

Ewa Dyka
Instytut Elektroenergetyki

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.

Ćwiczenia prowadzone są w oparciu o dwa programy:

- MET-NUM – program napisany w Turbo Pascalu przez pracowników Instytutu Informatyki Uniwersytetu Wrocławskiego zawierający moduły składające się z programów demonstracyjnych oraz pakietów zawierających wybrane funkcje i procedury
- MATLAB – pakiet obliczeniowy firmy MathWorks umożliwiający dokonywanie dowolnych obliczeń numerycznych

Celem powyższych ćwiczeń jest zarówno poznanie metod przedstawionych w obydwu programach jak i porównanie ich ze sobą pod kątem dokładności otrzymywanych wyników.

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 nawet małe odchylenia danych wejściowych mają duży wpływ na wynik końcowy)
- stabilność algorytmów - w trakcie obliczeń następuje kumulacja błędów; obliczenia w programie MATLAB 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, a jeżeli powodują uzyskanie wyniku znacznie oddalonego od prawidłowego to mówimy o niestabilnym algorytmie obliczeniowym.

Jedynym używanym typem zmiennych 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.

operatory arytmetyczne:

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

operatory 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 – 0.333333333333334).

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. $(x.^4) \cdot \tan(x) + x \cdot \sin(x) - x / \cos(x)$). 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 |
| clc | czyszczenie okna poleceń (<i>Command Window</i>) |

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')**, lub ustawić tę ścieżkę w MATLAB-ie.

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 nowym pliku lub 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 pominięcie wyświetlania pośrednich obliczeń
disp() - wyświetlenie wyniku

Elementy macierzy lub tablicy wartości umieszcza się zawsze w nawiasach kwadratowych, natomiast nawiasy zwykle zarezerwowane są dla poleceń i funkcji MATLAB-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*')*).

Program pamięta wszystkie wykorzystywane zmienne, dlatego 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 all**.

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

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

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

b) dzielenie

macierzy

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

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

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

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

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

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

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

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

$$\mathbf{A} / \mathbf{B} = \begin{bmatrix} 3 & -1 \\ 4 & -1 \end{bmatrix}$$

tablic wartości

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

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

$$2 ./ \mathbf{B} = \begin{bmatrix} 2 & 2 \\ 1 & 2 \end{bmatrix}$$

2 / B - działania nie można wykonać

c) potęgowanie

macierzy

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

$$\mathbf{A}^3 = \begin{bmatrix} 21 & 34 \\ 34 & 55 \end{bmatrix}$$

tablic wartości

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

$$\mathbf{A}^{-1} = \begin{bmatrix} 1 & 8 \\ 8 & 27 \end{bmatrix}$$

d) dzielenie lewostronne

macierzy

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

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

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

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

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

np.:

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

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

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

tablic wartości

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

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

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 |
| : | (dwukropek) | 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) | - funkcje trygonometryczne | pi | - liczba Π |
| atan(x) | | | |
| sin(x) | | | |
| tan(x) | | | |

Ćwiczenia nr 1 i 2

PRZYBLIŻANIE FUNKCJI

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:

- Istnieją dwie podstawowe metody interpolacji:

- ### Przebieg ćwiczenia MET-NUM (program INTERPOL):

- ## Błędy w węzłach

[illegible]

Moduł z maksymalnej wartości błędu w przedziale interpolacji

[illegible]

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

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

następującymi metodami:

- **linear** - interpolacja liniowa
- **spline** - interpolacja funkcjami sklejanymi trzeciego stopnia
- **cubic** - interpolacja wielomianami trzeciego rzędu
- **nearest** - interpolacja za pomocą funkcji schodkowej

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.

Spośród metod dostępnych w programie MET-NUM metoda Lagrange'a (przybliżanie funkcji wielomianem algebraicznym o stopniu n zależnym od liczby węzłów) dla dwóch węzłów stanowi interpolację liniową, dla trzech węzłów będzie to przybliżanie za pomocą wielomianu stopnia drugiego, dla czterech za pomocą wielomianu stopnia trzeciego itd.

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, 4, 5 i 9 węzłów)
- 3.) wyznaczyć wartości y_z funkcji $y = f(x)$ w węzłach z ($y_z = f(z)$)
- 4.) wykorzystując polecenie **interp1** dokonać interpolacji rozpatrywanej funkcji o danych węzłach (z, yz) , w punktach x_i , dla wszystkich metod przedstawionych powyżej (wyznaczany jest wówczas wektor y_i wartości funkcji interpolującej w punktach x_i)
- 5.) obliczyć 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
- 7.) porównać między sobą wyniki otrzymane w programie MATLAB (narysować w skali logarytmicznej wykresy błędu w funkcji liczby węzłów i dokonać wyboru optymalnej metody), a następnie dokonać odpowiedniego porównania z wynikami z programu MET-NUM.

Porównanie wyników

| Metoda | | Liczba węzłów | | | | |
|---|---------|---------------|---|---|---|---|
| | | 2 | 3 | 4 | 5 | 9 |
| MET-NUM (m. Lagrange'a, węzły równoodległe) | | | | | | |
| MET-NUM (f. sklepane węzły równoodległe) | | | | | | |
| MATLAB (interp1) | linear | | | | | |
| | spline | | | | | |
| | cubic | | | | | |
| | nearest | | | | | |

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')
```

```
%(x,y) - współrzędne węzłów interpolacji
```

```
%(xi,yi) - współrzędne punktów w których
```

```
% dokonywana jest interpolacja
```

```
plot(x,y,'*',xi,yi)
```

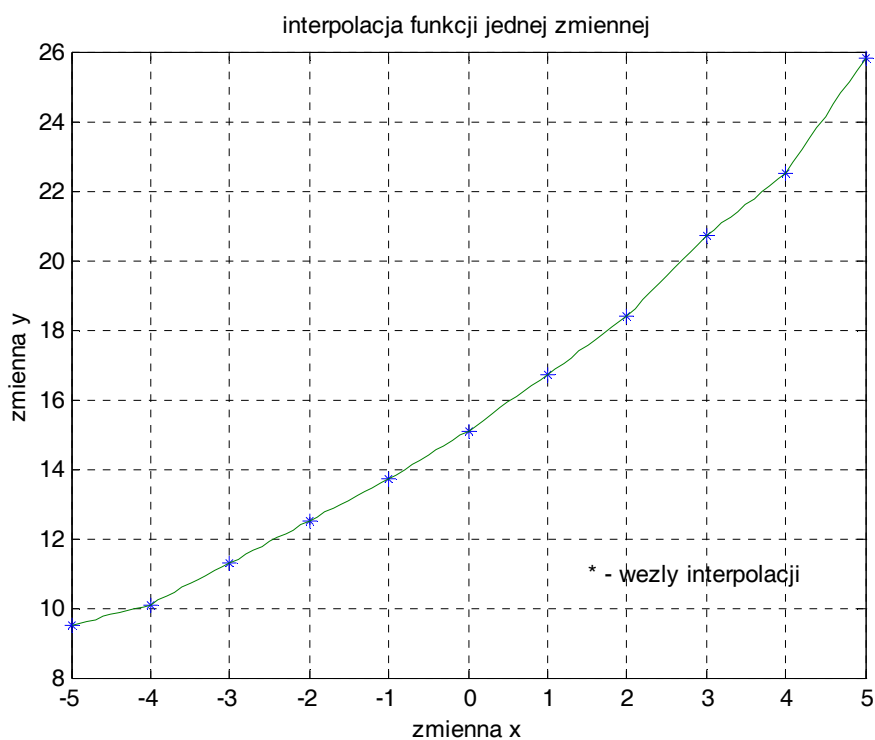
```
grid on
```

```
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
yz=(z.^2).*sin(pi*z)
yi=interp1(z,yz,x)

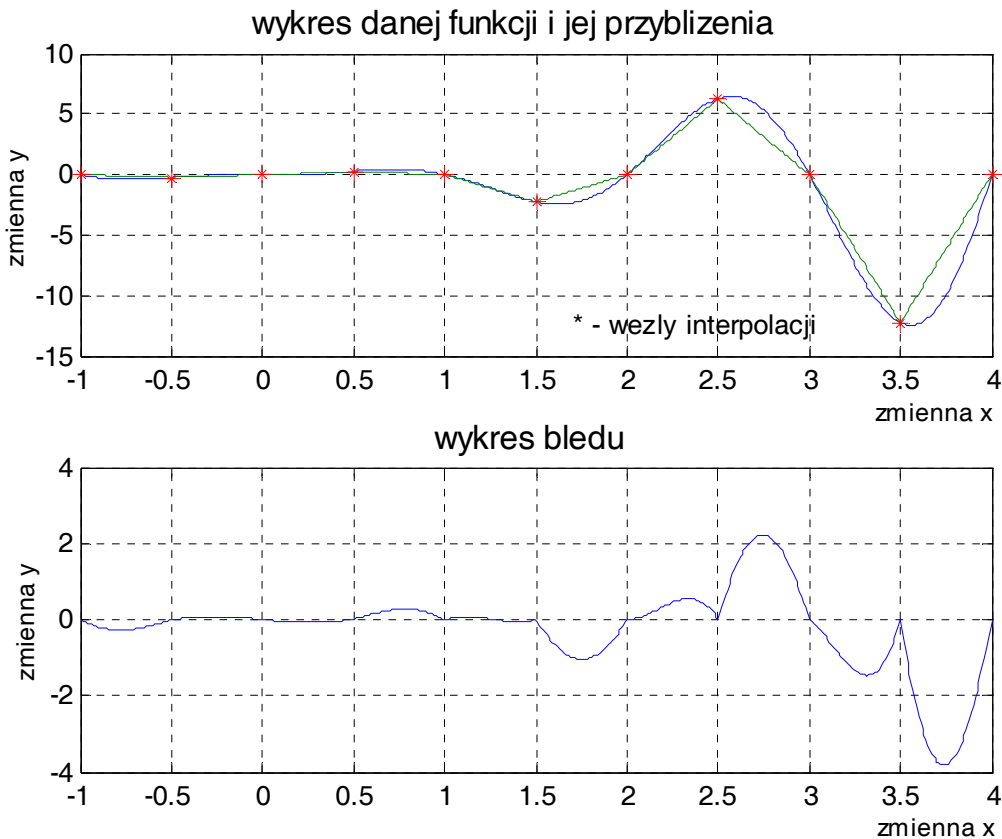
% (z,yz) - współrzędne węzłów interpolacji
% yi - wartości funkcji przybliżającej w punktach x
% przedziału interpolacji

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

% bl - błąd interpolacji
% blm - max wartość błędu interpolacji

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

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



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

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 pf taki, że:

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

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

Przebieg ćwiczenia MET-NUM (program UNIFAPPR):

- 1.) dla stopni wielomianu podanych w tabelce spisać wartości błędu bezwzględnego dla wszystkich metod; wyniki zamieścić w tabeli
- 2.) dokonać wyboru najlepszej metody aproksymacji spośród metod prawie optymalnych, pozostałe metody uszeregować ze względu na wielkość błędu
- 3.) w oparciu o wyniki dla najwyższego stopnia wielomianu, zbadać jak zachowują się współczynniki wielomianu aproksymującego, w zależności od charakteru funkcji
- 4.) dla najmniejszego i największego stopnia wielomianu przerysować wykres błędu dla aproksymacji optymalnej
- 5.) narysować w skali logarytmicznej wykresy błędu aproksymacji w funkcji stopnia wielomianu aproksymującego dla wszystkich metod aproksymacji (oprócz **B** – aproksymacji optymalnej)
- 6.) na podstawie wykresu określić jak zachowuje się błąd aproksymacji przy wzroście stopnia wielomianu

Błąd aproksymacji

| Metoda | Stopień wielomianu | | | | | | | |
|---------------|--------------------|-----|-----|-----|-----|-----|-----|-----|
| | n = | n = | n = | n = | n = | n = | n = | n = |
| B | | | | | | | | |
| P | | | | | | | | |
| S | | | | | | | | |
| I | | | | | | | | |
| J | | | | | | | | |
| G | | | | | | | | |
| K | | | | | | | | |
| L | | | | | | | | |
| M | | | | | | | | |
| MATLAB | | | | | | | | |

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

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

ważne jest, aby wartość \mathbf{J} była możliwie jak najmniejsza.

Dokonywana jest aproksymacja funkcji \mathbf{y} za pomocą wielomianu $\mathbf{W}(\mathbf{x})$:

$$\mathbf{W}(\mathbf{x}) = \mathbf{a}_r \mathbf{x}^r + \mathbf{a}_{r-1} \mathbf{x}^{r-1} + \dots + \mathbf{a}_1 \mathbf{x} + \mathbf{a}_0$$

Na podstawie posiadanych danych $\mathbf{y} = f(\mathbf{x})$

| | | | | |
|----------------|-------|-------|-------|-------|
| \mathbf{x}_i | x_1 | x_2 | | x_N |
| \mathbf{y}_i | y_1 | y_2 | | y_N |

konstruowana są:

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

$$\mathbf{X} = \begin{bmatrix} x_1^r & x_1^{r-1} & \dots & 1 \\ x_2^r & x_2^{r-1} & \dots & 1 \\ \dots & \dots & \dots & \dots \\ x_N^r & x_N^{r-1} & \dots & 1 \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} a_r \\ a_{r-1} \\ \dots \\ a_0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix}$$

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

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

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

poprzez rozwiązanie równania:

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

minimalizowany jest średniokwadratowy błąd przybliżenia \mathbf{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 \mathbf{X} lub liczby) a otrzymane wartości umieszcza w wektorze \mathbf{p} lub macierzy \mathbf{P} .

Przebieg ćwiczenia MATLAB:

- 1.) wyznaczyć wartości funkcji aproksymowanej $\mathbf{y} = \mathbf{f}(\mathbf{x})$ i narysować jej wykres w całym przedziale aproksymacji $< -1, 1 >$ z krokiem **0,01**
- 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 ($\mathbf{y} = \mathbf{f}(\mathbf{x})$) i wielomianu aproksymującego w jednym układzie współrzędnych a wykres bezwzględnego błędu aproksymacji w drugim; wykresy i napisany program zamieścić w sprawozdaniu
- 6.) wykres błędu w funkcji stopnia wielomianu aproksymującego umieścić na wspólnym wykresie z krzywymi uzyskanymi z programu MET-NUM
- 7.) porównać wyniki z obydwu programów

Przykład

1. 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.

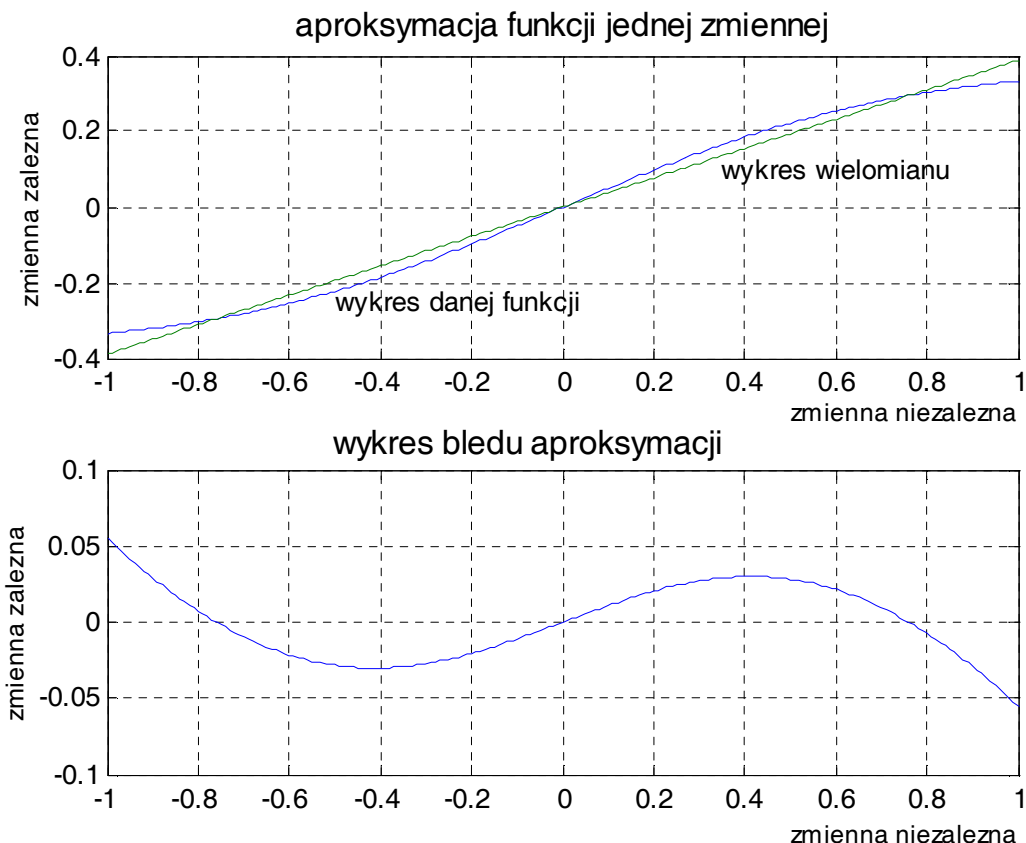
```
%%aproksymacja funkcji jednej zmiennej
```

```
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))

%r - stopień wielomianu przybliżającego
%a - wektor współczynników wielomianu przybliżającego
%p - wektor wartości wielomianu przybliżającego
%bl - błąd aproksymacji
%blm - max wartość błędu aproksymacji

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

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, wykorzystując wcześniej obliczone przybliżenia.

W programie MET - NUM przedstawione są 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 podprzedział na krańcach którego funkcja ma przeciwne 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.

Podobnie 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]**, przez punkty **a** i **b** prowadzona jest sieczna do wykresu funkcji. Sieczna dzieli przedział **[a, b]** na dwie części, wybierany jest ten podprzedział na krańcach którego funkcja przyjmuje przeciwne 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.

Przebieg ćwiczenia MET-NUM (program ROOT):

- 1.) przepisać współczynniki wielomianu
- 2.) dla wybranego zera zamieścić w tabeli wartości $|f|$ dla kolejnych iteracji i wszystkich metod
- 3.) przepisać przybliżoną wartość zera wyznaczoną za pomocą każdej z metod i określić dokładność otrzymanego wyniku
- 4.) na podstawie wyników zamieszczonych w tabeli wykonać w skali logarytmicznej wykresy $|f| = f(l. \text{ iteracji})$ i dokonać oceny zbieżności metod

Moduł wartości funkcji dla kolejnych iteracji

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

W programie MATLAB jest dostępna 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 **a₀, a₁, a_{n-1}, a_n**.

Jeżeli $W(x) = a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \dots + a_{n-1} x + a_n$

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
- 4.) porównać wyniki z obydwu programów

Przykład

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

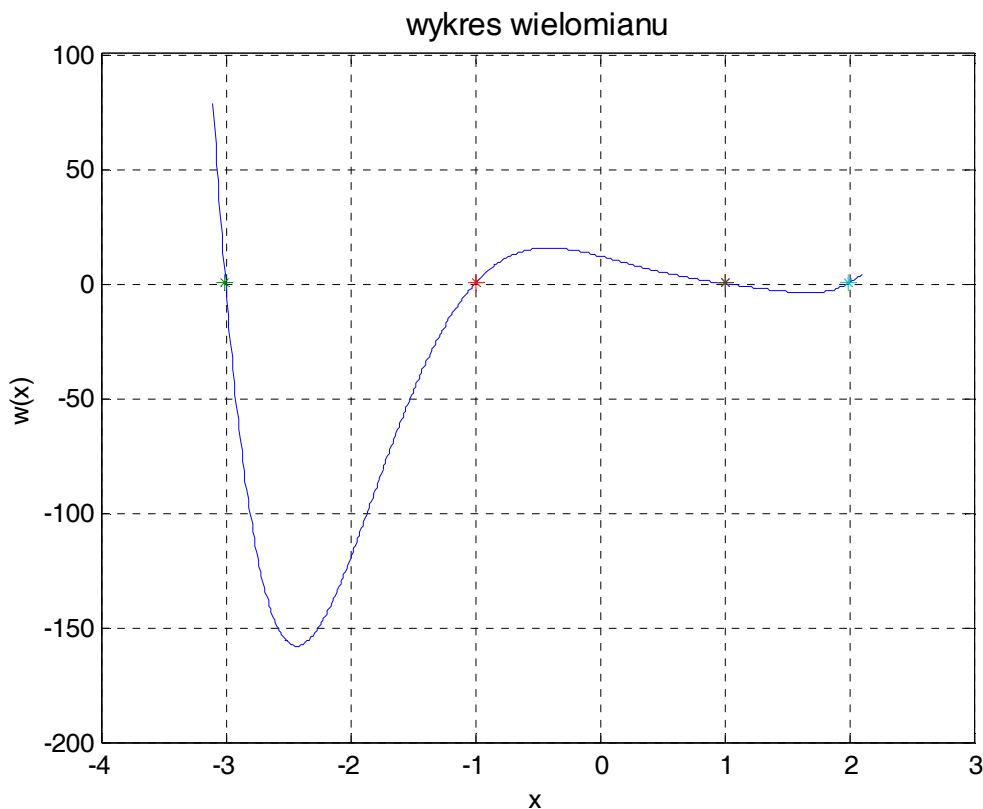
$$w(x) = x^6 - x^5 - 7x^4 + 15x^3 - 6x^2 - 14x + 12$$

`%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

Jeżeli \mathbf{x}_0 jest zerem krotności \mathbf{k} to spełniona jest zależność:

$$\mathbf{f}(\mathbf{x}_0) = \mathbf{f}'(\mathbf{x}_0) = \dots = \mathbf{f}^{k-1}(\mathbf{x}_0) = \mathbf{0} \quad \text{natomiast} \quad \mathbf{f}^k(\mathbf{x}_0) \neq \mathbf{0}$$

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).

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

jako wartość zera przyjmowane jest przybliżenie \mathbf{x}_{n+1} .

Przebieg ćwiczenia MET-NUM (program ZERAFUNK):

- 1.) na podstawie wykresu funkcji **f** i wykresu funkcji przez pochodną **f/pf** określić przedział zawierający wszystkie zera funkcji (w przypadku funkcji oscylacyjnych przedział zawierający 8 zer)
- 2.) zlokalizować zera dla funkcji **f** (zera o nieparzystej krotności) oraz dla funkcji przez pochodną **f/pf** (wszystkie zera) i dokonać wyboru zer o parzystej krotności (długość przedziału tablicowania nie może być większa od 3, natomiast krok tablicowania **h** nie może być mniejszy niż 10^{-4})
- 3.) dla dokładności **10^{-5}** wyznaczyć wartości wszystkich zer wszystkimi metodami (funkcje oscylacyjne 8 zer); wyniki zamieścić w tabeli
- 4.) dla zer wielokrotnych policzyć ich krotność

Wartości zer dla funkcji f i funkcji f/pf

[illegible]

W programie MATLAB rzeczywiste zera funkcji można wyznaczyć wykorzystując polecenie **fzero**. W tym celu należy w skrypcie (z rozszerzeniem **'m'**, np. *plik.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 obliczeń **tol**.

Składnia polecenia **fzero**, które należy napisać innym pliku lub w oknie programu, jest następująca:

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

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 $2,22 \cdot 10^{-16}$. 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)
- 4.) porównać wyniki z obydwu programów

Przykład

1. W przedziale $\langle -3, 3 \rangle$ wyznaczyć wszystkie zera funkcji: $y = (x + 2) \arctg(x - 1)$ 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:

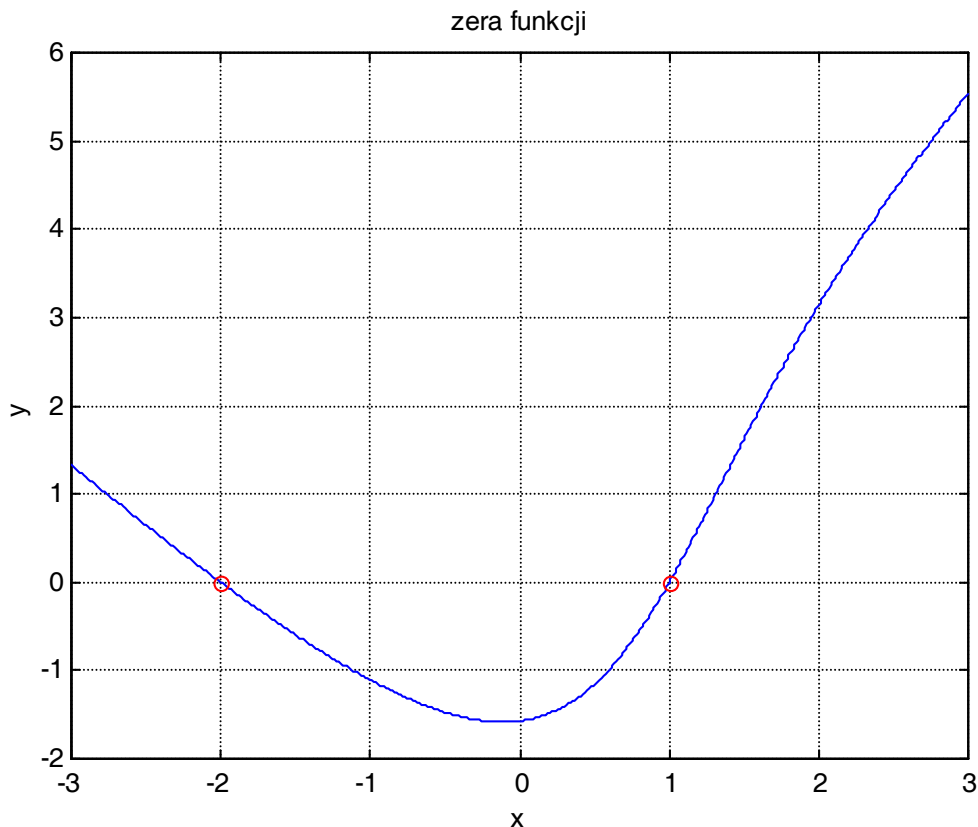
function y=f(x)
y=(x+2).*atan(x-1)

%w nowym skrypcie zamieścić polecenia służące do wyznaczenia zer i narysowania wykresu:

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

%ustalić punkty w otoczeniu których poszukiwane będą zera (np. 3 i -3)
z1=fzero('zera',-3)           %obliczenia dokonywane są z dokładnością 2,22*10-16
z2=fzero('zera',3,1e-6)       %obliczenia dokonywane są z dokładnością 10-6

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



W przedziale $\langle -3, 3 \rangle$ funkcja ta posiada dwa zera rzeczywiste:

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

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.

Podczas 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 MET-NUM (program ZERAWIEL):

- 1.) wprowadzić współczynniki wielomianu (współczynniki są tak dobrane, że rozpatrywane wielomiany stopnia siódmego posiadają 5 zer rzeczywistych i jedną parę zer zespolonych sprzężonych)
- 2.) obejrzeć wykres wielomianu i dokonać lokalizacji obszaru zawierającego zera
- 3.) dla dokładności 10^{-5} wyznaczyć wszystkie zera metodą Laguerre'a
- 4.) kolejno dla dokładności 10^{-3} , 10^{-6} , 10^{-9} , 10^{-12} i 10^{-15} wyznaczać wszystkie zera najpierw metodą Laguerre'a a następnie zera rzeczywiste pozostałymi metodami; dla wybranego zera rzeczywistego przepisać liczbę iteracji dla poszczególnych metod
- 5.) dla powyższego zera narysować wykresy w skali logarytmicznej **l. iteracji = f(dokładności)** dla wszystkich metod
- 6.) na podstawie wykresów określić koszty metod

Przebieg ćwiczenia MATLAB:

- 1.) narysować wykres wielomianu $W(x)$ w przedziale zawierającym wszystkie zera
- 2.) określić wektor a współczynników wielomianu $W(x)$
- 3.) wykorzystując polecenie **roots** wyznaczyć wszystkie zera wielomianu $W(x)$
- 4.) zmieniając dokładność w poleceniu **fzero** (10^{-3} , 10^{-6} , 10^{-9} , 10^{-12} i 10^{-15}) wyznaczać kolejno wartość tego samego zera rzeczywistego co w ćwiczeniu MET-NUM
- 5.) porównać wyniki z obydwu programów

Wartość zera i liczba iteracji dla poszczególnych metod w funkcji dokładności

| Metoda | Dokładność | | | | | | | | | |
|----------------------------|--------------|--------|--------------|--------|--------------|--------|--------------|--------|--------------|--------|
| | 10^{-3} | | 10^{-6} | | 10^{-9} | | 10^{-12} | | 10^{-15} | |
| | Wartość zera | l. it. | Wartość zera | l. it. | Wartość zera | l. it. | Wartość zera | l. it. | Wartość zera | l. it. |
| Bisekcji | | | | | | | | | | |
| Siecznych | | | | | | | | | | |
| Newtona | | | | | | | | | | |
| Steffensena | | | | | | | | | | |
| Laguerre'a | | | | | | | | | | |
| Laguerre'a (z deflacją) | | | | | | | | | | |
| MATLAB | | x | | x | | x | | x | | x |

Ćwiczenia nr 5 i 6

ALGEBRA LINIOWA

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:

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

$\mathbf{A} = [a_{ij}]$ - macierz układu
 $\mathbf{b} = [b_1, b_2, \dots, b_n]^T$ - 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 metodami przedstawionymi w programie MET-NUM:

- 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:

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

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.

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

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

$\mathbf{\Sigma}$ - macierz przekątniowa

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

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

Moduł (wartość bezwzględna wyznacznika macierzy **A**) jest iloczynem jej wszystkich wartości szczególnych

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

Dokładność numerycznie 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:

$$\text{cond}(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| \quad \text{gdzie } \|\mathbf{A}\| = \sigma_1$$

($\|\mathbf{A}\|$ - norma macierzy **A** zdefiniowana jako największa wartość szczególna σ_1 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:

$$\text{cond}_2(\mathbf{A}) = \sigma_1 / \sigma_n$$

σ_1 - największa wartość szczególna macierzy **A**
 σ_n - najmniejsza wartość szczególna macierzy **A**

2.) dla normy Frobeniusa:

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

3.) współczynniki numerycznej osobliwości macierzy:

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

n - stopień macierzy **A**
 σ_1 - największa wartość szczególna macierzy **A**
macheps – dokładność maszynowa
(tzn. najmniejsza liczba dodatnia reprezentowana w komputerze taka, że $1 + \text{macheps} > 1$)
w programie MET-NUM równa $\sim 1.8189894 \cdot 10^{-12}$

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

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

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

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

4.) Współczynnik numerycznej poprawności algorytmu:

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

$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{y}$ - wektor residuum

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

\mathbf{y} - rozwiązanie numeryczne

$\|\mathbf{y}\|_2$ - druga norma wektora rozwiązania numerycznego
(pierwiastek z sumy kwadratów jego współrzędnych)

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

macheps - dokładność maszynowa

Jeżeli \mathbf{wsp} jest rzędu \underline{n} bądź $\underline{n^2}$, to przyjmuje się, że algorytm jest numerycznie poprawny.

5.) Błąd względny rozwiązania:

$$\mathbf{W} = \|\mathbf{x} - \mathbf{y}\|_2 / \|\mathbf{x}\|_2$$

\mathbf{x} - rozwiązanie dokładne

\mathbf{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 \mathbf{x} . Ustalona wartość rozwiązania jest następująca: $\mathbf{x}[i] = i$; następnie mnożąc macierz układu \mathbf{A} przez wektor rozwiązania \mathbf{x} otrzymuje się wektor prawych stron \mathbf{b} . Na podstawie dokładnej wartości macierzy układu \mathbf{A} i wektora prawych stron \mathbf{b} wyznaczane są wektory rozwiązania numerycznego \mathbf{x}_1 , \mathbf{x}_2 i \mathbf{x}_3 trzema wymienionymi wyżej metodami.

Przebieg ćwiczenia MET-NUM (program GAUSS):

- 1.) przepisać informację na temat macierzy i zależność określającą jej elementy
- 2.) dla stopnia macierzy $\mathbf{n} = 5$ wypisać jej elementy
- 3.) wynotować, dla dostępnych trzech metod rozwiązywania układów równań liniowych wartości: wyznacza macierzy, błędu względnego \mathbf{W} i współczynnika numerycznej poprawności algorytmu; a ponadto dokładność maszynową, najmniejszy element główny, największy element główny, współczynniki numerycznej osobliwości macierzy \mathbf{tol} i $\mathbf{tol1}$, spektralny wskaźnik uwarunkowania z SVD, największą wartość szczególną i najmniejszą wartość szczególną.
- 4.) zgodnie z tabelką wprowadzać zaburzenia do elementu \mathbf{a}_{11} macierzy \mathbf{A} ; przepisać wartości błędu względnego \mathbf{W} dla rozpatrywanych metod
- 5.) zmienić stopień macierzy na 7 i powtórzyć wszystkie polecenia
- 6.) na podstawie powyższych współczynników i wskaźników określić poprawność algorytmu i numeryczne uwarunkowanie macierzy
- 7.) w oparciu o wartości błędu względnego (bez zaburzenia) uszeregować poznane metody
- 8.) zbadać wrażliwość rozwiązania na wprowadzane zaburzenia (wykorzystać wartości wskaźników uwarunkowania z SVD)
- 9.) określić wpływ stopnia macierzy na uzyskane wyniki

Wyniki obliczeń dla poszczególnych metod

| Wskaźniki, wyznaczniki i współczynniki | Metoda (MET-NUM) | | |
|--|------------------|-----------------|---------------------|
| | Gauss bez wyboru | Gauss z wyborem | Odwracanie macierzy |
| dokładność maszynowa (macheps) | | | |
| współczynnik numerycznej poprawności algorytmu | | | |
| wyznacznik macierzy | | | x |
| błąd względny W | | | |
| najmniejszy element główny | | | x |
| największy element główny | | | x |
| współczynnik numerycznej osobliwości tol | | | |
| najmniejsza wartość szczególna | | | |
| największa wartość szczególna | | | |
| współczynnik numerycznej osobliwości tol1 | | | |
| spektralny wskaźnik uwarunkowania z SVD | | | |

Wpływ zaburzenia na wartość błędu

| Zaburzenie | Błąd względny W | | |
|----------------|------------------|-----------------|---------------------|
| | Gauss bez wyboru | Gauss z wyborem | Odwracanie macierzy |
| bez zaburzenia | | | |
| + 0.01 | | | |
| - 0.01 | | | |
| + 0.001 | | | |
| - 0.001 | | | |
| + 0.0001 | | | |
| - 0.0001 | | | |

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

| | |
|-------------------------|--|
| inv(A) | - odwracanie macierzy |
| A\b | - rozwiązanie 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 jej elementów |

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 $n = 5$ i porównać z macierzą z programu MET-NUM
- 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 ($tol = 10*n*eps*norm(A,'fro')$ i $tol1 = n*eps*\sigma_1$)
- 8.) wyznaczyć błędy względne rozwiązań ($W2 = norm(x - x2)/norm(x)$ i $W3 = norm(x-x3)/norm(x)$)
- 9.) obliczyć współczynniki numerycznej poprawności algorytmu:
 $wp2 = norm(r2)/(eps*norm(x2)*norm(A,'fro'))$
 $wp3 = norm(r3)/(eps*norm(x3)*norm(A,'fro'))$
- 10.) wyznaczyć czas obliczeń dla obydwu metod
- 11.) porównać wyniki z obydwu programów

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 | | |
| najmniejszy element główny | | |
| największy element główny | | |
| współczynnik numerycznej osobliwości tol | | |
| najmniejsza wartość szczególna | | |
| największa wartość szczególna | | |
| współczynnik numerycznej osobliwości tol1 | | |
| wskaźnik 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  
    2 1 -1 -2  
    5 1 2 -1  
    3 -1 3 -3]
```

`%macierz układu`

```
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
20.0000
13.0000
-5.0000

-10.0000
20.0000
13.0000
-5.0000

Algebra liniowa (wartości własne macierzy)

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

$$Ax = \lambda x$$

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

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

$\lambda_1 \dots \lambda_n$ - wartości własne macierzy A

q_j - wektory własne macierzy A (kolumny macierzy Q)

$$A q_j = \lambda_j q_j$$

$j = 1 \dots n$

Wszystkie wartości własne są pierwiastkami wielomianu charakterystycznego:

$$\det(A - \lambda I)$$

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

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

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

$$B = Z^T A Z$$

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

Macierz B jako podobna do macierzy A ma takie same wartości własne.

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

W procedurach dostępnych w programie MET-NUM koszty metod stanowią:

- czas wykonywania obliczeń
- liczba cykli i obrotów (JACSYM)
- liczba podziałów (BISECT)
- liczba iteracji (QLSYM)

Do pomiaru czasu wykorzystano procedury pakietu DOS-GETTIME; czasy obliczeń obarczone są 55 ms błędem wynikającym z realizacji procedury i mogą różnić się nawet w przypadku powtarzania obliczeń.

Należy pamiętać, że procedura BISECT nie wyznacza wektorów własnych.

Zadanie obliczania wartości własnych macierzy symetrycznej jest bardzo dobrze uwarunkowane. Oznacza to, że wartości własne nie są wrażliwe na małe zmiany elementów macierzy 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.

Przebieg ćwiczenia MET-NUM (program JACOBI):

- 1.) dla stopnia macierzy $n = 2, 4 \dots 20, 20$ tworzyć kolejno macierze A symetryczne o zadanych wartościach własnych tworzących ciąg arytmetyczny, geometryczny lub harmoniczny
- 2.) dla stopnia $n = 4$ wynotować elementy macierzy A i jej wartości własne
- 3.) dane dotyczące kosztów metod umieścić w tabeli
- 4.) dla wszystkich metod narysować wykresy: **czas = f(st. macierzy)** i **l. operacji = f(st. macierzy)** (do czasów metod BISECT i QLSYM należy dodać czas procedury TRIDIAG)
- 5.) zamieścić wnioski dotyczące kosztów poszczególnych metod

Koszty metod

| Stopień macierzy | Macierz SYMETR RAND | | | | | | |
|------------------|---------------------|--------|-------|--------|----------|-------|----------|
| | TRIDIAG | JACSYM | | BISECT | | QLSYM | |
| | czas | czas | cykle | czas | podziały | czas | iteracje |
| 2 | | | | | | | |
| 4 | | | | | | | |
| 6 | | | | | | | |
| 8 | | | | | | | |
| 10 | | | | | | | |
| 12 | | | | | | | |
| 14 | | | | | | | |
| 16 | | | | | | | |
| 18 | | | | | | | |
| 20 | | | | | | | |
| 20 | | | | | | | |

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

- $L = \text{eig}(A)$ - wektor L zawiera wartości własne macierzy kwadratowej A
- $[V, D] = \text{eig}(A)$ - macierz D zawiera wartości własne macierzy A umieszczone na przekątnej
- kolumny macierzy V są wektorami własnymi macierzy A

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

$$AV = VD$$

$$D = V^T AV$$

Przebieg ćwiczenia MATLAB:

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

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                          %pętla for generujące macierz
    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 własne 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

Całkowanie

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

Jeżeli $F'(x) = f(x)$ to $\int_a^b f(x)dx = F(b) - F(a)$ F - 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:

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

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.

Przebieg ćwiczenia MET-NUM (program TRAPEZY):

- 1.) zaobserwować w jaki sposób wyznaczana jest całka dla różnych funkcji podcałkowych za pomocą prostych kwadratur
- 2.) opierając się na ilustracji krokowej wybrać dla danej funkcji podcałkowej tę z prostych kwadratur, która zapewnia najmniejszy błąd bezwzględny $w_d - w_k$, (w_d - wartość dokładna całki, w_k - wartość całki policzona daną kwadraturą)
- 3.) korzystając z kwadratur złożonych zaobserwować w jaki sposób konstruowane jest rozwiązanie dla różnych kwadratur i przepisać, dla danej funkcji podcałkowej, wartości błędu bezwzględnego dla kolejnych wartości podziałów przedziału całkowania (2, 4, 8, 16, 32)
- 4.) narysować wykres **błąd = f(liczba podziałów przedziału)** dla wszystkich metod; wybrać metodę zapewniającą najmniejszy błąd

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 $x_0 = a$ 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:

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

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

Łatwo zauważyć, 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 wstępnym 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 Q_0 całki funkcji $f(x)$ na danym przedziale (a, b) , oblicza się jej wartość po podziale przedziału na dwie części. Jeżeli moduł różnicy tych wartości jest mniejszy niż iloczyn modułu wstępnego przybliżenia i względnej tolerancji, to przybliżenie Q_0 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 trzy funkcje **quad**, **quad8** i **quadl** umożliwiające całkowanie numeryczne w oparciu o trzy różne procedury:

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

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

quadl – adaptacyjna kwadratura Lobbatta

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

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

Q = quadl('plik', a, b, tol, trace)

a, b - przedział całkowania

tol - tolerancja względna (domyślna 10^{-6})

trace - parametr ten, jeżeli ma wartość niezerową, umożliwia wyświetlenie wykresu funkcji podcałkowej z zaznaczonymi węzłami kwadratury

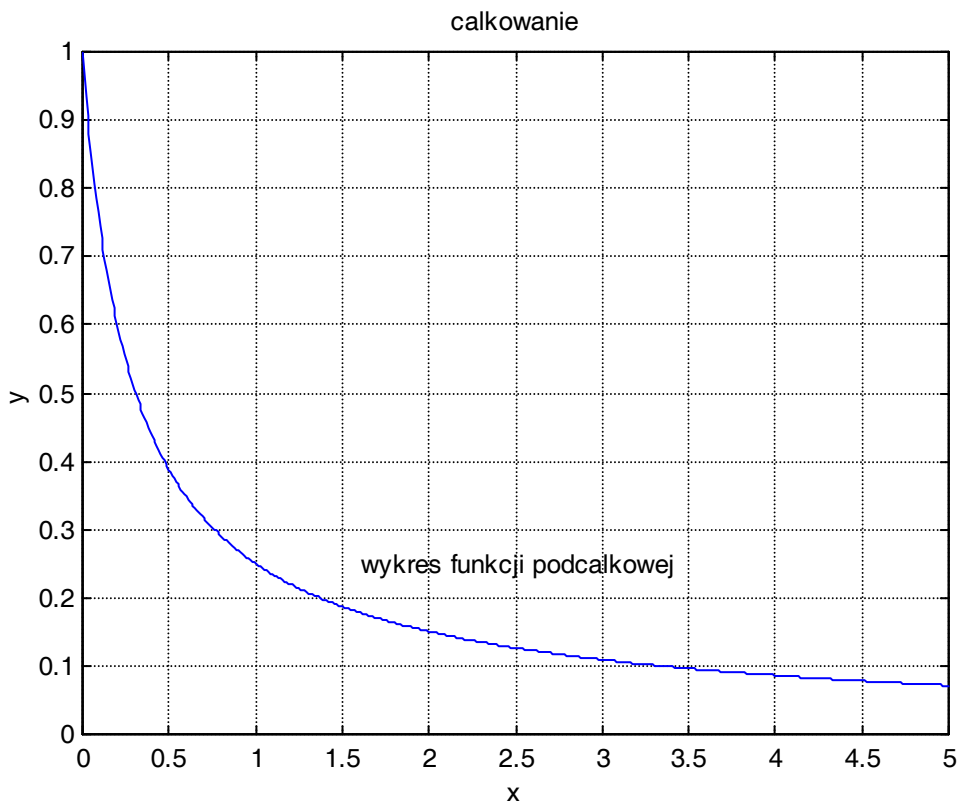
Przebieg ćwiczenia MATLAB:

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

Przykład

1. 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 nowym skrypcie zamieścić polecenie służące do obliczenia całki i narysowania wykresu:  
  
x=0:0.01:5  
y=1./(2*x+sqrt(3*x+1))  
  
Q=quad('calk',0,5)    %wyznaczenie wartości całki za pomocą funkcji quad  
  
plot(x,y)  
grid on  
title('calkowanie')  
text(1.2,0.25,'wykres funkcji podcałkowej')  
xlabel('x')  
ylabel('y')
```

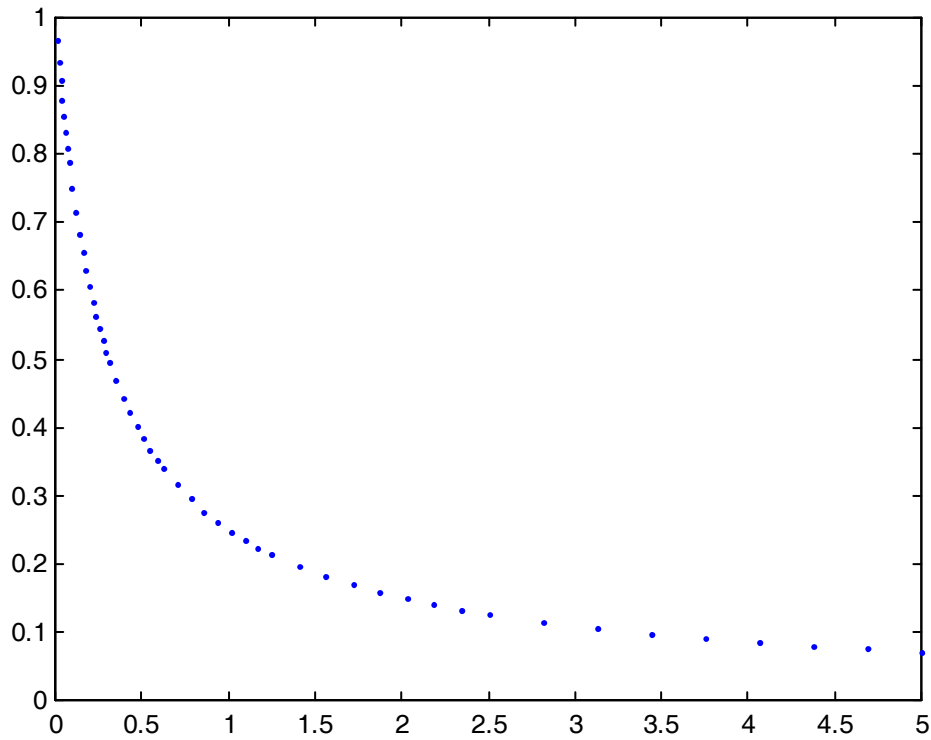


Wartość całki wynosi:

$$Q = 0.9437$$

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

`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.

Pakiet INTEGRAL umożliwia porównanie 9 algorytmów całkowania automatycznego (w tym jeden adaptacyjny). Kwadratury te można stosować do obliczania 20 całek. Funkcje podcałkowe zostały podzielone na cztery grupy:

- funkcje gładkie (nie zmieniające się szybko w przedziale **(a, b)** i posiadające ciągłe pochodne w tym przedziale); **A, B, C, D, E**
- funkcje prawie osobliwe (funkcje ciągłe w przedziale **(a, b)** o rosnących nieograniczenie modułach dla $x \rightarrow c$, gdzie c - punkt leżący w pobliżu przedziału **(a, b)**); **F, G, H, I, J**
- funkcje szybko oscylujące (funkcje mające wiele max i min lokalnych w przedziale całkowania); **K, L, M, N, O**
- funkcje przedziałami ciągłe, lub mające pierwsze pochodne przedziałami ciągłe (funkcje lub ich pochodne mają skończoną liczbę nieciągłości w przedziale całkowania); **P, Q, R, S, T**

Przebieg ćwiczenia MET-NUM (program INTEGRAL):

- 1.) obejrzeć wykres danej funkcji podcałkowej; przepisać jej wzór i przedział całkowania
- 2.) dla kolejnych dokładności 10^{-3} , 10^{-5} i 10^{-7} i wszystkich metod przepisać dla danej funkcji przybliżone wartości całki - **res**, błąd względny - **er**, oszacowanie błędu względnego - **est**, liczbę odwołań do funkcji podcałkowej - **nf**, czas obliczeń – **czas**
- 3.) na podstawie powyższych danych dokonać wyboru optymalnej metody całkowania pod kątem zapewnienia najmniejszej wartości błędu i najmniejszych kosztów (tzn. czasu obliczeń i liczby odwołań do funkcji podcałkowej)

Przebieg ćwiczenia MATLAB:

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

Tabele wyników kolejno dla dokładności 10^{-3} , 10^{-5} i 10^{-7}

| Metoda | Wyniki | | | | |
|----------------|--------|----|-----|----|------|
| | res | er | est | nf | czas |
| compNC | | | | | |
| compGL | | | | | |
| compLob | | | | | |
| compRad | | | | | |
| Simpson | | | | | |
| Quanc | | | | | |
| CCquad | | | | | |
| RomBerg | | | | | |
| Filon | | | | | |
| MATLAB (quad) | | x | x | x | x |
| MATLAB (quad8) | | x | x | x | x |
| MATLAB (quadl) | | x | x | x | x |

Ćwiczenia nr 8, 9 i 10

RÓŻNICZKOWANIE

Różniczkowanie

W sposób przybliżony ma być rozwiązane 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:

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

można sprowadzić do układu m równań rzędu pierwszego:

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

Przedstawione w programie MET-NUM przykłady rozwiązywania równań różniczkowych oparte są o jawne metody Rungego - Kuty. Ogólnie metody te można opisać wzorem:

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

gdzie

$$\begin{cases} K_1 = F(t_n, Y_n), \\ K_i = F(t_n + b_i h; Y_n + h \sum_{j=1}^{i-1} a_{ij} K_j) \end{cases}$$

dla $i = 2, 3, \dots, s$.

Współczynniki metody można przedstawić w tablicy:

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

Pomiędzy współczynnikami metody zachodzą następujące związki:

$$b_i = \sum_{j=1}^{i-1} a_{ij} \quad i = 2, 3, \dots, s \quad s - \text{liczba etapów metody}, \quad h - \text{krok}$$

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

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

$$Y_n \approx Y(t_n)$$

$$Y_{n+1} \approx Y(t_{n+1})$$

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

Przegląd 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 rozwiązanie numeryczne otrzymuje się z przedstawionych powyżej zależności opisujących metody Rungego-Kutty, opierając się o tablicę współczynników charakteryzującą daną metodę. Łatwo zauważyć, że dla metody Eulera wzór ten jest równaniem prostej.

W pozostałych metodach rozwiązanie numeryczne konstruowane jest w sposób analogiczny.

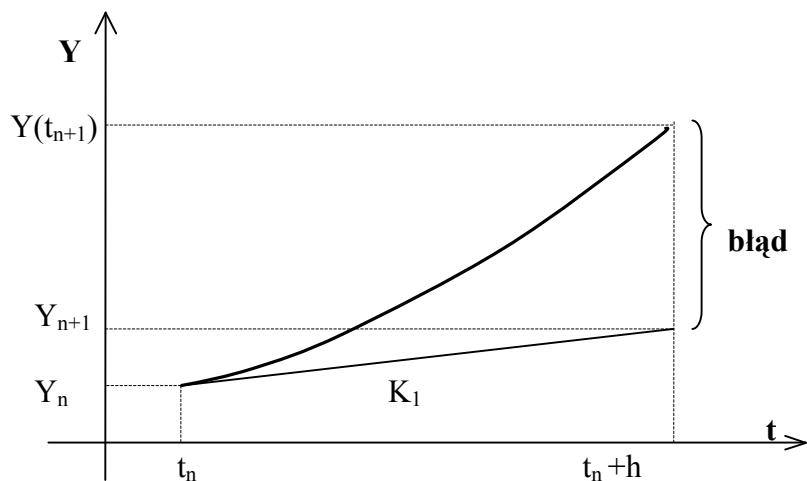
1) Metoda Eulera (1,1)

$$\left. \begin{array}{c} \begin{array}{|c|} \hline 1 \\ \hline \end{array} \quad b_i = 0; \quad a_{ij} = 0 \\ \omega_1 = 1 \end{array} \right\} \Rightarrow \underbrace{K_1 = F(t_n, Y_n), \quad K_i = 0}_{\Downarrow} \\ Y_{n+1} = Y_n + hK_1$$

tablica
współczynników

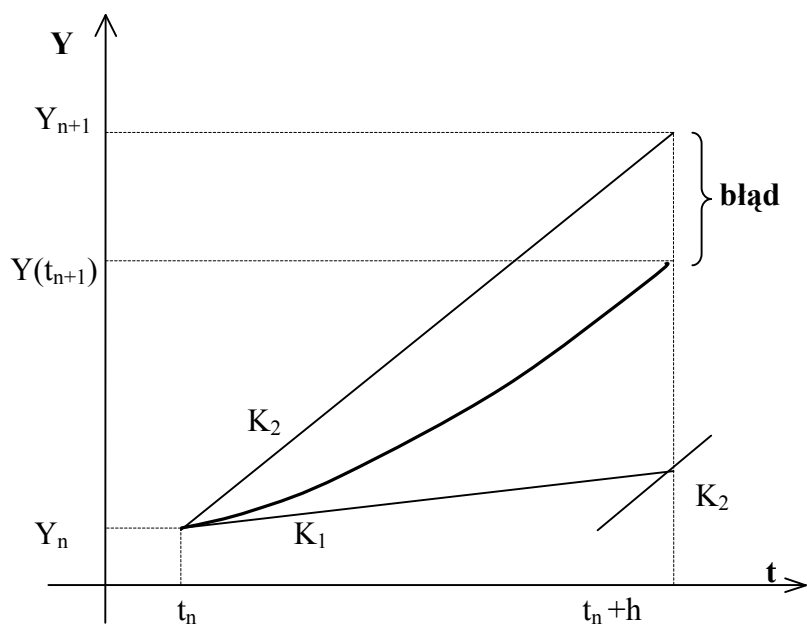
współczynniki

rozwiązanie numeryczne



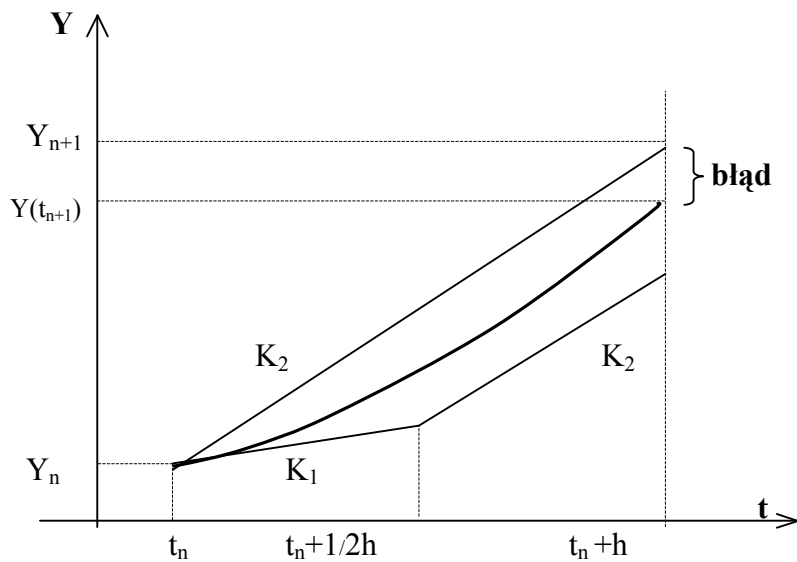
2) Modyfikacja metody Eulera (1,2)

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



3) Metoda Runge'go (2,2)

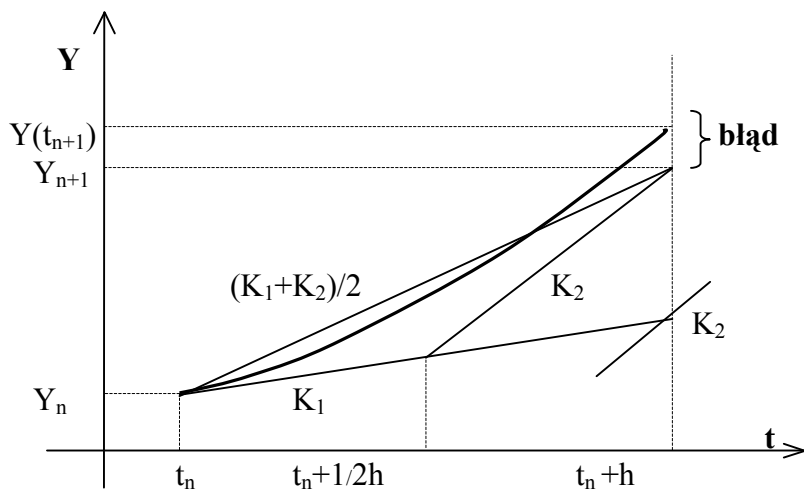
$$\left. \begin{array}{c|cc} 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array} \right\} \Rightarrow \underbrace{K_1 = F(t_n, Y_n), \quad K_2 = F(t_n+1/2h, Y_n+1/2hK_1)}_{\Downarrow} \\ Y_{n+1} = Y_n + hK_2$$



4) Metoda Eulera – Cauchy’go (2,2)

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

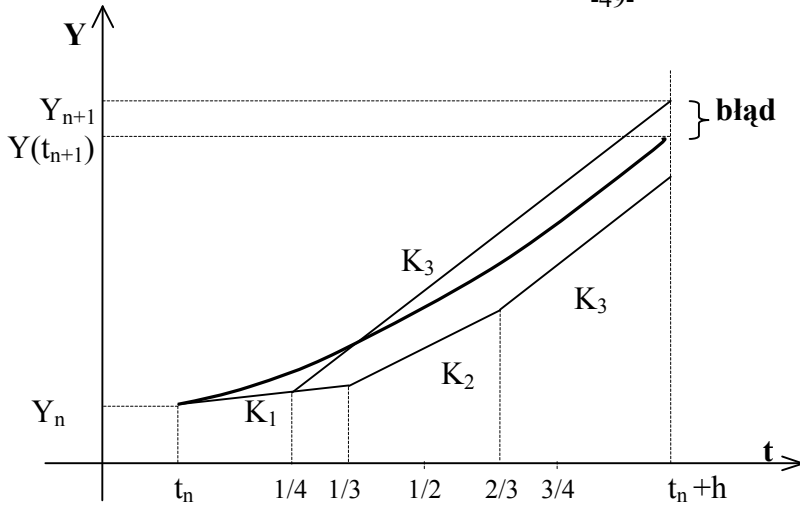
$$Y_{n+1} = Y_n + h(1/2K_1 + 1/2K_2)$$



5) Metoda Heuna (3,3)

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

$$Y_{n+1} = Y_n + h(1/4K_1 + 3/4K_3)$$



6) Metoda Rungego-Kutty (4,4)

| | | | | |
|-----|-----|-----|-----|-----|
| 1/2 | 1/2 | | | |
| 1/2 | 0 | 1/2 | | |
| 1 | 0 | 0 | 1 | |
| | 1/6 | 1/3 | 1/3 | 1/6 |

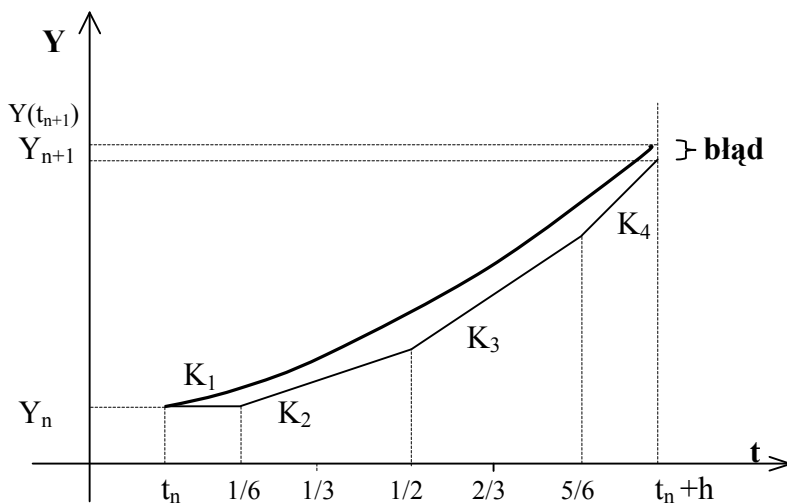
$$\begin{aligned}
 b_2 &= 1/2; & a_{21} &= 1/2; \\
 b_3 &= 1/2; & a_{31} &= 0; & a_{32} &= 1/2; \\
 b_4 &= 1; & b_{41} &= 0; & b_{42} &= 0; & b_{43} &= 1; \\
 \omega_1 &= 1/6; & \omega_2 &= 1/3; & \omega_3 &= 1/3; & \omega_4 &= 1/6;
 \end{aligned}$$

⇓

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

⇓

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



Przebieg ćwiczenia MET-NUM (program RK - ELEM):

- 1.) obejrzeć wszystkie przykłady (*Rodzina rozwiązań*)
- 2.) wybrać przykład **A** (*Elementarz metod*) i stosując kolejno wszystkie metody, prześledzić konstruowanie rozwiązania numerycznego oraz zaobserwować jak zachowuje się ciąg rozwiązań przy zmniejszaniu długości kroku
- 3.) dla przykładów **K** i **J** zastosować metodę *Eulera(1,1)* i zaobserwować zachowanie się rozwiązania przy zmianie długości kroku (*K* – niestabilny, *J* – nadstabilny)
- 4.) dla podanego zagadnienia początkowego przepisać zależności określające to zagadnienie jego warunki brzegowe i rozwiązanie dokładne
- 5.) dla powyższego zagadnienia (*Porównanie metod rzędu 1 – 4*) zbadać wpływ zmiany długości kroku (l. kroków zmieniać przez podwajanie **2, 4, 8 512**) na wartość błędu w punkcie końcowym dla czterech przedstawionych w programie metod
- 6.) przepisać dla podanych wyżej ilości kroków wartości błędów w punkcie końcowym
- 7.) dla wszystkich metod narysować wykresy (w skali logarytmicznej): **błąd = f(l. kroków)** i wyciągnąć wnioski nt. efektywności metod

Błąd w punkcie końcowym

| Liczba podziałów | Metoda | | | | | |
|------------------|--------------|--------------|-------------|----------------------|----------------|----------------|
| | Euler (1, 1) | Runge (2, 2) | Heun (3, 3) | Runge – Kutta (4, 4) | MATLAB (ode23) | MATLAB (ode45) |
| 2 | | | | | x | x |
| 4 | | | | | x | x |
| 8 | | | | | x | x |
| 16 | | | | | x | x |
| 32 | | | | | x | x |
| 64 | | | | | x | x |
| 128 | | | | | x | x |
| 256 | | | | | x | x |
| 512 | | | | | x | x |
| | | | | | | x |
| | | | | | x | |

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_1 i K_i oraz rozwiązanie numeryczne Y_{n+1} zamiast wartości przybliżonej Y_n przyjmowana jest wartość dokładna $Y(t_n)$ i liczone są wartości tych współczynników oraz rozwiązania numerycznego dla pewnego kroku h , a następnie wyznaczany jest błąd lokalny stanowiący różnicę pomiędzy rozwiązaniem dokładnym i numerycznym.

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

$$r_{n+1}(h) = Y(t_n + h) - (Y(t_n) + h \sum_{i=1}^s \omega_i K_i(h))$$

Dla dostatecznie małych wartości h , błąd aproksymacji można rozwinąć w szereg potęgowy względem h :

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

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

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

czyli błąd lokalny ma postać:

$$\mathbf{r}_{n+1}(\mathbf{h}) = \Phi(\mathbf{t}_n, \mathbf{Y}(\mathbf{t}_n))\mathbf{h}^{p+1} + \mathbf{O}(\mathbf{h}^{p+2})$$

$$\Phi(\mathbf{t}_n, \mathbf{Y}(\mathbf{t}_n))\mathbf{h}^{p+1} \quad \text{część główna błędu lokalnego}$$

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:

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

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

Błędem globalnym albo całkowitym metody w punkcie \mathbf{t}_n nazywamy wielkość:

$$\varepsilon_n = \mathbf{Y}_n - \mathbf{Y}(\mathbf{t}_n)$$

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

$$\frac{dy}{dt} = \mathbf{F}(t, \mathbf{y}), \quad \mathbf{y}(t_0) = \mathbf{y}_0; \quad \mathbf{y}, \mathbf{y}_0 \in R^n$$

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**).

$$[\mathbf{t}, \mathbf{Y}] = \text{ode23}(\text{'plik'}, \mathbf{t0}, \mathbf{tk}, \mathbf{y0}, \mathbf{tol}, \mathbf{tr})$$

$$[\mathbf{t}, \mathbf{Y}] = \text{ode45}(\text{'plik'}, \mathbf{t0}, \mathbf{tk}, \mathbf{y0}, \mathbf{tol}, \mathbf{tr})$$

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

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 rozwiązania należy, po zadeklarowaniu funkcji $\mathbf{F}(t, \mathbf{y})$, napisać w nowym pliku lub w oknie programu polecenie, o postaci jak powyżej, zawierające nazwę odpowiedniej funkcji **ode**.

Po wprowadzeniu oznaczenia $\mathbf{dy} = \mathbf{F}(t, \mathbf{y})$, funkcję $\mathbf{F}(t, \mathbf{y})$ 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 $\mathbf{F}(t, \mathbf{y})$ zamieścić wszystkie równania w macierzy.

Przebieg ćwiczenia MATLAB:

- 1.) w nowym pliku zadeklarować funkcję $F(t, y)$
- 2.) w kolejnym pliku umieścić zależność określającą rozwiązanie dokładne i wyznaczyć jego wartość w przedziale (t_0, t_k) oraz wykonać następujące obliczenia:
 - korzystając z polecenia **ode23** wyznaczyć wektor rozwiązania numerycznego w przedziale (t_0, t_k)
 - obliczyć błąd rozwiązania, tzn. różnicę pomiędzy rozwiązaniem dokładnym a numerycznym
 - policzyć liczbę kroków wykorzystując polecenie **size**
 - narysować w jednym układzie współrzędnych wykres rozwiązania dokładnego i numerycznego a w drugim wykres błędu
- 3.) przepisać wartość błędu w punkcie końcowym
- 4.) błąd rozwiązania w punkcie końcowym porównać z wartościami błędów uzyskanymi (przy odpowiedniej ilości kroków) dla metod z programu MET-NUM rzędu 2 i 3
- 5.) punkty 1-4 powtórzyć dla polecenia **ode 45**; wartości błędu w punkcie końcowym porównać z odpowiednimi wartościami błędów wyznaczonymi w programie MET-NUM dla metod rzędu 4

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:
```

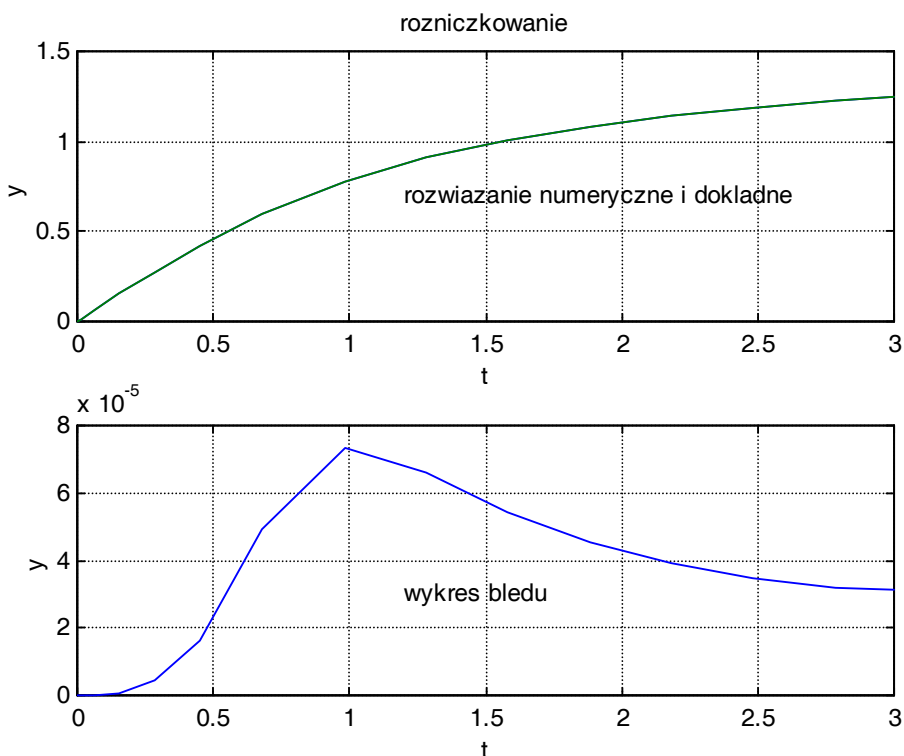
```
function dy=F(t,y)                                %dy=y'
dy=1./(1+t.^2)
```

```
%w nowym skrypcie zamieścić polecenia służące do wyznaczenia rozwiązania dokładnego,
% numerycznego i błędu różniczkowania oraz do narysowania wykresu
```

```
y=atan(t)                                           %y - rozwiązanie dokładne
[t,Y]=ode23('rozn',[0 3],[0])                      %Y - rozwiązanie numeryczne
bl=y-Y                                              %bl - błąd różniczkowania
```

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

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



Rozwiązania uzyskane w MATLAB-ie:

| czas | rozwiązanie numeryczne | rozwiązanie dokładne | błąd |
|---------------|---------------------------|-------------------------|---------------|
| t = | Y = | y = | 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 $\frac{d^2y}{dt^2} + 2\frac{dy}{dt} + y = 0$ 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:

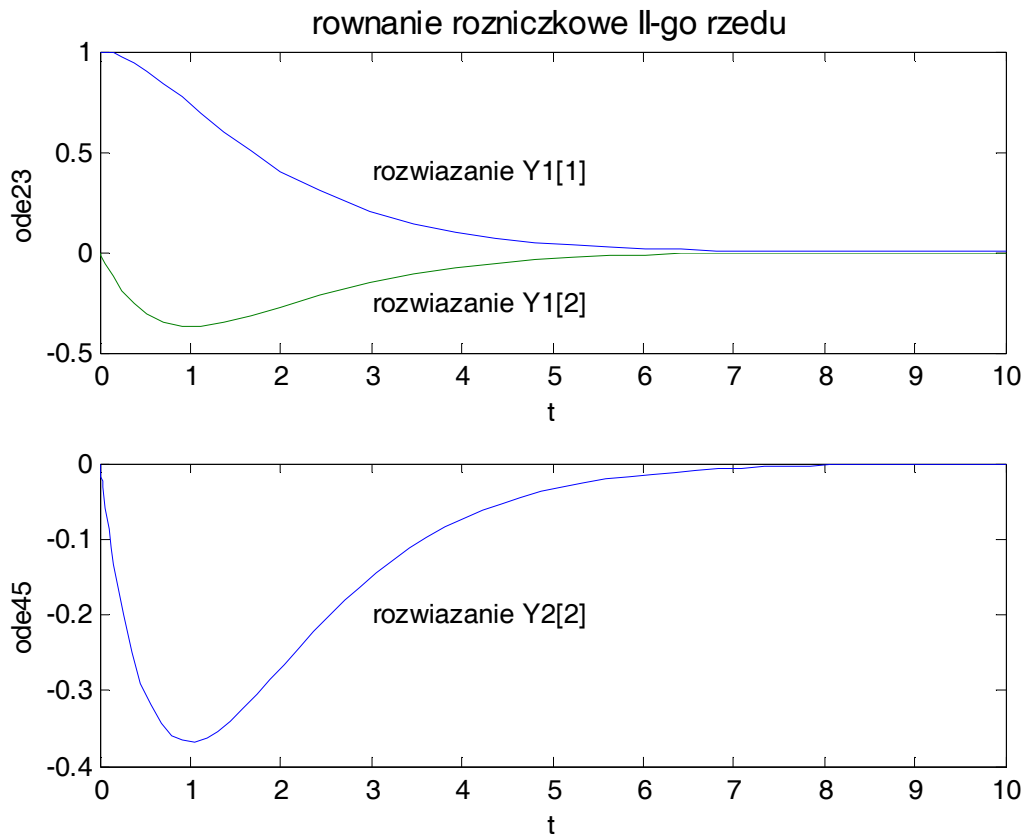
```
function d2y=F(t,y) %d2y=y''
d2y=[y(2); -y(1)-2*y(2)] %y(1)=y y(2)=y'
```

```
% w nowym skrypcie zamieścić polecenia służące do wyznaczenia rozwiązania numerycznego
% i narysowania wykresów

[t1,Y1]=ode23('rozn2',[0 10],[1 0]')           %Y1 - rozwiązanie numeryczne metodą ode23
[t2,Y2]=ode45('rozn2',[0 10],[1 0]')           %Y2 - rozwiązanie numeryczne metodą ode45

subplot(2,1,1)
plot(t1,Y1)                                     %Y1 - pierwsze i drugie rozwiązanie metodą ode23
xlabel('t')
title('rownanie rozniczkowe II-go rzędu')
ylabel('ode23')
text(3,0.4,'rozwiązanie Y1[1]')
text(3,-0.25,'rozwiązanie Y1[2]')

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



Stabilność metod Rungego-Kutty

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

Gdy te warunki nie są spełnione to po wykonaniu nawet niewielkiej ilości kroków rozwiązania przybliżone na ogół 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 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 h takim, że λh 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 .

Program RK-STAB kreśli na ekranie brzeg obszaru stabilności absolutnej jawnych dwuetapowych metod Rungego-Kutty rzędu 1 i 2 zdefiniowanych tabelą współczynników:

| | |
|----------|----------|
| | a |
| u | v |

Dla metod rzędu co najmniej pierwszego spełniona jest zależność:

$$u + v = 1$$

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:

$$y(n+1) = [1 + \lambda h + av(\lambda h)^2] y(n)$$

Zakładamy, że: λh jest liczbą zespoloną oraz wprowadzamy parametr \mathbf{P} :

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

wówczas równanie brzegu obszaru stabilności absolutnej będzie następujące:

$$|\mathbf{y}(\mathbf{n}+1)| = |\mathbf{y}(\mathbf{n})|$$

więc po uwzględnieniu wcześniejszych zależności otrzymujemy:

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

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

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

Przebieg ćwiczenia MET-NUM (program RK - STAB):

- 1.) dla wybranego przykładu przepisać tabele współczynników dla wszystkich metod rzędu 1 i 2 a następnie wyznaczyć dla każdej z nich wartość parametru **P** (*RK-ELEM – Elementarz metod*)
- 2.) przerysować kształt obszaru stabilności absolutnej (*RK-STAB*) dla powyższych wartości parametru **P**
- 3.) w przedziale $0 < \mathbf{P} < 0,5$ określić jak zachowuje się kształt obszaru stabilności absolutnej, gdy zmienia się wartość **P**; dla jakich wartości **P** obszar jest spójny a dla jakich otrzymuje się najdłuższy przedział stabilności absolutnej (zamieścić odpowiednie wykresy)
- 4.) w przypadku $\mathbf{P} > 0,5$ obejrzyć wykresy dla **P = 0,6; 1,5; 2; 5; 10; 100; 1000**, zaobserwować jaki wpływ na kształt i wielkość obszaru stabilności absolutnej oraz na długość przedziału stabilności absolutnej ma zwiększanie parametru **P**
- 5.) w całym przedziale $< 0, 1000 >$ określić jak zmieniają się wartości **Re_{min}**, **Re_{max}**, **Im_{min}**, **Im_{max}**, $\Delta \mathbf{Re} = \mathbf{Re}_{\max} - \mathbf{Re}_{\min}$ i $\Delta \mathbf{Im} = \mathbf{Im}_{\max} - \mathbf{Im}_{\min}$ (zamieścić wykresy $|\Delta \mathbf{Re}| = f(\mathbf{P})$ i $|\Delta \mathbf{Im}| = f(\mathbf{P})$)

[illegible]

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:

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

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 rozwiązanie 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:

$$Y_{n+1} = \begin{bmatrix} 1 & h \\ -\omega^2 h & 1 \end{bmatrix} Y_n = A Y_n \quad A - \text{macierz przejścia}$$

Po obliczeniu wyznacznika Δ macierzy A można znaleźć jej wartości własne λ :

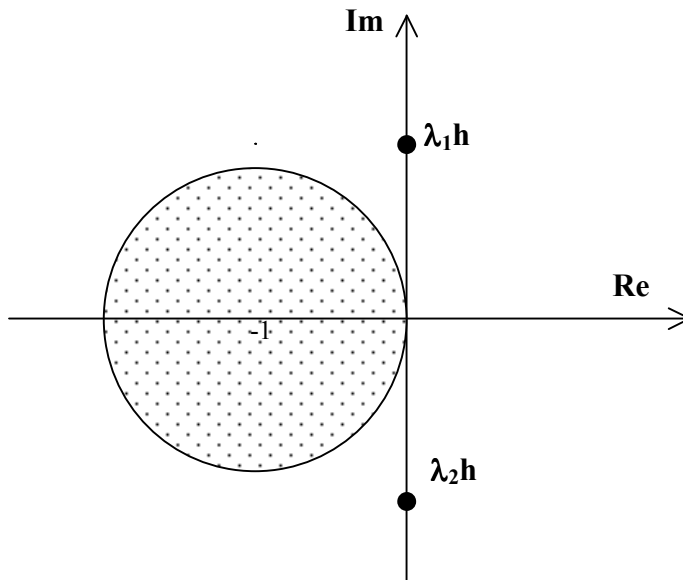
$$\Delta = 1 + \omega^2 h^2 = (1 + i \omega h)(1 - i \omega h) \quad \Rightarrow \quad \lambda = 1 \pm i \omega h$$

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:

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

wartości własne wyrażają się wzorem:

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

natomiast 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 skalarnie równanie testowe o postaci:

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

Rozwiązanie 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 zewnątrz 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 skalarne 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 .

Przebieg ćwiczenia MET-NUM (program RK-OKRĄG):

- 1.) przy ustalonej wartości ω (odpowiednio $\omega = 1; 2; 3...$) zmieniać wartość kroku h i zaobserwować jak zachowuje się ciąg rozwiązań równania różniczkowego okręgu rozwiązywanego metodą Eulera; odpowiednie wykresy zamieścić w sprawozdaniu
- 2.) dla tej samej wartości ω przeanalizować metodę Eulera-Cauchego
- 3.) w przypadku metody stabilnej dla równania różniczkowego okręgu tak dobrać wartość parametru P , aby uzyskać możliwie największy zakres zmian kroku h przy stałej wartości ω , a następnie dla tej wartości P określić graniczną wartość kroku h zapewniającą stabilność metody
- 4.) analogicznie jak w p.1 przeanalizować metodę niejawną Eulera i wzór trapezów; wykresy zamieścić w sprawozdaniu
- 5.) dla danej wartości ω rozwiązać w MATLAB-ie równanie różniczkowe okręgu; zamieścić w sprawozdaniu napisany program i wykres rozwiązania (rozwiązaniem jest okrąg)

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 rozwiązanie 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

Sterowanie długością kroku dokonywane jest poprzez:

- połowienie / podwajanie kroku
- mnożenie kroku przez współczynnik

Przedstawiając przybliżenie normy części głównej błędu lokalnego w postaci:

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

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:

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

wówczas nowy krok $H = 2h$ 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:

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

$$\gamma^k z < \varepsilon \text{ SafetyFac}$$

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

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

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

czyli:

$$h_{\text{Min}} \leq H \leq h_{\text{Max}}$$

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.

Przebieg ćwiczenia MET-NUM (program RK-ROZW):

1.) dla danego zagadnienia początkowego zastosować:

- ♦ metodę 3-go rzędu klasyczną (3,3)
 - towarzyszącą 4-go rzędu klasyczną (4,4)
 - krok stały
 - szacowanie błędu bezwzględnego
 - bez szacowania błędu
 - podwajanie / połowienie kroku
 - szacowanie błędu bezwzględnego
 - bez szacowania błędu
 - mnożenie kroku przez współczynnik
 - szacowanie błędu bezwzględnego
 - bez szacowania błędu
 - ekstrapolację Richardsona
 - krok stały
 - szacowanie błędu bezwzględnego
 - bez szacowania błędu
 - podwajanie / połowienie kroku
 - szacowanie błędu bezwzględnego
 - bez szacowania błędu
 - mnożenie kroku przez współczynnik
 - szacowanie błędu bezwzględnego
 - bez szacowania błędu

2.) dla powyższego zagadnienia początkowego zastosować:

- ♦ metodę 4-go rzędu klasyczną (4,4)
 - towarzyszącą Shanks'a (5,5)
 - ekstrapolację Richardsona

wraz ze sterowaniem długością kroku i szacowaniem błędu bezwzględnego jak w p.1

3.) wyniki zamieścić w tabeli:

| Liczba kroków (liczba kroków akceptowanych) | Liczba odwołań do funkcji | Błąd lokalny | t | Błąd globalny | Norma max błędu bezwzględnego | | % Udział kroków akceptowanych |
|---|---------------------------------|-----------------|---|------------------|----------------------------------|------------------------------------|-------------------------------------|
| | | | | | dla błędu globalnego | dla błędu w punkcie końcowym | |
| | | | | | | | |

4.) zaobserwować jaki wpływ na wartości błędów oraz na koszty obliczeń i sprawność metod ma:

- sterowanie długością kroku i szacowanie błędu bezwzględnego
- zwiększenie rzędu metody
- zastosowanie metody towarzyszącej lub ekstrapolacji Richardsona

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., Cegiela R. – MATLAB – obliczenia numeryczne i ich zastosowanie, Wydawnictwo Nakom, Poznań 1999
9. Dyka E., Markiewicz P., Sikora R. – Modelowanie w elektrotechnice z wykorzystaniem środowiska Matlab, Wydawnictwo Politechniki Łódzkiej, Łódź 2006