Paweł Wojarnik (276027)

Szeregi Fouriera

Analiza Sygnałów

1 Cel raportu

Celem raportu jest analityczne przedstawienie rozwinięcia w szereg Fouriera funkcji sygnału prostokątnego, trójkątnego, piłokształtnego, przedstawienie rozwiązania w programie Scilab oraz na wykresach.

2 Obliczanie szeregów Fouriera

Szeregiem Fouriera funkcji f nazywamy szereg funkcyjny postaci:

$$\mathrm{S}(x)=\frac{a_0}{2}+\sum_{n=1}^{\infty}\left(a_n\cos\left(\frac{2n\pi}{T}x\right)+b_n\sin\left(\frac{2n\pi}{T}x\right)\right)$$

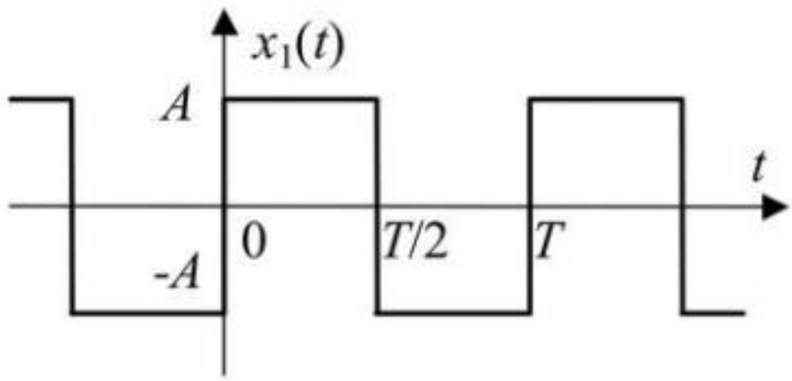
gdzie:

$$a_n=\frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}}f(x)\cos\left(\frac{2n\pi}{T}x\right)\mathrm{dx},\quad n=0,1,2,\dots$$

$$b_n=\frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}}f(x)\sin\left(\frac{2n\pi}{T}x\right)dx,\quad n=1,2,3,\dots$$

a T oznacza okres funkcji f.

3 Sygnał prostokątny



Sygnał prostokątny o okresie T i Amplitudzie A możemy zapisać jako:

$$f(t) = \begin{cases} A & \text{dla } 0 < t < \frac{T}{2} \\ -A & \text{dla } \frac{T}{2} < t < T \\ 0 & \text{dla } t = 0 \ i \ t = \frac{T}{2} \end{cases}$$

$$a_0 = \frac{2}{T} \cdot \left(\int_{-\frac{T}{2}}^{0} (-A)dt + \int_{0}^{\frac{T}{2}} Adt \right) = 0$$

$$a_n = \frac{2}{T} \cdot \left(\int_{-\frac{T}{2}}^{0} (-A) \cdot \cos\left(\frac{2\pi n}{T} t\right)dt + \int_{0}^{\frac{T}{2}} A \cdot \cos\left(\frac{2\pi n}{T} t\right)dt \right) = 0$$

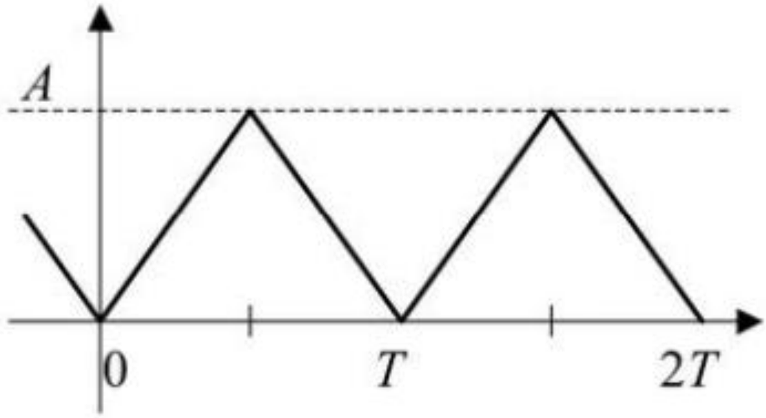
$$b_n = \frac{2}{T} \cdot \left(\int_{-\frac{T}{2}}^{0} (-A) \cdot \sin\left(\frac{2\pi n}{T} t\right) dt + \int_{0}^{\frac{T}{2}} A \cdot \sin\left(\frac{2\pi n}{T} t\right) dt \right)$$

$$= \frac{2A}{\pi n}(1 - \cos(n\pi)) = \begin{cases} 0 & \text{dla n parzystych} \\ \frac{4A}{\pi n} & \text{dla n nieparzystych} \end{cases}$$

Współczynniki $a_0 \ i \ a_n$ są równe 0, a b_n zeruje się dla parzystych n. Dlatego możemy zamienić n na 2k-1. Ostatecznie otrzymujemy szereg Fouriera:

$$f(x) = \sum_{k=1}^{\infty} \left(\frac{4A}{\pi(2k-1)} sin\left(\frac{(2k-1)2\pi}{T} t\right)\right)$$

4 Sygnał trójkątny



Sygnał trójkątny o okresie T i Amplitudzie A możemy zapisać jako:

$$f(t) = \begin{cases} -\frac{2A}{T} \cdot t + 2A & \text{dla } -\frac{T}{2} < t < 0 \\ \frac{2A}{T} \cdot t & \text{dla } 0 < t < \frac{T}{2} \end{cases}$$

$$a_0 = \frac{2}{T} \cdot \left(\int_{-\frac{T}{2}}^{0} \left(\frac{2A}{T} \cdot t + 2A \right) dt + \int_{0}^{\frac{T}{2}} \frac{2A}{T} \cdot t \cdot dt \right) = A$$

$$a_n = \frac{2}{T} \cdot \left(\int_{-\frac{T}{2}}^{0} (-A) \cdot \cos\left(\frac{2\pi n}{T} t \right) dt + \int_{0}^{\frac{T}{2}} A \cdot \cos\left(\frac{2\pi n}{T} t \right) dt \right)$$

$$= -\frac{2A}{\pi^2 n^2} (1 - \cos(\pi n))$$

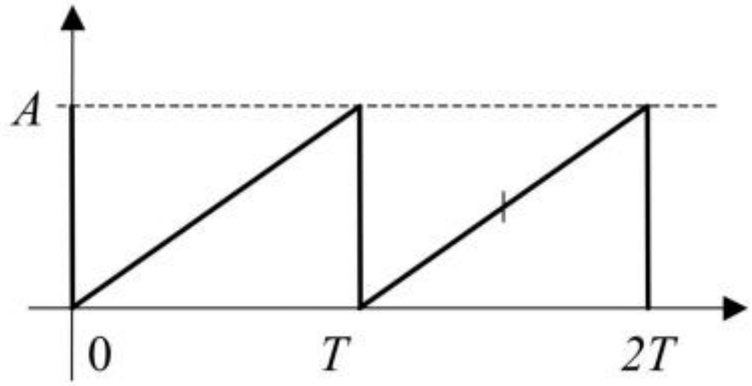
$$b_n = 0$$

Ponieważ funkcja jest nieparzysta na przedziale $<-\frac{T}{2}, \frac{T}{2}>$.

Współczynniki a_n zeruje się dla parzystych n. Dlatego możemy zamienić n na 2k-1. Ostatecznie otrzymujemy szereg Fouriera:

$$f(x) = \frac{A}{2} - \sum_{k=1}^{\infty}\left(\frac{2A}{\pi^2(2k-1)^2} cos\left(\frac{(2k-1)2\pi}{T} t\right)\right)$$

5 Sygnał piłokształtny



Sygnał piłokształtny o okresie T i Amplitudzie A możemy zapisać jako:

$$f(t) = \begin{cases} \dfrac{A}{T} \cdot t & \text{dla } 0 < t < T \\ 0 & \text{dla } t = 0 \; i \; t = T \end{cases}$$

$$a_0 = \frac{2}{T} \cdot \int_0^T \frac{A}{T} \cdot t \cdot dt = A$$

$$a_n = \frac{2}{T} \cdot \int_0^T \frac{A}{T} \cdot t \cdot \cos\left(\frac{2\pi n}{T} t\right) \cdot dt = 0$$

$$b_n = \frac{2}{T} \cdot \int_0^T \frac{A}{T} \cdot t \cdot \sin\left(\frac{2\pi n}{T} t\right) \cdot dt = -\frac{A}{\pi n}$$

Ostatecznie otrzymujemy szereg Fouriera:

$$f(x) = \frac{A}{2} - \sum_{n=1}^{\infty} \frac{A}{\pi n} sin\left(\frac{2\pi n}{T} t\right)$$

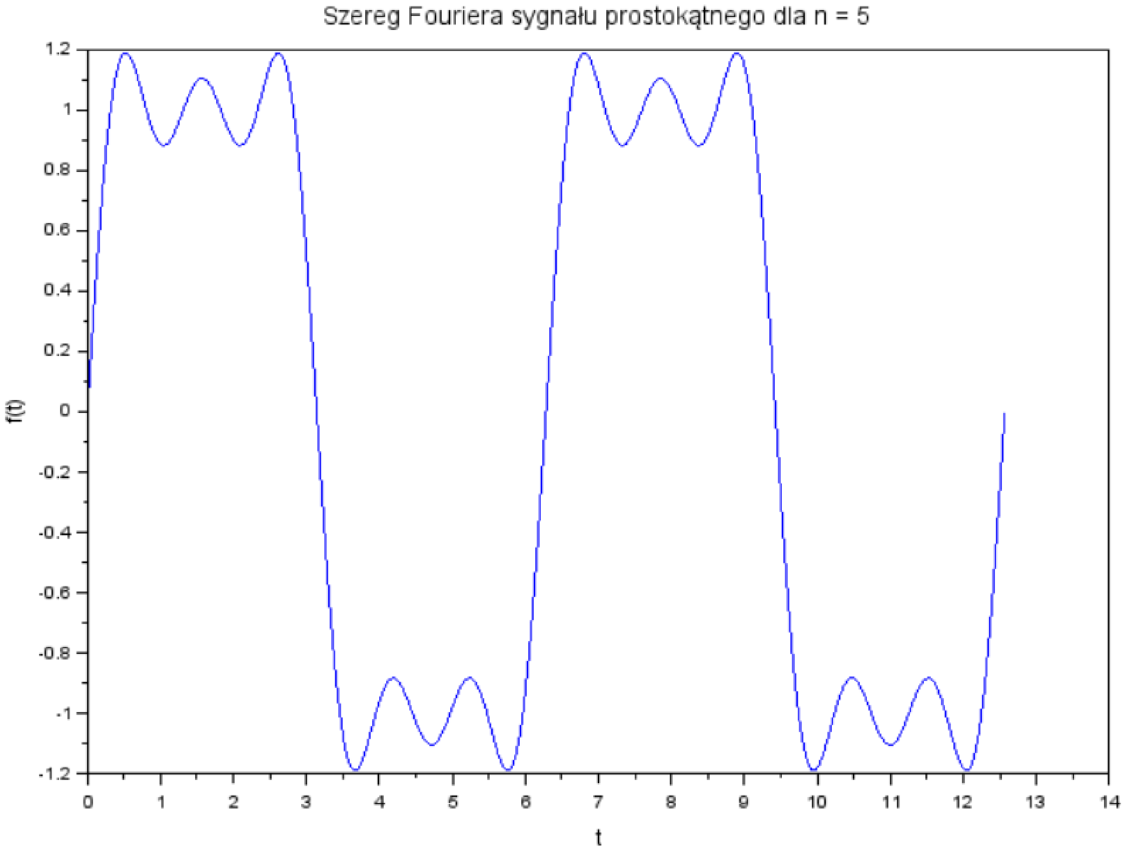
6 Wizualizacje i kod w Scilab

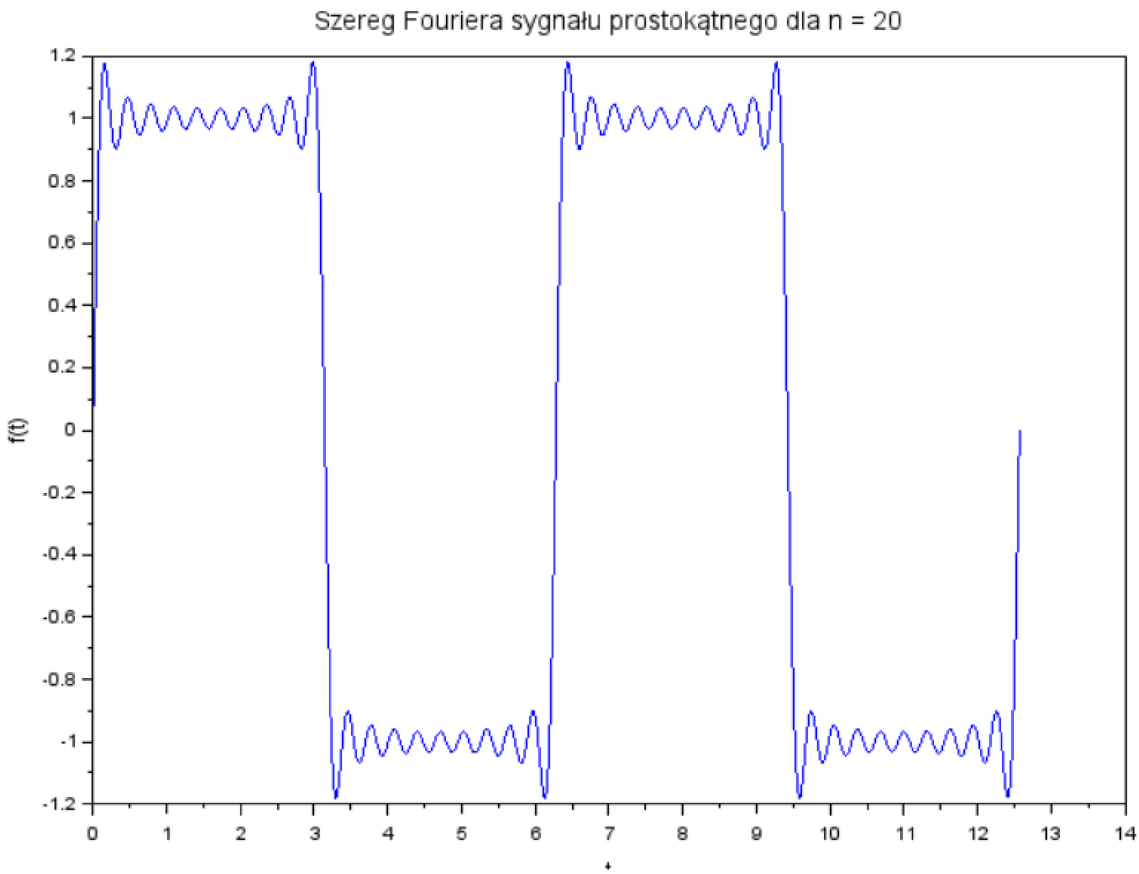
6.1 Sygnał prostokątny

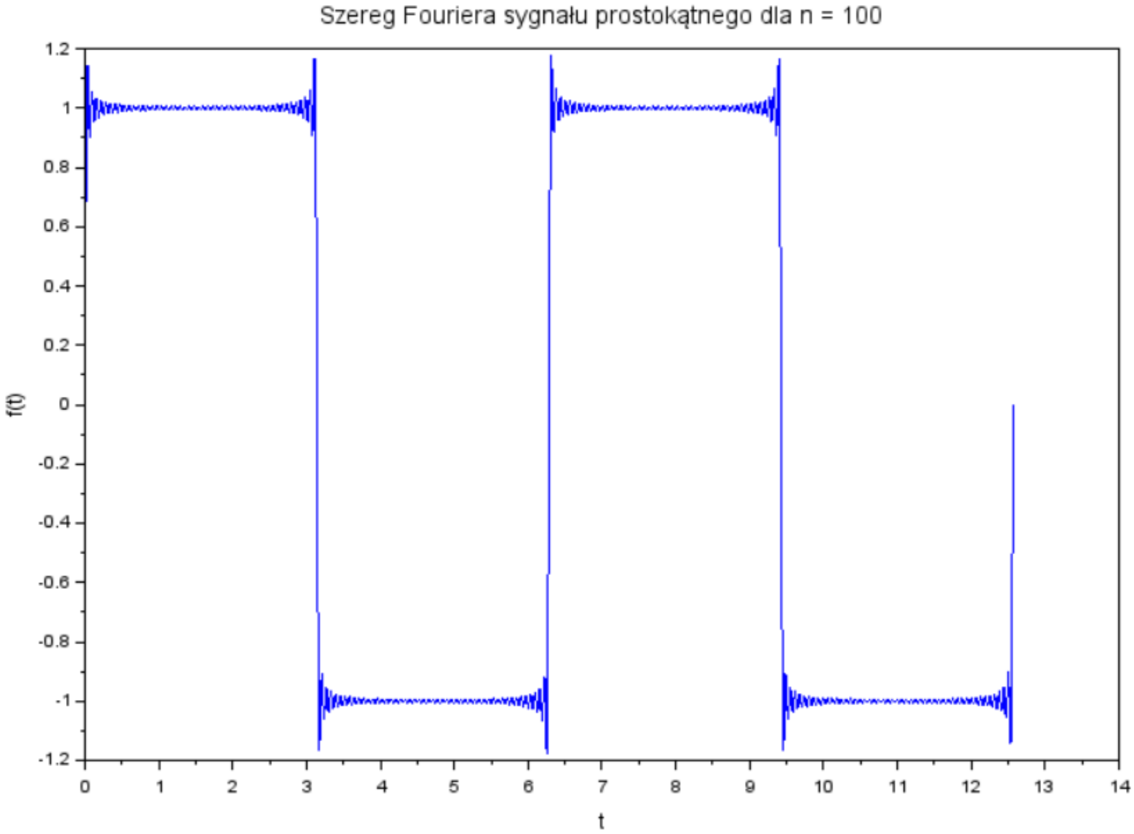
Kod w Scilabie użyty do wygenerowania wykresów fali prostokątnej dla wartości A=1, T=2π:

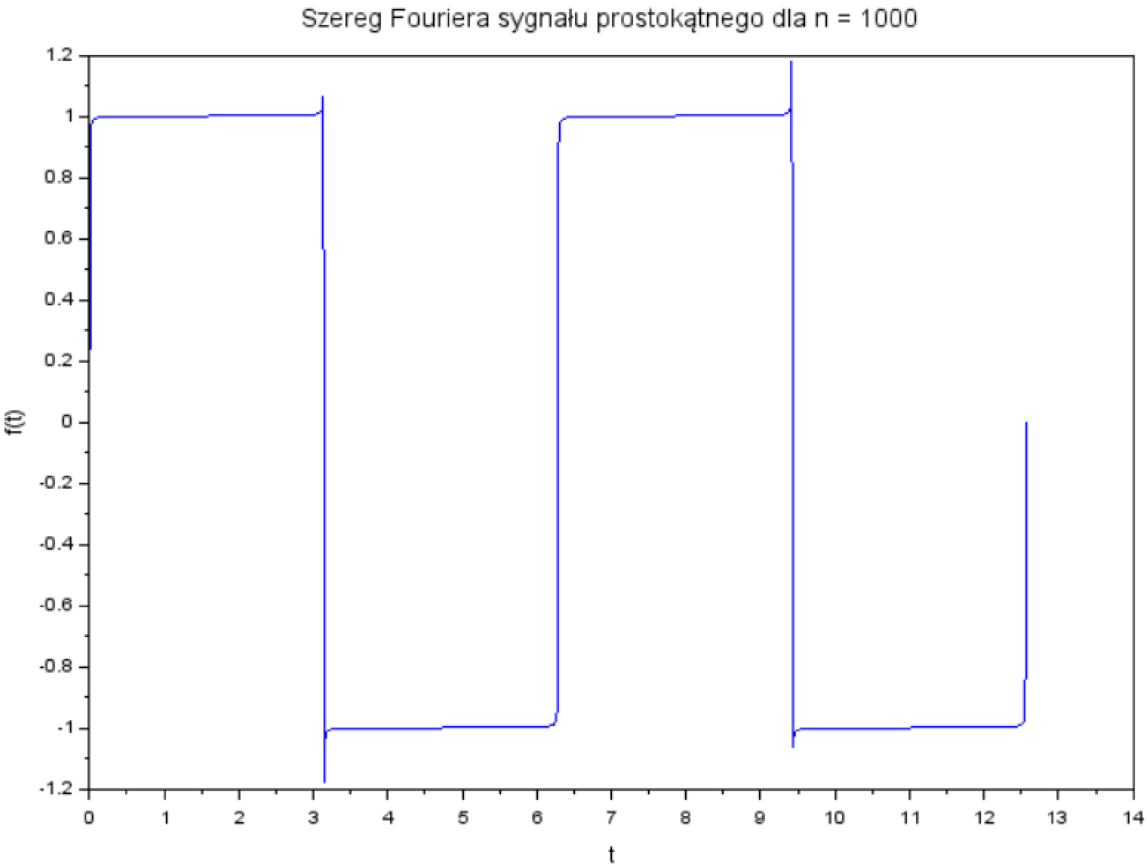
```
// Definicja funkcji prostokątnej
function y = square_wave(t, T, A)
    y = zeros(t);
    for i = 1:length(t)
        if mod(t(i), T) < T/2 then
            y(i) = A;
        elseif mod(t(i), T) < T
            y(i) = -A;
        else
            y(i) = 0;
        end
    end
end
// Obliczanie szeregu Fouriera
function y = fourier_series(t, T, A, n_max)
    y = zeros(t);
    for k = 1:n_max
        n = 2*k - 1; // Tylko nieparzyste n
        y = y + (4*A / (n * %pi)) * sin((2 * n * %pi * t) / T);
    end
end
// Parametry
T = 2 * %pi;
A = 1;
n = 10;
k_max = ceil(n / 2); // Przeliczenie n na k_max
// Wektor czasu
t = linspace(0, 2*T, 1000);
// Obliczanie szeregu Fouriera dla przypisanego n
y = fourier_series(t, T, A, k_max);
// Rysowanie wykresu szeregu Fouriera
clf;
plot(t, y);
title("Szereg Fouriera sygnału prostokątnego dla n = " + string(n));
xlabel("t");
ylabel("f(t)");
xticks([0, %pi/2, %pi, 3*%pi/2, 2*%pi]);
xticklabels(["0", "\pi/2", "\pi", "3\pi/2", "2\pi"]);
```

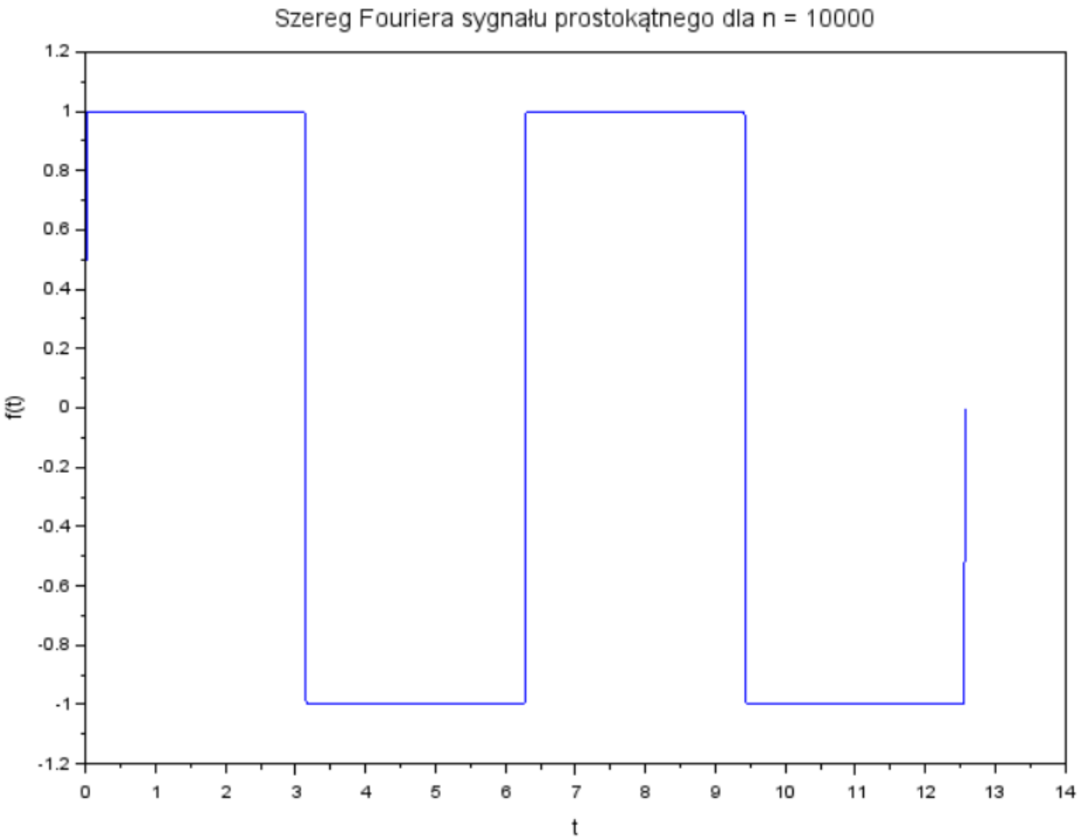
Otrzymane wykresy rozwinięcia szeregu Fouriera dla różnych n:






Szereg Fouriera sygnału prostokątnego dla n = 100
f(t)
t
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14


Szereg Fouriera sygnału prostokątnego dla n = 1000
f(t)
t
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14



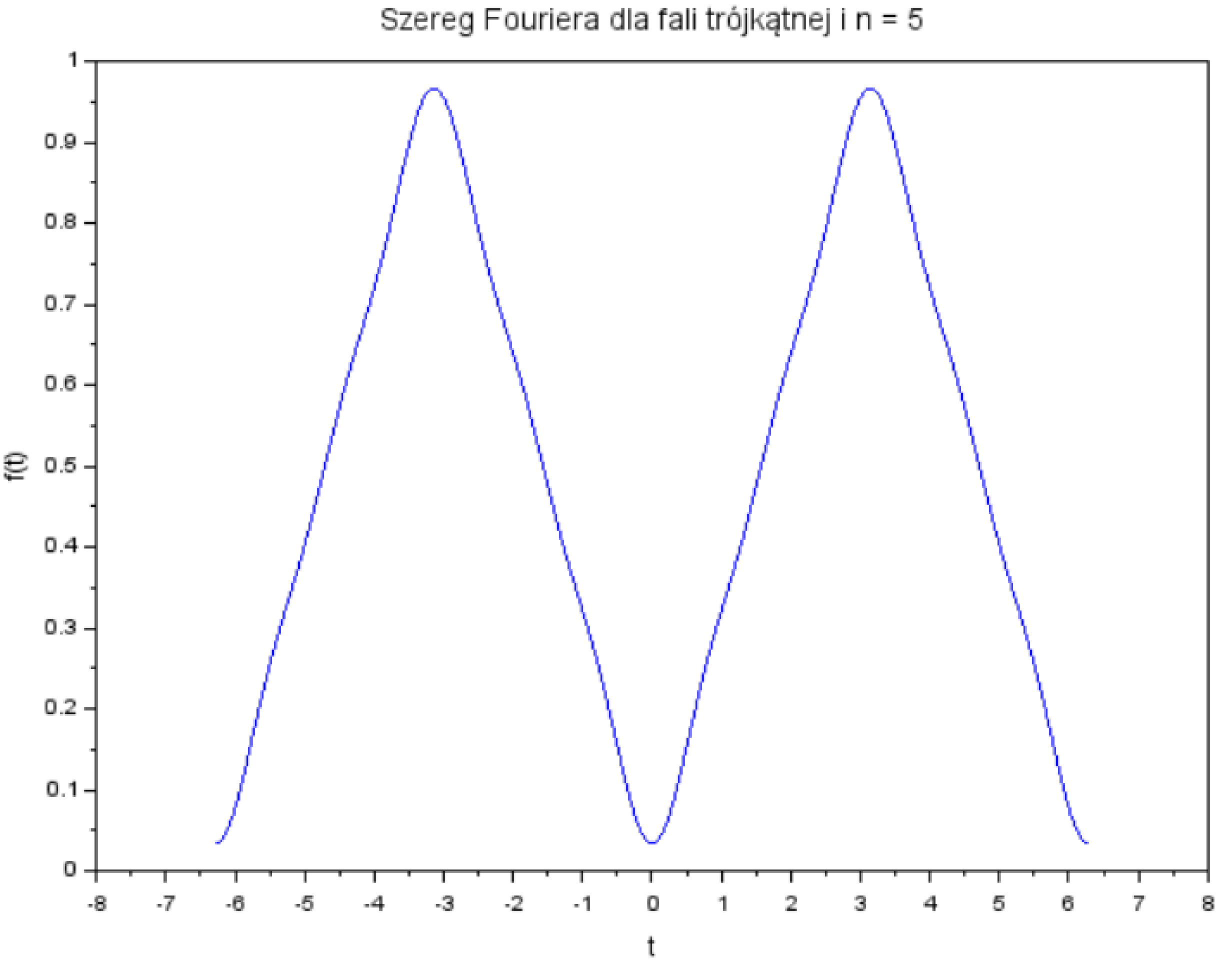
Na przedstawionych wykresach widać wpływ wartości n na dokładność funkcji. Im większe n tym lepsze odwzorowanie. Aby wykres był zbliżony do faktycznej funkcji potrzebna jest wysoka wartość n na poziomie 10 000.

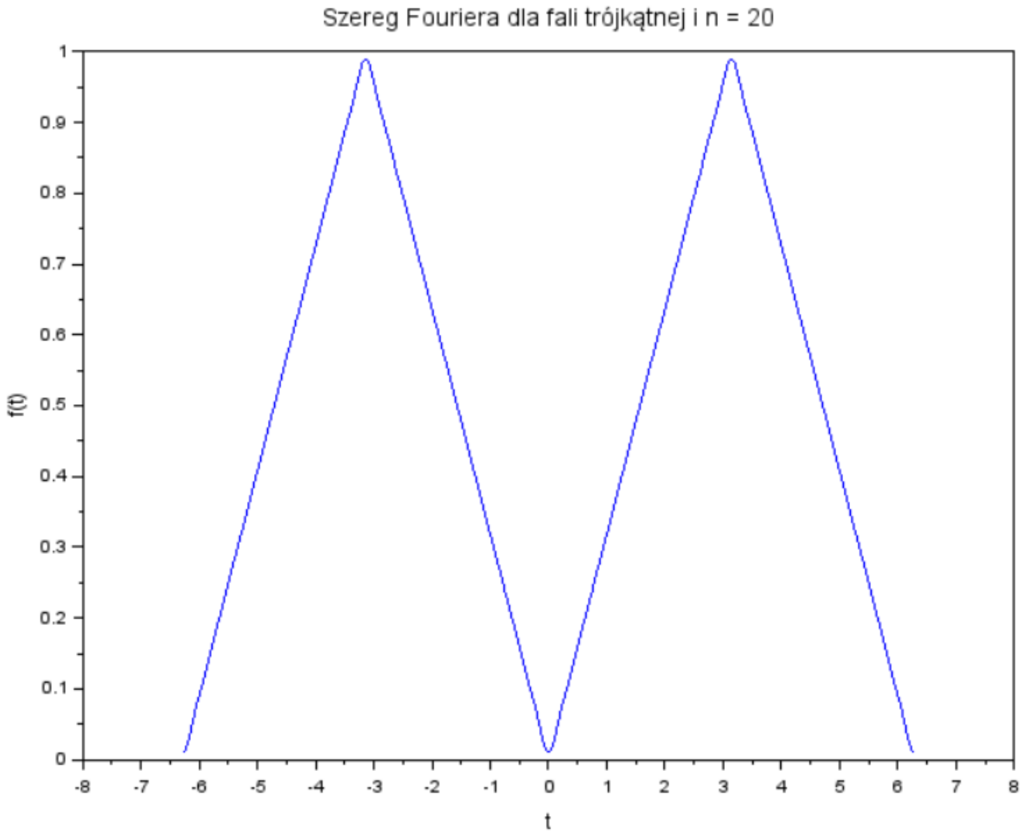
6.2 Sygnał trójkątny

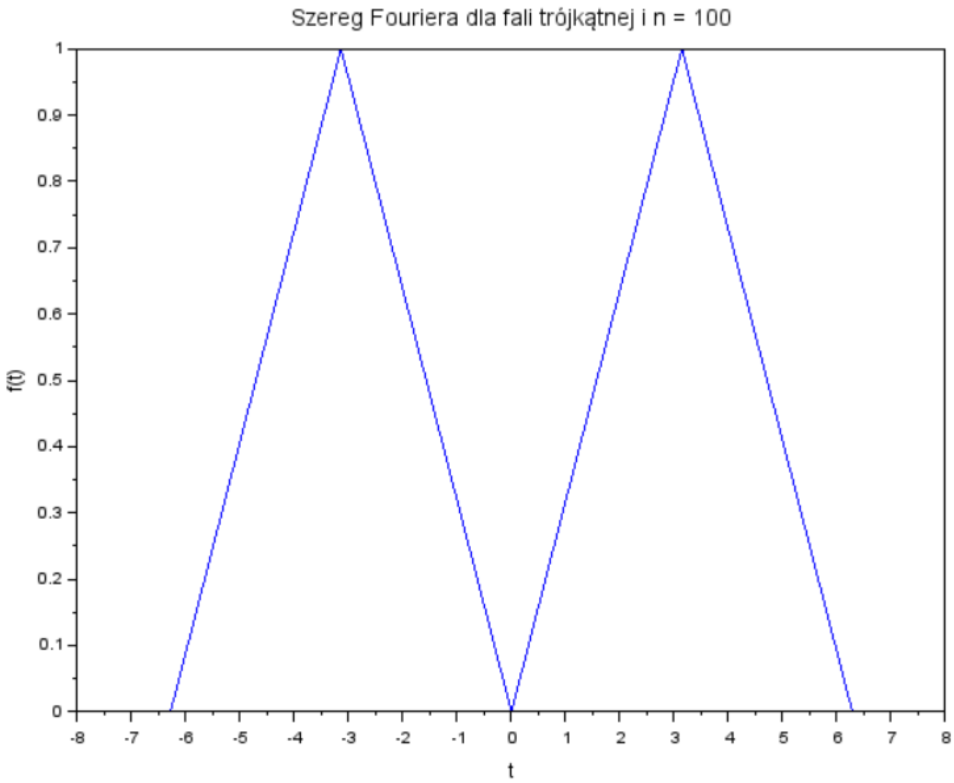
Kod w Scilabie użyty do wygenerowania wykresów fali trójkątnej dla wartości A=1, T=2π:

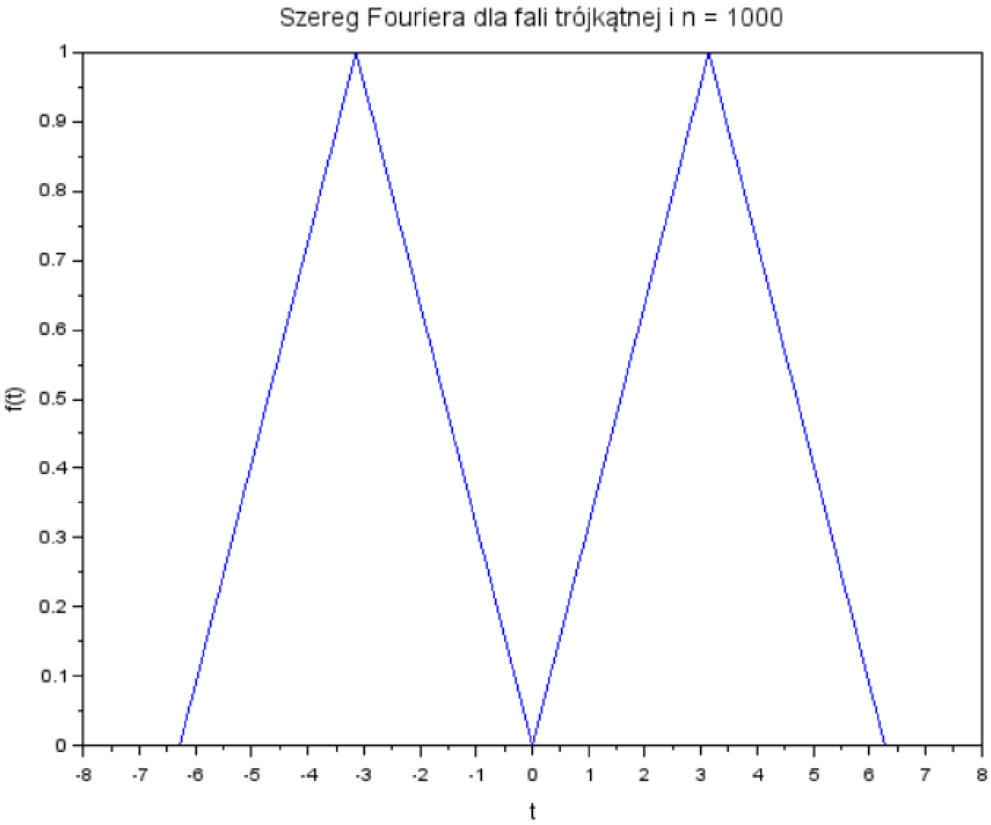
```
// Definicja funkcji trójkątnej
function y = triangle_wave(t, T, A)
    y = zeros(t);
    for i = 1:length(t)
        if mod(t(i), T) < T/2 then
            y(i) = (4 * A / T) * mod(t(i), T) - A;
        else
            y(i) = - (4 * A / T) * mod(t(i), T) + 3 * A;
        end
    end
end
// Obliczanie szeregu Fouriera
function y = fourier_series(t, T, A, n_max)
    y = A/2 * ones(t); // a0 term
    for k = 1:n_max
        n = 2 * k - 1; // Tylko nieparzyste n
        y = y - (4 * A / (n^2 * %pi^2)) * cos((2 * n * %pi * t) / T);
    end
end
// Parametry
T = 2 * %pi;
A = 1;
n = 2;
k_max = ceil(n / 2); // Przeliczenie n na k_max
// Wektor czasu dla dwóch okresów
t = linspace(-T, T, 1000); // Zmiana górnej granicy na 2*T
// Obliczanie szeregu Fouriera dla przypisanego n
y = fourier_series(t, T, A, k_max);
// Rysowanie wykresu szeregu Fouriera
clf;
plot(t, y);
title("Szereg Fouriera dla fali trójkątnej i n = " + string(n));
xlabel("t");
ylabel("f(t)");
xticks([0, %pi, 2 * %pi, 3 * %pi, 4 * %pi]);
xticklabels(["0", "\pi", "2\pi", "3\pi", "4\pi"]);
// Wyświetlanie funkcji trójkątnej dla porównania
figure();
plot(t, triangle_wave(t, T, A));
```

Otrzymane wykresy rozwinięcia szeregu Fouriera dla różnych n:




Szereg Fouriera dla fali trójkątnej i n = 20
f(t)
t
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
-8
-7
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5
6
7
8


Szereg Fouriera dla fali trójkątnej i n = 100
f(t)
t
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
-8
-7
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5
6
7
8



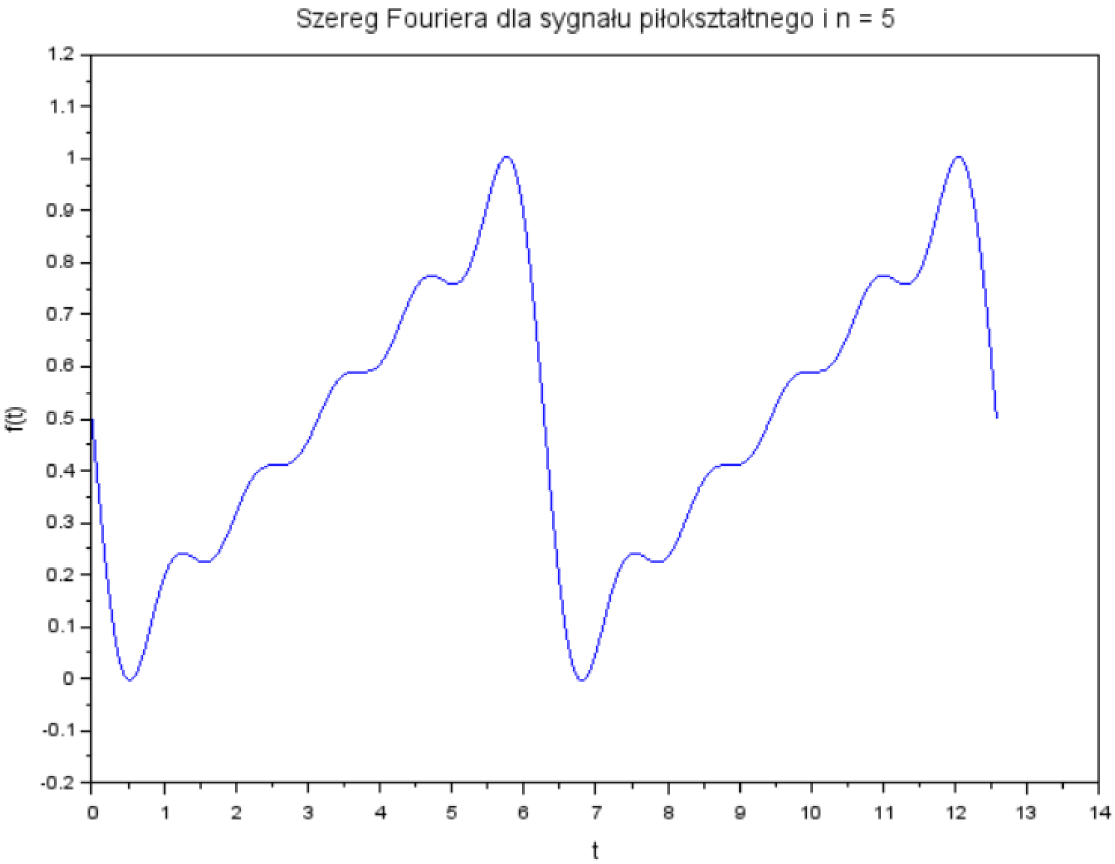
Na przedstawionych wykresach widać wpływ wartości n na dokładność funkcji. Im większe n tym lepsze odwzorowanie. Jednak wykresy są zbliżone do faktycznej funkcji już dla n o wartości 100.

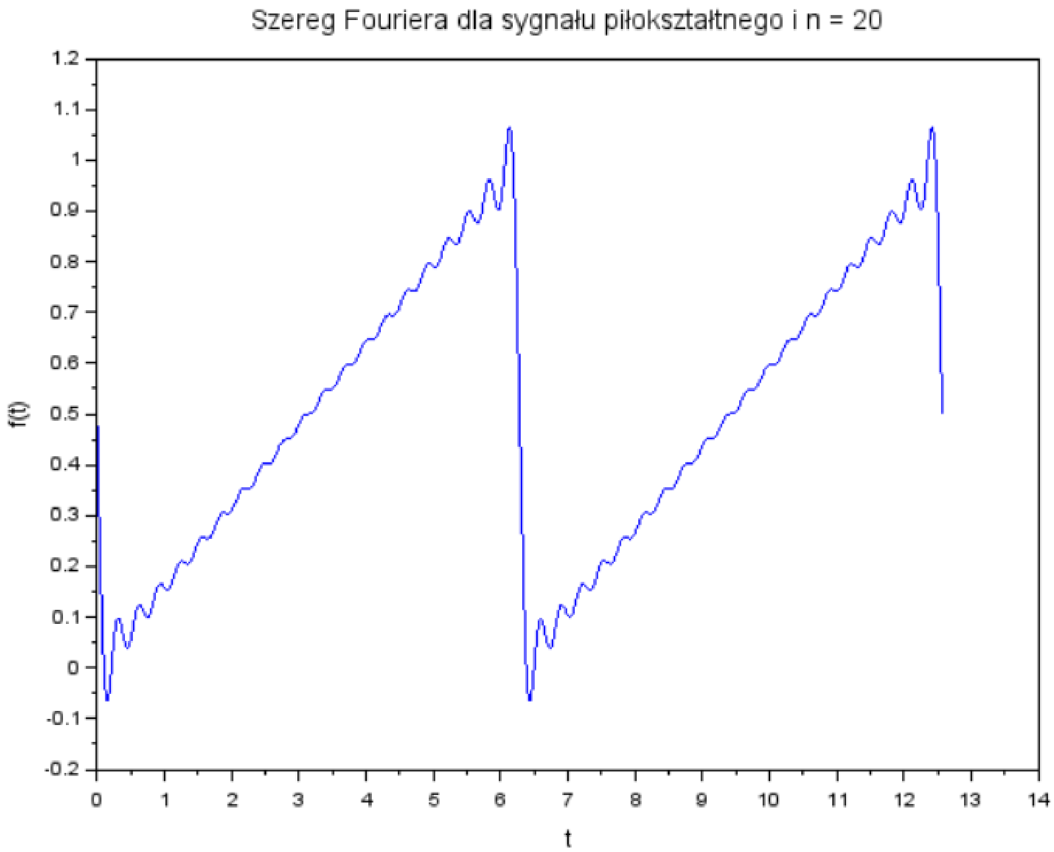
6.3 Sygnał piłokształtny

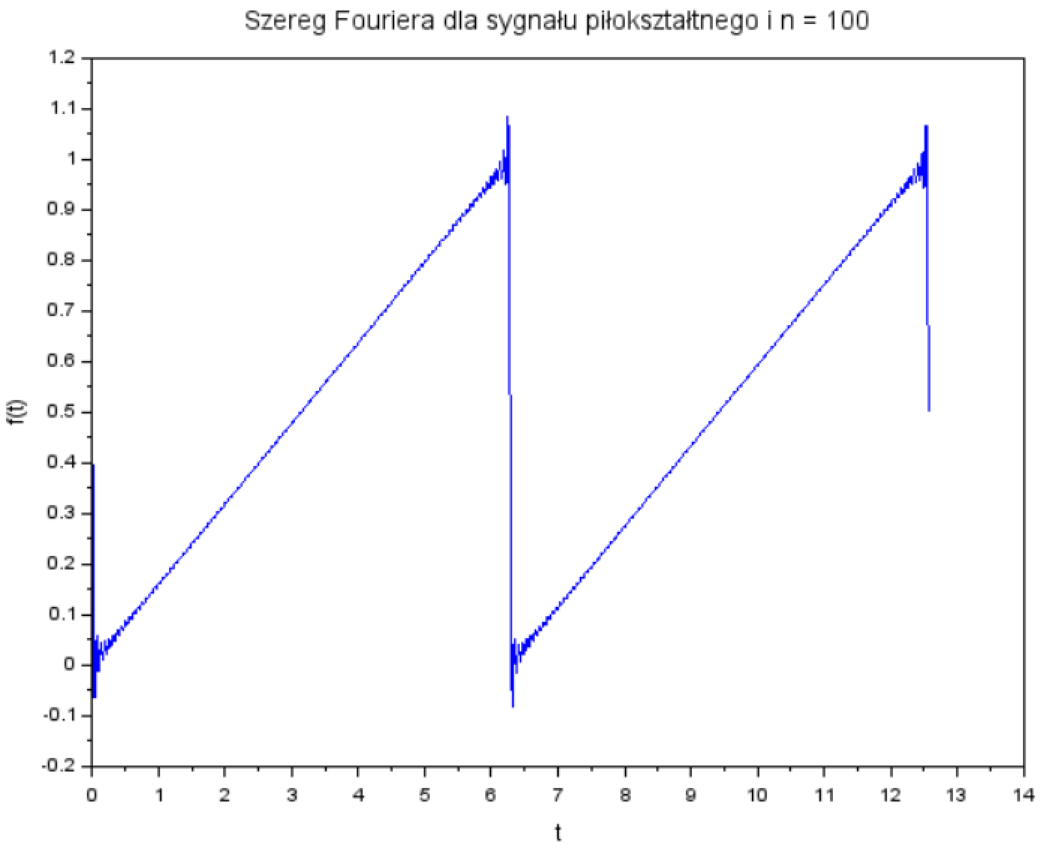
Kod w Scilabie użyty do wygenerowania wykresów fali piłokształtnej dla wartości A=1, T=2π:

```
// Definicja funkcji piłokształtnej
function y = sawtooth_wave(t, T, A)
    y = zeros(t);
    for i = 1:length(t)
        if mod(t(i), T) < T then
            y(i) = (A / T) * mod(t(i), T);
        else
            y(i) = 0;
        end
    end
end
// Obliczanie szeregu Fouriera
function y = fourier_series(t, T, A, n_max)
    y = A/2 * ones(t); // a0 term
    for n = 1:n_max
        y = y - (A / (n * %pi)) * sin((2 * %pi * n * t) / T);
    end
end
// Parametry
T = 2 * %pi;
A = 1;
n = 5; // Ręcznie przypisana wartość n (można zmieniać)
k_max = n; // Przeliczenie n na k_max
// Wektor czasu dla dwóch okresów
t = linspace(0, 2 * T, 1000); // Zmiana górnej granicy na 2*T
// Obliczanie szeregu Fouriera dla przypisanego n
y = fourier_series(t, T, A, k_max);
// Rysowanie wykresu szeregu Fouriera
clf;
plot(t, y);
title("Szereg Fouriera dla sygnału piłokształtnego i n = " + string(n));
xlabel("t");
ylabel("f(t)");
xticks([0, %pi, 2 * %pi, 3 * %pi, 4 * %pi]);
xticklabels(["0", "\pi", "2\pi", "3\pi", "4\pi"]);
// Wyświetlanie funkcji piłokształtnej dla porównania
figure();
plot(t, sawtooth_wave(t, T, A));
```

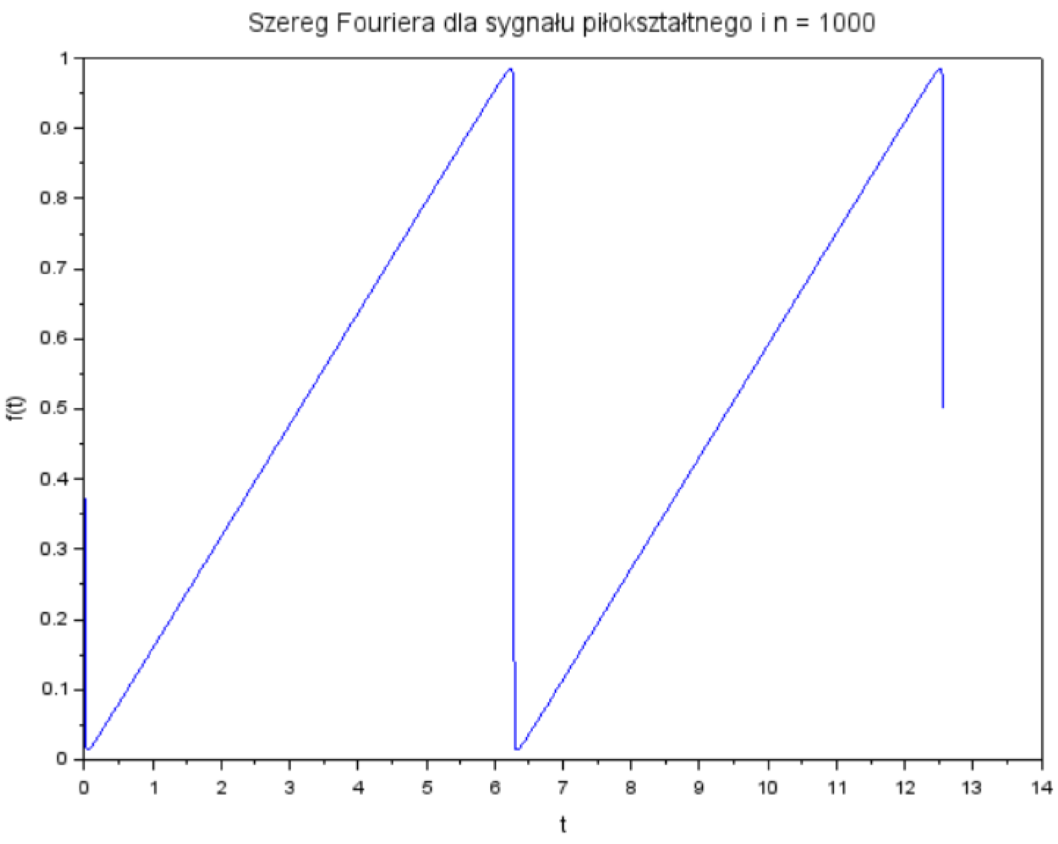
Otrzymane wykresy rozwinięcia szeregu Fouriera dla różnych n:



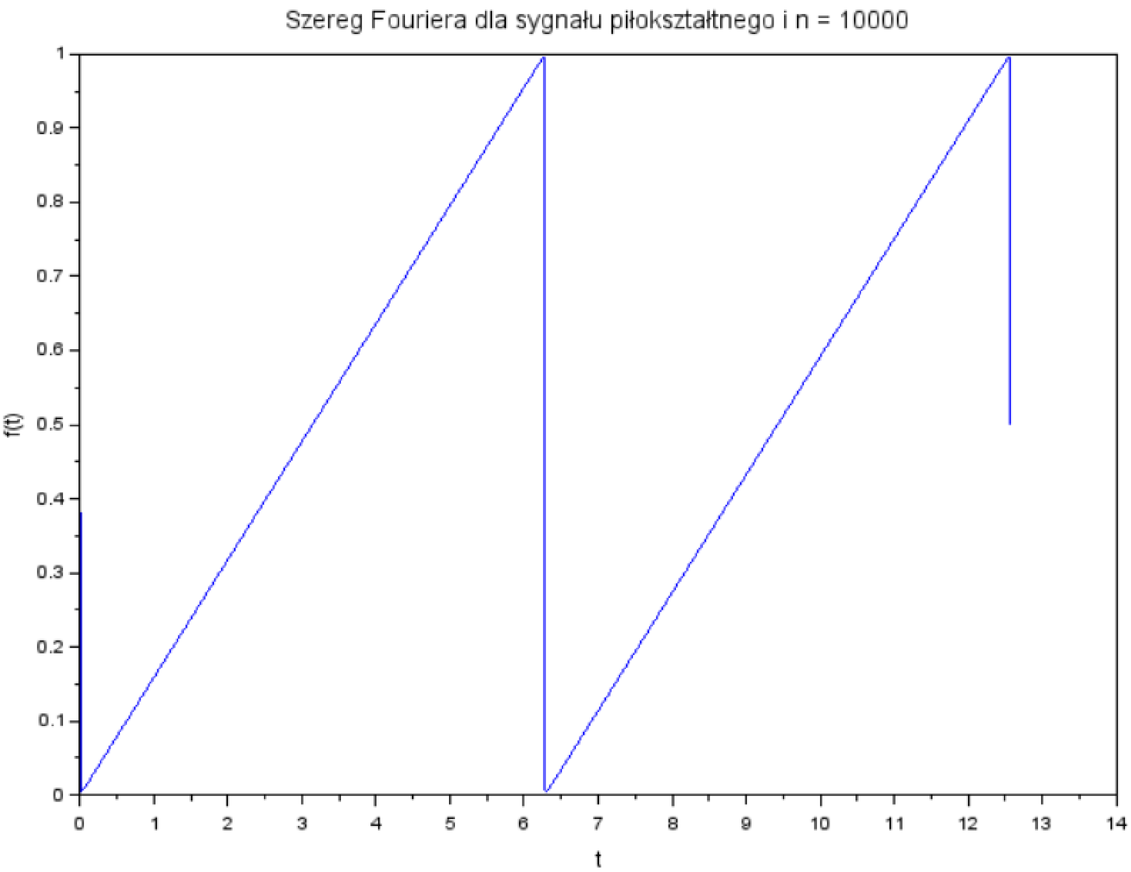




Szereg Fouriera dla sygnału piłokształtnego i n = 100
f(t)
t



Szereg Fouriera dla sygnału piłokształtnego i n = 1000
f(t)
t



Na przedstawionych wykresach widać wpływ wartości n na dokładność funkcji. Im większe n tym lepsze odwzorowanie. Wykresy są zbliżone do faktycznej funkcji dla n o wartości 1000.

7 Podsumowanie

Wyznaczyliśmy i zbadaliśmy rozwinięcia w szeregi Fouriera 3 sygnały: prostokątny, trójkątny i piłokształtny. W numerycznym rozwiązaniu przy pomocy Scilaba przedstawiliśmy funkcje dla różnych n na wykresach, wartość n miała największe znaczenie przy generowaniu sygnału prostokątnego, ponieważ potrzebna była największa wartość n, aby wykres przypominał faktyczną funkcję. Najgorzej odwzorowane miejsca tych funkcji były miejsca o nagłej zmianie wartości, w tak zwanych punktach skokowych.