

Homework 2

- Author: Ruchit Jerambhai Vithani
- Created on: 09/18/2022
- GTID: 903829303

How to Run

- Download and extract the zip file
- There are three c++ files, namely `main.cpp` , `Bond.cpp` and `UnitTest.cpp` .
- Compile the program using following command

Compiling

```
g++ -o A1 Bond.cpp main.cpp UnitTest.cpp
```

Running the code

```
./A1
```

Implementation details

1. Bond Pricing Function

```
double Bond::Price(double face_value, double maturity_time, double interest_rate) const {  
    double present_value = 0;  
  
    int n = floor(maturity_time / this->frequency);  
    double regular_coupon = face_value * this->coupon_rate * this->frequency;  
    double last_coupon = (maturity_time - n * this->frequency) * this->coupon_rate * face_value;  
    double last_amount = last_coupon + face_value;  
  
    double discount_factor;  
    for (int i=1; i <= n; i++){
```

```

        discount_factor = exp(-1 * interest_rate * this->frequency * i);
        present_value += discount_factor*regular_coupon;
    }

    discount_factor = exp(-1 * interest_rate * maturity_time);
    present_value += discount_factor * last_amount;

    return present_value;
}

```

- First we find the number of coupon payments from the maturity time and frequency.
- We then calculate the last coupon amount separately as it is different from the normal coupon payment. Last coupon is `coupon_rate * (remaining maturity in the last payment) * face_value`
- We also calculate the regular coupon payment amount.
- Discount factor is calculated based on the formula provided in the homework pdf
- Finally, the present value is also calculated based on the formula in the homework sheet and the value is returned.

2. Calculating the maturity between start date and end date

- It was assumed that each month contains approximately 30 days. This makes the maturity calculations easy and accurate.
- Total number of days between given two dates was calculated based on year, month and day difference.

3. Finding the closest maturity from the interest rate date

- Since the maturity was sorted in descending order, binary search algorithm can be used to search for the closest maturity in the table to the target maturity.

```

double find_interest_rate(vector<double> maturity, vector<double> interest_rate, double target)
// Binary Search Algorithm to find the target interest rate
int start = 0;
int end = int(maturity.size()) - 1;
int mid;

while (start <= end){
    mid = (start + end) / 2;
    if (abs(maturity[mid] - target_maturity) <= 0.0001){
        return interest_rate[mid];
    } else if (maturity[mid] > target_maturity){

```

```

        start = mid + 1;
    } else {
        end = mid - 1;
    }
}
return interest_rate[end + 1];
}

```

4. Unit tests

- One test case was picked up from the excel sheet provided with the homework
- Two other test cases were manually generated to increase the confidence of the function
- Unit test function is called before pricing the bond in question 3 to make sure that things are working as expected.

Results:

- Please find the screenshot attached containing the output of running the program:

```

iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
ruchitvithani@lawn-143-215-94-214:~/Documents/georgia_tech/Assignments/sem-1/System Design for Computational Finance/Homework 2
> g++ -o A1 Bond.cpp main.cpp UnitTest.cpp
> ./A1
Bond 1 characteristics (Bond initialised by default constructor): Bond(00/00/0000,0.000000,0.000000)
Bond 2 characteristics: Bond(19/11/2035,0.500000,0.070000)

Running tests....
Face Value: 100 Maturity: 4.2 Interest Rate: 0.07 Coupon Rate: 0.08 Frequency: 0.5 Target Price: 103.144 Achieved Price: 103.145 OK!
Face Value: 100 Maturity: 6 Interest Rate: 0.08 Coupon Rate: 0.1 Frequency: 0.3 Target Price: 108.961 Achieved Price: 108.961 OK!
Face Value: 100 Maturity: 1 Interest Rate: 0.09 Coupon Rate: 0.04 Frequency: 0.4 Target Price: 95.156 Achieved Price: 95.1565 OK!

Underlying bond matures on: 01/01/2020
Current Date: 01/01/2016 Maturity Date: 01/01/2020 Maturity value: 4 Interest Rate: 0.003263
Current Date: 01/01/2017 Maturity Date: 01/01/2020 Maturity value: 3 Interest Rate: 0.0054175
Current Date: 01/01/2018 Maturity Date: 01/01/2020 Maturity value: 2 Interest Rate: 0.0124867
Current Date: 01/01/2019 Maturity Date: 01/01/2020 Maturity value: 1 Interest Rate: 0.01455
Current Date: 01/01/2020 Maturity Date: 01/01/2020 Maturity value: 0 Interest Rate: 0.0254056

Arithmetic average price between 2016 and 2020: 108.537
Current Date: 03/08/2015 Maturity Date: 01/01/2020 Maturity value: 5.41111 Interest Rate: 0.004419

Actual Price of security is: 105.972. It's a good investment!!!

```