American University of Armenia
BS in Data Science

Pricing of Fixed Income Instruments: Creation of Bond Valuation Calculator

Students:
Knarik Badalyan
Sona Stepanyan

Supervisor:
Arsen Grigoryan

May 4, 2023

Abstract

Bond valuation is an essential component in the world of finance, given the fact that bonds are a well-liked investment choice for investors globally. They are a kind of debt instrument provided by businesses or the government that pays the shareholder the coupon periodically and the principal amount at maturity. Bond valuation is the process of figuring out a bond's fair value or market value, which takes into account several variables like the coupon rate, face value, settlement, maturity dates, and yield to maturity. The main goal of the project has been to create a reliable tool for specialists in the field. Hence, the project targeted the creation of a simple calculator for pricing a single bond. The precision of calculation was tested by comparing calculated prices with the ones published by the Central Bank of Armenia. Furthermore, the tool was enhanced to provide a fast and simple API - like tool for the speedy pricing of large sets of bonds (around 1 million bonds at a time). The research conducted in the local financial market showed that there is a real need for a product that can help investors to figure out the price or other parameters of the bond. For this reason, a Python-based calculator with a graphical user interface (GUI) was created that accepts various bond-related parameters like coupon rate, coupon frequency, face value, time to maturity, yield to maturity, settlement date, and maturity date. As an output, it returns dirty price, clean price, accrued interest, face value, coupon rate, coupon frequency, yield to maturity, time to maturity, Macaulay duration, modified duration, convexity, settlement date, maturity date, and convention. In the second part of the project, a real bond dataset was obtained by scraping the Central Bank of Armenia (CBA) website. The dataset includes 28 rows of bond data, which were used to test the accuracy of the bond valuation calculator. The results were analyzed and compared to the actual bond prices to evaluate the accuracy of the calculator. Other research in the local financial market revealed that there is a need for a reliable and efficient tool for valuing large bond portfolios. As a result, it was concluded that it is important to make the bond valuation process as fast as possible. For this reason, a million-row dataset was generated in Excel, and the code was run on it to find out how much time it took to obtain the results. The prompts showed that it takes quite a long time to evaluate huge amounts of bonds, and the need for optimization arose. Hence, the NumPy package was used to optimize the process, and with the help of the optimization, the code could be run on a million-row dataset in 276 seconds (4.6 minutes). This was a success, as before the optimization, the code was running on the dataset for more than 40 minutes. Overall, this project demonstrates the importance of bond valuation and how Python programming can be used to develop a tool that can be very useful for the financial market. The results also indicate that the calculator is accurate and can be used to analyze large datasets of bonds.

Introducation

The bond market is an essential part of the world of finance, and it offers financial resources through the sales of the bonds. The value of the bonds depends on different components, such as the state of the interests, the reliability of the issuer, and the time until its maturity. So, for the people who are in this and want to make their investments wisely, it is critical to determine the fair value of the bond. Regardless of the essentiality of band valuation, the local Armenian market is in demand of an accessible and easy-to-use tool to calculate the value of bonds and do a quick valuation of large bond portfolios. Consequently, poor investment decisions can be made by the investors, which can, in their turn, lead to big financial losses. To tackle this issue, it was proposed to create a bond valuation calculator that can help investors of the local financial market to evaluate the bonds with high accuracy and speed. The best models of pricing will be integrated for the creation of the bond calculator to give accessible and user-friendly results for the local financial market participants having different kinds of expertise. By offering the market participants this tool, the risk of financial losses will be reduced, contributing to the progress of the local financial market of Armenia. Overall, this paper will emphasize the importance of creating a bond valuation calculator for the Armenian market and the ways used to achieve the goal. In the methodology section, the techniques and the design of the calculator, the methods of testing the results, and optimization will be described. It is believed that the space bond valuation calculator in the Armenian market was empty and that it will be a way to facilitate the growth and development of the bond market in Armenia.

Methodology

This bond valuation project consists of three parts: the creation of the calculator, accuracy checking, and optimization. In the first part of the project, a bond valuation calculator was created. The calculator includes a Python script and a simple graphical user interface (GUI) to make it accessible, user-friendly, and easy to use. The calculator takes a principal value, coupon rate, coupon frequency, settlement date, maturity date, yield to maturity, and convention as input and returns dirty price, clean price, accrued interest, face value, coupon rate, coupon frequency, yield to maturity, time to maturity, Macaulay duration, modified duration, convexity, settlement date, maturity date, and convention. All the code is written in Python, and several Python packages are used, including NumPy, Pandas, datetime, dateutil.relativedelta, and SciPy. NumPy is used for numerical calculations. Pandas package is used for manipulating and analyzing tabular data. The datetime package is used to work with dates and dateutil.relativedelta is used to compute the time between the dates. Lastly, SciPy is used for scientific and technical computations, specifically for the optimization of bond valuation. For the development of the GUI PyQT package is used.

Bond Pricing: bond basics

To achieve the goal of valuing bonds and other bond instruments many financial formulas were used. In this section of the methodology, a detailed explanation of the formulas used to create the calculator is provided. It is vital to mention that the calculator is multifunctional and evaluates not only the prices of fixed-rate bonds but also the prices of zero-coupon bonds. Other than this, it accepts all 4 possible conventions: ACT/ACT, 30/360, ACT/360, and ACT/365.

Price

The price of a bond can be computed as the present value of its future cash flows. Those cash flows include the coupons (if any) and the principal value (or face value) of the bond that is repaid at maturity. Mathematically, the bond price at time t is given by

$$B_t = Pe^{-y(T-t)} + \sum_{i=1}^{N} C_i e^{-y(t_i-t)} \tag{1}$$

where B_t is the bond price, P is the principal value, C_i is the coupon paid at time t_i, T is the maturity time and y is the yield to maturity. Here, we assume that the last coupon is paid at maturity(i.e., $t_N = T$).

Note that y is continuously compounded. Suppose we have a yield of $y_a an$ with annual compounding, the corresponding continuously compounded yield is given by

$$y = \log(1 + y_{ann}). \tag{2}$$

This above price is the dirty price of a bond. Between coupon dates, the price quoted by bond dealers, referred to as the clean price, is different from the dirty price. This is because the dirty price includes accrued interest while the clean price does not. The clean price is therefore the dirty price minus accrued interest. Accrued interest is given by

$$I_{acc} = C_i \frac{t - t_{i-1}}{t_i - t_{i-1}} \tag{3}$$

where the numerator is the number of days since the last coupon and the denominator is the number of days from the last coupon date to the next coupon date (i.e., t_i) is the time when the next coupon is to be paid and $t_i - 1$ is the time when the last coupon was paid). Note that there are different conventions to calculate the number of days (e.g., Actual/Actual, 30/360).

Yield to Maturity

When the bond is traded, its price is readily available. Given the price, we can compute the yield to maturity. This is the value of y that makes the price equation above equal to the market price of the bond. Unless the bond is a zero-coupon bond, y must be found with an optimization algorithm (e.g., Newton). The price of a zero-coupon bond is simply

$$Z_t = Pe^{-y(T-t)}. \tag{4}$$

In this case, we can easily derive a closed-form solution for the yield to maturity,

$Z_t = Pe^{-y(T-t)}$
$\log Z_t = \log\left(Pe^{-y(T-t)}\right)$
$\log Z_t = -y(T-t) + \log P$
$\log \frac{Z_t}{P} = -y(T-t)$
$y = -\frac{\log \frac{Z_t}{P}}{T-t}.$

Current yield

The current yield is a simple yield measure. It is defined as

$$y_{curr} = \frac{C}{B_t}. \tag{5}$$

Duration

Let's differentiate the equation for B_t with respect to y. We get

$$\frac{\partial B_t}{\partial y} = -(T-t)Pe^{-y(T-t)} - \sum_{i=1}^{N}(t_i - t)C_i e^{-y(t_i-t)}. \tag{6}$$

The Macaulay duration is given by

$$D_{mac} = -\frac{1}{B_t}\frac{\partial B_t}{\partial y}. \tag{7}$$

The modified duration is given by

$$D_{mod} = \frac{D_{mac}}{1 + \frac{y_{ann}}{f}} \tag{8}$$

where f is the number of coupons paid in a year.

Convexity

The second order derivative of the bond price with respect to y is given by

$$\frac{\partial^2 B_t}{\partial y^2} = (T-t)^2 P e^{-y(T-t)} + \sum_{i=1}^{N} (t_i - t)^2 C_i e^{-y(t_i - t)}. \tag{9}$$

The convexity of a bond is given by

$$\mathcal{C} = \frac{1}{B_t} \frac{\partial^2 B_t}{\partial y^2}. \tag{10}$$

Approximating change in price for change in yield

The change in bond price for a given change in yield can be approximated with the modified duration and convexity. Modified duration alone can be used for a small change in yield but for larger changes in yield, it is necessary to use a second order approximation. The percentage change in bond price is given by

$$\frac{\Delta B_t}{B_t} \approx -D_{mod} \Delta y + \frac{1}{2} \mathcal{C} (\Delta y)^2. \tag{11}$$

The second part of the project was the process of accuracy checking which included data scraping, data cleaning and manipulation, and testing. For scraping the necessary data urllib.request, Pandas, and NumPy packages were used. The urllib request package was used to access the Central Bank of Armenia website link(`https://www.cba.am/IMRM/YC/YC.xlsx`), which includes the necessary data. The scraped dataset included the following columns: ISIN, Coupon Rate , Maturity Date, Dirty Price, Yield to Maturity, and Settlement Date. Then, several rows of data that did not meet the requirements of the bond calculator were dropped, for example, the data was in foreign currency or there were other impending circumstances. Later columns for Coupon Frequency, Principal Value, and Convention were added manually to match the dataset to the input requirement of the code. The resulting dataset was a 28-row dataset, which included all the necessary variables to calculate the bond price and other bond parameters. The dataset was saved as a CSV file. To check the accuracy of the bond calculator the bond valuation code was applied to the dataset. The code took scraped bond parameters and calculated the dirty price for each bond, which was later compared to the scraped dirty prices. The calculated dirty prices were identical to the scraped dirty prices. Therefore, a conclusion can be drawn that the bond calculator has hundred percent accuracy. The last part of the project includes running the dataset on a million-row dataset. For this part of the project, a million-row dataset was generated in Excel, which includes all the necessary attributes that the bond valuation calculator takes as input. These values include principal value, coupon rate, coupon frequency, settlement date, maturity date, yield to maturity, and convention. Principal values were all set to 100, coupon frequency was set to either 1, 2, 4, or 12, and the convention was set to either ACT/ACT, ACT/360, ACT/365, or 30/360. Other parameters of the dataset were generated randomly. The dataset was then stored in Google Drive to ensure its accessibility(`https://docs.google.com/spreadsheets/d/1Xg4XZdcN1ZfFuqs9puTYxf-M2cNITBo9/edit?usp=sharing&ouid=102726307159039228958&rtpof=true&sd=true`). Then, the code was run on the dataset, but it took a very long time to work. Therefore, the need for code optimization arose. The NumPy package was used to optimize the bond valuation code. The new optimized version of the code takes the dataset and makes all the necessary changes: it uses the string split function to bring the dates to the necessary format and calculates all percentage values. The optimized version of the code, instead of calculating the dirty price for each bond and appending it to a list, applies the bond valuation function to the whole dataset. The optimized version of the code calculated dirty prices for a million bonds in only 278 seconds or 4.6 minutes, making it much faster and suitable for applying to a large dataset.

Results and Analysis

The aim of the project was the creation of a bond valuation calculator, which would evaluate the dirty price and other parameters of the bond, and, most importantly, would be fast and accurate. To check the accuracy of the calculator, the calculator was applied to a dataset scraped from the Central Bank of Armenia website. The desired output included scraped dirty prices on the left side and real dirty prices on the right side. The results showed that the calculator has a hundred percent accuracy and can be applied to evaluate bonds from the Armenian financial market.

	Real Dirty Price	Calculated Dirty Price
0	99.4308863	99.4608286
1	99.0564787	99.0564787
2	110.5500513	110.5500513
3	107.3681638	107.3681638
4	95.02643	95.02643
5	110.9941415	110.9941415
6	100.8239823	100.8239823
7	114.1387587	114.1387587
8	93.9323582	93.9323582
9	91.8127554	91.8127554
10	92.2551987	92.2551987
11	95.4863007	95.4863007
12	88.9702636	88.9702636
13	96.8674257	96.8674257
14	88.995324	88.995324
15	85.8693699	85.8693699
16	81.578031	81.578031
17	96.1797845	96.1797845
18	93.5832212	93.5832212
19	85.733115	85.733115
20	88.8805116	88.8805116
21	103.2282975	103.2260979
22	98.80505	98.80505
23	97.4865924	97.4865924
24	105.3948127	105.3948127
25	93.1624877	93.1624877
26	92.5713198	92.5713198
27	89.7124007	89.7124007

The other goal of this project was to make the code suitable to apply to large datasets. For reaching this goal, the need for optimization arose. The optimized version of the code applies the calculator on the whole dataset instead of calculating bond prices one by one, which makes the process way faster. The desired running time to achieve was a maximum of 5 minutes. However, the optimized code calculated dirty prices for million rows of data in 278 seconds, which is approximately 4.6 minutes.

Conclusion

The desirable output of this capstone project was to create an infrastructure that would be in the form of a calculator for market participants to use, as it should be open source and accessible to everyone. The work done in all phases of the project has brought this project to its intended goal. As a result, the bond valuation calculator was created and performed well on the real bond dataset, accurately calculating the values of the bonds. The results were compared to the actual bond prices to ensure the accuracy of the calculator. The interface created for the calculator made it easy to input bond information and receive the relevant output. More than this, the calculator is capable of calculating the dirty prices of a million bonds in less than 5 minutes. The calculator was created to fill a gap in the Armenian market and can be used by various financial institutions. Moreover, the calculator is open source, and everyone can use it for their own purposes, as well as make changes to it to adjust the calculator for their specific needs and regulations of the region they are living in. Overall, the development of a bond valuation calculator using Python can streamline the process of valuing bonds, making it more efficient. accurate and fast. Future work can include the improvement of the interface of the calculator, turning it into a website accessible to everyone having any kind of expertise. It can also include the development of additional features for the calculator, such as the ability to analyze bond portfolios or compare bonds across different issuers. Moreover, parallelism can be used to make the calculation process way more optimized, taking less time to run the code on the dataset than now.

References

- Frank J. Fabozzi. (2012). The Handbook of Fixed Income Securities: The Handbook of Fixed Income Securities Eighth Editionhttp://dspace.vnbrims.org jspui bitstream The ...

- Wes McKinney. (2013). Python for Data Analysis: Data Wrangling with Pandas, NumPy Jupyter. Python for Data Analysishttps://bedford-computing.co.uk uploads 2015/10

- AMX — . https://amx.am/

- Central Bank of Armenia. https://www.cba.am/am/SitePages/Default.