Fixed Income Derivatives E2025 - Problem Set Week 1

Problem 1

In this problem, we will initially consider a fixed rate bullet bond with an annual coupon rate of 6 and a principal of 100 USD. This bond has just been issued and matures in exactly 10 years. The market price of this bond is currently 98.74 USD.

In this problem, we will construct a function that will allow us find the yield-to-maturity of a known stream of future cash-flows when the true present value of these cash-flows is known. To do so, we will use the package 'scipy' in Python and minimize the squared error between the true present value Π of the future cash-flows and the present value $\hat{\Pi}(y|\mathbf{C})$ of the future cash-flows $\mathbf{C}$ as a function of the yield-to-maturity y. That is, we will consider the following minimization problem

$$\min_y \text{SE} = \Big(\Pi - \hat{\Pi}(y|\mathbf{C})\Big)^2 \tag{1}$$

a) Write a function in Python which takes as inputs the yield-to-maturity, a vector of the times of the cash-flows, and a vector of cash-flows. This function should return the present value of the future cash-flows.

b) Write a function in Python which takes as inputs the yield-to-maturity, the true present value of the future cash-flows, a vector of the times of the cash-flows, and a vector of cash-flows. This function should return the squared deviation between the true present value and the present value for that specific value of the yield-to-maturity.

c) Compute the yield-to-maturity of the bond both using both the method 'nelder-mead' and the method 'powell'.

The methods 'nelder-mead' and 'powell' do not make use of the first- and second order derivatives of the objective function with respect to the variable, y, that we are optimizing over, but the optimization procedure can be improved by using the information in these derivatives. In this part of the problem, we will learn to use methods, namely 'BFGS' and 'Newton-CG' that converge faster but rely on the first- and second order derivatives of the objective function.

d) Compute $\frac{\partial SE}{\partial y}$ and $\frac{\partial^2 SE}{\partial y^2}$

e) Write a function in Python that returns the first order derivative of the objective function and find the yield-to-maturity using the 'BFGS' method and the first order derivative.

f) Write a function in Python that returns the second order derivative of the objective function and find the yield-to-maturity using the 'Newton-CG' method and both the first- and second order derivative.

Lastly, we will use the 'trust-constr' method to impose bounds on the variable we are optimizing over

g) Use the 'trust-constr' method to find the yield-to-maturity when imposing that $y \in [0, 0.12]$ and $y \in [0.08, 0.12]$.