**Libor Interest Rate Swap Pricing using R code**

**Swap Pricing**

A fixed versus floating interest rate swap exchanges a stream of cash flows determined by 1) a predetermined fixed rate and 2) floating rates which will be determined periodically in the future respectively. Since we don’t know exactly the evolution of floating rates in the future, forward rate is used for the alternative variable coupon rate, which is implied in the market forward looking information. Market participants expect the forward rate to be the expected future rate in the perspective of fair pricing.   
  
It is worth noting that floating rates are determined at refixing dates before its corresponding interest periods in advance. Whenever we price a swap, its first variable cash flow is always known but remaining cash flows are unknown. For this unknown future variable rates, we use forward rates for its corresponding interest periods, which are implied in the current market yield curve.   
  
Swap pricing is to calculate the net present value (NPV), which is the difference between the sum of present values of fixed legs and floating legs. When a swap participant receives fixed and pays floating cash flows, his/her swap value at time t is   
  
\[\begin{align} NPV(t) & = \underbrace{\sum\_{i=1}^{n\_i} {CF\_{t\_i}^{fixed} DF^{libor}(t,t\_i)}}\_{\text{PV of fixed CFs}} \\ & – \underbrace{\sum\_{j=1}^{n\_j} {CF\_{t\_j}^{float} DF^{libor}(t,t\_j)}}\_{\text{PV of floating CFs}} \\ \end{align}\] \[\begin{align} & DF^{libor} = \text{Libor discount factor}\\ & DF^{libor}(t,t\_i) = DF^{libor} \text{ from } t\_i \text{ to } t \text{ for the fixed leg} \\ & DF^{libor}(t,t\_j) = DF^{libor} \text{ from } t\_j \text{ to } t \text{ for the floating leg} \\ & t\_i = i \text{-th payment date for the fixed leg}, i=1,2,…, n\_i \\ & t\_j = j \text{-th payment date for the floating leg}, j=1,2,…, n\_j \\ & s = \text{spot date} \end{align}\]

**Cash Flows**

Since discount factor is the market information, we need to calculate a stream of cash flows of two legs (NA = notional amount).

**Fixed leg**

\[\begin{align} CF\_{t\_i}^{fixed} = \underbrace{ \underbrace{ \underbrace{\text{C}} \_{\text{coupon rate}} \times \tau(t\_{i-1},t\_i) } \_{\text{semi-annual fixed coupon rate}} \times NA} \_{\text{semi-annual fixed coupon amount}} \\ \end{align}\] \[\begin{align} C &= \text{fixed rate} \\ \tau(t\_{i-1},t\_i) &= \text{day fraction(30I/360)} \\ &= (360 × \Delta Year + 30 × \Delta Month + \Delta Day) / 360 \end{align}\]

**Floating leg**

\[\begin{align} CF\_{t\_j}^{float} = \underbrace{ \underbrace{ \underbrace{FD^{libor}(t, t\_{j-1},t\_j)} \_{\text{forward rate}} \times \tau(t\_{j-1},t\_j) } \_{\text{quarterly variable coupon rate}} \times NA} \_{\text{quarterly variable coupon amount}} \\ \end{align}\] \[\begin{align} FD^{libor}(t, t\_{j-1},t\_j) &= \text{forward rate between } t\_{j-1} \text{ and } t\_j \\ &\quad \text{ implied in the time } t \text{ Libor curve} \\ \tau(t\_{j-1},t\_j) &= \text{day fraction(ACT/360)} \\ &= \text{actual days in-between} / 360 \end{align}\]

**Discount Factor and Forward Rate**

For the IRS swap pricing to be completed, discount factors and forward rates are needed to be calculated in the following way.

**Discount Factor at time t**

\[\begin{align} & DF^{libor}(t,t\_i) = \exp \left(-R^{libor}(t,t\_i) \times \frac{t\_i – t}{365} \right) \\ \\ & R^{libor}(t,t\_i) = \text{zero rate from } t\_i \text{ to } t \text{ implied in the Libor curve} \end{align}\]

**Forward Rate at time t**

\[\begin{align} FD^{libor}(t, t\_{j-1},t\_j) = \frac{365}{t\_j – t} \times \left(\frac{DF^{libor}(t,t\_{j-1})}{DF^{libor}(t,t\_j)}-1 \right) \end{align}\]

**IRS Specification**

As of 2021/06/30, consider the following 5-year IRS (Pay Float & Rec Fixed) for Libor 3M index with market information (swap rates and zero curve), which are from the Bloomberg.

As can be seen the above table, the fixed coupon rate of the fixed leg is 0.96495 which is the 5-year market swap rate as definition. Payment frequency and day count convention are different between the fixed leg and float leg. This specification is not absolute but chosen conventionally. Furthermore, there are many kinds of day count conventions. For more information about the day count conventions and swap specifications, you can find lots of wep pages.   
  
Before going to the calculation, we typically need to determine a series of 6 dates, which are dates for Interest Begin, Interest End, Accrual Begin, Accrual End, Reset Date, Payment Date. Determination of these dates requires market conventions and is somewhat complicated and important. In principal, the general swap pricing encompasses the determination of these dates.   
  
However, since most in-house pricing system or Bloomberg or Reuter provide that information, we can calculate the following swap cash flow schedules and the net present value (NPV) with these dates. This topic will be discussed in the later post. This time, **we assume away them by using the Bloomberg information** regarding cash flow schedules (paymemt dates) and zero rate curve.   
  
We try to price the 5-year IRS at spot date (s; **two business days from the trade date**). Therefore, pricing formula is as follows.   
\[\begin{align} & NPV(s) = \sum\_{i=1}^{n\_i} {CF\_{t\_i}^{fixed} DF^{libor}(s, t\_i)} -\sum\_{j=1}^{n\_j} {CF\_{t\_j}^{float} DF^{libor}(s, t\_j)} \\ \\ & s = \text{spot date} \end{align}\]   
Price of IRS at spot date is zero because there is no profit or loss between two counterparties at inception. Therefore, we can validate our swap pricing model by investigating **whether the swap price at the spot date is zero or not**.   
  
In real pricing, we use **linearly interpolated zero curve** since payment dates do not coincide with dates of market zero rate curve.

**R code**

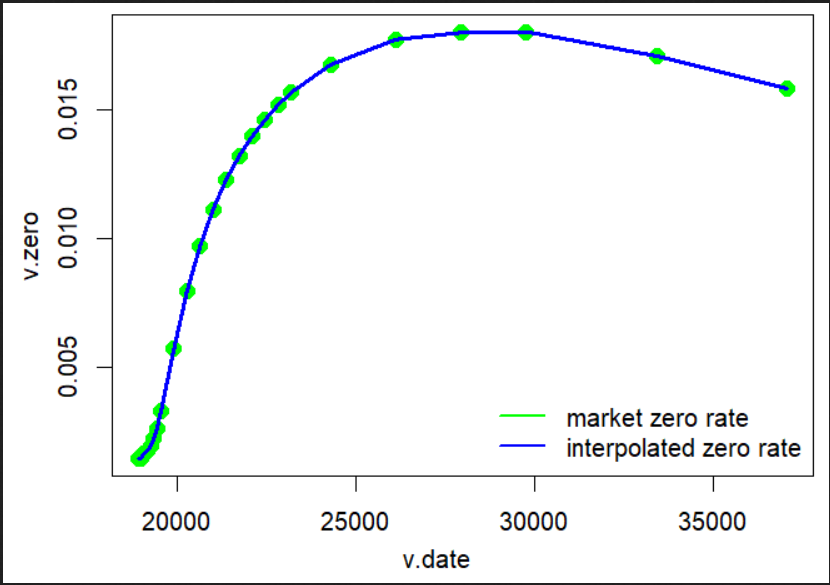
The following R code implements 5-year LIBOR 3M IRS pricing with the curve date of 2021/06/30 and the spot date of 2021/07/02.

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184 | #=========================================================================#  # Financial Econometrics & Derivatives, ML/DL using R, Python, Tensorflow  # by Sang-Heon Lee  #  # <https://kiandlee.blogspot.com>  #————————————————————————-#  # Libor 5-year fixed versus floating IRS Pricing  #=========================================================================#    graphics.off()  # clear all graphs  rm(list = ls()) # remove all files from your workspace    #————————————————————————–  # 1. Market Information  #————————————————————————–    # Zero curve from Bloomberg as of 2021-06-30  df.zero <– data.frame(      date = c(“2021-10-04”,“2021-12-15”,“2022-03-16”,“2022-06-15”,               “2022-09-21”,“2022-12-21”,“2023-03-15”,“2023-07-03”,               “2024-07-02”,“2025-07-02”,“2026-07-02”,“2027-07-02”,               “2028-07-03”,“2029-07-02”,“2030-07-02”,“2031-07-02”,               “2032-07-02”,“2033-07-05”,“2036-07-02”,“2041-07-02”,               “2046-07-02”,“2051-07-03”,“2061-07-05”,“2071-07-02”),        rate = c(0.00147746193495074, 0.00144337757980778,               0.00166389741542625,0.00175294804717070,0.00196071374597585,               0.00224582504806747,0.00264462838911974,0.00328408008984121,               0.00571530169527018,0.00795496282359075,0.00970003866673104,               0.01113416387898720,0.01229010329346910,0.01320660291639990,               0.01396222829363160,0.01461391064905110,0.01518876914165160,               0.01567359620429550,0.01673867348140660,0.01771539830734830,               0.01798302077085120,0.01801516858533200,0.01707008589009480,               0.01580574448899780      )  )    #————————————————————————–  # 2. Libor Swap Specification  #————————————————————————–    spot\_date\_ymd  <– as.Date(“2021-07-02”)   # spot date    no\_amt     <– 10000000      # notional amount  fixed\_rate <– 0.0096495    # cf\_scedule from Bloomberg  lt.cf\_date <– list(      fixed = c(“2022-01-04”,“2022-07-05”,“2023-01-03”,“2023-07-03”,                “2024-01-02”,“2024-07-02”,“2025-01-02”,“2025-07-02”,                “2026-01-02”,“2026-07-02”),      float = c(“2021-10-04”,“2022-01-04”,“2022-04-04”,“2022-07-05”,                “2022-10-03”,“2023-01-03”,“2023-04-03”,“2023-07-03”,                “2023-10-02”,“2024-01-02”,“2024-04-02”,“2024-07-02”,                “2024-10-02”,“2025-01-02”,“2025-04-02”,“2025-07-02”,                “2025-10-02”,“2026-01-02”,“2026-04-02”,“2026-07-02”)  )    #————————————————————————–  # 3. Swap Pricing – Preprocessing  #————————————————————————–    # spot date as serial number  spot\_date <– as.numeric(as.Date(spot\_date\_ymd))    # Interpolation of zero curve  v.date   <– as.numeric(as.Date(df.zero$date))  v.zero   <– df.zero$rate  f\_linear <– approxfun(v.date, v.zero, method=“linear”)  v.date.inter <– spot\_date:max(v.date)  v.zero.inter <– f\_linear(v.date.inter)    # Figures for zero curve  x11(width=6, height=5);  plot(v.date, v.zero, type = “b”, col = “green”, pch = 16, cex = 1.5)  lines(v.date.inter, v.zero.inter, col = “blue”, type=“l”, lwd = 3)  legend(“bottomright”,         legend = c(“market zero rate”, “interpolated zero rate”),         col = c(“green”, “blue”), lty = 1, bty = “n”, lwd = 2)    # number of CFs  ni <– length(lt.cf\_date$fixed)  nj <– length(lt.cf\_date$float)    # output dataframe with CF dates and its interpolated zero  df.fixed = data.frame(ymd = as.Date(lt.cf\_date$fixed),                        date    = as.numeric(as.Date(lt.cf\_date$fixed)))    df.float = data.frame(ymd = as.Date(lt.cf\_date$float),                        date = as.numeric(as.Date(lt.cf\_date$float)))    #————————————————————————–  # 4. Swap Pricing – Calculation  #————————————————————————–    #———————————————————-  #  1)  Fixed Leg  #———————————————————-  # zero rate for discounting  df.fixed$zero\_DC = f\_linear(df.fixed$date)    # discount factor  df.fixed$DF <– exp(–df.fixed$zero\_DC\*(df.fixed$date–spot\_date)/365)    # tau, CF  for(i in 1:ni) {        ymd      <– df.fixed$ymd[i]      ymd\_prev <– df.fixed$ymd[i–1]      if(i==1) ymd\_prev <– spot\_date\_ymd        d <– as.numeric(strftime(ymd, format = “%d”))      m <– as.numeric(strftime(ymd, format = “%m”))      y <– as.numeric(strftime(ymd, format = “%Y”))        d\_prev <– as.numeric(strftime(ymd\_prev, format = “%d”))      m\_prev <– as.numeric(strftime(ymd\_prev, format = “%m”))      y\_prev <– as.numeric(strftime(ymd\_prev, format = “%Y”))        # 30I/360      tau <– (360\*(y–y\_prev) + 30\*(m–m\_prev) + (d–d\_prev))/360        # cash flow rate      df.fixed$rate[i] <– fixed\_rate        # Cash flow at time ti      df.fixed$CF[i] <– fixed\_rate\*tau\*no\_amt # day fraction  }    # Present value of CF  df.fixed$PV = df.fixed$CF\*df.fixed$DF      #———————————————————-  #  2)  Floating Leg  #———————————————————-    # zero rate for discounting  df.float$zero\_DC = f\_linear(df.float$date)    # discount factor  df.float$DF <– exp(–df.float$zero\_DC\*(df.float$date–spot\_date)/365)    # tau, forward rate, CF  for(i in 1:nj) {        date      <– df.float$date[i]      date\_prev <– df.float$date[i–1]        DF        <– df.float$DF[i]      DF\_prev   <– df.float$DF[i–1]        if(i==1) {          date\_prev <– spot\_date          DF\_prev   <– 1      }        # ACT/360      tau <– (date – date\_prev)/360        # forward rate      fwd\_rate <– (1/tau)\*(DF\_prev/DF–1)        # cash flow rate      df.float$rate[i] <– fwd\_rate        # Cash flow amount at time ti      df.float$CF[i] <– fwd\_rate\*tau\*no\_amt # day fraction  }    # Present value of CF  df.float$PV = df.float$CF\*df.float$DF    #———————————————————-  #  3)  Swap Price at spot date  #———————————————————-    df.fixed[,–2]  df.float[,–2]  print(paste0(“Fixed Leg = “, round(sum(df.fixed$PV),6)))  print(paste0(“Float Leg = “, round(sum(df.float$PV),6)))  print(paste0(“Swap Price at spot date = “,               round(sum(df.fixed$PV) – sum(df.float$PV),6)))    [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

**Results**

The following figure draws the **market zero rate curve** (Bloomberg) and the **linearly interpolated zero rate curve** (from **approxfun()** R function) at 2021/06/30.

The following results indicate that the swap price is $2.719318. We expect this price to be $0 but cumulated numerical errors or unknown aspects of interpolation make this difference. But this is considered a zero value swap because the price ratio against nominal amount is 2.719318/10000000 = 0.00000027, which is nearly zero in the view of the numerical calculation.



The following results indicate that the swap price is $2.719318. We expect this price to be $0 but cumulated numerical errors or unknown aspects of interpolation make this difference. But this is considered a zero value swap because the price ratio against nominal amount is 2.719318/10000000 = 0.00000027, which is nearly zero in the view of the numerical calculation.

