FE - 680 Assignment 3

Instructor: Dragos Bozdog (PhD)

Completed By: Riley Heiman

In [ ]:
```
import os
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns

import pandas as pd
import numpy as np
from scipy.stats import norm
```

In [ ]:
```
print(os.getcwd())
```

```
d:\Documents\Stevens\FE-680\HW3
```

Problem 1

In the Hull–White model, $a = 0.08$ and $\sigma = .02$ .Calculate the price of a one-year **European call**

option on a zero-coupon bond that will mature in five years when the term structure is flat at 5, the principal of the bond is $\$100$, and the strike price is $\$70$.

Page 719 of Hull states:

> Some of the models just presented allow options on zero-coupon bonds to be valued analytically. For the Vasicek, Ho–Lee, and Hull–White one-factor models, the price at time zero of a call option that matures at time T on a zero-coupon bond maturing at time s is

$$LP(0,s)N(h) - KP(0,T)N(h-\sigma_P) \qquad (32.10)$$

Where:

- L = Principal of the bond
- K = Strike Price
- T = Maturity of option
- s = Maturity of Zero Coupon Bond

$$h = \frac{1}{\sigma_P} ln(\frac{LP(0,s)}{P(0,T)K} + \frac{\sigma_P}{2})$$

$$\sigma_P = \frac{\sigma}{a}[1 - e^{-a(s-T)}]\sqrt{\frac{1-e^{-2aT}}{2a}}$$

$$P(0,T) = risk - free \ discount \ factor$$

So

$$P(t,T) = e^{-r(T-t)}$$

Riley's thoughts: We know the Vasicek is a special case for Hull-White when $a(t)$, $b(t)$, and $c(t)$ are contants. What about Ho-Lee? Is Ho-Lee also a specical case of Vasicek? The textbook by Hull implied all three stocastic interest rate equations have the exact same analytical solution! (A little bit skeptical here.)

In [ ]:
```
# INPUTS
a = .08
sigma = .02
T = 1.0
s = 5.0
r = .05

L = 100
K = 70
```

In [ ]:
```
# Step 1: Find P(0,T), P(0,s)
P0T = np.exp(-r*T)
P0s = np.exp(-r*s)

# Step 2: Find Sigma P
sigma_p = (sigma / a) * ( 1 - np.exp( -a * (s-T) ) ) * ( np.sqrt( (1-np.exp( -2*a*T ) )/(2*a)  )  )

# Step 3: Find h
h = (1 / sigma_p) * np.log( ((L*P0s)/(K*P0T)) + (sigma_p / 2) )

# Step 4: Find the price
price = L*P0s*norm.cdf(h,loc=0, scale=1) - K*P0T*norm.cdf((h-sigma_p),loc=0, scale=1)

print('P(0,T) = ' + str(P0T))
print('P(0,s) = ' + str(P0s))
print('sigma_p = ' + str(sigma_p))
print('h = ' + str(h))

print('-----------------------')

print('The price = $' + str( np.round( price, decimals = 3) ))
```

```
P(0,T) = 0.951229424500714
P(0,s) = 0.7788007830714049
sigma_p = 0.06581336189047297
h = 2.802182536818634
-----------------------
The price = $11.303
```

Problem 2

Consider the example in lecture Notes: Vasicek Model Tree Construction and implement an algorithm to calculate the interest rates for 6 months ($\Delta t = 1 \ month$). The risk neutral dynamics of the Vasicek model is given by:

$$dr = k(\theta - r)dt + \sigma dW(t)$$

Where:

- $k = .025$
- $\sigma = 126$
- $r_0 = 5.121\%$
- $\theta = 15.339\%$

a) Report the interest rate values on the terminal nodes at $\Delta t = 6 \ months$.

b) Calculate the price of a zero-coupon bond with maturity in 6 months with face value 100.

c) Calculate the price of a European call option on the zero-coupon bond that expires at t = 6 months and has strike $90.

d) Calculate the price of an American call option on the zero-coupon bond that expires at t = 6 months and has strike $90.

Riley's Notes I don't believe the intent for this problem was to be complicated!

I attempted by best to create a Vasicek tree based upon the instructions on Lecture 4 **The General Hull-White Model and Super Calibration.PDF**. I've come to the realization however this is a **trinomial** tree, but the problem is asking for a **binomial** tree.

I will have to review this problem further next week.

In [ ]:
```
print("I dont know (Send Help)")
```

```
I dont know (Send Help)
```

Problem 3

Risk-free zero rates are flat at 5% in the U.S and flat at 9% in Australia (both rates are annually compounded).

In a 4-year differential swap Australian floating risk-free rate is received and **8%** is paid with both being applied to a USD principal of **$10 million**.

Payments are exchanged annually. The volatility of all 1-year forward rates in Australia is estimated to be 30%, the volatility of the forward USD/AUD exchange rate (AUD per USD) is 20% for all maturities, and the correlation between the two is 0.3. What is the value of the swap?

Riley's thoughts: This is just a swap, *not* a swap option.

$$Swap \, Value = floating \, value - Fixed \, value$$

$$V_i \rho_i \sigma_W \sigma_V t \qquad (34.3)$$

In [ ]:
```
principal = 10
diff_swap_rate = .08
n_years = 4
zero_rate = .05

aus_rate = .09

sigma_aus = .3
sigma_usd = .20
rho = .3


payment = principal*diff_swap_rate
tmp_sum = []

for i in range(1, n_years+1):
    tmp_sum.append(  payment / (1+zero_rate )**i  )


fixed_present_value = np.sum(tmp_sum)


# Find the quanto adjustment (34.3)
# Forward rate not provided. Use zero rate

def quanto_adj(t):
    return( 1 + zero_rate * rho * sigma_aus * sigma_usd*t)

payment2 = aus_rate * principal
tmp_sum = []
for i in range(1, n_years+1):
    tmp_sum.append(  payment2*quanto_adj(i-1) / (1+zero_rate )**i  )

floating_sum = np.sum(tmp_sum)

print('Payment Amount = $ ' + str(payment))

print('Present Value (Fixed)= $ '  + str( np.round( fixed_present_value, decimals = 2)) )
print('Present Value (Floating)= $ '  + str( np.round( floating_sum, decimals = 4)) )

print('Swap Value (millions) = Floating - Fixed = ' + str(floating_sum - fixed_present_value))
```

```
Payment Amount = $ 0.8
Present Value (Fixed)= $ 2.84
Present Value (Floating)= $ 3.1955
Swap Value (millions) = Floating - Fixed = 0.35872832821715184
```

Problem 4

Calculate the alternative duration measure for a 2-year bond with principal of $\$100$ paying coupon semiannually at the rate of $\$3$ per year when Vasicek's model is used with $a = .13$, $b = .012$, $\sigma = .01$ and $r = 1\%$

Recall from Hull

$$P(t,T) = A(t,T)e^{-B(t,T)r(t)} \qquad (31.6)$$

$$B(t,T) = \frac{1-e^{-a(T-t)}}{a} \qquad (31.7)$$

$$A(t,T) = exp[\frac{(B(t,T) - T + t)(a^2 b - \frac{\sigma^2}{2})}{a^2} - \frac{\sigma^2 B(t,T)^2}{4a}] \qquad (31.8)$$

$$\hat{D} = \Sigma_{i=1}^{n} \frac{c_i P(t,T_i)}{Q} \hat{D}_i$$

In [ ]:
```
a = .13
b = .012
sigma = .01
r = .01
t = 0
T = 2
c = 1.5

# Bond Price
Q = 103.95

def B(a, t, T):
    return(   ( 1-np.exp( -a*(T-t) ) ) / a  )

def A(B, t, T, a, b, sigma):
    return( np.exp(  (((b - T + t)*(a**2 *b - (sigma**2)/2))) /(a**2)  - ((sigma**2 * B**2)/(4*a)) ) )

def P(A,B,r):
    return( A * np.exp( - B * r ) )
```

In [ ]:
```
B1  = B( a, t, .5 )
B2 = B( a, t, 1 )
B3 = B( a, t, 1.5 )
B4 =B( a, t, 2.0 )

A1 = A(B1, t, .5 , a, b, sigma)
A2 = A(B2, t, 1  , a, b, sigma)
A3 = A(B3, t, 1.5, a, b, sigma)
A4 = A(B4, t, 2.0, a, b, sigma)

P1 = P(A1,B1,r)
P2 = P(A2,B2,r)
P3 = P(A3,B3,r)
P4 = P(A4,B4,r)

print('P(t,T) = P(0,0.5) = ' + str(P1))
print('P(t,T) = P(0,1.0) = ' + str(P2))
print('P(t,T) = P(0,1.5) = ' + str(P3))
print('P(t,T) = P(0,2.0) = ' + str(P4))

# Find D HAT
D_HAT = (c*P1*B1) + (c*P2*B2) + (c*P3*B3) + (c*P3*B3) + (101.5*P4*B4)
D_HAT = D_HAT /Q

print('')
print('(Alternative Duration) D-Hat = ' + str( np.round(D_HAT , decimals = 2) ) )
```

```
P(t,T) = P(0,0.5) = 0.9907448504454588
P(t,T) = P(0,1.0) = 0.9816903952951357
P(t,T) = P(0,1.5) = 0.9729341268140901
P(t,T) = P(0,2.0) = 0.9644645278840127

(Alternative Duration) D-Hat = 1.72
```