MATH40082 Mini Assignment 2 Solution

10945438

1 Solution

In the given non-standard derivative-pricing model, in which the value of a derivative contract $V(r,t)$ can be solved for by considering

$$V(r,t) = P(r,t,T)\,E[g(R)], \tag{1}$$

where t is time, r is the interest rate, R (r at time T) is a normally distributed random variable given by $R \sim \mathcal{N}(f, v^2)$, g is the payoff function, and $P(r,t,T)$ is given by

$$P(r,t,T) = \exp\left(\frac{1}{4}k^2(t,T) - \frac{1}{2}n(r,t,T)\right), \tag{2}$$

the value of a \$1 cash-or-nothing Put option can be trivially solved for by considering g being (in the case of a put) at maturity $t = T$ equal to \$1 if $R < X_r$ and nothing otherwise (shifted Heaviside step), and as such the expectation in (1) is given by

$$\mathbb{E}[g(R)] = \int_{-\infty}^{\infty} g(R) N_p\!\left(\tfrac{r-f}{v}\right) dr = \int_{-\infty}^{X_r} N_p\!\left(\tfrac{r-f}{v}\right) dr = N_c\!\left(\tfrac{X_r-f}{v}\right) = N_c(h(r,t,T)). \tag{3}$$

where [Task 1] $h(r,t,T)$ is given by

$$\boxed{h(r,t,T) = \frac{X_r - f(r,t,T)}{v(t,T)}},$$

as N_p is the standard normal PDF function and N_c is the standard normal CDF function and as such is specified by

$$N_c(z) \;=\; \int_{-\infty}^{z} N_p(u)\,du \quad \text{(its CDF)}. \tag{4}$$

The solved Put option value is given by

$$V(r,t) = P(r,t,T)N(h(r,t,T)), \tag{5}$$

and k^2, n, f, and v are given functions that can be found in the appendix.

The [Task 2] value of the (relevant) Put option given $T = 8$, $r_0 = 0.0283$, $X_r = 0.05$, $\kappa = 0.2848$, $\theta = 0.067$, and $\sigma = 0.0378$ was found to be

$$\boxed{V(r = r_0, t = 0, T) = \$0.410705}.$$

The bond price $P(r, t = 0, T)$ and the option price $V(r, t = 0, T)$ were computed [Task 3] for 100 values of r in the range $r \in [0, 0.2]$. Figure 1 is a plot of $P(r, t = 0, T)$ and $V(r, t = 0, T)$ against r.

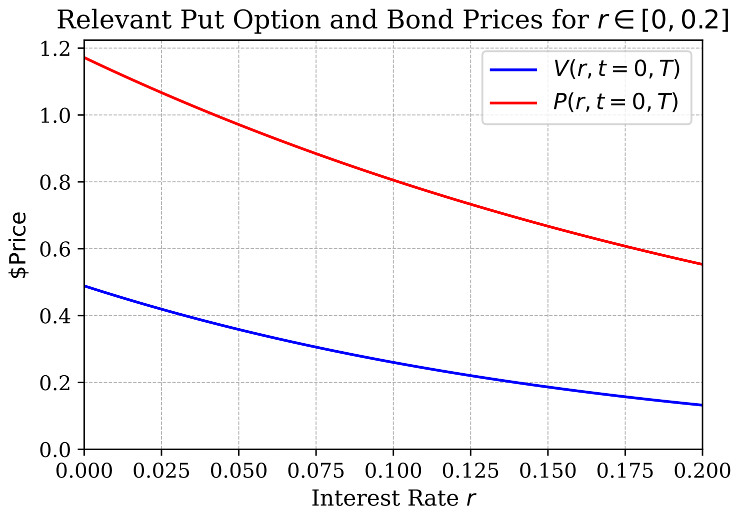

Figure 1: Relevant bond price $P(r, t = 0, T)$ and option price $V(r, t = 0, T)$ against interest rate r. [Task 3]

2 Code

The code to generate Table 1 is in Listing 1 and continued in 2.

Listing 1: Code for Mini-Assignment 2.

```
# Importing libraries
import math
from scipy.stats import norm
import numpy as np
import matplotlib.pyplot as plt

# Latex style for plots
plt.rcParams.update({
    "text.usetex": False,
    "font.family": "serif",
    "axes.labelsize": 12,
    "axes.titlesize": 14,
    "legend.fontsize": 12,
    "xtick.labelsize": 11,
    "ytick.labelsize": 11
})

# Defining relevant function arguments seperately
m = lambda r, t, T, kappa, theta: r*math.exp(-kappa*(T-t)) + theta*(1-math.exp(-kappa*(T-t))
    )

n = lambda r, t, T, kappa, theta: 0.5*r*(T-t) - ((theta-r)/kappa)*(1-math.exp(-3*kappa*(T-t)
    ))

k_squared = lambda t, T, kappa, sigma: ((sigma**2)/(2*(kappa**3)))*(5*math.exp(-kappa*(T-t))
     - 3*math.exp(-2*kappa*(T-t)) + 3*kappa*(T-t) - 2)

q  = lambda t, T, kappa, sigma: ((sigma**2)/(4*(kappa**2)))*((1-math.exp(-kappa*(T-t)))**3)


# Defining function for pure discount bond price
def P(r, t, T, kappa, theta, sigma):
  return math.exp(0.25*k_squared(t, T, kappa, sigma) - 0.5*n(r, t, T, kappa, theta))

# Defining function for mean of R under risk-neutral measure
def f(r, t, T, kappa, theta, sigma):
  return m(r, t, T, kappa, theta) - 0.5*q(t, T, kappa, sigma)

# Defining function for variance of R under risk-neutral measure
def v_squared(t, T, kappa, sigma):
  return ((sigma**2)/(3*kappa))*(1-math.exp(-3*kappa*(T-t)))

# Defining function for h argument in the Call and Put option pricing formulas
def h(X_r, r, t, T, kappa, theta, sigma):
  return (X_r-f(r, t, T, kappa, theta, sigma))/(v_squared(t, T, kappa, sigma)**0.5)

# Defining function for Put option price
def put_option_price(X_r, r, t, T, kappa, theta, sigma):
  return P(r, t, T, kappa, theta, sigma)*norm.cdf(h(X_r, r, t, T, kappa, theta, sigma))

#####
# Main to run the program
```

Listing 2: Code Continuation.

```
def main():
  # Evaluation of example put option price using the given parameters/constants
  kappa_val = 0.2848
  theta_val = 0.067
  sigma_val = 0.0378

  task_2_put_option_price = put_option_price(X_r=0.05, r=0.0283, t=0, T=8, kappa=kappa_val,
      theta=theta_val, sigma=sigma_val)

  print('Task 2: Put option price: %.6f' % task_2_put_option_price)

  ###
  # Put option and bond prices for different values of r
  option_price_array = np.array([])
  bond_price_array = np.array([])

  print('Task 3: Put option and bond prices for different values of r:')

  for r in np.linspace(0, 0.2, 100):
    option_price_r_i = put_option_price(X_r=0.05, r=r, t=0, T=8, kappa=kappa_val, theta=
        theta_val, sigma=sigma_val)
    bond_price_i = P(r=r, t=0, T=8, kappa=kappa_val, theta=theta_val, sigma=sigma_val)

    print('r: %.4f, Put option price: %.4f, Bond price: %.4f' % (r, option_price_r_i,
        bond_price_i))

    option_price_array = np.append(option_price_array, option_price_r_i)
    bond_price_array = np.append(bond_price_array, bond_price_i)

  # Plotting the put option and bond prices for different values of r, and saving the plot
      as a PDF
  # Clean style for the plot
  plt.figure(figsize=(6, 4))
  plt.plot(np.linspace(0, 0.2, 100), option_price_array, label=r'$V(r, t = 0, T)$', color='
      blue', linewidth=1.5)
  plt.plot(np.linspace(0, 0.2, 100), bond_price_array, label=r'$P(r, t = 0, T)$', color='red
      ', linewidth=1.5)
  plt.xlabel(r'Interest Rate $r$')
  plt.ylabel(r'$\$  \text{Price}$')
  plt.ylim(0)
  plt.xlim(0, 0.2)
  plt.title(r'Relevant Put Option and Bond Prices for $r \in [0, 0.2]$')
  plt.legend(frameon=True, loc='best')
  plt.grid(True, linestyle='dashed', linewidth=0.5)
  plt.savefig('task_3_plot.pdf', dpi=300, bbox_inches='tight')
  plt.show()

# Run main function
if __name__ == '__main__':
  main()
```

A Additional Given Equations

This section contains given formulae for the task, excluded from above for the sake of brevity.

$$
\begin{aligned}
f(r,t,T) &= m(r,t,T) \;-\; \tfrac{1}{2}\, q(t,T), \\
v^2(t,T) &= \tfrac{\sigma^2}{3\,\kappa} \left(1 - e^{-3\,\kappa\,(T-t)}\right), \\
m(r,t,T) &= r\, e^{-\kappa\,(T-t)} \;+\; \theta \left(1 - e^{-\kappa\,(T-t)}\right), \\
n(r,t,T) &= \tfrac{1}{2}\, r\,(T-t) \;-\; \tfrac{\theta - r}{\kappa} \left(1 - e^{-3\,\kappa\,(T-t)}\right), \\
k^2(t,T) &= \tfrac{\sigma^2}{2\,\kappa^3} \left(5\, e^{-\kappa\,(T-t)} \;-\; 3\, e^{-2\,\kappa\,(T-t)} \;+\; 3\,\kappa\,(T-t) \;-\; 2\right), \\
q(t,T) &= \tfrac{\sigma^2}{4\,\kappa^2} \left(1 - e^{-\kappa\,(T-t)}\right)^3.
\end{aligned}
$$