Constant Maturity Yield Curve Study: UK and US

Po-Hsuan Lin

NYU Tandon School of Engineering MFE Program

Bank of England Constant Maturity Yield Curve Methodology

We will start by examining the methodology used by Bank of England in constructing their constant maturity treasury rates. Constant maturity treasury rates are the set of yields on government bonds, which are known as gilts in UK. Since government bonds are issued in certain maturities, to obtain the yield for any other maturities requires building the constant maturity yield curve.

There are two major approaches in constructing the constant maturity yield curves: parametric model and cubic spline interpolation. Parametric models simplify the yield curve into a few important parameters. An example will be the Nelson-Siegel model; it can be summarized in the following expression:

$$y(m,\beta) = \beta_0 + \beta_1 e^{-\frac{m}{\tau_1}} + \beta_2 \frac{m}{\tau_1} e^{-\frac{m}{\tau_1}}$$

Its flexibility corresponds to the number of factors introduced. In the case of Nelson-Siegel, the degree of freedom is provided by these four factors $\beta_0, \beta_1, \beta_2$ and τ_1.

On the other end of the spectrum is the cubic spline interpolation. Instead of modeling the yield curve to parameters, it's is fitted to a piecewise cubic polynomial. The coefficients of the spline are restricted so that the curve is continuous and smooth at all maturities. A detailed discussion and comparison of the various parametric and cubic spline methods can be found in Bank of England Quarterly Bulletin article by Anderson and Sleath (1999).

Before 2000, Bank of England focused mostly on the parametric methods, which included the earlier Mastronikola method (1991) and the more widely used Svensson method (1994,1995). The Svensson model generalized formula can be seen below:

$$y(m,\beta) = \beta_0 + \beta_1 e^{-\frac{m}{\tau_1}} + \beta_2 \frac{m}{\tau_1} e^{-\frac{m}{\tau_1}} + \beta_3 \frac{m}{\tau_2} e^{-\frac{m}{\tau_2}}$$

It is similar to the Nelson-Siegel model but with two additional factors β_3 and τ_2. Despite having more factors, the Svensson model has a limitation. Its long-term yield is constrained to a constant level due to the expectation hypothesis. However, the expectation hypothesis in reality does not hold, as pointed out in Anderson and Sleath (1999). Thus, at the long-end of the yield curve, a discrepancy between the actual yield and Svensson's predicted yield emerges.

This limitation prompts the Bank of England to replace the Svensson method with a more cubic spline approach. The cubic spline method it currently uses is called variable roughness penalty or VRP. It is based on a cubic spline technique introduced by Waggoner (1997). According to Anderson and Sleath (1999) and Hurd (2005), VRP is superior "based upon the criteria of smoothness, flexibility and stability". It overcomes Svensson model's limitation by associating flexibility with maturity; the curve is more flexible at the short end than at the long end. VRP minimizes the following objective function:

$$X = X_{residuals} + \int_0^M \lambda_t(m)[f''(m)]^2 dm$$

M is the maturity of the longest bond. $\lambda_t(m)$ is the curvature penalty that links flexibility, $f''(m)$ with maturity, m. Bank of England Quarterly Bulletin November 2009 appendix provides a more detailed and mathematically rigorous exploration of the VRP technique.

Discussion of Expectation Hypothesis

Expectation hypothesis states that the forward rate should theoretically equal to the expected future interest rate. If the forward rate is different from the expected future interest rate, arbitrage opportunities arise. An example being the expected future interest rate being higher than the current forward rate, investors can borrow and lock in the forward rate right now while waiting to lend at the higher expected forward rate. The rate difference is an obvious arbitrage opportunity, thus in theory, expectation hypothesis should hold for a complete no-arbitrage market. However, in reality, due to the uncertainty of future interest rate, several premiums, with risk premium being the most important one, are incorporated into the expected short rates. This creates a discrepancy between expected rates and forward rates as shown in the graph provided by Brooke and Cooper (2000).

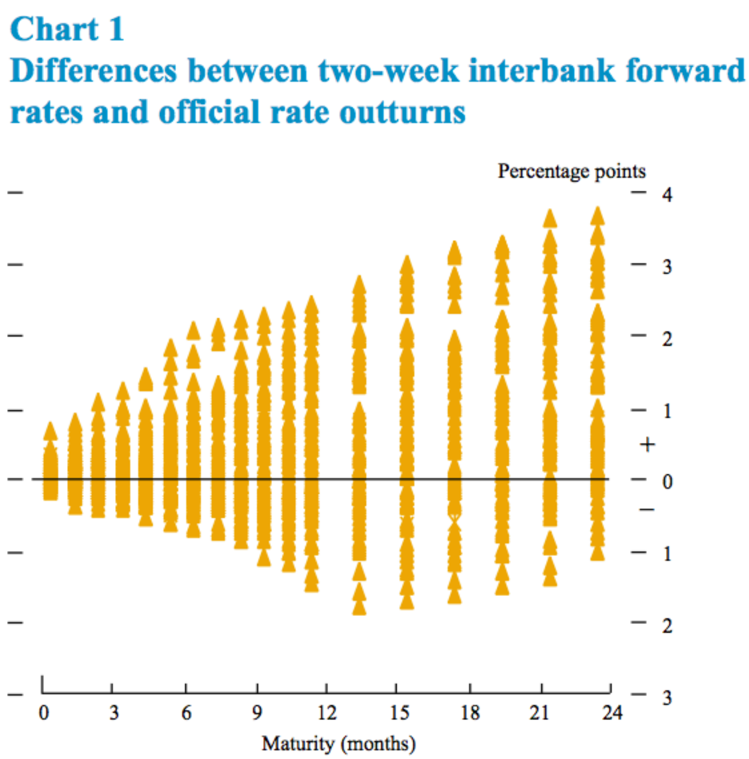

Figure 1 Differences between forward rates and expected rates

Therefore, we can conclude that expectation hypothesis may not hold for longer maturities. For shorter duration, the expected rates are very close to forward rates as expectation of the future values hardly deviate from current values. This short discussion of expectation hypothesis concludes our section on UK's constant maturity yield curve.

US Constant Maturity Yield Curve Methodology

The following section will briefly go over US's approach on constructing its constant maturity yield curve. The methodology is described in detail in Fisher et al (1995) and summarized in Fisher (2005). Similar to UK, US Department of Treasury uses a cubic-spline based approach. The methodology is classified as a "quasi-cubic spline hermite spline function" on Treasury Yield Curve Methodology website (2009). Judging from the summary provided by Fisher, this quasi-cubic spline hermite function is very similar to VRP technique. It is a summation of the squared deviations of price discrepancies in bond prices and a penalty of non-linearity, which coincides with the VRP's objective function.

Yet, another factor is incorporated into US' calculation. The weights at which these two factors are summed are calculated using a specific technique called generalized cross-validation (GCV) criterion. It can be simplified into the following ratio as formulated by Fisher (2005): the ratio of a quasi-out-of-sample goodness-of-fit measure to the effective number of parameters. This ratio provides a tradeoff between the fitness and flexibility of the model. For more information regarding this methodology, please look over Fisher at el (1995).

Data Processing and Representation

In this data processing and representation section, I extract the US constant maturity yield curve from Federal Reserve Economic Database using R. Original code is used in R in Finance professor course and modified here. Snippets of the code can be found below.

```r
# Select the data for extraction
symbols <- c("DGS1", "DGS2", "DGS3", "DGS5", "DGS10", "DGS20", "DGS30")
# Use the library quantmod which has a built-in library of functions
library(quantmod)
# Create a new environment for data
env_fred <- new.env()
# Extract data
getSymbols(symbols, env=env_fred, src="FRED")
# Show each individual series using chart_Series
chart_Series(env_fred$DGS1["1990/"],name="1-year constant maturity Treasury rate")
```

Code 1, Data Extraction

This code uses the library quantmod to download constant maturity yield curve data from FRED and save them in the environment. Time series graphs of 7 different maturities yield curve can be found in Appendix I.

After showing how the constant maturity yield changes over time, we can also show how the yield curve changes year after year. The yield curves drawn here are unprocessed, meaning no model or spline interpolation has been used.

```r
# get end-of-year dates since 2006
dates <- xts::endpoints(env_fred$DGS1["2006/"], on="years")
dates <- zoo::index(env_fred$DGS1["2006/"])[dates]
# create time series of end-of-year rates
rates <- eapply(env_fred, function(rate) rate[dates])
rates <- rutils::do_call(cbind, rates)
# rename columns and rows, sort columns, and transpose into matrix
colnames(rates) <- substr(colnames(rates), start=4, stop=11)
rates <- rates[, order(as.numeric(colnames(rates)))]
colnames(rates) <- paste0(colnames(rates), "yr")
rates <- t(rates)
colnames(rates) <- substr(colnames(rates), start=1, stop=4)
# plot matrix using plot.zoo()
col_ors <- colorRampPalette(c("red", "blue"))(NCOL(rates))
plot.zoo(rate_s, main="Yield curve since 2006", lwd=3, xaxt="n",
         plot.type="single", xlab="maturity", ylab="yield", col=col_ors)
# add x-axis
axis(1, seq_along(rownames(rates)), rownames(rates))
# add legend
legend("bottomright", legend=colnames(rates),
       col=col_ors, lty=1, lwd=4, inset=0.05, cex=0.8)
```

Code 2, Time Series of CMY

The time series result is shown below. As one can see from the graph, right before the financial crisis in 2008, the yield curve was flat, especially the one in 2006. As US was hit by the financial crisis, the federal reserve decreased the short-term borrowing rate. The sudden decrease in short-term rate caused the overall yield curve to be steeper, as reflected in 2008 and 2009 yield curve. Since then, the long-term yield gradually decreased down to a low point in 2011 and 2012 as market adjusted to the expectation of having low yield. The market started to recover slowly in the past five years which prompted the federal reserve to start raising interest rate. Under Mrs. Yellen's tenure, she raised the interest rate corresponding to the inflation and unemployment data while preventing overheating the economy. This led to today's data. The yield curve is again a mild upward slopping curve.

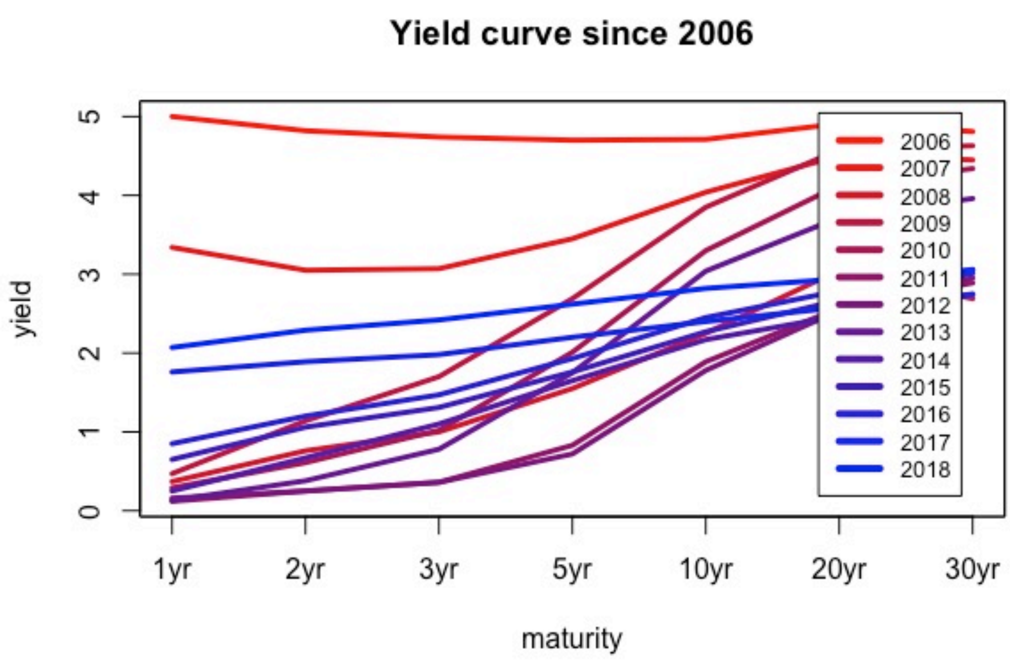

Figure 2, Time Series of Constant Maturity Yield Curve

Cubic Spline Interpolation

The following MATLAB cubic spline interpolation code is written by me for another class which utilizes LU decomposition of tridiagonal symmetric positive definite matrix to create a cubic spline interpolation. I decide to implement the interpolation on 2014 data since it has a good mixture of low short term rates and higher long term rates. The data is taken directly from the previous section. When we graph the data above directly in MATLAB, we get the figure 3.

2014	1yr	2yr	3yr	5yr	10yr	20yr	30yr
Rate	0.25	0.67	1.1	1.65	2.17	2.47	2.75

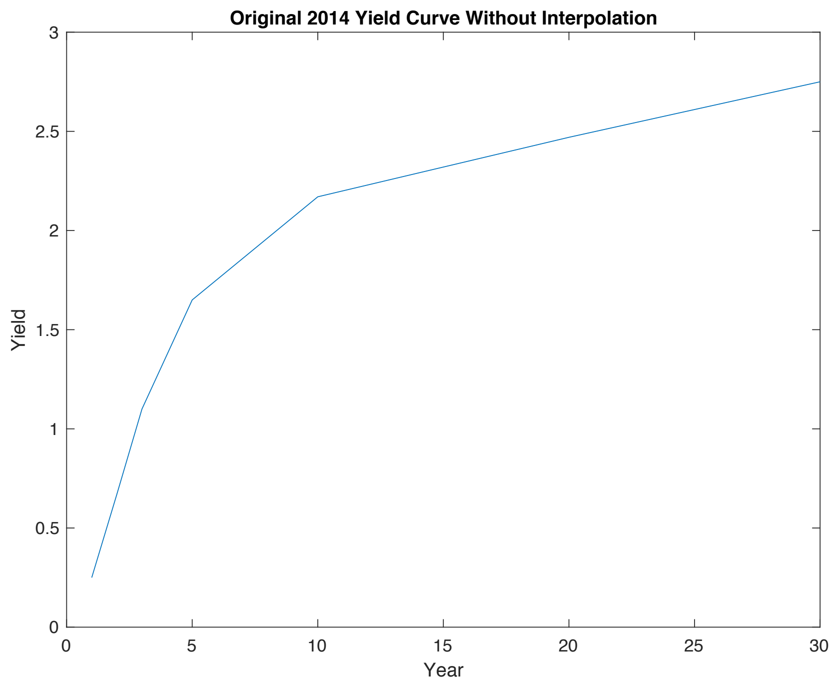

Figure 3, 2014 Yield Curve Without Interpolation

The code is attached in Appendix II. The process involves using a linear solver that runs the LU decomposition with no pivoting on a tridiagonal matrix, forward substitution on a lower triangular matrix and backward substation on an upper triangular matrix.

After running the codes, we can get the following equation for each interval:

$$y = -0.17 + 0.4373x - 0.0259x^2 + 0.0086x^3, 1 \leq x \leq 2$$

$$y = 0.1641 - 0.0538x + 0.2247x^2 - 0.0331x^3, 2 \leq x \leq 3$$

$$y = -0.9493 + 1.0495x - 0.1464x^2 + 0.0081x^3, 3 \leq x \leq 5$$

$$y = -0.1147 + 0.5487x - 0.0463x^2 + 0.0014x^3, 5 \leq x \leq 10$$

$$y = 1.1716 + 0.1629x - 0.0077x^2 + 0.0001x^3, 10 \leq x \leq 20$$

$$y = 2.4884 - 0.0347x + 0.0022x^2 - 0.000024x^3, 20 \leq x \leq 30$$

When we plot the set of equations above with the corresponding intervals, the following graph can be obtained. Due to the cubic polynomials nature, the curve is smooth and continuous at all points. Each interval is colored differently. An overlay of the interpolated curve on the original un-interpolated curve can be found in Appendix I, figure 12. The red curve in figure 12 is the same as the curve in figure 3.

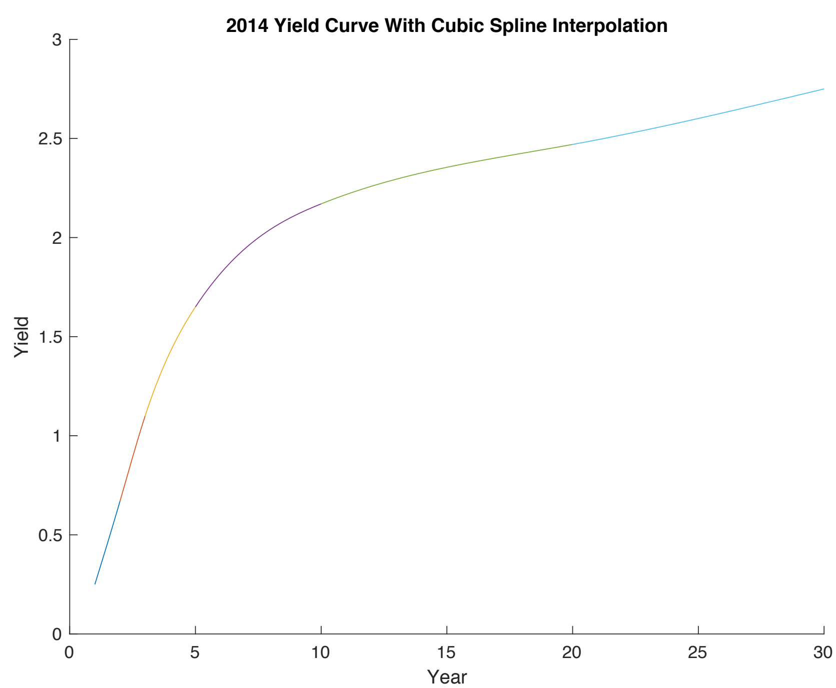

Figure 4, 2014 Yield Curve with Cubic Spline Interpolation

References

Fisher, M (1996): Fitting and interpreting the US yield curve at the Federal Reserve Board. Fisher, M, D Nychka and D Zervos (1995): "Fitting the term structure of interest rates with smoothing splines", Board of Governors of the Federal Reserve System, Federal Reserve Board Working Paper 95-1.

Anderson, N and J Sleath (1999): "New estimates of the UK real and nominal yield curves", Bank of England Quarterly Bulletin, November, pp 384-92.

——— (2001): "New estimates of the UK real and nominal yield curves", Bank of England Working Paper, no 126.

Brooke, M, N Cooper and C Scholtes (2000): "Inferring market interest rate expectations from money market rates", Bank of England Quarterly Bulletin, November, pp 392-402.

Waggoner, D (1997): "Spline methods for extracting interest rate curves from coupon bond prices", Federal Reserve Bank of Atlanta, Working Paper series, 97-10.

Appendix I

Figure 5, 1 Year Constant Maturity Yield Curve

Figure 6, 2 Year Constant Maturity Yield Curve

Figure 7, 3 Year Constant Maturity Yield Curve

Figure 8, 5 Year Constant Maturity Yield Curve

Figure 9, 10 Year Constant Maturity Yield Curve

Figure 10, 20 Year Constant Maturity Yield Curve

Figure 11, 30 Year Constant Maturity Yield Curve

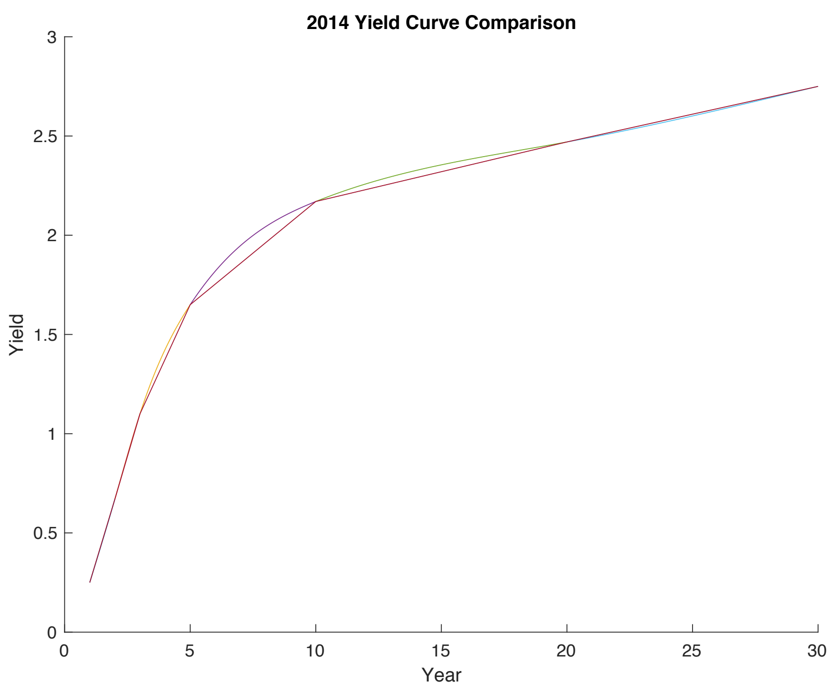

Figure 12, 2014 Yield Curve Comparison

Appendix II

```
%% FRE6411 Final Project
% Created by Michael (Po-Hsuan) Lin
% Last update: 03/16/2018

n = 6;
x = [1 2 3 5 10 20 30];
v = [0.25 0.67 1.1 1.65 2.17 2.47 2.75];
z = zeros(1,n-1);
M = zeros(n-1,n-1);
w = zeros(n+1,0);
a = zeros(1,n);
b = zeros(1,n);
c = zeros(1,n);
d = zeros(1,n);
q = zeros(1,n);
r = zeros(1,n);

for i = 2:n
    z(i-1) = 6*(((v(i+1)-v(i))/(x(i+1)-x(i)))-((v(i)-v(i-1))/(x(i)-x(i-1))));
end

for j = 2:n
    M(j-1,j-1) = 2*(x(j+1)-x(j-1));
end

for k = 2:n-1
    M(k-1,k) = x(k+1) - x(k);
end

for l = 3:n
    M(l-1,l-2) = x(l) - x(l-1);
end

% Call function lu solver
[L,U] = lu_no_pivoting(M);
y = forward_subst(L,transpose(z));
p = backward_subst(U,y);

w(1) = 0;
w(n+1) = 0;
for ii = 2:n
    w(ii) = p(ii-1);
end

for iii = 2:n+1
    c(iii-1) = (w(iii-1)*x(iii)-w(iii)*x(iii-1))/(2*(x(iii)-x(iii-1)));
    d(iii-1) = (w(iii) - w(iii-1))/(6*(x(iii)-x(iii-1)));
end

for iv = 2:n+1
    q(iv-1) = v(iv-1) - c(iv-1)*((x(iv-1))^2) - d(iv-1)*((x(iv-1))^3);
    r(iv-1) = v(iv) - c(iv-1)*((x(iv))^2) - d(iv-1)*((x(iv))^3);
end

for vi = 2:n+1
    a(vi-1) = (q(vi-1)*x(vi) - r(vi-1)*x(vi-1))/(x(vi)-x(vi-1));
    b(vi-1) = (r(vi-1)-q(vi-1))/(x(vi)-x(vi-1));
end

t1 = linspace(1,2,100);
f1 = a(1) + b(1)*t1 + c(1)*(t1.^2) + d(1)*(t1.^3);
t2 = linspace(2,3,100);
```

```
f2 = a(2) + b(2)*t2 + c(2)*(t2.^2) + d(2)*(t2.^3);
t3 = linspace(3,5,100);
f3 = a(3) + b(3)*t3 + c(3)*(t3.^2) + d(3)*(t3.^3);
t4 = linspace(5,10,100);
f4 = a(4) + b(4)*t4 + c(4)*(t4.^2) + d(4)*(t4.^3);
t5 = linspace(10,20,100);
f5 = a(5) + b(5)*t5 + c(5)*(t5.^2) + d(5)*(t5.^3);
t6 = linspace(20,30,100);
f6 = a(6) + b(6)*t6 + c(6)*(t6.^2) + d(6)*(t6.^3);

hold on
plot(t1,f1)
plot(t2,f2)
plot(t3,f3)
plot(t4,f4)
plot(t5,f5)
plot(t6,f6)
xlabel('Year')
ylabel('Yield')
title('2014 Yield Curve With Cubic Spline Interpolation')

%% LU decomposition without pivoting used in cubic_spline_interpolation.m
function [L,U] = lu_no_pivoting(A)
n = size(A,2);
L = zeros(n,n);
U = zeros(n,n);
for i = 1:(n-1)
    for k = i:n
        U(i,k) = A(i,k);
        L(k,i) = A(k,i)/U(i,i);
    end
    for j = (i+1):n
        for k = (i+1):n
            A(j,k) = A(j,k) - L(j,i)*U(i,k);
        end
    end
end
L(n,n) = 1;
U(n,n) = A(n,n);

%% Forward substitution used in cubic_spline_interpolation.m
function x = forward_subst(L,b)
n = size(L,2);
x = zeros(n,1);
x(1,1) = b(1,1) / L(1,1);
for j = 2:n
    sum = 0;
    for k = 1:(j-1)
        sum = sum + L(j,k)*(x(k,1));
    end
    x(j,1) = (b(j,1) - sum) / L(j,j);
end

%% Backward substitution used in cubic_spline_interpolation.m
function x = backward_subst(U,b)
n = size(U,2);
x=zeros(n,1);
for j = n:-1:1
    x(j) = b(j)/U(j,j);
    b(1:j-1) = b(1:j-1) - U(1:j-1,j) * (x(j));
end
```