Metody numeryczne i symulacja – Lab. 1

Magdalena Szymkowiak (magdalena.szymkowiak@put.poznan.pl)

1. Środowisko *Matlab/Octave* – przypomnienie
   (link do strony z instalacją *Matlaba*
   `https://ch.mathworks.com/campaigns/products/trials.html`
   – po wypełnieniu formularza można korzystać z *Matlaba* przez 30 dni)
   (instalacja *Octave* `https://www.gnu.org/software/octave`)

   (a) Oznaczenia

   - potęgi

     ```
     1e+2              v=5e-2            t=2/100000
     ans =             v =               t=
     100               0.0500            2.0000e-05
     ```

   - liczby zespolone
     `2+3i` lub `2+3j`

     ```
     a=2+sqrt(-9)      c=abs(a)          d =
     a =               c =               0.9828
     2.0000 +          3.6056
     3.0000i           d=angle(a)
     ```

     Zobacz `help i`

   - inne znaki i funkcje specjalne
     `NaN` – not a number
     `Inf` – nieskończoność
     `ans` – zmienna robocza
     `pi` – liczba π
     `load ('nazwa pliku')` – wczytuje zmienne z pliku
     `save('nazwa pliku','zmienna')` – zapisuje zmienne z przestrzeni roboczej

(b) Macierze

- Generowanie wektora

```
x=[0:0.1:1]
x =
Columns 1 through 7
0 0.1000 0.2000 0.3000 0.4000 0.5000 0.6000
Columns 8 through 11
0.7000 0.8000 0.9000 1.0000
y=[7:11]
y =
7 8 9 10 11
```

- Generowanie macierzy

  Przy ręcznym wpisywaniu elementów macierzy
  - poszczególne elementy wiersza macierzy oddziela się spacjami lub przecinkami
  - wiersze macierzy oddziela się średnikami

```
A=[1 2 3; 4 5 6; 7 8 9]
A =
1 2 3
4 5 6
7 8 9
```

```
B=[1, 2, 3; 4, 5, 6]
B =
1 2 3
4 5 6
```

Macierz można generować za pomocą istniejących macierzy.

```
A =
1 2 3
4 5 6
7 8 9
B =
1 2 3
4 5 6
```

```
C =
1 2 3 1 2 3
4 5 6 4 5 6
```

```
D=[A;B]
D=
1 2 3
4 5 6
7 8 9
1 2 3
4 5 6
```

```
G=repmat(B,2,1)
G =
1 2 3
4 5 6
1 2 3
4 5 6
```

```
C=[B B]
```

Uwaga 1. *Należy pamiętać o zgodności wymiarów macierzy!*

Można generować macierze jednostkowe, macierze wypełnione jedynkami, zerami lub z zadaną przekątną.

```
E=eye(3)
E =
1 0 0
0 1 0
0 0 1
```

```
Z =
0 0
0 0
0 0
0 0
Z=zeros(5,2)
```

```
J=ones(2,3)
J =
1 1 1
1 1 1

p=[2 3 4]
P=diag(p)
```

```
P =
2 0 0
0 3 0
0 0 4
```

Uwaga 2. *W macierzy dwuwymiarowej*
`macierz(wiersz, kolumna)`

Można generować macierze trójwymiarowe.

```
A =
1 2 3
4 5 6
7 8 9
```

```
E=eye(3)
E =
1 0 0
0 1 0
0 0 1
```

```
p=[2 3 4]
P=diag(p)
P =
2 0 0
0 3 0
0 0 4
```

```
T(:,:,1)=A;
T(:,:,2)=E;
T(:,:,3)=P;
T
T(2,3,2)
ans =
0
```

Uwaga 3. *W macierzy trójwymiarowej*
`macierz(wiersz, kolumna, warstwa)`

Uwaga 4. *Z pamięci Matlaba można usuwać*

- *poszczególne zmienne i funkcje – przy pomocy polecenia*
  `clear nazwa`
- *wszystkie zmienne i funkcje – przy pomocy polecenia*
  `clear all`
- *zawartość Command Window – przy pomocy polecenia*
  `clc`

- Odwołanie do elementów macierzy

```
A =[1 2 3; 4 5 6; 7 8 9]
A=
1 2 3
4 5 6
7 8 9

A(2,3)
ans=
6

A(:,2)
ans=
2
5
8
```

```
A(3,:)
ans=
7 8 9
A(2:3, 1:2)
ans=
4 5
7 8
A(1:2:3, :)
ans=
1 2 3
7 8 9
A(1:2:end, :)
ans=
1 2 3
7 8 9
```

- Informacje o macierzach

```
size(macierz)

B=[1, 2, 3; 4, 5, 6]
B =
1 2 3
4 5 6

= size(B)
liczba_wierszy=
2
liczba_kolumn=
3
[liczba_wierszy, liczba_kolumn]
```

Największy wymiar macierzy

```
length(macierz)
B=[1, 2, 3; 4, 5, 6]
B =
1 2 3
4 5 6
length(B)
ans=
3
```

- Suma elementów macierzy

```
sum(macierz)
sum(macierz,2)
sum(sum(macierz))
```

```
B=[1, 2, 3; 4, 5, 6]      sum(B,2)
B =                       ans=
1 2 3                     6
4 5 6                     15

sum(B)                    sum(sum(B))
ans=                      ans=
5 7 9                     21
```

- Niektóre operacje macierzowe

```
+  -   *   ^  /  \  '   inv  sum   det
```

```
G=[2 3;4 1]               inv(G)
G =                       ans =
2 3                       -0.1000 0.3000
4 1                       0.4000 -0.2000

det(G)                    G*inv(G)
ans =                     ans =
-10                       1.0000 -0.0000
                          0 1.0000
```

- Niektóre operacje blokowe

```
.*  .^  ./  .\
```

```
G=[2 3;4 1]      G^2             G.^2
G =              ans =           ans =
2 3              16 9            4 9
4 1              12 13           16 1
```

- Dzielenie
  Rozróżniamy dzielenie „dwustronne”
  - prawostronne (/)
  - lewostronne (\)

```
2/10
ans =
0.2000
```

```
2\10
ans =
5
```

Uwaga 5. *Dzielenie lewostronne* (\) *ma szczególne zastosowanie przy rozwiązywaniu układu równań.*

Rozwiąż układ równań liniowych:

$$\begin{cases} 2x_1 - 5x_2 = 9 \\ x_1 + 3x_2 = -1 \end{cases}$$

Układ ten można zapisać w postaci macierzowej $AX = B$
gdzie $A = \begin{bmatrix} 2 & -5 \\ 1 & 3 \end{bmatrix}$ $X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ $B = \begin{bmatrix} 9 \\ -1 \end{bmatrix}$
Wtedy $X = A^{-1}B$

Działania w *Matlabie*

```
A=[2 -5; 1 3];
B=[9; -1];
X=A^(-1)*B
```

```
X=inv(A)*B
X=A \ B
X =
```

```
2
-1
```

(c) Operatory

Relacyjne

- A<B
- A>B
- A<=B
- A>=B
- A==B
- A∼=B

Logiczne (dla macierzy zero-jedynkowych)

- A&B
- A|B
- xor(A,B)
- ∼ A

Zobacz `help relop`

```
A =[1 2 3; 4 5 6; 7 8 9]
```

```
A=
1 2 3
4 5 6
7 8 9
```

```
G=A>=5
G=
0 0 0
0 1 1
1 1 1
```

```
A(G)=-2
A=
1  2  3
4 -2 -2
-2 -2 -2
```

```
H=(A<5)&(A∼=2)
H=
1 0 1
1 0 0
0 0 0
```

```
K=(A<=2)|(A>=8)
K=
1 1 0
0 0 0
0 1 1
```

(d) m-pliki

Matlab rozgranicza dwa typy m-plików:

- skrypty
  - w momencie wywołania nie pobierają żadnych argumentów wejściowych oraz nie zwracają danych wyjściowych
  - stosuje się je aby uprościć lub zautomatyzować czynności wykonywane wielokrotnie
- funkcje
  - operują na określonych argumentach i po zakończeniu swego działania zwracają wynik
  - pozwalają na rozszerzenie możliwości *Matlaba* o te procedury i algorytmy, które nie zostały wcześniej zdefiniowane
  - występujące wewnątrz funkcji zmienne lokalne nie są widoczne poza nią

(e) Skrypty – przykłady

Skrypt zapisany jako m-plik `ZAJ_srgeomszcz.m`

```
szczp1=[1 2 3 4 5 6 7 8 9];
np1=length(szczp1);
srgeomp1=(prod(szczp1))^(1 / np1)
```

Skrypt zapisany jako m-plik `ZAJ_hitogram_szroz.m`

```
k=6;
h=0.1;
xd1=1.5;
ni=[7 11 25 33 19 5];
xdi=[xd1:h:xd1+(k-1)*h];
xgi=[xd1+h:h:xd1+k*h];
xi=(xdi+xgi)/2;
figure
bar(xi,ni,1,'m')
xlim([xdi(1)-0.1, xgi(length(xgi))+0.1])
xlabel('przedziały')
ylabel('liczności przedziałów')
title('histogram liczności szeregu rozdzielczego')
```

(f) Podstawowe instrukcje w *Matlabie*

- pętla FOR („dla”):
```
for zmienna_iterowana = macierz_wartości
ciąg_instrukcji
end
```
- pętla WHILE („dopóki”):
```
while wyrażenie_warunkowe
ciąg_instrukcji
end
```
- instrukcja warunkowa IF („jeżeli”):
```
if wyrażenie_warunkowe1
ciąg_instrukcji1
elseif wyrażenie_warunkowe2
ciąg_instrukcji2
elseif wyrażenie_warunkowe3
ciąg_instrukcji3
else
ciąg_instrukcji4
end
```

Uwaga 6. *Jeżeli polecenie zapisywane w jakimś oknie Matlaba jest zbyt długie, można je przełamać wykorzystując operator ...*

Uwaga 7. *Wykonywanie każdej funkcji w Matlabie można zatrzymać naciskając kombinacje klawiszy CTRL+C.*

2. Format liczby– może to być np. short, long, bank, rat

   np. exp(1)

   `format short` 2.7183
   fixed-decimal format with 4 digits after the decimal point

   `format long` 2.718281828459046
   fixed-decimal format with 15 digits after the decimal point

   `format bank` 2.72
   currency format with 2 digits after the decimal point

   `format rat` $\frac{1457}{536}$
   ratio of small integers

   `format shortE` $2.7183e+00$
   short scientific notation with 4 digits after the decimal point

   `format shortE` $2.718281828459046e+00$
   long scientific notation with 15 digits after the decimal point

3. Zapis liczby – system dwójkowy w arytmetyce zmiennoprzecinkowej (Floating-Point Numbers)

   Wyróżniamy dwa rodzaje liczb zmiennoprzecinkowych:

   32-bitowe (pojedynczej precyzji - ang. single precision)
   64-bitowe (podwójnej precyzji - ang. double precision)

   Matlab wykorzystuje podwójną precyzję

   $$x = \pm q \cdot 2^n$$

   $\pm$ – znak liczby

   n – cecha liczby (wykładnik całkowity)

   $q \in \left[\frac{1}{2}, 1\right)$ – mantysa (rozwinięcie dwójkowe tej liczby jest takie, że pierwsza cyfra po przecinku jest różna od zera)

   64 bity – (1 bit – znak) (11 bitów – cecha) (52 bity – mantysa)

   dokładność $\varepsilon = 2.2204e-16$

   największa liczba dodatnia `realmax` $= 1.7977e+308$

   najmniejsza liczba dodatnia `realmin`$=2.2251e-308$

   Liczby w systemie dwójkowym z mantysą o długości 52 bity

   – liczby o 16 cyfrach w systemie dziesiętnym

4. Elementy statystki

   dla próby n elementowej o wyrazach x_i (dla $i = 1, 2, \ldots, n$)

   średnia arytmetyczna

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

   średnia potęgowa rzędu 2 (wartość skuteczna)

$$\overline{x}_2 = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i)^2}$$

   wariancja

$$s^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i)^2 - (\overline{x})^2 = (\overline{x}_2)^2 - (\overline{x})^2$$

   odchylenie standardowe

$$s = \sqrt{s^2} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2}$$

5. Błędy

wartość rzeczywista x

wartość obliczeniow $\hat{x}$

błąd bezwględny

$$E_x = |x - \hat{x}|$$

błąd względny (gdy $x \neq 0$)

$$R_x = \frac{|x - \hat{x}|}{|x|}$$

dla próby n elementowej o wyrazach x_i (dla $i = 1, 2, \ldots, n$)

średni błąd bezwględny

$$\overline{E} = \frac{1}{n} \sum_{i=1}^{n} E_{x_i} = \frac{1}{n} \sum_{i=1}^{n} |x_i - \hat{x_i}|$$

średni błąd wględny

$$\overline{R} = \frac{1}{n} \sum_{i=1}^{n} R_{x_i} = \frac{1}{n} \sum_{i=1}^{n} \frac{|x_i - \hat{x_i}|}{|x_i|}$$

średni błąd kwadratowy

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \hat{x_i})^2}$$

średni błąd kwadratowy średniej arytmetycznej

$$\sigma_{\overline{x}} = \sqrt{\frac{1}{n(n-1)} \sum_{i=1}^{n} (x_i - \overline{x})^2}$$

ZADANIA

1. Utwórz skrypt w *Matlabie*, przy pomocy którego można zapisać w systemie dwójkowym i ósemkowym następujące liczby systemu dziesiętnego

   (a) 25

   (b) 1066

   (c) 125,625

   (d) 0,333

   (e) 0,1

2. Utwórz skrypt w *Matlabie*, przy pomocy którego możesz dokonać konwersji

   (a) $(101101110110)_2 \rightarrow (?)_8$

   (b) $((111101, 101)_2 \rightarrow (?)_{10}$

   (c) $(2716)_8 \rightarrow (?)_{10}$

3. Utwórz skrypt w *Matlabie*, który na podstawie plików:

   - arguments_of_sine.txt – sygnał wejściowy wymuszenia
   - sine.txt – odpowiedź w formie funkcji sinusoidalnej
   - sine_with_noise.txt - zaszumiony sygnał wyjściowy

   może obliczyć i wykreślić

   (a) wartości błędów bezwględnych między sygnałem zaszumionym i wzorcowym dla każdego sygnału wejściowego

   (b) wartości błędów względnych między sygnałem zaszumionym i wzorcowym dla każdego sygnału wejściowego

   (c) średni błąd bezwzględny między sygnałami zaszumionymi $\hat{x}_i$ i wzorcowymi x_i

   (d) średni błąd względny między sygnałami zaszumionymi $\hat{x}_i$ i wzorcowymi x_i

   (e) wartość skuteczną sygnałów sine.txt oraz sine_with_noise.txt

4. Dokonaj korekty sygnałów zaszumionych z uwzględnieniem średniego błędu bezwzględnego

$$\hat{y}_i = \begin{cases} \hat{x}_i - \overline{E}, & \text{gdy } \hat{x}_i > x_i \\ \hat{x}_i + \overline{E}, & \text{gdy } \hat{x}_i < x_i. \end{cases}$$

Utwórz skrypt w *Matlabie*, który na podstawie podanych plików oraz pliku po korekcie może obliczyć

(a) odchylenie standardowe sygnału zaszumionego oraz sygnalu po wprowadzonej korekcie

(b) średni błąd kwadratowy średniej arytmetycznej sygnału zaszumionego oraz sygnalu po wprowadzonej korekcie