IE523: Financial Computing
Fall, 2019
Mid-Term Programming Assignment: Portfolio Optimization Software in C++ under Parallel Shifts in Term Structure
Due Date: 18 October 2018

Instructions

1. **If you worked with someone on a specific problem, make sure you tell me about it. If you do not do this and I notice a verbatim reproduction of material, I consider it as plagiarism. There can be serious consequences to this.**
   (a) I will randomly pick ≈ 6 students in the class, who will be asked to explain all aspects of their submitted code to me. Do not be disturbed if your name appears in this list – I just need to go through it to ensure the grading is fair.
2. **If you picked-up the solution from some source, make sure you mention it. If you do not do this and I notice a verbatim reproduction of material, I consider it as plagiarism. There can be serious consequences to this.**
3. **Each submitted document should be personalized. I expect to see**
   (a) C++ code and header files (if any), the comments in the code should pretty much tell me what you are trying to do
   (b) A PDF version of sample-runs that accompanies each submission, have to be uploaded on Compass by the due date.
4. **The written In-Class Exam will be held during class hours on 17 October 2018.**

I want you to write a comprehensive, readable (i.e. put a lot of comments; have meaningful variable names; etc. etc.) *Portfolio Optimization Software Package* in C++ assuming you have parallel shifts in the term structure.

Input Specification: The input file to your software will have the following format:

1st-line:	#CFs									
2nd-line:	CF-1's PV	Maturity	cf_1^1	cf_2^1	cf_3^1	cf_4^1	$\cdots$	$cf_{n_1}^1$		
3rd-line:	CF-2's PV	Maturity	cf_1^2	cf_2^2	cf_3^2	cf_4^2	$\cdots$	$\cdots$	$cf_{n_2}^2$	
4th-line:	CF-3's PV	Maturity	cf_1^2	cf_2^2	cf_3^2	$\cdots$	$cf_{n_3}^2$			
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	
kth-line:	CF-k's PV	Maturity	cf_1^k	cf_2^k	cf_3^k	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$cf_{n_k}^k$
$(k+1)$*th-line:*	FV of user's debt obligation	due date								

(i.e. k-many Bonds; the 1st number in each row is the PV of the bond, followed by the bond's cash-flows; not all of them have the same maturity; followed by the FV of user's debt obligation and the time when its is due).

Output Specification 1: We do not assume a flat term structure, but we assume the term structure changes are *parallel* (i.e. the yield changes by the same %-age amount for all maturities).

1. Compute the *YTM* for each cash flow.
2. Compute the *duration* for each cash flow.
3. Compute the *convexity* for each cash flow.

Output Specification 2 Assuming no changes in term structure, compute the %-age of the face value of each cash flow you would have to purchase to meet the future debt obligation.

Output Specification 3: Pick the bond-portfolio that will meet this obligation when it is due, and has the largest convexity among all possible portfolio choices. You should pose this as a Linear Programming Problem and use lp_solve to find the optimal answer. Your final answer will say that we need to buy λ_1% of first cash flow, λ_2% of second cash flow, λ_3% of third cash flow, etc. No short-selling is permitted (i.e. all your λ's have to be non-negative).

1. In your write-up, present an explanation of the strategy/method that you used in picking the best portfolio.
2. Make sure your program will handle the case when there is no portfolio that will meet the debt obligation if we are worried about (parallel) movements in the term structure (cf. figure 2).

I also want to see a couple of sample runs (cook-up your own data; or use lesson 4 of my notes).

Partial Credit Information: This should help you decide how your efforts should be spent in the next two weeks.

1. (*20 points*) Making sure you catch every possible error/infeasibility that might occur in the general setting.
2. (*20 points*) Soundness of your theoretical arguments for the design-methodology that you adopt for the portfolio design.
3. (*60 points*) The correct functioning of your code.

A sample screen shot is shown in figure 1.

```
wirelessprv-10-194-149-149:Debug sreenivas$ ./Bond\ Immunization input1
Input File: input1
We owe 1790.85 in 10 years
Number of Cash Flows: 5
------------------------------------------------------------------
Cash Flow #1
Price = 1131.27
Maturity = 10
Yield to Maturity = 0.0499999
Duration = 7.7587
Convexity = 70.4264
------------------------------------------------------------------
Cash Flow #2
Price = 1069.88
Maturity = 15
Yield to Maturity = 0.0625639
Duration = 9.93582
Convexity = 119.831
------------------------------------------------------------------
Cash Flow #3
Price = 863.5
Maturity = 30
Yield to Maturity = 0.07
Duration = 13.6774
Convexity = 262.769
------------------------------------------------------------------
Cash Flow #4
Price = 1148.75
Maturity = 12
Yield to Maturity = 0.0574999
Duration = 8.58082
Convexity = 87.6798
------------------------------------------------------------------
Cash Flow #5
Price = 1121.39
Maturity = 11
Yield to Maturity = 0.0549998
Duration = 8.20531
Convexity = 79.1966
******************************************************************
Model name:
                C1       C2       C3       C4       C5
Minimize  -70.4264 -119.831 -262.769 -87.6798 -79.1966
R1               1        1        1        1        1  =        1
R2          7.7587  9.93582  13.6774  8.58082  8.20531  =       10
Type          Real     Real     Real     Real     Real
upbo           Inf      Inf      Inf      Inf      Inf
lowbo            0        0        0        0        0
Largest Convexity we can get is: 143.262
Optimal portfolio:
%Cash Flow:1  0.621321
%Cash Flow:2  0
%Cash Flow:3  0.378679
%Cash Flow:4  0
%Cash Flow:5  0
******************************************************************
To immunize against small changes in `r' for each $1 of PV, you should buy
$0.621321 of Cash Flow#1
$0.378679 of Cash Flow#3
If you need to immunize for a larger PV-value, just buy an appropriate proportion
******************************************************************
For example, if you want to immunize for $500 of PV, buy
$310.661 of Cash Flow#1
$189.339 of Cash Flow#3
******************************************************************
For example, if you want to immunize for $750 of PV, buy
$465.991 of Cash Flow#1
$284.009 of Cash Flow#3
******************************************************************
For example, if you want to immunize for $1000 of PV, buy
$621.321 of Cash Flow#1
$378.679 of Cash Flow#3
******************************************************************
For example, if you want to immunize for $1009.36 of PV, buy
$627.137 of Cash Flow#1
$382.223 of Cash Flow#3
******************************************************************
wirelessprv-10-194-149-149:Debug sreenivas$
```

Figure 1: A sample screen shot.

```
wirelessprv-10-194-149-149:Debug sreenivas$ ./Bond\ Immunization input2
Input File: input2
We owe 1790.85 in 10 years
Number of Cash Flows: 3
--------------------------------------------------------------------
Cash Flow #1
Price = 1131.27
Maturity = 10
Yield to Maturity = 0.0499999
Duration = 7.7587
Convexity = 70.4264
--------------------------------------------------------------------
Cash Flow #2
Price = 1121.39
Maturity = 11
Yield to Maturity = 0.0549998
Duration = 8.20531
Convexity = 79.1966
--------------------------------------------------------------------
Cash Flow #3
Price = 1148.75
Maturity = 12
Yield to Maturity = 0.0574999
Duration = 8.58082
Convexity = 87.6798
********************************************************************
Model name:
                C1       C2       C3
Minimize  -70.4264 -79.1966 -87.6798
R1               1        1        1  =        1
R2          7.7587  8.20531  8.58082  =       10
Type          Real     Real     Real
upbo           Inf      Inf      Inf
lowbo            0        0        0
There is no portfolio that meets the duration constraint of 10 years
wirelessprv-10-194-149-149:Debug sreenivas$
```

Figure 2: A sample screen shot of an infeasible set of cashflows.

```
wirelessprv-10-194-149-149:Debug sreenivas$ ./Bond\ Immunization input3
Input File: input3
We owe 1790.85 in 10 years
Number of Cash Flows: 3
-----------------------------------------------------------------
Cash Flow #1
Price = 1051.52
Maturity = 10
Yield to Maturity = 0.0600001
Duration = 7.6655
Convexity = 67.9958
-----------------------------------------------------------------
Cash Flow #2
Price = 1095.96
Maturity = 15
Yield to Maturity = 0.0599997
Duration = 10
Convexity = 121.484
-----------------------------------------------------------------
Cash Flow #3
Price = 986.24
Maturity = 30
Yield to Maturity = 0.0599996
Duration = 14.6361
Convexity = 296.143
*****************************************************************
Model name:
                C1       C2       C3
Minimize  -67.9958 -121.484 -296.143
R1               1        1        1  =        1
R2          7.6655       10  14.6361  =       10
Type          Real     Real     Real
upbo           Inf      Inf      Inf
lowbo            0        0        0
Largest Convexity we can get is: 144.404
Optimal portfolio:
%Cash Flow:1  0.665093
%Cash Flow:2  0
%Cash Flow:3  0.334907
*****************************************************************
To immunize against small changes in `r' for each $1 of PV, you should buy
$0.665093 of Cash Flow#1
$0.334907 of Cash Flow#3
If you need to immunize for a larger PV-value, just buy an appropriate proportion
*****************************************************************
For example, if you want to immunize for $500 of PV, buy
$332.546 of Cash Flow#1
$167.454 of Cash Flow#3
*****************************************************************
For example, if you want to immunize for $750 of PV, buy
$498.82 of Cash Flow#1
$251.18 of Cash Flow#3
*****************************************************************
For example, if you want to immunize for $1000 of PV, buy
$665.093 of Cash Flow#1
$334.907 of Cash Flow#3
*****************************************************************
For example, if you want to immunize for $1009.36 of PV, buy
$671.318 of Cash Flow#1
$338.042 of Cash Flow#3
*****************************************************************
wirelessprv-10-194-149-149:Debug sreenivas$
```

Figure 3: A sample screen shot for the example that is done in lesson 4. You can compare these results with those in the Brute-Force Excel Spreadsheets provided on Compass with Lesson 4.