

1 Rozdział

Strona: [LeIA](#)

Kurs: Metody numeryczne (2025Z)

Książka: 1 Rozdział

Wydrukowane przez użytkownika: Kinga Kondraciuk

Data: niedziela, 30 listopada 2025, 13:59

Spis treści

- 1. Elementy teorii błędów**
- 2. Błędy bezwzględne i względne**
- 3. Błędy funkcji jednej zmiennej**
- 4. Błędy funkcji wielu zmiennych**
- 5. Błędy działań arytmetycznych**
 - 5.1. 6. prawdzamy markdown

1. Elementy teorii błędów

Metody numeryczne dają nam możliwość rozwiązywania pewnych zagadnień w sposób przybliżony wtedy, gdy dokładne metody nie mogą być stosowane lub nie są znane.

Każde rozwiązywanie numeryczne wiąże się z popełnianiem błędów obliczeń. Błędy te mogą wynikać z różnych przyczyn.

Jedną z nich jest podawanie danych w sposób przybliżony - podczas pomiarów lub doświadczeń popełniamy nieścisłości związane np.: z dokładnością przyrządu pomiarowego.

Błędy danych mogą silnie wpływać na wyniki obliczeń, ale nie zawsze. Czasami można podać tzw.: wskaźniki uwarunkowania, które "przenoszą" błędy danych na błędy obliczeń końcowych. Podamy takie wskaźniki dla błędów funkcji jednej i wielu zmiennych.

Drugą przyczyną jest dokładność samego algorytmu stosowanego do obliczeń. Może się zdarzyć, że dokładny wzór np.: rekurencyjny nie nadaje się do obliczeń numerycznych, będącym o nim mówić, że nie jest stabilny - podamy przykład takiego algorytmu. Nawet kolejność działań na liczbach przybliżonych może mieć znaczenie i wpływać na wynik, a niektóre działania np.: odejmowanie liczb przybliżonych bliskich daje czasami zaskakująco duże błędy.

Trzeba również pamiętać o błędach maszynowych, wynikające z reprezentacji liczb w komputerze. Wszystkie liczby w komputerze są zaokrąglane do wartości, która jest.

Wprawdzie dzisiejsze komputery są bardzo dokładne, to jednak nakładanie się tych wszystkich błędów może dawać absurdalne wyniki. Będziemy ilustrować takie sytuacje.

2. Błędy bezwzględne i względne

Dużymi literami A, B, C,... będziemy oznaczać liczby dokładne, a małymi a, b, c ... przybliżone wartości tych liczb.

Błędem bezwzględnym liczby przybliżonej a nazywamy wartość bezwzględną różnicę między liczbą dokładną A a jej przybliżeniem, co zostało przedstawione we wzorze [1.2.1](#).

$$\Delta a = |A - a| \quad (1.2.1)$$

Na ogół nie jest znana wartość liczby A, wtedy a możemy tylko oszacować z góry. W praktyce błędem bezwzględnym nazywamy możliwie najmniejsze oszacowanie takiej różnicy. Na ogół błąd bezwzględny możemy przyjmować jako dokładność przyrządu pomiarowego.

Błędem względnym nazywamy stosunek błędu bezwzględnego do wartości bezwzględnej liczby przybliżonej (patrz wzór [1.2.2](#)):

$$\delta a = \frac{\Delta a}{|a|} \quad (1.2.2)$$

dla a różnego od zera.

Przykład 1.2.1

Przykład

Jeśli liczba dokładna $A = 1.88$, a chcemy podać jej przybliżenie z dokładnością do jednego miejsca po przecinku to $a = 1.9$ i błąd $\Delta a = |A - a| = 0.02$. Oczywiście nie "obcinamy" liczby dokładnej do jednego miejsca po przecinku (ang. truncate), tylko zaokrąglamy cyfrę 8 do cyfry 9. Gdybyśmy "obcięli" i za liczbę przybliżoną przyjęli $a^* = 1.8$ błąd bezwzględny byłby o wiele większy: $\Delta a^* = |A - a| = 0.08$.

Obliczymy jeszcze błędy względne obu przybliżeń. Błąd względny dla wartości zaokrąglonej wynosi:

$$\delta a = \frac{\Delta a}{|a|} = \frac{0.02}{1.9} = 0.0105, \quad (1.2.1.1)$$

co stanowi 1.05 liczby przybliżonej, natomiast błąd względny dla wartości obciętej wynosi (patrz [1.2.1.2](#)):

$$\delta a^* = \frac{\Delta a^*}{|a^*|} = \frac{0.08}{1.8} = 0.0444 \quad (1.2.1.2)$$

co stanowi 4.44 liczby przybliżonej.

Będziemy się posługiwać pojęciem cyfr dokładnych liczby przybliżonej. W podanym przykładzie w liczbie $a = 1.9$ wszystkie cyfry będą dokładne mimo, że w liczbie dokładnej nie występuje cyfra 9, natomiast w przybliżeniu $a^* = 1.8$ cyfra 8 nie jest dokładna mimo, że taka sama cyfra i na takim samym miejscu występuje w A. Liczba przybliżona będzie mieć wszystkie cyfry dokładne, jeśli jej błąd bezwzględny nie będzie przekraczać połowy ostatniego uwzględnionego miejsca dziesiętnego.

Przykład 1.2.2

Przykład

Dany jest szereg: $\sum_{n=1}^{\infty} (-1)^n \frac{2n}{(n+2)!}$, obliczymy jego przybliżoną sumę, przyjmując dokładność $\varepsilon = 10^{-10}$. Oczekiwany wynik przybliżony

powinien być obarczony błędem bezwzględnym mniejszym od wartości ε .

Sumujemy szereg naprzemienny zbieżny do zera. Jeśli za przybliżoną sumę szeregu będziemy brać n-tą sumę częściową (n pierwszych wyrazów), to błąd bezwzględny między dokładną sumą a jej przybliżeniem nie będzie przekraczał wartości bezwzględnej pierwszego odrzuconego wyrazu czyli $|a_n + 1|$. Zatem będziemy brać tyle wyrazów, aż sąsiednie sumy będą się różnić o mniej niż podana dokładność $10^{-10} = 0,0000000001$. Zatem kolejne wyrazy możemy obliczyć za pomocą ciągu:

$$s_1 = \frac{-2}{3!}, s_2 = \frac{-2}{3!} + \frac{4}{4!}, s_n = s_{n-1} + (-1)^n \frac{2n}{(n+2)!}$$

i sumujemy tak długo, aż $s_n - s_{n-1} < \varepsilon = 10^{-10}$. Okazuje się, że wystarczy zsumować $n = 13$ wyrazów i wtedy przybliżona suma będzie wynosić $s_n = -0.2072766470$.

Przyjrzyjmy się teraz krótkiemu programowi w MATLABie ilustrującemu powyższe obliczenia.

```
function p1_2_2
    format long
    % obliczamy sumę najpierw 13, a potem 14 pierwszych wyrazów
    c_13=ciag(13)
    c_14=ciag(14)
    c_13-c_14

    % obliczamy sumę częściową z założoną dokładnością
    % c_13 zawiera wartość, n_13 zawiera liczbę zsumowanych składników
    [c_13, n_13] = ciag_dokladosc(1e-10)
end

% wariant funkcji z określona maksymalną liczbą składników
function s = ciag(max_n)
    s=0;
    for n=1:max_n
        s = s + (-1)^n * (2*n)/factorial(n+2);
    end
end

% wariant funkcji z określona dokładnością
function [s,n] = ciag_dokladosc(max_eps)
    s=0;
    for n=1:50
        sn = (-1)^n * (2*n)/factorial(n+2);
        s = s + sn;
        if abs(sn) < max_eps
            break;
        end
    end
end
```

Powyższy listing powinien zwrócić wyniki zaprezentowane poniżej. Możemy zaobserwować, że obie wartości c_{13} są zgodne z podanymi w przykładzie. Wartość n_{13} zawiera rzeczywistą liczbę składników uwzględnionych w sumie. Jednocześnie wartość ans , która reprezentuje różnicę między dwoma kolejnymi przybliżeniami obliczonymi dla $n = 13$ oraz $n = 14$ jest mniejsza od założonej dokładności $\varepsilon = 10^{-10}$.

```
>> p1_2_2
c_13 =
-0.207276647029913
c_14 =
-0.207276647028574
ans =
-1.338262833883164e-12
c_13 =
-0.207276647029913
n_13 =
13
```

3. Błędy funkcji jednej zmiennej

Błędy funkcji jednej zmiennej

Błędem funkcji jednej zmiennej należy interpretować jako jej właściwość związaną z wrażliwością na dokładność zadawanych jej danym wejściowym. Niektóre funkcje przenoszą błędy danych wejściowych zwiększać inne funkcje tłumią błędy danych wejściowych. Wrażliwość funkcji określa się wskaźnikiem uwarunkowania, którego definicję wyprowadzimy w tym rozdziale.

Przykład 1.3.1

Przykład

Zmierzliśmy długość boku sześcianu i otrzymaliśmy wynik $x = 2.3$ cm, ale naszą miarką możemy zmierzyć z dokładnością do 0.03 cm.

Jak błąd długości boku wpłynie na błąd objętości tego sześcianu?

Mamy następujące dane: bok $x = 2.3$ cm, błąd $\Delta x = 0.03$ cm, objętość sześcianu jest funkcją boku i wynosi $v(x) = x^3$.

Szukamy Δv i δv , czyli błędu bezwzględnego i względnego objętości.

Aby wykonać obliczenia podamy ogólne wzory na te błędy.

Rozpatrujemy funkcję jednej zmiennej $f(x)$ i argument x jest obarczony błędem bezwzględnym Δx . Wtedy błąd bezwzględny funkcji, oznaczany przez Δf , równa się:

$$\Delta y = \Delta f = |f'(x)|\Delta x \quad (1.3.1)$$

gdzie pochodną we wzorze obliczamy dla wartości podanego argumentu. Wzór ten wynika ze wzoru Taylora funkcji jednej zmiennej:

$$f(x + \Delta x) = f(x) + \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) + \frac{\Delta x^3}{3!} f'''(x) + \dots$$

Przenieśmy składnik $f(x)$ na lewą równania i "obetnijmy" szereg po składniku z pierwszą pochodną. Wówczas możemy zapisać wyrażenie na błąd przybliżony:

$$f(x + \Delta x) - f(x) = \Delta x f'(x)$$

ponieważ $\Delta f = f(x + \Delta x) - f(x)$. Zauważmy dodaną wartość bezwzględną w wyrażeniu 1.3.1, która gwarantuje, że błąd jest wyrażony zawsze jako wartość dodatnia.

Z ogólnego wzoru na błąd względny możemy zapisać:

$$\delta y = \delta f = \frac{\Delta f}{|f(x)|} \quad (1.3.2)$$

Wzór ten można przekształcić, wstawiając do niego wzór na błąd bezwzględny i otrzymamy:

$$\delta y = \delta f = \frac{\Delta f}{|f(x)|} = \frac{|f'(x)\Delta x|}{|f(x)|} = \frac{|f'(x) \cdot x|}{|f(x)|} \frac{\Delta x}{x} = \omega \cdot \delta x$$

gdzie wielkość $\omega = \frac{|f'(x) \cdot x|}{|f(x)|}$ nazywamy **wskaźnikiem uwarunkowania** i za jego pomocą możemy zapisać wzór na błąd względny funkcji:

$$\delta f = \omega \cdot \delta x \quad (1.3.3)$$

Z tego wzoru widać, że wskaźnik ten "przenosi" błąd względny z argumentu na funkcję.

Wróćmy do przykładu 1.3.1. Korzystając z powyższych wzorów mamy:

$$v(x) = x^3, v'(x) = 3x^2, x = 2.3, \Delta x = 0.03$$

$$v(x) = 12.2, \Delta v = |3 \cdot (2.3)^2| \cdot 0.03 = 0.476$$

$$\delta v = \frac{0.476}{12.2} = 0.039, \delta x = \frac{0.03}{2.3} = 0.013, \omega = \frac{\delta v}{\delta x} = 3$$

Błąd wzgledny funkcji powiększył się 3 razy w stosunku do błędu wzglednego argumentu.

Oczywiście wyniki są zaokrąglone. Ponieważ błąd bezwzględny objętości wynosi 0.5 nie ma sensu podawać w wyniku więcej cyfr po przecinku, nawet cyfra 2 po przecinku nie jest cyfrą dokładną.

Wyniki na błędy wzgledne podane są z trzema cyframi po przecinku, żeby wyraźnie było widać, że błąd wzgledny wzrosł 3 razy.

W nastepnym przykładzie błąd wzgledny funkcji dla wartości $x = 1$ i $x = -1$ rośnie do nieskończoności, a im bliższe są argumenty tych wartości tym błąd jest większy. Wiąże się to z odejmowaniem liczb przybliżonych bliskich. Proszę porównać przykład z tematu: Błędy działań arytmetycznych.

Przykład 1.3.2

Przykład

Dana jest funkcja $f(x) = x^2 - 1$. Napisac wzór na wskaźnik uwarunkowania. Obliczyć go dla różnych wartości argumentu x . Obliczyć błędy wzgledne funkcji dla różnych argumentów, biorąc za błąd wzgledny argumentu 5% jego wartości (bezwzględnej).

Obliczymy pochodną funkcji i podamy wzory na błędy:

$$f'(x) = 2x, \quad \Delta f = |2x| \cdot \Delta x, \quad \delta x = 0.05, \quad \Delta x|x|,$$

$$\omega(x) = \left| \frac{f'(x) \cdot x}{f(x)} \right| = \left| \frac{2x \cdot x}{x^2 - 1} \right| = \left| \frac{2x^2}{x^2 - 1} \right|$$

$$\delta f(x) = \omega(x) \cdot \delta x$$

Dla $x = 1$ oraz $x = -1$ nie można obliczyć błędu wzglednego funkcji. Jeśli x będzie bliski jedności, wskaźnik uwarunkowania będzie duży i błąd wzgledny funkcji też będzie duży. Badając funkcję proszę wstawiać argumenty dalekie od 1 i bliskie np.: 1,03.

Przeanalizujmy krótki program w MATLABie, który pozwoli nam zbadać właściwości tej funkcji.

```
function p1_3_2
format short
% definiujemy wyrażenia (funkcje) dla przykładu
f = @(x) (x*x-1);
df = @(x) (2*x);
w = @(x) (df(x)*x/f(x));

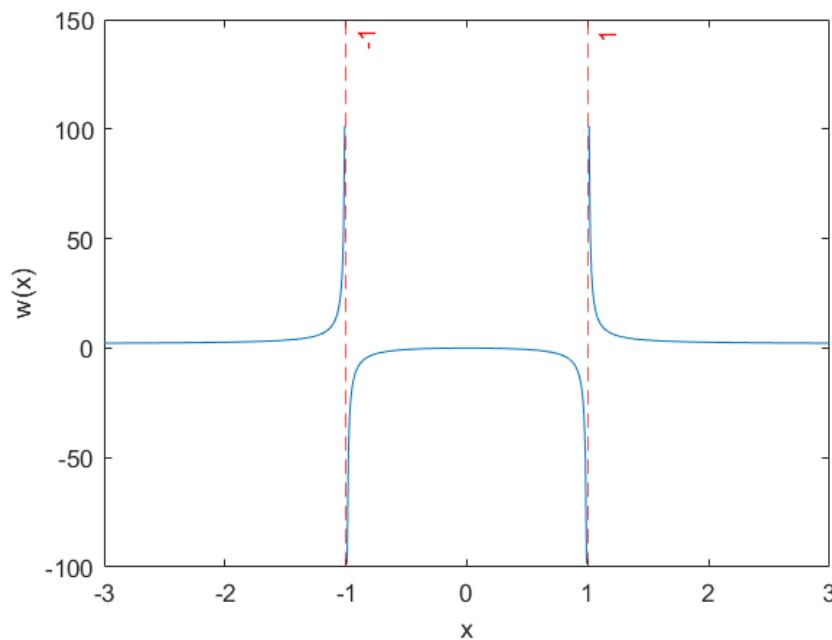
% Obliczamy kilka wyników
fprintf("Wartość w(1.01)\t= %f\n", w(1.01));
fprintf("Wartość w(1.03)\t= %f\n", w(1.03));
fprintf("Wartość w(-2)\t= %f\n", w(-2));
fprintf("Wartość w(10.5)\t= %f\n", w(10.5));

% aby przeanalizować zmienność funkcji narysujemy zależność
% współczynnika uwarunkowania od x
W = [];
X = -3:0.01:3;
for x=X
    % doklejamy do wektora wyników kolejną wartość
    W = [W w(x)];
end

% rysujemy przebieg zależności w(x)
plot(X,W)
xlabel('x')
ylabel('w(x)')

% dorysowujemy ważne linie wskazujące wartości osobliwe gdy wskaźnik uwarunkowania dąży do
% nieskończoności
xline(-1, '--r', {'-1'})
xline(1, '--r', {'1'})
end
```

Powyższy program wygeneruje rysunek reprezentujący zależność wskaźnika uwarunkowania $\omega(x)$ od x dla $x \in (-3, 3)$.



Rys. 1.3.1. Przebieg zależności wskaźnika uwarunkowania dla funkcji z przykładu 1.3.2

Po uruchomieniu program powinien wyświetlić cztery wartości współczynnika uwarunkowania dla przykładowych wartości x . Uruchomienie programu oraz obserwację wyników pozostawiamy czytelnikowi.

4. Błędy funkcji wielu zmiennych

Przykład 1.4.1

Przykład

Zmierzliśmy boki prostopadłościanu i otrzymaliśmy $x=1.2\text{cm}$, $y=1.8\text{cm}$ oraz $z=2.1\text{cm}$. Nasz przyrząd pomiarowy ma dokładność $0,01\text{cm}$. Jaki popełnimy błąd bezwzględny i względny licząc pole powierzchni całkowitej tego prostopadłościanu?

Mamy dane: $x=1.2$, $y=1.8$, $z=2.1$, błędy bezwzględne przyjmujemy dla wszystkich boków

$\Delta x = \Delta y = \Delta z = \Delta = 0.01$ wzór na pole $p(x, y, z) = 2xy + 2xz + 2yz$.

W obliczeniach skorzystamy z następujących wzorów definiujących błędy dla funkcji wielu zmiennych.

Rozważania przeprowadzimy dla funkcji 2 zmiennych, ale wszystkie wzory można uogólnić na więcej zmiennych. Dana jest funkcja $f(x, y)$ i argumenty są obarcone błędami Δx , Δy . Wzór na błąd bezwzględny funkcji wynika, tak jak dla funkcji jednej zmiennej, ze wzoru Taylora (nie będziemy go wyprowadzać) i jest następujący:

$$\Delta f(x, y) = \left| \frac{\partial f(x, y)}{\partial x} \right| \Delta x + \left| \frac{\partial f(x, y)}{\partial y} \right| \Delta y \quad (1.4.1)$$

Pochodne cząstkowe są liczone dla tych wartości argumentów, dla których liczymy błąd.

Z ogólnego wzoru na błąd względny otrzymujemy:

$$\delta f = \frac{\Delta f}{|f(x)|} \quad (1.4.2)$$

Przekształcimy ten wzór tak, jak wzór na błąd funkcji jednej zmiennej, aby wprowadzić wskaźniki uwarunkowania.

$$\delta f = \frac{\Delta f}{|f(x)|} = \frac{|f_x(x, y)| \Delta x + |f_y(x, y)| \Delta y}{|f(x, y)|} = \left| \frac{f_x(x, y) \cdot x}{f(x, y)} \right| \frac{\Delta x}{|x|} + \left| \frac{f_y(x, y) \cdot y}{f(x, y)} \right| \frac{\Delta y}{|y|} = \omega_x \cdot \delta x + \omega_y \cdot \delta y \quad (1.4.3)$$

Wielkości, które wprowadziliśmy:

$$\omega_x = \left| \frac{f_x(x, y) \cdot x}{f(x, y)} \right|, \omega_y = \left| \frac{f_y(x, y) \cdot y}{f(x, y)} \right| \quad (1.4.4)$$

nazywamy wskaźnikami uwarunkowania odpowiednio zmiennej x i y , "przenoszą" one błędy względne argumentów na błąd względny funkcji.

Powrócimy do przykładu 1.4.1, podając jednocześnie wzory dla funkcji 3 zmiennych.

Mamy dane: $x=1.2$, $y=1.8$, $z=2.1$, błędy bezwzględne przyjmujemy dla wszystkich boków $\Delta x = \Delta y = \Delta z = \Delta = 0.01$, wzór na pole $p(x, y, z) = 2xy + 2xz + 2yz$. Pochodne cząstkowe względem poszczególnych argumentów wynoszą:

$$p_x(x, y, z) = 2y + 2z, \quad p_y(x, y, z) = 2x + 2z, \quad p_z(x, y, z) = 2x + 2y$$

Wartość pola $p = 16,9\text{cm}^2$.

Zatem wartość błędu możemy wyznaczyć następująco:

$$\Delta p(x, y, z) = \left| \frac{\partial p(x, y, z)}{\partial x} \right| \Delta x + \left| \frac{\partial p(x, y, z)}{\partial y} \right| \Delta y + \left| \frac{\partial p(x, y, z)}{\partial z} \right| \Delta z = [|2 \cdot 1.8 + 2 \cdot 2.1|] + [|2 \cdot 1.2 + 2 \cdot 2.1|] + [|2 \cdot 1.2 + 2 \cdot 1.8|] \cdot 0.01 = 0.2$$

$$\delta p = \frac{\Delta p}{p(x, y, z)} = 0.012, \quad \omega_x = 0.553, \quad \omega_y = 0.702, \quad \omega_z = 0.745$$

Program przykładowy ilustrujący rozwiązanie w MATLABie:

```
function p1_4_1
    x = 1.2;
    y = 1.8;
    z = 2.1;
    Dx = 0.01; Dy = 0.01; Dz = 0.01;
    dx = Dx/x; dy = Dy/y; dz = Dz/z;
    p = @(x,y,z) (2*x*y+2*x*z+2*y*z);
    dpx = @(x,y,z) (2*y+2*z);
    dpy = @(x,y,z) (2*x+2*z);
    dpz = @(x,y,z) (2*x+2*y);

    % pierwszy sposób z definicji obliczenia
    Dp = @(x,y,z) (dpx(x,y,z)*Dx + dpy(x,y,z)*Dy + dpz(x,y,z)*Dz);
    dp = Dp(x,y,z) / p(x,y,z)

    % drugi sposób wykorzystujący wskaźniki uwarunkowania
    wx = dpx(x,y,z)*x / p(x,y,z)
    wy = dpy(x,y,z)*y / p(x,y,z)
    wz = dpz(x,y,z)*z / p(x,y,z)
    dp = wx*dx + wy*dy + wz*dz
end
```

Powyższy program po uruchomieniu powinien wyświetlić wyniki zgodne z obliczeniami:

```
>> p1_4_1
dp =
    0.0121
wx =
    0.5532
wy =
    0.7021
wz =
    0.7447
dp =
    0.0121
```

5. Błędy działań arytmetycznych

Błędy działań arytmetycznych

Skorzystamy ze wzoru (**1.4.1**) na błąd bezwzględny funkcji dwóch zmiennych, aby wyprowadzić wzory na błędy działań arytmetycznych.

Błąd sumy dwóch liczb przybliżonych: Argumenty x i y są obarczone odpowiednio błędami bezwzględnymi $(|\Delta x|, |\Delta y|)$, sumę argumentów zapisujemy jako $(s(x,y)=x+y)$. Pochodne cząstkowe tej funkcji zarówno po (x) jak i po (y) są równe 1, zatem:

$$(|\Delta s(x,y)| = |\Delta x + \Delta y|) = |\Delta x + \Delta y| \quad (1.5.1)$$

Zatem błąd bezwzględny sumy równa się sumie błędów bezwzględnych składników. Błąd wzgledny trzeba obliczyć z ogólnego wzoru:

$$(|\delta s(x,y)| = |\frac{\Delta s}{s}|) = |\frac{\Delta x + \Delta y}{x+y}| \quad (1.5.2)$$

Błąd różnicy dwóch liczb przybliżonych: Argumenty x i y są obarczone odpowiednio błędami bezwzględnymi $(|\Delta x|, |\Delta y|)$, różnicę argumentów zapisujemy jako $(r(x,y)=x-y)$. Pochodne cząstkowe tej funkcji: po (x) jest równa 1, po (y) jest równa -1, zatem

$$(|\Delta r(x,y)| = |\Delta x - \Delta y|) = |\Delta x - \Delta y| \quad (1.5.3)$$

Zatem błąd bezwzględny różnicy równa się **sumie** błędów bezwzględnych składników. Błąd wzgledny trzeba obliczyć z ogólnego wzoru:

$$(|\delta r(x,y)| = |\frac{\Delta r}{r}|) = |\frac{\Delta x - \Delta y}{x-y}| \quad (1.5.4)$$

Wzór na błąd wzgledny ma sens jeśli x jest różne od y .

Przykład 1.5.1

Przykład

Dane są dwie liczby przybliżone $(a=0.0035)$ i $(b=0.0033)$. Wszystkie cyfry tych liczb są dokładne, tzn.: błędy bezwzględne tych liczb są równe nie więcej niż 0,00005. Obliczymy błędy wzgledne tych liczb i błąd wzgledny różnicę $(a-b)$.

$$(|a|=0.0035), (|b|=0.0033), (|\Delta a|=0.00005), (|\Delta b|=0.00005), (|\delta a| = |\frac{\Delta a}{a}| = 0.015), (|\delta b| = |\frac{\Delta b}{b}| = 0.015)$$

Błędy wzgledne składników to 1.5% dla (a) i 1.5% dla (b) , natomiast błąd wzgledny różnicy jest bardzo duży w porównaniu z błędami składników i wynosi 100%. Ten niekorzystny efekt jest związany z odejmowaniem liczb przybliżonych bliskich. Jeśli jest możliwość zastąpienia różnicy innym działaniem, należy zastosować inny wzór, aby nie odejmować liczb przybliżonych bliskich.

Błąd iloczynu dwóch liczb przybliżonych: Argumenty x i y są obarczone odpowiednio błędami bezwzględnymi $(|\Delta x|, |\Delta y|)$, iloczyn argumentów zapisujemy jako $(i(x,y)=xy)$. Pochodna cząstkowa tej funkcji: po x jest równa y , po y jest równa x , zatem błąd bezwzględny iloczynu jest równy:

$$(|\Delta i(x,y)| = |y| |\Delta x + x| |\Delta y|) \quad (1.5.5)$$

Błąd wzgledny trzeba obliczyć z ogólnego wzoru:

$$(|\delta i(x,y)| = |\frac{\Delta i}{i}|) = |\frac{\Delta x + x \Delta y}{xy}| = |\frac{|\Delta x| + |\Delta y|}{xy}| = |\frac{|\Delta x|}{x} + |\frac{\Delta y}{y}| \quad (1.5.6)$$

Zatem błąd wzgledny iloczynu równa się **sumie** błędów wzglednych czynników. Wzór na błąd wzgledny ma sens jeśli x i y są różne od zera.

Błąd ilorazu dwóch liczb przybliżonych: Argumenty x i y są obarczone odpowiednio błędami bezwzględnymi $(|\Delta x|, |\Delta y|)$, iloraz argumentów zapisujemy jako $(ir(x,y)=x/y)$ dla y różnego od zera. Pochodna cząstkowa tej funkcji: po x jest równa $(\frac{1}{y})$, po y jest

równa $(\frac{-x}{y^2})$, zatem

$$(\Delta ir(x,y) = \frac{1}{y} \cdot (\Delta x + \frac{-x}{y^2} \cdot \Delta y)) \quad (1.5.7)$$

Błąd względny trzeba obliczyć z ogólnego wzoru:

$$(\delta ir(x,y) = \frac{|\Delta ir(x,y)|}{|ir(x,y)|} = \frac{|\frac{1}{y} \cdot (\Delta x + \frac{-x}{y^2} \cdot \Delta y)|}{|\frac{x}{y^2}|} = \frac{|\frac{\Delta x}{y} + \frac{-x}{y^3} \cdot \Delta y|}{|\frac{x}{y^2}|} = \frac{|\Delta x| \cdot |\frac{1}{y}| + |\frac{-x}{y^2}| \cdot |\Delta y|}{|\frac{x}{y^2}|} = \frac{|\Delta x| + |\frac{x}{y^2}| \cdot |\Delta y|}{|\frac{x}{y^2}|} = \delta x + \delta y) \quad (1.5.8)$$

Zatem błąd względny ilorazu równa się sumie błędów względnych czynników. Wzór na błąd względny ma sens jeśli x i y są różne od zera. Na koniec tego tematu podamy przykład, który pokazuje, że sumowanie liczb za pomocą pewnych programów może nie być przemienne.

Przykład 1.5.2

Przykład

W programie, w którym różnica rzędu między liczbami nie może przekraczać (10^{15}) , obliczymy na dwa sposoby sumę znacznej ilości liczb bardzo małych i jednej dużej. Zmiana kolejności sumowania (najpierw małe potem duża, albo najpierw duża potem małe) będzie miała znaczący wpływ na wynik.

I sposób:

Do największej liczby $(c=26)$ dodajemy po kolei liczby $(d_i=9 \cdot 10^{16})$ gdzie $(i=0,1,\dots,n)$, a liczba $(n=300000)$. Korzystamy ze wzoru rekurencyjnego $(s_0=c, s_{i+1}=s_i+d_i)$ i w wyniku otrzymujemy sumę $(s_{n+1}=S=26.0000000000000)$ (13 zer po przecinku, $(n+1)$) dlatego, że suma uwzględnia jeszcze dodatkowo dużą liczbę.

II sposób:

Sumujemy po kolei liczby małe według wzoru rekurencyjnego: $(s_0=0, s_i=s_{i-1}+d_i)$, a potem dodajemy wynik do liczby dużej: $(s_{n+1}=s_n+c)$ i w wyniku otrzymujemy $(s_{n+1}=S=26.0000000002700)$.

Poniżej przedstawiony został program ilustrując przykład w MATLABie.

```
function p1_5_2
    format long
    d=9e-16;
    c = 26;

    % I sposób (najpierw duża, potem małe)
    s = c;
    n = 300000;
    for i=0:n
        s = s + d;
    end
    s

    % II sposób (najpierw małe, potem duża)
    s = 0;
    n = 300000;
    for i=0:n
        s = s + d;
    end
    s = s + c;
    s
end
```

Uruchomienie programu powinno zwrócić wynik:

```
>> p1_5_2
s =
26
s =
26.00000000269999
```

5.1. 6. sprawdzamy markdown

2.1. Numeryczna algebra liniowa

Algebra liniowa jest często osobnym kursem ujętym w programie typowych studiów inżynierskich. Niemniej istnieją pewne specjalne techniki, nieuwzględnione w standardowym kursie, które są związane ze specyfiką rozwiązywania zagadnień algebry liniowej na komputerach. Niniejszy rozdział przedstawia kilka wybranych takich metod.

Typowym zagadnieniem z zakresu algebry liniowej wykorzystywanym w ramach problemów inżynierskich jest rozwiązywanie układu równań liniowych. Mamy tutaj na myśli nie układy z trzema czy czterema niewiadomymi, które standardowo są rozwiązywane analitycznie, lecz układy z setkami czy nawet dziesiątkami tysięcy niewiadomych. Tego typu problemy wymagają specjalnych technik, które nie tylko gwarantują znalezienie rozwiązania, ale również znajdują je w sposób minimalizujący nakłady oraz błędy obliczeniowe. Wśród metod rozwiązywania układów równań liniowych wyróżniamy metody bezpośrednie oraz metody iteracyjne. Do metod bezpośrednich zaliczamy takie jak: eliminacja Gaussa, faktoryzacje LU czy QR, faktoryzacje SVD. Do metod iteracyjnych zaliczamy Jacobiego, SOR, Gradientów sprężonych, GMRES, i inne. W niniejszym kursie skupimy się tylko na wybranych metodach bezpośrednich.

Drugim klasycznym zagadnieniem algebry liniowej, który ma duże znaczenie z punktu widzenia inżynierskiego jest wyznaczanie wartości własnych macierzy. Należy zaznaczyć, że wyznaczanie wartości własnych macierzy metodami analitycznymi jest niezmiernie czasochłonne i często sprowadza się do rozwiązania bardzo źle uwarunkowanego wielomianowego równania charakterystycznego. Wśród metod numerycznych pozwalających wyznaczyć wartości i wektory własne są metody pozwalające wyznaczyć wartości własne rzeczywiste i urojone, maksymalną wartość własną, minimalną wartość własną lub wszystkie wartości. W niniejszym podręczniku przedstawimy jedynie podstawowe metody wyznaczania wartości własnych minimalnej i maksymalnej. Pozostałe metody wykraczają poza program studiów inżynierskich.

Na początku wprowadźmy podstawowe oznaczenia. Wiele zagadnień naukowych oraz inżynierskich prowadzi do układu równań liniowych $\{Ax=b\}$, który w formie macierzowej przyjmuje postać:

```
 $$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \left( \begin{array}{rrrr} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{array} \right) \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}
```

\right)

```
\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{2.1.1} $$
```

Przykład 2.1

Rozważmy przykład techniczny obwodu elektrycznego zaprezentowanego na rysunku 2.1. Naszym zadaniem jest przedstawić w zapisie macierzowym układ równań pozwalający znaleźć rozkład prądów w obwodzie. Należy wykorzystać prawa Kirchhoffa. Parametry obwodu to: $I=10\text{[A]}$, $E_1=5\text{[V]}$, $E_2=8\text{[V]}$, $R_1=5\text{[\Omega]}$, $R_2=5\text{[\Omega]}$, $R_3=3\text{[\Omega]}$, $R_4=7\text{[\Omega]}$, $R_5=2\text{[\Omega]}$.

image-20221021185329183

Rysunek 2.1 Obwód elektryczny zbudowany z trzech oczek, zawierający pięć rezystancji, dwa źródła napięcia E_1 , E_2 oraz jedno źródło prądu I_1 .

Z równań Kirchhoffa otrzymujemy pięć równań. Pierwsze trzy równania przedstawiają bilans prądów w węzłach, a pozostałe dwa bilans napięć w oczkach. Układ pięciu niezależnych liniowo równań zawiera pięć niewiadomych i posiada jednoznaczne rozwiązanie.

```
 $$ \begin{aligned} & \text{\begin{split}} l_1 + l_4 &= l_1 \cdot l_3 + l_4 &= l_5 \cdot l_2 + l_3 &= l_1 \cdot R_1 \cdot l_1 + R_3 \cdot l_3 - R_4 \cdot l_4 &= E_1 \cdot R_2 \cdot l_2 - R_3 \cdot l_3 - R_5 \cdot l_5 &= -E_2 \\ & \text{\end{split}} \end{aligned} \label{eq:obwod Algebraicznie} \tag{2.1.2} $$
```

Powyższy układ równań zapisany algebraicznie możemy przekształcić do postaci macierzowej:

```
 $$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & -1 \\ -1 & 1 & 1 & 0 & 0 & \mathbf{R}_1 & 0 & \mathbf{R}_3 & -\mathbf{R}_4 & 0 & 0 & \mathbf{R}_2 & -\mathbf{R}_3 & 0 & -\mathbf{R}_5 \end{bmatrix} \begin{bmatrix} 1 & 1 & 2 & 3 & 4 & 5 \end{bmatrix}
```

\end{bmatrix}

```
\begin{bmatrix} 1 & 0 & 0 & \mathbb{E} 1 \\ -\mathbb{E} 2 \end{bmatrix} \label{obwod:macierzowo} \tag{2.1.3} $$$
```

Ostatecznie po podstawieniu wartości liczbowych otrzymujemy układ równań przedstawiony w równaniu \$ref{obvod:liczbowo}\$.

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & -1 \\ 1 & 1 & 0 & 0 & 5 & 0 & 3 & -7 & 0 \\ 1 & 4 & 1 & 5 & 0 & 5 & -3 & 0 & -2 \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \end{bmatrix}$$

\end{bmatrix}

$$\begin{bmatrix} 10 \\ 0 \\ 0 \\ 5 \\ -8 \end{bmatrix} \text{label}{obvod:liczbowo}\text{tag}{2.1.3}$$

2.2 Normy wektorów

W algebrze liniowej, analizie funkcyjnej i pokrewnych dziedzinach matematyki, norma to funkcja przyporządkowująca dodatnią wartość liczbową określającą długość lub wielkość wektora, lub przestrzeni wektorowej gdy obliczamy normę macierzy. W ogólności normy są wykorzystywane do określania odległości między punktami wskazywanymi przez wektory oraz porównywania ich długości. Istnieje wiele różnych rodzajów norm (różnych funkcji), które mogą spełnić to zadanie. Aby funkcja przekształcająca wektor lub macierz w liczbę mogła być nazywana normą, to musi ona posiadać następujące właściwości:

1. $\|av\| = |a| \|v\|$ (skalowalność)
2. $\|u + v\| \leq \|u\| + \|v\|$ (nierówność trójkątna)
3. $\|v\| \geq 0$ (niewjemność)
4. Jeżeli $\|v\| = 0$ to $v = 0$, czyli v jest wektorem zerowym (jednoznaczność),

gdzie a to dowolna wartość skalarna, v i u to dowolne wektory. W dalszej części przedstawimy kilka najczęściej wykorzystywanych funkcji, które charakteryzują się przed chwilą wymienionymi właściwościami, zatem są normami macierzy:

Norma euklidesowa (L2-norm)

To najbardziej intuicyjna norma wśród norm wektorowych, która stanowi euklidesową długość wektora, czyli pierwiastek sumy kwadratów jego składowych x_i :
$$\sqrt{\sum_{i=1}^n x_i^2} \text{label}{eq:norma_euklidesowa}$$

p-norma

p -norma to funkcja zwracająca wartość skalarną zdefiniowaną jako pierwiastek stopnia sumy składowych wektora podniesionych do p -tej potęgi. W tym kontekście, zdefiniowana wcześniej norma euklidesowa jest szczególnym przypadkiem p -normy dla $p=2$.
$$\left(\sum_{i=1}^n |x_i|^p \right)^{1/p} \text{label}{eq:pnorma}$$

Norma nieskończoność

Przykład klasycznej normy, w której funkcja przyporządkowuje wektorowi wartość maksymalnej składowej x_i . Zaletą tej normy jest jej bardzo prosta i szybka implementacja oraz brak wrażliwości na błędy operacji arytmetycznych. Na rysunku 2.2 przedstawiona została interpretacja graficzna wymienionych norm indukowanych w przestrzeni wektorów dwuwymiarowych. Przedstawiono na nim „koła jednostkowe”, stanowiące zbiory punktów znajdujących się na końcach wszystkich wektorów, które dla poszczególnych zdefiniowanych norm uzyskują wartość równą jeden. Wektory są zaczepione w początku układu współrzędnych.

image-20221021185621838

Rys. 2.2. "Koła jednostkowe" dla trzech definicji norm: L_1 dla $p=1$, L_2 dla normy euklidesowej oraz L_∞ dla normy nieskończoności.

[Rysunek dla normy L_1 przedstawia obrócony o 45° kwadrat, którego wierzchołki są na osiach współrzędnych ox i oy , a jego środek ciężkości w początku układu współrzędnych. Środkowy rysunek dla normy L_2 przedstawia okrąg ze środkiem w początku układu współrzędnych. Trzeci, dolny rysunek dla normy L_∞ przedstawia kwadrat ze środkiem ciężkości w początku układu współrzędnych i bokami równoległymi do osi ox i oy .]

2.3 Normy macierzowe

Macierze często są nazywane operatorami liniowymi, które przekształcają wektory, np. za pomocą operacji $x' = Ax$. Przekształceniem może być wydłużenie lub skrócenie wektora. W tym kontekście, normy macierzowe są funkcjami, które przyporządkowują danej macierzy liczbę skalarną wyrażającą zdolność macierzy do wydłużania wektorów. Każda norma wektorowa pozwala nam zdefiniować normę macierzową, która wyraża maksymalne wydłużenie wektora jednostkowego w danej normie po przekształceniu przez macierz A . Takie normy nazywa się normami indukowanymi. Normy indukowane, zatem charakteryzują jak dana macierz A rozciąga / przekształca wektory jednostkowe w odniesieniu do

danej normy. Wyraża to równanie $\|\alpha\|_{\text{norma:indukowana}}$, które normie przyporządkowuje maksymalne wydłużenie dowolnego wektora \mathbf{x} przez macierz A . $\|\alpha\|_p = \max\{\|\mathbf{x}\|_p \mid \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_p = 1\}$ Podobnie do norm wektorowych, normy macierzowe spełniają podstawowe kryteria pozwalające na ich wykorzystanie do porównywania macierzy. Dla skalarów α w \mathbb{K} oraz wszystkich macierzy A i B w przestrzeni $\mathbb{K}^{m \times n}$,

- $\$|\alpha A| = |\alpha| |A|$ (skalowalność)
 - $|A+B| \leq |A| + |B|$ (nierówność trójkątna)
 - $|A| \geq 0$ (nieujemność)
 - $|A| = 0$ jeżeli $A = \{m, n\}$ (jednoznaczność)

Na rysunku 2.3 przedstawiono graficzną interpretację przekształcenia zbioru "kół jednostkowych" dla norm $\| \cdot \|_1$, $\| \cdot \|_2$ oraz $\| \cdot \|_{\infty}$ przez przykładową macierz: $A = \begin{bmatrix} 1 & 2 & 0 & 2 \end{bmatrix}$

image-20230106105700167

Rys. 2.3. "Koła jednostkowe" dla trzech definicji norm: L_1 dla p -normy i $p=1$, L_2 dla normy euklidesowej oraz L_{∞} dla normy nieskończoności.

Normy macierzowe są naturalnym rozszerzeniem notacji norm dedykowanej dla wektorów. Na początku wymienimy p -normy indukowane.

Norma \$L_1\$-maksimum $\|A\|_1 = \max \{ \sum_{j=1}^n |a_{ij}| : i = 1, \dots, m \}$

Norma $\$L_{\{\infty\}}$ - nieskończoność \$\$ |A|_{\{\infty\}} = \max \{ \sum_{j=1}^n |a_{ij}| : \text{label(norma:rieskonczonosc)} \} \tag{2.3.3} \$\$

Norma $\$L_2$ $\$ \sigma_{\max}(A) = \sqrt{\lambda_{\max}(A^H A)}$ gdzie σ_{\max} to maksymalna wartość osobienna macierzy A oraz λ_i oznacza wartości własne macierzy $A^H A$ (gdzie A^H to macierz Hermitowska, czyli transponowana macierz sprzężona spełniająca warunek $[a_{ij}] = \bar{[a_{ji}]}$).

Norma Frobenius $\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$

2.4 Współczynnik uwarunkowania

Powszechnym zastosowaniem norm macierzowych jest obliczanie współczynnika uwarunkowania macierzy. Współczynnik uwarunkowany macierzy $\$cond(A)$ informuje za pomocą liczby skalarnej jak bardzo dana macierz jest podatna na niedokładności danych wejściowych. Danymi wejściowymi są w tym przypadku wartości samej macierzy A oraz wartości wektora prawych stron b układu równań $Ax=b$. Współczynnik uwarunkowania określa jak bardzo zmiany wartości liczbowych przenoszą się na rozwiązanie układu równań.

Założmy, że \$e\$ jest błędem wektora prawych stron \$b\$ układu równań \$Ax=b\$ pm \$e\$, wówczas rozwiążanie jest również obarczone tym błędem i wyraża się równaniem: \$x + \Delta x = A^{-1}b + A^{-1}e\$ zatem stosunek względnego błędu rozwiązania \$\frac{\|\Delta x\|}{\|A^{-1}b\|} \leq \frac{\|A^{-1}e\|}{\|A^{-1}b\|}\$ do względnego błędu \$b\$ \$\frac{\|\Delta x\|}{\|b\|} \leq \frac{\|A^{-1}e\|}{\|b\|}\$ wyraża się: \$\frac{\|\Delta x\|}{\|b\|} = \frac{\|\Delta x\|}{\|A^{-1}b\|} \cdot \frac{\|A^{-1}b\|}{\|b\|} = \frac{\|A^{-1}e\|}{\|A^{-1}b\|} \cdot \frac{\|A^{-1}b\|}{\|b\|} = \frac{\|A^{-1}e\|}{\|b\|}\$ Wartość maksymalna tego stosunku może więc postrzegana jako iloczyn dwóch norm, zgodnie z poniższym przekształceniem: \$\max_{\{e,b\}} \left(\frac{\|A^{-1}e\|}{\|b\|} \right) = \max_{\{e\}} \left(\frac{\|A^{-1}e\|}{\|A^{-1}b\|} \right) \cdot \max_{\{b\}} \left(\frac{\|A^{-1}b\|}{\|b\|} \right)\$ Wartość maksymalna jest określana właśnie współczynnikiem uwarunkowania: \$\text{cond}(A) = \frac{\|A\|}{\|A^{-1}\|}\$. Wartość współczynnika uwarunkowania o wartości 1 nie wprowadza zatem żadnego dodatkowego błędu rozwiązania. Rozwiążanie jest zatem nie gorsze od samych danych wejściowych. Wartość współczynnika uwarunkowania zależy od przyjętych norm. Najczęściej stosowane definicje to:

- współczynnik uzależniony od wartości osobliwych gdy zastosujemy normę $\|A\|_{cond} = \frac{\sigma_{max}(A)}{\sigma_{min}(A)}$
 - współczynnik uzależniony od wartości minimalnych i maksymalnych gdy macierz jest trójkątna i zastosujemy normę $\|A\|_{cond} = \frac{\max(|a_{ii}|)}{\min(|a_{ii}|)}$

Przykład 2.2

Przeanalizujmy wpływ niedokładności danych wejściowych dla przykładowego układu równań wraz z obliczeniem współczynnika uwarunkowania. Rozważamy układ równań $\begin{aligned} Ax = b \\ \mathbf{A} &= \begin{bmatrix} 1 & 2 & 3 \\ 7 & 1 & 9 \\ 4 & 5 & 6 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 999 \\ 999 \\ 999 \end{bmatrix} \end{aligned}$. Współczynnik uwarunkowania tej macierzy możemy obliczyć wyznaczając go z normy L_{∞} w następujących krokach: $\begin{aligned} \text{cond}(\mathbf{A}) &= \|A^{-1}\| \|A\| \approx \max_i |\sum_j a_{ij}| = 5,999 \\ A^{-1} &= \frac{1}{\det(\mathbf{A})} \mathbf{A}^T D = \frac{1}{(3.999 - 2 \cdot 2)} \begin{bmatrix} 3.999 & -2 & -2 \\ -2 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 3.999 & -2 & -2 \\ -2 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \\ \|A^{-1}\| \|A\| &= 5,999 * 5999 = 35988 \end{aligned}$. Współczynnik uwarunkowania dla tej macierzy o wymiarach 2×3 jest stosunkowo bardzo duży. Oznacza to, że spodziewamy się znacznego wpływu niedokładności na wynik rozwiązania.

Sprawdźmy to.

Obliczymy rozwiązanie układu równań przy założeniu oryginalnego wektora $b = \begin{bmatrix} 4 \\ 7,999 \end{bmatrix}$. Wówczas rozwiązanie układu równań możemy obliczyć odejmując pierwszy wiersz od drugiego: $\begin{aligned} x_1 + 2x_2 &= 4 \\ 2x_1 + 3,999x_2 &= 7,999 \end{aligned}$

$$\begin{aligned} -2x_1 - 4x_2 &= -8 \\ 2x_1 + 3,999x_2 &= 7,999 \end{aligned}$$

$\circ \quad -0,001x_2 = -0,001 \rightarrow x_2 = 1$

następnie $x_1 = 4 - 2x_2 = 4 - 2 \cdot 1 = 2$

Zmieśmy teraz nieznacznie wektor prawych stron b i przyjmijmy: $b = \begin{bmatrix} 4 \\ 8 \end{bmatrix}$. Wówczas analogicznie przeprowadzająca obliczenia jak powyżej otrzymamy wynik: $x_1 = 0$. Jak można zauważyć, niewielka zmiana o wartość \$0,001\$ wektora b spowodowała bardzo dużą zmianę rozwiązania. **Mówimy o takim układzie równań, że jest źle uwarunkowany.**

Zmieśmy współczynniki macierzy A : $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 7 & 1 & 9 \\ 4 & 5 & 6 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} 999 \\ 999 \\ 999 \end{bmatrix}$. Gdybyśmy powtórzyli obliczenia współczynnika uwarunkowania dla tej macierzy to otrzymalibyśmy wartość $\text{cond}(A) = 25$, która jest mniejsza od poprzedniej o ponad 100 razy (dwa rzędy). Widać, że macierz ta jest znacznie lepiej uwarunkowana. Powtórzyliśmy serię obliczeń dla oryginalnego $b = \begin{bmatrix} 999 \\ 999 \\ 999 \end{bmatrix}$ oraz zmodyfikowanego $b = \begin{bmatrix} 8 \\ 8 \\ 8 \end{bmatrix}$ i porównajmy wyniki.

dla $b = \begin{bmatrix} 999 \\ 999 \\ 999 \end{bmatrix}$ otrzymamy wynik $x = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$,
a dla $b = \begin{bmatrix} 8 \\ 8 \\ 8 \end{bmatrix}$ otrzymamy $x = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$.

Jak widać, tym razem wyniki rozwiązania różnią się nieznacznie.

2.5 Metoda eliminacji Gaussa

Eliminacja Gaussa jest podstawowym sposobem rozwiązywania układu równań linowych. W eliminacji Gaussa wykorzystujemy operacje wierszowe dodawania do siebie wierszy układu równań. Wykonujemy jednak te operacje w ścisłe określony sposób i kolejności. Zasadniczym celem eliminacji Gaussa jest przekształcenie układu równań do postaci górnopróbkowej. To znaczy, że wszystkie współczynniki macierzy A układu równań poniżej diagonali są zerowane. Uważny czytelnik zauważał zapewne, że mówimy o eliminacji elementów macierzy A , ale operacje wykonujemy na całych wierszach układu równań. Nie możemy zatem zapomnieć o wektorze prawych stron b . W celu uproszczenia implementacji algorytmu zazwyczaj tworzy się zazwyczaj macierz rozszerzoną, która powstaje poprzez dołączenie wektora prawych stron b dodatkowej kolumny do macierzy A . $\begin{bmatrix} A & b \end{bmatrix} \rightarrow [A \mid b]$

Następnie przekształcamy macierz A do postaci górnopróbkowej za pomocą dodawania wierszy 'diagonalnych' wymnożonych przez stosowne współczynniki skalujące (odejmujemy wiersze 'diagonalne' od wierszy poniżej). Kolejność operacji (zaprezentowana na rysunku 2.4) jest następująca:

- najpierw odejmujemy pierwszy wiersz macierzy A od drugiego wiersza wymnożony przez współczynnik, który spowoduje po odjęciu wyzerowanie elementu a_{21} , równy $\left(\frac{a_{21}}{a_{11}}\right)$,
- w kolejnych kilku krokach odejmujemy pierwszy wiersz macierzy wymnożony przez stosowne współczynniki, aż wyzerowane zostaną wszystkie elementy poniżej diagonali,
- dalej, przechodzimy do zerowania elementów poniżej diagonali w drugiej kolumnie,
- itd., aż wyzerujemy wszystkie elementy poniżej diagonali.

Powyższy schemat został zilustrowany na rysunku 2.4.

image-20230106120225859

Rysunek 2.4 Ilustracja przebiegu eliminacji Gaussa

Przykład 2.3

Przeprowadź eliminację Gaussa dla poniższego układu równań z trzema niewiadomymi. $\begin{aligned} x_1 + 3x_2 + 4x_3 &= 2 \\ -2x_1 + 2x_2 + 3x_3 &= -1 \\ x_1 + x_2 + 2x_3 &= 3 \end{aligned}$ Powyższy układ równań możemy zapisać w postaci macierzowej: $\begin{bmatrix} 1 & 3 & 4 \\ -2 & 2 & 3 \\ 1 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$

\end{bmatrix}

$\begin{bmatrix} 2 & -1 & 3 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$ Następnie możemy odjąć pierwszy wiersz od kolejno drugiego i trzeciego: $\begin{bmatrix} 1 & 3 & 4 \\ -2 & 2 & 3 \\ 1 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$

\end{bmatrix}

$\begin{bmatrix} 2 & -1 & 3 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$ W wyniku otrzymamy: $\begin{bmatrix} 1 & 3 & 4 \\ 0 & 8 & 11 \\ 0 & 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$

\end{bmatrix}

$\begin{bmatrix} 1 & 3 & 4 \\ 0 & 8 & 11 \\ 0 & 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$ Kolejnym krokiem jest odjęcie drugiego wiersza od trzeciego w celu eliminacji elementu a_{32} : $\begin{bmatrix} 1 & 3 & 4 \\ 0 & 8 & 11 \\ 0 & 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$

\end{bmatrix}

$\begin{bmatrix} 1 & 3 & 4 \\ 0 & 8 & 11 \\ 0 & 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$ Ostatecznie otrzymuje górną-trójkątną postać układu równań: $\begin{bmatrix} 1 & 3 & 4 \\ 0 & 8 & 11 \\ 0 & 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$

\end{bmatrix}

$\begin{bmatrix} 2 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$

W celu implementacji komputerowej algorytmu eliminacji Gaussa warto posłużyć się zapisem algorytmicznym procedury. Zakładając, że operujemy na macierzy rozszerzonej \mathbf{Ag} , procedura ma postać:

```

1. function Ag = gaussian(A, b)
2.     Ag = [A b]
3.     n = size(Ag, 1);
4.     for k=1:n-1
5.         for i = k+1:n
6.             l = Ag(i, k) / Ag(k, k);
7.             for j = k+1:n
8.                 Ag(i, j) = Ag(i, j) - l * Ag(k, j);
9.             end
10.        end
11.    end
12. end

```

Tak sformułowany algorytm pozwala nam wygodnie zaimplementować funkcje w MATLAB, która przyjmuje jako argument macierz A oraz wektor prawych stron b , a zwraca macierz rozszerzoną Ag , która jest górną-trójkątna. Proszę zwrócić uwagę, na bardzo podobne oznaczenia, które ułatwiają interpretację kodu.

```

1. function Ag = gaussian(A, b)
2.     Ag = [A b]
3.     n = size(Ag, 1);
4.     for k=1:n-1
5.         for i = k+1:n
6.             l = Ag(i, k) / Ag(k, k);
7.             for j = k+1:n
8.                 Ag(i, j) = Ag(i, j) - l * Ag(k, j);
9.             end
10.        end
11.    end
12. end

```

Jednym z bardzo ważnych elementów, których nie możemy pominąć jest wrażliwość algorytmu na wystąpienie w dowolnym momencie procesu zera na diagonali, które będzie skutkowało dzieleniem przez zero podczas obliczania współczynników wykorzystywanego do eliminacji. W celu eliminacji tego problemu stosuje się **selekcję elementu głównego**.

Selekcja elementu głównego polega na takiej zamianie wierszy macierzy rozszerzonej Ag , aby w kolejnym kroku algorytmu na diagonali znalazły się **maksymalny co modułu element**, jednocześnie pozostawiając już wyzerowane elementy nadal zerowymi.

Występują trzy rodzaje selekcji:

1. w kolumnie (selekcja częściowa)- najprostsza wymaga tylko zamiany równań (rysunek 2.5a),
2. w wierszu (selekcja częściowa) - zamieniamy kolejność niewiadomych w wektorze x (rysunek 2.5b),

3. podmacierz A[r:end, c:end] (selekcja pełna) - zamieniamy zarówno wiersze jak i kolejność zmiennych w wektorze \$x\$ (rysunek 2.5c).

image-20230106140428588

Rys. 2.5 Elementy macierzy, w których poszukuje się wartości maksymalnych co do modułu w przypadku procedury selekcji elementu głównego:
a) dla selekcji częściowej w kolumnie, b) dla selekcji częściowej w wierszu, c) dla selekcji pełnej w bloku macierzy.

Przykład 2.4

Rozwiążemy układ równań stosując skończoną precyżję sztucznie zaokrąglając wyniki obliczeń arytmetycznych aby pokazać wpływ selekcji elementu głównego również na dokładność rozwiązania.

Na początku rozwiążmy układ z następującym układem wierszy.

$$\begin{bmatrix} 0.003 & 59.1 & -6.13 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

\end{bmatrix}

$$\begin{bmatrix} 59.17 & 46.78 \end{bmatrix}$$

$$m = \frac{5.291}{0.003} = 1763.666... \approx 1764.0$$

$$\begin{bmatrix} 0.003 & 59.1 & 0 & -104200 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

\end{bmatrix}

$$\begin{bmatrix} 59.17 & -104300 \end{bmatrix}$$

$$x_2 = \frac{-104300}{-104200} \approx 1.001$$

$$x_1 = \frac{59.17 - 59.1 \cdot 1.001}{0.003} \approx 3.633$$

Teraz, powtórzmy obliczenia, ale wcześniej zamieniając miejscami wiersz pierwszy i drugi układu równań.

$$\begin{bmatrix} 5.291 & -6.13 & 0.003 & 59.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

\end{bmatrix}

$$\begin{bmatrix} 46.78 & 59.17 \end{bmatrix}$$

$$m = \frac{0.003}{5.291} \approx 0.000567$$

$$\begin{bmatrix} 5.291 & -6.13 & 0 & 59.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

\end{bmatrix}

$$\begin{bmatrix} 46.78 & 59.14 \end{bmatrix}$$

$$x_2 = \frac{59.14}{59.1} \approx 1.001$$

$$x_1 = \frac{46.78 - (-6.13) \cdot 1.001}{5.291} \approx 10.001$$

Widzimy, że zmiana wierszy (zaznaczmy, że przy zastosowaniu sztucznie zauważonego błędu zaokrągleń) dała nam wynik znacznie inny od poprzedniego. Który wynik jest poprawny? Ten drugi, ponieważ na diagonali znajdował się największy co do modułu w danej kolumnie element.

Wykonanie tych obliczeń w MATLABie (nawet z oryginalnym układem wierszy) daje nam wynik drugi. Po pierwsze, MATLAB zamienia wiersze automatycznie, po drugie operacje wykonane są z dużo większą precyżją ($\$eps \approx 10^{-14}$).

```

1. A = [0.003 59.1;
2.      5.291 -6.13];
3. b = [59.17;
4.      46.78];
5. A\b
6.
7. ans =
8.
9.      10.0008
10.     1.0007

```

2.6 Wsteczne podstawienie

Jeżeli macierz rozszerzona reprezentująca nasz układ równań jest już w postaci trójkątnej, to wynik rozwiązania bardzo łatwo znaleźć stosując

procedurę wstecznego podstawienia. Przyjrzyjmy się przykładowemu układowi równań w postaci górnorójkątnej: $\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ & b_1 & 0 \\ & a_{22} & a_{23} \\ & b_2 & 0 \\ & 0 & a_{33} \\ & b_3 & 0 \end{bmatrix}$. Mając taką postać, procedura kolejno znajduje wartości x_{ij} zaczynając od ostatniego, czyli dla $i=n, n-1, \dots, 1$. Otrzymujemy zatem: $x_3 = \frac{b_3}{a_{33}}$, $x_2 = \frac{b_2 - a_{23}x_3}{a_{22}}$, $x_1 = \frac{b_1 - a_{13}x_3 - a_{12}x_2}{a_{11}}$. Zauważmy, że w kolejnych wierszach wykorzystujemy wartości x_{ij} obliczone wcześniej. Powyższą procedurę możemy uogólnić dla macierzy górnoprójkątnej o dowolnym rozmiarze i przedstawić matematycznie: $x_i = \frac{c_i - \sum_{j=i+1}^n u_{ij} x_j}{u_{ii}}$ dla $i = n, n-1, \dots, 1$.

Zaczynając tym razem od pierwszego wiersza: $x_i = \frac{c_i - \sum_{j=1}^{i-1} u_{ij} x_j}{u_{ii}}$ dla $i = 1, 2, \dots, n$.

```

1. % U - macierz górnoprójkątna, c - wektor prawych stron
2. function x = wsteczne_gornoprójkątna(U,c)
3.     n = size(U,1);
4.     x = zeros(n,1);
5.     for i = n:-1:1
6.         s = 0;
7.         for j = i+1:n
8.             s = s + U(i,j)*x(j);
9.         end
10.        x(i) = (c(i) - s) / U(i,i);
11.    end
12. end
13.
14. % U - macierz dolnotrójkątna, c - wektor prawych stron
15. function x = wsteczne_dolnotrójkątna(L,c)
16.     n = size(L,1);
17.     x = zeros(n,1);
18.     for i = 1:n
19.         s = 0;
20.         for j = 1:i-1
21.             s = s + L(i,j)*x(j);
22.         end
23.         x(i) = (c(i) - s) / L(i,i);
24.     end
25. end

```

2.7. Eliminacja Gaussa-Jordana

Eliminacja Gaussa-Jordana jest rozwinięciem eliminacji Gaussa o dodatkowe kroki. W eliminacji tej oprócz zerowania elementów poniżej diagonali układu równań, eliminujemy również elementy powyżej diagonali. Dodatkowo, skalujemy wartości we wszystkich wierszach, tak aby po eliminacji na diagonali były same jedynki. Pamiętając, że operacje, które wykonujemy są operacjami wierszowymi to rozwiązania układu równań oryginalnego i układu po eliminacji są takie same. Łatwo zauważyc, że po eliminacji Gaussa-Jordana, ponieważ w części macierzy A będziemy mieli wyzerowane wszystkie elementy poniżej i powyżej diagonali oraz na diagonali będą wartości 1 , to ostatnia kolumna macierzy rozszerzonej będzie rozwiązaniem układu równań. Nie potrzebujemy zatem wstecznego podstawienia. Pozornie, algorytm wydaje się być szybszy obliczeniowo, ale z uwagi na konieczność wyzerowania elementów powyżej diagonali jego złożoność jest praktycznie taka sama jak eliminacji Gaussa połączonej z wstecznym podstawieniem. Niemniej występują sytuacje, że zastosowanie eliminacji Gaussa-Jordana jest korzystne. Jedną z nich jest np. konieczność obliczenia macierzy odwrotnej.

Przebieg algorytmu eliminacji Gaussa-Jordana jest następujący:

1. Inicjalizacja (definicja macierzy rozszerzonej) $\begin{aligned} a_{ij}^{(0)} &= a_{ij} \text{ dla } i=1,2, \dots, n; j=1,2, \dots, n \\ &= b_i \text{ dla } i=1,2, \dots, n+1 \end{aligned}$
2. Normalizacja (znormalizować element diagonalny do wartości 1) $a_{kj}^{(k)} = \frac{a_{kj}}{a_{kk}}$ dla $j=k, k+1, \dots, n+1$
3. Redukcja (zredukować wszystkie elementy pozadiagonalne w kolumnie k) $a_{ij}^{(k)} = a_{ij} - a_{ik} \cdot a_{kj}^{(k)}$ dla $j=k, k+1, \dots, n+1$; $i=1,2, \dots, n$ (and $j \neq k$)

Kroki 2-3 muszą zostać wykonane dla wszystkich kolumn $k=1,2, \dots, n$.

Przykład 2.5

Znajdź rozwiązanie układu równań przedstawionego w postaci macierzy rozszerzonej.

1. W pierwszym kroku podzielimy wszystkie elementy pierwszego wiersza przez wartość elementu diagonalnego \$2\$.
 2. W drugim kroku odejmiemy pierwszy wiersz od drugiego wymnożony przez \$4\$.
 3. W trzecim kroku odejmiemy pierwszy wiersz od trzeciego wymnożony przez \$3\$.
 4. W czwartym kroku podzielimy elementy w drugim wierszu przez wartość diagonalną równą \$5\$.
 5. W piątym kroku odejmiemy drugi wiersz od pierwszego wymnożony przez \$\frac{-1}{2}\$.
 6. W szóstym kroku odejmiemy drugi wiersz od trzeciego wymnożony przez \$\frac{7}{2}\$.
 7. ... kontynuując obliczenia otrzymamy ostateczny wynik w czwartej kolumnie macierzy rozszerzonej: \$x=[1,2,4]\$.

Jak wcześniej wspominano, eliminację Gaussa-Jordana można skutecznie wykorzystać do obliczania macierzy odwrotnej o niewielkim rozmiarze. Aby wyprowadzić ten sposób przypomnijmy definicję macierzy odwrotnej: $\mathbf{A}^{-1} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} a'_{11} & a'_{12} & \cdots & a'_{1n} \\ a'_{21} & a'_{22} & \cdots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a'_{m1} & a'_{m2} & \cdots & a'_{mn} \end{bmatrix}$

$\begin{bmatrix} 1 & 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & & & & \vdots & & \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \end{bmatrix}$
\label{def:maceirz_odwrotna} tag{2.7.1} Zauważmy, że w tym równaniu pierwszą i kolejne kolumny macierzy A^{-1} możemy oznaczyć jako zmienne x_{ij} :

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{bmatrix} = I_m$$

\begin{bmatrix} 1 & 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & & & & \vdots & & \\ 0 & 0 & \cdots & 1 \end{bmatrix} \notag Przy takim zapisie, okazuje się, że aby znaleźć kolejne kolumny macierzy odwrotnej A^{-1} wystarczy rozwiązać n niezależnych układów równań, w których niewiadomymi będą kolumny macierzy A^{-1} a wektorami prawych stron kolejne kolumny macierzy jednostkowej. $\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(m)} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

```

}
\begin{bmatrix} 1 & 0 & \vdots & 0 \end{bmatrix} \notag \\

\$ \$ \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}
```

↳ Using `ifelse` to create a `0` or `1` vector to store `0`'s and `1`'s in a matrix.

i tak dalej

Możemy jednak podejść do tego zagadnienia skuteczniej i zamiast rozwiązywać \$n\$ osobnych układów równań rozwiążemy jeden ale z wszystkimi wektorami prawych stron (całą macierzą jednostkową) doklejoną do macierzy A . $\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$ Wykorzystamy do tego metodę eliminacji Gaussa-Jordana, w wyniku której przekształcimy układ do postaci $\begin{bmatrix} 1 & 0 & \cdots & 0 \\ a'_{12} & a'_{13} & \cdots & a'_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a'_{m1} & a'_{m2} & \cdots & a'_{mn} \end{bmatrix}$

$\&a'_{22}\&\cdots\&a'_{2n} \ \vdots\&\vdots\&\ddots\&\vdots\&\vdots\&\ddots\&\vdots\& 0\&0\&\cdots\&1 \ &a'_{m1}\&a'_{m2}\&\cdots\&a'_{mn}$
 $\end{bmatrix} \notag \quad \text{gdzie współczynniki } a'_{ij} \text{ stanowią współczynniki szukanej macierzy odwrotnej.}$

Przykład 2.6

Wykorzystując eliminację Gaussa-Jordana znajdź macierz odwrotną do A :

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 4 & 5 \end{bmatrix}$$

Rozwiązanie

$$\left[\begin{array}{cccc} 2 & 1 & 4 & 5 \\ 0 & 0 & 1 & 0 \end{array} \right] \rightarrow \left[\begin{array}{cccc} 1 & \frac{1}{2} & 2 & 0 \\ 0 & 0 & 1 & 0 \end{array} \right] \rightarrow \left[\begin{array}{cccc} 1 & \frac{1}{2} & 0 & -4 \\ 0 & 0 & 1 & 0 \end{array} \right] \rightarrow \left[\begin{array}{cccc} 1 & 0 & 0 & -5 \\ 0 & 0 & 1 & 0 \end{array} \right]$$

$$\left[\begin{array}{cccc} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & 0 \end{array} \right] \rightarrow \left[\begin{array}{cccc} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & 0 \end{array} \right] \rightarrow \left[\begin{array}{cccc} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & 0 \end{array} \right]$$

Odpowiedź

$$\mathbf{A}^{-1} = \begin{bmatrix} 0 & -5 \\ 0 & 0 \end{bmatrix}$$

2.8 Rozkład na czynniki - faktoryzacja macierzy

Innym sposobem rozwiązywania układów równań jest wstępny rozkład na czynniki macierzy A . W algebrze liniowej stosuje się powszechnie kilka rodzajów faktoryzacji:

1. Faktoryzacja LU: $A=L\cdot U$, macierz L jest macierzą dolno-trójkątną, a macierz U górną-trójkątną,
2. Faktoryzacja QR: $A=Q\cdot R$, macierz Q jest macierzą ortonormalną $Q^T Q=I$ $\Rightarrow Q^{-1}=Q^T$, macierz R jest macierzą górną-trójkątną,
3. Faktoryzacja SVD: $A=S\cdot V\cdot U^T$, macierze S, V są ortogonalne, a U diagonalna.

W ramach niniejszego podręcznika przyjrzymy się wyłącznie faktoryzacji LU. Zakładając w właściwości górną i dolno-trójkątne macierzy L i U , oryginalny układ równań możemy zapisać następująco: $Ax=b$ $\Rightarrow Lx=b$ $\Rightarrow Ux=b$. Rozwiążanie możemy uzyskać dwukrotnie stosując wsteczne podstawienie. Najpierw podstawimy $Ux=y$, wówczas uzyskamy $Ly=b$. Macierz L jest dolno-trójkątna więc rozwiązanie tego równania możemy szybko znaleźć stosując wsteczne podstawienie od góry. Znając wynik y , możemy przejść do drugiego etapu, korzystając z wprowadzonego wcześniej podstawienia: $Ux=y$ $\Rightarrow x$. Wykorzystując ponownie wsteczne podstawienie uzyskamy ostateczne rozwiązanie.

2.8.1 Metoda Doolittle'a

Pozostaje pytanie jak skutecznie znaleźć rozkład LU. Pierwszym podejściem jest zastosowanie **metody Doolittle'a**. Aby wyprowadzić tą metodę posłużmy się pełnym zapisem faktoryzacji LU: $A=LU$.

$\left[\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{array} \right] = \left[\begin{array}{cccc} 1 & 0 & \cdots & 0 \\ l_{11} & l_{12} & \cdots & l_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{m1} & l_{m2} & \cdots & l_{mn} \end{array} \right] \left[\begin{array}{cccc} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1} & u_{m2} & \cdots & u_{mn} \end{array} \right]$

1. Najpierw wyznaczmy współczynniki pierwszego wiersza macierzy U . Zgodnie ze schematem na rysunku 2.6 wartości współczynników macierzy A w pierwszym siedem równe:

$$a_{11} = u_{11} + u_{12} + \cdots + u_{1n} \quad a_{12} = u_{11} + u_{12} + \cdots + u_{1n} \quad \dots \quad a_{1n} = u_{11} + u_{12} + \cdots + u_{1n}$$

Stąd błyskawicznie (z uwagi na większość zer), możemy wyznaczyć $u_{1i}=a_{1i}$ dla $i=1,2,\dots,n$.

W kolejnym drugim kroku wyznaczmy współczynniki pierwszej kolumny macierzy L :
 $a_{11} = u_{11} + u_{12} + \cdots + u_{1n}$
 $a_{21} = u_{21} + u_{22} + \cdots + u_{2n}$
 \dots
 $a_{m1} = u_{m1} + u_{m2} + \cdots + u_{mn}$

Stąd łatwo wyprowadzić wyrażenia na wartości $l_{11}, l_{21}, \dots, l_{m1}$ zakładając, że wartość u_{11} została już obliczona w poprzednim kroku. W trzecim kroku algorytmu wróćmy do macierzy U , tym razem do drugiego wiersza, analogicznie wyprowadzając wzory na współczynniki z operacji mnożenia wiersza razy kolumna:
 $u_{12} = a_{12} - l_{11}u_{11}$
 $u_{22} = a_{22} - l_{21}u_{11}$
 \dots
 $u_{m2} = a_{m2} - l_{m1}u_{11}$

image-20230106143609520

Rys. 2.6 Schemat z kolejnością obliczeń wierszy i kolumn macierzy $\$L\$$ i $\$U\$$ w faktoryzacji $\$LU\$$

Wykorzystując ten schemat możemy określić algorytm faktoryzacji LU metodą Doolittle'a w następujący sposób

1. pierwszy wiersz U kopujemy z pierwszego wiersza A $\$u_{\{1\}} = a_{\{1\}}\text{\textbackslash text\{ dla \} } i=1,2,\text{\textbackslash ldots,n\$}$
 2. pierwsza kolumna L obliczana jest za pomocą: $\$L_{\{i\}} = a_{\{i\}} / u_{\{1\}}\text{\textbackslash text\{ dla \} } i=2,3,\text{\textbackslash ldots,n\$}$
 3. następnie dla każdej pary $(\$i^{\{ty\}}\$ wiersz \$U\$) oraz $(\$i^{\{ta\}}\$ kolumna \$L\$)$:$

```
 $$ \begin{aligned} & \text{\textbackslash begin\{align<em>\}} \text{\textbackslash text\{dla } i=2,3,\text{\ldots},n \& \text{\textbackslash u\_{ik}} = a_{ik} - \sum \limits_{j=1}^{i-1} l_{ij} u_{jk} \right) \text{\textbackslash text\{ for } k=i+1,\text{\ldots},n \text{\textbackslash l\_{ki}} \\ & \& = \frac{(a_{ik} - \sum \limits_{j=1}^{i-1} l_{ij} u_{jk})}{(u_{ii})} \text{\textbackslash text\{ for } k=i+1,i+2,\text{\ldots},n \end{aligned} \text{\textbackslash end\{align</em>\}} $$
```

Implementacja tego kodu w postaci funkcji MATLABa jest następująca.

```

1. function [L, U] = doolittle( A )
2. n = size( A,1 );
3. L = eye( n ); % inicjujemy macierz jednoscikowa, poniewaz zawsze na diagonali sa jedynki
4. U = zeros( n ); % pusta (na razie) macierz gornotrojkatna
5. U( 1, : ) = A( 1, : ); % kopujemy pierwszy wiersz
6. L( 2 : n, 1 ) = A( 2 : n, 1 ) / U( 1,1 ); % obliczamy pierwsza kolumna
7.
8. % wykonyujemy parami obliczenia kolejno wierszy U i kolumn L
9. for i = 2:n
10.     for k = i:n
11.         s = 0;
12.         for j=1:i-1
13.             s = s + L( i,j ) *U( j,k );
14.         end
15.         U( i,k ) = A( i,k ) - s;
16.     end
17.     for k = i+1:n
18.         s = 0;
19.         for j=1:i-1
20.             s = s + L( k,j ) *U( j,i );
21.         end
22.         L( k,i ) = ( A( k,i ) - s ) / U( i,i );
23.     end
24. end
25. end

```

Przykład 2.7

Wykorzystując przykładową, powyższą funkcję MATLABa oraz wcześniej zdefiniowane funkcje do wstecznego podstawienia znajdź rozwiązańek układu równań: $\begin{bmatrix} 1 & -1 & 1 & 1 \\ 4 & 3 & -1 & 2 \\ 3 & 2 & 2 & 5 \\ 8 & 9 & 5 & 8 \end{bmatrix} \begin{bmatrix} 4 \\ 6 \\ 15 \\ 1 \end{bmatrix}$ Rozwiązanie:

```

1. function L04_lu
2. A = [1 -1 1 1
3.      4 3 -1 2
4.      3 2 2 5
5.      8 9 5 8];
6. b = [4 6 15 1]'
7. [L, U] = doolittle( A )
8. A = L*U % powinna być macierz zerowa
9. y = wsteczne_dolnotrojkatne( L,b )
10. x = wsteczne_gornotrojkatne( U, y )
11.
12. % sprawdzam rozwiązanie - norma powinna być zero
13. norm( A*x-b )
14. end
15.
16. L =
17.    1.0000      0      0      0
18.    4.0000    1.0000      0      0
19.    3.0000    0.7143    1.0000      0
20.    8.0000    2.4286    3.5556    1.0000
21.
22. U =
23.    1.0000   -1.0000    1.0000    1.0000
24.      0    7.0000   -5.0000   -2.0000
25.      0      0    2.5714    3.4286
26.      0      0      0   -7.3333
27.
28. ans =
29.    0      0      0      0
30.    0      0      0      0
31.    0      0      0      0

```

2.8.2 Metoda eliminacji Gaussa

Drugim, najbardziej użytecznym z praktycznego punktu widzenia sposobem jest wykorzystanie eliminacji Gaussa. Znaczenie tego podejścia jest bardzo istotne, gdyż pozwala na selekcję elementu głównego w kolejnych krokach metody. Metoda Doolittle'a nie pozwala na to. Dzięki selekcji jesteśmy zabezpieczeni przed dzieleniem przez zero na diagonali oraz redukujemy błędy zaokrągleń.

Algorytm faktoryzacji LU z wykorzystaniem eliminacji Gaussa jest bardzo prosty do implementacji. Pomijając szczegóły związane z wyprowadzeniem (wyraziliśmy operacje wierszowe za pomocą operatorów macierzowych oraz metodą Gaussa-Jordana wyprowadziliśmy macierze odwrotne tych operatów) przedstawmy algorytm.

Faktoryzacja LU metodą eliminacji Gaussa przebiega zgodnie z procesem eliminacji, ale w trakcie procesu współczynniki L_{ij} , które używaliśmy do wymnożenia macierzy diagonalnych podczas zerowania elementów zapamiętujemy w odpowiednich miejscach i,j macierzy L . Macierz wynikowa eliminacji Gaussa staje się wynikową macierzą U .

Zatem, współczynnik L_{21} z rysunku 2.7 wstawiamy w miejsce L_{11} macierzy docelowej L (patrz rysunek 2.8).

image-20230106153458673

Rys. 2.7 Ilustracja operacji odejmowania pierwszego wiersza macierzy w trakcie eliminacji Gaussa od wiersza drugiego, z zaznaczonym współczynnikiem L_{21} , który wykorzystywany jest do wstawienia do macierzy L

image-20230106153620984

Rys. 2.8 Ilustracja z zaznaczonym elementem L_{21} macierzy L

Implementacja faktoryzacji LU z wykorzystaniem eliminacji Gaussa w środowisku MATLAB (bez selekcji elementu głównego) została przedstawiona poniżej.

```
1. function [L, U] = lu_gaussian(A)
2.     n = size( A, 1 );
3.     L = eye( n );
4.     for j = 1:n-1
5.         for i = j+1:n
6.             f = A( i,j ) / A( j,j );
7.             A(i, :) = A(i, :) - f*A(j,: );
8.             L(i, j) = f; % tutaj zapamietujemy współczynnik
9.         end
10.    end
11.    U = A;
12. end
```

Przykład 2.8

Wykorzystaj implementację faktoryzacji LU na przykładowej macierzy losowej o rozmiarach 4×4 . Sprawdź wynik faktoryzacji obliczając normę $\|LU-A\|$.

Rozwiązanie:

```
1. function l05_lu_gaussian
2.     A = rand( 4 )
3.     [L, U] = lu_gaussian( A )
4.     norm( L * U - A, 2 )
5. end
6.
7. >> l05_lu_gaussian
8. A =
9. 0.8147  0.6324  0.9575  0.9572
10. 0.9058  0.0975  0.9649  0.4854
11. 0.1270  0.2785  0.1576  0.8003
12. 0.9134  0.5469  0.9706  0.1419
13.
14. L =
15. 1.0000      0      0      0
16. 1.1118  1.0000      0      0
17. 0.1559  -0.2972  1.0000      0
18. 1.1211  0.2676  3.5869  1.0000
19.
20. U =
21. 0.8147  0.6324  0.9575  0.9572
22. 0  -0.6055  -0.0996  -0.5788
23. 0      0  -0.0212  0.4791
24. 0      0      0  -2.4948
25.
26. ans =
27. 2.2204e-16
```

normy

2 Rozdział

Strona: [LeIA](#)

Kurs: Metody numeryczne (2025Z)

Książka: 2 Rozdział

Wydrukowane przez użytkownika: Kinga Kondraciuk

Data: niedziela, 30 listopada 2025, 14:01

Spis treści

- 1. Normy wektorów**
- 2. Numeryczna algebra liniowa**
- 3. Normy macierzowe**
- 4. Współczynnik uwarunkowania**
- 5. Metoda eliminacji Gaussa**
- 6. Wsteczne podstawienie**
- 7. Eliminacja Gaussa-Jordana**
- 8. Rozkład na czynniki - faktoryzacja macierzy**
 - 8.1. Metoda Doolittle'a
 - 8.2. Metoda eliminacji Gaussa

1. Normy wektorów

W algebrze liniowej, analizie funkcyjnej i pokrewnych dziedzinach matematyki, norma to funkcja przyporządkowująca dodatnią wartość liczbową określającą długość lub wielkość wektora, lub przestrzeni wektorowej gdy obliczamy normę macierzy. W ogólności normy są wykorzystywane do określania odległości między punktami wskazywanymi przez wektory oraz porównywania ich długości. Istnieje wiele różnych rodzajów norm (różnych funkcji), które mogą spełnić to zadanie. Aby funkcja przekształcająca wektor lub macierz w liczbę mogła być nazywana normą, to musi ona posiadać następujące właściwości:

1. $\|av\| = |a|\|v\|$ (skalarność)
2. $\|u + v\| \leq \|u\| + \|v\|$ (nierówność trójkątna)
3. $\|v\| \geq 0$ (niewjemność)
4. Jeżeli $\|v\| = 0$ to $v = 0$, czyli v jest wektorem zerowym (jednoznaczność),

gdzie a to dowolna wartość skalarna, i to dowolne wektory. W dalszej części przedstawimy kilka najczęściej wykorzystywanych funkcji, które charakteryzują się przed chwilą wymienionymi właściwościami, zatem są normami macierzy:

Norma euklidesowa (L2-norm)

To najbardziej intuicyjna norma wśród norm wektorowych, która stanowi euklidesową długość wektora, czyli pierwiastek sumy kwadratów jego składowych x_i

$$\|x\|_2 := \sqrt{x_1^2 + \dots + x_n^2} \quad (2.1.4)$$

p-norma

p -norma to funkcja zwracająca wartość skalarną zdefiniowaną jako pierwiastek p -tego stopnia sumy składowych wektora podniesionych do p -tej potęgi. W tym kontekście, zdefiniowana wcześniej norma euklidesowa jest szczególnym przypadkiem p -normy dla $p = 2$.

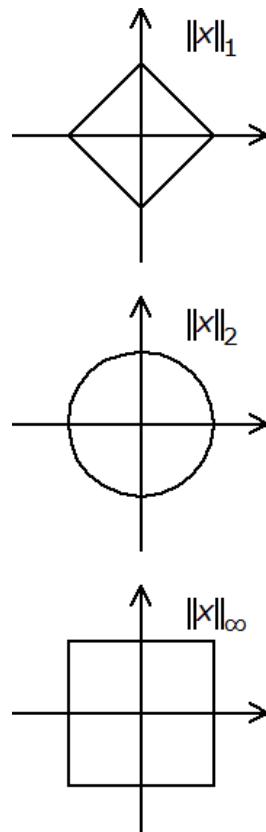
$$\|x\|_p := \left(\sum_{i=0}^n |x_i|^p \right)^{1/p} \quad (2.1.5)$$

Norma nieskończoność

Przykład klasycznej normy, w której funkcja przyporządkowuje wektorowi wartość maksymalnej składowej x_i . Zaletą tej normy jest jest bardzo prosta i szybka implementacja oraz brak wrażliwości na błędy operacji arytmetycznych.

$$\|x\|_\infty := \max_i |x_i|. \quad (2.1.6)$$

Na rysunku 2.2 przedstawiona została interpretacja graficzna wymienionych norm indukowanych w przestrzeni wektorów dwuwymiarowych. Przedstawiono na nim „koła jednostkowe”, stanowiące zbiory punktów znajdujących się na końcach wszystkich wektorów, które dla poszczególnych zdefiniowanych norm uzyskują wartość równą jeden. Wektory są zaczepione w początku układu współrzędnych.



Rys. 2.2. "Koła jednostkowe" dla trzech definicji norm: L_1 dla p -normy i $p = 1$, L_2 dla normy euklidesowej oraz L_∞ dla normy nieskończoność.

[Rysunek dla normy L_1 przedstawia obrócony o 45° kwadrat, którego wierzchołki są na osiach współrzędnych ox i oy , a jego środek ciężkości w początku układu współrzędnych. Środkowy rysunek dla normy L_2 przedstawia okrąg ze środkiem w początku układu współrzędnych. Trzeci, dolny rysunek dla normy L_∞ przedstawia kwadrat ze środkiem ciężkości w początku układu współrzędnych i bokami równoległymi do osi ox i oy .]

2. Numeryczna algebra liniowa

Algebra liniowa jest często osobnym kursem ujętym w programie typowych studiów inżynierskich. Niemniej istnieją pewne specjalne techniki, nieuwzględnione w standardowym kursie, które są związane ze specyfiką rozwiązywania zagadnień algebry liniowej na komputerach. Niniejszy rozdział przedstawia kilka wybranych takich metod.

Typowym zagadnieniem z zakresu algebry liniowej wykorzystywanym w ramach problemów inżynierskich jest rozwiązanie układu równań liniowych. Mamy tutaj na myśli nie układy z trzema czy czterema niewiadomymi, które standardowo są rozwiązywane analitycznie, lecz układy z setkami czy nawet dziesiątkami tysięcy niewiadomych. Tego typu problemy wymagają specjalnych technik, które nie tylko gwarantują znalezienie rozwiązania, ale również znajdują je w sposób minimalizujący nakłady oraz błędy obliczeniowe. Wśród metod rozwiązywania układów równań liniowych wyróżniamy metody bezpośrednie oraz metody iteracyjne. Do metod bezpośrednich zaliczamy takie jak: eliminacja Gaussa, faktoryzacje LU czy QR, faktoryzacje SVD. Do metod iteracyjnych zaliczamy Jacobiego, SOR, Gradientów sprężonych, GMRES, i inne. W niniejszym kursie skupimy się tylko na wybranych metodach bezpośrednich.

Drugim klasycznym zagadnieniem algebry liniowej, który ma duże znaczenie z punktu widzenia inżynierskiego jest wyznaczanie wartości własnych macierzy. Należy zaznaczyć, że wyznaczanie wartości własnych macierzy metodami analitycznymi jest niezmiernie czasochłonne i często sprowadza się do rozwiązywania bardzo źle uwarunkowanego wielomianowego równania charakterystycznego. Wśród metod numerycznych pozwalających wyznaczyć wartości i wektory własne są metody pozwalające wyznaczyć wartości własne rzeczywiste i ujęte, maksymalną wartość własną, minimalną wartość własną lub wszystkie wartości. W niniejszym podręczniku przedstawimy jedynie podstawowe metody wyznaczania wartości własnych minimalnej i maksymalnej. Pozostałe metody wykraczają poza program studiów inżynierskich.

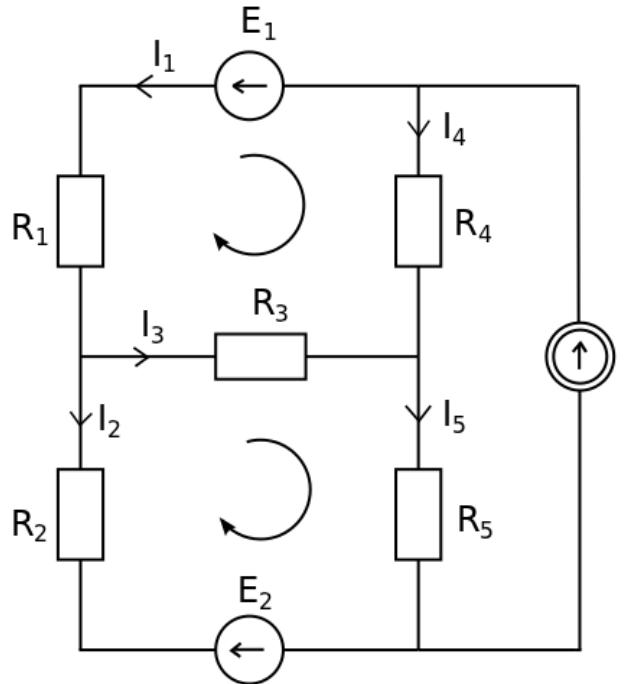
Na początku wprowadźmy podstawowe oznaczenia. Wiele zagadnień naukowych oraz inżynierskich prowadzi do układu równań liniowych $Ax = b$, który w formie macierzowej przyjmuje postać:

$$Ax = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (2.1.1)$$

Przykład 2.1

Przykład

Rozważmy przykład techniczny obwodu elektrycznego zaprezentowanego na rysunku 2.1. Naszym zadaniem jest przedstawić w zapisie macierzowym układ równań pozwalający znaleźć rozkład prądów w obwodzie. Należy wykorzystać prawa Kirchhoffa. Parametry obwodu to: $I = 10[A]$, $E1 = 5[V]$, $E2 = 8[V]$, $R1 = 5[\Omega]$, $R2 = 5[\Omega]$, $R3 = 3[\Omega]$, $R4 = 7[\Omega]$, $R5 = 2[\Omega]$



Rysunek 2.1. Obwód elektryczny zbudowany z trzech oczek, zawierający pięć rezystancji, dwa źródła napięcia E_1 , E_2 oraz jedno źródło prądu I .

Z równań Kirchhoffa otrzymujemy pięć równań. Pierwsze trzy równania przedstawiają bilans prądów w węzłach, a pozostałe dwa bilans napięć w oczkach. Układ pięciu niezależnych liniowo równań zawiera pięć niewiadomych i posiada jednoznaczne rozwiązańe.

$$\begin{aligned}
 I_1 + I_4 &= I \\
 I_3 + I_4 &= I_5 \\
 I_2 + I_3 &= I_1 \\
 R_1 I_1 + R_3 I_3 - R_4 I_4 &= E_1 \\
 R_2 I_2 - R_3 I_3 - R_5 I_5 &= -E_2
 \end{aligned} \tag{2.1.2}$$

Powyższy układ równań zapisany algebraicznie możemy przekształcić do postaci macierzowej:

$$\left[\begin{array}{ccccc} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & -1 \\ -1 & 1 & 1 & 0 & 0 \\ R_1 & 0 & R_3 & -R_4 & 0 \\ 0 & R_2 & -R_3 & 0 & -R_5 \end{array} \right] \left[\begin{array}{c} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{array} \right] = \left[\begin{array}{c} I \\ 0 \\ 0 \\ E_1 \\ -E_2 \end{array} \right] \tag{2.1.3}$$

Ostatecznie po podstawieniu wartości liczbowych otrzymujemy układ równań przedstawiony w równaniu 2.1.3.

$$\left[\begin{array}{ccccc} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & -1 \\ -1 & 1 & 1 & 0 & 0 \\ 5 & 0 & 3 & -7 & 0 \\ 0 & 5 & -3 & 0 & -2 \end{array} \right] \left[\begin{array}{c} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{array} \right] = \left[\begin{array}{c} 10 \\ 0 \\ 0 \\ 5 \\ -8 \end{array} \right] \tag{2.1.3}$$

3. Normy macierzowe

Macierze często są nazywane operatorami liniowymi, które przekształcają wektory, np. za pomocą operacji $x' = Ax$. Przekształceniem może być wydłużenie lub skrócenie wektora. W tym kontekście, normy macierzowe są funkcjami, które przyporządkowują danej macierzy liczbę skalarną wyrażającą zdolność macierzy do wydłużania wektorów. Każda norma wektorowa pozwala nam zdefiniować normę macierzową, która wyraża maksymalne wydłużenie wektora jednostkowego w danej normie po przekształceniu przez macierz A . Takie normy nazywa się normami indukowanymi. Normy indukowane, zatem charakteryzują jak dana macierz A rozciąga / przekształca wektory jednostkowe w odniesieniu do danej normy. Wyraża to równanie 2.1.3, które normie przyporządkowuje maksymalne wydłużenie dowolnego wektora x przez macierz A .

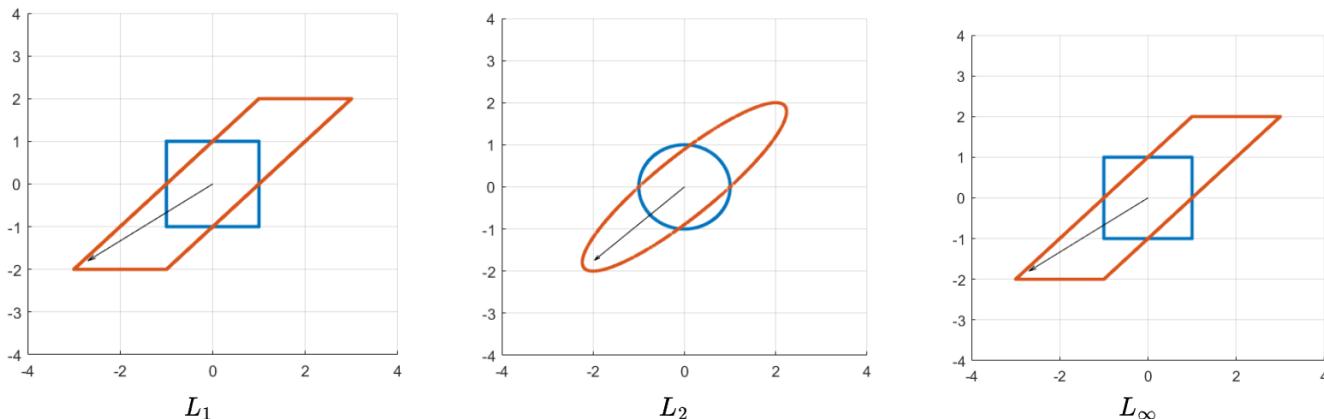
$$\|A\|_p = \max_{\|x\|_p \neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\|x\|_p=1} \|Ax\|_p \quad (2.3.1)$$

Podobnie do norm wektorowych, normy macierzowe spełniają podstawowe kryteria pozwalające na ich wykorzystanie do porównywania macierzy. Dla skalarów α w K oraz wszystkich macierzy A oraz B w przestrzeni $K^{m \times n}$,

- $\|\alpha A\| = |\alpha| \|A\|$ (skalarność)
- $\|A + B\| \leq \|A\| + \|B\|$ (nierówność trójkątna)
- $\|A\| \geq 0$ (nieujemność)
- $\|A\| = 0$ jeżeli $A = 0_{m,n}$ (jednoznaczność)

Na rysunku 2.3 przedstawiono graficzną interpretację przekształcenia zbioru "kół jednostkowych" dla norm L_1 , L_2 , L_∞ przez przykładową macierz:

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix}$$



Rys. 2.3. "Koła jednostkowe" dla trzech definicji norm: L_1 dla p -normy i $p = 1$, L_2 dla normy euklidesowej oraz L_∞ dla normy nieskończoność.

Normy macierzowe są naturalnym rozszerzeniem notacji norm dedykowanej dla wektorów. Na początku wymienimy p -normy indukowane.

Norma L_1 - maksimum

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=0}^n |a_{ij}| \quad (2.3.2)$$

Norma L_∞ - nieskończoność

$$\|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=0}^n |a_{ij}| \quad (2.3.3)$$

Norma L_2

$$\sigma_{\max}(A) = \sqrt{\max \lambda_i} \quad (2.3.4)$$

gdzie σ_{\max} to maksymalna wartość osobienna macierzy A oraz λ_i oznacza wartości własne macierzy $A^H A$ (gdzie A^H to macierz Hermitowska, czyli transponowana macierz sprzężona spełniająca warunek $[a_{ij}] = [a_{ji}]$.

Norma Frobeniusa

$$\|A\|_2 = \sigma_{\max}(A) \leq \left(\sum_{i=0}^n \sum_{j=0}^n |a_{ij}^2| \right)^{1/2} = \|A\|_F \quad (2.3.5)$$

4. Współczynnik uwarunkowania

Powszechnym zastosowaniem norm macierzowych jest obliczanie współczynnika uwarunkowania macierzy $\text{cond}(A)$ informującego o błędzie rozwiązania układu równań $Ax = b$. Współczynnik uwarunkowania określany jest jako stosunek wartości wektora prawych stron b do wartości wektora prawych stron $A^{-1}b$. Założymy, że e jest błędem wektora prawych stron b układu równań $Ax = b \pm e$, wówczas rozwiązanie jest również obarczone tym błędem i wyraża się równaniem:

$$x + \Delta x = A^{-1}b + A^{-1}e$$

zatem stosunek względnego błędu rozwiązania $\frac{\|A^{-1}b\|}{\|A^{-1}e\|}$ do względnego błędu b $\frac{\|e\|}{\|b\|}$ wyraża się:

$$\frac{\frac{\|A^{-1}b\|}{\|e\|}}{\frac{\|b\|}{\|b\|}} = \frac{\|A^{-1}b\| / \|A^{-1}e\|}{\|e\| / \|b\|} = \frac{\|A^{-1}e\|}{\|e\|} \cdot \frac{\|b\|}{\|A^{-1}b\|}$$

Wartość maksymalna tego stosunku może więc postrzegać się jako iloczyn dwóch norm, zgodnie z poniższym przekształceniem:

$$\begin{aligned} \max_{e, b \neq 0} \left\{ \frac{\|A^{-1}e\|}{\|e\|} \cdot \frac{\|b\|}{\|A^{-1}b\|} \right\} &= \max_{e \neq 0} \left\{ \frac{\|A^{-1}e\|}{\|e\|} \right\} \max_{b \neq 0} \left\{ \frac{\|b\|}{\|A^{-1}b\|} \right\} \\ &= \max_{e \neq 0} \left\{ \frac{\|A^{-1}e\|}{\|e\|} \right\} \max_{x \neq 0} \left\{ \frac{\|Ax\|}{\|x\|} \right\} \\ &= \|A^{-1}\| \|A\|. \end{aligned}$$

Wartość maksymalna jest określana właśnie współczynnikiem uwarunkowania:

$$\text{cond}(A) = \|A^{-1}\| \|A\| \geq \|A^{-1}A\| = 1$$

Współczynnik uwarunkowania o wartości 1 nie wprowadza zatem żadnego dodatkowego błędu rozwiązania. Rozwiązanie jest zatem nie gorsze od samych danych wejściowych. Wartość współczynnika uwarunkowania zależy od przyjętych norm. Najczęściej stosowane definicje to:

- współczynnik uzależniony od wartości osobliwych gdy zastosujemy normę L_2

$$\text{cond}(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

- współczynnik uzależniony od wartości minimalnych i maksymalnych gdy macierz jest trójkątna i zastosujemy normę L_∞

$$\text{cond}(A) \geq \frac{\max_i(|a_{ii}|)}{\min_i(|a_{ii}|)}$$

Przykład 2.2

Przykład

Przeanalizujmy wpływ niedokładności danych wejściowych dla przykładowego układu równań wraz z obliczeniem współczynnika uwarunkowania. Rozważamy układ równań $Ax = b$

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 3,999 \end{bmatrix}; b = \begin{bmatrix} 4 \\ 7,999 \end{bmatrix}$$

Współczynnik uwarunkowania tej macierzy możemy obliczyć wyznaczając go z normy L_∞ w następujących krokach:

$$cond(A) = ||A^{-1}|| \cdot ||A||$$

$$||A||_{\infty} = \max_i \sum_j |a_{ij}| = 5,999$$

$$A^{-1} = \frac{1}{\det(A)} A^D = \frac{1}{3.999 - 2 \cdot 2} \begin{bmatrix} 3.999 & -2 \\ -2 & 1 \end{bmatrix} = \begin{bmatrix} -3999 & 2000 \\ 2000 & -1000 \end{bmatrix}$$

$$\| A^{-1} \|_{\infty} = \max_i \sum_j |a_{ij}| = 5999$$

$$cond(A) = ||A^{-1}|| \cdot ||A|| = 5,999 * 5999 = 35988$$

Współczynnik uwarunkowania dla tej macierzy o wymiarach 2×2 jest stosunkowo bardzo duży. Oznacza to, że spodziewamy się znacznego wpływu niedokładności na wynik rozwiązania. Sprawdźmy to.

Obliczmy rozwiązanie układu równań przy założeniu oryginalnego wektora $\begin{bmatrix} 4 \\ 7,999 \end{bmatrix}$. Wówczas rozwiązanie układu równań możemy obliczyć odejmując pierwszy wiersz od drugiego:

następnie

$$x_1 = 4 - 2x_2 = 4 - 2 \cdot 1 = 2$$

otrzymujemy zatem rozwiązanie:

$$x = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Zmieńmy teraz nieznacznie wektor prawych stron b i przyjmijmy: $b = \begin{bmatrix} 4 \\ 8 \end{bmatrix}$. Wówczas analogicznie przeprowadzająca obliczenia jak powyżej otrzymamy wynik:

$$x = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$$

Jak można zauważyć, niewielka zmiana o wartość 0,001 wektora b spowodowała bardzo dużą zmianę rozwiązania. **Mówimy o takim układzie równań, że jest źle uwarunkowany.**

Zmieńmy współczynniki macierzy A :

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}; b = \begin{bmatrix} 4 \\ 7,999 \end{bmatrix}$$

Gdybyśmy powtórzyli obliczenia współczynnika uwarunkowania dla tej macierzy to otrzymalibyśmy wartość $cond(A) = 25$, która jest mniejsza od poprzedniej o ponad 100 razy (dwa rzedy). Widać, że macierz ta jest znacznie lepiej uwarunkowana. Powtórzymy serie obliczeń

dla oryginalnego $b = \begin{bmatrix} 4 \\ 7,999 \end{bmatrix}$ oraz zmodyfikowanego $b = \begin{bmatrix} 4 \\ 8 \end{bmatrix}$ i porównajmy wyniki

dla $b = \begin{bmatrix} 4 \\ 7,999 \end{bmatrix}$ otrzymamy wynik $x = \begin{bmatrix} 3,998 \\ 0,001 \end{bmatrix}$,

a dla $b = \begin{bmatrix} 4 \\ 8 \end{bmatrix}$ otrzymamy $x = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$.

Jak widać, tym razem wyniki rozwiązania różnią się nieznacznie.

5. Metoda eliminacji Gaussa

Eliminacja Gaussa jest podstawowym sposobem rozwiązywania układu równań liniowych. W eliminacji Gaussa wykorzystujemy operacje wierszowe dodawania do siebie wierszy układu równań. Wykonujemy jednak te operacje w ścisłe określony sposób i kolejności. Zasadniczym celem eliminacji Gaussa jest przekształcenie układu równań do postaci górnopróbkowej. To znaczy, że wszystkie współczynniki macierzy $\langle A \rangle$ układu równań poniżej diagonali są zerowane. Uważny czytelnik zauważał zapewne, że mówimy o eliminacji elementów macierzy $\langle A \rangle$, ale operacje wykonujemy na całych wierszach układu równań. Nie możemy zatem zapomnieć o wektorze prawych stron $\langle b \rangle$. W celu uproszczenia implementacji algorytmu zazwyczaj tworzy się zazwyczaj macierz rozszerzoną, która powstaje poprzez dołączenie wektora prawych stron $\langle b \rangle$ dodatkowej kolumny do macierzy $\langle A \rangle$.

$$\langle [A|b] \rangle$$

Następnie przekształcamy macierz $\langle Ag \rangle$ do postaci górnopróbkowej za pomocą dodawania wierszy wierszy 'diagonalnych' wymnożonych przez stosowne współczynniki skalujące (odejmujemy wiersze 'diagonalne' od wierszy poniżej). Kolejność operacji (zaprezentowana na rysunku 2.4) jest następująca:

- najpierw odejmujemy pierwszy wiersz macierzy $\langle Ag \rangle$ od drugiego wiersza wymnożony przez współczynnik, który spowoduje po odjęciu wyzerowanie elementu $\langle a_{21} \rangle$, równy $\langle l_{21} = \frac{a_{21}}{a_{11}} \rangle$,
- w kolejnych krokach odejmuję pierwszy wiersz macierzy wymnożony przez stosowne współczynniki, aż wyzerowane zostaną wszystkie elementy poniżej diagonali,
- dalej, przechodzimy do zerowania elementów poniżej diagonali w drugiej kolumnie,
- itd., aż wyzerujemy wszystkie elementy poniżej diagonali.

Powyższy schemat został zilustrowany na rysunku 2.4.

$$\begin{aligned} -l_{21} &= -\left(\frac{a_{21}}{a_{11}}\right) \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ \cancel{a_{21}} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \\ -l_{31} &= -\left(\frac{a_{31}}{a_{11}}\right) \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ \cancel{a_{31}} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b_3 \end{bmatrix} \\ -l_{32} &= -\left(\frac{a'_{32}}{a'_{22}}\right) \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & \cancel{a'_{32}} & a'_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \end{bmatrix} \rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \end{bmatrix} \end{aligned}$$

Rysunek 2.4. Ilustracja przebiegu eliminacji Gaussa

Przykład 2.3

Przykład

Przeprowadź eliminację Gaussa dla poniższego układu równań z trzema niewiadomymi.

$$\langle x_1+3x_2+4x_3=2 \rangle \langle -2x_1+2x_2+3x_3=-1 \rangle \langle x_1+x_2+2x_3=3 \rangle$$

Powyższy układ równań możemy zapisać w postaci macierzowej:

$$\langle \left(\begin{matrix} 1 & 3 & 4 & 2 \\ -2 & 2 & 3 & -1 \\ 1 & 1 & 2 & 3 \end{matrix} \right) \cdot \left(\begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \right) = \left(\begin{matrix} 2 \\ -1 \\ 3 \end{matrix} \right) \rangle$$

Następnie możemy odjąć pierwszy wiersz od kolejno drugiego i trzeciego:

$$\langle \left(\begin{matrix} 1 & 3 & 4 & 2 \\ -2 & 2 & 3 & -1 \\ 1 & 1 & 2 & 3 \end{matrix} \right) \cdot \left(\begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \right) = \left(\begin{matrix} 2 \\ -1 \\ 3 \end{matrix} \right) \rangle$$

$$\{1\}) 2\backslash end\{matrix\}\right] \backslash)$$

W wyniku otrzymamy:

$$\left(\begin{pmatrix} 1 & 3 & 4 & 0 & 8 & 11 & 0 & -2 & -2 \end{pmatrix} \cdot \begin{pmatrix} x_1 & x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 & 3 & 1 \end{pmatrix} \right)$$

Kolejnym krokiem jest odjęcie drugiego wiersza od trzeciego w celu eliminacji elementu a_{32} :

$$\left(\begin{pmatrix} 1 & 3 & 4 & 0 & 8 & 11 & 0 & -2 & -2 \end{pmatrix} - \frac{-2}{8} \begin{pmatrix} 2 & 3 & 1 \end{pmatrix} \right) = \begin{pmatrix} 2 & 3 & 1 & 0 & 0 & 11 & 0 & -2 & -2 \end{pmatrix}$$

Ostatecznie otrzymuje górną-trójkątną postać układu równań:

$$\left(\begin{pmatrix} 1 & 3 & 4 & 0 & 8 & 11 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 & x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 & 3 & 1 & \frac{1}{8} \end{pmatrix} \right)$$

W celu implementacji komputerowej algorytmu eliminacji Gaussa warto posłużyć się zapisem algorytmicznym procedury. Zakładając, że operujemy na macierzy rozszerzonej $\begin{pmatrix} A & g \end{pmatrix}$ dla $i, j = 1, 2, \dots, n$ o wymiarach $(n \times n+1)$,

$$a_{ij}^k = a_{ij}^{(k-1)} - \frac{a_{ij}^{(k-1)} a_{kk}^{(k-1)}}{a_{jj}^{(k-1)}}$$

dla

$\forall (k=1, 2, \dots, n-1)$ (kolejne kolumny bez ostatniej)

$\forall (i=k+1, k+2, \dots, n)$ (kolejne wiersze poniżej k-tego)

$\forall (j=k, k+1, \dots, n+1)$ (wszystkie kolumny z pominięciem już wcześniej wyzerowanych)

Tak sformułowany algorytm pozwala nam wygodnie zaimplementować funkcje w MATLAB, która przyjmuje jako argument macierz A oraz wektor prawych stron b , a zwraca macierz rozszerzoną $\begin{pmatrix} A & g \end{pmatrix}$, która jest górną-trójkątna. Proszę zwrócić uwagę, na bardzo podobne oznaczenia, które ułatwiają interpretację kodu.

```
function Ag = gaussian(A,b)
Ag = [A b];
n = size(Ag,1);
for k=1:n-1
    for i = k+1:n
        l = Ag(i,k) / Ag(k,k);
        for j = k:n+1
            Ag(i,j) = Ag(i,j) - l * Ag(k,j);
        end
    end
end
end
```

Jednym z bardzo ważnych elementów, których nie możemy pominąć jest wrażliwość algorytmu na wystąpienie w dowolnym momencie procesu zero na diagonali, które będzie skutkowało dzieleniem przez zero podczas obliczania współczynników wykorzystywanego do eliminacji. W celu eliminacji tego problemu stosuje się **selekcję elementu głównego**.

Selekcja elementu głównego polega na takiej zamianie wierszy macierzy rozszerzonej $\begin{pmatrix} A & g \end{pmatrix}$, aby w kolejnym kroku algorytmu na diagonali znalazła się **maksymalny co modułu element**, jednocześnie pozostawiając już wyzerowane elementy nadal zerowymi.

Występują trzy rodzaje selekcji:

1. w kolumnie (selekcja częściowa)- najprostsza wymaga tylko zamiany równań (rysunek 2.5a),
2. w wierszu (selekcja częściowa) - zamieniamy kolejność niewiadomych w wektorze x (rysunek 2.5b),
3. podmacierz $A[r:end, c:end]$ (selekcja pełna) - zamieniamy zarówno wiersze jak i kolejność zmiennych w wektorze x (rysunek 2.5c).

$$\begin{array}{c} \text{a)} \\ \left[\begin{array}{ccccc} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{array} \right] \end{array}$$

$$\begin{array}{c} \text{b)} \\ \left[\begin{array}{ccccc} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{array} \right] \end{array}$$

$$\begin{array}{c} \text{c)} \\ \left[\begin{array}{ccccc} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{array} \right] \end{array}$$

Rys. 2.5. Elementy macierzy, w których poszukuje się wartości maksymalnych co do modułu w przypadku procedury selekcji elementu głównego: a) dla selekcji częściowej w kolumnie, b) dla selekcji częściowej w wierszu, c) dla selekcji pełnej w bloku macierzy.

Przykład 2.4

Przykład

Rozwiążemy układ równań stosując skończoną precyzję sztucznie zaokrąglając wyniki obliczeń arytmetycznych aby pokazać wpływ selekcji elementu głównego również na dokładność rozwiązania.

Na początku rozwiążmy układ z następującym układem wierszy.

$$\begin{aligned}
 & \left(\begin{matrix} 0.003 & 59.1 & 5.291 & -6.13 \end{matrix} \right) \left(\begin{matrix} x_1 & x_2 \end{matrix} \right) = \left(\begin{matrix} 59.17 & 46.78 \end{matrix} \right) \\
 & \frac{5.291}{0.003} = 1763.666... \approx 1764.0 \\
 & \left(\begin{matrix} 0 & -104200 \end{matrix} \right) \left(\begin{matrix} x_1 & x_2 \end{matrix} \right) = \left(\begin{matrix} 59.17 & -104300 \end{matrix} \right) \\
 & x_2 = \frac{-104300}{-104200} \approx 1.001 \quad x_1 = \frac{59.17 - 59.1}{-104200} \approx 3.633
 \end{aligned}$$

Teraz, powtórzmy obliczenia, ale wcześniejszej zamieniając miejscami wiersz pierwszy i drugi układu równań.

$$\begin{aligned}
 & \left(\begin{matrix} 5.291 & -6.13 & 0.003 & 59.1 \end{matrix} \right) \left(\begin{matrix} x_1 & x_2 \end{matrix} \right) = \left(\begin{matrix} 46.78 & 59.17 \end{matrix} \right) \\
 & \frac{0.003}{5.291} = 0.000567 \\
 & \left(\begin{matrix} 46.78 & 59.14 \end{matrix} \right) \left(\begin{matrix} x_1 & x_2 \end{matrix} \right) = \frac{46.78 - (-6.13)}{5.291} \approx 10.001
 \end{aligned}$$

Widzimy, że zmiana wierszy (zaznaczmy, że przy zastosowaniu sztucznie zawyżonego błędu zaokrągleń) dała nam wynik znacznie inny od poprzedniego. Który wynik jest poprawny? Ten drugi, ponieważ na diagonali znajdował się największy co do modułu w danej kolumnie element.

Wykonanie tych obliczeń w MATLABie (nawet z oryginalnym układem wierszy) daje nam wynik drugi. Po pierwsze, MATLAB zamienia wiersze automatycznie, po drugie operacje wykonane są z dużo większą precyzją ($\text{eps} \approx 10^{-14}$).

```

A = [0.003 59.1;
      5.291 -6.13];
b = [59.17
      46.78];
A\b

ans =
    10.0008
    1.0007
  
```

6. Wsteczne podstawienie

Jeżeli macierz rozszerzona reprezentująca nasz układ równań jest już w postaci trójkątnej, to wynik rozwiązania bardzo łatwo znaleźć stosując procedurę wstecznego podstawienia. Przyjrzyjmy się przykładowemu układowi równań w postaci górnopróbkątnej:

$$\left(\begin{matrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{22} & a_{23} & b_2 \\ a_{33} & b_3 \end{matrix} \right)$$

Mając taką postać, procedura kolejno znajduje wartości zaczynając od ostatniego, czyli dla $\forall i=n, n-1, \dots, 1$. Otrzymujemy zatem:

$$\begin{aligned} x_3 &= \frac{b_3 - a_{23}x_2 - a_{13}x_1}{a_{33}} \\ x_2 &= \frac{b_2 - a_{32}x_3 - a_{12}x_1}{a_{22}} \\ x_1 &= \frac{b_1 - a_{31}x_3 - a_{21}x_2 - a_{11}x_3}{a_{11}} \end{aligned}$$

Zauważmy, że w kolejnych wierszach wykorzystujemy wartości x_i obliczone wcześniej. Powyższą procedurę możemy uogólnić dla macierzy górnopróbkątnej o dowolnym rozmiarze i przedstawić matematycznie:

$$\begin{aligned} x_i &= \frac{c_i - \sum_{j=i+1}^n (u_{ij} \cdot x_j)}{u_{ii}} \quad \text{dla } i = n, n-1, \dots, 1 \quad (2.6.1) \\ \text{a w przypadku macierzy dolno-próbkątnej, odwrotnie będziemy kolejno obliczać wartości zaczynając tym razem od pierwszego wiersza:} \\ x_i &= \frac{c_i - \sum_{j=1}^{i-1} (l_{ij} \cdot x_j)}{l_{ii}} \quad \text{dla } i = 1, 2, \dots, n \quad (2.6.2) \end{aligned}$$

Oto dodatkowo dwie funkcje implementujące oba podstawienia.

```
% U - macierz górnopróbkątna, c - wektor prawych stron
function x = wsteczne_gornopróbkątnie(U,c)
    n = size(U,1);
    x = zeros(n,1);
    for i = n:-1:1
        s = 0;
        for j = i+1:n
            s = s + U(i,j)*x(j);
        end
        x(i) = (c(i) - s) / U(i,i);
    end
end

% U - macierz dolno-próbkątna, c - wektor prawych stron
function x = wsteczne_dolno-próbkątnie(L,c)
    n = size(L,1);
    x = zeros(n,1);
    for i = 1:n
        s = 0;
        for j = 1:i-1
            s = s + L(i,j)*x(j);
        end
        x(i) = (c(i) - s) / L(i,i);
    end
end
```

7. Eliminacja Gaussa-Jordana

Eliminacja Gaussa-Jordana jest rozwinięciem eliminacji Gaussa o dodatkowe kroki. W eliminacji tej oprócz zerowania elementów poniżej diagonali układu równań, eliminujemy również elementy powyżej diagonali. Dodatkowo, skalujemy wartości we wszystkich wierszach, tak aby po eliminacji na diagonali były same jedynki. Pamiętając, że operacje, które wykonujemy są operacjami wierszowymi to rozwiązania układu równań oryginalnego i układu po eliminacji są takie same. Łatwo zauważyć, że po eliminacji Gaussa-Jordana, ponieważ w części macierzy A będziemy mieli wyzerowane wszystkie elementy poniżej i powyżej diagonali oraz na diagonali będą wartości , to ostatnia kolumna macierzy rozszerzonej będzie rozwiązaniem układu równań. Nie potrzebujemy zatem wstecznego podstawienia. Pozornie, algorytm wydaje się być szybszy obliczeniowo, ale z uwagi na konieczność wyzerowania elementów powyżej diagonali jego złożoność jest praktycznie taka sama jak eliminacji Gaussa połączonej z wstecznym podstawieniem. Niemniej występują sytuacje, że zastosowanie eliminacji Gaussa-Jordana jest korzystne. Jedną z nich jest np. konieczność obliczenia macierzy odwrotnej. Przebieg algorytmu eliminacji Gaussa-Jordana jest następujący:

- ## 1. Inicjalizacja (definicja macierzy rozszerzonej)

\(a_{ij}^{(0)} = a_{ij} \) dla \(i=1,2,\dots,n; j=1,2,\dots,n \)

\(a_{ij}^{(0)} = b_i \) dla \(i=1,2,\dots,n; j=n+1 \)

2. Normalizacja (znormalizować element diagonalny do wartości 1)

$$\backslash(a_{kj})^{(0)} = \frac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}} \quad \text{dla } j=k, k+1, \dots, n+1$$

3. Redukcja (zredukować wszystkie elementy pozadiagonalne w kolumnie k)

```
\backslash( a_{ij})^{\wedge \{(0)\}} = a_{ij})^{\wedge \{(k-1)\}} - a_{ik})^{\wedge \{(k-1)\}} \cdot \ldots \cdot a_{kj})^{\wedge \{(k)\}} \text{for } j=k, k+1, \dots, n+1; i=1, 2, \dots, n \text{(and } i \neq k \text{) } \backslash
```

Kroki 2-3 muszą zostać wykonane dla wszystkich kolumn $\backslash(k=1,2,\dots,n\backslash)$.

Przykład 2.5

Przykład

Znajdź rozwiązanie układu równań przedstawionego w postaci macierzy rozszerzonej.

- W pierwszym kroku podzielimy wszystkie elementy pierwszego wiersza przez wartość elementu diagonalnego $\backslash(2\backslash)$.
 - W drugim kroku odejmiemy pierwszy wiersz od drugiego wymnożony przez $\backslash(4\backslash)$.
 - W trzecim kroku odejmiemy pierwszy wiersz od trzeciego wymnożony przez $\backslash(3\backslash)$.
 - W czwartym kroku podzielimy elementy w drugim wierszu przez wartość diagonalną równą $\backslash(5\backslash)$.
 - W piątym kroku odejmiemy drugi wiersz od pierwszego wymnożony przez $\backslash(\frac{-1}{2}\backslash)$.
 - W szóstym kroku odejmiemy drugi wiersz od trzeciego wymnożony przez $\backslash(\frac{7}{2}\backslash)$.
 - ... kontynuując obliczenia otrzymamy ostateczny wynik w czwartej kolumnie macierzy rozszerzonej: $\backslash(x=[1,2,4]\backslash)$.
$$\backslash(\left[\begin{matrix}2&-1&1&4\\4&3&-1&6\\3&2&2&15\end{matrix}\right]\rightarrow\left[\begin{matrix}1&\frac{-1}{2}&\frac{1}{2}&\frac{1}{2}\end{matrix}\right]\rightarrow\left[\begin{matrix}1&\frac{-1}{2}&\frac{1}{2}&\frac{1}{2}\\0&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}\end{matrix}\right]\rightarrow\left[\begin{matrix}1&\frac{-1}{2}&\frac{1}{2}&\frac{1}{2}\\0&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}\\0&\frac{7}{2}&\frac{1}{2}&\frac{1}{2}\end{matrix}\right]\rightarrow\left[\begin{matrix}1&\frac{-1}{2}&\frac{1}{2}&\frac{1}{2}\\0&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}\\0&0&\frac{1}{2}&\frac{1}{2}\end{matrix}\right]\rightarrow\left[\begin{matrix}1&\frac{-1}{2}&\frac{1}{2}&\frac{1}{2}\\0&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}\\0&0&1&\frac{1}{2}\end{matrix}\right]\rightarrow\left[\begin{matrix}1&\frac{-1}{2}&\frac{1}{2}&\frac{1}{2}\\0&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}\\0&0&0&\frac{1}{2}\end{matrix}\right]\rightarrow\left[\begin{matrix}1&\frac{-1}{2}&\frac{1}{2}&\frac{1}{2}\\0&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}\\0&0&0&1\end{matrix}\right]$$

Jak wcześniej wspominano, eliminację Gaussa-Jordana można skutecznie wykorzystać do obliczania macierzy odwrotnej o niewielkim rozmiarze. Aby wyprowadzić ten sposób przypomnijmy definicję macierzy odwrotnej:

$$\begin{aligned} & \left(AA^{-1} = \left[\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{matrix} \right] \cdot \left[\begin{matrix} a'_{11} & a'_{12} & \dots & a'_{1n} \\ a'_{21} & a'_{22} & \dots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a'_{m1} & a'_{m2} & \dots & a'_{mn} \end{matrix} \right] \right) = \left[\begin{matrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{matrix} \right] \end{aligned} \quad (2.7.1)$$

Zauważmy, że w tym równaniu pierwszą i kolejne kolumny macierzy (A^{-1}) możemy oznaczyć jako zmienne (x_{ij})

$$\begin{aligned} & \left(\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{matrix} \right) \cdot \left(\begin{matrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{matrix} \right) = \left(\begin{matrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{matrix} \right) \end{aligned}$$

Przy takim zapisie, okazuje się, że aby znaleźć kolejne kolumny macierzy odwrotnej (A^{-1}) wystarczy rozwiązać (n) niezależnych układów równań, w których niewiadomymi będą kolumny macierzy (A^{-1}) a wektorami prawych stron kolejne kolumny macierzy jednostkowej.

$$\begin{aligned} & \left(\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{matrix} \right) \cdot \left(\begin{matrix} x_{11} \\ x_{21} \\ \vdots \\ x_{m1} \end{matrix} \right) = \left(\begin{matrix} 1 \\ 0 \\ \vdots \\ 0 \end{matrix} \right) \\ & \left(\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{matrix} \right) \cdot \left(\begin{matrix} x_{12} \\ x_{22} \\ \vdots \\ x_{m2} \end{matrix} \right) = \left(\begin{matrix} 0 \\ 1 \\ \vdots \\ 0 \end{matrix} \right) \end{aligned}$$

i tak dalej.

Możemy jednak podejść do tego zagadnienia skuteczniej i zamiast rozwiązywać osobnych układów równań rozwiążemy jeden ale z wszystkimi wektorami prawych stron (całą macierzą jednostkową) doklejoną do macierzy (A) .

$$\left(\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & 0 & 0 & \dots & 1 \end{matrix} \right)$$

Wykorzystamy do tego metodę eliminacji Gaussa-Jordana, w wyniku której przekształcimy układ do postaci

$$\left(\begin{matrix} 1 & 0 & \dots & 0 & a'_{11} & a'_{12} & \dots & a'_{1n} & 0 & 1 & \dots & 0 \\ 0 & 1 & \dots & 0 & a'_{21} & a'_{22} & \dots & a'_{2n} & 0 & 0 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & a'_{m1} & a'_{m2} & \dots & a'_{mn} & 0 & 0 & \dots & 1 \end{matrix} \right)$$

gdzie współczynniki (a'_{ij}) stanowią współczynniki szukanej macierzy odwrotnej.

Przykład 2.6

Przykład

Wykorzystując eliminację Gaussa-Jordana znajdź macierz odwrotną do (A) :

$$(A = \left[\begin{matrix} 2 & 1 & 4 & 5 \end{matrix} \right])$$

Rozwiązanie

$$\begin{aligned} & \left(\begin{matrix} 2 & 1 & 4 & 5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 2 & 2.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \\ & \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \\ & \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \end{aligned}$$

Odpowiedź

$$(A^{-1} = \left[\begin{matrix} \frac{1}{6} & -\frac{1}{6} & \frac{1}{3} & -\frac{1}{3} \end{matrix} \right])$$

8. Rozkład na czynniki - faktoryzacja macierzy

Innym sposobem rozwiązywania układów równań jest wstępny rozkład na czynniki macierzy (A) . W algebrze liniowej stosuje się powszechnie kilka rodzajów faktoryzacji:

1. Faktoryzacja LU: $(A=L \cdot U)$, macierz L jest macierzą dolno-trójkątną, a macierz U górnou-trójkątną,
2. Faktoryzacja QR: $(A=Q \cdot R)$, macierz Q jest macierzą ortonormalną $(Q^T Q = 1 \rightarrow Q^{-1} = Q^T)$, macierz jest macierzą górnou-trójkątną,
3. Faktoryzacja SVD: $(A=S \cdot V)$, macierze (S) , (V) są ortogonalne, a (D) diagonalna.

W ramach niniejszego podręcznika przyjrzymy się wyłącznie faktoryzacji (LU) . Zakładając w właściwości górnou i dolno-trójkątne macierzy (L) i (U) oryginalny układ równań możemy zapisać następująco:

$$(Ax=b)$$

$$(LUx=b)$$

Rozwiązanie możemy uzyskać dwukrotnie stosując wsteczne podstawienie. Najpierw podstawimy $(Ux=y)$, wówczas uzyskamy

$$(Ly=b)$$

Macierz (L) jest dolno-trójkątna więc rozwiązywanie tego równani moźemy szybko znaleźć stosując wsteczne podstawienie od góry. Znając wynik (y) , moźemy przejść do drugiego etapu, korzystając z wprowadzonego wcześniej podstawienia:

$$(Ux=y)$$

Wykorzystując ponownie wsteczne podstawienie uzyskamy ostateczne rozwiązanie.

8.1. Metoda Doolittle'a

Pozostaje pytanie jak skutecznie znaleźć rozkład LU. Pierwszym podejściem jest zastosowanie **metody Doolittle'a**. Aby wyprowadzić tą metodę posłużmy się pełnym zapisem faktoryzacji LU:

$$\backslash(A=LU)$$

$$\backslash(\left(\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{matrix} \right) = \left(\begin{matrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \dots & \dots & \dots & 0 \\ l_{n1} & l_{n2} & \dots & 1 \end{matrix} \right) \left(\begin{matrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & u_{nn} \end{matrix} \right))$$

Zapis ten pozwoli nam wyprowadzić procedurę Doolittle'a bezpośrednio z definicji faktoryzacji. W metodzie tej kluczowa jest kolejność w jakiej będziemy wyznaczać współczynniki kolejno parami poszczególnych wierszy macierzy $\backslash(U)$ i kolumn macierzy $\backslash(L)$.

- Najpierw wyznaczamy współczynniki pierwszego wiersza macierzy $\backslash(U)$. Zgodnie ze schematem na rysunku 2.6 wartości współczynników macierzy $\backslash(A)$ w pierwszym są równe:

$$\backslash(\begin{aligned} a_{11} &= 1 u_{11} + 0 \cdot u_{12} + \dots + 0 \cdot u_{1n} \\ a_{12} &= 1 u_{21} + 0 \cdot u_{22} + \dots + 0 \cdot u_{2n} \\ \vdots &= \vdots u_{n1} + 0 \cdot u_{n2} + \dots + 0 \cdot u_{nn} \end{aligned})$$

Stąd błyskawicznie (z uwagi na większość zer), możemy wyznaczyć $\backslash(u_{1i}=a_{1i})$ dla $\backslash(i=1,2,\dots,n)$.

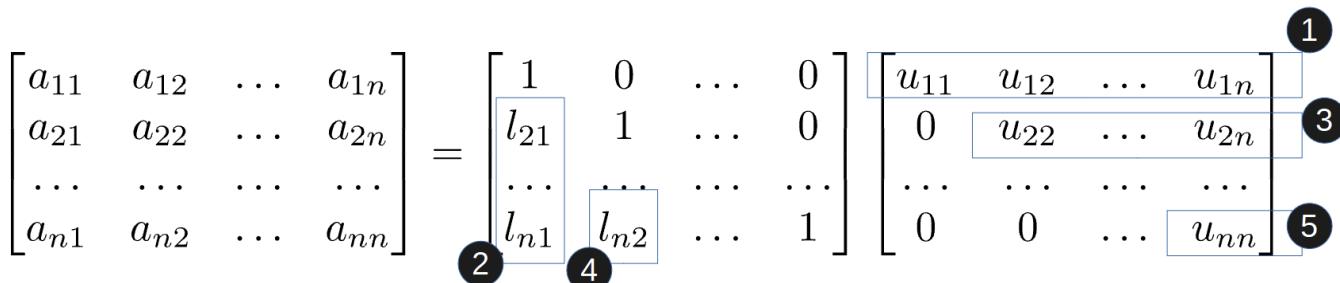
W kolejnym drugim kroku wyznaczamy współczynniki pierwszej kolumny macierzy $\backslash(L)$:

$$\backslash(\begin{aligned} a_{21} &= l_{21} u_{11} + 1 \cdot u_{22} + \dots + 0 \cdot u_{2n} \\ a_{31} &= l_{31} u_{11} + l_{32} u_{21} + 1 \cdot u_{32} + \dots + 0 \cdot u_{3n} \\ \vdots &= \vdots l_{n1} u_{11} + l_{n2} u_{21} + \dots + l_{nn} u_{nn} \end{aligned})$$

stąd łatwo wyprowadzić wyrażenia na wartości $\backslash(l_{i1})$ zakładając, że wartości $\backslash(u_{11})$ została już obliczona w poprzednim kroku. W trzecim kroku algorytmu wróćmy do macierzy $\backslash(U)$, tym razem do drugiego wiersza, analogicznie wyprowadzając wzory na współczynniki z operacji mnożenia wiersza razy kolumna:

$$\backslash(\begin{aligned} a_{22} &= l_{21} u_{12} + 1 u_{22} + \dots + 0 \cdot u_{2n} \\ a_{32} &= l_{31} u_{12} + l_{32} u_{21} + 1 u_{32} + \dots + 0 \cdot u_{3n} \\ \vdots &= \vdots l_{n1} u_{12} + l_{n2} u_{21} + \dots + l_{nn} u_{nn} \end{aligned})$$

i tak dalej.



Rys. 2.6. Schemat z kolejnością obliczeń wierszy i kolumn macierzy $\backslash(L)$ i $\backslash(U)$ w faktoryzacji $\backslash(LU)$.

Wykorzystując ten schemat możemy określić algorytm faktoryzacji LU metodą Doolittle'a w następujący sposób:

- pierwszy wiersz U kopujemy z pierwszego wiersza A
 $\backslash(u_{1i}=a_{1i})$ dla $\backslash(i=1,2,\dots,n)$
- pierwsza kolumna L obliczana jest za pomocą:
 $\backslash(l_{i1}=a_{i1}/u_{11})$ dla $\backslash(i=1,2,\dots,n)$ dla $\backslash(i=2,3,\dots,n)$
- następnie dla każdej pary $\backslash(i^t y)$ wiersz $\backslash(U)$ oraz $\backslash(i^t a)$ kolumna $\backslash(L)$:

$$\backslash(\begin{aligned} \text{dla } i=2,3, \dots, n, \text{ dla } j=1, \dots, i-1, \text{ dla } k=i+1, \dots, n \\ l_{ik} = a_{ik} - \sum_{j=1}^{i-1} l_{ij} u_{jk} \end{aligned})$$

Implementacja tego kodu w postaci funkcji MATLABa jest następująca.

```
function [L, U] = doolittle(A)
n = size(A,1);
L = eye( n ); % inicjujemy macierz jednsotkowa, poniewaz zawsze na diagonali sa jedynki
U = zeros( n ); % pusta (na razie) macierz gornotrojkatna
U(1,: ) = A(1,: ) % kopujemy pierwszy wiersz
L(2:n,1) = A(2: n ,1) / U(1,1); % obliczamy pierwsza kolumnę

% wykonujemy parami obliczenia kolejno wierszy U i kolumn L
for i = 2:n
    for k = i:n
        s = 0;
        for j=1:i-1
            s = s + L(i,j)*U(j,k);
        end
        U(i,k) = A(i,k) - s;
    end
    for k = i+1:n
        s = 0;
        for j=1:i-1
            s = s + L(k,j)*U(j,i);
        end
        L(k,i) = (A(k,i) - s) / U(i,i);
    end
end
end
```

Przykład 2.7

Przykład

Wykorzystując przykładową, powyższą funkcję MATLABa oraz wcześniej zdefiniowane funkcje do wstecznego podstawienia znajdź rozwiązańe układu równań:

$$\begin{pmatrix} 1 & -1 & 1 & 1 \\ 4 & 3 & -1 & 2 \\ 3 & 2 & 2 & 5 \\ 8 & 9 & 5 & 8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \\ 15 \\ 1 \end{pmatrix}$$

Rozwiązanie:

```
function L04_lu
A = [1 -1 1 1
      4 3 -1 2
      3 2 2 5
      8 9 5 8];
b = [4 6 15 1]';
[L, U] = doolittle(A)
A = L*U % powinna być macierz zerowa
y = wsteczne_dolnotrojkatne(L,b)
x = wsteczne_gornotrojkatne(U, y)

% sprawdzam rozwiazanie - norma powinna być zero
norm(A*x-b)
end

L =
1.0000      0      0      0
4.0000    1.0000      0      0
3.0000    0.7143    1.0000      0
8.0000    2.4286    3.5556    1.0000

U =
1.0000   -1.0000    1.0000    1.0000
0     7.0000   -5.0000   -2.0000
0       0     2.5714    3.4286
0       0       0    -7.3333

ans =
0   0   0   0
0   0   0   0
0   0   0   0
0   0   0   0

y =
4.0000
-10.0000
10.1429
-42.7778

x =
-0.5000
-2.5000
-3.8333
5.8333

ans =
1.9860e-15
```

8.2. Metoda eliminacji Gaussa

Drugim, najbardziej użytecznym z praktycznego punktu widzenia sposobem jest wykorzystanie eliminacji Gaussa. Znaczenie tego podejścia jest bardzo istotne, gdyż pozwala na selekcję elementu głównego w kolejnych krokach metody. Metoda Doolittle'a nie pozwala na to. Dzięki selekcji jesteśmy zabezpieczeni przed dzieleniem przez zero na diagonali oraz redukujemy błędy zaokrągleń.

Algorytm faktoryzacji LU z wykorzystaniem eliminacji Gaussa jest bardzo prosty do implementacji. Pomijając szczegóły związane z wyprowadzeniem (wyrazilibyśmy operacje wierszowe za pomocą operatorów macierzowych oraz metodą Gaussa-Jordana wyprowadziliśmy macierze odwrotne tych operatorów) przedstawmy algorytm.

Faktoryzacja (LU) metodą eliminacji Gaussa przebiega zgodnie z procesem eliminacji, ale w trakcie procesu współczynniki (l_{ij}) , które używaliśmy do wymnożenia macierzy diagonalnych podczas zerowania elementów zapamiętujemy w odpowiednich miejscach (ij) macierzy (L) . Macierz wynikowa eliminacji Gaussa staje się wynikową macierzą (U) .

Zatem, współczynnik (l_{21}) z rysunku 2.7 wstawiamy w miejsce $(2,1)$ macierzy docelowej (L) (patrz rysunek 2.8).

$$-l_{21} = -\left(\frac{a_{21}}{a_{11}}\right) \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ \cancel{a_{21}} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Rys. 2.7. Ilustracja operacji odejmowania pierwszego wiersza macierzy w trakcie eliminacji Gaussa od wiersza drugiego, z zaznaczonym współczynnikiem (l_{21}) , który wykorzystywany jest do wstawienia do macierzy (L) .

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix}$$

Rys. 2.8. Ilustracja z zaznaczonym elementem (l_{21}) macierzy (L) .

Implementacja faktoryzacji (LU) z wykorzystaniem eliminacji Gaussa w środowisku MATLAB (bez selekcji elementu głównego) została przedstawiona poniżej.

```
function [L, U] = lu_gaussian( A )

n = size( A, 1 );
L = eye( n );
for j = 1:n-1
    for i = j+1:n
        f = A( i,j ) / A( j,j );
        A( i, : ) = A( i, : ) - f*A( j,: );
        L( i, j ) = f; % tutaj zapamietujemy współczynnik
    end
end
U = A;
end
```

Przykład 2.8

Przykład

Wykorzystaj implementację faktoryzacji LU na przykładowej macierzy losowej o rozmiarach (4×4) . Sprawdź wynik faktoryzacji

obliczając normę $\|LU-A\|$

Rozwiążanie

```
function l05_lu_gaussian
    A = rand( 4 )
    [L, U] = lu_gaussian( A )
    norm( L * U - A, 2 )
end

>> l05_lu_gaussian
A =
    0.8147    0.6324    0.9575    0.9572
    0.9058    0.0975    0.9649    0.4854
    0.1270    0.2785    0.1576    0.8003
    0.9134    0.5469    0.9706    0.1419

L =
    1.0000      0      0      0
    1.1118    1.0000      0      0
    0.1559   -0.2972    1.0000      0
    1.1211    0.2676    3.5869    1.0000

U =
    0.8147    0.6324    0.9575    0.9572
    0   -0.6055   -0.0996   -0.5788
    0      0   -0.0212    0.4791
    0      0      0   -2.4948

ans =
    2.2204e-16
```

normy

3 Rozdział

Strona: [LeIA](#)

Kurs: Metody numeryczne (2025Z)

Książka: 3 Rozdział

Wydrukowane przez użytkownika: Kinga Kondraciuk

Data: niedziela, 30 listopada 2025, 14:01

Spis treści

- 1. Funkcje interpolacyjne**
- 2. Wielomiany interpolacyjne**
- 3. Wielomian Lagrange'a**
- 4. Wielomian Newtona**

1. Funkcje interpolacyjne

Funkcje interpolacyjne

W lekcji tej wprowadzimy pojęcie funkcji interpolacyjnej, wielomianu interpolacyjnego i węzłów interpolacji. Podamy również podstawowe twierdzenie o istnieniu i jednoznaczności wielomianu interpolacyjnego. Zastosujemy wielomiany interpolacyjne Lagrange'a i Newtona do rozwiązywania zadań interpolacyjnych.

Z doświadczeń lub pomiarów określiliśmy w $n + 1$ różnych punktach: x_0, x_1, \dots, x_n z przedziału $\langle a, b \rangle$ wartości funkcji $y = f(x)$ i te wartości oznaczyliśmy przez:

$$y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n) \quad (3.1.1)$$

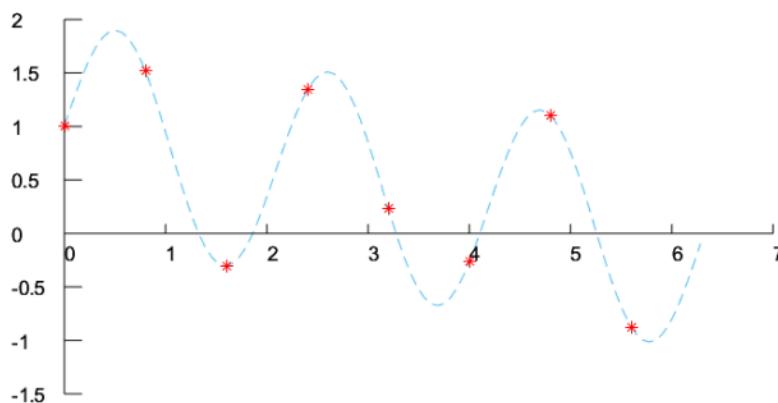
Interpolacja służy do znajdowania przybliżonych wartości funkcji $f(x)$ w dowolnym punkcie przedziału nawet w przypadku, gdy znane jest tylko kilka wartości funkcji $f(x)$ w tym przedziale.

Zadaniem interpolacji jest wyznaczenie funkcji $F(x)$, zwanej **funkcją interpolacyjną**, określonej w przedziale $\langle a, b \rangle$, która w punktach x_0, x_1, \dots, x_n , zwany**m węzłami interpolacji**, przyjmuje wartości funkcji $f(x)$ i w punktach poza węzłami przybliża wartość tej funkcji.

Zatem dla punktów x_i dla $i = 0, 1, 2, \dots, n$ w przedziale $\langle a, b \rangle$ funkcja $F(x)$ musi spełniać $n + 1$ warunków:

$$F(x_i) = y_i = f(x_i) \quad i = 0, 1, 2, \dots, n \quad (3.1.2)$$

Wykres funkcji interpolacyjnej musi przechodzić przez punkty (x_i, y_i) $i = 0, 1, 2, \dots, n$ zaznaczone czerwonymi krzyżkami. Funkcja interpolacyjna jest narysowana niebieską przerywaną linią.



Rys 3.1. - Funkcja interpolacyjna.

Jako funkcje interpolacyjne stosuje się bardzo często:

- wielomiany algebraiczne stopnia n , oznaczmy je przez $W_n(x)$,
- funkcje sklejane $S(x)$.

Wielomiany algebraiczne stopnia n będziemy zapisywać w znanej postaci:

$$W_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} + a_nx^n \quad (3.1.3)$$

gdzie a_i , $i = 0, 1, 2, \dots, n$ o współczynnikach rzeczywistych wielomianu.

W dalszych rozważaniach właśnie tym wielomianom poświęcimy najczęściej miejsca i będziemy je wykorzystywać jako funkcje interpolacyjne. Funkcje sklejane tzw. "splajny" będziemy omawiać w dalszych lekcjach.

2. Wielomiany interpolacyjne

Wielomiany interpolacyjne

Dla funkcji $y = f(x)$, która w $n + 1$ różnych punktach: x_0, x_1, \dots, x_n z przedziału $\langle a, b \rangle$ przyjmuje wartości $y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n)$ zbudujemy wielomian interpolacyjny algebraiczny w postaci:

$$W_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} + a_nx^n$$

Ponieważ wielomian n-tego stopnia ma $n + 1$ niewiadomych współczynników a_i $i = 0, 1, 2, \dots, n$, aby go jednoznacznie określić trzeba wyznaczyć $n + 1$ równań, w których te współczynniki są niewiadomymi. Inaczej mówiąc trzeba podać $n + 1$ punktów, przez które ma przechodzić wykres tego wielomianu. I tak wielomian pierwszego stopnia $W_1(x) = a_0 + a_1x$ graficznie przedstawia prostą, ma dwa współczynniki a_0, a_1 i wiadomo, że przez dwa punkty przechodzi jedna prosta.

Wielomian drugiego stopnia $W_2(x) = a_0 + a_1x + a_2x^2$ ma trzy współczynniki a_0, a_1, a_2 i wymaga trzech punktów, aby określić te współczynniki jednoznacznie, graficznie taki wielomian przedstawia parabolę (przez trzy punkty przechodzi jedna parabola). Ogólnie zatem prawdziwe jest twierdzenie:

Twierdzenie

Twierdzenie o istnieniu i jednoznaczności wielomianu interpolacyjnego:

Istnieje jedyny wielomian interpolacyjny $W_n(x)$ stopnia co najwyżej n , który w $n + 1$ różnych punktach x_0, x_1, \dots, x_n z przedziału $\langle a, b \rangle$ pokrywa się z funkcją $y = f(x)$, tzn.:

$$W(x_i) = y_i = f(x_i) \quad i = 0, 1, 2, \dots, n$$

Dowód: Dla wielomianu $W_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} + a_nx^n$ warunki $W(x_i) = y_i = f(x_i)$ $i = 0, 1, 2, \dots, n$ sprowadzają się do rozwiązania układu $n + 1$ następujących równań liniowych z $n + 1$ niewiadomymi a_i $i = 0, 1, 2, \dots, n$:

$$a_0 + a_1x_0 + a_2x_0^2 + \dots + a_nx_0^n = y_0$$

$$a_0 + a_1x_1 + a_2x_1^2 + \dots + a_nx_1^n = y_1$$

$$\vdots$$

$$a_0 + a_1x_n + a_2x_n^2 + \dots + a_nx_n^n = y_n$$

Przypominamy, że układ równań liniowych $(n + 1) \times (n + 1)$ ma jednoznaczne rozwiązania na niewiadome a_i $i = 0, 1, 2, \dots, n$, jeśli wyznacznik tego układu jest różny od zera. Jak wygląda wyznacznik naszego układu?

$$V = \begin{vmatrix} 1 & x_0 & x_0^2 & \dots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{vmatrix} \quad (3.2.1)$$

Jest to wyznacznik Vandermonde'a i jest on równy iloczynowi wszystkich możliwych różnic $x_j - x_i$ gdzie $j > i$. Zapisujemy ten fakt w postaci

$$\det A = \prod_{1 \leq i < j \leq n} (x_j - x_i)$$

Ponieważ węzły x_i $i = 0, 1, 2, \dots, n$ są różne to wyrazy $x_j - x_i$ są różne od zera, zatem wyznacznik jest różny od zera i układ ma zawsze jedynie rozwiązanie. Oznacza to, że istnieje zawsze jedyny szukany wielomian interpolacyjny. Może być stopnia niższego niż n , bo współczynnik a_n może być równy zero, oczywiście jeszcze jakieś inne współczynniki mogą się zerować.

Przykład 3.1

Przykład

Aby wyjaśnić obliczanie wyznacznika Vandermonde'a obliczymy ten wyznacznik dla stopnia 2 i 3, dalsze obliczanie wynika z indukcji matematycznej - nie będziemy jej prowadzić, aby nie komplikować rozważań.

Dla $n = 2$ mamy

$$V_2 = \begin{vmatrix} 1 & x_0 \\ 1 & x_1 \end{vmatrix} = x_1 - x_0 \neq 0 \text{ bo węzły } x_0 \neq x_1$$

Dla $n = 3$ wyznacznik obliczamy następująco:

$$V_3 = \begin{vmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{vmatrix} = \begin{vmatrix} 1 & x_0 & x_0^2 \\ 0 & x_1 - x_0 & x_1^2 - x_0^2 \\ 0 & x_2 - x_0 & x_2^2 - x_0^2 \end{vmatrix} = \begin{vmatrix} x_1 - x_0 & x_1^2 - x_0^2 \\ x_2 - x_0 & x_2^2 - x_0^2 \end{vmatrix} = (x_1 - x_0)(x_2 - x_0) \begin{vmatrix} 1 & x_1 + x_0 \\ 1 & x_2 + x_0 \end{vmatrix} = (x_1 - x_0)(x_2 - x_0)(x_2 - x_1)$$

Jak powstały te przekształcenia? W pierwszym kroku od drugiego wiersza i od trzeciego wiersza został odjęty wiersz pierwszy i wyzerowały się wyrazy w pierwszej kolumnie w drugim i trzecim wierszy. Dalej następuje rozwinięcie wyznacznika względem pierwszej kolumny i zostaje jeden wyznacznik drugiego stopnia. Wyciągamy z pierwszego wiersza wyrażenie $x_1 - x_0$, a z drugiego wiersza $x_2 - x_0$ przed wyznacznik korzystając ze wzoru skróconego mnożenia $a^2 - b^2 = (a - b)(a + b)$. I znów ponieważ węzły są różne otrzymujemy wyrażenie na wyznacznik, które jest różne od zera.

Potem prowadzimy indukcję względem stopnia wyznacznika.

3. Wielomian Lagrange'a

Wielomian Lagrange'a

Poszukujemy wielomianu algebraicznego w postaci:

$$WL_n(x) = \frac{(x-x_1)(x-x_2) \cdots (x-x_n)}{(x_0-x_1)(x_0-x_2) \cdots (x_0-x_n)} y_0 + \frac{(x-x_0)(x-x_2) \cdots (x-x_n)}{(x_1-x_0)(x_1-x_2) \cdots (x_1-x_n)} y_1 + \cdots + \frac{(x-x_0) \cdots (x-x_{k-1})(x-x_{k+1}) \cdots (x-x_n)}{(x_k-x_0) \cdots (x_k-x_{k-1})(x_k-x_{k+1}) \cdots (x_k-x_n)} y_k + \cdots + \frac{(x-x_0)(x-x_1) \cdots (x-x_{n-1})}{(x_n-x_0)(x_n-x_1) \cdots (x_n-x_{n-1})} y_n$$

Wielomian ten nosi nazwę **wielomianu interpolacyjnego Lagrange'a**.

Wielomian ten ma $n+1$ składników, w każdym ze składników przy y_k $k = 0, 1, 2, \dots, n$ w liczniku nie występuje czynnik $(x - x_k)$ to znaczy w liczniku jest wielomian n -tego stopnia, a w mianowniku od węzła x_k odejmowane są wszystkie inne węzły i te różnice są pomnożone przez siebie. Zwróćmy uwagę, że w mianowniku nie występuje również czynnik $(x_k - x_k)$. Ponieważ suma wielomianów stopnia n jest wielomianem co najwyżej n -tego stopnia (jakie wyraże mogą się zredukować i możemy dostać wielomian niższego stopnia, ale nigdy wyższego) wielomian Lagrange'a $WL_n(x)$ jest wielomianem co najwyżej n -tego stopnia.

Sprawdzamy, czy jest spełniony warunek $WL_n(x_k) = y_k = f(x_k)$ $k = 0, 1, 2, \dots, n$.

Wstawiając za x do wielomianu $WL_n(x)$ węzeł x_k otrzymujemy:

$$\begin{aligned} WL_n(x) &= \frac{(x-x_1)(x-x_2) \cdots (x-x_n)}{(x_0-x_1)(x_0-x_2) \cdots (x_0-x_n)} y_0 \\ &+ \frac{(x-x_0)(x-x_2) \cdots (x-x_n)}{(x_1-x_0)(x_1-x_2) \cdots (x_1-x_n)} y_1 + \cdots \\ &+ \frac{(x-x_0) \cdots (x-x_{k-1})(x-x_{k+1})}{(x_k-x_0) \cdots (x_k-x_{k-1}) \cdots (x_k-x_n)} y_k \\ &+ \cdots + \frac{(x-x_0)(x-x_1) \cdots (x-x_{n-1})}{(x_n-x_0)(x_n-x_1) \cdots (x_n-x_{n-1})} y_n \end{aligned}$$

W każdym składniku oprócz tego, który stoi przy y_k jest w liczniku różnica $(x_k - x_k)$ czyli zero, natomiast w tym składniku przy y_k w liczniku jest takie samo wyrażenie jak w mianowniku, to znaczy, że przy jest współczynnik 1. Zatem

$$WL_n(x_k) = 0 \cdot y_0 + 0 \cdot y_1 + \cdots + 1 \cdot y_k + \cdots + 0 \cdot y_n = y_k$$

Wielomian Lagrange'a spełnia wymagania z twierdzenia o istnieniu, zatem jest szukanym wielomianem interpolacyjnym. Jeśli wprowadzimy następujące oznaczenie:

$$\Phi_k(x) = \frac{(x-x_0) \cdots (x-x_{k-1})(x-x_{k+1}) \cdots (x-x_n)}{(x_k-x_0) \cdots (x_k-x_{k-1})(x_k-x_{k+1}) \cdots (x_k-x_n)} \quad (3.3.2)$$

to wielomian będzie mieć postać:

$$WL_n(x) = \sum_{k=0}^n \Phi_k(x) y_k \quad (3.3.3)$$

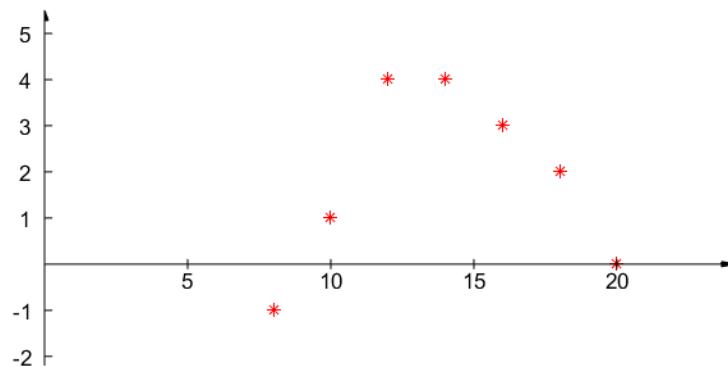
Przykład 3.2.

Przykład

Zmierzliśmy w pierwszym dniu wiosny - 21 marca 2005 roku- w Warszawie na Mokotowie temperaturę za oknem i wyniki zapisaliśmy w tabeli:

Godzina pomiaru	8	10	12	14	16	18	20
Temperatura w stopniach	-1	1	4	4	3	2	0

Na wykresie wygląda to następująco:

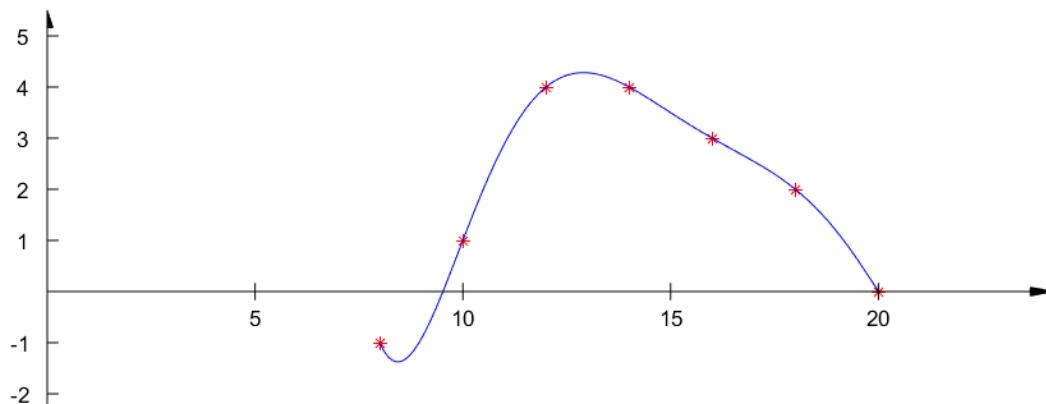


Rys 3.3.1. Rozkład temperatury z danych pomiarowych

Mamy 7 pomiarów, to znaczy wartość funkcji - temperatury - jest dana w 7 równoodległych węzłach. Szukany wielomian interpolacyjny będzie zatem $n = 6$ stopnia. Po zastosowaniu wzoru na wielomian Lagrange'a dostajemy wynik:

$$W_6(x) = 1187 - \frac{7949}{15}x + \frac{22847}{240}x^2 - \frac{283}{32}x^3 + \frac{173}{384}x^4 - \frac{23}{1920}x^5 + \frac{1}{7680}x^6 \quad (1)$$

Poniżej podajemy rysunek pomiarowych i wielomianu interpolacyjnego.



Możemy teraz obliczać przybliżoną temperaturę w dowolnej porze między godziną 8 a godziną 20-tą. Otrzymujemy np. że o godzinie 13:30 temperatura wynosiła 4, 187 stopnia, o godzinie 15:45 wynosiła 3, 119 stopnia, a o godzinie 10:15 miała wartość 1, 52 stopnia.

Zaimplementujmy teraz funkcję w MATLABie, która będzie otrzymywać jako parametry węzły interpolacji oraz zadany punkt x dla którego powinien zostać obliczona wartość wielomianu interpolacyjnego Lagrange'a. W programie wykorzystujemy wyrażenie na wielomian Lagrange'a:

$$WL_n(x) = \sum_{i=0}^n y_i \prod_{j=0, i \neq j}^n \frac{x - x_j}{x_i - x_j} \quad (3.3.4)$$

```
% x,y - wektory wezlow interpolacji
% xval - wspolrzędna x dla której należy obliczyć wartość wielomianu
function y = lagr(x,y,xval)
    n = length(x);
    suma = 0;
    for i = 1:n
        ilocz = 1;
        for j = 1:n
            if (i ~= j)
                ilocz = ilocz * (xval - x(j)) / (x(i) - x(j));
            end
        end
        suma = suma + y(i) * ilocz;
    end
    y = suma;
end
```

Przykład 3.3.**Przykład**

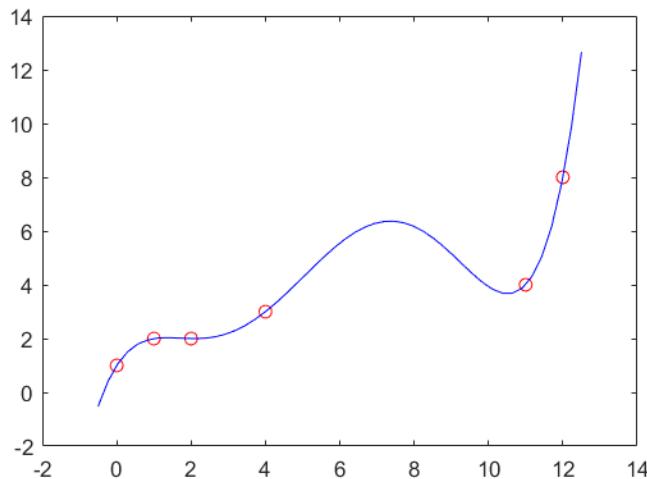
Wykorzystując funkcję do wyznaczania wartości wielomianu Lagrange'a napisz program, który narysuje przebieg wielomianu interpolacyjnego dla zadanych węzłów interpolacji.

Rozwiązanie:

```
x = [0 1 2 4, 11, 12]
y = [1 2 2 3, 4, 8] % wygeneruj 50 punktów równoodległych pomiędzy najmniejszą i największą
% wartością zbioru węzłów interpolacji
xd = linspace(min(x)-0.5,max(x)+0.5, 50);
yd = zeros(length(xd),1);

for i = 1:length(xd)
    yd(i) = lagr(x,y,xd(i));
end
plot(x,y, 'or', xd, yd, 'b')
```

Program zwraca wykres jak następuje:



4. Wielomian Newtona

Wielomian interpolacyjny Newtona

Inną postacią wielomianu interpolacyjnego jest **wielomian interpolacyjny Newtona**. Do zdefiniowania tego wielomianu wykorzystamy **ilorazy różnicowe** n-tego rzędu dla funkcji $y = f(x)$, która w $(n+1)$ różnych punktach: $(x_0, x_1, x_2, \dots, x_n)$ z przedziału (a, b) przyjmuje wartości $y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n)$.

Ilorazy różnicowe I rzędu:

$$(f(x_0, x_1) = \frac{y_1 - y_0}{x_1 - x_0}, \dots, f(x_{k-1}, x_k) = \frac{y_k - y_{k-1}}{x_k - x_{k-1}}, \dots, f(x_{n-1}, x_n) = \frac{y_n - y_{n-1}}{x_n - x_{n-1}}).$$

Ilorazy różnicowe II rzędu:

$$(f(x_0, x_1, x_2) = \frac{f(x_1, x_2) - f(x_0, x_1)}{x_2 - x_0}, \dots, f(x_{n-2}, x_{n-1}, x_n) = \frac{f(x_{n-1}, x_n) - f(x_{n-2}, x_{n-1})}{x_n - x_{n-2}}).$$

I ogólnie iloraz (m) -tego rzędu $(m=2, \dots, n, k=2, \dots, n)$

$$(f(x_{k-m}, \dots, x_{k-1}, x_k) = \frac{f(x_{k-m+1}, \dots, x_{k-1}, x_k) - f(x_{k-m}, \dots, x_{k-1})}{x_k - x_{k-m}}) \quad (3.3.5)$$

Iloraz (n) -tego rzędu jest tylko jeden i ma postać:

$$(f(x_0, x_1, \dots, x_n) = \frac{f(x_1, \dots, x_n) - f(x_0, \dots, x_{n-1})}{x_n - x_0}).$$

Zauważmy, że aby obliczyć ilorasy różnicowe rzędu (m) , trzeba podzielić różnicę ilorazów rzędu $(m-1)$ przez różnicę wartości węzłów o współrzędnych różniących się o (m) . Jest to na pierwszy rzut oka dość skomplikowane. Zapiszemy w tabeli ilorasy różnicowe dla funkcji, dla której są dane wartości tylko w trzech punktach:

x_0	y_0	Ilorasy I rzędu	Iloraz II rzędu
x_1	y_1	$f(x_0, x_1) = \frac{y_1 - y_0}{x_1 - x_0}$	
x_2	y_2	$f(x_1, x_2) = \frac{y_2 - y_1}{x_2 - x_1}$	$f(x_0, x_1, x_2) = \frac{f(x_1, x_2) - f(x_0, x_1)}{x_2 - x_0}$

Przykład 3.4

Przykład

Funkcja $f(x)$ dana jest za pomocą tabelki

$$(\begin{array}{cccc} x_i & y_i & & \\ \hline -1 & 0 & & \\ 1 & 4 & & \\ 3 & 6 & & \\ 5 & 2 & & \end{array})$$

Przykład 3.4.1 : Funkcja $f(x)$ dana jest za pomocą tabelki: $\begin{array}{cccc} x_i & y_i & & \\ \hline -1 & 0 & & \\ 1 & 4 & & \\ 3 & 6 & & \\ 5 & 2 & & \end{array}$. Obliczymy dla niej ilorasy różnicowe do $n=3$ rzędu włącznie. Zauważmy, że węzłów jest $n+1=4$, wtedy ostatni iloraz jest n -tego rzędu - czyli trzeciego

x_i	y_i	I rzędu	II rzędu	III rzędu
-1	0			
1	4	$\frac{4-0}{1-(-1)} = 2$		
3	6	$\frac{6-4}{3-1} = 1$	$\frac{1-2}{3-1} = -\frac{1}{2} = -0,25$	
5	2	$\frac{2-6}{5-3} = -2$	$\frac{-2-1}{5-1} = -\frac{3}{4} = -0,75$	$\frac{-0,75-(-0,25)}{5-(-1)} = \frac{-0,5}{6} = -\frac{1}{12}$

Możemy teraz podać postać wielomianu interpolacyjnego Newtona dla funkcji $(f(x))$:

$$(\begin{aligned} W_n(x) = y_0 + & f(x_0, x_1)(x-x_0) + f(x_0, x_1, x_2)(x-x_0)(x-x_1) + \dots + \\ & f(x_0, x_1, \dots, x_n)(x-x_0)(x-x_1)\dots(x-x_{n-1}) \end{aligned}) \quad (3.4.1)$$

Widać z tej postaci, że jest to wielomian co najwyżej $\lfloor n \rfloor$ -tego stopnia. W ostatnim składniku jest co najwyżej $\lfloor x \rfloor$ do potęgi $\lfloor n \rfloor$ -tej (ale iloraz różnicowy $\lfloor n \rfloor$ -tego rzędu może być równy zero). Trudniej jest sprawdzić, że wielomian ten w węzłach pokrywa się z funkcją $f(x)$. Sprawdzimy ten warunek tylko dla wielomianu drugiego stopnia, który ma postać:

$$\langle WN_2(x) = y_0 + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1) \rangle$$

Dla $\langle x=x_0 \rangle$ widać, że

$$\langle W_2(x_0) = y_0 + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1) \rangle$$

zatem $\langle WN_2(x_0) = y_0 \rangle$

Dla $\langle x=x_1 \rangle$ mamy:

$$\langle WN_2(x_1) = y_0 + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1) \rangle$$

Ostatni składnik znika, zostają dwa składniki, w tym drugim skorzystamy ze wzoru na iloraz różnicowy pierwszego rzędu i otrzymamy:

$$\langle W_2(x_1) = y_0 + f(x_0, x_1)(x - x_0) = y_0 + \frac{y_1 - y_0}{x_1 - x_0} (x - x_0) = y_0 + y_1 - y_0 = y_1 \rangle$$

Dla $\langle x=x_2 \rangle$

$$\langle WN_2(x_2) = y_0 + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1) \rangle$$

i rozpiszemy iloraz drugiego rzędu w ostatnim składniku:

$$\begin{aligned} & \langle \begin{aligned} & \text{\& W N_2}\left(x_2\right)=y_0+f\left(x_0, x_1\right)\left(x-x_0\right)+\frac{f\left(x_1, x_2\right)-f\left(x_0, x_1\right)}{\left(x_2-x_0\right)}\left(x-x_0\right)\left(x-x_1\right) \\ & \text{\& } y_0+f\left(x_0, x_1\right)\left(x-x_0\right)+f\left(x_0, x_1, x_2\right)\left(x-x_0\right)\left(x-x_1\right)=y_0+\frac{y_2-y_0}{x_2-x_0}\left(x-x_0\right)\left(x-x_1\right)+\frac{y_2-y_1}{x_2-x_1}\left(x-x_1\right) \\ & \text{\& } x_1\left(x-x_1\right)=y_0+y_1-y_0+y_2-y_1=y_2 \end{aligned} \rangle \end{aligned}$$

Można to sprawdzić ogólnie, że $\langle WN_n(x_k) = y_k = f(x_k) \sim k=0,1,\dots,n \rangle$.

Powróćmy do przykładu 3.4, dla tej funkcji wielomian Newtona będzie następujący:

$$\langle WN_3(x) = 0 + 2(x+1) - \frac{1}{4}(x+1)(x-1) + \frac{1}{12}(x+1)(x-1)(x-3) \rangle$$

Przykład 3.5

Przykład

Zmierzliśmy wartość funkcji w przedziale $\langle x=1,4 \rangle$ tylko w dwóch węzłach i otrzymaliśmy wyniki: dla $\langle x=1 \rangle \langle y=3 \rangle$ i dla $\langle x=4 \sim y=6 \rangle$. Wielomian interpolacyjny dla tej funkcji jest prostą przechodzącą przez te punkty. Węzőw jest $\langle n+1=2 \rangle$, wielomian jest stopnia $\langle n=1 \rangle$. Potem dodaliśmy jeszcze punkt dla $\langle x=2 \rangle$ wartość funkcji $\langle y=3 \rangle$ i poprowadziliśmy wielomian stopnia 2. Następnie dodaliśmy jeszcze jeden pomiar i dla $\langle x=3 \rangle$ otrzymaliśmy $\langle y=2 \rangle$. Teraz mamy 4 węzły w przedziale, a wielomian jest 3-stopnia.

Rozwiązańie

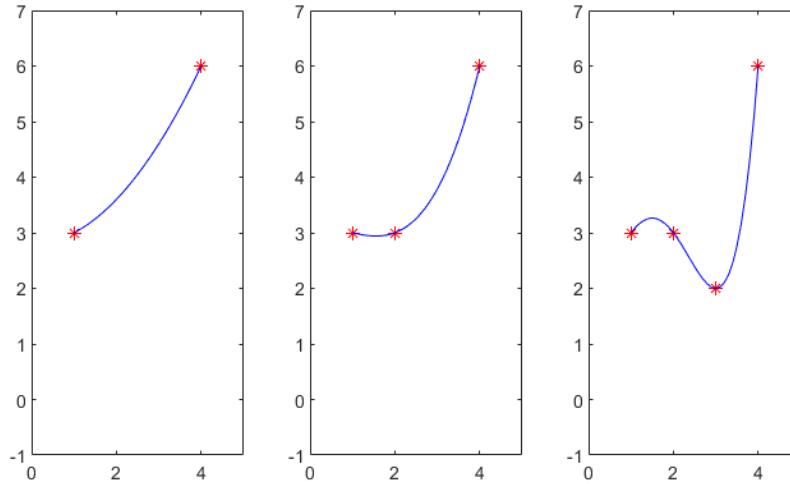
Do rozwiązania zadania posłużymy się środowiskiem MATLAB oraz wbudowaną funkcją *polyfit* oraz *polyval*. Program ilustrujący to zadanie wygląda następująco.

```
% definiujemy funkcję do wygenerowania gładkiego wykresu
xd = 1:0.1:4
% pierwszy przypadek - 1 stopień
x = [1 4]
y = [3 6]
a = polyfit(x,y,2)
yd = polyval(a,xd)
subplot(1,3,1)
plot(x,y,'*r',xd,yd, 'b')
xlim([0,5])
ylim([-1,7])

% drugi przypadek - 2 stopień
x = [1 2 4]
y = [3 3 6]
a = polyfit(x,y,3)
yd = polyval(a,xd)
subplot(1,3,2)
plot(x,y,'*r',xd,yd, 'b')
xlim([0,5])
ylim([-1,7])

% trzeci przypadek - 3 stopień
x = [1 2 3 4]
y = [3 3 2 6]
a = polyfit(x,y,4)
yd = polyval(a,xd)
subplot(1,3,3)
plot(x,y,'*r',xd,yd, 'b')
xlim([0,5])
ylim([-1,7])
```

Oto wynik programu z wygenerowanymi przebiegami.



Rysunek 3.7. Przebiegi funkcji interpolującej z rozwiążaniem zadania z przykładu 3.5

4 Rozdział

Strona: [LeIA](#)

Kurs: Metody numeryczne (2025Z)

Książka: 4 Rozdział

Wydrukowane przez użytkownika: Kinga Kondraciuk

Data: niedziela, 30 listopada 2025, 14:13

Spis treści

- 1. Błąd interpolacji**
- 2. Węzły Czebyszewa**
- 3. Zbieżność procesów interpolacyjnych**

1. Błąd interpolacji

Błąd interpolacji

W lekcji tej omawiamy błąd interpolacji, wprowadzamy węzły Czebyszewa, które minimalizują tę część błędu, która zależy od węzłów. Zajmujemy się również zbieżnością procesów interpolacyjnych.

Jeśli funkcja $f(x)$ jest dana za pomocą tabelki, to znaczy jej wartości są wynikiem doświadczeń lub pomiarów, nie możemy określić błędu jaki popełniamy biorąc za wartość funkcji w punkcie nie będącym węzłem wielomianu interpolacyjnego. Ale są przypadki gdy funkcja jest dana wzorem analitycznym $y = f(x)$ w przedziale a, b , a mimo to potrzebujemy zbudować dla niej wielomian interpolacyjny. Taka sytuacja ma miejsce przede wszystkim przy całkowaniu, o czym będziemy mówić w rozdziale dotyczącym całkowania numerycznego. W takim przypadku można obliczyć błąd interpolacji, zależy on od funkcji, a właściwie od pochodnej rzędu $n + 1$, oraz od sposobu rozmieszczenia węzłów interpolacji. Ponieważ wzór (podajemy ten wzór bez wyprowadzania) na różnicę między funkcją interpolowaną (oczywiście taką, która ma w tym przedziale wszystkie pochodne do rzędu $n + 1$ włącznie) a wielomianem interpolacyjnym jest następujący:

$$f(x) - W_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)(x - x_1) \dots (x - x_n) \quad (4.1.1)$$

gdzie ξ jest pewnym punktem w przedziale a, b , to błąd bezwzględny interpolacji można oszacować przez:

$$|f(x) - W_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |(x - x_0)(x - x_1) \dots (x - x_n)| \quad (4.1.2)$$

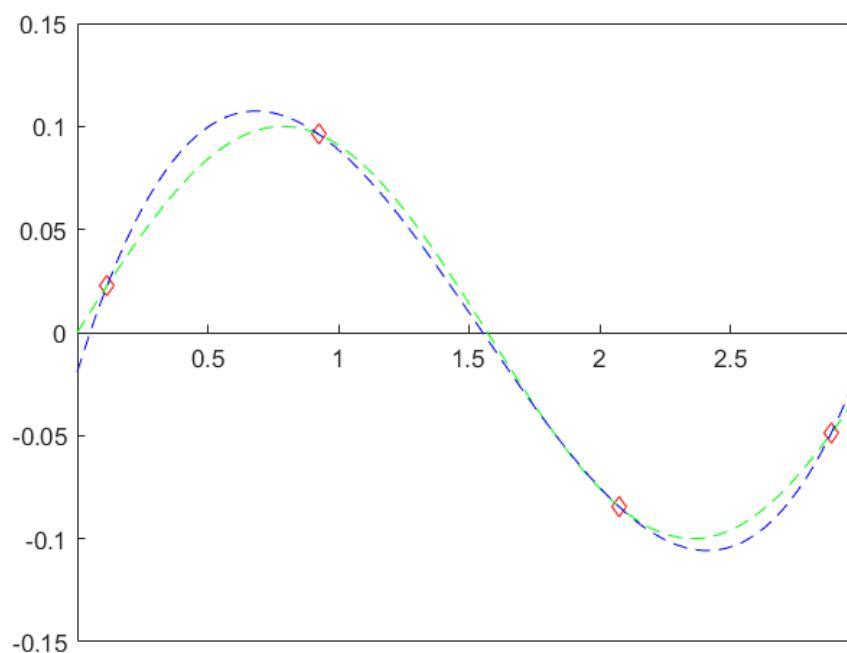
gdzie $M_{n+1} = \sup_{x \in [a, b]} |f^{(n+1)}(x)|$

porównajmy, dla przykładu funkcję która ma ograniczone pochodne z jej wielomianem interpolacyjnym.

Przykład 4.1

Przykład

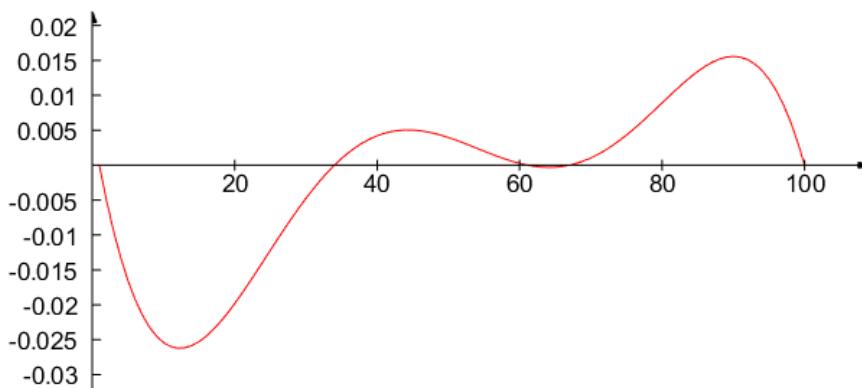
Weźmy funkcję $f(x) = 0.1 \sin(2x)$ w przedziale $0, 3$ i jako węzły interpolacji przyjmiemy cztery punkty równoodległe w tym przedziale $0, 1, 2, 3$. Funkcję i wielomian przedstawimy na wykresie:



Rys 4.1. Wykres funkcji $f(x) = 0.1\sin 2x$ i jej wielomianu interpolacyjnego 3 stopnia.

Na osi $0x$ zaznaczone są węzły, widać, że funkcja i wielomian pokrywają się w węzłach. Funkcja jest narysowana zieloną linią, wielomian niebieską przerywaną.

Na następnym wykresie przedstawiona jest funkcja będąca różnicą $f(x)$ i wielomianu interpolacyjnego. Jak widać błąd bezwzględny nie przekracza 0,03 w rozpatrywanym przedziale.



Rys 4.2. Wykres różnicy między funkcją f i wielomianem interpolacyjnym

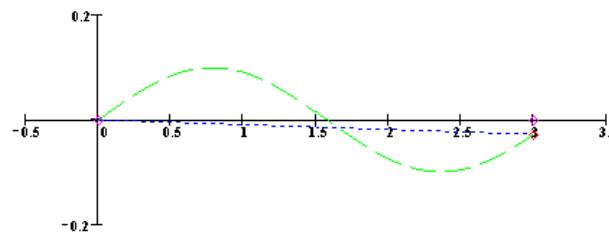
W poprzednim przykładzie obliczyliśmy wielomian 3 stopnia dla podanej funkcji, w następnym będziemy zmieniać stopień wielomianu dla tej samej funkcji.

Przykład 4.2

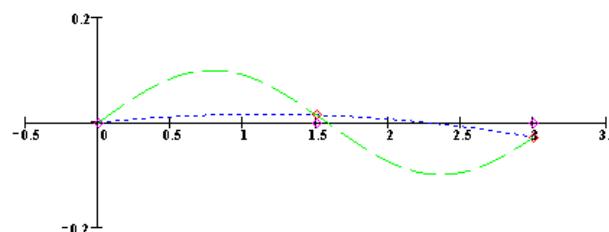
Przykład

Będziemy rozpatrywać jeszcze raz poprzednią funkcję, a mianowicie: $f(x) = 0.1\sin 2x$ w przedziale $<0, 3>$. Funkcja ta ma ograniczone pochodne w przedziale, wraz ze wzrostem ilości węzłów (bierzemy węzły równoodległe) wielomian interpolacyjny coraz lepiej będzie przybliżał daną funkcję. Gdy zmieniamy stopień wielomianu n , ilość węzłów $n + 1$ to wyraźnie widzimy, że wraz ze wzrostem stopnia

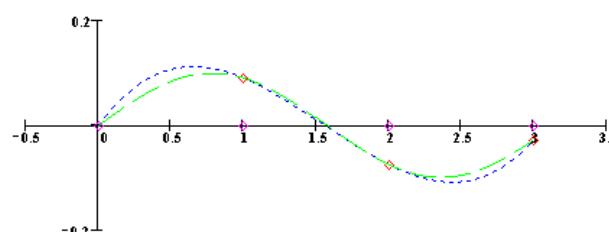
zmniesza się błąd interpolacji. Na osi $0x$ zaznaczone są węzły, widać na rysunku, że funkcja w węzłach pokrywa się z wielomianem. Wykres funkcji narysowany jest na zielono, dla $n = 8$ wykresy na naszym rysunku prawie się pokrywają.



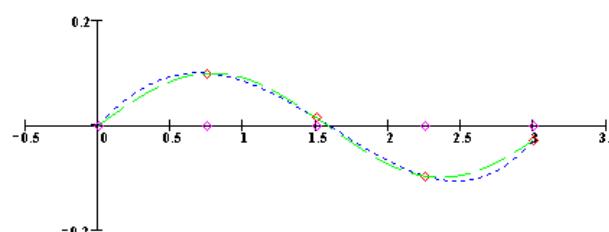
Rys 4.3a. Wykres funkcji f i jej wielomianu interpolacyjnego 1 stopnia



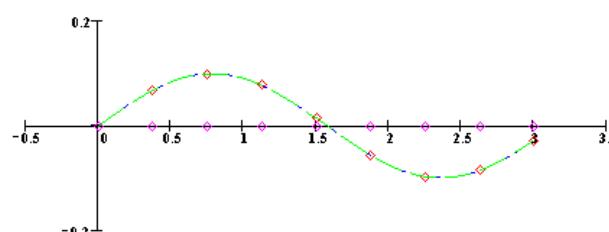
Rys 4.3b. Wykres funkcji f i jej wielomianu interpolacyjnego 2 stopnia.



Rys 4.3c. Wykres funkcji f i jej wielomianu interpolacyjnego 3 stopnia



Rys 4.3d. Wykres funkcji f i jej wielomianu interpolacyjnego 4 stopnia.



Rys4.3e. Wykres funkcji f i jej wielomianu interpolacyjnego 8 stopnia.

Przyjrzyjmy się teraz implementacji rysującej te wykresy.

```
close all
f = @(x) (0.1*sin(2*x))
x = linspace(0, 3, 100);
y = f(x)

% zmieniaj poniżej rzad w zakresie 2-5,
% pamiętaj, że n=4 oznacza 4 węzły
% ale 3ci stopień wielomianu
n = 4
xk = linspace(0, 3, n);
yk = f(xk);
a = polyfit(xk, yk, n-1);
y_interp = polyval(a, x);

plot(xk, yk, 'dr', x, y, '--g', x, y_interp, '--b');

%centeraxes(gca);

ax = gca;
ax.XAxisLocation = 'origin';
ay.YAxisLocation = 'origin';

figure

plot(y-y_interp, 'r')

%centeraxes(gca);

ax = gca;
ax.XAxisLocation = 'origin';
ay.YAxisLocation = 'origin';
```

Dla dwóch węzłów maksymalny błąd interpolacji w tym przedziale równa się 0,107, dla trzech węzłów równa się 0,098, dla czterech 0,026, dla pięciu 0,016, a dla $n = 8$ czyli dla dziewięciu węzłów błąd nie przekracza 0,000084.

2. Węzły Czebyszewa

Najczęściej stosuje się węzły równoodległe dla funkcji interpolacyjnej, które są proste w użyciu, a podczas doświadczeń można mierzyć badaną wartość funkcji co ustaloną jednostkę czasu. Jednak jak wynika ze wzoru na błąd interpolacji:

$$|f(x) - W_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |(x - x_0)(x - x_1)\dots(x - x_n)|$$

jego wartość zależy w istotny sposób od rozmieszczenia węzłów. Okazuje się, że węzły równoodległe nie zawsze są najlepsze. Tę część błędu, zależną od węzłów minimalizują tzw.: węzły Czebyszewa, które podamy tutaj bez wyprowadzania. Jeśli szukamy w dowolnym przedziale a, b $n+1$ optymalnych węzłów, można je wyliczyć ze wzoru:

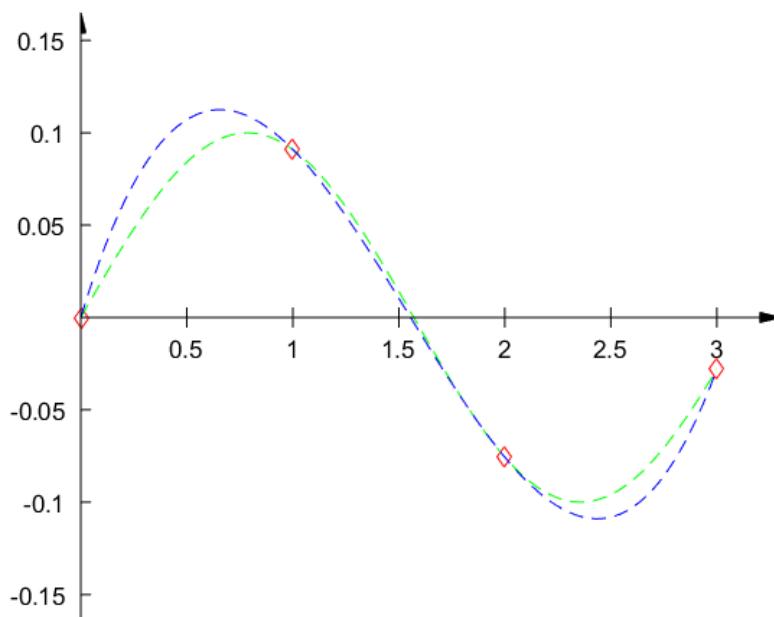
$$x_k = \frac{b-a}{2} \cos\left(\frac{2k+1}{2n+2}\pi\right) + \frac{b+a}{2} \quad k = 0, 1, 2\dots n \quad (4.2.1)$$

Wróćmy do przykładu 4.1.1 z funkcją $f(x) = 0.1 \sin 2x$ w przedziale $0, 3$.

Przykład 4.3

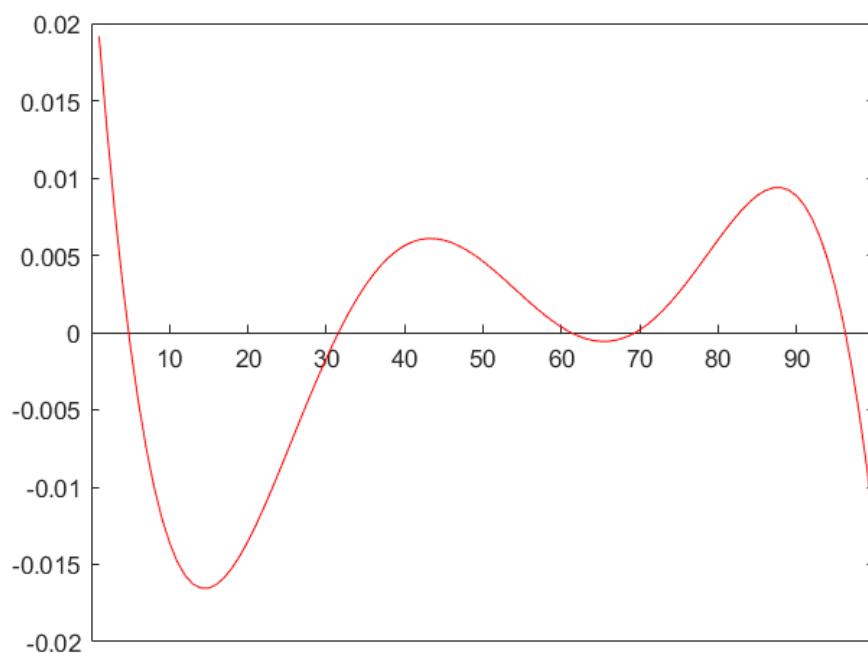
Przykład

Poprzednio dla zbudowania wielomianu interpolacyjnego braliśmy cztery węzły równoodległe : 0,1 2 i 3. Teraz obliczymy 4 węzły Czebyszewa w tym przedziale z powyższego wzoru. Będą to : 0,114; 0,926; 2,074; 2,886 (podajemy te wartości z dokładnością do trzech cyfr po przecinku). Te węzły są zaznaczone na osi Ox , funkcja jest narysowana zieloną linią, wielomian niebieską przerywaną:



Rys. 4.4. Wykres funkcji f i jej wielomianu interpolacyjnego 3 stopnia z 4 węzłami Czebyszewa

Tak jak poprzednio na następnym rysunku przedstawiamy różnicę między funkcją daną, a jej wielomianem interpolacyjnym opartym na węzłach Czebyszewa. Największy błąd bezwzględny nie przekroczy 0,02 w tym przedziale (jest on trochę mniejszy niż w poprzednim przykładzie z węzłami równoodległymi – tam było 0,03).



Rys. 4.5. Wykres różnicy między funkcją f i jej wielomianem interpolacyjnym.

Od razu dla porównania przyjrzyjmy się zmodyfikowanej implementacji.

```
close all
f = @(x) (0.1*sin(2*x))
x = linspace(0,3,100);
y = f(x)

% zmieniaj poniżej rząd w zakresie 2-5,
% pamiętaj, że n=4 oznacza 4 wezły
% ale 3ci stopień wielomianu
n = 4

% odkomentuj dla wezłów równoodległych
% xk = linspace(0,3,n);

% odkomentuj dla wezłów Czebyszewa
k=0:n-1;
a=0;
b=3;
xk = (b-a)/2*cos((2*k+1)/(2*(n-1)+2)*pi) + (b+a)/2;

yk = f(xk)

a = polyfit(xk,yk,n-1);
y_interp = polyval(a,x);

plot(xk,yk, 'dr', x,y, '--g', x, y_interp, '--b');
%centeraxes(gca);
ax = gca;
ax.XAxisLocation = 'origin';
ax.YAxisLocation = 'origin';

figure

plot(y-y_interp,'r')
%centeraxes(gca);
ax = gca;
ax.XAxisLocation = 'origin';
ax.YAxisLocation = 'origin';
```

3. Zbieżność procesów interpolacyjnych

Zbieżność procesów interpolacyjnych

Jeszcze raz podamy wzór na błąd interpolacji dla funkcji określonej w przedziale a, b i mającej w tym przedziale pochodne do rzędu $n + 1$ włącznie.

$$|f(x) - W_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |(x - x_0)(x - x_1)\dots(x - x_n)|$$

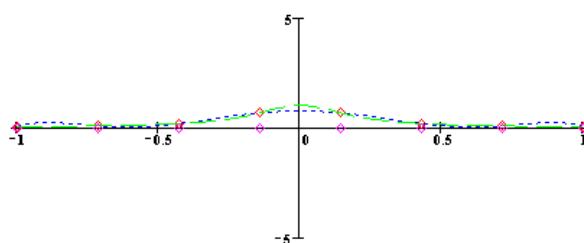
gdzie $M_{n+1} = \sup_{x \in [a, b]} |f^{(n+1)}(x)|$.

Z tego wzoru wynika, że wraz ze wzrostem ilości węzłów mianownik szybko rośnie, bo jest w nim wyraz $(n + 1)!$ zatem cały ułamek winien maleć i przez to maleć powinien błąd. Ale na błąd ma wpływ wielkość ograniczająca pochodną $n + 1$ rzędu. Podamy popularny w literaturze przykład funkcji, która ma wszystkie pochodne ograniczone, ale na tyle dużej wartości, że wraz ze wzrostem ilości węzłów błąd interpolacji rośnie tzn.: "rozjeżdża" się wielomian z funkcją interpolowaną.

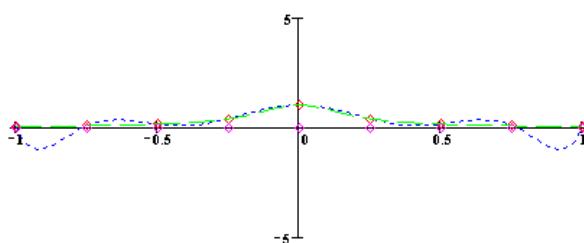
Przykład 4.4

Przykład

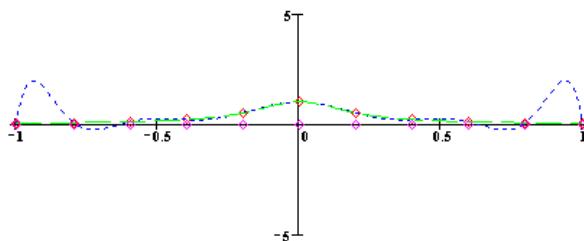
Będziemy rozpatrywać funkcję: $f(x) = \frac{1}{1+25x^2}$ w przedziale $-1, 1$. Na początku będziemy brać osiem węzłów równoodległych tzn. $n = 7$. Wtedy wielomian interpolacyjny i funkcja będą zachowywać się "poprawnie", niezbyt się od siebie różnić. Pierwsza seria rysunków obrazuje tę sytuację. Jeśli będziemy zwiększać ilość węzłów interpolacja będzie obarczona coraz to większym błędem. Zjawisko to widać na rysunkach 4.6a-d.



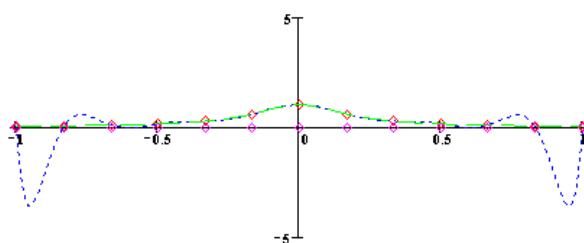
Rys. 4.6a. Wykres funkcji i jej wielomianu interpolacyjnego 7 stopnia.



Rys. 4.6b. Wykres funkcji i jej wielomianu interpolacyjnego 8 stopnia.

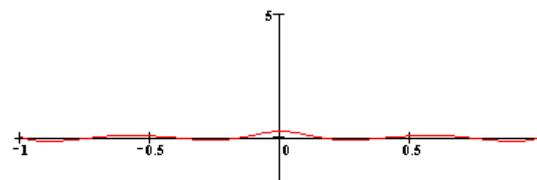


Rys. 4.6c. Wykres funkcji i jej wielomianu interpolacyjnego 10 stopnia

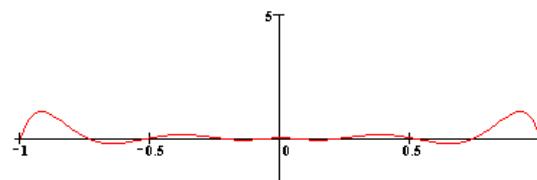


Rys. 4.6d. Wykres funkcji i jej wielomianu interpolacyjnego 12 stopnia.

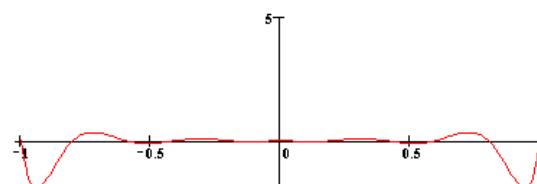
Podobnie będzie z rysunkiem przedstawiającym różnice między funkcją interpolowaną a wielomianem interpolacyjnym o równoodległych węzłach. Te różnice dla $n = 12$ czyli dla 13 węzłów będą bardzo duże w porównaniu do wartości funkcji w tym przedziale. Zaprezentowane to zostało na rysunkach 4.7a-d.



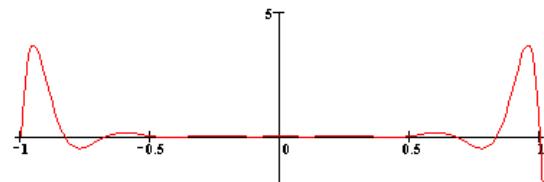
Rys. 4.7a. Wykres różnic między funkcją i jej wielomianem interpolacyjnym 7 stopnia.



Rys. 4.7b. Wykres różnic między funkcją i jej wielomianem interpolacyjnym 8 stopnia.



Rys. 4.7c. Wykres różnic między funkcją i jej wielomianem interpolacyjnym 10 stopnia.



Rys. 4.7d. Wykres różnic między funkcją i jej wielomianem interpolacyjnym 12 stopnia.

Trzeba sobie zdać sprawę z takich faktów, że dla $n = 12$ pochodna rozpatrywanej funkcji $n + 1 = 13$ rzędu równa jest w punkcie $0, 1$ wartości $-3, 29 \cdot 10^{17} a_{13!} = 6.277 \cdot 10^9$

Bibliografia

1. Katarzyna Litewska, Jerzy Muszyński - Analiza matematyczna t I i II . Oficyna Wydawnicza Politechniki Warszawskiej. Warszawa 2000.
2. Julian Klukowski, Ireneusz Nabiałek - Algebra dla studentów . Wydawnictwo Naukowo-Techniczne . Warszawa 1999.
3. Adam Grabarski, Irena Musiał-Walczak, Wawrzyniec Sadkowski, Alicja Smoktunowicz, Janusz Wąsowski- Ćwiczenia z metod numerycznych. Oficyna Wydawnicza Politechniki Warszawskiej. Warszawa 2002.
4. Zenon Fortuna, Bohdan Macukow, Janusz Wąsowski - Metody numeryczne. Wydawnictwo Naukowo-Techniczne. Warszawa 1993.
5. J.Klamka, Z.Ogonowski, M.Jamicki, M.Stasiak - Metody numeryczne. Wydawnictwo Politechniki Śląskiej. Gliwice 2004

5 rozdział

Strona: [LeIA](#)

Kurs: Metody numeryczne (2025Z)

Książka: 5 rozdział

Wydrukowane przez użytkownika: Kinga Kondraciuk

Data: niedziela, 30 listopada 2025, 14:14

Spis treści

- 1. Baza funkcji sklejanych**
- 2. Interpolacja splajnami na bazie równoodległej**

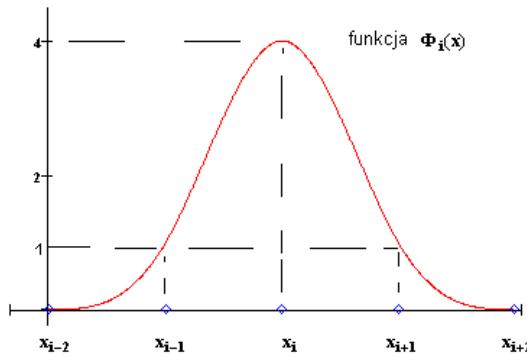
1. Baza funkcji sklejanych

Baza funkcji sklejanych

Stosując do interpolacji wielomian interpolacyjny nie możemy narzucać stopnia wielomianu, ten stopień zależy od ilości węzłów. Jeśli mamy 20 różnych węzłów (20 pomiarów) to wielomian interpolacyjny może być nawet 19 stopnia. Wraz ze wzrostem ilości węzłów rośnie na ogół stopień wielomianu. Natomiast stopień niżej zdefiniowanej funkcji sklejanej tzw.: splajnu, nie będzie zależał od ilości węzłów.

Ograniczymy się w tym opracowaniu do splajnu 3-ego stopnia, jest on na ogół najczęściej używany do interpolacji. Będziemy rozpatrywać przedział i podzielimy go na n części, czyli na n podprzedziałów o długości $h = \frac{b-a}{n}$. Otrzymamy węzły równoodległe $x_i = a + i \cdot h$ $i = 0, 1, \dots, n$. Funkcją sklejaną 3-ego stopnia będziemy nazywać funkcję, która na każdym podprzedziale jest wielomianem 3 stopnia, ale posklejaną tak, aby była ciągła i miała pierwszą i drugą pochodną ciągłą na $\langle a, b \rangle$. Aby dokładnie określić funkcję sklejaną 3-ego stopnia $S_3(x)$ na przedziale $\langle a, b \rangle$ określmy najpierw bazę splajnu 3-iego stopnia dla węzłów równoodległych. Jedna funkcja bazowa jest podana za pomocą bardzo skomplikowanego wzoru, ale musi spełniać powyższe wymagania, tzn.: musi być wielomianem 3-ego stopnia na każdym podprzedziale, mieć pierwszą i drugą pochodną ciągłą na $\langle a, b \rangle$. Funkcja bazowa o numerze i , oznaczona przez $\Phi_i(x)$ ma w węźle o numerze i maksimum równe 4, w węzłach obok ma wartość 1, a w węzłach o numerach $i-2$ i $i+2$ ma wartość 0. Oto wzór i wykres takiej funkcji:

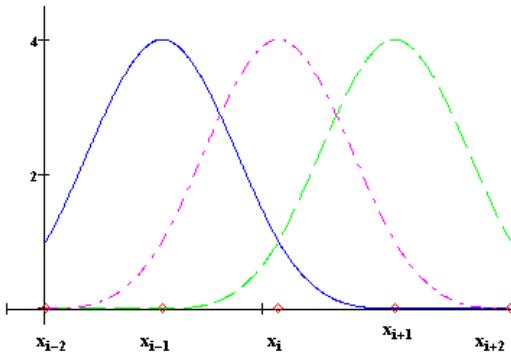
$$\Phi_i(x) = \frac{1}{h^3} \cdot \begin{cases} (x - x_{i-2})^3 & \text{dla } x \in \langle x_{i-2}, x_{i-1} \rangle \\ (x - x_{i-2})^3 - 4(x - x_{i-1})^3 & \text{dla } x \in \langle x_{i-1}, x_i \rangle \\ (x_{i+2} - x)^3 - 4(x_{i+1} - x)^3 & \text{dla } x \in \langle x_i, x_{i+1} \rangle \\ (x_{i+2} - x)^3 & \text{dla } x \in \langle x_{i+1}, x_{i+2} \rangle \\ 0 & \text{dla } x \in R - \langle x_{i-2}, x_{i+2} \rangle \end{cases} \quad (5.1.1)$$



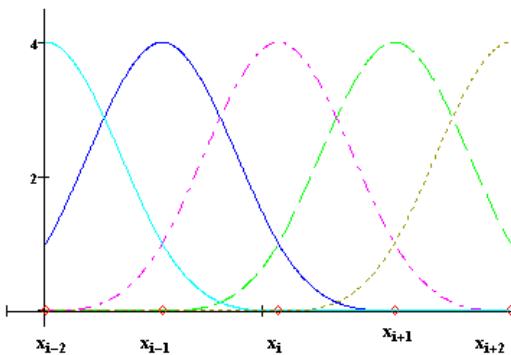
Rys. 5.1. Funkcja bazowa o numerze i , takim samym jak węzeł. Wartość maksymalną równą 4 funkcja przyjmuje dla wartości $x = x_i$.

Na podstawie tych funkcji bazowych będziemy określać w przedziale $\langle a, b \rangle$ z $n+1$ węzłami równoodległymi splajn 3-ego stopnia. Ile jest takich funkcji bazowych w przedziale $\langle a, b \rangle$

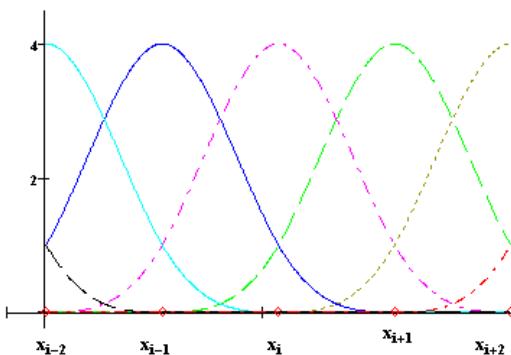
Narysujemy po kolej funkcje bazowe dla $n = 4$ tzn.: dla 5 węzłów. Narysujemy najpierw trzy $\Phi_1(x), \Phi_2(x), \Phi_3(x)$ (patrz rysunek 5.1a). Dodamy jeszcze dwie: $\Phi_0(x), \Phi_4(x)$ (patrz rysunek 5.1b). I mamy tyle funkcji ile jest węzłów. Ale są jeszcze dwie funkcje, które nie są równe 0 w całym przedziale, te funkcje odpowiadają węzłom, których na rysunku nie ma, jednemu o numerze wcześniejszym niż 0 i jednemu o numerze późniejszym niż 4. Te funkcje oznaczymy przez $\Phi_{-1}(x), \Phi_{n+1}(x)$ (patrz rysunek 5.1c). Okazuje się, że niezerowych funkcji jest o dwie więcej niż węzłów, czyli o 3 więcej niż n .



Rys. 5.1a. Trzy funkcje bazowe dla 3 węzłów: x_{i-1}, x_i, x_{i+1} , przy założeniu $i = 2$.



Rys. 5.1b. Pięć funkcji bazowych dla 5 węzłów: $x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}$, przy założeniu $i = 2$.



Rys. 5.1c. Siedem funkcji bazowych dla 7 węzłów: $x_{i-1}, x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}, x_{n+1}$, przy założeniu $i = 2$.

Oto funkcja napisana w MATLABie, który implementuje funkcję bazową zakładając, że przekazujemy mu trzy parametry:

- x_i - współrzędną środka danej funkcji bazowej, odpowiada to funkcji $\Phi_i(x)$,
- h - odległość pomiędzy węzłami interpolacji; odległość ta wyznacza granice między przedziałami istotnymi dla funkcji bazowej,
- x - wartość, dla której należy obliczyć wartość funkcji bazowej.

```
% xi - punkt środkowy funkcji bazowej
% h - odległość między węzłami
% x - współrzędna x, dla której obliczamy wartość
function y = phi(xi,h,x)
    if (x < xi-2*h) || (x > xi+2*h)
        y = 0;
    elseif x < xi-h
        y = (x - (xi-2*h))^3;
    elseif x < xi
        y = (x - (xi-2*h))^3 - 4*(x - (xi-h))^3;
    elseif x < xi+h
        y = ((xi+2*h) - x)^3 - 4*((xi+h) - x)^3;
    else
        y = ((xi+2*h) - x)^3;
    end
    y=y/h^3;
end
```

Przykład 5.1

Przykład

Wykorzystaj funkcję bazową w MATLABie i narysuj przebieg wszystkich 7 funkcji bazowych dla przypadku z pięcioma węzłami interpolacji równoodległymi w przedziale $<0, 5>$.

Rozwiązanie przedstawimy bezpośrednio w kodzie.

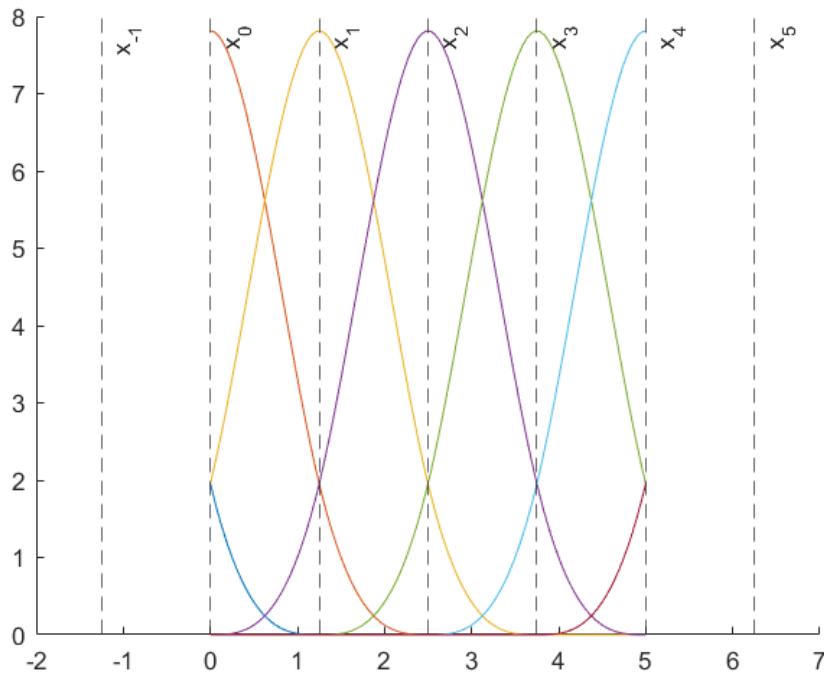
```
close all
a = 0;
b = 5;
xd = linspace(a,b,100);
xk = linspace(a,b,5);

h = xk(2)-xk(1);

yd = zeros(length(xd),1);

% używamy hold on, ponieważ chcemy dorysować kolejne przebiegi
% na tym samym rysunku
hold on
for i = -1:5
%for i = 0:4
%for i = 1:3
    % obliczamy współrzędna wezła środkowego i-tej funkcji
    % bazowej
    xi = xk(1) + (i)*h;
    % narysujemy pionowa linie aby zaznaczyć środek funkcji bazowej
    xline(xi, '--', {sprintf('x_{%d}',i)})

    for j = 1:length(xd)
        yd(j) = phi(xi, h, xd(j));
    end
plot(xd,yd)
end
hold off
```



Rys. 5.2. Wynik uruchomienia skryptu przykładowego, prezentujący siedem funkcji bazowych narysowanych w przedziale $<0, 5>$ z zaznaczonymi za pomocą pionowych przerywanych linii środkami funkcji bazowych.

Za pomocą tych funkcji zdefiniujemy funkcję sklejaną (splajn) 3-iego stopnia:

$$S_3(x) = c_{-1}\Phi_{-1}(x) + c_0\Phi_0(x) + c_1\Phi_1(x) + \dots + c_n\Phi_n(x) + c_{n+1}\Phi_{n+1}(x) \quad (5.1.2)$$

lub w skrócie:

$$S_3(x) = \sum_{i=-1}^{n+1} c_i\Phi_i(x) \quad (5.1.3)$$

Współczynniki są liczbami rzeczywistymi, będziemy je dobierać tak, aby splajn był funkcją interpolacyjną dla funkcji $f(x)$.

2. Interpolacja splajnami na bazie równoodległej

Zastosujemy funkcję sklejaną $S_3(x)$ do interpolacji funkcji $f(x)$ danej w przedziale a, b .

Dzielimy przedział na n części, $h = \frac{b-a}{n}$, węzły równoodległe $x_i = a + i \cdot h$ $i = 0, 1, \dots, n$.

Funkcja interpolacyjna musi się pokrywać w węzłach z funkcją $f(x)$ tzn.:

$$S_3(x_i) = y_i = f(x_i) \quad i = 0, 1, 2, \dots, n \quad (5.2.1)$$

Otrzymaliśmy z tych związków $n+1$ równań, a współczynników jest $n+3$, przypominamy wzór:

$$S_3(x) = c_{-1}\Phi_{-1}(x) + c_0\Phi_0(x) + c_1\Phi_1(x) + \dots + c_n\Phi_n(x) + c_{n+1}\Phi_{n+1}(x) \quad (5.2.2)$$

Nasz układ ma zatem dwa stopnie swobody i aby jednoznacznie wyznaczyć $S_3(x)$ musimy mieć jeszcze dwa równania. Na ogół zadaje się wartości pochodnej funkcji $S_3'(x)$ w punktach a i b - tzn.: zadaje się współczynniki kierunkowe stycznych pod jakimi funkcja interpolacyjna ma startować z punktu a w prawo i jak ma wpadać do b z lewej strony. Dodatkowe warunki to:

$$S_3'(a^+) = \alpha, S_3'(b^-) = \beta$$

Z warunków 5.2.1 i z własności funkcji bazowych i ich pochodnych dostajemy układ równań:

$$c_{i-1} + 4c_i + c_{i+1} = y_i \quad i = 0, 1, \dots, n \quad (5.2.3)$$

$$-c_{-1} + c_1 = \frac{h}{3} \cdot \alpha$$

$$-c_{n-1} + c_{n+1} = \frac{h}{3} \cdot \beta$$

Po wyliczeniu współczynników c_{-1}, c_{n+1} z równań (5.2.3) i po wstawieniu ich do (5.2.2) otrzymujemy następujący układ $n+1$ równań liniowych z $n+1$ niewiadomymi c_0, c_1, \dots, c_n :

$$\begin{aligned} 4c_0 + 2c_1 &= y_0 + \frac{h}{3} \cdot \alpha \\ c_0 + 4c_1 + c_2 &= y_1 \\ c_1 + 4c_2 + c_3 &= y_2 \\ \ddots &= \dots \\ c_{n-2} + 4c_{n-1} + c_n &= y_{n-1} \\ 2c_{n-1} + 4c_n &= y_n - \frac{h}{3} \cdot \beta \end{aligned}$$

Układ ten ma zawsze jedyne rozwiązanie na c_0, c_1, \dots, c_n , pozostałe 2 współczynniki obliczymy ze wzorów:

$$-c_{-1} + c_1 = \frac{h}{3} \cdot \alpha,$$

$$-c_{n-1} + c_{n+1} = \frac{h}{3} \cdot \beta.$$

Nie wyprowadzaliśmy układu równań, aby nie rozbudowywać tego tematu. Zainteresowanych obliczeniami odsyłamy do podanej literatury.

Przykład 5.2**Przykład**

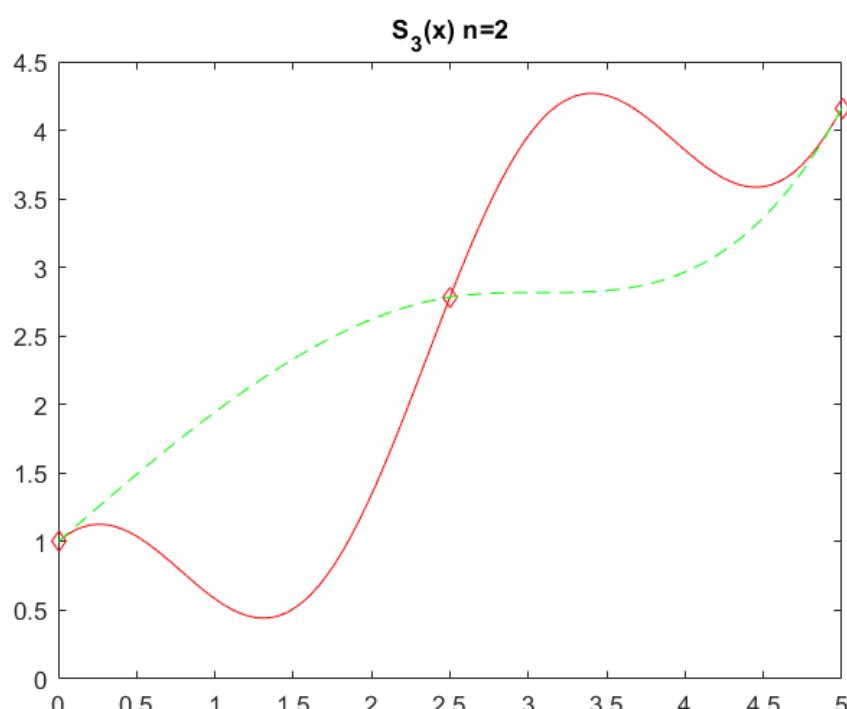
Dana jest funkcja $f(x) = x + \cos(2x)$ w przedziale $<0, 5>$. Znajdziemy dla niej funkcję sklejane 3-iego stopnia dla różnej ilości węzłów równoodległych. Liczba n oznacza ilość podprzedziałów, węzłów jest $n+1$. Ponieważ $f'(x) = 1 - 2\sin(2x)$ oraz $\alpha = f'(0) = 1, \beta = f'(5) = 2.088$ przyjmujemy, że $S'_3(a^+) = \alpha = 1, S'_3(b^-) = \beta = 2.088$.

Rozwiązujeć układ dla $n=2$ (3 węzły) dostajemy następujące współczynniki:

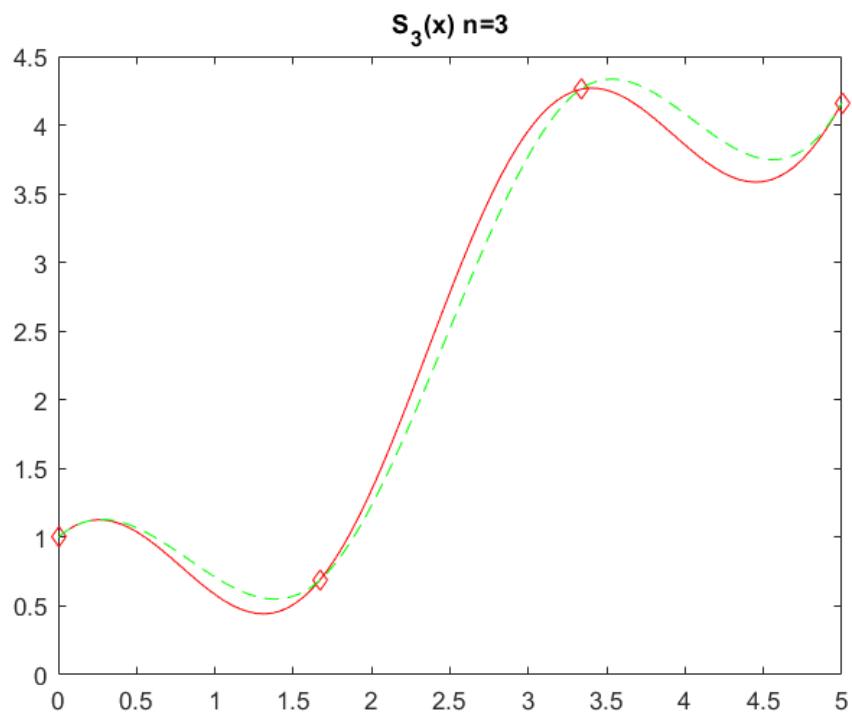
$$c_{-1} = -0,26; c_0 = 0,172; c_1 = 0,573; c_2 = 0,319; c_3 = 2,313$$

Narysujeśmy ten wykres, a później będziemy zwiększać liczbę podprzedziałów (węzłów): (nie podajemy współczynników następnych funkcji sklejanych, a tylko ich wykresy i błędy).

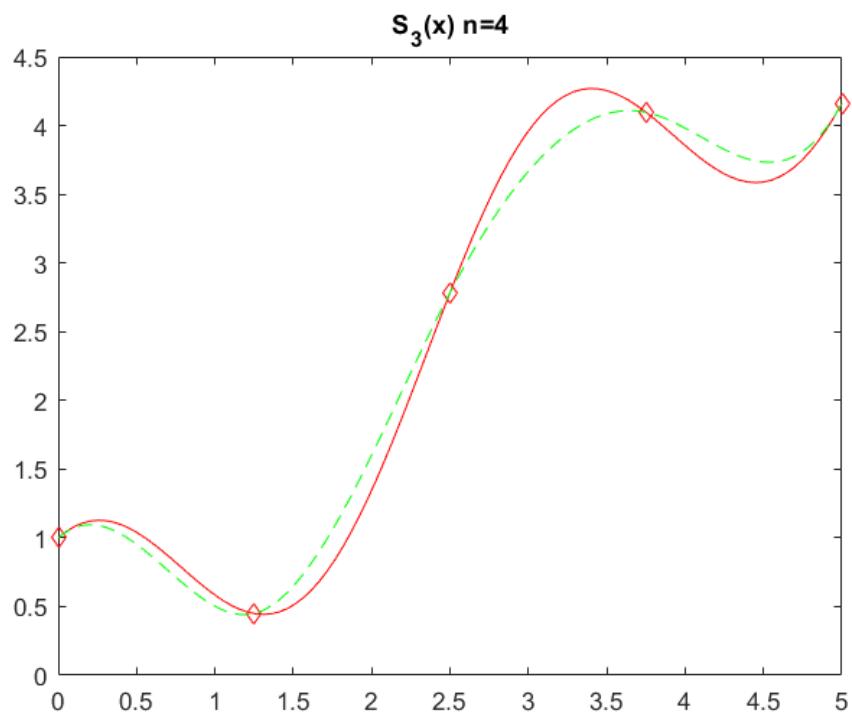
Dla $n = 2$ $n = 3$ $n = 4$ $n = 5$ $n = 8$ przedstawiono serię wykresów na rysunkach 5.3a-e.



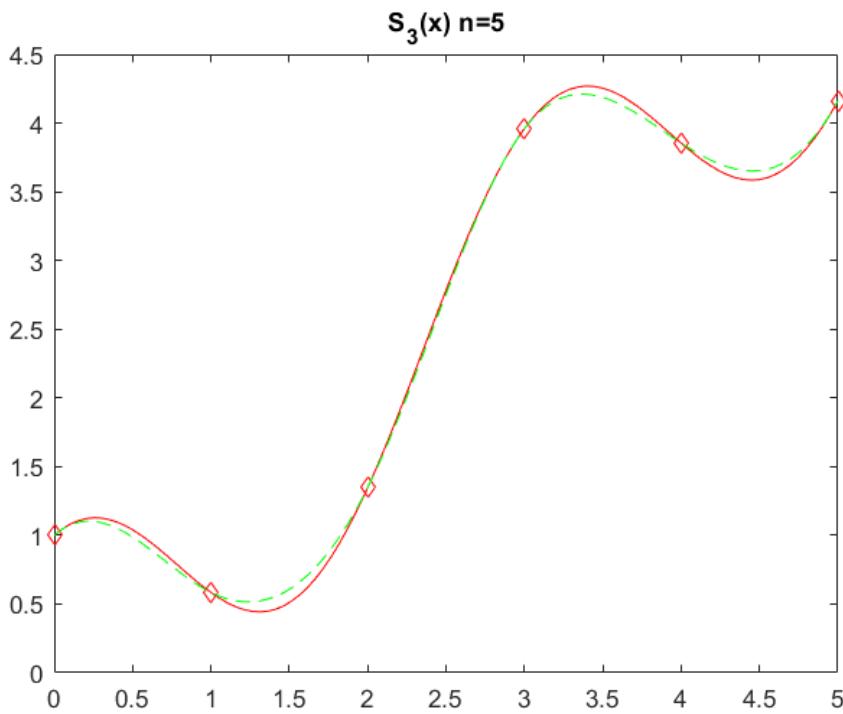
Rys. 5.3a. Wykres funkcji f i interpolacyjnych splajnów dla $n = 2$.



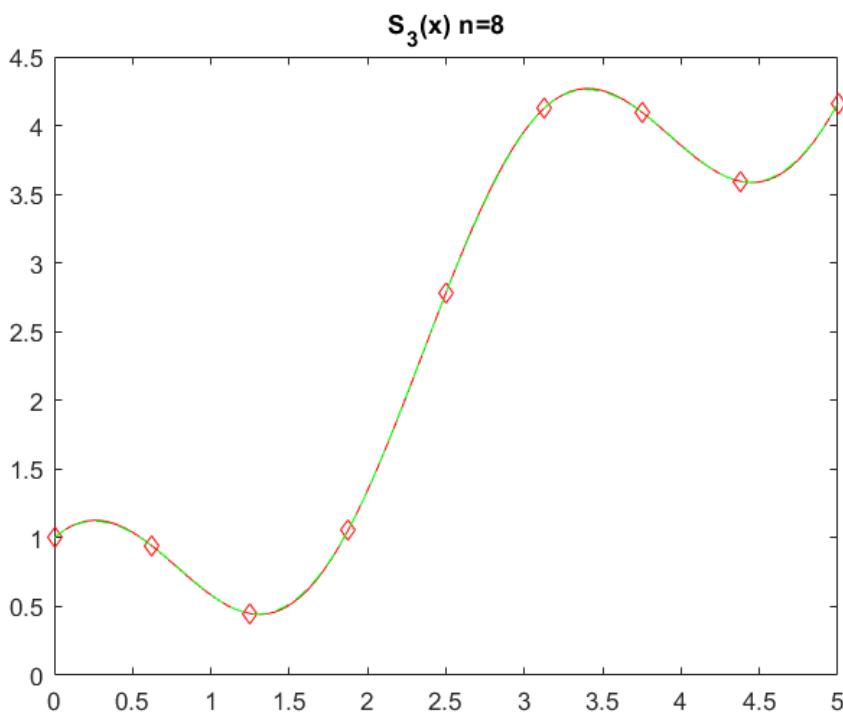
Rys. 5.3b. Wykres funkcji f i interpolacyjnych splajnów dla $n = 3$.



Rys. 5.3c. Wykres funkcji f i interpolacyjnych splajnów dla $n = 4$.



Rys. 5.3d. Wykres funkcji f i interpolacyjnych splajnów dla $n = 5$.



Rys. 5.3e. Wykres funkcji f i interpolacyjnych splajnów dla $n = 8$.

Błędy dla $n = 2$ są jak widać na rysunku bardzo duże, rzędu 1,822, dla $n = 3$ maksymalny błąd interpolacji równa się 0,273, dla $n = 4$ błąd wynosi 0,306- jest większy!, dla $n = 5$ jest 0,097, a dla $n = 8$ już tylko 0,008. Na rysunku funkcje się pokrywają. Obliczenia podaliśmy z dokładnością do trzech cyfr po przecinku. Powyższe wykresy oraz rozwiązanie zadania można uzyskać poniższym programem MATLAB. W poniższym programie zakładamy, że w tym samym folderze znajduje się dodatkowo funkcja MATLABa obliczająca wartość funkcji bazowej, przedstawiona w części 5.1.

```
close all
```

```
f = @(x) (x+cos(2*x))
df = @(x) (1-2*sin(2*x))

a = 0;
b = 5;
x = linspace(a,b,100);
y = f(x)

n=2;
xk = linspace(a,b,n+1);
yk = f(xk);

h = xk(2)-xk(1);

% potrzebujemy n+3 funkcji bazowych!
c = zeros(n+3,1);

alpha = df(a)
beta = df(b)

% budujemy uklad rownan
A = zeros(n+1);
b = zeros(n+1,1);

A(1,1) = 4;
A(1,2) = 2;
b(1) = yk(1) + h/3*alpha;

for i=2:n
    A(i,i-1)=1;
    A(i,i)=4;
    A(i,i+1)=1;
    b(i)=yk(i);
end

A(n+1,n+1) = 4;
A(n+1,n) = 2;
b(n+1) = yk(n+1) - h/3*beta;

% rozwiazujemy wspolczynniki i wstawiamy je od razu do wektora c
c(2:n+2) = A\b;

% c_{-1}
c(1) = c(3) - h/3*alpha;
% c_{n+1}
c(n+3) = c(n+1) + h/3*beta;

yS3 = zeros(length(x),1);
for j = 1:length(x)
    s = 0;
    for i = -1:n+1
        xi = a + (i)*h;
        s = s + c(i+2) * phi(xi, h, x(j));
    end
    yS3(j) = s;
end
plot(x,y,'r', xk, yk, 'dr', x, yS3, '--g')
title(sprintf('S_3(x) n=%d',n))

% wyswietlmy wspolczynniki c
c
```

6 Rozdział - Aproksymacja

Strona: [LeIA](#)

Kurs: Metody numeryczne (2025Z)

Książka: 6 Rozdział - Aproksymacja

Wydrukowane przez użytkownika: Kinga Kondraciuk

Data: niedziela, 30 listopada 2025, 14:15

Opis

Lekcja jest poświęcona aproksymacji dyskretnej, ograniczamy się jednak tylko do metody najmniejszych kwadratów. Choć głównie są omawiane wielomiany aproksymacyjne algebraiczne, to wzory są wyrowadzone dla dowolnej funkcji aproksymacyjnej, będącej kombinacją liniową zbioru funkcji bazowych.

Spis treści

- 1. Aproksymacja dyskretna**
- 2. Funkcje aproksymacyjne**
- 3. Aproksymacja wielomianami algebraicznymi**
- 4. Wielomiany trygonometryczne**
- 5. Błąd aproksymacji**
- 6. Aproksymacja ciągła**

1. Aproksymacja dyskretna

Aproksymacja

Aproksymacja, tak jak interpolacja, służy do znajdowania przybliżonych wartości funkcji $f(x)$ w dowolnym punkcie przedziału. Jednak funkcja aproksymacyjna na ogół jest inną niż funkcja interpolacyjna. W przypadku funkcji interpolacyjnej pokrywała się ona w pewnych punktach z funkcją interpolowaną, na funkcję aproksymacyjną nie będziemy narzucać takiego warunku. Będziemy od niej żądać aby była "bliska" funkcji aproksymowanej. Wyjaśnimy co będziemy uważały za "bliskość" dwóch funkcji. Ogólnie chodzi o to, aby wartości tych funkcji w pewnych wyróżnionych punktach były sobie bliskie. Założymy, podobnie jak w interpolacji, że z doświadczeń lub pomiarów określiliśmy w $n + 1$ różnych punktach:

$$x_0, x_1, x_2, \dots, x_n$$

z przedziału a, b wartości funkcji $y = f(x)$ i te wartości oznaczyliśmy przez:

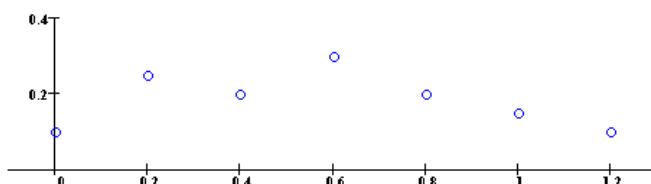
$$y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n)$$

Funkcję aproksymacyjną oznaczamy przez $F(x)$ i będziemy wymagać, aby kwadraty odległości między wartościami y_i a $F(x_i)$ w sumie były jak najmniejsze, tzn.: aby suma

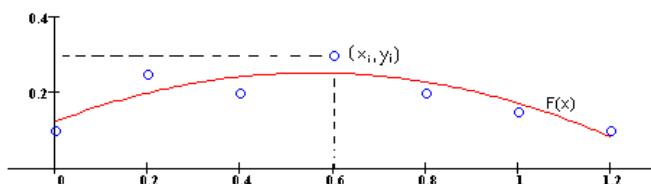
$$H = \sum_{i=0}^n (y_i - F(x_i))^2$$

była minimalna. Taka metoda aproksymacyjna nazywana jest metodą najmniejszych kwadratów. Na rysunku przedstawiona jest na czerwono funkcja aproksymacyjna $F(x)$ i są zaznaczone te odcinki (na czarno), których suma kwadratów długości ma być najmniejsza. Wartości funkcji $F(x_i)$ oznaczone są przez F_i , wartości (x_i, y_i) są zaznaczone kółkami.

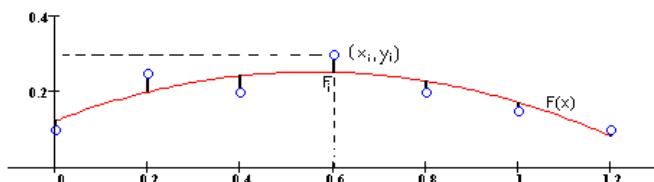
Na rysunku 6.1a) zaznaczone są węzły i wartości funkcji, na rysunku 6.1b) dochodzi jeszcze $F(x)$, na rysunku 6.1c) zaznaczone są dodatkowo odcinki- różnice między $f(x)$ i $F(x)$ w węzłach.



Rys. 6.1a. Wartości funkcji f w węzłach.



Rys. 6.1b. Wartości funkcji f w węzłach i funkcję aproksymacyjną.



Rys. 6.1c. Wartości funkcji f w węzłach i funkcja aproksymacyjna $F(x)$ wraz z zaznaczonymi błędami, czyli różnicami: $(y_i - F(x_i))$.

2. Funkcje aproksymacyjne

Funkcje aproksymacyjne

Założmy, że z doświadczeń lub pomiarów określiliśmy w $n + 1$ różnych punktach :

$$x_0, x_1, x_2, \dots, x_n$$

z przedziału a, b wartości funkcji $y = f(x)$ i te wartości oznaczyliśmy przez:

$$y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n)$$

Będziemy rozpatrywać funkcje aproksymacyjne w różnej postaci, w szczególności wielomiany algebraiczne i wielomiany trygonometryczne. Jeśli za funkcję aproksymacyjną będziemy brać wielomian m-tego stopnia, to ten wielomian zapisywać będziemy w postaci:

$$W_M(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_{m-1} x^{m-1} + a_m x^m \quad (6.2.1)$$

gdzie $a_i \quad i = 0, 1, 2, \dots, n$ to współczynniki rzeczywiste wielomianu, które trzeba znaleźć.

Jeśli za funkcję aproksymacyjną będziemy brać m-tą wielomian trygonometryczny to będzie to następująca funkcja:

$$T_m(x) = a_0 + a_1 \cos(c \cdot x) + b_1 \sin(c \cdot x) + a_2 \cos(2c \cdot x) + b_2 \sin(2c \cdot x) + \dots + a_m \cos(mc \cdot x) + b_m \sin(mc \cdot x) \quad (6.2.2)$$

W takim m-tym wielomianie występują cosinusy i sinusy wielokrotności kąta $c \cdot x$, współczynnik c jest znany, niewiadome są współczynniki $a_0, a_i, b_i \quad i = 1, 2, \dots, m$.

Ogólnie, jeśli funkcja aproksymacyjna oparta będzie na $m + 1$ znanych niezależnych liniowo funkcjach bazowych: $\phi_0(x), \phi_1(0), \dots, \phi_m(x)$ to będzie mieć postać: $F(x) = a_0 \phi_0(x) + a_1 \phi_1(x) + \dots + a_m \phi_m(x)$ gdzie $a_i \quad i = 0, 1, 2, \dots, m$ są szukanymi współczynnikami.

Metoda najmniejszych kwadratów polega zatem na znalezieniu współczynników przy funkcjach bazowych takich, aby funkcja określająca sumę kwadratów odchyлеń $S = \sum_{i=0}^n (y_i - F(x_i))^2$ była jak najmniejsza.

Oznaczmy przez:

$$\begin{aligned} H(a_0, a_1, \dots, a_m) &= \sum_{i=0}^n (y_i - F(x_i))^2 \\ &= \sum_{i=0}^n (y_i - (a_0 \phi_0(x_i) + a_1 \phi_1(x_i) + \dots + a_m \phi_m(x_i)))^2 \end{aligned} \quad (6.2.3)$$

Będziemy szukać minimum tej funkcji $m+1$ zmiennych $a_i \quad i = 0, 1, 2, \dots, m$. Punkty, w których funkcja wielu zmiennych przyjmuje minimum są punktami, w których zerują się pochodne cząstkowe (jeśli istnieją) tej funkcji po $a_i \quad i = 0, 1, 2, \dots, m$. Funkcja $H(a)$ jest wielomianem ze względu na niewiadome $a_i \quad i = 0, 1, 2, \dots, m$, więc te pochodne istnieją i są ciągłe. Otrzymujemy układ $m+1$ równań liniowych na $m+1$ niewiadomych

$$\frac{\partial H}{\partial a_j} = 0 \quad j = 0, 1, 2, \dots, m$$

Po obliczeniu pochodnych dostajemy:

$$\frac{\partial H}{\partial a_j} = 2 \sum_{i=0}^n (y_i - (a_0 \phi_0(x_i) + a_1 \phi_1(x_i) + \dots + a_m \phi_m(x_i))) \cdot (-\phi_j(x_i)) = 0$$

dla $j = 0, 1, 2, \dots, m$, tzn.: układ :

$$a_0 \sum_{i=0}^n \phi_0(x_i) \phi_j(x_i) + a_1 \sum_{i=0}^n \phi_1(x_i) \phi_j(x_i) + \dots + a_m \sum_{i=0}^n \phi_m(x_i) \phi_j(x_i) = \sum_{i=0}^n y_i \phi_j(x_i) \quad (6.2.4)$$

gdzie $j = 0, 1, 2, \dots, m$.

Rozpisując te równania otrzymujemy:

$$\begin{aligned}
 a_0 \sum_{i=0}^n \phi_0^2(x_i) + a_1 \sum_{i=0}^n \phi_1(x_i) \phi_0(x_i) + \dots + a_m \sum_{i=0}^n \phi_m(x_i) \phi_0(x_i) &= \sum_{i=0}^n y_i \phi_0(x_i) \\
 a_0 \sum_{i=0}^n \phi_0(x_i) \phi_1(x_i) + a_1 \sum_{i=0}^n \phi_1^2(x_i) + \dots + a_m \sum_{i=0}^n \phi_m(x_i) \phi_1(x_i) &= \sum_{i=0}^n y_i \phi_1(x_i) \\
 \dots \dots \dots \\
 a_0 \sum_{i=0}^n \phi_0(x_i) \phi_m(x_i) + a_1 \sum_{i=0}^n \phi_1(x_i) \phi_m(x_i) + \dots + a_m \sum_{i=0}^n \phi_m^2(x_i) &= \sum_{i=0}^n y_i \phi_m(x_i)
 \end{aligned} \tag{6.2.5}$$

Jeśli oznaczmy przez:

$$M = \begin{bmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \dots & \varphi_m(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \dots & \varphi_m(x_1) \\ \dots & \dots & \dots & \dots \\ \varphi_0(x_n) & \varphi_1(x_n) & \dots & \varphi_m(x_n) \end{bmatrix}, Y = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}, A = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} \tag{6.2.6}$$

to układ powyższy można zapisać w prostej formie macierzowej:

$$M^T M \cdot A = M^T \cdot Y \tag{6.2.7}$$

W metodzie najmniejszych kwadratów funkcja H jest tak skonstruowana, że posiada zawsze minimum (w najlepszym wypadku jest to zero), nie musimy za każdym razem sprawdzać warunków dostatecznych na istnienie ekstremum funkcji wielu zmiennych. Z liniowej niezależności funkcji bazowych wynika nieosobliwość macierzy układu równań liniowych (macierze mają różne od zera wyznaczniki). Wynika z tego, że rozwiązanie na współczynniki $a_i \quad i = 0, 1, 2, \dots, m$ istnieje, jest jedynie, zatem możemy znaleźć funkcję aproksymacyjną spełniającą narzucone warunki - minimalizacja sumy kwadratów różnic między funkcją daną a aproksymacyjną w wybranych punktach.

Uwagi:

1. W interpolacji ilość węzłów narzucała stopień wielomianu interpolacyjnego, w aproksymacji wielomianu może być stopnia 2, a ilość węzłów 100. Możemy sami sterować stopniem wielomianu. Jeśli natomiast węzłów będzie tyle ile funkcji bazowych to funkcja aproksymacyjna pokrywa się z funkcją interpolacyjną.
2. Macierz, powyżej zdefiniowana, ma tyle wierszy ile jest węzłów, a tyle kolumn ile jest funkcji bazowych. W pierwszej kolumnie jest zerowa funkcja bazowa we wszystkich węzłach, w drugiej kolumnie następna funkcja bazowa dla wszystkich węzłów po kolej, w ostatniej kolumnie jest ostatnia funkcja bazowa dla wszystkich węzłów.
3. Macierz $M^T M$ jest macierzą kwadratową wymiaru $(m+1) \times (m+1)$.

Przykład 6.1

Przykład

Dla $n+1=7$ punktów zmierzliśmy wartości funkcji $f(x)$ i otrzymaliśmy następujące wyniki (w tabelce):

x_i	0	0.2	0.4	0.6	0.8	1.0	1.2
y_i	0.1	0.25	0.2	0.3	0.2	0.15	0.1

Dla $n = 6$ oraz $i = 0, 1, \dots, n$. Szukamy funkcji aproksymacyjnej $F(x)$

$$F(x) = a_0 + a_1 \sin x + a_2 e^x$$

to znaczy, że bazą dla tej funkcji jest układ $\{1, \sin x, e^x\}$

Macierz M ma w pierwszej kolumnie same jedynki, w drugiej wartości funkcji $\sin x$ dla wszystkich x_i , a w trzeciej kolumnie wartości funkcji e^x dla x_i i wygląda następująco (wyniki podaliśmy z dokładnością do trzech cyfr po przecinku):

$$M = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0.199 & 1.221 \\ 1 & 0.389 & 1.492 \\ 1 & 0.565 & 1.822 \\ 1 & 0.717 & 2.226 \\ 1 & 0.841 & 2.718 \\ 1 & 0.932 & 3.32 \end{pmatrix}$$

Z układu równań $M^T M \cdot A = M^T \cdot Y$, w którym macierze $M^T M$ i $M^T Y$ są następujące:

$$M^T \cdot M = \begin{pmatrix} 7 & 3.644 & 13.799 \\ 3.644 & 2.601 & 8.831 \\ 13.799 & 8.831 & 31.403 \end{pmatrix} \quad M^T \cdot Y = \begin{pmatrix} 1.3 \\ 0.66 \\ 2.435 \end{pmatrix}$$

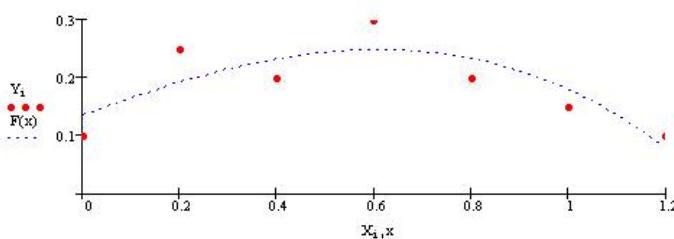
otrzymujemy współczynniki :

$$a = \begin{pmatrix} 0.39 \\ 0,572 \\ -0,255 \end{pmatrix}$$

Zatem szukana funkcja aproksymacyjna ma postać:

$$F(x) = 0,39 + 0,572 \sin x - 0,255 e^x$$

Wykres:



Rys. 6.2. Wykres funkcji aproksymacyjnej $F(x)$.

3. Aproksymacja wielomianami algebraicznymi

Aproksymacja wielomianami algebraicznymi

Załóżmy, że z doświadczeń lub pomiarów określiliśmy w $n+1$ różnych punktach :

$$x_0, x_1, x_2, \dots, x_n$$

z przedziału a, b wartości funkcji $y = f(x)$ i te wartości oznaczyliśmy przez:

$$y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n)$$

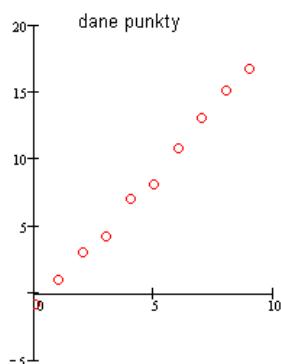
Będziemy rozpatrywać funkcje aproksymacyjne w postaci wielomianów algebraicznych:

$$W_m(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_{m-1} x^{m-1} + a_m x^m \quad (6.3.1)$$

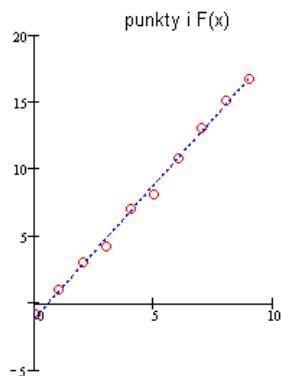
gdzie $a_i \quad i = 0, 1, 2, \dots, m$ to współczynniki rzeczywiste wielomianu, które trzeba znaleźć. Bazą takiego wielomianu są funkcje: $\{1, x, x^2, \dots, x^m\}$. Zbudujemy macierz M dla tej bazy:

$$M = \begin{bmatrix} 1 & x_0 & \dots & x_0^m \\ 1 & x_1 & \dots & x_1^m \\ \dots & \dots & \dots & \dots \\ 1 & x_n & \dots & x_n^m \end{bmatrix} \quad (6.3.2)$$

i aby znaleźć współczynniki $a_i \quad i = 0, 1, 2, \dots, m$, trzeba rozwiązać układ $M^T M \cdot A = M^T \cdot Y$ z powyższą macierzą. Ponieważ macierz układu $M^T M$ jest na ogół dla wysokich stopni wielomianów źle uwarunkowana (małe błędy danych powodują duże błędy wyników), stosuje się najczęściej aproksymację wielomianami niskich stopni tzn.: $m=1, 2$ lub 3 . Różne programy numeryczne liczą wskaźniki uwarunkowania macierzy i rozwiązuje układy równań liniowych. Prześledzimy tylko jeszcze raz powstawanie tego układu dla wielomianu pierwszego i drugiego stopnia. Założymy, że z doświadczeń dostaliśmy takie wartości badanej funkcji (x_i, y_i) , że punkty ułożyły się tak, jak na wykresie na rysunku 6.3. Na rysunku a) są tylko dane punkty, na rysunku b) również wielomian interpolacyjny stopnia 1- oznaczony jako $F(x)$.



Rys. 6.3a. Dane pomiarowe.



Rys. 6.3b. Dane i wykres funkcji liniowej- wielomianu aproksymacyjnego 1 stopnia.

Wtedy naturalnie jest stosować jako funkcję aproksymacyjną wielomian pierwszego stopnia czyli $F(x) = a_0 + a_1 x$.

Jeśli będziemy korzystać z powyższych gotowych wzorów to dostaniemy:

$$M = \begin{bmatrix} 1 & x_0 \\ 1 & x_1 \\ \dots & \dots \\ 1 & x_n \end{bmatrix} M^T = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_0 & x_1 & \dots & x_n \end{bmatrix} M^T M = \begin{bmatrix} n+1 & \sum_{i=0}^n x_i \\ n & n \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 \end{bmatrix} M^T Y = \begin{bmatrix} \sum_{i=0}^n y_i \\ \sum_{i=0}^n x_i y_i \end{bmatrix}$$

Zatem układ na współczynniki niewiadome a_0, a_1 będzie następujący:

$$\begin{aligned} a_0(n+1) + a_1 \sum_{i=0}^n x_i &= \sum_{i=0}^n y_i \\ a_0 \sum_{i=0}^n x_i + a_1 \sum_{i=0}^n x_i^2 &= \sum_{i=0}^n x_i y_i \end{aligned} \tag{6.3.3}$$

Ten sam układ otrzymamy wracając do funkcji :

$$H(a_0, a_1) = \sum_{i=0}^n (y_i - F(x_i))^2 = \sum_{i=0}^n (y_i - (a_0 + a_1 x_i))^2$$

i obliczając jej minimum.

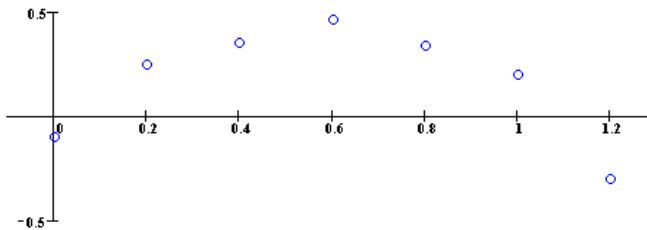
Obliczone pochodne cząstkowe przyrównamy do zera :

$$\begin{aligned} \frac{\partial H}{\partial a_0} &= 2 \sum_{i=0}^n (y_i - (a_0 + a_1 x_i))(-1) = 0 \\ \frac{\partial H}{\partial a_1} &= 2 \sum_{i=0}^n (y_i - (a_0 + a_1 x_i))(-x_i) = 0 \end{aligned} \tag{6.3.4}$$

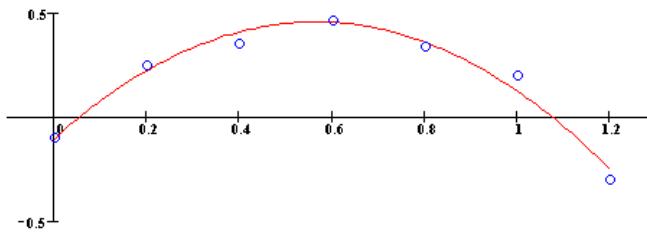
A stąd otrzymamy ten sam układ co podany powyżej.

Jeśli punkty pomiarowe ułożą się tak jak na rysunku poniżej, to nie ma co szukać funkcji aproksymacyjnej jako wielomianu stopnia 1, tylko co najmniej stopnia 2.

Na rysunku 6.4a są tylko dane punkty, na rysunku 6.4b również wielomian interpolacyjny stopnia 2.



Rys. 6.4a. Dane wejściowe - węzły aproksymacji.



Rys. 6.4b. Dane i wykres funkcji kwadratowej- wielomianu aproksymacyjnego 2 stopnia.

W tym wypadku za funkcję aproksymacyjną można przyjąć wielomian $F(x) = a_0 + a_1x + a_2x^2$. Wtedy

$$M = \begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ \dots & \dots & \dots \\ 1 & x_n & x_n^2 \end{bmatrix} M^T M = \begin{bmatrix} n+1 & \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i^3 \\ \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i^3 & \sum_{i=0}^n x_i^4 \end{bmatrix} M^T Y = \begin{bmatrix} \sum_{i=0}^n y_i \\ \sum_{i=0}^n x_i y_i \\ \sum_{i=0}^n x_i^2 y_i \end{bmatrix}$$

I układ na współczynniki a_0, a_1, a_2 jest następujący

$$\begin{aligned} a_0(n+1) + a_1 \sum_{i=0}^n x_i + a_2 \sum_{i=0}^n x_i^2 &= \sum_{i=0}^n y_i \\ a_0 \sum_{i=0}^n x_i + a_1 \sum_{i=0}^n x_i^2 + a_2 \sum_{i=0}^n x_i^3 &= \sum_{i=0}^n x_i y_i \\ a_0 \sum_{i=0}^n x_i^2 + a_1 \sum_{i=0}^n x_i^3 + a_2 \sum_{i=0}^n x_i^4 &= \sum_{i=0}^n x_i^2 y_i \end{aligned} \tag{6.3.5}$$

Taki sam układ otrzymamy, jeśli określmy funkcję:

$$H(a_0, a_1, a_2) = \sum_{i=0}^n (y_i - F(x_i))^2 = \sum_{i=0}^n (y_i - (a_0 + a_1 x_i + a_2 x_i^2))^2$$

obliczymy jej pochodne cząstkowe po i przyrównamy je do zera.

Przykład 6.2

Przykład

Funkcja $f(x)$ jest dana w 7 węzłach za pomocą tabelki

x_i	0	0.2	0.4	0.6	0.8	1.0	1.2
y_i	0.5	0.4	0.3	0.3	0.2	0.15	0.1

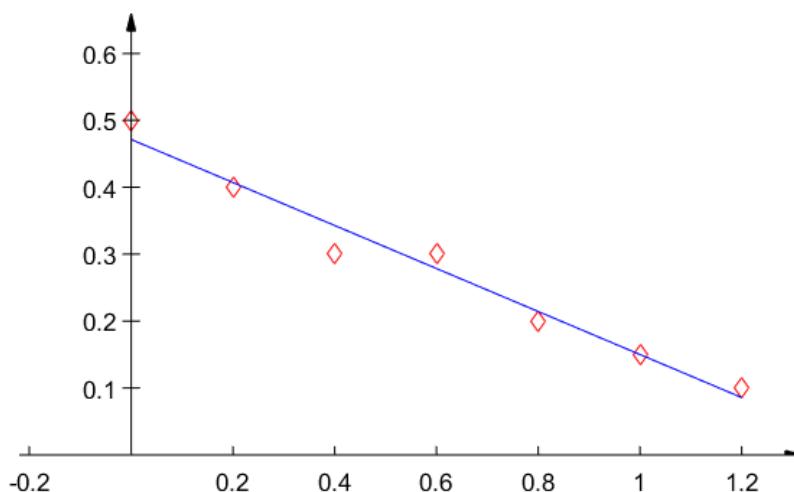
Będziemy szukać wielomianu aproksymacyjnego stopnia 1. Dla tych danych macierz M ma w pierwszej kolumnie jedynki, w drugiej węzły. Układ dwóch równań na współczynniki a jest bardzo prosty:

$$7a_0 + 4,2a_1 = 1,95$$

$$4,2a_0 + 3,64a_1 = 0,81$$

Rozwiążując ten układ dostajemy: $a_0 = 0,471$, $a_1 = -0,321$

Zatem wielomian aproksymacyjny pierwszego stopnia ma postać: $W_1(x) = 0,471 - 0,321x$



Rys. 6.5. Wykres funkcji $W_1(x)$.

Powyższy wykres został wygenerowany za pomocą poniższego:

```

xk = [ 0,0.2,0.4,0.6,0.8,1.0,1.2];
yঃ;yk = [ 0.5,0.4,0.3,0.3,0.2,0.15,0.1];

n = length(xk);
m = 1;

M = zeros(n,m+1)
M(:,1) = 1;
for i = 2:m+1
    M(:,i) = xk.^(i-1);
end

a = M'*M \ M'*yk'

xd = linspace(min(xk),max(xk),100);
yd = ones(n,1)*a(1)
for i = 2:m+1
    yd = yd + a(i)*xd.^(i-1);
end
plot(xk,yk,'dr',xd, yd,'b')
ylim([0,0.6])
xlim([-0.2,1.2])
%centeraxes(gca)

```

4. Wielomiany trygonometryczne

Załóżmy, że z doświadczeń lub pomiarów określiliśmy w $n + 1$ różnych punktach: $x_0, x_1, x_2, \dots, x_n$ z przedziału a, b wartości funkcji $y = f(x)$ i te wartości oznaczyliśmy przez:

$$y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n) \quad (1)$$

Będziemy rozpatrywać jako funkcje aproksymacyjne wielomiany trygonometryczne:

$$\begin{aligned} T_m(x) = & a_0 + a_1 \cos(cx) + b_1 \sin(cx) + a_2 \cos(2cx) + b_2 \sin(2cx) + \dots \\ & + \dots a_m \cos(mc \cdot x) + b_m \sin(mc \cdot x) \end{aligned} \quad (6.4.1)$$

W takim m-tym wielomianie występują cosinusy i sinusy wielokrotności kąta cx , współczynnik c jest znany, niewiadome są współczynniki a_0, a_i, b_i $i = 1, 2, \dots, m$. W praktyce, często przyjmuje się współczynnik $c = 1$ lub wielokrotności pewnej częstotliwości $f : c = 2\pi f$.

komentarz -

W celu uproszczenia obliczeń oraz ostatecznych wzorów ograniczymy się do węzłów równoodległych, podzielimy przedział a, b na n części, otrzymamy podprzedziały o długości $h = \frac{b-a}{n}$ i węzły $x_i = i \cdot h$, $i = 0, 1, \dots, n$. Najczęściej w obliczeniach przyjmuje się $a = 0$.

Ze względu na okresowość funkcji sin i cos przyjmujemy $c = \frac{\pi}{l}$, $l = \frac{n+1}{2}h$. Wtedy pierwszy wielomian trygonometryczny ma postać:

$$T_1(x) = a_0 + a_1 \cos\left(\frac{\pi}{l}x\right) + b_1 \sin\left(\frac{\pi}{l}x\right)$$

gdzie współczynniki a_0, a_1, b_1 wyliczamy z zerowania się pochodnych cząstkowych po a_0, a_1, b_1 funkcji:

$$H(a_0, a_1, b_1) = \sum (y_i - (a_0 + a_1 \cos\left(\frac{\pi}{l}x_i\right) + b_1 \sin\left(\frac{\pi}{l}x_i\right)))^2 \quad (6.4.2)$$

lub (co ostatecznie będzie równoważne) budujemy macierz M :

$$\begin{aligned} M &= \begin{bmatrix} 1 & \cos\left(\frac{\pi}{l}x_0\right) & \sin\left(\frac{\pi}{l}x_0\right) \\ 1 & \cos\left(\frac{\pi}{l}x_1\right) & \sin\left(\frac{\pi}{l}x_1\right) \\ \dots & \dots & \dots \\ 1 & \cos\left(\frac{\pi}{l}x_n\right) & \sin\left(\frac{\pi}{l}x_n\right) \end{bmatrix} \\ M^T M &= \begin{bmatrix} n+1 & 0 & 0 \\ 0 & \frac{n+1}{2} & 0 \\ 0 & 0 & \frac{n+1}{2} \end{bmatrix} \\ M^T Y &= \begin{bmatrix} \sum_{i=0}^n y_i \\ \sum_{i=0}^n y_i \cos\left(\frac{\pi}{l}x_i\right) \\ \sum_{i=0}^n y_i \sin\left(\frac{\pi}{l}x_i\right) \end{bmatrix} \end{aligned}$$

W tym wypadku macierz układu $M^T M$ jest dobrze uwarunkowana, jest macierzą diagonalną i układ ma bardzo prostą postać:

$$\begin{aligned}
 (n+1)a_0 &= \sum_{i=0}^n y_i, \\
 \frac{n+1}{2}a_1 &= \sum_{i=0}^n y_i \cos\left(\frac{\pi}{l}x_i\right), \\
 \frac{n+1}{2}b_1 &= \sum_{i=0}^n y_i \sin\left(\frac{\pi}{l}x_i\right)
 \end{aligned} \tag{6.4.3}$$

Możemy podać wzory na współczynniki a_0, a_1, b_1 :

$$\begin{aligned}
 a_0 &= \frac{1}{n+1} \sum_{i=0}^n y_i, \\
 a_1 &= \frac{2}{n+1} \sum_{i=0}^n y_i \cos\left(\frac{\pi}{l}x_i\right), \\
 b_1 &= \frac{2}{n+1} \sum_{i=0}^n y_i \sin\left(\frac{\pi}{l}x_i\right)
 \end{aligned} \tag{6.4.4}$$

Dla drugiego wielomianu trygonometrycznego:

$$T_2(x) = a_0 + a_1 \cos\left(\frac{\pi}{l}x\right) + b_1 \sin\left(\frac{\pi}{l}x\right) + a_2 \cos\left(2\frac{\pi}{l}x\right) + b_2 \sin\left(2\frac{\pi}{l}x\right)$$

współczynniki można wyliczyć analogicznie do powyższych, dostajemy wtedy:

$$\begin{aligned}
 a_0 &= \frac{1}{n+1} \sum_{i=0}^n y_i, \\
 a_1 &= \frac{2}{n+1} \sum_{i=0}^n y_i \cos\left(\frac{\pi}{l}x_i\right), \\
 b_1 &= \frac{2}{n+1} \sum_{i=0}^n y_i \sin\left(\frac{\pi}{l}x_i\right) \\
 a_2 &= \frac{2}{n+1} \sum_{i=0}^n y_i \cos\left(2\frac{\pi}{l}x_i\right), \\
 b_2 &= \frac{2}{n+1} \sum_{i=0}^n y_i \sin\left(2\frac{\pi}{l}x_i\right)
 \end{aligned} \tag{6.4.5}$$

Można uogólnić to postępowanie dla wielomianów trygonometrycznych zawierających wyrazy z \cos i \sin kąta m-krotnego.

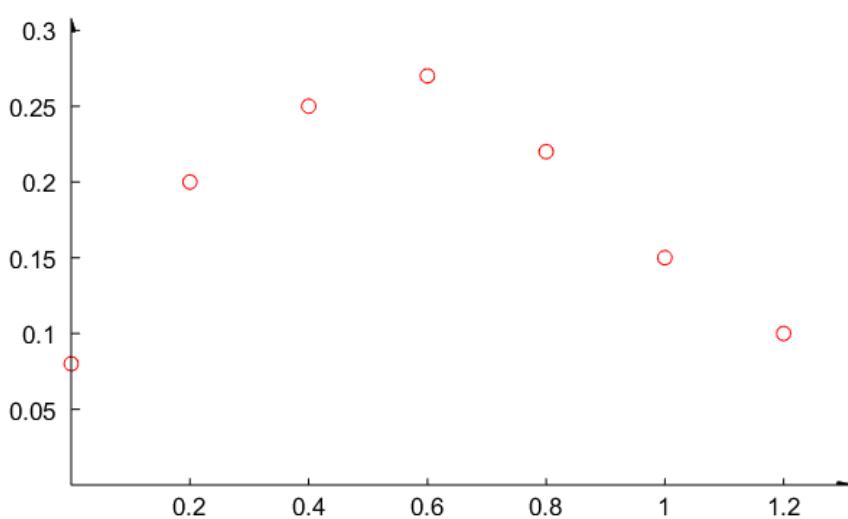
Przykład 7.1

Przykład

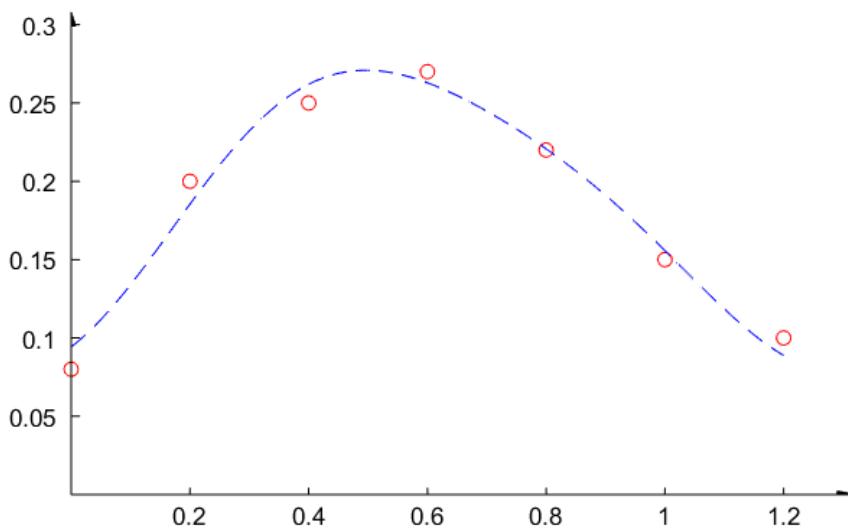
Funkcja jest dana za pomocą tabelki:

xi	yi
0	0,08
0,2	0,2
0,4	0,25
0,6	0,27
0,8	0,22
1,0	0,15
1,2	0,1

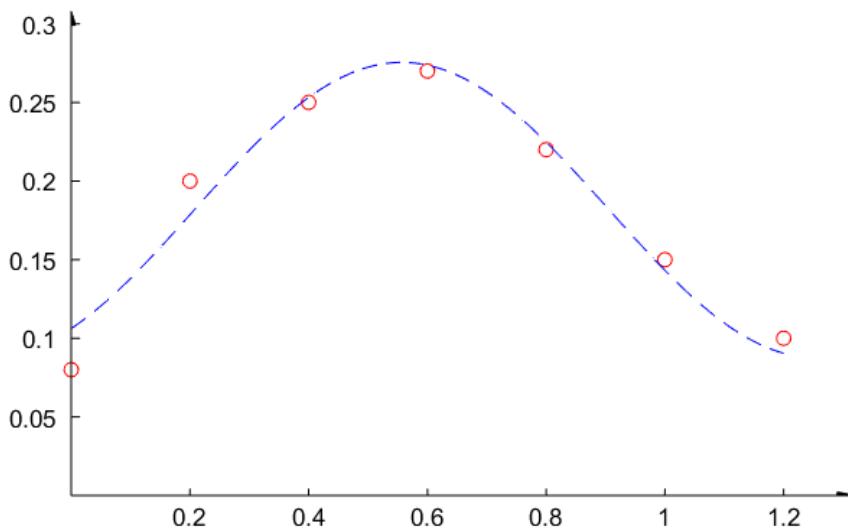
Dane: $n = 6, i = 0, 1, \dots, n, h = 0, 2, l = \frac{n+1}{2}h = 0,7$. Rysunki: 7.1a dane, 7.2b wielomian $T1x$, 7.1c wielomian $T2x$



Rys. 6.6a. Dane aproksymacyjne.



Rys. 6.6b. Dane i wielomian aproksymacyjny trygonometryczny T_1 .



Rys. 6.6c. Dane i wielomian aproksymacyjny trygonometryczny drugiego stopnia T_2 .

Po obliczeniu współczynników korzystając z powyższych wzorów otrzymujemy:

$$T_1(x) = 0.181 - 0.075 \cos\left(\frac{\pi}{l}x\right) + 0.056 \sin\left(\frac{\pi}{l}x\right) \quad (2)$$

$$\begin{aligned} T_2(x) = & 0.181 - 0.075 \cos\left(\frac{\pi}{l}x\right) + 0.056 \sin\left(\frac{\pi}{l}x\right) + \\ & - 0.019 \cos\left(2\frac{\pi}{l}x\right) + 0.004 \sin\left(2\frac{\pi}{l}x\right) \end{aligned} \quad (3)$$

Skrypt w MATLABie z rozwiązaniem zadania:

```
dane = [
0.0 0.08
0.2 0.2
0.4 0.25
0.6 0.27
0.8 0.22
1.0 0.15
1.2 0.1];
x = dane(:,1);
y = dane(:,2);

n=length(x)-1
h=0.2
l=(n+1)/(2)*h

a0 = 1/(n+1)*sum(y)
a1 = 2/(n+1)*sum(y .* cos(pi/l * x))
b1 = 2/(n+1)*sum(y .* sin(pi/l * x))
a2 = 2/(n+1)*sum(y .* cos(2*pi/l * x))
b2 = 2/(n+1)*sum(y .* sin(2*pi/l * x))

xd = 0:0.01:max(x);
yd = a0 + a1 * cos(pi/l * xd) + b1 * sin(pi/l * xd)

close all
plot(x,y,'or')
centeraxes(gca);

figure
plot(x,y,'or', xd,yd,'--b')
centeraxes(gca);

figure
yd = a0 + a1 * cos(pi/l * xd) + b1 * sin(pi/l * xd) ...
+ a2 * cos(2 * pi/l * xd) + b2 * sin(2 * pi/l * xd)
plot(x,y,'or', xd,yd,'--b')
centeraxes(gca);
```

5. Błąd aproksymacji

Czym będziemy się kierować decydując się na tę, a nie inną funkcję aproksymacyjną? Ponieważ chcemy, aby suma kwadratów odchyleń między funkcją daną a funkcją aproksymacyjną w węzłach była jak najmniejsza, możemy przyjąć dla prostoty, że najlepsza będzie ta funkcja, dla której ta suma jest jak najmniejsza. Będziemy posługiwać się wzorem na średni błąd przypadający na jeden węzeł, to znaczy:

$$bl = \sqrt{\frac{\sum_{i=0}^n (y_i - F(x_i))^2}{n+1}} \quad (6.5.1)$$

We wzorze pod pierwiastkiem w liczniku jest suma kwadratów odchyleń, którą minimalizowaliśmy, w mianowniku jest ilość węzłów.

Jeśli będziemy szukać funkcji aproksymacyjnej spośród danych możliwych, wybierając tę dla której wartość powyższego błędu bl jest najmniejsza.

Przykład 7.2

Przykład

Funkcja jest dana za pomocą tabelki:

xi	y_i
0	0,08
0,2	0,2
0,4	0,25
0,6	0,27
0,8	0,22
1,0	0,15
1,2	0,1

Dane: $n = 6, i = 0, 1, \dots, n, h = 0, 2, l = \frac{n+1}{2}h = 0.7$.

W przykładzie tym błędy są odpowiednio równe:

dla podanego w poprzednim temacie wielomianu:

$$T_1(x) = 0.181 - 0.075 \cos\left(\frac{\pi}{l}x\right) + 0.056 \sin\left(\frac{\pi}{l}x\right)$$

$T_1(x)$ błąd średni wynosi $bl = 0.014$. Błąd ten wynika z obliczenia sumy:

$$bl = \sqrt{\frac{(0.08-T_1(0))^2+(0.2-T_1(0.2))^2+(0.25-T_1(0.4))^2+\dots+(0.1-T_1(1.2))^2}{7}}$$

a dla wielomianu $T_2(x)$

$$\begin{aligned} T_2(x) = & 0.181 - 0.075 \cos\left(\frac{\pi}{l}x\right) + 0.056 \sin\left(\frac{\pi}{l}x\right) + \\ & - 0.019 \cos\left(2\frac{\pi}{l}x\right) + 0.004 \sin\left(2\frac{\pi}{l}x\right) \end{aligned}$$

błąd średni równa się $bl = 0.01$.

Do ewentualnych dalszych obliczeń, modelu numerycznego, itp., wybierzemy z tych dwóch funkcji wielomian drugi $T_2(x)$, bo ma mniejszy średni błąd.

Można stosować inne kryteria doboru funkcji aproksymacyjnej, czasami stosuje się błąd średni statystyczny, ale nie będziemy komplikować rozważań i zostaniemy przy tym najprostszym wzorze na błąd.

6. Aproksymacja ciągła

Będziemy aproksymować funkcję ciągłą $f(x)$ w przedziale (a, b) funkcją $F(x)$ postaci: $F(x) = a_0\phi_0(x) + a_1\phi_1(x) + \dots + a_m\phi_m(x)$ gdzie a_i ($i=0, 1, 2, \dots, m$) są szukanymi współczynnikami. Jeśli założymy, że funkcje bazowe $\phi_i(x)$ ($i=0, 1, 2, \dots, m$) są w przedziale (a, b) całkowalne z kwadratem (tzn. istnieje skończona wartość całki $\int_a^b \phi_i(x)^2 dx$) to funkcję aproksymacyjną $F(x)$ będziemy poszukiwać taką, aby funkcja:

$$\begin{aligned} H(a_0, a_1, \dots, a_m) &= \int_a^b [f(x) - F(x)]^2 dx \\ &= \int_a^b [f(x) - (a_0\phi_0(x) + a_1\phi_1(x) + \dots + a_m\phi_m(x))]^2 dx \end{aligned} \quad (7.3.1)$$

miała jak najmniejszą wartość. Podobnie jak poprzednio, gdy funkcja $f(x)$ dana była tylko w skończonej ilości punktów, warunkiem koniecznym na minimum funkcji $H(a_0, a_1, \dots, a_m)$ jest zerowanie się pochodnych cząstkowych $\frac{\partial H}{\partial a_j} = 0$ ($j=0, 1, 2, \dots, m$). I tak jak poprzednio układ tych równań posiada jednoznaczne rozwiązanie na współczynniki a_i ($i=0, 1, 2, \dots, m$), a warunek konieczny w tym wypadku zapewnia (ze względu na postać funkcji $H(a_0, a_1, \dots, a_m)$) istnienie minimum.

Przykład 7.3

Przykład

Wyznaczyć wielomian aproksymacyjny pierwszego stopnia najlepiej aproksymujący funkcję $f(x) = \frac{1}{x}$ w przedziale $(1, 2)$. Funkcja aproksymacyjna ma postać $F(x) = a_0 + a_1 x$ gdzie dwa współczynniki znajdziemy z zerowania się pochodnych funkcji:

$$H(a_0, a_1) = \int_1^2 [f(x) - (a_0 + a_1 x)]^2 dx$$

po a_0 i po a_1 .

$$\begin{aligned} \frac{\partial H}{\partial a_0} &= -2 \int_1^2 [f(x) - (a_0 + a_1 x)] dx \\ &= -2 \int_1^2 \left(\frac{1}{x} - (a_0 + a_1 x) \right) dx \end{aligned}$$

Mnożąc obie strony obu równań przez $\frac{1}{2}$ oraz rozdzielając na sumy wyrażenia podcałkowe otrzymujemy:

$$\begin{aligned} -\int_1^2 2f(x) dx + \int_1^2 2(a_0 + a_1 x) dx &= 0 \\ -\int_1^2 2f(x) dx &= -\int_1^2 2(a_0 + a_1 x) dx \end{aligned}$$

Stąd ostatecznie otrzymujemy układ równań, w którym łatwo wyznaczamy wartości całek analitycznie:

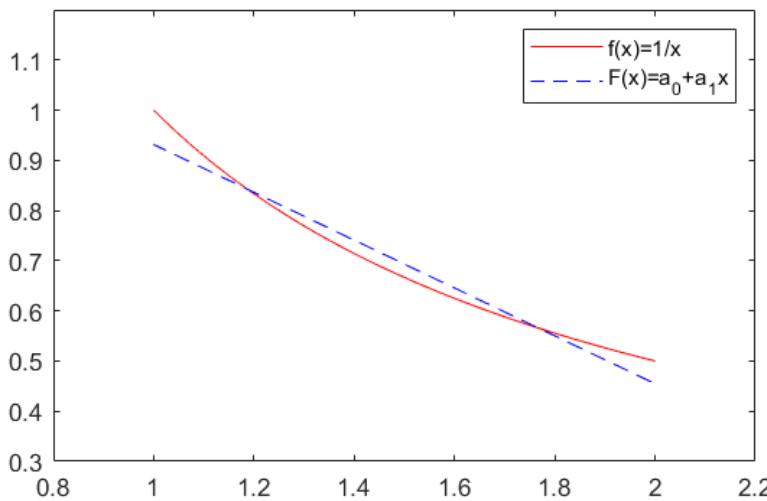
$$\begin{aligned} a_0 \int_1^2 \frac{1}{x} dx + a_1 \int_1^2 x dx &= \int_1^2 2f(x) dx \\ a_0 \int_1^2 \frac{1}{x} dx + a_1 \int_1^2 x dx &= \int_1^2 2f(x) dx \end{aligned} \quad (6.6.1)$$

gdzie całki zostały wyznaczone analitycznie:

$$\begin{aligned} \int_1^2 \frac{1}{x} dx &= \ln 2 \\ \int_1^2 x dx &= \frac{1}{2}x^2 \Big|_1^2 = \frac{3}{2} \end{aligned}$$

Rozwiązaniem tego układu są liczby: $a_0 = 1.408$, $a_1 = -0.477$, zatem szukana funkcja aproksymacyjna ma postać:

$$F(x) = 1.408 - 0.477x$$



Rys. 6.7. Wykres funkcji $f(x)$ i funkcji aproksymacyjnej $F(x)$.

Implementacja rozwiązania przykładu w MATLABie znajduje się poniżej. Na uwagę zasługuje zastąpienie całki analitycznej numeryczną obliczoną metodą prostokątów (które wyjaśnimy w kolejnych rozdziałach). W tym celu generujemy wektor $\langle x \rangle$ zawierający zbiór 1001 równoodległych punktów w przedziale $\langle 1, 2 \rangle$. Dla każdego punktu obliczamy wartość funkcji podcałkowej i mnożymy przez długość h (w przykładowej implementacji $\langle h=0.001 \rangle$). Zatem, możemy zapisać:

$$\langle \begin{aligned} \int_1^2 1 dx &= 2-1 \int_1^2 x dx \approx h \sum_{i=0}^{n-1} x_i \cdot h \int_1^2 x^2 dx \approx h \sum_{i=0}^{n-1} x_i^2 \end{aligned} \rangle$$

zatem powstaje nam układ równań $\langle Aa=b \rangle$ o postaci:

$$\langle \begin{bmatrix} 2-1 & \sum_{i=0}^{n-1} x_i^1 \\ a_0 & a_1 \end{bmatrix} = \begin{bmatrix} h \sum_{i=0}^{n-1} x_i^1 & h \sum_{i=0}^{n-1} x_i^2 \\ h \sum_{i=0}^{n-1} \frac{1}{x_i} & h \sum_{i=0}^{n-1} x_i^0 \end{bmatrix} \rangle$$

Należy zauważyć, że dla powyższych funkcji bazowych całki zostały wyznaczone bardzo łatwo analitycznie. Niemniej w ogólnym przypadku, gdy funkcje bazowe są bardziej złożone, to metoda wyznaczania całki numerycznie może okazać się bardziej praktyczna.

Powyższy przykład został zaimplementowany w MATLABie.

```

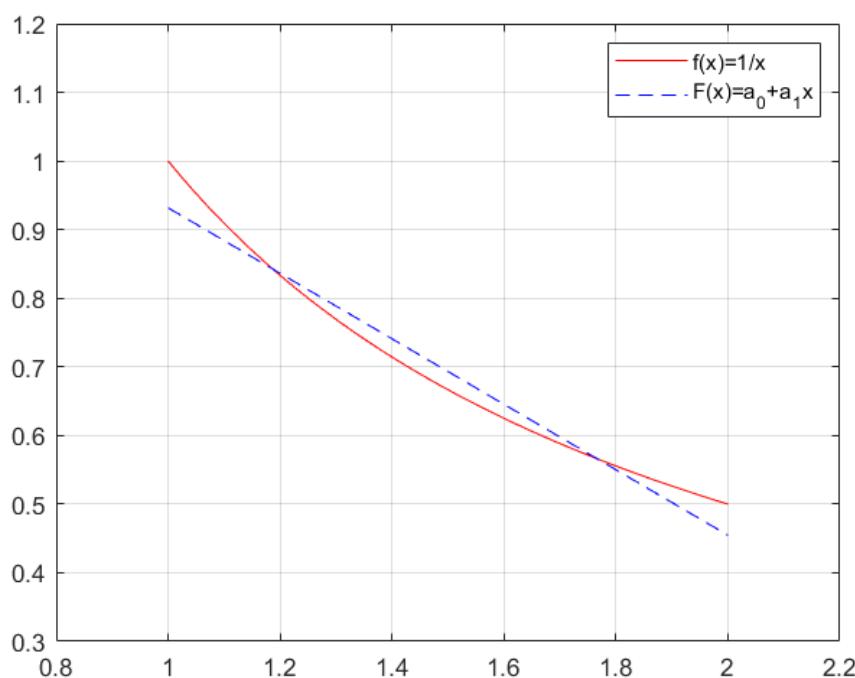
f = @(x) (1./x);
h = 0.001;
a=1;
b=2;
x = a:h:b;
n = length(x)
A = [
    b-a sum(x(1:end-1).*1)*h
    sum(x(1:end-1).*1)*h sum(x(1:end-1).^2)*h
];
b = [
    sum(1./x(1:end-1))*h
    sum(x(1:end-1).^0*1)*h
];
a = A\b
a_0 = a(1)
a_1 = a(2)
F = @(x) (a_0 + a_1*x);
close all
plot(x,f(x),'-r', x, F(x), 'b--')
xlim([0.8,2.2])
legend('f(x)=1/x', 'F(x)=a_0+a_1x')
ylim([0.3,1.2])

```

W wyniku uruchomienia powyższego skryptu otrzymamy bardzo podobny wynik do analitycznego:

```
a_0 =  
1.4086  
a_1 =  
-0.4770
```

oraz wykres



Rys. 6.8. Wykres funkcji $f(x)$ i funkcji aproksymacyjnej $F(x)$ uzyskane za pomocą całek obliczonych numerycznie.

7 Rozdział

Strona: [LeIA](#)

Kurs: Metody numeryczne (2025Z)

Książka: 7 Rozdział

Wydrukowane przez użytkownika: Kinga Kondraciuk

Data: niedziela, 30 listopada 2025, 14:15

Spis treści

- 1. Równania nieliniowe**
- 2. Wprowadzenie**
- 3. Izolacja pierwiastków**
- 4. Uwagi o dokładności**
- 5. Rząd metody**
- 6. Metoda bisekcji**
- 7. Metoda siecznych**
- 8. Metoda stycznych - Newtona**
- 9. Pierwiastki wielokrotne**
- 10. Układy równań nieliniowych**

1. Równania nieliniowe

Równanie nieliniowe np. równanie kwadratowe, logarytmiczne, wykładnicze, trygonometryczne, będziemy zapisywać ogólnie jako równanie postaci:

$$f(x) = 0 \quad (7.1.1)$$

gdzie funkcja f jest funkcją nieliniową zmiennej rzeczywistej x i jest funkcją ciągłą na pewnym przedziale skończonym lub nieskończonym (a, b) . Zwróćmy szczególną uwagę na zero z prawej strony równania, które jest istotne z punktu widzenia większości istniejących algorytmów. Z uwagi na tą postać (zero z prawej strony) często mówi się, że poszukujemy **zer** równania, czyli pierwiastków, dla których wartość wyrażenia z lewej strony przyjmuje wartość zero. Jedną z możliwych metod rozwiązywania równań nieliniowych jest metoda graficzna, polegająca na narysowaniu przebiegu funkcji z lewej strony równania i odczytanie wartości x dla których przebieg przecina oś OX .

2. Wprowadzenie

Poznane metody analityczne rozwiązywania równań nieliniowych dotyczą pewnej wąskiej klasy funkcji $f(x)$, np. wielomianów stopnia nie większego niż czwarty, natomiast olbrzymiej klasy równań nieliniowych - głównie równań przestępnych - nie da się rozwiązać dokładnie. Potrzebne są metody przybliżone, które umożliwiają znalezienie pierwiastków rzeczywistych tych równań z góry podaną dokładnością. Niektóre z tych metod, oraz problemy z nimi związane, będą przedstawione w tym rozdziale.

Definicja

Liczبę rzeczywistą p , która spełnia równanie $f(x) = 0$ tzn. dla której $f(p) = 0$, nazywamy pierwiastkiem rzeczywistym równania lub zerem funkcji f .

Definicja

Liczبę rzeczywistą p nazywamy k - krotnym pierwiastkiem równania $f(x) = 0$ lub k -krotnym miejscem zerowym funkcji f , jeśli dla wartości p funkcja i jej pochodne do $k-1$ rzędu włącznie przyjmują wartość zero, natomiast wartość pochodnej k -tego rzędu jest różna od zera , tzn:

$$f(p) \equiv 0, \quad f'(p) \equiv 0, \quad \dots \quad f^{(k-1)}(p) \equiv 0, \quad f^{(k)}(p) \neq 0$$

Przedstawione poniżej przybliżone metody rozwiązywania równań nieliniowych, można stosować jedynie pod warunkiem, że znany jest pewien przedział, w którym znajduje się jeden i tylko jeden pierwiastek rzeczywisty danego równania. Taki przedział będziemy nazywać *przedziałem izolacji* dla równania $f(x) = 0$. Przedziały izolacji, przed przystąpieniem do rozwiązywania równań, będziemy wyznaczać graficznie.

Wybrane przybliżone metody rozwiązywania równań nieliniowych są metodami iteracyjnymi, polegającymi na budowaniu ciągu przybliżeń liczb rzeczywistych:

$$x_0, x_1, x_2, \dots, x_n, \dots$$

zbieżnego do szukanego rozwiązania - pierwiastka p - równania $f(x) = 0$. Oczywiście, aby ciąg $\{x_n\}$ był zbieżny do pierwiastka p niezbędne są na ogół dodatkowe, oprócz ciągłości, założenia o funkcji f , które będą podane przy każdej metodzie osobno.

Ograniczymy się do metod jednokrokowych, polegających na znalezieniu następnego przybliżenia x_{n+1} , mając dane jego poprzednie przybliżenie x_n , oraz do metod dwukrokowych, polegających na znalezieniu następnego przybliżenia x_{n+1} , mając obliczone dwa poprzednie x_{n-1} i x_n . Formuły określające ciągi przybliżeń można zapisać ogólnie w postaci:

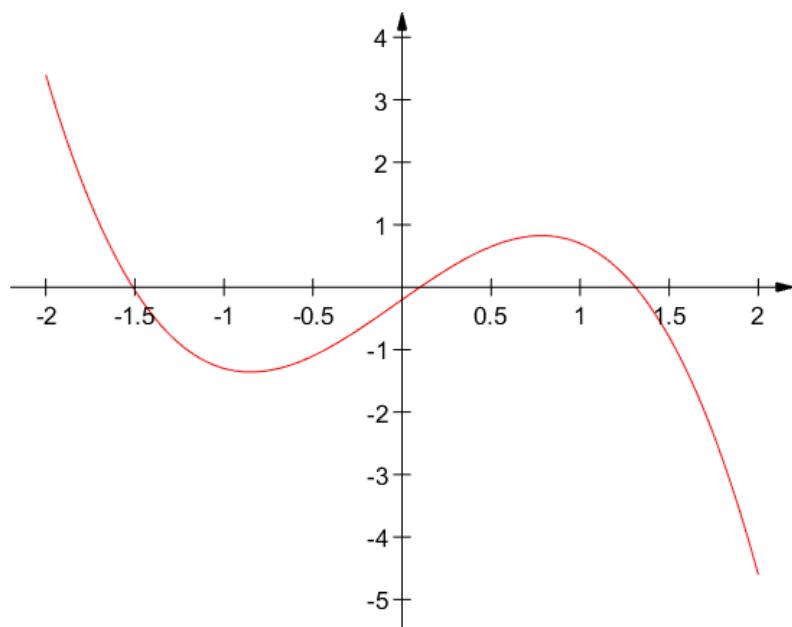
- $x_{n+1} = F(x_n)$ - dla metod jednokrotnych,
- $x_{n+1} = F(x_{n-1}, x_n)$ - dla metod dwukrotnych.

Będziemy korzystać z następujących twierdzeń:

Twierdzenie

Jeżeli funkcja f ciągła w przedziale $[a, b]$ ma na końcach tego przedziału różne znaki tzn. $f(a)f(b) < 0$, to wewnątrz tego przedziału istnieje co najmniej jeden pierwiastek równania $f(x) = 0$.

Ilustracja twierdzenia: Na rysunku funkcja jest ciągła o postaci $f(x) = -0.2 + 2x - 0.1x^2 - x^3$ i $f(b) < 0$, $f(a) > 0$. Funkcja przecina osią OX trzy razy, ma zatem trzy pierwiastki rzeczywiste w przedziale a, b .

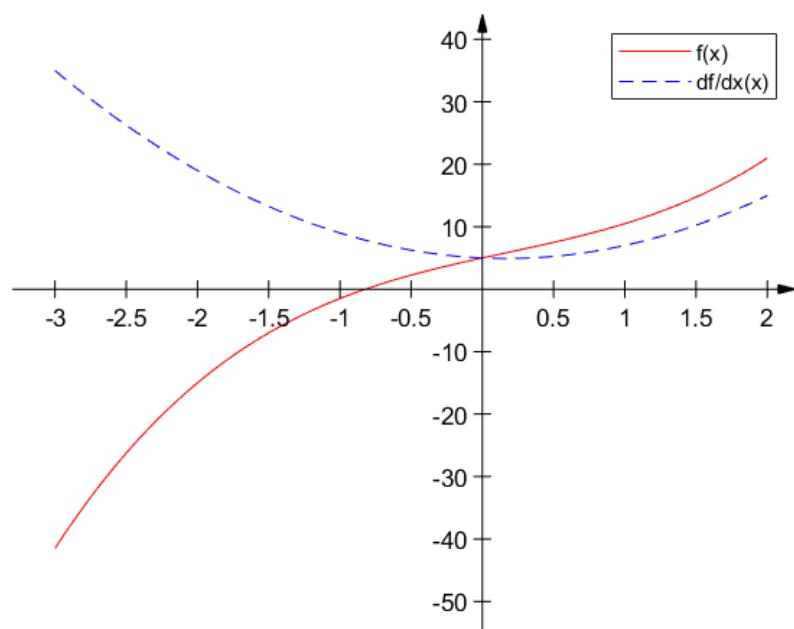


Rys 7.1. - Wykres funkcji ciągłej f zmieniającej znak trzy razy w przedziale $[-2, 2]$.

Twierdzenie

Jeżeli w przedziale (a, b) istnieje pochodna $f'(x)$ i nie zmienia znaku w tym przedziale tzn. albo jest w nim cała czas dodatnia albo ujemna, a $f(a)f(b) < 0$, to równanie $f(x) = 0$ ma dokładnie jeden pierwiastek jednokrotny.

Ilustracja twierdzenia: Na rysunku funkcja $f(x) = 5 + 5x - 0.5x^2 + x^3$ jest ciągła, ma ciągłą pochodną $f'(x) = 5 - x + 3x^2$, która jest cała czas dodatnia w a, b , tzn.: funkcja w a, b rośnie, oraz $f(a)f(b) < 0$, raz tylko wykres funkcji przecina się z osią OX .



Rys. 7.2. Wykres funkcji ciągłej f (linia czerwona), zmieniającej znak raz w przedziale $[-3, 2]$, oraz jej pochodnej $f'(x)$, która w całym przedziale jest dodatnia (przerywana linia niebieska).

3. Izolacja pierwiastków

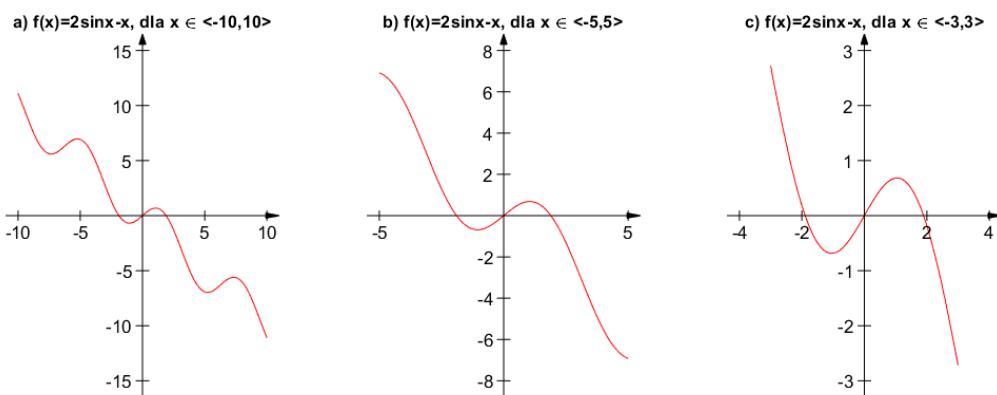
Najprostszą metodą przekonania się czy równanie $f(x) = 0$ posiada pierwiastki rzeczywiste jest narysowanie funkcji f i osi OX , i przekonanie się czy funkcja przecina się z osią. Można tym sposobem znaleźć przedziały, które się nie przecinają, a w których znajduje się po jednym pierwiastku danego równania. Zilustrujemy to na przykładzie.

Przykład 7.1

Przykład

Znaleźć przedziały izolacji równania: $2 \sin(x) - x = 0$

Rysujemy w kartezjańskim układzie współrzędnych funkcję f na możliwie dużym przedziale, a potem tak zauważamy ten przedział, aby nie stracić pierwiastków - aby wszystkie nadal były na wykresie. Rysunek 8.3a) przedstawia funkcję w przedziale $<-10, 10>$, rysunek 8.3b) w przedziale $<-5, 5>$, a rysunek 8.3c) w przedziale $<-3, 3>$.



Rys. 7.3. Izolacja pierwiastków równania $f(x) = 0$, funkcja $f(x)$.

Jak widać na rysunku funkcja $f(x)$ dąży do nieskończoności, gdy argumenty dążą do minus nieskończoności i dąży do minus nieskończoności, gdy argumenty dążą do nieskończoności. Można przyjąć, że wszystkie pierwiastki danego równania tzn. wszystkie punkty przecięcia funkcji z osią Ox , są w przedziale $<-3, 3>$. Widzimy, że równanie ma pierwiastek $x = 0$, co łatwo sprawdzić, drugi pierwiastek w przedziale np. $(1, 5; 2, 5)$ i trzeci pierwiastek w przedziale np. $(-2, 5; -1)$. Zatem równanie ma trzy przedziały izolacji $(-2, 5; -1)$, $(-0, 5; 0, 5)$ i $(1, 2, 5)$. Oczywiście widać, że przedziały izolacji nie są wyznaczone jednoznacznie, można podać je w postaci: $(-2, 2; -1, 3)$, $(-0, 2; 0, 2)$ i $(1, 3; 2, 2)$. Więcej pierwiastków równanie nie posiada.

Inną metodą graficzną jest rysowanie równania $f(x) = 0$ za pomocą dwóch funkcji, jeśli równanie da się przedstawić w postaci różnicicy funkcji $g(x) - h(x) = 0$. Punkty przecięcia funkcji g i h , są pierwiastkami równania $f(x) = 0$.

Przykład 7.2

Przykład

Ponieważ rozpatrywane równanie można przedstawić jako równanie:

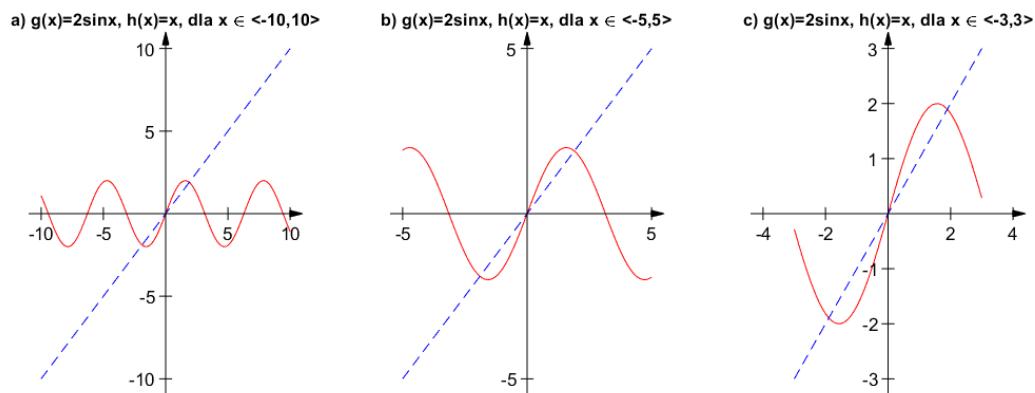
$$2 \sin(x) = x$$

to oznaczając przez $g(x) = 2 \sin(x)$, a przez $h(x) = x$, narysujemy te funkcje w takim samym przedziale jak poprzednio funkcję f .

Funkcje g i h przecinają się w zerze i dla tych samych x , w których funkcja f przecina się z osią OX . Funkcje te nie przecinają się w innym przedziale, bo funkcja g jest funkcją ograniczoną o wartościach w przedziale $[-2, 2]$, a funkcja h rośnie na prawo do nieskończoności, a na

lewo maleje do minus nieskończoności.

Rysunek 8.4a) przedstawia funkcje w przedziale $\langle -10, 10 \rangle$, rysunek 8.4b) w przedziale $\langle -5, 5 \rangle$, a rysunek 8.4c) w przedziale $\langle -3, 3 \rangle$.



Rys. 7.4. Izolacja pierwiastków równania $g(x) = h(x)$.

4. Uwagi o dokładności

Istotnym problemem w metodach iteracyjnych jest decyzja, którą iterację wziąć za przybliżenie pierwiastka równania $f(x) = 0$, co ma decydować o zakończeniu postępowania iteracyjnego (wyboru warunku "stopu"), lub jak się da oszacować przyjęte przybliżenie w stosunku do nieznanej dokładnej wartości pierwiastka. Do każdej metody będzie ten problem rozważany osobno, tutaj podamy jak można wykorzystać następujące twierdzenie:

Twierdzenie

Niech p będzie dokładną, a x^* przybliżoną wartością pierwiastka równania $f(x) = 0$, przy czym obie te liczby znajdują się w przedziale domkniętym $[a, b]$. Jeśli f posiada pochodną i jeśli dla x z przedziału $[a, b]$ zachodzi nierówność $|f'(x)| \geq m_1 \geq 0$ to prawdziwe jest oszacowanie :

$$|x^* - p| \leq \frac{|f(x^*)|}{m_1} \quad (1)$$

Dowód. Stosując wzór Lagrange'a otrzymujemy: $f(x^*) - f(p) = (x^* - p)f'(c)$ gdzie wartość c jest liczbą między p i x^* .

Ponieważ $f(p) = 0$ i $f'(c) \geq m_1$ to $|f(x^*) - f(p)| = |f(x^*)| \geq m_1|x^* - p|$ zatem $|x^* - p| \leq \frac{|f(x^*)|}{m_1}$.

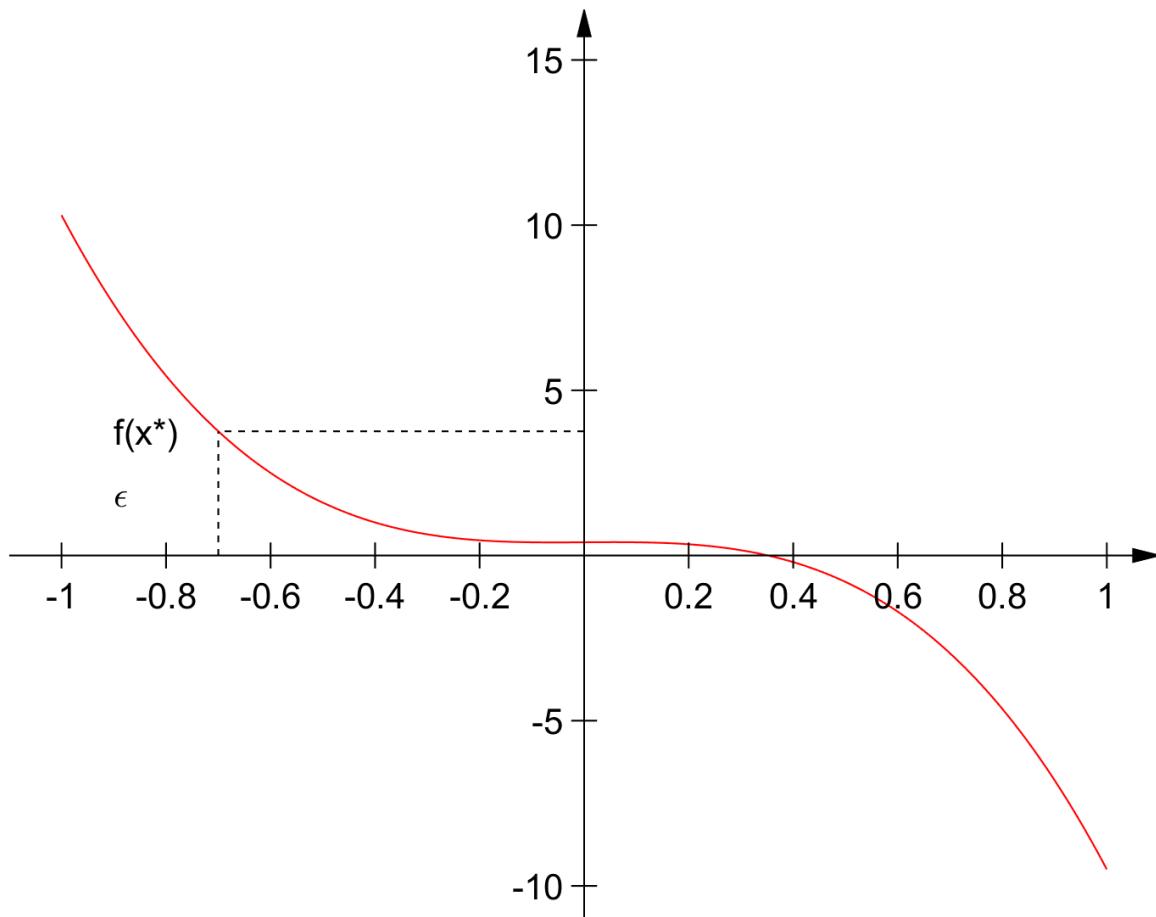
Przykład 7.3

Przykład

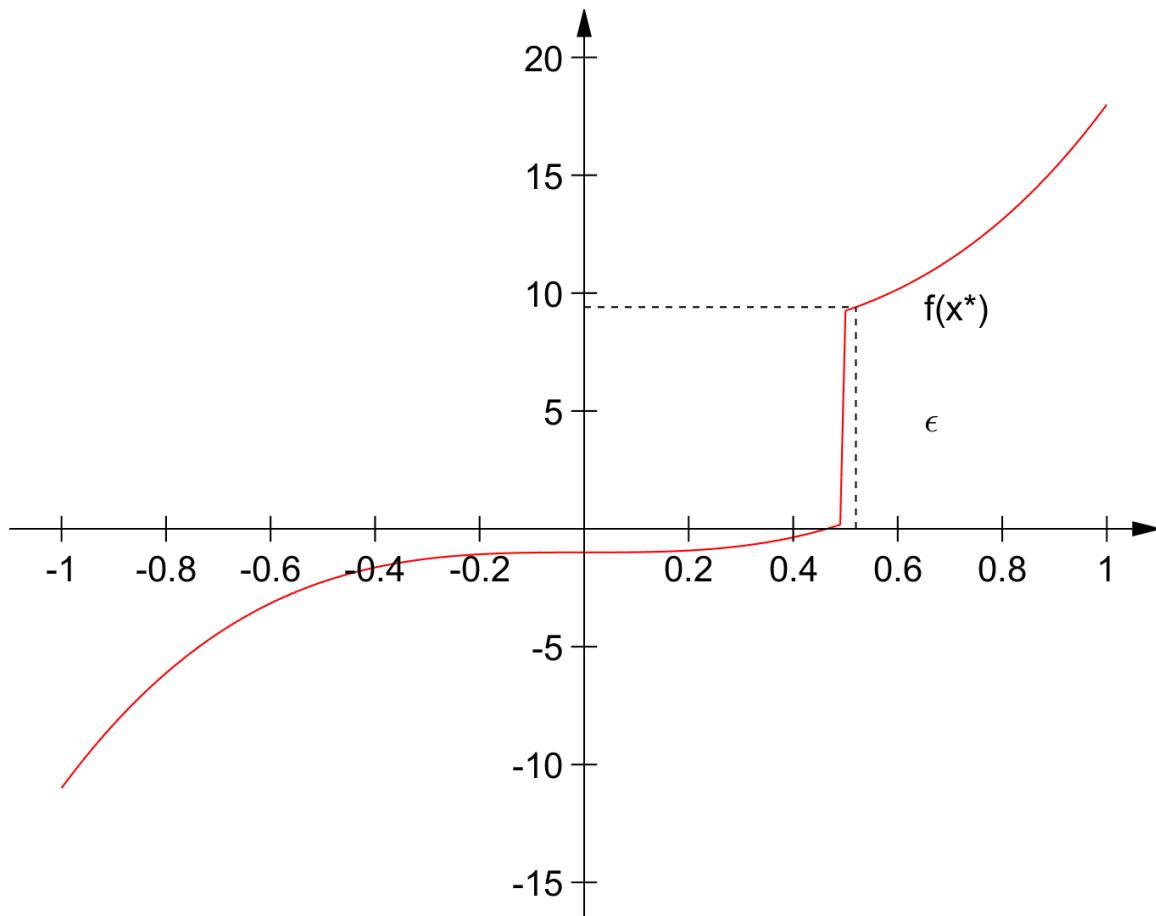
Rozpatrzymy równanie $x^4 - x - 1 = 0$. Weźmy za $x^* = 1.22$. Oszacujemy błąd bezwzględny tego przybliżenia. Mamy $f(x^*) = -0.0047$. Dla $x^* = 1.23$ wartość funkcji $f(x^*) = 0.0588$, dokładny pierwiastek p jest zatem w przedziale $(1, 22; 1.23)$. Pochodna $f'(x) = 4x^3 - 1$ jest w tym przedziale rosnąca, a najmniejszą wartość przyjmuje dla $x=1.22$, zatem $m_1 = 4 \cdot (1.22)^3 - 1 = 6.264$ więc $|x^* - p| \leq \frac{0.0047}{6.264} < 0.001$.

Uwaga

Niekiedy w praktyce ocenia się dokładność przybliżenia pierwiastka x^* według tego, czy liczba $|f(x^*)|$ jest mała, czy duża. Jeśli jest mała, to uważa się że x^* jest dobrym przybliżeniem dokładnej wartości pierwiastka p i na odwrót, jeśli $|f(x^*)|$ jest duże to x^* zostaje uznane za złe przybliżenie. Jak widać z następujących rysunków takie podejście nie zawsze jest prawidłowe. Nie należy również zapominać, że po pomnożeniu równania $f(x) = 0$ przez dowolną liczbę N różną od zera, otrzymujemy równanie równoważne, a liczbę $Nf(x^*)$ można uczynić dowolnie dużą lub dowolnie małą, dzięki doborowi N .



Rys. 7.5. Sytuacja, gdy x^* nie jest bliskie p .



Rys. 7.6. Sytuacja gdy $f(x^*)$ nie jest bliskie zeru.

W dalszych rozważaniach, aby zapobiec takim opisanym wyżej sytuacjom, będziemy zakładać brak punktów przegięcia funkcji f w przedziałach izolacji, tzn., tak będziemy dobierać (zawężać) przedział izolacji, aby druga pochodna funkcji opisującej lewą stronę równania miała stały znak w rozpatrywanym przedziale.

5. Rząd metody

Podstawowym warunkiem, jaki powinna spełniać dana metoda jest zbieżność ciągu iteracyjnego do pierwiastka równania. Oczywiście, tym lepsza jest metoda im szybciej ciąg przybliżeń jest zbieżny do p . Szybkość zbieżności można określić za pomocą dwóch wielkości:

- rzędu metody,
- i stałej asymptotycznej C błędu metody.

Definicja. Mówimy, że metoda jest rzędu r , jeśli istnieje granica :

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - p|}{|x_n - p|^r} = C \neq 0 \quad (7.4.1)$$

Liczba C nazywamy stałą asymptotyczną błędu metody.

Uwagi:

1. Jeśli $r = 1$ i $C < 1$ to x_n dąży do p dla dowolnego x_0 - punktu startowego.
2. Jeśli $r > 1$ i x_0 jest dostatecznie bliskie p to x_n zbiega do p .
3. Im większy jest rząd metody i im mniejsza jest stała asymptotyczna błędu, tym szybciej ciąg jest zbieżny do pierwiastka.

Zilustrujemy te uwagi na przykładzie.

Przykład 7.4

Przykład

Założymy, że metoda jest rzędu 2, ze stałą asymptotyczną równą 5. Czyli :

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - p|}{|x_n - p|^2} = 5$$

Dla każdego $\epsilon > 0$ istnieje takie N , że dla $n > N$ spełnione są nierówności:

$$5|x_n - p|^2 - \epsilon \leq |x_{n+1} - p| \leq 5|x_n - p|^2 + \epsilon$$

Oznacza to, że różnica pomiędzy $(n+1)$ -szym przybliżeniem x_{n+1} i dokładnym pierwiastkiem p może przyjmować, dla wystarczająco dużych n wartości dowolnie mało różniące się od $5|x_n - p|^2$

$$|x_{n+1} - p| \approx 5|x_n - p|^2$$

Przyjmijmy, że obliczyliśmy przybliżenie x_n pierwiastka, różniące się od niego o mniej niż 0,1 tzn: $|x_n - p| < 0.1$. Wówczas:

$$\begin{aligned} |x_{n+1} - p| &\cong 5 \cdot (0,1)^2 = 0.05 \\ |x_{n+2} - p| &\cong 5 \cdot (0.05)^2 = 0.0025 \\ |x_{n+3} - p| &\cong 5 \cdot (0.0025)^2 = 0.00003125 \end{aligned}$$

Widzimy więc, że różnice pomiędzy kolejnymi przybliżeniami i pierwiastkiem dokładnym szybko maleją. Założymy, że metoda jest rzędu 3, a stała asymptotyczna równa się 5. Zakładając, że $|x_n - p| < 0.1$ otrzymujemy teraz:

$$\begin{aligned} |x_{n+1} - p| &\cong 5 \cdot (0,1)^3 = 0.005 \\ |x_{n+2} - p| &\cong 5 \cdot (0.005)^3 = 0.000000625 \end{aligned}$$

Zatem metoda rzędu 3 jest wyraźnie szybsza od metody rzędu 2.

Gdyby wziąć metodę rzędu 2 ze stałą $C = 1$, to przy takiej samej przyjętej dokładności między n -tym przybliżeniem a pierwiastkiem: $|x_n - p| < 0.1$ dostajemy:

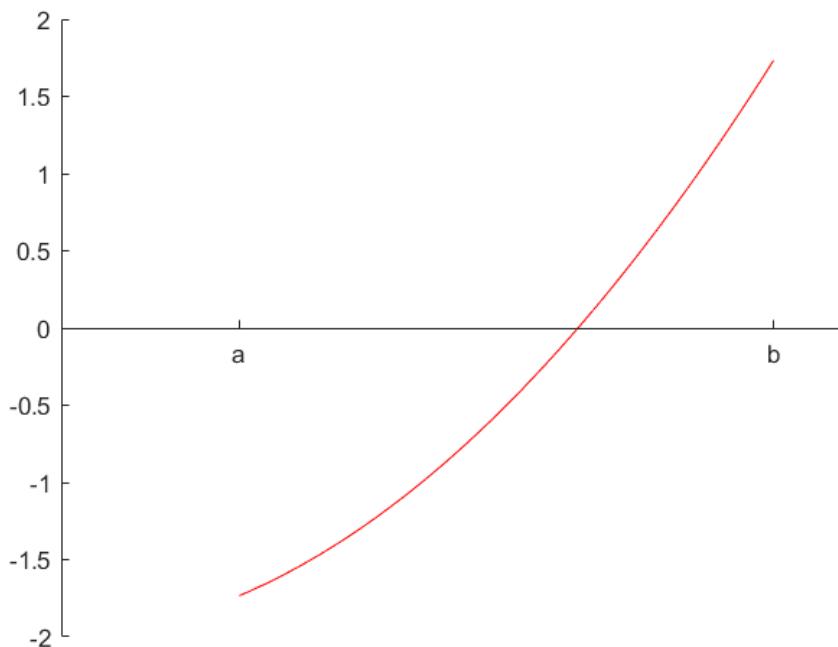
$$\begin{aligned} |x_{n+1} - p| &\cong 0.01 \\ |x_{n+2} - p| &\cong 0.0001 \\ |x_{n+3} - p| &\cong 0.0000001 \end{aligned}$$

Metoda rzędu 2 ze stałą 1 jest nieco szybsza od metody rzędu 2 ze stałą 5, ale wolniejsza od metody rzędu 3 ze stałą większą.

6. Metoda bisekcji

Omówimy teraz najprostsze metody znajdowania pierwiastków równania nieliniowego $f(x) = 0$. Podamy za każdym razem warunki wystarczające i konieczne, aby ciąg iteracyjny był zbieżny do szukanego pierwiastka. W każdej z omawianych metod rozpatrujemy przedział izolacji (a, b) w którym znajduje się aktualnie szukany pierwiastek, jeśli pierwiastków jest więcej - więcej jest przedziałów izolacji, metodę stosujemy po kolejno do każdego przedziału izolacji osobno.

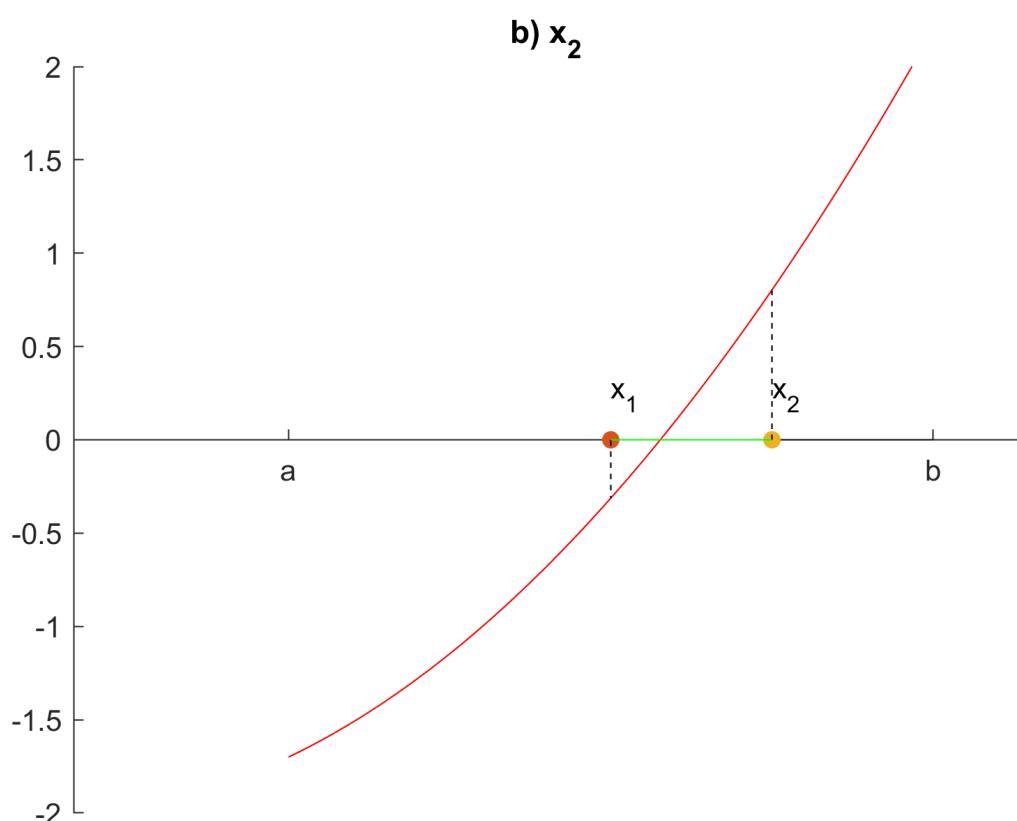
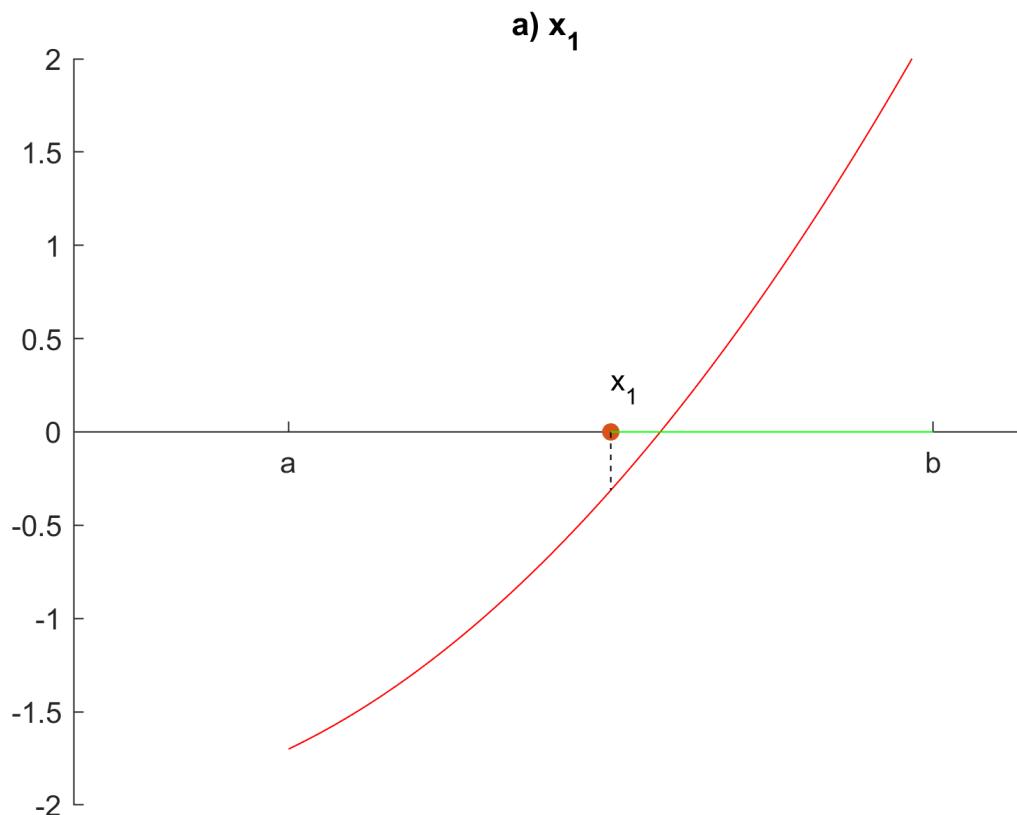
Metoda bisekcji. Założenia: Funkcja f jest funkcją ciągłą na $[a, b]$, oraz $f(a)f(b) < 0$.

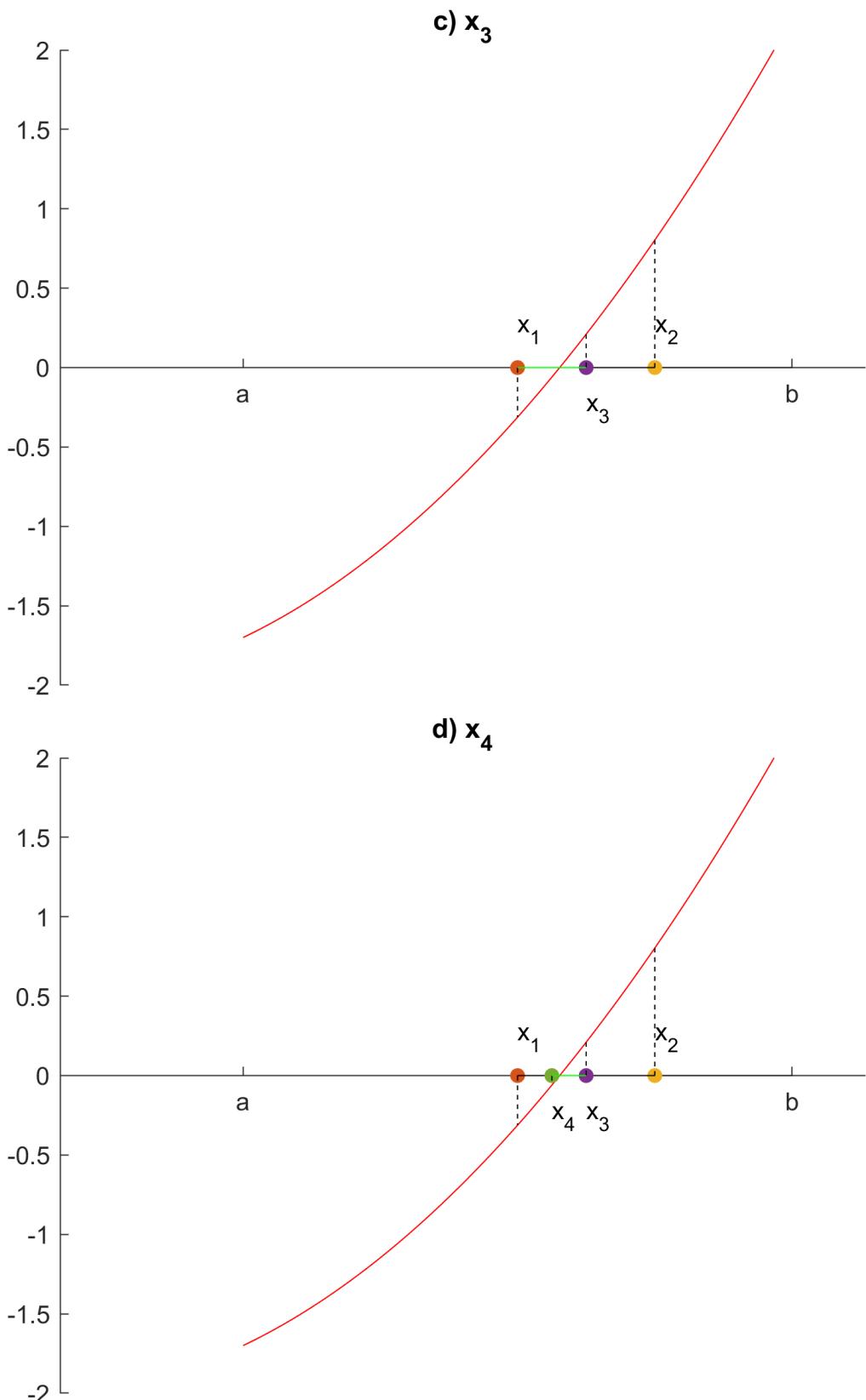


Rys. 7.7. W przedziale $[a, b]$ jest jeden pierwiastek rzeczywisty równania $f(x) = 0$.

Opis metody: Dzielimy przedział $[a, b]$ na połowy punktem: $x_1 = \frac{a+b}{2}$

- a) (Rys. 7.8a) Wybieramy przedział, w którym funkcja zmienia znak, czyli w naszym przypadku $[x_1, b]$,
- b) (Rys. 7.8b) Dzielimy go na połowę punktem x_2 i wybieramy znów przedział, w którym funkcja zmienia znak, w naszym przypadku $[x_1, x_2]$,
- c) (Rys. 7.8c) Dzielimy go na połowę punktem x_3 , wybieramy przedział $[x_1, x_3]$,
- d) (Rys. 7.8d) dzielimy znów na pół punktem x_4 wybieramy przedziałik $[x_4, x_3]$ itd.





Rys. 7.8. Kolejne cztery iteracje szukanego pierwiastka.

Jeśli $f(x_1) = 0$ to x_1 jest pierwiastkiem równania. Jeśli $f(x_1)$ jest różne od zera to z otrzymanych dwóch podprzedziałów $[a, x_1]$ i $[x_1, b]$ wybieramy ten, w którym funkcja f zmienia znak. Z kolei ten przedział dzielimy na połowy punktem x_2 i badamy wartość funkcji w x_2 oraz znaki w otrzymanych podprzedziałach, wybierając do dalszych obliczeń zawsze ten, w którym funkcja zmienia znak. Otrzymujemy albo po n krokach $f(x_n) = 0$ albo ciąg

podprzedziałów takich, że $f(x_n)f(x_{n+1}) < 0$ przy czym x_n, x_{n+1} są końcami przedziału, a jego długość $|x_n - x_{n+1}| < \frac{1}{2^n}(b - a)$.

Ponieważ, z konstrukcji, lewe końce przedziałów tworzą ciąg niemalejący i ograniczony z góry (przez p), a prawe końce przedziałów tworzą ciąg nieskończony i ograniczony dołu (przez p), to istnieje granica wspólna dla tych ciągów równa p .

Podstawową zaletą tej metody jest jej prostota i pewność, że w każdej kolejnej iteracji szukany pierwiastek leży między dwiema wartościami zmiennej x , dla których funkcja zmienia znak. Teoretycznie można uzyskać dowolną dokładność przy obliczeniach pierwiastka, stosować iterację tak dugo, aż

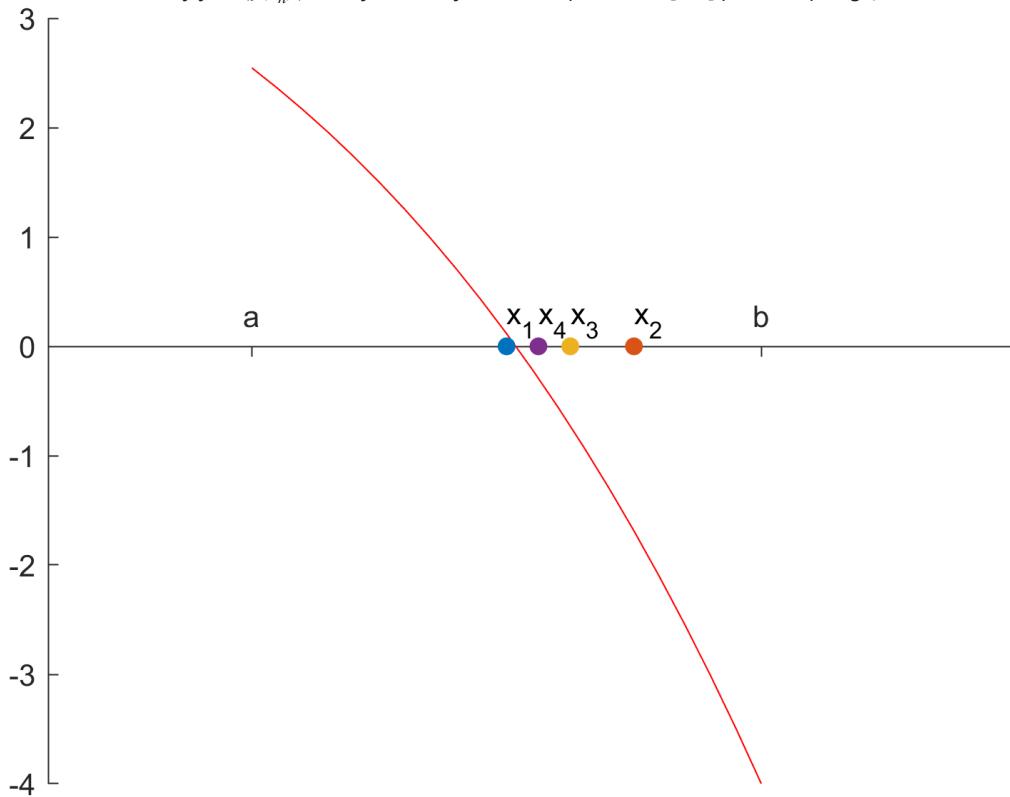
$$|x_n - x_{n+1}| < \frac{1}{2^n}(b - a) < \varepsilon \quad (7.5.1)$$

Jednak przy dużej ilości kroków, błędy zaokrągleń mogą nie dopuścić do otrzymania żądanej dokładności. Metoda ta jest wolno zbieżna, bo z konstrukcji wynika, że przedziały mają za każdym razem mniejszą długość tylko o połowę. Rząd metody bisekcji jest równy 1.

Przykład 7.5

Przykład

Szukamy pierwiastka wielomianu $f(x) = x^8 - 10x^6 + 5$ w przedziale $[a, b]$, gdzie $a = 0.8, b = 1$ z dokładnością $d = 0.0001$, gdzie dokładność oznacza dla nas zakończenie iteracji jeśli $|f(x_n)| < d$, i jeśli funkcja nie ma w przedziale $[a, b]$ punktów przegięcia.



Rys. 7.9. Ilustracja graficzna 4 iteracji.

Na rysunku pokazany jest wielomian w rozpatrywanym przedziale. Widać, że funkcja f nie ma punktów przegięcia w tym przedziale, i zmienia znak w $[a, b]$. Punkt $x_1 = \frac{a+b}{2} = 0.9$.

Otrzymujemy następujące wyniki, ciąg $\{x_n\}$ dla $j = 1, \dots, 13$

[0.9000]

```
0.9500
0.9250
0.9125
0.9062
0.9031
0.9047
0.9039
0.9035
0.9037
0.9036
0.9037
0.9036 ]
```

Ciąg $\{x_n\}$ nie jest monotoniczny, oscyluje wokół pierwiastka p wielomianu $f(x)$.

Ostatnia iteracja daje nam pierwiastek z podaną dokładnością $x_{13} = 0.903638$ i $f(x_{13}) = 0.000016$.

Jeśli warunkiem stopu będzie warunek $|x_{n-2} - x_{n-1}| < d$ to dostaniemy jako pierwiastek 11-tą iterację $x_{11} = 0.903613$ i wartość funkcji $f: f(x_{11}) = 0.000771$.

Oto skrypt w MATLABie rozwiązuający zadanie oraz generujący powyższy wykres

```
a=0.8;
b=1;
f = @(x) (x.^8)-10*x.^6+5;
xd = a:0.01:b;

% konfiguracja ładnego wykresu
close all
figure
set(gca,'visible','off')
axes('color', 'none', 'YAxisLocation', 'origin', 'XAxisLocation', 'origin')
hold on
plot(xd,f(xd),'r')
xlim([a-0.1*a,b+b*0.1*b]);
xticks([a, b])
xticklabels({'a','b'})

% w tej zmiennej zapamiętamy wyniki poszczególnych iteracji
X_i = []
d = 1e-4;
for i=1:100
    x = (a+b)/2
    X_i = [X_i x];

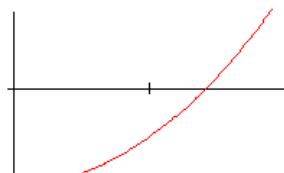
    % rysujemy tylko cztery pierwsze punkty
    if i < 5
        scatter(X_i,0,'filled');
        text(x,0.25, sprintf('x_%d',i))
    end

    % wybieramy przedział do kolejnej iteracji
    if (f(a)*f(x) < 0)
        b = x;
    else
        a = x;
    end

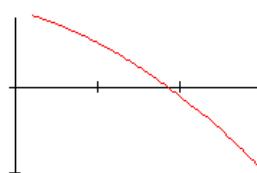
    % kryterium stopu procedury
    if (i>1) && (abs(X_i(end-1) - X_i(end))< d )
        break;
    end
end
X_i'
f(x)
```


7. Metoda siecznych

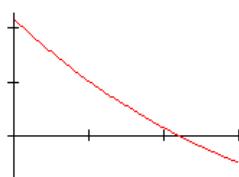
Założenia: Funkcja f jest klasy $C^2(a,b)$, zmienia znak w przedziale (a, b) oraz pochodne pierwsza i druga mają stały znak w rozpatrywanym przedziale. To znaczy, że w przedziale izolacji (a, b) może zachodzić któryś z czterech podanych na rysunkach przypadków:



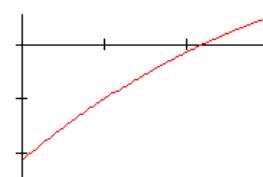
funkcja rośnie, jest wypukła



funkcja maleje , jest wkłasta



funkcja maleje , jest wypukła



funkcja rośnie , jest wkłasta

Rys 7.10. Możliwe przypadki wykresów funkcji f w przedziale izolacji (a, b) .

Na rysunku opieramy się o przypadek, kiedy pierwsza i druga pochodna są dodatnie i startujemy z punktu $(b=x_0)$ oraz z punktu (x_1) leżącego po lewej stronie (b) , ale po prawej stronie od pierwiastka.

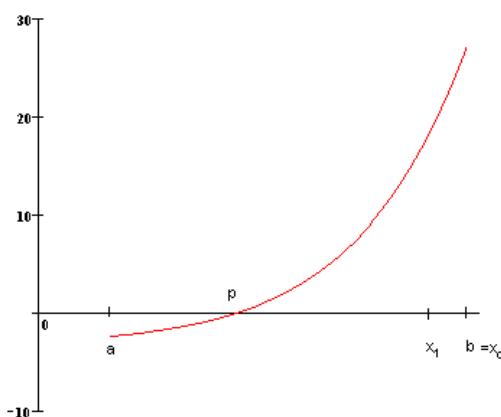
Opis metody: Metoda siecznych jest metodą dwukrokową, startujemy z dwóch punktów (x_0) i (x_1) takich, że $f(x_0)f''(x_0)>0$, $f(x_1)f''(x_1)>0$. Przez punkty $(x_0, f(x_0))$ i $(x_1, f(x_1))$ prowadzimy sieczną i przecinamy ją z osią (Ox) , punkt przecięcia wyznacza następną iterację (x_2) .

$$\begin{aligned} y - f(x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_1) \end{aligned}$$

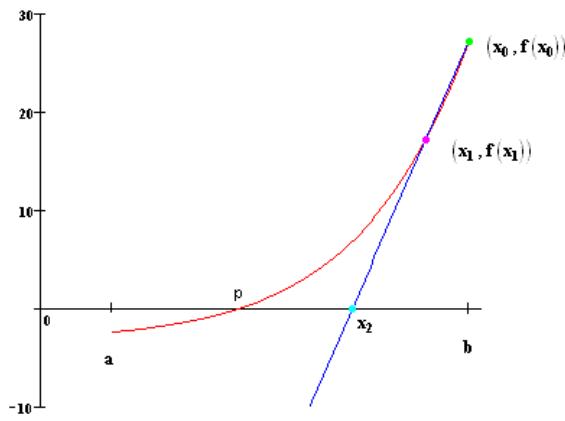
stąd

$$x_2 = x_1 - \frac{f(x_1) - f(x_0)}{f(x_1) - f(x_0)}(x_1 - x_0)$$

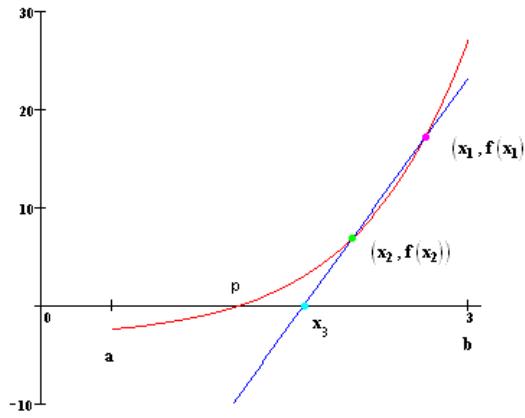
Jeśli $f(x_2) = 0$ to x_2 jest pierwiastkiem, jeśli nie, przez punkty $(x_1, f(x_1))$ i $(x_2, f(x_2))$ prowadzimy sieczną i przecinamy ją z osią (Ox) , dostajemy następną iterację: $x_3 = x_2 - \frac{f(x_2) - f(x_1)}{f(x_2) - f(x_1)}$.



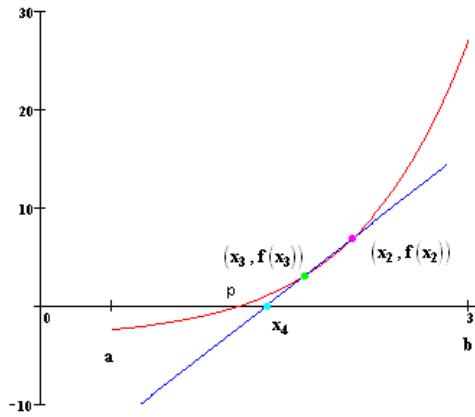
a) Przez punkty $(x_0, f(x_0))$ i $(x_1, f(x_1))$ prowadzimy prostą i przecinamy ją z osią Ox , punkt przecięcia oznaczamy przez x_2 ,



b) Przez punkty $(x_1, f(x_1))$ i $(x_2, f(x_2))$ prowadzimy prostą i przecinamy ją z osią Ox , punkt przecięcia oznaczamy przez x_3 ,



c) Przez punkty $(x_2, f(x_2))$ i $(x_3, f(x_3))$ prowadzimy prostą i przecinamy ją z osią Ox , punkt przecięcia oznaczamy przez x_4 itd.,



Rys. 7.11. Kolejne trzy iteracje w metodzie siecznych.

Postępując kolejno w wyżej opisany sposób otrzymamy wzór ogólny na ciąg iteracyjny:

$$(x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}) \quad (7.6.1)$$

Jeśli nie będziemy przestrzegać spełnienia warunków na punkty startu, ciąg może być zbieżny do pierwiastka, ale nie zawsze (przykład 7.7).

Dla tej metody jest prawdziwe twierdzenie:

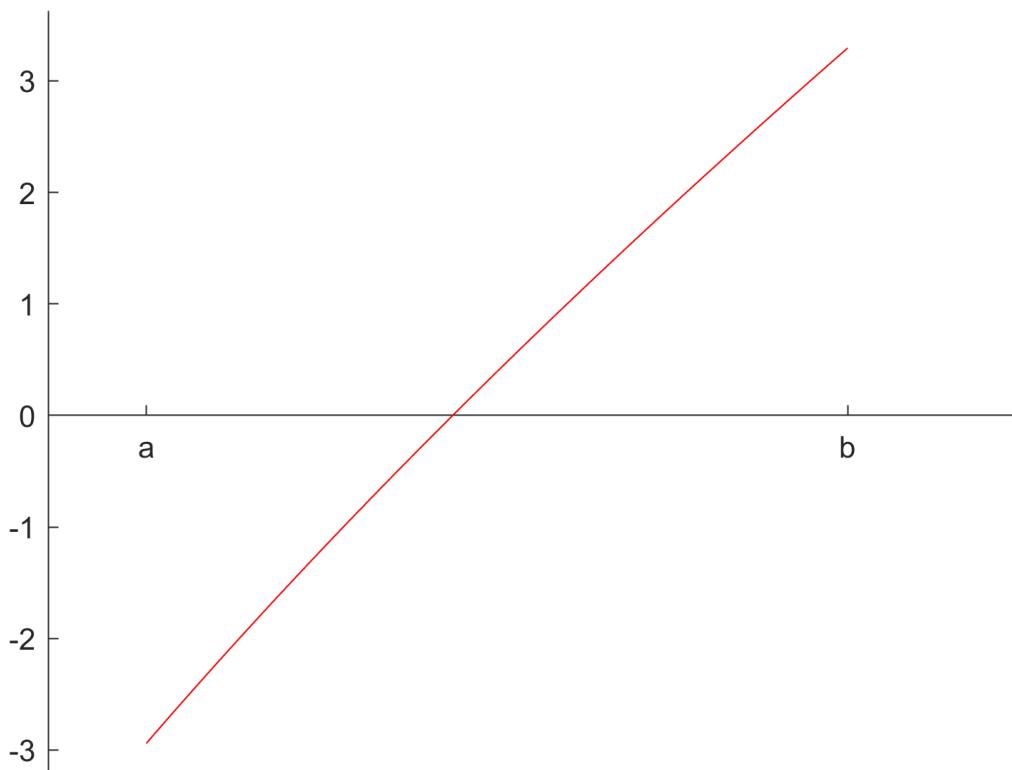
Twierdzenie

Jeśli w otoczeniu $|x-p|<\delta$ pierwiastka p równania $f(x)=0$ funkcja f ma ciągłą drugą pochodną, a pierwsza i druga pochodna jest różna od zera w tym otoczeniu oraz przybliżenia x_0 i x_1 ($x_0 \neq x_1$) są dostatecznie bliskie pierwiastka p , to metoda siecznych jest zbieżna, jej rzad jest równy $\frac{\sqrt{5}+1}{2} \approx 1.618 \dots$, a stała asymptotyczna błędu jest równa $C = \left(\frac{|f''(p)|}{2|f'(p)|}\right)^{\frac{1}{2}} \frac{\sqrt{5}-1}{2}$.

Przykład 7.6

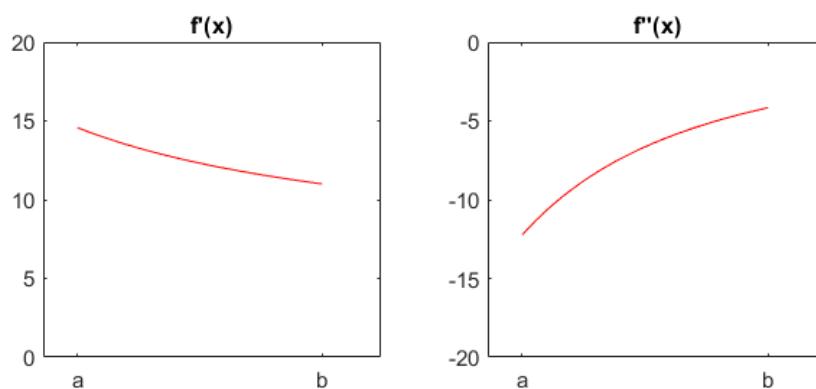
Przykład

Rozpatrujemy równanie $6x+6\ln x-5=0$ w przedziale $(0.7, 1.2)$ z dokładnością $d=10^{-8}$.



Rys. 7.12. Wykres funkcji f w podanym przedziale.

Na rysunku widać, że funkcja zmienia znak w danym przedziale. Na następnych rysunkach są zilustrowane pierwsza i druga pochodna funkcji f w tym samym przedziale.



Rys. 7.13. Wykresy pochodnych funkcji $\lambda(f)$.

Widac, że pierwsza pochodna jest dodatnia w rozpatrywanym przedziale, a druga pochodna jest ujemna w $\langle [a, b] \rangle$. Spełnione są założenia dla metody siecznych, pochodne są ciągłe i nie zmieniają znaku w $\langle [a, b] \rangle$.

Możemy zastosować metodę siecznych, wybierając za punkty startu takie punkty, w których funkcja ma taki sam znak jak druga pochodna. Ponieważ druga pochodna jest ujemna wybieramy punkty po lewej stronie pierwiastka, w którym funkcja też jest ujemna
 $(X_0=a, x_1=a+0.01)$

Za pomocą wzoru iteracyjnego na x_{n+1} dostajemy wektor iteracji, wzór jest przeliczany tak długo dopóki nie będzie osiągnięta dokładność, tzn. aż $|f(x_n)| < \epsilon$. Gdzie ϵ jest stosunkowo mała założona przez nas liczbą, która stanowi kryterium zbieżności.

Wektor iteracji:

```
\begin{aligned} &x_j = \begin{array}{r} \hline 0.7 \\ \hline 0.71 \\ \hline 0.9026114008 \\ \hline 0.9173854594 \\ \hline 0.9184219035 \\ \hline 0.91842661 \\ \hline 0.9184266114 \\ \hline \end{array} \end{aligned}
```

Rozwiæzaniem jest: $x_6 = 0.9184266114$, dla którego $f(x_6) = -2,309 \cdot 10^{-14}$

Przykładowa implementacja w MATLABie:

```

a=0.7;
b=1.2;
f = @(x) 6*x+6*log(x)-5;

x = [a a+0.01];
d = 1e-13;
max_iter = 20;
iter = 1;
n=2;
while abs(f(x( n ))) > d && iter < max_iter
x(n+1) = x( n ) - f(x( n ))*(x( n )-x(n-1)) / (f(x( n )) - f(x(n-1)));
iter = iter + 1;
n = n + 1;
end
format long
x
format short

```

Uruchomienie programu daje wynik:

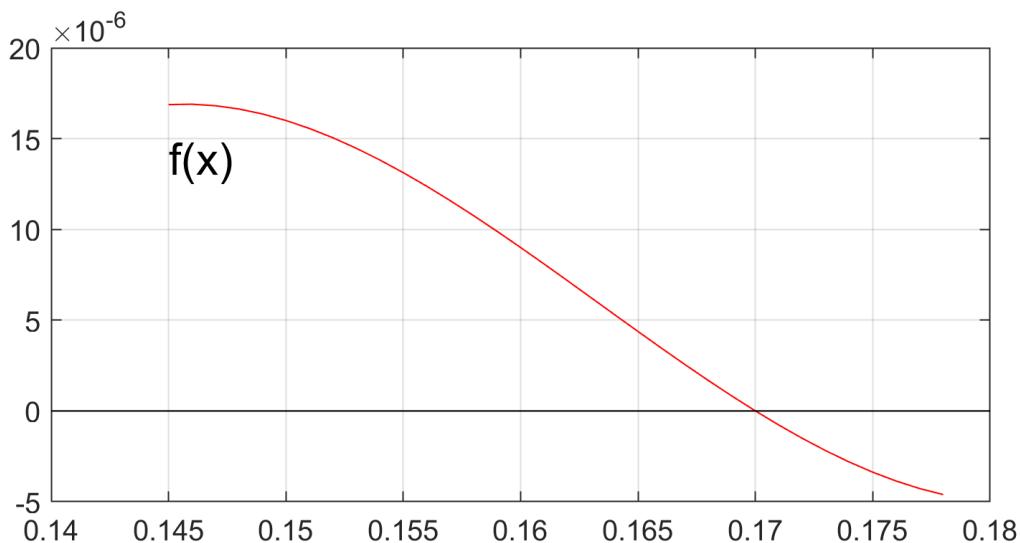
```
x =  
    0.700000000000000  0.710000000000000  0.902611400765419  0.917385459390865  0.918421903512899  0.918426609980826  
0.918426611372439
```

Przykład 7.7

Przykład

Przykład ilustruje sytuację, w której nie są spełnione założenia przy jakich możemy stosować tę metodę. Dane jest równanie: $\sqrt{x^3 - 0.49x^2 + 0.0791} - x - 0.004199 = 0$

Szukamy pierwiastka tego równania w przedziale $\langle [0.145, 0.178] \rangle$. Pierwiastek istnieje, bo jak widać na rysunku, funkcja zmienia znak, ten pierwiastek jest blisko punktu 0,17.

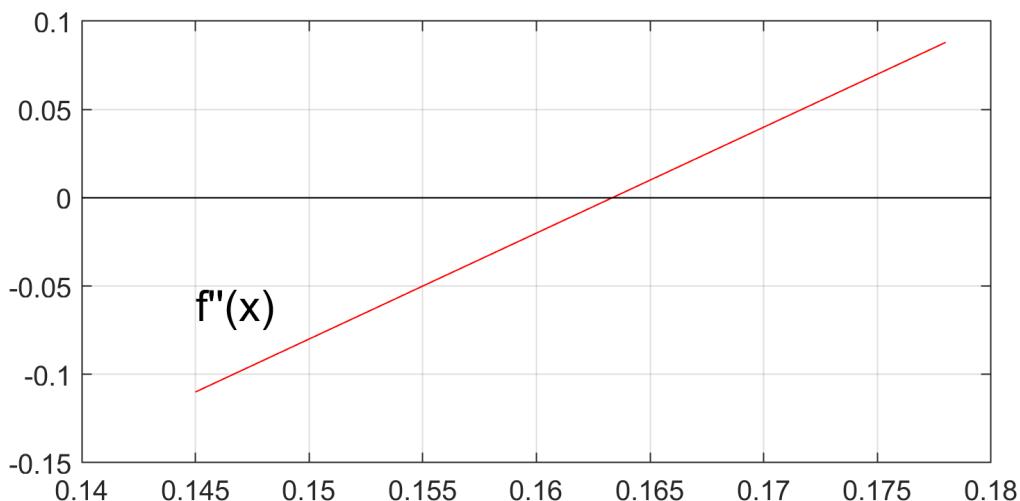


Rys. 7.14. Wykres funkcji f w podanym przedziale.

Znajdziemy ten pierwiastek metodą siecznych. Startujemy z punktów: $x_0 = a$ i $x_1 = a + 0,01$ i stosujemy wzór:

$$(x_{n+1} = x_n - \frac{f(x_n) - f(x_{n-1})}{f'(x_n) - f'(x_{n-1})})$$

Dostajemy dla 10 iteracji $x_{10} = 0.19$ i $f(x_{10}) = 0$. Jednak **to nie jest pierwiastek z tego przedziału**, nasz pierwiastek był blisko punktu 0.17. Dlaczego tak się stało? Pochodna druga zmienia znak w tym przedziale, w dodatku wystartowaliśmy ze złych punktów. Ponieważ wykres drugiej pochodnej jest następujący:



Rys. 7.15. Wykres drugiej pochodnej, która zmienia znak w rozpatrywanym przedziale.

Przedział nie może być w tym wypadku taki duży, powinniśmy zmienić go na $[0.165, 0.178]$ i sprawdzić pozostałe założenia.

8. Metoda stycznych - Newtona

Podobnie jak w metodzie siecznych w metodzie stycznych założymy, że funkcja f jest klasy $C^2(a,b)$, zmienia znak w przedziale (a, b) oraz pochodne pierwsza i druga mają stały znak w rozpatrywanym przedziale. To znaczy, że w przedziale izolacji (a, b) może zachodzić któryś z czterech podanych na rysunku 7.10 przypadków.

Opis metody: Metodę opiszemy korzystając z przypadku pierwszego, kiedy pierwsza pochodna jest dodatnia (funkcja rośnie) i druga pochodna jest dodatnia (funkcja jest wypukła).

Jako punkt startu obieramy taki punkt x_0 , w którym funkcja ma taki sam znak jak druga pochodna: $f'(x_0) > 0$, w naszym przypadku punkt b - ponieważ w tym punkcie funkcja jest dodatnia tak jak druga pochodna. Z punktu $(x_0, f(x_0))$ wystawiamy styczną do krzywej $y = f(x)$. Równanie stycznej ma postać:

$$y - f(x_0) = f'(x_0)(x - x_0)$$

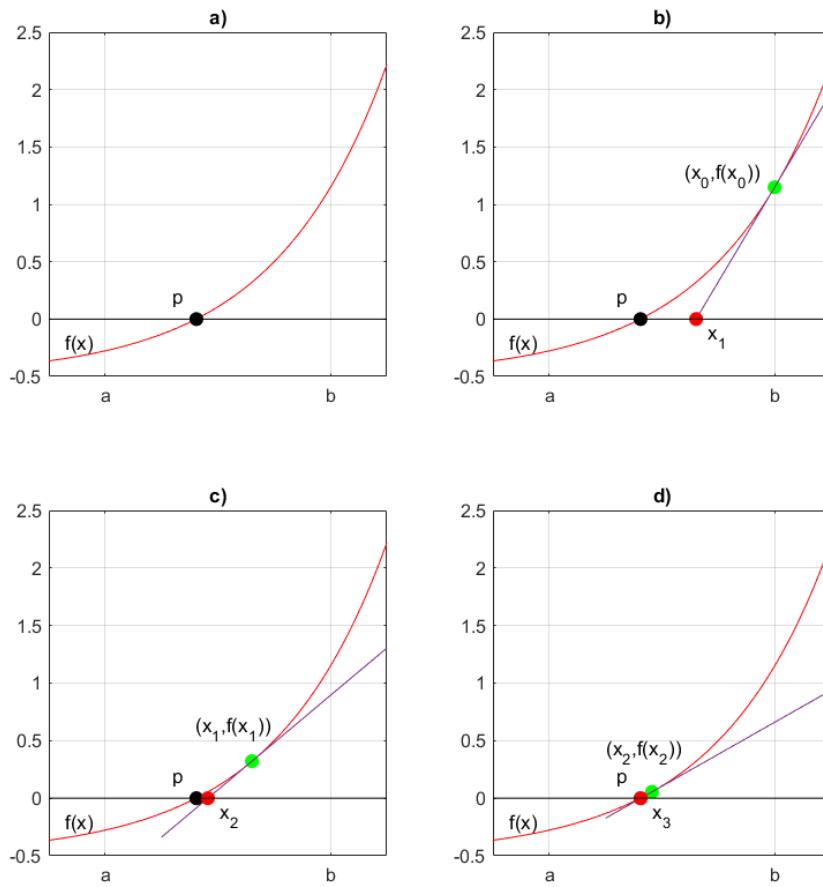
Przecinamy styczną z osią Ox i otrzymany punkt przecięcia jest pierwszym przybliżeniem pierwiastka. Wstawiając zatem za y zero a za x wartość x_1 otrzymujemy:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Jeśli $f(x_1) = 0$ to x_1 jest pierwiastkiem, jeśli nie, postępujemy analogicznie dalej, z punktu $(x_1, f(x_1))$ wystawiamy styczną do krzywej i przecinamy ją z osią Ox : $y - f(x_1) = f'(x_1)(x - x_1)$, przyjmujemy $^*y = 0$, i otrzymujemy: $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$.

Na rysunku 7.16 przedstawione zostały kolejne iteracje metody Newtona dla przykładowej funkcji z dodatnimi wartościami pierwszej i drugiej pochodnej $f'(x)$. Kolejne rysunki przedstawiają:

- a) - przebieg funkcji $f(x)$ z poszukiwanym pierwiastkiem p ,
- b) Przez punkt $(x_0, f(x_0))$ prowadzimy styczną do $f(x)$ i przecinamy ją z osią Ox , punkt przecięcia oznaczamy przez x_1 ,
- c) Przez punkt $(x_1, f(x_1))$ prowadzimy styczną i przecinamy ją z osią Ox , punkt przecięcia oznaczamy przez x_2 ,
- d) Przez punkt $(x_2, f(x_2))$ prowadzimy styczną i przecinamy ją z osią Ox , punkt przecięcia oznaczamy przez x_3 .



Rys. 7.16. a) Przebieg funkcji $f(x)$ i b), c), d) kolejne trzy iteracje w metodzie stycznych.

Powtarzając w ten sposób budowanie kolejnej iteracji otrzymujemy ciąg iteracyjny x_n określony wzorem :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (7.7.1)$$

Ciąg ten przy podanych założeniach jest zbieżny do szukanego pierwiastka p . Może się zdarzyć, że startując z innego punktu, nie spełniającego podany warunek $f(x_0)f'(x_0) > 0$, ciąg iteracyjny też będzie zbieżny do szukanego pierwiastka, ale bez tego warunku nie mamy gwarancji, że ciąg x_n zbiega do p (przykład 7.9).

Dla tej metody jest prawdziwe twierdzenie:

Twierdzenie

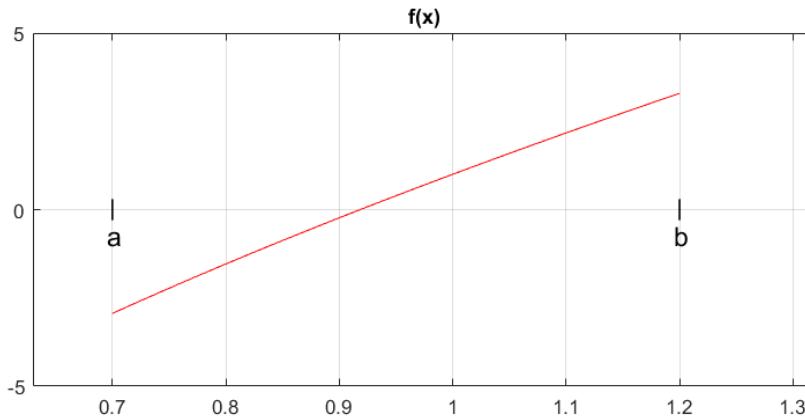
Jeżeli w otoczeniu $|x-p|<\delta$ pierwiastka p równania $f(x)=0$ funkcja f ma ciągłą drugą pochodną oraz pierwsza i druga pochodna są różne od zera w tym otoczeniu oraz x_0 leży wystarczająco blisko pierwiastka p , to metoda Newtona jest rzędu 2 ze stałą asymptotyczną błędem $C=\frac{|f''(p)|}{|2f'(p)|}$.

Metoda Newtona jest szybkozbieżną metodą jednokroikową wymagającą na każdym kroku obliczania jednej wartości funkcji i jednej wartości pierwszej pochodnej.

Przykład 7.8

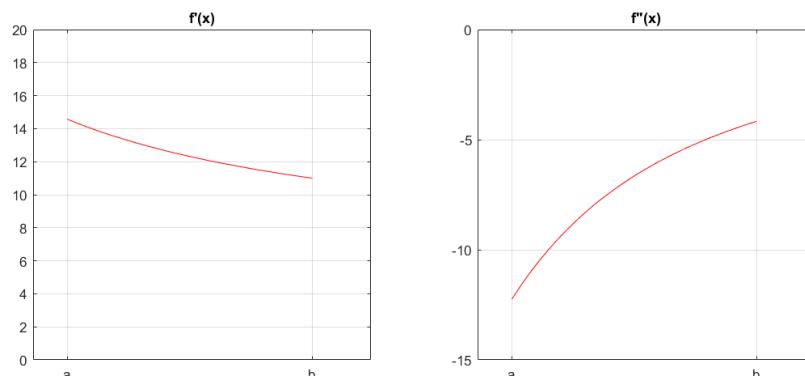
Przykład

Rozpatrujemy równanie $6x+6 \ln x - 5 = 0$ w przedziale $[0.7, 1.2]$. Na rysunku 7.17 widać, że funkcja zmienia znak w danym przedziale



Rys. 7.17. Wykres funkcji f w przedziale $[0.7, 1.2]$.

Na rysunkach 7.18a i 7.18b zilustrowane są pierwsza i druga pochodna funkcji f w tym samym przedziale. Widać, że pierwsza pochodna jest dodatnia w rozpatrywanym przedziale, a druga pochodna jest ujemna w $[a, b]$. Spełnione są założenia dla metody Newtona, pochodne są ciągłe i nie zmieniają znaku w $[a, b]$.



Rys. 7.18. Wykres pochodnych w rozpatrywanym przedziale.

Możemy zastosować metodę Newtona przyjmując za punkt startu ten koniec przedziału $[a, b]$, dla którego jest spełniony warunek $f(x_0)f''(x_0)>0$. W tym przypadku jest to punkt a , zatem $x_0 = a$. Dla dokładności $d=10^{-8}$ otrzymujemy 4 iteracje i $x_4=0.9184266114$ jest przybliżonym pierwiastkiem równania oraz $f(x_4)=0$ (przyjmujemy za zero wszystkie liczby mniejsze niż 10^{-15}).

Wektor iteracji ma postać: $\{j = 0, 1..4\}$

$$\begin{aligned} x_j = & 0.7 \\ & \hline 0.9017681142 \\ & \hline 0.9183466866 \\ & \hline 0.9184266096 \\ & \hline 0.9184266114 \end{aligned}$$

Ten sam przykład, dla tej samej dokładności obliczyliśmy w poprzednim temacie metodą siecznych. Aby otrzymać żądaną dokładność

trzeba było dla tamtej metody wziąć o dwie iteracje więcej. Metoda stycznych (Newtona) jest zatem szybciej zbieżną metodą.

Przykład 7.18

Przykład

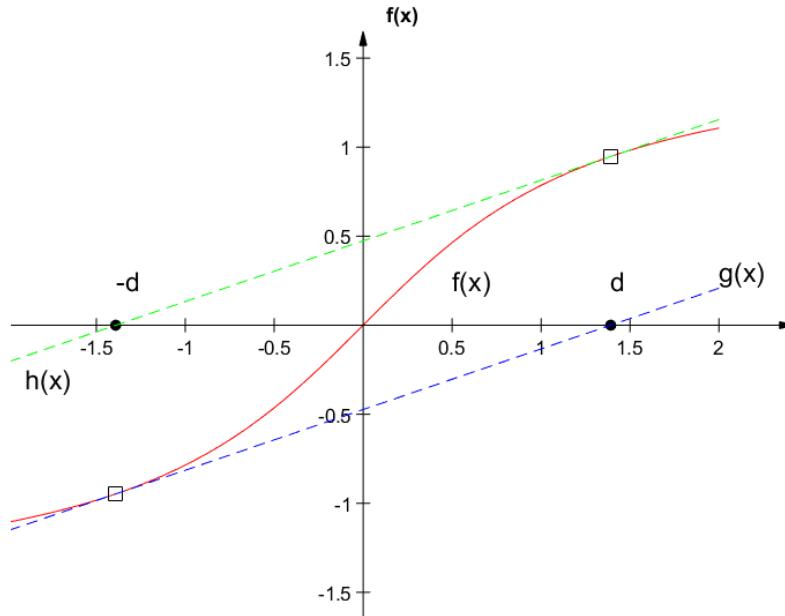
Rozpatrujemy równanie $\arctg(x)=0$ w przedziale $[-2, 2]$. Jako punkt startu, w celu demonstracji zjawiska zapętlenia, obieramy dokładny pierwiastek równania:

$$\arctg x - \frac{2}{1+x^2} = 0$$

Oznaczmy ten pierwiastek przez d i w przybliżeniu równa się on 1.39125043 . Będziemy stosować metodę Newtona dla równania: $\arctg x = 0$ z punktem startowym $x_0 = d$.

Wstawiamy z punktu $(d, f(d))$ styczną do krzywej $\arctg(x)$: $g(x) = f'(x-d) + f(d)$ i przecinamy ją z osią Ox wyznaczając punkt x_1 .

Okazuje się że punkt przecięcia będzie $x_1 = -d$. Jeśli z punktu $(-d, f(-d))$ wystawimy do krzywej $\arctg(x)$ styczną: $h(x) = f'(-d)(x+d) + f(-d)$ i przetniemy ją z osią Ox dostaniemy znów punkt $x_2 = d$. W ten sposób metoda Newtona "zapętliła" się i ze wzoru Newtona dostajemy na zmianę punkty d i $-d$ jako kolejne iteracje, a widać na rysunku, że pierwiastkiem równania jest $p=0$.



Rys. 7.19. Wykres funkcji f i stycznych wychodzących z punktów $(d,0)$ i $(-d,0)$.

To zapętlenie wynika z tego, że druga pochodna zmienia znak w przedziale $[-2, 2]$, ma w zerze punkt przegięcia. Nie są zatem spełnione założenia podane do metody Newtona.

9. Pierwiastki wielokrotne

Metody iteracyjne wymagają na ogół, aby szukany pierwiastek był pierwiastkiem jednokrotnym. Tak jest przy metodzie Newtona i metodzie siecznych. Metoda bisekcji dopuszcza pierwiastki nieparzystokrotne, przy parzystokrotnych funkcja nie zmienia znaku w przedziale izolacji. Na ogół nie znamy krotności szukanych pierwiastków.

Wprowadzamy funkcję pomocniczą $(u(x) = \frac{f(x)}{f'(x)})$ i rozwiążujemy równanie $(u(x)=0)$ zamiast równania $(f(x)=0)$. Równanie $(u(x)=0)$ ma takie same pierwiastki jak równanie $(f(x)=0)$, ale wszystkie są jednokrotne.

Ponieważ:

$$\begin{aligned} u'(x) &= \frac{f'(x)f'(x) - f(x)f''(x)}{(f'(x))^2} = \frac{f'(x)(f'(x) - f(x)f''(x))}{(f'(x))^2} \\ &= \frac{f'(x)(1 - u(x)f''(x))}{(f'(x))^2} = 1 - u(x)\frac{f''(x)}{f'(x)} \end{aligned}$$

wzory na metodę Newtona i metodę siecznych przybierają postać:

dla metody Newtona (stycznych):

$$(x_{n+1} = x_n - \frac{u(x_n)}{u'(x_n)}) \quad (7.8.1)$$

gdzie $(u'(x_n) = 1 - u(x_n)\frac{f''(x_n)}{f'(x_n)})$

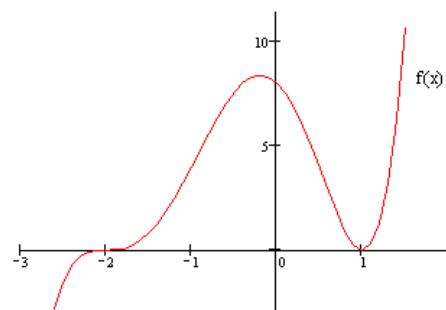
dla metody siecznych:

$$(x_{n+1} = x_n - \frac{u(x_n) - u(x_{n-1})}{u'(x_n) - u'(x_{n-1})}) \quad (7.8.2)$$

Przykład 7.19

Przykład

Funkcja nieliniowa $(f(x) = (x-1)^2(x+2)^3)$ będzie wielomianem stopnia piątego mającym jeden pierwiastek dwukrotny i jeden trzykrotny.

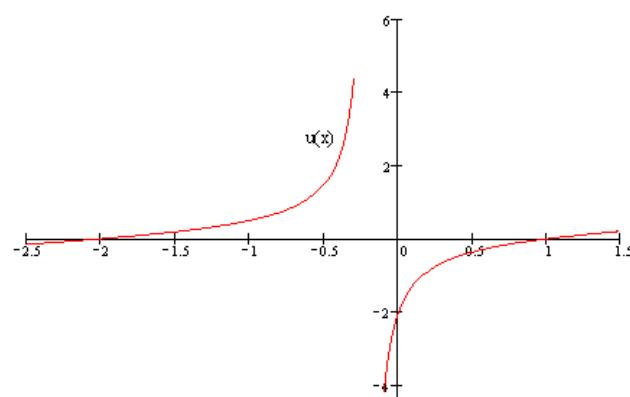


Rys. 7.20. Wykres funkcji (f) .

Obliczymy jej pochodną i następnie wprowadzimy funkcję $(u(x) = \frac{f(x)}{f'(x)})$.

$$\begin{aligned} f(x) &= (x-1)^2(x+2)^3 \\ f'(x) &= 2(x-1)(x+2)^3 + 3(x-1)^2(x+2)^2 = (x-1)(x+2)^2[2(x+2) + 3(x-1)] = (x-1)(x+2)^2(5x+1) \end{aligned}$$

Równanie $(u(x)=0)$ ma dwa pierwiastki, takie jak funkcja (f) ale są już jednokrotne. Funkcja (u) nie jest ciągła na całej osi (R) , ale istnieją przedziały izolacji pierwiastków, w których jest ciągła i ma ciągłe pochodne.



Rys. 7.21. Wykres funkcji $u(x)$.

10. Układy równań nieliniowych

Dany jest układ $\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases}$ niewiadomymi:

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases} \quad (7.9.1)$$

który będziemy zapisywać wektorowo: $\begin{cases} F(x)=0 \end{cases}$, gdzie $x \in \mathbb{R}^n$; $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$.

Warunki istnienia rozwiązań układu są znacznie trudniejsze do sprawdzenia, a nawet nie można sformułować jednolitego kryterium istnienia rozwiązania bez założenia szczególnych własności odwzorowania F , takich jak różniczkowalność itd. Będziemy zakładać istnienie rozwiązania układu $\begin{cases} F(x)=0 \end{cases}$ i ograniczymy się do jednej metody: poszukiwania rozwiązań metodą Newtona.

Rozpatrzmy metodę iteracyjną jednokrokową daną ogólnym wzorem: $\begin{cases} x^{k+1} = G(x^k) \\ x^0 \text{ wektor początkowy} \end{cases}$ i wektor początkowy x^0 , który będziemy dobierać dostatecznie blisko rozwiązania.

Definicja

Niech $G: \mathbb{R}^n \rightarrow \mathbb{R}^n$. Punkt w nazywamy punktem przyciągania metody iteracyjnej (ewentualnie punktem stacjonarnym), jeżeli istnieje takie otoczenie U tego punktu, że obierając dowolny wektor początkowy x^0 z tego otoczenia uzyskamy ciąg punktów x^1, x^2, \dots, x^n zbieżny do w . Największe z tych otoczeń nazywamy obszarem przyciągania punktu w (punktu stacjonarnego).

Oznaczmy przez :

$$\begin{cases} F(x) = \begin{aligned} & f_1(x_1, \dots, x_n) \\ & \vdots \\ & f_n(x_1, \dots, x_n) \end{aligned} \end{cases} \quad (7.9.2)$$

oraz

$$\begin{cases} J(x) = \begin{aligned} & \frac{\partial f_1}{\partial x_1}(x_1, \dots, x_n) & \dots & \frac{\partial f_1}{\partial x_n}(x_1, \dots, x_n) \\ & \vdots \\ & \frac{\partial f_n}{\partial x_1}(x_1, \dots, x_n) & \dots & \frac{\partial f_n}{\partial x_n}(x_1, \dots, x_n) \end{aligned} \end{cases} \quad (7.9.3)$$

Jeśli funkcje $f_i(x)$ są różniczkowalne w sposób ciągły w pewnym otoczeniu punktu p , w którym $F(p)=0$, i macierz $J(x)$ jest w tym otoczeniu nieosobliwa, to jeśli wektor startu dobierzemy odpowiednio blisko punktu p to punkt p jest punktem przyciągania metody iteracyjnej danej wzorem (metoda Newtona):

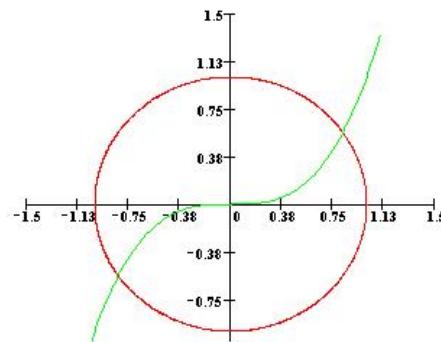
$$x^{n+1} = x^n - J^{-1}(F(x^n)) \quad (7.9.4)$$

Przykład 7.20

Przykład

Rozpatrujemy układ równań:

$$\begin{aligned} & x^2 + y^2 - 1 = 0 \\ & x^3 - y = 0 \end{aligned}$$



Rys. 7.22. Graficzna interpretacja układu.

Na rysunku czerwona linia opisuje pierwsze równanie, zielona drugie. Widać, że krzywe przecinają się w dwóch punktach, układ ma dwa rozwiązania. Lewą stronę pierwszego równania oznaczamy przez $f_1(x,y)$, lewą stronę drugiego równania oznaczamy przez $f_2(x,y)$. Oznaczmy przez:

$$\begin{aligned} F(x, y) = & \left[\begin{array}{l} f_1(x, y) \\ f_2(x, y) \end{array} \right] \text{ oraz } J(x, y) = \left[\begin{array}{ll} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{array} \right] \end{aligned}$$

Znajdziemy rozwiązanie w pierwszej ćwiartce. Jako wektor startu bierzemy (odczytujemy z rysunku), wektor (z) ma pierwszą współrzędną (x) a drugą (y) .

$$(z^{\langle 0 \rangle}) = \left[\begin{array}{l} 0.9 \\ 0.5 \end{array} \right]$$

Korzystając ze wzoru Newtona dla układów równań:

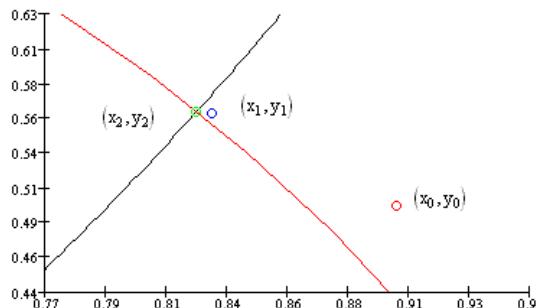
$$(x^{\langle 1 \rangle}) = (x^{\langle 0 \rangle}) - \left(\begin{array}{l} F(x^{\langle 0 \rangle}) \\ F(x^{\langle 0 \rangle}) \end{array} \right)^{-1} F'(x^{\langle 0 \rangle})$$

dostajemy dokładności $(d=10^{-8})$, tzn. $|f(z^{\langle n \rangle})| \leq d$ | całą macierz iteracji, w kolumnach której są kolejne wektory iteracyjne:

$$\begin{aligned} (\mathbf{z}) = & \left[\begin{array}{l} 0.9 \\ 0.831678 \\ 0.826062 \\ 0.826031 \\ 0.5 \\ 0.562979 \\ 0.563608 \\ 0.563624 \end{array} \right] \\ & \vdots \end{aligned}$$

Rozwiązaniem jest trzecia iteracja skąd $(x = 0.826031)$, a $(y = 0.563624)$.

Na rysunku, w dużym powiększeniu, czerwone kółeczko to punkt startu, pierwsze przybliżenie to niebieskie kółeczko, drugie przybliżenie to zielone kółeczko, a rozwiązanie przybliżone, czyli trzecia iteracja pokrywa się na rysunku z drugą.



Rys. 7.23. Kolejne trzy iteracje rozwiązywania układu.

Wartość funkcji wektorowej opisującej równania jest dla tego rozwiązania następująca:

$$(F(z^{\langle \rangle})) = \left[\begin{array}{l} 1.19 \cdot 10^{-9} \\ 2.294 \cdot 10^{-9} \end{array} \right]$$

Jeśli będziemy brać jako wektor startu wektor o współrzędnych o przeciwnych znakach

$$(z^{\langle 0 \rangle}) = \left[\begin{array}{l} -0.9 \\ -0.5 \end{array} \right]$$

dostaniemy symetryczne rozwiązanie $(x = -0.826031)$, a $(y = -0.563624)$.

Przykład rozwiązywany w MATLABie w postaci implementacji naiwnej - ręcznie wpisanych iteracji, oraz w postaci algorytmicznej.

```
% Implementacja naiwna (niezalecana)

% w poniższych wzorach przyjęta: x(1) - x, x(2) - y
F = @(x) ([x(1).^2 + x(2).^2 - 1; x(1).^3 - x(2)]);
DF = @(x) ([2*x(1) 2*x(2); 3*x(1)^2 -1])

x0 = [0.9
      0.5]

x1 = x0 - inv(DF(x0))*F(x0);
x2 = x1 - inv(DF(x1))*F(x1);
x3 = x2 - inv(DF(x2))*F(x2)

F(x3)

% Implementacja algorytmiczna,
% z określeniem maksymalnej liczby iteracji
x0 = [0.9
      0.5];
X = [x0];

max_iter = 10;
n = 1;
d = 1e-8;
while (n < max_iter && norm(F(X(:,n))) > d)
    X(:,n+1) = X(:,n) - inv(DF(X(:,n)))*F(X(:,n));
    % lub wydajniej, aby uniknąć odwracania macierzy
    % X(:,n+1) = X(:,n) - DF(X(:,n)) \ F(X(:,n));
    n = n+1;
end

n
F(X(:,n))
format long
X
format short
```

W wyniku uruchomienia skryptu otrzymamy wynik:

```
x3 =
  0.8260
  0.5636

ans =
  1.0e-08 *
  0.1190
  0.2294

n =
  4

ans =
  1.0e-08 *
  0.1190
  0.2294

X =
  Columns 1 through 3
  0.900000000000000  0.831678486997636  0.826061782413291
  0.500000000000000  0.562978723404255  0.563607908816801
  Column 4
  0.826031358607699
  0.563624161819281
```

8 Rozdział

Strona: [LeIA](#)

Kurs: Metody numeryczne (2025Z)

Książka: 8 Rozdział

Wydrukowane przez użytkownika: Kinga Kondraciuk

Data: niedziela, 30 listopada 2025, 14:16

Spis treści

- 1. Całkowanie numeryczne**
- 2. Metody proste trapezów i parabol**
- 3. Metody złożone trapezów i parabol**
- 4. Węzły Legendre'a**
- 5. Uwagi o dokładności**

1. Całkowanie numeryczne

Bardzo dużo zagadnień technicznych, fizycznych, mechanicznych sprowadza się do obliczania całek oznaczonych funkcji jednej zmiennej. Można za pomocą tych całek liczyć np.: długości łuków, pola obszarów, pola powierzchni obrotowych, objętości brył obrotowych, masy ciał, momenty statyczne i bezwładności, wartości napięcia dla pól elektrycznych, itd. Dokładne obliczenie tych całek wymaga znajomości funkcji pierwotnych dla funkcji podcałkowych, nie każda jednak funkcja posiada funkcję pierwotną. Zachodzi konieczność znalezienia całki oznaczonej metodą przybliżoną. Dotyczy to zwłaszcza sytuacji gdy przebieg określony jest za pomocą ciągu danych pomiarowych w punktach.

Zajmiemy się w tym rozdziale obliczaniem całek oznaczonych za pomocą wielomianów interpolacyjnych. W całce:

$$\int_a^b f(x) dx$$

będziemy zastępować funkcję $f(x)$ jej wielomianem interpolacyjnym n -tego stopnia, którego węzły x_0, x_1, \dots, x_n będą leżały w przedziale całkowania $[a, b]$ i będą teraz węzłami całkowania. Ograniczymy się w tym opracowaniu do całkowania funkcji bez osobliwości w przedziale $[a, b]$, tzn.: funkcji przyjmującej skończone wartości w rozpatrywanym przedziale, a przedział $[a, b]$ jest skończony. Wynika z tych założeń, że nie będziemy się zajmować całkowaniem całek niewłaściwych.

Do szacowania błędu całkowania wykorzystamy podane już wcześniej wzory na błąd interpolacji. Błąd ten zależy od pochodnych funkcji podcałkowych (a właściwie ich ekstremalnych wartości w granicach przedziału całkowania) i od węzłów. Będziemy rozpatrywać węzły równoodległe - dla prostoty obliczeń, a również węzły optymalne, tzn.: takie, które minimalizują tę część błędu całkowania, która zależy od węzłów.

Obliczając zatem całkę wstawiamy za $f(x)$ wielomian interpolacyjny Lagrange'a n -tego stopnia $L_n(x)$ dany wzorem:

$$L_n(x) = \sum_{k=0}^n \Phi_k(x) f(x_k) \quad (8.0.1)$$

gdzie :

$$\Phi_k(x) = \frac{(x-x_0)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)} \quad (8.0.2)$$

jest wielomianem Lagrange'a, a x_0, x_1, \dots, x_n są węzłami w przedziale $[a, b]$.

Otrzymamy:

$$\begin{aligned} I(f) &= \int_a^b f(x) dx \cong \int_a^b L_n(x) dx \\ &= \int_a^b \sum_{k=0}^n \Phi_k(x) f(x_k) dx \\ &= \sum_{k=0}^n f(x_k) \left(\int_a^b \Phi_k(x) dx \right) \\ &= \sum_{k=0}^n A_k f(x_k) \\ &= S(f) \end{aligned} \quad (8.0.3)$$

gdzie $A_k = \int_a^b \Phi_k(x) dx$.

A_k są współczynnikami zależnymi od węzłów, nie zależą od funkcji podcałkowej $f(x)$, łatwo je wyliczyć, bo są to całki ze zwykłych wielomianów w przedziale $[a, b]$.

Wzór na wartość $S(f)$ będziemy nazywać **kwadraturą**.

Błąd przybliżenia: $E(f) = I(f) - S(f)$, to różnica między dokładną wartością całki $I(f)$, a jej wartością przybliżoną $S(f)$. Do celu oszacowania jego górnej granicy będziemy używać wartości całki błędu interpolacji w przedziale $[a, b]$. Więcej informacji na ten temat w podrozdziale Uwagi o dokładności.

2. Metody proste trapezów i parabol

Rozpatrzymy skończony przedział $[a, b]$ oraz równoodległe węzły x_0, x_1, \dots, x_n w tym przedziale: $h = \frac{b-a}{n}$, $x_k = a + k \cdot h$, $k = 0, 1, \dots, n$. Wtedy całka jest równa w przybliżeniu:

$$I(f) = \int_a^b f(x) dx \cong \sum_{k=0}^n A_k f(x_k) = S(f)$$

gdzie $A_k = \int_a^b \Phi_k(x) dx$, oraz $\Phi_k(x)$ to wielomian Lagrange'a.

Wzór prosty trapezów.

Ustalmy $n = 1$ i wtedy $h = b - a$ i mamy dwa węzły $x_0 = a, x_1 = b$. Wielomian interpolacyjny Lagrange'a jest stopnia 1 i wstawiając za funkcję $f(x)$ wielomian $L_1(x)$ mamy:

$$\begin{aligned} I(f) &= \int_a^b f(x) dx \cong \int_a^b L_1(x) dx = \int_a^b \left(\frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) \right) dx = \\ &= f(x_0) \int_a^b \frac{x - x_1}{x_0 - x_1} dx + f(x_1) \int_a^b \frac{x - x_0}{x_1 - x_0} dx = \frac{h}{2} f(x_0) + \frac{h}{2} f(x_1) = S(f) \end{aligned} \quad (8.1.1)$$

Całki w ostatniej linijce wzoru łatwo obliczyć, bo są to całki oznaczone z wielomianów pierwszego stopnia.

W tym wypadku

$$\begin{aligned} A_0 &= \int_a^b \frac{x - x_1}{x_0 - x_1} dx = \\ &= \frac{1}{x_0 - x_1} \int_a^b (x - x_1) dx = \\ &= \frac{1}{-h} \int_a^b (x - b) dx = \\ &= \frac{1}{-h} \left(\frac{1}{2} x^2 - bx \right) \Big|_a^b = \\ &= \frac{h}{2} \end{aligned}$$

$$\begin{aligned} A_1 &= \int_a^b \frac{x - x_0}{x_1 - x_0} dx = \\ &= \frac{1}{h} \int_a^b (x - a) dx = \\ &= \frac{1}{h} \left(\frac{1}{2} x^2 - ax \right) \Big|_a^b = \\ &= \frac{h}{2} \end{aligned}$$

Możemy wzór na przybliżoną wartość $I(f)$ zapisać w postaci:

$$I(f) = \int_a^b f(x) dx \cong A_0 f(x_0) + A_1 f(x_1) = \frac{h}{2} [f(x_0) + f(x_1)] \quad (8.1.2)$$

Korzystając ze wzoru na błąd interpolacji dostajemy:

$$\begin{aligned}
 E(f) &= I(f) - S(f) = \int_a^b (I(f) - S(f)) df = \\
 &= \frac{1}{2} \int_a^b (x-a)(x-b)f''(\xi) dx = \\
 &= -\frac{1}{12} h^3 f''(\xi^*) \tag{8.1.3}
 \end{aligned}$$

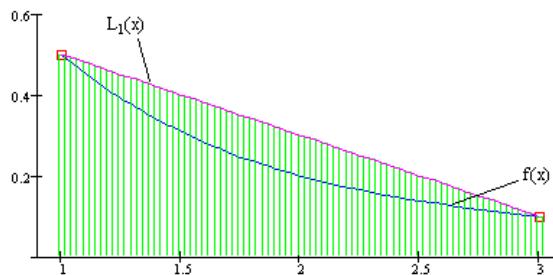
gdzie $\xi, \xi^* \in [a, b]$.

Wzór na przybliżoną wartość całki ma prostą interpretację geometryczną. Pokażemy ją na przykładzie.

Przykład 8.1

Przykład

Obliczymy przybliżoną wartość całki z funkcji $f(x) = \frac{1}{1+x^2}$ w przedziale $[1, 3]$ za pomocą wielomianu interpolacyjnego stopnia $n = 1$. Graficznie, ponieważ dana funkcja jest dodatnia, całka z tej funkcji równa się polu pod krzywą opisaną daną funkcją. Na rysunku 8.1 krzywa jest zaznaczona na niebiesko. Zamiast pola pod krzywą liczymy ze wzoru przybliżonego pole pod prostą łączącą punkty $(a, f(a))$ i $(b, f(b))$ (czyli pod wielomianem interpolacyjnym pierwszego stopnia) zaznaczoną na czerwono. Pole, które otrzymamy jest zakreskowane na zielono. To zielone pole jest polem trapezu, który "leży" na swojej wysokości h . I wzór dlatego nosi nazwę wzoru trapezów, a jak widać, we wzorze jest suma podstawa trapezu dzielona przez 2 i pomnożona przez wysokość.



Rys. 8.1. Interpretacja geometryczna wzoru prostego trapezów.

Po wykonaniu obliczeń dostajemy:

$$S(f) = \frac{h}{2}(f(x_0) + f(x_1)) = \frac{h}{2}(f(a) + f(b)) = \frac{2}{2}\left(\frac{1}{2} + \frac{1}{10}\right) = 0.6$$

wartość przybliżona całki $S(f) = 0.6$, a ponieważ dokładną wartość możemy w tym wypadku podać, bo funkcja pierwotna dla funkcji podcałkowej to $\arctg(x)$, zatem $I(f) = \arctg(b) - \arctg(a) = 0.463648$ (z dokładnością do 6 cyfr po przecinku), to błąd bezwzględny równy się $bl = |I(f) - S(f)| = 0.136352$, i stanowi aż 23%. Widać na rysunku, że wartości przybliżona i dokładna znacznie się różnią (pole pod funkcją i pole pod prostą)

Wzór prosty parabol (Simpsona).

Ponieważ w podanym przykładzie wartość całki obarczona jest dużym błędem, wstawimy zamiast funkcji wielomian interpolacyjny stopnia 2. Ustalmy $n = 2$ i wtedy $h = (b - a)/2$ i mamy trzy węzły $x_0 = a, x_1 = a + h = \frac{a+b}{2}, x_2 = b$.

Wielomian interpolacyjny Lagrange'a jest stopnia 2 i wstawiając za funkcję $f(x)$ wielomian $L_2(x)$ mamy:

$$\begin{aligned}
 I(f) &= \int_a^b f(x)dx \cong \int_a^b L_2(x)dx = \\
 &= \int_a^b \left(\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \right. \\
 &\quad \left. + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f(x_2) \right) dx = \\
 &= f(x_0) \int_a^b \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} dx + \\
 &\quad + f(x_1) \int_a^b \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} dx + f(x_2) \int_a^b \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} dx = \\
 &= \frac{h}{3}f(x_0) + \frac{4h}{3}f(x_1) + \frac{h}{3}f(x_2) = S(f)
 \end{aligned} \tag{8.1.4}$$

Całki w przedostatniej linijce wzoru łatwo obliczyć, bo są to całki oznaczone z wielomianów drugiego stopnia.

W tym wypadku

$$A_0 = \int_a^b \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} dx = \frac{h}{3}$$

$$A_1 = \int_a^b \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} dx = \frac{4h}{3}$$

$$A_2 = \int_a^b \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} dx = \frac{h}{3}$$

Możemy wzór na przybliżoną wartość $I(f)$ zapisać w postaci:

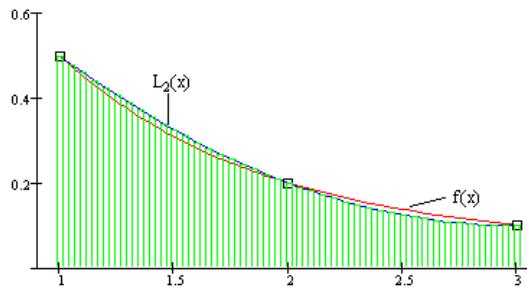
$$I(f) = \int_a^b f(x)dx \cong A_0f(x_0) + A_1f(x_1) + A_2f(x_2) = \frac{h}{3}(f(x_0) + 4f(x_1) + f(x_2)) \tag{8.1.5}$$

Korzystając ze wzoru na błąd interpolacji dostajemy:

$$E(f) = I(f) - S(f) = \int_a^b (I(f) - S(f)) df = -\frac{1}{90}h^5 f^{(4)}(\xi^*) \tag{8.1.6}$$

gdzie $\xi, \xi^* \in [a, b]$.

Powróćmy do przykładu 8.1. Teraz prowadzimy parabolę przez trzy punkty $(a, f(a)), ((a+h), f(a+h))$ i $(b, f(b))$. Na rysunku 8.2 pole zakreskowane na zielono równa się polu pod parabolą (wielomianem interpolacyjnym stopnia 2) narysowaną na niebiesko, funkcja jest narysowana na czerwono.



Rys 8.2. Interpretacja geometryczna wzoru prostego parabol.

Po obliczeniu przybliżonej wartości całki według wzoru parabol

$$\begin{aligned}
 S(f) &= A_0f(x_0) + A_1f(x_1) + A_2f(x_2) = \frac{h}{3}(f(x_0) + 4f(x_1) + f(x_2)) = \\
 &= \frac{1}{3}\left(\frac{1}{2} + \frac{4 \cdot 1}{5} + \frac{1}{10}\right) = \frac{7}{15}
 \end{aligned}$$

otrzymujemy: $S(f) = 0.466667$, błąd bezwzględny $|I(f) - S(f)| = 0.003019$, co stanowi tylko 0.3.

3. Metody złożone trapezów i parabol

W poprzedniej sekcji rozpatrywaliśmy przybliżone całkowanie funkcji za pomocą wielomianów interpolacyjnych 1 i 2 stopnia z równoodległymi węzłami. Można by wyprowadzić również podobne wzory dla wielomianów wyższych stopni, ale okazało się, że lepiej podzielić przedział całkowania na m części i w każdym otrzymanym podprzedziale zastosować wzór prosty trapezów lub parabol, korzystając z tego faktu, że całka po przedziale $[a, b]$ jest sumą całek po otrzymanych podprzedziałach. W ten sposób wyprowadzimy dwa wzory: złożony trapezów i złożony parabol (Simpsona).

Wzór złożony trapezów

Dzielimy przedział $[a, b]$ na m części: $h = \frac{b-a}{m}$, $x_k = a + k \cdot h$, $k = 0, 1, \dots, m$. Otrzymamy m podprzedziałów o długości h . W każdym z podprzedziałów $[x_i, x_{i+1}]$, $i = 0, 1, \dots, m-1$ stosujemy wzór prosty trapezów:

$$\int_{x_i}^{x_{i+1}} f(x) dx \cong \frac{h}{2} (f(x_i) + f(x_{i+1})) \quad (8.2.1)$$

Sumując całki po wszystkich podprzedziałach dostajemy przybliżoną wartość całki w przedziale $[a, b]$:

$$\begin{aligned} I(f) &= \int_a^b f(x) dx = \\ &= \int_a^{x_1} f(x) dx + \dots + \int_{x_i}^{x_{i+1}} f(x) dx + \dots + \int_{x_{m-1}}^b f(x) dx \cong \\ &\cong \frac{h}{2} (f(x_0) + f(x_1)) + \dots + \frac{h}{2} (f(x_i) + f(x_{i+1})) + \dots \\ &\dots + \frac{h}{2} (f(x_{m-1}) + f(x_m)) = \\ &= \frac{h}{2} (f(x_0) + 2(f(x_1) + f(x_2) + \dots + f(x_{m-1})) + f(x_m)) = \\ &= S(f) \end{aligned} \quad (8.2.2)$$

Jeśli przesumujemy błędy po wszystkich podprzedziałach otrzymamy:

$$E(f) = -\frac{(b-a)^3}{12m^2} f''(\xi) \quad , \quad \xi \in [a, b] \quad (8.2.3)$$

Zauważmy, że we wzorze na $S(f)$, w ostatniej linijce, wartości funkcji podcałkowej w skrajnych węzłach są w nawiasie wzięte z mnożnikiem 1, a w pozostałych węzłach z mnożnikiem 2. Prostą interpretację geometryczną tego faktu ilustrujemy na przykładzie, który był przeliczany w poprzedniej sekcji.

Przykład 8.3

Przykład

Obliczymy przybliżoną wartość całki z funkcji $f(x) = \frac{1}{1+x^2}$ w przedziale $[1, 3]$, dzieląc przedział na 4 części. Mamy: $m = 4$, $h = 0.5$, zatem

$$S(f) = \frac{0,5}{2} (f(1) + f(3) + 2(f(1.5) + f(2) + f(2.5))) =$$

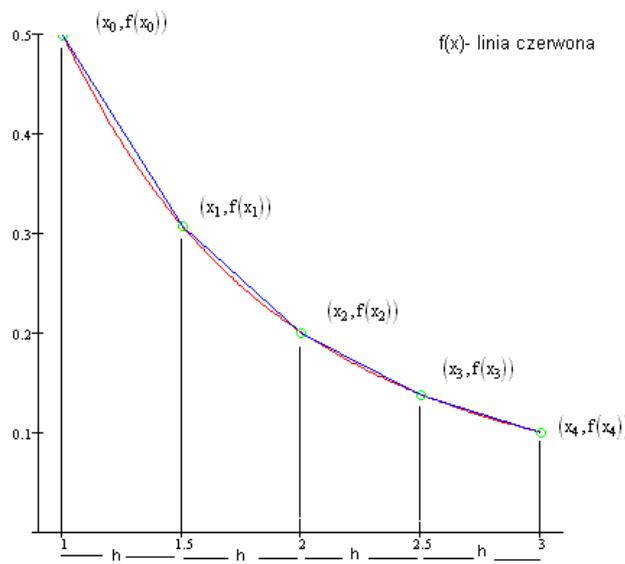
$$= 0.25 \left(\frac{1}{2} + \frac{1}{10} + 2 \cdot \left(\frac{1}{1+(1.5)^2} + \frac{1}{1+2^2} + \frac{1}{1+(2.5)^2} \right) \right) = 0.473$$

Obliczając wartość całki za pomocą funkcji pierwotnej $F(x) = \arctg(x)$, otrzymamy $I(f) = 0.464$, zatem błąd bezwzględny wyniesie 0.009, tzn. 0.9%.

Graficznie: trzeba obliczyć pola czterech trapezów o tej samej wysokości h , pierwszy trapez ma podstawy równe $f(x_0)$ i $f(x_1)$, drugi ma

podstawy $f(x_1)$ i $f(x_2)$, podstawami trzeciego trapezu są $f(x_2)$ i $f(x_3)$ i podstawami czwartego trapezu są $f(x_3)$ i $f(x_4)$.

Jak widać trzy podstawy są wspólne w tych czterech trapezach i dlatego są we wzorze wzięte podwójnie. Skrajne podstawy są uwzględnione tylko raz. Po przesumowaniu pół tych trapezów dostajemy przybliżoną wartość całki $S(f)$.



Rys. 8.3. Interpretacja geometryczna wzoru złożonego trapezów.

Wzór złożony parabol (Simpsona)

Dzielimy przedział $[a, b]$ na m części (ale bierzemy m **parzyste**): $h = \frac{b-a}{m}$, $x_k = a + k \cdot h$, $k = 0, 1, \dots, m$. Otrzymamy m podprzedziałów o długości h , inaczej $m/2$ podprzedziałów o długości $2h$, w każdym z podprzedziałów o długości $2h$: $[x_i, x_{i+2}]$, $i = 0, 1, \dots, m-2$ stosujemy wzór prosty parabol:

$$\int_{x_i}^{x_{i+2}} f(x) dx \cong \frac{h}{3} (f(x_i) + 4f(x_{i+1}) + f(x_{i+2})) \quad (8.2.4)$$

Sumując otrzymane całki po $m/2$ w podprzedziałach otrzymujemy:

$$\begin{aligned}
 I(f) &= \int_a^b f(x) dx = \\
 &= \int_a^{x_2} f(x) dx + \dots + \int_{x_i}^{x_{i+2}} f(x) dx + \dots + \int_{x_m-2}^b f(x) dx \cong \\
 &\cong \frac{h}{3} (f(x_0) + 4f(x_1) + f(x_2)) + \\
 &\quad + \frac{h}{3} (f(x_2) + 4f(x_3) + f(x_4)) + \dots \\
 &\quad + \frac{h}{3} (f(x_i) + 4f(x_{i+1}) + f(x_{i+2})) + \dots \\
 &\quad + \frac{h}{3} (f(x_{m-2}) + 4f(x_{m-1}) + f(x_m)) = \\
 &= \frac{h}{3} (f(x_0) + \\
 &\quad + 2(f(x_2) + f(x_4) + \dots + f(x_{m-2})) + \\
 &\quad + 4(f(x_1) + f(x_3) + \dots + f(x_{m-1})) + \\
 &\quad + f(x_m)) = \\
 &= S(f)
 \end{aligned} \quad (8.2.5)$$

W nawiasie w ostatnim wyrażeniu wartości funkcji w skrajnych węzłach są wzięte z mnożnikiem 1, wartości funkcji w węzłach pozostałych numerach parzystych są z mnożnikiem 2, a w węzłach o numerach nieparzystych z mnożnikiem 4.

Sumując błędy, podane w poprzednim temacie dla wzoru parabol, po $m/2$ podprzedziałach dostajemy:

$$E(f) = -\frac{(b-a)^5}{180m^4} f^{(4)}(\xi^*) \quad \xi^* \in [a, b] \quad (8.2.6)$$

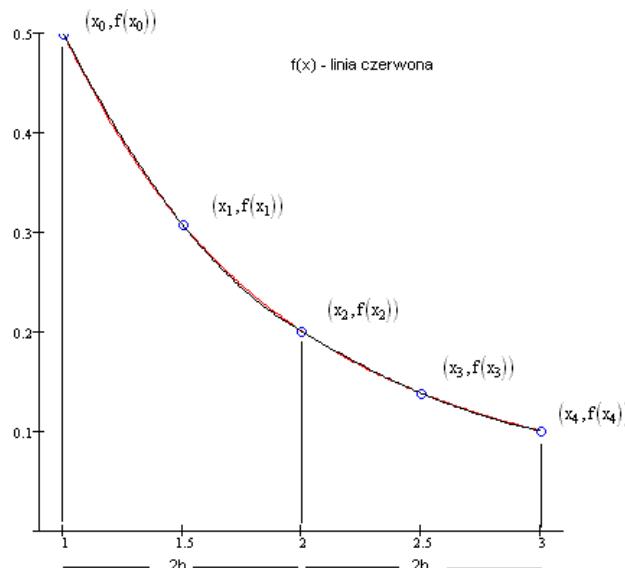
Wróćmy do przykładu 8.1 i obliczmy całkę metodą parabol biorąc też $m = 4, h = 0.5$.

Bierzemy podprzedział $[1, 2]$ i $[2, 3]$, w każdym z nich stosujemy wzór prosty parabol.

Dostajemy:

$$\begin{aligned} S(f) &= \frac{h}{3} \left(f(x_0) + 4f(x_1) + f(x_2) \right) + \frac{h}{3} \left(f(x_2) + 4f(x_3) + f(x_4) \right) = \\ &= \frac{h}{3} (f(1) + f(3) + 2 \cdot f(2) + 4 \cdot (f(1.5) + f(2.5))) = 0.4637 \end{aligned} \quad (1)$$

Korzystając z funkcji pierwotnej, tak jak w przykładzie powyżej dostajemy błąd bezwzględny równy 0.0001, czyli 0.01%. Na rysunku 8.4 są zaznaczone dwie parbole (na czarno), jedna w przedziale $[1, 2]$, druga w przedziale $[2, 3]$, ale błędy są tak małe, że prawie się pokrywają się z funkcją (na czerwono). Przybliżona wartość całki to suma pól pod tymi parabolami.



Rys. 8.4. Interpretacja geometryczna wzoru złożonego parabol.

4. Węzły Legendre'a

W tej sekcji omawiamy kwadratury Gaussa z węzłami Legendre'a. Podajemy również wzory na zależność między ilością podprzedziałów we wzorach złożonych, a dokładnością obliczeń.

Do tej pory obliczaliśmy przybliżoną wartość całki oznaczonej z funkcji $\int_a^b f(x) dx$ zastępując ją wielomianami interpolacyjnymi z równoodległymi węzłami. Ale błąd całkowania zależy od położenia węzłów, tak jak w interpolacji, więc choć węzły równoodległe są wygodne do liczenia, nie zawsze są najlepsze. Okazuje się, że optymalnymi węzłami są pierwiastki pewnych wielomianów, które noszą nazwę wielomianów Legendre'a. Nie będziemy wprowadzać tutaj teorii wielomianów ortogonalnych, tylko podamy wartości tych pierwiastków i wartości związanych z nimi współczynników. Ponieważ pierwiastki wielomianów Legendre'a są w przedziale $[-1, 1]$, aby z nich skorzystać, jako z węzłów, zamienimy naszą całkę po przedziale $[a, b]$ na przedział $[-1, 1]$ za pomocą podstawienia: $x = \frac{b-a}{2}t + \frac{b+a}{2}$, wtedy $d x = \frac{b-a}{2} dt$ i otrzymujemy:

$$\begin{aligned} I(f) &= \int_a^b f(x) dx = \int_{-1}^1 f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right) \cdot \frac{b-a}{2} dt = \\ &= \frac{b-a}{2} \int_{-1}^1 f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right) dt \end{aligned} \quad (8.3.1)$$

Jeśli t_k dla $k = 0, 1, \dots, n$ będą węzłami Legendre'a, to wstawiając za funkcję f wielomian interpolacyjny stopnia n z tymi węzłami otrzymamy wzór na przybliżoną kwadraturę :

$$\begin{aligned} I(f) &= \int_a^b f(x) dx = \int_{-1}^1 f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right) dt \cong \int_{-1}^1 A_k \Phi_k(t) dt = S(f) \end{aligned} \quad (8.3.2)$$

gdzie $A_k = \int_{-1}^1 \Phi_k(x) dx$ nie zależy od funkcji, tylko od węzłów, a w wielomianach $\Phi_k(x)$ też występują węzły t_k .

W każdej książce z metod numerycznych jest podana tablica węzłów i współczynników Legendre'a. Podajemy poniżej tabelkę dla wielomianów interpolacyjnych stopnia $n=1, 2, 3$ i 4 .

n	k	Węzły t_k	Współczynniki A_k
1	0; 1	± 0.577350	1
2	0; 2	± 0.774597	5/9
	1	0	8/9
3	0; 3	± 0.861136	0.347855
	1; 2	± 0.339981	0.652145
	0; 4	± 0.906180	0.236927
4	1; 3	± 0.538469	0.478629
	2	0	0.568889

Przykład 8.4

Przykład

Obliczymy przybliżoną wartość całki z funkcji $f(x) = \frac{1}{1+x^2}$ w przedziale $[1, 3]$ za pomocą wielomianu interpolacyjnego stopnia 1 wykorzystując dwa węzły Legendre'a: $t_0 = -0,577350, t_1 = 0,577350, A_0 = 1, A_1 = 1$.

Dla $n=1$ wzór na $S(f)$ ma postać:

$$\begin{aligned} I(f) &= \int_a^b f(x) dx \cong \frac{b-a}{2} \left[f\left(\frac{t_0 + t_1}{2}\right) + \frac{A_0 f(t_0) + A_1 f(t_1)}{2} \right] \\ &= \frac{1}{2} \left[f\left(\frac{-0,577350 + 0,577350}{2}\right) + \frac{1 \cdot f(-0,577350) + 1 \cdot f(0,577350)}{2} \right] = S(f) \end{aligned}$$

Przeliczając węzły z przedziału $[-1, 1]$ do przedziału $[1, 3]$ otrzymujemy

$$(x_0 = \frac{b-a}{2} t_0 + \frac{b+a}{2} = 1.42265, x_1 = \frac{b-a}{2} t_1 + \frac{b+a}{2} = 2.57735)$$

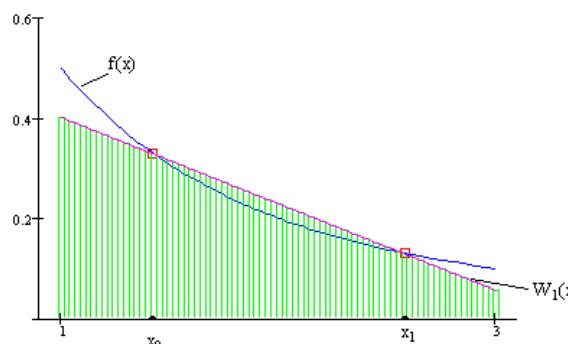
Zatem:

$$\begin{aligned} S(f) &= \frac{1}{2} \left[f\left(\frac{x_0 + x_1}{2}\right) + \frac{A_0 f(x_0) + A_1 f(x_1)}{2} \right] \\ &= \frac{1}{2} \left[f\left(\frac{1.42265 + 2.57735}{2}\right) + \frac{1 \cdot f(1.42265) + 1 \cdot f(2.57735)}{2} \right] = 0,462 \end{aligned}$$

Jak porównamy ten wynik z całką obliczoną za pomocą funkcji pierwotnej $\operatorname{arctg}(x)$ dostajemy błąd równy 0.002 %, czyli 0.2% .

Ten przykład był przeliczany na różne sposoby, proszę porównać wynik otrzymany wzorem prostym trapezów.

Graficznie, zamiast pola pod krzywą liczymy ze wzoru przybliżonego pole pod prostą łączącą punkty $(x_0, f(x_0))$, $(x_1, f(x_1))$ (czyli pod wielomianem interpolacyjnym pierwszego stopnia). Pole, które otrzymamy jest zakreskowane na zielono.



Rys. 8.5. Interpretacja graficzna wzoru z dwoma węzłami Legendre'a.

Jeśli zastosujemy wielomian stopnia 2, będą nam potrzebne trzy węzły i odpowiadające im trzy współczynniki Legendre'a:

$$(t_0 = -0,774597, t_1 = 0, t_2 = 0,774597, A_0 = \frac{5}{9}, A_1 = \frac{8}{9}, A_2 = \frac{5}{9})$$

Wtedy:

$$(I(f) = \int_a^b f(x) dx \cong \frac{b-a}{3} \left[f\left(\frac{t_0 + 2t_1 + t_2}{3}\right) + \frac{A_0 f(t_0) + A_1 f(t_1) + A_2 f(t_2)}{3} \right] = S(f))$$

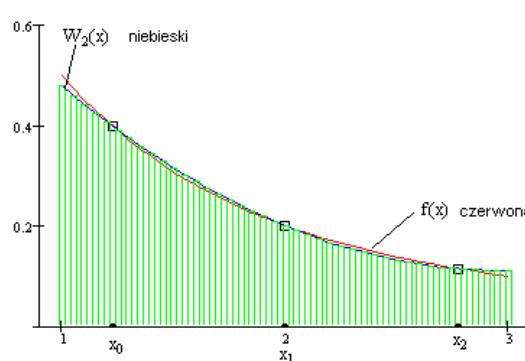
gdzie

$$(x_0 = \frac{b-a}{2} t_0 + \frac{b+a}{2} = 1.225403, x_1 = \frac{b-a}{2} t_1 + \frac{b+a}{2} = 2, x_2 = \frac{b-a}{2} t_2 + \frac{b+a}{2} = 2.774597)$$

Wstawiając te wartości otrzymujemy:

$$(\left. S(f) = \frac{1}{3} \left[f\left(\frac{x_0 + 2x_1 + x_2}{3}\right) + \frac{A_0 f(x_0) + A_1 f(x_1) + A_2 f(x_2)}{3} \right] \right|_{f(x) = \frac{1}{1+x^2}} = 0.4637)$$

a porównując przybliżoną wartość całki z otrzymaną za pomocą funkcji pierwotnej otrzymujemy błąd 0.00008 %, czyli 0.008% . Graficznie, zamiast pola pod krzywą liczymy ze wzoru przybliżonego pole pod parabolą przechodzącą przez punkty $(x_0, f(x_0))$, $(x_1, f(x_1))$, $(x_2, f(x_2))$ (czyli pod wielomianem interpolacyjnym drugiego stopnia). Pole, które otrzymamy jest zakreskowane na zielono.



Rys. 8.6. Interpretacja graficzna wzoru z trzema węzłami Legendre'a.

Porównując wyniki, które otrzymaliśmy w powyższych przykładach z wynikami otrzymwanymi za pomocą węzłów równoodległych widzimy, że w tym wypadku liczona tą metodą wartość całki jest bardzo bliska jej wartości "dokładnej". W tym wypadku węzły Legendre'a dają dużo lepszy wynik. Można również, analogicznie jak w przypadku wzorów trapezów i parabol wyprowadzić wzory złożone oparte na 2 lub 3 węzłach Legendre'a i wtedy dokładność jeszcze się poprawi.

5. Uwagi o dokładności

W podanych przykładach obliczaliśmy całkę z funkcji, która miała funkcję pierwotną, można było określić błąd wyników, przez porównanie wartości przybliżonej z wartością dokładną. Na ogół całkujemy w sposób numeryczny funkcję taką, dla której nie istnieje funkcja pierwotna, albo trudno ją znaleźć, jednak chcemy aby nasze obliczenia nie przekraczały z góry zadanego błędu. Można to zrobić, jeśli oszacujemy w przedziale całkowania maksymalną wartość modułu drugiej pochodnej dla wzoru trapezów i czwartej pochodnej dla wzoru parabol. Wtedy możemy dobrać tak liczbę podprzedziałów, na które dzielimy przedział $\langle [a, b] \rangle$, aby uzyskać żądaną dokładność.

Ponieważ wzór na błąd całkowania dla metody trapezów był następujący:

$$\langle E(f) = -\frac{1}{3} \cdot (b-a)^3 \cdot \int_a^b f''(x) dx, \quad \forall x \in [a, b] \rangle \quad (8.4.1)$$

to oznaczając przez $\langle M_2 = \sup_{x \in [a, b]} |f''(x)| \rangle$, błąd bezwzględny całkowania nie przekroczy wartości $\langle \varepsilon = \left| \frac{1}{3} \cdot (b-a)^3 \cdot M_2 \right| \rangle$, skąd przy danym maksymalnym dopuszczalnym błędzie $\langle \varepsilon \rangle$ możemy obliczyć $\langle m \rangle$, czyli ilość podprzedziałów w metodzie trapezów aby nie został przekroczyony.

Mamy: $\langle m = \sqrt{\frac{1}{3} \cdot (b-a)^3 \cdot M_2} \rangle$, ale ponieważ $\langle m \rangle$ musi być liczbą naturalną, bierzemy za $\langle m \rangle$:

$$\langle m = \left\lceil \sqrt{\frac{1}{3} \cdot (b-a)^3 \cdot M_2} \right\rceil + 1 \rangle \quad (8.4.2)$$

gdzie nawias kwadratowy $\langle [u] \rangle$ oznacza część całkowitą liczby $\langle u \rangle$.

Dla metody parabol wzór na błąd był następujący:

$$\langle E(f) = -\frac{1}{180} \cdot (b-a)^5 \cdot \int_a^b f^{(4)}(x) dx, \quad \forall x \in [a, b] \rangle \quad (8.4.3)$$

Oznaczamy przez $\langle M_4 = \sup_{x \in [a, b]} |f^{(4)}(x)| \rangle$, wtedy błąd bezwzględny nie przekroczy wartości $\langle \varepsilon = \left| \frac{1}{180} \cdot (b-a)^5 \cdot M_4 \right| \rangle$, zatem wzór na $\langle m \rangle$ jest następujący: $\langle m = \sqrt[4]{\frac{1}{180} \cdot (b-a)^5 \cdot M_4} \rangle$, ale $\langle m \rangle$ jest naturalne i musi być **parzyste** zatem:

$$\langle m = 2 \cdot \left\lceil \sqrt[4]{\frac{1}{180} \cdot (b-a)^5 \cdot M_4} \right\rceil + 2 \rangle \quad (8.4.4)$$

To sztuczne podzielenie pierwiastka przez 2, a później pomnożenie znów przez 2, zapewnia parzystość otrzymanego $\langle m \rangle$.

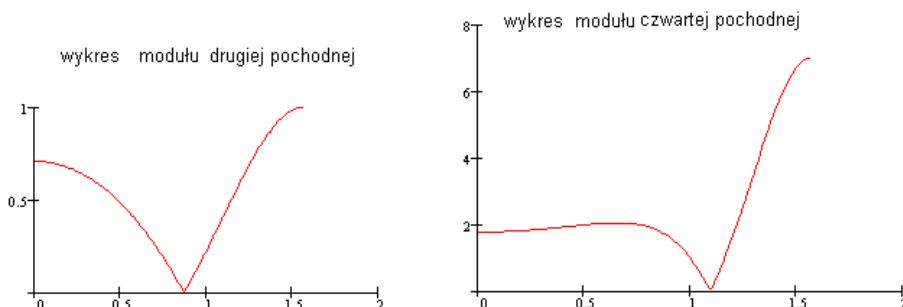
Przykład 8.5

Przykład

Obliczyć z dokładnością $\varepsilon = 10^{-6}$ długość łuku krzywej $y = \sin(x)$ w przedziale $[0, \frac{\pi}{2}]$. Skorzystamy ze wzoru na długość łuku: $I = \int_a^b \sqrt{1+y'(x)^2} dx$, jeśli krzywa jest opisana wzorem $y=y(x)$ w przedziale $[a, b]$.

W naszym przypadku funkcją podcałkową będzie $f(x) = \sqrt{1+\cos^2(x)}$, a ta funkcja nie posiada funkcji pierwotnej, zatem całkę z niej w przedziale $[a, b]$ liczymy numerycznie. Skorzystamy z metody trapezów i parabol.

Oszacujemy z rysunku wartości modułu drugiej i czwartej pochodnej:



Rys. 8.7. Wykresy wartości bezwzględnych pochodnych: drugiej i czwartej.

Przymajemy $M_2=1$, oraz $M_4=7$. Wstawiając do odpowiednich wzorów otrzymujemy:

- stosując metodę złożoną trapezów, aby uzyskać dokładność 10^{-6} , musimy podzielić przedział $[a, b]$ na $m=569$ części i wtedy otrzymamy wynik $I = 1.910099$,
- a stosując metodę złożoną parabol, aby uzyskać ten sam wynik, wystarczy podzielić przedział na $m=26$ części.

Rozdział 9

Strona: [LeIA](#)

Kurs: Metody numeryczne (2025Z)

Książka: Rozdział 9

Wydrukowane przez użytkownika: Kinga Kondraciuk

Data: niedziela, 30 listopada 2025, 14:16

Spis treści

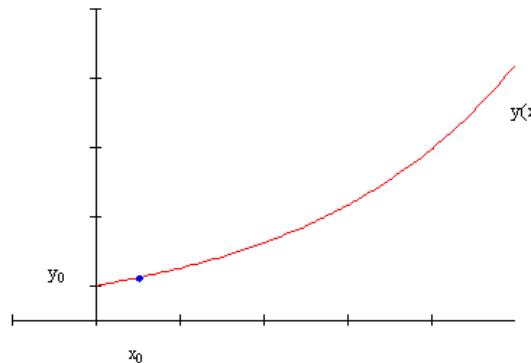
- 1. Równania różniczkowe zwyczajne**
- 2. Metoda prosta Eulera**
- 3. Metoda ulepszona Eulera**
- 4. Metoda klasyczna Rungego-Kutty**
- 5. Metoda trapezów i Heuna**
- 6. Uwagi**

1. Równania różniczkowe zwyczajne

Rozpatrujemy problem początkowy (Cauchy`ego): znaleźć krzywą całkową równania

$$y' = f(x, y)$$

przechodzącą przez punkt (x_0, y_0) . Przy założeniach: funkcja f jest ciągła po x i ma ciągłą pochodną cząstkową po y , istnieje jedyna taka krzywa w pewnym otoczeniu punktu początkowego.



Rys. 9.1. Szukane rozwiązanie przechodzące przez punkt początkowy.

Będziemy rozwiązań tego zagadnienia szukać metodami przybliżonymi.

Zakładamy, że istnieje rozwiązanie podanego problemu, które ma wszystkie pochodne do rzędu $n + 1$ włącznie, i rozwijamy je w szereg Taylora w punkcie x_0 .

$$\begin{aligned} y(x) &= y(x_0) + \frac{y'(x_0)}{1!}(x - x_0) + \frac{y''(x_0)}{2!}(x - x_0)^2 + \frac{y'''(x_0)}{3!}(x - x_0)^3 + \\ &\dots + \frac{y^{(n)}(x_0)}{n!}(x - x_0)^n + R_n(x) \end{aligned} \quad (9.0.1)$$

gdzie

$$R_n(x) = \frac{y^{(n+1)}(x_0 + \theta(x - x_0))}{(n+1)!}(x - x_0)^{n+1} \quad (9.0.2)$$

a θ jest liczbą z otwartego przedziału $(0, 1)$. Przybliżonym rozwiązaniem będzie kilka pierwszych wyrazów tego szeregu (będzie to wielomian stopnia n).

Przykład 9.1

Przykład

Dane jest równanie różniczkowe I rzędu: $y' = 2xy$ z warunkiem początkowym: $y(0) = 1$.

Podane równanie to równanie o zmiennych rozdzielonych i można podać jego dokładne rozwiązanie przechodzące przez punkt $(0, 1)$. Jest to funkcja $r(x) = e^{x^2}$.

Znajdziemy również przybliżone rozwiązanie metodą szeregów potęgowych i ograniczymy się do pięciu wyrazów szeregu. Zobaczmy jaki jest błąd między rozwiązaniem dokładnym a przybliżonym.

Oznaczmy przez $f(x, y) = 2xy = y'$ (z podanego równania) i przez $y_0 = 1$ (warunek początkowy). Obliczymy cztery kolejne pochodne funkcji $y(x)$ w punkcie początkowym $x_0 = 0$. Wartość pierwszej pochodnej wyliczymy bezpośrednio z równania:

$$y_1 = y'(x_0) = f(x_0, y_0) = 0$$

Następne pochodne obliczymy różniczkując pierwszą pochodną czyli funkcję $f(x, y) = f(x, y(x))$ po zmiennej x . Otrzymamy:

$$y_2 = y''(x_0) = \frac{d}{dx} f(x, y)_{(x_0, y_0)} = 2$$

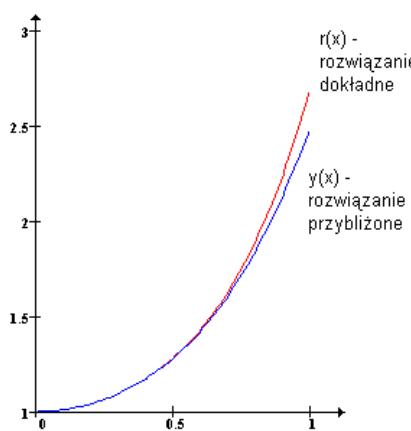
$$y_3 = y'''(x_0) = \frac{d^2}{dx^2} f(x, y)_{(x_0, y_0)} = 0$$

$$y_4 = y^{(4)}(x_0) = \frac{d^3}{dx^3} f(x, y)_{(x_0, y_0)} = 12$$

Zatem rozwiązanie przybliżone w postaci szeregu potęgowego z pięcioma wyrazami jest następujące:

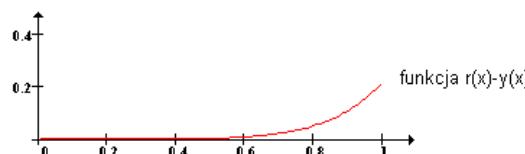
$$y(x) = y_0 + y_1 x + \frac{1}{2} y_2 x^2 + \frac{1}{3!} y_3 x^3 + \frac{1}{4!} y_4 x^4 = 1 + x^2 + \frac{1}{2} x^4$$

Porównamy te dwa rozwiązania na rysunku:



Rys. 9.2. Wykres rozwiązania dokładnego i przybliżonego.

Na następnym rysunku jest przedstawiona funkcja błędu $r(x) - y(x)$ w przedziale $<0, 1>$ i widać, że maksymalny błąd wynosi około 0.2 (w przybliżeniu do trzech cyfr 0.218).



Rys. 9.3. Wykres funkcji błędu.

2. Metoda prosta Eulera

Zakładamy, że istnieje jedyne rozwiązanie problemu początkowego (Cauchy'ego):

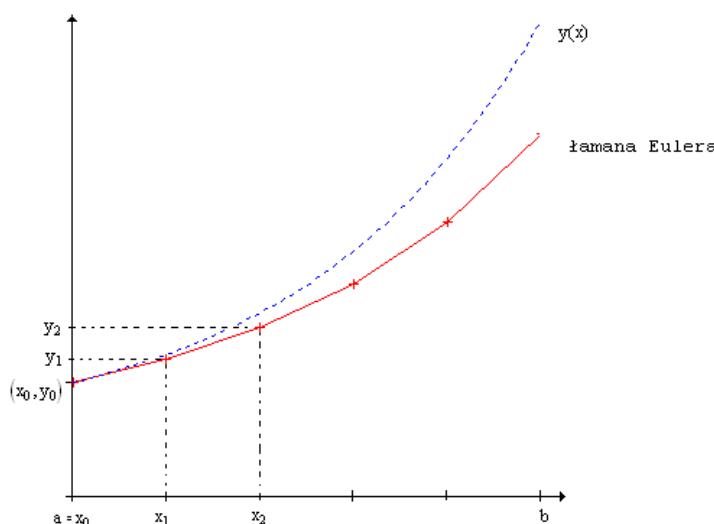
$$\begin{aligned} y' &= f(x, y) \\ y(x_0) &= y_0 \end{aligned} \tag{9.1.1}$$

Znajdziemy przybliżone rozwiązanie metodą prostą Eulera.

Będziemy szukać rozwiązania na przedziale a, b , gdzie $a = x_0$. Podzielimy przedział na n części o długości h . W punkcie początkowym wystawiamy styczną do szukanego rozwiązania. Mamy z równania dokładną wartość współczynnika kierunkowego tej prostej $f(x_0, y_0)$. Zatem szukana styczna ma postać

$$y = y_0 + f(x_0, y_0)(x - x_0)$$

Przecinamy tą styczną z prostą $x = x_1$ i otrzymujemy przybliżoną wartość rozwiązania w punkcie x_1 .



Rys. 9.3. Łamana Eulera.

$$y_1 = y_0 + f(x_0, y_0)(x_1 - x_0) = y_0 + hf(x_0, y_0)$$

Następnie obliczamy z równania współczynnik kierunkowy stycznej w punkcie (x_1, y_1) i prowadzimy przez punkt (x_1, y_1) prostą

$$y = y_1 + f(x_1, y_1)(x - x_1)$$

Przecinamy ją z prostą $x = x_2$ i otrzymujemy wartość y_2

$$y_2 = y_1 + hf(x_1, y_1)$$

Kontynuując to postępowanie otrzymujemy ciąg wartości y_i ze wzoru

$$y_{i+1} = y_i + hf(x_i, y_i) \tag{9.1.2}$$

Dostajemy zatem rozwiązanie przybliżone w postaci tabelki z wartościami (x_i, y_i) . Na rysunku te punkty połączone są łamaną i widać, że za każdym kolejnym krokiem rośnie błąd między dokładnym rozwiązaniem (przerywana niebieska linia) i łamaną.

Przykład 9.2

Przykład

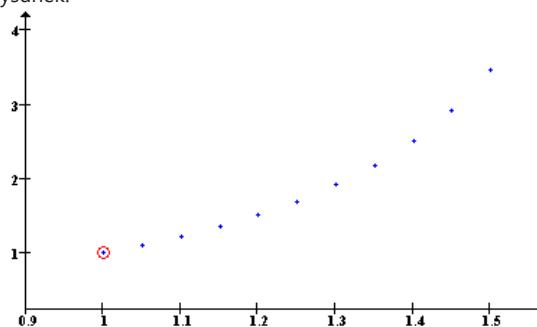
Dane jest równanie $y' = x^2 + y^2$ i warunek początkowy $y(1) = 1$. Będziemy szukać przybliżonego rozwiązania w przedziale $< 1, 1.5 >$, $a = 1$, $b = 1.5$. Podzielimy przedział $< a, b >$ na $n = 10$ części.

$$h = \frac{b-a}{n}, \quad i = 0, \dots n, \quad x_i = a + ih$$

Korzystając ze wzoru (9.2) otrzymamy rozwiązanie w postaci tabelki:

$x_i =$	$y_i =$
1	1
1.05	1.1
1.1	1.216
1.15	1.35
1.2	1.507
1.25	1.693
1.3	1.914
1.35	2.182
1.4	2.511
1.45	2.924
1.5	3.457

Ilustrację graficzną rozwiązania przedstawia rysunek:



Rys. 9.4. Graficzne przedstawienie rozwiązania.

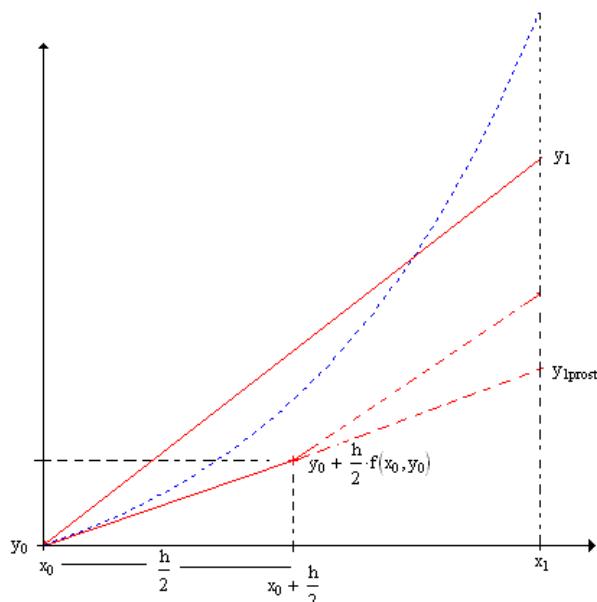
3. Metoda ulepszona Eulera

Stosując identyczne założenia jak w metodzie Eulera będziemy szukać rozwiązań na przedziale $a < b$, gdzie $a = x_0$. Podzielimy przedział na n części o długości h . Punkty podziału: $x_i = a + ih$ gdzie $i = 0, 1, \dots, n$. Wartości funkcji będącej rozwiązaniem danego zagadnienia będziemy liczyć ze wzoru:

$$y_n = y_{n-1} + h f\left(x_{n-1} + \frac{h}{2}, y_{n-1} + \frac{h}{2} f(x_{n-1}, y_{n-1})\right) \quad (9.2.1)$$

Dostajemy rozwiązanie w postaci tabelki, w której są wartości (x_i, y_i) gdzie $i = 0, 1, 2, \dots, n$. Wzór wyjaśnimy na rysunku dla pierwszego kroku:

$$y_1 = y_0 + h f\left(x_0 + \frac{h}{2}, y_0 + \frac{h}{2} f(x_0, y_0)\right)$$



Rys. 9.5. Interpretacja graficzna pierwszego kroku.

Idea zmodyfikowanego wzoru polega na tym, że będziemy "posuwać" się wzdłuż prostej stycznej do wykresu nie w punkcie (x_0, y_0) , tylko wzdłuż prostej o współczynniku kierunkowym równym współczynnikowi stycznej do krzywej w punkcie oddalonym od x_0 o $h/2$.

Przykład 9.3

Przykład

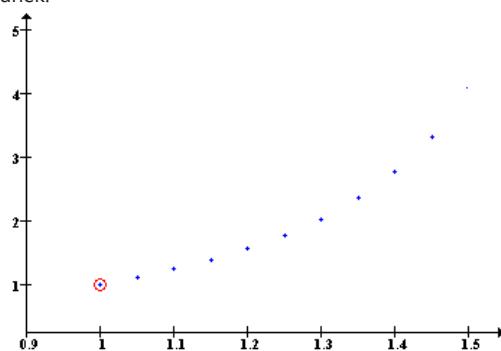
Dane jest równanie (to samo co w przykładzie 9.1): $y' = x^2 + y^2$ i warunek początkowy $y(1) = 1$. Będziemy szukać przybliżonego rozwiązania w przedziale $1 < 1.5$, $a = 1$, $b = 1.5$. Podzielimy przedział $a < b$ na $n = 10$ części.

$$h = \frac{b-a}{n}, \quad i = 0, \dots, n, \quad x_i = a + ih$$

Korzystając ze wzoru (9.2.1) otrzymamy rozwiązanie w postaci tabelki:

$x_i =$	$y_i =$
1	1
1.05	1.108
1.1	1.233
1.15	1.381
1.2	1.557
1.25	1.769
1.3	2.028
1.35	2.352
1.4	2.768
1.45	3.323
1.5	4.098

Ilustrację graficzną rozwiązania przedstawia rysunek:



Rys. 9.6. Graficzne przedstawienie rozwiązania.

4. Metoda klasyczna Rungego-Kutty

Jak się okazuje metoda ulepszona wykazuje większą dokładność od metody prostej. Łatwo jest to uzasadnić intuicyjnie. Metoda prosta zakłada, że wartość pochodnej definiującej prędkość zmian zmiennej y jest stała w przedziale $(x_i, x_i + h)$ i jest równa pochodnej z początku przedziału. W metodzie ulepszonej na początku obliczamy prognozowaną wartość $y_{i+1/2}$ w połowie przedziału $(x_i, x_i + h)$ i dla tej chwili w połowie obliczamy wartość prognozowanej pochodnej. Następnie tą pochodną używamy jako obowiązującą w całym przedziale. Łatwo zauważyc, że w większości przypadków, wartość w połowie przedziału jest bliższa rzeczywistej wartości średniej niż wartość z początku przedziału.

Metody Rungego-Kutty idą o krok dalej. Średnią wartość pochodnej nad przedziałem $(x_i, x_i + h)$ przybliżają za pomocą różnych strategii wykorzystania prognozowanych wartości w różnych punktach tego przedziału. Najbardziej popularną metodą jest klasyczna metoda Rungego-Kutty czwartego stopnia.

Przyjmijmy, że wartość wyrażenia na pochodną w różnych punktach będziemy oznaczać literą k . W skrócie metodę Rungego-Kutta 4-tego stopnia można opisać:

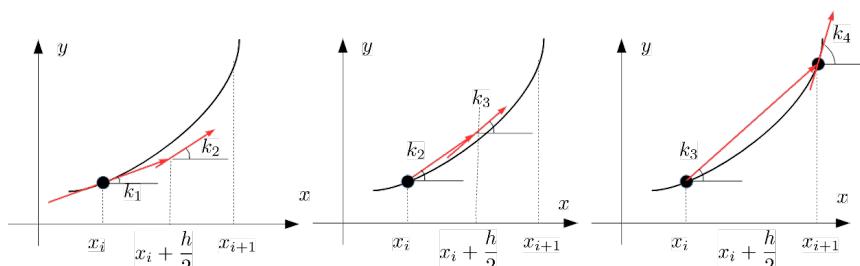
1. Oblicz wartość pochodnej na początku przedziału (k_1).
 2. Używając k_1 oblicz prognozowaną wartość funkcji w połowie przedziału $t_i + h/2$, dla tej wartości oblicz pochodną w połowie przedziału (k_2).
 3. Używając k_2 oblicz jeszcze raz prognozowaną wartość $y_{i+1/2}$ w połowie przedziału i dla niej wyznacz wartość pochodnej (k_3).
 4. Następnie używając k_3 oblicz prognozowaną wartość funkcji na końcu przedziału ($t_i + h$) i dla niej wyznacz wartość pochodnej (k_4).
 5. Ostatecznie średnia wartość pochodnej nad całym przedziałem jest przybliżona za pomocą wzoru na średnią ważoną:
- $$k_{średnie} = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$
6. A wzór na wartość poszukiwanej funkcji w kolejnej chwili.

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (9.3.1)$$

Podsumowując metoda Rungego-Kutty może być zapisana w następującej postaci:

$$\begin{aligned} k_1 &= f(x_i, y_i) \\ k_2 &= f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1\right) \\ k_3 &= f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2\right) \\ k_4 &= f(x_i + h, y_i + hk_3) \\ y_{i+1} &= y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \quad (9.3.2)$$

gdzie $f(x, y)$ to wyrażenie definiujące wartość pochodnej. Na rysunku 9.7 przedstawiona została interpretacja graficzna



Rys. 9.7. Interpretacja graficzna metody klasycznej Rungego-Kutty 4-tego stopnia.

Przykład 9.4

Przykład

Napisz skrypt w MATLABie, który wyznaczy rozwiązanie układu równań różniczkowych zwyczajnych:

$$\begin{aligned}\frac{dy_1}{dt} &= y_2 \\ \frac{dy_2}{dt} &= -10y_1\end{aligned}$$

Przyjmij $t \in [0, 5]$, $h = 0.1$.

Rozwiązanie

Skrypt z rozwiązaniem podzielimy na dwie części. Część główną, w której zdefiniujemy wartości początkowe oraz funkcję definiującą wyrażenia na pochodne. Część numeryczną dotyczącą metody Rungego-Kutty czwartego stopnia. Dzięki takiej realizacji fragment dotyczący metody numerycznej będziemy mogli wykorzystywać również dla innych równań - będzie on uniwersalny.

Wyniki w kolejnych iteracjach będziemy przechowywać w kolumnach dwuwierszowej macierzy. Pierwszy wiersz będzie zawierał wartości zmiennej y_1 a drugi y_2 .

```
function rk4_funkcja
y0 = [ 5;
       3.1623];
[y, t] = rk4(@f, 5, 0.1, y0);

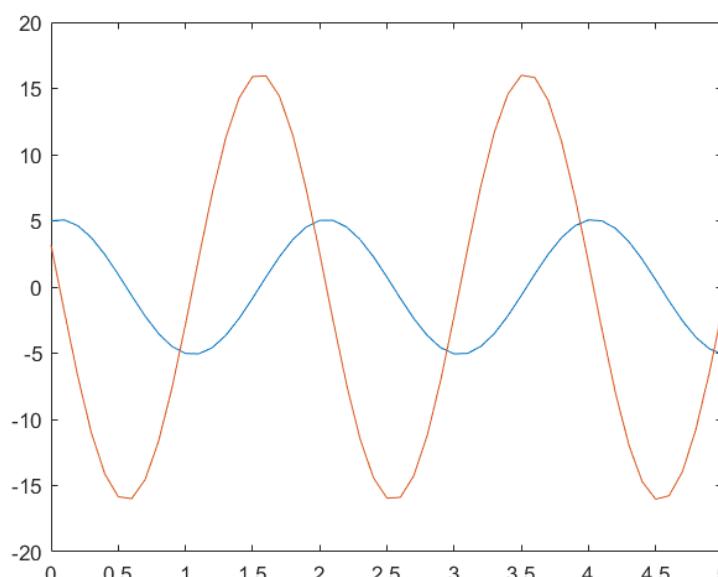
plot(t, y(1,:), t, y(2,:));

end

function [y, t] = rk4(f, t_max, h, y0)
t = 0:h:t_max;
y = zeros(size(y0,1), length(t));
y(:,1) = y0;
for i=1:length(t)-1
    k1 = f(t(i), y(:,i));
    k2 = f(t(i)+h/2, y(:,i) + h/2*k1);           k3 = f(t(i)+h/2, y(:,i) + h/2*k2);
    k4 = f(t(i)+h, y(:,i) + h*k3);
    y(:,i+1) = y(:,i) + h/6*(k1+2*k2+2*k3+k4);
end
end

function dy = f(t, y)
a = 10;
dy = [y(2)
      -a*y(1)];
end
```

Po uruchomieniu program powinien wygenerować rysunek z wykresem rozwiązania.



Rys. 9.8. Przebiegi y_1 i y_2 rozwiązania przykładu.

5. Metoda trapezów i Heuna

Inną kategorią metod rozwiązywania równań różniczkowych zwyczajnych są **metody niejawne**. Charakteryzują się one tym, że wartość pochodnej obliczana jest w punktach, które zamierzamy dopiero obliczyć. Metody niejawne nie precyzuje skąd weźmiemy wartości, które dopiero chcemy obliczyć. Przykładem takiej metody jest metoda trapezów, zgodnie z którą kolejna wartość oblicza się z wzoru:

$$y_{i+1} = y_i + h \cdot \left[\frac{f(t_i, y_i) + f(t_{i+1}, y_{i+1})}{2} \right] \quad (9.4.1)$$

We wzorze tym widzimy drugi składnik licznika ułamka: $f(t_{i+1}, y_{i+1})$, który dotyczy wartości pochodnej w punkcie, który chcemy dopiero obliczyć y_{i+1} .

Są dwa zasadnicze podejścia do rozwiązywania tego problemu. Pierwsze, algebraiczne, polega na wstawieniu do wzoru (9.4.1) wyrażenia algebraicznego definiującego pochodną i przekształceniu względem zmiennej y_{i+1} .

Przykład 9.5

Przykład

Założymy równanie

$$\frac{dy}{dt} = -10y + t$$

stosując sformułowanie niejawne, po prostu podstawiamy wyrażenie na pochodną w miejsce $f(t_{i+1}, y_{i+1})$, otrzymamy zatem:

$$\begin{aligned} y_{i+1} &= y_i + h \cdot \left[\frac{f(t_i, y_i) + f(t_{i+1}, y_{i+1})}{2} \right] \\ y_{i+1} &= y_i + h \cdot \left[\frac{-10y_i + t_i - 10y_{i+1} + t_{i+1}}{2} \right] \\ y_{i+1} + \frac{h \cdot 10y_{i+1}}{2} &= y_i + h \cdot \left[\frac{-10y_i + t_i + t_{i+1}}{2} \right] \\ y_{i+1} &= \frac{1}{1 + 5h} \left[y_i + h \cdot \left[\frac{-10y_i + t_i + t_{i+1}}{2} \right] \right] \end{aligned}$$

Rozwiązanie znajdujemy rekurencyjnie wyznaczając kolejne wartości y_{i+1} .

Drugim podejściem do metod niejawnych jest zastosowanie strategii predyktor-korektor. W metodzie tej, w miejsce wartości y_{i+1} w wyrażeniu na pochodną wstawiamy wartość prognozowaną za pomocą dowolnej metody jawnej. Przykładem takiej metody może być nawet jawna metoda prosta Eulera. W takim przypadku w pierwszym kroku wyznaczamy wartość prognozowaną, którą oznaczmy $y_{i+1}^{(0)}$:

$$y_{i+1}^{(0)} = y_i + hf(x_i, y_i)$$

Następnie wykorzystujemy tą wartość we wzorze na metodę trapezów:

$$y_{i+1} = y_i + h \cdot \left[\frac{f(t_i, y_i) + f(t_{i+1}, y_{i+1}^{(0)})}{2} \right]$$

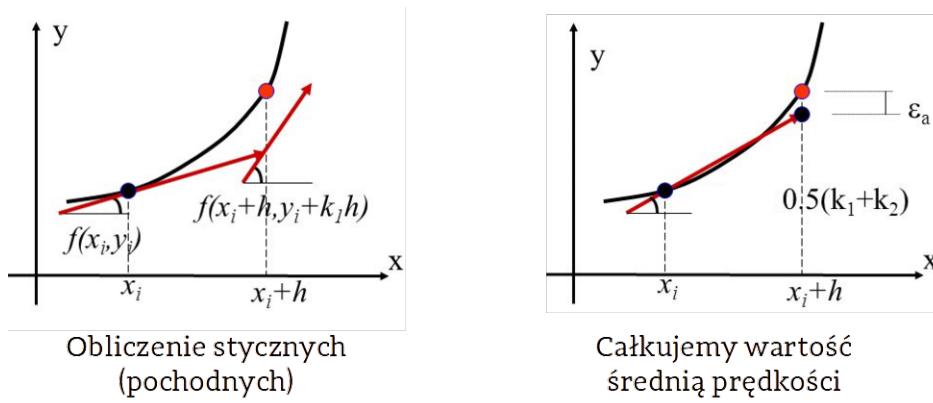
Ostatecznie wzór metody trapezów z wykorzystaniem strategii predyktor-korektor oraz metody jawnej Eulera jako predyktora, przyjmuje postać:

$$\begin{aligned}
 k_1 &= f(t_i, y_i) \\
 k_2 &= f(t_{i+1}, y_i + h k_1) \\
 y_{i+1} &= y_i + h \cdot \left[\frac{k_1 + k_2}{2} \right]
 \end{aligned} \tag{9.4.2}$$

Wzór (9.4.2) nazywany jest **metodą Heuna**. Spróbujmy opisać tą metodę słownie:

- najpierw obliczam wartość prognozowaną na końcu przedziału metodą jawną Eulera,
- potem obliczam średnią arytmetyczną pochodnych na początku przedziału i na końcu przedziału, przy czym pochodna na końcu przedziału obliczona jest dla wartości prognozowanej $y_{i+1}^{(0)}$.

Na rysunku 9.9 przedstawiona została interpretacja graficzna metody.



Rys. 9.9. Interpretacja graficzna metody Heuna.

6. Uwagi

W technice, fizyczne metody rozwiązywania zagadnień początkowych (równań różniczkowych zwyczajnych) używane są do poszukiwania zmian stanu układów dynamicznych, które zbudowane są z wielu, powiązanych ze sobą zmiennymi nazywanymi zmiennymi stanu. W przypadku rozpatrywanych wcześniej metod przedstawione wzory dotyczyły równań z jedną zmienną. Przyjrzyjmy się jak możemy wykorzystać je do rozwiązania zagadnień z wieloma zmiennymi. Rozważmy zatem nie pojedyncze równanie różniczkowe zwyczajne, ale układ takich równań, gdzie zmienne są ze sobą powiązane. Układ taki możemy zapisać:

$$\begin{aligned} \frac{dy_1}{dx} &= f_1(x, y_1, y_2, \dots, y_n) \\ \vdots & \\ \frac{dy_n}{dx} &= f_n(x, y_1, y_2, \dots, y_n) \end{aligned} \quad (9.5.1)$$

lub w sposób skrócony:

$$\frac{dY}{dx} = F(x, Y), \text{ notag}$$

gdzie: $\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, F = \begin{bmatrix} f_1(x, y_1, \dots, y_n) \\ \vdots \\ f_n(x, y_1, \dots, y_n) \end{bmatrix}$

W takim przypadku możemy analogicznie wykorzystywać przedstawione wcześniej metody zastępując jedynie zapis algorytmów przy pomocy jednej zmiennej (y_i) wektorem zmiennych (Y_i) . Tak, **metoda prosta Eulera** będzie postaci:

$$Y_{i+1} = Y_i + h \cdot F(t_i, Y_i), \text{ notag}$$

lub w pełnym zapisie:

$$\begin{pmatrix} Y_{i+1} \\ \vdots \\ Y_n \end{pmatrix} = \begin{bmatrix} y_1(t_{i+1}) \\ \vdots \\ y_n(t_{i+1}) \end{bmatrix} = \begin{bmatrix} y_1(t_i) \\ \vdots \\ y_n(t_i) \end{bmatrix} + h \cdot \begin{bmatrix} f_1(x, y_1(t_i), \dots, y_n(t_i)) \\ \vdots \\ f_n(x, y_1(t_i), \dots, y_n(t_i)) \end{bmatrix}, \text{ notag}$$

Metoda ulepszona Eulera, aby uprościć zapis obliczona zostanie dwukrotkowo. Najpierw wyznaczymy prognozowaną wartość zmiennych (y_k) w połowie przedziału: $(t_{i+1/2})$:

$$\begin{pmatrix} Y_{i+1/2} \\ \vdots \\ Y_n \end{pmatrix} = \begin{bmatrix} y_1(t_{i+1/2}) \\ \vdots \\ y_n(t_{i+1/2}) \end{bmatrix} = \begin{bmatrix} y_1(t_i) \\ \vdots \\ y_n(t_i) \end{bmatrix} + \begin{bmatrix} f_1(x, y_1(t_i), \dots, y_n(t_i)) \\ \vdots \\ f_n(x, y_1(t_i), \dots, y_n(t_i)) \end{bmatrix} \cdot h, \text{ notag}$$

a następnie wstawimy tą wartość do ostatecznego wzoru **metody ulepszonej Eulera**:

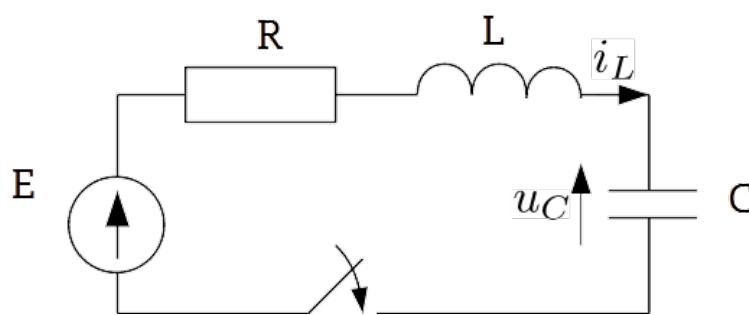
$$\begin{pmatrix} Y_{i+1} \\ \vdots \\ Y_n \end{pmatrix} = \begin{bmatrix} y_1(t_{i+1}) \\ \vdots \\ y_n(t_{i+1}) \end{bmatrix} = \begin{bmatrix} y_1(t_i) \\ \vdots \\ y_n(t_i) \end{bmatrix} + h \cdot \begin{bmatrix} f_1(x, y_1(t_{i+1/2}), \dots, y_n(t_{i+1/2})) \\ \vdots \\ f_n(x, y_1(t_{i+1/2}), \dots, y_n(t_{i+1/2})) \end{bmatrix}, \text{ notag}$$

Przykład 9.6

Przykład

Znajdź przebieg prądu płynącego przez cewkę i napięcia na kondensatorze w szeregowym obwodzie RLC z wymuszeniem napięciowym o wartości (1 [V]) od momentułączenia obwodu do czasu (0.5 [s]) . Przyjmij parametry: $(E=1 \text{ [V]})$, $(R = 1 \text{ [Omega]})$, $(L=0.1 \text{ [H]})$, $(C = 0.01 \text{ [F]})$. Użyj metody Eulera z krokiem $(h=0.01 \text{ [s]})$.

Przedstawmy schemat obwodu szeregowego RLC:



Rys. 9.10. Schemat szeregowego obwodu RLC.

Rozkład prądów i napięć w czasie w tym obwodzie opisany jest za pomocą równań Kirkchhoffa:

$$\begin{aligned} i_L - C \frac{\text{d}u_C}{\text{d}t} = E - R_i L - \frac{1}{L} \frac{\text{d}(i_L u_C)}{\text{d}t} \end{aligned}$$

gdy wykonamy poniższe podstawienie oraz przekształcimy go algebraicznie

$$\begin{aligned} y_1 &= u_C \\ y_2 &= i_L \end{aligned}$$

to powyższy układ przyjmie postać bardziej "matematyczną":

$$\begin{aligned} \frac{\text{d}y_1}{\text{d}t} &= C y_2 \\ \frac{\text{d}y_2}{\text{d}t} &= \frac{1}{L} (E - R y_2 - y_1) \end{aligned}$$

Zwróćmy uwagę, że jest to układ z dwiema zmiennymi stanu (niewiadomymi funkcjami): (y_1, y_2) .

Z uwagi na założenia techniczne związane z tym, że obserwujemy przebiegi połączenia włącznika, możemy przyjąć, że wartości początkowe napięcia i prądu są równe 0 (czyli na początku nie ma napięcia ani nie płynie prąd):

$$\begin{aligned} y_1(0) &= 0 \\ y_2(0) &= 0 \end{aligned}$$

Do rozwiązania zadania użyjemy metody Eulera. W celu ilustracji procesu obliczeń wyznaczymy dwa kolejne kroki algebraicznie a następnie przedstawimy program, który znajduje rozwiązanie.

Przypomnijmy wzór Eulera, w którym użyjemy notacji wektorowej dla wektora zmiennych stanu \mathbf{Y} oraz wyrażeń algebraicznych definiujących pochodne $\mathbf{F}(\dots)$:

$$\mathbf{Y}_{i+1} = \mathbf{Y}_i + h \cdot \mathbf{F}(t_i, \mathbf{Y}_i)$$

Wprowadźmy oznaczenie, które w indeksie dolnym będzie zawierało numer zmiennej stanu oraz numer iteracji $i+1$ będzie reprezentowany przez chwilę czasową: t_{i+1}

$$\mathbf{Y}_{i+1} = \begin{bmatrix} y_1(t_{i+1}) \\ y_2(t_{i+1}) \end{bmatrix}$$

Przy takim założeniu, dla pierwszego kroku możemy napisać:

$$\begin{aligned} y_1(t_1) &= y_1(t_0) + h \cdot \frac{1}{C} (E - R y_2(t_0) - y_1(t_0)) \\ &= y_1(t_0) + h \cdot \frac{1}{C} (E - R y_2(t_0) - y_1(t_0)) \end{aligned}$$

Po podstawieniu wartości liczbowych otrzymamy:

$$\begin{aligned} y_1(0.01) &= y_1(0) + h \cdot \frac{1}{C} (E - R y_2(0) - y_1(0)) \\ &= 0 + 0.01 \cdot \frac{1}{C} (E - R y_2(0) - y_1(0)) \end{aligned}$$

Dla drugiego kroku otrzymamy:

$$\begin{aligned} y_1(t_2) &= y_1(t_1) + h \cdot \frac{1}{C} (E - R y_2(t_1) - y_1(t_1)) \\ &= y_1(t_1) + h \cdot \frac{1}{C} (E - R y_2(t_1) - y_1(t_1)) \end{aligned}$$

oraz po podstawieniu wartości liczbowych:

$$\begin{aligned} y_1(0.02) &= y_1(0.01) + h \cdot \frac{1}{C} (E - R y_2(0.01) - y_1(0.01)) \\ &= 0.1 + 0.01 \cdot \frac{1}{C} (E - R y_2(0.01) - y_1(0.01)) \end{aligned}$$

W poniższym programie w MATLABie celowo użyto, krótszej wartości kroku całkowania h . Wynika to z tego, że przy zbyt długim kroku (tym z treści zadania) prosta metoda Eulera powoduje bardzo dużą kumulację błędów w kolejnych iteracjach i rozwiązanie o ile jest poprawne matematycznie to odbiega od oczekiwanej przebiegu technicznie.

Program w MATLABie

```

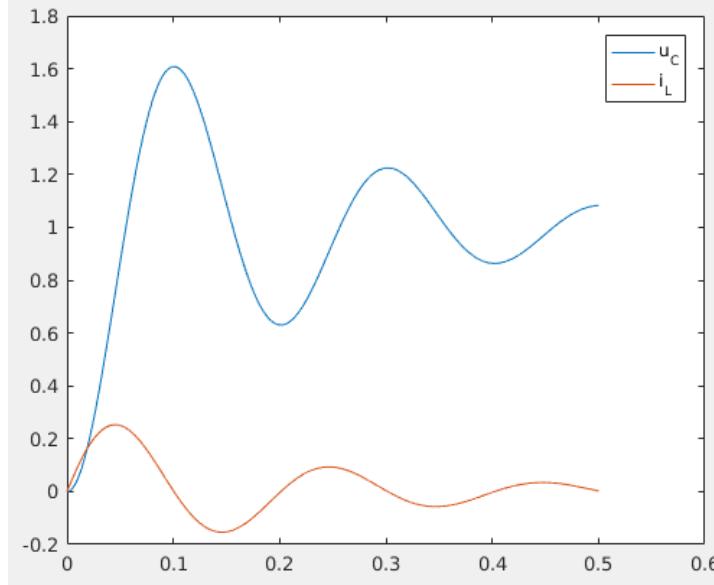
function rlc
    y = [0;
          0];
    T = [0];
    % wartość z treści zadania jest za dłuża,
    % h = 0.01;
    h = 0.0001;
    t = 0; i = 1;

    while t < 0.5
        y(:,i+1) = y(:,i) + h * f(t, y(:,i));
        t = t + h; i = i + 1;
        T(i) = t;
    end
    plot(T, y(1,:), T, y(2,:));
    legend('u_C', 'i_L');
end

function dy = f(t,y)
    E = 1; R = 1; L = 0.1; C = 0.01;
    dy = [1/C*y(2)
          1/L*(E - R*y(2) - y(1)) ];
end

```

W wyniku uruchomienia otrzymujemy przebieg graficzny:



Rys. 9.11. Przebiegi prądu i napięcia w analizowanym szeregowym obwodzie RLC.

Równania wyższego rzędu

Drugą kwestią uzupełniającą temat rozwiązywania równań różniczkowych zwyczajnych są równania wyższego rzędu. Rozważmy równanie (n) -tego rzędu.

$$(\frac{d^n y}{dt^n} + a_{n-1}(t) \frac{d^{n-1}y}{dt^{n-1}} + \dots + a_1(t) \frac{dy}{dt} + a_0(t)y = g(t)) \quad (9.5.2)$$

W ogólności równanie (n) tego rzędu jest zastępowane układem (n) równań 1-go rzędu, stosując ciąg podstawień jak w poniższym przykładzie. Na początku przyjmujemy $y_1=y$. Pierwsze $n-1$ równań wynika z prostego wprowadzenia zmiennych pomocniczych dla kolejnych pochodnych głównej zmiennej stanu:

$$\begin{aligned}
 & \frac{dy_1}{dt} = y_2 \\
 & \frac{dy_2}{dt} = y_3 \\
 & \vdots \\
 & \frac{dy_{n-1}}{dt} = y_n \\
 & \frac{dy_n}{dt} = g(t)
 \end{aligned} \quad (9.5.3)$$

ostatnie równanie różniczkowe wynika z podstawienia wprowadzonych zmiennych pomocniczych $\{y_k\}$ dla $(k=1,2,\dots,n)$ do głównego równania (9.5.1):

$$\left(\frac{dy_n}{dt} + a_{n-1}(t)y_n + \dots + a_1(t)y_2 + a_0(t)y_1 = g(t) \right) \quad (9.5.4)$$

Oczywiście w dalszym etapie przekształcamy je do postaci takiej, że wyrażenie na pochodną względem y_n jest po jego lewej stronie:

$$\left(\frac{dy_n}{dt} = g(t) - a_{n-1}(t)y_n - \dots - a_1(t)y_2 - a_0(t)y_1 \right) \quad (9.5.5)$$

Operacja ta najlepiej będzie zilustrowana na przykładzie.

Przykład 9.7

Przykład

Zastąp równanie trzeciego rzędu układem równań 1-rzędu, który może zostać wykorzystany w metodach numerycznych rozwiązywania równań różniczkowych zwyczajnych.

$$(4\frac{dy}{dt}^3 - 2\frac{dy}{dt}^2 - 5y = 0) \notag$$

Rozwiązanie

Najpierw wprowadzamy zmienne pomocnicze:

$$\begin{aligned} y_1 &= y \\ y_2 &= \frac{dy}{dt} \\ y_3 &= \frac{d^2y}{dt^2} \end{aligned} \notag$$

Następnie wstawiamy je do głównego równania:

$$(4\frac{dy_3}{dt} - 2y_2 - 5y_1 = 0) \notag$$

z czego po przekształceniach otrzymujemy ostateczny układ równań:

$$\begin{aligned} \left(\begin{aligned} \frac{dy_1}{dt} &= y_2 \\ \frac{dy_2}{dt} &= y_3 \\ \frac{dy_3}{dt} &= \frac{1}{4}(2y_2 + 5y_1) \end{aligned} \right) \notag \end{aligned}$$