial with Cubic Constraint n = 6.

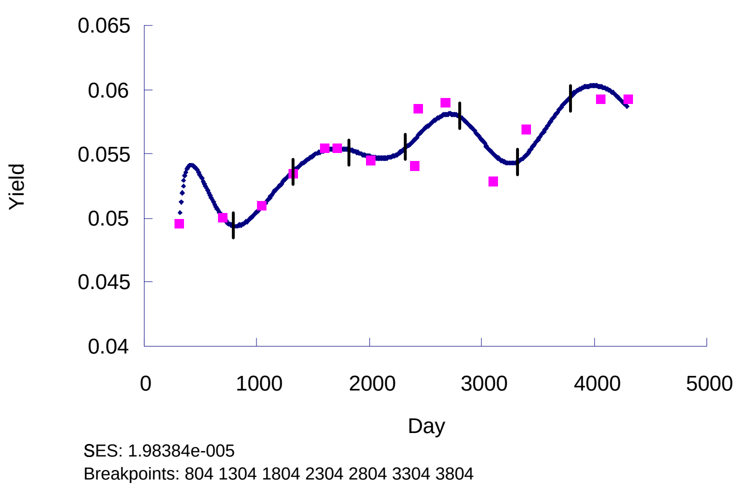

Figure 5.9: Least Squares Polynomial with Cubic Constraint n = 8.

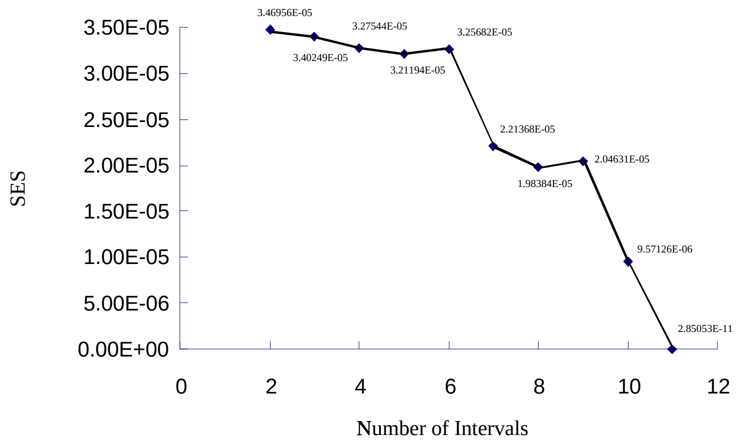

Figure 5.10: SES AGAINST NUMBER OF INTERVALS.

through the input data. In other words, the SES here should be 0, but consult the output that there exists little error from truncation error.

Consider the least squares polynomial with cubic constraint of equally spaced breakpoints case, with number of intervals n = 2, 4, the output are given in 5.13 and 5.14. And the relationship between SES and n is shown in 5.15.

In this jumping data case, it can take advantage by specifying right positions of breakpoints. For example, in 2 intervals condition we pick the only breakpoint be 3200, which makes the SES = 7.87668e-005. The output is given in figure 5.16. This is much better than that in equally spaced condition(which SES = 0.000144894). The same result holds in all n $\geq$ 2.

Look at the position of breakpoint carefully, it falls before the bond 88101's maturity. This makes both the cubic splines separated by the breakpoint with smooth input data points except the endpoint. Think it intuitive, a cubic spline can fit data like this precisely. That's why these breakpoints leads smaller error(SES).

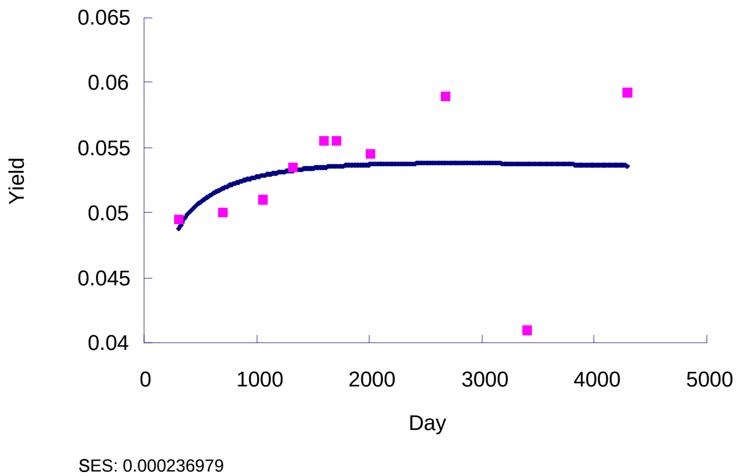

Figure 5.11: Least Square with n = 2 in jumping data.

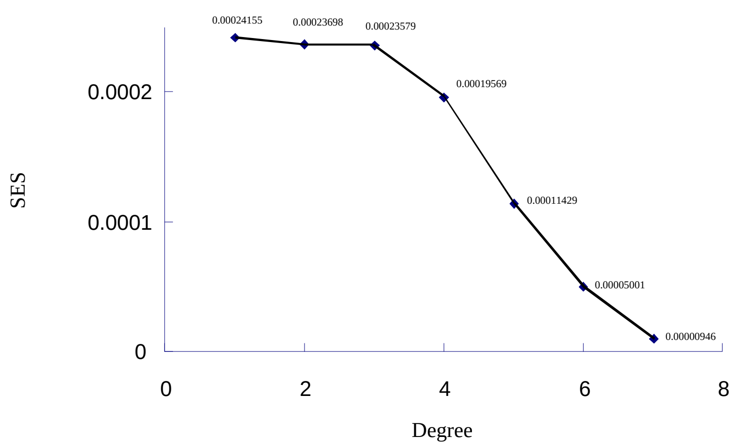

Figure 5.12: SES against n in jumping data.

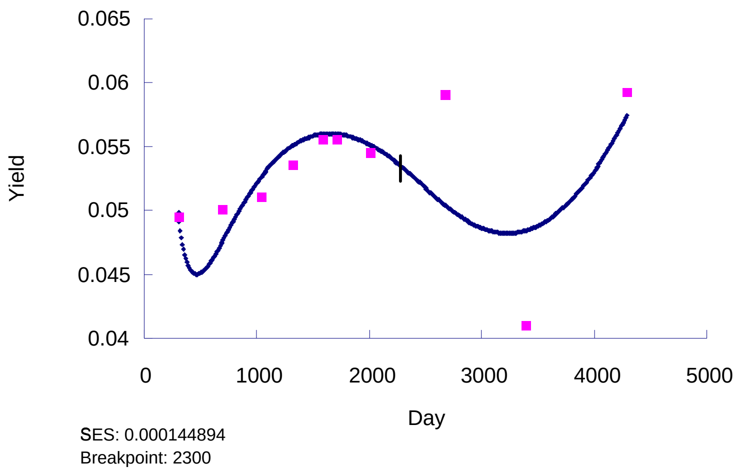

Figure 5.13: Least Squares Polynomial with Cubic Constraint n = 2 in jumping data.

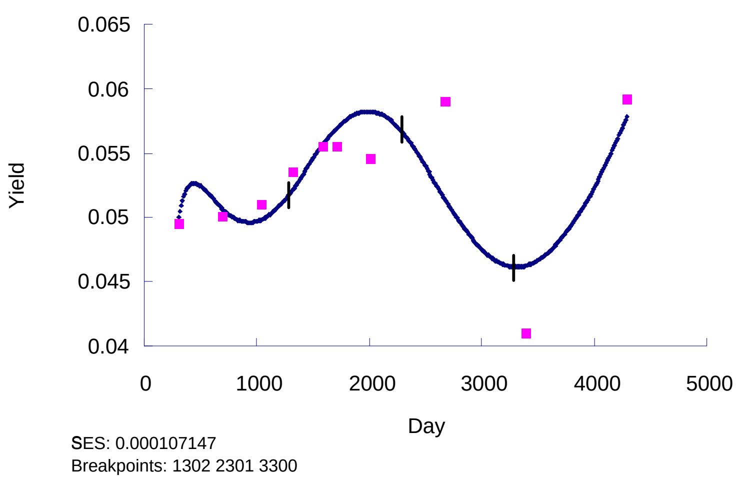

Figure 5.14: Least Squares Polynomial with Cubic Constraint n = 4 in jumping data.

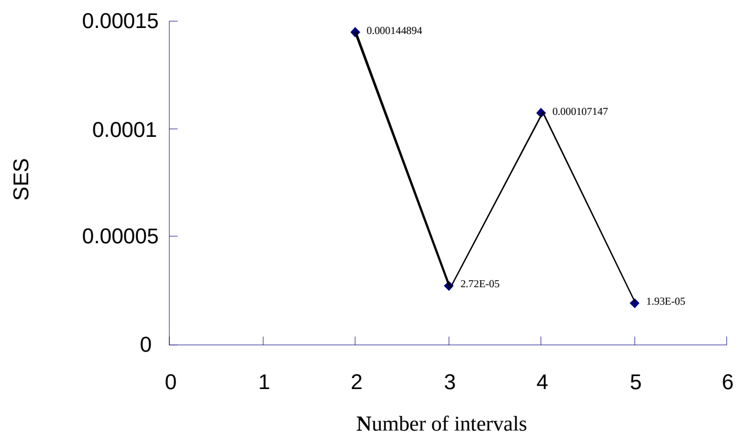

Figure 5.15: SES against number of intervals in jumping data.

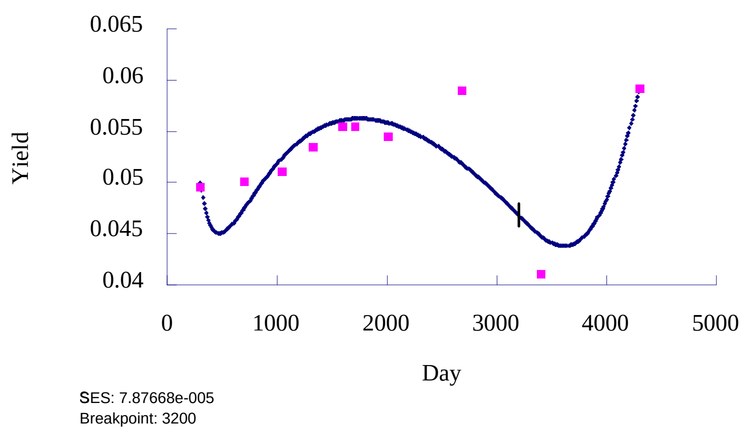

Figure 5.16: Least Squares Polynomial with Cubic Constraint of specified breakpoints in jumping data.

Chapter 6

Conclusion

This thesis gives three curve fitting methods to fit the given yield-maturity data. In that both the least squares polynomial and the cubic spline methods are heavily used in scientific field, and this thesis gives a third method with the mixed properties of the first two. This makes the third fitting method a harder one, both theoretical and practical using, but more powerful in its usage(meet both properties).

In least squares polynomial method, consider figure 5.4, the SES decreases with the degree n increase. This is just what expected and is reasonable. In other hand, although the SES goes down with larger degree n, the plot becomes more and more oscillational. In the last fitting method of least square polynomial with cubic constraints, this phenomenon holds, too. This makes a tradeoff between smooth and preciseness, consider the case of this thesis, choosing the degree or number of intervals between 2 and 4 seemed suitable.

In the least squares polynomial with cubic constraint, there is another degree of freedom of taking decision on choosing the breakpoints. For the purpose of figuring out the effect, this thesis makes another example of jumping data input. The conclusion of section 5.5 recommended to take breakpoints near the jumped data point for smaller estimation error.

After all the results given in this thesis, the least squares polynomial with cubic spline constraints method may be a better choice because of the freedom of breakpoints choosing. Besides, although this method involves more complicate matrix calculation than others, the computation time of these methods are all within 1 or 2 seconds. More precisely, the computational complexity are all $\mathbf{O}(n^3)$ which implies that the third method is no harder than the first two.

Bibliography

[1] B. Noble and J. W. Daniel, *Applied Linear Algebra*, 3rd edition. Prentice-Hall. 1988.

[2] C. de Boor, *A Practical Guide to Splines*. Springer-Verlag. 1978.

[3] C. F. Gerald and P. O. Wheatley, *Applied Numerical Analysis*, 5th edition. Addison Wesley. 1994.

[4] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Prentice-Hall. 1974.

[5] D. G. Luenberger, *Investment Science*. Oxford University Press, Inc. 1998.

[6] D. R. Chambers, W. T. Carleton, and D. W. Waldman, "A New Approach to Estimation of the Term Structure of Interest Rates." *Journal of Financial and Quantitative Analysis* Vol. 19, No. 3, 1084, pp. 233–252.

[7] J. C. Van Horne, *Financial Market Rates & Flows*, 4th editio. Prentice-Hall. 1994.

[8] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, second edition. Springer-Verlag. 1992.

[9] J. W. Longley, *Least Squares Computations Using Orthogonalization Methods*. Marcel Dekker, Inc. 1984.

[10] Lyuu, Yuh-Dauh, *Introduction to Financial Computation: Principles, Mathematics, Algorithms.* Manuscripts, Feb. 1995–1997.

[11] S. Sundaresan, *Fixed Income Markets and Their Derivatives.* South-Western. 1997.

[12] McCulloch, J. H., "Measuring the Term Structure of Interest Rates." *Journal of Business* Vol. 44(January 1971), pp. 19–31.