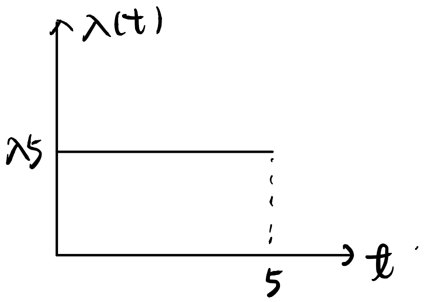
P8 Jiaqi Li & Xiangui Mei HW3

(a)



$S(5) = \ell \lambda_5$

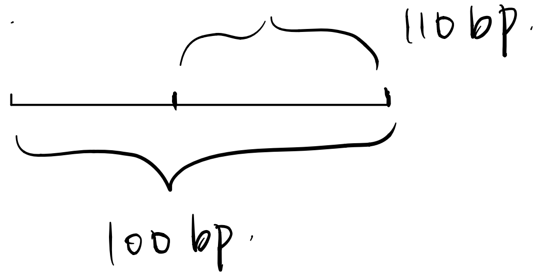
$100bp = \ell \lambda_5$

$\lambda_5 = 166.67 \text{ bps} = 0.01667$

(b) 1-year survival prob $= e^{-\lambda_5 \cdot 1} = 0.983$

1-year default prob $= 1-e^{-\lambda_5 \cdot 1} = 0.0165$

(c)



$S_1 = \ell \lambda_1$

$\lambda_1 = \frac{0.011}{0.6} = 0.018$

$$V_1^{fee} = \delta \sum_{i=1}^{n} p^d(0, t_j) = 0.5 \sum_{i=1}^{2} p^d(0, t_j)$$

$$= 0.5 \sum_{j=1}^{2} e^{-0.5\lambda_1} + e^{-\lambda_1} = 0.987$$

$$V_1^{port} = \ell \cdot Q(\tau \le 1) = 0.6(1-e^{-\lambda_1}) = 0.0107$$

$$\text{value} = V_1^{port} - S_0 V_1^{fee}$$

$$= 0.0107 - 0.01 \cdot 0.987$$

$$= 0.00083$$

(d) at maturity. the payoff is:

for protection leg: ①

$$E[L 1\{\tau \le T\} e^{-\int_0^\tau r_u du}] = E[L 1\{\tau \le T\}] = 0.6 \times Q[\tau \le T] = 0.6(1-e^{-\lambda})$$

for discrete coupon:

$$\hat{V}_{ofee} = \sum_{j=1}^{1} E\left[e^{-\int_0^{t_2} r_s ds} S_0 \, 1\{\tau > t_j\}\right] \quad ②$$

$$= S_0 \sum_{j=1}^{1} Q(\tau > t_j) = S_0 \left(e^{-\lambda}\right)$$

$S_0 = 110bp = 0.11$

$$① = ② \qquad 0.6(1-e^{-\lambda}) = 0.011(e^{-\lambda})$$

$$\lambda = 0.018167$$

P9

(a) CIR model: $d\lambda_t = k(\theta - \lambda_t)dt + \sigma\sqrt{\lambda_t}\, dW_t$

for affine models: $E(e^{-\int_0^t \lambda_u du}) = e^{A(t) - B(t)\lambda_0}$ $\quad \gamma = \sqrt{k^2 + 2\cdot\sigma^2} = 1.004$

$$A(t) = \frac{2k\theta}{\sigma^2} \ln\left(\frac{2\gamma e^{(k+\gamma)t/2}}{2\gamma + (k+\gamma)(e^{\gamma t}-1)}\right) \qquad B(t) = \frac{2(e^{\gamma t}-1)}{2\gamma + (k+\gamma)(e^{\gamma t}-1)}$$

(b) assume $R=0$

for defaultable bond:

$$P_d = E\left[e^{-\int_0^T r_s ds} 1\{\tau > T\}\right] = E\left[e^{-\int_0^T r_s ds}\right] E\left[1\{\tau > T\}\right]$$

$$= P(0,T)\, Q(\tau > T)$$

$$= P(0,T)\, E\left[e^{-\int_0^T \lambda_s ds}\right]$$

$$= P(0,T)\; e^{A(T) - B(T)\lambda_0}$$

$$= e^{-rT}\; e^{A(T) - B(T)\lambda_0}$$

$$= e^{-rT + A(T) - B(T)}$$

$r_d \cdot T = -\ln P \qquad r_d = -\ln P / T \qquad \text{Spread} = r_d - r$

After implementing in R, result:

```
1: 0.0136553716693598
2: 0.015616182001414644
3: 0.01673990210505128
4: 0.017428620113432013
5: 0.01787836270364123
6: 0.01818913790061906
7: 0.01841449600822616
8: 0.018584577332531806
9: 0.018717202680343187
10: 0.018823413011352354
```

(c). When $T \to 0$. $r_d \to r$

$\because$ when $T \to 0$ $A \to 0$ $B \to 0$

$\underset{T\to 0}{P_d} = e^{-rT}$ $\quad \underset{T \to 0}{r_d} = -\ln P_d / T = r$.

difference: for Merton model. it involves $N(d_1)$ $\quad d_1 = \dfrac{\ln(\frac{S}{K}) + (r+\frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}$

when $T \to 0$ and $t=0$. the denominator $\to 0$. d_1 will be undefined so Merton model can not have $T=0$.

(d). $R=0.5$ $\quad C=0.02$

$$P_c^d(0,T) = \sum_{k=1}^{2T} E\left[e^{-\int_0^{t_k} r_u du} C_k 1\{\tau > t_k\}\right]$$
$$+ E\left[e^{-\int_0^T r_u du} 1\{\tau > T\}\right]$$
$$+ E\left[e^{-\int_0^{\tau} r_s ds} R_\tau 1\{\tau \le T\}\right]$$
$$= \sum_{k=1}^{2T} C_k E\left[e^{-\int_0^{t_k} \lambda_s ds} \quad P(0,t_k)\right]$$
$$+ P(0,T)\, E\left[e^{-\int_0^T \lambda_s ds}\right]$$
$$+ R\int_0^T P(0,s)\, E\left[\lambda_s e^{-\int_0^s \lambda_u du}\right] ds.$$

$$E\left[\lambda_t e^{-\int_0^t \lambda_u du}\right] = E\left[\partial_t\left(1-e^{-\int_0^t \lambda_u du}\right)\right]$$
$$= -E\left[\partial_t e^{-\int_0^t \lambda_u du}\right] = -\partial_t E\left[e^{-\int_0^t \lambda_u du}\right]$$

$$= \sum_{k=1}^{2T} C_k\, e^{A(t_k) - B(t_k)\lambda_0} P(0,t_k)$$
$$+ P(0,T)\, e^{A(T) - B(T)\lambda_0}$$
$$+ R \quad P(0,s)\,(C(t) + H(t)\lambda_0)\, e^{A(t) - B(t)\lambda_0} \quad) ds.$$

After implementing in R, result:

```
1: 1.003023835127196
2: 1.004059729309519 7
3: 1.004399089415227 3
4: 1.004495251993172 5
5: 1.004507082385171 4
6: 1.004490092778390 6
7: 1.004463657258157 8
8: 1.00443452440974 4
9: 1.004405033474714 8
10: 1.004375984260437
```

Appendix :

```r
library(cubature)

#----------------------------b----------------------------------#
f_zero = function(K,theta,sig,lambda0,r,t){
  gamma = sqrt(K^2+2*sig^2)
  lead = 2*K*theta/(sig^2)
  A = lead*log(2*gamma*exp((K+gamma)*t/2)/(2*gamma+(K+gamma)*(exp(gamma*t)-1)))
  B = 2*(exp(gamma*t)-1)/(2*gamma+(K+gamma)*(exp(gamma*t)-1))
  result = exp(A-B*lambda0)*exp(-r*t)
  return(result)
}

K = 1; theta = 0.02; sig = 0.15; lambda0 = 0.01; r = 0.01
bonds = seq(10)
yields = seq(10)
spreads = seq(10)

for(i in 1:10){
  bonds[i] = f_zero(K,theta,sig,lambda0,r,i)
  yields[i] = -log(bonds[i])/i
  spreads[i] = yields[i] - r
}

#----------------------------c----------------------------------#
limit_bond = f_zero(K,theta,sig,lambda0,r,0.0000001)
limit_yield = -log(limit_bond)/0.0000001
limit_spread = limit_yield - r
```

```r
#----------------------------d----------------------------------#
freq = 0.5
t = c(1:10)
R = 0.5
c = 0.02

f_coupon = function(K,theta,sig,lambda0,r,t,freq,c){
  result = 0
  period = t/freq
  for(i in 1:period){
    result = result + exp(-r*K*freq)*c*freq*f_zero(K,theta,sig,lambda0,r,t)
  }
  return(result)
}

f_recovery = function(K,theta,sig,lambda0,r,t,R){
  f_temp = function(K,theta,sig,lambda0,r,t){
    result = (-r)*exp(-r*t)*(1-f_zero(K,theta,sig,lambda0,r,t))
    return(result)
  }
  integral = adaptIntegrate(f_temp,lower = 0, upper = t)$integral
  recovery = R*(exp(-r*t)*(1-f_zero(K,theta,sig,lambda0,r,t))-integral)
  return(recovery)
}

f_notional = function(K,theta,sig,lambda0,r,t){
  return(exp(-r*t)*f_zero(K,theta,sig,lambda0,r,t))
}

price = seq(10)
for(i in 1:10){
  price = f_coupon(K,theta,sig,lambda0,r,i,freq,c) +
    f_recovery(K,theta,sig,lambda0,r,i,R) +
    f_notional(K,theta,sig,lambda0,r,i)
}
```