

UNIWERSYTET GDAŃSKI
WYDZIAŁ MATEMATYKI, FIZYKI I INFORMATYKI

Jakub Hinc

Kierunek studiów: **Modelowanie matematyczne i analiza danych**
Numer albumu: **275123**

Projekt
Analiza instrumentów finansowych

Spis treści

1	Wstęp	2
1.1	Schemat projektu i założenia	3
2	Kod	5
2.1	efektywna stopa zwrotu	7
2.2	Struktura terminowa	10
2.3	Przedłużenie I metoda	10
2.4	Przedłużenie II metoda	12
3	Wyniki i wnioski	14
4	Źródła:	15

Rozdział 1

Wstęp

Celem projektu jest analiza obligacji pod względem ich rentowności. W tym celu zostało wybranych pięć instrumentów, z których wszystkie wyceniane są w walucie Euro. Dane dla obligacji zostały bezpośrednio pobrane ze strony GPW dla rynku catalyst. Wycena będzie dokonywana na 2022-10-23 (data zakupu). Poniżej przedstawiono obligacje wraz z parametrami wejściowymi:

- ECH1024:
 - Wartość nominalna: 1000 EUR
 - Data wykupu: 2024-10-23
 - Oprocentowanie: 0.04500
 - Kupon (ile razy wypłacany w ciągu roku): 2
 - Data ostatniego kuponu: 2022-10-23
- MBHPE11
 - Wartość nominalna: 1000 EUR
 - Data wykupu: 2025-04-24
 - Oprocentowanie: 0.01285
 - Kupon (ile razy wypłacany w ciągu roku): 1
 - Data ostatniego kuponu: 2022-04-24

- EUR0126
 - Wartość nominalna: 1000 EUR
 - Data wykupu: 2026-01-19
 - Oprocentowanie: 0.01500
 - Kupon (ile razy wypłacany w ciągu roku): 1
 - Data ostatniego kuponu: 2022-01-19
- EUR1027
 - Wartość nominalna: 1000 EUR
 - Data wykupu: 2027-10-22
 - Oprocentowanie: 0.01375
 - Kupon (ile razy wypłacany w ciągu roku): 1
 - Data ostatniego kuponu: 2022-10-22
- EUR1028
 - Wartość nominalna: 1000 EUR
 - Data wykupu: 2028-10-25
 - Oprocentowanie: 0.01000
 - Kupon (ile razy wypłacany w ciągu roku): 1
 - Data ostatniego kuponu: 2022-10-25

1.1 Schemat projektu i założenia

W pierwszej kolejności na podstawie danych przedstawionych na wstępie zostanie stworzona tabela w excelu zawierająca wszystkie niezbędne dane. W oparciu o wprowadzone wartości zostaną obliczone dodatkowo kolumny potrzebne w dalszej części projektu, jak np. wartość kuponu, cena brudna. Następnie dla każdej obligacji zostanie wyliczona efektywną stopę zwrotu, która jest procentową wartością, mówiącą nam, jakiego rodzaju stopy zwrotu możemy oczekiwać po uwzględnieniu wszystkich czynników, takich jak kupon, wartość zakupu oraz

czas do wykupu. Jest to ważne dla inwestorów, ponieważ pozwala porównać potencjalne zyski pomiędzy różnymi instrumentami. Następnie mając już dla każdej obligacji ERR zostanie wyznaczona struktura terminowa w oparciu o wzór:

$$B(t_i) = (1 + r_i)^{-t_i}$$

Otrzymując strukturę terminową obligacji jesteśmy teraz w stanie przedłużyć je, czyli tak naprawdę interpolować w oparciu o dwie metody:

$$B(t) = B(t_{i-1})^{\frac{t_{i-1}-t}{t_i-t_{i-1}}} B(t_i)^{\frac{t-t_{i-1}}{t_i-t_{i-1}}}$$

$$B(t) = B(t_{i-1}) + (B(t_i) - B(t_{i-1})) \frac{t - t_{i-1}}{t_i - t_{i-1}}$$

Rozdział 2

Kod

W pierwszej kolejności zostały załadowane dane wejściowe dla wybranych obligacji, a dodatkowo tabela ta została wyświetlona:

```
[1]: df = pd.read_excel("obligacje-dane.xlsx")
df
```

```
[1]:
```

	Nazwa	Wartość nominalu	Data wykupu	Oprocentowanie	kupon	\
0	ECH1024	1000	2024-10-23	0.04500	2	
1	MBHPE11	1000	2025-04-24	0.01285	1	
2	EUR0126	1000	2026-01-19	0.01500	1	
3	EUR1027	1000	2027-10-22	0.01375	1	
4	EUR1028	1000	2028-10-25	0.01000	1	

	wartosc rynkowa	Data zakupu	Data ostatniego kuponu
0	1.0000	2022-12-01	2022-10-23
1	1.0000	2022-12-01	2022-04-24
2	1.0300	2022-12-01	2022-01-19
3	0.9907	2022-12-01	2022-10-22
4	1.0650	2022-12-01	2022-10-25

Dodanie liczby dni od ostatniego kuponu wyrażona w dniach

```
[2]: df["liczba dni"] = (df["Data zakupu"] - df["Data ostatniego ↵
↵ kuponu"]).dt.days
```

Kolejny krok miał na celu obliczenie ceny czystej, a więc pomnożenie ceny nominalnej przez wartość rynkową obligacji:

```
[3]: df["cena czysta"] = df["Wartość nominalu"] * df["wartosc rynkowa "]
```

W kolejnym kroku została obliczona wartość kuponu z uwzględnieniem jak często w ciągu roku jest wypłacany

```
[4]: df["wartosc kuponu"] = df["Wartość nominalu"] * df["Oprocentowanie_↵↵"] / df["kupon"]
```

Następnie została obliczona cena brudna, na podstawie wcześniej obliczonych wartości:

```
[5]: df["cena brudna"] = df["cena czysta"] + ((df["liczba dni"] + 4) / ↵↵365) * df["wartosc kuponu"]
```

Do tabeli dodałem także czas życia obligacji, a więc wyrażony w latach okres, który pozostał do wykupu obligacji

```
[6]: df["czas zycia"] = (df["Data wykupu"] - df["Data zakupu"]).dt.days / ↵↵365
```

Liczba kuponów obligacji została obliczona funkcją ceil, a dodatkowo wyrażenie zostało pomnożone przez liczbę kuponów w ciągu roku

```
[7]: df["liczba_kuponow"] = df["czas zycia"].apply(ceil) * df["kupon"]
```

Poniżej można zobaczyć jak wygląda nasz df z dodanymi nowymi kolumnami:

```
[8]: df
```

```
[9]:
```

	Nazwa	Wartość nominalu	Data wykupu	Oprocentowanie	kupon	\
0	ECH1024	1000	2024-10-23	0.04500	2	
1	MBHPE11	1000	2025-04-24	0.01285	1	
2	EUR0126	1000	2026-01-19	0.01500	1	
3	EUR1027	1000	2027-10-22	0.01375	1	
4	EUR1028	1000	2028-10-25	0.01000	1	

```

wartosc rynkowa  Data zakupu  Data ostatniego kuponu  liczba dni ↵↵
↵↵\

```

0	1.0000	2022-12-01	2022-10-23	39
1	1.0000	2022-12-01	2022-04-24	221
2	1.0300	2022-12-01	2022-01-19	316
3	0.9907	2022-12-01	2022-10-22	40
4	1.0650	2022-12-01	2022-10-25	37

	cena czysta	wartosc kuponu	cena brudna	czas zycia	
	↪liczba_kuponow				
0	1000.0	22.50	1002.650685	1.895890	↪
↪ 4					
1	1000.0	12.85	1007.921233	2.397260	↪
↪ 3					
2	1030.0	15.00	1043.150685	3.136986	↪
↪ 4					
3	990.7	13.75	992.357534	4.893151	↪
↪ 5					
4	1065.0	10.00	1066.123288	5.904110	↪
↪ 6					

2.1 efektywna stopa zwrotu

W tej części zostanie obliczona efektywna stopa zwrotu, z wykorzystaniem dwóch funkcji fx oraz bis. Funkcja bis działa rekurencyjnie i jej celem jest znalezienie takiego x dla funkcji fx , która będzie zwracać wartość 0. Zwrócona wartość przez funkcję bis jest naszą szukaną efektywną stopą zwrotu.

```
[10]: def fx(x):
    tn = tn1
    suma = 0
    for i in range(ilosc_kuponow):
        suma += wartosc_kuponu*pow(1+x,i)
    suma = (suma-P*pow(1+x,tn)+cena_nominalna)
    return suma
```



```
[11]: def bis(fx,a,b,err):
    while np.absolute(b-a)>err:
        midPoint=(a+b)*0.5
        if fx(midPoint)*fx(a)<0:
            b=midPoint
        midPoint=(a+b)*0.5
        if fx(midPoint)*fx(b)<0:
            a=midPoint
    return b-(b-a)*fx(b)/(fx(b)-fx(a))

    if metoda==2:#bisekcja
        i=1
        a=int(input("Podaj pierwszy kraniec przedziału: "))
        b=int(input("Podaj drugi kraniec przedziału: "))
        while i<n:
            err=0.0001
        print("x= ",bis(fx,a,b,err))
        i=i+1
```

W oknie poniżej znajduje się pętla, której celem jest znalezienie odpowiednich parametrów wejściowych dla naszej obligacji oraz obliczenie dla niej ERR. Jak widać za a i b przyjęto odpowiednio wartości -1 i 1. Liczba ujemna została dodana dlatego, że dla dwóch obligacji okazało się, że efektywna stopa procentowa jest ujemna.

```
[12]: r_list = []
for i in range(0, 5):
    ilosc_kuponow = df["liczba_kuponow"][i]
    P = df["cena brudna"][i] * 1.019
    wartosc_kuponu = df["wartosc kuponu"][i]
    cena_nominalna = df["Wartość nominału"][i]
    tn1 = df["czas zycia"][i]
    kupon = df["kupon"][i]
    r = bis(fx, -1, 1, 0.001)
    r_list.append(r)
```

Obliczona wartość R została dodana do naszej tabeli i po wyświetleniu możemy zobaczyć jej wartości.

```
[13]: df["R"] = r_list
df
```

```
[13]:      Nazwa  Wartość nominału Data wykupu Oprocentowanie  kupon  \
0  ECH1024             1000  2024-10-23         0.04500        2
1  MBHPE11             1000  2025-04-24         0.01285        1
2  EUR0126             1000  2026-01-19         0.01500        1
3  EUR1027             1000  2027-10-22         0.01375        1
4  EUR1028             1000  2028-10-25         0.01000        1

      wartosc rynkowa  Data zakupu Data ostatniego kuponu  liczba dni  \
0                  1.0000  2022-12-01          2022-10-23         39
1                  1.0000  2022-12-01          2022-04-24        221
2                  1.0300  2022-12-01          2022-01-19        316
3                  0.9907  2022-12-01          2022-10-22         40
4                  1.0650  2022-12-01          2022-10-25         37

      cena czysta  wartosc kuponu  cena brudna  czas zycia  \
0      1000.0          22.50  1002.650685    1.895890
1      1000.0          12.85  1007.921233    2.397260
2      1030.0          15.00  1043.150685    3.136986
3      990.7          13.75   992.357534    4.893151
4      1065.0          10.00  1066.123288    5.904110

      liczba kuponow  \
0      4
1      3
2      4
3      5
4      6

      liczba kuponow      R
0      4  0.037302
1      3  0.004720
2      4 -0.000917
3      5  0.011688
4      6 -0.004256
```

2.2 Struktura terminowa

W kolejnym kroku w oparciu o przedstawiony na wstępie wzór zostanie obliczona struktura terminowa dla wszystkich obligacji. Poniżej znajduje się kod, który

```
[14]: B1 = (1 + df["R"][0])**(-df["czas zycia"][0])
      B2 = (1 + df["R"][1])**(-df["czas zycia"][1])
      B3 = (1 + df["R"][2])**(-df["czas zycia"][2])
      B4 = (1 + df["R"][3])**(-df["czas zycia"][3])
      B5 = (1 + df["R"][4])**(-df["czas zycia"][4])

      B = list([1, B1, B2, B3, B4, B5])
      t = [0, df["czas zycia"][0], df["czas zycia"][1], df["czas zycia"][2], df["czas zycia"][3], df["czas zycia"][4]]

      df_B = pd.DataFrame(list(zip(t, B)), columns=['t', 'B'])
      df_B
```

```
[14]:
```

	t	B
0	0.000000	1.000000
1	1.895890	0.932922
2	2.397260	0.988774
3	3.136986	1.002881
4	4.893151	0.944727
5	5.904110	1.025504

2.3 Przedłużenie I metoda

W tym porożdziale zostanie zaprezentowany sposób przedłużania struktury terminowej dla naszych obligacji. Jest to inaczej interpolacja pomiędzy naszymi wcześniej wyliczonymi punktami. Poniżej znajduje się funkcja, która dla danych wejściowych oblicza dla nas zinterpolowane punkty.

```
[15]: def przedluzenie1(t, t1, t2, b1, b2):
      return b1*((t2 - t)/(t2 - t1))*b2*((t - t1)/(t2 - t1))
```

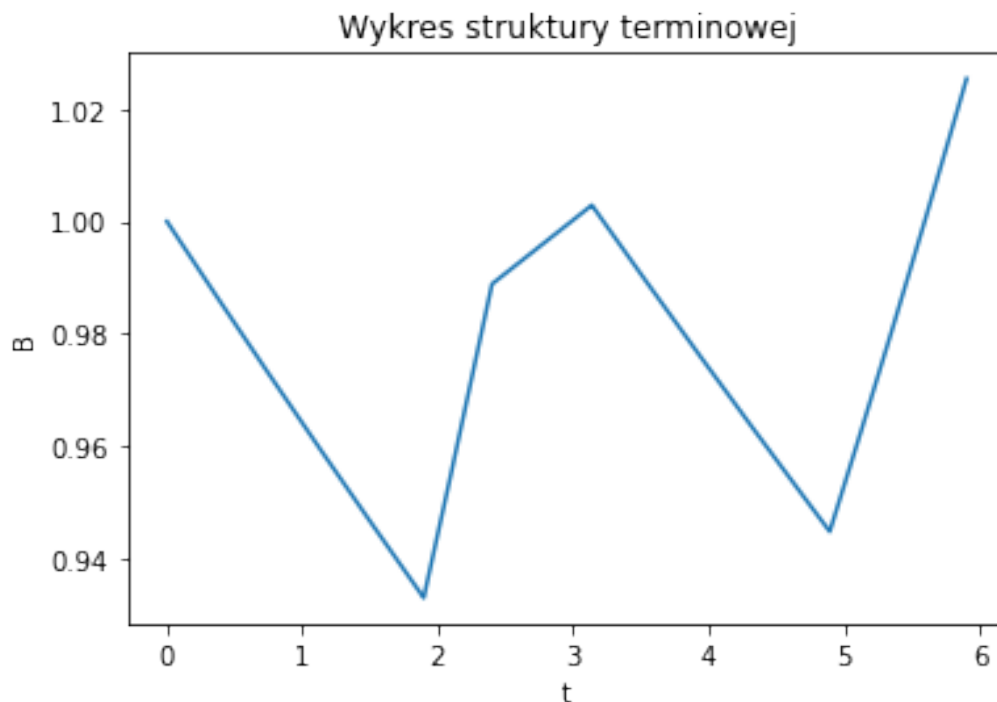
```
[16]: df_B1 = pd.DataFrame(columns=['t', 'B'])
```

W pętli tej dla każdej pary np. (B0, B1) (B2,B3) zostaną obliczone wartości pomiędzy tymi punktami.

```
[17]: for i in range(1, 6):
        punkty_t = np.arange(df_B['t'][i-1], df_B['t'][i], 0.001)
        B_punkty = przedluzenie1(punkty_t, df_B['t'][i-1],
        ↪df_B['t'][i], df_B['B'][i-1], df_B['B'][i])
        dane = pd.DataFrame(list(zip(punkty_t, B_punkty)), columns=
        ↪['t', 'B'])
        df_B1 = df_B1.append(dane)
```

Poniżej można także zauważyć jak na wykresie wyglądają obliczone punkty. Wiadać, że krzywa ta jest łamana i w końcowej fazie

```
[18]: fig, ax = plt.subplots(figsize =(20, 10))
        plt.plot(df_B1['t'], df_B1['B'])
        plt.xlabel('t')
        plt.ylabel('B')
        plt.title('Wykres struktury terminowej')
        plt.show()
```



2.4 Przedłużenie II metoda

W drugiej metodzie wykorzystujemy drugi inny wzór, dlatego też obliczenia będziemy opierać na funkcji przedłużenie2.

```
[19]: def przedluzenie2(t, t1, t2, b1, b2):  
       return b1 + (b2 - b1) * ((t - t1) / (t2 - t1))
```

```
[20]: df_B2 = pd.DataFrame(columns=['t', 'B'])
```

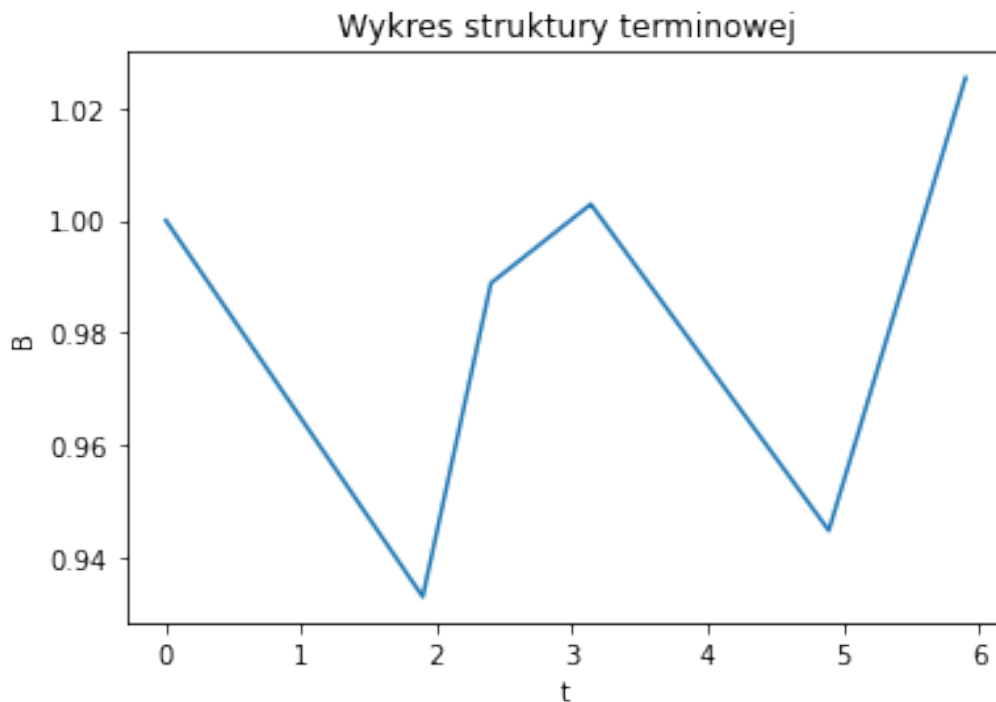
Tak jak w poprzednim przykładzie również tutaj całość działa na pętli, która oblicza miesiące pomiędzy wcześniej wyznaczonymi punktami.

```
[21]: for i in range(1, 6):  
       punkty_t = np.arange(df_B['t'][i-1], df_B['t'][i], 0.001)
```

```
B_punkty = przedluzenie2(punkty_t, df_B['t'][i-1],  
↳df_B['t'][i], df_B['B'][i-1], df_B['B'][i])  
dane = pd.DataFrame(list(zip(punkty_t, B_punkty)), columns=  
↳['t', 'B'])  
df_B2 = df_B2.append(dane)
```

Jak widać wykres ten jest identyczny jak w przypadku metody pierwszej.

```
[22]: fig, ax = plt.subplots(figsize =(20, 10))  
plt.plot(df_B2['t'], df_B2['B'])  
plt.xlabel('t')  
plt.ylabel('B')  
plt.title('Wykres struktury terminowej')  
plt.show()
```



Rozdział 3

Wyniki i wnioski

Celem projektu była analiza 5 wybranych obligacji pod kątem ich efektywnej stopy zwrotu oraz struktury terminowej. Analiza ta została przeprowadzona na podstawie danych dotyczących długości trwania obligacji oraz wysokości wypłacanych odsetek. W wyniku przeprowadzonych badań, uzyskano dwa wyniki dla efektywnej stopy zwrotu (r_i) oraz struktury terminowej ($B(i)$). Dodatkowo, przygotowano dwa wykresy, które przedstawiały przedłużoną strukturę terminową obliczoną dwoma różnymi metodami. Przeprowadzona analiza pozwoliła na lepsze zrozumienie ryzyka związanego z inwestowaniem w obligacje oraz na wybór najlepszej opcji inwestycyjnej.

Rozdział 4

Źródła:

<https://gpwcatalyst.pl/o-instrumentach-instrument?nazwa=EUR1028>
<https://gpwcatalyst.pl/o-instrumentach-instrument?nazwa=MBHPE11>
<https://gpwcatalyst.pl/o-instrumentach-instrument?nazwa=ECH1024>
<https://gpwcatalyst.pl/o-instrumentach-instrument?nazwa=EUR0126>
<https://gpwcatalyst.pl/o-instrumentach-instrument?nazwa=EUR1027>