Fixed Income HW3

YiTao Hu, Jin (Jane) Huangfu, Junyu (Kevin) Wu, Charles Rambo

24/01/2020

Question 1

Use the zero-coupon curve you created in Homework 2 to compute the par rates for semiannual pay bonds with maturities ranging from 1 year to 25 years.

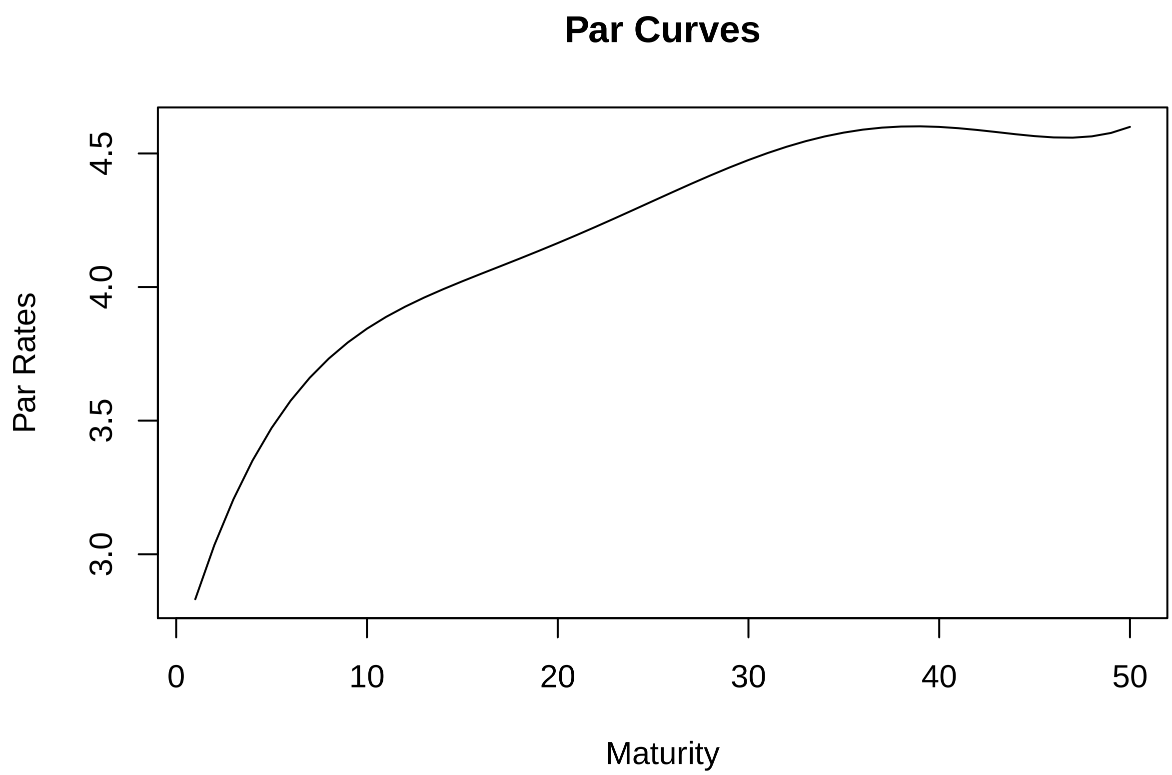
```r
#import packages
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 3.5.2
```

```r
library(derivmkts)
```

```
## Warning: package 'derivmkts' was built under R version 3.5.2
```

```r
#import the data
ZCB_data=read_excel("~/Desktop/UCLAMFE/Fixed Income/week3/Hw3data_2020.xlsx")
# drop the nas
ZCB_data=na.omit(ZCB_data)
# compute the semi-annual par rates from the discount factors
ZCB_data$parRates=(2*((100-100*ZCB_data$`D(T)`))/cumsum(ZCB_data$`D(T)`))
#plot the par curves
plot(ZCB_data$parRates,xlab = 'Maturity',ylab = 'Par Rates',
     main = 'Par Curves',type = 'l')
```

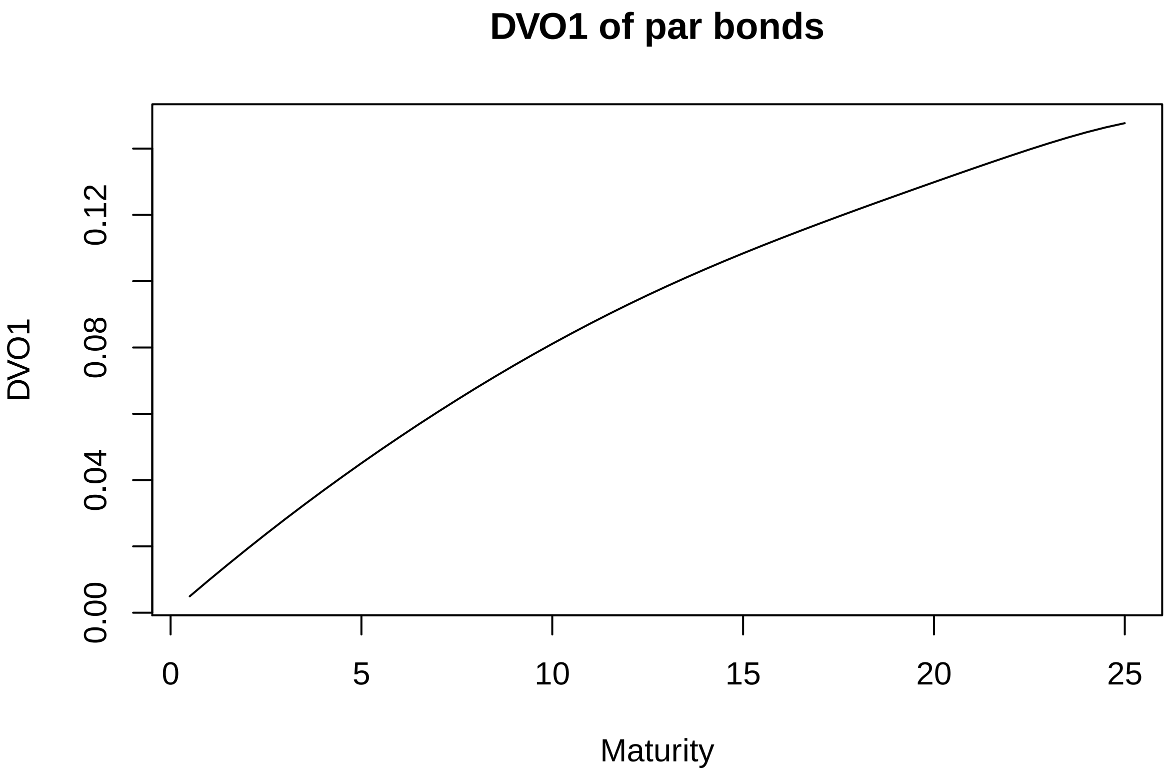


Question 2

For each of these bonds, compute their DV01.

```
ZCB_data$MacD=0 #initialization
#compute the Maculey Duration for the bond
for (i in 1:nrow(ZCB_data)){
  #compute the Maculey Duration for the bond
  ZCB_data$MacD[i]=duration(price = 100,coupon = ZCB_data$parRates[i],mat = ZCB_data$T[i],principal = 1
}

#compute the modified duration of the bond
ZCB_data$ModiD=ZCB_data$MacD/(1+ZCB_data$parRates/200)
#compute DV01
ZCB_data$DV01=ZCB_data$ModiD*100*0.0001
#plot the DV01 of the par bonds
plot(x=seq(0.5,25,0.5),y = ZCB_data$DV01,xlab = 'Maturity',ylab = 'DV01',main = 'DV01 of par bonds',typ
```



So, the DV01 of the par bonds are:

```
#initialize the result dataframe
output=data.frame(matrix(data = 0,nrow = nrow(ZCB_data),ncol = 1))
row.names(output)=seq(0.5,25,0.5)
colnames(output)=c('DV01')
output$DV01=ZCB_data$DV01
output
```

```
##            DV01
## 0.5  0.004930198
## 1    0.009776975
## 1.5  0.014531721
## 2    0.019189658
```

```
## 2.5  0.023748833
## 3    0.028209255
## 3.5  0.032572190
## 4    0.036839599
## 4.5  0.041013706
## 5    0.045096675
## 5.5  0.049090391
## 6    0.052996317
## 6.5  0.056815429
## 7    0.060548205
## 7.5  0.064194663
## 8    0.067754439
## 8.5  0.071226884
## 9    0.074611195
## 9.5  0.077906549
## 10   0.081112242
## 10.5 0.084227833
## 11   0.087253268
## 11.5 0.090189005
## 12   0.093036107
## 12.5 0.095796323
## 13   0.098472137
## 13.5 0.101066790
## 14   0.103584283
## 14.5 0.106029319
## 15   0.108407263
## 15.5 0.110724028
## 16   0.112985952
## 16.5 0.115199643
## 17   0.117371795
## 17.5 0.119508971
## 18   0.121617361
## 18.5 0.123702507
## 19   0.125768993
## 19.5 0.127820102
## 20   0.129857437
## 20.5 0.131880506
## 21   0.133886216
## 21.5 0.135868432
## 22   0.137817379
## 22.5 0.139719084
## 23   0.141554781
## 23.5 0.143300326
## 24   0.144925657
## 24.5 0.146394356
## 25   0.147663373
```

Question 3

Compute the Macauley and modified durations for the 1, 2, 3, 4, and 5 year bonds in question 1 above.

The Macauley and modified durations for 1,2,3,4,5 year bonds are:

```r
#initialize the result dataframe
output=data.frame(matrix(data = 0,nrow = 5,ncol = 2))
```

```r
row.names(output)=seq(1,5,1)
colnames(output)=c('Macauley Duration','Modified Duration')
output$`Macauley Duration`=ZCB_data$MacD[seq(2,10,2)]
output$`Modified Duration`=ZCB_data$ModiD[seq(2,10,2)]
output
```

```
##   Macauley Duration Modified Duration
## 1         0.9925286         0.9776975
## 2         1.9511125         1.9189658
## 3         2.8713454         2.8209255
## 4         3.7527086         3.6839599
## 5         4.5963451         4.5096675
```

Question 4

You have a \$5,000,000 liability due in 3 years. How much do you need to invest in a 3 year zero-coupon bond to defease the liability? Use the same zero-coupon curve as in 1.

To have \$5,000,000 cash flows 3 years later, we need to invest the present value of this amount in 3-year the zero-coupon Bond, which is:

$$5,000,000 \times D_3 =$$

```r
5000000*ZCB_data$`D(T)`[6]
```

```
## [1] 4494051
```

Question 5

Using the data in question 1, compute the convexities of the 1, 2, 3, 4, and 5 year bonds.

```r
ZCB_data$Convex=0 #initialization
#compute the Convexity for the par bonds
for (i in 1:nrow(ZCB_data)){
  ZCB_data$Convex[i]=convexity(price = 100,coupon = ZCB_data$parRates[i],mat = ZCB_data$T[i],principal =
}
#initialize the result dataframe
output=data.frame(matrix(data = 0,nrow = 5,ncol = 1))
row.names(output)=seq(1,5,1)
colnames(output)=c('Convexity')
output$Convexity=ZCB_data$Convex[seq(2,10,2)]
output
```

```
##   Convexity
## 1  1.441007
## 2  4.679181
## 3  9.556696
## 4 15.923195
## 5 23.640910
```

Question 6

Use the computed dollar durations and convexities for the 1, 2, 3, 4, and 5 year bonds, compute the price change of a 100 basis point upward and downward parallel shift in the zero-curve. Compare the price changes with the actual price change obtained by recomputing the price of the bond from the shifted spot curve.

The approximate price change given by duration and given by the following formula:

$$\Delta P = -DV01 \times 10,000 \times \Delta y + \frac{1}{2} \times Convex \times \Delta y^2 \times P$$

```
# grab related information
output=data.frame(matrix(data = 0,nrow = 5,ncol = 2))
row.names(output)=seq(1,5,1)
colnames(output)=c('DV01','Convexity')
output$DV01=ZCB_data$DV01[seq(2,10,2)]
output$Convexity=ZCB_data$Convex[seq(2,10,2)]
output$Coupon=ZCB_data$parRates[seq(2,10,2)]
output$mat=ZCB_data$T[seq(2,10,2)]
output$Price_ini=100
# compute approx price change upward and downward 100  bp
output$P_Approx_up=-output$DV01*10000*0.01+0.5*output$Convexity*(0.01)^2*100
output$P_Approx_down=-output$DV01*10000*(-0.01)+0.5*output$Convexity*(-0.01)^2*100
# compute the real bond price when the parallel change in rates
for (i in 1:nrow(output))
  {#compute the actual price change when rates go up
  output$act_P_up[i]=bondpv(coupon = output$Coupon[i],mat = output$mat[i],yield = (output$Coupon[i]/100
  # compute the actual price change when rates go down
  output$act_P_down[i]=bondpv(coupon = output$Coupon[i],mat = output$mat[i],yield = (output$Coupon[i]/1
}
```

The approximate price changes and actual price changes are as followed.

```
output[,6:9]
```

```
##   P_Approx_up P_Approx_down   act_P_up act_P_down
## 1  -0.9704925     0.9849026 -0.9705395  0.9849501
## 2  -1.8955699     1.9423617 -1.8957971  1.9425928
## 3  -2.7731420     2.8687090 -2.7737565  2.8693372
## 4  -3.6043439     3.7635759 -3.6056153  3.7648818
## 5  -4.3914630     4.6278721 -4.3937133  4.6301947
```