The Cooper Union
EID378 Finance
Prof. Fred L. Fontaine

Problem Set II: Interest Rates and Bonds

March 9, 2022

1. The following code will perform backsubstitution: if A is a lower triangular matrix with nonzero elements on the diagonal, this will solve $y = Ax$. Here, n is the length of the x, y vectors (A is $n \times n$):

```
import numpy as np

x= np.zeros(n)

for i in range(0,n):

    tmp= y[i]

    for j in range(0,i):

        tmp -= x[j]*A[i,j]

    x[i]= tmp/A[i,i]
```

Feel free to modify it, convert it to a function, etc. It does not do error checking, and is based on the premise that $A[i,j] = 0$ for $j > i$ (the code will still run if not, it just won't solve $y = Ax$ correctly), and that $A[i,i] \neq 0$.

Write Python code to do the following. Assume spot rates r and forward rates f are in continuous-compounding form, for simplicity. Given N bonds with prescribed coupon rates (coupons payable semi-annually), face value, current prices, and maturities exactly $0.5m$ years from today (i.e., coinciding with the coupon payments), with $m = 1, 2, \cdots, N$. You can have these values stored in whatever form you prefer- in separate *numpy*.array (vectors) or all together in a dataframe; note the above code, however, assumes y and A are *numpy* arrays.

(a) Solve for the discount factors $Z(0, 0.5m)$ and from that term structure, i.e., $r(0, 0.5m)$. Specifically, set up the matrix and use backsubstitution to solve for the term structure.

(b) From the term structure, compute the forward rates $f(0, 0.5m, 0.5m + 0.5)$ for $0 \leq m \leq N - 1$.

(c) Now apply your code to the following problem, assuming face value \$1000 for each bond.

Bond Name	A	B	C	D
Maturity	0.5	1	1.5	2
Bond Price	985.86	974.70	967.99	966.82
Coupon	2%	3%	4%	5%

(d) Plot the term structure (as a continuous curve), and display the forward rates.

2. Refer to the Python code below. It will compute the YTM from information about a bond (face value, price and coupon rate).

```
## From Mastering Python in Finance (with typo corrections by FF)

import scipy.optimize as optimize

def bond_ytm(price,FaceVal,T,coup,freq=2,guess=0.05):

  freq= float(freq)

  periods= T*freq

  coupon= coup/100.*FaceVal/freq

  dt= [(i+1)/freq for i in range(int(periods))]

  ytm_func= lambda y:  sum([coupon/(1+y/freq)**(freq*t) \

      for t in dt])+ FaceVal/(1+y/freq)**periods-price

  return optimize.newton(ytm_func,guess)
```

Write Python code that: given the term structure (annual) $r_1(0,m)$, $1 \le m \le N$, and *annual* coupon rate (i.e., assume the coupons are paid annually) of a bond, will compute the price per \$1 face value and YTM of the bond.

Now take $N = 10$ and assume the Nelson-Siegel model with parameters $\beta_0 = 0.02$, $\beta_1 = 0.02$, $\beta_2 = 0.20$, $\tau = 5$:

$$r_1(0,T) = \beta_0 + (\beta_1 + \beta_2)\frac{\tau_1}{T}\left(1 - e^{-T/\tau_1}\right) - \beta_2 e^{-T/\tau_1}$$

Compute the price per \$1 face value and YTM for coupon rates $0\%, 1\%, \cdots, 9\%$, and plot each curve.

3. **Extra Credit:** In the previous problem, you used the Nelson-Siegel model to determine the annual term structure. Now go backwards: from the computed term structure, fit the parameters $\beta_0, \beta_1, \beta_2, \tau_1$, and see if you get back what you started with. Now let's try some noise: to each $r_1(0,T)$, add or subtract 5 basis points (each independently, with $\pm$ equally likely), and try to fit the model again. If this noise level seems too high or too low (giving either radically different answers or too similar) try varying the size of the noise (i.e, 5 basis points). In any case, comment on the sensitivity of the model parameters to variations in the data.

4. The Macauley duration D_{mac} assuming annual compounding can be expressed as:

$$D_{mac} = 1 + \frac{1}{y_1} + \frac{T(y_1 - c) - (1 + y_1)}{c\left[(1+y_1)^T - 1\right] + y_1}$$

where y_1 is the YTM, c is the coupon rate and T is time-to-maturity in years. As a hint to its behavior, regardless of c:

$$\lim_{T \longrightarrow \infty} D_{mac} = 1 + \frac{1}{y_1}$$

   (a) Write a function in Python to compute D_{mac} from these three parameters.

   (b) As an example, set $y_1 = 10\%$ and graph duration as a function of time to maturity, up to $T = 100$ years, for $c = 2\%$ and 10% (superimposed) with a horizontal line indicating the limiting value $D_{mac}(T = \infty)$. [The reason for going out to 100 years is to show the convergence.] This should replicate a graph in the Brandimarte text.

   (c) Generate several plots to help us visualize how D_{mac} varies with these parameters. What you do is up to you. We don't want 1,000 plots, and the plots should have reasonable values (the above was an exception- don't take $T > 30 years$ normally). Do whatever you think is reasonable to illustrate how D_{mac} varies with these parameters.

5. Companies A and B have been offered the following rates per year on a \$1million 5-year investment:

	Fixed Rate	Floating Rate
A	8.8%	LIBOR
B	8.0%	LIBOR

   (a) Company A prefers a fixed-rate loan, and company B prefers a floating-rate loan. Bank X has been engaged as an intermediary for a swap. The swap should be equally attractive to each company, and the bank should earn 0.2% annually. Design the swap.

   (b) Suppose instead company A were offered a fixed rate of 8.0% and company B a rate of 8.8%. If you repeat your calculation, you will find a problem with it, and neither company would actually engage in the transaction. Explain in *words,* based on the discussion in the lecture: why doesn't a swap make sense here?