

INTRODUCERE ÎN SOFTWARE MATEMATIC

CURS #1

Liviu Marin

Facultatea de Matematică și Informatică, Universitatea din București, România

E-mails: marin.liviu@gmail.com; liviu.marin@fmi.unibuc.ro

Liviu Marin

Introducere în software matematic: Curs #1

Liviu Marin

Introducere în software matematic: Curs #1

PROBLEME ORGANIZATORICE

STRUCTURA: Curs (2 ore/săpt.) + Laborator (2 ore/săpt.)

Curs: Prof. Liviu Marin

Laborator:

- Gr. 101: Prof. Liviu Marin
- Gr. 102, 103, 106: Dr. Andrei Halanay
- Gr. 104, 105: Drd. Andreea-Paula Voinea-Marinescu

CONTACT:

- Sala 312
- marin.liviu@gmail.com
- <https://sites.google.com/site/marinliviu/home/teaching/>

Liviu Marin

Introducere în software matematic: Curs #1

Liviu Marin

Introducere în software matematic: Curs #1

CUPRINS - CURS #1

- 1 Probleme organizatorice
- 2 Bibliografie
- 3 Introducere în MATLAB®
 - Prezentare generală
 - Ferestrele MATLAB
 - Fereastra de comenzi (Command Window)
 - Operații aritmetice cu scalari
 - Formate de afișare a datelor
 - Funcții matematice elementare predefinite
 - Variabile scalare
 - Fișiere script

VERIFICARE: test laborator (120 minute)

$$\text{NOTA} = 0,60 \times (\text{NOTA TEST}) \\ + 0,40 \times (\text{NOTA LABORATOR}) + \text{BONUS}$$

OBSERVAȚII:

- (a) **Condiție necesară** pentru intrarea în verificare (test laborator):
 $\text{NOTA LABORATOR} \geq 5$
- (b) Neîndeplinirea condiției necesare (a): **RESTANȚĂ**
- (c) În cazul unei restanțe cauzate de neîndeplinirea condiției (a), trebuie refăcut laboratorul pentru a obține $\text{NOTA LABORATOR} \geq 5$
- (d) **RESTANȚĂ:** test laborator (120 minute)
 $\text{NOTA} = 0,60 \times (\text{NOTA TEST}) \\ + 0,40 \times (\text{NOTA LABORATOR (REFĂCUT)})$

CONȚINUTUL CURSULUI:

1. Introducere în MATLAB®
2. Tablouri: definire, apelare
3. Operații cu tablouri
4. Fișiere script și manipularea datelor
5. Reprezentări grafice bidimensionale
6. Programare în MATLAB®
7. Fișiere funcție și funcții MATLAB®
8. Polinoame, interpolare polinomială în MATLAB®
9. Reprezentări grafice tridimensionale
10. Calcul simbolic
11. Aplicații

BIBLIOGRAFIE



MATLAB® The Language of Technical Computing, Using MATLAB.

The MathWorks Inc., 2002.



Amos Gilat.

MATLAB® An Introduction with Applications, Fifth Edition.

John Wiley & Sons, London, UK, 2014.



Desmond J. Higham, Nicholas J. Higham.

MATLAB® Guide, Third Edition.

SIAM: Society for Industrial and Applied Mathematics, Philadelphia, USA, 2017.



(a)



(b)

Figure: (a) Cleve Moler și (b) Jack Little, fondatorii The MathWorks Inc., USA.

Designul bobinelor de inducție pentru rezonanță magnetică (RMN)

PROBLEMA: Cunoscându-se **densitatea de curent, J** , să se determine **inducția magnetică, B** , generată de J , i.e.

$$\begin{cases} \nabla \cdot \mathbf{B}(x, y, z) = 0, & (x, y, z) \in \mathbb{R}^3 \quad (\text{absența polilor magnetici}) \\ \nabla \times \mathbf{B}(x, y, z) = \mu_0 \mathbf{J}(x, y, z), & (x, y, z) \in \mathbb{R}^3 \quad (\text{legea lui Ampère}) \end{cases} \quad (1)$$

unde

$$\mathbf{J}(x, y, z) = \begin{cases} \mathbf{J}^{\text{coil}}(x, y, z), & (x, y, z) \in \Gamma_{\text{coil}} \\ 0, & (x, y, z) \in \mathbb{R}^3 \setminus \Gamma_{\text{coil}} \end{cases} \quad (2a)$$

$$\nabla \cdot \mathbf{J}^{\text{coil}}(x, y, z) = 0, \quad \mathbf{J}^{\text{coil}}(x, y, z) \cdot \mathbf{n}^{\text{coil}}(x, y, z) = 0, \quad (x, y, z) \in \Gamma_{\text{coil}} \quad (2b)$$

Notății:

$\mathbf{B} = (B_x, B_y, B_z)^T$ inducția magnetică; $\mathbf{J} = (J_x, J_y, J_z)^T$ densitatea de curent;

$\mu_0 = 4\pi \times 10^{-10}$ H/m permeabilitatea spațiului liber;

$\mathbf{J}^{\text{coil}} = (J_x^{\text{coil}}, J_y^{\text{coil}}, J_z^{\text{coil}})^T$ densitatea de curent de suprafață (pe bobină);

$\mathbf{n}^{\text{coil}} = (n_x^{\text{coil}}, n_y^{\text{coil}}, n_z^{\text{coil}})^T$ normala exterioră la suprafața bobinei.

PROBLEMA: Cunoscându-se o componentă a inducției magnetice, $\tilde{B}_x(x, y, z)$ sau $\tilde{B}_z(x, y, z)$ pentru $(x, y, z) \in \Omega \subset \mathbb{R}^3$, să se determine densitatea de curent, $\mathbf{J}^{\text{coil}}(x, y, z)$ pentru $(x, y, z) \in \Gamma_{\text{coil}}$, a.i.

$$\begin{cases} \nabla \cdot \mathbf{B}(x, y, z) = 0, & (x, y, z) \in \mathbb{R}^3 \\ \nabla \times \mathbf{B}(x, y, z) = \mu_0 \mathbf{J}(x, y, z), & (x, y, z) \in \mathbb{R}^3 \\ \mathbf{J}(x, y, z) = \begin{cases} \mathbf{J}^{\text{coil}}(x, y, z), & (x, y, z) \in \Gamma_{\text{coil}} \\ \mathbf{0}, & (x, y, z) \in \mathbb{R}^3 \setminus \Gamma_{\text{coil}} \end{cases} \\ \nabla \cdot \mathbf{J}^{\text{coil}}(x, y, z) = 0, \quad \mathbf{J}^{\text{coil}}(x, y, z) \cdot \mathbf{n}^{\text{coil}}(x, y, z) = 0, & (x, y, z) \in \Gamma_{\text{coil}} \\ B_x(x, y, z) = \tilde{B}_x(x, y, z) \quad (B_z(x, y, z) = \tilde{B}_z(x, y, z)), & (x, y, z) \in \Omega \end{cases} \quad (3)$$

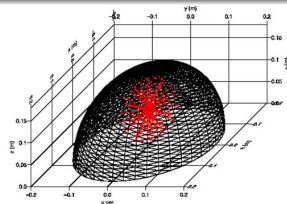


Figure: Bobină de inducție semisferică ($\Gamma_{\text{coil}} = \partial B(\mathbf{0}, R) \times \{z > 0\}$, $R = 0.175$ m), regiunea de interes $\Omega = B(\mathbf{x}^c, r)$, cu $\mathbf{x}^c = (0, 0, 0.081)^T$ m, $r = 0.065$ m, și gradientul de inducție $\tilde{B}_x(x, y, z) = G_x x$, $G_x = 1.0$ T/m.

Liviu Marin

Introducere în software matematic: Curs #1

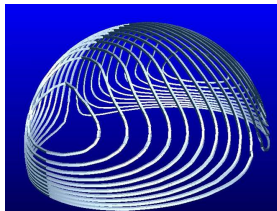


Figure: Bobină de inducție semisferică, cu gradientul inducției magnetice pe direcția Ox, i.e. $\tilde{B}_x(x, y, z) = G_x x$, $G_x = 1.0$ T/m.

Liviu Marin

Introducere în software matematic: Curs #1

Liviu Marin

Introducere în software matematic: Curs #1

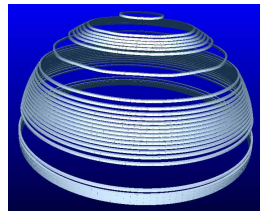


Figure: Bobină de inducție semisferică, cu gradientul inducției magnetice pe direcția Oz, i.e. $\tilde{B}_z(x, y, z) = G_z z$, $G_z = 1.0$ T/m.

Liviu Marin

Introducere în software matematic: Curs #1

MATLAB®

- Este o **platformă de programare performantă** pentru calcul științific.
- Este atât un **limbaj de programare**, cât și un **sistem de dezvoltare**.
- Integrează calculul, vizualizarea și programarea într-un **mediu ușor de utilizat**, problemele și soluțiile lor fiind exprimate într-un limbaj matematic accesibil.
- Sistem interactiv al cărui element de bază este **tabloul (array)**, fără a fi necesară dimensionalizarea sa.
- Numele provine de la **MATRIX LABORATORY** – firma producătoare: **The MathWorks Inc., USA**.
- A evoluat atât în **mediul universitar** (pachet standard pentru cursurile introductive și avansate de matematică, inginerie și științe), cât și în **industrie** (cercetare-dezvoltare).

MATLAB®

- **Domenii de utilizare:**
 - Matematică și calcul științific
 - Dezvoltare de algoritmi
 - Modelare, simulare și testare de prototipuri
 - Analiza și vizualizarea datelor
 - Grafică în inginerie și științele aplicate
 - Dezvoltare de aplicații, inclusiv **GUI (Graphic User Interface)**
- **Sistemul MATLAB®** este format din cinci părți principale:
 - Limbajul MATLAB®
 - Mediul de lucru MATLAB®
 - Manipularea graficelor
 - Biblioteca de funcții matematice MATLAB®
 - Interfața programului de aplicații, **API (Application Program Interface)** a MATLAB®

Liviu Marin

Introducere în software matematic: Curs #1

Liviu Marin

Introducere în software matematic: Curs #1

Sistemul MATLAB®

- (1) **Limbajul MATLAB®** este un limbaj performant de tip matrice/tablou cu instrucțiuni de control al salturilor, funcții, structuri de date, intrări/ieșiri și proprietăți de programare pe obiecte.

Facilitățile de programare sunt organizate în șase directoare:

- ops – Operators and special characters
- Lang – Programming language construct
- Strfun – Character strings
- Iofun – File input/output
- Timefun – Times and dates
- Datatypes – Data types structures

Sistemul MATLAB®

- (2) **Mediul de lucru MATLAB®** reprezintă un set de facilități care permit manevrarea variabilelor în spațiul de lucru, importul și exportul de date, dezvoltarea, manipularea și depanarea fișierelor MATLAB® (*.m) și a aplicațiilor MATLAB®.

Facilitățile acestea sunt organizate în directorul:

- **general** – General purpose commands

Liviu Marin

Introducere în software matematic: Curs #1

Liviu Marin

Introducere în software matematic: Curs #1

Sistemul MATLAB®

- (3) **Manipularea graficelor** reprezintă sistemul grafic al MATLAB®. Cuprinde comenzi performante pentru vizualizarea datelor bidimensionale și tridimensionale, procesarea imaginilor, animații, prezentări de grafice. Permite, de asemenea, utilizarea unor comenzi mai puțin performante pentru crearea unor interfețe GUI.

Funcțiile grafice sunt organizate în cinci directoare:

- graph2d – Two-dimensional graphs
- Graph3d – Three-dimensional graphs
- Specgraph – Specialized graphs
- Graphics – Handle Graphics
- Uitools – Graphical user interface tools

Sistemul MATLAB®

- (4) **Biblioteca de funcții matematice MATLAB®** reprezintă o colecție complexă de algoritmi de calcul pornind de la funcții elementare (sin, cos etc.) până la funcții sofisticate (inversare de matrice, valori proprii, funcții Bessel etc.).

Funcțiile matematice sunt organizate în opt directoare:

- elmat – Elementary matrices and matrix manipulation
- Elfun – Elementary math functions
- Specfun – Specialized math functions
- Matfun – Matrix functions – numerical linear algebra
- Datafun – Data analysis and Fourier transforms
- Polyfun – Interpolation and polynomials
- Funfun – Function functions and ODE solvers
- Sparfun – Sparse matrices

Sistemul MATLAB®

- (5) **Interfața programului de aplicații (API) a MATLAB®** este o bibliotecă ce permite scrierea de programe în C, C++, FORTRAN, Java sau Python, care interacționează cu MATLAB®.

Observație: Informații detaliate despre directoarele asociate părților principale ale sistemului MATLAB® se pot obține folosind comanda **help** în fereastra de comenzi (Command Window), e.g.

```
>> help Elfun
```

Fereastra MATLAB	Scop
Fereastra de comenzi (Command Window) • COMMAND	Fereastra principală Se introduc variabile Se evaluează expresii Se rulează programe
Fereastra de editare a figurilor (Figure Window) • FIGURE	Conține outputul comenzilor grafice
Fereastra de editare (Editor Window) • EDITOR	Se creează fișierele script și funcțiile
Fereastra de ajutor (Help Window) • HELP	Conține documentația MATLAB®
Fereastra istoric al comenzilor (Command History Window)	Stochează comenzile introduse în fereastra principală
Fereastra de lucru (Workspace Window)	Furnizează date și/sau informații despre variabilele folosite
Fereastra folderului actual (Current Folder Window)	Arată fișierele din folderul actual

Probleme organizatorice
Bibliografie
Introducere în MATLAB®

Prezentare generală
Ferestrele MATLAB
Ferestra de comenzi (Command Window)
Operații aritmetice cu scalari
Formate de afișare a datelor
Funcții matematice elementare predefinite
Variabile scalare
Fișiere script

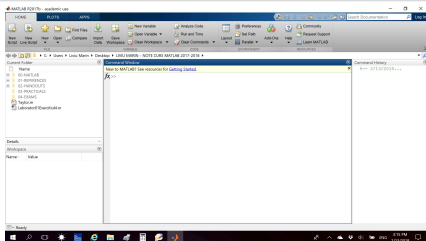


Figure: Vizualizare implicită a desktop-ului MATLAB® • MATLAB

Liviu Marin Introducere în software matematic: Curs #1

Probleme organizatorice
Bibliografie
Introducere în MATLAB®

Prezentare generală
Ferestrele MATLAB
Ferestra de comenzi (Command Window)
Operații aritmetice cu scalari
Formate de afișare a datelor
Funcții matematice elementare predefinite
Variabile scalare
Fișiere script

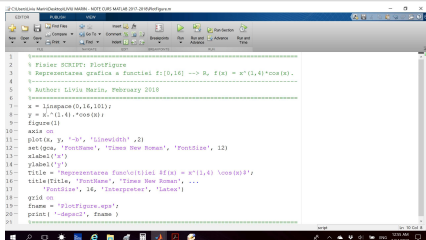


Figure: Ferestră de editare (Editor Window) • MATLAB

Liviu Marin Introducere în software matematic: Curs #1

Probleme organizatorice
Bibliografie
Introducere în MATLAB®

Prezentare generală
Ferestrele MATLAB
Ferestra de comenzi (Command Window)
Operații aritmetice cu scalari
Formate de afișare a datelor
Funcții matematice elementare predefinite
Variabile scalare
Fișiere script

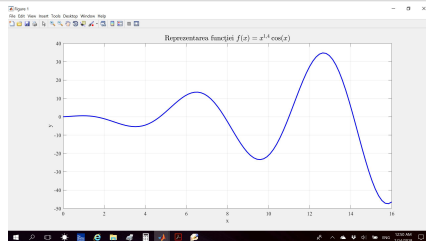


Figure: Ferestră de editare a figurilor (Figure Window) • MATLAB

Liviu Marin Introducere în software matematic: Curs #1

Probleme organizatorice
Bibliografie
Introducere în MATLAB®

Prezentare generală
Ferestrele MATLAB
Ferestra de comenzi (Command Window)
Operații aritmetice cu scalari
Formate de afișare a datelor
Funcții matematice elementare predefinite
Variabile scalare
Fișiere script

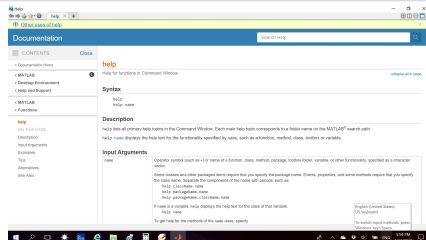


Figure: Ferestră de ajutor (Help Window) • MATLAB

Liviu Marin Introducere în software matematic: Curs #1

Fereastra de comenzi (Command Window)

- Comenzile se scriu în dreptul cursorului (\gg).
- După scrierea comenzii și apăsarea tastei **Enter**, comanda respectivă este executată. Doar ultima comandă scrisă este executată în fereastra de comenzi, comenzile anterioare rămânând neschimbate.
- Nu se poate merge spre capătul din stânga al unei comenzi deja afișate pentru a face corecturi și a reexecuta.
- Mai multe comenzi pot fi scrise în aceeași linie dacă acestea sunt despărțite de câte o virgulă. După apăsarea tastei **Enter**, comenzile sunt executate în ordine, de la stânga la dreapta.
- O comandă scrisă anterior în fereastra de comenzi poate fi reapelată în dreptul cursorului cu ajutorul tastei **↑**.

Fereastra de comenzi (Command Window)

- O comandă (foarte) lungă poate fi continuată pe linia următoare prin scrierea a trei puncte de suspensie (...) și apăsarea tastei **Enter**.
- Dacă semnul punct și virgulă (;) este adăugat la capătul unei comenzi, rezultatul acesteia nu este afișat.
- Dacă mai multe comenzi sunt scrise în aceeași linie, despărțite de semnul punct și virgulă (;), atunci rezultatul niciuneia dintre acele comenzi nu va fi afișat.
- Simbolul procent (%) scris la începutul unei linii transformă acea linie într-o linie de comentariu.
- Simbolul procent (%) urmat de text (comentariu) poate fi scris după o comandă, în aceeași linie, fără a afecta comanda respectivă.
- Prin scrierea comenzii **clc** și apăsarea tastei **Enter**, se șterg toate comenzile din fereastra de comenzi.

Operații aritmetice cu scalari

- Scalarii** (i.e. numerele reale sau complexe) pot fi folosiți în calcule aritmetice directe (ca un calculator de birou) sau pot fi atribuiți unor variabile, care pot fi utilizate în calcule ulterioare.
- Ordinea** operațiilor aritmetice în MATLAB® este cea uzuală: (1) ridicarea la putere; (2) înmulțirea și împărțirile la stânga și la dreapta; (3) adunarea și scăderea.

Simbol	Operație aritmetică	Exemplu
+	Adunare	$\gg 4 + 2$
-	Scădere	$\gg 4 - 2$
*	Înmulțire	$\gg 4 * 2$
/	Împărțire la dreapta	$\gg 4/2$
\	Împărțire la stânga	$\gg 4 \backslash 2$
^	Ridicare la putere	$\gg 4^2$

Table: Operații aritmetice

Formate de afișare a datelor

- Afișarea implicită a valorilor numerice în MATLAB se face **în virgulă fixă cu 4 zecimale**, e.g. valoarea implicită afișată lui π este 3.1416.
- Acesta este formatul implicit al valorilor numerice în MATLAB și corespunde valorii implicite a comenzii MATLAB **format**.
- Formatul de afișare a valorilor numerice în MATLAB se poate schimba prin comanda **format**.

Comandă	Descriere	Exemplu 290/7
format short	Virgulă fixă cu 4 zecimale din intervalul [0.001, 1000]. Altfel, format short e	41.4286
format long	Virgulă fixă cu 15 zecimale din intervalul [0.001, 100]. Altfel, format long e	41.428571428571431
format short e	Virgulă mobilă cu 5 cifre	4.1429e + 01
format long e	Virgulă mobilă cu 15 cifre	4.142857142857143e + 01
format short g	Cea mai bună reprezentare (v.m. sau v.f.) cu 5 cifre	41.429
format long g	Cea mai bună reprezentare (v.m. sau v.f.) cu 15 cifre	41.4285714285714
format bank	Virgulă fixă cu 2 zecimale	41.43

Table: Formate de afișare a datelor

Funcții matematice elementare predefinite

- Pe lângă operațiile aritmetice, expresiile din MATLAB pot conține și funcții.
- MATLAB are o vastă bibliotecă de funcții predefinite.
- O funcție are un **nume** și un **argument** conținut între paranteze.
- Când o funcție este utilizată, atunci **argumentul** său poate fi:
 - o un număr;
 - o o variabilă căreia i-a fost alocată o valoare numerică;
 - o o expresie formată din numere și/sau variabile.
- Funcțiile pot fi conținute în:
 - o argumentul/argumentele unei funcții;
 - o expresii.

Funcție	Descriere	Exemplu
sqrt(x)	\sqrt{x}	<pre>>> sqrt(64) ans = 8</pre>
nthroot(x,n)	$\sqrt[n]{x}$	<pre>>> nthroot(64,3) ans = 4</pre>
exp(x)	e^x	<pre>>> exp(2) ans = 7.3891</pre>
abs(x)	$ x $	<pre>>> abs(-3) ans = 3</pre>

Table: Funcții matematice elementare

Funcție	Descriere	Exemplu
log(x)	$\ln x$	<pre>>> log(4) ans = 1.3863</pre>
log10(x)	$\lg x$	<pre>>> log10(4) ans = 0.6021</pre>
factorial(x)	$n!$	<pre>>> factorial(5) ans = 120</pre>

Table: Funcții matematice elementare – continuare

Funcție	Descriere	Exemplu
<code>sin(x)</code> <code>sind(x)</code>	$\sin x$ (x în radiani) $\sin x$ (x în grade)	<pre>>> sin(pi/6) ans = 0.5000</pre>
<code>cos(x)</code> <code>cosd(x)</code>	$\cos x$ (x în radiani) $\cos x$ (x în grade)	<pre>>> cosd(30) ans = 0.8660</pre>
<code>tan(x)</code> <code>tand(x)</code>	$\tan x$ (x în radiani) $\tan x$ (x în grade)	<pre>>> tan(pi/6) ans = 0.5774</pre>
<code>cot(x)</code> <code>cotd(x)</code>	$\cot x$ (x în radiani) $\cot x$ (x în grade)	<pre>>> cotd(30) ans = 1.7321</pre>

Table: Funcții trigonometrice

Funcție	Descriere	Exemplu
<code>asin(x)</code> <code>asind(x)</code>	$\arcsin x$ (în radiani) $\arcsin x$ (în grade)	<pre>>> asin(0.5) ans = pi/6</pre>
<code>acos(x)</code> <code>acosd(x)</code>	$\arccos x$ (în radiani) $\arccos x$ (în grade)	<pre>>> acosd(0.8660) ans = 30</pre>
<code>atan(x)</code> <code>atand(x)</code>	$\arctg x$ (în radiani) $\arctg x$ (în grade)	<pre>>> atan(0.5774) ans = pi/6</pre>
<code>acot(x)</code> <code>acotd(x)</code>	$\operatorname{arcctg} x$ (în radiani) $\operatorname{arcctg} x$ (în grade)	<pre>>> acotd(1.7231) ans = 30</pre>

Table: Funcții trigonometrice inverse

Funcție	Descriere	Exemplu
<code>sinh(x)</code>	$(e^x + e^{-x})/2$	<pre>>> sinh(2) ans = 3.6269</pre>
<code>cosh(x)</code>	$(e^x + e^{-x})/2$	<pre>>> cosh(2) ans = 3.7622</pre>
<code>tanh(x)</code>	$(e^{2x} - 1)/(e^{2x} + 1)$	<pre>>> tanh(2) ans = 0.9640</pre>
<code>coth(x)</code>	$(e^{2x} + 1)/(e^{2x} - 1)$	<pre>>> coth(2) ans = 1.0373</pre>

Table: Funcții trigonometrice hiperbolice

Funcție	Descriere	Exemplu
<code>round(x)</code>	Rotunjire către cel mai apropiat număr întreg	<pre>>> round(17/5) ans = 3</pre>
<code>fix(x)</code>	Rotunjire către zero	<pre>>> round(13/5) ans = 2</pre>
<code>ceil(x)</code>	Rotunjire către ∞	<pre>>> ceil(11/5) ans = 3</pre>
<code>floor(x)</code>	Rotunjire către $-\infty$	<pre>>> floor(-9/4) ans = -3</pre>

Table: Funcții de rotunjire

Funcție	Descriere	Exemplu
rem(x, y)	Restul împărțirii lui x la y	<pre>>> rem(-12, 5) ans = -2</pre>
sign(x)	Funcția signum	<pre>>> sign(6) ans = 1</pre>

Table: Funcții de rotunjire – continuare

Variabile scalare

- O **variabilă scalară** este un nume alcătuit dintr-o literă sau mai multe litere și cifre și căreia îi este atribuită valoarea numerică.
- Odată ce o valoare numerică este atribuită unei variabile scalare, variabila respectivă poate fi folosită în:
 - o expresii matematice;
 - o funcții;
 - o comenzi și declarații MATLAB.
- În fapt, o **variabilă scalară** este un nume alocat unui spațiu de memorie, unde este stocată valoarea numerică atribuită variabilei respective.
- Dacă unei variabile scalare (deja definite și cu o valoare numerică atribuită) i se atribuie o nouă valoare numerică, atunci vechea valoare numerică este înlocuită în memorie de această nouă valoare numerică.

- **Operatorul de atribuire** din MATLAB este semnul egal (=) și acesta atribuie o valoare unei variabile:

NumeVariabila = Expresie

- o **NumeVariabila** este numele variabilei scalare;
- o **Expresie** este o valoare numerică sau o expresie matematică.

EXEMPLUL #1: Operatorul de atribuire

```
>> x = 15      Numărul 15 este atribuit variabilei x
x =
    15      MATLAB afișează variabila x
             și valoarea atribuită variabilei x

>> x = 3 * x - 12  O nouă valoare este atribuită variabilei x
x =
    33      MATLAB afișează variabila x
             și valoarea atribuită variabilei x
```

EXEMPLUL #2: Definirea unei variabile prin intermediul a două variabile definite anterior

```
>> a = 12      Numărul 12 este atribuit variabilei a
a =
    12      MATLAB afișează variabila a
             și valoarea atribuită variabilei a

>> B = 4      Numărul 4 este atribuit variabilei B
B =
    12      MATLAB afișează variabila B
             și valoarea atribuită variabilei B

>> C = (a - B) + 40 ...  Valoarea unei expresii este atribuită
>> -a/B * 10      variabilei C
C =
    18      MATLAB afișează variabila C
             și valoarea atribuită variabilei C
```

EXEMPLUL#3: EXEMPLUL#2, scris folosind semnul punct și virgulă

```
>> a = 12;           Numărul 12 este atribuit variabilei a
>> B = 4;           Numărul 4 este atribuit variabilei B
>> C = (a - B) + 40 ... Valoarea unei expresii este atribuită
>> -a/B * 10;        variabilei C
>> C
C =
    18               MATLAB afișează variabila C
                        și valoarea atribuită variabilei C
```

EXEMPLUL#4: EXEMPLUL#2, scris mai compact

```
>> a = 12, B = 4;    Numerele 12 și 4 atribuite variabilelor a și B
a =                  MATLAB afișează variabila a
    12               și valoarea atribuită variabilei a
>> C = (a - B) + 40 ... Valoarea unei expresii este atribuită
>> -a/B * 10         variabilei C
C =                  MATLAB afișează variabila C
    18               și valoarea atribuită variabilei C
```

Reguli pentru numele variabilelor

- Numele unei variabile
 - o trebuie să înceapă cu o literă;
 - o poate avea cel mult 63 de caractere;
 - o poate conține litere, cifre și semnul `_`;
 - o nu poate conține semne de punctuație (e.g. punct, virgulă, punct și virgulă).
- MATLAB face distincție între literele mari și cele mici (*case sensitive*).
- Spațiile nu sunt permise între caracterele numelui unei variabile.
- Trebuie evitate numele funcțiilor predefinite (e.g. `sin`, `exp`, `sqrt` etc.).

Cuvinte cheie predefinite

- Există 20 de cuvinte rezervate de MATLAB pentru diverse scopuri și care nu pot fi folosite ca nume de variabile.
- Când sunt folosite, cuvintele cheie apar scrise cu culoare albastră.
- Un mesaj de eroare este afișat când se încearcă folosirea unui cuvânt cheie ca nume de variabilă.
- Cuvintele cheie sunt afișate prin intermediul comenzii MATLAB `iskeyword`.

<code>break</code>	<code>case</code>	<code>catch</code>	<code>classdef</code>
<code>continue</code>	<code>else</code>	<code>elseif</code>	<code>end</code>
<code>for</code>	<code>function</code>	<code>global</code>	<code>if</code>
<code>otherwise</code>	<code>parfor</code>	<code>persistent</code>	<code>return</code>
<code>spmd</code>	<code>switch</code>	<code>try</code>	<code>while</code>

Variabile predefinite

ans Variabilă a cărei valoare corespunde ultimei expresii neatribuite explicit unei variabile

pi π

eps Cea mai mică diferență dintre două numere, i.e. 2^{-52}

inf ∞

i $\sqrt{-1}$

j $\sqrt{-1}$

NaN Not-a-Number – corespunde unei nedeterminări, e.g. $0/0$

Comenzi MATLAB pentru manipularea datelor

clear Șterge din memorie toate variabilele

clear x y z Șterge din memorie variabilele x, y și z

who Afișează lista variabilelor curente din memorie

whos Afișează lista variabilelor curente din memorie și dimensiunile acestora, precum și informații despre memoria ocupată și tipul variabilelor

Fișiere script

- Folosirea ferestrei de comenzi (Command Window) pentru executarea unei înșiruri de comenzi - în special când acestea sunt legate între ele (i.e. **un program**) - **nu este convenabilă** și poate fi dificilă/imposibilă.
- Fereastra de comenzi (Command Window) **nu este interactivă**, i.e. prin apăsarea tastei **Enter**, doar ultima comandă scrisă este executată, comenzile anterioare rămânând neschimbate.
- Un mod convenabil de execuție a comenzilor în MATLAB constă în crearea unui fișier ce conține o serie de comenzi (i.e. **program**), salvarea și rularea/execuția sa. Un astfel de fișier s.n. **fișier script**.

Observații:

- Un **fișier script** este o înșiruire de comenzi MATLAB, numit și **program**.
- La rularea/execuția unui fișier script, comenzile conținute în acest fișier sunt executate în ordine în care acestea sunt scrise.
- Dacă un fișier script conține un output (e.g. o instrucțiune de atribuire fără semnul punct și virgulă), atunci output-ul va apărea în fereastra de comenzi (Command Window).
- Folosirea unui fișier script este convenabilă deoarece fișierul poate fi editat (corectat sau chiar schimbat) și rulat/executat în mod repetat.
- Fișierele script pot fi editate în orice editor de text și apoi copiate în editorul MATLAB.
- Fișierele script se mai numesc și **fișiere M** în conformitate cu extensia *.m obținută implicit după ce acestea sunt salvate.

PROBLEMA #1: Soldul S al unui cont de economii după timpul t (măsurat în ani) de investiție a capitalului C , cu o rată anuală a dobânzii d și o dobândă calculată în n tranșe anuale, este calculat cu formula

$$S = P \left(1 + \frac{d}{n} \right)^{nt}.$$

- Calculați soldul unui cont de economii după 17 ani de investiție a sumei de 5.000 USD cu o rată anuală a dobânzii de 8,5% și o dobândă calculată într-o singură tranșă anuală.
- Calculați t pentru soldul obținut la (a) și investiția aceluiași capital de cu o rată anuală a dobânzii de 8,5% și o dobândă calculată lunar.
- Determinați numărul de ani și de luni corespunzătoare lui t obținut la (b).

PROBLEMA #1: Fișier script [Curs1Problem1.m](#)

```
clear % Sterge din memorie toate var.

C = 5000; % Capitalul investit
d = 0.085; % Rata dobanzii anuale
n1 = 1; % Transe anuale dobanda calculata (a)
t1 = 17; % Nr ani de investitie cu o dobanda
        % calculata in n1 transe anuale
n2 = 12; % Transe anuale dobanda calculata (b)

S = C * (1 + d/n1)^(n1 * t1) % PROBLEMA #1(a)

t2 = log(S/C)/(n2 * log(1 + d/n2)) % PROBLEMA #1(b)

ani = fix(t2) % PROBLEMA #1(c)
luni = ceil((t2 - ani) * 12)
```