MGMTMFE 405 - Project 8

Yanxiang Zhao

March 8, 2019

Answers

For this project, I used 30/360 day-count convension. ## Problem 1: Vasicek Model

(a) Price of a pure discout bond

```
bond_1a
```

```
## [1] 975.9473
```

(b) Price of a coupon bond

```
bond_1b
```

```
## [1] 1081.822
```

(c) Price of an European call option on a pure discount bond

Calculated with explicit bond price.

```
opt_1c
```

```
## [1] 11.65509
```

(d) Price of an European call option on a coupon bond

Calculated by Monte Carlo simulations.

```
opt_1d
```

```
## [1] 159.9455
```

Problem 2: CIR Model

(a) Price of an European call option on a pure discount bond

```
opt_2a
```

```
## [1] 1.510533
```

(b) Explicit call opion price

```
opt_2b
```

```
## [1] 1.56044
```

Comment: The call option price calculated from the simulation in part (a) is slightly larger than the price calculated using the explicit formula. The difference is about 2.5% of the explicit formula price.

Problem 3: G2++ Model

Price of an European put option on a pure discount bond

```
opt_3
```

```
## [1] 13.34838
```

Main

```
# Zhao_Yanxiang_Projcet8
source('fixedIncome.R')
########################### MAIN ###########################
##### Question 1 #####
# set parameters
r0 <- .05
sig <- .18
kappa <- .82
r_bar <- .05
nSim <- 30000
fv <- 1000
# (a) pure discout bond
t <- 0.5
# generate paths
paths1a <- vasicekPath(r0, sig, kappa, r_bar, t, nSim)
# find pure discount bond price
bond_1a <- zcBond(paths1a, fv)
# (b) coupon paying bond
t <- 4
pmtT <- seq(0.5, t, 0.5)
c <- 30
# generate paths
```

```r
paths1b <- vasicekPath(r0, sig, kappa, r_bar, t, nSim)
# find pure discount bond price
bond_1b <- cpnBond(paths1b, pmtT, c, fv)
# (c) European Call option on pure discount bond
exerT <- .25
t <- .5
k <- 980
opt_1c <- zcOption_vasicek(paths1a, sig, kappa, r_bar, exerT, t, k, fv, 'call')
# (d) European Call option on coupon bond
exerT <- .25
t <- 4
k <- 980
c <- 30
mSim <- 100
opt_1d <- cpnOption(paths1b, sig, kappa, r_bar, exerT, t, k, c, fv, "call", "Vasicek", mSim)

##### Question 2 #####
# set parameters
r0 <- .05
sig <- .18
kappa <- .92
r_bar <- .055
nSim <- 1000
fv <- 1000
# (a) European Call option
exerT <- 0.5
t <- 1
k <- 980
mSim <- 100
paths2a <- cirPath(r0, sig, kappa, r_bar, t, nSim)
opt_2a <- zcOption(paths2a, sig, kappa, r_bar, exerT, t, k, fv, "call", "CIR", mSim)
# (b) Explicite solution
opt_2b <- zcCall_cir(r0, sig, kappa, r_bar, exerT, t, k, fv)
##### Question 3 #####
# set parameters
x0 <- y0 <- 0
r0 <- phi_t <- 0.03
rho <- 0.7
a <- 0.1
b <- 0.3
sig <- 0.03
eta <- 0.08
t <- 0.5
nSim <- 1000
# paths
paths3 <- g2ppPath(x0, y0, r0, phi_t, rho, a, b, sig, eta, t, nSim)
# price the put option
exerT <- 0.5
t <- 1
k <- 985
mSim <- 100
opt_3 <- zcOption_gcpp(paths3, x0, y0, r0, phi_t, rho, a, b, sig, eta, exerT, t, k, fv, "put", mSim)
```

fixedIncome.R

```
# Zhao_Yanxiang_Projcet8
########################### FUNCTIONS ###########################
##### PATH #####
# Vasicek model
vasicekPath <- function(r0, sig, kappa, r_bar, t, nSim){
  dt <- 1/360 # assume 30/360 conversion
  n <- t/dt
  sim <- list()
  for (i in 1:(nSim/2)){
    dw <- sqrt(dt)*rnorm(n)
    r_sim1 <- rep(r0,n)
    r_sim2 <- rep(r0,n)
    for (s in 2:n){
      # create anthithetic paths
      r_sim1[s] <- r_sim1[s-1] + kappa*(r_bar - r_sim1[s-1])*dt + sig*dw[s]
      r_sim2[s] <- r_sim2[s-1] + kappa*(r_bar - r_sim2[s-1])*dt - sig*dw[s]
    }
    sim_i <- list(r_sim1, r_sim2)
    sim <- append(sim, sim_i)
  }
  out <- matrix(unlist(sim), nSim, n, byrow = T)
  return(out)
}
# CIR model
cirPath <- function(r0, sig, kappa, r_bar, t, nSim){
  dt <- 1/360 # assume 30/360 conversion
  n <- t/dt
  sim <- list()
  for (i in 1:(nSim/2)){
    dw <- sqrt(dt)*rnorm(n)
    r_sim1 <- rep(r0,n)
    r_sim2 <- rep(r0,n)
    for (s in 2:n){
      # create anthithetic paths
      # use partial truncation
      r_sim1[s] <- r_sim1[s-1] + kappa*(r_bar - r_sim1[s-1])*dt + sig*sqrt(ifelse(r_sim1[s-1]>0,r_sim1[
      r_sim2[s] <- r_sim1[s-1] + kappa*(r_bar - r_sim2[s-1])*dt - sig*sqrt(ifelse(r_sim2[s-1]>0,r_sim2[
    }
    sim_i <- list(r_sim1, r_sim2)
    sim <- append(sim, sim_i)
  }
  out <- matrix(unlist(sim), nSim, n, byrow = T)
  return(out)
}
# G2++ model
g2ppPath <- function(x0, y0, r0, phi_t, rho, a, b, sig, eta, t, nSim){
  dt <- 1/360 # assume 30/360 conversion
  n <- t/dt
  sim <- list()
  for (i in 1:nSim){
    dw1 <- sqrt(dt)*rnorm(n)
```

```
    dw2 <- sqrt(dt)*rnorm(n)
    x_sim <- rep(x0,n)
    y_sim <- rep(y0,n)
    for (s in 2:n){
      # create anthithetic paths
      x_sim[s] <- x_sim[s-1]-a*x_sim[s-1]*dt+sig*dw1[s]
      y_sim[s] <- y_sim[s-1]-b*y_sim[s-1]*dt+eta*(rho*dw1[s]+sqrt(1-rho^2)*dw2[s])
    }
    sim_i <- c(r0, x_sim[2:n]+y_sim[2:n]+phi_t)
    sim <- append(sim, sim_i)
  }
  out <- matrix(unlist(sim), nSim, n, byrow = T)
  return(out)
}
##### BOND PRICING #####
# Zero-coupon bond
zcBond <- function(paths, fv){
  dt <- 1/360
  if (!is.null(ncol(paths))){
    pv <- fv*mean(exp(-apply(paths[,2:ncol(paths)]*dt, 1, sum)))
  } else{
    # deal with a single paths
    pv <- fv*mean(exp(sum(paths[2:length(paths)]*dt)))
  }
  return(pv)
}
# Coupon bond
cpnBond <- function(paths, pmtT, c, fv){
  pmtT <- pmtT*360
  pmtC <- c(rep(c, length(pmtT)-1), c+fv)
  pmtPV <- c()
  for (i in 1:length(pmtT)){
    pmtPV[i] <- zcBond(paths[,1:pmtT[i]],pmtC[i])
  }
  return(sum(pmtPV))
}
zcBond_cir <- function(r0, sig, kappa, r_bar, t, fv){
  h1 <- sqrt(kappa^2+2*sig^2)
  h2 <- (kappa+h1)/2
  h3 <- (2*kappa*r_bar)/sig^2
  A <- ((h1*exp(h2*t))/(h2*(exp(h1*t)-1)+h1))^h3
  B <- (exp(h1*t)-1)/(h2*(exp(h1*t)-1)+h1)
  return(fv*A*exp(-B*r0))
}
##### OPTION PRICING #####
# European option on ZC bond - vasicek
zcOption_vasicek <- function(paths, sig, kappa, r_bar, exerT, t, k, fv, type){
  dt <- 1/360
  exerT_day <- exerT*360+1
  rt <- paths[,exerT_day]
  B <- 1/kappa*(1-exp(-kappa*(t-exerT)))
  A <- exp((r_bar - sig^2/(2*kappa^2))*(B-(t-exerT)) - sig^2/(4*kappa)*B^2)
  bond_exerT <- fv*(A*exp(-B*rt))
```

```r
  payoff <- switch (type,
    "call" = ifelse(bond_exerT>k, bond_exerT-k, 0),
    "put"  = ifelse(bond_exerT<k, k-bond_exerT, 0)
  )
  option <- mean(payoff*exp(-apply(paths[,2:exerT_day]*dt, 1, sum)))
  return(option)
}
# European call option on ZC bond - CIR explicit
zcCall_cir <- function(r0, sig, kappa, r_bar, exerT, t, k, fv){
  h1 <- sqrt(kappa^2+2*sig^2)
  h2 <- (kappa+h1)/2
  h3 <- (2*kappa*r_bar)/sig^2
  A_TS <- ((h1*exp(h2*(t-exerT)))/(h2*(exp(h1*(t-exerT))-1)+h1))^h3
  B_TS <- (exp(h1*(t-exerT))-1)/(h2*(exp(h1*(t-exerT))-1)+h1)
  theta <- sqrt(kappa^2+2*sig^2)
  phi <- 2*theta/(sig^2*(exp(theta*t)-1))
  psi <- (kappa + theta)/sig^2
  r_star <- log(A_TS/(k/fv))/B_TS
  P_tS <- zcBond_cir(r0, sig, kappa, r_bar, t, fv)/fv
  P_tT <- zcBond_cir(r0, sig, kappa, r_bar, exerT, fv)/fv
  # find the call option price with chi-dist
  x1 <- 2*r_star*(phi + psi + B_TS)
  q1 <- (2*phi^2*r0*exp(theta*t))/(phi+psi+B_TS)
  p <- (4*kappa*r_bar)/sig^2
  x2 <- 2*r_star*(phi + psi)
  q2 <- (2*phi^2*r0*exp(theta*t))/(phi+psi)
  # call option price
  C <- fv*P_tS*pchisq(x1, p, q1)-k*P_tT*pchisq(x2, p, q2)
  return(C)
}
# European option on ZC bond
zcOption <- function(paths, sig, kappa, r_bar, exerT, t, k, fv, type, method, mSim){
  rt <- paths[,ncol(paths)]
  paths_before <- paths
  option <- c()
  for (i in 1:length(rt)){
    paths_after <- switch (method,
                           "Vasicek" = vasicekPath(rt[i], sig, kappa, r_bar, t-exerT, mSim),
                           "CIR" = cirPath(rt[i], sig, kappa, r_bar, t-exerT, mSim)
    )
    bond_exerT <- zcBond(paths_after, fv)
    payoff <- switch (type,
                      "call" = ifelse(bond_exerT>k, bond_exerT-k, 0),
                      "put"  = ifelse(bond_exerT<k, k-bond_exerT, 0)
    )
    option[i] <- zcBond(paths_before[i,], payoff)
  }
  return(mean(option))
}
# European option on coupon bond
cpnOption <- function(paths, sig, kappa, r_bar, exerT, t, k, c, fv, type, method, mSim){
  rt <- paths[,ncol(paths)]
  paths_before <- paths
```

```r
  option <- c()
  for (i in 1:length(rt)){
    paths_after <- switch (method,
      "Vasicek" = vasicekPath(rt[i], sig, kappa, r_bar, t-exerT, mSim),
      "CIR" = cirPath(rt[i], sig, kappa, r_bar, t-exerT, mSim)
    )
    pmtT <- seq(0.5, t, 0.5)-exerT # so far only for exerT < 0.5
    bond_exerT <- cpnBond(paths_after, pmtT, c, fv)
    payoff <- switch (type,
      "call" = ifelse(bond_exerT>k, bond_exerT-k, 0),
      "put"  = ifelse(bond_exerT<k, k-bond_exerT, 0)
    )
    option[i] <- zcBond(paths_before[i,], payoff)
  }
  return(mean(option))
}
# European opion on zc bond - G2++
zcOption_gcpp <- function(paths, x0, y0, r0, phi_t, rho, a, b, sig, eta, exerT, t, k, fv, type, mSim){
  rt <- paths[,ncol(paths)]
  paths_before <- paths
  option <- c()
  for (i in 1:length(rt)){
    paths_after <- g2ppPath(x0, y0, rt[i], phi_t, rho, a, b, sig, eta, t, mSim)
    bond_exerT <- zcBond(paths_after, fv)
    payoff <- switch (type,
                      "call" = ifelse(bond_exerT>k, bond_exerT-k, 0),
                      "put"  = ifelse(bond_exerT<k, k-bond_exerT, 0)
    )
    option[i] <- zcBond(paths_before[i,], payoff)
  }
  return(mean(option))
}
```