

METODY NUMERYCZNE

Krzysztof Małczewski

Interpolacja

Sformułowanie zagadnienia interpolacji

3

Wiele zjawisk fizycznych jest opisywanych przez funkcje, których postaci nie znamy, ale potrafimy obliczyć lub zmierzyć wartości tych funkcji oraz ich pochodnych dla określonych wartości argumentu. Na przykład, w określonych chwilach czasowych możemy mierzyć niektóre parametry sygnałów elektrycznych.

Interpolacją nazywamy postępowanie prowadzące do znalezienia wartości pewnej funkcji $f(x)$ w dowolnym punkcie przedziału (x_0, x_n) na podstawie znanych wartości tej funkcji w punktach x_0, x_1, \dots, x_n , zwanych ***węzłami interpolacji*** ($x_0 < x_1 < \dots < x_n$).

Sformułowanie zagadnienia interpolacji

4

Postępowanie prowadzące do znalezienia wartości funkcji $f(x)$ w punkcie leżącym poza przedziałem (x_0, x_n) nazywamy *ekstrapolacją*. Interpolację lub ekstrapolację stosujemy najczęściej w następujących przypadkach:

- a) gdy nie znamy samej funkcji $f(x)$, a tylko jej wartości w pewnych punktach (tak przeważnie bywa w naukach doświadczalnych);

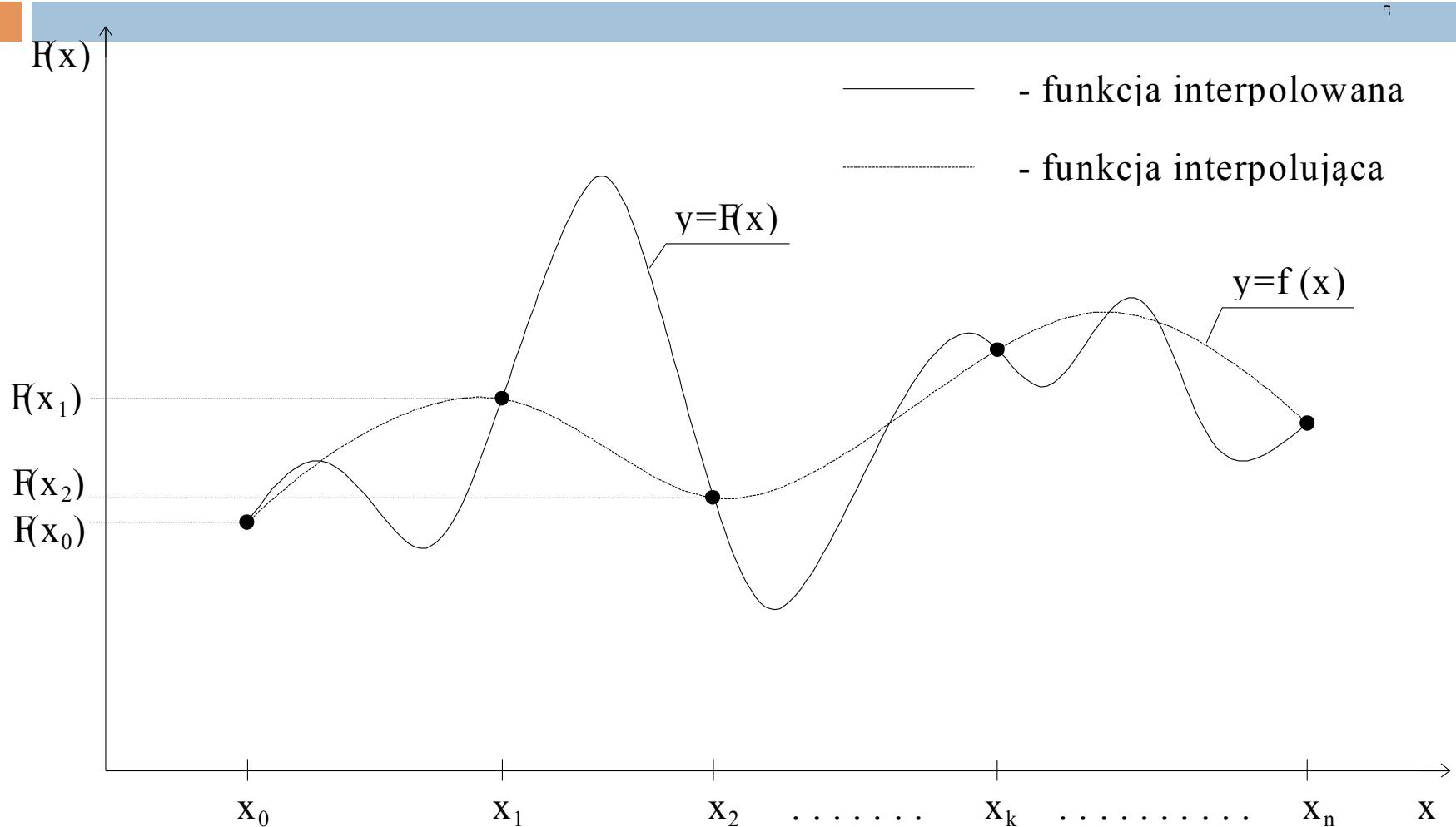
Sformułowanie zagadnienia interpolacji

5

b) gdy obliczanie wartości pewnej funkcji $F(x)$ bezpośrednio z określającego ją wzoru nastręcza zbyt duże trudności rachunkowe; wtedy zastępujemy ją prostszą funkcją $f(x)$, o której zakładamy, że w punktach x_0, x_1, \dots, x_n ma te same wartości co funkcja $F(x)$; w tym przypadku $F(x)$ nazywamy *funkcją interpolowaną*, zaś $f(x)$ *funkcją interpolującą* (rys. 3.1).

Sformułowanie zagadnienia interpolacji

6



Sformułowanie zagadnienia interpolacji

7

Funkcji interpolującej poszukuje się zwykle w pewnej określonej postaci. Najczęściej zakłada się, że jest to wielomian lub funkcja wymierna. Przedmiotem naszych rozważań będzie interpolacja wielomianami algebraicznymi, wielomianami trygonometrycznymi oraz funkcjami sklejonymi. Zastosowanie do obliczeń bardzo szybkich komputerów o dużych pamięciach spowodowało, że interpolacja, będąca niegdyś jedną z podstawowych metod numerycznych, straciła nieco na znaczeniu. Obecnie stosuje się albo proste metody, jak interpolacja liniowa czy kwadratowa, albo też bardziej złożone, wymagające użycia komputera, jak np. interpolacja za pomocą funkcji sklejanych.

Zastosowanie interpolacji

8

Wzory interpolacyjne są punktem wyjścia do wyprowadzenia wielu metod stosowanych w różnych działach metod numerycznych (rozwiązywanie równań, różniczkowanie i całkowanie numeryczne, numeryczne rozwiązywanie równań różniczkowych zwyczajnych)

Interpolacja wielomianowa

9

Wielomianem interpolacyjnym $W_n(x)$ nazywamy wielomian stopnia co najwyżej n , który w punktach x_0, x_1, \dots, x_n przyjmuje wartości y_0, y_1, \dots, y_n .

Twierdzenie 3.1

Istnieje dokładnie jeden wielomian interpolacyjny, który w punktach x_0, x_1, \dots, x_n przyjmuje wartości y_0, y_1, \dots, y_n .

Dowód

Węzły interpolacji x_0, x_1, \dots, x_n mogą być rozmiieszczone w zupełnie dowolny sposób.

Interpolacja wielomianowa

10

Szukany wielomian zapisujemy w postaci:

$$W_n(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n. \quad (3.1)$$

Korzystając z definicji wielomianu interpolacyjnego otrzymujemy układ $n+1$ równań z $n+1$ niewiadomymi współczynnikami a_0, a_1, \dots, a_n :

$$\begin{aligned} a_0 + a_1 x_0 + a_2 x_0^2 + \dots + a_n x_0^n &= y_0, \\ a_0 + a_1 x_1 + a_2 x_1^2 + \dots + a_n x_1^n &= y_1, \\ &\dots \\ a_0 + a_1 x_n + a_2 x_n^2 + \dots + a_n x_n^n &= y_n. \end{aligned} \quad (3.2)$$

Interpolacja wielomianowa

11

Macierz współczynników tego układu jest *macierzą Vandermonde'a* postaci:

$$\mathbf{A} = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix}, \quad (3.3)$$

zaś wyznacznik D macierzy **A** ma postać:

$$D = \begin{vmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{vmatrix}. \quad (3.4)$$

Interpolacja wielomianowa

12

Przy założeniach, że $x_i \neq x_j$ dla $i \neq j$, mamy zawsze $D = \prod_{1 \leq j \leq i \leq n} (x_i - x_j) \neq 0$.

Zatem układ (3.2) ma dokładnie jedno rozwiązanie, a wartości a_i według twierdzenia Cramera są określone wzorem:

$$a_i = \frac{1}{D} \sum_{j=0}^n y_j D_{ij}, \quad (3.5)$$

gdzie D_{ij} ($j = 0, 1, 2, \dots, n$) są kolejnymi dopełnieniami algebraicznymi elementów i -tej kolumny wyznacznika D .

Wielomian interpolacyjny Lagrange'a

13



de LAGRANGE
Joseph Louis
(1736-1813)

Francuski matematyk, astronom i fizyk. Genialny samouk, profesor Szkoły Artyleryjskiej w Turynie, członek a potem prezes (1766- 1787) Akademii Nauk w Berlinie. Od 1772 członek francuskiej Akademii Nauk. W dziele "Mechanika analityczna" (1788) usystematyzował dotychczasową wiedzę z zakresu mechaniki teoretycznej, wprowadzając wiele własnych koncepcji, m.in. sformułował Lagrange'a równania mechaniki, wprowadził tzw. funkcję Lagrange'a, rozwinał teorię mechaniki nieba, podał metodę wyznaczania orbit. W optyce zaproponował wzór wyrażający powiększenie układu optycznego za pomocą odpowiedniej aparatury wejściowej i wyjściowej (tzw. wzór Lagrange'a-Helmholza), sformułował równanie ruchu płynu, matematyczne twierdzenie o wartości średniej oraz zajmował się równaniami różniczkowymi (Lagrange'a równanie różniczkowe). W okresie rewolucji francuskiej zaangażowany w proces modernizacji miar i wag.

Wielomian interpolacyjny Lagrange'a

14

Twierdzenie 3.2

Wielomian $W_n(x)$ postaci (3.6) jest wielomianem interpolacyjnym dla dowolnego wyboru $n+1$ węzłów interpolacji x_0, x_1, \dots, x_n .

$$\begin{aligned} W_n(x) &= y_0 \frac{(x - x_1)(x - x_2)\dots(x - x_n)}{(x_0 - x_1)(x_0 - x_2)\dots(x_0 - x_n)} + \\ &+ y_1 \frac{(x - x_0)(x - x_2)\dots(x - x_n)}{(x_1 - x_0)(x_1 - x_2)\cdot\dots\cdot(x_1 - x_n)} + \dots + \\ &+ y_n \frac{(x - x_0)(x - x_1)\dots(x - x_{n-1})}{(x_n - x_0)(x_n - x_1)\dots(x_n - x_{n-1})} = \sum_{j=0}^n y_j \prod_{\substack{k=0 \\ k \neq j}}^n \frac{(x - x_k)}{(x_j - x_k)} \end{aligned} \quad (3.6)$$

Wielomian interpolacyjny Lagrange'a

15

Przyjmując oznaczenia:

$$\omega_k(x) = (x - x_0)(x - x_1)\dots(x - x_n),$$

$\omega'_n(x_j)$ - wartość pochodnej wielomianu $\omega_n(x)$ w punkcie x_j będącym zerem tego wielomianu,

można wielomian interpolacyjny $W_n(x)$ zapisać w postaci:

$$W_n(x) = \sum_{j=0}^n y_j \frac{\omega_n(x)}{(x - x_j)\omega'_n(x_j)}.$$

Wzór (3.7) nazywamy **wzorem interpolacyjnym Lagrange'a** opartym na węzłach x_0, x_1, \dots, x_n .

Przykład

16

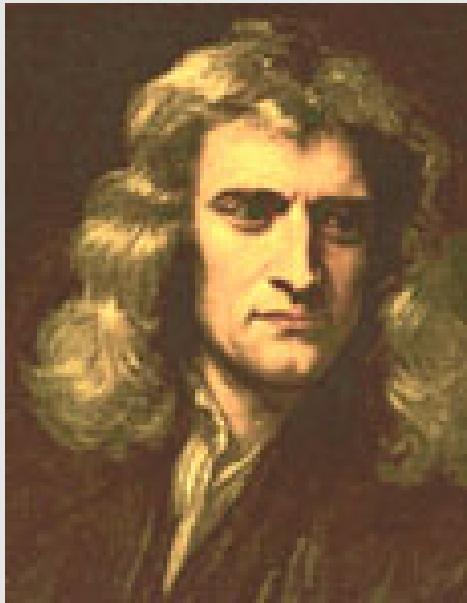
Znaleźć wielomian interpolacyjny, który w punktach -2, -1, 0, 2 przyjmuje odpowiednio wartości 0, 1, 1, 2.

Stosując wzór Lagrange'a (3.6) dla $n = 3$ otrzymujemy:

$$\begin{aligned}W_3(x) &= 0 \frac{(x+1)(x-0)(x-2)}{(-2+1)(-2-0)(-2-2)} + 1 \frac{(x+2)(x-0)(x-2)}{(-1+2)(-1-0)(-1-2)} + \\&+ 1 \frac{(x+2)(x+1)(x-2)}{(0+2)(0+1)(0-2)} + 2 \frac{(x+2)(x+1)(x-0)}{(2+2)(2+1)(2-0)} = \\&= \frac{1}{3}(x^3 - 4x) - \frac{1}{4}(x^3 + x^2 - 4x - 4) + \frac{1}{12}(x^3 + 3x^2 + 2x) = \\&= \frac{1}{6}x^3 - \frac{1}{6}x + 1.\end{aligned}$$

Wzór interpolacyjny Newtona

17



**NEWTON Isaac
(1642 - 1727)**

Urodzony w Anglii, studiował na Uniwersytecie w Cambridge, w wieku 21 lat rozpoczął własne badania, które zrewolucjonizowały świat. Pierwszą opublikowaną pracą Newtona były badania nad naturą światła (prawa odbicia i załamania), dzięki którym zaprojektował i zbudował pierwszy teleskop zwierciadlany. W wieku 29 lat odkrycia te oraz wyniki wielu innych doświadczeń z dziedziny optyki przedstawił Brytyjskiemu Towarzystwu Królewskiemu (*British Royal Society*). Prace w dziedzinie optyki mają jednakże mniejsze znaczenie niż osiągnięcia w dziedzinie czystej matematyki i mechaniki. Jego największym wkładem do matematyki jest odkrycie rachunku całkowego, w dziedzinie mechaniki - poszerzenie nauki o ruchu ciał. W 1687 r. opublikował dzieło: *Matematyczne zasady filozofii przyrody* (*Philosophiae naturalis principia mathematica*), w którym przedstawił prawo ciążenia i prawo ruchu, pokazując również, jak można wykorzystać te prawa do precyzyjnego określenia ruchu planet dookoła Słońca. Wniósł znaczny wkład do termodynamiki i akustyki, ogłosił zasady zachowania pędu i momentu pędu, odkrył słynny dwumian Newtona oraz podał pierwsze przekonujące wyjaśnienie pochodzenia gwiazd.

Wzór interpolacyjny Newtona

18

Wzór interpolacyjny Lagrange'a jest niewygodny do stosowania w przypadku, gdy zmienia się liczba węzłów. Wtedy do wyznaczenia wielomianu określonego stopnia trzeba powtarzać obliczenia od początku. Zatem poprzez dodanie nowych węzłów interpolacyjnych nie można modyfikować wcześniej wyznaczonego wielomianu Lagrange'a. Wzór interpolacyjny Newtona równoważny wielomianowi Lagrange'a usuwa tę niedogodność.

Wzór interpolacyjny Newtona

19

Niech x_0, x_1, \dots, x_n , będą węzłami interpolacji, w których wielomian interpolacyjny przyjmuje odpowiednio wartości y_0, y_1, \dots, y_n . Można wówczas zdefiniować wyrażenia zwane *ilorazami różnicowymi*:

- *pierwszego rzędu*:

$$[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0}$$

$$[x_1, x_2] = \frac{y_2 - y_1}{x_2 - x_1}$$

.....

$$[x_{n-1}, x_n] = \frac{y_n - y_{n-1}}{x_n - x_{n-1}}$$

Wzór interpolacyjny Newtona

20

- drugiego rzędu (analogicznie):

$$[x_0, x_1, x_2] = \frac{[x_1, x_2] - [x_0, x_1]}{x_2 - x_0}$$

$$[x_1, x_2, x_3] = \frac{[x_2, x_3] - [x_1, x_2]}{x_3 - x_1}$$

.....

$$[x_{n-2}, x_{n-1}, x_n] = \frac{[x_{n-1}, x_n] - [x_{n-2}, x_{n-1}]}{x_n - x_{n-2}}$$

- **k-tego rzędu:**

$$[x_i, x_{i+1}, \dots, x_{i+k}] = \frac{[x_{i+1}, x_{i+2}, \dots, x_{i+k}] - [x_i, x_{i+1}, \dots, x_{i+k-1}]}{x_{i+k} - x_i}.$$

dla $k = 1, 2, \dots$ oraz $i = 0, 1, 2, \dots$.

Wzór interpolacyjny Newtona

21

Z ilorazów różnicowych tworzy się zazwyczaj tablicę (tabela 3.1).

Tabela 3.1. Tablica ilorazów różnicowych

x_i	y_i	rzędu 1	rzędu 2	rzędu 3	rzędu 4	rzędu 5
x_0	y_0	$[x_0, x_1]$				
x_1	y_1	$[x_1, x_2]$	$[x_0, x_1, x_2]$	$[x_0, x_1, x_2, x_3]$		
x_2	y_2	$[x_2, x_3]$	$[x_1, x_2, x_3]$	$[x_1, x_2, x_3, x_4]$	$[x_0, x_1, x_2, x_3, x_4]$	\dots
x_3	y_3	$[x_3, x_4]$	$[x_2, x_3, x_4]$	$[x_2, x_3, x_4, x_5]$	$[x_1, x_2, x_3, x_4, x_5]$	
x_4	y_4	$[x_4, x_5]$	$[x_3, x_4, x_5]$			
x_5	y_5					
...	...					

Wzór interpolacyjny Newtona

22

Korzystając z definicji ilorazów różnicowych, otrzymujemy **wzór interpolacyjny Newtona** z ilorazami różnicowymi:

$$y = W_n(x) = y_0 + [x_0, x_1](x - x_0) + [x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + [x_0, x_1, \dots, x_n](x - x_0)(x - x_1)\dots(x - x_{n-1}) \quad (3.9)$$

Wzór interpolacyjny Newtona ma tę własność, że rozszerzenie zadania interpolacji o dodatkowy węzeł sprowadza się do obliczenia i dołączenia do poprzednio wyznaczonego wielomianu dodatkowego składnika.

Interpolacja trygonometryczna

23

Interpolację trygonometryczną stosuje się do wyznaczania funkcji okresowych, często sinusoidalnych, np. funkcji opisujących sygnały elektryczne lub drgania w mechanice.

Zakładamy, że dana jest funkcja f zmiennej rzeczywistej o wartościach zespolonych, okresowa o okresie 2π . Czyli dla każdego x :

$$f(x + 2\pi) = f(x).$$

Jeśli funkcja g byłaby funkcją okresową o okresie t , tzn. dla każdego y :

$$g(y + t) = g(y),$$

Interpolacja trygonometryczna

24

to dokonując podstawienia $x = \frac{2\pi}{t}y$ otrzymamy funkcję okresową:

$$f(x) = g\left(\frac{xt}{2\pi}\right)$$

o okresie 2π . Można więc, bez zmniejszenia ogólności, rozpatrywać tylko funkcje o okresie 2π .

Zadanie interpolacji trygonometrycznej

25

Potrzeba wyznaczania współczynników wielomianu interpolacyjnego funkcji f opartego na dowolnych węzłach pojawia się w praktyce bardzo rzadko. Z tego powodu ograniczymy się tylko do przypadku węzłów równoodległych:

$$x_k = \frac{2k\pi}{n+1}, \quad k = 0, 1, \dots, n. \quad (3.12)$$

Przy tym założeniu rozwiązanie zadania interpolacyjnego upraszcza się w istotny sposób.

Zadanie interpolacji trygonometrycznej

26

Zadanie interpolacji trygonometrycznej polega na znalezieniu dla danej funkcji f wielomianu trygonometrycznego postaci:

$$F_n(x) = \sum_{m=0}^n c_m e^{jmx} = \sum_{m=0}^n c_m (\cos mx + j \sin mx), \quad (3.10)$$

gdzie $j = \sqrt{-1}$. Wielomian ten w $n+1$ różnych punktach $x_k, k=0, 1, \dots, n$, $x_k \neq x_l$ dla $k \neq l$, z przedziału $(0, 2\pi)$ przyjmuje te same wartości, co funkcja f . Tzn. dla $k=0, 1, \dots, n$:

$$F_n(x_k) = f(x_k).$$

Zadanie interpolacji trygonometrycznej

27

Trygonometryczny wielomian interpolacyjny funkcji f , oparty na węzłach (3.12) może być przedstawiony w następującej postaci:

$$F_n(x) = \frac{1}{2}a_0 + \sum_{m=0}^p (a_m \cos mx + b_m \sin mx) + \frac{1}{2}\delta a_{p+1} \cos(p+1)x \quad (3.16)$$

przy czym dla n parzystego $\delta=0$, $p=0.5n$; dla n nieparzystego $\delta=1$, $p=0.5(n-1)$, zaś współczynniki a_m oraz b_m mają postać:

Zadanie interpolacji trygonometrycznej

28

$$\begin{aligned} a_m &= \frac{2}{n+1} \sum_{k=0}^n f(x_k) \cos mx_k, \quad m = 0, 1, 2, \dots, p \\ b_m &= \frac{2}{n+1} \sum_{k=0}^n f(x_k) \sin mx_k, \quad m = 1, 2, \dots, p \end{aligned} \tag{3.17}$$

Wielomian (3.16) jest trygonometrycznym odpowiednikiem wzoru Lagrange'a. Łatwo można sprawdzić, że jeśli $f(x)$ jest **funkcją parzystą**, tzn. $f(x)=f(-x)$, to $b_m=0$ dla każdego m . Jeśli natomiast $f(x)$ jest **funkcją nieparzystą** tzn. $f(x)=f(-x)$, to $a_m = 0$ dla każdego m .

Przykład

29

Znaleźć trygonometryczny wielomian interpolacyjny $F_4(x)$ przechodzący przez węzły interpolacji (3.12) i dla $n = 4$ przybliżający funkcję daną w postaci dyskretnych wartości $f_0 = f_1 = f_2 = 0, f_3 = f_4 = 1$.

Zgodnie z wzorem (3.16) dla $p=0.5, n = 2$ mamy:

$$\begin{aligned} F_4(x) &= \frac{1}{2}a_0 + \sum_{m=0}^2 (a_m \cos mx + b_m \sin mx) = \\ &= 0.5a_0 + a_1 \cos x + b_1 \sin x + a_2 \cos 2x + b_2 \sin 2x. \end{aligned}$$

Wyznaczamy węzły interpolacji:

$$x_0 = 0, \quad x_1 = 2/5\pi, \quad x_2 = 4/5\pi, \quad x_3 = 6/5\pi, \quad x_4 = 8/5\pi,$$

Przykład

30

zaś ze wzorów (3.17) otrzymujemy:

$$a_m = \frac{2}{5} \sum_{k=0}^4 f(x_k) \cos mx_k, \quad m = 0, 1, 2,$$

$$b_m = \frac{2}{5} \sum_{k=0}^4 f(x_k) \sin mx_k, \quad m = 1, 2.$$

Zatem:

$$a_0 = \frac{2}{5} \sum_{k=0}^4 f(x_k) = \frac{2}{5}(1+1) = \frac{4}{5},$$

$$a_1 = \frac{2}{5} \sum_{k=0}^4 f(x_k) \cos x_k = \frac{2}{5}(-0.809 + 0.309) = -0.200,$$

$$a_2 = \frac{2}{5} \sum_{k=0}^4 f(x_k) \cos 2x_k = \frac{2}{5}(0.309 - 0.809) = -0.200,$$

$$b_1 = \frac{2}{5} \sum_{k=0}^4 f(x_k) \sin x_k = \frac{2}{5}(-0.558 - 0.951) = -0.616,$$

$$b_2 = \frac{2}{5} \sum_{k=0}^4 f(x_k) \sin 2x_k = \frac{2}{5}(0.951 - 0.588) = 0.145$$

Przykład

31

oraz

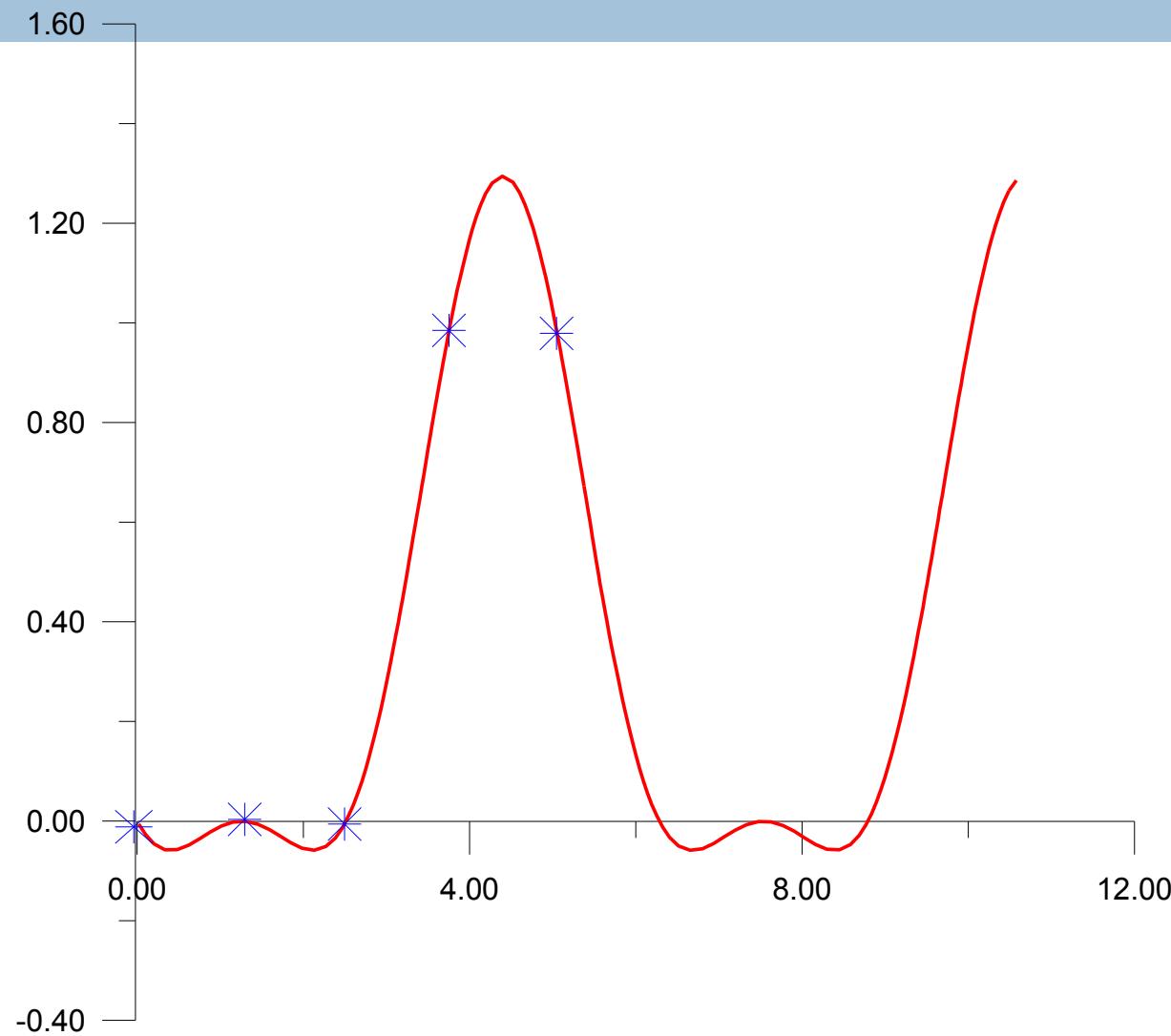
$$F_4(x) = 0.4 a_0 - 0.2 \cos x - 0.616 \sin x - 0.2 \cos 2x + 0.145 \sin 2x.$$

Można sprawdzić, że: $F_4(x_0)=0$, $F_4(x_1)=1.00062$, $F_4(x_2)=0.99997$,
 $F_4(x_3)=0.00002$, $F_4(x_4)=0.00621$.

Ilustracja graficzna przykładu 3.2 (rys. 3.2) pokazuje, że funkcja interpolacyjna jest funkcją okresową o okresie 2π . Wartości funkcji $F_4(x)$ pokrywają się w węzłach interpolacji z wartościami danymi.

Przykład

32



Funkcje sklejane

33

Niech w przedziale $\langle a, b \rangle$ danych będzie $(n+1)$ punktów x_0, x_1, \dots, x_n , przy czym $a = x_0 < x_1 < \dots < x_{n-1} < x_n = b$. Punkty x_i , $i = 1, \dots, n$ określają pewien podział przedziału $\langle a, b \rangle$ na n podprzedziałów. Podział ten oznaczamy symbolem Δ_n .

Funkcję $S(x) = S(x, \Delta_n)$ określoną na przedziale $\langle a, b \rangle$ nazywamy **funkcją sklejaną** stopnia m ($m \geq 1$), jeżeli:

- 1) $S(x)$ jest wielomianem stopnia co najwyżej m na każdym podprzedziale (x_i, x_{i+1}) , $i = 0, 1, \dots, n-1$.
- 2) $S(x)$ i jej pochodne stopnia $1, 2, \dots, m-1$ są ciągłe w rozpatrywanych przedziałach.

Funkcje sklejane

34

Zbiór wszystkich funkcji sklejanych stopnia m o węzłach w punktach x_i oznaczamy $S_m(\Delta_n)$. Jeśli $S(x) \in S_m(\Delta_n)$, to na każdym przedziale (x_i, x_{i+1}) , $i = 0, 1, \dots, n$ funkcja $S(x)$ jest wielomianem stopnia co najwyżej m :

$$S(x) = c_{i,m}x^m + c_{i,m-1}x^{m-1} + \dots + c_{i,1}x + c_{i,0} \text{ dla } x \in (x_i, x_{i+1}). \quad (3.18)$$

Mamy więc $n(m+1)$ dowolnych stałych c_{ij} . Żądanie ciągłości pochodnych rzędu $0, 1, \dots, m-1$ w każdym węźle wewnętrznym x_i daje $m(n-1)$ warunków. Tak więc funkcja $S(x)$ zależy od $n(m+1) - m(n-1) = n+m$ parametrów.

Funkcje sklejane

35

Dowolne funkcje bardzo często przybliża się funkcjami sklejanymi. Wiąże się to z łatwością wyznaczania ich wartości oraz ze zbieżnością dla licznych klas funkcji. W praktyce często stosuje się funkcje sklejane stopnia trzeciego, które dla wielu zagadnień są wystarczająco gładkie, a szybkość ich zbieżności jest zadowalająca.

Interpolacyjne funkcje sklejane stopnia trzeciego

Funkcję $S(x) \in S_3(\Delta_n)$ nazywamy interpolacyjną funkcją sklejaną stopnia trzeciego dla funkcji $f(x)$, jeżeli $S(x_i) = f(x_i) = y_i$, $i = 0, 1, \dots, n$, $n > 1$.

Funkcja $S(x)$ stopnia trzeciego zależy od $n+3$ parametrów. Ponieważ dane są wartości funkcji $f(x_i) = y_i$ w $n+1$ punktach, to na interpolacyjne funkcje sklejane stopnia trzeciego należy nałożyć dwa dodatkowe warunki.

Najczęściej są to następujące warunki dodatkowe:

$$S'(a+0) = \alpha_1, \quad S'(b-0) = \beta_1 \quad (3.19)$$

lub

$$S''(a+0) = \alpha_2, \quad S''(b-0) = \beta_2, \quad (3.20)$$

gdzie $\alpha_1, \beta_1, \alpha_2, \beta_2$ są ustalonymi liczbami rzeczywistymi

Interpolacyjne funkcje sklejane stopnia trzeciego

37

Do wyznaczania interpolacyjnych funkcji sklejanych o przedstawionej wyżej postaci, są niezbędne następujące dane: węzły x_i , wartości drugiej pochodnej funkcji sklejanej $M_j = S''(x)$ oraz wartości funkcji $y_i = f(x_i)$.

Często wygodnie jest przedstawić poszukiwaną funkcję $S(x)$ w postaci kombinacji liniowej elementów bazy przestrzeni $S_m(\Delta_m)$.

Interpolacyjne funkcje sklejane stopnia trzeciego

38

Wyznaczamy bazę przestrzeni funkcji $S(x)$ stopnia trzeciego z węzłami

równoodległymi $x_i = x_0 + ih$, $h = \frac{b-a}{n}$, $i = 0, 1, \dots, n$.

Dodatkowo przez $x_{-3}, x_{-2}, x_{-1}, x_{n+1}, x_{n+2}, x_{n+3}$ oznaczmy punkty $x_i = x_0 + ih$ dla $i = -3, -2, -1, n+1, n+2, n+3$ i określmy funkcje $\Phi_i^3(x)$, $i = -1, 0, 1, \dots, n, n+1$ za pomocą wzoru:

Interpolacyjne funkcje sklejane stopnia trzeciego

39

$$\Phi_j^3(x) = \begin{cases} (x - x_{j-2})^3 & \text{dla } x \in (x_{j-2}, x_{j-1}), \\ h^3 + 3h^2(x - x_{j-1}) + 3h(x - x_{j-1})^2 - 3(x - x_{j-1})^3 & \text{dla } x \in (x_{j-1}, x_j), \\ h^3 + 3h^2(x_{j+1} - x) + 3h(x_{j+1} - x)^2 - 3(x_{j+1} - x)^3 & \text{dla } x \in (x_j, x_{j+1}), \\ (x_{j+2} - x)^3 & \text{dla } x \in (x_{j+1}, x_{j+2}), \\ 0 & \text{dla pozostałych } x \in R. \end{cases} \quad (3.29)$$

Interpolacyjne funkcje sklejane stopnia trzeciego

40

Funkcje te są klasy C^2 . W tablicy 3.2 podane są wartości funkcji $\Phi_j^3(x)$ oraz jej pierwszej i drugiej pochodnej w punktach x_k dla $k = j-2, j-1, j, j+1, j+2$. Poza przedziałem (x_{j-2}, x_{j+2}) funkcja ta jest tożsamościowo równa zeru.

Tabela 3.2. Wartości funkcji $\Phi_j^3(x)$ i jej pochodnych

	x_{j-2}	x_{j-1}	x_j	x_{j+1}	x_{j+2}
$\Phi_j^3(x)$	0	1	4	1	0
$(\Phi_j^3(x))'$	0	$3/h$	0	$-3/h$	0
$(\Phi_j^3(x))''$	0	$3/h^2$	$-12/h^2$	$6/h^2$	0

Interpolacyjne funkcje sklejane stopnia trzeciego

41

Twierdzenie 3.7

Funkcje $\Phi_i^3(x)$, $i = -1, 0, 1, \dots, n+1$, określone na przedziale $\langle a, b \rangle$ stanowią bazę przestrzeni funkcji sklejanych trzeciego stopnia. Każdą funkcję $S(x)$ można przedstawić w postaci kombinacji liniowej:

$$S(x) = \sum_{i=-1}^{n+1} c_i \Phi_i^3(x), \quad a \leq x \leq b, \quad (3.30)$$

gdzie $\Phi_i^3(x)$ są określone wzorem (3.29), c_i są liczbami rzeczywistymi.

Interpolacyjne funkcje sklejane stopnia trzeciego

42

W przypadku węzłów równoodległych szukamy interpolacyjnej funkcji sklejanej w postaci kombinacji liniowej (3.30). Na podstawie tablicy 3.2 można stwierdzić, że stałe c_i muszą spełniać układ $(n+1)$ równań:

$$c_{i-1} + 4c_i + c_{i+1} = y_i, \quad i = 0, 1, \dots, n. \quad (3.31)$$

Jeśli funkcja spełnia dodatkowe warunki (3.19), to dodatkowe dwa równania będą następujące:

$$-c_{-1} + c_1 = \frac{h}{3}\alpha_1 \quad -c_{n-1} + c_{n+1} = \frac{h}{3}\beta_1.$$

Interpolacyjne funkcje sklejane stopnia trzeciego

43

Po wyeliminowaniu z układu współczynników c_{-1} oraz c_{n+1} , pozostałe współczynniki $c_j, j = 0, 1, \dots, n$ rozwinięcia będą rozwiązaniem układu:

$$\begin{bmatrix} 4 & 2 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & \dots & 0 & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 2 & 4 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \dots \\ \dots \\ c_{n-1} \\ c_n \end{bmatrix} = \begin{bmatrix} y_0 + h\alpha_1 / 3 \\ y_1 \\ \dots \\ \dots \\ y_{n-1} \\ y_n + h\beta_1 / 3 \end{bmatrix}, \quad (3.32)$$

którego macierz współczynników jest macierzą trójdziagonalną o dominujących elementach na głównej przekątnej. Układ ma więc jednoznaczne rozwiązanie.

Przykład

Mając dane węzły interpolacyjne jak w tabeli, znaleźć sześcienną funkcję sklejaną.

i	0	1	2	3
x_i	1	3	5	8
y_i	2	4	7	9

Korzystając z rozważań bieżącego rozdziału, otrzymujemy:

a) w przedziale $< x_0 , x_1 > = < 1, 3 >$:

$$S(x) = 2.00 + 0.84(x - 1.00) + 0.00(x - 1.00)^2 + 0.04(x - 1.00)^3$$

b) w przedziale $< x_1 , x_2 > = < 3, 5 >$:

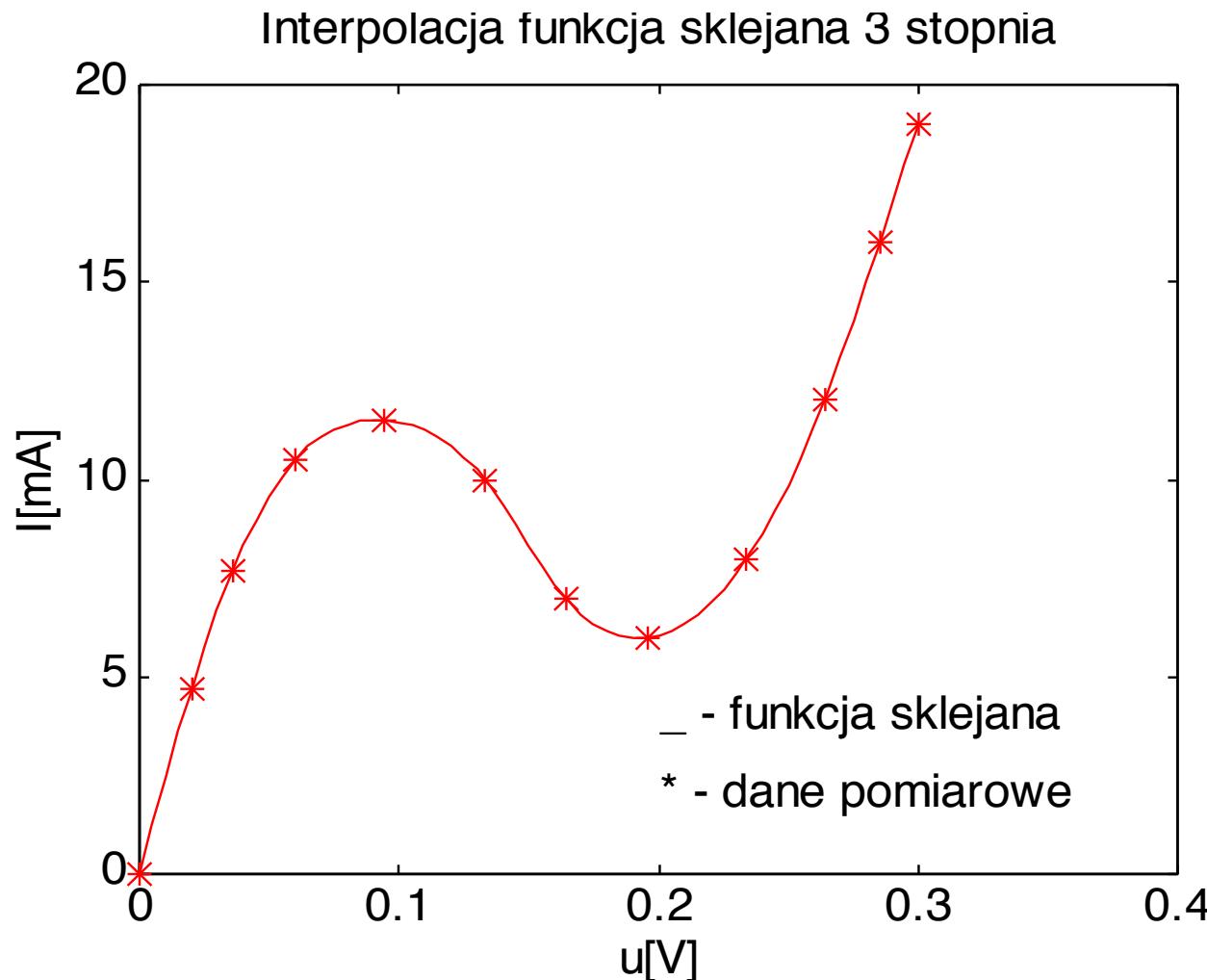
$$S(x) = 4.00 + 1.33(x - 3.00) + 0.25(x - 3.00)^2 - 0.08(x - 3.00)^3$$

c) w przedziale $< x_2 , x_3 > = < 5, 8 >$:

$$S(x) = 7.00 + 1.35(x - 5.00) - 0.24(x - 5.00)^2 + 0.00(x - 5.00)^3$$

Przykład interpolacji funkcją sklejaną

45



PODSTAWY JĘZYKA

MATLAB
(MATrix LABoratory)

METODY NUMERYCZNE

dr inż. Krzysztof Małczewski

Literatura:

Literatura dotycząca MATLAB'a po polsku:

A.Zalewski R.Cegieła, Matlab - *Obliczenia numeryczne i ich zastosowania* , Wyd. Nakom, Poznań 1996.

B.Mrozek, Z.Mrozek: „MATLAB uniwersalne środowisko obliczeń naukowo-technicznych”. PLJ 1996.

J.Brzózka, L.Dorobczyński: „Programowanie w Matlab”, wyd.Mikom 1998.

B.Mrozek, Z.Mrozek: „MATLAB 5.x, Simulink 2.x”., wyd. PLJ 1998.

Z.Wróbel, R.Koprowski: „Przetwarzanie obrazu w programie MATLAB”. Wyd. Uniw. Śl., K-ce 2001.

A.Kamińska, B.Pańczyk: „Matlab - przykłady i zadania” - wyd. Mikom 2002.

Marcin Stachurski: Metody numeryczne w programie Matlab. wyd.MIKOM 2003.

W.Regel: Statystyka matematyczna w Matlab. wyd.MIKOM 2003.

Wiesława Regel: Wykresy i obiekty graficzne w MATLAB. wyd.MIKOM 2003.

R.Jankowski, I.Lubowiecka, W.Witkowski: „Podstawy programowania w języku Matlab”. Gdańsk 2003.

B.Mrozek, Z.Mrozek: MATLAB i Simulink. Poradnik użytkownika. wyd.HELION 2004.

Matlab w Internecie:

<http://www.mathworks.com/>

Lutecki: Przykłady zastosowań MATLABA
<http://www.lutecki.republika.pl/matlab.html>

Ł.Przewoźnik: Metody numeryczne i MATLAB
http://oen.dydaktyka.agh.edu.pl/dydaktyka/matematyka/c_metody_numeryczne/

R.Buczyniski, R.Kasztelanic: Kurs Matlaba
<http://www.igf.fuw.edu.pl/ZOI/Matlab/index.html>

Zestawienie aktualnych propozycji książkowych dostępnych w księgarniach - porównanie produktów

<http://www.nokaut.pl/szukaj/matlab.html>

Działania na wektorach i macierzach

Podstawowe polecenia

Definicja macierzy:

```
>> A=[1,3,6;2,7,8;0,3,9]
```

Odpowiedź:

```
A =  
1 3 6  
2 7 8  
0 3 9
```

Wielkość macierzy:

```
>> size(A)
```

Odpowiedź:

```
ans =  
3 3
```

Transpozycja macierzy:

Odpowiedź:

```
>> A'
```

```
ans =  
1 2 0  
3 7 3  
6 8 9
```

Odwołanie do pojedynczych kolumn czy wierszy macierzy:

>> A(:,3)

Odpowiedź:

ans =

6
8
9

(pierwsza kolumna macierzy A)

>> A(1,:)

Odpowiedź:

ans =

1 3 6

(trzeci wiersz macierzy A)

Działania na wybranych kolumnach i wierszach

np. dodawanie:

>> A(1,:)+A(3,:)

Odpowiedź:

ans =

1 6 15

Dodawanie macierzy:

```
>> B=[3,4,5;6,7,2;8,1,0];  
>> C=A+B
```

Odpowiedź:

```
C =  
4 7 11  
8 14 10  
8 4 9
```

Odejmowanie macierzy:

```
>> C=A-B
```

Odpowiedź:

```
C =  
-2 -1 1  
-4 0 6  
-8 2 9
```

Mnożenie macierzy:

```
>> C=A*B
```

Odpowiedź:

```
C =  
69 31 11  
112 65 24  
90 30 6
```

Funkcje wykorzystywane przy działaniach macierzowych

Symbol	Wyjaśnienie
inv	obliczanie macierzy odwrotnej
det	obliczanie wyznacznika macierzy
rank	obliczanie rzędu macierzy (liczba niezależnych wierszy i kolumn)
cond	wskaźnik uwarunkowania macierzy (oszacowanie z jaką (maksymalnie) dokładnością (do ilu miejsc po przecinku) możemy podać wynik)
eye(n)	definicja macierzy jednostkowej o wymiarach n na n
trace	obliczanie śladu macierzy (suma elementów na głównej przekątnej macierzy)
zeros(n,m)	definicja macierzy wypełnionej zerami
expm	potęgowanie macierzy
eig	obliczanie wartości własnych macierzy
lu	dekompozycja macierzy na macierz trójkątną górną i dolną
chol	rozkład Cholewskiego
qr	Dekompozycja macierzy na ortogonalną i trójkątną górną
\	znacznik używany do rozwiązyania liniowych równań algebraicznych

Funkcje wykorzystywane przy analizie danych

Symbol	Wyjaśnienie
min (max)	zwraca najmniejszy (największy) element wektora lub gdy jest to macierz najmniejszy (największy) element każdej kolumny
sum	zwraca sumę elementów wektora lub gdy jest to macierz sumy elementów każdej kolumny
std	Zwraca odchylenie standardowe wektora lub gdy jest to macierz odchylenia standardowe elementów każdej kolumny
sort	sortuje wektor w porządku wartości rosnących lub gdy jest to macierz sortuje poszczególne elementy każdej kolumny
mean	zwraca średnia arytmetyczną elementów wektora lub gdy jest to macierz średnie arytmetyczne elementów każdej kolumny

Funkcje wykorzystywane przy analizie równań wyższych rzędów

Symbol	Wyjaśnienie
poly	oblicza współczynnik wielomianu charakterystycznego
roots	zwraca pierwiastki równania
polyval	zwraca wartość wielomianu
polyfit	aproksymacja wielomianem

Funkcje wykorzystywane przy analizie nieliniowych równań algebraicznych

Symbol	Wyjaśnienie
fmin	szuka najmniejszej wartości funkcji
fzero	szuka miejsca zerowego funkcji

Funkcje matematyczne

Symbol	Wyjaśnienie
$\sin(x)$	sinus
$\cos(x)$	cosinus
$\tan(x)$	tangens
$\text{asin}(x)$	arcus sinus
$\text{acos}(x)$	arcus cosinus
$\text{atan}(x)$	arcus tangens
$\sinh(x)$	sinus hiperboliczny
$\cosh(x)$	cosinus hiperboliczny
$\tanh(x)$	tangens hiperboliczny
$\text{asinh}(x)$	arcus sinus hiperboliczny
$\text{acosh}(x)$	arcus cosinus hiperboliczny
$\text{atanh}(x)$	arcus tangens hiperboliczny
\sqrt{x}	pierwiastek kwadratowy
e^x	e do x
$\ln(x)$	logarytm naturalny
$\log_2(x)$	logarytm przy podstawie z 2
$\log_{10}(x)$	logarytm przy podstawie z 10

Funkcje związane z obliczeniami w dziedzinie liczb zespolonych

Symbol	Wyjaśnienie
<code>abs(x)</code>	macierz modułów elementów macierzy x
<code>angle(x)</code>	macierz argumentów macierzy x
<code>real(x)</code>	macierz części rzeczywistych elementów macierzy x
<code>imag(x)</code>	macierz części urojonych elementów macierzy x
<code>conj(x)</code>	macierz o elementach sprzężonych z elementami macierzy x

Funkcje dodatkowe

Symbol	Wyjaśnienie
<code>round(x)</code>	zaokrąglę elementy macierzy x do najbliższej liczby całkowitej
<code>rem(x,y)</code>	oblicza resztę z dzielenia odpowiadających sobie elementów macierzy x i y
<code>gcd(a,b)</code>	oblicza największy wspólny dzielnik liczb a i b
<code>lcm(a,b)</code>	Oblicza najmniejszą wspólną wielokrotną liczb a i b

METODY NUMERYCZNE

Temat 1 INTERPOLACJA I APROKSYMACJA

INTERPOLACJA

INTREPOLACJA FUNKCJI JEDNEJ ZMIENNEJ

Standardowe procedury MATLAB-a realizują interpolację za pomocą następujących metod:

- interpolacja wielomianami
- interpolacja za pomocą funkcji sklejanych

1) interpolacja wielomianami pierwszego i trzeciego stopnia

Powszechnie znanym wielomianem jest wielomian interpolacyjny Lagrange'a postaci:

$$W_N(x) = \sum_{i=1}^N y_i \frac{(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}$$

Podstawową wadą takich interpolacji jest skłonność do występowania silnych oscylacji między węzłami interpolacji dla wielomianów interpolacyjnych wysokich stopni

2) interpolacja za pomocą funkcji sklejanych

Metoda ta sprowadza się do podziału przedziału, w którym dokonywana jest interpolacja, na kilka podprzedziałów oraz dokonywania interpolacji na kolejnych obszarach interpolowanej funkcji za pomocą wielomianów niskiego rzędu przy jednoczesnej gwarancji „gładkiego” przejścia między kolejnymi przedziałami.

W dziedzinie obliczeń numerycznych interpolacja spełnia zwykle role pomocniczą

np. dla konstrukcji metod całkowania i różniczkowania numerycznego
lub w metodach optymalizacji.

FUNKCJA BIBLIOTECZNA

Interpolacja funkcji jednej zmiennej w punktach określonych wektorem x_i

```
yi=interp1(x,y,xi,'metoda')
```

gdzie:

x, y - wezły interpolacji (elementy wektora x muszą tworzyć ciąg rosnący)

'metoda' - opcjonalny parametr pozwalający na wybór metody interpolacji

'linear' - interpolacja funkcja liniowa,

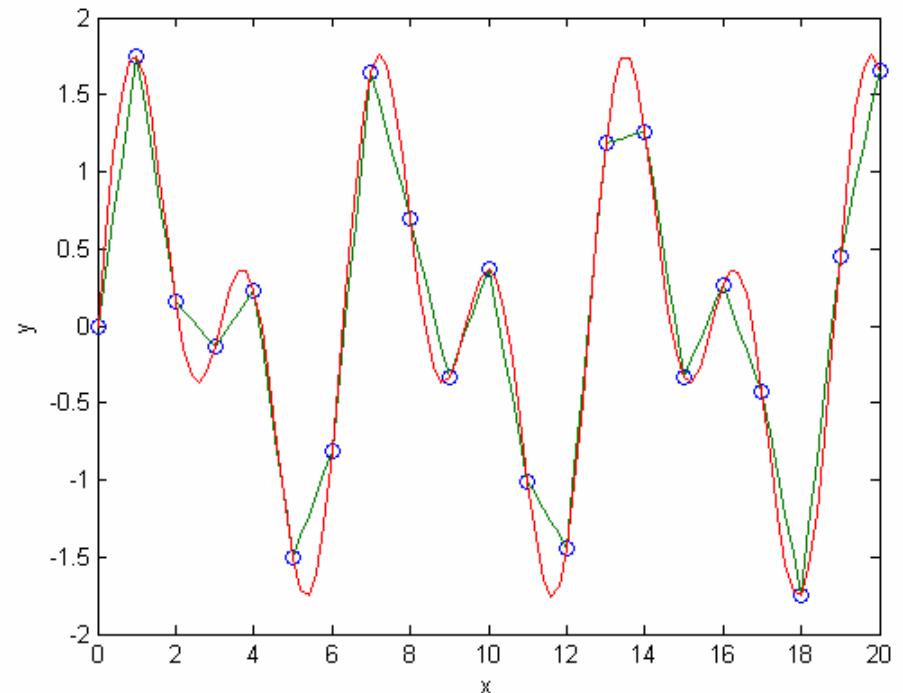
'spline' - interpolacja funkcjami sklejonymi trzeciego stopnia,

'cubic' - interpolacja wielomianami trzeciego stopnia (odległości między elementami x muszą być równe).

PRZYKŁAD - interpolacja różnymi metodami

```
x=0:20; y=sin(x)+sin(2*x);  
xi=0:.2:20;  
figure(1)  
yi=interp1(x,y,xi,'linear');  
plot(x,y,'o',xi,yi,xi,sin(xi)+sin(2*xi))  
xlabel('x')  
ylabel('y')
```

interpolacja funkcji $y=\sin(x)+\sin(2x)$
łamana, uzyskaną wskutek
uwzględnienia 21 węzłów
rozmieszczonych równomiernie w
przedziale $<0,20>$

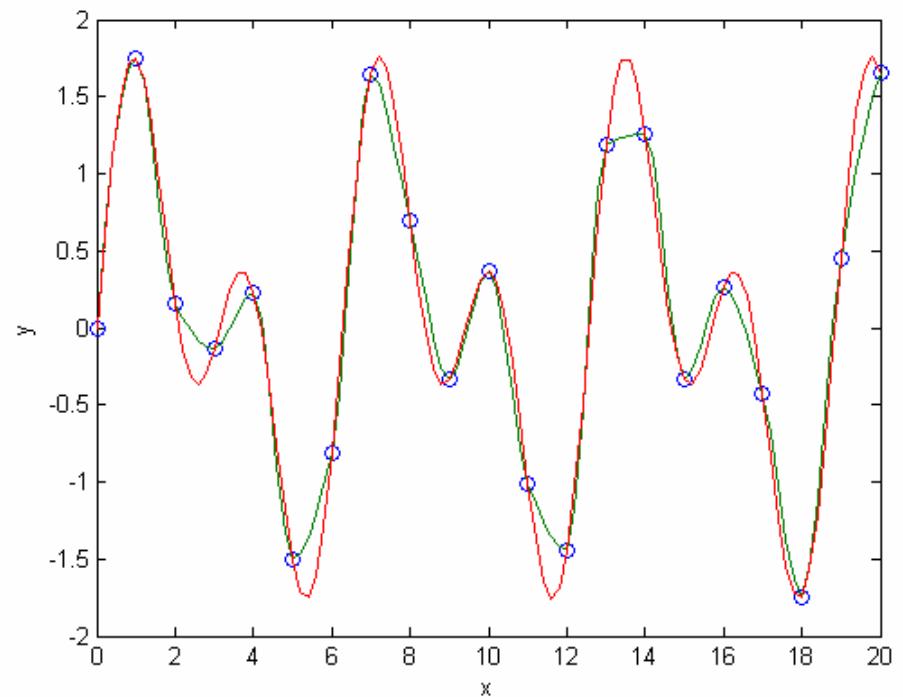


```

figure(2)
yi=interp1(x,y,xi,'cubic');
plot(x,y,'o',xi,yi,xi,sin(xi)+sin(2*xi))
xlabel('x')
ylabel('y')

```

interpolacja funkcji $y=\sin(x)+\sin(2x)$
 wielomianami trzeciego stopnia, przy
 uwzględnieniu 21 węzłów
 rozmieszczonych równomiernie w
 przedziale $<0,20>$

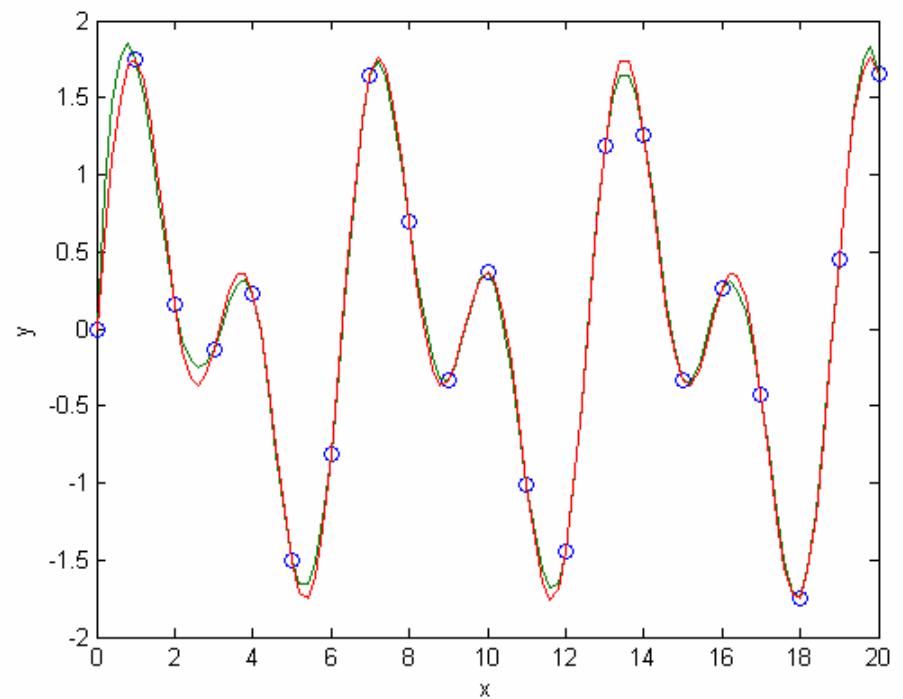


```

figure(3)
yi=interp1(x,y,xi,'spline');
plot(x,y,'o',xi,yi,xi,sin(xi)+sin(2*xi))
xlabel('x')
ylabel('y')

```

interpolacja funkcji $y=\sin(x)+\sin(2x)$
 funkcjami sklejonymi, przy
 uwzględnieniu 21 węzłów
 rozmiieszczonych równomiernie w
 przedziale $<0,20>$



INTREPOLACJA FUNKCJI DWÓCH ZMIENNYCH

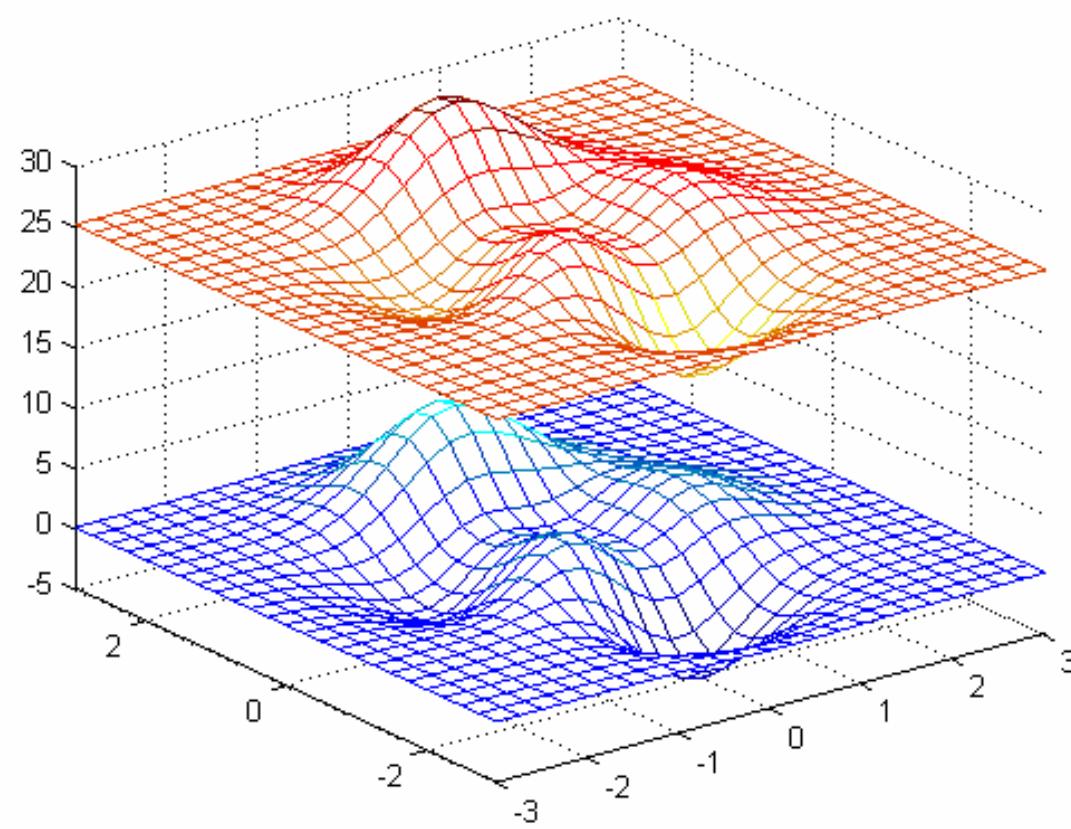
FUNKCJA BIBLIOTECZNA

Interpolacja funkcji dwóch zmiennych w punktach określonych macierzą [X, Y]

```
zi=interp2(x,y,z,xi,yi,zi,'metoda')
```

PRZYKŁAD

```
[X,Y] = meshgrid(-3:.25:3);
Z = peaks(X,Y);%dowolna funkcja dwóch zmiennych
[XI,YI] = meshgrid(-3:.125:3);
ZI = interp2(X,Y,Z,XI,YI,'linear');
mesh(X,Y,Z), hold, mesh(XI,YI,ZI+25)
hold off
axis([-3 3 -3 3 -5 30])
```



APROKSYMACJA

Aproksymacja wielomianem

FUNKCJA BIBLIOTECZNA

$$f=polyfit(x,y,r)$$

gdzie:

x,y - wektor danych

r - stopien wielomianu

Funkcja ta dla wektorów danych x i y generuje współczynniki wielomianu stopnia r przyblizajacego najlepiej w sensie sredniokwadratowym zaleznosc miedzy seria danych x i y.

PRZYKŁAD

```
x=[6.1 8 10.3 11.3 12.3 13.2 14.2 16.1];%zbiór rzednych  
y=[61.3 50 46.4 46.4 40.8 31.6 29.9 22];%zbiór wartosci  
n=1;%stopien wielomianu (dla n=1 funkcja liniowa)
```

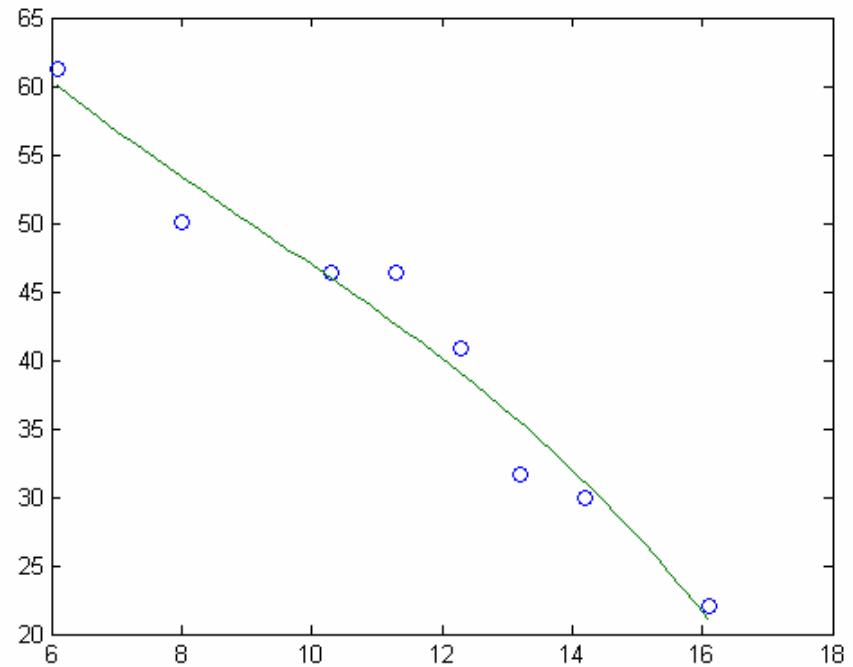
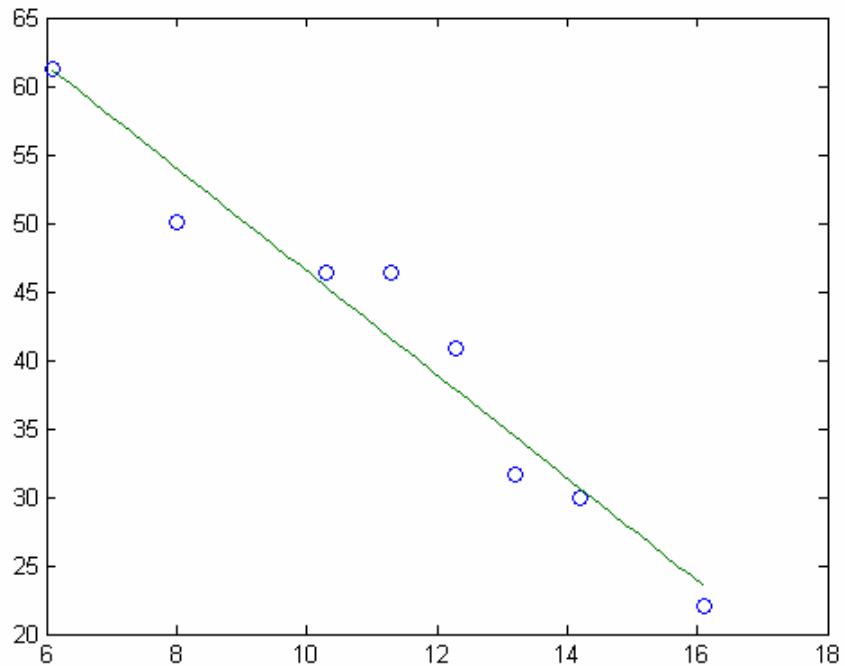
```
f=polyfit(x,y,n);%tablica wspolczynnikow wielomianu
```

```
p=min(x)%poczatek zbioru rzednych xn  
k=max(x)%koniec zbioru rzednych xn  
dx=(k-p)/50; %krok podzialu zbioru xn  
xn=(p:dx:k); %zbiór rzednych xn  
for i=1:length(xn)  
    yn(i)=polyval(f,xn(i));%zbiór wartosci funkcji  
end  
figure(1)  
plot(x,y,'o',xn,yn)
```

```

n=3;%stopien wielomianu
f=polyfit(x,y,n);
for i=1:length(xn)
    yn(i)=polyval(f,xn(i));%zbiór wartosci wielomianu
end
figure(2)
plot(x,y,'o',xn,yn)

```



ZADANIE DOMOWE nr 1

Dokonaj aproksymacji mając dany zbiór danych empirycznych przedstawionych w tabeli x i y

x	y
2.0	12.0
1.2	8.0
14.8	76.4
8.3	17.0
8.4	21.3
3.0	10.0
4.8	12.5
15.6	97.3
11.5	25.0
14.2	38.6
14.0	47.3
16.1	88.0

- 1) Aproksymuj dane prostą oraz wielomianami stopnia 2, 3 i 4 wykorzystując dostępne funkcje języka Matlab.
- 2) Napisz własną procedurę w przypadku aproksymacji liniowej.
- 3) Określ błędy korelacji i determinacji każdej z wykonanych aproksymacji.

METODY NUMERYCZNE

Temat 2 WIELOMIANY, MIEJSCA ZEROWE

MATLAB posiada kilka funkcji bibliotecznych znajdujących zera funkcji jednej zmiennej oraz pozwalających wygodnie operować na wielomianach.

FUNKCJA BIBLIOTECZNA

fzero - oblicza miejsce zerowe funkcji o podanej nazwie zaczynając obliczenia z punktu startowego x_0 , który winien być początkowym przybliżeniem wartości szukanego miejsca zerowego.

```
z=fzero('F',x0,esp)
```

gdzie:

esp - zadana dokładność,

PRZYKŁAD

Tworzymy skrypt funkcji w pliku o nazwie fun.m

```
function [Y]=fun(X)
```

```
%funkcja
```

```
Y=sin(X)-X*cos(X)
```

```
% koniec fun.m
```

Obliczamy wartość miejsca zerowego w oknie komend

```
z=fzero('fun',4.71239,10^-3)
```

```
z=4.4934
```

PRZYKŁAD

Wyznaczenie wartość stałej π przy wykorzystaniu funkcji fzero.
Poszukiwanie miejsca zerowego funkcji sinus w pobliżu wartości 3.

```
>> x = fzero(@sin,3)  
x =  
3.1416
```

PRZYKŁAD

Poszukiwanie miejsc zerowych funkcji sinus w przedziale pomiędzy wartościami 1 i 2.

```
>> x = fzero(@cos,[1 2])  
x =  
1.5708
```

PRZYKŁAD

Poszukiwanie miejsca zerowego funkcji $f(x) = x^3 - 2x + 5$.

```
>> f = @(x)x.^3-2*x-5;  
>> z = fzero(f,2)  
z =  
2.0946
```

Funkcja fzero poszukuje tylko wyników w zbiorze liczb rzeczywistych.

```
>> roots([1 0 -2 -5])  
ans =  
2.0946  
-1.0473 + 1.1359i  
-1.0473 - 1.1359i
```

WIELOMIANY

FUNKCJA BIBLIOTECZNA

poly - wyznacza współczynniki $a(1), a(2), \dots, a(n+1)$ wielomianu

$$W(x, a) = a(1) x^n + a(2) x^{n-1} + \dots + a(n+1)$$

o pierwiastkach podanych w wektorze r.

$$a=\text{poly}(r)$$

Współczynniki są uszeregowane według malejących potęg zmiennej x .

PRZYKŁAD

W oknie komend wpisujemy

r=[1 2 3 4 5]

a=poly(r)

a=[1 -15 85 -225 274 -120]

FUNKCJA BIBLIOTECZNA

polyval – służy wyznaczaniu wartości wielomianów

PRZYKŁAD

Wyznacz wartości wielomianu postaci $p(x) = 3x^2 + 2x + 1$ dla x równego kolejno 5, 7 i 9.

```
>> p = [3 2 1];
>> polyval(p,[5 7 9])
ans =
    86   162   262
```

FUNKCJA BIBLIOTECZNA

roots - funkcja oblicza pierwiastki wielomianu $W(x, a)$ o postaci, jak podano w opisie funkcji poly

r=roots(a)

PRZYKŁAD

W oknie komend wpisujemy

a=[1 -15 85 -225 274 -120]

r=roots(a)

r=[5 4 3 2 1]

FUNKCJA BIBLIOTECZNA

fminbnd – funkcja pozwala na wyznaczenie wartości minimalnych wybranej funkcji w określonym przedziale.

PRZYKŁAD

Wyznacz wartość minimalną funkcji $f(x) = x^3 - 2x - 5$ w przedziale od 0 do 2.

```
>> f = @(x)x.^3-2*x-5;  
>> x = fminbnd(f, 0, 2)  
x =  
0.8165
```

wartość

```
>> f=[1,0,-2,-5]  
>> polyval(f,x)  
ans =  
-6.0887
```

METODY NUMERYCZNE

Temat 3

CAŁKOWANIE NUMERYCZNE

CAŁKOWANIE NUMERYCZNE

Metody całkowania numerycznego, zwane *kwadraturami* sprowadzają się do przybliżenia funkcji podcałkowej f lub jej kolejnych fragmentów na danym przedziale $\langle a, b \rangle$ za pomocą innej funkcji, dla której wartości całki jest określona analitycznie.

Stosuje się w trakcie analizy kwadratury:

- złożone,
- adaptacyjne.

Idea kwadratury złożonej polega na podziale przedziału $\langle a, b \rangle$ na pewną liczbę podprzedziałów (np. ze względu na dużą zmienność funkcji w przedziale całkowania) i obliczenie wartości całki na każdym z tych przedziałów, a następnie zsumowanie tych wartości.

Techniki adaptacyjne polegają na wstępny podziale przedziału całkowania na dwie części równej długości i obliczenie wartości całek w każdym z nich za pomocą kwadratury. Przedział, w którym nie osiągnięto wystarczającej dokładności obliczeń (porównuje się wartość funkcji w danym przedziale z wartością funkcji po podziale na dwie części) ponownie jest dzielony na dwie części i tak do czasu aż osiągniemy zadana zbieżność.

Standardowe procedury MATLAB-a pozwalające na numeryczne obliczenie wartości całki oznaczonej funkcji jednej zmiennej to:

- quad - kwadratura adaptacyjna oparta o interpolacje wielomianem 2 stopnia,
- quadl - kwadratura adaptacyjna oparta o aproksymacje wielomianem 8 stopnia.

FUNKCJE BIBLIOTECZNE

```
Q1=quad(f,a,b,tol)
Ql=quadl(f,a,b,tol)
```

gdzie:

f - lancuch zawierajacy nazwe funkcji umieszczonej w osobnym skrypcie,
a, b - przedzial całkowania,
tol - wymagana tolerancja względna (domyślnie 10^{-3} - przy braku parametru).

PRZYKŁAD - Całkowanie numeryczne

$$\int_0^{\pi} \sin(x^2) dx$$

Tworzymy skrypt funkcji podcałkowej w pliku o nazwie quadfun.m

function [Y]=quadfun(X)

%funkcja podcalkowa

Y=sin(X.*X);

% koniec quadfun.m

Obliczamy wartość całki w oknie komend

Q1=quad('quadfun',0,pi)

i

Ql=quadl('quadfun',0,pi)

Q1=Ql=0.7727

METODY NUMERYCZNE

Temat 4 UKŁADY RÓWNAŃ LINIOWYCH I DEKOMPOZYCJA MACIERZY

WŁASNOŚCI MACIERZY

Rząd macierzy – największy stopień uzyskanej z niej podmacierzy kwadratowej nieosobliwej tzn. takiej której wyznacznik jest różny od zera.

Do obliczenia rzędu macierzy służy funkcja rank.

$$r=\text{rank}(A)$$

Według twierdzenia Cronecera –Capelliego układ równań liniowych $\mathbf{Ax}=\mathbf{b}$ **można jednoznacznie rozwiązać** gdy rząd macierzy głównej jest równy rzędowi macierzy dołączonej

PRZYKŁAD

$$A = [1 \ 2 \ -1; \ 2 \ 1 \ -2; \ 1 \ 2 \ 2];$$

$$b = [1 \ 2 \ 3]';$$

$$r_A = \text{rank}(A) = 3$$

$$r_{Ab} = \text{rank}([A \ b]) = 3$$

Ślad macierzy - suma elementów znajdujących się na głównej przekątnej.
Do obliczenia sladu macierzy służy funkcja trace.

$$t = \text{trace}(A)$$

PRZYKŁAD

$$A = [1 \ 2 \ -1 \ 3; 2 \ 1 \ -2 \ 1; 1 \ 2 \ 2 \ 3; 2 \ 1 \ 1 \ 1]$$

$$t = \text{trace}(A) = 5$$

Wyznacznik macierzy

$$|A| = \det(A) = \sum_{p \in P} \text{sgn}(p) \cdot a_{1p_1} \cdot a_{2p_2} \cdot \dots \cdot a_{np_n}$$

A - macierz kwadratowa $n \times n$

P – zbiór wszystkich permutacji liczb (możliwych uszeregowień tych liczb) od 1 do n

Dla każdej permutacji p obliczamy następujący iloczyn:

$$\text{sgn}(p) \cdot a_{1p_1} \cdot a_{2p_2} \cdot \dots \cdot a_{np_n}$$

$$\operatorname{sgn}(p) = \begin{cases} 1, & \text{gdy } p \text{ jest permutacją patrzystą} \\ -1, & \text{gdy } p \text{ jest permutacją niepatrzystą} \end{cases}$$

i sumujemy dla kolejnych permutacji.

Obliczanie wyznacznika z definicji jest niezmiernie czasochłonne, gdyż wymaga wyznaczenia $n!$ permutacji, dlatego w praktyce nie wykonuje się go w ten sposób.

Metoda obliczania wyznacznika polega na sprowadzeniu macierzy do postaci, w której wyznacznik jest łatwo obliczyć np. trójkątnej, gdzie wyznacznikiem macierzy jest iloczyn elementów na przekątnej.

Do obliczenia wyznacznika macierzy służy funkcja \det .

$$d=\det(A)$$

PRZYKŁAD

$$A=[1\ 2\ 10\ -2; 3\ -3\ 6\ 9; 2\ 4\ -1\ 8; 4\ 5\ 6\ 10]$$

$$d=\det(A)$$

$$d=-351$$

UKŁADY RÓWNAŃ LINIOWYCH I DEKOMPOZYCJA MACIERZY

W wielu sytuacjach obliczeniowych (wyznaczniki i rozwiązywanie układów równań) korzystne jest przedstawienie danej macierzy A w postaci iloczynu macierzy trójkątnych dolnej i górnej: $A=LU$

Funkcja lu znajduje macierze L i U takie, ze $L^*U=A$

$$[L, U] = \text{lu}(A)$$

PRZYKŁAD

$$A = [1 \ 2 \ 1; 0 \ 4 \ 2; 6 \ 1 \ 1]$$

$$[L, U] = \text{lu}(A)$$

$$L =$$

$$\begin{array}{ccc} 0.1667 & 0.4583 & 1.0000 \\ 0 & 1.0000 & 0 \\ 1.0000 & 0 & 0 \end{array}$$

macierz nie jest macierzą trójkątną dolna

$$U =$$

$$\begin{array}{ccc} 6.0000 & 1.0000 & 1.0000 \\ 0 & 4.0000 & 2.0000 \\ 0 & 0 & -0.0833 \end{array}$$

dla macierzy zachodzi zależność $A=LU$

$$[L, U, P] = lu(A)$$

gdzie zachodzi własność $LU=PA$, a P to macierz permutacji (nieosobliwa zbudowana z zer i jedynek; $P^{-1}=P$)

$$[L, U, P] = lu(A)$$

$L =$

$$\begin{matrix} 1.0000 & 0 & 0 \\ 0 & 1.0000 & 0 \\ 0.1667 & 0.4583 & 1.0000 \end{matrix}$$

$U =$

$$\begin{matrix} 6.0000 & 1.0000 & 1.0000 \\ 0 & 4.0000 & 2.0000 \\ 0 & 0 & -0.0833 \end{matrix}$$

$P =$

$$\begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{matrix}$$

ZASTOSOWANIE ROZKŁADU LU UKŁADY RÓWNAŃ LINIOWYCH

Rozkład LU ułatwia rozwiązywanie różnych zadań. Najprostszym jest rozwiązywanie układów równań liniowych postaci: $\mathbf{Ax}=\mathbf{b}$

Jeżeli znamy rozkład macierzy:

$$\mathbf{A=PLU},$$

równanie wyjściowe wówczas ma postać:

$$\mathbf{PLUx=b \text{ lub } LUX= Pb} \quad (\mathbf{P^{-1}=P}).$$

Aby wyznaczyć \mathbf{x} trzeba rozwiązać pomocniczy układ równań:

$$\mathbf{Ly=Pb},$$

a uzyskany \mathbf{y} wstawić do równania

$$\mathbf{Ux=y}.$$

PRZYKŁAD - Rozwiązywanie układu równań liniowych

$A = [10 \ 2 \ 1; 0 \ 40 \ 2; 6 \ 1 \ 10];$

$b = [20 \ 30 \ 40]'$; - to musi być wektor kolumnowy

% Sprawdzenie rzędu macierzy

$\text{rank}(A) = 3$

$\text{rank}([A \ b]) = 3$

% Rozwiązywanie układu równan

$$x = A \setminus b$$

$x =$

1.5808

0.6004

2.9915

Funkcja ta rozwiązuje układ równan dwukrotnie szybciej niż polecenie:

$$x = \text{inv}(A) * b$$

$\text{inv}(A)$ - wyznacza odwrotność macierzy A

ZASTOSOWANIE ROZKŁADU LU DO ODWRACANIA MACIERZY

$$A = PLU \quad A^{-1} = (PLU)^{-1} = U^{-1}L^{-1}P^{-1} = U^{-1}L^{-1}P$$

PRZYKŁAD

$$A = [1 \ 2 \ 1; 0 \ 4 \ 2; 6 \ 1 \ 1];$$

$$\text{inv}(A) =$$

$$\begin{matrix} 1.0000 & -0.5000 & -0.0000 \\ 6.0000 & -2.5000 & -1.0000 \\ -12.0000 & 5.5000 & 2.0000 \end{matrix}$$

$$[L, U, P] = \text{lu}(A);$$

$$A = P^* L^* U$$

$$\text{inv}A = \text{inv}(U)^* \text{inv}(L)^* P$$

$$\text{inv}A =$$

$$\begin{matrix} 1.0000 & -0.5000 & -0.0000 \\ 6.0000 & -2.5000 & -1.0000 \\ -12.0000 & 5.5000 & 2.0000 \end{matrix}$$

ZASTOSOWANIE ROZKŁADU LU DO WYZNACZANIA WARTOŚCI WYZNACZNIKA MACIERZY

PRZYKŁAD

$$A = [1 \ 2 \ 1; 0 \ 4 \ 2; 6 \ 1 \ 1];$$

$$d = \det(A) = 2$$

$$[L, U, P] = lu(A)$$

wyznacznik A jest równy iloczynowi elementów na głównej przekątnej U

$$\det(A) = \det(U) = \prod_{i=1}^n u_{ii}$$

$$a = \text{diag}(U)$$

$$d = \text{abs}(a(1)*a(2)*a(3)) = 2$$

ROZKŁAD CHOLEWSKY'EGO

Obok znajdującego różnorodne zastosowanie rozkładu macierzy $A=LU$ należy wspomnieć o rozkładzie Cholewsky'ego zwanym także rozkładem Banachiewicza. Polega on na zapisaniu macierzy A (dodatnio określonej - zawsze odwracalna i jej odwrotność jest również dodatnio określona) w postaci iloczynu LL^T , gdzie L jest macierzą trójkątną dolną.

Rozkład ten realizuje funkcja:

$$L=\text{chol}(A)$$

PRZYKŁAD

$A=[10 \ 2 \ 1; \ 2 \ 40 \ 2; \ 6 \ 1 \ 10];$
 $L=\text{chol}(A)$

$L =$

$$\begin{matrix} 3.1623 & 0.6325 & 0.3162 \\ 0 & 6.2929 & 0.2860 \\ 0 & 0 & 3.1334 \end{matrix}$$

ROZKŁAD NA MACIERZ ORTOGONALNĄ I TÓJKĄTNĄ GÓRNĄ

Funkcja:

$$[Q, R] = qr(A)$$

$$A = QR$$

gdzie:

R - macierz trójkątna górnna

Q - macierz ortogonalna

PRZYKŁAD

$$A = [10 \ 2 \ 1; 2 \ 40 \ 2; 6 \ 1 \ 10];$$

$$Q =$$

$$\begin{matrix} -0.8452 & 0.1427 & -0.5151 \\ -0.1690 & -0.9856 & 0.0043 \\ -0.5071 & 0.0907 & 0.8571 \end{matrix}$$

$R =$

$$\begin{matrix} -11.8322 & -8.9586 & -6.2541 \\ 0 & -39.0480 & -0.9212 \\ 0 & 0 & 8.0646 \end{matrix}$$

Sprawdzenie ortogonalności macierzy Q

$$Q^T Q = Q Q^T = I$$

$Q^T Q =$

$$\begin{matrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{matrix}$$

$Q Q^T =$

$$\begin{matrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{matrix}$$

WEKTORY I WARTOŚCI WŁASNE MACIERZY

Jeżeli pewien wektor x zostanie poddany przekształceniu opisanemu przez macierz A w wyniku którego otrzymamy wektor y : $y=A*x$. Jeżeli $y=\lambda*x$ to wtedy x jest wektorem własnym macierzy A , zaś λ jest jej wartością własną. Spełnione są związki:

$$\det(A-\lambda*I)=0$$

oraz

$$(A-\lambda*I)*X=0$$

Numeryczne wyznaczanie wartości własnych i wektorów własnych nie jest zadaniem łatwym, a dostępne algorytmy ich uzyskania nie są niezawodne zwłaszcza gdy mamy do czynienia z wielokrotnymi wartościami własnymi macierzy lub wartościami w postaci liczb zespolonych.

PRZYKŁAD

Obliczyć wartości własne macierzy

$A=[3 \ -1 \ 1; \ -1 \ 5 \ -1; \ 1 \ -1 \ 3];$

Współczynniki wielomianu charakterystycznego według malejących potęg

$p=poly(A)$

$p =$

1.0000 -11.0000 36.0000 -36.0000

Wartości własne macierzy

$\lambda=roots(p)'$

$\lambda =$

6.0000 3.0000 2.0000

Wartości własne można też wyznaczyć przy pomocy polecenia

$\lambda=eig(A)$

lub

$$[X, \lambda] = \text{eig}(A)$$

gdzie:

X - macierz wektorów własnych macierzy **A**

lambda - macierz diagonalana, której główna przekątna zawiera wartości własne

PRZYKŁAD

$A = [3 \ -1 \ 1; \ -1 \ 5 \ -1; \ 1 \ -1 \ 3];$

$\lambda = \text{eig}(A)'$

$\lambda =$

2.0000 3.0000 6.0000

$[X, \lambda] = \text{eig}(A)$

$X =$

0.7071	-0.5774	0.4082
-0.0000	-0.5774	-0.8165
-0.7071	-0.5774	0.4082

wektory własne macierzy A

lambda =

2.0000	0	0
0	3.0000	0
0	0	6.0000

wartości własne macierzy A na głównej przekątnej

LABORATORIUM METOD NUMERYCZNYCH

Wstęp

Rozwój techniki komputerowej spowodował, że wiele skomplikowanych problemów naukowych rozwiązywanych jest przy pomocy maszyn cyfrowych. Różnorodność zagadnień i złożoność obliczeń wymaga niejednokrotnie dobrej znajomości problemów mających wpływ na dokładność obliczeń czy też na szybkość ich wykonania. Istnieje wiele metod rozwiązywania typowych zagadnień, dlatego wszechstronne ich poznanie umożliwia właściwe podejście do problemów związanych z obliczeniami numerycznymi.

W ramach laboratorium z przedmiotu Metody Numeryczne przedstawione zostały podstawowe działy metod numerycznych: interpolacja, aproksymacja, równania nieliniowe, układy równań liniowych, wartości własne macierzy, całkowanie i równania różniczkowe zwyczajne.

- MATLAB – pakiet obliczeniowy firmy MathWorks umożliwiający dokonywanie dowolnych obliczeń numerycznych

PODSTAWY MATLAB-a

1. Wprowadzenie.

MATLAB jest programem służącym do obliczeń numerycznych.

Na prawidłowość wyników uzyskiwanych w trakcie obliczeń mają wpływ dwa podstawowe elementy:

- uwarunkowanie zadania - (złe uwarunkowanie powoduje, że małe odchylenia danych wejściowych mają duży wpływ na wynik końcowy)
- stabilność algorytmów - w trakcie obliczeń następuje kumulacja błędów, obliczenia w MATLABie dokonywane są na liczbach zmiennoprzecinkowych, zarówno te liczby jak i wykonywane na nich operacje obarczone są pewnymi błędami uzależnionymi od precyzji zapisu. Błędy te w trakcie obliczeń mają tendencję do przenoszenia się i kumulowania, jeżeli powodują uzyskanie wyniku znacznie oddalonego od prawidłowego to mówimy o niestabilnym algorytmie obliczeniowym.

Jednym używanym typem danych są macierze, przy czym MATLAB umożliwia również dokonywanie operacji arytmetycznych dla poszczególnych elementów macierzy, przy wykorzystaniu tzw. operatorów tablicowych.

Macierze należy oznaczać dużymi literami, natomiast wektory bądź tablice wartości mogą być oznaczane małymi lub dużymi literami. Tę samą zmienną zapisaną raz dużą literą raz małą MATLAB traktuje jako dwie różne zmienne.

operatorы arytmetyczne:

*	mnożenie
^	potęgowanie
+ -	dodawanie, odejmowanie
/	dzielenie (dzielenie prawostronne),
\	dzielenie lewostronne ($A/B = (A' \cdot B')$)
'	transpozycja macierzy (tablicy)
.	tablica wartości

operatorы porównania

==	równe
~=	różne
<	mniejsze
>	większe
<=	mniejsze równe
>=	większe równe

Części dziesiętne oddzielane są kropką (np. 3.5, 100.9). Liczby ułamkowe postaci $a \cdot 10^{-n}$ zapisywane są następująco: ae-n.

Można podać sposób wyświetlania obliczeń pisząc polecenie **format** z odpowiednim parametrem (np. liczba 1/3; format short – 0.3334; format long – 0.3333333333334).

W celu odróżnienia działań dokonywanych na macierzach od działań dokonywanych na tablicach wartości, w przypadku tablic wartości należy **zawsze po zmiennej umieścić kropkę przed znakiem mnożenia, dzielenia i potęgowania** (np. $(x.^4).*tan(x) + x.*sin(x) - x./cos(x)$). W przypadku dzielenia umieszcza się **kropkę również po stałej przed znakiem dzielenia** (np. $2./x$).

Najczęściej używane znaki przy pisaniu własnego programu:

%	na początku linii – linia ta jest komentarzem
;	na końcu linii zawierającej wzory – program nie wyświetla pośrednich obliczeń
%%	na początku pierwszej linii po której jest pusty wiersz – linia ta jest helpem do pliku
...	na końcu linii – dalszy ciąg danej linii w następnym wierszu

Napisany program należy zachowywać w skrypcie z rozszerzeniem "**m**" i umieszczać w katalogu o nazwie **MATLAB**. Katalog ten należy założyć na dysku sieciowym użytkownika. Program obliczeniowy uruchamiany jest poprzez napisanie w oknie MATLAB-a nazwy pliku bez rozszerzenia. Przed jego wywołaniem należy podać ścieżkę dostępu: np. **path(path,'g:\MATLAB')**.

Niektóre obliczenia wymagają zastosowania wbudowanych funkcji MATLAB-a (np. całkowanie, różniczkowanie, wyznaczanie zer funkcji), wówczas w skrypcie deklaruje się dane zadanie jako własną funkcję pisząc "function $y = f(x)$ ", natomiast w oknie programu należy napisać polecenie zawierające odpowiednią funkcję MATLAB-a (np. quad, ode23, fzero).

2. Definiowanie macierzy.

a) przez wyliczenie elementów:

np. : $A = [2 \ 2 \ 1; 3 \ 4 \ 5; 5 \ 6 \ 7]$

$$A = \begin{bmatrix} 2 & 2 & 1 \\ 3 & 4 & 5 \\ 5 & 6 & 7 \end{bmatrix}$$

Macierz deklaruje się poprzez umieszczenie jej elementów w nawiasach kwadratowych; poszczególne elementy macierzy oddzielane są spacjami; koniec wiersza oznaczany jest średnikiem lub deklarowany jest poprzez wciśnięcie klawisza "Enter".

b) przez wygenerowanie elementów:

np. $x = -5 : 0.1 : 8$ macierz wierszowa (-5 - pierwszy element; 0.1 - krok, 8 - ostatni (131) element)
 $y = f(x)$ macierz wierszowa ($y_1 = f(5)$ - pierwszy element, $y_{131} = f(8)$ - ostatni (131) element)

c) przez podanie zależności określającej elementy macierzy:

np. dla macierzy Hilberta elementy macierzy określone są następującą zależnością:
 $a(i, j) = 1 / (i+j-1)$, aby wyznaczyć elementy tej macierzy można skorzystać z biblioteki programu pisząc polecenie **hilb(n)**, gdzie **n** - stopień macierzy lub napisać własny program:

```
n =  
for i = 1:n  
    for j = 1:n  
        A(i,j) = 1/(i+j-1);  
    end  
end  
disp(A)
```

napisanie średnika na końcu linii powoduje brak wyświetlania pośrednich obliczeń,
disp() - wyświetlenie wyniku

Elementy macierzy lub tablicy wartości umieszcza się zawsze w nawiasach kwadratowych, natomiast nawiasy zwykłe zarezerwowane są dla polecień i funkcji MATLABA-a.

Poszczególne elementy polecenia bądź funkcji oddzielane są przecinkami i jeżeli nie stanowią zmiennej lub liczby umieszczane są w apostrofach (np. `plot(x, y, 'r*')`).

Ponieważ program pamięta wszystkie wykorzystywane zmienne, więc w przypadku wprowadzenia nowej zmiennej pod używaną wcześniej nazwą, należy usunąć starą zmienną poleceniem **clear nazwa_zmiennej**; można też usunąć wszystkie dotychczasowe zmienne pisząc **clear**.

3. Podstawowe działania arytmetyczne na macierzach i tablicach wartości.

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_{11} & \mathbf{a}_{12} \\ \mathbf{a}_{21} & \mathbf{a}_{22} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \mathbf{b}_{11} & \mathbf{b}_{12} \\ \mathbf{b}_{21} & \mathbf{b}_{22} \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$$

a) mnożenie

macierzy $\mathbf{A}^* \mathbf{B} = \begin{bmatrix} \mathbf{a}_{11}\mathbf{b}_{11} + \mathbf{a}_{12}\mathbf{b}_{21} & \mathbf{a}_{11}\mathbf{b}_{12} + \mathbf{a}_{12}\mathbf{b}_{22} \\ \mathbf{a}_{21}\mathbf{b}_{11} + \mathbf{a}_{22}\mathbf{b}_{21} & \mathbf{a}_{21}\mathbf{b}_{12} + \mathbf{a}_{22}\mathbf{b}_{22} \end{bmatrix}$

$$\mathbf{A}^* \mathbf{B} = \begin{bmatrix} 5 & 3 \\ 8 & 5 \end{bmatrix}$$

tablic wartości $\mathbf{A} \cdot \mathbf{B} = \begin{bmatrix} \mathbf{a}_{11}\mathbf{b}_{11} & \mathbf{a}_{12}\mathbf{b}_{12} \\ \mathbf{a}_{21}\mathbf{b}_{21} & \mathbf{a}_{22}\mathbf{b}_{22} \end{bmatrix}$

$$\mathbf{A} \cdot \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}$$

b) dzielenie

macierzy $\mathbf{A} / \mathbf{B} = \mathbf{A}^* \mathbf{B}^{-1} = \mathbf{A}^* \frac{\mathbf{B}^{D^*}}{|\mathbf{B}|}$

$$\mathbf{B}^{D^*} = \begin{bmatrix} \mathbf{B}_{11}^* & \mathbf{B}_{12}^* \\ \mathbf{B}_{21}^* & \mathbf{B}_{22}^* \end{bmatrix}^T$$

$$[\mathbf{B}_{11}^*] = (-1)^2 \mathbf{b}_{22} \quad [\mathbf{B}_{12}^*] = (-1)^3 \mathbf{b}_{21}$$

$$[\mathbf{B}_{21}^*] = (-1)^3 \mathbf{b}_{12} \quad [\mathbf{B}_{22}^*] = (-1)^4 \mathbf{b}_{11}$$

$$\mathbf{B}^{D^*} = \begin{bmatrix} \mathbf{b}_{22} & -\mathbf{b}_{12} \\ -\mathbf{b}_{21} & \mathbf{b}_{11} \end{bmatrix}$$

$$|\mathbf{B}| = \mathbf{b}_{11}\mathbf{b}_{22} - \mathbf{b}_{12}\mathbf{b}_{21}$$

$$\mathbf{B}^{-1} = \frac{\mathbf{B}^{\text{D}*}}{|\mathbf{B}|} = \begin{bmatrix} \frac{\mathbf{b}_{22}}{\mathbf{b}_{11}\mathbf{b}_{22} - \mathbf{b}_{12}\mathbf{b}_{21}} & \frac{-\mathbf{b}_{12}}{\mathbf{b}_{11}\mathbf{b}_{22} - \mathbf{b}_{12}\mathbf{b}_{21}} \\ \frac{-\mathbf{b}_{21}}{\mathbf{b}_{11}\mathbf{b}_{22} - \mathbf{b}_{12}\mathbf{b}_{21}} & \frac{\mathbf{b}_{11}}{\mathbf{b}_{11}\mathbf{b}_{22} - \mathbf{b}_{12}\mathbf{b}_{21}} \end{bmatrix}$$

$$\mathbf{A} * \mathbf{B}^{-1} = \begin{bmatrix} \frac{\mathbf{a}_{11}\mathbf{b}_{22} - \mathbf{a}_{12}\mathbf{b}_{21}}{\mathbf{b}_{11}\mathbf{b}_{22} - \mathbf{b}_{12}\mathbf{b}_{21}} & \frac{-\mathbf{a}_{11}\mathbf{b}_{12} + \mathbf{a}_{12}\mathbf{b}_{11}}{\mathbf{b}_{11}\mathbf{b}_{22} - \mathbf{b}_{12}\mathbf{b}_{21}} \\ \frac{\mathbf{a}_{21}\mathbf{b}_{22} - \mathbf{a}_{22}\mathbf{b}_{21}}{\mathbf{b}_{11}\mathbf{b}_{22} - \mathbf{b}_{12}\mathbf{b}_{21}} & \frac{-\mathbf{a}_{21}\mathbf{b}_{12} + \mathbf{a}_{22}\mathbf{b}_{11}}{\mathbf{b}_{11}\mathbf{b}_{22} - \mathbf{b}_{12}\mathbf{b}_{21}} \end{bmatrix}$$

$$\mathbf{A} / \mathbf{B} = [3 \ -1 \\ 4 \ -1]$$

tablic wartości

$$\mathbf{A}./\mathbf{B} = \begin{bmatrix} \frac{\mathbf{a}_{11}}{\mathbf{b}_{11}} & \frac{\mathbf{a}_{12}}{\mathbf{b}_{12}} \\ \frac{\mathbf{a}_{21}}{\mathbf{b}_{21}} & \frac{\mathbf{a}_{22}}{\mathbf{b}_{22}} \end{bmatrix}$$

$$\mathbf{A.} / \mathbf{B} = [1 \ 2 \\ 1 \ 3]$$

$$2. / \mathbf{B} = [2 \ 2 \\ 1 \ 2]$$

2 / B - działania nie można wykonać

c) potęgowanie

macierzy

$$\mathbf{A}^{\wedge}3 = \mathbf{A} * \mathbf{A} * \mathbf{A}$$

$$\mathbf{A}^{\wedge}3 = [21 \ 34 \\ 34 \ 55]$$

tablic wartości

$$\mathbf{A.}^{\wedge}3 = \begin{bmatrix} \mathbf{a}_{11}^3 & \mathbf{a}_{12}^3 \\ \mathbf{a}_{21}^3 & \mathbf{a}_{22}^3 \end{bmatrix}$$

- 8 -
 $A.^3 = [1 \ 8
8 \ 27]$

d) dzielenie lewostronne

macierzy

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases}$$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$Ax = b \quad \text{to} \quad x = A^{-1}b \quad \text{lub} \quad x = A \setminus b$$

$x = A \setminus b$ - zapis rozwiązania układu równań liniowych metodą eliminacji Gaussa z częściowym wyborem elementu głównego (dzielenie lewostronne)

$x = A^{-1}b$ - rozwiązanie układu równań metodą odwracania macierzy

np.:

$$\begin{cases} x_1 + 2x_2 = 4 \\ 2x_1 + 3x_2 = 7 \end{cases}$$

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \quad b = \begin{bmatrix} 4 \\ 7 \end{bmatrix} \quad x = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$x = A^{-1}b = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad x = A \setminus b = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

tablic wartości $A \setminus B = \begin{bmatrix} \frac{b_{11}}{a_{11}} & \frac{b_{12}}{a_{12}} \\ \frac{b_{21}}{a_{21}} & \frac{b_{22}}{a_{22}} \end{bmatrix} = B ./ A$

$$A \setminus B = \begin{bmatrix} 1 & 1/2 \\ 1 & 1/3 \end{bmatrix}$$

4. Rysowanie wykresów.

Rysowanie wykresów na płaszczyźnie umożliwia polecenie **plot()**, np.: **plot(x, y, 'typ linii')**, przy czym **x** - zmienna niezależna, **y** - zmienna zależna, natomiast **typ linii** określa kolor i znacznik linii. Dopuszczalne kolory i znaczniki linii:

znacznik		kolor
.	(kropka)	y - żółty
o	(mała litera o)	m - fioletowy
x	(mała litera x)	c - jasno-niebieski
+	(plus)	r - czerwony
*	(znak mnożenia)	g - zielony
-	(minus)	b - niebieski
:	(dwukroppek)	w - biały
-.	(minus, kropka)	k - czarny
--	(minus, minus)	

przykładowe polecenie: **plot(x, y, 'r*')**

Można umieścić kilka krzywych w jednym układzie współrzędnych np.: $y = f(x)$, $z = f(x)$, $u = f(x)$ pisząc polecenie: **plot(x, y, 'r*', x, z, 'c+', x, u, 'co')**.

Polecenie **subplot** pozwala na umieszczenie krzywych w kilku układach współrzędnych w jednym oknie. Składnia tego polecenia jest następująca:

subplot(ilość wykresów w pionie, ilość wykresów w poziomie, kolejność)

np. polecenia:

```
subplot(2, 1, 1)
plot(x, y)
subplot(2, 1, 2)
plot(z, w)
```

umożliwiają narysowanie dwóch wykresów w pionie, przy czym wykres (x, y) jako pierwszy umieszczony jest na górze ekranu, natomiast (z, w) jako drugi na dole ekranu.

Polecenia **xlabel('tekst')**, **ylabel('tekst')** i **title('tekst')** umożliwiają dołączenie etykiet do osi x i y oraz tytułu wykresu, natomiast polecenie **text(x, y, 'tekst')** pozwala na dodanie dowolnego tekstu na wykresie, przy czym (x, y) są to współrzędne określające początek tekstu.

Polecenie **grid on** powoduje wyświetlenie linii siatki, **grid off** usuwa linie siatki z wykresu

5. Wybrane elementarne funkcje matematyczne.

abs(x)	- wartość bezwzględna	log(x)	- logarytm naturalny
max(x)	- wartość maksymalna	log10(x)	- logarytm dziesiętny
real(x)	- część rzeczywista	sqrt(x)	- pierwiastek kwadratowy
imag(x)	- część urojona	exp(x)	- funkcja wykładnicza
cos(x)		pi	- liczba Π
atan(x)			
sin(x)			
tan(x)			
	{ - funkcje trygonometryczne		

Ćwiczenia nr 1 i 2

PRZYBLIŻANIE FUNKCJI

Interpolacja

Dane są wartości funkcji w pewnych punktach zwanych węzłami interpolacji. Należy wyznaczyć przybliżone wartości tej funkcji w punktach nie będących węzłami w taki sposób, aby błąd w tych punktach był jak najmniejszy. W tym celu należy dobrać:

- metodę
- rozmieszczenie węzłów
- liczbę węzłów

Istnieją dwie podstawowe metody interpolacji:

- **liniowa** - w danym przedziale funkcja zastępowana jest odcinkami linii prostej
- **paraboliczna** - w danym przedziale funkcja zastępowana jest wielomianem określonego stopnia (mogą to być wielomiany algebraiczne, trygonometryczne bądź funkcje sklejane)

Błędy w węzłach

Metoda	Rozmieszczenie węzłów											
	równoodległe			zera			punkty ekstremalne			punkty zagęszczone		
	3	5	9	3	5	9	3	5	9	3	5	9
Lagrange'a												
Newtona												
Funkcji sklejanych												
Thielego												
Paszkowskiego												

Moduł z maksymalnej wartości błędu w przedziale interpolacji

Metoda	Rozmieszczenie węzłów											
	równoodległe			zera			punkty ekstremalne			punkty zagęszczone		
	3	5	9	3	5	9	3	5	9	3	5	9
Lagrange'a												
Newtona												
Funkcji sklejanych												
Thielego												
Paszkowskiego												

Program MATLAB, dzięki swojej funkcji bibliotecznej **interp1**, umożliwia dokonanie interpolacji funkcji jednej zmiennej $y = f(x)$ w punktach x_i nie będących węzłami

$$y_i = \text{interp1}(z, y_1, x_i, \text{'metoda'}) \quad (z, y_1) - \text{węzły interpolacji}$$

następującymi metodami:

- **'linear'** - interpolacja liniowa
- **'spline'** - interpolacja funkcjami sklejonymi trzeciego stopnia
- **'cubic'** - interpolacja wielomianami trzeciego rzędu

We wszystkich przypadkach elementy wektora **z** muszą stanowić ciąg rosnący, natomiast trzecią metodę należy stosować tylko dla węzłów równoodległych. W składni polecenia można pominąć nazwę metody; wówczas metodą domyślną jest interpolacja liniowa.

Przebieg ćwiczenia MATLAB:

- 1.) wyznaczyć wartości funkcji interpolowanej $y = f(x)$ i narysować jej wykres w całym przedziale interpolacji $<-1;1>$ z krokiem **0.01**
- 2.) dobrać krok dla węzłów interpolacji z (kolejno dla 2, 3, 5 i 9 węzłów)
- 3.) wyznaczyć wartości y_1 funkcji $y = f(x)$ w węzłach z
- 4.) dokonać interpolacji funkcji $y = f(x)$ w punktach x_i dla, węzłów (z, y_1), używając polecenia **interp1** (wyznaczany jest wektor y_i wartości funkcji interpolującej w punktach x_i)
- 5.) wyznaczyć maksymalny bezwzględny błąd interpolacji (wartość bezwzględną z maksimum różnicy pomiędzy funkcją interpolowaną a interpolującą); wyniki zamieścić w tabeli
- 6.) narysować wykresy funkcji interpolowanej i funkcji interpolującej w jednym układzie współrzędnych (zaznaczyć * węzły interpolacji), natomiast wykres błędu interpolacji w drugim; wykresy i napisany program zamieścić w sprawozdaniu

Przykłady

1. Dla wartości zapisanych w tabeli dokonać interpolacji liniowej z krokiem **0,1** a następnie narysować wykres, przy czym wartości z tabeli zaznaczyć na wykresie *.

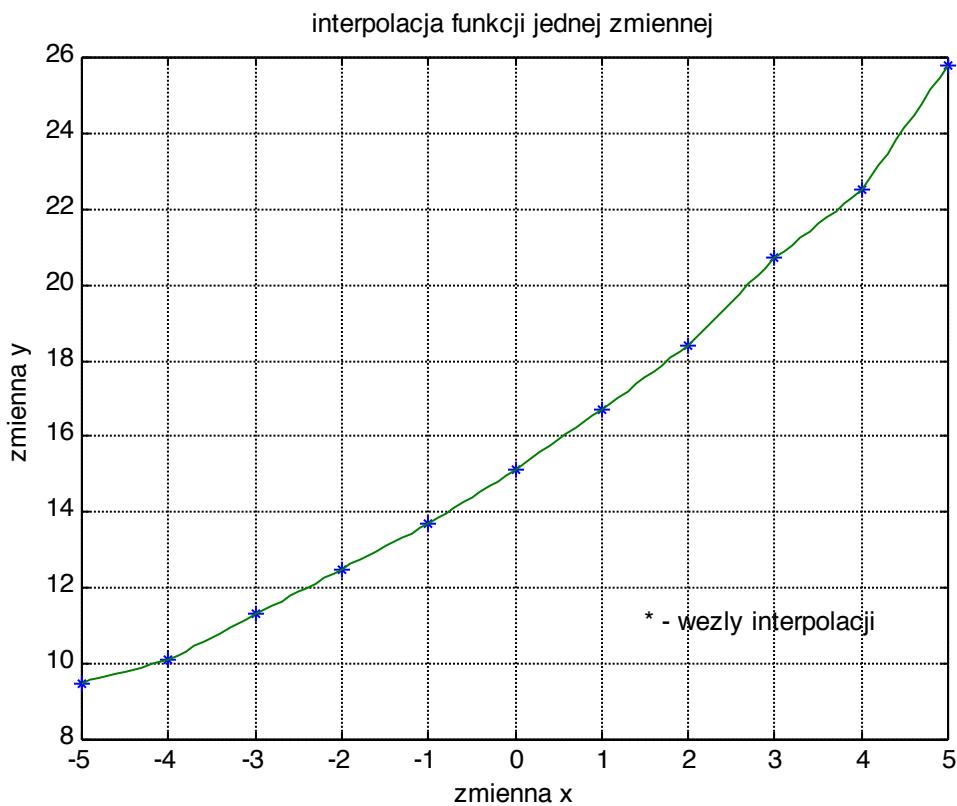
x	-5	-4	-3	-2	-1	0	1	2	3	4	5
y	9,5	10,1	11,3	12,5	13,7	15,1	16,7	18,4	20,7	22,5	25,8

```
%%interpolacja funkcji jednej zmiennej
```

```

x=-5:1:5
y=[9.5 10.1 11.3 12.5 13.7 15.1 16.7 18.4 20.7 22.5 25.8]
xi=-5:0.1:5
yi=interp1(x,y,xi,'linear')

plot(x,y,'*',xi,yi)
grid on
title('interpolacja funkcji jednej zmiennej')
xlabel('zmienna x')
ylabel('zmienna y')
text(1.5,11,'* - wezly interpolacji')
```



2. Dokonać interpolacji liniowej

funkcji $y = x^2 \sin(\Pi x)$ w przedziale $< -1; 4 >$ z

krokiem **0,5**. Narysować wykres danej funkcji i funkcji przybliżającej w jednym układzie współrzędnych natomiast wykres błędu interpolacji w drugim; węzły interpolacji zaznaczyć *. Wyznaczyć maksymalną wartość bezwzględnego błędu interpolacji w rozpatrywanym przedziale.

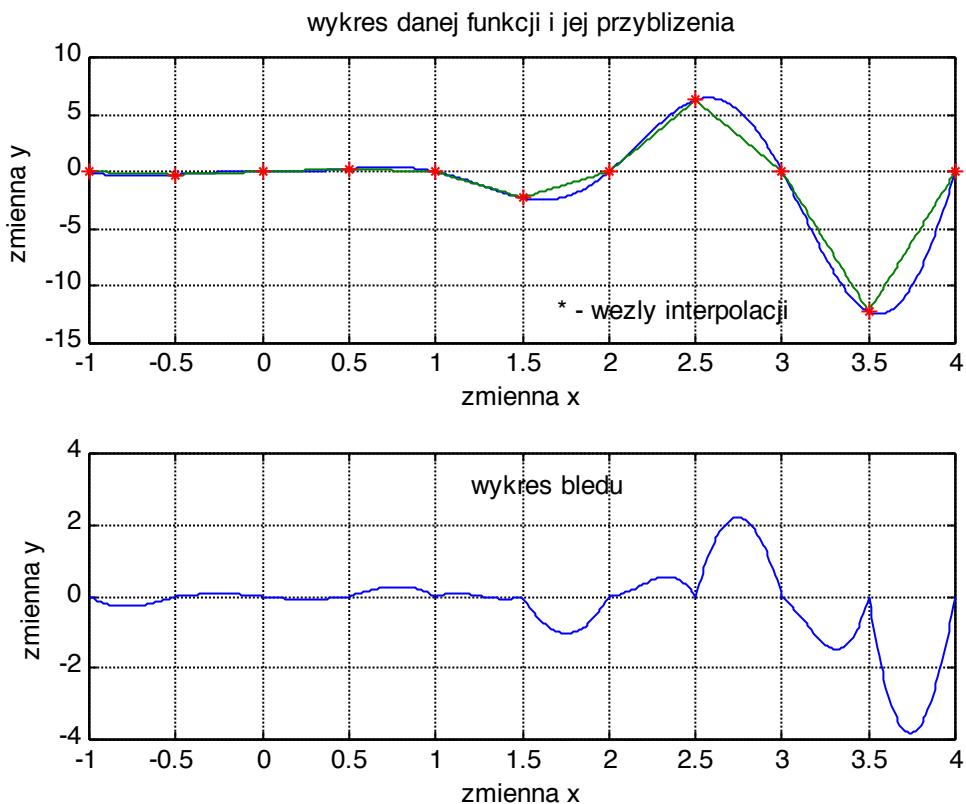
```
%%interpolacja funkcji jednej zmiennej
```

```
x=-1:0.01:4
y=(x.^2).*sin(pi*x)
z=-1:0.5:4
y1=(z.^2).*sin(pi*z)
yi=interp1(z,y1,x)

bl=y-yi
blm=max(abs(bl))

subplot(2,1,1)
plot(x,y,x,yi,z,y1,'*')
grid on
title('wykres danej funkcji i jej przyblizenia')
xlabel('zmienna x')
ylabel('zmienna y')
text(1.7,-12.5,'* - węzły interpolacji')

subplot(2,1,2)
plot(x,bl)
grid on
title('wykres błędu')
xlabel('zmienna x')
ylabel('zmienna y')
```



Maksymalna wartość błędu interpolacji w rozpatrywanym przedziale wynosi: blm = 3,8265

Ćwiczenie nr 2

Aproksymacja

Aproksymacja jest to przybliżanie funkcji za pomocą wielomianów.

Dla danej funkcji $F(x)$ określonej w przedziale $\langle a, b \rangle$ poszukiwana jest funkcja $f(x)$ dająca najmniejsze max różnicy pomiędzy funkcją $F(x)$ a $f(x)$ w całym przedziale $\langle a, b \rangle$:

$$\| F(x) - f(x) \| = \sup_{x \in \langle a, b \rangle} | F(x) - f(x) |$$

Aproksymacja jednostajna jest to aproksymacja funkcji z przestrzeni $C(T)$ funkcji rzeczywistych ciągłych w ustalonym zbiorze domkniętym T zgodnie z normą:

$$\| f \|_{\infty} = \max_{x \in T} | f(x) |$$

tzn. poszukiwany jest wielomian optymalny p_f taki, że:

$$\| f - p_f \|_{\infty} \leq \| f - q \|_{\infty} \quad \text{dla dowolnego } q$$

Znalezienie wielomianu optymalnego nie jest łatwe, dlatego często zastępuje się go wielomianem prawie optymalnym.

Błąd aproksymacji

Dana jest seria N danych (np. wyniki pomiarów) $x = [x_1, \dots, x_n]^T$ i odpowiadająca jej seria N wielkości $y = [y_1, \dots, y_n]^T$. Zadaniem aproksymacji jest znalezienie funkcji $f(x)$ przybliżającej w sposób optymalny zależność pomiędzy x i y . Błędy przybliżenia są sumowane po N pomiarach, otrzymuje się wówczas tzw. odchylenie średniokwadratowe:

$$J = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i))^2$$

ważne jest, aby wartość J była możliwie jak najmniejsza.

Dokonywana jest aproksymacja funkcji y za pomocą wielomianu $W(x)$:

$$W(x) = a_r x^r + a_{r-1} x^{r-1} + \dots + a_1 x + a_0$$

Na podstawie posiadanych danych $y = f(x)$

x_i	x_1	x_2	x_N
y_i	y_1	y_2	y_N

konstruowana są:

- macierz wartości X o wymiarach $N \times (r+1)$;
gdzie N - ilość danych (x_i, y_i), r - stopień wielomianu przybliżającego $W(x)$;
- macierz współczynników wielomianu a ;
- wektor wartości y :

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^r & \mathbf{x}_1^{r-1} & \dots & 1 \\ \mathbf{x}_2^r & \mathbf{x}_2^{r-1} & \dots & 1 \\ \dots & \dots & \dots & \dots \\ \mathbf{x}_N^r & \mathbf{x}_N^{r-1} & \dots & 1 \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} a_r \\ a_{r-1} \\ \dots \\ a_0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix}$$

Iloczyn $\mathbf{X}\mathbf{a}$ daje kolumnowy wektor wartości wielomianu \mathbf{W} dla poszczególnych danych \mathbf{x}_i . Szukany jest taki wektor \mathbf{a} , aby $\mathbf{X}\mathbf{a}$ było jak najbliższe wektorowi \mathbf{y} :

$$\min_a \sum_{i=1}^N (y_i - x_i a)^2 = \min_a \| \mathbf{y} - \mathbf{X}\mathbf{a} \|$$

$$J = \frac{1}{N} \| \mathbf{y} - \mathbf{X}\mathbf{a} \|_2$$

poprzez rozwiązywanie równania:

$$\mathbf{a} = \mathbf{X}\backslash\mathbf{y}$$

minimalizowany jest średniokwadratowy błąd przybliżenia J .

Tę metodę aproksymacji w MATLAB-ie realizuje funkcja **polyfit**:

$$\mathbf{a} = \text{polyfit}(\mathbf{x}, \mathbf{y}, r)$$

r - stopień wielomianu

Funkcja ta dla danych wektorów \mathbf{x} i \mathbf{y} znajduje wektor współczynników \mathbf{a} wielomianu stopnia r przybliżającego najlepiej w sensie średniokwadratowym zależność pomiędzy wartościami \mathbf{x} a \mathbf{y} .

Dla $r = 1$ otrzymuje się najprostszą metodę aproksymacji która nazywana jest regresją liniową; jest to aproksymacja za pomocą funkcji liniowej.

Aby otrzymać wartości wielomianu przybliżającego $\mathbf{W}(\mathbf{x})$ należy posłużyć się funkcją MATLAB-a **polyval**:

$$\mathbf{p} = \text{polyval}(\mathbf{a}, \mathbf{x})$$

Funkcja ta wyznacza wartości wielomianu o współczynnikach określonych wektorem \mathbf{a} dla wszystkich elementów wektora \mathbf{x} (macierzy X lub liczby) a otrzymane wartości umieszcza w wektorze \mathbf{p} lub macierzy \mathbf{P} .

Przebieg ćwiczenia MATLAB:

- 1.) wyznaczyć wartości funkcji aproksymowanej $y = f(x)$ i narysować jej wykres w całym przedziale aproksymacji $<-1, 1>$
- 2.) zmieniając kolejno, zgodnie z tabelką, stopień wielomianu, wyznaczyć współczynniki wielomianów aproksymujących używając funkcji **polyfit**
- 3.) dla danego stopnia wielomianu wyznaczyć wartości wielomianu aproksymującego wykorzystując funkcję **polyval**
- 4.) dla danego stopnia wielomianu wyznaczyć maksymalny błąd bezwzględny aproksymacji (wartość bezwzględną z maksimum różnicy pomiędzy funkcją aproksymowaną a wielomianem aproksymującym)
- 5.) dla najniższego stopnia wielomianu narysować wykresy funkcji aproksymowanej ($y = f(x)$) i wielomianu aproksymującego w jednym układzie współrzędnych a wykres bezwzględnego błędu aproksymacji w drugim; wykresy i napisany program zamieścić w sprawozdaniu

Przykład

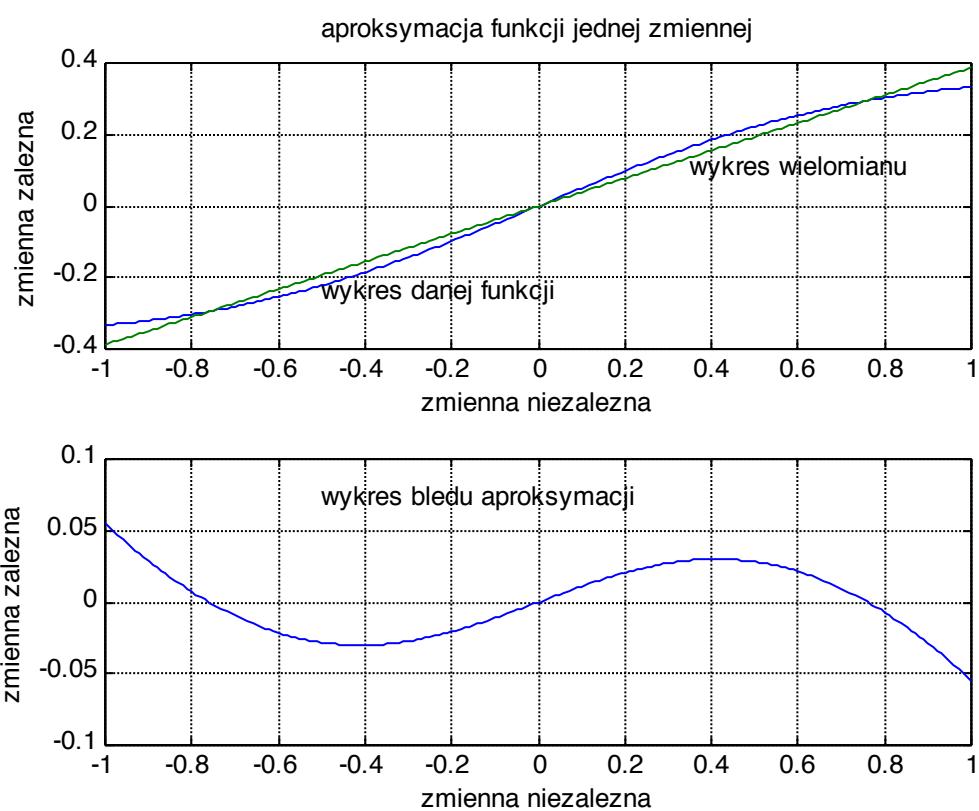
- Dokonać aproksymacji średniokwadratowej funkcji $y = \frac{x}{x^2 + 2}$ wielomianem 2-go stopnia w przedziale $<-1;1>$ z krokiem 0,01. Narysować wykres danej funkcji i funkcji przybliżającej w jednym układzie współrzędnych natomiast wykres błędu aproksymacji w drugim. Wyznaczyć maksymalną wartość bezwzględnego błędu aproksymacji w rozpatrywanym przedziale.

%%aproksymacja funkcji jednej zmiennej

```
x=-1:0.01:1
y=x./(x.^2+2)
r=2
a=polyfit(x,y,r)
p=polyval(a,x)
bl=y-p
blm=max(abs(bl))

subplot(2,1,1)
plot(x,y,x,p)
grid on
title('aproksymacja funkcji jednej zmiennej')
text(0.35,0.1,'wykres wielomianu')
text(-0.5,-0.25,'wykres danej funkcji')
xlabel('zmienna niezależna')
ylabel('zmienna zależna')

subplot(2,1,2)
plot(x,bl)
grid on
text(-0.5,0.07,'wykres błędu aproksymacji')
xlabel('zmienna niezależna')
ylabel('zmienna zależna')
```



Maksymalna wartość błędu aproksymacji w rozpatrywanym przedziale wynosi: blm = 0,0546

Ćwiczenia nr 3 i 4

ZERA FUNKCJI I WIELOMIANÓW

Ćwiczenie nr 3

Zera funkcji i zera wielomianów

Zera wielomianów

Analityczne wyznaczanie rozwiązań równań o skomplikowanych funkcjach jest często niemożliwe, dlatego duże znaczenie mają przybliżone iteracyjne metody rozwiązywania równań. Metoda iteracyjna polega na obliczaniu kolejnych przybliżeń wartości zera, wykorzystującym wcześniej obliczone przybliżenia. Przedstawiono trzy metody iteracyjne wyznaczania zer funkcji:

- **bisekcji (połowienia)**
- **siecznych**
- **Newtona**

Metoda **bisekcji** pozwala znaleźć zera funkcji o nieparzystej krotności. Wykorzystuje się tutaj fakt, że wartość funkcji zmienia znak w otoczeniu takiego zera. Po ustaleniu przedziału $[a, b]$ zawierającego jedno takie zero (np. metodą tablicowania) jako kolejne przybliżenie przyjmuje się środek przedziału $x = (a + b)/2$, a następnie rozpatruje się ten przedział na krańcach którego funkcja ma przeciwnie znaki. Postępowanie to kontynuuje się tak długo, aż zostanie osiągnięta założona dokładność.

Miarą dokładności może być długość przedziału zawierającego poszukiwane zero lub $|f|$ w środku przedziału. Metoda bisekcji jest zawsze zbieżna.

Również w metodzie siecznych i w metodzie Newtona wykorzystywany jest fakt zmiany znaku funkcji w otoczeniu zera. W metodzie **siecznych**, po ustaleniu początkowego przybliżenia $[a, b]$, prowadzona jest sieczna do wykresu funkcji w punkcie a . Sieczna dzieli przedział $[a, b]$ na dwie części, wybierany jest ten podprzedział na krańcach którego funkcja przyjmuje przeciwnie znaki. Postępowanie kontynuowane jest do osiągnięcia założonej dokładności.. Jako wartość zera przyjmuje się punkt przecięcia siecznej z osią odciętych.

W metodzie **Newtona** postępuje się podobnie z tym, że w punkcie początkowym przedziału $[a, b]$ prowadzona jest styczna do wykresu funkcji. Jako wartość zera przyjmuje się punkt przecięcia stycznej z osią odciętych. Elementem decydującym o zbieżności metody Newtona jest właściwy dobór przybliżenia początkowego.

Metody siecznych i Newtona mogą niekiedy być rozbieżne.

Moduł wartości funkcji dla kolejnych iteracji

Metoda	Liczba iteracji					
	1	2	3	4	5	6
Bisekcji						
Siecznych						
Newtona						

W programie MATLAB istnieje funkcja **roots(a)**, gdzie a jest macierzą wierszową zawierającą współczynniki wielomianu, dzięki której można wyznaczyć wektor z zawierający zera (zarówno rzeczywiste jak i zespolone) wielomianu $W(x)$ o znanych współczynnikach $a_0, a_1, \dots, a_{n-1}, a_n$.

Jeżeli $W(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-1}x + a_n$

to $a = [a_0 \ a_1 \ \dots \ a_{n-1} \ a_n]$

wówczas $z = \text{roots}(a)$

Przebieg ćwiczenia MATLAB:

- 1.) określić wektor **a** współczynników wielomianu **W(x)**
- 2.) wykorzystując polecenie **roots** wyznaczyć zera wielomianu **W(x)**
- 3.) narysować wykres wielomianu w przedziale zawierającym zera (zera zaznaczyć „o”); wykresy i napisany program zamieścić w sprawozdaniu

Przykład

1. Wyznaczyć wszystkie zera następującego wielomianu i narysować jego wykres w przedziale zawierającym zera (zera zaznaczyć *):

$$w(x) = x^6 - x^5 - 7x^4 + 15x^3 - 6x^2 - 14x + 12$$

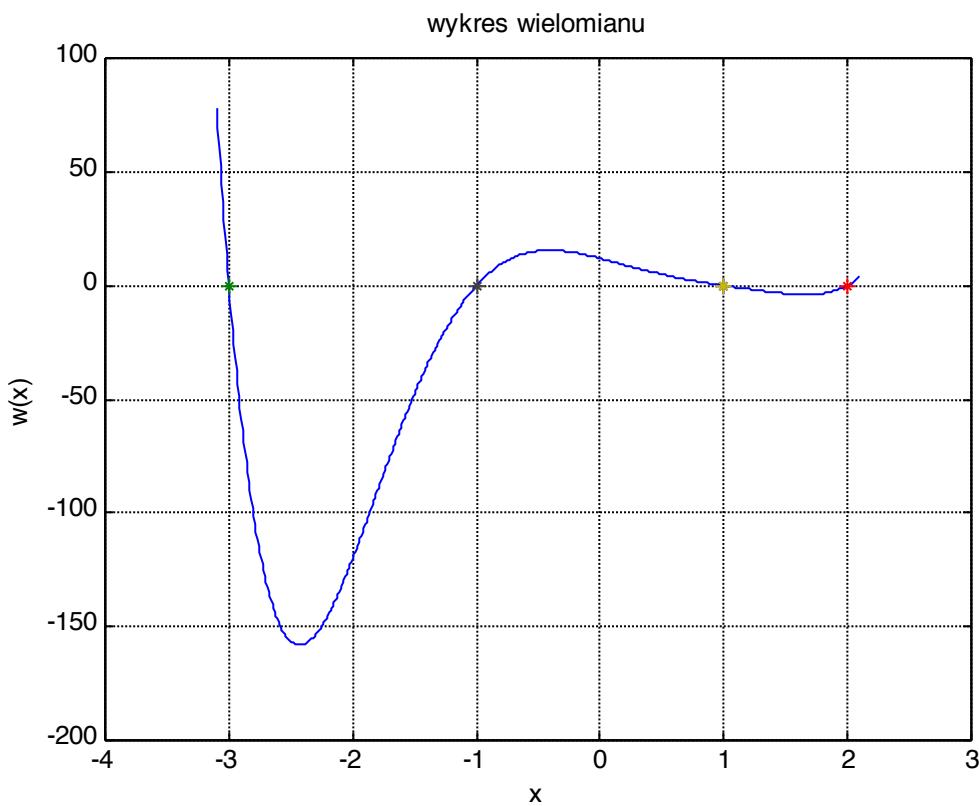
%%zera wielomianu

```
a=[1 -1 -7 15 -6 -14 12]      %a - wektor zawierający współczynniki wielomianu
z=roots(a)                      %z - wektor zawierający zera wielomianu
```

```
x=-3.1:0.01:2.1
```

```
y=x.^6-x.^5-7*x.^4+15*x.^3-6*x.^2-14*x+12
```

```
plot(x,y,real(z),0,'*')          %real(z) - wektor zawierający części rzeczywiste wektora z
grid on
title('wykres wielomianu')
xlabel('x')
ylabel('w(x)')
```



Wartości zer rozpatrywanego wielomianu są następujące:

$z =$

- 3.000
- 2.000
- $1 + 1.000i$
- $1 - 1.000i$
- 1.000
- 1.000

Zera funkcji

Istotnym zagadnieniem w metodach iteracyjnych jest znajdowanie przybliżeń początkowych. W przypadku, gdy nie ma żadnych informacji o zerach funkcji (o ich liczbie i krotności) a interesują nas zera rzeczywiste, wówczas jedyną metodą umożliwiającą wyznaczenie wartości początkowych jest metoda tablicowania polegająca na znalezieniu w danym przedziale $[a, b]$ podprzedziałów o długości h na krańcach których funkcja ma różne znaki. Jeżeli funkcja jest ciągła, to w takich przedziałach ma nieparzystą liczbę zer.

Metody iteracyjne umożliwiają znalezienie zer wielokrotnych o nieparzystej krotności.

Jeżeli x_0 jest zerem krotności k to spełniona jest zależność:

$$f(x_0) = f'(x_0) = \dots = f^{k-1}(x_0) = 0 \quad \text{natomiast} \quad f^k(x_0) \neq 0$$

Metoda tablicowania nie wykrywa zer o parzystej krotności. W tym przypadku należy rozpatrywać funkcję: $u(x) = f(x)/f'(x)$, której wszystkie zera są pojedyncze i jednocześnie są zerami funkcji $f(x)$.

Zakładana dokładność ϵ nie może być mniejsza niż 10^{-18} ; obliczenia zostają przerwane, jeżeli w metodzie bisekcji długość przedziału zawierającego poszukiwane zero będzie mniejsza od ϵ (za wartość zera przyjmowany jest środek ostatniego przedziału).

W pozostałych metodach zakończenie iteracji następuje, gdy spełniony jest warunek:

$$\frac{m | x_{n+1} - x_n |}{1 - m} \leq \epsilon \quad \text{gdzie} \quad m = \left| \frac{x_{n+1} - x_n}{x_n - x_{n-1}} \right|$$

jako wartość zera przyjmowane jest przybliżenie x_{n+1} .

Wartości zer dla funkcji f i funkcji f/pf

W programie MATLAB rzeczywiste zera funkcji można wyznaczyć wykorzystując polecenie **fzero**. W tym celu należy w skrypcie (z rozszerzeniem ‘m’) zadeklarować rozpatrywaną funkcję w sposób następujący:

```
function y = f(x)
y =
```

Następnie należy określić przybliżenie początkowe **x0**, czyli punkt wokół którego będzie poszukiwane zero, można również podać dokładność **tol** oraz parametr **w**, który, jeżeli ma wartość niezerową, powoduje wyświetlenie pośrednich obliczeń. Składnia polecenia **fzero**, które należy napisać w oknie programu, jest następująca:

```
z = fzero('plik', x0, tol, w)
```

gdzie **plik** jest to nazwa skryptu (bez rozszerzenia) zawierającego rozpatrywaną funkcję, natomiast **x0** jest to przybliżenie początkowe. Jeżeli nie zostanie podana wartość **tol** wówczas obliczenia będą wykonywane z dokładnością równą **eps** (dokładność maszynowa w MATLAB-ie) czyli $2,22 \cdot 10^{-16}$, natomiast brak parametru **w** powoduje pominięcie wyświetlania pośrednich obliczeń.

Algorytm polecenia **fzero** oparty jest na kombinacji metod bisekcji, siecznych i interpolacji.

Przebieg ćwiczenia MATLAB:

- 1.) narysować wykres funkcji $y = f(x)$ w przedziale zawierającym zera i określić przedziały zmienności funkcji
- 2.) w oddzielnym pliku zadeklarować funkcję $y = f(x)$
- 3.) dla dokładności 10^{-5} i dokładności **eps**, wykorzystując polecenie **fzero** i zmieniając odpowiednio punkt **x0** wokół którego poszukiwane jest zero, wyznaczyć wszystkie (dla funkcji oscylacyjnych 3) zera powyższej funkcji (punkt **x0** nie może być zerem funkcji)

Przykład

- W przedziale $<-3, 3>$ wyznaczyć wszystkie zera funkcji: $y = (x+2)\arctan(x-1)$ i narysować jej wykres w tym przedziale; miejsca zerowe zaznaczyć **o**.

```
%%zera funkcji
```

```
%funkcję y=f(x) zadeklarować w skrypcie o nazwie np. zera.m:
```

```
function y=f(x)
y=(x+2).*atan(x-1)
```

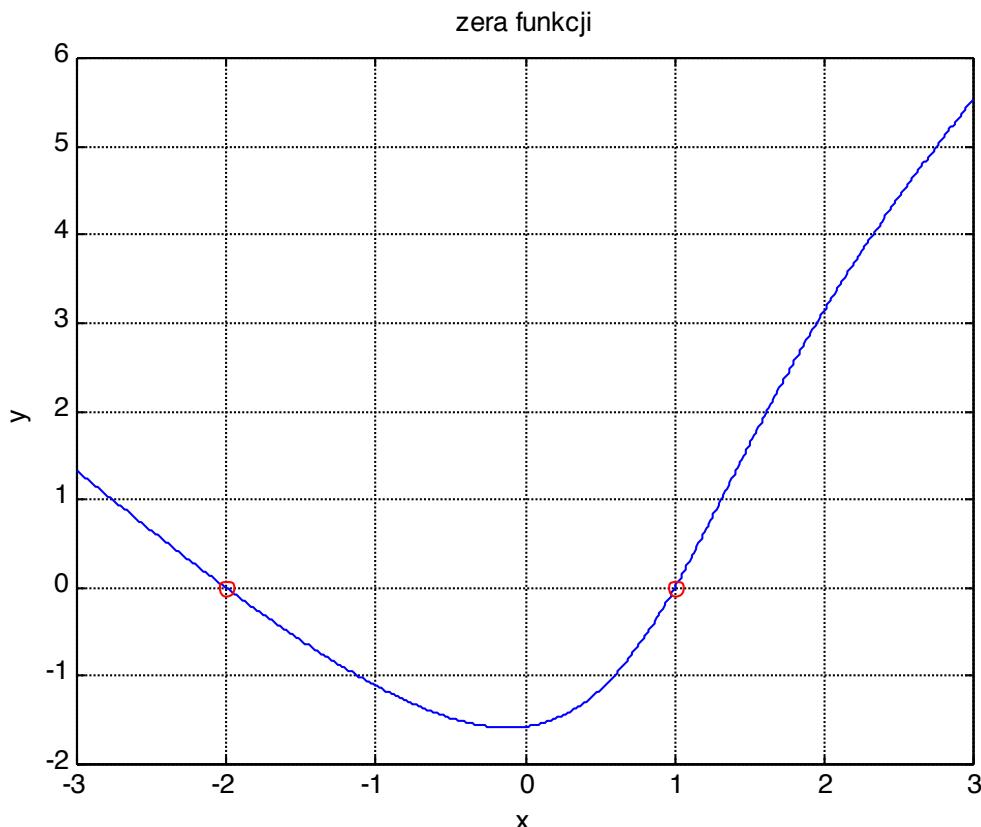
```
%ustalić punkty w otoczeniu których poszukiwane będą zera (np. 3 i -3)
%w oknie MATLAB-a kolejno napisać i wywołać polecenia:
```

```
z1=fzero('zera',-3)          %obliczenia dokonywane są z dokładnością  $2,22 \cdot 10^{-16}$ 
z2=fzero('zera',3,1e-6,1)      %obliczenia dokonywane są z dokładnością  $10^{-6}$ 
```

```
%w nowym skrypcie zamieścić polecenia służące do narysowania wykresu:
```

```
x=-3:0.01:3
y=(x+2).*atan(x-1)

plot(x,y,z1,0,'ro',z2,0,'ro')
grid on
title('zera funkcji')
xlabel('x')
ylabel('y')
```



W przedziale $<-3; 3>$ funkcja ta posiada dwa zera rzeczywiste:

$$\begin{aligned} z1 &= -2 \\ z2 &= 1 \end{aligned}$$

Ćwiczenie nr 4

Zera wielomianów

Wielomian stopnia n o rzeczywistych współczynnikach posiada dokładnie n zer, przy czym dla każdego zera zespolonego istnieje dokładnie jedno zero z nim sprzężone. Przedziały zawierające wszystkie zera rzeczywiste można otrzymać z twierdzenia Maclaurina. Można również znaleźć promień okręgu o środku w początku układu współrzędnych $(0, 0)$ zawierającego wszystkie zera zarówno rzeczywiste jak i zespolone.

Podeczas obliczania wszystkich zer wielomianu bardzo pomocna jest możliwość eliminacji z danego wielomianu obliczonego już zera (deflacja wielomianu), przy czym istnieją wielomiany źle uwarunkowane dla których deflacja powoduje znaczne zniekształcenie kolejnych obliczanych zer.

Przebieg ćwiczenia MATLAB:

- 1.) narysować wykres wielomianu $\mathbf{W(x)}$ w przedziale zawierającym wszystkie zera
- 2.) określić wektor \mathbf{a} współczynników wielomianu $\mathbf{W(x)}$
- 3.) wykorzystując polecenie **roots** wyznaczyć wszystkie zera wielomianu $\mathbf{W(x)}$
- 4.) zmieniając dokładność w poleceniu **fzero** (10^{-3} , 10^{-6} , 10^{-9} , 10^{-12} i 10^{-15}) wyznaczać kolejno wartość zera rzeczywistego

Ćwiczenia nr 5 i 6

ALGEBRA LINIOWA

Ćwiczenie nr 5

Algebra liniowa (układy równań liniowych)

Układ równań liniowych o postaci:

$$\sum_{i,j=1}^n a_{ij}x_i = b_i \quad i = 1, 2, \dots, n$$

można przedstawić w postaci macierzowej:

$$\mathbf{Ax} = \mathbf{b}$$

$\mathbf{A} = [\mathbf{a}_{ij}]$ - macierz układu
 $\mathbf{b} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]^T$ - wektor prawych stron
 $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ - wektor rozwiązania

Jeżeli macierz \mathbf{A} jest macierzą nieosobliwą ($\det(\mathbf{A}) \neq 0$), wówczas rozpatrywany układ równań ma dokładnie jedno rozwiązanie:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \quad \mathbf{A}^{-1} \text{ - macierz odwrotna do macierzy } \mathbf{A}$$

Rozwiązanie powyższego układu równań można uzyskać trzema:

- **eliminacja Gaussa bez wyboru elementu głównego**
- **eliminacja Gaussa z częściowym wyborem elementu głównego**
- **za pomocą macierzy odwrotnej wyznaczonej metodą eliminacji Gaussa z częściowym wyborem elementu głównego**

Podczas eliminacji Gaussa dokonywany jest rozkład macierzy \mathbf{A} na iloczyn macierzy trójkątnych \mathbf{L} i \mathbf{U} , gdzie \mathbf{L} - macierz trójkątna dolna (z jedynkami na przekątnej) i \mathbf{U} - macierz trójkątna górna:

$$\mathbf{Ax} = \mathbf{b} \quad i \quad \mathbf{A} = \mathbf{LU} \quad \Rightarrow \quad \mathbf{LUx} = \mathbf{b}$$

dokonując podstawienia: $\mathbf{Ux} = \mathbf{y} \quad \Rightarrow \quad \mathbf{Ly} = \mathbf{b}$

otrzymujemy układ równań:

$$\begin{cases} \mathbf{Ux} = \mathbf{y} \\ \mathbf{Ly} = \mathbf{b} \end{cases}$$

W przypadku eliminacji Gaussa z częściowym wyborem elementu głównego dokonywany jest rozkład trójkątny macierzy ze zmienioną kolejnością równań.

Jeżeli elementy macierzy trójkątnych wyznaczonych metodą eliminacji Gaussa bez wyboru elementu głównego są wszystkie nieujemne, to należy się spodziewać, że rozwiązanie uzyskane bez wyboru elementu głównego będzie tak samo dokładne, albo nawet dokładniejsze od rozwiązania obliczonego z częściowym wyborem elementu głównego.

Każda macierz \mathbf{A} można przedstawić w postaci rozkładu **SVD**:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T \quad \text{gdzie } \mathbf{U} \text{ i } \mathbf{V} \text{ - macierze ortogonalne } (\mathbf{U}^{-1} = \mathbf{U}^T; \mathbf{V}^{-1} = \mathbf{V}^T) \\ \Sigma \text{ - macierz przekątniowa}$$

$$\Sigma = \text{diag}(\sigma_i) \quad \sigma_i \text{ - wartości szczególne (osobliwe) macierzy } \mathbf{A}$$

Jeżeli macierz \mathbf{A} jest macierzą nieosobliwą, to jej wszystkie wartości szczególne są dodatnie i uporządkowane nierośnaco. Jeżeli którakolwiek wartość szczególna macierzy jest mniejsza od 0, to macierz ta jest macierzą osobliwą.

Moduł (wartość bezwzględna wyznacznika macierzy \mathbf{A}) jest iloczynem jej wszystkich wartości szczególnych

$$|\det(\mathbf{A})| = \sigma_1 \sigma_2 \dots \sigma_n$$

Dokładność numeryczne obliczonego rozwiązania rozpatrywanego układu równań liniowych zależy od zastosowanego algorytmu i uwarunkowania zadania.

Jako **wskaźnik uwarunkowania zadania** rozwiązywania układu równań liniowych przyjmuje się następującą zależność:

$$\mathbf{cond}(\mathbf{A}) = \|\mathbf{A}\| * \|\mathbf{A}^{-1}\| \quad \text{gdzie } \|\mathbf{A}\| = \sigma_1 \\ (\|\mathbf{A}\| - \text{norma macierzy } \mathbf{A} \text{ zdefiniowana jako największa wartość szczególna } \sigma_1 \text{ tej macierzy})$$

Im większa jest wartość **cond(A)**, tym rozwiązanie układu równań jest bardziej wrażliwe na małe zmiany (zaburzenia) elementów macierzy **A** i wektora **b** oraz na błędy zaokrągleń powstające w trakcie obliczeń.

Oprócz powyższych rozpatrywane są jeszcze następujące wskaźniki i wyznaczniki:

Wskaźniki z SVD:

1.) spektralny:

$$\mathbf{cond}_2(\mathbf{A}) = \sigma_1 / \sigma_n \quad \sigma_1 - \text{największa wartość szczególna macierzy } \mathbf{A} \\ \sigma_n - \text{najmniejsza wartość szczególna macierzy } \mathbf{A}$$

2.) dla normy Frobeniusa:

$$\mathbf{cond}_F(\mathbf{A}) = \left(\left(\sum_j \sigma_j^2 \right) \left(\sum_j \frac{1}{\sigma_j^2} \right) \right)^{\frac{1}{2}}$$

3.) współczynniki numerycznej osobliwości macierzy:

$$\mathbf{tol1} = n * \text{macheps} * \sigma_1$$

n - stopień macierzy **A**

σ₁ - największa wartość szczególna macierzy **A**

macheps – dokładność maszynowa

(tzn. najmniejsza liczba dodatnia reprezentowana w komputerze taka, że $1 + \text{macheps} > 1$)

Jeżeli którakolwiek wartość szczególna macierzy **A** jest mniejsza niż **tol1** to macierz **A** jest numerycznie osobliwa.

$$\mathbf{tol} = 10 * n * \text{macheps} * \|\mathbf{A}\|_F$$

$\|\mathbf{A}\|_F$ - norma Frobeniusa macierzy **A** (pierwiastek sumy kwadratów wszystkich elementów macierzy)

z

Jeżeli moduł któregokolwiek elementu głównego macierzy **A** jest mniejszy od **tol**, to macierz ta jest macierzą numerycznie osobliwą.

4.) Współczynnik numerycznej poprawności algorytmu:

$$\mathbf{wsp} = \|\mathbf{r}\|_2 / (\text{macheps} * \|\mathbf{y}\|_2 * \|\mathbf{A}\|_F)$$

r = **b** - **Ay** - wektor residuum

$\|\mathbf{r}\|_2$ - druga norma wektora residuum (pierwiastek z sumy kwadratów jego współrzędnych)

y - rozwiązanie numeryczne

$\ y\ _2$	- druga norma wektora rozwiązania numerycznego (pierwiastek z sumy kwadratów jego współrzędnych)
$\ \mathbf{A}\ _F$	- norma Frobeniusa macierzy \mathbf{A}
macheps	- dokładność maszynowa

Jeżeli wsp jest rzędu n bądź n^2 , to przyjmuje się, że algorytm jest numerycznie poprawny.

5.) Błąd względny rozwiązania:

$$W = \|x - y\|_2 / \|x\|_2$$

x - rozwiązanie dokładne
 y - rozwiązanie obliczone

Jeżeli wartości przekraczają zakres liczb reprezentowanych w komputerze, to wyświetlana jest wartość 1E38.

W trakcie ćwiczenia liczony jest błąd względny rozwiązania; aby można było wyznaczyć dokładną wartość tego błędu należy ustalić rozwiązanie x . Ustalona wartość rozwiązania jest następująca: $x[i] = i$; następnie mnożąc macierz układu A przez wektor rozwiązania x otrzymuje się wektor prawych stron b . Na podstawie dokładnej wartości macierzy układu A i wektora prawych stron b wyznaczane są wektory rozwiązania numerycznego x_1 , x_2 i x_3 trzema wymienionymi wyżej metodami.

W programie MATLAB dostępne są m.in. następujące funkcje dotyczące macierzy:

inv(A)	- odwracanie macierzy
A\b	- rozwiązywanie układu równań liniowych metodą eliminacji Gaussa z częściowym wyborem elementu głównego (A - macierz układu, b - wektor prawych stron)
det(A)	- wyznacznik macierzy
[L,U] = lu(A)	- rozkład macierzy na macierz trójkątną górną L i macierz trójkątną dolną U
[U,S,V] = svd(A)	- rozkład SVD macierzy (A = USV ^T , U , V - macierze ortogonalne, S - macierz przekątniowa z wartościami szczególnymi σ_i na przekątnej)
s = svd(A)	- wektor wartości szczególnych
c = cond(A)	- wskaźnik uwarunkowania zadania rozwiązywania układów równań liniowych równy iloczynowi największych wartości szczególnych macierzy A i A ⁻¹
eps	- dokładność maszynowa
norm(x)	- norma wektora równa pierwiastkowi z sumy kwadratów jego współrzędnych
norm(A,'fro')	- norma Frobeniusa macierzy równa pierwiastkowi z sumy kwadratów elementów
jej	

Ponadto można obliczyć czas wyznaczania rozwiązania (**tic** - początek pomiaru czasu, **toc** - koniec pomiaru czasu):

```
tic, x3=inv(A)*b, toc
tic, x2=A\b, toc
```

Przebieg ćwiczenia MATLAB:

- 1.) na podstawie zależności określającej elementy macierzy utworzyć macierz stopnia **n = 5**
- 2.) zdefiniować wektor rozwiązania **x** i obliczyć wektor prawych stron **b**
- 3.) wyznaczyć macierz **A1** odwrotną do macierzy **A** i rozwiązanie **x3** metodą odwracania macierzy
- 4.) obliczyć wyznacznik macierzy **A**, dokonać rozkładu tej macierzy na macierze trójkątne i wyznaczyć rozwiązanie **x2** metodą eliminacji Gaussa z częściowym wyborem elementu głównego
- 5.) dla obu metod obliczyć błąd bezwzględny (**bl2** = **x** - **x2** i **bl3** = **x** - **x3**) i wektory residuum (**r2** = **b** - **A*x2** i **r3** = **b** - **A*x3**)
- 6.) dokonać rozkładu **SVD** macierzy **A** i wyznaczyć najmniejszą σ_n i największą σ_1 wartość szczególną tej macierzy oraz wskaźnik uwarunkowania **cond(A)**
- 7.) wyznaczyć dokładność maszynową **eps** i obliczyć współczynniki numerycznej osobliwości macierzy (**tol** = $10 * n * \text{eps} * \text{norm}(A, 'fro')$ i **tol1** = $n * \text{eps} * \sigma_1$)
- 8.) wyznaczyć błędy względne rozwiązań (**W2** = $\text{norm}(x - x2) / \text{norm}(x)$ i **W3** = $\text{norm}(x - x3) / \text{norm}(x)$)
- 9.) obliczyć współczynniki numerycznej poprawności algorytmu (**wp2** = $\text{norm}(r2) / ((\text{eps} * \text{norm}(x2)) * \text{norm}(A, 'fro'))$ i **wp3** = $\text{norm}(r3) / ((\text{eps} * \text{norm}(x3)) * \text{norm}(A, 'fro'))$)
- 10.) wyznaczyć czas obliczeń dla obydwu metod

Wyniki obliczeń w programie MATLAB

Wskaźniki, wyznaczniki i współczynniki	Metoda (MATLAB)	
	Gauss z wyborem	Odwracanie macierzy
dokładność maszynowa (eps)		
współczynnik numerycznej poprawności algorytmu		
wyznacznik macierzy		
błąd względny W		
czas obliczeń		
współczynnik numerycznej osobliwości tol		
najmniejsza wartość szczególna		

największa wartość szczególna	
współczynnik numerycznej osobliwości toll	
wskaznik uwarunkowania (cond)	

Przykład

1. Stosując metodę eliminacji Gaussa z częściowym wyborem elementu głównego i metodę odwracania macierzy rozwiązać następujący układ równań liniowych:

$$\begin{cases} x - 2y + 5z + u = 10 \\ 2x + y - z - 2u = -3 \\ 5x + y + 2z - u = 1 \\ 3x - y + 3z - 3u = 4 \end{cases}$$

%%układy równań liniowych

```
A=[1 -2 5 1 %macierz układu  
 2 1 -1 -2  
 5 1 2 -1  
 3 -1 3 -3]  
  
b=[10 -3 1 4]' %wektor prawych stron  
  
x1=A\b %rozwiązanie metodą eliminacji Gaussa  
%z częściowym wyborem elementu głównego  
  
x2=inv(A)*b %rozwiązanie metodą odwracania macierzy
```

Rozwiązanie powyższego układu równań jest następujące:

x1 =	x2 =
-10.0000	-10.0000
20.0000	20.0000
13.0000	13.0000
-5.0000	-5.0000

Ćwiczenie nr 6

Algebra liniowa (wartości własne macierzy)

Dana jest macierz rzeczywista \mathbf{A} stopnia n , wartości własne λ tej macierzy i jej wektor własny \mathbf{x} są zdefiniowane następująco:

$$\mathbf{Ax} = \lambda \mathbf{x}$$

Jeżeli macierz \mathbf{A} jest macierzą symetryczną to wszystkie jej wartości własne są rzeczywiste i istnieje macierz ortogonalna \mathbf{Q} ($\mathbf{Q}^{-1} = \mathbf{Q}^T$) taka, że :

$$\mathbf{Q}^T \mathbf{A} \mathbf{Q} = \text{diag}(\lambda_j)$$

$\lambda_1, \dots, \lambda_n$ - wartości własne macierzy \mathbf{A}

\mathbf{q}_j - wektory własne macierzy \mathbf{A} (kolumny macierzy \mathbf{Q})

$$\mathbf{Aq}_j = \lambda_j \mathbf{q}_j$$

$j = 1, \dots, n$

Wszystkie wartości własne są pierwiastkami wielomianu charakterystycznego:

$$\det(\mathbf{A} - \lambda \mathbf{I})$$

Wartości własne macierzy symetrycznej można wyznaczyć m. in. metodami:

- **Jacobiego**
- **bisekcji**
- **QL**

Metodę Jacobiego stosuje się bezpośrednio do macierzy \mathbf{A} , aby zastosować metodę bisekcji należy najpierw przekształcić macierz \mathbf{A} przez podobieństwo ortogonalne do postaci trójkątnej symetrycznej \mathbf{B} (procedura TRIDIAG):

$$\mathbf{B} = \mathbf{Z}^T \mathbf{A} \mathbf{Z}$$

\mathbf{Z} - macierz ortogonalna ($\mathbf{Z}^{-1} = \mathbf{Z}^T$)

Macierz \mathbf{B} jako podobna do macierzy \mathbf{A} ma takie same wartości własne.

Podobnie postępuje się w przypadku metody QL, która służy do wyznaczania wartości własnych macierzy symetrycznych trójkątniowych.

Zadanie obliczania wartości własnych macierzy symetrycznej jest bardzo dobrze uwarunkowane. Oznacza to, że wartości własne nie są wrażliwe na małe zmiany elementów macierzy \mathbf{A} , ponadto przekształcenie macierzy przez podobieństwo ortogonalne nie zmienia uwarunkowania.

Wszystkie wartości własne macierzy symetrycznej dodatnio określonej są dodatnie.

Dokładność z jaką metody Jacobiego, bisekcji i QL wyznaczają najmniejsze wartości własne jest w ogólnym przypadku taka sama. Dokładność obliczeniowa sprawia, że w niektórych przypadkach uzyskuje się ujemne najmniejsze wartości własne.

Koszty metod

W programie MATLAB istnieją polecenia umożliwiające wyznaczenie wartości własnych i wektorów własnych macierzy:

$$\mathbf{L} = \text{eig}(\mathbf{A})$$

- wektor \mathbf{L} zawiera wartości własne macierzy kwadratowej \mathbf{A}

$$[\mathbf{V}, \mathbf{D}] = \text{eig}(\mathbf{A})$$

- macierz \mathbf{D} zawiera wartości własne macierzy \mathbf{A} umieszczone na przekątnej

- kolumny macierzy \mathbf{V} są wektorami własnymi macierzy \mathbf{A}

Zgodnie z definicją wartości własnych i wektorów własnych dla macierzy symetrycznej \mathbf{A} istnieje macierz ortogonalna \mathbf{V} taka, że $\mathbf{V}^T = \mathbf{V}^{-1}$ i wówczas spełnione są zależności:

$$\begin{aligned}\mathbf{AV} &= \mathbf{VD} \\ \mathbf{D} &= \mathbf{V}^T \mathbf{AV}\end{aligned}$$

Przebieg ćwiczenia MATLAB:

- 1.) zadeklarować macierz \mathbf{A} stopnia $n = 4$ i wyznaczyć jej wartości własne
- 2.) porównać wyniki z obydwu programów
- 3.) policzyć iloczyny \mathbf{AV} oraz \mathbf{VD} i porównać je ze sobą
- 4.) wyznaczyć iloczyn $\mathbf{V}^T \mathbf{AV}$ i porównać z macierzą \mathbf{D}

Przykład

1. Wyznaczyć wartości własne i wektory własne macierzy kwadratowej rzędu 5, której elementy określone są zależnością:

$$a(i,j) = 1/(i+j-1)$$

%%Wartości własne i wektory własne macierzy

```
n=5 %stopień macierzy

for i=1:n
    for j=1:n
        A(i,j)=1/(i+j-1);
    end
end

disp(A) %wyświetlenie macierzy

[V,D]=eig(A)
%V - macierz wektorów własnych macierzy A
%D - macierz zawierająca wartości własne macierzy A
```

Wygenerowana macierz A:

n =

5

1.0000	0.5000	0.3333	0.2500	0.2000
0.5000	0.3333	0.2500	0.2000	0.1667
0.3333	0.2500	0.2000	0.1667	0.1429
0.2500	0.2000	0.1667	0.1429	0.1250
0.2000	0.1667	0.1429	0.1250	0.1111

Kolumny macierzy V są wektorami własnymi macierzy A:

V =

0.0062	0.0472	0.2142	-0.6019	0.7679
-0.1167	-0.4327	-0.7241	0.2759	0.4458
0.5062	0.6674	-0.1205	0.4249	0.3216
-0.7672	0.2330	0.3096	0.4439	0.2534
0.3762	-0.5576	0.5652	0.4290	0.2098

Wartości szczególne macierzy A znajdują się na przekątnej macierzy D:

D =

0.0000	0	0	0	0
0	0.0003	0	0	0
0	0	0.0114	0	0
0	0	0	0.2085	0
0	0	0	0	1.5671

Ćwiczenia nr 7

CAŁKOWANIE

Ćwiczenie nr 7

Całkowanie

Dana jest funkcja $f(x)$ ciągła w przedziale (a, b) :

Jeżeli

$$F'(x) = f(x) \quad \text{to} \quad \int_a^b f(x) dx = F(b) - F(a) \quad F - \text{funkcja pierwotna funkcji } f$$

Całkowanie numeryczne polega na obliczeniu całki oznaczonej na podstawie funkcji podcałkowej w pewnych punktach przedziału całkowania. Odpowiednie wzory dające poszukiwaną wartość przybliżoną całki nazywane są kwadraturami.

Funkcję podcałkową zastępuje się w przedziale (a, b) funkcją interpolującą lub aproksymującą o możliwie prostej postaci (np. wielomianu) dla której znana jest funkcja pierwotna.

Punkty w których obliczane są wartości funkcji podcałkowej występującej w kwadraturze nazywane są węzłami kwadratury.

Rozpatrywane będą trzy kwadratury:

- metoda prostokątów
- metoda trapezów
- metoda Simpsona

Metoda prostokątów polega na zastąpieniu funkcji podcałkowej funkcją stałą g taką, że:

$$g(x) = f(x_0) \quad x_0 - \text{punkt przedziału } (a, b)$$

wówczas całka dana jest wzorem:

$$(b - a)f(x_0)$$

Jest to kwadratura zbudowana na jednym węźle.

Metoda trapezów polega na zastępowaniu funkcji podcałkowej funkcją liniową g , która przechodzi przez punkty $(a, f(a))$; $(b, f(b))$, wówczas wartość całki wynosi:

$$(b - a) \frac{f(a) - f(b)}{2}$$

Jest to kwadratura oparta na dwóch węzłach.

Metoda Simpsona wykorzystuje trzy punkty przedziału (a, b) : a , $\frac{a+b}{2}$, b . Na tych trzech punktach zbudowana jest parabola; wartość całki jest równa:

$$\frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right)$$

Jest to kwadratura oparta na trzech węzłach.

Zwiększając liczbę węzłów można przybliżać funkcję $f(x)$ wielomianami coraz wyższych stopni; nie jest to jednak właściwy sposób postępowania, ponieważ zwiększanie stopnia wielomianu nie zawsze poprawia dokładność otrzymywanych wyników.

Poprawę dokładności można zawsze uzyskać stosując podział przedziału **(a, b)** na podprzedziały o równej długości z zastosowaniem w każdym z podprzedziałów jednej z prostych kwadratur. Odpowiada to zastąpieniu funkcji podcałkowej linią łamaną lub odcinkami parabol. Są to tzw. kwadratury złożone.

Błąd całkowania

Metoda	Liczba podziałów przedziału całkowania					
	1	2	4	8	16	32
Lewych prostokątów						
Punktu środkowego						
Prawych prostokątów						
Trapezów						
Simpsona						

Wyznaczenie funkcji pierwotnej jest bardzo często niemożliwe lub bardzo trudne, dlatego stosuje się numeryczne metody całkowania, które polegają na przybliżeniu funkcji podcałkowej **f(x)**, lub jej kolejnych fragmentów, na danym przedziale **(a, b)** za pomocą innej funkcji dla której wartość całki jest określona analitycznie.

Funkcją taką może być wielomian. Metody całkowania numerycznego rozmieszczają w przedziale całkowania numerycznego **(a, b)** punkty w których zostanie dokonana interpolacja wielomianem. Wspomniane punkty oznaczone jako **x_k** ($k = 0, 1 \dots N$) nazywane są węzłami kwadratury. Jeżeli odległości pomiędzy kolejnymi węzłami są takie same, a interpolacji dokonujemy wielomianem Lagrange'a oraz **x₀ = a** i **x_N = b**, to kwadraturę taką nazywamy kwadraturą Newtona-Cotesa.

Dla **N = 0** otrzymuje się wzór całkowania metodą prostokątów (jeden węzeł), dla **N = 1** - metodą trapezów (dwa węzły) a dla **N = 2** metodą Simpsona (trzy węzły).

Kwadratury Newtona-Cotesa mają następującą postać:

$$Q(f, a, b) = \sum_{k=0}^{N} A_k f_k(x_k)$$

gdzie współczynniki **A_k** wynikają z aproksymacji funkcji wielomianem interpolacyjnym Lagrange'a (przy równych odległościach między węzłami).

Łatwo zauważyc, że przybliżenie funkcji o dużej zmienności w przedziale całkowania za pomocą wielomianu interpolacyjnego (zwłaszcza niskiego rzędu) może okazać się mało dokładne.

Dlatego stosuje się:

- kwadratury złożone
- kwadratury adaptacyjne

Kwadratury złożone zostały omówione powyżej, stosuje się je zwykle łącznie z technikami adaptacyjnego obliczania całek. Kwadratury adaptacyjne polegają na wstępny podziale przedziału całkowania na dwa podprzedziały o równej długości i obliczeniu wartości całek na każdym z nich za pomocą kwadratury.

Przedział w którym nie osiągnięto wymaganej dokładności jest ponownie dzielony na dwa podprzedziały o równej długości i powtarzane jest całkowanie na każdym z nich oraz sprawdzana dokładność; w razie potrzeby dokonywany jest kolejny podział jednego lub obu tych przedziałów aż do osiągnięcia założonej dokładności.

Istotą kwadratury adaptacyjnej jest określenie, czy została osiągnięta wymagana dokładność. Sprawdzenia tego dokonuje się w prosty sposób: mając obliczoną wartość **Q₀** całki funkcji **f(x)** na danym przedziale **(a, b)**, oblicza się jej wartość po podziale przedziału na dwie części . Jeżeli moduł różnic tych

wartości jest mniejszy niż iloczyn modułu wstępniego przybliżenia i względnej tolerancji, to przybliżenie **Q₀** przyjmuje się za wystarczające, w przeciwnym wypadku postępuje się zgodnie z procedurą adaptacyjną, czyli dzieli się przedział na pół.

W bibliotece MATLAB-a zawarte są dwie funkcje **quad** i **quad8** umożliwiające całkowanie numeryczne w oparciu o dwie różne procedury:

quad - adaptacyjna kwadratura oparta o regułę Simpsona stosowana dla funkcji wolnozmiennych
(interpolacja wielomianem drugiego stopnia)

quad8 - adaptacyjna kwadratura ośmiorzedziałowa Newtona-Cotesa stosowana dla funkcji szybkozmiennych (aproksymacja wielomianem ósmego stopnia)

Q = quad('plik', a, b, tol, trace)
Q = quad8('plik', a, b, tol, trace)

a, b - przedział całkowania
tol - wymagana tolerancja względna (domyślana 10^{-3})
trace - parametr ten, jeżeli ma wartość niezerową, umożliwia wyświetlenie wykresu funkcji podcałkowej z zaznaczonymi węzłami kwadratury

Przebieg ćwiczenia MATLAB:

- 1.) narysować wykres funkcji podcałkowej w przedziale całkowania
- 2.) korzystając z poleceń **quad** i **quad8** wyznaczyć dla danej funkcji podcałkowej wartość całki oznaczonej i narysować jej wykres z zaznaczonymi węzłami kwadratury

Przykład

- Obliczyć wartość całki: $\int_0^5 \frac{dx}{2x + \sqrt{3x+1}}$ i narysować wykres funkcji podcałkowej w przedziale całkowania.

%%całkowanie

%funkcję y=f(x) zadeklarować w skrypcie o nazwie np. calk.m:

```
function y=f(x)
y=1./(2*x+sqrt(3*x+1))
```

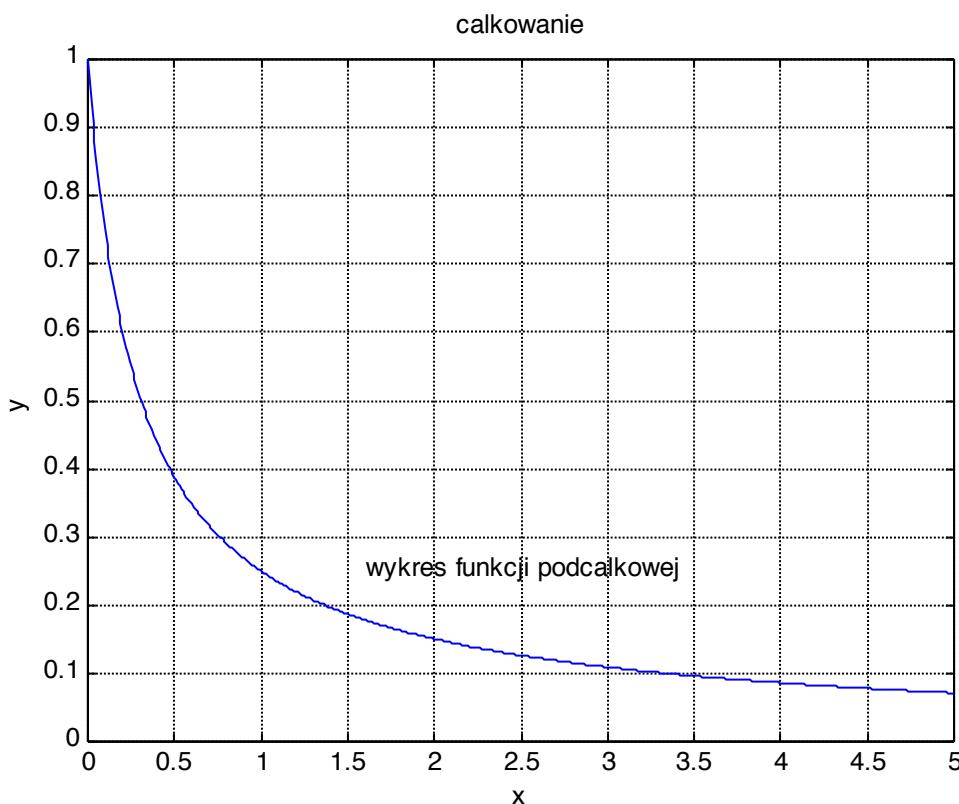
%w oknie MATLAB-a napisać i wywołać polecenie:

```
Q=quad('calk',0,5)
```

%w nowym skrypcie zamieścić polecenia służące do narysowania wykresu:

```
x=0:0.01:5
y=1./(2*x+sqrt(3*x+1))
```

```
plot(x,y)
grid on
title('calkowanie')
text(1.2,0.25,'wykres funkcji podcałkowej')
xlabel('x')
ylabel('y')
```

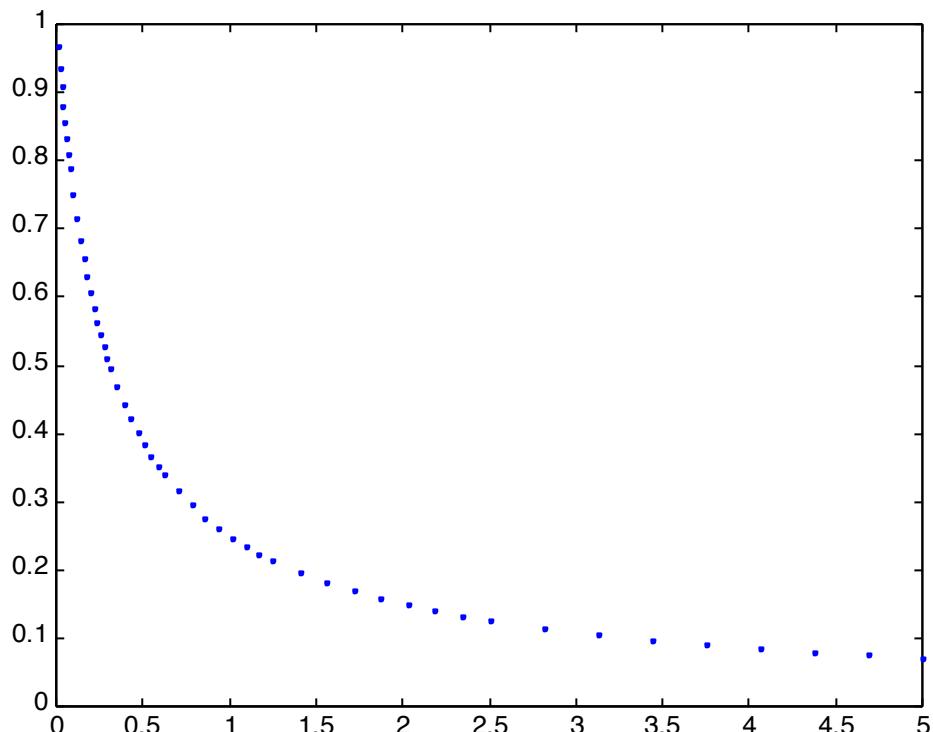


Wartość całki wynosi:

$Q = 0.9437$

Wykres funkcji podcałkowej z zaznaczonymi węzłami kwadratury można również uzyskać deklarując w poleceniu **quad** bądź **quad8** niezerową wartość parametru **w**:

Q=quad('calk', 0.5, 1e-3, 1)



Ogólnie w kwadraturze adaptacyjnej punkty w których obliczane są wartości funkcji podcałkowej wybiera się zależnie od zachowania się tej funkcji, natomiast w kwadraturze nieadaptacyjnej punkty te wybiera się w pewien ustalony sposób niezależnie od charakteru funkcji.

Przebieg ćwiczenia MATLAB:

- 1.) narysować wykres funkcji podcałkowej w przedziale całkowania i obliczyć wartość całki oznaczonej wykorzystując polecenia **quad** i **quad8** dla dokładności 10^{-3} , 10^{-5} i 10^{-7}
- 2.) porównać wyniki z obu programów

Ćwiczenia nr 8, 9 i 10

RÓŻNICZKOWANIE

Ćwiczenie nr 8

Różniczkowanie

W sposób przybliżony ma być rozwiązyane zagadnienie początkowe:

$$\begin{cases} Y' = F(t, Y) \\ Y(t_0) = Y_0 \end{cases}$$

w przedziale (t_0, T_{\max}) zmiennej niezależnej t , którego jednoznaczne rozwiązanie $Y = Y(t)$ jest różniczkowalne dostatecznie wiele razy.

Rozwiązanie $Y(t)$ może być funkcją skalarną lub wektorem

$$Y(t) = [y_1(t), y_2(t), \dots, y_m(t)]^T$$

w zależności od ilości równań w układzie.

Każde równanie różniczkowe wyższego rzędu postaci:

$$y^{(m)} = f(t, y, y', \dots, y^{(m-1)})$$

po wprowadzeniu oznaczeń:

$$\begin{aligned} y_1 &= y, \\ y_2 &= y', \\ y_3 &= y'', \\ &\dots \\ y_m &= y^{(m-1)} \end{aligned}$$

można sprowadzić do układu m równań rzędu pierwszego:

$$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ \dots \\ y_{m-1}' = y_m \\ y_m' = f(t, y_1, \dots, y_m) \end{cases}$$

Rozwiązywanie równań różniczkowych oparte są o jawne metody Rungego – Kutta można opisać wzorem:

$$Y_{n+1} = Y_n + h \sum_{i=1}^s \omega_i K_i$$

gdzie

$$\begin{cases} K_1 = F(t_n, Y_n), \\ K_i = F(t_n + b_i h; Y_n + h \sum_{j=1}^{i-1} a_{ij} K_j) \end{cases} \quad \text{dla } i = 2, 3, \dots, s.$$

Współczynniki metody można przedstawić w tablicy:

b_2	a_{21}				
b_3	a_{31}	a_{32}			
\dots	\dots	\dots			
b_s	a_{s1}	a_{s2}	\dots	$a_{s,s-1}$	
	ω_1	ω_2	\dots	ω_{s-1}	ω_s

Pomiędzy współczynnikami metody zachodzą następujące związki:

$$b_i = \sum_{j=1}^{i-1} a_{ij} \quad i = 2, 3, \dots, s$$

s - liczba etapów metody, **h** - krok

$$\sum_{i=1}^s \omega_i = 1$$

Przez \mathbf{Y}_n i \mathbf{Y}_{n+1} oznaczone zostały rozwiązania numeryczne, natomiast $\mathbf{Y}(t_n)$ i $\mathbf{Y}(t_{n+1})$ stanowią rozwiązania dokładne w punktach t_n i t_{n+1} , gdzie $t_n = t_{n+1} + h$, więc:

$$Y_n \approx Y(t_n)$$

Metody Rungego-Kutty należą do metod jednokrokowych, tzn. rozwiązanie przybliżone \mathbf{Y}_{n+1} w punkcie t_{n+1} jest wyznaczane tylko na podstawie rozwiązania \mathbf{Y}_n w punkcie t_n i nie zależy od wcześniej policzonych przybliżonych rozwiązań $\mathbf{Y}_{n-1}, \mathbf{Y}_{n-2}, \dots$. Ułatwia to sterowanie długością kroku w dowolnym momencie pracy algorytmu.

Przeglad metod

Zasady konstruowania rozwiązania numerycznego dla poszczególnych metod zostały poniżej przedstawione w sposób graficzny. Na wykresach grubą linią zaznaczone jest rozwiązanie dokładne. W oparciu o zagadnienie początkowe i warunek brzegowy wyznaczana jest wartość K_1 współczynnika kierunkowego stycznej do wykresu rozwiązania dokładnego w punkcie początkowym (t_n, Y_n) , a następnie rysowana jest styczna do tego wykresu w tym punkcie. Dla metody Eulera rozwiązanie numeryczne stanowi wartość Y_{n+1} w punkcie (t_{n+1}, Y_{n+1}) . Zmniejszając długość kroku (tzn. dzieląc kolejno przedział (t_n, t_{n+1}) na podprzedziały o równej długości) uzyskuje się ciąg stycznych zbieżny do rozwiązania dokładnego.

Wzór określający rozwiązywanie numeryczne otrzymuje się z przedstawionych powyżej zależności opisujących metody Rungego-Kutty, opierając się o tablicę współczynników charakteryzującą daną metodę. Łatwo zauważać, że dla metody Eulera wzór ten jest równaniem prostym.

W pozostałych metodach rozwiązanie numeryczne konstruowane jest w sposób analogiczny.

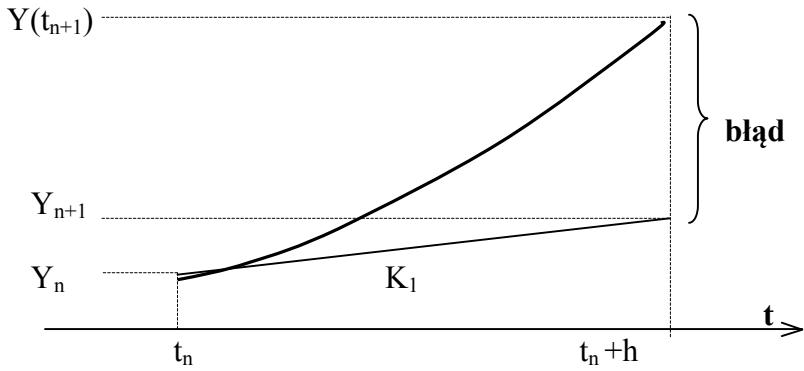
1) Metoda Eulera (1,1)

$$\left| \begin{array}{l} b_i = 0; \\ 1 \end{array} \right. \quad \omega_1 = 1 \quad \left. \begin{array}{l} a_{ij} = 0 \\ \end{array} \right\} \Rightarrow K_1 = F(t_n, , Y_n), \quad \underbrace{K_i = 0}_{\Downarrow} \\ Y_{n+1} = Y_n + h K_1$$

(tablica
współczynników)

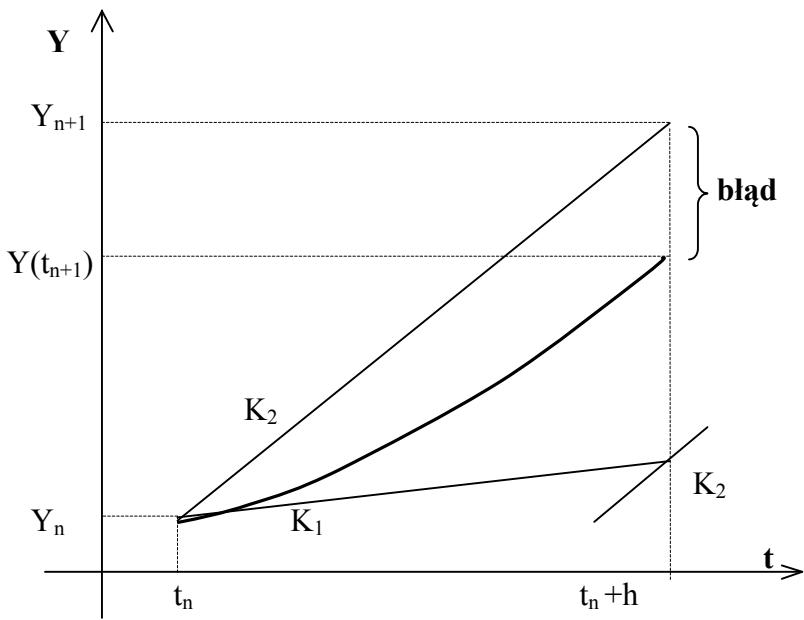
(współczynniki)

(rozwiązanie numeryczne)



2) Modyfikacja metody Eulera (1,2)

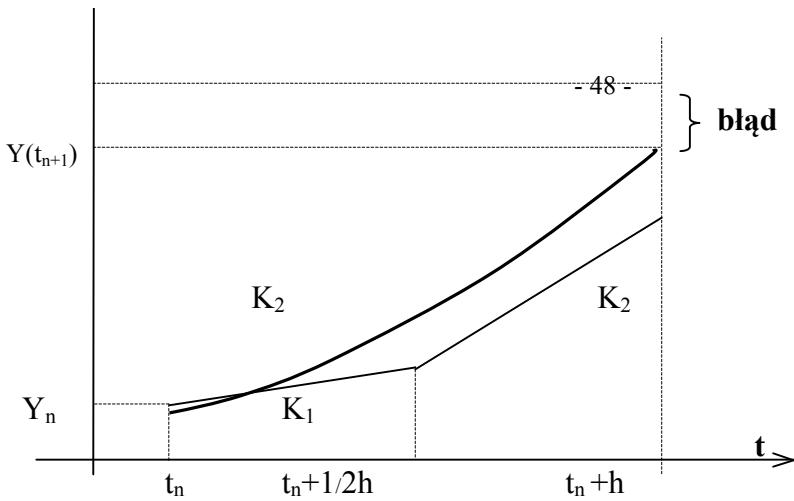
$$\begin{array}{c|cc} 1 & 1 \\ \hline 1 & 0 & 1 \end{array} \quad \left. \begin{array}{l} b_2 = 1; \\ \omega_1 = 0; \end{array} \right. \quad \left. \begin{array}{l} a_{21} = 1 \\ \omega_2 = 1 \end{array} \right\} \Rightarrow \underbrace{K_1 = F(t_n, Y_n),}_{\Downarrow} \quad \underbrace{K_2 = F(t_n + h, Y_n + hK_1)}_{Y_{n+1} = Y_n + hK_2}$$



3) Metoda Runge'go (2,2)

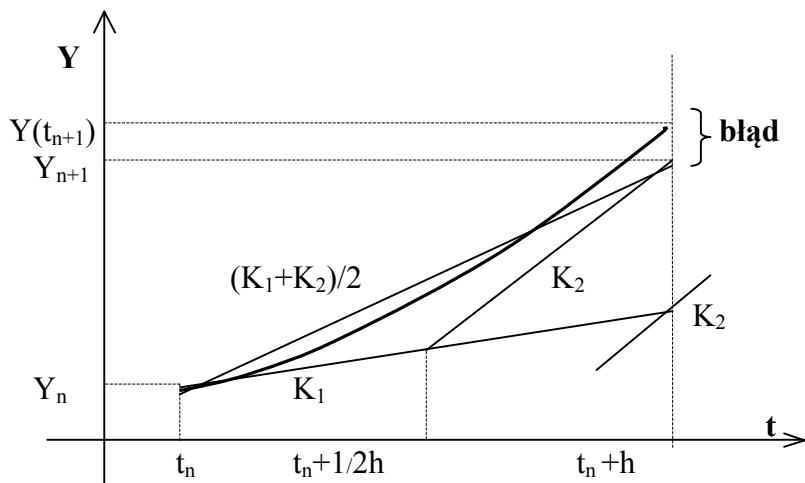
$$\begin{array}{c|cc} 1/2 & 1/2 \\ \hline 0 & 1 \end{array} \quad \left. \begin{array}{l} b_2 = 1/2; \\ \omega_1 = 0; \end{array} \right. \quad \left. \begin{array}{l} a_{21} = 1/2; \\ \omega_2 = 1 \end{array} \right\} \Rightarrow \underbrace{K_1 = F(t_n, Y_n),}_{\Downarrow} \quad \underbrace{K_2 = F(t_n + 1/2h, Y_n + 1/2hK_1)}_{Y_{n+1} = Y_n + hK_2}$$





4) Metoda Eulera – Cauche'go (2,2)

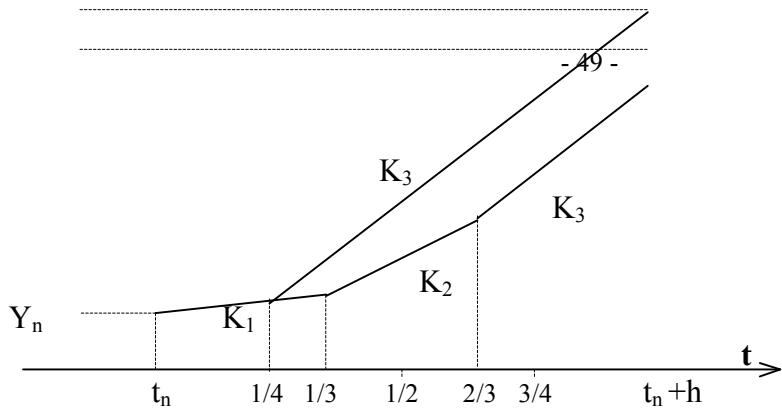
$$\begin{array}{c|cc}
 1 & 1 \\
 \hline
 & 1/2 & 1/2
 \end{array} \quad \left. \begin{array}{l} b_2 = 1; \quad a_{21} = 1; \\ \omega_1 = 1/2; \quad \omega_2 = 1/2 \end{array} \right\} \Rightarrow \underbrace{\begin{array}{l} K_1 = F(t_n, , Y_n), \\ K_2 = F(t_n+h, Y_n+hK_1) \end{array}}_{\Downarrow} \quad Y_{n+1} = Y_n + h(1/2K_1 + 1/2K_2)$$



5) Metoda Heuna (3,3)

$$\begin{array}{c|ccc}
 1/3 & 1/3 & & \\
 \hline
 2/3 & 0 & 2/3 & \\
 \hline
 & 1/4 & 0 & 3/4
 \end{array} \quad \left. \begin{array}{lll} b_2 = 1/3; & a_{21} = 1/3; \\ b_3 = 2/3; & a_{31} = 0; & a_{32} = 2/3 \\ \omega_1 = 1/4; & \omega_2 = 0; & \omega_3 = 3/4; \end{array} \right\} \Rightarrow \underbrace{\begin{array}{l} K_1 = F(t_n, , Y_n), \\ K_2 = F(t_n+1/3h, Y_n+1/3hK_1) \\ K_3 = F(t_n+2/3h, Y_n+2/3hK_2) \end{array}}_{\Downarrow} \quad Y_{n+1} = Y_n + h(1/4K_1 + 3/4K_3)$$





6) Metoda Rungego-Kutty (4,4)

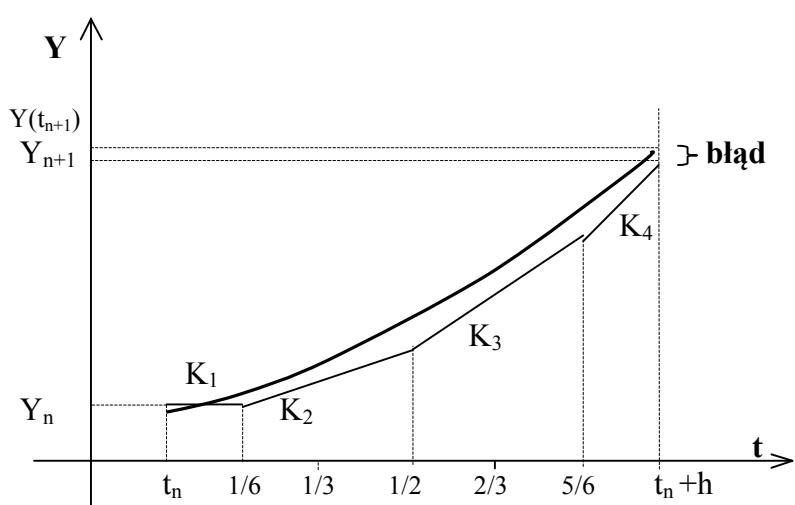
$\frac{1}{2}$	$\frac{1}{2}$			$b_2 = \frac{1}{2};$	$a_{21} = \frac{1}{2};$		
$\frac{1}{2}$	0	$\frac{1}{2}$		$b_3 = \frac{1}{2};$	$a_{31} = 0;$	$a_{32} = \frac{1}{2};$	
1	0	0	1	$b_4 = 1;$	$b_{41} = 0;$	$b_{42} = 0;$	$b_{43} = 1;$
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\omega_1 = \frac{1}{6};$	$\omega_2 = \frac{1}{3};$	$\omega_3 = \frac{1}{3};$	$\omega_4 = \frac{1}{6};$

\Downarrow

$$\begin{aligned} K_1 &= F(t_n, Y_n), \\ K_2 &= F(t_n + 1/2h, Y_n + 1/2hK_1) \\ K_3 &= F(t_n + 1/3h, Y_n + 1/3hK_2) \\ K_4 &= F(t_n + h, Y_n + hK_3) \end{aligned}$$

\Downarrow

$$Y_{n+1} = Y_n + h(1/6K_1 + 1/3K_2 + 1/3K_3 + 1/6K_4)$$



Błąd w punkcie końcowym

Dokładność metod Rungego-Kutty jest rozumiana w sensie błędu aproksymacji, tzn. liczona jest różnica pomiędzy rozwiązaniem dokładnym a numerycznym. W tym celu we wzorach określających współczynniki K_i i K_i oraz rozwiązanie numeryczne Y_{n+1} zamiast wartości przybliżonej Y_n przyjmowana jest wartość

dokładna $\mathbf{Y}(t_n)$ i liczone są wartości tych współczynników oraz rozwiązania numerycznego dla pewnego kroku \mathbf{h} , a następnie wyznaczany jest błąd lokalny stanowiący różnicę pomiędzy rozwiązaniem dokładnym i numerycznym.

Błąd aproksymacji (błąd lokalny) metody Rungego-Kutty ma postać:

$$\mathbf{r}_{n+1}(\mathbf{h}) = \mathbf{Y}(t_n + \mathbf{h}) - (\mathbf{Y}(t_n) + h \sum_{i=1}^s \omega_i \mathbf{K}_i(\mathbf{h}))$$

Dla dostatecznie małych wartości \mathbf{h} , błąd aproksymacji można rozwinąć w szereg potęgowy względem \mathbf{h} :

$$\mathbf{r}_{n+1}(\mathbf{h}) = \mathbf{r}_{n+1}(0) + h \mathbf{r}'_{n+1}(0) + \frac{h^2}{2} \mathbf{r}''_{n+1}(0) + \dots$$

Metoda Rungego-Kutty jest rzędem p , jeżeli dla każdego zagadnienia początkowego zachodzi:

$$\mathbf{r}_{n+1}(0) = 0, \quad \mathbf{r}'_{n+1}(0) = 0, \dots, \quad \mathbf{r}_{n+1}^{(p)}(0) = 0 \quad \mathbf{r}_{n+1}^{(p+1)}(0) \neq 0$$

czyli błąd lokalny ma postać:

$$\begin{aligned} \mathbf{r}_{n+1}(\mathbf{h}) &= \Phi(t_n, \mathbf{Y}(t_n)) \mathbf{h}^{p+1} + \mathbf{O}(\mathbf{h}^{p+2}) \\ \Phi(t_n, \mathbf{Y}(t_n)) \mathbf{h}^{p+1} &\quad \text{część główna błędu lokalnego} \end{aligned}$$

Najprostsza metoda automatycznego dobierania długości kroku \mathbf{h} polega na wyznaczeniu przybliżonej wartości części głównej błędu lokalnego i dobraniu takiego kroku \mathbf{h} , aby spełniona była zależność:

$$\|\Phi(t_n, \mathbf{Y}(t_n)) \mathbf{h}^{p+1}\| < \varepsilon$$

dla zadanej wartości ε , gdzie $\|\cdot\|$ oznacza normę wektora.

Błądem globalnym albo całkowitym metody w punkcie t_n nazywamy wielkość:

$$\boldsymbol{\varepsilon}_n = \mathbf{Y}_n - \mathbf{Y}(t_n)$$

W programie MATLAB dostępnych jest kilka funkcji pozwalających na rozwiązanie zagadnienia początkowego dla układów równań różniczkowych zwyczajnych postaci:

$$\frac{dy}{dt} = F(t, y), \quad y(t_0) = y_0; \quad y, y_0 \in R^n$$

Przykładowo rozpatrzone zostaną dwie z nich (**ode23** i **ode45**).

Każda z tych funkcji korzysta z pary metod Rungego-Kutty rzędu 2 i 3 (**ode23**) lub rzędu 3 i 4 (**ode34**).

$$\begin{aligned} [t, Y] &= \text{ode23('plik', t0, tk, y0, tol, tr)} \\ [t, Y] &= \text{ode45('plik', t0, tk, y0, tol, tr)} \end{aligned}$$

plik - nazwa pliku (bez rozszerzenia) w którym zdefiniowana jest funkcja $F(t, y)$

t0, tk - przedział czasu w którym poszukiwane jest rozwiązanie

y0 - warunek początkowy (wektor kolumnowy zawierający wartość rozwiązania w chwili początkowej)

tol - parametr określający dokładność, domyślna wartość: 10^{-3} dla ode23 i 10^{-6} dla ode45

tr - parametr ten, jeżeli ma wartość niezerową umożliwia wypisanie na ekranie kolejnych kroków metody

Aby wyznaczyć wartość rozwiązań należy, po zadeklarowaniu funkcji $F(t, y)$, napisać w oknie programu polecenie, o postaci jak powyżej, zawierające nazwę odpowiedniej funkcji **ode**.

Po wprowadzeniu oznaczenia $dy = F(t, y)$, funkcję $F(t, y)$ można zadeklarować w skrypcie z rozszerzeniem '**m**' w sposób następujący:

```
function dy = F(t, y)
dy =
```

W przypadku równania różniczkowego zwyczajnego wyższego rzędu należy, wprowadzając dodatkowe zmienne, sprowadzić to równanie do układu równań rzędu pierwszego i definiując funkcję $F(t, Y)$ zamieścić wszystkie równania w macierzy.

Przebieg ćwiczenia MATLAB:

- 1.) zadeklarować funkcję $F(t, y)$, a następnie korzystając z polecenia **ode23** wyznaczyć wektor rozwiązania numerycznego w przedziale **(t0, tk)**
- 2.) w nowym pliku umieścić zależność określającą rozwiązanie dokładne i wyznaczyć jego wartość w przedziale **(t0, tk)**
- 3.) obliczyć błąd rozwiązania, tzn. różnicę pomiędzy rozwiązaniem dokładnym a numerycznym, przepisać wartość błędu w punkcie końcowym oraz policzyć liczbę kroków wykorzystując polecenie **size**
- 4.) narysować w jednym układzie współrzędnych wykres rozwiązania dokładnego i numerycznego a w drugim wykres błędu

Przykłady

1. W przedziale $< 0; 3 >$, stosując metodę **ode23**, znaleźć rozwiązanie następującego równania różniczkowego: $\frac{dy}{dt} = \frac{1}{1+t^2}$, spełniającego warunek początkowy $y(0) = 0$. Wyznaczyć błąd w punkcie końcowym i narysować wykres rozwiązania numerycznego i dokładnego w jednym układzie współrzędnych a wykres błędu w drugim. Rozwiązanie dokładne określone jest zależnością: $y = \arctg(t)$.

```
%różniczkowanie
```

```
%funkcję y'=f(t,y) zadeklarować w skrypcie o nazwie np. rozn.m:
```

```
function dy=F(t,y) %dy=y'  
dy=1./(1+t.^2)
```

```
%w oknie MATLAB-a napisać i wywołać polecenie:
```

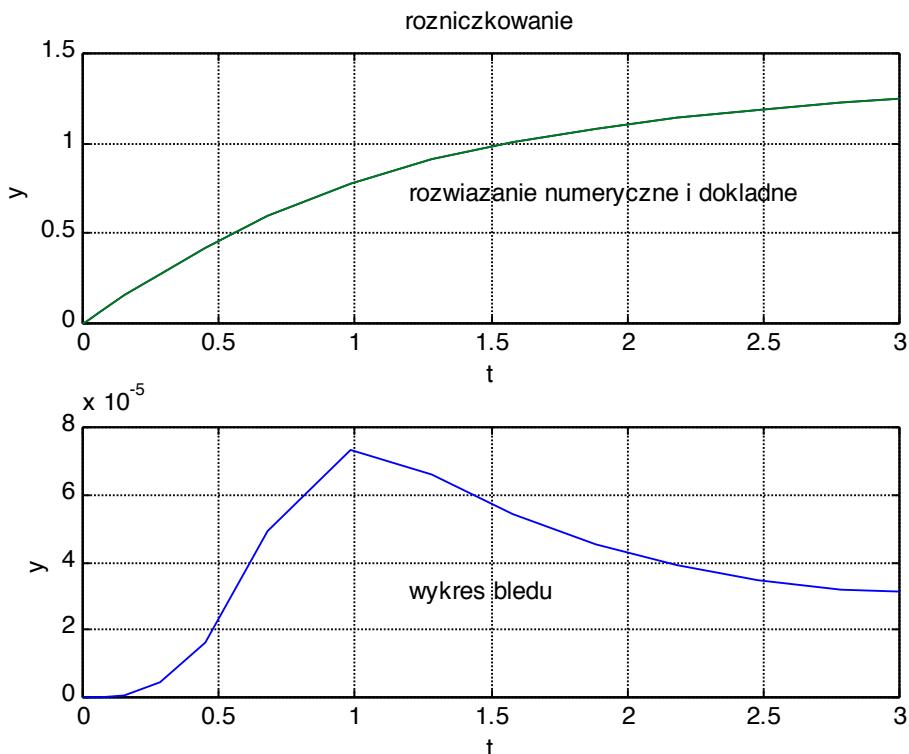
```
[t,Y]=ode23('rozn',[0 3],[0]) %Y - rozwiązanie numeryczne
```

```
%w nowym skrypcie zamieścić polecenia służące do narysowania wykresu  
%i wyznaczenia wartości błędu różniczkowania
```

```
y=atan(t) %y - rozwiązanie dokładne  
bl=y-Y %bl - błąd różniczkowania
```

```
subplot(2,1,1)  
plot(t,Y,t,y)  
grid on  
title('różniczkowanie')  
text(1.2,0.7,'rozwiązańe numeryczne i dokładne')  
xlabel('t')  
ylabel('y')
```

```
subplot(2,1,2)  
plot(t,bl)  
grid on  
text(1.2,3e-5,'wykres błędu')  
xlabel('t')  
ylabel('y')
```



Rozwiązań uzyskane w MATLAB-ie:

czas $t =$	rozwiązań numeryczne $Y =$	rozwiązań dokładne $y =$	błąd $bl =$ $1.0e-004 *$
0	0	0	0
0.0001	0.0001	0.0001	0.0000
0.0005	0.0005	0.0005	0.0000
0.0025	0.0025	0.0025	0.0000
0.0125	0.0125	0.0125	0.0000
0.0625	0.0624	0.0624	0.0002
0.1543	0.1531	0.1531	0.0061
0.2810	0.2740	0.2740	0.0424
0.4472	0.4205	0.4205	0.1640
0.6819	0.5984	0.5985	0.4944
0.9819	0.7762	0.7762	0.7331
1.2819	0.9082	0.9083	0.6621
1.5819	1.0070	1.0071	0.5453
1.8819	1.0823	1.0824	0.4527
2.1819	1.1410	1.1410	0.3896
2.4819	1.1877	1.1878	0.3482
2.7819	1.2257	1.2257	0.3210
3.0000	1.2490	1.2490	0.3157

Wartość błędu w punkcie końcowym wynosi $0,3157 \cdot 10^{-4}$.

2. Dla $t \in [0, 10]$ znaleźć rozwiązanie następującego równania różniczkowego II-rzędu
- $$\frac{d^2y}{dt^2} + 2\frac{dy}{dt} + y = 0 \text{ o zadanym warunku brzegowym } y(0) = [1 \ 0].$$

Aby rozwiązać powyższe równanie należy sprowadzić je do układu dwóch równań I-rzędu wprowadzając dodatkowe zmienne y_1 i y_2 :

$$\begin{aligned} y_1 &= y \\ y_2 &= \frac{dy}{dt} \end{aligned} \Rightarrow \begin{cases} \frac{dy_1}{dt} = y_2 \\ \frac{dy_2}{dt} = -2y_2 - y_1 \end{cases}$$

%%równanie różniczkowe II-go rzędu

%funkcję d2y=F(t,y) zadeklarować w skrypcie o nazwie np. rozn2.m:

```
function d2y=F(t,y)
d2y=[y(2);-y(1)-2*y(2)]
```

```
%d2y=y'
%y(1)=y y(2)=y'
```

%w oknie MATLAB-a napisać i wywołać kolejno polecenia:

```
[t1,Y1]=ode23('rozn2',[0 10],[1 0]')
[t2,Y2]=ode45('rozn2',[0 10],[1 0]')
```

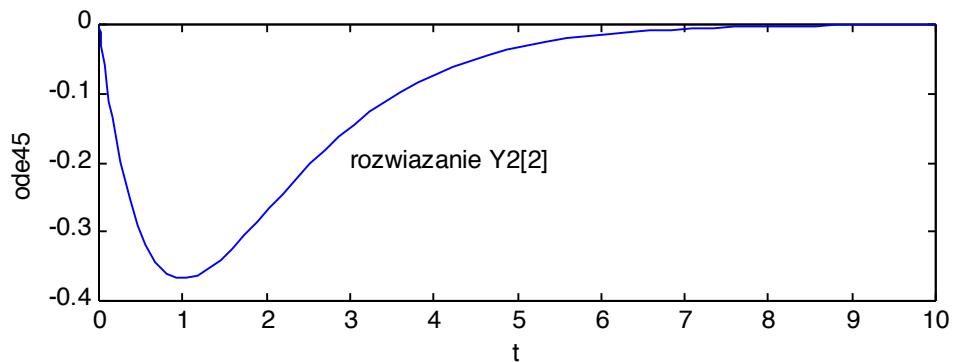
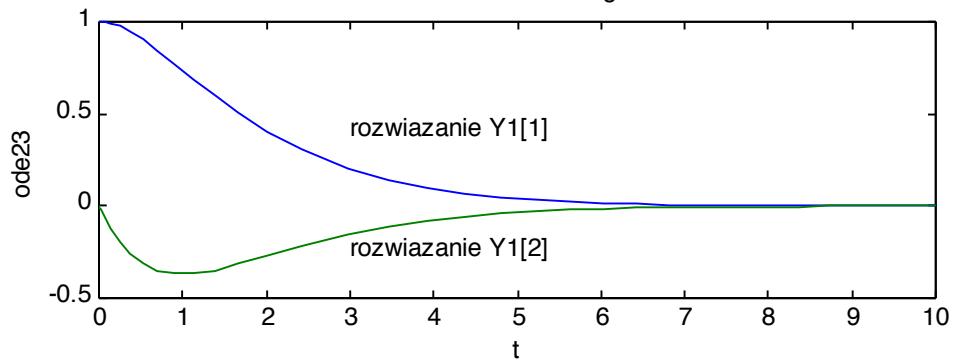
%Y1 - rozwiązanie numeryczne metodą ode23
%Y2 - rozwiązanie numeryczne metodą ode45

%w nowym skrypcie zamieścić polecenia służące do narysowania wykresów

```
subplot(2,1,1)
plot(t1,Y1)
xlabel('t')
title('rownanie różniczkowe II-go rzedu')
ylabel('ode23')
text(3,0.4,'rozwiæzanie Y1[1]')
text(3,-0.25,'rozwiæzanie Y1[2]')

subplot(2,1,2)
plot(t2,Y2(:,2))
xlabel('t')
ylabel('ode45')
text(3,-0.2,'rozwiæzanie Y2[2]')
```

rownanie różniczkowe II-go rzędu



Ćwiczenie nr 9

Stabilność metod Rungego-Kutty

Przyjmujemy, że metoda numeryczna jest absolutnie stabilna dla danej długości kroku całkowania \mathbf{h} , jeżeli zastosowanie tej metody do liniowego układu stabilnego daje ciąg rozwiązań przybliżonych \mathbf{y}_n zbieżny do zera, gdy $n \rightarrow \infty$ dla $\mathbf{h} = \text{const.}$

Gdy te warunki nie są spełnione to po wykonaniu nawet niewielkiej ilości kroków rozwiązania przybliżone na ogólnie szybko rosną dając tzw. lawinę błędów. W celu uniknięcia gwałtownego narastania błędu należy zmniejszyć krok całkowania. Jeżeli odpowiednio zmniejszymy długość kroku możemy otrzymać układ na granicy stabilności.

Stabilność absolutna metody różniczkowej zależy od wyboru zagadnienia początkowego i od długości kroku całkowania. Aby łatwiej można było określić zakres dopuszczalnych zmian długości kroku \mathbf{h} kreślony jest na płaszczyźnie zmiennej zespolonej tzw. obszar stabilności absolutnej metody.

Obszarem stabilności absolutnej nazywa się podzbiór Ω płaszczyzny zespolonej C taki, że ciąg $\{y(n)\}$ wartości rozwiązania numerycznego (otrzymanego badaną metodą ze stałym krokiem \mathbf{h} takim, że $\lambda\mathbf{h}$ należy do wnętrza tego podzbioru) dąży do 0 przy n rosnącym:

$$\lim_{n \rightarrow \infty} \{y(n)\} \rightarrow 0$$

Przedziałem stabilności absolutnej nazywa się wspólną część obszaru Ω i osi Re . Dla metod rzędu co najmniej pierwszego spełniona jest zależność:

$$\mathbf{u} + \mathbf{v} = \mathbf{1}$$

W celu zbadania stabilności absolutnej metod liniowych rozwiązywania równań różniczkowych rozpatrywane jest następujące równanie testowe:

$$y' = \lambda y \quad \text{o warunku początkowym:} \quad y(0) = 1$$

Rozwiązanie numeryczne powyższego równania za pomocą dwuetapowej metody Runge'go-Kutty wyraża się wzorem:

$$y(n+1) = [1 + \lambda\mathbf{h} + \mathbf{a}\mathbf{v}(\lambda\mathbf{h})^2] y(n)$$

Zakładamy, że: $\lambda\mathbf{h}$ jest liczbą zespoloną oraz wprowadzamy parametr \mathbf{P} :

$$\lambda\mathbf{h} = \mathbf{r} + i\mathbf{s} \quad i \quad \mathbf{P} = \mathbf{a}\mathbf{v}$$

wówczas równanie brzegu obszaru stabilności absolutnej będzie następujące:

$$|y(n+1)| = |y(n)|$$

więc po uwzględnieniu wcześniejszych zależności otrzymujemy:

$$|\mathbf{P}(\mathbf{r} + i\mathbf{s})^2 + (\mathbf{r} + i\mathbf{s}) + 1| = 1$$

Równanie brzegu obszaru stabilności absolutnej wyprowadza się w zależności od parametru **P**. Dla **P ≤ 0,5** najpierw znajduje się przedziały stabilności absolutnej, tzn. wyznacza się wartość **r** dla **s = 0**, a następnie dla każdej wartości **r** z tych przedziałów określa się **s = s(r)**. W przypadku **P > 0,5** krzywa będąca wykresem powyższego równania staje się wklęsła, więc nie można jej w sposób jednoznaczny opisać, dlatego w tym przypadku korzysta się z zależności **r = r(s)**.

Wartość parametru **P** jest ściśle powiązana z rzędem metody:

- **P = 0** - metoda Eulera (1,1)
- **P = 0,5** - metoda rzędu 2
- **P = 1** - modyfikacja metody Eulera (1,2)
- pozostałe wartości **P** – modyfikacje metod Rungego-Kutty 1-go rzędu 2-etapowych

Równanie różniczkowe okręgu

Równanie okręgu na płaszczyźnie ma postać następującą:

$$\begin{cases} x = \cos \omega t \\ y = \sin \omega t \end{cases} \quad \begin{array}{l} \omega - \text{parametr} \\ t \in [0; T_{\max}] \end{array}$$

Po dwukrotnym zróżniczkowaniu drugiego równania otrzymuje się równanie różniczkowe II rzędu, które jest równaniem różniczkowym okręgu:

$$y' = \omega \cos \omega t \quad \Rightarrow \quad y'' = -\omega^2 \sin \omega t \quad \Rightarrow \quad y'' + \omega^2 y = 0$$

Po wprowadzeniu zmiennej $z = \omega x = \omega \cos \omega t$ można powyższe równanie zapisać w postaci układu dwóch równań rzędu pierwszego lub w postaci macierzowej:

$$\begin{cases} y' = z \\ z' = -\omega^2 y \end{cases} \quad \Rightarrow \quad \begin{bmatrix} y' \\ z' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \quad \Rightarrow \quad Y' = F(t, Y)$$

gdzie:

$$Y = \begin{bmatrix} y \\ z \end{bmatrix} \quad \text{i} \quad F(t, Y) = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} Y$$

a warunek brzegowy jest następujący: $y(0) = 0 \quad \text{i} \quad z(0) = \omega$

Pierwiastki λ równania $Y' = F(t, Y)$, czyli wartości własne macierzy układu równań różniczkowych, są liczbami czysto urojonymi:

$$\Delta = \omega^2 = (i\omega)(-i\omega) \Rightarrow \lambda = \pm i\omega$$

Jeżeli wartości własne macierzy układu równań różniczkowych leżą w obszarze stabilności absolutnej danej metody, to metoda ta jest stabilna dla tego układu równań.

Powyższe równanie różniczkowe okręgu będzie poniżej rozwiązywane ze stałym krokiem pięcioma metodami Rungego-Kutty rzędu pierwszego i drugiego.

1.) Metoda jawna Eulera (1,1)

W metodzie jawnej Eulera rozwiązywanie numeryczne równania różniczkowego wyraża się wzorem:

$$Y_{n+1} = Y_n + hF(t_n, Y_n)$$

dla równania różniczkowego okręgu zachodzi:

$$F(t_n, Y_n) = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} Y_n$$

tak więc:

$$Y_{n+1} = \begin{bmatrix} 1 & h \\ -\omega^2 h & 1 \end{bmatrix} Y_n = AY_n$$

macierz \mathbf{A} nazywana jest macierzą przejścia.

Po obliczeniu wyznacznika Δ macierzy \mathbf{A} można znaleźć jej wartości własne λ :

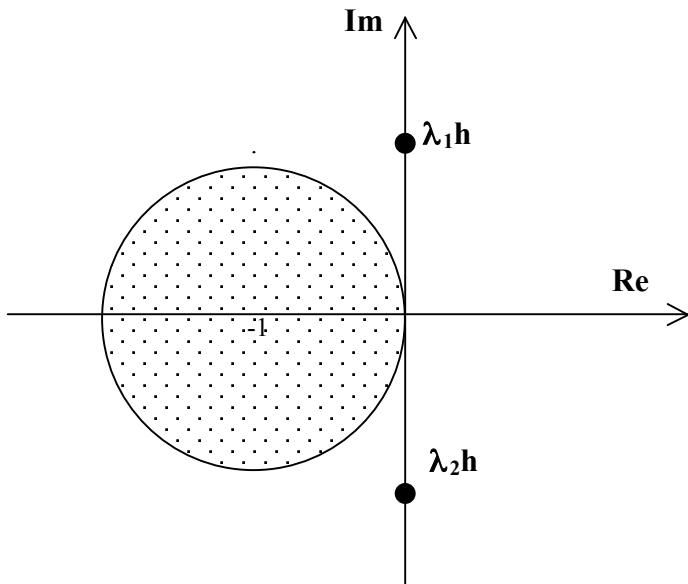
$$\Delta = 1 + \omega^2 h^2 = (1 + i\omega h)(1 - i\omega h) \Rightarrow \lambda = 1 \pm i\omega h$$

Jeżeli moduły wartości własnych macierzy przejścia są większe od 1, to dana metoda jest metodą niestabilną dla rozpatrywanego równania różniczkowego.

$$|\lambda| = \sqrt{1 + \omega^2 h^2} > 1$$

Metoda jawna Eulera jest metodą niestabilną dla równania różniczkowego okręgu.

Obszarem stabilności absolutnej metody Eulera na płaszczyźnie zmiennej zespolonej λh jest koło o środku w punkcie (-1, 0) i promieniu 1. Warunkiem stabilności jest, aby wszystkie wartości własne macierzy układu λh leżały w tym obszarze. W przypadku równania różniczkowego okręgu wartości te są czysto urojone, leżą więc na osi **Im** niezależnie od kroku h .



2.) Metoda Eulera-Cauchego (2,2)

W metodzie Eulera-Cauchego macierz przejścia dla równania różniczkowego okręgu ma postać następującą:

$$\mathbf{A} = \begin{bmatrix} 1 - (\omega h)^2 / 2 & h \\ -\omega^2 h & 1 - (\omega h)^2 / 2 \end{bmatrix}$$

wartości własne wyrażają się wzorem:

$$\lambda = (1 - 1/2(\omega h)^2) \pm i\omega h$$

natomast ich moduły:

$$|\lambda| = \sqrt{1 + \frac{1}{4}\omega^4 h^4} > 1$$

są większe od 1, więc metoda Eulera-Cauchego jest również metodą niestabilną dla równania różniczkowego okręgu. Jak można zauważyć na podstawie przykładów z pakietu RK-STAB obszarem stabilności absolutnej dla tej metody jest obszar nieco większy niż w przypadku metody jawnej Eulera, ale wartości własne leżą również poza tym obszarem.

3.) Metoda stabilna dla równania różniczkowego okręgu (1,2)

Metoda ta jest modyfikacją metody Eulera-Cauchego, a macierz przejścia dla równania różniczkowego okręgu określona jest zależnością:

$$A = \begin{bmatrix} 1 - P(\omega h)^2 & h \\ -\omega^2 h & 1 - P(\omega h)^2 \end{bmatrix}$$

gdzie: $P = av$ (a i v - współczynniki metody)

Wartości własne wyrażają się wzorem:

$$\lambda = (1 - P(\omega h)^2) \pm i\omega h$$

Aby omawiana metoda była rzędu co najmniej pierwszego współczynniki u i v muszą spełniać następującą zależność:

$$u + v = 1$$

Korzystając z warunkiem stabilności metody:

$$|\lambda| \leq 1$$

otrzymujemy:

$$(\omega h)^2 \leq (2P - 1) / P^2$$

Warunek ten może być spełniony jedynie dla $P \geq 0,5$ a wówczas:

$$|\omega h| \in (0; \sqrt{(2P - 1)} / P)$$

czyli tak należy dobrać krok h , aby wartość ωh należała do powyższego przedziału.

4.) Metoda niejawna Eulera (1,1)

Metoda niejawna Eulera opisana jest zależnością:

$$Y_{n+1} = Y_n + hF(t_{n+1}, Y_{n+1})$$

W celu wyznaczenia obszaru stabilności absolutnej dla tej metody rozwiązuje się skalarne równanie testowe o postaci:

$$y' = \lambda y \quad \text{gdzie } \lambda - \text{liczba zespolona}$$

Rozwiązywanie numeryczne powyższego równania metodą niejawną Eulera jest następujące:

$$y_{n+1} = \frac{1}{1-\lambda h} y_n$$

Warunkiem stabilności metody jest aby:

$$|y_{n+1}| < |y_n|$$

Warunek ten będzie spełniony jeżeli $|1-\lambda h| > 1$, czyli jeżeli wartość λh będzie leżała na płaszczyźnie zmiennej zespolonej poza kołem jednostkowym o środku w punkcie $(\text{Re}, \text{Im}) = (1, 0)$. Tak więc obszarem stabilności absolutnej metody niejawnej Eulera jest zewnętrze powyższego koła jednostkowego.

Wartości własne macierzy układu równań różniczkowych opisujących równanie różniczkowe okręgu są czysto urojone, leżą więc w obszarze stabilności absolutnej metody niejawnej Eulera. Metoda ta jest metodą stabilną dla równania różniczkowego okręgu.

5.) Wzór trapezów (2,2)

Metoda trapezów opisana jest zależnością:

$$Y_{n+1} = Y_n + 0,5h(F(t_n, Y_n) + F(t_{n+1}, Y_{n+1}))$$

natomiast rozwiązanie numeryczne skalarnego równania testowego ma postać następującą:

$$y_{n+1} = \frac{1 + \lambda h}{1 - \lambda h} y_n$$

Wartości własne macierzy układu są czysto urojone $\lambda = \pm i\omega$, więc:

$$|y_{n+1}| = |y_n| \quad \text{gdyż:} \quad \left| \frac{1 + \lambda h}{1 - \lambda h} \right| = 1$$

Obszarem stabilności absolutnej w przypadku wzoru trapezów jest lewa półpłaszczyzna zmiennej zespolonej $\text{Re} < 0$ wraz z osią urojonych. Wartości własne macierzy układu leżą na osi urojonych, czyli na brzegu obszaru stabilności absolutnej metody trapezów. Metoda ta dla równania różniczkowego okręgu jest zawsze na granicy stabilności niezależnie od długości kroku h .

Ćwiczenie nr 10

Zastosowanie metod Rungego-Kutty

Dokładność rozwiązania równania różniczkowego uzyskanego metodami numerycznymi zależy m.in. od zastosowanej metody i od algorytmu sterowania długością kroku. Jeżeli znane jest rozwiązania dokładne możliwe jest wyznaczenie norm błędu względnego, bezwzględnego, globalnego oraz błędu w punkcie końcowym rozwiązania numerycznego.

Sprawność metody określona jest jako procentowy udział kroków akceptowanych do ogólnej liczby kroków w obliczeniach ze zmiennym krokiem, natomiast koszty metody stanowi liczba odwołań do podprogramu obliczającego wartość prawej strony równania.

Sterowanie długością kroku przeprowadzane jest na podstawie oszacowania wartości błędu lokalnego (względnego lub bezwzględnego), które dokonywane jest również dla kroku stałego.

Wartość błędu lokalnego można szacować wykorzystując:

- własny wzór szacujący metody
- metodę włożoną
- ekstrapolację Richardsona
- metodę towarzyszącą

Sterowania długością kroku dokonywane jest poprzez:

- połowienie / podwajanie kroku
- mnożenie kroku przez współczynnik

Przedstawiając przybliżenie normy części głównej błędu lokalnego w postaci:

$$z = C h^k \quad C > 0 \text{ - zależy od uzyskanego rozwiązania} \\ k \text{ - zależy od rzędu metody}$$

dla połowienia kroku otrzymuje się następujące kryterium akceptacji kroku:

$$C h^k \leq \epsilon \quad \epsilon \text{ - zadana wartość}$$

Jeżeli powyższa nierówność jest spełniona to obliczenia powtarzane będą z nowym krokiem $H = h / 2$ pod warunkiem, że krok ten będzie jeszcze krokiem dopuszczalnym tzn.: $h_{\min} \leq H$.

Można również podwoić wartość kroku pod warunkiem spełnienia zależności:

$$C (2 h)^k < \epsilon \quad \text{SafetyFac} \quad 0 < \text{SafetyFac} < 1 \quad - \text{współczynnik bezpieczeństwa}$$

wówczas nowy krok $H = 2 h$ musi być sprowadzany jeszcze do dopuszczalnego przedziału kroku:

$$H := \min(H, h_{\max})$$

Dla mnożenia kroku przez współczynnik $H = \gamma h$, współczynnik γ wyznaczany jest również z nierówności zawierającej współczynnik bezpieczeństwa:

$$C (\gamma h)^k < \epsilon \quad \text{SafetyFac}$$

$$\gamma^k z < \epsilon \quad \text{SafetyFac}$$

$$\gamma < (\epsilon \text{ SafetyFac} / z)^{(1/k)}$$

Krok ten nie może ulegać zbyt dużym i nagłym zmianom, dlatego współczynnik γ musi spełniać warunek:

$$\mathbf{FacMin} \leq \gamma \leq \mathbf{FacMax}$$

czyli:

$$\mathbf{hMin} \leq H \leq \mathbf{hMax}$$

Wartości **hMax**, **hMin**, **SafetyFac**, **FacMin**, **FacMax**, są ustalonymi przez użytkownika parametrami służącymi do korygowania kroku obliczeń. Jeżeli krok jest zbyt duży jego wartość jest redukowana do **hMax**, natomiast jeżeli jest mniejszy niż **hMin** następuje przerwanie obliczeń.

Spis literatury

1. Fortuna Z. - Metody numeryczne, WNT, Warszawa 1983
2. Jankowska J. i M. – Przegląd metod i algorytmów numerycznych, cz. 1, WNT, Warszawa 1981
3. Ralston A. – Wstęp do analizy numerycznej, PWN, Warszawa 1971
4. Stoer J. – Wstęp do metod numerycznych, t. 1, PWN, Warszawa 1979
5. Stoer J. , Bulirsch R.– Wstęp do metod numerycznych, t. 2, PWN, Warszawa 1980
6. Krupowicz A. – Metody numeryczne zagadnień początkowych równań różniczkowych zwyczajnych, PWN, Warszawa 1986
7. Demidowicz B. P., Maron I. A. - Metody numeryczne, PWN, Warszawa 1965
8. Zalewski A., Cegieła R. – MATLAB – obliczenia numeryczne i ich zastosowanie, Wydawnictwo Nakom, Poznań 1999