Queens College, CUNY, Department of Computer Science

Computational Finance

CSCI 365 / 765

Spring 2018

Instructor: Dr. Sateesh Mane

Midterm 1 Spring 2018

due Monday Feb. 19, 2018, 11:59 pm

- **NOTE: It is the policy of the Computer Science Department to issue a failing grade to any student who either gives or receives help on any test.**
- This is an **open-book** test.
- **Any problem to which you give two or more (different) answers receives the grade of zero automatically.**
- This is a **take home exam.**
  Please submit your solution via email, as a file attachment, to `Sateesh.Mane@qc.cuny.edu`.
  The file name should have either of the formats:

  `StudentId_first_last_CS365_midterm1_Feb2018`

  `StudentId_first_last_CS765_midterm1_Feb2018`

  Acceptable file types are txt, doc/docx, pdf (also cpp, with text in comment blocks).
- **In all questions where you are asked to submit programming code, programs which display any of the following behaviors will receive an automatic F:**
  1. Programs which do not compile successfully (compiler warnings which are not fatal are excluded, e.g. use of deprecated features).
  2. Array out of bounds.
  3. Dereferencing of uninitialized variables (including null pointers).
  4. Operations which yield NAN or infinity, e.g. divide by zero, square root of negative number, etc. *Infinite loops.*
  5. Programs which do NOT implement the public interface stated in the question.
- **In addition, note the following:**
  1. Programs which compile and run successfully but have memory leaks will receive a poor grade (but not F).
  2. All debugging and/or output statements (e.g. `cout` or `printf`) will be commented out.
  3. Program performance will be tested solely on function return values and the values of output variable(s) in the function arguments.
  4. In other words, program performance will be tested solely via the public interface presented to the calling application. (I will write the calling application.)

1 Question 1

1.1

Today's time is denoted by t_0, in all the cases below.

1.2

The future value of a cashflow at time t_1 is F_1. The present value (today) of the cashflow is F_0. The discount factor of the future cashflow is d. Write the formula for d in terms of the input parameters:

$$d = (\text{formula})\,. \tag{1.1}$$

Let r denote the **continuously compounded interest rate** over the time interval from t_0 to t_1. Write the formula for d in terms of r and any other input parameters:

$$d = \text{function of } r, \text{ etc.} \tag{1.2}$$

1.3

Suppose $F_1 = 120.0$, $t_0 = 0.0$ and $t_1 = 1.5$. Let $r = 4.0\%$ (so in decimal $r = 0.04$). Calculate the numerical value of the discount factor d in eq. (1.2):

$$d = (\text{number})\,. \tag{1.3}$$

Express your answer to **four decimal places.**
Your answer should lie between 0.9 *and* 1.0.

1.4

Use the number from eq. (1.3) (rounded to 4 d.p.) to calculate the present value of the cashflow (i.e. the value of F_0) for the scenario in Sec. 1.3:

$$F_0 = (\text{number})\,. \tag{1.4}$$

Express your answer to **two decimal places.**

2 Question 2

- In this question, we shall perform a simple bootstrap of a yield curve.
- The time today is $t_0 = 0$ in this question.
- You are given three newly issued par bonds, say $B_{0.5}$, B_1 and $B_{1.5}$.
  1. All the bonds have face $F = 100$.
  2. The maturities of the bonds are $T_{0.5} = 0.5$, $T_1 = 1.0$ and $T_{1.5} = 1.5$ (in years).
  3. All the bonds pay semiannual coupons (two coupons per year).
  4. All the bonds have constant rate coupons (not variable rate).
- The yields of the bonds are obtained as follows:
  1. **Take the first 4 digits of your student id. Define:**
  $$\Delta y_1 = 2 \times \frac{\text{first 4 digits of your student id}}{10^4} \,. \tag{2.1}$$
  2. **Take the last 4 digits of your student id. Define (note the minus sign):**
  $$\Delta y_{1.5} = -2 \times \frac{\text{last 4 digits of your student id}}{10^4} \,. \tag{2.2}$$
  3. For example if your student id is 23054617, then
  $$\Delta y_1 = 2 \times 0.2305 \,, \qquad \Delta y_{1.5} = -2 \times 0.4617 \,. \tag{2.3}$$
  4. Note that $\Delta y_{1.5}$ is **negative.**
  5. The bonds have the following yields (in percent):
  $$y_{0.5} = 5.0\% \,, \qquad y_1 = (5.0 + \Delta y_1)\% \,, \qquad y_{1.5} = (5.0 + \Delta y_{1.5})\% \,. \tag{2.4}$$
  6. **Write down the values of y_1 and $y_{1.5}$ to four decimal places.**
  7. *If you have done your work correctly, then* $5.0 \le y_1 \le 7.0$ *and* $3.0 \le y_{1.5} \le 5.0$.
- **Calculate the discount factors $d_{0.5}$, $d_{1.0}$, $d_{1.5}$.**
  Express your results to **four decimal places.**
- **Calculate the continuously compounded spot rates $r_{0.5}$, $r_{1.0}$, $r_{1.5}$.**
  Express your answers as **percentages, to two decimal places.**

3 Question 3

- The time today is $t_0 = 0$ in this question.
- Suppose we have three cashflows $CF_{0.5}$, CF_1 and $CF_{1.5}$ at (future) times $t_{0.5}$, t_1 and $t_{1.5}$, respectively. Let the discount factors of the three cashflows be $d_{0.5}$, d_1 and $d_{1.5}$, respectively. Let the present value of the total set of the three cashflows be CF_0 (i.e. the value today).
- **Write a mathematical formula for CF_0 in terms of the input parameters:**

$$CF_0 = \text{function of } \{CF_{0.5}, CF_1, CF_{1.5}\}, \text{ etc.} \tag{3.1}$$

- Let the times be $t_{0.5} = 0.5$, $t_1 = 1.0$ and $t_{1.5} = 1.5$.
- Let the values of the cashflows be $CF_{0.5} = 2.0$, $CF_1 = 2.0$ and $CF_{1.5} = 102.0$.
- **The discount factors $d_{0.5}$, d_1 and $d_{1.5}$ are given by the values you calculated in Question 2 (to 4 d.p.).**
- **Calculate the numerical value of CF_0 in eq. (3.1) (answer to FOUR decimal places).**

$$CF_0 = (\text{number})\,. \tag{3.2}$$

4 Question 4

- The time today is $t_0 = 0$ in the question below.
- A newly issued bond has a maturity of 18 months (= 1.5 years).
- The bond has a face $F = 100$.
- The bond pays semi-annual coupons with an annualized coupon rate of $c = 4$.
- The times of the cashflows are $t_1 = 0.5$, $t_2 = 1.0$ and $t_3 = 1.5$.
- The yield of the bond is y.
- **Fill in the table below with the values of $B(y)$ (answers to four decimal places).**

y (%)	$B(y)$
0	(4 d.p.)
1	(4 d.p.)
2	(4 d.p.)
3	(4 d.p.)
4	(4 d.p.)
5	(4 d.p.)
6	(4 d.p.)
7	(4 d.p.)
8	(4 d.p.)
9	(4 d.p.)

- **The market price of the bond is given by the value of CF_0 from eq. (3.2).**
- Denote this value by B_{market}.
- **Find the smallest interval $(y_{\text{low}}, y_{\text{high}})$ in the above table which gives a (lower, upper) bound for the bond yield.**
  1. Find the *smallest interval* in the above table.
  2. **Solutions which state $(y_{\text{low}}, y_{\text{high}}) = (0.0, 9.0)$ will receive zero credit.**
  3. Don't try to be too clever and calculate your own "improved" (lower, upper) bound.
  4. Use the numbers in the above table.
- **Calculate the value of the midpoint:**

$$y_{\text{mid}} = \frac{y_{\text{low}} + y_{\text{high}}}{2.0} \,. \tag{4.1}$$

- **Calculate the bond price $B(y_{\text{mid}})$ (answer to 4 decimal places).**

5 Question 5: Mandatory for graduate students (optional for undergraduates)

- Use the same data as in Question 4.
- **State which value y_{low} or y_{high} should be updated for the next iteration step.**
- Update the value y_{low} or y_{high} for the next iteration step.
- **Calculate the new value of y_{mid} and the bond price $B(y_{\text{mid}})$.**
- **Stop.** Do not perform further iterations.

6 Question 6 (bonus question)

- Consider a newly issued bond, which pays semi-annual coupons. Let the bond pay n cashflows:

$$B = \frac{\frac{1}{2}c}{1+\frac{1}{2}y} + \frac{\frac{1}{2}c}{(1+\frac{1}{2}y)^2} + \cdots + \frac{F+\frac{1}{2}c}{(1+\frac{1}{2}y)^n}\,. \tag{6.1}$$

- Here is a C++ function to sum the above cashflows (the input yield value is a percentage):

```
double bond_price_from_yield(double F, double c, double y, int n)
{
  double B = 0.0;
  if ((F <= 0.0) || (c < 0.0) || (y < 0.0) || (n <= 0)) {
    return B;
  }
  double temp1 = 1.0 + (0.01*y)/2.0;
  double temp2 = temp1;

  for (int i = 1; i < n; ++i) {
    B = B + (0.5*c) / temp2;
    temp2 = temp2 * temp1;
    if (i == n-1) {
      B = B + (F + 0.5*c) / temp2;
    }
  }
  return B;
}
```

- **Question: does the above implementation work correctly?**
- *Justify your answer, i.e. do not merely reply "yes" (or "no") but explain your reasons.*
- ***Note: "Is it correct?" is NOT the same as "is it computationally efficient?"***
  This question is NOT asking about efficiency.
  Obviously one can find a more efficient implementation.
- The question is: *does the above function output the correct value for B, if the input values to the function contain valid data?* **Justify your answer.**