Short Rate Models

Summary

1.Bonds

A bond is a type of debt security where the issues borrows money from investors and agrees to pay them b ack with interest over a specified period of time. Bonds are commonly issued by governements and companies looking to raise funds for various projects and goals (Various bonds can be found in the : London Stock Exchange website)

Bonds have multiple key features :

- Issuer : The entity that borrows money
- Face Value F : Which is the amount that will be repaid to the bondholder at the bond's maturity. It is usually 1000$ per bond.
- Coupon Rate r : The interest rate with wich the issuer agrees to pay the owner of the bond. It is almost always a fixed percentage of the bonds Face Value.
- Annual Coupon Payment : $C = r * F$ it's the ammount that the owner of the bonds get
- Time to Maturity N : The date when the bond's face value will be paid back
- Market interest rate r_c : bonds can be bought and sold in the secondary markets before their maturity dates. The price of a bond fluctuates based the current interest rates of the market which is why having an idea of where they will be in the future is important for bond holders

The value of the bond at the time of the acquirement is given by the following formula:

$$\text{Bond Price} = \sum_{t=1}^{N} \frac{C}{(1+r_c)^t} + \frac{F}{(1+r_c)^N} = \sum_{t=1}^{N} \frac{r*F}{(1+r_c)^t} + \frac{F}{(1+r_c)^N}$$

let's say for example that the company Apple : issued bonds on the 10 of february 2023 with the following features :

- Face Value F = 1000$
- Coupon Rate r = 5%
- Number of coupons = 5
- Time to Maturity N = 5

on the 10 february 2024, just before the coupon is expected to be paid, let's consider 3 different scenarios :

- The market interest rate didn't change $r = r_c$
- The markest interest rate went up by 100 base points $r_{c1} = 6\%$
- The markest interest rate went down by 100 base points $r_{c2} = 4\%$

let's see how the Bond Price changes in these different scenarios

In [1]:
```
import matplotlib as plt
from math import *

def Bondprice(F,r,N,rc):
    P = 0
    for i in range(N) :
        P += r*F/(1+rc)**i
    P = P + F/(1+rc)**N
    return P

print(f" The value of the bond when the interest rate doesn't change is : Bo
```

```
 The value of the bond when the interest rate doesn't change is : Bond price
= 5999.36$
 When it goes up by a 100 base points the value is : Bond price = 5833.05 $
 When it goes down by a 100 base points the value is : 6248.32$
```

the potential loss for a 100 base point variance is $Delta = 166.64$
Which grows extremely fast if a company owns many of these bonds in their portfolio, this is why understanding how market interest rates can change is extremely crucial.

2 Short-Rate Models

Short-rate models focus on the short rate, which is the interest rate applicable over an incredibly short period of time, known as an infinitesimal. An Infinitesimal is a non-zero quantity that is so close to 0 that it’s smaller than any

other non-zero real number. This short rate plays a crucial role in many financial models because it is used to construct the entire term structure of interest rates (also known as the yield curve), which shows the relationship between interest rates and different time horizons.

How Short-Rate interest models work ?

A short-rate model typically represents the short rate $r(t)$ as a stochastic process, meaning that its path is driven by both deterministic (predictible) and Brownian (random)factors. The short rate keeps fluctuating over time due to the inherent uncertainty of financial markets.

- **deterministic Factors** : this component models predictibal tendencies like mean-reverting behavior as interest rates tend to move towards a long-term equilibrum
- **Brownian Motion** : Often Modeled using brownian motion, it reprents the uncertainty and volatility of financian markets

as mentionned, the short rate is modeled by a stochastic differential equation (SDE) under the form :

$$dr(t) = \mu(r(t), t)dt + \sigma(r(t), t)dW(t)$$

in this equation :

- $\mu(r(t), t)$ is called the **drift** term it represents the expected change of the rate $r(t)$ over an infinitesimal time interval.
- $\sigma(r(t), t)$ is the **volatility** term, is reflects the level of uncertainty in the movement of the short rate
- $dW(t)$ represents a Brownian Motion, a standard mathematical model for random fluctuations in financial processes

2.1 Vacisek Model (1977)

The Vasicek Model, introduced in 1977, is one of the earliest and most recognized models used to describe how interest rates behave over time. It's a **mean-reverting** short-rate model, meaning it assumes that while interest rates can fluctuate randomly, they eventually tend to revert back toward a long-term average. The Vasicek model was revolutionary for its time because it gave a mathematically manageable way to track the evolution of interest rates, especially in times of economic uncertainty (end of the gold standard in 1971, 1973 Oil Crisis, Collapse of Breton Woods etc ...)

The Vacisek Model uses an SDE as seen above :

$$dr(t) = a(b - r(t))dt + \sigma dW(t)$$

Where :

- $r(t)$ os the short rate at time t.
- a is the speed of mean reversion, in simpler terms it represents how quickly the sort rate returns to its long-term mean.
- b is the long-term mean rate that the rate is supposed to revert to.
- σ is the **Volatility** which is the standard
- $dW(t)$ is the increment of the Wiener process (a Brownian motion with $\mathbb{E}[X]$ = 0, $\mathrm{Var}(X) = f(t)$)

Interpretation :

$$a(b - r(t))$$

This part of the model is known as the **Deterministic Component**. It pulls the short rate back toward its long-term mean, denoted as k. Whenever the short rate $r(t)$ rises above the long-term mean k, the deterministic component becomes negative, pulling the rate back down. Since the **Wiener process** has a mean of zero, it doesn't affect this tendency. The parameter a controls the speed of this reversion, meaning it dictates how fast or slow the rate moves back toward the mean k. A large a means a faster adjustment, while a smaller a results in slower reversion.

$$\sigma dW(t)$$

This term represents the random fluctuations in the sort rate due to market volatility, it models the uncertainty in interest rate movements. The parameter σ represents the size of these fluctuations (i.e, **the volatility**). The higher the σ the more volatile the interest rates. Following here is a simple implementation of the Vasicek Model

In [ ]:

2.1.1 Test

In this section we are going to try to implement the Vasicek Model on the US Federal_Fund_Rates date that we will fetch using opensource Free API AlphaVantage : https://www.alphavantage.co/documentation/#

My goal, was to use the Maximum likelihood estimation Method f to estimate Vasicek parameters, and then stimulate different possible scenarios for our

function.

In [3]:
```
#Importing all libraries and relevant tools for conveniency

import requests
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize

# Using AlphaVantage free API Key and documentation we can import data from
api_key = 'MNBX49QP1JKFK3KG'
url = f'https://www.alphavantage.co/query?function=FEDERAL_FUNDS_RATE&apikey
response = requests.get(url)
data = response.json()


# Converting the JSON data to a DataFrame because dataframes are easier to w
ffr_df = pd.DataFrame(data['data'])
ffr_df['date'] = pd.to_datetime(ffr_df['date'])
ffr_df['value'] = pd.to_numeric(ffr_df['value'])
```

Let's display the rates we got from our 2 tables

1- Estimating the Parameters : in order to do so, we are going to use the Maximum likelihood estimation. First we use the Ito formula to $r_t \exp(-a\delta)$ to obtain the explicit form of the Vacicek rate

$$r_{t+\delta} = r_t e^{-a\delta} + k(1 - e^{-a\delta}) + \sigma\sqrt{\frac{1 - e^{-2a\delta}}{2a}}\varepsilon, \quad \varepsilon \sim \mathcal{N}(0,1)$$

with ϵ the Standard Normal Noise $\mathcal{N}(0,1)$

$$\log(L(a,b,\sigma)) = \sum_{t=1}^{n} \frac{1}{E\sqrt{2\pi}} \exp\left(-0.5\frac{(x - sig)^2}{E^2}\right)$$

with $E = V\sqrt{\frac{1-\exp(-2at)}{2a}}$
and $sig = r\exp(-a * t) + K(1 - \exp(-at))$

In [57]:
```
#We supposed that the volatility is constant ( which is not plausible)
data_ = ffr_df['value']
X_train = data_ [:100]
X_test = data_ [50:]


STD = np.std(ffr_df['value']/100)
v = STD* np.sqrt(12)
print(v)
```

```
0.12373113986190391
```

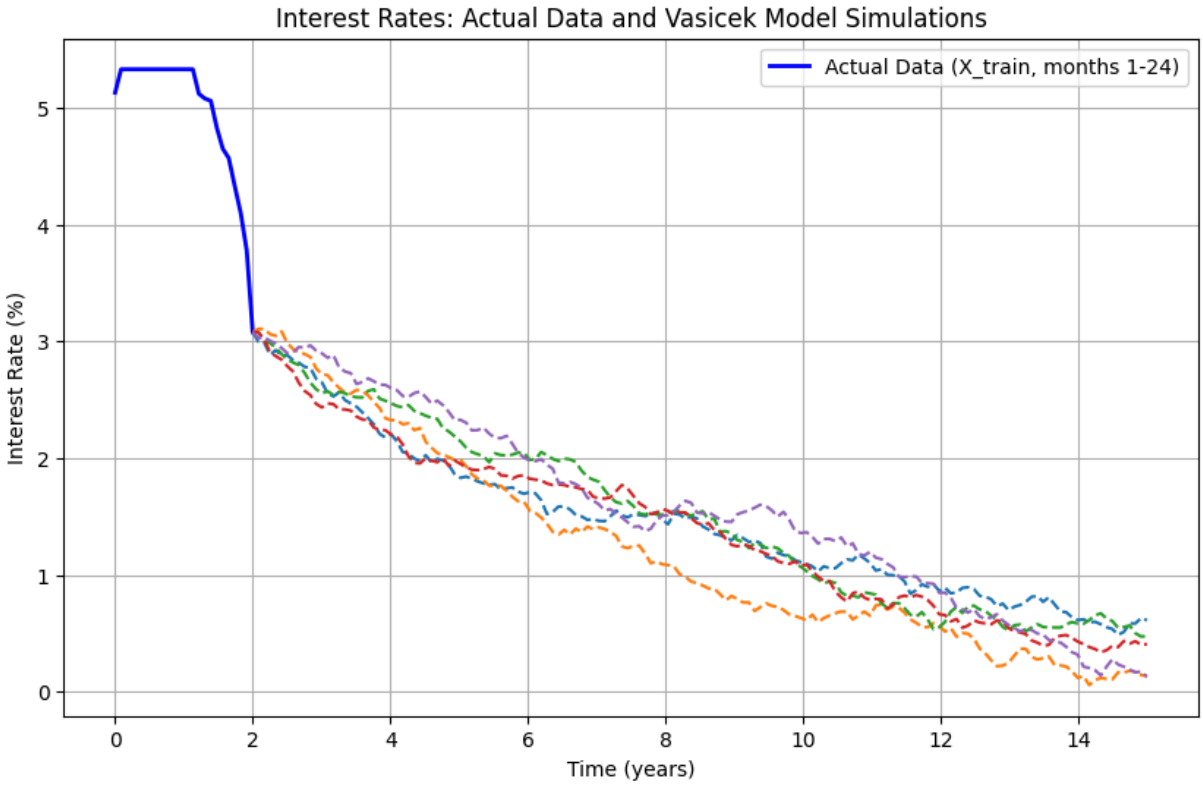
In [13]:
```
#Defining the normal distribution affiliated with the model
def N_Vas(x,a,k,t,r):

    E = v * np.sqrt((1 - np.exp(-2 * a * t)) / (2 * a))

    sig = r * np.exp(-a * t) + k * (1 - np.exp(-a * t))

    return ((1 / (E * np.sqrt(2 * np.pi))) * np.exp(-0.5 * ((x - sig) / E)**
# Defining the maximum likelihood estimation
def log_likelihood(parameters, X_train):
    a, k = parameters
    S=0
    t=1/12
    epsilon = 1e-15
    #Here I added a very small number because I had some cases where P was e
    for i in range (len(X_train)-1):
        P = N_Vas(X_train[i+1], a, k, t, X_train[i])
        S-=np.log(P + epsilon)
    return(S)
```

In [14]:
```
# Estimating the parameters with the constraints on a and k being positive
constraints = ({'type': 'ineq', 'fun': lambda params: np.exp(params[0])},  #
               {'type': 'ineq', 'fun': lambda params: params[1]})  # k >= 0

estimateresults = minimize(log_likelihood, [1, 2], args=(X_train,), constrai
estimated_parameters = estimateresults.x
# print the estimated parameters
estimated_a, estimated_k = estimated_parameters
print(f" the estimated parameter a is : {estimated_a} \n the estimated param
```

```
the estimated parameter a is : 0.1387347675353312
the estimated parameter b is : -3.8377500895705816e-13
```

In [69]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Our Vasicek parameters, we estimated a and k prior to that and calculated
a = 0.1387347675353312
b = -3.8377500895705816e-13
sigma = 0.12373113986190391
r0 = X_train[23]

# Simulation settings
T = (180 - 24) / 12   # Total time in years (13 years from month 24 to 180)
dt = 1 / 12  # Monthly time steps
N = int(T / dt)  # Number of steps (156 months or 13 years)
t = np.linspace(0, T, N)  # Time for the simulations (years)
num_simulations = 5
t_real = np.linspace(0, 2, 24)  # Time for the actual data (2 years)

# Vasicek Path
def sim_vasicek(a, b, sigma, r0, N, dt):
    r = np.zeros(N)
```

```
    r[0] = r0
    dW = np.random.normal(0, np.sqrt(dt), N)
    for i in range(1, N):
        dr = a * (b - r[i-1]) * dt + sigma * dW[i]
        r[i] = r[i-1] + dr
    return r

# Plot the actual data (X_train) for the first 24 months
plt.figure(figsize=(10,6))
plt.plot(t_real, X_train[:24], label='Actual Data (X_train, months 1-24)', c

# Plot multiple Vasicek simulations from month 24 to month 180
for i in range(num_simulations):
    r_vasicek = sim_vasicek(a, b, sigma, r0, N, dt)
    plt.plot(np.linspace(2, 15, N), r_vasicek, linestyle='--')
#you can add  label=f'Simulation {i+1}' inside the plt.plot to have the colo
plt.title('Interest Rates: Actual Data and Vasicek Model Simulations')
plt.xlabel('Time (years)')
plt.ylabel('Interest Rate (%)')
plt.grid(True)
plt.legend(loc='upper right')
plt.show()
```



This plot showcase the interest rate in the last 2 years (01-09-2022 to 01-09-2024) and then 10 different simulations for the next 15 upcoming years

This Model presents multiples limits :

- Constant votality : as you can see we assumed that volatility is constant which is not the case, volatility often

In [ ]: