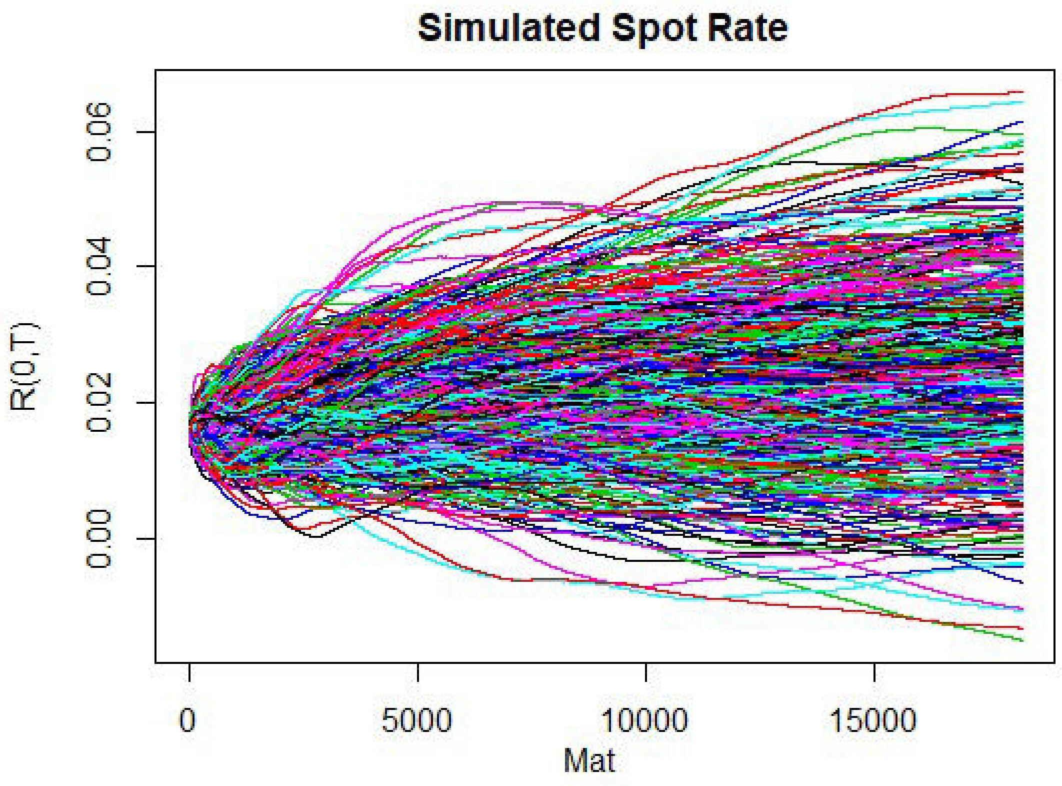
Hull-White 1-factor model using R code

Purpose of this post simulate future spot rates and other related time series using Hull-White 1-factor model like the following figures which is the simulation of future spot rates.



For detailed derivations and explanations regarding useful theorems, refer to the earlier posts on Hull-White 1-factor model

- Hull-White 1-factor model : 1) Introduction
- Hull-White 1-factor model : 2) Zero coupon bond
- Hull-White 1-factor model : 3) Simulation
- Hull-White 1-factor model : 4) Numerical integration
- Hull-White 1-factor model : 5) Numerical calculation

We summarize all results in Hull-White 1-factor model from previous posts and provide R code for the simulation of short rate, discount factors, and so on.

Hull and White (1990) introduced the no-arbitrage condition of Ho and Lee (1986) to Vasicek (1977). This model generates an exact fitting to the given initial term structure so that it can be used to price interest rate contingent claims such as IR option, swaption, structured IR products, and so on. It also provides the closed-form solution for interest rate cap, floor, and swaption.

As a starting point for developing this model, we assume that under the risk-neutral measure Q using money market account (\(B_t\)) as the numeraire, the stochastic process of short rates

(\(r(t)\\)) is as follows.
\[dr(t) = {\theta (t) – a(t)r(t)}dt + \sigma (t) dW(t),\] Here, \(r(t)\) can be divided into two parts : the stochastic (\(x(t)\\)) and deterministic parts (\(φ(t)\\)).
\[\begin{align} r(t) &= x(t) + φ(t),\\ dx(t) &= -a(t)x(t)dt+σ(t)dW(t),x(0)=0,\\ dφ(t) &= {θ(t)-a(t)φ(t)}dt,φ(0)=r(0) \end{align}\] \(θ(t)\) and \(φ(t)\) have the following forms after some derivations.
\[\begin{align} θ(t) &= \frac{\partial f(0,t)}{\partial t} + a(t)f(0,t) \\ &\quad + \int_{0}^{t} σ(u)^{2} e^{-2 \int_{u}^{t} a(v)dv} du, \\ φ(t) &= f(0,t) + \int_{0}^{t} σ(u)^{2} e^{- \int_{u}^{t} a(v)dv} B(u,t) du \end{align}\] For any \(s(< t)\), \(x(t)\) can be expressed as integrated form.
\[x(t) = x(s)e^{- \int_{s}^{t} a(v)dv} + \int_{s}^{t} σ(u) e^{- \int_{u}^{t} a(v)dv} dW(u) \]

1. Zero-coupon bond

Let \(P(t,T)\) denotes the time \(t\) price of zero-coupon bond with a maturity of \(T\). If \(\mathscr{F_t}\) is the information generated by \(x(t)\) available up to the time \(t\), \(P(t,T)\) is defined as \[\begin{align} P(t,T) &= E \left[\exp \left(-\int_{t}^{T} r(u)du \right)|\mathscr{F_t} \right] \\ &= E \left[\exp \left(-\int_{t}^{T} x(u)+φ(u) du \right)|\mathscr{F_t}\right] \end{align}\] We also define \(B(t,T)\) and \(V(t,T)\) for convenience.
\[\begin{align} B(t,T)&= \int_{t}^{T} e^{-\int_{t}^{u} a(v) dv}du, \\ V(t,T)&= \int_{t}^{T} \sigma (u)^2 B(u,T)^2 du \end{align}\] We can have the integrated form of \(x(t)\) from \(t\) to \(T\). \[\int_{t}^{T} x(u)du =x(t)B(t,T)+\int_{t}^{T} σ(u)B(u,T) dW(u)\] From the above result, we can find that \(\int_{t}^{T} x(u)du\) follows the normal distirbution with mean \(x(t)B(t,T)\) and variance \(V(t,T)\). When random variable follows the normal distribution with mean \(\mu\) and variance \(σ^2\), \(E[\exp(Y)]=\exp \left(\mu + \frac{1}{2}σ^2 \right) \). Using this theorem, \(P(t,T)\) can be expressed as follows. \[\begin{align} P(t,T) &= \exp \left(-\int_{t}^{T} φ(u)du \right) E \left[\exp \left(-\int_{t}^{T} x(u)du \right)|\mathscr{F_t} \right] \\ &= \exp \left(-\int_{t}^{T} φ(u)du -x(t)B(t,T) + \frac{1}{2}V(t,T) \right) \end{align}\] The no-arbitrage condition says that \(P(t,T)\) can explain the initial term structure with the perfect fit. The above equation meets this no-arbitrage condition if the market discount factor \(P(0,T)\) is incorporated into \(P(t,T)\) of the Hull-White model.
\[\begin{align} &P(0,T) = \exp \left(-\int_{0}^{T} φ(u)du + \frac{1}{2}V(0,T) \right)\\ \rightarrow &\exp \left(-\int_{0}^{T} φ(u)du \right) = P(0,T) \exp \left(– \frac{1}{2}V(0,T) \right) \end{align}\] Using the above no-arbitrage condition, the following relationship holds regarding \(φ(.)\) function. \[\begin{align} \exp \left(-\int_{t}^{T} φ(u)du \right) = \frac{P(0,T)}{P(0,t)} \exp \left(-\frac{1}{2}\{V(0,T)-V(0,t)\} \right) \end{align}\] Therefore, the zero-coupon bond price is \[P(t,T) = \frac{P(0,T)}{P(0,t)} \exp \left(-x(t)B(t,T) + \frac{1}{2}\{V(t,T)-V(0,T)+V(0,t)\} \right)\] Substituting with \(V(t,T)\), a reduced expression for \(P(t,T)\) is available. \[\begin{align} P(t,T) &= \frac{P(0,T)}{P(0,t)} \exp \left(-x(t)B(t,T) + \frac{1}{2}\Omega(t,T) \right)\\ \Omega(t,T) &= \int_{0}^{t} σ(u)^2 \{B(u,t)^2-B(u,T)^2\} du \end{align}\]

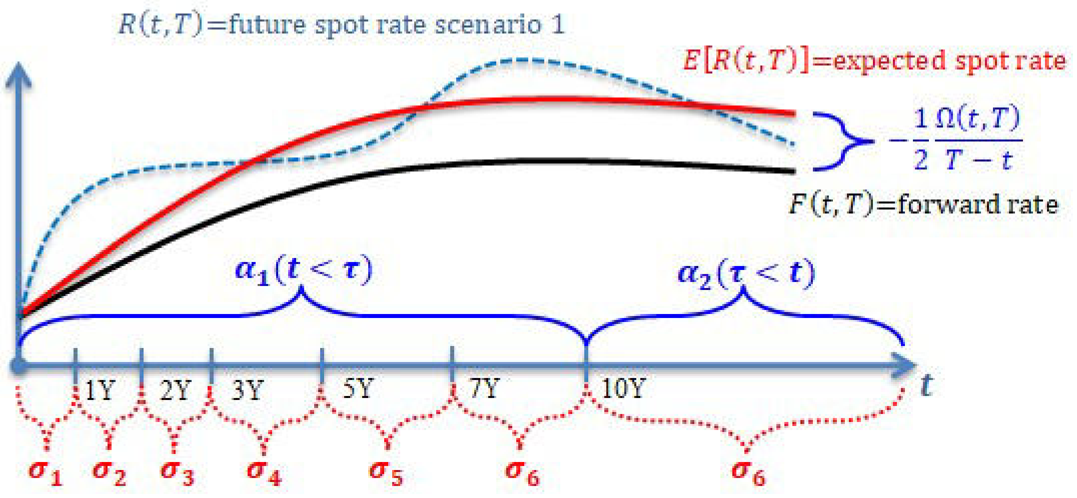
2. Simulation

We assume that at given times \(T_1\),\(T_2\),…,\(T_N\), cash flows of a derivaties take places with \(f_1\),\(f_2\),…,\(f_N\). The risk-neutral price of this derivatives is
\[P_0 = \displaystyle\sum_{j=1}^{N} E\left[\frac{f(T_j)}{B_{T_j}} \right] \]
At first, let's discretize time axis with \(\Delta t_i = t_{i+1} – t_i\).
\[\begin{align} 0 = t_0 &< t_1 < t_2 < t_3 < ... < t_{M_1 -1} < t_{M_1} = T_1 \\ &< t_{M_1 +1} < t_{M_1 +2} < ... < t_{M_2 -1} < t_{M_2} = T_2 \\ &< t_{M_2 +1} < t_{M_2 +2} <... \end{align}\]
The discretized process of \(x(t)\) has the following form. \[\begin{align} x_{t_{i+1}} &= x_{t_i} e^{-\int_{t_i}^{t_{i+1}} a(v)dv} \\ &+ \epsilon\sqrt{\int_{t_i}^{t_{i+1}}σ(u)^2 e^{-2 \int_{u}^{t_{i+1}}a(v)dv}du} \end{align}\] Here, \(\epsilon\) is the standard normal random

number. From the above scenario, since we can get \(x_{t_0}\),\(x_{t_1}\), \(x_{t_2}\), \(x_{t_3}\),…, discount factor at time \(T_j\) is \[\frac{1}{B_{T_j}} = \prod_{i=0}^{M_j-1} P(t_i , t_{i+1}) \] \[\begin{align} &P(t_i , t_{i+1}) = \frac{P(0 , t_{i+1})}{P(0 , t_i)} \\ &\times \exp\left(-x_{t_i} B(t_i,t_{i+1})+\frac{1}{2}\ int_{0}^{t_i}σ(u)^2 \{ B(u,t_i)^2-B(u,t_{i+1})^2 \}du \right) \end{align}\] Cash flow at time \(T_j\) is calculated as follows
\[R(t_i , {t_i}+\tau) = \frac{1}{{\tau}} \left\{ {\frac{1}{P(t_i , {t_i}+\tau)} -1} \right\} \] \[\begin{align} &P(t_i , {t_i}+\tau) = \frac{P(0 , {t_i}+\tau)}{P(0 , t_i)} \\ & \times \exp \left(-x_{t_i} B(t_i, {t_i}+\tau)+\frac{1}{2} \int_{0}^{t_i} σ(u)^2 \{ B(u,t_i)^2 – B(u,{t_i}+\tau)^2 \}du \right) \end{align}\] Finally, using scenarios for discount factors and cash flows, the present value of a derivatives with cash flows \(f(T_1)\),\(f(T_2)\),…,\(f(T_N)\) at times \(T_1\),\(T_2\),…,\(T_N\) under the risk-neutral measure (\(P_0\)) is \[P_0 = \displaystyle\sum_{j=1}^{N} E\left[\frac{f(T_j)}{B_{T_j}} \right] \] In other words, the present value of derivatives product is an average of values from many iterated simulation.

3. Numerical Integration

Since market data is not continuous, parameters for mean-reversion speed and volatility are also treated as a discrete case. But constant parameter is too restrictive to use practically. As you can see the following figure, it is typical to use piecewise constant volatility function and constant or two-regime mean-reversion speed function.



At first, we assume that \(a(t)\) have two regime according to the threshold year which divide time axis into short-term and long-term. \[a(t)=\begin{cases} a_1 & \text{if}\ t < \tau \\ a_2 & \text{if}\ t \geq \tau \end{cases}\] \(σ(t)\) is assumed to have the following piecewise constant function. \[σ(t)=\begin{cases} σ_1 & \text{if}\ t_0 \leq t < t_1 \\ σ_2 & \text{if}\ t_1 \leq t < t_2 \\ ... \\ σ_{M-1} & \text{if}\ t_{M-2} \leq t < t_{M-1} \\ σ_M & \text{if}\ t_{M-1} \leq t \end{cases}\] Using these functional forms of parameters, we need to calculate the following numerical integration before running a simulation. \[\begin{align} A(t,T) &= e^{-\int_{t}^{u} a(v)dv} \\ B(t,T) &= \int_{t}^{T} e^{-\int_{t}^{u} a(v)dv} du \\ Z(t) &= \int_{0}^{t} σ(u)^2 e^{-\int_{u}^{t} a(v)dv}B(u,t) du \\ \xi(t) &= \int_{0}^{t} σ(u)^2 e^{-2\int_{u}^{t} a(v)dv} du \\ \Omega(t,T) &= \int_{0}^{t} σ(u)^2 \{ B(u,t)^2 - B(u,T)^2 \} du \end{align}\] For these numerical integrations, \(a(t)\) and \(σ(t)\) are

applied differently according to which time is selected. \[\begin{align} A(t,T)&=\begin{cases} e^{-a_1 (T-t)} & \text{if}\ T < \tau \\ e^{-a_2 (T-t)} & \text{if}\ t > \tau \\ e^{-a_1 (\tau-t)-a_2(T-\tau)} & \text{if}\ t \leq \tau \leq T \end{cases} \\ \\ B(t,T)&=\begin{cases} \dfrac{1-e^{-a_1 (T-t)}}{a_1} & \text{if}\ T < \tau \\ \dfrac{1-e^{-a_2 (T-t)}}{a_2} & \text{if}\ t > \tau \\ \dfrac{1-e^{-a_1 (\tau-t)}}{a_1}+ \\ e^{-a_1 (\tau-t)}\dfrac{1-e^{-a_2 (T-\tau)}}{a_2} & \text{if}\ t \leq \tau \leq T \end{cases} \end{align}\] \[\begin{align} Z(t)&=\begin{cases} \displaystyle\int_{0}^{t} σ(u)^2 e^{-a_1 (t-u)} \dfrac{1-e^{-a_1 (t-u)}}{a_1} du & \text{if}\ t < \tau \\ e^{-a_2 (t-\tau)} \displaystyle \int_{0}^{\tau} σ(u)^2 e^{-a_1 (\tau-u)} \left\{ \dfrac{1-e^{-a_1 (\tau-u)}}{a_1} \right\} du \\ +e^{-a_2 (t-\tau)} \displaystyle\int_{0}^{\tau} σ(u)^2 e^{-a_1 (\tau-u)} \left\{ e^{-a_1 (\tau-u)} \dfrac{1-e^{-a_2 (t-\tau)}}{a_2} \right\} du \\ + \displaystyle\int_{\tau}^{t} σ(u)^2 e^{-a_2 (t-u)} \dfrac{1-e^{-a_2 (t-u)}}{a_2} du & \text{if}\ t \geq \tau \end{cases} \\ \\ \xi(t)&=\begin{cases} \displaystyle \int_{0}^{t} σ(u)^2 e^{-2 a_1 (t-u)} du & \text{if}\ t < \tau \\ e^{-2 a_2 (t-\tau)} \displaystyle \int_{0}^{\tau} σ(u)^2 e^{-2 a_1 (\tau-u)} du \\ +\displaystyle\int_{\tau}^{t} σ(u)^2 e^{-2 a_2 (t-u)} du & \text{if}\ t \geq \tau \end{cases} \end{align}\] \[\Omega(t,T) = -2B(t,T)Z(t) - B(t,T)^2\xi(t) \]
With closer scrutiny, these numerical integrations have the following ingredient in common. \[\begin{align} **I**(t) = \int_{0}^{t} σ(u)^2 e^{au} du \end{align}\] When maximum value is \(m\) which are \(t_j < t \), calculation of \(**I**(t) \) have the following form of summation.
(i) \(a ≠ 0 \) : \[\begin{align} &**I**(t) = \sum_{j=1}^{m} σ_j^2 \int_{t_{j-1}}^{t_j} e^{au} du + σ_{m+1}^2 \int_{t_m}^{t} e^{au}du \\ & = \sum_{j=1}^{m} σ_j^2 \frac{e^{a t_j} – e^{a t_{j-1}}}{a} + σ_{m+1}^2 \frac{ e^{a t_t} – e^{a t_m} }{a} \end{align}\] (ii) \(a = 0 \) : \[\begin{align} **I**(t) = \sum_{j=1}^{m} σ_j^2 (t_j – t_{j-1}) + σ_{m+1}^2 (t – t_m) \end{align}\]
Now let's express \(Z(t)\) and \(\xi(t)\) using \(**I**(t,a,b) = \int_{0}^{t} σ(u)^2 a e^{bu} du\).

\(Z(t)\) has the following functional form using \(**I**(t,a,b)\).

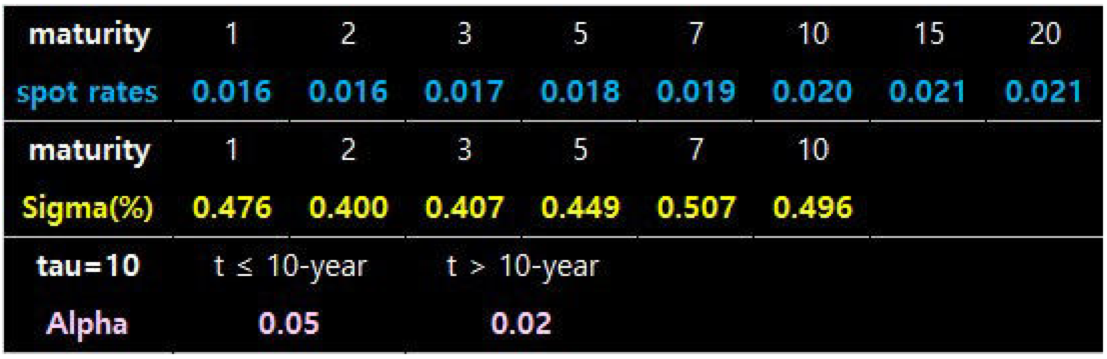
(i) \(t < τ\) \[\begin{align} Z(t) = \frac{1}{a_1} e^{-a_1 t} **I**(t,1,a_1) - \frac{1}{a_1} e^{-2a_1 t} **I**(t,1,2a_1) \end{align}\]
(ii) \(τ≤t \) \[\begin{align} Z(t) &= e^{-a_2 (t-\tau)} Z(\tau,1,a_1) \\ & + e^{-a_2 (t-\tau) – 2 a_1 \tau} B(\tau, t, a_2) **I**(\tau,1,2 a_1) \\ &+ Z(t,1,a_2) – \\ &\left(\frac{1}{a_2} e^{-a_2 t} **I**(\tau,1,a_2) – \frac{1}{a_2} e^{-2 a_2 t} **I**(\tau,1,2 a_2) \right) \end{align}\]
\(\xi(t)\) has the following functional form using \(**I**(t,a,b)\).

(i) \(t < τ\) \[\begin{align} \xi(t) = e^{-2 a_1 t} **I**(t,1,2a_1) \end{align}\] (ii) \(τ≤t \) \[\begin{align} \xi(t) &= e^{-2 a_2 (t-\tau) - 2 a_1 \tau} **I**(\tau,1,2 a_1) \\ & + e^{-2 a_2 t} (**I**(t,1,2 a_2) - **I**(\tau,1,2 a_2)) \end{align}\]
We can simulate \(x(t)\) using the following discretized stochastic process for \(x(t)\).
\[\begin{align} x_{t_{i+1}} &= x_{t_i} A(t_i, t_{i+1}) \\ &+ \epsilon\sqrt{\xi(t_{i+1}) – A(t_i, t_{i+1})^2\xi(t_i)} \end{align}\]

4. Simulation : R code

For ease of exposition, we assume that model parameters are given after some calibration.

* Calibrated parameters for Hull-White 1 factor model



maturity	1	2	3	5	7	10	15	20
spot rates	0.016	0.016	0.017	0.018	0.019	0.020	0.021	0.021
maturity	1	2	3	5	7	10		
Sigma(%)	0.476	0.400	0.407	0.449	0.507	0.496		
tau=10	t ≤ 10-year	t > 10-year						
Alpha	0.05	0.02						

The following R code is for simulating short rates, discount factors, and so on using the Hull-White 1 factor model with given calibrated parameters.

```
#==============================================================
==============#
# Financial Econometrics & Derivatives, ML/DL using R, Python, Tensorflow
# by Sang-Heon Lee
#
# https://kiandlee.blogspot.com
#————————————————————————————-#
# Numerical Simulation for Hull-White 1 factor model
#==============================================================
==============#

library(Rfast)  # colCumProds

graphics.off()  # clear all graphs
rm(list = ls()) # remove all files from your workspace

setwd("D:/a_book_FIER_Ki_Lee/ch05_HW1F/code")

# Functions for numerical Integration

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
#  I(t) = Int_0^t sigma(s)^2 A exp(Bs) ds
#—————————————————————————-#
#           t
#  I(t) = ∫  σ(u)^2 A exp(Bu) du
#          0
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
fI<–function(t, A, B, lt.HW) {
    M <– 0; value <– 0

    tVol <– lt.HW$tsig   # volatility tenor
    Vol  <– lt.HW$sigma  # volatility vector
    nVol <– lt.HW$nsig   # # of volatility

    # find Maximum M from j which is t_j < t
    M <– ifelse(length(which(tVol<=t))==0,1,max(which(tVol<=t))+1)

    # summation part
    if (B==0) {
        if (M==1) value <– value + Vol[1]^2*A*t
        else {
            for (i in 1:(M–1)) {
                add <– Vol[i]^2*A*(tVol[i] – ifelse(i==1,0,tVol[i–1]))
                value <– value + add
```

cs

```
            }
            add <– Vol[ifelse(M==(nVol+1),M–1,M)]^2*A*(t–tVol[M–1])
            value <– value + add
        }
    }
    else {
      if (M==1) { value <– value + Vol[1]^2*A/B*(exp(B*t)–1)}
      else {
          for (i in 1:(M–1)) {
              add <– Vol[i]^2*A/B*
                     (exp(B*tVol[i])–ifelse(i==1,1,exp(B*tVol[i–1])))
              value <– value + add
          }
          add <– Vol[ifelse(M==(nVol+1),M–1,M)]^2*A/B*
                 (exp(B*t)–exp(B*tVol[M–1]))
          value <– value + add
      }
    }
    return(value)
}

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
#  A(s,t)=e^(-Int_s^t a(v) dv)
#————————————————————————————————————————-#
#                    s
#  A(s,t) = exp( -∫ a(v)dv )
#                   t
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
fA<–function(s, t, lt.HW) {
    tau <– lt.HW$tkap       # tau
    K1  <– lt.HW$kappa[1]   # short-term kappa
    K2  <– lt.HW$kappa[2]   # long-term kappa

    if      (tau <= s) f <– exp(–K2*(t–s))
    else if (t < tau ) f <– exp(–K1*(t–s))
    else               f <– exp(–K1*(tau–s)–K2*(t–tau))

    return(f)
}

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
#  B(s,t)=Int_s^t e^(-Int_t^u a(v) dv) du
#————————————————————————————————————————-#
#             t       u
#  B(s,t) = ∫ exp( -∫ a(v)dv ) du
#            s       t
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
fB1<–function(s, t, kappa) {return((1 – exp(–kappa*(t–s)))/ kappa)}

fB<–function(s, t, lt.HW) {
    tau <– lt.HW$tkap       # tau
    K1  <– lt.HW$kappa[1]   # short-term kappa
    K2  <– lt.HW$kappa[2]   # long-term kappa

    if      (tau <= s) f <– fB1(s, t, K2)
    else if (t < tau ) f <– fB1(s, t, K1)
    else  f <– fB1(s,tau,K1)+exp(–K1*(tau–s))*fB1(tau,t,K2)

    return(f)
}
```

```

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
#  Zeta(t) = Int_0^t σ(u)^2 e^(-2 Int_u^t a(v) dv) du
#——————————————————————————————————————————-#
#            t               t
#  Zeta(t) = ∫ σ(u)^2 exp( -2∫ a(v)dv ) du
#           0               u
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
fZeta<–function(t, lt.HW) {
    tau <– lt.HW$tkap       # tau
    K1  <– lt.HW$kappa[1]   # short-term kappa
    K2  <– lt.HW$kappa[2]   # long-term kappa

    if (t < tau) f = exp(–2*K1*t)*fI(t,1,2*K1,lt.HW)
    else  f = exp(–2*K2*(t–tau)–2*K1*tau)*fI(tau,1,2*K1,lt.HW)+
          exp(–2*K2*t)*(fI(t,1,2*K2,lt.HW)–fI(tau,1,2*K2,lt.HW))

    return(f)
}

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
#  Z(t) = Int_0^t σ(u)^2 e^(-Int_u^t a(v) dv) B(u,t) du
#——————————————————————————————————————————-#
#         t             t
#  Z(t) = ∫ σ(u)^2 exp( -∫ a(v)dv ) B(u,t) du
#        0             u
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
fZ1<–function(t, kappa, lt.HW) {
    I1 = exp(  –kappa*t)*fI(t,1,  kappa, lt.HW) / kappa
    I2 = exp(–2*kappa*t)*fI(t,1,2*kappa, lt.HW) / kappa
    return(I1 – I2)
}

fZ<–function(t, lt.HW) {
    tau <– lt.HW$tkap       # tau
    K1  <– lt.HW$kappa[1]   # short-term kappa
    K2  <– lt.HW$kappa[2]   # long-term kappa

    if (t < tau)
        f = fZ1(t, K1, lt.HW)
    else {
        I1 = exp(–K2*(t–tau))*fZ1(tau, K1, lt.HW)
        I2 = exp(–K2*(t–tau))*fB(tau,t,lt.HW)*
             exp(–2*K1*tau)*fI(tau,1,2*K1,lt.HW)
        I3 = exp(–K2*t) * fI(tau, 1, K2, lt.HW) / K2
        I4 = exp(–2*K2*t) * fI(tau, 1, 2*K2, lt.HW) / K2
        f =  I1 + I2 + fZ1(t, K2, lt.HW) – I3 + I4
    }
    return(f)
}

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
#  Omega(t,T) = Int_0^t sigma(s)^2 [B(s,t)^2 – B(s,T)^2] ds
#——————————————————————————————————————————-#
#               t
#  Omega(t,T) = ∫ σ(s)^2 [B(s,t)^2 – B(s,T)^2] ds
#              0
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
fOmega<–function(t, T, lt.HW) {
    return(–fB(t,T,lt.HW) * (2.0*fZ(t,lt.HW) +
```

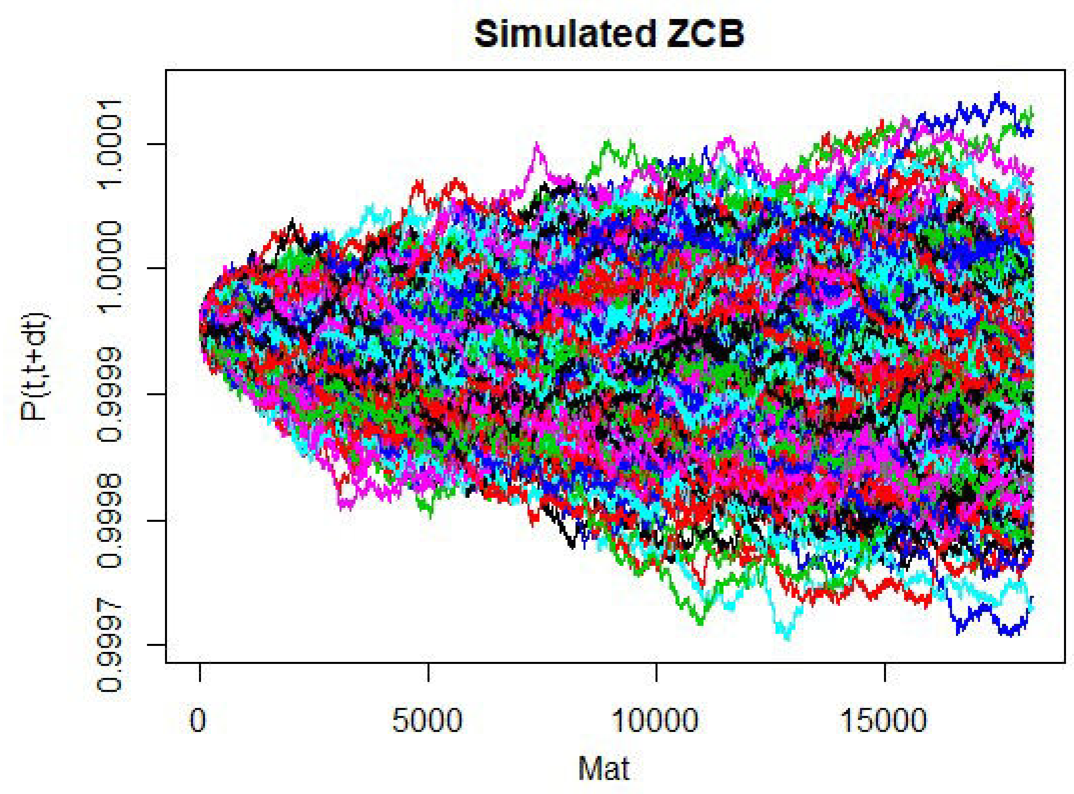
```
                        fB(t,T,lt.HW)*fZeta(t,lt.HW)))
}

#==============================================================
==============#
#           Main : Hull-White 1 Factor Model Simulation
#==============================================================
==============#

  #———————————————————————————————————————#
  # Information List for the Hull-White model
  #———————————————————————————————————————#
  # – tkap  : threshold year which divide mean-reversion speed
  # – kappa : mean-reversion speed parameters
  # – tsig  : maturity vector for volatility parameters
  # – sigma : volatility parameter vector
  # – tDF   : maturity vector for spot rates
  # – rc    :spot rates curve
  #———————————————————————————————————————#

  # list object which contain Hull-White model related information
  lt.HW <– list(
     tkap  = 10,
     kappa = c(0.05, 0.02),
     tsig  = c(1.0, 2.0, 3.0, 5.0, 7.0, 10.0),
     sigma = c(0.004761583,0.004000462,0.004073902,
             0.004487176,0.00507169,0.00496086),
     tDF   = c(1.0, 2.0,3.0,5.0,7.0,10.0,15.0,20.0),
     rc    = c(0.01596,0.01608,0.016525,0.01756,
             0.0185,0.01973,0.02056,0.020925)
     )

  # Add other information to list
  lt.HW$nDF  <– length(lt.HW$tDF)   # # of spot
  lt.HW$nsig <– length(lt.HW$sigma) # # of vol
  lt.HW$nkap <– length(lt.HW$kappa) # # of kappa

  # Check for Numerical Integration Functions for HW1F
  m.temp <– matrix(NA,15,5)
  colnames(m.temp) <– c("I", "B", "Zeta", "Z", "Omega")
  for(i in 1:15) {
     m.temp[i,1] <– fI    (i, 2, 3, lt.HW)
     m.temp[i,2] <– fB    (0.5,  i, lt.HW)
     m.temp[i,3] <– fZeta (i,       lt.HW)
     m.temp[i,4] <– fZ    (i,       lt.HW)
     m.temp[i,5] <– fOmega(0.5,  i, lt.HW)
  }
  print("Check for Numerical Integration Functions for HW1F")
  print(m.temp)

  # Discount Factor
  lt.HW$DF   <– exp(–lt.HW$tDF*lt.HW$rc)

  #———————————————————————————————————————#
  # Preprocessing for simulation
  #———————————————————————————————————————#

  # Simulation information
  denom.1y     <– 365    # # of dt in 1-year

```

```r
# t : valuation date, T : maturity
lt.HW.sim    <– list(t=0, T=50, dt=1/denom.1y, nscenario =5000)

lt.HW.sim$nt <– round(lt.HW.sim$t*denom.1y,0)
lt.HW.sim$nT <– round(lt.HW.sim$T*denom.1y,0)

# spit the time axis by dt
v.Ti <– seq(lt.HW.sim$dt, lt.HW.sim$T, length = lt.HW.sim$nT)

#————————————————————————————#
# Linear Interpolation of spot rate curve
#————————————————————————————#
# rule=2 : For outside the interval [min(x), max(x)],
#          the value at the closest data extremeis used.
#————————————————————————————#
frci <–approxfun(x=lt.HW$tDF, y=lt.HW$rc, rule=2)

v.rci <– frci(v.Ti)          # interpolated spot rates
v.DFi <– exp(–v.Ti*v.rci) # interpolated DF

#————————————————————————————#
# temporary use for blog width adjustment
#————————————————————————————#
sim <– lt.HW.sim
par <– lt.HW
dt  <– lt.HW.sim$dt

# standard normal random error
set.seed(123456)

# predetermined vector
v.A <– v.Zeta <– v.dZeta.sqrt <– v.B <– v.Omega <– rep(0, sim$nT)

for (n in 1:sim$nT) {
  v.A[n]     <– fA    (v.Ti[n]–dt, v.Ti[n], par)
  v.Zeta[n]  <– fZeta (v.Ti[n],             par)
  v.B[n]     <– fB    (v.Ti[n]–dt, v.Ti[n], par)
  v.Omega[n] <– fOmega(v.Ti[n]–dt, v.Ti[n], par)
}

v.dZeta.sqrt <– c(sqrt(v.Zeta[1]),
                 sqrt(v.Zeta[–1]–v.A[–1]^2*v.Zeta[–sim$nT]))

# selecting some indices because plotting is time-consuming
v.idx.sample <– sample(1:sim$nscenario, 500)

#————————————————————————————#
# Simulation Part
#————————————————————————————#

# interpolated discount factor from initial yield curve
v.P0 <– v.DFi
# ratio of bond price P(0,t+dt)/P(0,t)
v.P0T_P0T1 <– c(v.P0[1]/1,v.P0[–1]/v.P0[–sim$nT])

m.P.ts   <– matrix(0, sim$nT, sim$nscenario ) # P(t,t+dt)
m.Rsc.ts <– matrix(0, sim$nT, sim$nscenario ) # short rate

# Simulate from now on.

# for n=1
m.P.ts  [1,] <– v.P0T_P0T1[1]
m.Rsc.ts[1,] <– –log(m.P.ts[1,])/dt
xt <– rnorm(sim$nscenario, 0, 1)*v.dZeta.sqrt[1]

for(n in 2:sim$nT) {
   print(n)
   m.P.ts[n,] <– v.P0T_P0T1[n]*exp(–xt*v.B[n]+0.5*v.Omega[n])
   xt <– xt*v.A[n] + rnorm(sim$nscenario, 0, 1)*v.dZeta.sqrt[n]
}

m.Rsc.ts <– –log(m.P.ts)/dt      # spot rates
m.DF.ts  <– colCumProds(m.P.ts)  # Dscount Factors
m.R0T.ts <– –log(m.DF.ts)/v.Ti   # future spot rates

## plot paths
t <– seq(dt, lt.HW.sim$T, dt)

x11(width=6, height=5);
   matplot(m.P.ts[,v.idx.sample], type=“l”, lty=1,
         xlab=“Mat”,ylab=“P(t,t+dt)”,main=“Simulated ZCB”)
x11(width=6, height=5);
   matplot(m.Rsc.ts[,v.idx.sample], type=“l”, lty=1,
         xlab=“Mat”,ylab=“R(t,t+dt)”,main=“Simulated Short Rate”)
x11(width=6, height=5);
   matplot(m.DF.ts[,v.idx.sample], type=“l”, lty=1,
         xlab=“Mat”,ylab=“DF(0,T)”  ,main=“Simulated Discount Factor”)
x11(width=6, height=5);
   matplot(m.R0T.ts[,v.idx.sample], type=“l”, lty=1,
         xlab=“Mat”,ylab=“R(0,T)”   ,main=“Simulated Spot Rate”)
```

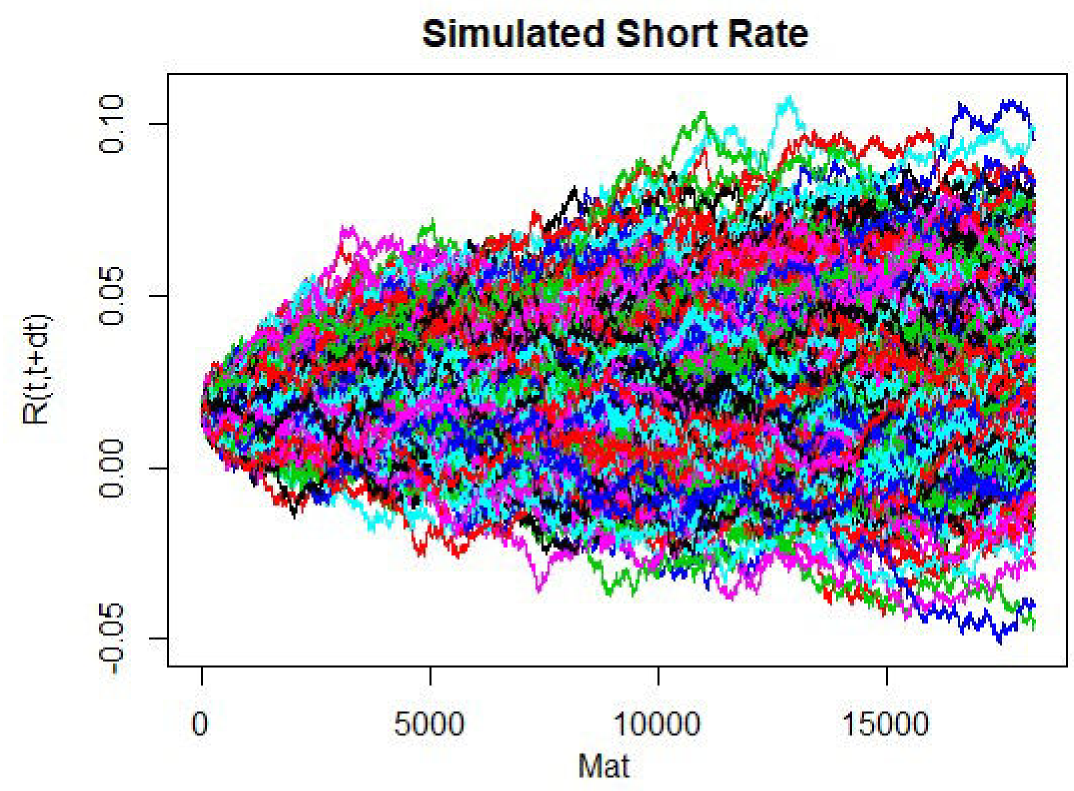
5. Simulation Results

After running the above R code, you can find the simulated outputs. To further illustrate the dynamic characteristics of simulated variables, we draw four graphs for a clear understanding of the Hull-White model simulation.

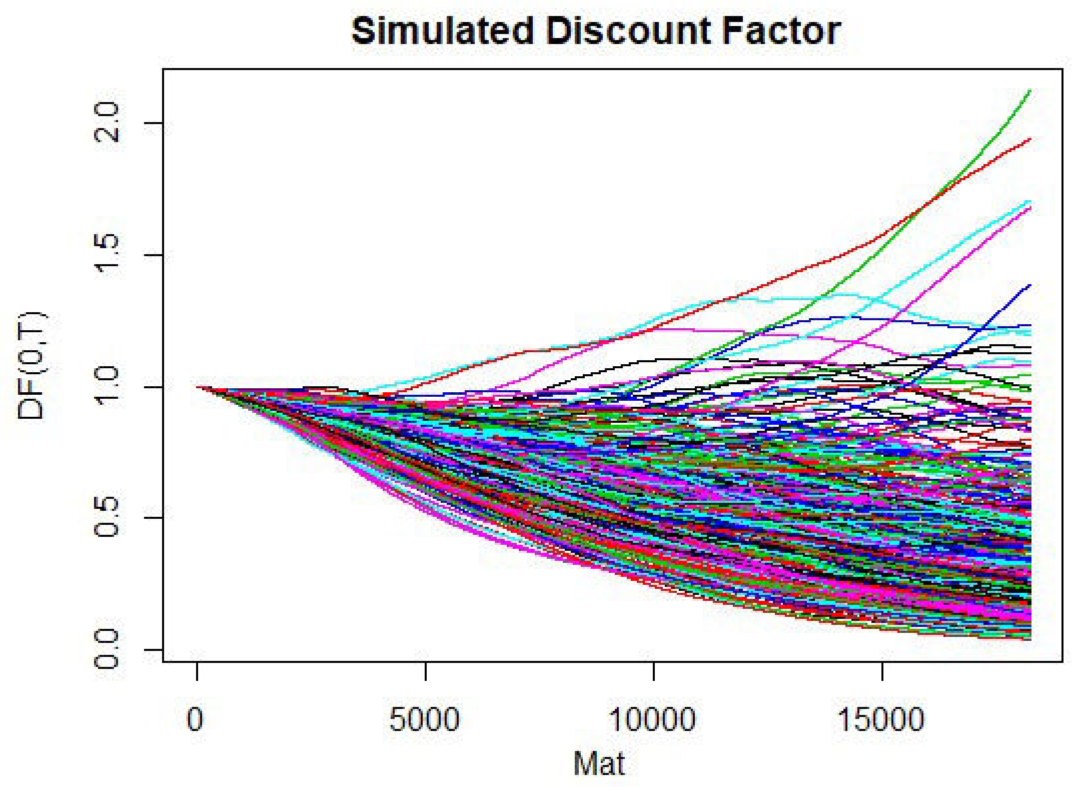
The following graph draws future zero coupon bond prices with \(dt\) maturity. Since maturity is too short, most simulated prices are centered on the neighborhood of 1.



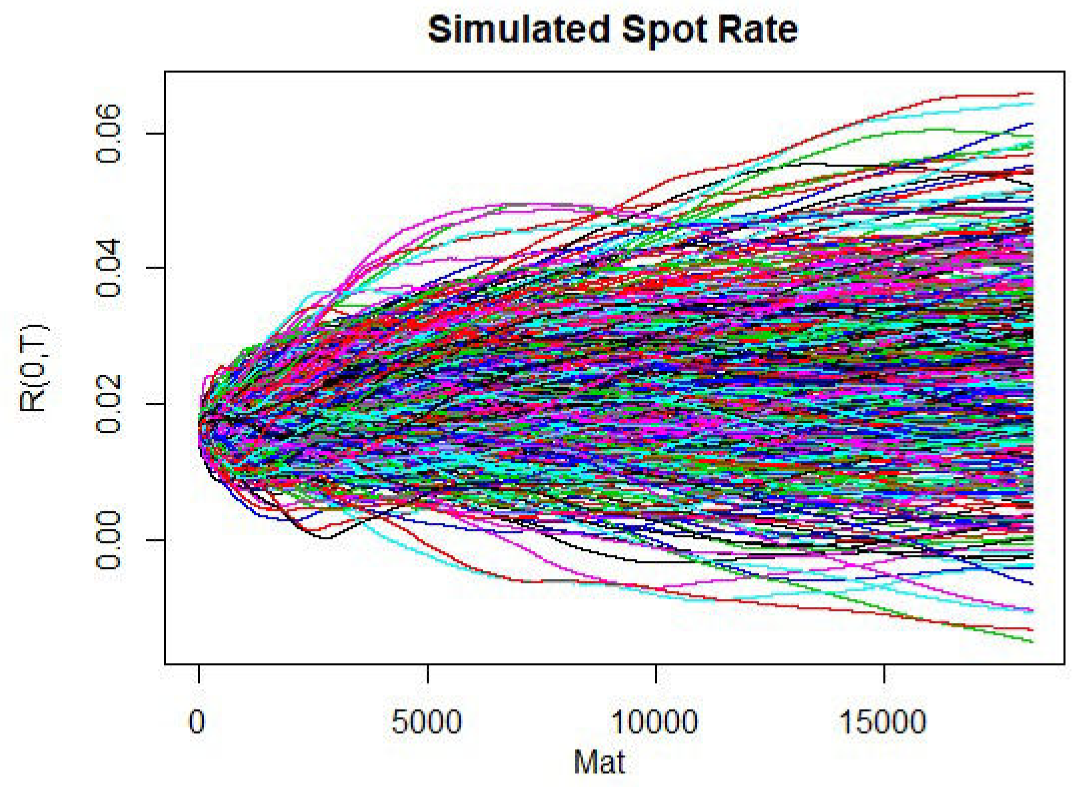
The following graph is the result of future short rates. As the Hull-White model is the normal model, we can find some of the future short rates below zero which is negative.



The following graph shows the simulated discount factors. As the Hull-White model is the normal model, we can find some of the discount factors exceeding 1.



The following graph is about the simulation of future spot rates. Due to the same reason, we also observe some negative values.



The remaining job is to calibrate parameters of the Hull-White 1 factor model with market data such as the swaption volatility matrix. This topic will be discussed next time. \(\blacksquare\)