


INTRODUCERE ÎN MATLAB

MATLAB este un pachet de programe de înaltă performanță, ce integrează calculul numeric și reprezentărilor grafice în domeniul științei și ingineriei. MATLAB oferă, pe baza unor lucrări matematice, o analiză matricială, sinteza și identificarea sistemelor și programe grafice inginerești atât 2D cât și 3D. MATLAB-ul integrează toate acestea într-un mediu ușor de învățat și folosit, în care enunțurile problemelor și rezolvările acestora sunt exprimate în modul cel mai natural posibil, așa cum sunt scrise matematic, fără a fi necesară programarea tradițională.

O caracteristică importantă a acestui limbaj de programare este usurița cu care acesta poate fi extins. Prin aceasta, orice utilizator poate adăuga propriile programe scrise în MATLAB la fișierele originale, dezvoltând aplicații specifice domeniului în care lucrează. Structural MATLAB-ul este realizat sub forma unui nucleu de bază, cu interpretor propriu, în jurul căruia sunt construite toolbox-urile. Toolbox-urile sunt colecții extinse de funcții MATLAB care dezvoltă mediul de programare de la o versiune la alta, pentru a rezolva probleme specifice anumitor domenii. Firma The MATHWORKS Inc. (care a realizat acest limbaj) a pus în circulație o serie de toolbox-uri cum ar fi: SIGNAL PROCESSING, IMAGE PROCESSING, SYMBOLIC MATH, NEURAL NETWORK, CONTROL SYSTEM DESIGN, ROBUST CONTROL, SYSTEM IDENTIFICATION, OPTIMISATION, SPLINE, STATISTICS, etc.

1.1. Lansarea în execuție

Programul se lansează în execuție din mediul WINDOWS, prin selecția pictogramei MATLAB  , în momentul în care se lansează suprafața de lucru (desktop) a sistemului de operare (fig.1.1). Versiunea descrisă în această lucrare este MATLAB 6.5 – Release 13.

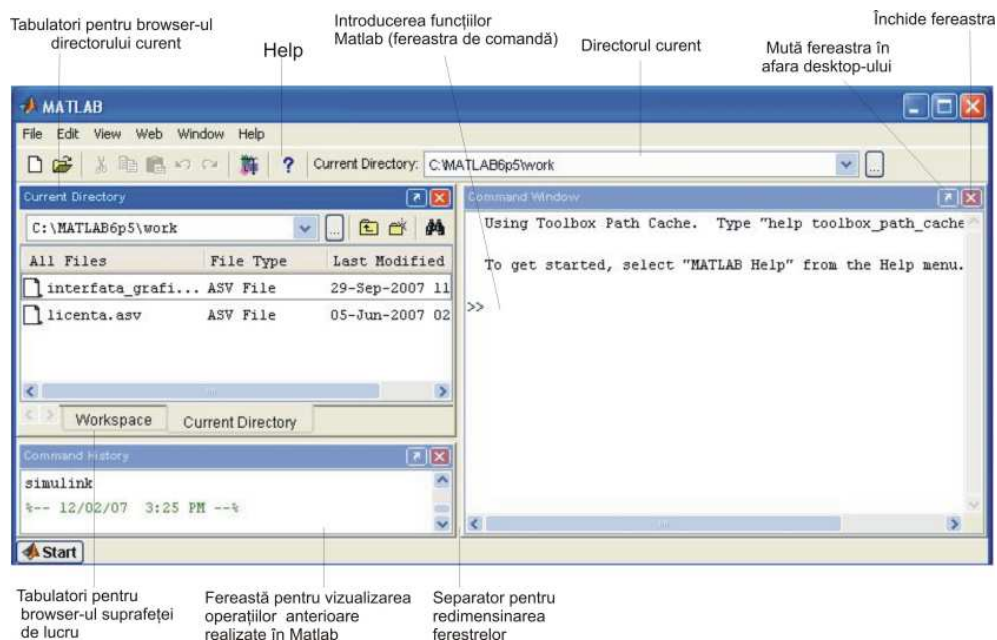


Fig.1.1. Suprafața de lucru MATLAB

Din afara mediului WINDOWS (de exemplu DOS) programul poate fi lansat în execuție direct, cu o comandă de forma:

„win c:\MATLAB\BIN\matlab”,

dacă fișierul „matlab.exe” se găsește în calea „c:\MATLAB\BIN”.

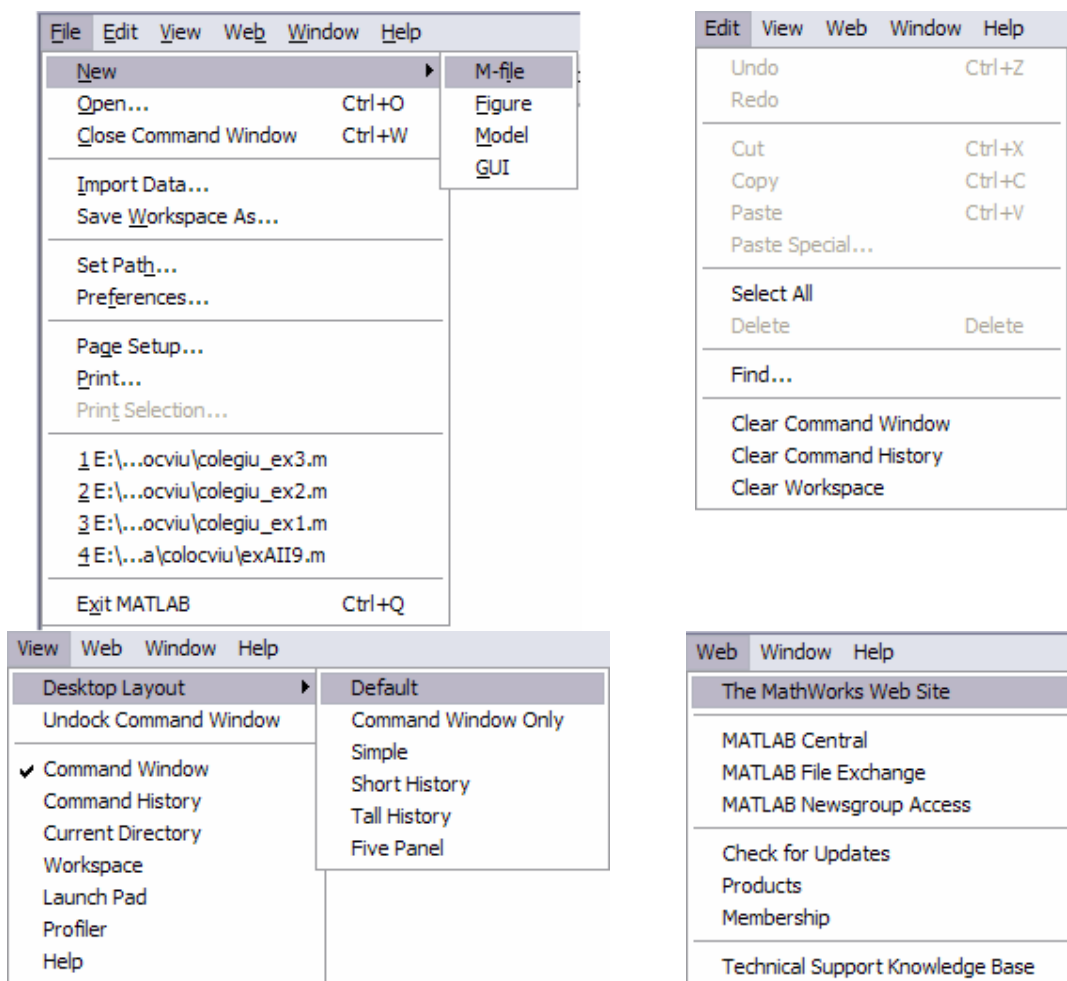
1.2. Ferestrele de lucru

Programul MATLAB lucrează cu două tipuri de ferestre: o fereastră de comenzi și o fereastră pentru reprezentări grafic.

1.2.1. Fereastra de comenzi

Este utilizată pentru scrierea, modificarea și depanarea funcțiilor programelor (fișierelor de tip M). Meniul din bara superioară este accesibil prin tastarea simultană a tastei [Alt] și a literei subliniate a comenzii dorite sau prin selecția directă cu mouse-ul.

Fiecare comandă din meniul principal furnizează un meniu specific, selecția comenzii dorite făcându-se prin deplasarea zonei active cu ajutorul săgeților sau prin selecția directă cu ajutorul mouse-ului. Aceste submeniuuri sunt prezentate în fig.1.2.



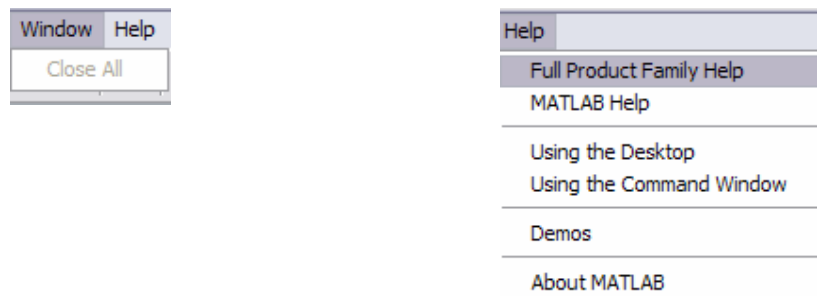


Fig.1.2. Submeniurile limbajului Matlab

Meniul **F**ile conține o serie de comenzi legate de fișiere:

- **N**ew – este folosit pentru deschiderea unui fișier nou de tip M (**M -file**), a unei noi ferestre grafice (**F**igure), a unui model SIMULINK nou (**M**odel) sau a unei interfețe grafice de tip Graphic User Interface (**G**UI);
- **O**pen – permite deschiderea unor fișiere de tip M deja existente pentru editare și lansare;
- **C**lose **C**ommand **W**indow – închide fereastra de comandă MATLAB;
- **I**mport **D**ata – opțiune destinată importului de date din fișiere;
- **S**ave **W**orkspace **A**s – salvează toate variabilele din spațiul de lucru împreună cu valorile lor atribuite într-un fișier desemnat;
- **S**et **P**ath – permite setarea căii fișierelor MATLAB;
- **P**reference – acesează o casetă de dialog în care se pot seta o serie de caracteristici ale programului MATLAB;
- **P**rint și **P**rint **S**election – tipărește conținutul ferestrei de comandă, respectiv a părții selectate a acesteia;
- **E**xit **M**ATLAB – opțiune folosită pentru părăsirea mediului MATLAB.

Meniul **E**dit conține o serie de comenzi cunoscute din Windows destinate editării: **U**ndo, **R**edo, **C**ut, **C**opy, **P**aste, **P**aste **S**pecial, **S**elect **A**ll, **D**elete, precum și trei comenzi specifice pentru ștergerea conținutului ferestrei de comandă, a ferestrei istoric și a suprafeței de lucru: **C**lear **C**ommand **W**indow, **C**lear **C**ommand **H**istory și **C**lear **W**orkspace.

În meniul **V**iew se regăsesc comenzi destinate manipulării diferitelor ferestre aferente instrumentelor utilizate în MATLAB. Comenzile din meniul **W**eb pot accesa resursele din INTERNET ale companiei producătoare (pagina de web, suportul tehnic, etc.).

În meniul **W**indow sunt afișate toate ferestrele deschise în timpul sesiunii de lucru curente, putându-se afișa fereastra activă. Ultimul meniu este meniul **H**elp, în care prin intermediul comenzilor existente se poate accesa documentația atașată programului.

1.2.2. Fereastra de reprezentări grafice

Este o formă elevată de reprezentare a graficelor. Pot exista mai multe ferestre grafice deschise simultan, dar numai o singură fereastră de comenzi.

1.3. Programarea în MATLAB

1.3.1. Matrice, vectori și scalari

MATLAB-ul lucrează numai cu un singur tip de obiecte, matrice numerice rectangulare, cu elemente reale sau complexe. În acest sens, scalarii sunt asimilați matricelor cu o linie și o coloană (1x1), iar vectorii sunt asimilați matricelor cu o linie (1xN) sau cu o coloană (Nx1).

Elementele unei matrice pot fi identificate prin una dintre notațiile: A_{ij} , $A[i, j]$, $A(i, j)$ etc. și semnifică elementul de la intersecția liniei i cu coloana j . Notăția adoptată în MATLAB este ultima și anume $A(i, j)$. Dimensiunea unei matrice este precizată de o pereche de numere care arată numărul de linii și coloane al matricei respective.

Pentru a face referire la un element $A(i, j)$ al unei matrice A , sunt necesari doi indici, indicele de linie și indicele de coloană, în această ordine. Referirea unui element al unui vector poate fi făcută numai cu un singur indice.

Definirea matricelor se poate face prin una din următoarele metode:

- introducerea explicită a listei de elemente;
- generarea prin instrucțiuni și funcții;
- crearea de fișiere M^1 ;
- încărcarea din fișiere de date externe.

MATLAB-ul nu conține instrucțiuni de dimensionare și declarații de tip, iar memoria este alocată în mod automat până la valoarea maxim disponibilă. Cea mai simplă metodă de definire a matricelor mici constă în utilizarea unei liste explicite. La introducerea unei astfel de liste trebuie respectate următoarele reguli:

- elementele unei linii trebuie separate prin blanc-uri sau virgule;
- liniile se separă prin semnul punct-virgulă “;”;
- elementele unei matrici sunt cuprinse între paranteze drepte “[]”.

1.3.2. Declarații și variabile

MATLAB-ul este un limbaj de expresii. Expresiile tipărite de utilizator sunt interpretate și evaluate. Instrucțiunile MATLAB sunt, de cele mai multe ori, de forma:

variabilă = expresie

sau, mai simplu:

expresie

Expresiile sunt compuse din operatori sau alte caractere speciale, din funcții și nume de variabile. Evaluarea expresiei produce o matrice care este afișată pe ecran și atribuită unei variabile. Dacă numele variabilei și semnul egal („variabilă =”) sunt omise, MATLAB-ul crează automat o variabilă cu numele „ans”.

Orice instrucțiune este în mod normal terminată cu „Enter”. Dacă ultimul caracter al unei instrucțiuni este punct-virgulă “;”, instrucțiunea este executată, dar tipărirea este suprimată. Utilizarea acestui caracter la sfârșitul unei instrucțiuni în fișiere M este necesară în situațiile în care nu se dorește afișarea datelor intermediare.

¹ Fișierele de tip M sunt fișiere ce conțin instrucțiuni MATLAB și se numesc așa deoarece au extensia “m”. Acestea sunt de fapt programe MATLAB.

Numele de variabile și de funcții au ca prim caracter o literă, urmată de litere, cifre sau caracterul special „liniuță de subliniere” (adică „_”). MATLAB-ul face deosebirea între litere mari și mici. Funcția “casesen” permite trecerea MATLAB-ului în modul senzitiv/nesenzitiv, în vederea separării literelor mari de cele mici. La lansare MATLAB-ul este în modul senzitiv, adică face distincție între litere mari și mici, dar cu comanda casesen off trece în modul nesenzitiv. Revenirea se face cu comanda casesen on.

Numele de funcții este obligatoriu să fie redactat cu litere mici!

1.3.3. Structura programelor în MATLAB

MATLAB-ul lucrează fie în modul linie de comandă, caz în care fiecare linie este prelucrată imediat și rezultatele sunt afișate, fie cu programe scrise în fișiere. Aceste două moduri de lucru formează împreună un mediu de programare ușor de învățat și de utilizat.

Fișierele ce conțin instrucțiuni MATLAB se numesc fișiere M (deoarece au extensia “.m”) și sunt programe MATLAB. Un fișier M constă dintr-o succesiune de instrucțiuni MATLAB, cu posibilitatea apelării altor fișiere M precum și a apelării recursive.

Un program MATLAB poate fi scris sub forma unor fișiere „**script**” sau sub formă de fișiere „**function**”. Ambele tipuri de fișiere sunt scrise în format ASCII.

1.3.3.1. *Fișierele script*

Aceste fișiere sunt fișiere externe ce conțin secvențe de instrucțiuni MATLAB. Prin apelarea numelui fișierului, se execută secvența de instrucțiuni MATLAB conținută de acesta. După execuția completă, variabilele cu care a operat fișierul rămân în zona de memorie a aplicației. Fișierele script nu permit integrarea în programe mari realizate pe principiul modularizării, ele sunt folosite pentru rezolvarea unor probleme care cer comenzi succesive lungi, ce fac greoi lucrul în modul linie de comandă.

1.3.3.2. *Fișierele funcție (function)*

Dacă prima linie a fișierului conține cuvântul cheie „**function**”, fișierul respectiv este declarat ca fișier funcție. Spre deosebire de un fișier script, un fișier funcție poate lucra cu argumente. Variabilele definite și utilizate în interiorul fișierului funcție sunt localizate la nivelul acestuia. Deci, la terminarea execuției unei funcții, în memorie nu rămân decât variabilele de ieșire ale funcției.

Forma generală a primei linii a unui fișier funcție este următoarea:

function [param_ieșire]=nume_funcție(param_intrare)

unde:

function - este cuvântul cheie care declară fișierul ca fișier funcție. Prezența acestui cuvânt cheie este obligatorie;

nume_funcție - este numele funcției, adică numele sub care se salvează fișierul, fără extensie. Nu poate fi identic cu cel al unui fișier M preexistent.

param_ieșire - parametrii de ieșire trebuie separați cu virgulă și cuprinși între paranteze drepte. Dacă funcția nu are parametrii de ieșire, parantezele drepte și semnul egal nu mai au sens, pot să lipsească.

param_intrare - parametrii de intrare trebuie separați cu virgulă și cuprinși între paranteze rotunde. Dacă funcția nu are parametri de intrare, parantezele rotunde nu mai au sens, pot să lipsească.

1.3.4. Comentariile și help-ul într-un program

Într-un program MATLAB, un comentariu se introduce prin caracterul procent “%” plasat la începutul liniei. Dacă caracterul procent apare pe prima poziție într-o linie, aceasta este omisă de compilator, iar dacă apare într-o linie de program, partea de linie care urmează după semnul procent va fi omisă.

Într-un program, un comentariu poate apare în orice poziție, dar este recomandată însă prezența unui comentariu după prima linie care declară o funcție. În acest caz, comentariul, care apare imediat după prima linie de declarare a funcției, constituie help-ul fișierului respectiv. Dacă se apelează în fereastra de comandă:

help nume_funcție

se va afișa help-ul fișierului funcție respectiv.

1.4. Instrucțiuni și funcții de control logic

Instrucțiunile de control logic în MATLAB sunt: **if, else, elseif, for, while, break, return, end, error.**

1.4.1. Instrucțiunea condițională „if”

Această instrucțiune este folosită în cazul în care este necesară o selecție a grupului de instrucțiuni ce urmează a fi executat, selecție condiționată de valoarea de adevăr a unei expresii. Instrucțiunile condiționale utilizează operatorii relaționali și operatorii logici.

MATLAB-ul are șase operatori relaționali, care sunt utilizați pentru a compara două matrice de dimensiuni egale. Lista acestor operatori este prezentată în tabelul 1.1.

Tab.1.1. Operatorii relaționali

Operatorii relaționali	Semnificația
<	mai mic
< =	mai mic sau egal
>	mai mare
> =	mai mare sau egal
= =	identic
~ =	diferit

Operatorii relaționali compară două matrice sau două expresii matriceale, element cu element. Rezultatul este o matrice de aceeași dimensiune cu a matricelor care se

compară, cu elemente 1 când relația este ADEVĂRATĂ și cu elemente 0 când relația este FALSĂ. Primii patru operatori compară numai partea reală a operatorilor, partea imaginară fiind ignorată, iar ultimii doi operatori tratează atât partea reală cât și partea imaginară. Dacă unul dintre operanzi este un scalar și celălalt este o matrice, scalarul se „extinde” până la dimensiunea matricei. Forma generală de utilizare a acestor operatori este:

rezultat = expresie_1 operator_relațional expresie_2

Pentru combinarea a două sau mai multe expresii logice se utilizează operatorii logici prezentați în tabelul 1.2. Operatorii logici ȘI, SAU compară doi scalari sau două matrice de dimensiuni egale. Pentru cazul în care cei doi operanzi sunt matrice, se operează element cu element. Operatorul logic NU (sau complementul logic) este operator unar.

Tab.1.2. Operatorii logici

Operatorii logici	Simbol MATLAB	Prioritatea
NU (not)	~	1
ȘI (and)	&	2
SAU (or)		3

Operatorii logici au prioritate mai mică decât operatorii relaționali sau cei aritmetici. În tabelul 1.3. este dată tabela de adevăr a operatorilor logici.

Tab.1.3. Tabela de adevăr a operatorilor logici

A	B	~A	A B	A&B
FALS	FALS	ADEV.	FALS	FALS
FALS	ADEV.	ADEV.	ADEV.	FALS
ADEV.	FALS	FALS	ADEV.	FALS
ADEV.	ADEV.	FALS	ADEV.	ADEV.

Instrucțiunea if poate fi implementată ca instrucțiune **if** simplă, sau poate include clauzele **else** sau **elseif**. Forma generală a unei instrucțiuni **if** simplă este următoarea:

```
if expresie logică
    grup de instrucțiuni
end
```

Dacă expresia logică este adevărată, se execută grupul de instrucțiuni cuprins între instrucțiunea **if** și instrucțiunea **end**. În caz contrar, se trece la prima instrucțiune care urmează după instrucțiunea **end**.

Clauza **else** este folosită pentru a executa un set de instrucțiuni (grupul de instrucțiuni A), dacă expresia logică este adevărată și un alt set de instrucțiuni (grupul de instrucțiuni B) dacă expresia logică este falsă. Forma generală este:

```
if expresie logică
    grup de instrucțiuni A
else
    grup de instrucțiuni B
end
```

Dacă funcția de calculat are mai multe nivele de instrucțiuni **if-else** este recomandată utilizarea clauzei **elseif**. Aceasta are următoarea formă generală:

```
if expresie logică 1
    grup de instrucțiuni A
elseif expresie logică 2
    grup de instrucțiuni B
elseif expresie logică 3
    grup de instrucțiuni C
end
```

Această instrucțiune este evaluată în modul următor:

- dacă expresia logică 1 este adevărată, se execută numai grupul de instrucțiuni A;
- dacă expresia logică 1 este falsă și expresia logică 2 este adevărată, se execută numai grupul de instrucțiuni B;
- dacă expresiile logice 1 și 2 sunt false și expresia logică 3 este adevărată, se execută numai grupul de instrucțiuni C;
- dacă mai multe expresii logice sunt adevărate, prima expresie logică adevărată determină care grup de instrucțiuni se execută prima dată;
- dacă nici o expresie logică nu este adevărată, nu se execută nici un grup de instrucțiuni din structura **if**.

Clauza **elseif** poate fi combinată cu clauza **else** într-o structură generală de forma:

```
if expresie logică 1
    grup de instrucțiuni A
elseif expresie logică 2
    grup de instrucțiuni B
elseif expresie logică 3
    grup de instrucțiuni C
else
    grup de instrucțiuni D
end
```

În acest caz, dacă nici o expresie logică nu este adevărată, se execută grupul de instrucțiuni D.

1.4.2. Instrucțiunea repetitivă „for”

Instrucțiunea **for** permite repetarea unui grup de instrucțiuni din corpul buclei, de un anumit număr de ori. Structura generală a acestei instrucțiuni este următoarea:

```
for index = expresie
    grup de instrucțiuni
end
```

unde:

index- este numele contorului;

expresie- este o matrice, un vector sau un scalar;
grupul de instrucțiuni - este orice expresie MATLAB.

În aplicații, „**index = expresie**” are de cele mai multe ori următoarea formă:

k = inițial:pas:final

unde:

inițial – este prima valoare a lui k;

pas – este pasul. Dacă acesta este omis el este considerat implicit 1;

final – este cea mai mare valoare pe care o poate lua k.

La fiecare pas de calcul „**index**” are valoarea unuia dintre elementele expresiei. Dacă expresia este o matrice, ciclarea se face pe coloane. Pentru un ciclu **for** cu pasul negativ sau neîntreg se generează mai întâi un vector cu pasul și limitele dorite și apoi se citesc valorile acestuia în cadrul buclei **for**. La folosirea buclei **for** trebuie respectate următoarele reguli:

- indexul buclei **for** trebuie să fie o variabilă;
- dacă expresia este o matrice goală, bucla nu se execută. Se va trece la următoarea instrucțiune după **end**;
- dacă expresia este un scalar, bucla se execută o singură dată, cu indexul dat de valoarea scalarului;
- dacă expresia este un vector linie, bucla se execută de atâtea ori câte elemente are vectorul, de fiecare dată indexul având valoarea egală cu următorul element din vector;
- dacă expresia este o matrice, indexul va avea la fiecare iterație valorile conținute în următoarea coloană a matricei;
- la terminarea ciclului **for**, indexul are ultima valoare utilizată;
- dacă se utilizează operatorul două puncte „:” pentru a defini expresia, bucla se execută de $n = \left\lfloor \frac{final - initial}{pas} \right\rfloor + 1$ ori, dacă n este pozitiv, și nu se execută dacă n este negativ. Prin [] s-a notat valoarea întreagă a numărului.

1.5. Instrucțiunea repetitivă „while”

Instrucțiunea repetitivă **while** este o structură care se utilizează pentru repetarea unui set de instrucțiuni, atâta timp cât o condiție specificată este adevărată. Formatul general al acestei instrucțiuni este următorul:

while expresie
 grup de instrucțiuni
end

Grupul de instrucțiuni este executat cât timp expresia este adevărată. Dacă expresia nu este adevărată se trece la executarea primei instrucțiuni care urmează după **end**. Dacă expresia este totdeauna adevărată logic, bucla devine infinită.

Dintr-o buclă infinită se iese forțat prin apăsarea concomitentă a tastelor [Ctrl]+[C] !!!

1.5.1. Instrucțiunea „break”

Instrucțiunea **break** se utilizează pentru a ieși dintr-o buclă înainte ca aceasta să se fi terminat. Se recomandă utilizarea ei dacă o condiție de eroare este detectată în interiorul unei bucle. Această instrucțiune încetează execuția ciclurilor **for** și **while**. În cazul unor cicluri imbricate, **break** comandă ieșirea din ciclul cel mai interior. Se apelează cu sintaxa:

break

1.5.2. Instrucțiunea „return”

Instrucțiunea **return** comandă o ieșire normală din fișierul M către funcția care l-a apelat sau către tastatură. Se apelează cu sintaxa:

return

1.5.3. Instrucțiunea „error”

Instrucțiunea **error** permite afișarea unor mesaje la întâlnirea unei erori. Se apelează cu sintaxa:

error('mesaj')

După afișarea textului „**mesaj**” controlul este redat tastaturii.

1.5.4. Funcții de control logic

În continuare, se vor prezenta pe scurt funcțiile de control logic. În MATLAB avem următoarele funcții de control logic:

exist - verifică dacă variabilele sau funcțiile argument sunt definite;
any - testează dacă cel puțin un element al unei matrice verifică o condiție logică dată (testează dacă un vector are cel puțin un element diferit de zero; returnează 1 dacă condiția este verificată și 0 în caz contrar);
all - testează dacă toate elementele unei matrice verifică o condiție logică dată (testează dacă un vector are toate elementele diferite de zero; returnează 1 dacă condiția este adevărată și 0 în caz contrar);
find - returnează indicii elementelor diferite de zero;
isnan - testează dacă elementele unei matrice sunt NaN²;
isinf - testează dacă elementele unei matrice sunt infinite;
finite - testează dacă elementele unei matrice sunt finite.

Aceste funcții sunt des asociate cu instrucțiunea **if**.

1.6. Variabile speciale în MATLAB

eps – variabilă în care este memorată variabila relativă, pentru calcule efectuate în virgulă mobilă, valoarea implicită fiind 2.2204e-016;

² NaN – Not a Number, nu este un număr

pi – variabilă permanentă ce are asociată valoarea 3.14159265358;

$i = j = \sqrt{-1}$ - variabilă folosită pentru reprezentarea unității imaginare;

inf – variabilă folosită pentru reprezentarea lui plus infinit în aritmetica IEEE;

nargin – variabilă permanentă folosită pentru testarea numărului argumentelor de intrare ce trebuie introduse pentru apelarea unei funcții;

nargout - variabilă permanentă folosită pentru testarea numărului argumentelor de ieșire ale unei funcții;

flops – returnează numărul de operații în virgulă mobilă efectuate de către calculator;

computer – variabilă folosită pentru obținerea informațiilor referitoare la tipul calculatorului și numărul maxim de elemente pe care le poate gestiona versiunea respectivă de MATLAB;

realmax – reprezintă cea mai mare valoare pozitivă în virgulă mobilă care poate fi folosită în calcule, respectiv 1.7977e+308;

realmin - reprezintă cea mai mică valoare pozitivă în virgulă mobilă care poate fi folosită în calcule, respectiv 2.2251e-308;

isieee – funcție ce returnează 1 dacă calculatorul cu aritmetică IEEE și respectiv 0 în caz contrar;

version, **ver** – funcții pentru determinarea versiunii MATLAB și a toolbox-urilor instalate pe calculator.

1.7. Exerciții propuse

1. Să se introducă și să se afișeze următoarele matrice:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}; \quad B = [1 \quad 3 \quad 5 \quad 7]; \quad C = \begin{bmatrix} 1 \\ 4 \\ 7 \\ 9 \end{bmatrix};$$

Să se afișeze următoarele elemente:

- elementul de pe linia 2 și coloana 1 din matrice;
- elementul al treilea din vectorul linie;
- elementul al patrulea din vectorul coloană.

2. Să se scrie un fișier funcție, denumit “funcție1”, în care să se calculeze următoarea funcție:

$$f(x, y) = \begin{cases} x^3 + y^3, & \text{dacă } 0 \leq x - y \leq 10 \\ x^2 + y^2, & \text{dacă } x - y < 0 \text{ și } y \geq 0 \\ (x - y)^2, & \text{în restul cazurilor} \end{cases}$$

Această funcție se va implementa în două moduri:

- folosind instrucțiunea **if-else**;
- folosind instrucțiunea **if-elseif**.

Se vor calcula următoarele valori: $f(1, 3)$, $f(1, -3)$, $f(-3, 1)$, $f(-4, -1)$.

3. Să se calculeze suma primelor 100 de numere naturale. Se va calcula în două moduri: folosind instrucțiunea repetitivă **for** și folosind instrucțiunea repetitivă **while**. Pentru rezolvare se vor deschide două fișiere de tip M, în care se vor scrie cele două programe.

4. Să se scrie un program, care calculează suma elementelor vectorului $X = [5 \ 2 \ -9 \ 10 \ -1 \ 9 \ 1]$ până când întâlnește un număr mai mare de 8.

5. Fie $X = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 1 \\ 0 & 0 & 4 \end{bmatrix}$

Să se determine:

- coloanele în care matricea are cel puțin un element diferit de zero;
- coloanele în care matricea are toate elementele diferite de zero;
- coloanele în care matricea are cel puțin un element mai mare ca -1 ;