Fixed Income Fundamentals (with Python)

Bond Yield Calculation with Semiannual Coupons and Cash Price

Alexandre Landi

IBM, Skema Business School

September 15, 2024

Problem Definition

Exercise:
A 3-year bond provides a coupon of 8% semiannually and has a cash price of 104. The bond pays \$4 in 6, 12, 18, 24, and 30 months, and \$104 in 36 months. What is the bond's yield?

Source: Hull, J. C. (2021), Options, Futures, and Other Derivatives (11th global ed.), Pearson, p. 122, practice question 4.11

This exercise requires calculating the bond's yield to maturity (YTM) by solving the equation for y, the yield, which equates the bond's price with the present value of future cash flows.

Bond Payments Overview

Coupon Payments: The bond pays \$4 semiannually over 3 years:

$$[\textit{Coupon} \quad \textit{Coupon} \quad \textit{Coupon} \quad \textit{Coupon} \quad \textit{Coupon} \quad (\textit{Coupon} + \textit{FaceValue})]$$

$$[\$4 \quad \$4 \quad \$4 \quad \$4 \quad \$4 \quad \$104]$$

These payments are made at:

- 6 months
- 12 months
- 18 months
- 24 months
- 30 months
- 36 months (final payment includes the face value)

The bond provides five coupon payments of \$4 and a final payment of \$104, which includes the last coupon and the face value.

Introduction to Discounting with Continuous Compounding

What is Discounting?

- **Discounting** is the process of determining the present value of a future amount of money or stream of cash flows given a specific interest rate.
- In continuous compounding, the discount factor for a future payment at time t is:

$$e^{-rt}$$

where:

  - e is the base of the natural logarithm.
  - r is the continuous compounding rate (or yield).
  - t is the time in years until the payment is made.

Discounting reflects the time value of money, capturing how the value of a future payment decreases with time.

Understanding the Discount Factor e^{-ty}

Why Use e^{-ty}?

- The expression e^{-ty} represents the **discount factor** under continuous compounding.
- It calculates the present value of a future payment by accounting for the continuous decrease in value over time.
- For a payment at time t, the factor e^{-ty} discounts the payment back to the present using yield y.

How It Works:

$$\text{Present Value} = \text{Future Value} \times e^{-ty}$$

The use of e^{-ty} aligns with the principle of continuous compounding, reflecting the continuous nature of interest accrual.

Applying e^{-ty} in Bond Yield Calculation

Why Use e^{-ty} in Our Case?

To calculate the bond's yield, we need to find the yield y such that the present value of all future payments equals the bond's current market price.

- Each cash flow (coupon payment or face value) is discounted to its present value using e^{-ty}, where:
  - t is the time in years until the payment.
  - y is the yield to maturity we are solving for.
- The sum of these present values is set equal to the bond's market price:

$$4e^{-0.5y} + 4e^{-1.0y} + \ldots + 104e^{-3.0y} = 104$$

Applying e^{-ty} ensures we accurately calculate the present value of future payments, accounting for the yield y and time t.

Bond Yield Formula

The bond yield (y) is the value that solves:

$$4e^{-0.5y} + 4e^{-1.0y} + 4e^{-1.5y} + 4e^{-2.0y} + 4e^{-2.5y} + 104e^{-3.0y} = 104$$

Where:

- Each term represents the present value of the bond's payments.
- e^{-ty} discounts the payment at time t with the yield y.
- The total present value must equal the bond's current price (104).

This equation equates the present value of all coupon and face value payments with the bond's current price of \$104, discounted by the bond yield.

Why Use a Numerical Method?

Why Can't We Solve Algebraically?

The bond yield equation:

$$4e^{-0.5y} + 4e^{-1.0y} + 4e^{-1.5y} + 4e^{-2.0y} + 4e^{-2.5y} + 104e^{-3.0y} = 104$$

is **nonlinear** and involves **exponential functions** of y.

Key Points:

- The equation is **transcendental**, meaning it contains terms like e^{-ty} which cannot be easily isolated or simplified algebraically.
- There is no simple closed-form solution for y; thus, solving directly with algebra is impractical.
- Numerical methods (like SciPy's `fsolve`) efficiently find roots of such complex functions.

Therefore, a numerical solver is used to find the precise yield (y) that equates the present value of the bond's cash flows to its price.

Introduction to fsolve

What is `fsolve`?

`fsolve` is a function from the SciPy library in Python used to solve non-linear equations.

Key Features:

- Finds the roots of a function (i.e., values where the function equals zero).
- Uses numerical methods like the Newton-Raphson method.
- Requires an initial guess to start the iteration.

`fsolve` is particularly useful for solving complex equations that cannot be rearranged into a simple algebraic form.

Introduction to the Yield Approximation Formula

Objective: Derive an approximate formula for the yield to maturity (YTM) of a bond.

Approximation Formula:

$$y \approx \frac{\text{Coupon Payment} + \frac{\text{Face Value} - \text{Cash Price}}{\text{Number of Periods}}}{\text{Cash Price}}$$

Where:

- Coupon Payment: Periodic payment by the bond.
- Face Value: The bond's value at maturity.
- Cash Price: Current market price of the bond.
- Number of Periods: Total periods until maturity.

This formula provides a quick estimate of a bond's yield, helping to understand its returns without complex calculations.

Derive the Approximate Yield Formula

Objective: Derive an approximate formula for the yield to maturity (YTM) of a bond using the average annual return method.

Key Concepts:

- The approximate yield is based on the bond's average annual return.
- It considers both coupon income and capital gain or loss.
- The yield is calculated as the total average annual return divided by the cash price.

This approach provides an intuitive and practical method for estimating the yield to maturity, especially useful for bonds with regular coupon payments and near par value.

Step 1: Consider the Average Annual Return

Components of Average Annual Return:

1. **Annual Coupon Income:** The total coupon payments received each year.
2. **Average Annual Capital Gain or Loss:** The difference between the bond's face value and its purchase price, spread over the years to maturity.

Formula for Average Annual Return:

$$\text{Average Annual Return} = \text{Annual Coupon Payment} + \frac{\text{Face Value} - \text{Price}}{\text{Years to Maturity}}$$

This formula captures the total expected return per year from holding the bond until maturity.

Step 2: Approximate the Yield

Calculate the Approximate Yield to Maturity (YTM):

Formula:

$$\text{Approximate YTM} = \frac{\text{Average Annual Return}}{\text{Cash Price}}$$

Substituting the previous formulas:

$$\text{Approximate YTM} = \frac{\text{Annual Coupon Payment} + \dfrac{\text{Face Value} - \text{Price}}{\text{Years to Maturity}}}{\text{Cash Price}}$$

This formula provides an estimate of the bond's yield based on its expected annual return and average investment.

Step 4: Applying the Approximate Yield Formula

Given Bond Parameters:

- **Face Value (F)**: \$100
- **Price (P)**: \$104
- **Annual Coupon Payment (C_a)**: \$8 (8% of face value)
- **Years to Maturity (T)**: 3 years

Calculate Average Annual Return:

$$\text{Average Annual Return} = \$8 + \frac{\$100 - \$104}{3} = \$8 - \$1.33 = \$6.67$$

Using these values, we can estimate the bond's yield to maturity.

Step 5: Calculating the Approximate Yield

Compute the Approximate YTM:

$$\text{YTM} \approx \frac{\text{Average Annual Return}}{\text{Cash Price}} \approx \frac{\$6.67}{\$104} \approx 0.06410 \text{ or } 6.410\%$$

Interpretation:

- The approximate yield to maturity is 6.410%.
- This is an estimate of the annual return expected from holding the bond until maturity.

This approximation is useful for quickly estimating the bond's yield without complex calculations.

Why Use an Estimated Yield?

Why is the Estimation Only an Approximation?

The algebraic method provides an estimated yield but is not totally accurate because:

- The formula simplifies the bond price calculation by linearizing the exponential discounting.
- It does not account for the precise timing and compounding effects of cash flows.
- Assumes a constant yield, which may not be the case in reality.

Why is it Good Enough?

- Provides a reasonable starting point for numerical solvers.
- Reduces the number of iterations required for finding the exact yield.

The estimated yield serves as a "guess" to feed into the numerical solver, allowing it to converge more quickly to the precise solution.

Python Function Definition (Yield Estimation)

Function to Estimate the Yield:

```
def estimate_yield(
    coupon, face_value, cash_price, periods, payment_freq
    ):
    t = periods / payment_freq
    annual_coupon = coupon * payment_freq
    face_cash_diff = (face_value - cash_price) / t
    annual_return = annual_coupon + face_cash_diff
    estimated_yield = annual_return / cash_price
    return estimated_yield
```

This function calculates an estimated yield using an algebraic formula. The estimation serves as a good initial guess for the numerical solver ('fsolve'), helping it converge faster to the exact bond yield.

Python Example Usage (Yield Estimation)

Using the Estimated Yield Function:

```
# Define variables
coupon = 4
face_value = 100
cash_price = 104
periods = 6
payment_freq = 2

# Call the function to estimate the initial yield
estimated_yield = estimate_yield(coupon, face_value,
    cash_price, periods, payment_freq)
```

The example demonstrates how to use the ‘estimate_yield‘ function to compute an initial guess for the yield, which will be used as input for a numerical solver.

Understanding fsolve

What is `fsolve`?

`fsolve` is a function in the SciPy library used to solve non-linear equations.

Key Features:

- Finds the roots of a function, i.e., values where the function output is zero.
- Uses numerical methods (like the Newton-Raphson method) to iteratively find solutions.
- Requires an initial guess to start the iteration process.

Syntax:

```
fsolve(func, x0, args=(), ...)
```

- `func`: The function to solve.
- `x0`: Initial guess for the solution.
- `args`: Extra arguments to pass to the function.

'fsolve' is a powerful tool for solving complex, non-linear equations that cannot be rearranged into a simple algebraic form.

Applying fsolve in Our Case

How We Used `fsolve`:

To find the bond yield (YTM), we applied the `fsolve` function as follows:

- `bond_price`: The function to solve, defined to calculate the difference between the present value of future cash flows and the bond's current market price.
- `estimated_yield`: The initial guess for the yield, calculated using an algebraic approximation.
- `args`: A tuple containing extra arguments to pass to the `bond_price` function:
  - `cash_flows`: Array of coupon payments and final face value.
  - `times`: Array of times in years for each payment.
  - `cash_price`: The current market price of the bond.

In our case, 'fsolve' iteratively adjusts the yield ('y') to find the point where the present value of the bond's cash flows equals the bond's market price (104).

Python Function Definition

Function to Compute Bond Price:

```
import numpy as np

def bond_price(y, cash_flows, times, cash_price):
    present_values = cash_flows * np.exp(-y * times)
    return np.sum(present_values) - cash_price
```

This function calculates the bond price for a given yield y, cash flows, and times, based on the current market price (cash price).

The cash price is essential for solving the equation to find the yield.

Python Example Usage (Solving for Yield)

Solving for the Yield:

```
from scipy.optimize import fsolve

t = np.arange(0, periods / payment_freq, 1 / payment_freq) +
    1 / payment_freq

# Use fsolve with the estimated yield
ytm = fsolve(
    bond_price,
    estimated_yield,
    args=(cash_flows, times, cash_price)
    )[0]
```

Output:

```
Bond Yield (YTM): 6.407%
```

The ‘fsolve‘ function is initialized with the estimated yield, allowing it to converge quickly to the precise bond yield (YTM) of 6.407

Introducing QuantLandi's Newsletter

- Practical insights into quantitative finance
- Expert advice from industry professionals
- Actionable strategies to enhance your financial expertise

Subscribe to QuantLandi's Newsletter:
https://quantlandi.substack.com/