MGMTMFE 405 - Project 9

Yanxiang Zhao

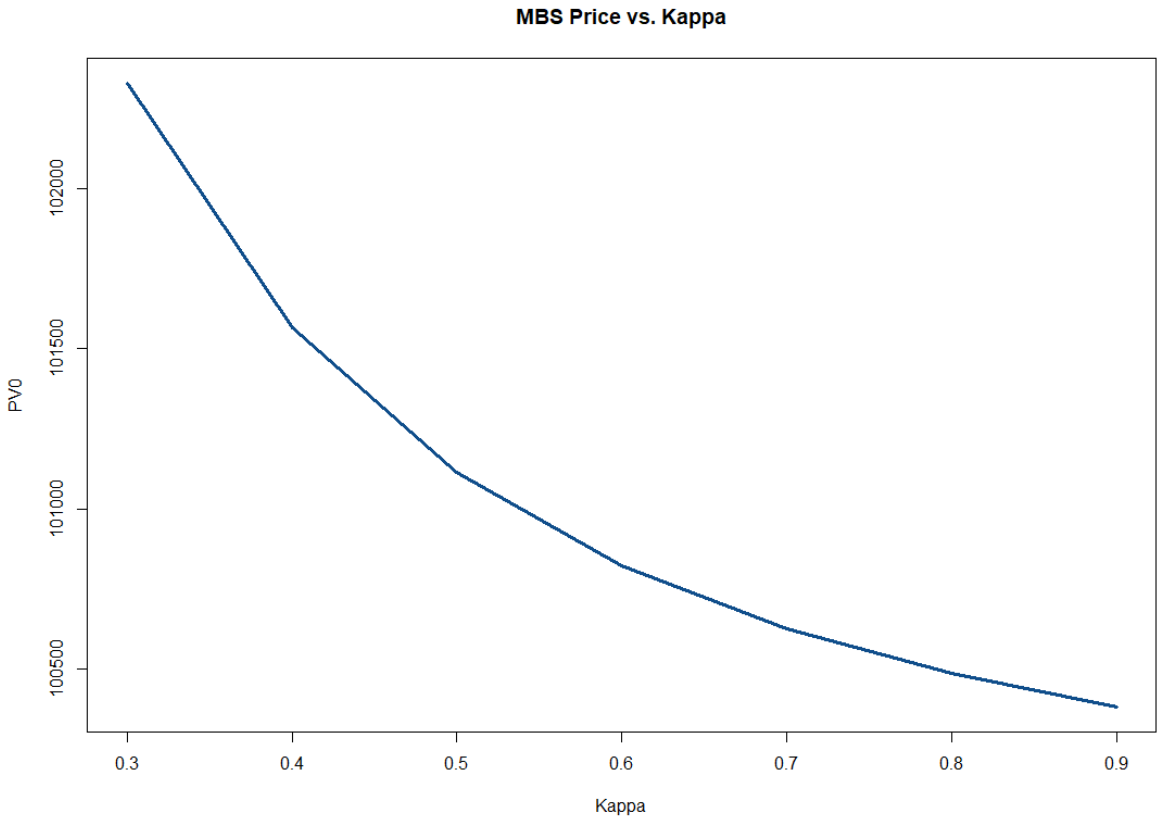
March 14, 2019

1. Consider the Numerix Prepayment Model

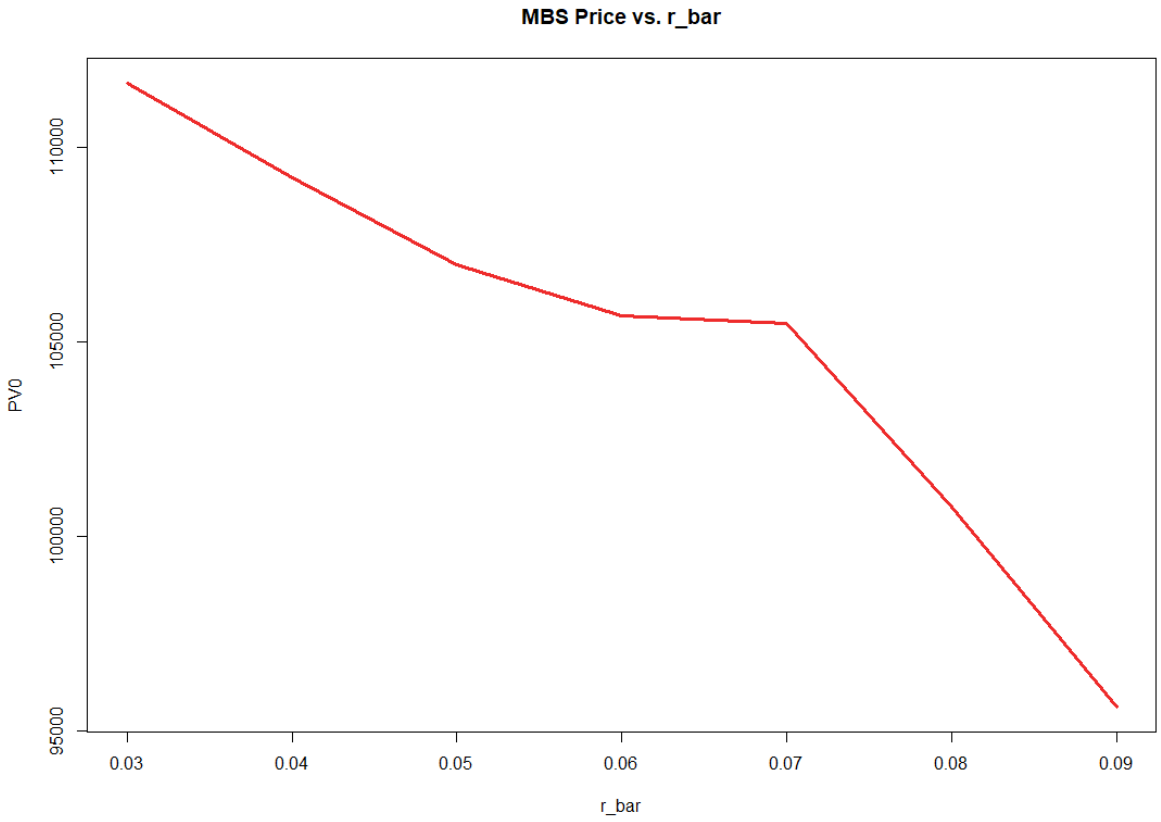
(a) Compute the price of the MBS using this model for prepayments

```
> source("promptMbs.R")
[1] "please enter the following inputs for the MBS: "
The notional amount of the loan is ($): 100000
The weighted average coupon is (%): 8
The maturity of the loan is (years): 30
The initial interest rate is (%): 7.8
The average interest rate is (%): 8
The annual volatility is (%): 12
The mean reversion coefficient is: .6
The number of simulations to run: 10000
The price of the MBS is: $100735.4
> 
```

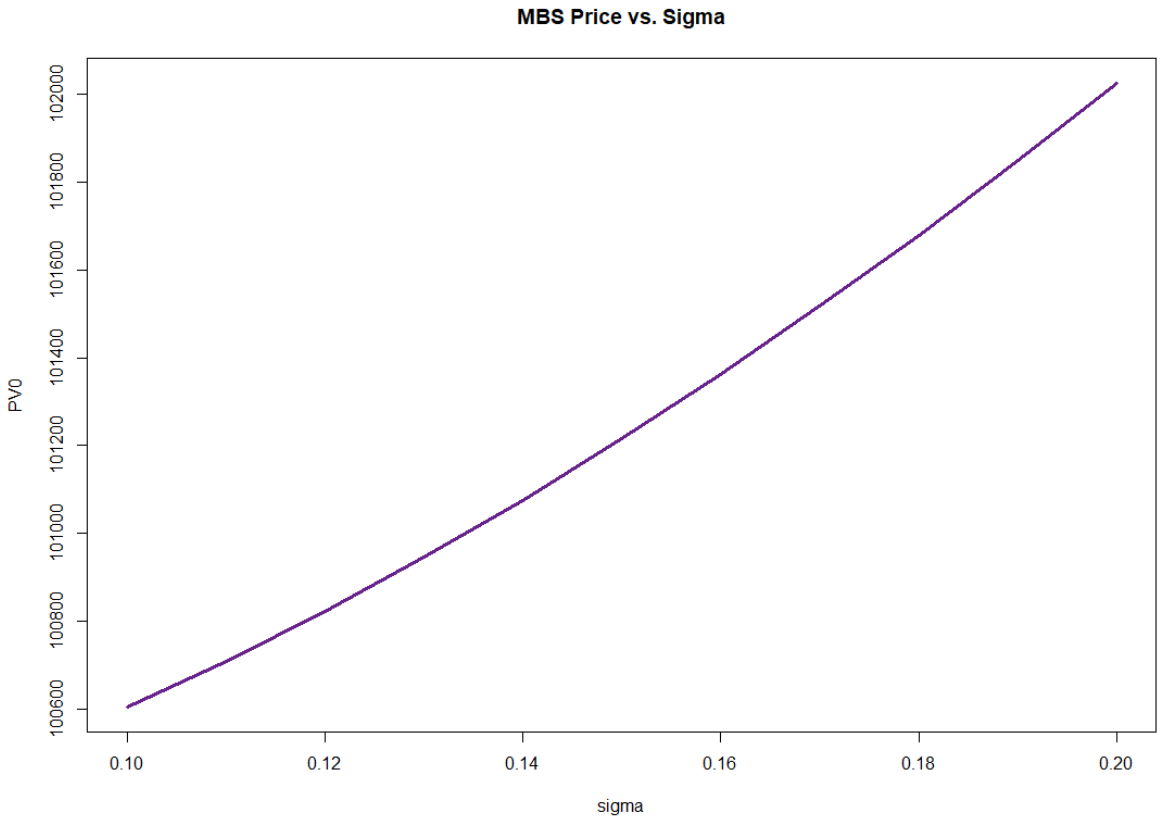
(b) Compute the price of the MBS for the following range of kappa: in [0.3, 0.9] range, in increments of 0.1. Draw the graph of the price with respect to kappa.



(c) Compute the price of the MBS for the following range of r_bar: in [0.03, 0.09] range, in increments of 0.01. Draw the graph of the price with respect to r_bar.



(d) Compute the price of the MBS for the following range of sigma: in [0.10, 0.20] range, in increments of 0.01. Draw the graph of the price with respect to sigma.



2. Compute the Option-Adjusted-Spread (OAS) for the Numerix-Prepayment Model above with the Market Price of MBS being $110,000.

The OAS is about -0.01175835.

3. Compute the OAS-adjusted Duration and Convexity of the MBS, considered in the previous question.

The change in OAS is set to be 50 bps.
The OAS-adjusted Duration is 7.625456.
The OAS-adjusted Convexity is 44.47026.

Code

Main

```
# Zhao_Yanxiang_Project6
setwd("C:/Users/harry/OneDrive/Documents/GitHub/MGMTMFE405-Comutaional-Methods/Project9/")
source('abs.R')
##### Question 1 #####
# a)
source("promptMbs.R")

# b) for different kappa
kappa <- seq(0.3, 0.9, 0.1)
q1b <- c()
for (k in kappa) {
  value <- findMBS(kappa=k)
  q1b <- c(q1b, value)
}
plot(x= kappa, y = q1b, type = "l", lwd = 3, col = "dodgerblue4",
     main = "MBS Price vs. Kappa", ylab = "PV0", xlab = "Kappa")
# c) for different r_bar
r_bar <- seq(0.03, 0.09, 0.01)
q1c <- c()
for (rbar in r_bar) {
  value <- findMBS(r_bar = rbar)
  q1c <- c(q1c, value)
}
plot(x= r_bar, y = q1c, type = "l", lwd = 3, col = "firebrick2",
     main = "MBS Price vs. r_bar", ylab = "PV0", xlab = "r_bar")
# d) for different sigma
sig <- seq(0.1, 0.2, 0.01)
q1d <- c()
for (sigma in sig) {
  value <- findMBS(sig = sigma, nSim=200000)
  q1d <- c(q1d, value)
}
plot(x= sig, y = q1d, type = "l", lwd = 3, col = "darkorchid4",
     main = "MBS Price vs. Sigma", ylab = "PV0", xlab = "Sigma")


##### Question 2 #####
rPaths <- cirPath(r0=.078, sig=.12, kappa=.6, r_bar=.08,t = 40, nSim = 30000)
x <- uniroot(fitOAS, r_t = rPaths, lower = -0.05, upper = 0)$root

##### Question 3 #####
y <- 0.0005
p0 <- 110000
p_plus <- fitOAS(x+y, rPaths, expPv = 0)
p_minus <- fitOAS(x-y, rPaths, expPv = 0)

duration <- (p_minus-p_plus)/(2*y*p0)
convexity <- (p_plus+p_minus-2*p0)/(2*p0*y^2)
```

abs.R

```
cirPath <- function(r0, sig, kappa, r_bar, t, nSim){
  dt <- 1/12
  n <- t/dt+1
  sim <- list()
  for (i in 1:(nSim/2)){
    dw <- sqrt(dt)*rnorm(n-1)
    r_sim1 <- rep(r0,n)
    r_sim2 <- rep(r0,n)
    for (s in 2:n){
      # create anthithetic paths
      # use full truncation
      r_sim1[s] <- r_sim1[s-1] + kappa*(r_bar - ifelse(r_sim1[s-1]>0,r_sim1[s-1],0))*dt + sig*sqrt(ifels
      r_sim2[s] <- r_sim2[s-1] + kappa*(r_bar - ifelse(r_sim2[s-1]>0,r_sim2[s-1],0))*dt - sig*sqrt(ifels
    }
    sim_i <- list(r_sim1, r_sim2)
    sim <- append(sim, sim_i)
  }
  out <- matrix(unlist(sim), nSim, n, byrow = T)
  return(out)
}

findMBS <- function(pv0=100000, r0=.078, r_bar=.08, wac =.08, kappa=.6, sig=.12, years=30, nSim=30000){
  # function to find discount factor
  disFac <- function(r, t, n){
    factor <- 1/(1-(1+r)^(-n+(t-1)))
  }
  # function to find long-term interest rate
  bondRate <- function(paths){
    rate <- -1/(ncol(paths))*apply(paths*(1/12), 1, sum)
    return(rate)
  }

  # set constants
  r <- wac/12
  N <- years*12+1
  # find CPR_t related terms
  t_seq <- seq(0, years*12)
  SG_t <- ifelse(t_seq/30<1, t_seq/30, 1)
  SY_t <- rep(c(.94,.76,.74,.95,.98,.92,.98,1.1,1.18,1.22,1.23,.98), years)

  # generate interest rate paths
  r_t <- cirPath(r0, sig, kappa, r_bar, years+10, nSim)
  # set containers for PV_t and PV_CF_t
  PV_t <- cbind(rep(pv0,nSim), diag(0, nSim, N-1))
  PV_CF_t <- diag(0, nSim, N)
  for (s in 2:N){
    now <- s-1
    # find r_t-1(10)
    r10 <- bondRate(r_t[,now:(now+(10*12)-1)])
    RI_s <- 0.28+0.14*atan(-8.57 + 430*(wac - r10))
    BU_s <- 0.3+0.7*(PV_t[,s-1]/pv0)
```

```
    # find CPR_s
    CPR_s <- RI_s*BU_s*SG_t[now]*SY_t[now]
    # find cahs flow s
    dis_s <- disFac(r, now, N)
    SP_s <- PV_t[,s-1]*r*(dis_s-1)
    SMM_s <- 1-(1-CPR_s)^(1/12)
    IP_s <- PV_t[,s-1]*r
    TPP_s <- SP_s + (PV_t[,s-1] - SP_s) * SMM_s
    # update PV_t
    PV_t[,s] <- PV_t[,s-1] - TPP_s
    # update PV_CF_t
    r_0_s <- if (now == 1) matrix(r_t[,1], ncol = 1) else  r_t[,1:now]
    PV_CF_t[,s] <- (TPP_s + IP_s)*exp(-apply(r_0_s*(1/12), 1, sum))
  }
  P_0 <- mean(apply(PV_CF_t, 1, sum))
  return(P_0)
}

fitOAS <- function(x, r_t, pv0=100000, wac =.08, expPv = 110000){
  # function to find discount factor
  disFac <- function(r, t, n){
    factor <- 1/(1-(1+r)^(-n+(t-1)))
  }
  # function to find long-term interest rate
  bondRate <- function(paths){
    rate <- -1/(ncol(paths))*apply(paths*(1/12), 1, sum)
    return(rate)
  }

  # set constants
  r <- wac/12
  N <- ((ncol(r_t)-1)/12-10)*12+1
  nSim <- nrow(r_t)

  # find CPR_t related terms
  t_seq <- seq(0, N-1)
  SG_t <- ifelse(t_seq/30<1, t_seq/30, 1)
  SY_t <- rep(c(.94,.76,.74,.95,.98,.92,.98,1.1,1.18,1.22,1.23,.98), (N-1)/12)

  # add spread
  r_t <- r_t+x
  # set containers for PV_t and PV_CF_t
  PV_t <- cbind(rep(pv0,nSim), diag(0, nSim, N-1))
  PV_CF_t <- diag(0, nSim, N)
  for (s in 2:N){
    now <- s-1
    # find r_t-1(10)
    r10 <- bondRate(r_t[,now:(now+(10*12)-1)])
    RI_s <- 0.28+0.14*atan(-8.57 + 430*(wac - r10))
    BU_s <- 0.3+0.7*(PV_t[,s-1]/pv0)
    # find CPR_s
    CPR_s <- RI_s*BU_s*SG_t[now]*SY_t[now]
    # find cahs flow s
```

```
    dis_s <- disFac(r, now, N)
    SP_s <- PV_t[,s-1]*r*(dis_s-1)
    SMM_s <- 1-(1-CPR_s)^(1/12)
    IP_s <- PV_t[,s-1]*r
    TPP_s <- SP_s + (PV_t[,s-1] - SP_s) * SMM_s
    # update PV_t
    PV_t[,s] <- PV_t[,s-1] - TPP_s
    # update PV_CF_t
    r_0_s <- if (now == 1) matrix(r_t[,1], ncol = 1) else  r_t[,1:now]
    PV_CF_t[,s] <- (TPP_s + IP_s)*exp(-apply(r_0_s*(1/12), 1, sum))
  }
  P_0 <- mean(apply(PV_CF_t, 1, sum))
  return(P_0-expPv)
}
```

promptMbs.R

```
source('abs.R')

print("please enter the following inputs for the MBS: ")
pv0 <- as.numeric(readline("The notional amount of the loan is ($): "))
wac <- as.numeric(readline("The weighted average coupon is (%): "))/100
years <- as.numeric(readline("The maturity of the loan is (years): "))
r0 <- as.numeric(readline("The initial interest rate is (%): "))/100
r_bar <- as.numeric(readline("The average interest rate is (%): "))/100
sig <- as.numeric(readline("The annual volatility is (%): "))/100
kappa <- as.numeric(readline("The mean reversion coefficient is: "))
nSim <- as.numeric(readline("The number of simulations to run: "))
cat('The price of the MBS is: $')
cat(findMBS(pv0, r0, r_bar, wac, kappa, sig, years, nSim))
```