Spring 2020 Final Exam

Victor Liu
FRE6411 - Fixed Income Securities
Tandon School of Engineering

April 22, 2020

Question 1

From the following site https://www.treasury.gov/resource-center/data-chart-center/interest-rates/pages/TextView.aspx?data=yieldYear&year=2020
Download the daily constant maturity US treasury yields. (CMY)

A)

Estimate daily continuous yield curve and instantaneous forward rate curve by fitting a cubic spline function to the above discrete CMY.

Daily continuous yield curve
The daily constant maturity US treasury yields from Jan 2,2020 to April 8,2020 are loaded from the website as a DataFrame "YieldCurve", shown in table below.

Table: YieldCurve

Date	1 MO	2 MO	3 MO	6 MO	1 YR	2 YR	3 YR	5 YR	7 YR	10 YR	20 YR	30 YR
2020-01-02	1.53	1.55	1.54	1.57	1.56	1.58	1.59	1.67	1.79	1.88	2.19	2.33
2020-01-03	1.52	1.55	1.52	1.55	1.55	1.53	1.54	1.59	1.71	1.80	2.11	2.26
2020-01-06	1.54	1.54	1.56	1.56	1.54	1.54	1.56	1.61	1.72	1.81	2.13	2.28
2020-01-07	1.52	1.53	1.54	1.56	1.53	1.54	1.55	1.62	1.74	1.83	2.16	2.31
2020-01-08	1.50	1.53	1.54	1.56	1.55	1.58	1.61	1.67	1.78	1.87	2.21	2.35

The index of "YieldCurve" is date, so the rows of "YieldCurve" are discrete CMY on corresponding dates.
Natural cubic spline method has been applied to these curves for interpolation. So the interpolation method "CubicSpline" from package "scipy" has been invoked to return continuous functions as continuous CMYs. The functions on different dates have been stored in a dictionary named "yield_d".

```
yield_d = {}
for i in YieldCurve.index:
    yield_d[i] = sp.interpolate.CubicSpline(x, YieldCurve.loc[i,], bc_type = 'natural')
```

The continuous yield curve on Jan 2,2020 has been shown in figure 1.

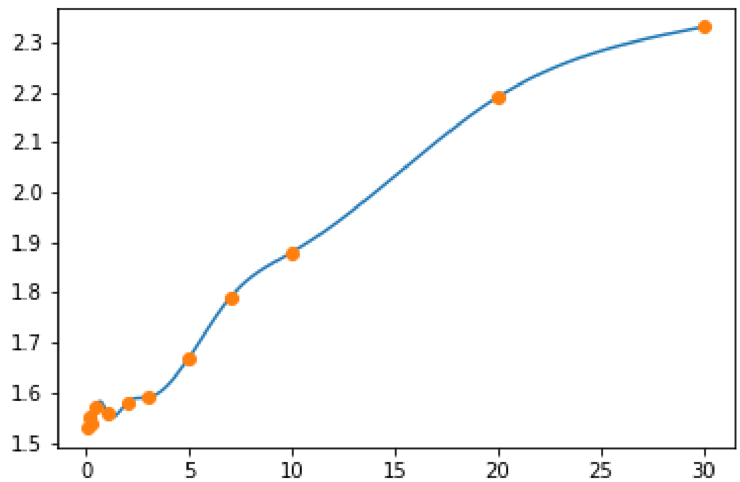

Figure 1: Continuous yield curve on Jan 2,2020

Instantaneous forward rate curve

By using formula:

$$f_t(T) = y_t(T) + T\frac{dy_t(T)}{dT}$$

where $f_t(T)$ is an instantaneous forward rate at T and $y_t(T)$ is the continuous yield curve at T, the daily instantaneous forward rate curves have been linked to the continuous yield curves.

The function "forwardrate" shown below is defined to get instantaneous forward rate:

```
def forwardrate(cs, x):
    return cs(x)+cs.derivative()(x)*x
```

Take the instantaneous forward rate curve in Jan 2,2020 as an example in figure 2:

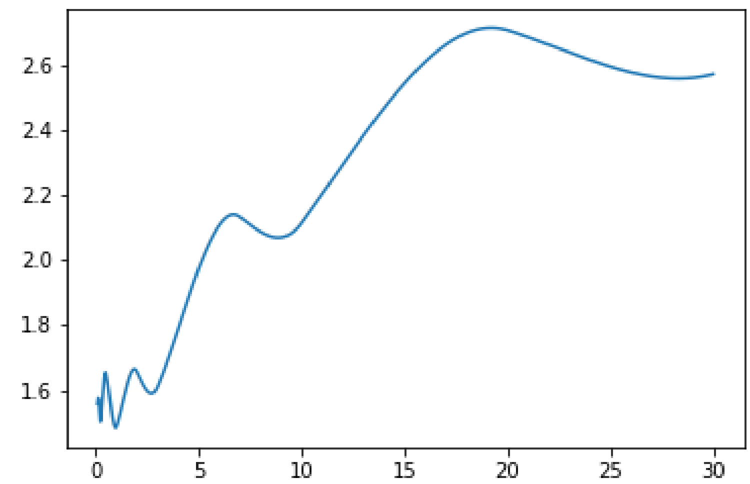

Figure 2: Instantaneous forward rate curve on Jan 2,2020

B)

Graph the end on the months yield curves and instantaneous forward rate curve, and comment on the monthly changes of the two curves.

Figure 3 shows the yield curves on the end of Jan, Feb, Mar 2020.

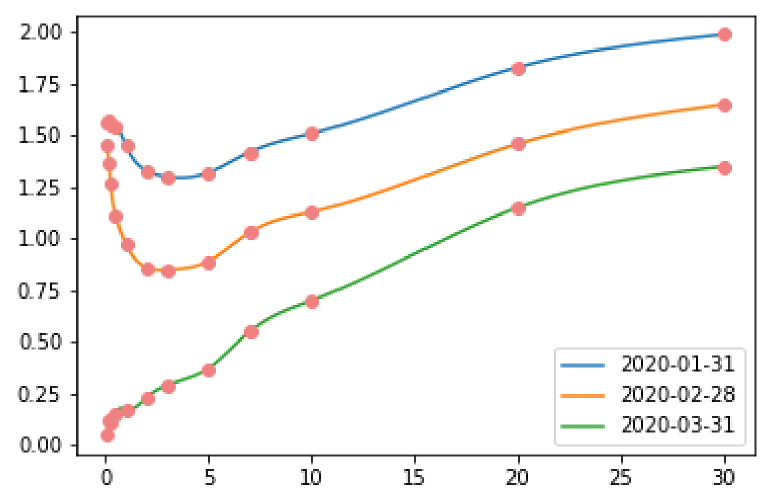

Figure 3: Yield curves on the end of months

The instantaneous forward rate curves on the end of Jan, Feb, Mar 2020 was displayed in figure 4.

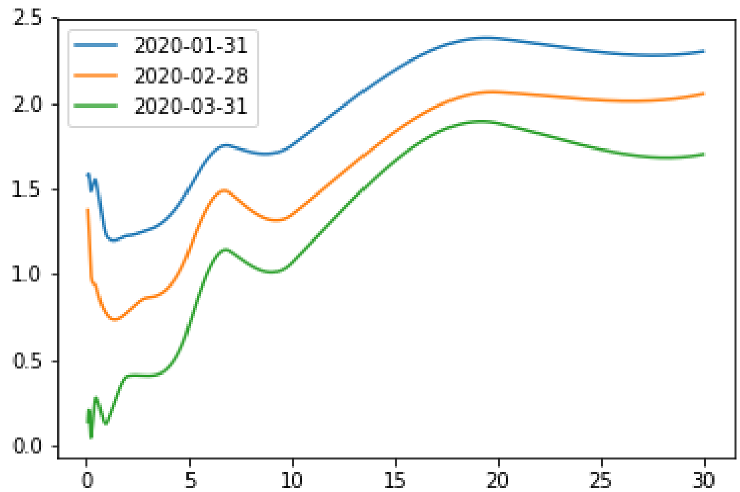

Figure 4: Instantaneous forward rate curves on the end of months

Comment:
At the end of the January, the yield curve shows that the short-term yields from 1-month to three years are inverted, which might be a signal of economic recession. However, at the end of the February, the medium-term and long-term yields decreased rapidly, the spread between the 10 year and 3-month yield became negative. It indicated that suffered from COVID-19 pandemic the long-term US Treasury yields went down as markets took on a more risk averse tone. In March, the Federal Reserve cut the Fed fund rates to 0.00%-0.25%. As a result, the short-term yields went down rapidly, the shape of yield curve became upward-sloping.
For the monthly change of instantaneous forward rate, almost same movement with the CMYs, as time went by, the level of the whole curve decreased, and short-term forward rate went down rapidly so that the shape of short-term forward rate curve on March (0-5years) became upward-sloping from inverted.

Question 2

For ten year US Government bonds futures contract expiring in June 2020(TYM0), the following bonds are deliverable.

A)

For each of deliverable bonds using the estimated yield curve in 1) find the time series of daily price of the bonds in 2020.

Methodology
In order to calculate the daily price of the bonds, a class named “Bond” in Python has been created with attributes “coupon”, “maturity”, “conversion factor”. The method “couponDate” could return all coupon dates instead of the maturity date of a specific bond. The

method "price" can return a bond price at designated date by calling method "couponDate".

```
class Bond:
    def __init__(self, coupon, maturity, cf):
        self.coupon = coupon
        self.maturity = maturity
        self.cf = cf
    def couponDate(self, currentDate):
        from calendar import monthrange
        d = self.maturity.day
        m1 = self.maturity.month
        m2 = (m1 + 6)%12
        dates = []
        ...

        return dates

    def price(self, currentDate, curve):
        price = 0
        for i in self.couponDate(currentDate):
            years = (i-currentDate).days/365
            price = price + (self.coupon/2)/(1+0.01*curve(years))**(years)
        years = (self.maturity-currentDate).days/365
        price = price + (100+self.coupon/2)/(1+0.01*curve(years))**(years)
        return price
```

As a result, by utilizing the price formula,

$$P_t(T_j, C_j) = \Sigma_{i=1}^{N-1} \frac{C_j}{[1 + y(t, T_i)]^{\frac{(T_i - t)}{365}}} + \frac{(100 + C_j)}{[1 + y(t, T_j)]^{\frac{(T_j - t)}{365}}}$$

where T_j is bond maturity and C_j is the coupon payment, N is the number of semi-annual coupon payment, $y(t, T_i)$ is the interpolated yield T_i at t, the daily prices of all bonds have been calculated and stored in the table "Bond_price":

Table: Bond_price

Date	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17
2020-01-02	103.984	99.8379	98.7686	95.9635	104.343	104.055	92.2699	103.524	108.086	108.538	109.426	110.957	107.567	104.855	98.6072	99.247	97.2035
2020-01-03	104.52	100.36	99.2899	96.4814	104.893	104.617	92.7824	104.101	108.69	109.163	110.069	111.627	108.244	105.544	99.2929	99.9565	97.9239
2020-01-06	104.468	100.309	99.2403	96.4335	104.843	104.567	92.7365	104.051	108.638	109.108	110.012	111.565	108.181	105.477	99.2248	99.8829	97.8467
2020-01-07	104.346	100.189	99.1199	96.3124	104.716	104.436	92.6148	103.916	108.497	108.962	109.862	111.41	108.022	105.316	99.0629	99.7161	97.6769
2020-01-08	104.08	99.9321	98.8634	96.0582	104.442	104.156	92.3643	103.628	108.194	108.649	109.54	111.074	107.685	104.973	98.724	99.3652	97.3212
2020-01-09	104.156	100.005	98.9358	96.1295	104.521	104.239	92.4342	103.716	108.292	108.756	109.656	111.203	107.823	105.123	98.8819	99.5382	97.5067
2020-01-10	104.356	100.201	99.1315	96.3245	104.725	104.444	92.6275	103.923	108.502	108.967	109.866	111.414	108.028	105.322	99.0719	99.7261	97.689
2020-01-13	104.242	100.089	99.0199	96.2138	104.608	104.326	92.5181	103.802	108.376	108.835	109.73	111.271	107.882	105.172	98.9202	99.5677	97.5267
2020-01-14	104.38	100.222	99.1542	96.3483	104.754	104.479	92.6525	103.964	108.551	109.023	109.929	111.485	108.106	105.408	99.1638	99.8265	97.7971
2020-01-15	104.586	100.421	99.354	96.5474	104.968	104.699	92.8502	104.191	108.789	109.268	110.182	111.747	108.37	105.675	99.4276	100.098	98.0707

Result
The time series of daily price of the bonds in 2020 are displayed in Figure 5:

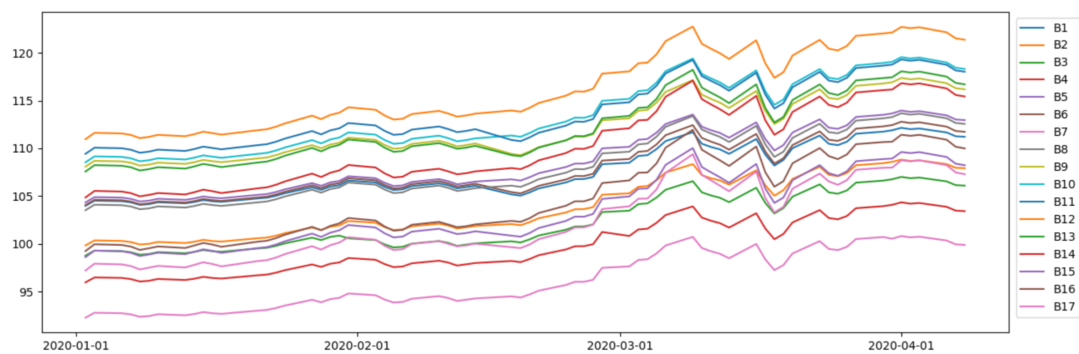

Figure 5: Daily prices of the bonds

B)

Find the daily rate of return of each bond and conversion factor adjusted return (conversion factor adjusted return is simply the daily return multiply by the conversion factor)

Methodology
The bonds' conversion factors have been derivatized from CME method.
(site: https://www.cmegroup.com/trading/interest-rates/calculating-us-treasury-futures-conversion-factors.html)

A bond's conversion factor is defined as:

$$
\begin{aligned}
\text{factor} &= a \times \left[\frac{\text{coupon}}{2} + c + d\right] - b \\
a &= \frac{1}{1.03^{\frac{v}{6}}} \\
b &= \frac{\text{coupon}}{2} \times \frac{6-v}{6} \\
c &= \frac{1}{1.03^{2n}} \quad \text{if } z < 7 \quad \text{or} \quad \frac{1}{1.03^{2n+1}} \quad \text{if otherwise} \\
d &= \frac{\text{coupon}}{0.06} \times (1-c) \\
v &= z \quad \text{if } z < 7 \quad \text{or} \quad (z-6) \quad \text{if otherwise}
\end{aligned}
$$

where factor is rounded to four decimal places, and coupon is the bond's annual coupon in decimals, n is the number of whole years from the first day of the delivery month to the maturity date of the bond, z is the number of whole months between n and the maturity date rounded down to the nearest quarter.
A function "factor" is created to check the conversion factors of these bonds.

```
def factor(Bond, Futures_maturity):
    if(Bond.maturity.month - Futures_maturity.month >=0):
        n = Bond.maturity.year - Futures_maturity.year
        m = Bond.maturity.month - Futures_maturity.month
    else:
        n = Bond.maturity.year - Futures_maturity.year -1
        m = 12+(Bond.maturity.month - Futures_maturity.month)
    z = (m//3)*3
    v = z if z<7 else (z-6)
    a = 1/1.03**(v/6)
    b = (Bond.coupon*0.01/2)*(6-v)/6
    c = 1/1.03**(2*n) if (z<7) else 1/1.03**(2*n+1)
    d = (Bond.coupon*0.01/0.06)*(1-c)
    factor = a * (Bond.coupon*0.01/2+c+d) -b
    return factor
```

The result shows that all bonds' conversion factors are correct.

```
Output:
0.8006
0.7740
0.7607
0.7408
0.8012
0.7882
0.7052
0.7821
0.8060
0.8085
0.8037
0.8150
0.7778
```

```
0.7560
0.6991
0.7016
0.6777
```

The daily rates of return of each bond are stored in table "Bond_return", while conversion factor adjusted returns are in table "Bond_adjreturn".

Table: Bond_return

Date	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13
2020-01-03	0.515518	0.522736	0.527721	0.539699	0.527607	0.539897	0.555444	0.557006	0.558783	0.575318	0.5879	0.603721	0.628837
2020-01-06	-0.0491379	-0.0510184	-0.0499515	-0.049626	-0.0482387	-0.0471615	-0.0494601	-0.0478813	-0.0480905	-0.0502796	-0.0519388	-0.0552964	-0.0585399
2020-01-07	-0.116971	-0.118853	-0.121268	-0.125595	-0.12093	-0.125315	-0.131224	-0.130082	-0.129867	-0.133599	-0.136585	-0.139482	-0.146159
2020-01-08	-0.254834	-0.256722	-0.25879	-0.263944	-0.261759	-0.267966	-0.270508	-0.276677	-0.278701	-0.287068	-0.293047	-0.301118	-0.312133
2020-01-09	0.072751	0.0730083	0.0731907	0.0742292	0.0754889	0.0790813	0.0756999	0.0849268	0.0901009	0.0980674	0.106246	0.116108	0.128317

Table: Bond_adjreturn

Date	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13
2020-01-03	0.412724	0.404598	0.401437	0.399809	0.422718	0.425546	0.391699	0.435635	0.450379	0.465145	0.472496	0.492033	0.489109
2020-01-06	-0.0393398	-0.0394882	-0.0379981	-0.0367629	-0.0386488	-0.0371727	-0.0348792	-0.037448	-0.038761	-0.0406511	-0.0417432	-0.0450666	-0.0455323
2020-01-07	-0.0936467	-0.0919924	-0.0922485	-0.0930407	-0.096889	-0.098773	-0.0925389	-0.101737	-0.104672	-0.108015	-0.109774	-0.113678	-0.113683
2020-01-08	-0.20402	-0.198703	-0.196862	-0.195529	-0.209721	-0.211211	-0.190762	-0.216389	-0.224633	-0.232094	-0.235522	-0.245411	-0.242777
2020-01-09	0.0582445	0.0565084	0.0556761	0.054989	0.0604817	0.0623319	0.0533836	0.0664213	0.0726213	0.0792875	0.0853897	0.0946281	0.0998048

Result

The daily rates of return of each bond shown in figure 6, and conversion factor adjusted returns are in figure 7.

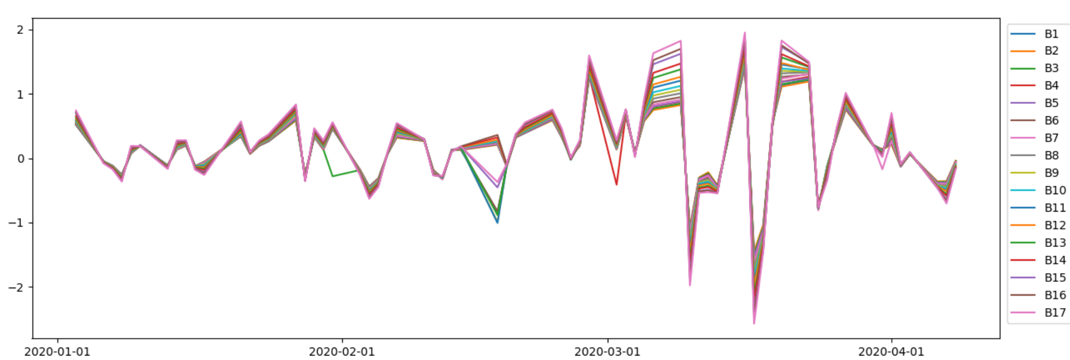

Figure 6: Daily rate of return of each bond

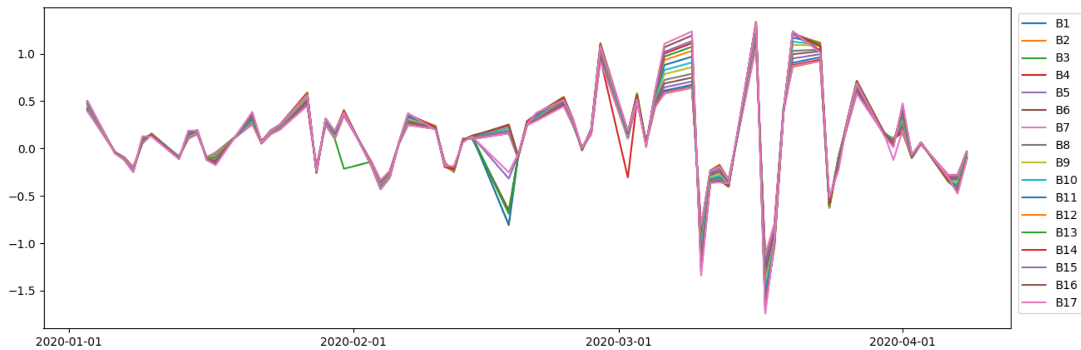

Figure 7: Conversion factor adjusted returns

C)

Compare and comment on the conversion factor adjusted return of the above bonds.

Conversion factor adjusted price matters because it determines the Cheapest-to-deliver bond which can make an impact on the quote of the US Treasury bond Futures. So the conversion factor adjusted return should be monitored carefully in case that current Cheapest-to-deliver bond changes.

In general, conversion factor adjusted return of all bonds move at almost the same direction and degree. However, because a bond's return will be negative at its coupon date after paying the coupon, its conversion factor adjusted return will deviate from others. Like B3 on Jan 31, 2020 and B4 on Feb 29, 2020. And another exception happens when aggressive monetary policy is implemented. Because Federal Reserve cut Fed Fund rates on Mar 5, 2020 and Mar 15, 2020. Bond conversion factor adjusted returns faced with dramatically changed yield curve deviate from each other. At these times, an obvious pattern can be found that the longer maturity a bond has, the more volatile its conversion factor adjusted return will be.