MF728_hw1-Copy2

February 8, 2024

0.1 MF 728: Fixed Income

0.1.1 Problem set # 1

1. Yield Curve Construction: Consider the following table of USD swap rates:

Term	Rate (%)
1Y	2.8438
2Y	3.060
3Y	3.126
4Y	3.144
5Y	3.150
7Y	3.169
10Y	3.210
30Y	3.237

Note: You may assume that these swaps pay coupons semi-annually (every 6 months). For simplicity, you may use a year fraction of 0.5 in all swap coupon payments.

(a) Extract the constant forward rate for the first year that enables you to match the 1Y market swap rate.

[74]:
```
import numpy as np
from scipy.optimize import fsolve

#
def swap_present_value(f1, sw1):
    #   PV
    PV = 0.5 * sw1 / (1 + 0.5 * f1) + 0.5 * sw1 / (1 + 0.5 * f1)**2
    #  PV
    return PV - 0.5 * f1 / (1 + 0.5 * f1) - 0.5 * f1 / (1 + 0.5 * f1)**2

#   swap_to_forward_1Y   args    sw1
def swap_to_forward_1Y(sw1):
    # fsolve  swap_present_value   sw1
    forward_rate = fsolve(swap_present_value, x0=sw1, args=(sw1,))
    return forward_rate[0]
```

[75]:
```
f1 = swap_to_forward_1Y(2.8438 / 100)
print(f1)
```

```
0.028437999999999998
```

(b) Holding this first year forward rate fixed, find the forward rate from one year to two years that enables you to match the two year swap (while also matching the one year).

[76]:
```
#
def swap_present_value(f2, sw2, f1):
    #
    PV1 = sw2 / (1 + 0.5 * f1) + sw2 / (1 + 0.5 * f1)**2 + sw2 / ((1 + 0.5 *␣
 ↪f1)**2 * (1 + 0.5 * f2)) + sw2 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2)
    #
    PV2 = f1 / (1 + 0.5 * f1) + f1 / (1 + 0.5 * f1)**2 + f2 / ((1 + 0.5 *␣
 ↪f1)**2 * (1 + 0.5 * f2)) + f2 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2)
    #
    return PV1 - PV2

#
def swap_to_forward_2Y(sw2, f1):
    #  fsolve
    forward_rate_2Y = fsolve(swap_present_value, x0=sw2, args=(sw2, f1))
    return forward_rate_2Y[0]

#         f1
f1 = 2.8438 / 100  #
#
f2 = swap_to_forward_2Y(3.060 / 100, f1)
#
D1 = 1 / ((1 + 0.5 * f1)**2)  #     D1
D2 = D1 / ((1 + 0.5 * f2)**2)

f2, D2
```

[76]:
```
(0.03283113038626899, 0.9410092691367571)
```

(c) Continue this process and extract piecewise constant forward rates for the entire curve. Comment on the forward rates vs. the swap rates.

[77]:
```
#
def swap_present_value_3Y(f3, sw3, f1, f2):
    #
    PV1 = sw3 / (1 + 0.5 * f1) + sw3 / (1 + 0.5 * f1)**2 + sw3 / ((1 + 0.5 *␣
 ↪f1)**2 * (1 + 0.5 * f2)) + sw3 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2) +␣
 ↪sw3 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2 * (1 + 0.5 * f3)) + sw3 / ((1 +␣
 ↪0.5 * f1)**2 * (1 + 0.5 * f2)**2 * (1 + 0.5 * f3)**2)
    #print(PV1)
```

```
    #
    PV2 = f1 / (1 + 0.5 * f1) + f1 / (1 + 0.5 * f1)**2 + f2 / ((1 + 0.5 *␣
 ↪f1)**2 * (1 + 0.5 * f2)) + f2 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2) + f3␣
 ↪/ ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2 * (1 + 0.5 * f3)) + f3 / ((1 + 0.5␣
 ↪* f1)**2 * (1 + 0.5 * f2)**2 * (1 + 0.5 * f3)**2)
    #print(PV2)
    #
    return PV1 - PV2

#
def swap_to_forward_3Y(sw3, f1, f2):
    #  fsolve
    forward_rate_3Y = fsolve(swap_present_value_3Y, x0=sw3, args=(sw3, f1, f2))
    return forward_rate_3Y[0]

#       f1 f2
sw3 = 3.126 / 100

#
f3 = swap_to_forward_3Y(sw3, f1, f2)

#
D3 = D2 / ((1 + 0.5 * f3)**2)

f3, D3
```

[77]: (0.03264530551203542, 0.9110258289278597)

[78]:
```
#    swap_present_value
def swap_present_value_4Y(f4, sw4, f1, f2, f3):
    PV = sw4/(1+0.5*f1) + sw4/(1+0.5*f1)**2 + sw4/((1+0.5*f1)**2*(1+0.5*f2)) +␣
 ↪sw4/((1+0.5*f1)**2*(1+0.5*f2)**2) + sw4/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.
 ↪5*f3)) + sw4/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2) + sw4/((1+0.
 ↪5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)) + sw4/((1+0.5*f1)**2*(1+0.
 ↪5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)**2)
    PV2 = f1/(1+0.5*f1) + f1/(1+0.5*f1)**2 + f2/((1+0.5*f1)**2*(1+0.5*f2)) + f2/
 ↪((1+0.5*f1)**2*(1+0.5*f2)**2) + f3/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3))␣
 ↪+ f3/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2) + f4/((1+0.5*f1)**2*(1+0.
 ↪5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)) + f4/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.
 ↪5*f3)**2*(1+0.5*f4)**2)
    return PV - PV2

#    swap_to_forward_4Y
def swap_to_forward_4Y(sw4, f1, f2, f3):
    #   lambda        swap_present_value
    forward_rate_4Y = fsolve(lambda f4: swap_present_value_4Y(f4, sw4, f1, f2,␣
 ↪f3), sw4)
```

```
    return forward_rate_4Y[0]

sw4 = 3.144 / 100
f4 = swap_to_forward_4Y(sw4, f1, f2, f3)

#
D4 = D3 / ((1 + 0.5 * f4)**2)

f4,D4
```

[78]: (0.03201590223021914, 0.8825442234289774)

[79]:
```
#    swap_present_value
def swap_present_value_5Y(f5, sw5, f1, f2, f3, f4):
    PV = sw5*sum((1+0.5*f1)**-i for i in range(1, 3)) + sw5/((1+0.5*f1)**2*(1+0.
 ↪5*f2)) + sw5/((1+0.5*f1)**2*(1+0.5*f2)**2) + sw5/((1+0.5*f1)**2*(1+0.
 ↪5*f2)**2*(1+0.5*f3)) + sw5/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2) + sw5/
 ↪((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)) + sw5/((1+0.
 ↪5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)**2) + sw5/((1+0.
 ↪5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)**2*(1+0.5*f5)) + sw5/((1+0.
 ↪5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)**2*(1+0.5*f5)**2)
    PV2 = f1/(1+0.5*f1) + f1/(1+0.5*f1)**2 + f2/((1+0.5*f1)**2*(1+0.5*f2)) + f2/
 ↪((1+0.5*f1)**2*(1+0.5*f2)**2) + f3/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3))␣
 ↪+ f3/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2) + f4/((1+0.5*f1)**2*(1+0.
 ↪5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)) + f4/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.
 ↪5*f3)**2*(1+0.5*f4)**2) + f5/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.
 ↪5*f4)**2*(1+0.5*f5)) + f5/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.
 ↪5*f4)**2*(1+0.5*f5)**2)
    return PV - PV2

#    swap_to_forward_5Y
def swap_to_forward_5Y(sw5, f1, f2, f3, f4):
    #   lambda        swap_present_value
    forward_rate_5Y = fsolve(lambda f5: swap_present_value_5Y(f5, sw5, f1, f2,␣
 ↪f3, f4), sw5)
    return forward_rate_5Y[0]

#      f5
f5 = swap_to_forward_5Y(0.0315, f1, f2, f3, f4)

#   D5
D5 = D4 / (1 + 0.5 * f5)**2

f5,D5
```

[79]: (0.031760049949048276, 0.8551683806917486)

[80]:
```
#    swap_present_value
def swap_present_value_7Y(f7, sw7, f1, f2, f3, f4, f5, D1, D2, D3, D4, D5):
    PV = (sw7 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
          sw7 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
          sw7 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
          sw7 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
          sw7 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
          sw7 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)))
    PV2 = (f1 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
           f2 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
           f3 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
           f4 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
           f5 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
           f7 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)))
    return PV - PV2

#    swap_to_forward_7Y
def swap_to_forward_7Y(sw7, f1, f2, f3, f4, f5, D1, D2, D3, D4, D5):
    #   lambda        swap_present_value
    forward_rate_7Y = fsolve(lambda f7: swap_present_value_7Y(f7, sw7, f1, f2,␣
 ↪f3, f4, f5, D1, D2, D3, D4, D5), sw7)
    return forward_rate_7Y[0]

#      f7
f7 = swap_to_forward_7Y(0.03169, f1, f2, f3, f4, f5, D1, D2, D3, D4, D5)

#   D7
D7 = D5 / (1 + f7 * 0.5) ** 4

f7,D7
```

[80]:
```
(0.03222150590023792, 0.802208843979025)
```

[81]:
```
#    swap_present_value
def swap_present_value_10Y(f10, sw10, f1, f2, f3, f4, f5, f7, D1, D2, D3, D4,␣
 ↪D5, D7):
    PV = (sw10 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
          sw10 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
          sw10 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
          sw10 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
          sw10 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
          sw10 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)) +
          sw10 * D7 * sum((1 + 0.5 * f10) ** -i for i in range(1, 7)))
    PV2 = (f1 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
           f2 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
           f3 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
           f4 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
```

```
            f5 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
            f7 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)) +
            f10 * D7 * sum((1 + 0.5 * f10) ** -i for i in range(1, 7)))
    return PV - PV2

#    swap_to_forward_10Y
def swap_to_forward_10Y(sw10, f1, f2, f3, f4, f5, f7, D1, D2, D3, D4, D5, D7):
    #   lambda        swap_present_value
    forward_rate_10Y = fsolve(lambda f10: swap_present_value_10Y(f10, sw10, f1,␣
 ↪f2, f3, f4, f5, f7, D1, D2, D3, D4, D5, D7), sw10)
    return forward_rate_10Y[0]

#      f10
f10 = swap_to_forward_10Y(0.0321, f1, f2, f3, f4, f5, f7, D1, D2, D3, D4, D5,␣
 ↪D7)

#   D10
D10 = D7 / (1 + f10 * 0.5) ** 6
f10,D10
```

[81]:
```
(0.033225985137036694, 0.7266972400950599)
```

[82]:
```
# Standalone swap_present_value function
def swap_present_value_30Y(f30, sw30, f1, f2, f3, f4, f5, f7, f10, D1, D2, D3,␣
 ↪D4, D5, D7, D10):
    PV = (sw30 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
          sw30 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
          sw30 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
          sw30 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
          sw30 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
          sw30 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)) +
          sw30 * D7 * sum((1 + 0.5 * f10) ** -i for i in range(1, 7)) +
          sw30 * D10 * sum((1 + 0.5 * f30) ** -i for i in range(1, 41)))
    PV2 = (f1 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
           f2 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
           f3 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
           f4 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
           f5 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
           f7 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)) +
           f10 * D7 * sum((1 + 0.5 * f10) ** -i for i in range(1, 7)) +
           f30 * D10 * sum((1 + 0.5 * f30) ** -i for i in range(1, 41)))
    return PV - PV2

# Refactored swap_to_forward_30Y function
def swap_to_forward_30Y(sw30, f1, f2, f3, f4, f5, f7, f10, D1, D2, D3, D4, D5,␣
 ↪D7, D10):
```

```
    forward_rate_30Y = fsolve(lambda f30: swap_present_value_30Y(f30, sw30, f1,␣
 ↪f2, f3, f4, f5, f7, f10, D1, D2, D3, D4, D5, D7, D10), sw30)
    return forward_rate_30Y[0]

# Using the refactored function to calculate f30
f30 = swap_to_forward_30Y(0.03237, f1, f2, f3, f4, f5, f7, f10, D1, D2, D3, D4,␣
 ↪D5, D7, D10)

# Calculating D30
D30 = D10 / (1 + f30 * 0.5) ** 20
f30,D30
```

[82]:
```
(0.03258650990142389, 0.5259841369171494)
```

[83]:
```
f_list=[f1,f2,f3,f4,f5,f7,f10,f30]
f_list
```

[83]:
```
[0.028437999999999998,
 0.03283113038626899,
 0.03264530551203542,
 0.03201590223021914,
 0.031760049949048276,
 0.03222150590023792,
 0.033225985137036694,
 0.03258650990142389]
```

(d) Compute the fair market, breakeven swap rate of a 15Y swap. That is, find the swap rate that equates the present values of the fixed and floating legs.

[84]:
```
def sw15(f1, f2, f3, f4, f5, f7, f10, f30, D1, D2, D3, D4, D5, D7, D10):
    # Calculate the numerator of PV
    numerator = (
        f1 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
        f2 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
        f3 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
        f4 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
        f5 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
        f7 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)) +
        f10 * D7 * sum((1 + 0.5 * f10) ** -i for i in range(1, 7)) +
        f30 * D10 * sum((1 + 0.5 * f30) ** -i for i in range(1, 11))
    )

    # Calculate the denominator of PV
    denominator = (
        sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
        D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
        D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
        D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
```

```
        D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
        D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)) +
        D7 * sum((1 + 0.5 * f10) ** -i for i in range(1, 7)) +
        D10 * sum((1 + 0.5 * f30) ** -i for i in range(1, 11))
    )

    # Calculate PV by dividing the numerator by the denominator
    PV = numerator / denominator

    return PV
sw15(f1, f2, f3, f4, f5, f7, f10, f30, D1, D2, D3, D4, D5, D7, D10)
```

[84]:
```
0.03223672517257003
```

(e) Compute discount factors. Compute zero rates by finding the constant rate that leads to the calibrated discount factors. Comment on the differences in the zero rates and swap rates.

[85]:
```
D_list=[D1,D2,D3,D4,D5,D7,D10,D30]
zero_list = [np.log(D)/-T for D,T in zip(D_list,[1,2,3,4,5,7,10,30])]
zero_list
```

[85]:
```
[0.028237716361644027,
 0.030401144570267404,
 0.031061343281080404,
 0.03123659497399354,
 0.03129137859064855,
 0.0314837572929969,
 0.031924533922686626,
 0.021416140821712493]
```

Differences between Zero Rates and Swap Rates:

- Level of Rates: The zero rates and swap rates follow a similar trend up to the 10Y maturity, where zero rates increase with maturity, reflecting a normal upward-sloping yield curve. However, for the 30Y maturity, the zero rate significantly drops compared to the swap rate, indicating a discrepancy in long-term interest rate expectations or market conditions.
- Market Expectations and Liquidity: Swap rates are influenced by the expectations of future interest rates, credit risk, and liquidity in the swap market. In contrast, zero rates are purely derived from the current market prices of zero-coupon bonds or derived from bond prices, reflecting the market's view on future interest rates without the swap market's credit and liquidity considerations.

(f) Shift all forward rates up 100 basis points and re-calculate the breakeven swap rates for each benchmark point. Generate a table of new swap rates. Are these rates equivalent to having shifted the swap rates directly?

[86]:
```
def for_up_swr(f_list, basis_list):
```

```
    for i in range(len(f_list)):
        f_list[i] += basis_list[i]

    f1, f2, f3, f4, f5, f7, f10, f30 = f_list

    sw1 = f1 * sum((1+0.5*f1)**-i for i in range(1,3)) / sum((1+0.5*f1)**-i for i in range(1,3))

    D1 = 1 / (1+0.5*f1)**2

    sw2 = f1 * sum((1+0.5*f1)**-i for i in range(1,3)) + f2 * D1 * sum((1+0.5*f2)**-i for i in range(1,3))
    sw2 /= sum((1+0.5*f1)**-i for i in range(1,3)) + D1 * sum((1+0.5*f2)**-i for i in range(1,3))

    D2 = D1 / (1+0.5*f2)**2

    sw3 = f1 * sum((1+0.5*f1)**-i for i in range(1,3)) + f2 * D1 * sum((1+0.5*f2)**-i for i in range(1,3)) + f3 * D2 * sum((1+0.5*f3)**-i for i in range(1,3))
    sw3 /= sum((1+0.5*f1)**-i for i in range(1,3)) + D1 * sum((1+0.5*f2)**-i for i in range(1,3)) + D2 * sum((1+0.5*f3)**-i for i in range(1,3))

    D3 = D2 / (1+0.5*f3)**2

    sw4 = f1 * sum((1+0.5*f1)**-i for i in range(1,3)) + f2 * D1 * sum((1+0.5*f2)**-i for i in range(1,3)) + f3 * D2 * sum((1+0.5*f3)**-i for i in range(1,3)) + f4 * D3 * sum((1+0.5*f4)**-i for i in range(1,3))
    sw4 /= sum((1+0.5*f1)**-i for i in range(1,3)) + D1 * sum((1+0.5*f2)**-i for i in range(1,3)) + D2 * sum((1+0.5*f3)**-i for i in range(1,3)) + D3 * sum((1+0.5*f4)**-i for i in range(1,3))

    D4 = D3 / (1+0.5*f4)**2

    sw5 = f1 * sum((1+0.5*f1)**-i for i in range(1,3)) + f2 * D1 * sum((1+0.5*f2)**-i for i in range(1,3)) + f3 * D2 * sum((1+0.5*f3)**-i for i in range(1,3)) + f4 * D3 * sum((1+0.5*f4)**-i for i in range(1,3)) + f5 * D4 * sum((1+0.5*f5)**-i for i in range(1,3))
    sw5 /= sum((1+0.5*f1)**-i for i in range(1,3)) + D1 * sum((1+0.5*f2)**-i for i in range(1,3)) + D2 * sum((1+0.5*f3)**-i for i in range(1,3)) + D3 * sum((1+0.5*f4)**-i for i in range(1,3)) + D4 * sum((1+0.5*f5)**-i for i in range(1,3))

    D5 = D4 / (1+0.5*f5)**2
```

```
    sw7 = f1 * sum((1+0.5*f1)**-i for i in range(1,3)) + f2 * D1 * sum((1+0.
↪5*f2)**-i for i in range(1,3)) + f3 * D2 * sum((1+0.5*f3)**-i for i in␣
↪range(1,3)) + f4 * D3 * sum((1+0.5*f4)**-i for i in range(1,3)) + f5 * D4 *␣
↪sum((1+0.5*f5)**-i for i in range(1,3)) + f7 * D5 * sum((1+0.5*f7)**-i for i␣
↪in range(1,5))
    sw7 /= sum((1+0.5*f1)**-i for i in range(1,3)) + D1 * sum((1+0.5*f2)**-i␣
↪for i in range(1,3)) + D2 * sum((1+0.5*f3)**-i for i in range(1,3)) + D3 *␣
↪sum((1+0.5*f4)**-i for i in range(1,3)) + D4 * sum((1+0.5*f5)**-i for i in␣
↪range(1,3)) + D5 * sum((1+0.5*f7)**-i for i in range(1,5))

    D7 = D5 / (1+0.5*f7)**4

    sw10 = f1 * sum((1+0.5*f1)**-i for i in range(1,3)) + f2 * D1 * sum((1+0.
↪5*f2)**-i for i in range(1,3)) + f3 * D2 * sum((1+0.5*f3)**-i for i in␣
↪range(1,3)) + f4 * D3 * sum((1+0.5*f4)**-i for i in range(1,3)) + f5 * D4 *␣
↪sum((1+0.5*f5)**-i for i in range(1,3)) + f7 * D5 * sum((1+0.5*f7)**-i for i␣
↪in range(1,5)) + f10 * D7 * sum((1+0.5*f10)**-i for i in range(1,7))
    sw10 /= sum((1+0.5*f1)**-i for i in range(1,3)) + D1 * sum((1+0.5*f2)**-i␣
↪for i in range(1,3)) + D2 * sum((1+0.5*f3)**-i for i in range(1,3)) + D3 *␣
↪sum((1+0.5*f4)**-i for i in range(1,3)) + D4 * sum((1+0.5*f5)**-i for i in␣
↪range(1,3)) + D5 * sum((1+0.5*f7)**-i for i in range(1,5)) + D7 * sum((1+0.
↪5*f10)**-i for i in range(1,7))

    D10 = D7 / (1+0.5*f10)**6

    sw30 = f1 * sum((1+0.5*f1)**-i for i in range(1,3)) + f2 * D1 * sum((1+0.
↪5*f2)**-i for i in range(1,3)) + f3 * D2 * sum((1+0.5*f3)**-i for i in␣
↪range(1,3)) + f4 * D3 * sum((1+0.5*f4)**-i for i in range(1,3)) + f5 * D4 *␣
↪sum((1+0.5*f5)**-i for i in range(1,3)) + f7 * D5 * sum((1+0.5*f7)**-i for i␣
↪in range(1,5)) + f10 * D7 * sum((1+0.5*f10)**-i for i in range(1,7)) + f30 *␣
↪D10 * sum((1+0.5*f30)**-i for i in range(1,41))
    sw30 /= sum((1+0.5*f1)**-i for i in range(1,3)) + D1 * sum((1+0.5*f2)**-i␣
↪for i in range(1,3)) + D2 * sum((1+0.5*f3)**-i for i in range(1,3)) + D3 *␣
↪sum((1+0.5*f4)**-i for i in range(1,3)) + D4 * sum((1+0.5*f5)**-i for i in␣
↪range(1,3)) + D5 * sum((1+0.5*f7)**-i for i in range(1,5)) + D7 * sum((1+0.
↪5*f10)**-i for i in range(1,7)) + D10 * sum((1+0.5*f30)**-i for i in␣
↪range(1,41))

    return sw1,sw2,sw3,sw4,sw5,sw7,sw10,sw30

basis_list = [0.01 for i in range(8)]
```

[87]:
```
import pandas as pd

original_swaps = [2.8438, 3.060, 3.126, 3.144, 3.150, 3.169, 3.210, 3.237]
original_swaps = [i/100 for i in original_swaps]
```

```
modified_swaps = for_up_swr(f_list, basis_list)

rates = {
    'Term': [1, 2, 3, 4, 5, 7, 10, 30],
    'Original': [round(r, 4) for r in original_swaps],
    'Modified': [round(r, 4) for r in modified_swaps]
}

df = pd.DataFrame(rates)
print("\nOriginal vs Modified Swap Rates:\n")
print(df)
```

```
Original vs Modified Swap Rates:

   Term  Original  Modified
0     1    0.0284    0.0384
1     2    0.0306    0.0406
2     3    0.0313    0.0412
3     4    0.0314    0.0414
4     5    0.0315    0.0415
5     7    0.0317    0.0417
6    10    0.0321    0.0421
7    30    0.0324    0.0423
```

The modified swap rates produced by for_up_swr() are equivalent to having directly shifted the swap rates by 0.01 basis points. The end result is the same in both cases.

Shifting the forward rates and recalculating is equivalent to shifting the swap rates directly.

(g) Consider a bearish steepener to the swap rates, that is perform the following shifts on each swap rate:

Term	Rate change (bps)
1Y	+0
2Y	+0
3Y	+0
4Y	+5
5Y	+10
7Y	+15
10Y	+25
30Y	+50

[88]:
```
swap_list=[0.028438,0.03060,0.03126,0.03144,0.03150,0.03169,0.03210,0.03237]

basis_list=[0,0,0,0.0005,0.001,0.0015,0.0025,0.005]
```

```
sw1,sw2,sw3,sw4,sw5,sw7,sw10,sw30=[sw+basis for sw,basis in␣
 ↪zip(swap_list,basis_list)]
sw1,sw2,sw3,sw4,sw5,sw7,sw10,sw30
```

[88]:
```
(0.028438,
 0.0306,
 0.03126,
 0.03194,
 0.0325,
 0.033190000000000004,
 0.0346,
 0.03737)
```

(h) Re-run your bootstrapping procedure with this new curve. Comment on the changes to the forward rates.

[90]:
```
#
def swap_present_value(f1, sw1):
    #   PV
    PV = 0.5 * sw1 / (1 + 0.5 * f1) + 0.5 * sw1 / (1 + 0.5 * f1)**2
    #  PV
    return PV - 0.5 * f1 / (1 + 0.5 * f1) - 0.5 * f1 / (1 + 0.5 * f1)**2

#   swap_to_forward_1Y   args    sw1
def swap_to_forward_1Y(sw1):
    # fsolve  swap_present_value   sw1
    forward_rate = fsolve(swap_present_value, x0=sw1, args=(sw1,))
    return forward_rate[0]
```

[91]:
```
f1 = swap_to_forward_1Y(2.8438 / 100)
print(f1)
D1 = 1 / (1 + 0.5*f1)**2
print(D1)
```

```
0.028437999999999998
0.9721572416487895
```

[92]:
```
#
def swap_present_value(f2, sw2, f1):
    #
    PV1 = sw2 / (1 + 0.5 * f1) + sw2 / (1 + 0.5 * f1)**2 + sw2 / ((1 + 0.5 *␣
 ↪f1)**2 * (1 + 0.5 * f2)) + sw2 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2)
    #
    PV2 = f1 / (1 + 0.5 * f1) + f1 / (1 + 0.5 * f1)**2 + f2 / ((1 + 0.5 *␣
 ↪f1)**2 * (1 + 0.5 * f2)) + f2 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2)
    #
    return PV1 - PV2
```

```
#
def swap_to_forward_2Y(sw2, f1):
    #  fsolve
    forward_rate_2Y = fsolve(swap_present_value, x0=sw2, args=(sw2, f1))
    return forward_rate_2Y[0]

#
f2 = swap_to_forward_2Y(sw2,f1)
#
D2 = D1 / ((1 + 0.5 * f2)**2)

f2, D2
```

[92]: (0.032831130386268975, 0.9410092691367571)

[93]:
```
#
def swap_present_value_3Y(f3, sw3, f1, f2):
    #
    PV1 = sw3 / (1 + 0.5 * f1) + sw3 / (1 + 0.5 * f1)**2 + sw3 / ((1 + 0.5 *␣
 ↪f1)**2 * (1 + 0.5 * f2)) + sw3 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2) +␣
 ↪sw3 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2 * (1 + 0.5 * f3)) + sw3 / ((1 +␣
 ↪0.5 * f1)**2 * (1 + 0.5 * f2)**2 * (1 + 0.5 * f3)**2)
    #print(PV1)
    #
    PV2 = f1 / (1 + 0.5 * f1) + f1 / (1 + 0.5 * f1)**2 + f2 / ((1 + 0.5 *␣
 ↪f1)**2 * (1 + 0.5 * f2)) + f2 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2) + f3␣
 ↪/ ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2 * (1 + 0.5 * f3)) + f3 / ((1 + 0.5␣
 ↪* f1)**2 * (1 + 0.5 * f2)**2 * (1 + 0.5 * f3)**2)
    #print(PV2)
    #
    return PV1 - PV2

#
def swap_to_forward_3Y(sw3, f1, f2):
    #  fsolve
    forward_rate_3Y = fsolve(swap_present_value_3Y, x0=sw3, args=(sw3, f1, f2))
    return forward_rate_3Y[0]

#
f3 = swap_to_forward_3Y(sw3, f1, f2)

#
D3 = D2 / ((1 + 0.5 * f3)**2)

f3, D3
```

[93]: (0.03264530551203545, 0.9110258289278597)

[94]:
```
#    swap_present_value
def swap_present_value_4Y(f4, sw4, f1, f2, f3):
    PV = sw4/(1+0.5*f1) + sw4/(1+0.5*f1)**2 + sw4/((1+0.5*f1)**2*(1+0.5*f2)) +␣
 ↪sw4/((1+0.5*f1)**2*(1+0.5*f2)**2) + sw4/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.
 ↪5*f3)) + sw4/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2) + sw4/((1+0.
 ↪5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)) + sw4/((1+0.5*f1)**2*(1+0.
 ↪5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)**2)
    PV2 = f1/(1+0.5*f1) + f1/(1+0.5*f1)**2 + f2/((1+0.5*f1)**2*(1+0.5*f2)) + f2/
 ↪((1+0.5*f1)**2*(1+0.5*f2)**2) + f3/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3))␣
 ↪+ f3/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2) + f4/((1+0.5*f1)**2*(1+0.
 ↪5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)) + f4/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.
 ↪5*f3)**2*(1+0.5*f4)**2)
    return PV - PV2

#    swap_to_forward_4Y
def swap_to_forward_4Y(sw4, f1, f2, f3):
    #   lambda        swap_present_value
    forward_rate_4Y = fsolve(lambda f4: swap_present_value_4Y(f4, sw4, f1, f2,␣
 ↪f3), sw4)
    return forward_rate_4Y[0]

f4 = swap_to_forward_4Y(sw4, f1, f2, f3)

#
D4 = D3 / ((1 + 0.5 * f4)**2)

f4,D4
```

[94]:
```
(0.03411899988499989, 0.8807202227901273)
```

[95]:
```
#    swap_present_value
def swap_present_value_5Y(f5, sw5, f1, f2, f3, f4):
    PV = sw5*sum((1+0.5*f1)**-i for i in range(1, 3)) + sw5/((1+0.5*f1)**2*(1+0.
 ↪5*f2)) + sw5/((1+0.5*f1)**2*(1+0.5*f2)**2) + sw5/((1+0.5*f1)**2*(1+0.
 ↪5*f2)**2*(1+0.5*f3)) + sw5/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2) + sw5/
 ↪((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)) + sw5/((1+0.
 ↪5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)**2) + sw5/((1+0.
 ↪5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)**2*(1+0.5*f5)) + sw5/((1+0.
 ↪5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)**2*(1+0.5*f5)**2)
    PV2 = f1/(1+0.5*f1) + f1/(1+0.5*f1)**2 + f2/((1+0.5*f1)**2*(1+0.5*f2)) + f2/
 ↪((1+0.5*f1)**2*(1+0.5*f2)**2) + f3/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3))␣
 ↪+ f3/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2) + f4/((1+0.5*f1)**2*(1+0.
 ↪5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)) + f4/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.
 ↪5*f3)**2*(1+0.5*f4)**2) + f5/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.
 ↪5*f4)**2*(1+0.5*f5)) + f5/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.
 ↪5*f4)**2*(1+0.5*f5)**2)
    return PV - PV2
```

```
#    swap_to_forward_5Y
def swap_to_forward_5Y(sw5, f1, f2, f3, f4):
    #   lambda        swap_present_value
    forward_rate_5Y = fsolve(lambda f5: swap_present_value_5Y(f5, sw5, f1, f2,␣
 ↪f3, f4), sw5)
    return forward_rate_5Y[0]

#      f5
f5 = swap_to_forward_5Y(sw5, f1, f2, f3, f4)

#   D5
D5 = D4 / (1 + 0.5 * f5)**2

f5,D5
```

[95]: (0.034936952865818174, 0.8507384140726433)

[96]:
```
#    swap_present_value
def swap_present_value_7Y(f7, sw7, f1, f2, f3, f4, f5, D1, D2, D3, D4, D5):
    PV = (sw7 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
          sw7 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
          sw7 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
          sw7 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
          sw7 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
          sw7 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)))
    PV2 = (f1 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
           f2 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
           f3 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
           f4 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
           f5 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
           f7 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)))
    return PV - PV2

#    swap_to_forward_7Y
def swap_to_forward_7Y(sw7, f1, f2, f3, f4, f5, D1, D2, D3, D4, D5):
    #   lambda        swap_present_value
    forward_rate_7Y = fsolve(lambda f7: swap_present_value_7Y(f7, sw7, f1, f2,␣
 ↪f3, f4, f5, D1, D2, D3, D4, D5), sw7)
    return forward_rate_7Y[0]

#      f7
f7 = swap_to_forward_7Y(sw7, f1, f2, f3, f4, f5, D1, D2, D3, D4, D5)

#   D7
D7 = D5 / (1 + f7 * 0.5) ** 4
```

```
f7,D7
```

[96]: (0.035134972565236094, 0.7934931028923888)

[97]:
```
#    swap_present_value
def swap_present_value_10Y(f10, sw10, f1, f2, f3, f4, f5, f7, D1, D2, D3, D4,␣
 ↪D5, D7):
    PV = (sw10 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
          sw10 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
          sw10 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
          sw10 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
          sw10 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
          sw10 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)) +
          sw10 * D7 * sum((1 + 0.5 * f10) ** -i for i in range(1, 7)))
    PV2 = (f1 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
           f2 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
           f3 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
           f4 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
           f5 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
           f7 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)) +
           f10 * D7 * sum((1 + 0.5 * f10) ** -i for i in range(1, 7)))
    return PV - PV2

#    swap_to_forward_10Y
def swap_to_forward_10Y(sw10, f1, f2, f3, f4, f5, f7, D1, D2, D3, D4, D5, D7):
    #   lambda        swap_present_value
    forward_rate_10Y = fsolve(lambda f10: swap_present_value_10Y(f10, sw10, f1,␣
 ↪f2, f3, f4, f5, f7, D1, D2, D3, D4, D5, D7), sw10)
    return forward_rate_10Y[0]

#      f10
f10 = swap_to_forward_10Y(sw10, f1, f2, f3, f4, f5, f7, D1, D2, D3, D4, D5, D7)

#   D10
D10 = D7 / (1 + f10 * 0.5) ** 6
f10,D10
```

[97]: (0.03853787653877865, 0.7076368151287791)

[98]:
```
# Standalone swap_present_value function
def swap_present_value_30Y(f30, sw30, f1, f2, f3, f4, f5, f7, f10, D1, D2, D3,␣
 ↪D4, D5, D7, D10):
    PV = (sw30 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
          sw30 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
          sw30 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
          sw30 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
          sw30 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
```

```
          sw30 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)) +
          sw30 * D7 * sum((1 + 0.5 * f10) ** -i for i in range(1, 7)) +
          sw30 * D10 * sum((1 + 0.5 * f30) ** -i for i in range(1, 41)))
    PV2 = (f1 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
           f2 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
           f3 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
           f4 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
           f5 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
           f7 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)) +
           f10 * D7 * sum((1 + 0.5 * f10) ** -i for i in range(1, 7)) +
           f30 * D10 * sum((1 + 0.5 * f30) ** -i for i in range(1, 41)))
    return PV - PV2

# Refactored swap_to_forward_30Y function
def swap_to_forward_30Y(sw30, f1, f2, f3, f4, f5, f7, f10, D1, D2, D3, D4, D5,␣
 ↪D7, D10):
    forward_rate_30Y = fsolve(lambda f30: swap_present_value_30Y(f30, sw30, f1,␣
 ↪f2, f3, f4, f5, f7, f10, D1, D2, D3, D4, D5, D7, D10), sw30)
    return forward_rate_30Y[0]

# Using the refactored function to calculate f30
f30 = swap_to_forward_30Y(sw30, f1, f2, f3, f4, f5, f7, f10, D1, D2, D3, D4,␣
 ↪D5, D7, D10)

# Calculating D30
D30 = D10 / (1 + f30 * 0.5) ** 20
f30,D30
```

[98]:
```
(0.03978366720532815, 0.47723043447913865)
```

[99]:
```
f_list_new = [f1,f2,f3,f4,f5,f7,f10,f30]
f_list_new
```

[99]:
```
[0.028437999999999998,
 0.032831130386268975,
 0.03264530551203545,
 0.03411899988499989,
 0.034936952865818174,
 0.035134972565236094,
 0.03853787653877865,
 0.03978366720532815]
```

(i) Consider a bull steepener to the swap rates, that is perform the following shifts on each swap rate:

Term	Rate change (bps)
1Y	-50
2Y	-25
3Y	-15
4Y	-10
5Y	-5
7Y	+0
10Y	+0
30Y	+50

Print the new swap rates.

[100]:
```
swap_list=[0.028438,0.03060,0.03126,0.03144,0.03150,0.03169,0.03210,0.03237]
basis_list=[-0.005,-0.0025,-0.0015,-0.001,-0.0005,0,0,0]

sw1,sw2,sw3,sw4,sw5,sw7,sw10,sw30=[sw + basis for sw,basis in␣
 ↪zip(swap_list,basis_list)]
sw1,sw2,sw3,sw4,sw5,sw7,sw10,sw30
```

[100]:
```
(0.023438,
 0.0281,
 0.02976,
 0.030440000000000002,
 0.031,
 0.03169,
 0.0321,
 0.03237)
```

(j) Re-run your bootstrapping procedure with this new curve. Comment on the changes to the forward rates.

[105]:
```
#
def swap_present_value(f1, sw1):
    #   PV
    PV = 0.5 * sw1 / (1 + 0.5 * f1) + 0.5 * sw1 / (1 + 0.5 * f1)**2
    #  PV
    return PV - 0.5 * f1 / (1 + 0.5 * f1) - 0.5 * f1 / (1 + 0.5 * f1)**2

#   swap_to_forward_1Y   args    sw1
def swap_to_forward_1Y(sw1):
    # fsolve  swap_present_value   sw1
    forward_rate = fsolve(swap_present_value, x0=sw1, args=(sw1,))
    return forward_rate[0]

f1 = swap_to_forward_1Y(sw1)
print(f1)
D1 = 1 / (1 + 0.5*f1)**2
```

```
print(D1)
```

```
0.023438
0.9769676601655329
```

[106]:
```
#
def swap_present_value(f2, sw2, f1):
    #
    PV1 = sw2 / (1 + 0.5 * f1) + sw2 / (1 + 0.5 * f1)**2 + sw2 / ((1 + 0.5 *␣
 ↪f1)**2 * (1 + 0.5 * f2)) + sw2 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2)
    #
    PV2 = f1 / (1 + 0.5 * f1) + f1 / (1 + 0.5 * f1)**2 + f2 / ((1 + 0.5 *␣
 ↪f1)**2 * (1 + 0.5 * f2)) + f2 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2)
    #
    return PV1 - PV2

#
def swap_to_forward_2Y(sw2, f1):
    #  fsolve
    forward_rate_2Y = fsolve(swap_present_value, x0=sw2, args=(sw2, f1))
    return forward_rate_2Y[0]

#
f2 = swap_to_forward_2Y(sw2,f1)
#
D2 = D1 / ((1 + 0.5 * f2)**2)

f2, D2
```

[106]:
```
(0.03290535948429921, 0.9455965031506183)
```

[107]:
```
#
def swap_present_value_3Y(f3, sw3, f1, f2):
    #
    PV1 = sw3 / (1 + 0.5 * f1) + sw3 / (1 + 0.5 * f1)**2 + sw3 / ((1 + 0.5 *␣
 ↪f1)**2 * (1 + 0.5 * f2)) + sw3 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2) +␣
 ↪sw3 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2 * (1 + 0.5 * f3)) + sw3 / ((1 +␣
 ↪0.5 * f1)**2 * (1 + 0.5 * f2)**2 * (1 + 0.5 * f3)**2)
    #print(PV1)
    #
    PV2 = f1 / (1 + 0.5 * f1) + f1 / (1 + 0.5 * f1)**2 + f2 / ((1 + 0.5 *␣
 ↪f1)**2 * (1 + 0.5 * f2)) + f2 / ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2) + f3␣
 ↪/ ((1 + 0.5 * f1)**2 * (1 + 0.5 * f2)**2 * (1 + 0.5 * f3)) + f3 / ((1 + 0.5␣
 ↪* f1)**2 * (1 + 0.5 * f2)**2 * (1 + 0.5 * f3)**2)
    #print(PV2)
    #
    return PV1 - PV2
```

```
#
def swap_to_forward_3Y(sw3, f1, f2):
    #  fsolve
    forward_rate_3Y = fsolve(swap_present_value_3Y, x0=sw3, args=(sw3, f1, f2))
    return forward_rate_3Y[0]

#
f3 = swap_to_forward_3Y(sw3, f1, f2)

#
D3 = D2 / ((1 + 0.5 * f3)**2)

f3, D3
```

[107]: (0.033243751341449745, 0.9149280790879107)

[108]:
```
#    swap_present_value
def swap_present_value_4Y(f4, sw4, f1, f2, f3):
    PV = sw4/(1+0.5*f1) + sw4/(1+0.5*f1)**2 + sw4/((1+0.5*f1)**2*(1+0.5*f2)) +␣
 ↪sw4/((1+0.5*f1)**2*(1+0.5*f2)**2) + sw4/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.
 ↪5*f3)) + sw4/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2) + sw4/((1+0.
 ↪5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)) + sw4/((1+0.5*f1)**2*(1+0.
 ↪5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)**2)
    PV2 = f1/(1+0.5*f1) + f1/(1+0.5*f1)**2 + f2/((1+0.5*f1)**2*(1+0.5*f2)) + f2/
 ↪((1+0.5*f1)**2*(1+0.5*f2)**2) + f3/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3))␣
 ↪+ f3/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2) + f4/((1+0.5*f1)**2*(1+0.
 ↪5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)) + f4/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.
 ↪5*f3)**2*(1+0.5*f4)**2)
    return PV - PV2

#    swap_to_forward_4Y
def swap_to_forward_4Y(sw4, f1, f2, f3):
    #   lambda        swap_present_value
    forward_rate_4Y = fsolve(lambda f4: swap_present_value_4Y(f4, sw4, f1, f2,␣
 ↪f3), sw4)
    return forward_rate_4Y[0]

f4 = swap_to_forward_4Y(sw4, f1, f2, f3)

#
D4 = D3 / ((1 + 0.5 * f4)**2)

f4,D4
```

[108]: (0.03261670364494173, 0.8858005940260291)

[109]:
```
#    swap_present_value
def swap_present_value_5Y(f5, sw5, f1, f2, f3, f4):
    PV = sw5*sum((1+0.5*f1)**-i for i in range(1, 3)) + sw5/((1+0.5*f1)**2*(1+0.
 ↪5*f2)) + sw5/((1+0.5*f1)**2*(1+0.5*f2)**2) + sw5/((1+0.5*f1)**2*(1+0.
 ↪5*f2)**2*(1+0.5*f3)) + sw5/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2) + sw5/
 ↪((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)) + sw5/((1+0.
 ↪5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)**2) + sw5/((1+0.
 ↪5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)**2*(1+0.5*f5)) + sw5/((1+0.
 ↪5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)**2*(1+0.5*f5)**2)
    PV2 = f1/(1+0.5*f1) + f1/(1+0.5*f1)**2 + f2/((1+0.5*f1)**2*(1+0.5*f2)) + f2/
 ↪((1+0.5*f1)**2*(1+0.5*f2)**2) + f3/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3))␣
 ↪+ f3/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2) + f4/((1+0.5*f1)**2*(1+0.
 ↪5*f2)**2*(1+0.5*f3)**2*(1+0.5*f4)) + f4/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.
 ↪5*f3)**2*(1+0.5*f4)**2) + f5/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.
 ↪5*f4)**2*(1+0.5*f5)) + f5/((1+0.5*f1)**2*(1+0.5*f2)**2*(1+0.5*f3)**2*(1+0.
 ↪5*f4)**2*(1+0.5*f5)**2)
    return PV - PV2

#    swap_to_forward_5Y
def swap_to_forward_5Y(sw5, f1, f2, f3, f4):
    #   lambda        swap_present_value
    forward_rate_5Y = fsolve(lambda f5: swap_present_value_5Y(f5, sw5, f1, f2,␣
 ↪f3, f4), sw5)
    return forward_rate_5Y[0]

#      f5
f5 = swap_to_forward_5Y(sw5, f1, f2, f3, f4)

#   D5
D5 = D4 / (1 + 0.5 * f5)**2

f5,D5
```

[109]:
```
(0.03343139562806712, 0.856913350718001)
```

[110]:
```
#    swap_present_value
def swap_present_value_7Y(f7, sw7, f1, f2, f3, f4, f5, D1, D2, D3, D4, D5):
    PV = (sw7 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
          sw7 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
          sw7 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
          sw7 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
          sw7 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
          sw7 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)))
    PV2 = (f1 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
           f2 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
           f3 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
           f4 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
```

```
            f5 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
            f7 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)))
    return PV - PV2

#    swap_to_forward_7Y
def swap_to_forward_7Y(sw7, f1, f2, f3, f4, f5, D1, D2, D3, D4, D5):
    #   lambda        swap_present_value
    forward_rate_7Y = fsolve(lambda f7: swap_present_value_7Y(f7, sw7, f1, f2,␣
 ↪f3, f4, f5, D1, D2, D3, D4, D5), sw7)
    return forward_rate_7Y[0]

#      f7
f7 = swap_to_forward_7Y(sw7, f1, f2, f3, f4, f5, D1, D2, D3, D4, D5)

#   D7
D7 = D5 / (1 + f7 * 0.5) ** 4

f7,D7
```

[110]:
```
(0.03362707951203227, 0.8016256901458785)
```

[111]:
```
#    swap_present_value
def swap_present_value_10Y(f10, sw10, f1, f2, f3, f4, f5, f7, D1, D2, D3, D4,␣
 ↪D5, D7):
    PV = (sw10 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
          sw10 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
          sw10 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
          sw10 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
          sw10 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
          sw10 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)) +
          sw10 * D7 * sum((1 + 0.5 * f10) ** -i for i in range(1, 7)))
    PV2 = (f1 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
           f2 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
           f3 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
           f4 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
           f5 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
           f7 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)) +
           f10 * D7 * sum((1 + 0.5 * f10) ** -i for i in range(1, 7)))
    return PV - PV2

#    swap_to_forward_10Y
def swap_to_forward_10Y(sw10, f1, f2, f3, f4, f5, f7, D1, D2, D3, D4, D5, D7):
    #   lambda        swap_present_value
    forward_rate_10Y = fsolve(lambda f10: swap_present_value_10Y(f10, sw10, f1,␣
 ↪f2, f3, f4, f5, f7, D1, D2, D3, D4, D5, D7), sw10)
    return forward_rate_10Y[0]
```

```
#      f10
f10 = swap_to_forward_10Y(sw10, f1, f2, f3, f4, f5, f7, D1, D2, D3, D4, D5, D7)

#   D10
D10 = D7 / (1 + f10 * 0.5) ** 6
f10,D10
```

[111]: (0.03323013440399479, 0.7261600868744981)

[112]:
```
# Standalone swap_present_value function
def swap_present_value_30Y(f30, sw30, f1, f2, f3, f4, f5, f7, f10, D1, D2, D3,␣
 ↪D4, D5, D7, D10):
    PV = (sw30 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
          sw30 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
          sw30 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
          sw30 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
          sw30 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
          sw30 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)) +
          sw30 * D7 * sum((1 + 0.5 * f10) ** -i for i in range(1, 7)) +
          sw30 * D10 * sum((1 + 0.5 * f30) ** -i for i in range(1, 41)))
    PV2 = (f1 * sum((1 + 0.5 * f1) ** -i for i in range(1, 3)) +
           f2 * D1 * sum((1 + 0.5 * f2) ** -i for i in range(1, 3)) +
           f3 * D2 * sum((1 + 0.5 * f3) ** -i for i in range(1, 3)) +
           f4 * D3 * sum((1 + 0.5 * f4) ** -i for i in range(1, 3)) +
           f5 * D4 * sum((1 + 0.5 * f5) ** -i for i in range(1, 3)) +
           f7 * D5 * sum((1 + 0.5 * f7) ** -i for i in range(1, 5)) +
           f10 * D7 * sum((1 + 0.5 * f10) ** -i for i in range(1, 7)) +
           f30 * D10 * sum((1 + 0.5 * f30) ** -i for i in range(1, 41)))
    return PV - PV2

# Refactored swap_to_forward_30Y function
def swap_to_forward_30Y(sw30, f1, f2, f3, f4, f5, f7, f10, D1, D2, D3, D4, D5,␣
 ↪D7, D10):
    forward_rate_30Y = fsolve(lambda f30: swap_present_value_30Y(f30, sw30, f1,␣
 ↪f2, f3, f4, f5, f7, f10, D1, D2, D3, D4, D5, D7, D10), sw30)
    return forward_rate_30Y[0]

# Using the refactored function to calculate f30
f30 = swap_to_forward_30Y(sw30, f1, f2, f3, f4, f5, f7, f10, D1, D2, D3, D4,␣
 ↪D5, D7, D10)

# Calculating D30
D30 = D10 / (1 + f30 * 0.5) ** 20
f30,D30
```

[112]: (0.03258709705552466, 0.5255923083478515)

[113]:
```
f_list_new = [f1,f2,f3,f4,f5,f7,f10,f30]
f_list_new
```

[113]:
```
[0.023438,
 0.03290535948429921,
 0.033243751341449745,
 0.03261670364494173,
 0.03343139562806712,
 0.03362707951203227,
 0.03323013440399479,
 0.03258709705552466]
```

[ ]:
```

```