

Numeryczne przetwarzanie danych

Wstęp

Dane z pomiarów i symulacji często wymagają przetworzenia przed analizą. W ćwiczeniu omówiono różniczkowanie i całkowanie numeryczne, interpolację oraz aproksymację danych [1, 2].

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 [1, 2].

- 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 metody zależy od właściwości funkcji i danych. Krok Δx powinien równoważyć błąd obcięcia i błędy zaokrągleń; zbyt mały krok nie zawsze poprawia wynik.

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

- analiza danych eksperymentalnych $\rightarrow$ 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 $\rightarrow$ 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 $\rightarrow$ 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 $\rightarrow$ w numerycznym rozwiązywaniu równań różniczkowych często pojawia się konieczność obliczania pochodnych. Metody różniczkowania numerycznego mogą być używane do tego celu.

Różniczkowanie numeryczne danych dyskretnych zwraca tablicę przybliżonych wartości pochodnej, a nie jej wzór analityczny.

W środowisku **Scilab** funkcja `diff()` wyznacza różnice kolejnych elementów wektora lub macierzy. Przyjmuje jeden argument obowiązkowy i argumenty opcjonalne [3]:

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

gdzie:

- `y` — wektor lub macierz różnic;
- `f` — wejściowy wektor lub macierz;
- `n` — rząd różnicy (domyślnie pierwszy);
- `"typ"` — kierunek działania dla macierzy: `"r"`, `"c"` albo `"*"`.

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.

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

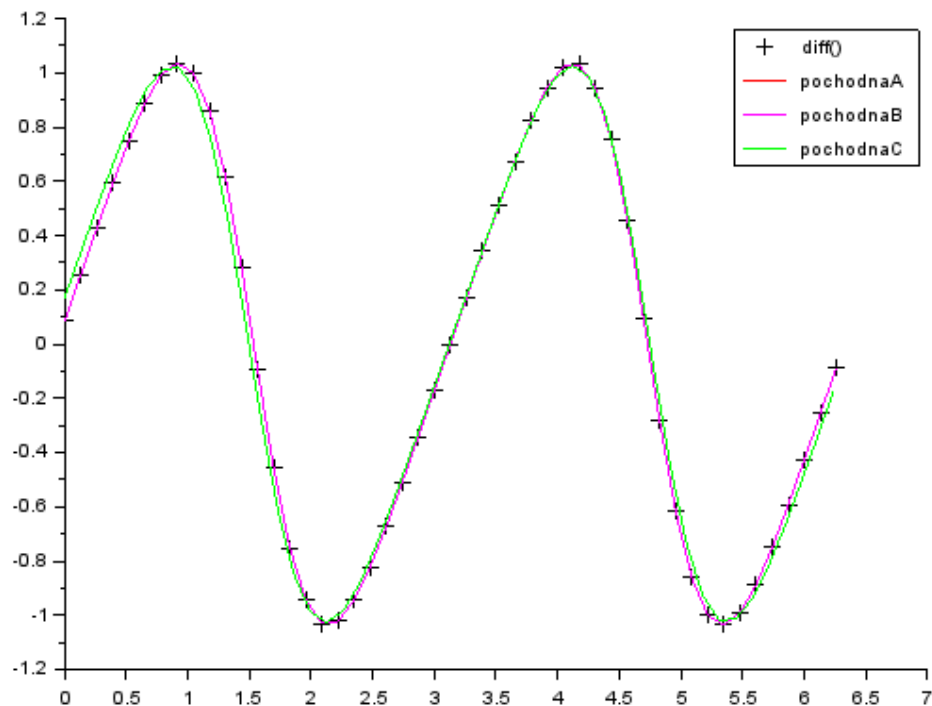
```
clear;clc;clf;
function y=forward(t,f)
    n=size(t)(2);
    for i=1:n-1
        y(i)=(f(i+1)-f(i))/(t(i+1)-t(i));
    end
endfunction
function y=backwards(t,f)
    n=size(t)(2);
    for i=2:n
        y(i-1)=(f(i)-f(i-1))/(t(i)-t(i-1));
    end
endfunction
function y=central(t,f)
    n=size(t)(2);
```

```

    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 ryc. 1.



Ryc. 1. Przebiegi zróżniczkowanej funkcji

2 Całkowanie numeryczne

Metody numerycznego całkowania wyznaczają przybliżoną wartość całki oznaczonej [1, 2]. W środowisku **Scilab** można użyć funkcji `integrate()` [3]:

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

gdzie:

- 'funkcja' — wyrażenie opisujące funkcję podcałkową;
- 'zmienna' — nazwa zmiennej całkowania;
- start i koniec — granice całkowania.

Przykład porównania funkcji wbudowanej z metodą trapezów:

```
f_txt='8*sin(0.5*x)-5*sin(3*x)-3*cos(2*x+%pi/3) ...  
      -0.15*x^2+1.5*x+6';  
I_ref=integrate(f_txt,'x',0,15);  
x=0:0.1:15;  
y=8*sin(0.5*x)-5*sin(3*x)-3*cos(2*x+%pi/3) ...  
  -0.15*x.^2+1.5*x+6;  
I_trap=sum((y(1:$-1)+y(2:$)).*diff(x)/2);  
mprintf("integrate: %.8f, trapezy: %.8f\n",I_ref,I_trap);
```

Proste kwadratury zastępują całkę sumą ważoną wartości funkcji w podprzedziałach. Krok wpływa na błąd dyskretyzacji, ale jego nadmierne zmniejszanie zwiększa koszt i może ujawnić wpływ błędów zaokrągleń.

Sposób postępowania zależy od reprezentacji funkcji. Dla wzoru analitycznego punkty kwadratury można dobierać swobodnie. Dla danych tabelarycznych są one narzucone przez próbkowanie; zmianę siatki należy poprzedzić uzasadnioną interpolacją.

Podstawowe metody różnią się funkcją przybliżającą całkowany przebieg w podprzedziale: stałą, prostą lub wielomianem wyższego stopnia.

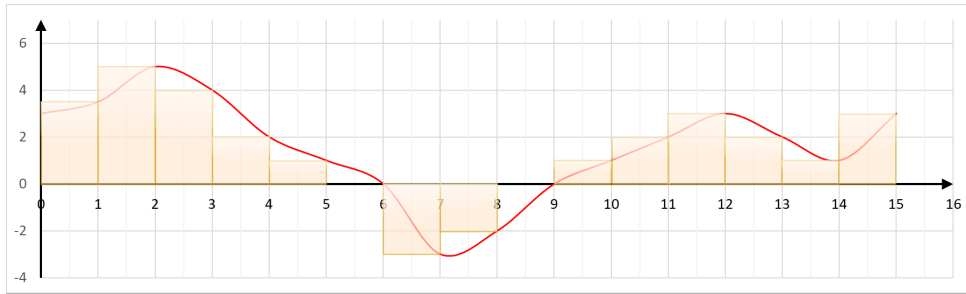
2.1 Metoda prostokątów

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.

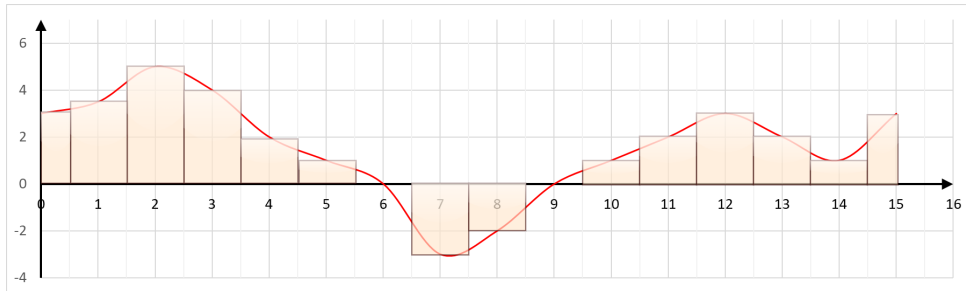
Metoda występuje w trzech wariantach, w zależności od sposobu przybliżania do stałej dla kroku obliczeniowego:

- lewych prostokątów $\rightarrow \langle t - \Delta t, t \rangle$, jak pokazano na ryc. 2
- środkowych prostokątów $\rightarrow \langle t - \frac{\Delta t}{2}, t + \frac{\Delta t}{2} \rangle$, jak pokazano na ryc. 3
- prawych prostokątów $\rightarrow \langle t, t + \Delta t \rangle$, jak pokazano na ryc. 4

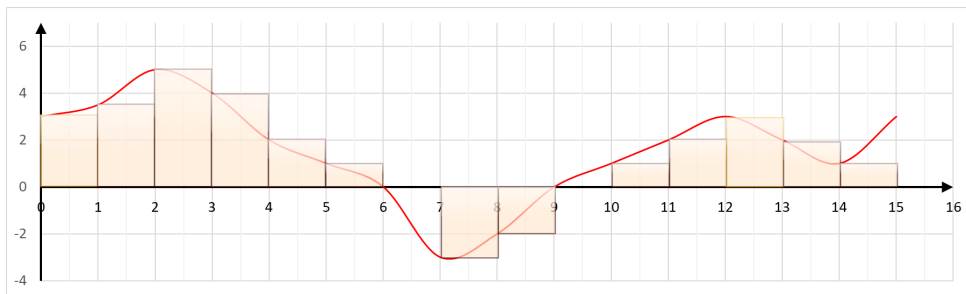
,gdzie Δt jest krokiem obliczeń.



Ryc. 2. Metoda prostokątów (lewych)



Ryc. 3. Metoda prostokątów (środkowych)



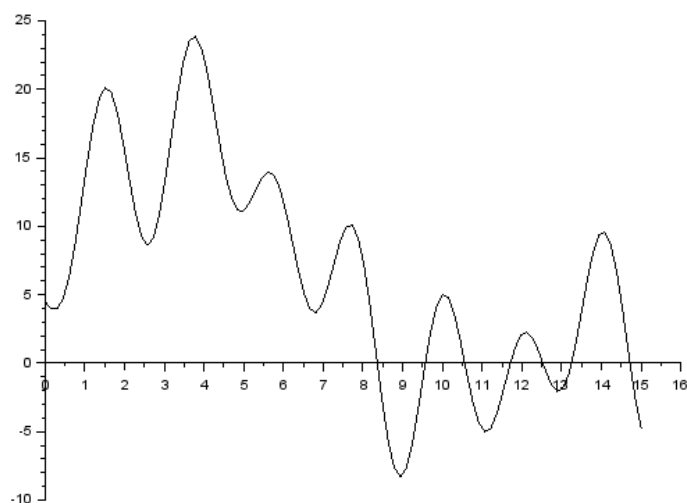
Ryc. 4. Metoda prostokątów (prawych)

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 wariantu metody prostokątów należy odpowiednio dobrać punkty, w których oblicza się wartości funkcji w sumie (2). Porównanie wykonano dla funkcji (3), pokazanej na ryc. 5, w przedziale (0, 15).

$$f(x) = 8 \sin(0,5x) - 5 \sin(3x) - 3 \cos\left(2x + \frac{\pi}{3}\right) - 0,15x^2 + 1,5x + 6 \quad (3)$$



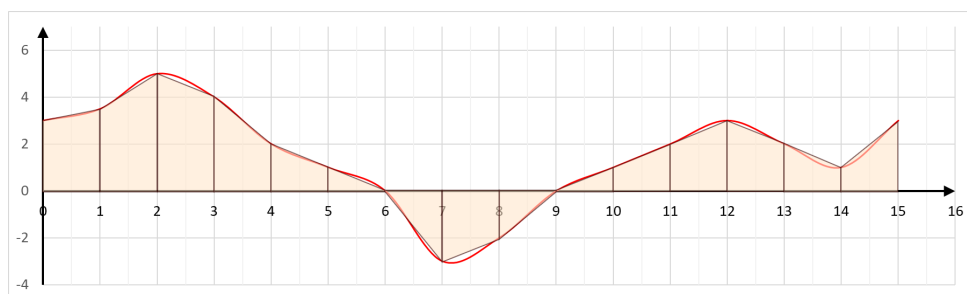
Ryc. 5. Przebieg funkcji podcałkowej

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 ryc. 6.



Ryc. 6. Metoda trapezów

Metoda wykorzystuje wartości funkcji na końcach podprzedziału. Niezależnie od sposobu reprezentacji danych wykonuje się te same działania. Złożoną metodę trapezów opisuje równanie (4).

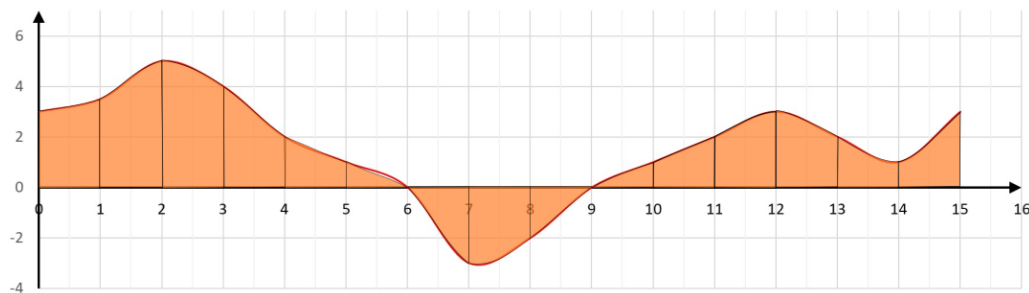
$$\int_a^b f(x)dx = \sum_{n=a}^{b-\Delta x} (f(n + \Delta x) + f(n)) \cdot \frac{\Delta x}{2} \quad (4)$$

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 ryc. 7.



Ryc. 7. Metoda Simpsona

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 $-0,008\%$
 - $\Delta t = 0,1 \rightarrow 101,50239$ z błędem $-0,0000005\%$

Gdy funkcja podcałkowa jest opisana tablicą danych, dla reguły Simpsona 1/3 liczba podprzedziałów musi być parzysta, a dla reguły 3/8 — podzielna przez trzy. Algorytmy korzystają odpowiednio z kolejnych trzech lub czterech punktów:

- 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 $d(n)$ oznacza wartości funkcji w węzłach. Różne współczynniki wynikają z tego, że pojedynczy panel reguły obejmuje odpowiednio dwa lub trzy podprzedziały o szerokości Δx .

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 zestandaryzowanej funkcji na podstawie zebranych danych (węzłów interpolacyjnych) poprzez wyznaczenie numeryczne współczynników funkcji interpolacyjnej. Zazwyczaj ma ona postać wielomianu algebraicznego lub trygonometrycznego. Cechą charakterystyczną interpolacji jest przyjmowanie przez funkcję interpolującą takich samych wartości w węzłach interpolacyjnych [1, 2].

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 węzeł jest opisany współrzędnymi (x_k, y_k) . Dla $n + 1$ różnych węzłów macierz Vandermonde'a X jest kwadratowa. Współczynniki najbezpieczniej wyznaczyć bez jawnego odwracania macierzy:

```
a=X\y;
```

Macierze Vandermonde'a mogą być źle uwarunkowane, zwłaszcza dla wielu równoodległych węzłów. Wielomian niższego stopnia nie przechodzi na ogół przez wszystkie punkty i stanowi aproksymację średniokwadratową, a nie interpolację. Dla prostokątnej macierzy X można użyć operatora $\backslash$:

```
a=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
```

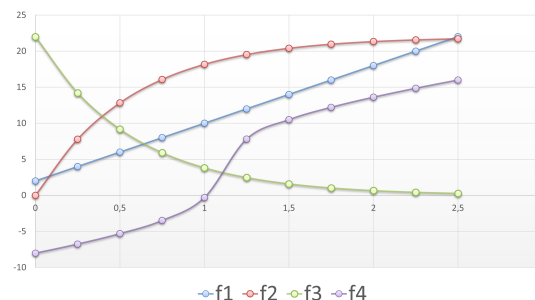
W oparciu o zapisaną powyżej funkcję można zaproponować algorytm wyznaczający współczynniki algebraicznego wielomianu interpolacyjnego i wykreślić przebieg interpolowanych danych wejściowych, jak pokazano poniżej.

```
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=tablicaX(x,k+1);
a=X\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 algorytmu przetestowano na danych zestawionych w tab. 2 i pokazanych na ryc. 8.

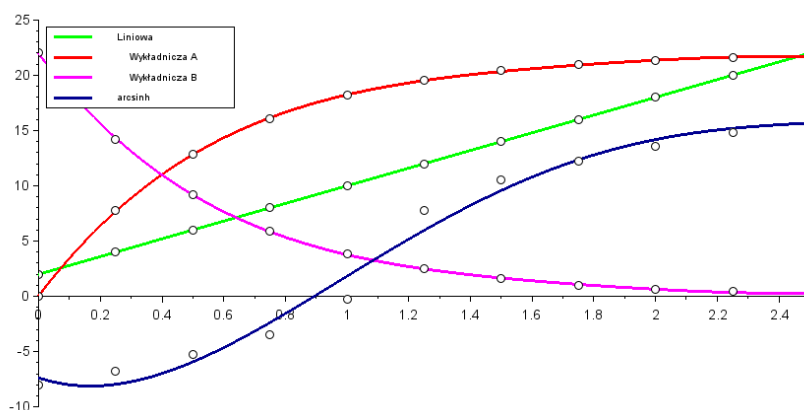
Tab. 2. Tablica węzłów interpolacyjnych

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



Ryc. 8. Przebiegi interpolowanych danych

Na podstawie danych z tab. 2 wykonano przybliżenie wielomianem piątego stopnia; wyniki pokazano na ryc. 9.



Ryc. 9. Interpolacja wielomianem rzędu piątego

Na ryc. 9 widać, że jakość przybliżenia zależy od funkcji i rozmieszczenia węzłów. Zwiększanie stopnia wielomianu dla równoodległych węzłów może powodować oscylacje na krańcach przedziału (zjawisko Rungego). Nie oznacza to jednak, że interpolacja wielomianowa nadaje się wyłącznie do funkcji o stałym znaku pochodnych; istotne są także dobór węzłów i użyta postać wielomianu.

Postacie Lagrange'a i Newtona opisują ten sam jednoznaczny wielomian interpolacyjny, dlatego nie usuwają zjawiska Rungego. Różnią się sposobem obliczeń; dla wielu punktów korzystniejsze bywają funkcje sklejane lub węzły Czebyszewa.

3.1.1 Interpolacja Lagrange'a

Postać Lagrange'a wielomianu interpolacyjnego jest opisana wzorem (11).

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

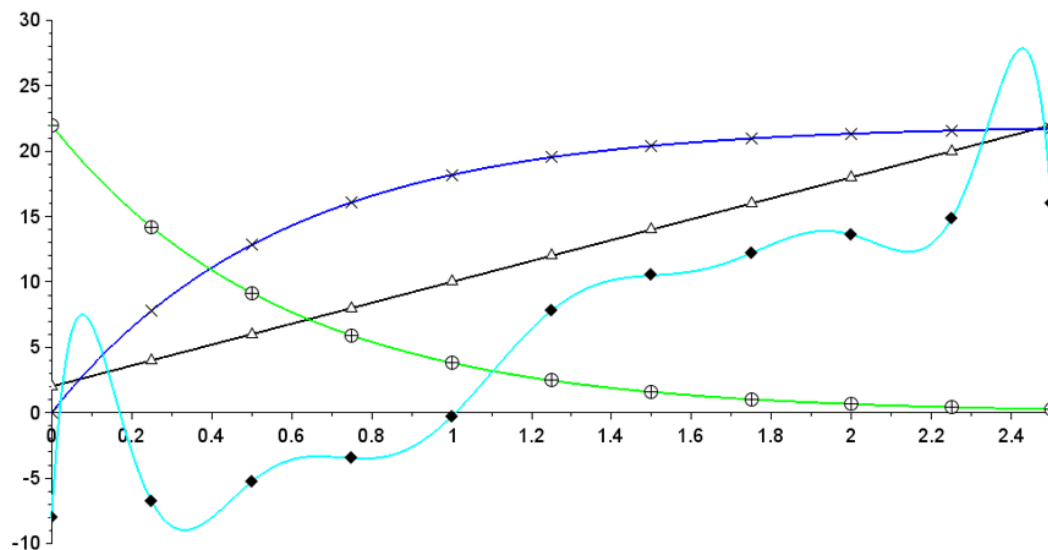
Dla $n+1$ różnych węzłów stopień wielomianu jest nie większy niż n . Postać Lagrange'a pozwala obliczać jego wartości bez rozwiązywania układu z macierzą Vandermonde'a. Przykładowa implementacja wzoru (11) ma postać:

```

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

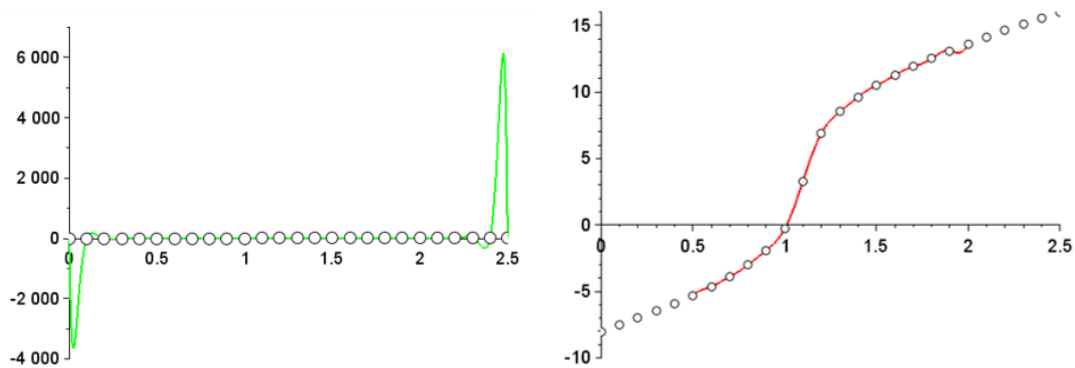
```

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



Ryc. 10. Interpolacja wielomianem Lagrange'a

Dla wielu równoodległych węzłów mogą wystąpić oscylacje na krańcach przedziału, jak pokazano na ryc. 11.



Ryc. 11. Interpolacja wielomianem Lagrange'a

3.1.2 Interpolacja Newtona

Postać Newtona wykorzystuje ilorazy różnicowe wyznaczone na podstawie węzłów. Procedura składa się 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
endfunction
```

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

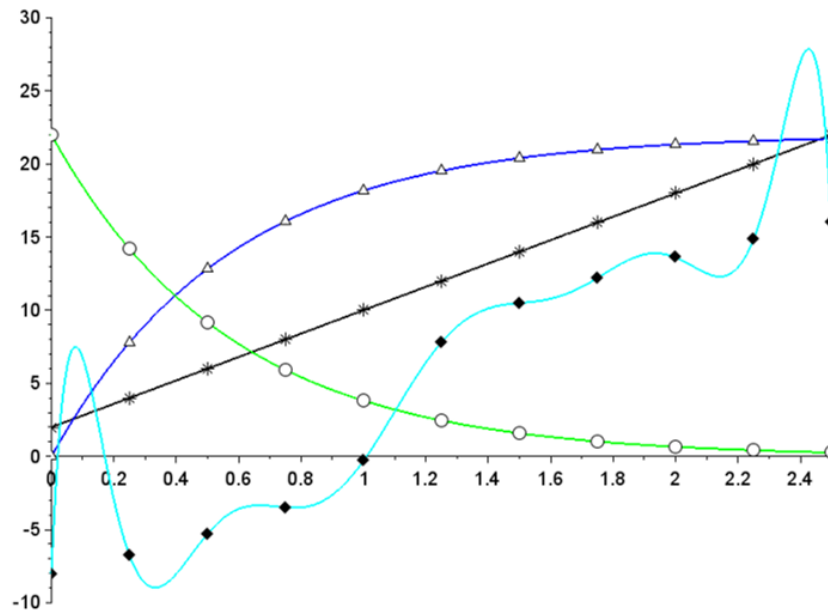
```
function fun=newton(x,f,c,m)
    n=size(x,'c');
    t=linspace(x(1),x(n),m);
    R=roznicowa(x,f)
    fun=[0];
    for i=1:m
        suma = 0;
        for j=2:n
            iloczyn = 1;
            for k=1:j-1
                iloczyn = iloczyn * (t(i)-x(k));
            end
```

```

        suma = suma + R(j,j)*iloczyn;
    end
    fun(i)=R(1,1)+suma;
end
plot2d(t,fun,c);
endfunction

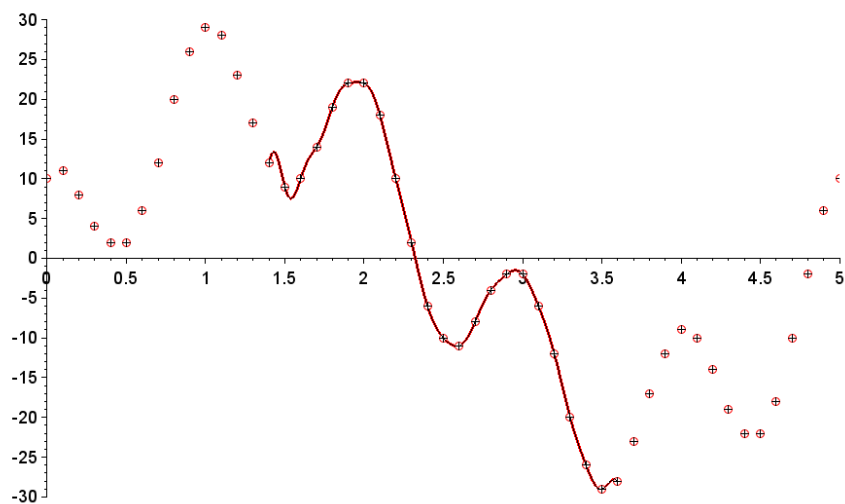
```

W dokładnej arytmetyce postaci Newtona i Lagrange'a dają ten sam wielomian. Różnice obserwowane numerycznie wynikają z algorytmu i błędów zaokrągleń (ryc. 12).



Ryc. 12. Interpolacja wielomianem Newtona

Porównanie obu postaci dla sygnału okresowego pokazano na ryc. 13.



Ryc. 13. Interpolacja sygnału sinusoidalnego postaciami Newtona 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()*

Dla równomiernie próbkowanych sygnałów okresowych naturalnym narzędziem jest interpolacja trygonometryczna. Współczynniki wyznacza dyskretna transformata Fouriera, dostępna w **Scilabie** jako `fft()` [3]. Aby ograniczyć przeciek widma, analizowany przedział powinien obejmować całkowitą liczbę okresów.

Argumentem funkcji jest tablica próbek. Na podstawie widma oraz częstotliwości próbkowania wyznacza się składowe szeregu trygonometrycznego (12).

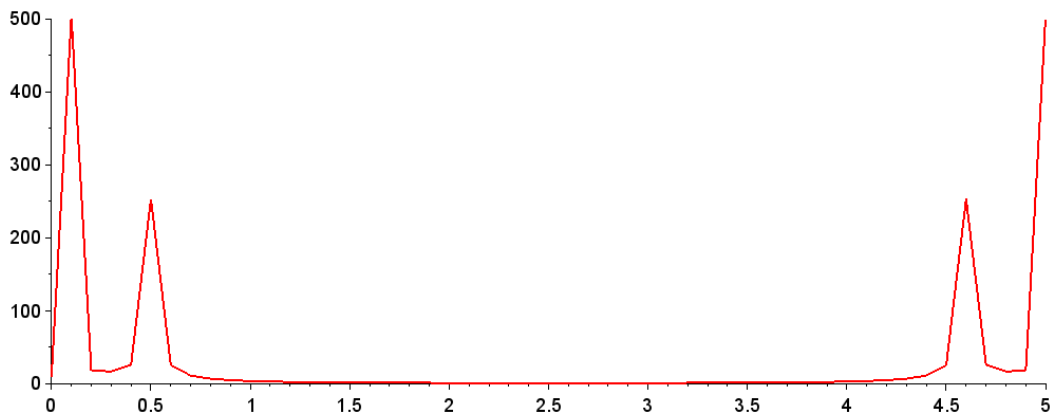
$$F(x) = \sum_{i=1}^n A_i \cdot \sin(f_i \cdot x + \varphi_i) \quad (12)$$

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 ryc. 14.



Ryc. 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 ryc. 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);

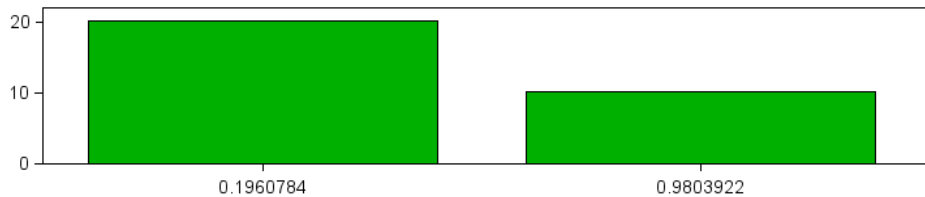
```

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);

```



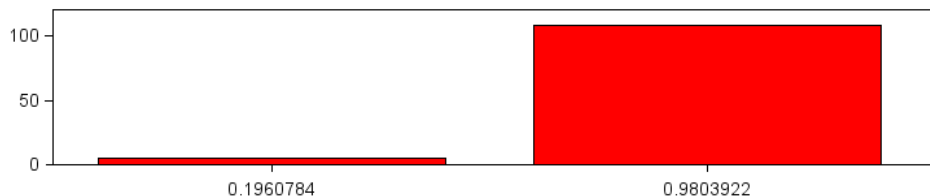
Ryc. 15. Wartości amplitudy dla określonych częstotliwości

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

```

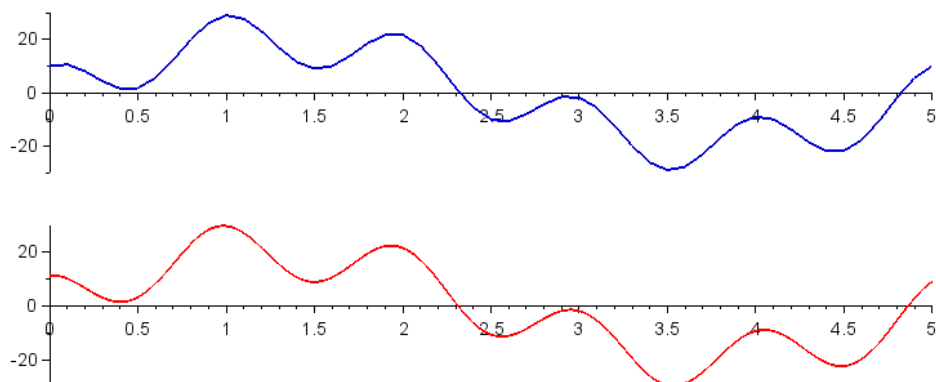
fi=atan(imag(fft_y),real(fft_y));
fi=angle(fft_y); //alternatywne obliczenie fi
faza=fi(k1)+%pi/2;
kat=faza*180/%pi;

```



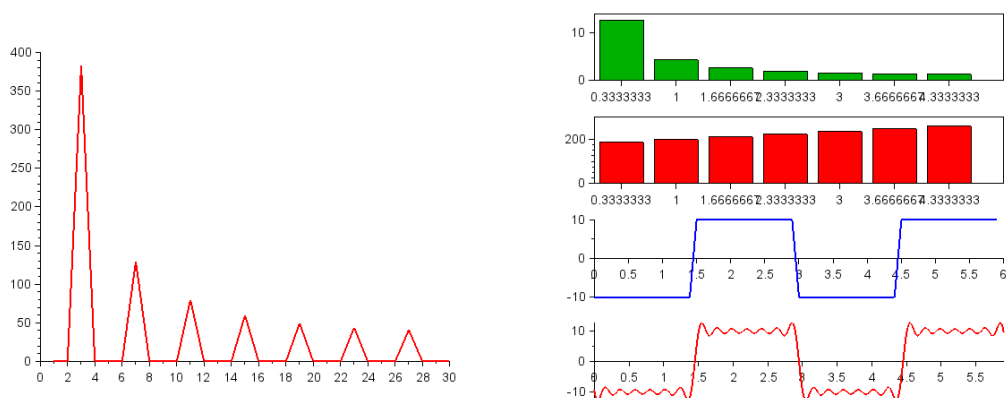
Ryc. 16. Wartości kąta fazowego dla określonych częstotliwości

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

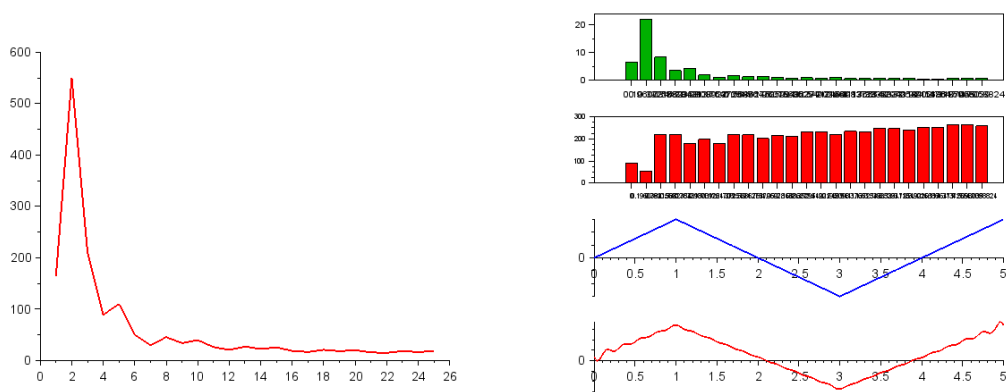


Ryc. 17. Przebiegi sygnału źródłowego (niebieski) i jego interpolacji wielomianem trygonometrycznym (czerwony)

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



Ryc. 18. Interpolacja wielomianem trygonometrycznym sygnału prostokątnego



Ryc. 19. Interpolacja wielomianem trygonometrycznym sygnału piło kształtnego

Można zauważyć oscylacje w pobliżu nieciągłości (zjawisko Gibbsa). Dla danych nieokresowych bez odpowiedniego przygotowania krańców przedziału interpolacja trygonometryczna zwykle daje słabe wyniki.

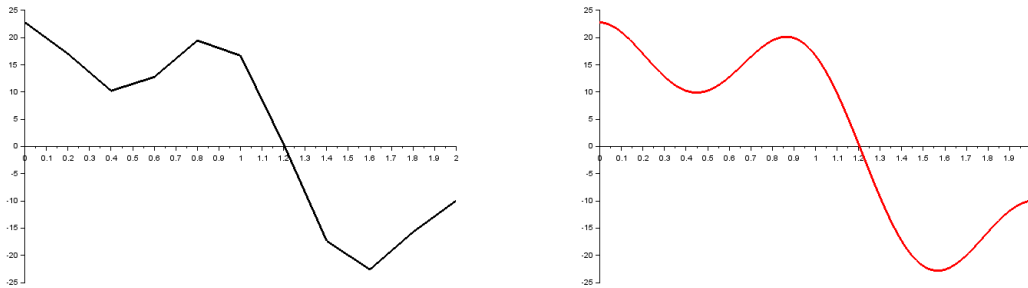
3.3 Interpolacja funkcjami sklejanymi

Funkcja sklejana (*spline*) jest zbudowana z wielomianów niskiego stopnia określonych osobno w kolejnych podprzedziałach. Pozwala to ograniczyć oscylacje charakterystyczne dla jednego wielomianu wysokiego stopnia [1, 2].

Najprostszy wariant jest odcinkami liniowymi: w przedziale $[t_k, t_{k+1}]$ przyjmuje się $f(t) = A_k t + B_k$. Współczynniki wynikają z dwóch sąsiednich węzłów:

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

Liczba przedziałów jest o jeden mniejsza od liczby węzłów. Dla każdego z nich określonego przez węzły n i $n+1$ wyznaczone 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 ryc. 20.



Ryc. 20. Interpolacja funkcjami sklejanymi I stopnia dla 11 i 201 węzłów

- wielomianie trzeciego stopnia — funkcja sklejana trzeciego stopnia

Gładszy przebieg daje interpolacja funkcją sklejaną trzeciego stopnia. W każdym przedziale stosuje się wielomian $f(t) = At^3 + Bt^2 + Ct + D$, a warunki sklejania zapewniają ciągłość wartości oraz pierwszej i drugiej pochodnej.

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.

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

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

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

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

Drugie pochodne w węzłach wyznacza się z warunku ciągłości pierwszej pochodnej. Dla węzła wewnętrznego otrzymuje się 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 (19) tworzy się układ dla drugich pochodnych. Dodatkowe dwa warunki określają typ funkcji sklejanej. Dla funkcji naturalnej przyjmuje się $f''_1 = f''_n = 0$; nie są to dowolne wartości, lecz jawnie wybrane warunki brzegowe.

Dla stałego kroku $x_{i+1} - x_i = h$ układ upraszcza się do postaci

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

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

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 trójdagonalną **Z**, oraz wektor **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.

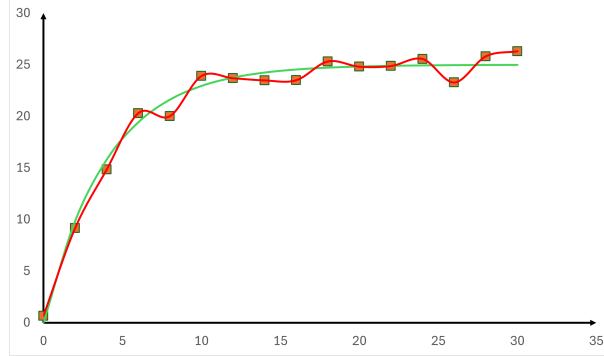
W praktyce w **Scilabie** naturalną funkcję sklejaną można wyznaczyć funkcjami **splin()** i **interp()**:

```
x=[0 1 2 3 4];
y=[0 1 0 1 0];
xx=linspace(x(1),x($),201);
d=splin(x,y,"natural");
yy=interp(xx,x,y,d);
plot(x,y,"o",xx,yy,"-");
```

4 Aproksymacja

W odróżnieniu od interpolacji funkcja aproksymująca nie musi przechodzić przez wszystkie punkty. Aproksymację stosuje się między innymi wtedy, gdy dane są obarczone błędem pomiarowym lub gdy potrzebny jest prosty model trendu [1, 2].

Funkcja interpolująca przechodzi przez wszystkie węzły, natomiast funkcja aproksymująca przebiega możliwie blisko danych w sensie przyjętej miary błędu (ryc. 21).



Ryc. 21. Przebiegi funkcji interpolującej i aproksymującej dane

Do przeprowadzenia aproksymacji należy wybrać rodzinę funkcji $f(x)$ i miarę rozbieżności względem danych $F(x_i)$. Przykładowym kryterium jakości jest warunek

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

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 doboru 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.

Najczęściej stosuje się aproksymację jednostajną (minimalizacja największego błędu bezwzględnego) albo średniokwadratową (minimalizacja sumy kwadratów reszt). Druga z nich jest wrażliwa na obserwacje odstające, ale dla niezależnych błędów o jednakowej wariancji ma dobre uzasadnienie statystyczne.

Aproksymacja średniokwadratowa polega na takim doborze współczynników, aby suma kwadratów reszt była najmniejsza:

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

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 .

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

Po przekształceniu powyższą zależność można zapisać w postaci:

$$\begin{aligned}n \cdot a_0 + a_1 \cdot \sum_{i=1}^n x_i &= \sum_{i=1}^n y_i \\ a_0 \cdot \sum_{i=1}^n x_i + a_1 \cdot \sum_{i=1}^n x_i^2 &= \sum_{i=1}^n x_i \cdot y_i\end{aligned}\tag{25}$$

Układ równań normalnych ma postać

$$\begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix}.\tag{26}$$

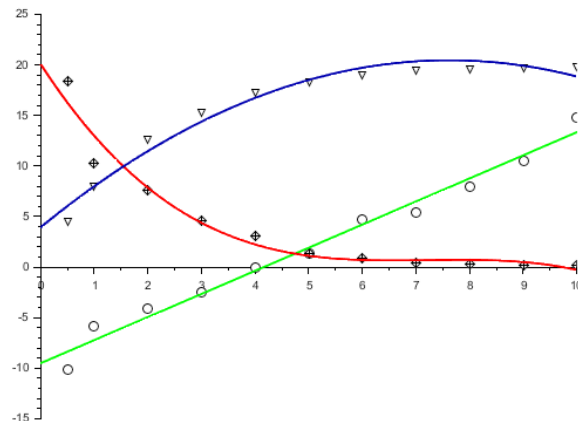
Współczynniki można wyznaczyć przez rozwiązanie układu liniowego. Dla wielomianu stopnia k wygodnie tworzy się macierz projektu X i rozwiązuje zadanie najmniejszych kwadratów

$$X^T X a = X^T y, \quad X_{ij} = x_i^{j-1}.\tag{27}$$

Ze względu na pogorszenie uwarunkowania przez iloczyn $X^T X$ w programie należy użyć bezpośrednio operatora rozwiązującego zadanie najmniejszych kwadratów:

```
// x, y - kolumnowe wektory danych; k - stopień wielomianu
X=ones(x);
for j=1:k
    X=[X x.^j];
end
a=X\y;
y_fit=X*a;
reszta=y-y_fit;
fprintf('RMSE = %.6g\n',sqrt(mean(reszta.^2)));
```

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.



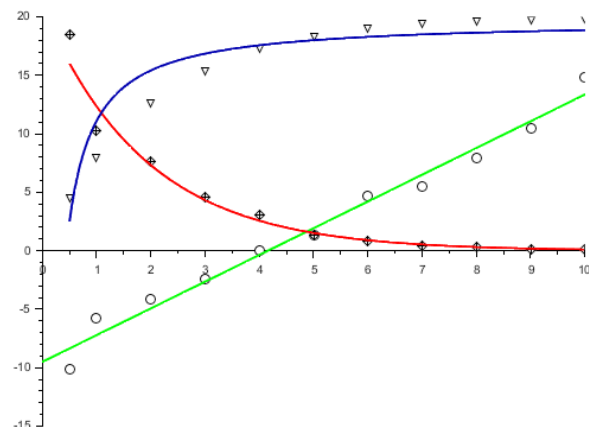
Ryc. 22. Aproksymacja średniokwadratowa funkcją wielomianową trzech zestawów danych pomiarowych

Niektóre modele nieliniowe można zlinearyzować przez podstawienie. Na ryc. 22 krzywe czerwoną i niebieską należałoby opisać odpowiednio modelami (28) i (29). Trzeba pamiętać, że transformacja zmienia model błędu i wagi obserwacji.

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

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

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



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

Zadania

Do ćwiczenia należy wykorzystać pliki CSV udostępnione przez prowadzącego, zawierające obszerne przebiegi czasowe t , $u_C(t)$ i $i(t)$. Każdy student tworzy indywidualny zestaw z przebiegu referencyjnego, dodając szum normalny o parametrach i ziarnie z tabeli. W danych z lukami usuwa co dziesiątą próbkę, zaczynając od indeksu m .

Zestaw	σ_u [mV]	σ_i [mA]	ziarno	m
1	2,0	0,50	20260701	10
2	2,2	0,55	20260702	11
3	2,4	0,60	20260703	12
4	2,6	0,65	20260704	13
5	2,8	0,70	20260705	14
6	3,0	0,75	20260706	15
7	3,2	0,80	20260707	16
8	3,4	0,85	20260708	17
9	3,6	0,90	20260709	18
10	3,8	0,95	20260710	19
11	4,0	1,00	20260711	20
12	4,2	1,05	20260712	21
13	4,4	1,10	20260713	22
14	4,6	1,15	20260714	23
15	4,8	1,20	20260715	24

Wspólne parametry modelu: $R = 20 \Omega$, $L = 0,1 \text{ H}$, $C = 100 \mu\text{F}$, $E = 10 \text{ V}$ i $\Delta t = 50 \mu\text{s}$.

1. Wczytać plik `rlc_pomiar_pelny.csv` i sprawdzić jednostki, krok próbkowania, monotoniczność czasu, brakujące wartości oraz podstawowe statystyki przebiegów referencyjnych. Następnie, korzystając z parametrów swojego zestawu, utworzyć indywidualne przebiegi pomiarowe przez dodanie szumu do kolumn referencyjnych.
2. Wyznaczyć prąd kondensatora z zależności

$$i_C(t) = C \frac{du_C(t)}{dt}$$

za pomocą różnic skończonych. Porównać wynik z prądem referencyjnym i obliczyć błąd RMSE.

3. Odtworzyć napięcie kondensatora z całki

$$u_C(t) - u_C(0) = \frac{1}{C} \int_0^t i(\tau) d\tau.$$

Zastosować co najmniej dwie metody całkowania i porównać błędy rekonstrukcji.

4. Z indywidualnego przebiegu pomiarowego odłożyć wartości kontrolne, a następnie usunąć co dziesiątą próbkę, rozpoczynając od indeksu m podanego w tabeli zestawów. Odtworzyć usunięte próbki interpolacją liniową oraz funkcją sklejaną. Obliczyć RMSE względem odłożonych wartości kontrolnych i porównać zachowanie metod w pobliżu ekstremów przebiegu.

Zadanie rozszerzające. Wyznaczyć dominującą częstotliwość oscylacji za pomocą FFT albo oszacować tłumienie z obwiedni kolejnych maksimów. Dla przebiegu aperiodycznego z pliku `rlc_aperiodyczny.csv` można dopasować model wykładniczy i przeanalizować reszty.

Bibliografia

- [1] Richard L. Burden, J. Douglas Faires i Annette M. Burden. *Numerical Analysis*. 10 wyd. Cengage Learning, 2016.
- [2] Zenon Fortuna, Bohdan Macukow i Janusz Wąsowski. *Metody numeryczne*. Warszawa: Wydawnictwa Naukowo-Techniczne, 2005.
- [3] *Strona Scilab Enterprises S.A.S.* 2024. URL: <http://www.scilab.org>.