

Laboratorium Metod Numerycznych

Numeryczna przetwarzanie danych

Laboratorium 8

Dane otrzymywane z pomiarów i symulacji numerycznych często mogą wymagać przetworzenia przed ich analizą - wykorzystaniem. Do podstawowych i najczęściej stosowanych metod przetwarzania danych można zaliczyć różniczkowanie i całkowania oraz ich interpolacje i aproksymację. W niniejszym ćwiczeniu omówione i przećwiczone zostaną wymienione operacje na danych.

1) Różniczkowanie

Różniczkowanie numeryczne to technika przybliżonego obliczania pochodnych funkcji matematycznych za pomocą metod numerycznych. Jest to szczególnie przydatne w przypadkach, gdy funkcje są trudne do różniczkowania analitycznego lub gdy dane są dostępne tylko w postaci dyskretnej. Znanych i stosowanych jest kilka metod różniczkowania numerycznego.

- Różnice skończone, dzielą się na dwie główne kategorie:
 - Różnice w przód obliczają przybliżoną pochodną na podstawie wartości funkcji w punktach x i $x + \Delta x$, gdzie Δx jest niewielką wartością kroku.
 - Różnice wsteczne korzystają z wartości funkcji w punktach x i $x - \Delta x$ do obliczenia przybliżonej pochodnej.
- Różnice centralne, są bardziej dokładne niż różnice skończone. Obliczają przybliżoną pochodną, korzystając z wartości funkcji w punktach $x + \Delta x$ i $x - \Delta x$.
- Metoda różnic skończonych, łączy różnice skończone z użyciem wielu punktów, aby uzyskać dokładniejsze przybliżenie pochodnej.
- Metoda różnic skończonych centralnych, rozwinięcie metody różnic skończonych, używające wielu punktów i obliczające przybliżoną pochodną za pomocą średnich różnic.
- Metoda różnic dzielonych, opiera się na interpolacji wielomianowej, a następnie oblicza pochodną tego wielomianu.

W praktyce dobór odpowiedniej metody zależy od konkretnej sytuacji i charakterystyki funkcji. Ważne jest również odpowiednie dobranie wartości kroku Δx , aby

uzyskać równowagę między dokładnością a stabilnością numeryczną.

Numeryczne różniczkowanie znajduje szerokie zastosowanie w nauce i technice, między innymi w:

- analiza danych eksperymentalnych → w przypadku eksperymentów, w których dane są zbierane w formie punktów dyskretnych, różniczkowanie numeryczne pozwala na obliczanie pochodnych, co może być istotne w analizie wyników eksperymentalnych.
- symulacje numeryczne → w modelowaniu matematycznym i symulacjach numerycznych często używa się różniczkowania numerycznego do obliczania pochodnych i gradientów. Jest to istotne w dziedzinach takich jak dynamika płynów, fizyka komputerowa, analiza wytrzymałości materiałów, etc.
- optymalizacja → metody optymalizacji często wykorzystują pochodne w celu znajdowania minimum lub maksimum funkcji celu. Różniczkowanie numeryczne może być używane do obliczania tych pochodnych.
- rozwiązywanie równań różniczkowych → w numerycznym rozwiązywaniu równań różniczkowych często pojawia się konieczność obliczania pochodnych. Metody różniczkowania numerycznego mogą być używane do tego celu.

Różniczkowanie numeryczne realizowane jest na podstawie tablic danych. Jest to w zasadzie proces odwrotny do stosowanego przy rozwiązywaniu równań różniczkowych. W efekcie końcowy uzyskuje się nie równanie funkcji a tablicę danych, charakterystykę na wykresie.

W środowisku **Scilab** jest dostępna funkcja realizująca zadanie wyznaczania przyrostów - `diff()`. Pozwala na wyznaczenia przyrostów określonego rzędu dla wektora lub macierzy danych. Przyjmuje trzy atrybuty wejściowe, przy czym tylko pierwszy jest obowiązkowy, składnię zapisano poniżej.

```
y=diff(f,n,"typ");
```

gdzie:

- `y` → wektor lub macierzy wyjściowa danych zróżniczkowanych
- `f` → wektor lub macierz wejściowa, może to być także funkcja zwracająca wektor lub macierz danych
- `n` → rząd różniczkowania, jeżeli atrybut nie zostanie podany funkcja zwraca pochodną pierwszego rzędu
- `"typ"` → atrybut tekstowy określający sposób różniczkowania danych przekazanych w postaci macierzy, może przyjmować trzy wartości: `"r"` - różniczkowanie wierszowe, `"c"` - różniczkowanie kolumnowe, `"*"` - różniczkowania standardowe

Wyznaczenie pochodnej wymaga obliczenia przyrostu funkcji w stosunku do przyrostu zmiennej $f'(x) = \frac{\Delta f(x)}{\Delta x}$. W środowisku Scilab pochodna jest więc stosunkiem dwóch wektorów przyrostowych, funkcji (danych) i zmiennej (zazwyczaj czasu), jak pokazano poniżej.

```
y=diff(wektor)./diff(t);
```

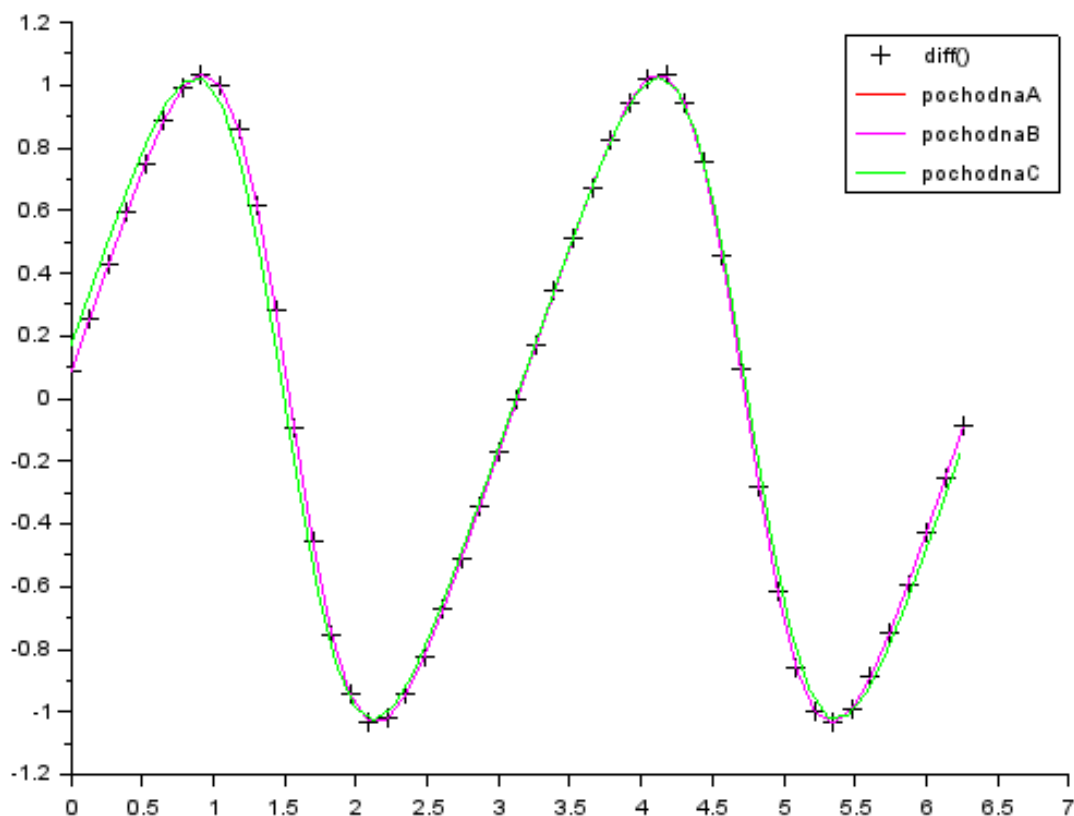
gdzie: **wektor** jest funkcją zmiennej **t** lub wektorem danych, a **t** jest wektorem zmiennej podcałkowej (np. kolejne kroki czasowe). Należy zwrócić uwagę na użycie znaku "." przed znakiem dzielenia w celu wykonania dzielenia algebraicznego.

Skrypt zapisany poniżej prezentuje przykład realizacji różniczkowania przykładowej tablicy danych za pomocą funkcji `diff()` i opracowanych funkcji różniczkujących. Tablica danych została utworzona na podstawie zapisanej poniżej funkcji.

$$f(x) = \frac{2 * \sin^2(x)}{2 + \cos^2(x)} \quad (1)$$

```
clear;clc;clf;
function y=forward(t,f)
    n=size(t)(2);
    for i=1:n-1
        y(i)=(f(i+1)-f(i))/(t(i+1)-t(i));
    end
endfunction
function y=backwards(t,f)
    n=size(t)(2);
    for i=2:n
        y(i-1)=(f(i)-f(i-1))/(t(i)-t(i-1));
    end
endfunction
function y=central(t,f)
    n=size(t)(2);
    for i=2:n-1
        a=f(i+1)-f(i-1)
        b=2*(t(i+1)-t(i));
        y(i-1)=a/b;
    end
endfunction
dt=0.02;
n=1/dt;
t=linspace(0,2*%pi,n);
t2=linspace(0,2*%pi-dt,n-1);
t3=linspace(0,2*%pi-2*dt,n-2);
f=2*sin(t).^2./(2+cos(t).^2);
f2=diff(f)./diff(t);
f3=forward(t,f);
f4=backwards(t,f);
f5=central(t,f);
plot2d(t2,[f2' f3 f4],[-1 5 6]);
plot2d(t3,f5,3);
legends(["diff()" "po...A" "po...B" "po...C"],[-1 5 6 3], opt="ur");
```

W powyższym przykładzie zdefiniowano trzy funkcje realizujące różniczkowanie: w przód, w tył i centralne. Przebiegi zróżniczkowanej funkcji pokazano na Rys.1.



Rys. 1: Przebiegi zróżniczkowanej funkcji

2) Całkowanie numeryczne

Metody numerycznego całkowania polegają na wyznaczeniu wartości przybliżonej całki oznaczonej. W środowisku **Scilab**'a obliczenia całki oznaczonej realizuje funkcja `integrate()`. Składnia podstawowego wywołania zapisana poniżej wymaga podania czterech atrybutów.

```
result = integrate('funkcja','zmienna',start,koniec);
```

gdzie:

- 'funkcja' → jest kodem lub odwołaniem do funkcji podcałkowej
- 'zmienna' → jest zmienną funkcji podcałkowej względem której realizowane jest całkowanie
- start i koniec → definiują przedział całkowania

Proste metody całkowania numerycznego polegają na przybliżeniu całki za pomocą odpowiedniej sumy ważonej wartości całkowanej funkcji w kilku punktach. Aby uzyskać dokładniejsze przybliżenie dzieli się przedział całkowania na niewielkie fragmenty. Ostateczny wynik jest sumą oszacowań całek w poszczególnych podprzedziałach. Najczęściej przedział dzieli się na równe podprzedziały, ale bardziej wyszukane algorytmy potrafią dostosowywać krok do szybkości zmienności funkcji (analiza pochodnej, przyrostu funkcji). Największy wpływ na dokładność całkowania numerycznego będzie w przypadku każdej z metod będzie miała wielkość kroku obliczeniowego. Im mniejsze przedziały tym wynik obliczeń obarczony będzie mniejszym błędem.

Duży wpływ na procedurę całkowania numerycznego ma sposób opisu całkowanej funkcji. Dane mogą być przekazane w postaci równania algebraicznego lub tablicy danych. W pierwszym przypadku skok podziału może być dobierany dowolnie. W przypadku przetwarzania danych stabelaryzowanych krok obliczeń jest ustalony. Sterowanie jego krokiem jest możliwe w przypadku za aproksymowania danych do funkcji.

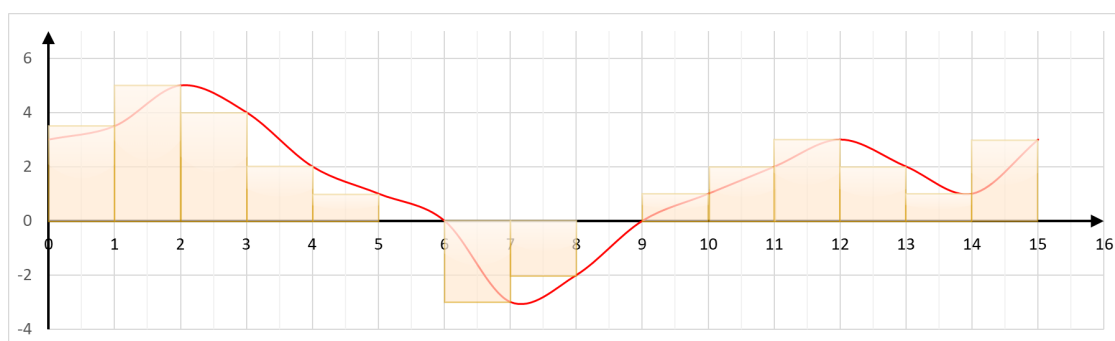
Podstawowe metody całkowania numerycznego różnią się sposobem aproksymacji kształtu całkowanej krzywej w poszczególnych iteracjach obliczeniowych. Zalicza się do nich aproksymację do stałej, odcinka prostej i paraboli. Wybór metody determinuje więc sposób obliczania przybliżonej wartości pola po krzywa dla pojedynczego kroku.

2.1) Metoda prostokątów

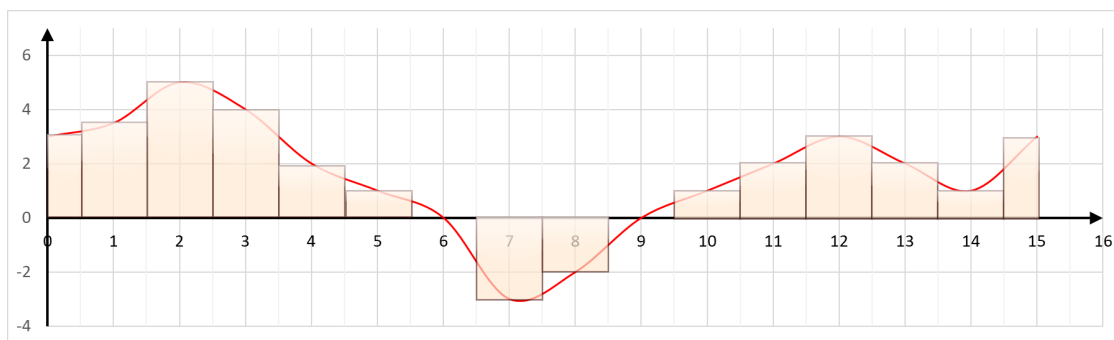
Przybliżenie zależności pola pod krzywą dla pojedynczego kroku do pola prostokąta przy aproksymacji kształtu krzywej do wartości stałej pozwala na wyznaczenie przybliżonej wartości całki oznaczonej w zadanym przedziale. Spośród omawianych metod algorytm ten obarczony jest największym błędem. Poprawę dokładności metody można osiągnąć jedynie poprzez zmniejszenie kroku obliczeniowego.

Metoda występuje w trzech wariantach, w zależności od sposobu przybliżania do stałej dla kroku obliczeniowego:

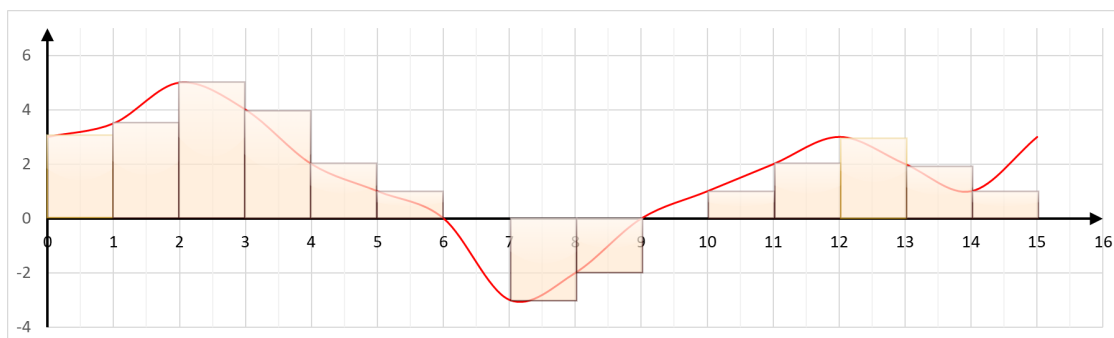
- lewych prostokątów $\rightarrow \langle t - \Delta t, t \rangle$, jak pokazano na Rys.2
 - środkowych prostokątów $\rightarrow \langle t - \frac{\Delta t}{2}, t + \frac{\Delta t}{2} \rangle$, jak pokazano na Rys.3
 - prawych prostokątów $\rightarrow \langle t, t + \Delta t \rangle$, jak pokazano na Rys.4
- ,gdzie Δt jest krokiem obliczeń.



Rys. 2: Metoda prostokątów (lewych)



Rys. 3: Metoda prostokątów (środkowych)



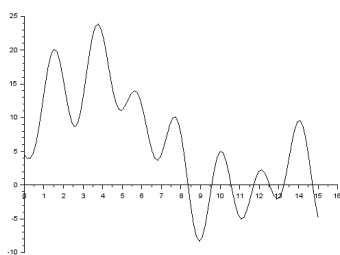
Rys. 4: Metoda prostokątów (prawych)

Spośród wymienionych powyżej odmian metody prostokątów, metoda *środkowych prostokątów* cechuje się najlepszym przybliżeniem, co jest konsekwencją uśrednienia wartości funkcji w każdym kroku obliczeń. Uogólniony algorytm metody można zapisać równaniem (2).

$$\int_a^b f(x)dx = \sum_{n=a}^b f(n) \cdot \Delta x \quad (2)$$

W zależności od zastosowanego sposobu aproksymacji przedziałowej funkcji podcałkowej dla metody prostokątów należy odpowiednio w równaniu (2) obrać granice sumowania. Skuteczność omówionych metod w odniesieniu do funkcji wbudowanej dla wybranych wartości kroku obliczeniowego zestawiono w tabeli. Symulację przeprowadzono dla funkcji opisanej równaniem (3) i pokazanej na Rys. 5 w przedziale (0, 15).

$$f(x) = 8 \sin(0.5x) - 5 \sin(3x) - 3 \cos\left(2x + \frac{\pi}{3}\right) - 0.15x^2 + 1.5x + 6 \quad (3)$$



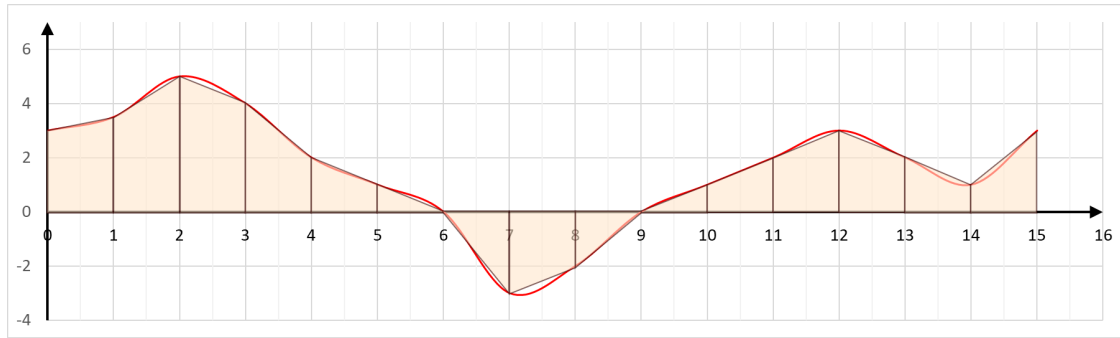
Rys. 5: Przebieg funkcji podcałkowej

Tab. 1: Przykładowa tabela

Lp.	Metoda	Δt	Całka	Błąd
1	<code>integrate()</code>	brak	101.50239	0%
2	lewych	1	96.307788	-5.1%
3	lewych	0.1	101.03132	-0.46%
4	środkowych	1	101.74762	0.25%
5	środkowych	0.1	101.50545	0.003%
6	prawych	1	105.60666	4%
7	prawych	0.1	101.96121	0.46%

2.2) Metoda trapezów

Metoda zakłada przybliżenie krzywej funkcji podcałkowej w rozpatrywanym kroku obliczeniowym za pomocą funkcji liniowej $f(x) = ax + b$, jak pokazano na Rys.6.



Rys. 6: Metoda trapezów

Metoda wykorzystuje wartości funkcji na krańcach kroku obliczeniowego. Niezależnie od sposobu opisu całkowanej funkcji (tablica lub równanie) model obliczeniowy realizuje te same obliczenia.. Algorytm wyznaczania wartości całki oznaczonej metodą trapezów opisano równaniem (4).

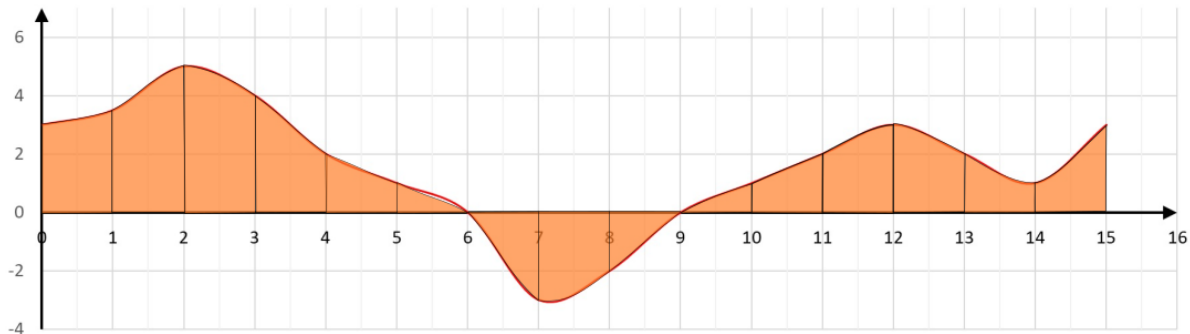
$$\int_a^b f(x)dx = \sum_{n=a}^{b-\Delta x} (f(n + \Delta x) + f(n)) \cdot \frac{\Delta x}{2} \quad (4)$$

Dla przykładowej funkcji opisanej równaniem (3) uzyskane wartości całki oznaczonej dla dwóch długości kroku obliczeniowego zapisano poniżej:

- dla $\Delta t = 1 \rightarrow 100.95723$ z błędem -0.5%
- dla $\Delta t = 0.1 \rightarrow 101.49627$ z błędem -0.006%

2.3) Metoda Simpsona - paraboli

W metodzie funkcja podcałkowa jest aproksymowana przez wielomian każdego kroku obliczeniowego, jak pokazano na Rys.7.



Rys. 7: Metoda Simsona

Zazwyczaj funkcja aproksymowana jest do wielomianu drugiego lub trzeciego stopnia. Algorytmy obliczeniowe metody w zależności od stopnia wielomianu gdy funkcja podcałkowa jest opisana równaniem zapisano poniżej.

- wielomian drugiego stopnia

$$\int_a^b f(x)dx = \sum_{n=a}^{b-\Delta x} \frac{\Delta x}{6} \left(f(n) + 4f\left(n + \frac{\Delta x}{2}\right) + f(n + \Delta x) \right) \quad (5)$$

- wielomian trzeciego stopnia

$$\int_a^b f(x)dx = \sum_{n=a}^{b-\Delta x} \frac{\Delta x}{8} \left(f(n) + 3f\left(n + \frac{\Delta x}{3}\right) + 3f\left(n + \frac{2\Delta x}{3}\right) + f(n + \Delta x) \right) \quad (6)$$

Dla zapisanych powyżej algorytmów uzyskano zestawione poniżej wyniki całkowania numerycznego:

- wielomian drugiego rzędu
 - $\Delta t = 1 \rightarrow 101.48416$ z błędem -0.018%
 - $\Delta t = 0.1 \rightarrow 101.50239$ z błędem -0.000001%
- wielomian trzeciego rzędu
 - $\Delta t = 1 \rightarrow 101.49463$ z błędem -0.008%
 - $\Delta t = 0.1 \rightarrow 101.50239$ x błędem -0.0000005%

Sposób implementacji algorytmów opisanych równaniami (5 i 6) komplikuje się gdy funkcja podcałkowa opisana jest tablica danych. W takim przypadku istotna jest liczba przedziałów obliczeniowych. Dla wielomianu drugiego stopnia musi być liczbą parzystą, natomiast w przypadku wielomianu trzeciego stopnia liczbą podzielną przez trzy. W obu przypadkach algorytmy muszą odwoływać się do kolejnych trzech lub czterech elementów tablicy danych. Po modyfikacji można je zapisać w postaci zestawionych poniżej równań:

- wielomian drugiego stopnia, liczba przedziałów parzysta

$$\int_a^b f(x)dx = \sum_{n=a}^{b-\Delta x} \frac{\Delta x}{3} (d(n) + 4d(n + 1) + d(n + 2)) \quad (7)$$

- wielomian trzeciego stopnia

$$\int_a^b f(x)dx = \sum_{n=a}^{b-\Delta x} \frac{3\Delta x}{8} (d(n) + 3d(n + 1) + 3d(n + 2) + d(n + 3)) \quad (8)$$

gdzie $d(n)$ jest tablicą danych całkowanej funkcji. Należy zwrócić uwagę na różnicę w wartościach współczynników przez które przemnażane są poszczególne czynniki sum. Równanie (5) ma współczynnik dwukrotnie mniejszy od równania (7), a równanie (6) trzykrotnie mniejszy jak równanie (8). Wynika to z rozmiaru kroku całkowania, przy obliczenia realizowanych przy danym równaniu funkcji całkowej Δx jest odległością

między kolejnymi krokami, natomiast w przypadku opisu funkcji za pomocą tablicy danych pojedynczy krok obliczeniowy składa się odpowiednio z dwóch lub trzech przedziałów zdefiniowanych przez kolejne trzy lub cztery punkty z tablicy danych.

Poniżej zestawiono wyniki symulacji zrealizowanych na podstawie algorytmów opisanych równaniami (7) i (8) na podstawie tablic danych:

- dla wielomianu II stopnia, przy 31 punktach \rightarrow 101.48416 z błędem -0.018%
- dla wielomianu III stopnia, przy 31 punktach \rightarrow 101.42795 z błędem -0.07%
- dla wielomianu II stopnia, przy 151 punktach \rightarrow 101.50237 z błędem -0.00002%
- dla wielomianu III stopnia, przy 151 punktach \rightarrow 101.50234 z błędem -0.00004%

3) Interpolacja

Dane pomiarowe zebrane w postaci tablicy liczbowej można zobrazować za pomocą wykresu. W wielu przypadkach równanie funkcji opisującej taki przebieg jest nieznane lub trudne do określenia. Możliwe jest jednak interpolowanie danego przebiegu do zstandaryzowanej funkcji na podstawie zebranych danych (węzłów interpolacyjnych) poprzez wyznaczenie numeryczne współczynników funkcji interpolacyjnej. Zazwyczaj ma ona postać wielomianu algebraicznego lub trygonometrycznego. Cecha charakterystyczną interpolacji jest przyjmowanie przez funkcję interpolującą takich samych wartości w węzłach interpolacyjnych.

3.1) Interpolacja wielomianowa

Podstawową funkcją interpolującą jest wielomian algebraiczny, zapisany w postaci ogólnej jak pokazano poniżej na równaniu (9).

$$W(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3 + \dots + a_n \cdot x^n \quad (9)$$

Wyznaczenie współczynników równania (9) przeprowadza się na podstawie zebranych danych w postaci listy węzłów interpolujących. Na ich podstawie i równania (9) można zapisać układ równań macierzowych $Xa_k = y_k$, jak pokazano poniżej za pomocą równania (10). W najprostszej formie algebraiczny wielomian interpolacyjny będzie miał tyle samo współczynników co węzłów danych wejściowych.

$$\begin{bmatrix} x_0^0 & x_0^1 & x_0^2 & \dots & x_0^n \\ x_1^0 & x_1^1 & x_1^2 & \dots & x_1^n \\ x_2^0 & x_2^1 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n^0 & x_n^1 & x_n^2 & \dots & x_n^n \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (10)$$

Każdy z węzłów interpolacyjnych jest opisany przez współrzędne (x_k, y_k) . Macierz X jest w takim przypadku jest macierzą kwadratową, a wyznaczenie rozwiązania w postaci wektora współczynników a_k można zrealizować tak jak dla każdego układu

równań liniowych. W środowisku *Scilab*'a można to zrealizować stosując jedną z pokazanych poniżej metod:

```
a1=linsolve(X,-y);
a2=lusolve(sparse(X),y);
a3=inv(X)*y;
```

Podejście trzecie wydaje się najbardziej optymalne. Na podstawie wyznaczonych współczynników można zapisać wielomian interpolacyjny. W przypadku dużej liczby węzłów interpolacyjnych wyznaczony zgodnie z pokazaną powyżej metodą wielomian może cechować się znacznymi odchyłami od przebiegu wzorcowego, szczególnie na krańcach przedziału. W takim wypadku korzystniejsze może się okazać interpolowanie funkcji wielomianem niższego rzędu. W efekcie macierz X w układzie równań $X \cdot a = y$ nie będzie macierzą kwadratową i wyznaczenie wektora współczynników wielomianu a metodami pokazanymi powyżej nie będzie możliwe. W takim wypadku rozwiązanie układu równań można uzyskać modyfikując trzecią metodę obliczeniową, stosując funkcję odwracania macierzy prostokątnej, jak pokazano na poniższym kodzie.

```
a=pinv(X)*y;
```

Macierz X z równania macierzowego współczynników wielomianu interpolacyjnego określa się na podstawie danych wejściowych (tablicy węzłów interpolacyjnych) i przyjętej liczby współczynników wielomianu. W większości przypadków optymalny poziom interpolacji uzyskuje się dla wielomianu 5-6 rzędu. Kod algorytmu tworzącego tablicę X na podstawie tablicy węzłów interpolacyjnych i przyjętego stopnia wielomianu zapisano poniżej.

```
function X=tablicaX(x,k)
    n=size(x,'c');
    X=[0];
    for i=1:n
        for j=1:k
            X(i,j) = x(i)^(j-1);
        end
    end
endfunction
```

W oparciu o zapisaną powyżej funkcję można zaproponować algorytm wyznaczający współczynniki algebraicznego wielomianu interpolacyjnego i wykreślić przebieg interpolowanych danych wejściowych, jak pokazano poniżej.

```
function fun=wielomian(x,y,c,m,k)
//x,y - tablice danych węzłów interpolacji
//c - kolor krzywej na wykresie interpolacji danych
//m - liczba punktów krzywej interpolacyjnej
```

```

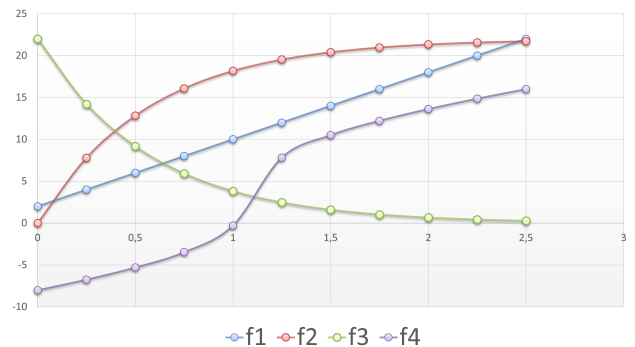
//k - stopień wielomianu
X=tablica(x,k);
a=pinv(A)*(y');
n=size(x,'c');
t=linspace(x(1),x(n),m);
fun=[0];
for i=1:m
    fun(i)=0;
    for j=1:k
        fun(i)=fun(i)+a(j)*t(i)^(j-1);
    end
end
plot2d(t,fun,c);
endfunction

```

Działanie opracowanego algorytmu interpolacyjnego przetestowano na typowych sygnałach dla których dane zestawiona w tabeli 2 i zobrazowano wykresem pokazanym na Rys.8.

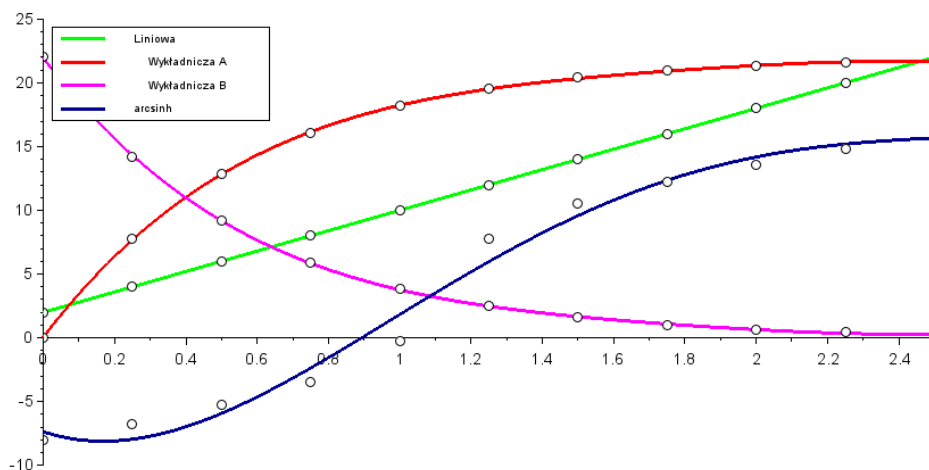
Tab. 2: Tablica węzłów interpolacyjnych

x	f_1	f_2	f_3	f_4
0	2	0	22	-8.02
0.25	4	7.796	14.204	-6.75
0.50	6	12.829	9.171	-5.30
0.75	8	16.079	5.921	-3.48
1.00	10	18.177	3.823	-0.29
1.25	12	19.532	2.468	7.81
1.50	14	20.406	1.594	10.50
1.75	16	20.971	1.029	12.21
2.00	18	21.336	0.664	13.61
2.25	20	21.571	0.429	14.85
2.50	22	21.723	0.277	16.00



Rys. 8: Przebiegi interpolowanych danych

W oparciu o zapisane powyżej funkcje przeprowadzono dla zestawionych w tabeli 2 interpolację algebraiczną z wielomianem rzędu piątego, wyniki pokazano na Rys.9.



Rys. 9: Interpolacja wielomianem rzędu piątego

Zauważyć można na Rys.9 że interpolacja przebiegów podstawowych (wykładniczych i liniowego) została zrealizowana bardzo dobrze, natomiast w przypadku przebiegu funkcji $\text{arcsinh}()$ nie przedstawia zadowalającej zgodności. Przy zwiększeniu stopnia wielomianu pojawiają się znaczne oscylacje na krańcach wykresu. Poprawy nie jest w stanie zapewnić także większa liczba węzłów interpolacyjnych. Znacznie trudniejsza sytuacja występuje przy próbie interpolacji wielomianem algebraicznych funkcji okresowych. W efekcie należy uznać że ta metoda interpolacji nadaje się wyłącznie do funkcji nieokresowych, o stałym znaku pierwszej i drugiej pochodnej.

Ze względu na przytoczone wady interpolacji wielomianem algebraicznym, korzystniejsze może być zastosowanie interpolacji wielomianami **Lagrange'a** lub **Newton'a** które cechują się znacznie lepszym stopniem interpolacji, także w przypadku przebiegów okresowych

3.1.1) Interpolacja Lagrange'a

Metoda Lagrange'a zakłada interpolowanie nieznannej funkcji opisanej węzłami interpolacyjnymi do wielomianu iteracyjnego opisanego wzorem (11).

$$L(x) = \sum_{i=0}^n \left(y_i \cdot \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} \right) \quad (11)$$

Stopień wielomianu Lagrange'a jest zgodny z liczbą węzłów interpolowanego przebiegu. W odróżnieniu od omówionej powyżej metody algebraicznej, algorytm Lagrange'a nie wymaga wyznaczania współczynników algorytmu, jest prostszy dzięki temu obliczeniowo. Przykładową implementację wzoru (11) do funkcji w środowisku Scilab'a zapisano poniżej.

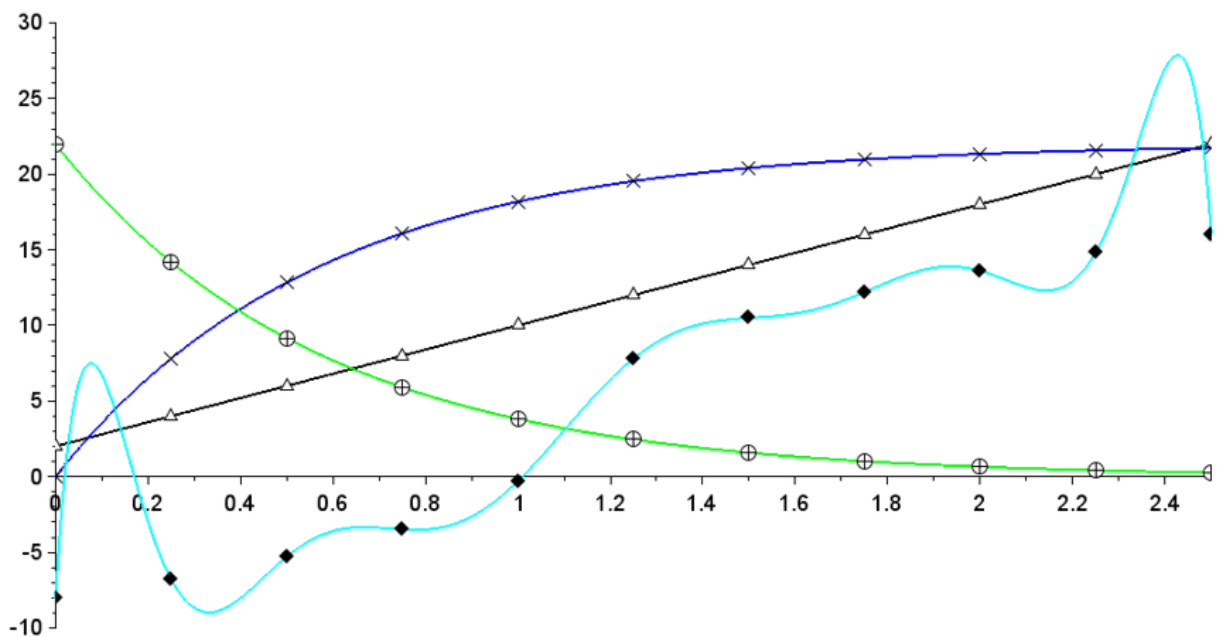
```
function Lag=lagrange(x,y,c,m)
    n=size(x,'c');
```

```

t=linspace(x(1),x(n),m);
Lag=[0];
for i=1:m
    Lag(i)=0
    for j=1:n
        iloczyn=1;
        for k=1:n
            if(k<>j)
                iloczyn=iloczyn*(t(i)-x(k))/(x(j)-x(k));
            end
        end
        Lag(i)=Lag(i)+y(j)*iloczyn;
    end
end
plot2d(t,Lag,c);
endfunction

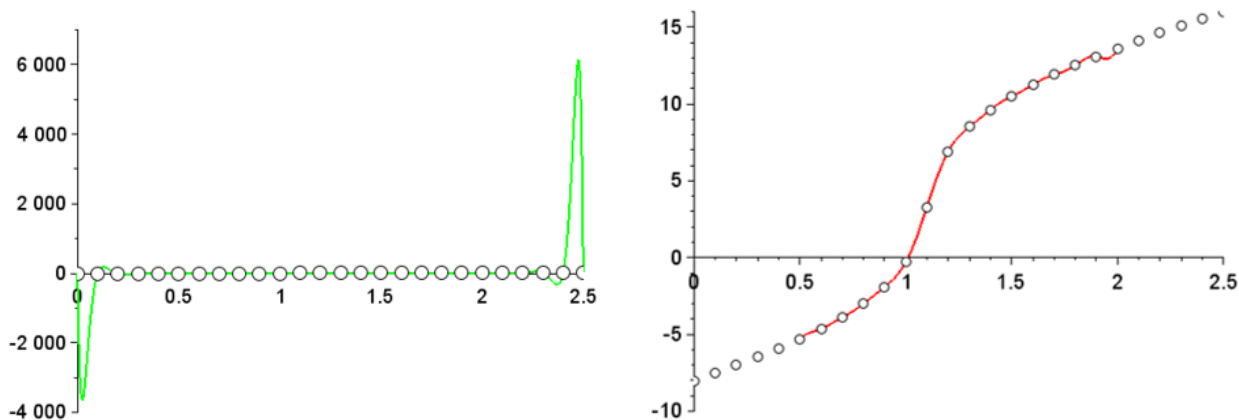
```

Zapisana powyżej funkcja podobnie jak opisany wcześniej wielomian algebraiczny n -tego stopnia dobrze interpoluje proste funkcje liniowe i wykładnicze, jak pokazano na Rys.10.



Rys. 10: Interpolacja wielomianem Lagrange'a

W przypadku funkcji złożonych, przemennych i okresowych także występując odkształcenia, ale w odróżnieniu od wielomianu algebraicznego na obu krańcach przedziału interpolacji. Metoda wraz ze wzrostem liczby węzłów zwraca przebiegi o większej magnitudzie odkształceń na krańcach przedziału, ale jednocześnie większej zgodności interpolacyjnej w środkowej części przedziału, jak pokazano na Rys.11.



Rys. 11: Interpolacja wielomianem Lagrange'a

3.1.2) Interpolacja Newton'a

Metoda interpolacji Newton'a opiera się na wykorzystaniu wyrażeń ilorazowych wyznaczanych na podstawie węzłów opisujących interpolowaną funkcję. Procedura składa się więc z dwóch etapów:

- wyznaczenie tablicy wyrażeń różnicowych, zgodnie z poniższym algorytmem

```
function R=roznicowa(x,f)
    n=size(x,'c');
    R=[0];
    for i=1:n
        R(i,1)=f(i);
    end
    for i=1:n-1
        for j=2:n
            if i+1>j
                continue;
            end
            R(j,i+1) =(R(j,i)-R(j-1,i))/(x(j)-x(j-i));
        end
    end
end
endfunction
```

- obliczenie wartości interpolowanej funkcji dla danego punktu przebiegu na podstawie wyrazów z przekątnej diagonalnej tablicy różnicowej, zgodnie z algorytmem zapisanym w poniższej funkcji

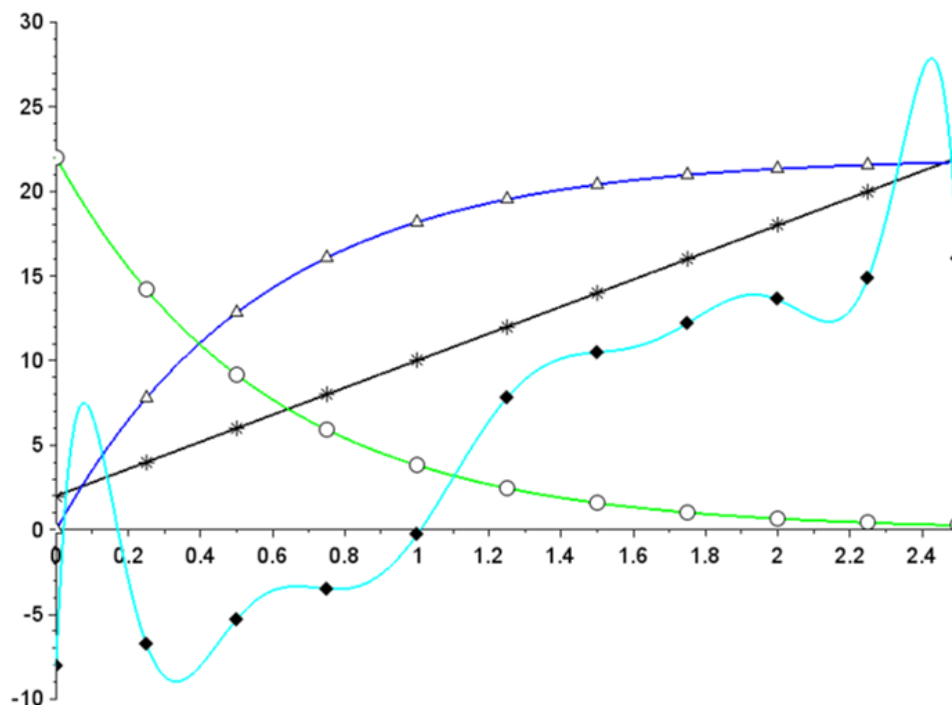
```
function fun=newton(x,f,c,m)
    n=size(x,'c');
    t=linspace(x(1),x(n),m);
    R=roznicowa(x,f)
```

```

fun=[0];
for i=1:m
    suma = 0;
    for j=2:n
        iloczyn = 1;
        for k=1:j-1
            iloczyn = iloczyn * (t(i)-x(k));
        end
        suma = suma + R(j,j)*iloczyn;
    end
    fun(i)=R(1,1)+suma;
end
plot2d(t,fun,c);
endfunction

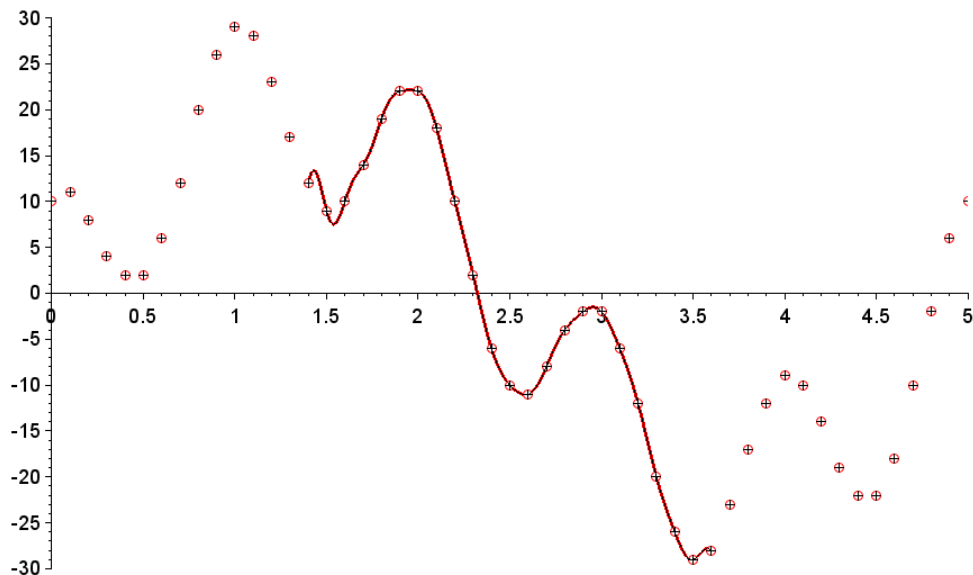
```

Uzyskiwana zgodność przebiegu interpolowanego metodą Newton'a jest na podobnym poziomie dokładności jak uzyskiwana z metody Lagrange'a, jak pokazano na Rys.12.



Rys. 12: Interpolacja wielomianem Newton'a

W przypadku funkcji złożonych, przemiennych i okresowych metoda Newton'a cechuje się podobnym poziomem przegięć na krańcach przedziału interpolacji i zgodności z centralnej części przedziału, jak pokazano na Rys.13.



Rys. 13: Interpolacja sygnału sinusoidalnego, odkształconego metodami Newton'a i Lagrange'a

W przypadku sygnałów w których wartości zmieniają się skokowo, przy interpolacji Lagrange'a występują nieznacznie mniejsze amplitudy oscylacji.

3.2) Szybka transformata Fouriera *FFT()*

W przypadku sygnałów sinusoidalnych i sinusoidalnych odkształconych najlepsze efekty interpolacji daje interpolacja trygonometryczna. Wyznaczanie współczynników szeregu na podstawie tablicy węzłów interpolacyjnych najlepiej przeprowadzić w oparciu o odwrotną transformatę Fouriera. W środowisku *Scilab*'a zrealizować to można za pomocą funkcji `fft()`. Optymalne efekty osiąga się jeżeli analizie podlega okres sygnału lub jego wielokrotność, ewentualnie przy zwiększonej częstotliwości próbkowania.

Atrybutem wejściowym funkcji jest tablica wartości jakie przyjmuje interpolowana funkcji w analizowanym przedziale. Na podstawie otrzymanych wyników i informacji o próbkowaniu interpolowanego sygnału określa się współczynniki szeregu trygonometrycznego, opisanego równaniem (12).

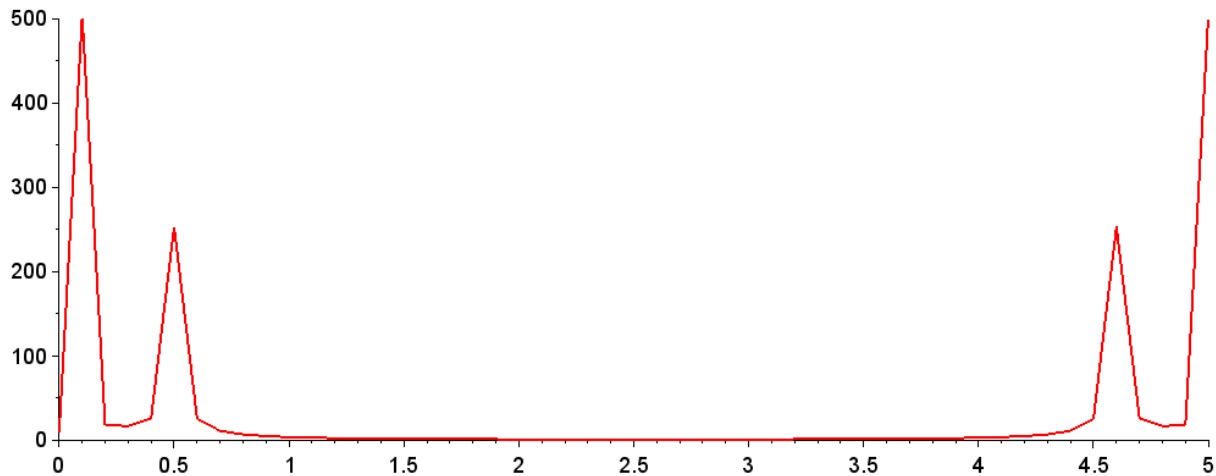
$$F(x) = \sum_{i=1}^n A_i \cdot \sin(f_i \cdot x + \varphi_i) \quad (12)$$

Procedura realizowana jest na podstawie poniższego algorytmu.

1. wywołanie funkcji `fft()` dla tablicy wartości y interpolowanej funkcji

```
fft_y=fft(y);
```

2. na podstawie modułu wartości otrzymanej z funkcji `fft()` można określić liczbę członów szeregu trygonometrycznego, jak pokazano na Rys.14.



Rys. 14: Wykres ekstremów funkcji $fft()$

Ze względu na jego symetryczną budowę do dalszej analizy wykorzystywana jest tylko pierwsza połowa danych z tablicy `fft_y`

3. Współczynniki szeregu trygonometrycznego wyznaczane są na podstawie ekstremów z wykresu z Rys.14. Ich automatyczne wykrycie można zrealizować poprzez wyszukanie indeksów wartości z tablicy `fft_y` których moduły wartości są większe od średniej wartości z tej tablicy

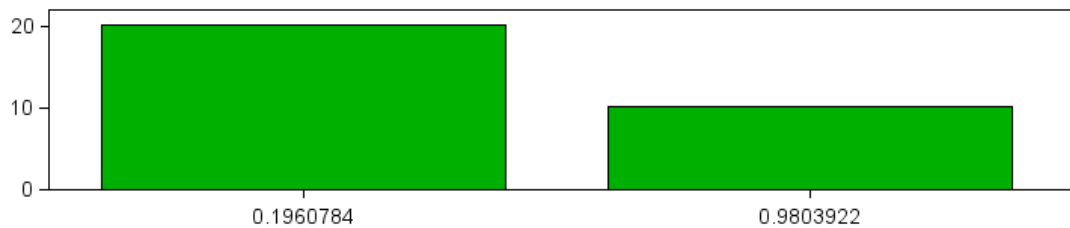
```
for x=1:N2
    modulFFT(x)=abs(fft_y(x));
end
fi=atan(imag(fft_y),real(fft_y));
k1=find(modulFFT>mean(modulFFT));
```

4. Częstotliwość f_i poszczególnych składowych interpolacyjnego szeregu trygonometrycznego oblicza się na podstawie częstotliwości próbkowania, liczby próbek i liczby okresów

```
sample_rate=1/(t(2)-t(1));
czestotliwosc=(sample_rate/(N*okresy))*(0:N2);
f=czestotliwosc(k1);
```

5. Amplitudy składowych szeregu trygonometrycznego A_i określa się na podstawie wektora modułów wartości uzyskanych z funkcji `fft()`, połowy liczby próbek i liczby okresów sygnału

```
A=modulFFT(k1)/(N2*okresy);
```



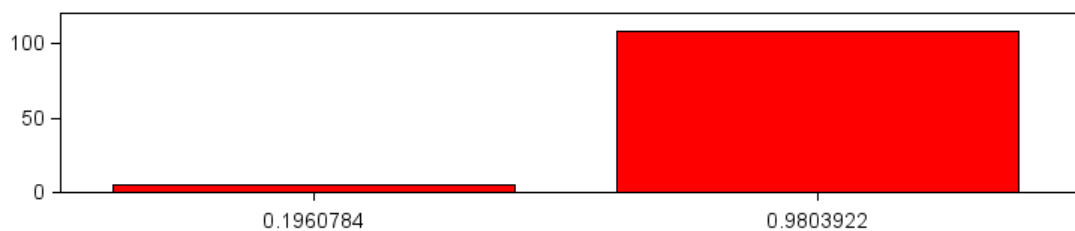
Rys. 15: Wartości amplitudy dla określonych częstotliwości

6. Kąty fazowe φ_i składowych szeregu określa się jako argument wartości zespolonych uzyskanych z funkcji `fft()`

```

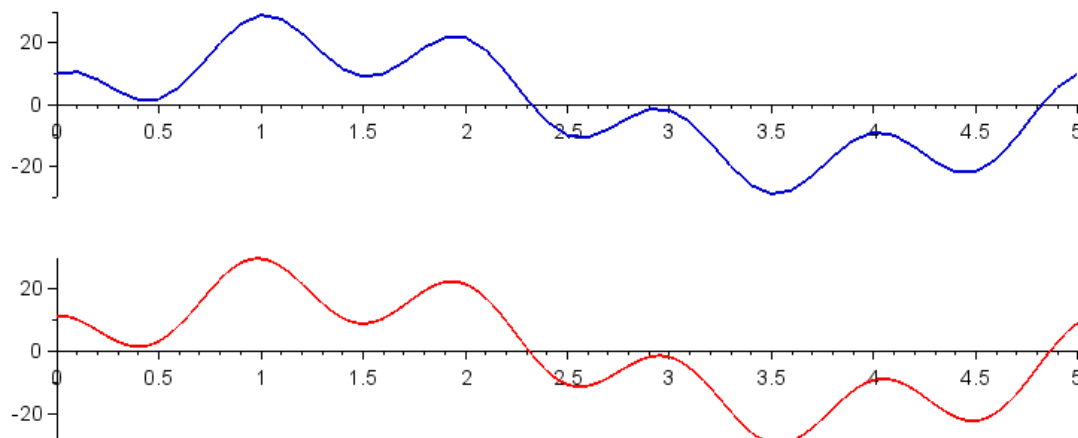
fi=atan(imag(fft_y),real(fft_y));
fi=angle(fft_y); //alternatywne obliczenie fi
faza=fi(k1)+%pi/2;
kat=faza*180/%pi;

```



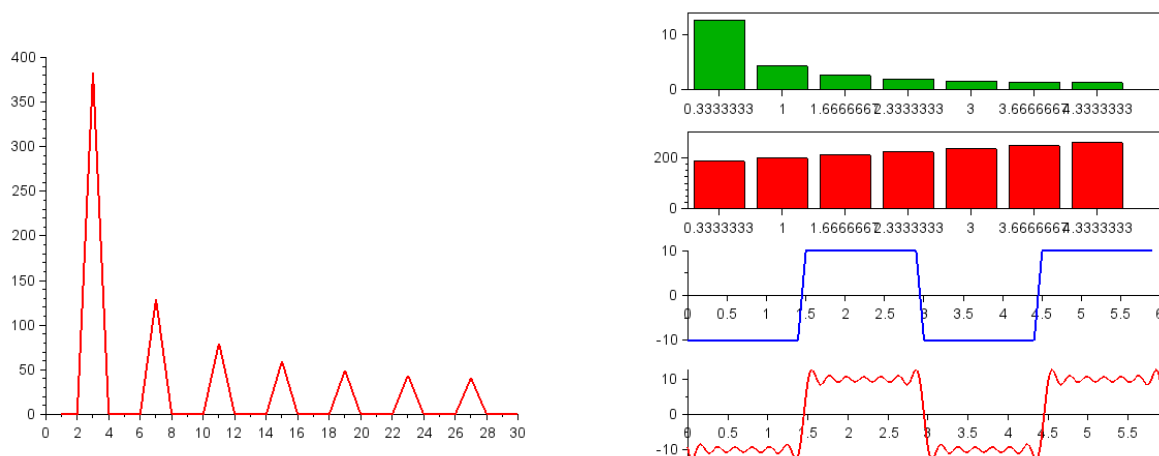
Rys. 16: Wartości kąta fazowego dla określonych częstotliwości

Na podstawie określonych tablic współczynników wielomianu trygonometrycznego możliwe jest wykreślenie interpolacji sygnału, jak pokazano na Rys.17. Sygnał źródłowy został utworzony na podstawie funkcji $F(x)=20\sin(0.2t)+10\cos(t)$.

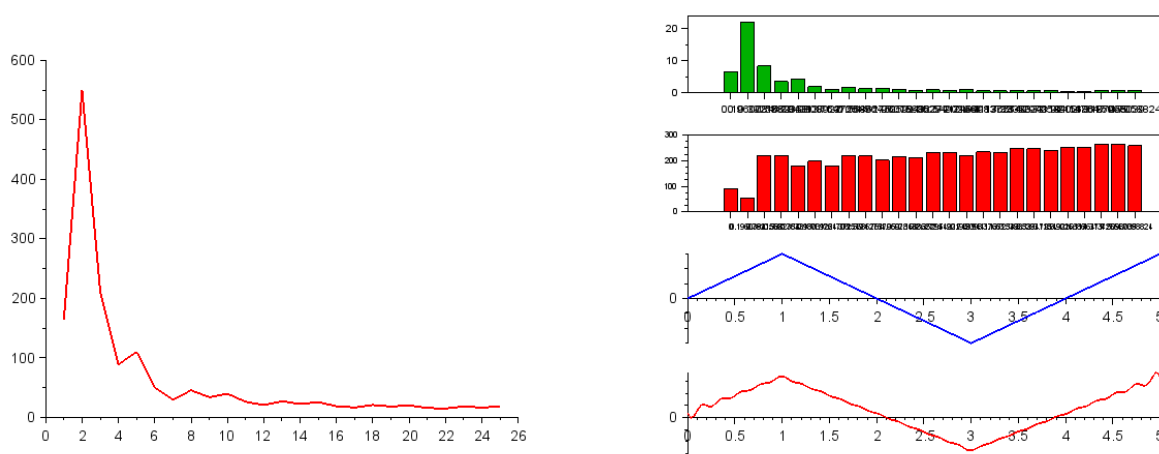


Rys. 17: Przebiegi sygnału źródłowego (niebieski) i jego interpolacji wielomianem trygonometrycznym (czerwony)

Za pomocą wielomianu trygonometrycznego i funkcji `fft()` można interpolować nie tylko funkcje sinusoidalne, jednakże jego efektywność jest znacznie niższa, jak pokazano na Rys.18 i Rys.19. Poprawę efektywności uzyskuje się po zwiększeniu częstotliwości próbkowania, lub zwiększeniu liczby elementów wielomianu trygonometrycznego.



Rys. 18: Interpolacja wielomianem trygonometrycznym sygnału prostokątnego



Rys. 19: Interpolacja wielomianem trygonometrycznym sygnału piło kształtnego

Zauważyć można oscylacje przy skokowych zmianach wartości interpolowanej funkcji oraz na krańcach przedziału interpolacji. W przypadku funkcji nieokresowych metoda nie zapewnia zadowalających wyników.

3.3) Interpolacja funkcjami sklejanymi

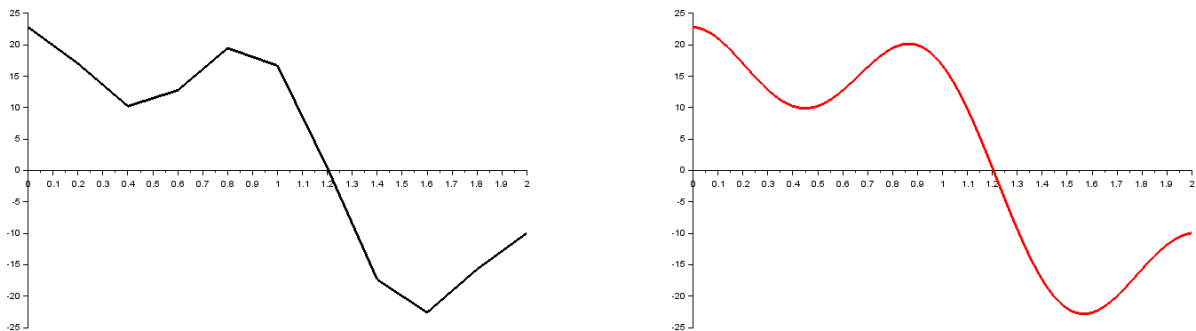
Ze względu na pojawiające się problemy z oscylacjami na krańcach przedziału przy interpolacji wielomianowej optymalnym rozwiązaniem wydaje się interpolowanie prostymi funkcjami przedziałami i następnie sklejenie interpolacji cząstkowych w cały przebieg. Interpolacje funkcjami sklejanymi opiera się na:

- funkcji liniowej

Jest to najprostsza metoda interpolacji splajaniem (z ang. *spline*), w której przedział pomiędzy dwoma węzłami interpolacyjnymi przybliża się przebiegiem liniowym $f(t) = A \cdot t + B$. Proces interpolacji liniowej sprowadza się więc do wyznaczeniu współczynników A i B równań prostych dla poszczególnych przedziałów interpolacji. Algorytm wyznaczania współczynników opiera się na dwóch granicznych węzłach interpolacyjnych $[t(k), F(k)]$ i $[t(k + 1), F(k + 1)]$ analizowanego przedziały, jak pokazano poniżej na równaniu (13).

$$\begin{cases} A[i] = \frac{F(k+1)-F(k)}{t(k+1)-t(k)} \\ B[i] = F(k) - A[i] \cdot t(k) \end{cases} \quad (13)$$

Liczba przedziałów jest o jeden mniejsza od liczby węzłów. Dla każdego z nich określonego przez węzły n i $n+1$ wyznaczane jest równanie prostej przechodzącej przez te punkty. W oparciu o wyznaczone tablice współczynników równań liniowych można interpolować przedziałami analizowane dane. Metoda daje dobre wyniki, gdy do dyspozycji jest duża liczba węzłów, jak pokazano na Rys.20.



Rys. 20: Interpolacja funkcjami sklejanyymi I stopnia dla 11 i 201 węzłów

- wielomianie trzeciego stopnia - splajan

Znacznie lepszą zgodnością wykreślanego przebiegu charakteryzuje się interpolacja funkcjami sklejanyymi wielomianów trzeciego rzędu. W metodzie tej przebieg funkcji interpolowanej jest do krzywej wielomianu opisanej równaniem w postaci ogólnej $f(t) = A \cdot t^3 + B \cdot t^2 + C \cdot t + D$.

W podstawowym podejściu do problemu na granicach przedziałów interpolacji mogą pojawić się tak zwane kanty. Metoda nie zapewni ciągłości przebiegu interpolowanej funkcji. Kryterium pozwalającym na zapewnienie ciągłości interpolacji jest zapewnienie zgodności wartości drugich pochodnych na granicach przedziałów interpolacji. W celu spełnienia powyższego kryterium należy określić dla każdego z przedziałów interpolacji współczynniki równania (14).

$$y_i(x) = \mathbf{a} \cdot f_i + \mathbf{b} \cdot f_{i+1} + \mathbf{c} \cdot f_i'' + \mathbf{d} \cdot f_{i+1}'' \quad (14)$$

Określenie współczynników realizowane jest na podstawie dwóch kolejnych węzłów interpolacji oraz wartości II pochodnych w tych punktach. Współczynniki występujące w równaniu (14) wyznaczone są z równań zapisanych poniżej.

$$\mathbf{a} = \frac{x_{i+1} - x}{x_{i+1} - x_i} \quad (15)$$

$$\mathbf{b} = \frac{x - x_{i+1}}{x_{i+1} - x_i} \quad (16)$$

$$\mathbf{c} = \frac{1}{6} \cdot (\mathbf{a}^3 - \mathbf{a}) \cdot (x_{i+1} - x_i)^2 \quad (17)$$

$$\mathbf{d} = \frac{1}{6} \cdot (\mathbf{b}^3 - \mathbf{b}) \cdot (x_{i+1} - x_i)^2 \quad (18)$$

Podstawowym problemem z zastosowaniem metody interpolacji splajnem jest konieczność posiadania informacji o wartości II pochodnej dla każdego z węzłów interpolacji. Problem ten można rozwiązać numerycznie zakładając ciągłość I pochodnej we wszystkich węzłach. W efekcie dla każdego przedziału interpolacji można zapisać równanie.

$$\frac{x_i - x_{i+1}}{6} \cdot f''_{i-1} + \frac{x_{i+1} - x_{i-1}}{3} \cdot f''_i + \frac{x_{i+1} - x_i}{6} \cdot f''_{i+1} = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} - \frac{f_i - f_{i-1}}{x_i - x_{i-1}} \quad (19)$$

Na podstawie równania (20) dla całego przedziału interpolacji można zapisać układ równań pozwalający na wyznaczenie II pochodnych w poszczególnych węzłach interpolacji. Niestety metoda zapewnia n-2 równań dla n niewiadomych, co powoduje że dla krańcowych punktów (f''_1 i f''_n) konieczne jest przyjęcie jakiś wartości II pochodnych. Zazwyczaj ustawienie wartości 0 dla II pochodnych krańcowych węzłów przedziału daje dobre wyniki, ale można ustawić dowolne wartości.

Zaleca się aby próbkowanie węzłów interpolacyjnych realizowane było z stałym krokiem $x_{i+1} - x_i = h = const$. Znacząco upraszcza to algorytm obliczeń, a układ równań przyjmuje wtedy postać układu zapisaną poniżej.

$$Z \cdot F'' = Y$$

$$\begin{bmatrix} 4 & 1 & 0 & \dots & 0 & 0 \\ 1 & 4 & 1 & \dots & 0 & 0 \\ 0 & 1 & 4 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 4 & 1 \\ 0 & 0 & 0 & \dots & 1 & 4 \end{bmatrix} \cdot \begin{bmatrix} f''_2 \\ f''_3 \\ f''_4 \\ \vdots \\ f''_{n-2} \\ f''_{n-1} \end{bmatrix} = \begin{bmatrix} 6 \cdot \frac{f_1 - 2 \cdot f_2 - f_3}{h^2} \\ 6 \cdot \frac{f_2 - 2 \cdot f_3 - f_4}{h^2} \\ 6 \cdot \frac{f_3 - 2 \cdot f_4 - f_5}{h^2} \\ \vdots \\ 6 \cdot \frac{f_{n-3} - 2 \cdot f_{n-2} - f_{n-1}}{h^2} \\ 6 \cdot \frac{f_{n-2} - 2 \cdot f_{n-1} - f_n}{h^2} \end{bmatrix} \quad (20)$$

Procedura interpolacji wielomianem III stopnia realizowana jest więc dwustopniowo:

- Wyznaczenie wartości drugich pochodnych w węzłach interpolacyjnych $\{2, n-1\}$, przy założonych wartościach pochodnej na krańcach przedziału interpolacji $f_1'' = 0$ i $f_n'' = 0$. Wartości wyznacza się rozwiązując układ równań (20) wykorzystując jedną z poznanych metod rozwiązywania układów równań liniowych (skończonych, iteracyjnych, a w przypadku środowisk obliczeniowych funkcji wbudowanych np. *linsolve()*). W zależności od liczby węzłów interpolacyjnych należy zdefiniować macierz trójdziagonalną \mathbf{Z} , oraz wektor \mathbf{Y} . Wektor wartości II pochodnych można wtedy wyznaczyć za pomocą funkcji *linsolve()*:

$$F2=linsolve(Z, -Y);$$

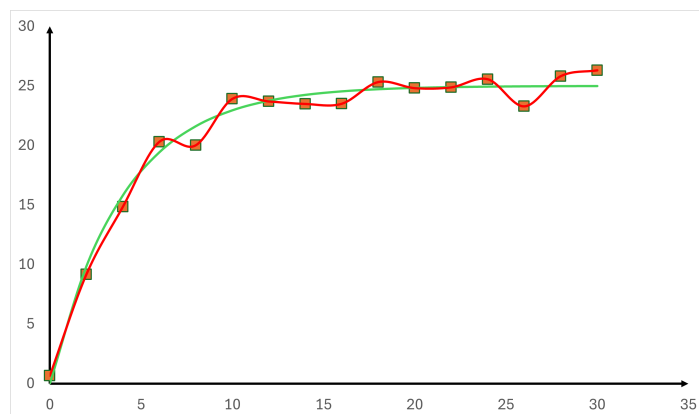
Należy jednak pamiętać że do kolejnego etapu konieczne jest dodanie krańcowych wartości II pochodnej dla punktów pierwszego f_1'' i n-tego f_n'' .

- W oparciu o równanie (14) wyznacza się wartości interpolujące daną funkcję w poszczególnych przedziałach określonych przez węzły interpolacji. W każdym z takich przedziałów określić można dowolną liczbę punktów interpolacyjnych.

4) Aproksymacja

W odróżnieniu od interpolacji, aproksymacja przybliża przebieg zmian opisany tablicą danych funkcją z możliwie jak największą dokładnością. Stosuje się ją gdy nie jest znany zapis matematyczny wiążący analizowane dane, lub wartości danych są obarczone błędem powodującym że ich interpolacja obciążona by była znaczącym odchyleniem od rzeczywistego przebiegu. Metoda pozwala na wyznaczenie, możliwie niskim kosztem numerycznym, wartości współczynników funkcji która w sposób przybliżony opisuje przetwarzane dane.

W uproszczeniu w wykres przebiegu funkcji interpolującej przechodzi przez wszystkie węzły, natomiast przebieg funkcji aproksymującej przebiega możliwie blisko węzłów, jak pokazano na Rys.21.



Rys. 21: Przebiegi funkcji interpolującej i aproksymującej dane

Przeprowadzenie procesu aproksymacji danych, argumentów funkcji bazowej $F(x)$, wymagana jest ich wcześniejsza analiza i określenie równania funkcji $f(x)$ której przebieg zapewnia jak największą zgodność z nimi. Analizowany jest więc warunek normy zapisany równaniem (21) dla ustalonej wartości dopuszczalnego błędu.

$$\|F(x) - f(x)\| \leq \varepsilon \quad (21)$$

Równanie funkcji aproksymującej w postaci uogólnionej można więc zapisać zależnością pokazaną w równaniu (22).

$$f(x) = a_0 \cdot \varphi_0(x) + a_1 \cdot \varphi_1(x) + \dots + a_n \cdot \varphi_n(x) \quad (22)$$

gdzie:

a_0, a_1, a_n : są współczynnikami funkcji aproksymacyjnej

$\varphi_0, \varphi_1, \varphi_n$: są funkcjami bazowymi aproksymacji

Dobór funkcji bazowych ma decydujące znaczenie dla prawidłowości i efektywności aproksymacji. Zadanie aproksymacji przy wybranych funkcjach bazowych sprowadza się więc do dobrania wartości poszczególnych współczynników a_n tak aby spełniony był warunek określony zależnością zapisaną równaniem (21). Funkcjami bazowymi są zazwyczaj wielomiany lub funkcje trygonometryczne dla sygnałów okresowych.

Wyznaczenie współczynników równania aproksymacyjnego najczęściej realizuje się w oparciu o aproksymacje jednostajną lub średniokwadratową. Aproksymacja jednostajna przykładu bardzo dużą wagę do dużych błędów i w ogóle nie zajmuje się jakością przybliżenia w innych punktach. Z tego powodu jest rzadziej używana w praktyce. Aproksymacja średniokwadratowa bardzo dobrze eliminuje błędy pomiarowe danych przybliżając funkcję $f(X)$ do rzeczywistej funkcji bazowej $F(x)$.

Aproksymacja średniokwadratowa sprowadza się do wyznaczenia współczynników funkcji aproksymującej takich aby osiągnięte była najmniejsze maksimum różnicy między $f(x)$ a $F(x)$ z zależności opisanej równaniem (23).

$$S(a_0, \dots, a_n) = \sum_{i=1}^k (F(x_i) - f(x_i))^2 \quad (23)$$

Wyznaczenie wartości współczynników spełniających założone kryterium sprowadza się do wyznaczenia miejsc zerowych pierwszych pochodnych równania (23) dla każdego z współczynników. Przy założeniu aproksymacji danych funkcją liniową $f(x) = a_1 \cdot x + a_0$ należy rozwiązać układ równań (24) dla zmiennych a_0 i a_1 .

$$\frac{\partial S(a_0, a_1)}{\partial a_0} = 2 \cdot \sum_{i=1}^n (y_i - (a_1 \cdot x_i + a_0)) \cdot (-1) = 0 \quad (24)$$

$$\frac{\partial S(a_0, a_1)}{\partial a_1} = 2 \cdot \sum_{i=1}^n (y_i - (a_1 \cdot x_i + a_0)) \cdot (-x_i) = 0$$

Po przekształceniu powyższą zależność można zapisać w postaci:

$$n \cdot a_0 + a_1 \cdot \sum_{i=1}^n x_i = \sum_{i=1}^n y_i \quad (25)$$

$$a_0 \cdot \sum_{i=1}^n x_i + a_1 \cdot \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i \cdot y_i$$

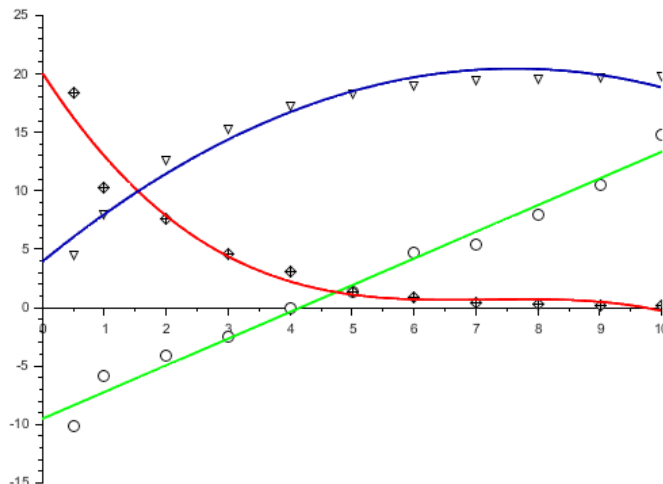
Co można zapisać w postaci równania macierzowego, jak pokazano poniżej.

$$2 \cdot \begin{vmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{vmatrix} \begin{vmatrix} a_0 \\ a_1 \end{vmatrix} = \begin{vmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{vmatrix} \quad (26)$$

Implementując dowolną metodę numeryczną rozwiązywania układ równań liniowych $Ax = b$ można wyznaczyć wartości współczynników a_0 i a_1 . Algorytm ten można w prosty sposób rozbudować w celu wyznaczenia współczynników wielomianu aproksymującego dowolnego rzędu, jak pokazano w poniższym równaniu dla wielomianu rzędu k .

$$(k+1) \cdot \begin{vmatrix} n & \sum_{i=1}^n x_i & \dots & \sum_{i=1}^n x_i^k \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \dots & \sum_{i=1}^n x_i^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^n x_i^k & \sum_{i=1}^n x_i^{k+1} & \dots & \sum_{i=1}^n x_i^{2k} \end{vmatrix} \begin{vmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{vmatrix} = \begin{vmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \\ \vdots \\ \sum_{i=1}^n x_i^k y_i \end{vmatrix} \quad (27)$$

Z wykorzystaniem algorytmu opisanego równaniem (27) przeprowadzono aproksymacje trzech zestawów danych jak pokazano na Rys 22. Zastosowano aproksymacje do wielomianu rzędu pierwszego, drugiego i trzeciego.



Rys. 22: Aproksymacja średniokwadratowa funkcją wielomianową trzech zestawów danych pomiarowych

Omówiony powyżej algorytm można zastosować także dla funkcji nieliniowych przeskalowując problem do funkcji liniowej za pomocą dedykowanego podstawienia.

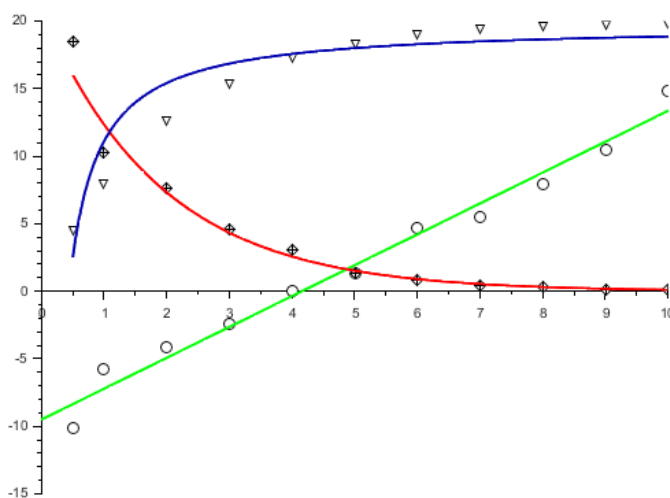
Zauważyć można na Rys. 22 że krzywe czerwona i niebieska należało by aproksymować funkcjami nieliniowymi, odpowiednio dla czerwonej równaniem (28) i niebieskiej równaniem (29).

$$\begin{aligned}
 y &= a_0 \cdot e^{a_1 \cdot x} \\
 \ln(y) &= \ln(a_0) + \ln(e^{a_1 \cdot x}) \\
 \ln(y) &= \ln(a_0) + a_1 \cdot x \\
 \begin{cases} Y &= \ln(y) \\ A_0 &= \ln(a_0) \end{cases}
 \end{aligned} \tag{28}$$

$$Y = A_0 + a_1 \cdot x$$

$$\begin{aligned}
 y &= \frac{a_1}{x} + a_0 \\
 X &= \frac{1}{x} \\
 y &= a_1 \cdot X + a_0
 \end{aligned} \tag{29}$$

Po przeliczeniu danych wejściowych zgodnie z przyjętymi w równaniach (28 i 29) założeniami można przeprowadzić aproksymację funkcjami nieliniowymi uzyskując przebiegi jak pokazano na Rys. 23.



Rys. 23: Aproksymacja średniokwadratowa funkcją wielomianową i funkcjami nieliniowymi zestawów danych pomiarowych

Zadania

Na podstawie informacji zawartych w niniejszej instrukcji i wiadomościach literaturowych stosując wybrane metody interpolacyjne i aproksymacyjne wykreślić przebiegi prądu i napięcia na kondensatorze w obwodzie szeregowym RLC załączanym na napięcie stałe. Rozpatrzyć przypadek oscylacyjny i aperiodyczny. W sprawozdaniu należy:

1. Studium wyboru równania funkcji aproksymacyjnej

2. Wyprowadzenie równań przeskalowujących równanie nieliniowe do liniowego
3. Zestawienie wartości współczynników
4. Studium wyboru metody interpolacyjnej
5. Wykresy pokazujące węzły danych (interpolacji i aproksymacji) oraz przebiegi interpolowane i aproksymowane
6. Wnioski