

Modelowanie i symulacja w programie MATLAB cz.2

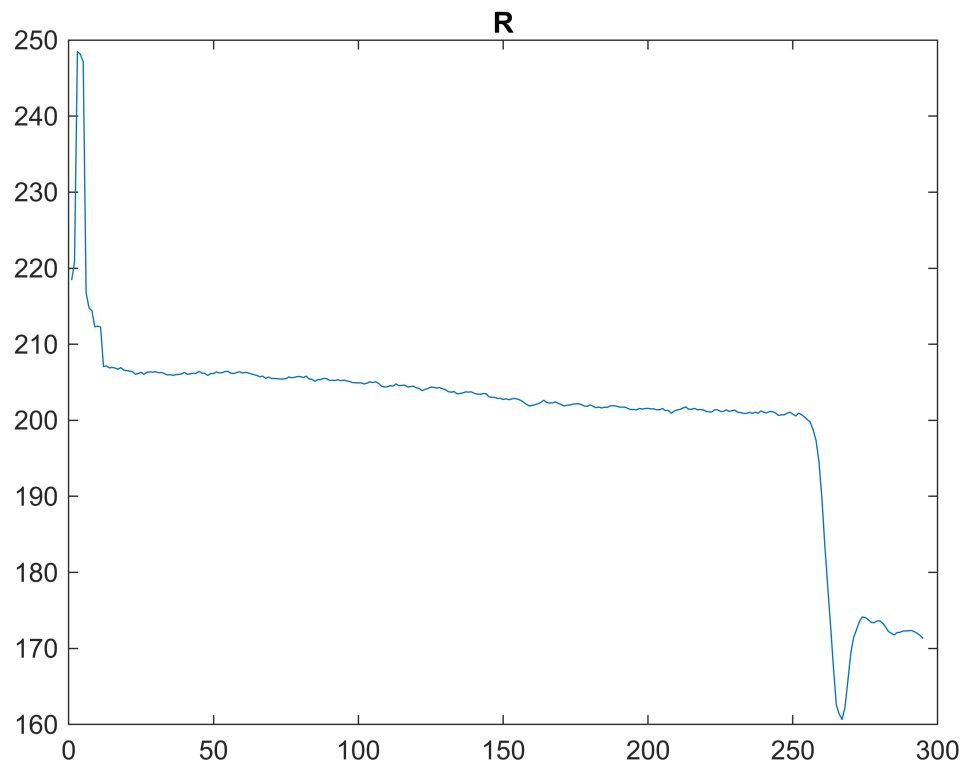
Jan Rosa 410269 AiR

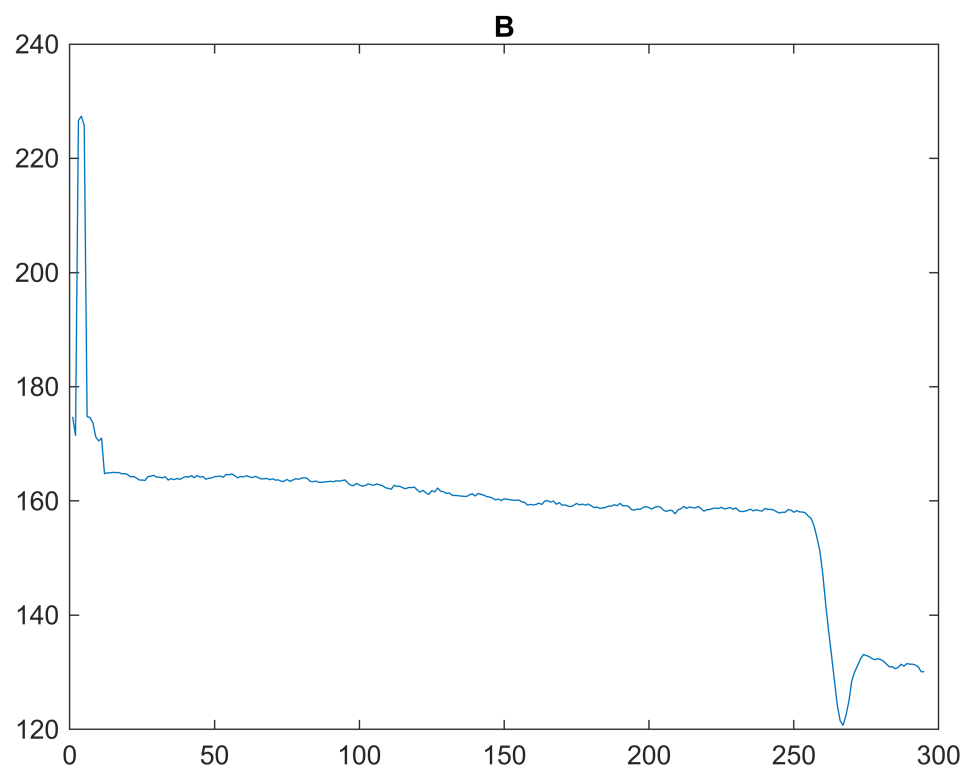
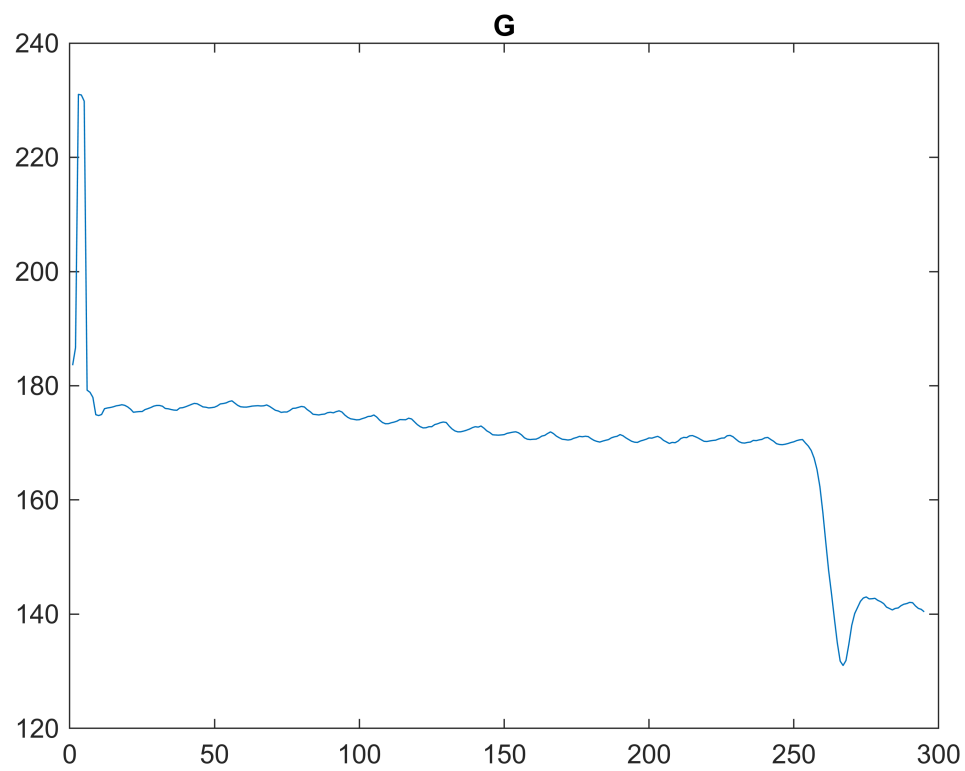
Zad 1

Zad a

Utwórz m-skrypt wczytujący dane z pliku daneP.csv a następnie realizującego wizualizację danych na wykresie typu plot (każda zmienna na osobnym wykresie). Dodaj do każdego wykresu osobny tytuł. - Wskazówka: użyj import wizarda do importu danych, następnie wygeneruj m-funkcję do importu danych i wykorzystaj ją w swoim m-skrypcie

```
import1
```





Kod m-skryptu:

```
daneP = readmatrix('daneP.csv');
```

```
R = daneP(:, 1);
```

```
G = daneP(:, 2);
```

```
B = daneP(:, 3);
```

```
figure
```

```
subplot(1, 3, 1)
```

```
plot(R)
```

```
title("R")
```

```
subplot(1, 3, 2)
```

```
plot(G)
```

```
title("G")
```

```
subplot(1, 3, 3)
```

```
plot(B)
```

```
title("B")
```

Zad b

Z zaimportowanych danych wybierz taki fragment, na którym widoczny jest sygnał okresowy. Utwórz nową zmienną zawierającą wybrany fragment sygnału i zwizualizuj ją na osobnym oknie wykresu. - Wskazówka: Przyglądnij się wykresowi każdej składowej i wybierz taką, na której okresowość jest najlepiej widoczna. Następnie, przy pomocy indeksowania, wybierz fragment pomijając zakłócenia na początku i na końcu sygnału.

```
okres = daneP(100:230, :);
```

```
okres = 131×3
```

```
204.9200 174.0600 162.8000  
204.9200 174.2500 162.6100  
204.7800 174.4000 162.6900  
204.8800 174.6200 162.9900  
205.0600 174.6600 162.8400  
204.9700 174.8800 162.7600  
205.0600 174.5200 162.9800  
204.8600 173.9700 162.8400  
204.4700 173.5600 162.6800  
204.3800 173.3500 162.3200  
⋮
```

```
m = 130;
```

```
m = 130
```

```
%fq = (0:m/2 + (m/2-1))*900
```

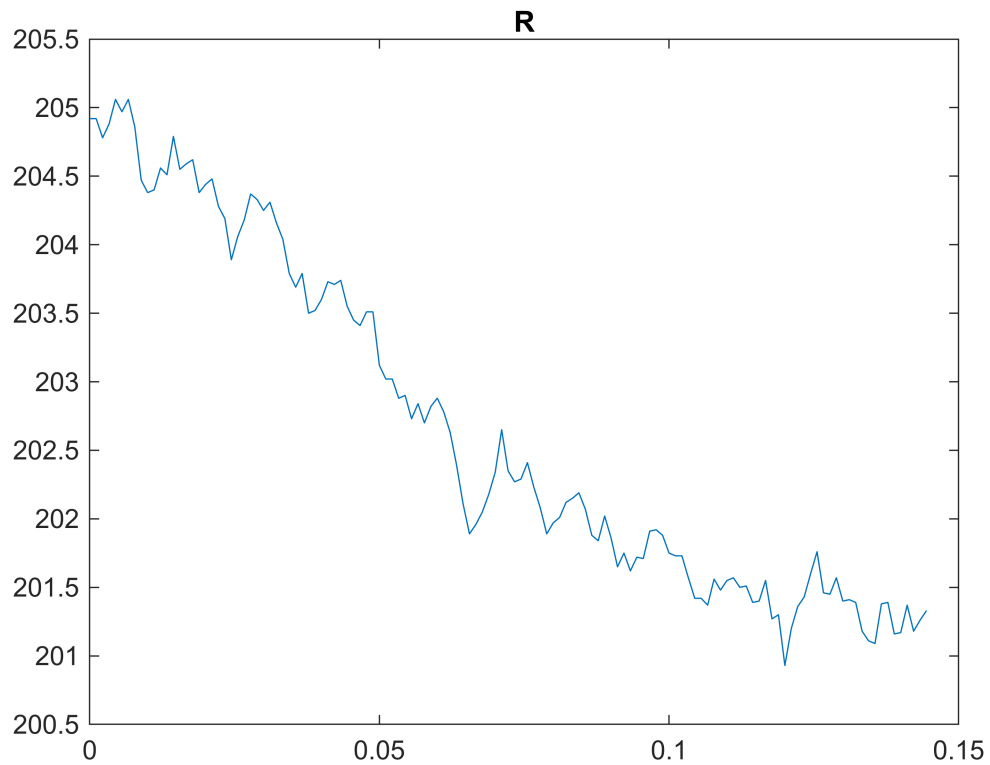
```
czas_pr = 1/900;
```

```
czas_pr = 0.0011
```

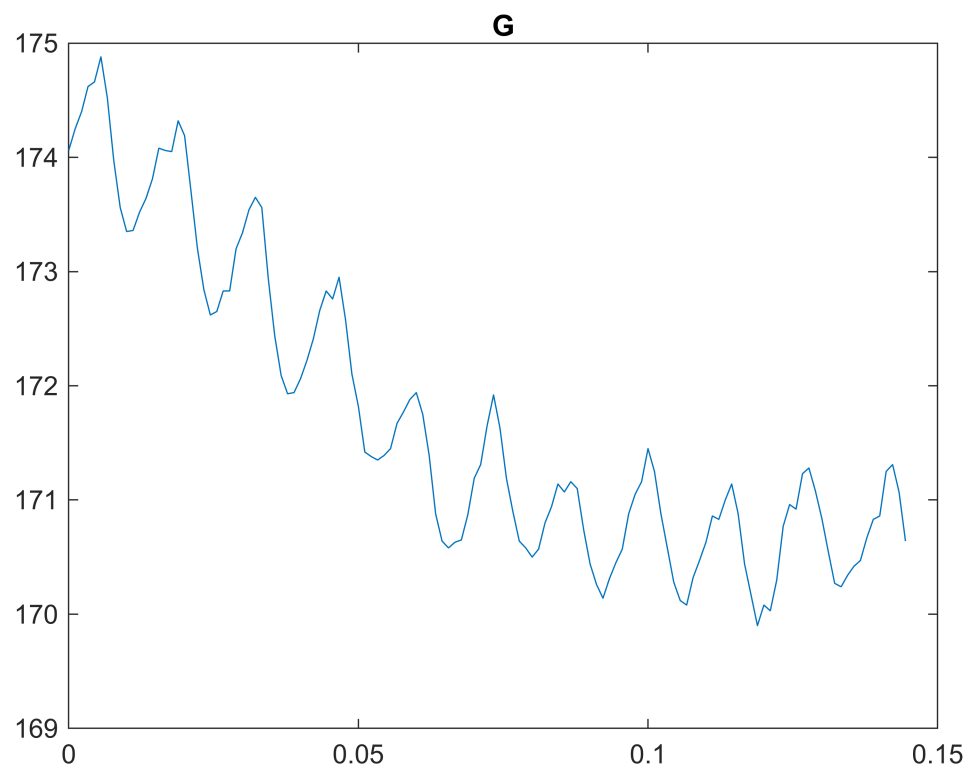
```
time = 0:czas_pr:130*czas_pr;
```

```
time = 1×131  
0 0.0011 0.0022 0.0033 0.0044 0.0056 0.0067 0.0078 ...
```

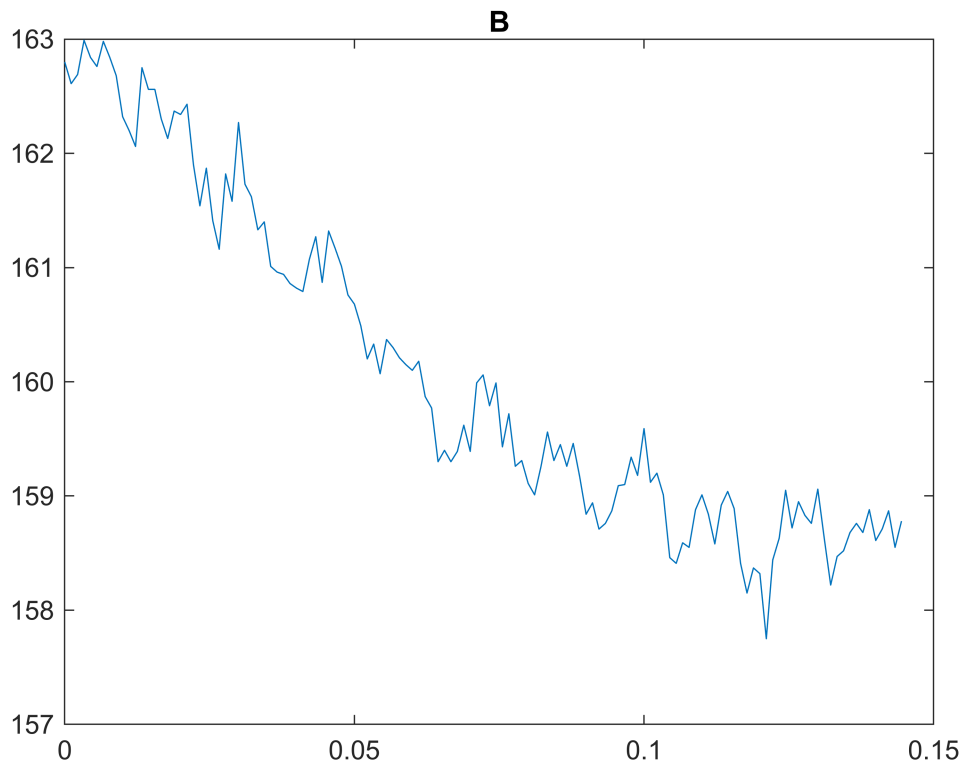
```
figure("Name", "R")  
plot(time, okres(:, 1))  
title("R")
```



```
figure("Name", "G")  
plot(time, okres(:, 2))  
title("G")
```



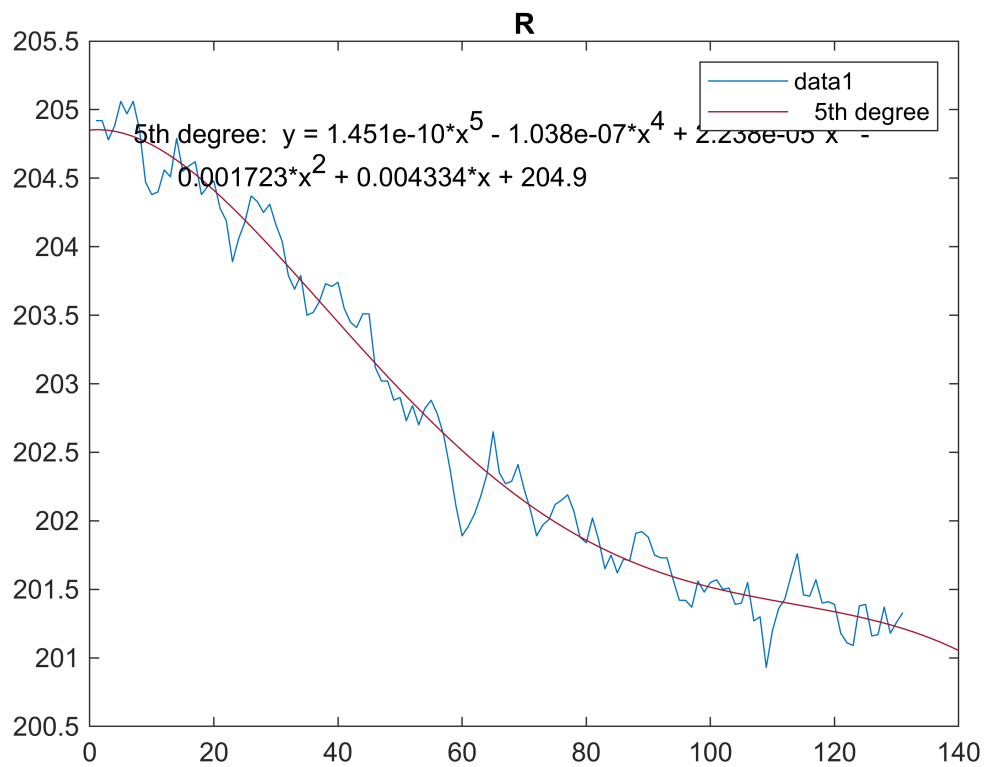
```
figure("Name","B")  
plot(time, okres(:, 3))  
title("B")
```



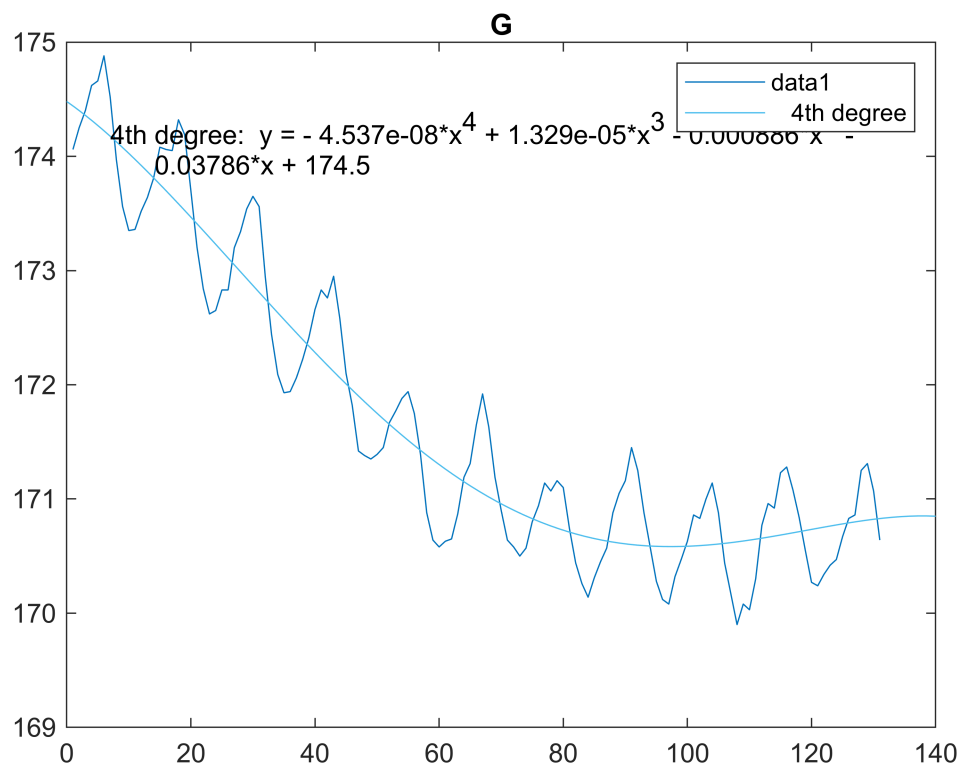
Zad c

Z wybranego fragmentu sygnału, usuń trend poprzez dopasowanie krzywej wielomianowej. Dobierz stopień wielomianu jak najmniejszego stopnia przy zachowaniu jak najmniejszego błędu dopasowania (funkcja norm). Do wykresu z poprzedniego punktu dodaj linię dopasowanego trendu (inny kolor). Utwórz nowy wykres zawierający sygnał z usuniętym trendem. - Wskazówka – możesz zrealizować dopasowanie przy pomocy narzędzia Basic Fitting Tool, a następnie wygenerować m-kod i wybrać z niego potrzebne fragmenty analizy sygnału

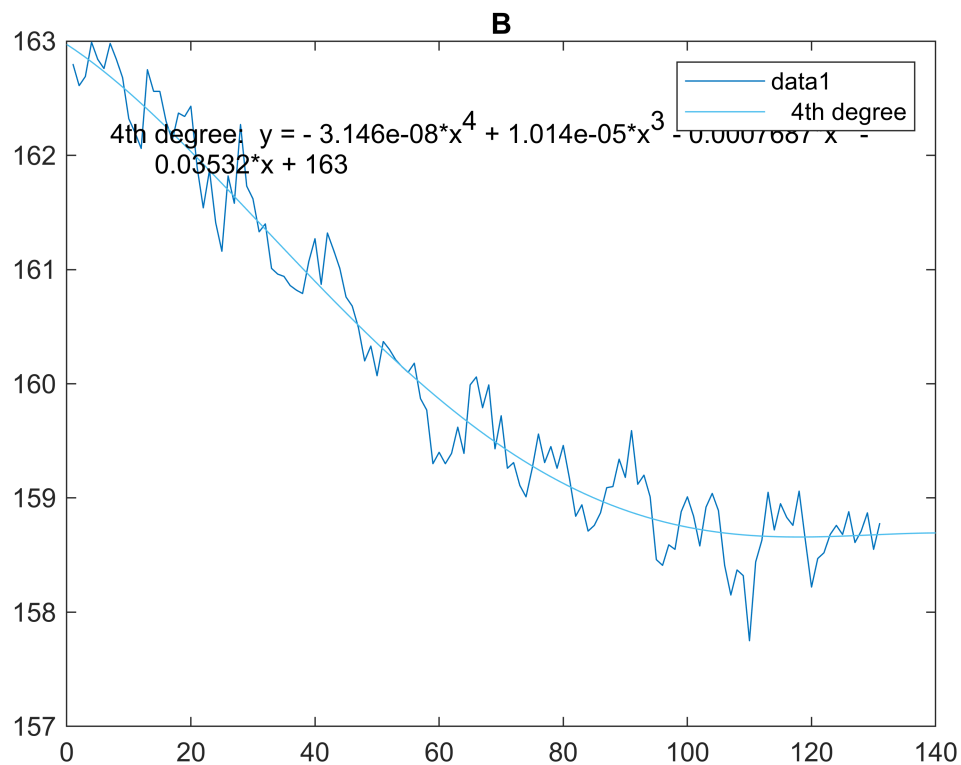
```
figure  
R_createfigure(okres(:, 1));
```



```
figure
G_createfigure(okres(:, 2));
```



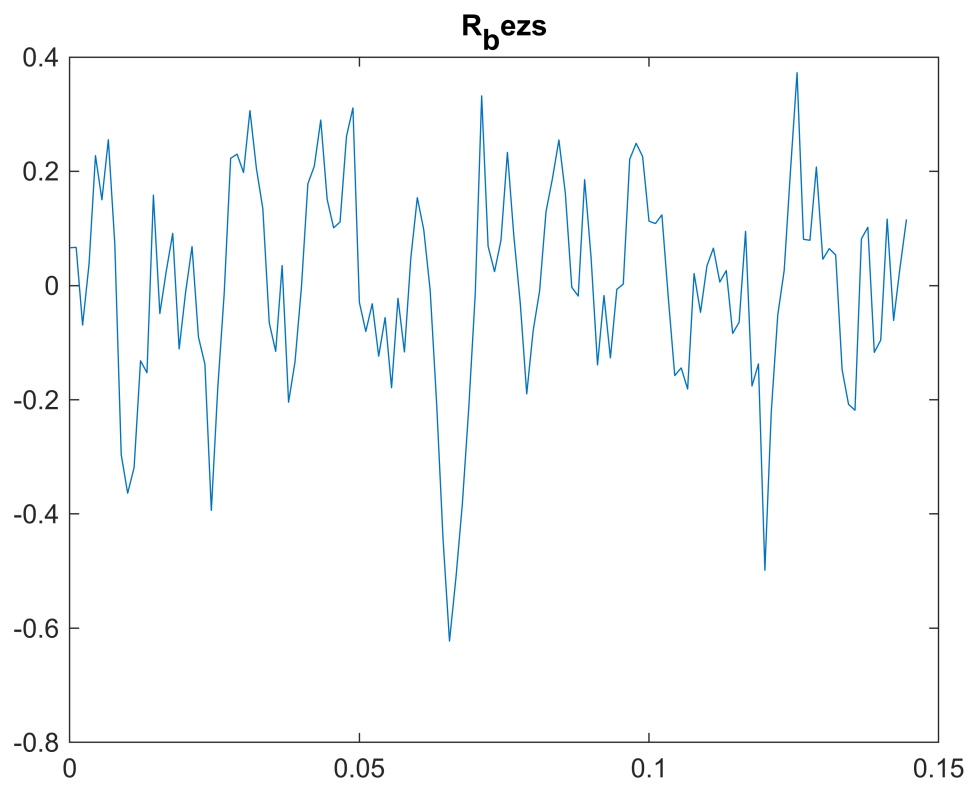
```
figure
B_createfigure(okres(:, 3));
```



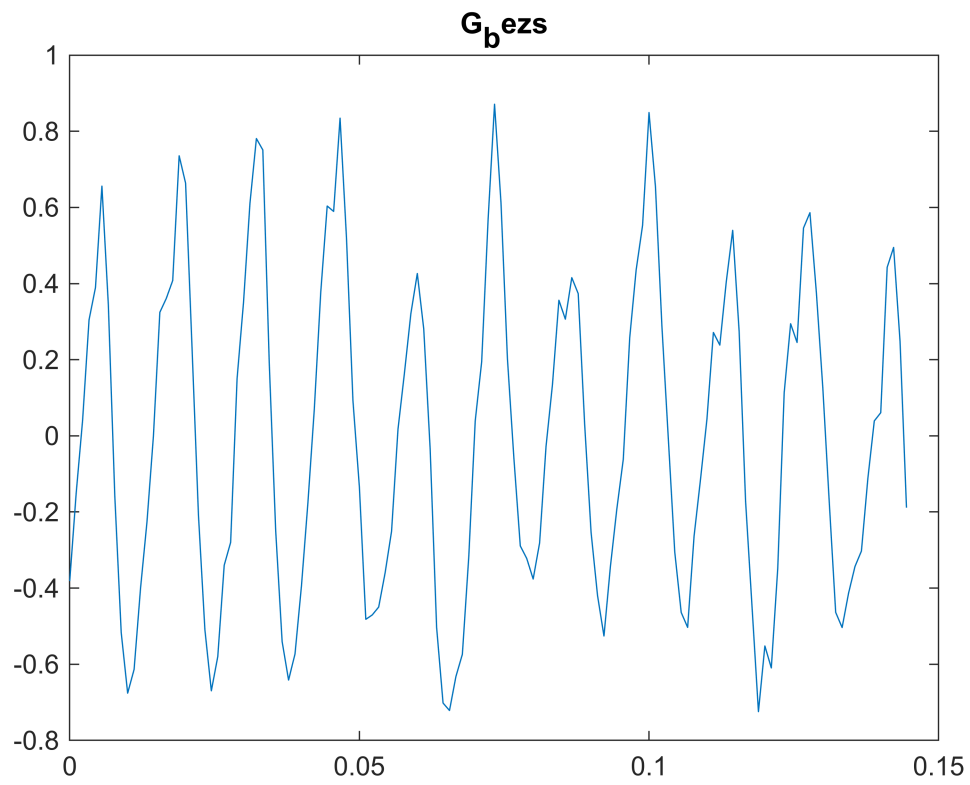
```
R_szum = polyval(polyfit(time, okres(:, 1), 5), time)';
G_szum = polyval(polyfit(time, okres(:, 2), 4), time)';
B_szum = polyval(polyfit(time, okres(:, 3), 2), time)';
```

```
%figure("Name","R poly")
%plot(R_szum)
%title("R poly")
%figure("Name","G poly")
%plot(G_szum)
%title("G poly")
%figure("Name","B poly")
%plot(B_szum)
%title("B poly")
R_bezs = okres(:,1) - R_szum;
G_bezs = okres(:,2) - G_szum;
B_bezs = okres(:,3) - B_szum;
```

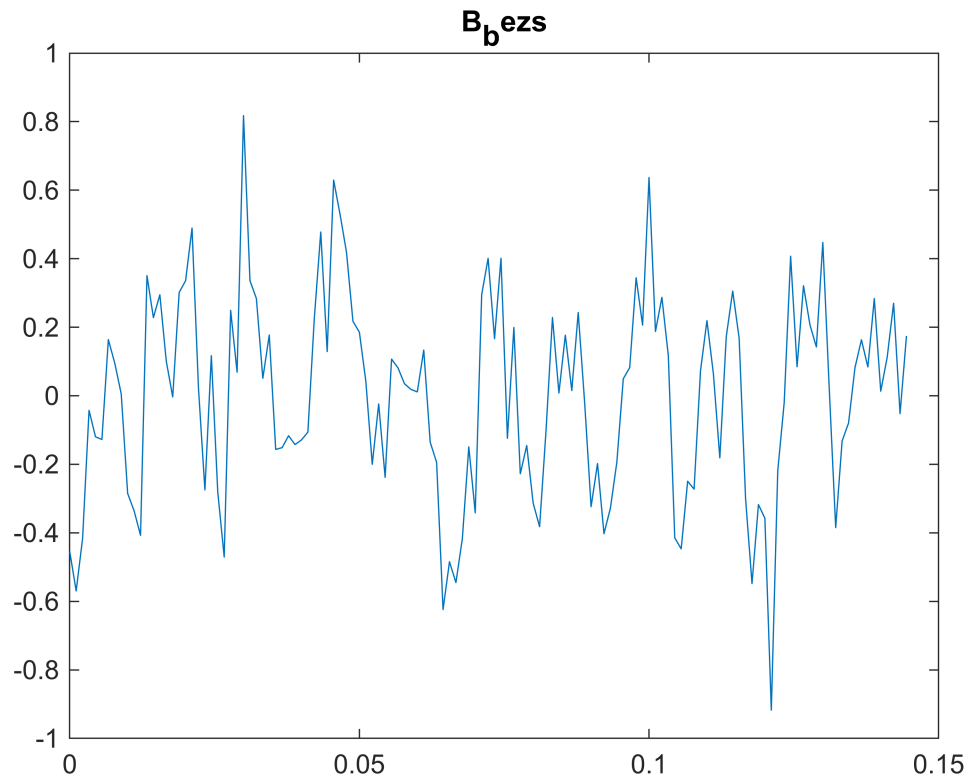
```
figure
plot(time, R_bezs)
title("R_bezs")
```

```
figure
plot(time, G_bezs)
title("G_bezs")
```



```
figure
plot(time, B_bezs)
title("B_bezs")
```



Zad d

Znajdź częstotliwość charakterystyczną sygnału. Sformatuj wykres dodając do niego w sposób programowy: opisy osi x,y i tytuł wykresu. Dodaj do wykresu punkt w miejscu maksimum (częstotliwość charakterystyczna) oraz opis informujący o wartości częstotliwości tego maksimum. - Wskazówka: użyj funkcji `fft` (patrz przykład w dokumentacji) z częstotliwością próbkowania odczytaną z pliku `daneP.csv` - Aby dodać do wykresu opis wykorzystaj funkcję `text` oraz sformatuj tekst przy pomocy funkcji `sprintf`.

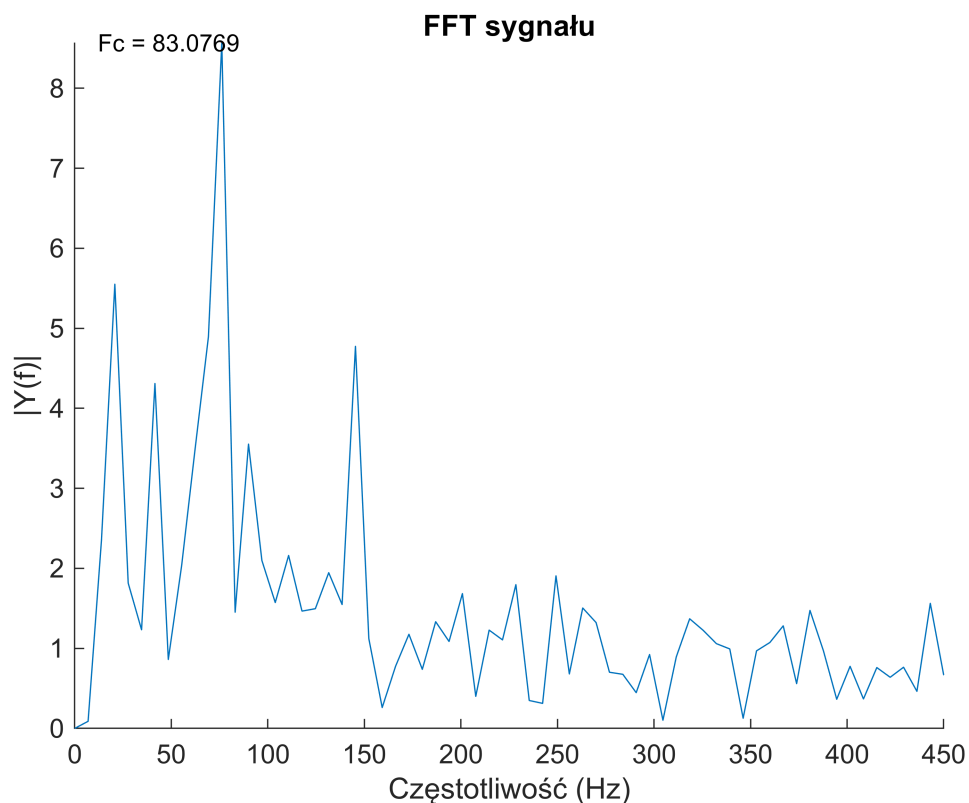
```
f_n = 900;
R_fft = abs(fft(R_bezs));
G_fft = abs(fft(G_bezs));
B_fft = abs(fft(B_bezs));
m = length(R_bezs);
fq = (0 : m/2 + (m/2 - 1)) * f_n/(m-1);

figure
hold on
plot(abs(fq), (R_fft))
title("FFT sygnału R")
xlabel("Częstotliwość (Hz)")
ylabel("|Y(f)|")
```

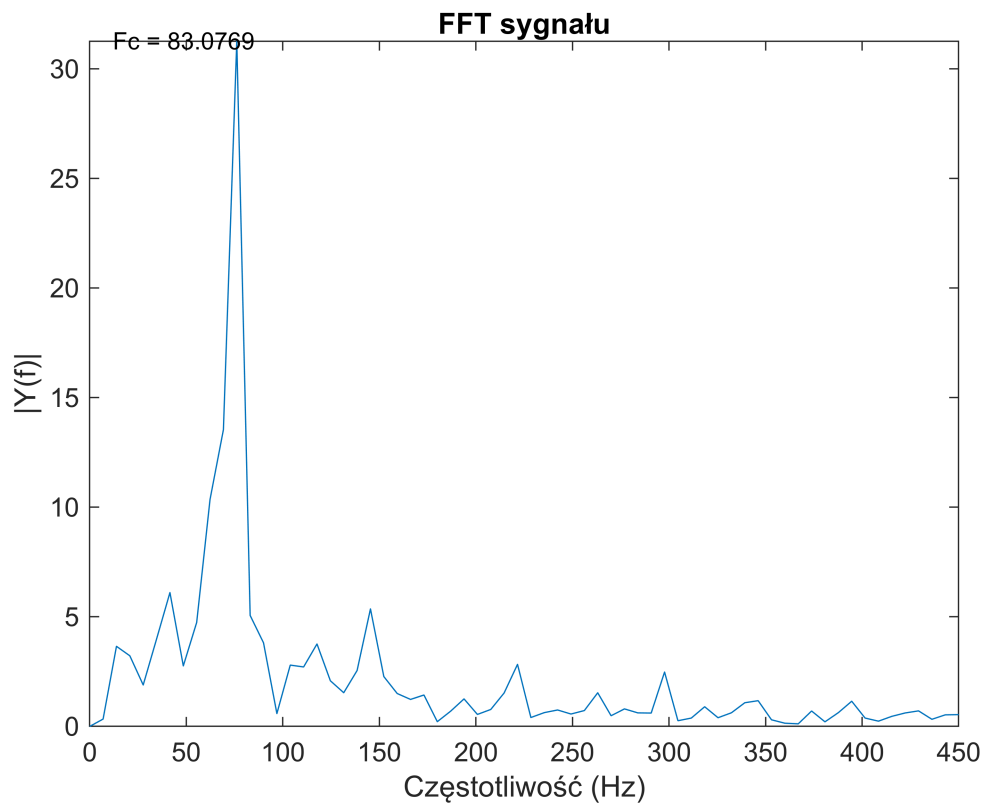
```
[R_max, index_r] = max(R_fft);
txt = "Fc = " + string(index_r * f_n/(m-1));
```

```
txt =
"Fc = 83.0769"
```

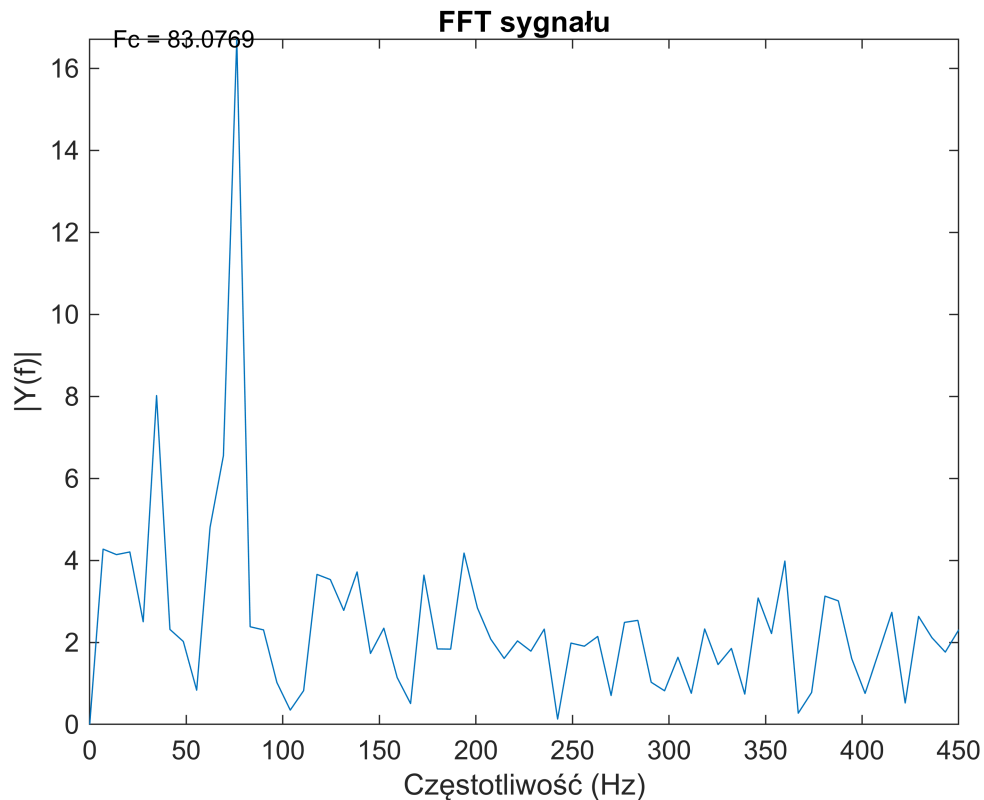
```
text(index_r, R_max, txt, fontsize = 9);
axis([0 450 0 inf]);
```



```
figure
plot(abs(fq), (G_fft))
title("FFT sygnału G")
xlabel("Częstotliwość (Hz)")
ylabel("|Y(f)|")
[G_max, index_g] = max(G_fft);
txt = "Fc = " + string(index_g * f_n/(m-1));
text(index_g, G_max, txt, fontsize = 9);
axis([0 450 0 inf])
```



```
figure
plot(abs(fq), B_fft)
title("FFT sygnału B")
xlabel("Częstotliwość (Hz)")
ylabel(" $|Y(f)|$ ")
[B_max, index_b] = max(B_fft);
txt = "Fc = " + string(index_b * f_n/(m-1));
text(index_b, B_max, txt, fontsize = 9);
axis([0 450 0 inf]);
```



Zad e

Uzupełnij m-plik o komentarze i dokumentację. Wygeneruj raport w formacie HTML

```
%przykładowy komentarz
%dokumentacja znajduje się w polach tekstowych LiveScript
%%Raport wygenerowany przez opcje w menu: Safe as
```

Zad 2

Zad a

Przy pomocy polecenia `randn` wygeneruj tablicę 3x3 liczb pseudolosowych `R` o rozkładzie normalnym (średnia 0 i odchylenie standardowe). Następnie utwórz zmienną `A` jako typ `UINT32`, zawierającą liczbę 100. Pomnóż zmienną `R` przez `A`, odpowiednio dostosowując typy danych. Rezultat (zmienna `B`) powinna być typu `UINT32`. Zwróć uwagę czy rezultaty mnożenia są poprawne ! W sprawozdaniu zanotuj liczbę bajtów potrzebną do zapamiętania jednej liczby typu `double` oraz jednej liczby typu `UINT32` (wskazówka – skorzystaj z polecenie `whos`).

```
R = normrnd(0, 1, 3, 3)
```

```
R = 3x3
    0.0859   -1.0616    0.7481
   -1.4916    2.3505   -0.1924
   -0.7423   -0.6156    0.8886
```

```
A = uint32(100)
```

```
A = uint32
    100
```

```
B = uint32(R * double(A))
```

```
B = 3x3 uint32 matrix
     9      0     75
     0    235      0
     0      0     89
```

Jeżeli przed przekształceniem na uint32 liczba była dodatnia to wynik jest zaokrągleniem liczby, jeśli była ujemna to wynik jest równy 0

double zawiera 8 bajtów

uint32 zawiera 4 bajty

Zad b

Utwórz dwie tablice znakowe zawierające teksty: „ćwiczenie 2” oraz „laboratorium 1”. Połącz te dwie tablice (wskazówka – skorzystaj z polecenia strvcats)

```
cw = 'ćwiczenie 2'
```

```
cw =
'ćwiczenie 2'
```

```
lab = 'laboratorium 1'
```

```
lab =
'laboratorium 1'
```

```
joined = strvcats(cw, lab)
```

```
joined = 2x14 char array
'ćwiczenie 2   '
'laboratorium 1'
```

Zad c

Utwórz tablicę znakową str1 zawierającą tekst „Krasnoludy przeszły przez rzekę w bród, nie zamoczywszy swych bród i do tego zmywszy ze swych nóg brud”. Znajdź indeksy słów zaczynających się na literę „b”,

kończących na literę „d” i nie zawierających litery „u”. Wskazówka – skorzystaj z wyrażeń regularnych – dokumentacja do polecenia regexp

```
str1 = 'Krasnoludy przeszły przez rzekę w bród, nie zamoczywszy swych bród i do tego zmywszy ze swych nóg brud'
```

```
str1 =  
'Krasnoludy przeszły przez rzekę w bród, nie zamoczywszy swych bród i do tego zmywszy ze swych nóg brud'
```

```
szuk = 'b+[^u]+d';
```

```
szuk =  
'b+[^u]+d'
```

```
[a, b, c] = regexp(str1, szuk, "start")
```

```
a = 35  
b = 70  
c = 1x1 cell array  
    {0x0 double}
```

Zad d

Utwórz tablicę komórkową o rozmiarze 2x2 zawierającą następujące dane jak na rysunku poniżej. Wybierz z tablicy komórkowej, tablicę liczb pseudolosowych znajdującą się w komórce 2-wiersz, 1- kolumna, dodaj do niej wartość 100, a rezultat zapisz w to samo miejsce do tablicy komórkowej.

```
incel = rand(3,3);  
d2 = {123, "abcd"; incel, 0.1}
```

```
d2 = 2x2 cell
```

	1	2
1	123	abcd
2	[0.6892,0.0838,0.1524;0.7482,0.2290,0.8258;0.4505,...	0.1

```
d2{2, 1} = d2{2, 1} + 100
```

```
d2 = 2x2 cell
```

	1	2
1	123	abcd
2	[100.6892,100.0838,100.1524;100.7482,100.2290,100.8258;1...	0.1

Zad e

Oblicz całkę oznaczoną w przedziale $\in (-2,2)$ z funkcji $f(x) = 2 - 2 \cdot x + 4$ i narysuj jej wykres dla tego przedziału. Wskazówka – zdefiniuj funkcję przy pomocy uchwytu do funkcji, wykorzystaj funkcję quad oraz fplot.

```
f = @(x) x.^2 - 2.*x + 4
```



```
f = function_handle with value:  
@(x)x.^2-2.*x+4
```

```
x = -2:0.1:2
```

```
x = 1×41  
-2.0000 -1.9000 -1.8000 -1.7000 -1.6000 -1.5000 -1.4000 -1.3000 ...
```

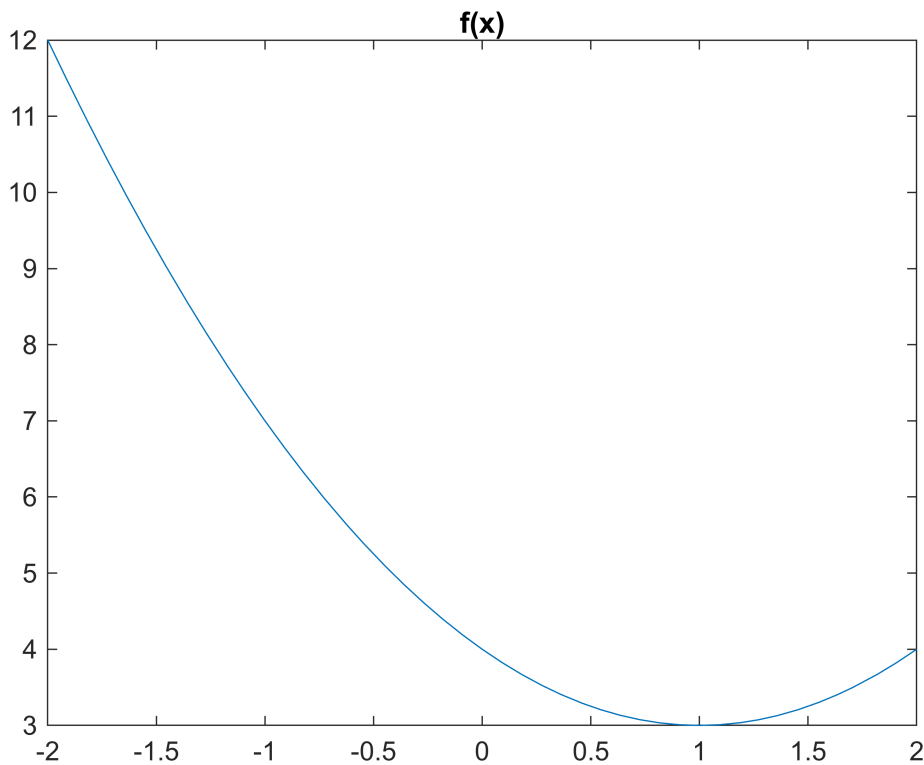
```
integral(f, -2, 2)
```

```
ans = 21.3333
```

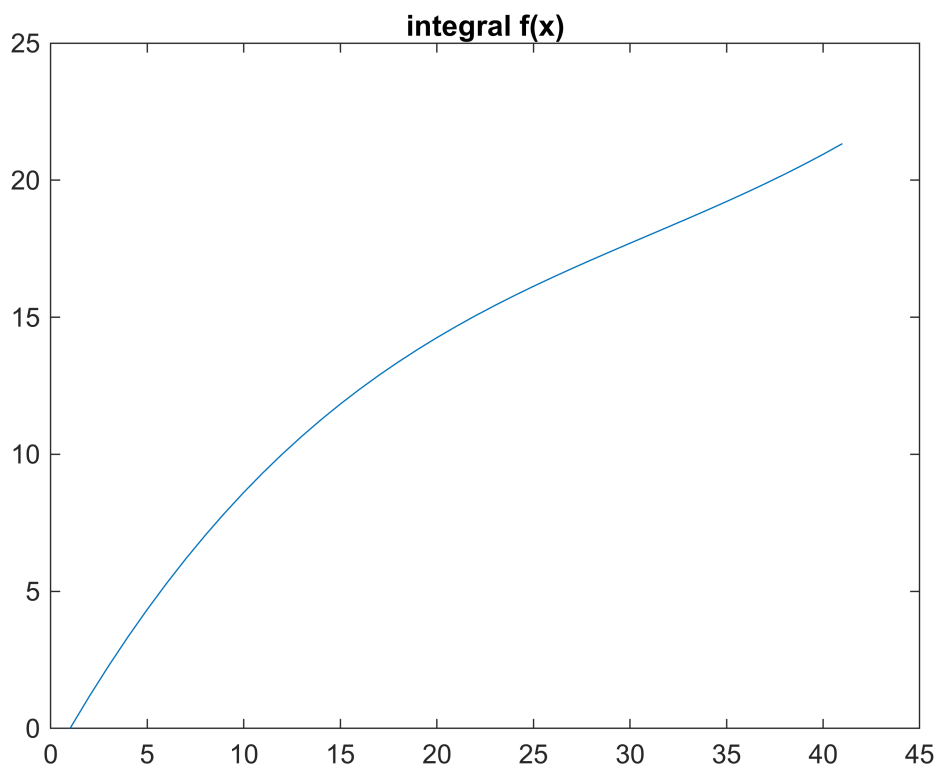
```
y = x
```

```
y = 1×41  
-2.0000 -1.9000 -1.8000 -1.7000 -1.6000 -1.5000 -1.4000 -1.3000 ...
```

```
for i = 1:41  
    y(i) = integral(f, -2, x(i));  
end  
figure  
fplot(f, [-2, 2])  
title("f(x)")
```



```
plot(y)  
title("integral f(x)")
```



Zad f

Utwórz typ danych tabelaryczny (table) zawierający dane jak na rysunku poniżej. Wyeksportuj dane z tabeli do pliku CSV. Wskazówka – sprawdź w systemie pomocy jak definiować nazwy kolumn oraz wierszy. Liczby do tabeli wygeneruj losowo. Skorzystaj z polecenia writetable

```
Matematyka = randi(100,5,1);
Fizyka = randi(100,5,1);
Chemia = randi(100,5,1);
tablica = table(Matematyka, Fizyka, Chemia);
tablica.Row = ["Rafał", "Monika", "Paweł", "Elżbieta", "Mirek"]
```

tablica = 5×3 table

	Matematyka	Fizyka	Chemia
1 Rafał	92	57	94
2 Monika	29	8	13
3 Paweł	76	6	57
4 Elżbieta	76	54	47
5 Mirek	39	78	2

```
writetable(tablica, "tablica.csv")
```