Homework 1

April 10, 2023

1 Homework 1

1.1 FINM 37500 - 2023

1.1.1 UChicago Financial Mathematics

- Mark Hendricks
- hendricks@uchicago.edu

2 Context

For use in these problems, consider the data below, discussed in Veronesi's *Fixed Income Securities* Chapters 9, 10. * interest-rate tree * current term structure

```
[ ]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import matplotlib as mpl
     %matplotlib inline
     plt.style.use('seaborn')
     mpl.rcParams['font.family'] = 'serif'
     import sympy
```

```
[ ]: rate_tree = pd.DataFrame({'0':[.0174,np.nan],'0.5':[.0339,.0095]})
     rate_tree.columns.name = 'time $t$'
     rate_tree.index.name = 'node'
     rate_tree.style.format('{:.2%}',na_rep='')
```

```
[ ]: <pandas.io.formats.style.Styler at 0x25e30b9b580>
```

The "tree" is displayed as a pandas dataframe, so it does not list "up" and "down" for the rows but rather an index of nodes. The meaning should be clear.

```
[ ]: term_struct = pd.DataFrame({'maturity':[.5,1,1.5],'price':[99.1338,97.8925,96.
       ↪1462]})
     term_struct['continuous ytm'] = -np.log(term_struct['price']/100) /␣
       ↪term_struct['maturity']
     term_struct.set_index('maturity',inplace=True)
term_struct.style.format({'price':'{:.4f}','continuous ytm':'{:.2%}'}).
 ↪format_index('{:.1f}')
```

[ ]:

```
<pandas.io.formats.style.Styler at 0x25e30be7220>
```

This is the current term-structure observed at $t = 0$.

3 1. Pricing a Swap

3.0.1 1.1

Calculate the tree of bond prices for the 2-period, $T = 1$, bond.

3.0.2 1.2

What is the risk-neutral probability of an upward movement of interest rates at $t = .5$?

3.1 The option contract

Consider a single-period swap that pays at time period 1 ($t = 0.5$), the expiration payoff (and thus terminal value) is * Payoff $= \frac{100}{2}(r_1 - c)$ * with $c = 2\%$ * payments are semiannual

Take the viewpoint of a fixed-rate payer, floating rate receiver.

3.1.1 1.3

What is the replicating trade using the two bonds (period 1 and period 2)?

3.1.2 1.4

What is the price of the swap?

[ ]:

```
# 1.1 Calculating the tree of bond prices

dt = .5
r0 = .0174
Rates = [.0339,.0095]
Z = np.exp(-r0 * .5)
A = np.exp(r0*.5)
P_1u = 100* np.exp(-rate_tree.iloc[0,1]*dt)
P_1d = 100*np.exp(-rate_tree.iloc[1,1]*dt)
P_df = pd.DataFrame([P_1u,P_1d], columns=['Period_1'])
P_df.insert(0,'Period_0',97.8925)
P_df.iloc[1,0] = 0
P_df['Period_2'] = 100
```

[ ]:

```
P_df
```

[ ]:

```
   Period_0   Period_1  Period_2
0   97.8925  98.319284       100
```

```
1      0.0000   99.526126        100
```

3.2 1.2

- The risk neutral probability of an upward movement in interest rates at T = .5 is equal to this:

$p^* = \frac{AP_{0|2}-P_{1d|2}}{P_{1u|2}-P_{1d|2}}$

[ ]:

```
p = (A*P_df.loc[0,'Period_0'] - P_1d)/(P_1u-P_1d)
print(f'The probability of an upward movement in interest rates at time t=0.5␣
  ↪is {np.round(p,5)}')
```

```
The probability of an upward movement in interest rates at time t=0.5 is 0.64486
```

[ ]:

```
P_float, P_fixed_u,P_fixed_d,Vswap_u, Vswap_d = sympy.symbols("P_float␣
  ↪P_fixed_u P_fixed_d V_swap_u V_swap_d")
A = sympy.Matrix([[100,98.3193],[100,99.5261]])
b = sympy.Matrix([50*(.0339-.02),50*(.0095-.02)])
alpha,beta = A.solve(b)
```

[ ]:

```
position_df = pd.DataFrame(index=['1-period bond','2-period␣
  ↪bond'],columns=['price','position','$ holding'],dtype=float)
position_df['price'] = term_struct.iloc[0:2,0].values
position_df["position"] = [alpha,beta]
position_df['$ holding'] = position_df['price']*position_df['position']
position_df.loc['net','$ holding'] = position_df['$ holding'].sum()
position_df.style.format('${:,.4f}')
```

```
[ ]: <pandas.io.formats.style.Styler at 0x25e30bdd760>
```

- The value of the time-0 swap is $0.2595

4 2. Using the Swap as the Underlying

As in the note, W.1, consider pricing the followign interest-rate option, * Payoff is $100\max(r_K - r_1, 0)$ * strike is r_K is 2% * expires at period 1, $(t = 0.5)$

Unlike the note, price it with the swap used as the underlying, not the two-period $(t = 1)$ bond. You will once again use the period-1 $(t = 0.5)$ bond as the cash account for the no-arbitrage pricing.

So instead of replicating the option with the two treasuries, now you're replicating/pricing it with a one-period bond and two-period swap.

4.0.1 2.1

Display the tree of swap prices.

4.0.2 2.2

What is the risk-neutral probability of an upward movement at $t = .5$ implied by the underlying swap tree?

Is this the same as the risk-neutral probability we found when the bond was used as the underlying?

4.0.3 2.3

What is the price of the rate option? Is it the same as we calculated in the note, W.1.?

[ ]:
```
swap_0 = position_df.iloc[2,2]
rk = .02

payoffs = 100*(rk - np.array(rate_tree.iloc[:,1]))
payoffs[payoffs<0] = 0
```

[ ]:
```
swap_tree = pd.DataFrame(np.array(b),columns=['Period_1'] )
swap_tree.insert(0,'Period_0',swap_0)

swap_tree.iloc[1,0] = 0

swap_tree
```

[ ]:
```
            Period_0           Period_1
0  0.259491206443485   0.695000000000000
1                  0  -0.525000000000000
```

[ ]:
```
P_UP = (swap_0-(swap_tree.iloc[1,1]))/(swap_tree.iloc[0,1]-(swap_tree.
 ↪iloc[1,1]))
P_UP
```

[ ]: 0.643025579052037

- **Yes**, the probability is exactly the same as the risk-neutral probability of an interest rate increase at time = .5.

4.1 2.3 | Pricing The Rate Option

[ ]:
```
payoffs
```

[ ]:
```
array([0.  , 1.05])
```

[ ]:
```
A_ = sympy.Matrix([[100,.695],[100,-.525]])
b_ = sympy.Matrix(payoffs)
alpha_, beta_ = A_.solve(b_)
A_.solve(b_)
```

[ ]:
$$\begin{bmatrix} 0.00598155737704918 \\ -0.860655737704918 \end{bmatrix}$$

[ ]:

```
position = pd.DataFrame(index=['1-period bond','2-period␣
 ↪swap'],columns=['price','position','$ holding'],dtype=float)

position.loc['1-period bond','price'] = term_struct.iloc[0,0]
position.loc['2-period swap','price'] = swap_0
position['position'] = [alpha_,beta_]
position['$ holding'] = position['price']*position['position']
position.loc['net','$ holding'] = position['$ holding'].sum()
position.style.format('${:,.4f}')
```

[ ]:

```
<pandas.io.formats.style.Styler at 0x25e30be7a30>
```

- **Yes** the price of the interest rate floor price is the same as we calculated in workbook 1.

Answer The price of the interest rate floor is $0.3696.

5 3. Pricing a Call on a Bond

Try using the same tree to price a call on the period-2 bond, (1-year), at period 1 (6-months). * Payoff $= \max(P_{1|2} - K, 0)$ * Strike = $99.00

5.0.1 3.1

What is the replicating trade using the two bonds (period 1 and period 2) as above? (That is, we are no longer using the swap as the underlying.)

5.0.2 3.2

What is the price of the European call option? * expiring at $T = .5$ * written on the bond maturing in 2 periods, $(t = 1)$

[ ]:

```
P_df
```

[ ]:

```
   Period_0   Period_1  Period_2
0   97.8925  98.319284       100
1    0.0000  99.526126       100
```

[ ]:

```
K = 99
payoffs_call = (P_df['Period_1'].values - K)
payoffs_call[payoffs_call <0] = 0
payoffs_call
A_call = sympy.Matrix([[100,98.319284],[100,99.526126]])
b_call = sympy.Matrix(payoffs_call)
alpha_c, beta_c = A_call.solve(b_call)
```

[ ]:

```
position_call = pd.DataFrame(index=['1-period bond','2-period␣
 ↪bond'],columns=['price','position','$ holding'],dtype=float)
position_call['price'] = term_struct.iloc[0:2,0].values
position_call["position"] = [alpha_c,beta_c]
```

```
position_call['$ holding'] = position_call['price']*position_call['position']
position_call.loc['net','$ holding'] = position_call['$ holding'].sum()
position_call.style.format('${:,.4f}')
```

```
[ ]: <pandas.io.formats.style.Styler at 0x25e30b0a070>
```

Answer: The price of the call is \$0.1852.

6 4 Two-Period Tree

Consider an expanded, **2 period** tree. (Two periods of uncertainty, so with the starting point, three periods total.)

[ ]:

```
new_col = pd.Series([.05,.0256,.0011],name='1')
rate_tree_multi = pd.concat([rate_tree,new_col],ignore_index=True,axis=1)
rate_tree_multi.columns = pd.Series(['0','0.5','1'],name='time $t$')
rate_tree_multi.index.name = 'node'
rate_tree_multi.style.format('{:.2%}',na_rep='')
```

```
[ ]: <pandas.io.formats.style.Styler at 0x25e30bfcb80>
```

[ ]:

```

```

6.0.1 4.1

Calculate and show the tree of prices for the 3-period bond, $T = 1.5$.

6.0.2 4.2

Report the risk-neutral probability of an up movement at $t = 1$.

(The risk-neutral probability of an up movement at $t = 0.5$ continues to be as you calculated in 2.3.

6.0.3 4.3

Calculate the price of the European **call** option? * expiring at $T = 1$ * written on the bond maturing in 3 periods, ($t = 1.5$)

6.0.4 4.4

Consider a finer time grid. Let dt in the tree now be 1/30 instead of 0.5.

Using this smaller time step, compute the $t = 0$ price of the following option: * option expires at $t = 1$ * written on bond maturing at \$t=1.5

[ ]:

```
P_2p = P_df[['Period_1']]
P_2p.insert(0,'Period_0',96.1462)
P_2p.iloc[1,0] = 0
P_2p.loc[0,'Period_2'] = 100*np.exp(-0.05*dt)
```

```
P_2p.loc[1,'Period_2'] = 100*np.exp(-.0256*dt)
P_2p.loc[2,'Period_2'] = 100*np.exp(-.0011*dt)
P_2p['Period_3'] = 100
```

```
[ ]: P_2p
```

```
[ ]:    Period_0   Period_1   Period_2  Period_3
     0   96.1462  98.319284  97.530991       100
     1    0.0000  99.526126  98.728157       100
     2       NaN        NaN  99.945015       100
```

7 5 American Style

7.0.1 5.1

Use the two-period tree from part 4, but this time to price an American-style **put** option.

Use a grid of $dt = .5$. * What is its value at $t = 0$? * Which nodes would you exercise it early?

7.0.2 5.2

Change the grid to $dt = 1/30$, as in 4.4. * What is its value at $t = 0$? * Make a visualization showing which nodes have early exercise. (I suggest using a dataframe and the `heatmap` from `seaborn`.