

AGH

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Inżynierskie techniki obliczeniowe 2021/2022

Wykład nr 1

Dr inż. Przemysław Korohoda
E-mail: korohoda@agh.edu.pl
Tel.wewn.AGH: (012-617)-27-52
Pawilon C3 - p.506

Strona www:

home.agh.edu.pl/~korohoda/rok_2021_2022_lato/ITO_EL_1

UPeL: ITOEL2022

Plan wykładu

1. Przedstawienie prowadzących, zasady współpracy, zasady zaliczania, organizacja semestru; podział na grupy itd.
2. Treści przedmiotu.
3. Literatura pomocnicza.
4. Wybrane elementy pakietu Matlab.

Zasady zaliczania

**Kartkówki przed zajęciami lab. (lub po), 5-10 minut:
np. 1 pytanie z odpowiedniego wykładu i 1 z materiału z
wcześniejszych zajęć), obecności aktywne, na koniec
spotkanie o charakterze powtórkowym (tzw. kolokwium)
Może być test końcowy na wykładzie (?)**

ocena końcowa według skali uniwersalnej:

punkty (procenty) -> ocena	
50-59	3
60-69	3,5
70-79	4
80-89	4,5
90-100	5

Treści kształcenia

Badanie i rozumienie wybranych technik obliczeniowych: wielomiany, funkcje trygonometryczne i zespolone, rachunek macierzowy, równania różniczkowe, całkowanie numeryczne, elementy statystyki, wybrane operatory i transformaty.

Zastosowania techniczne (analiza, modelowanie): elementy teorii obwodów, układów elektronicznych oraz przetwarzania sygnałów.

Nauka /utrwalenie podstaw korzystania z pakietu Matlab.

Podstawowa koncepcja

Głównym celem jest przygotowanie do wykorzystania komputera do „szybkiego eksperymentowania obliczeniowego” ...

...tak, żeby mając nie do końca jasny wzór/algorytm/teorię/itp. można było wykorzystać narzędzie obliczeniowe, aby samodzielnie wyjaśnić sobie wszystkie ważne wątpliwości/niejasności.

Zatem nie jest to kolejny kurs programowania! Sposoby wykorzystania elementów programowania w takim „improwowanym eksperymencie obliczeniowym” mogą się zasadniczo różnić od technik stosowanych w typowym współczesnym programowaniu, które zazwyczaj ma inne cele. Dlatego warto na te różnice zwrócić uwagę.

Trzy główne składniki eksperymentu obliczeniowego:

1. Świadomy, tego co robi, użytkownik.
2. Problem do rozwiązania.
3. Narzędzie (komputer z oprogramowaniem).

Literatura pomocnicza (według daty wydania):

1. A.Papoulis: Obwody i układy. Wyd. Komunikacji i Łączności, Warszawa 1988.
2. W.H.Press, S.A.Teukolsky, W.T.Vetterling, B.P.Flannery: Numerical recipes in C. 2nd ed. Cambridge University Press, 1992.
3. B.Mrozek, Z.Mrozek: Matlab 5.x, Simulink 2.x. Wyd. PLJ, Warszawa 1998.
4. A.Biran, M.Breiner: Matlab 5 for engineers. Prentice Hall, 1999.
5. A.Kaminska, B.Panczyk: Matlab – przykłady i zadania. Wyd. MIKOM, Warszawa 2002.
6. M.Stachurski: Metody numeryczne w programie Matlab. Wyd. MIKOM, Warszawa 2003.
7. A.Zalewski, E.Cegiela: Matlab – obliczenia numeryczne i ich zastosowania. Wyd. NAKOM, Poznań 2003.
8. W.Regel: Obliczenia symboliczne i numeryczne w programie Matlab. Wyd. MIKOM, Warszawa 2004.
9. S.Osowski, A.Cichocki, K.Siwiek: Matlab w zastosowaniu do obliczeń obwodowych i przetwarzania sygnałów. Oficyna wydawnicza Politechniki Warszawskiej, Warszawa 2006.

itp.

MATLAB = MATrix LABoratory

Początki:

Końcówka lat 1970. - Cleve Moler, University of New Mexico, zauważył nieefektywność ciągłego powtarzania podobnych programów w FORTRANie – MATLAB: interaktywny, łatwe we/wy, operacje na wektorach i macierzach „w jednym ruchu”.

Od 1983 - Inżynier Jack Little rozpoczął projekt komercyjny: MathWorks;

1994 – MATLAB 4.2c, Windows 3.1, wymagany koprocessor;

MATLAB staje się stopniowo powszechny także w polskich uczelniach;

2000 – MATLAB 6.0, wykorzystanie JVM (Java Virtual Machine);

Darmowe alternatywne propozycje:

Scilab (podobny, ale są różnice),

GNU octave (wymóg 100% kompatybilności z MATLAB'em).

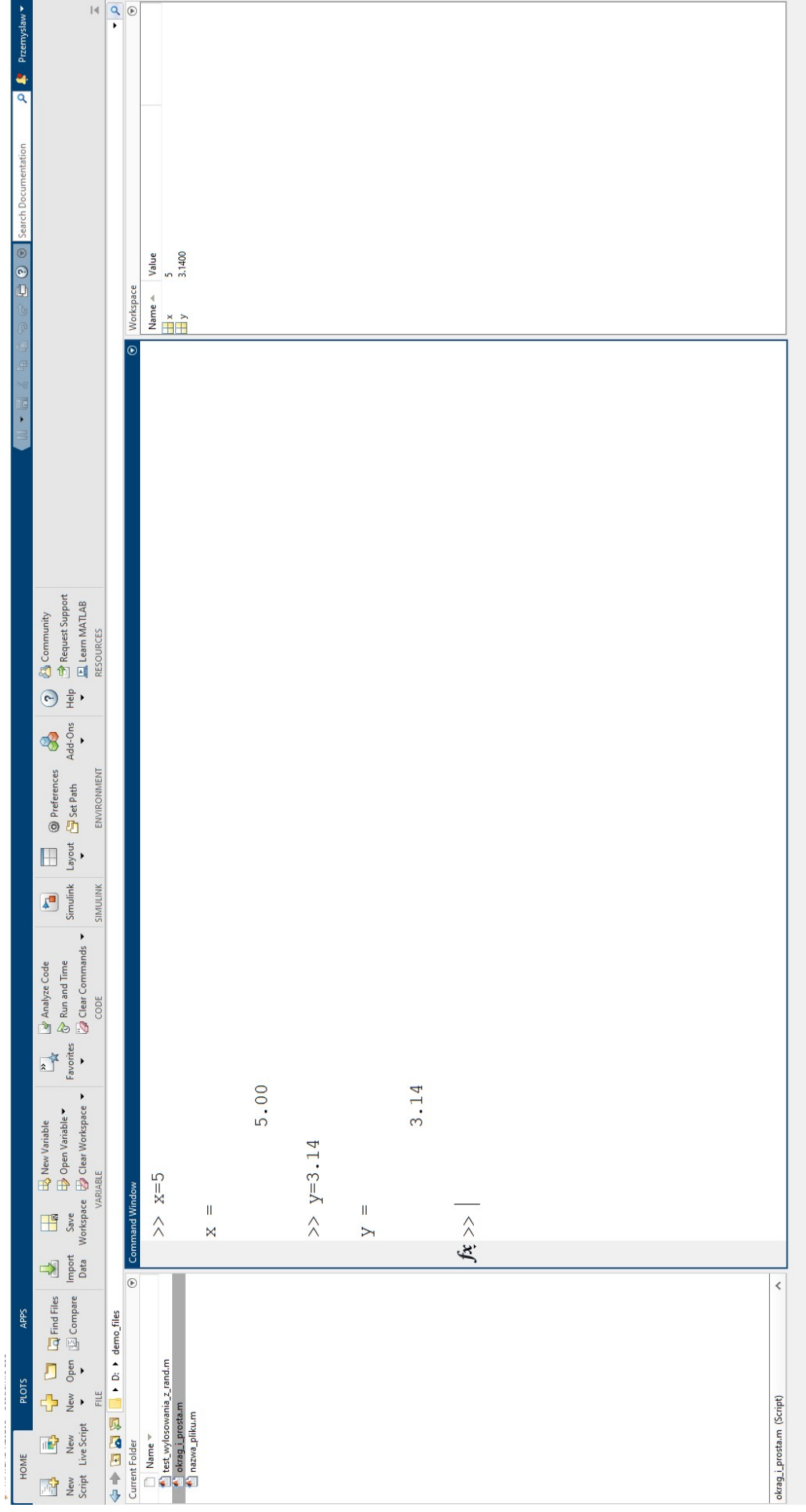
Komercyjne alternatywne propozycje : Mathematica, Maple, TK Solver, IDL.

Możliwości:

1. Importowania oprogramowania napisanego w C lub FORTRANie (mex-pliki).
2. Obliczeń symbolicznych (>>help sym).
3. Korzystania z interfejsu graficznego (GUI) i opracowywania symulacji w postaci schematów (Simulink).

itd..

Przykładowe multi-okno MATLABa



Wybrane elementy MATLABa

Praca w trybie komend realizowanych natychmiastowo,
w oknie o nazwie Command Window

```
>> a=5 [Enter]  
a =  
5  
>>
```

```
>> a=5, [Enter]  
a =  
5  
>>
```

```
>> a=5; [Enter]  
>>
```

```
>> a=5; [Enter]  
>> b=-4; [Enter]  
>> c=a*b [Enter]  
  
c =  
-20  
>>
```

>> to „znak zachęty”
a, b oraz c
to zmienne skalarne,

- * to operatory arytmetyczne

```
>> a=5; [Enter]  
>> b=-4; [Enter]  
>> c=a*b; [Enter]  
>>
```

**Uwaga – [Enter] to informacja o naciśnięciu klawisza
(nie jest to element syntaktyki MATLABa)**

Zmienne:

1. Nazwa zaczyna się od litery (dużej lub małej).
2. W nazwie nie wolno używać znaków specjalnych.
3. Rozróżniane są małe i duże litery.
4. Jest kilka zmiennych predefiniowanych, których nie warto nadpisywać.
5. Nie warto także nadpisywać nazw funkcji, których możemy potrzebować.

Przykładowe znaki specjalne:

1. Operatory, np.: **+** **-** ***** **/** **^** **** **|** **&** :
2. Separatory, np.: **,** **;** **.** *spacja*
3. Nawiasy: **()** **[]** **{}**
4. Cokolwiek, co może mieć specjalną interpretację w MATLABie ...

Zmienne predefiniowane:

pi – liczba π (ok. 3,14...)
i – jednostka urojona
j – jednostka urojona
eps – dostępna precyzja zapisu
ans – uniwersalna zmienna do odbierania wyników obliczeń

Wybrane elementy MATLABa

Przykłady poprawnych nazw zmiennych:

A
B_23
c21
tn
czas_obliczen_w_minutach

Niepoprawne:

1A
R%
T-start
Pierwsza zmienna

- bo zaczyna się od cyfry
- bo w nazwie użyto znaku komentarza (%)
- bo użyto operatora odejmowania
- bo zawiera spację

Nie zalecane:

cos
i

- bo zastąpiła wbudowaną funkcję cosinus
- bo zastąpiła wbudowaną jednostkę urojoną

Wybrane elementy MATLABa

W przypadku niepoprawnej nazwy MATLAB nas o tym poinformuje:

```
>> 1a=5 [Enter]
```

```
??? 1a=5
```

|

Error: Missing operator, comma, or semicolon.

Ale – jak widać - komentarz nie zawsze mówi, że nazwa jest niepoprawna...

W przypadku nazwy „nie zalecanej” sami się o tym zapewne kiedyś przekonamy, ale to może sporo kosztować (czasu, emocji ...)

Uwaga – od tego czasu będziemy pomijali informację o klawiszu [Enter]

Nazwy zmiennych

Wzór do zaimplementowania:

c=a+b

Preferowana wersja nazw zmiennych:

>> a=5; b=7; c=a+b

Przykładowa wersja „programistyczna”:

```
>> argument_pierwszy=5;  
>> argument_drugi=7;  
>> wynik_dodawania=argument_pierwszy+argument_drugi
```

Wybrane elementy MATLABa

Przykład wyliczenia:

MATLAB:

```
>>a=3; b=-2; c=2;  
>>delta=b^2-4*a*c,  
delta =  
    -20  
>>x1=(-b+delta^(1/2))/(2*a),  
x1 =  
    0.3333 + 0.7454i  
>>x2=(-b-delta^(1/2))/(2*a),  
x2 =  
    0.3333 - 0.7454i
```

Wzory:

$$\Delta = b^2 - 4 \cdot a \cdot c$$

nawiasy...

$$x_1 = \frac{-b + \sqrt{\Delta}}{2 \cdot a}$$

$$x_2 = \frac{-b - \sqrt{\Delta}}{2 \cdot a}$$

Kropka jako separator dziesiętny!
Liczby zespolone! Dla nich mamy wyjątek z zapisem, np...:
3j (tj. bez operatora mnożenia: 3*j).

Dodatkowe wyjaśnienia w trakcie wykładu...

Multi-okno MATLABa z oknem edytora



Wybrane elementy MATLABa

m-plik skryptowy

```
% nazwa m-pliku: przyklad_1.m
%
% To jest przykładowy plik
%
% P. Korohoda; 10/02/2015

clc; clear; close all;

a=3; b=-2; c=2;

delta = b^2-4*a*c,

x1 = (-b+delta^(0.5))/(2*a),
x2 = (-b-delta^(0.5))/(2*a),

% KONIEC
```

Kolejność
realizacji
po uruchomieniu

Elementy edycji tekstu:
a) spacje,
b) puste linie,
c) grupowanie poleceń.

Pamiętajmy: nasze pliki muszą zawsze „ładować” w odpowiednim katalogu!

Dodatkowe wyjaśnienia w trakcie wykładu...

Przykład uruchomienia programu z m-pliku:

```
>>przyklad_1  
delta =  
      -20  
x1 =  
      0.3333 + 0.7454i  
x2 =  
      0.3333 - 0.7454i  
>>
```

Nazwa m-pliku podlega analogicznym ograniczeniom jak nazwa zmiennej.

W razie wątpliwości można zawsze w oknie komend skorzystać z komendy „help”:

```
>>help proponowana_nazwa  
proponowana_nazwa.m not found
```

...czyli nie ma kolizji

Wektory i macierze

Wektor wierszowy (trójelementowy):

$$[1 \quad 2 \quad 3]$$

Wektor kolumnowy (trójelementowy):

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

**Macierz o wymiarach 2x3
(2 wiersze, 3 kolumny):**

$$\begin{bmatrix} 1 & 2 & 3 \\ -3 & -2 & -1 \end{bmatrix}$$

Indeksy

Zamiast x1 oraz x2 można użyć wektorowej zmiennej x:

```
% nazwa m-pliku: przyklad_1a.m
%
% To jest przykładowy plik
%
% P. Korohoda; 10/02/2015

clc; clear; close all;

a=3; b=-2; c=2;

    delta = b^2-4*a*c,

x(1) = (-b+delta^(0.5))/(2*a);
x(2) = (-b-delta^(0.5))/(2*a);
x,

% KONIEC
```

```
>>przyklad_1a
delta =
-20
x =
0.3333 + 0.7454i  0.3333 - 0.7454i
>>
```

```
x=x(:);
x,
```

**„Postawienie
do pionu”**

```
>>przyklad_1b
delta =
-20
x =
0.3333 + 0.7454i
0.3333 - 0.7454i
>>
```

Wartości indeksów zawsze zaczynają się od liczby jeden!

Wykres funkcji kosinus

```
% nazwa m-pliku: przyklad_2.m
%
% To jest kolejny przykladowy plik
%
% P. Korohoda; 02/03/2021

clc; clear; close all;

x = -pi : pi/99 : pi;
y=cos(x);

figure(1);
    plot(x,y,'b.-'); grid on;

% KONIEC
```

Nowe elementy:

- a) zapis ciągu: **start:krok:koniec**,
- b) użycie funkcji **cos**,
- c) użycie funkcji **figure**,
- d) użycie funkcji **plot**,
- e) użycie komendy **grid**.

Uwaga na elementy edycji, czyli: wcięcia, spacje, puste linie.

>>help plot

Wybrane elementy MATLABa

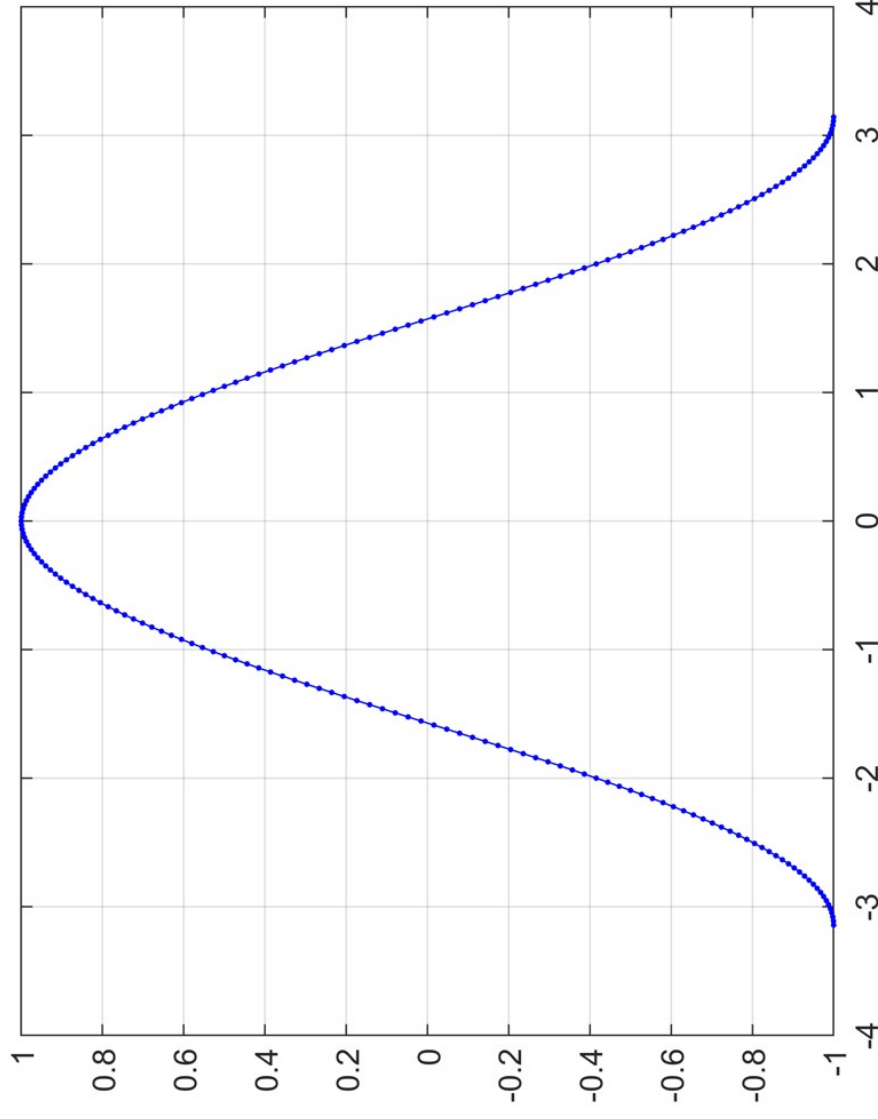
```
% nazwa m-pliku: przyklad_2.m
%
% To jest kolejny przykladowy plik
%
% P. Korohoda; 02/03/2021

clc; clear; close all;

x = -pi : pi/99 : pi;
y=cos(x);

figure(1);
    plot(x,y,'b.-'); grid on;

% KONIEC
```



Wybrane elementy MATLABa

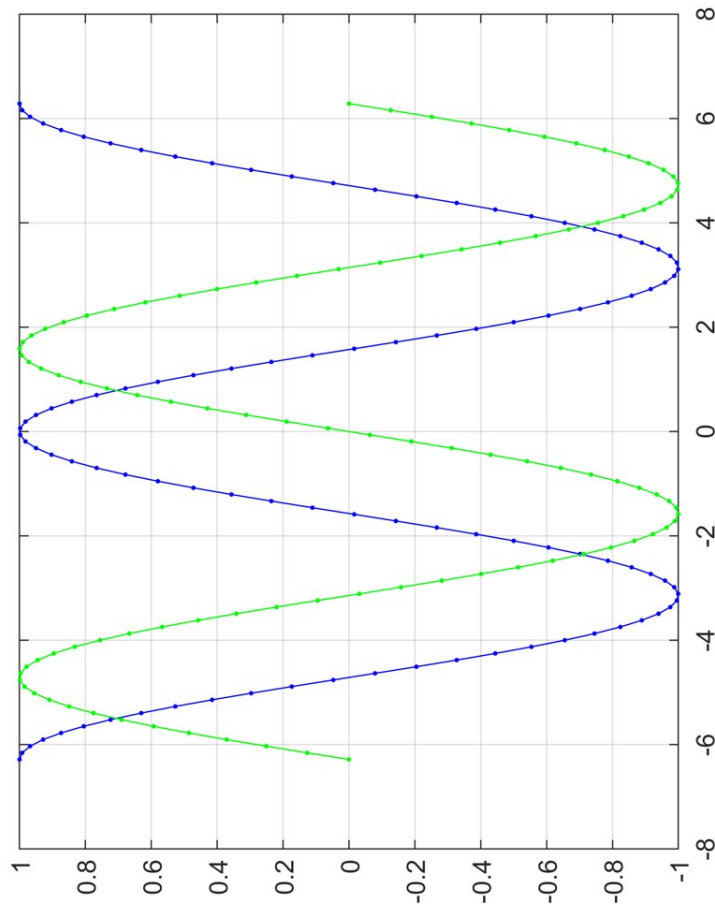
```
% nazwa m-pliku: przyklad_3.m
%
% To jest kolejny przykladowy plik
%
% P. Korohoda; 02/03/2021

clc; clear; close all;

x = -2*pi : 4*pi/99 : 2*pi;   % 100 elementow;
y1=cos(x);
y2=sin(x);

figure(1);
plot(x,y1,'b.-'); grid on; hold on;
plot(x,y2,'g.-');

% KONIEC
```



Wykres funkcji wielomianowej

```
% nazwa m-pliku: przyklad_4.m
%
% To jest kolejny przykladowy plik
%
% P. Korohoda; 02/03/2021

clc; clear; close all;

dt=1.1e-3; T=2;
x = -T : dt : T;
y = 2*x.^3 + 3*x.^2 - 2;

figure(1);
plot(x,y,'b.-'); grid on;

% KONIEC
```

Nowe elementy:

- a) notacja „naukowa” : **1.1e-3** ,
- b) zmienne jako parametry: **dt**, **T** ,
- c) operatory zastosowane macierzowo
(tu: dodawanie i odejmowanie),
- d) operator zastosowany skalarnie: **.****^**
(czyli „element po elemencie”).

Wykres funkcji wielomianowej

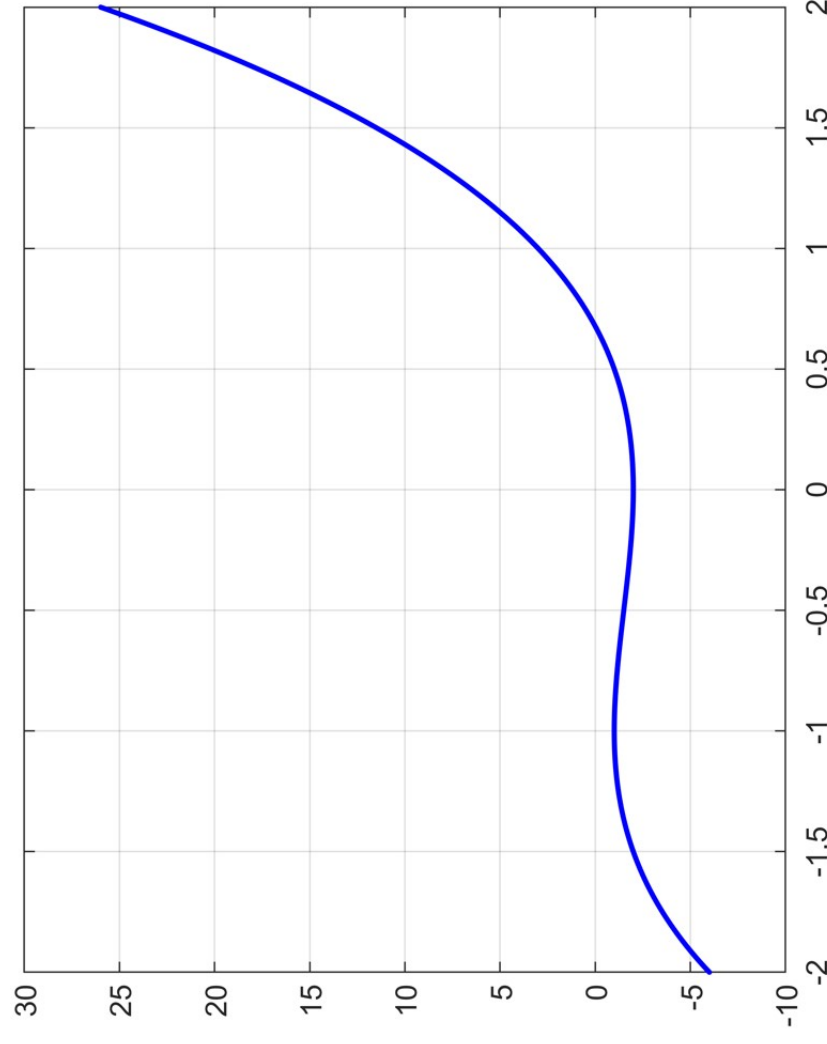
```
% nazwa m-pliku: przyklad_4.m
%
% To jest kolejny przykladowy plik
%
% P. Korohoda; 02/03/2021

clc; clear; close all;

dt=1.1e-3;  T=2;
x = -T : dt : T;
y = 2*x.^3 + 3*x.^2 - 2;

figure(1);
    plot(x,y,'b.-'); grid on;

% KONIEC
```



Wykres funkcji wielomianowej

```
% nazwa m-pliku: przyklad_5.m
%
% To jest kolejny przykładowy plik
%
% P. Korohoda; 02/03/2021

clc; clear; close all;

dt=1.1e-3;    T=2;    x = -T : dt : T;

a = [2,3,0,-2];
y = a(1)*x.^3 + a(2)*x.^2 + a(3)*x + a(4);

figure(1);
plot(x,y,'b.-'); grid on;

% KONIEC
```

Nowe elementy:

- a) wektor współczynników wielomianu,
- b) zapisywanie wiersza macierzy.

Wykres funkcji wielomianowej

```
% nazwa m-pliku: przyklad_6.m
%
% To jest kolejny przykladowy plik
%
% P. Korohoda; 02/03/2021

clc; clear; close all;

dt=1.1e-3;    T=2;    x = -T : dt : T;    Nx=length(x);

a = [2,3,0,-2];    Na=length(a);

y=zeros(1,Nx);
for n=1:Na
    y = y + a(n)*x.^(Na-n);
end

figure(1);
    plot(x,y,'b.-'); grid on;

% KONIEC
```

```
>>a=zeros(1,3)
```

```
a=
    0    0    0
```

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

```
A=
    0    0    0
    0    0    0
```

Wykres funkcji wielomianowej

```
% nazwa m-pliku: przyklad_7.m
%
% To jest kolejny przykladowy plik
%
% P. Korohoda; 02/03/2022

clc; clear; close all;

dt=1.1e-3;    T=2;    x = -T : dt : T;    Nx=length(x);

a = [2,3,0,-2]; % Na=length(a);
X=[x.^3 ; x.^2 ; x ; ones(1,Nx)]; % kolejne wiersze;
y=a*X;

figure(1);
plot(x,y,'b.-'); grid on;

% KONIEC
```

```
>>a=ones(1,3)
```

```
a=
    1    1    1
```

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

```
A=
    1    1    1
    1    1    1
```

Porównywanie dwóch wyników wektorowych

```
% nazwa m-pliku: przyklad_8.m
%
% To jest kolejny przykładowy plik
%
% P. Korohoda; 02/03/2022

clc; clear; close all;

a=-10:10; b=10:-1:-10;
c1=(a+b).^2;
c2=a.^2 + 2*a.*b + b.^2;

err=max(abs(c1-c2))

% KONIEC
```

Co otrzymamy w tych przypadkach?

```
.....
a=[1,-2,3,-5.7]; abs_a=abs(a)
.....
```

```
.....
a=[1,-2,3,-5.7]; max_a=max(a)
.....
```

```
.....
A=[1,-2;3,-5.7;2,1.1]; max_A=max(A)
.....
```

Generator liczb pseudolosowych

.....

N=10; rand(1,N)

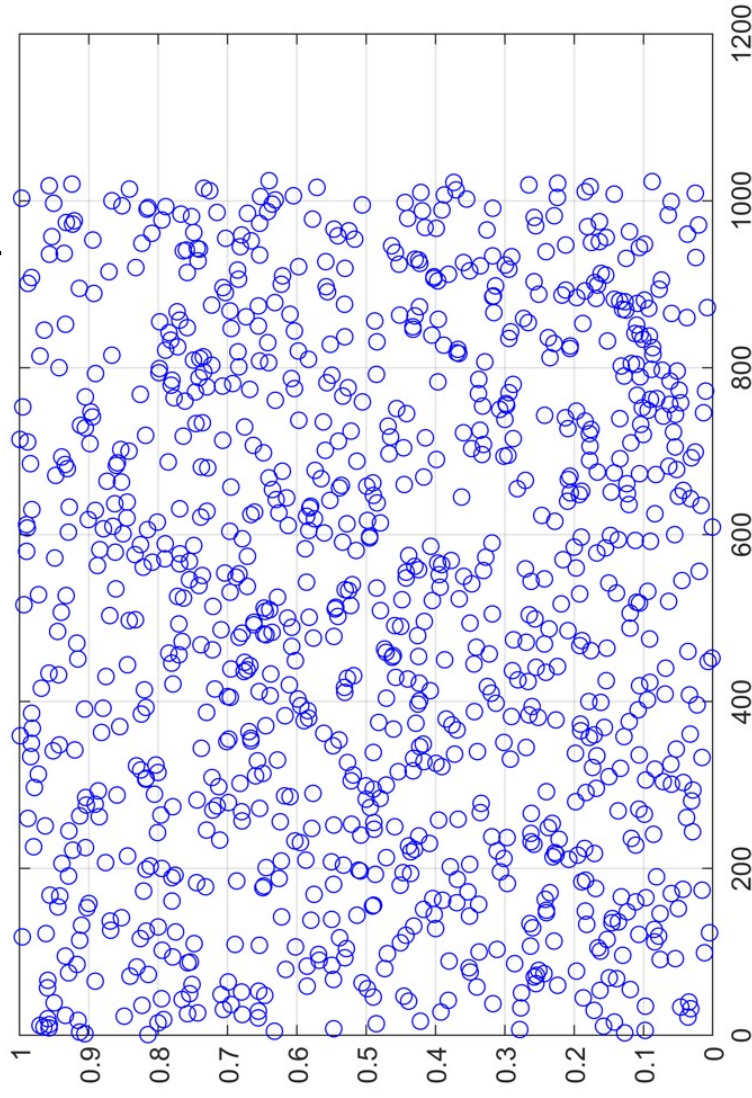
.....

.....

N=2¹⁰; rand(1,N);

.....

Graficzna forma studiowania efektów wywołania funkcji...



**Liczby z przedziału od 0 do 1,
o równomiernym rozkładzie gęstości prawdopodobieństwa.**

Porównywanie dwóch wyników wektorowych (**eksperyment: weryfikacja wzoru**)

```
% nazwa m-pliku: przyklad_9.m
%
% To jest kolejny przykladowy plik
%
% P. Korohoda; 02/03/2022

clc; clear; close all;

N=2^16;
tic;
for n=1:N
    a=rand; b=rand;
    c1(n)=(a+b)^2;
    c2(n)=a^2 + 2*a*b + b^2;
end
czas_w_sekundach=toc

err=max(abs(c1-c2))

% KONIEC
```

```
% nazwa m-pliku: przyklad_9a.m
%
% To jest kolejny przykladowy plik
%
% P. Korohoda; 02/03/2022

clc; clear; close all;

N=2^16;
a=rand(1,N); b=rand(1,N);
c1=(a+b).^2;
c2=a.^2 + 2*a.*b + b.^2;

err=max(abs(c1-c2))

% KONIEC
```

Macierzowo

Zadanie domowe

Na stronie:

<https://matlabacademy.mathworks.com/en>

Proszę zaliczyć kurs: **OnRamp i nadesłać uzyskane certyfikaty, umieszczając je w zakładce kursu na UPELu (typu zadanie): certyfikaty OnRamp**

Zapraszam do laboratorium ...