1. OPIS STANOWISKA

1.1. Stosowana aparatura

- komputer klasy PC z systemem operacyjnym Microsoft Windows (XP/Vista/7).

1.2. Oprogramowanie

- środowisko Matlab R2007b (Version 7.5.0.342), classroom license.

2. WSTĘP TEORETYCZNY

2.1. Wprowadzanie macierzy

Przy wprowadzaniu macierzy wszystkie jej elementy muszą być ujęte w nawiasy kwadratowe. Elementy oddziela się spacją lub przecinkiem. Średnik lub znak nowego wiersza **<Enter>** kończy wiersz macierzy i powoduje przejście do następnego. Liczba elementów w każdym wierszu musi być taka sama.

Wprowadzenie macierzy $A = \begin{bmatrix} 3 & 7 & 6 \\ 4 & 2 & 1 \end{bmatrix}$ będzie miało następującą postać:

```
>> A = [3 7 6; 4 2 1]
```

Można wprowadzać także wektory wierszowe i kolumnowe: $B = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}$ $C = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$

```
>> B = [1 2 3 4]
```

```
>> C = [1; 2; 3; 4]
```

Macierze mogą zawierać liczby zespolone: $D = \begin{bmatrix} 2+3i & 4-6i \\ -2-5i & 3i \end{bmatrix}$

```
>> D = [2+3i 4-6i; -2-5i 3i]
```

lub

```
>> D = [2 4; -2 0] + i*[3 -6; -5 3]
```

Jeśli kolejne elementy macierzy różnią się od siebie o określoną wartość to do utworzenia macierzy można wykorzystać dwukropek (**:**).

`min:max` - generuje wektor wierszowy zawierający liczby całkowite z przedziału <**min**,**max**> zwiększające się o **1**,

`min:krok:max` - generuje wektor wierszowy zawierający liczby od **min** do **max** o wartościach zmieniających się o krok,

```
>> B = 1:4

B =

    1  2  3  4
```

```
>> C = 5:3:15

C =

    5  8  11  14
```

```
>> A = [1:4; 1:0.5:2.5]

A =

    1.0000 2.0000 3.0000 4.0000

    1.0000 1.5000 2.0000 2.5000
```

W Matlabie zdefiniowano szereg funkcji do generowania macierzy specjalnych.

`eye(n)`	generuje macierz jednostkową o rozmiarze **n** $\times$ **n** (jedynki na głównej przekątnej, reszta elementów równa zeru)
`ones(n)`	generuje macierz o rozmiarze **n** $\times$ **n** o wszystkich elementach równych **1**
`zeros(n)`	generuje macierz o rozmiarze **n** $\times$ **n** o wszystkich elementach równych **0**
`rand(n)`	generuje macierz o rozmiarze **n** $\times$ **n** wypełnioną liczbami pseudolosowymi z przedziału **<0,1>**
`randn(n)`	generuje macierz o rozmiarze **n** $\times$ **n** wypełnioną liczbami pseudolosowymi o rozkładzie normalnym ze średnią równo **0** i wariancją równą **1**

```
>> eye(4)

ans =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

```
>> rand(3)

ans =

    0.4218    0.9595    0.8491
    0.9157    0.6557    0.9340
    0.7922    0.0357    0.6787
```

Powyższe funkcje generują macierze kwadratowe (**n** × **n**), dla macierzy prostokątnych należy podać dwa argumenty, np.

`>> ones(n,m)` **n** - liczba wierszy, **m** - liczba kolumn

2.2. Odwołania do elementów macierzy

Do elementu macierzy **A** znajdującego się w wierszu o indeksie **i** oraz kolumnie o indeksie **j** odwołujemy się poprzez **A(i,j)**. Elementem takim można posługiwać się jak każdą inną zmienną. Indeksy wierszy i kolumn rozpoczynają się od wartości **1**.

```
>> A = [3 7 6; 4 2 1]

A =

     3     7     6
     4     2     1

>> A(1,1)

ans =

     3

>> A(2,3)

ans =

     1
```

Do elementów macierzy można odwoływać się także przy użyciu jednego indeksu:

- jeśli **A** jest wektorem, to odwołanie **A(i)** oznacza odwołanie do i-tego elementu wektora,
- jeśli **A** jest macierzą dwuwymiarową, to odwołanie **A(i)** oznacza odwołanie do wektora kolumnowego uformowanego z kolejnych kolumn oryginalnej macierzy, umieszczonych jedna pod druga, np.

```
>> A = [1 2 3; 4 5 6; 7 8 9];
>> A(2,3)                         >> A(6)
ans =                             ans =
        6                                 8
```

Wykorzystując dwukropek można odwoływać się do wybranych fragmentów macierzy.

`A(i,:)`	**i**-ty wiersz macierzy **A**
`A(:,j)`	**j**-ta kolumna macierzy **A**
`A(:)`	cała macierz w postaci wektora kolumnowego
`A(:,:)`	cała macierz (dwuwymiarowa)
`A(i,j:k)`	elementy **i**-tego wiersza macierzy **A** o numerach od **j** do **k**
`A(i:j,k:l)`	elementy od **i**-tego do **j**-tego wiersza oraz od **k**-tej do **l**-tej kolumny
`A(X,i:j)`	wszystkie elementy w kolumnach od **i** do **j** i wierszach macierzy **A** o numerach określonych przez elementy wektora **X**

```
>> A = [1 2 3; 4 5 6; 7 8 9];
>> A(:)              >> A(2,:)            >> A(2:3,2:3)
ans =                ans =                ans =
       1                   4   5   6             5   6
       4                                         8   9
```

```
7
2              >> A(2,1:2)        >> A(:,[1 3])
5              ans =              ans =
8                    4   5              1   3
3                                       4   6
6                                       7   9
9
```

Można usunąć wybrane elementy macierzy przypisując im wartość w postaci macierzy pustej symbolizowanej przez puste nawiasy kwadratowe.

```
>> A = [1 2 3; 4 5 6; 7 8 9];
>> A(:,1)=[]
A =
     2 3
     5 6
     8 9
```

2.3. Macierze wielowymiarowe

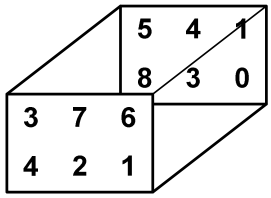
Matlab umożliwia definiowanie i wykonywanie operacji na macierzach wielowymiarowych. Odwołania do elementów takich macierzy wymagają podania odpowiedniej liczby indeksów, przy czym przyjmuje się, że:

- pierwszy indeks oznacza numer **wiersza** (wymiar 1),
- drugi indeks oznacza numer **kolumny** (wymiar 2),
- trzeci indeks oznacza numer **strony** (wymiar 3),
- kolejne indeksy oznaczają kolejne wymiary.

W przypadku macierzy trójwymiarowych definiuje się je stronami, np.

```
>> A(:,:,1) = [3 7 6; 4 2 1]        (strona 1)

>> A(:,:,2) = [5 4 1; 8 3 0]        (strona 2)
```



Przykładowe odwołania do powyższej macierzy:

```
>> A(2,2,1)            >> A(2,2,2)

ans =                  ans =

      2                      3
```

Macierze wielowymiarowe mogą być także tworzone za pomocą odpowiednich funkcji (`eye()`, `ones()`, `zeros()`, `rand()`, `randn()`), np.

```
>> A = rand(2,3,2)

A(:,:,1) =

           0.9501    0.6068    0.8913

           0.2311    0.4860    0.7621

A(:,:,2) =

           0.4565    0.8214    0.6154

           0.0185    0.4447    0.7919
```

2.4. Operacje na macierzach i wektorach

Operacje wykonywane na macierzach i wektorach można podzielić na dwie grupy:

- **operacje macierzowe** - określone regułami algebry liniowej,
- **operacje tablicowe** - wykonywane element po elemencie, operatory występujące w tych operacjach poprzedzone są kropką.

Podstawowe operacje i stosowane operatory zostały zestawione w poniższej tabeli.

Operacja	macierzowa	tablicowa	uwagi
dodawanie	`+`	`+`	
odejmowanie	`-`	`-`	
mnożenie	`*`	`.*`	
potęgowanie	`^`	`.^`	
dzielenie prawostronne	`/`	`./`	`A./B A(i,j)/B(i,j)`
dzielenie lewostronne	`\`	`.\`	`A.\B B(i,j)/A(i,j)`
transpozycja	`` ` ``	`` .` ``	

Poniżej przedstawiono sposób w jaki wykonywane są operacje macierzowe i tablicowe. Załóżmy, że mamy dwie macierze **A** i **B**.

$$A=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad B=\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

Dodawanie (macierzowe, tablicowe)

$$A+B=\begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} \\ a_{21}+b_{21} & a_{22}+b_{22} \end{bmatrix}$$

Odejmowanie (macierzowe, tablicowe)

$$A-B=\begin{bmatrix} a_{11}-b_{11} & a_{12}-b_{12} \\ a_{21}-b_{21} & a_{22}-b_{22} \end{bmatrix}$$

W przypadku dodawania i odejmowania operacje macierzowe i tablicowe wykonywane są w ten sam sposób.

Mnożenie (tablicowe)

$$A.*B=\begin{bmatrix} a_{11}\cdot b_{11} & a_{12}\cdot b_{12} \\ a_{21}\cdot b_{21} & a_{22}\cdot b_{22} \end{bmatrix}$$

Mnożenie (macierzowe)

$$A*B=\begin{bmatrix} a_{11}\cdot b_{11}+a_{12}\cdot b_{21} & a_{11}\cdot b_{12}+a_{12}\cdot b_{22} \\ a_{21}\cdot b_{11}+a_{22}\cdot b_{21} & a_{21}\cdot b_{12}+a_{22}\cdot b_{22} \end{bmatrix}$$

Mnożenie tablicowe jest przemienne, czyli: **`A.*B = B.*A`**. Mnożenie macierzowe nie jest przemienne, czyli: **`A*B ≠ B*A`.** W przypadku mnożenia macierzowego liczba wierszy macierzy **A** musi być równa liczbie kolumn macierzy **B**.

Potęgowanie (tablicowe)

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

Potęgowanie (macierzowe)

$$A\hat{}k = \underbrace{A * A * A * \ldots * A}_{k}$$

Dzielenie prawostronne (tablicowe)

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

Dzielenie lewostronne (tablicowe)

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

Dzielenie prawostronne (macierzowe)

$$A/B = A \cdot B^{-1}$$

Dzielenie lewostronne (macierzowe)

$$A \backslash B = A^{-1} \cdot B$$

Transpozycja macierzy zawierającej tylko elementy rzeczywiste daje takie same wyniki przy zastosowaniu obu operatorów - tablicowego i macierzowego. Jeśli macierz zawiera elementy zespolone to zastosowanie operatora tablicowego powoduje tylko zamianę wierszy macierzy z jej kolumnami, natomiast w przypadku operatora macierzowego otrzymana macierz zawiera elementy sprzężone z odpowiednimi elementami macierzy zespolonej.

```
>> A = [1+1i 2-2i; 3-3i 4+4i]

A =

   1.0000+ 1.0000i   2.0000- 2.0000i

   3.0000- 3.0000i   4.0000+ 4.0000i

>> A.'

ans =

   1.0000+ 1.0000i   3.0000- 3.0000i

   2.0000- 2.0000i   4.0000+ 4.0000i

>> A'

ans =

   1.0000- 1.0000i   3.0000+ 3.0000i

   2.0000+ 2.0000i   4.0000- 4.0000i
```

Jeśli jeden z argumentów przy operacjach jest skalarem, to zawsze wykonywane są operacje tablicowe, np.

`>> A+1` - do każdego elementu macierzy **A** zostanie dodana wartość **1**.

`>> A/5` - każdy element macierzy **A** zostanie podzielony przez **5**.

2.5. Funkcje przekształcające macierze i wyznaczające wielkości je charakteryzujące

Funkcje wyświetlające rozmiary macierzy:

`size(A)`	funkcja wyświetlająca liczbę wierszy i liczbę kolumn macierzy **A** w postaci dwuelementowego wektora wierszowego
`n = size(A,1)`	funkcja przypisująca zmiennej **n** liczbę wierszy macierzy **A**
`m = size(A,2)`	funkcja przypisująca zmiennej **m** liczbę kolumn macierzy **A**
`length(x)`	funkcja zwracająca długość wektora **x** lub dłuższy z wymiarów macierzy (jeśli **x** jest macierzą)

Funkcje wyświetlające wartości charakteryzujące macierze:

`rank(A)`	funkcja obliczająca rząd macierzy **A** (liczbę liniowo niezależnych wektorów tworzących wiersze lub kolumny macierzy)
`det(A)`	funkcja obliczająca wyznacznik macierzy kwadratowej **A**
`cond(A)`	funkcja obliczająca współczynnik uwarunkowania macierzy **A**, definiowany jako $\mathbf{cond(A)} = \|\|\mathbf{A}\|\| \cdot \|\|\mathbf{A}^{-1}\|\|$, stanowiący współczynnik z jakim błędy wejściowe przenoszą się na wyjście podczas operacji macierzowej; im większa jest wartość współczynnika uwarunkowania macierzy tym większa jest jej wrażliwość na błędy zaokrągleń podczas wykonywania operacji arytmetycznych
`trace(A)`	funkcja obliczająca ślad macierzy **A** (sumę elementów na przekątnej)

`E = eig(A)`	funkcja zwracająca wektor **E** zawierający wartości własne macierzy kwadratowej **A**
`[V,D] = eig(A)`	funkcja zwracająca: - macierz diagonalną **D** zawierającą na diagonali wartości własne macierzy kwadratowej **A** - macierz **V**, której kolumny są wektorami własnymi odpowiadającymi kolejnym wartościom własnym

```
>> A = [8 5 6; 3 4 7; 4 7 2];
>> eig(A)
ans =
       15.2888
        2.9210
       -4.2097
>> [V,D] = eig(A)
V =
    0.7293    0.8192   -0.1300
    0.4900   -0.5041   -0.6154
    0.4776   -0.2734    0.7774
D =
   15.2888         0         0
         0    2.9210         0
         0         0   -4.2097
```

Funkcje przekształcające macierze:

`inv(A)`	funkcja obliczająca macierz odwrotną do macierzy **A**, taki sam wynik uzyskamy podnosząc macierz **A** do potęgi **-1**: **`inv(A) = A^(-1)`**

```
>> A = [8 5 6; 3 4 7; 4 7 2];
>> inv(A)
ans =
    0.2181   -0.1702   -0.0585
   -0.1170    0.0426    0.2021
   -0.0266    0.1915   -0.0904
>> A = [7 6 4; 4 9 1; 6 7 3];
>> inv(A)
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 6.308085e-018.
ans =
  1.0e+015 *
    1.8476    0.9238   -2.7714
   -0.5543   -0.2771    0.8314
   -2.4019   -1.2010    3.6029
```

W powyższym przykładzie próbujemy obliczyć macierz odwrotną do macierzy osobliwej. Matlab ostrzega nas komunikatem: „Macierz jest prawie macierzą osobliwą lub jest źle wyskalowana. Wynik może być niepoprawny.” RCOND (reciprocal condition estimate) jest odwrotnością wskaźnika uwarunkowania macierzy.

`x = diag(A)`	funkcja tworząca wektor **x** z elementów znajdujących się na głównej przekątnej macierzy **A**

`A = diag(x)`	funkcja tworząca macierz przekątniową **A** z elementami wektora **x** na głównej przekątnej

```
>> x = [1:5];
>> A = diag(x)
A =
      1      0      0      0      0
      0      2      0      0      0
      0      0      3      0      0
      0      0      0      4      0
      0      0      0      0      5
```

`repmat(A,n,m)`	funkcja tworząca nową macierz przez powielenie podmacierzy **A** **m**-razy w poziomie i **n**-razy w pionie

```
>> A = [1 2;3 4];
>> repmat(A,2,3)
ans =
      1      2      1      2      1      2
      3      4      3      4      3      4
      1      2      1      2      1      2
      3      4      3      4      3      4
```

`reshape(A,n,m)`	funkcja tworząca macierz o **n**-wierszach i **m**-kolumnach z macierzy **A** (elementy nowej macierzy powstają z elementów macierzy **A** branych kolejno kolumnami), nowa macierz i macierz **A** powinny mieć tyle samo elementów

```
>> A = [1 5 9; 2 6 10; 3 7 11; 4 8 12]

A =

     1     5     9

     2     6    10

     3     7    11

     4     8    12

>> reshape(A,3,4)

ans =

     1     4     7    10

     2     5     8    11

     3     6     9    12
```

W Matlabie istnieje szereg funkcji wykonujących operacje na wektorach.

max(x)	funkcja zwracająca największy element wektora **x**
min(x)	funkcja zwracająca najmniejszy element wektora **x**
sum(x)	funkcja zwracająca sumę elementów wektora **x**
prod(x)	funkcja zwracająca iloczyn elementów wektora **x**
mean(x)	funkcja zwracająca średnią arytmetyczną elementów wektora **x**
norm	funkcja obliczająca normę wektora lub macierzy
dot(A,B)	funkcja obliczająca iloczyn skalarny wektorów **A** i **B**, wektory **A** i **B** powinny mieć taką samą długość
sort(V)	funkcja sortująca elementy wektora **V** w kolejności rosnącej
diff(V)	funkcja obliczająca różnice pomiędzy sąsiednimi elementami wektora **V**

`cumsum(V)`	funkcja obliczająca sumy skumulowane kolejnych elementów wektora **V**
`cumprod(V)`	funkcja obliczająca iloczyny skumulowane kolejnych elementów wektora **V**

Jeśli w powyższych funkcjach **x** jest macierzą dwuwymiarową, to funkcje te zwracają wyniki odnoszące się do poszczególnych jej **kolumn**.

```
>> V = [2 4 1 6 5];

>> mean(V)

ans =

       3.6000

>> sort(V)

ans =

       1     2     4     5     6

>> diff(V)

ans =

       2    -3     5    -1

>> cumsum(V)

ans =

       2     6     7    13    18

>> cumprod(V)

ans =

       2     8     8    48   240
```

2.6. Rozwiązywanie układów równań liniowych

Rozwiązywanie układów równań liniowych w postaci **A·x = b** (gdzie **A** - macierz współczynników, **x** - wektor niewiadomych, **b** - wektor wyrazów wolnych) jest jednym z podstawowych zagadnień algebry liniowej. Najprostsza metoda rozwiązania takiego układu polega na zastosowaniu macierzy odwrotnej lub dzielenia lewostronnego.

`x = inv(A)*b`	rozwiązuje układ **A·x = b** (**b** musi być wektorem kolumnowym)
`x = A\b`	rozwiązuje układ **A·x = b** (**b** musi być wektorem kolumnowym)

Rozwiążmy układ równań:

$$\begin{bmatrix} 5 & 4 & 2 \\ 1 & 2 & 0 \\ 3 & 0 & 4 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 13 \\ 5 \\ 7 \end{bmatrix}$$

```
>> A = [5 4 2; 1 2 0; 3 0 4];

>> b = [13; 5; 7];
```

```
>> x = inv(A)*b

x =

   -0.3333

    2.6667

    2.0000
```

```
>> x = A\b

x =

   -0.3333

    2.6667

    2.0000
```

2.7. Wielomiany

W Matlabie wielomiany reprezentowane są w postaci wektora wierszowego zawierającego współczynniki wielomianu. Współczynniki są umieszczone w kolejności malejących potęg (od współczynnika stojącego przy najwyższej potędze do współczynnika stojącego przy najniższej).

`Y = polyval(P,X)`	funkcja obliczająca wartość wielomianu **P** dla **X**; **P** jest wektorem, którego elementami są współczynniki wielomianu umieszczone w kolejności malejących potęg

Obliczamy wartość wielomianu ***P*(*x*)** dla **x = 3**:

$$P(x) = 15x^4 - 6x^3 + 4x - 3$$

```
>> P = [15 -6 0 4 -3];

>> y = polyval(P,3)

y =

        1062
```

Jeśli argument **X** funkcji **polyval** jest macierzą lub wektorem, to wartość wielomianu jest obliczana oddzielnie dla każdego elementu tej macierzy lub wektora.

`roots(P)`	funkcja obliczająca pierwiastki wielomianu **P**

Szukamy pierwiastków wielomianu ***P*(*x*)**:

$$P(x) = 15x^4 - 6x^3 + 4x - 3$$

```
>> P = [15 -6 0 4 -3];

>> roots(P)

ans =

  -0.7055

   0.2550 + 0.6412i

   0.2550 - 0.6412i

   0.5954
```

`poly(V)`	funkcja obliczająca współczynniki wielomianu na podstawie jego pierwiastków zawartych w wektorze **V**

```
>> V = [3 7 2 -3 4];
```

```
>> poly(V)

ans =

       1   -13    41    61  -450   504
```

Wielomian z powyższego przykładu ma postać:

$$P(x) = x^5 - 13x^4 + 41x^3 + 61x^2 - 450x + 504$$

Funkcja **poly** jest funkcją odwrotną do **roots**, a **roots** - funkcją odwrotną do **poly**:

```
>> P = [1 -13 41 61 -450 504];

>> roots(P)

ans =

        7.0000

       -3.0000

        4.0000

        3.0000

        2.0000
```

Jeśli argument funkcji **poly** jest macierzą kwadratową, to zwraca ona wektor zawierający współczynniki wielomianu charakterystycznego tej macierzy.

`polyder(P)`	funkcja wykonująca różniczkowanie wielomianu o współczynnikach przekazanych w wektorze **P**

Pochodna wielomianu:

$$P(x) = 15x^4 - 6x^3 + 4x - 3$$

wynosi:

$$P'(x) = 60x^3 - 18x^2 + 4$$

```
>> P = [15 -6 0 4 -3];
>> polyder(P)
ans =
        60   -18     0     4
```

Operacją odwrotną do różniczkowania wielomianu jest jego całkowanie wykonywane przez funkcję **polyint()**.

`polyint(P,C)`	funkcja wykonująca całkowanie wielomianu o współczynnikach przekazanych w wektorze **P** i stałej całkowania **C**

Całka wielomianu:

$$P(x) = 60x^3 - 18x^2 + 4$$

dla ***C* = -3** wynosi:

$$\int P(x)\,\mathrm{d}\,x = 15x^4 + 6x^3 + 4x - 3$$

```
>> P = [60 18 0 4];
>> polyint(P,-3)
ans =
        15     6     0     4    -3
```

3. PRZEBIEG ĆWICZENIA

Wykonaj podane poniżej zadania.

Zadanie 1

Wykorzystując dwukropek (**:**) lub odpowiednie funkcje wygeneruj poniższe macierze:

a) $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ -5 & -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 \end{bmatrix}$,

b) $\mathbf{B} = \begin{bmatrix} 1 & 3 & 5 & 7 & 9 & 11 & 13 \\ 0{,}5 & 1{,}5 & 2{,}5 & 3{,}5 & 4{,}5 & 5{,}5 & 6{,}5 \end{bmatrix}$,

c) $\mathbf{C} = \begin{bmatrix} \frac{-\pi}{2} & \frac{-\pi}{4} & 0 & \frac{\pi}{4} & \frac{\pi}{2} \\ -4 & -2 & 0 & 2 & 4 \end{bmatrix}$,

d) macierz **D** o rozmiarze **4×3** wypełnioną liczbami pseudolosowymi z przedziału **<0,1>**,

e) macierz jednostkową **E** o rozmiarze **4×4**.

Zadanie 2

Na podstawie macierzy **A** z Zadania 1 utwórz:

a) wektor **A1** zawierający drugi wiersz macierzy **A**,

b) macierz **A2** zawierającą kolumny od 3 do 7 macierzy **A**,

c) macierz **A3** zawierającą kolumny od 3 do 5 macierzy **A** z wierszy nr 2 i 3,

d) macierz **A4** zawierającą kolumny nr 1, 3, 5, 7, 9 z macierzy **A**.

Zadanie 3

Dane są macierze **A**, **B**, **C** i **D**. Rozwiąż poniższe równanie macierzowe.

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -1 & 2 \\ 3 & 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 4 & 1 \\ 2 & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \qquad \mathbf{C}(\mathbf{AX}-\mathbf{B}) = \mathbf{D}$$

Zadanie 4

Oblicz wartości poniższej funkcji dla elementów wektora **x**.

$$y = 2x\cos(1+x^2) \qquad \mathbf{x} = \begin{bmatrix} 0 & \frac{1}{2}\pi & \pi & \frac{3}{2}\pi & 2\pi \end{bmatrix}$$

Zadanie 5

Dane są dwie macierze:

$$\mathbf{A} = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 2 & 3 & 2 \\ 8 & 1 & 8 \\ 5 & 1 & 5 \end{bmatrix}$$

Stosując odpowiednie funkcje oblicz:

a) rzędy macierzy **A** i **B**,

b) wyznaczniki macierzy **A** i **B**,

c) wskaźniki uwarunkowań macierzy **A** i **B**,

d) wartości własne macierzy **A** i **B**,

e) ślady macierzy **A** i **B**,

f) macierze odwrotne macierzy **A** i **B**,

g) wartość najmniejszego elementu macierzy **A**,

h) sumę wszystkich elementów macierzy **A**,

i) sumę elementów na głównej przekątnej macierzy **A**.

Zadanie 6

Rozwiąż poniższy układ równań stosując:

a) macierz odwrotną,

b) dzielenie lewostronne.

$$\begin{cases} x_1 + 2x_2 + x_3 = 0 \\ \quad 4x_2 + 2x_3 = -3 \\ 6x_1 + x_2 + x_3 = 7{,}5 \end{cases}$$

Sprawdź, czy obie metody dały identyczne wyniki (odejmij otrzymane wyniki od siebie).

Zadanie 7

Oblicz pierwiastki oraz wartość wielomianu **f(x)** dla **x = 2**:

$$f(x) = 2x^6 - 3x^5 + 1{,}5x^3 - 4x^2 - 2$$

Zadanie 8

Pierwiastki pewnego wielomianu **f(x)** wynoszą:

$$x_1 = 1,\ \ x_2 = -1,\ \ x_3 = 2,\ \ x_4 = -2,\ \ x_5 = 3,\ \ x_6 = -3$$

Oblicz i podaj postać tego wielomianu.

Zadanie 9

Stosując funkcję **roots** oblicz pierwiastki wielomianu **f(x)**. Oceń dokładność wyznaczenia tych pierwiastków.

$$f(x) = (x-1)^6$$

4. LITERATURA

[1] Mrozek B., Mrozek Z.: MATLAB i Simulink. Poradnik użytkownika. Wydanie III. Helion, Gliwice, 2010.

[2] Stachurski M. Treichel W.: Matlab dla studentów. Ćwiczenia, zadania, rozwiązania. Witkom, Warszawa, 2009.

[3] Pratap R.: MATLAB 7 dla naukowców i inżynierów. Wydawnictwo Naukowe PWN, Warszawa, 2010.

[4] Brzóska J., Dorobczyński L.: Matlab: środowisko obliczeń naukowo-technicznych. „Mikom”, Wydawnictwo Naukowe PWN, Warszawa, 2008.

[5] Kamińska A., Pańczyk B.: Ćwiczenia z Matlab. Przykłady i zadania. Wydawnictwo MIKOM, Warszawa, 2002.

[6] Sobierajski M., Łabuzek M.: Programowanie w Matlabie dla elektryków. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2005.

[7] Dyka E., Markiewicz P., Sikora R.: Modelowanie w elektrotechnice z wykorzystaniem środowiska MATLAB. Wydawnictwa Politechniki Łódzkiej, Łódź, 2006.

[8] Czajka M.: MATLAB. Ćwiczenia. Helion, Gliwice, 2005.

5. ZAGADNIENIA NA ZALICZENIE

1. W jaki sposób w Matlabie wprowadza się macierze dwu- i wielowymiarowe?
2. Do czego w operacjach na macierzach może być wykorzystany dwukropek?
3. Wyjaśnij czym różnią się operacje macierzowe i tablicowe?
4. W jaki sposób w Matlabie reprezentowane są wielomiany?