

Laborator 1

Introducere MATLAB

numere/variabile/functii uzuale

Help

- Listarea subiectelor de help se face cu comanda
>> help

HELP topics:

matlab\general - General purpose commands.

matlab\ops - Operators and special characters.

matlab\lang - Programming language constructs.

matlab\elmat-Elementary matrices and matrix manipulation.

matlab\elfun - Elementary math functions.

matlab\specfun - Specialized math functions.

matlab\matfun - Matrix functions - numerical linear algebra.

Help

Aceasta comanda explica cum functioneaza help-ul

```
>> help help
```

- Utilizand *help-ul* in linie de comanda sa se gaseasca functia Matlab ptr calculul **arctangentei** si sa se specifice sintaxa de apel a acesteia.

Help (1)

>> *help*

- Din lista afisata se constata ca nu exista un director destinat doar functiilor trigonometrice.
- arctangenta e o functie elementare. Tastati

>> *help elfun*

- Pentru a obtine doar un ecran la un moment dat se poate introduce intai comanda *more on*, adica:

>> *more on*

>> *help elfun*

>> *help atan*

Pentru a trece la urmatoarea pagina se poate apasa orice tasta.

Observatii

1. Denumirile functiilor Matlab se scriu cu litere mici. Ele apar scrise in help cu litere mari doar pentru a fi scoase in evidenta.
2. Daca s-ar fi presupus ca denumirea functiei arctangenta din Matlab este aceeaasi cu cea folosita in matematica, arctg, si s-ar fi apelat help-ul functiei cu această denumire:

```
>> help arctg
```

Matlab afiseaza un mesaj de eroare, anuntand ca nu gaseste fisierul-M arctg (arctg.m not found.).
3. Denumirea functiei Matlab si denumirea fisierului in care este implementată functia trebuie sa fie identice.

Help (2)

O alta facilitate este utilizarea comenzii *lookfor* care cauta in fisierele de help un cuvânt cheie.

sintaxa este : *lookfor cuvânt-cheie*

Sa se gaseasca functia Matlab pentru functia care calculeaza radical indice 3

Help (3)

Help in modul on-line. Exista si un sistem bazat pe hipertext, care da detalii asupra asupra celor mai multe comenzi si exemple.

- `>> helpwin`
Apasati pe MATLAB, apoi pe Functions, apoi pe In Alphabetical Order'

`>> doc`

Apasati **butonul Help** (cel cu ? galben) **din interfata**

Comenzi

Comanda:

- >> path – listeaza toate directoarele din MATLAB
- >> pwd – intoarce numele directorului de lucru curent
- >> cd – schimba directorul curent
- >> dir sau ls – listeaza continutul directorului curent
- >> edit vas.m – deschide editorul de text ptr a crea fisierul
- >> type vas.m – afiseaza pe ecran continutul fisierului
- >> delete vas.m – sterge fisierul
- >> what – listeaza fisierele *.m din directorul curent
- >> diary azi1 – salveaza toate comenzile din sesiunea curenta in fisierul azi1.
- >> why ce face aceasta comanda?

Calcule

Operatiile aritmetice de baza sunt

+ - adunare

- - scadere

* - inmultire

/ - impartire

^ - ridicarea la putere

Ordinea implicita a operatiilor se poate schimba cu ajutorul parantezelor.

Expresii si calcule

- Efectuati simple operatii de adunare, scadere, inmultire, impartire folosind: “+” “-” “*” “/”. La sfarsitul comenzii se tasteaza Enter ptr a transmite operatia programului Matlab.

Exemple:

a) $2 + \text{round}(6 / 9 + 3 * 2) / 2 - 3$

b) $2 + \text{floor}(6 / 9 + 3 * 2) / 2 - 3$

c) $2 + \text{ceil}(6 / 9 + 3 * 2) / 2 - 3$

- Se observa aparitia variabilei *ans* – *variabila in care programul introduce automat rezultatul unei comenzi ce nu defineste o variabila proprie.*

Calcule

- Efectuati urmatoarele calcule:

$$\frac{\sin \frac{\pi}{4} + \cos \frac{\pi}{4}}{\sqrt{5 + \ln(7)} - 1.5^{0.17}}$$

Se va scrie comanda:

$(\sin(\pi/4) + \cos(\pi/4)) / (\sqrt{5 + \log(7)} - 1.5^{0.17})$

Argumentele functiilor trigonometrice sunt in radiani: $\sin(90)$? scriem $\sin(90 * \pi / 180)$

Calcule

Care dintre expresii e mai mare?

$$2^\pi \text{ sau } \pi^2;$$

$$\pi + \sqrt{1 + \pi^2} \text{ sau } \pi^{1 + \sqrt{1 + \pi^2}};$$

$$\left(\sin \frac{\pi}{3} + \cos \frac{\pi}{4} \right)^\pi \text{ sau } \pi^{\sin \frac{\pi}{3} + \cos \frac{\pi}{4}}.$$

Calcule

Sa se calculeze:

$$\frac{1 + \sqrt{5}}{2} =$$

$$\left[(\sqrt{2})^\pi \right]^{\sqrt{3}} =$$

$$\left| \pi^2 - 10 \cos^2 \frac{\pi}{3} \right| =$$

Raspuns: a) 1.6180 b) 6.5919 c) 7.3696

Calcule

Sa se calculeze:

$$\left(\frac{3^\pi - \pi^3}{\frac{1 + \sqrt{3}}{3}} \right)^{\frac{\pi}{\sqrt{3}}} =$$
$$\frac{\ln(56) + \sin(\cos \sqrt{44.67})}{\sqrt[3]{\ln(678) + \lg(223.38) - \operatorname{tg}\left(\sin \frac{\pi}{\sqrt{3}}\right)}} =$$
$$\frac{1}{1 + \frac{\sqrt{2}}{1 + \frac{\sqrt{3}}{1 + \frac{2}{1 + \frac{\sqrt{5}}{1 + \frac{\sqrt{6}}{1 + \sqrt{6}}}}}}}} =$$

Raspuns: a) 0.3849 b) 2.4231 c) 0.5576

Calcule cu constante speciale

Sa se calculeze:

$$\text{realmax}+2 =$$

$$\text{inf}-\text{inf} =$$

$$\text{nan}+2 =$$

$$\text{realmax}-\text{realmin} =$$

$$\text{inf}/\text{inf} =$$

$$0*\text{inf} =$$

Calcule

De la analiza matematica se stie ca urmatoarele limite sunt egale cu derivata functiei in x , $f'(x)$:

$$\lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad \text{și} \quad \lim_{h \rightarrow 0} \frac{f(x+h) - f(x-h)}{2h}$$

Insemana ca ptr valori mici ale lui h , fractiile din limitele de mai sus aproximeaza derivata.

Exercitiu:

Luand $f(x) = \sin x$ sa se calculeze eroarea dintre valoarea exacta a derivatei in $\pi/4$ si aceste aproximari ptr h :

- a) 10^{-6} b) 10^{-8} c) 10^{-10} d) 10^{-12} e) 10^{-14}

Calcule

Erorile date de:

$e1 = \text{abs}(\cos(\pi/4) - (\sin(\pi/4+h) - \sin(\pi/4))/h)$ respectiv

$e2 = \text{abs}(\cos(\pi/4) - (\sin(\pi/4+h) - \sin(\pi/4-h))/h/2)$

a) $h=0.000001$	$e1=3.5344e-007$	$e2=5.2650e-011$
b) $h=0.00000001$	$e1=3.0503e-009$	$e2=3.0503e-009$
c) $h=0.0000000001$	$e1=9.2454e-007$	$e2=3.6942e-007$
d) $h=0.0000000000001$	$e1=5.7368e-006$	$e2=6.1248e-005$
e) $h=0.000000000000001$	$e1=0.0034$	$e2=0.0021$

Fractiile nu tind la derivata in $\pi/4$!

Erorile se cumuleaza! Si functiile elementare $\sin(x)$ si $\cos(x)$ au erori de aproximare. In plus aici avem o nedeterminare $0/0$.

Ptr masinile pe 32 biti folositi valori la jumatatea preciziei maxime 10^{-7} , 10^{-8} .

Afisarea/ conversia caracterelor

Un caracter este reprezentat folosind ' ' (ex: 'a' sau 'x')

Caracterele sunt puse intr-o secventa ordonata = **character encoding**, fiecarui caracter i se atribuie o valoare intrega = **ASCII**

Pentru a converti un caracter la valoarea sa numerica se foloseste functia **double ()** si invers functia **char()**

```
>>double('a')
```

```
97
```

```
>> char(97)
```

```
a
```

Afisarea

- Sirul de caractere trebuie introdus intre ' '
s=' acesta e un text'
- Afisarea mesajelor la intalnirea unei erori cu functia:
error(' mesajul meu ')
Cand **error()** e intr-un fisier*.m se intrerupe executia
- Afisarea unui mesaj de avertizare, nu intrerupe executia programului:
warning('mesaj')
- Comenzile: **warning on, warning off, lasterr, lastwarn**
- Afisarea pe ecran a unei informatii cu functia disp:
disp(' orice text')
- Introducerea datelor se face cu functia input:
x= input('Introduceti valoarea lui x = ')

Fisier script

In MATLAB programele sursa numite M- files sunt:

- fisiere -script
- fisiere - functie

M- files se pot crea cu orice editor de texte, salvandu-le cu extensia .m

>> edit vasile.m

Scriptul = fisier ce contine o succesiune de comenzi / instructiuni, care se pot edita, se pot repeta etc.

Rularea unui script se face tastand numele fisierului (fara extensie) in linia de comanda.

>> *vasile*

Exemplu: Script

Crearea unui script ce are numele **volsf** si care va calcula volumul unei sfere :

Selectati: File\New\M-file sau scrieti comanda:

>> edit volsf.m

In fisier scrieti urmatoarele:

raza = 5

volum= 4/3*pi*r^3

Selectati: File\Save\

In Command window dati comanda:

>>volsf % executam scriptul scriind doar numele lui

Exemplu: Script

Adaugati urmatoarele instructiuni in volsf.m la sfarsit:

```
disp('Volumul unei sfere de raza ');  
disp( [ num2str(raza) 'este' [num2str(volum)] );
```

Salvati si rulati scriptul

Scrieti in command window :

```
>>volsf
```

Exemplu: Script

Crearea unui script ce are numele `sindoix` si va calcula $\sin 2x$:

Selectati: File\New\M-file sau sciety comanda:

>> edit

```
x= input(' Introduceti valoarea lui x=')
```

```
y=2*sin(x).*cos(x);
```

Selectati: File\Save As\sindoix.m

In Command window dati comenzile:

```
>>sindoix % executam scriptul scriind doar numele
```

```
>> Introduceti valoarea lui x=pi/3
```

```
>> y=0.8660
```

Rulati din nou scriptul ptr $x=[\pi/3 \ \pi/4 \ \pi/6 \ \pi/2]$

Algoritm

Programele noastre conțin algoritmi.

Algoritm = este o secvență finită de operații lipsite de ambiguitate sau instrucțiuni pentru a rezolva o problemă și de a obține un rezultat (wiki)

Dorim să oferim parametrii algoritmului, numit parametru de intrare și să primim un răspuns, numit parametru de ieșire.

Fisier functie

- Prima linie din fisierul functiei trebuie sa fie de forma:
function [param-iesire]=nume-functie(param-intrare)
- Daca exista doar un parametru de iesire (care poate fi un scalar, vector sau matrice) parantezele patrate pot lipsi.
- Parantezele rotunde de la lista-param-intrare pot lipsi doar daca nu exista nici un parametru de intrare.
- Variabilele folosite sunt locale, nu sunt disponibile in Workspace.
- ; de la sfarsitul fiecărei linii previne afisarea in Workspace a evaluarilor intermediare.
- Rularea fisierului functiei se face folosind sintaxa:
[valori_returnate]=nume_functie(valori_param-intrare)

Fisier: functie

Exemplu: **definirea unui functii** cu numele **volsfera** ce va calcula volumul unei sferei :

Selectati: File\New\M-file sau comanda

>> edit volsfera.m

```
function v = volsfera(r)
```

```
    v= 4/3*pi*r^3;
```

```
end
```

In Command window dati comenzile:

```
>> v=volsfera(5)
```

Fisier: functie

Exemplu: **definirea unui functii** cu numele cosdoix ce va calcula $\cos^2 x$:

Selectati: File\New\M-file sau comanda **edit**

```
function y=cosdoix(x)
```

```
    y=cos(x).*cos(x)-sin(x).*sin(x);
```

```
end
```

Selectati: File\Save As\cosdoix.m

In Command window dati comenzile:

```
>> y=cosdoix(x)
```

```
>> y = -0.5000
```

```
>> (1+cosdoix(x))./cosdoix(x)
```

```
    ans = -1.0000
```

```
>> 1+sindoix
```

??? Attempt to execute SCRIPT sindoix as a function.

Avantajele si Dezavantajele Scripturilor

Scriptul este o secventa de instructiuni si a aparut din necesitatea reutilizarii multiple a secventei respective.

Avantaje:

- utile in cazul programelor mari cu secvente de comenzi care se repeta aleatoriu (nu sunt bucle pur si simplu) deci utile doar programatorilor avansati;

Dezavantaje:

- creeaza si modifica variabile din zona de lucru fara nici un avertisment; **variabilele din scripturi sunt globale**
- nu au parametrii de intrare sau de iesire;
- sunt sursa unor erori greu de detectat;
- nu pot fi utilizate in expresii.

Avantajele si Dezavantajele Functiilor

Functia este tratata drept un subprogram, adica drept o unitate distincta de restul programului.

Avantaje:

- functia comunica cu programul apelant prin cele doua liste: listaparam-iesire si lista-param-intrare.
- nu modifica datele din zona de lucru a programului apelant.
- permite organizarea structurala a programarii.

Dezavantaje:

- folosirea in exces a functiilor poate duce la faramitarea excesiva a programului pierzandu-se din claritate.

Atasarea comentariilor

Comentariile sunt precedate de semnul procent (%).

Daca se plaseaza pe primele linii, imediat sub linia de definitie a functiei, formeaza Help-ul functiei care se apeleaza cu comanda *help nume_functie*.

Help-ul functiei se considera pana la prima linie goala sau pana la prima instructiune Matlab

Dupa ultima linie a help-ului se lasa o linie libera.

Exemplu :

function nume_functie

% Prima linie de help

% A doua linie de help ...

% Ultima linie de help

Linie libera (aceasta este obligatorie daca urmeaza un alt comentariu care nu este de tip help)

Instructiuni MATLAB

Exercitii

1. Scrieti functia care calculeaza aria si perimetrul unui dreptunghi.

- Adaugati help functiei
- Apelati functia dintr-un script.

2. Scrieti functia pentru a calcula suma primilor $N+1$ termeni a unei serii geometrice, $N < \infty$

$$gsum = \sum_{n=0}^N a^n.$$

Exercitii: functii

Raspuns:

Fisierul arieper.m

```
function [a,p]=arieper(L,l)
```

```
% Calcul arie și perimetru al dreptunghiului cu laturile L și l
```

```
a=L*l;
```

```
p=2*(L+l);
```

```
end
```

Fisierul script runarieper:

```
% Setarea valorilor laturile L și l apelul functiei arieper
```

```
L=5; % sau L=input('L=');
```

```
l=6; % sau l=input('l=');
```

```
[a,p]=drept(L,l) %Apelul functiei
```

Executarea scriptului:

```
>> runarieper
```


Exercitii: functii

Raspuns:

```
function gsum = geom(a,N)  
  n=0:N;  
  if a > 1  
    • error('suma divergenta')  
  end  
  gsum = sum(a.^n);  
end
```

Functia geom se apeleaza de la command prompt :

```
>> gsum=geom(0.4,10).
```

Funcții Anonime

Anonymous functions - funcțiile Matlab produse de la linia de comanda. Funcție implicită, apelabilă ca și o variabilă în lista de argumente intrate într-o funcție numită **function_handle** *@*. Au forma generală:

nume_funcție = @(param- de intrare) (funcția)

Exemplu: $\gg f = @(x) (x.^2 + \exp(x))$
 $\gg rez = f(x) \quad \% \text{apelul funcției}$

f - numele funcției,

@ - function handle,

x - parametrul de intrare

$x.^2 + \exp(x)$ – corpul funcției

Functii Anonime

```
>>> x = [1 3 5]
```

```
>>> y=[ 2 2 2]
```

```
>>> f = @(x, y) ( x .* y)
```

```
>>> f(x,y)
```

```
>>> a =10
```

```
>>> f = @(x, y) a*(x .* y)  % a face parte din corpul fctiei
```

```
>>> f(x,y)
```

```
>>> a=0
```

```
>>> f(x,y)  % nu se schimba valoarea functiei evaluate
```

```
>>> clear a
```

```
>>> f      % variabila a este definita in corpul functiei
```

Functii inline si comanda feval

Inline functions - functiile Matlab produse de la linia de comanda folosind comanda **inline**:

nume_functie = inline ('expr', 'arg1', 'arg2'...)

Exemplu: `>> f = inline('sqrt(x.^2+y.^2)', 'x', 'y')`

`>> rez = f(x,y) % apelul functiei`

- **string** -sir de caractere inclus intre apostrofi- *'test 10'*

Unele functii iau ca parametrii de intrare numele altor functii specificate ca string. Evaluarea acestora se face cu:

- **[param-de-iesire]=feval ('numefunctie', param-de-intrare)** – orice functie data sub forma sir de caractere.
- **[param-de-iesire]=builtin ('numefunctie', param-de-intrare)** – numai functiile interne ale Matlab-ului.

Exercitii functii

Scriti aceste functii in fisier sau anonymous sau inline.
Calculati valorile functiilor pentru $x=0.5$ si $t=\sqrt{2}$:

$$(x^3 + 1) \ln(1 + t + t^2) =$$

$$e^{x+t} (1 + x + \cos t) =$$

$$\frac{\cos^2 x + \operatorname{arctg} t}{x + t + e^x} =$$

Exercitii

Raspuns:

$$(x^3+1)*\log(1+t+t^2)= 1.6704$$

$$\exp(x+t)*(1+x+\cos(t))= 11.2300$$

$$((\cos(x))^2+\operatorname{atan}(t))/(x+t+\exp(x^t))= 0.5121$$

Funcțiile save și load

Salvează variabilele într-un fișier binar *.mat:

```
>>x = 0:0.1:1  
>> y = [exp(x)]  
>>save -mat calculexp.mat x y  
>>load calculexp.mat
```

Salvează variabilele într-un fișier text *.txt (se poate salva doar o variabilă):

```
>> save -ascii calculexp.mat y  
>>c = load ('calculexp.txt')
```

Dacă vreți să salvați mai multe variabile folosiți funcțiile:

fopen() fprintf() fclose()

Funcția fprintf -scrierea datelor in format

fprintf(fid, format, date)

fid- este un un numar intreg, asociat pentru identificarea fisierului in care se scrie, fid=1 sau e omis scriere se face pe ecran. Identificatorul fid al fisierului se obtine prin functia **fopen**.

format – formatul de scriere precedat de caracterul '%' urmat de: d, i, o, u, x, X, f, e, E, g, G, c si s.

date – datele ce trebuiesc scrise

Conversia datelor poate contine si caractere speciale neprintabile:

\n produce o linie noua,

\r reîntoarcerea cursorului la cap de rand

\t tab orizontal,

\b backspace,

\f forma libera

Conversia datelor

Conversia datelor incepe intotdeauna cu caracterul % si contine urmatoarele elemente optionale sau obligatorii:

- **semne, optionale, care controleaza alinierea :**

- caracterul “-”, care pozitioneaza datele la stanga si justificat, iar daca lipseste pozitionarea, se face la dreapta;

- caracterul “+”, care scrie intotdeauna semnul datei (plus,+ , sau minus,-);

- caracterul “0”, care scrie în spatiile libere zero (0);

- **marimea si precizia campului,optional:**

- marimea câmpului in digiti, ex 6 (prin %6f);

- precizia, numarul de digiti aflatii dupa punctual zecimal de exemplu 3 (prin %6.3f);

Conversia datelor

Conversia datelor incepe intotdeauna cu caracterul % si contine urmatoarele elemente obligatorii:

• **formatul de conversie obligatoriu, care poate fi :**

semnul “c” un singur caracter;

semnul “d” notatia zecimala;

semnul “e” sau “E” scrierea exponentiala, cu litera e mica sau cu litera E mare;

semnul “f” punct zecimal sau virgula fixa;

semnul “g” sau “G” forma mai compacta decat “e” sau “f”, nescrisind zerourile nesemnificative;

semnul “o” date in baza opt (octagonala);

semnul “s” pentru caractere;

semnul “u” date in baza decimala;

semnul “x” sau “X” date hexazecimale, cu litere a-f

Funcția fprintf, sprintf – scriere în fișier, pe ecran

fprintf(fid, format, date)

Formatul **fprintf** este vectorizat pentru cazul în care datele nu sunt scalare, fiind repetat de câte ori e necesar.

```
>> fprintf('A unit circle has circumference %g\n', 2*pi)
```

Scriu valorilor funcției exponentiale **intr-un fișier**:

```
>> x = 0:0.1:1; y = [exp(x)];  
>> fid = fopen('calculexp.txt','w');  
>> fprintf(fid, '%6.2f \n', y);  
>> fclose(fid);  
>> C = load ('calculexp.txt')
```

[s]=sprintf(format, date)- returnează datele sir de caractere.

Exemple:

```
sprintf('%d', round(pi))      ans = 3  
m=sprintf('%s', 'hello')     m=hello, adică m(3)=l
```

Debugging fisierelor .m

Depanarea sau ***Debugging*** este procesul prin care sunt identificate si rezolvate erorile din cadrul codului Matlab.

- **Erori de sintaxa:** introducerea gresita a unui nume de functie sau omiterea unei paranteze.

MATLAB afiseaza un mesaj de eroare in *Command Window* descriind eroarea si numarul liniei din fisierul m.

Clicati pe portiunea subliniata a mesajului de eroare, sau pozitionati cursorul pe mesajul de eroare si apasati *Ctrl+Enter*. Fisierul m ce contine eroarea se deschide in *Editor*, cu linia eronata afisata in prim plan.

- **Erori run-time:** aceste erori sunt in general de algoritm. Erorile run-time sunt vizibile cand fisierul m produce rezultate neasteptate.

Sfaturi pentru depanarea erorilor run-time

Stergeti simbolurile ”;” din instructiunile fisierelor .m cu pricina.

In urma acestei operatiuni MATLAB va afisa rezultatelor pe ecran pe durata executiei fisierului .m.

Adaugati instructiunii ***keyboard*** la fisierul.m . Instructiunile ***keyboard*** opresc executia fisierului .m la un anumit punct unde acestea apar si va permit examinarea si modificarea workspace-ul local al functiei. Acest mod este indicat de un prompt special: ***K>>*** Continuarea executiei functiei se realizeaza tastand ***return*** si apasand tasta ***Return***.

Folositi ***MATLAB Debugger*** sau functiile ***debugging***. Acestea sunt folosite pentru corectarea erorilor ***run-time*** deoarece puteti accesa workspace-ul functiilor si apoi puteti examina sau schimba valorile pe care le contin. Puteti defini sau sterge ***puncte de intrerupere*** sau ***breakpoints***, pentru oprirea executiei programului la o anumita locatie (la o anumita linie de program) in interiorul fisierelor .m.

De citit si exercitii

Stormy Attaway, Matlab, A Practical Introduction to Programming and Problem Solving, 3rd Edition, 2013 Elsevier Inc.

Capitolul 1 – Introducere in MATLAB (fara subcapitolul de Random Numbers)

Exercitiile din Capitolul 1 : 1, 6, 9, 13, 15, 16, 22, 23, 28, 29, 30

Capitolul – Introducere in programarea MATLAB subcapitolele 3.2 , 3.3, 3.4 3.6

Exercitiile din Capitolul 3 : 1, 2, 5, 6, 16

Capitolul 9

De citit si exercitii

Funcții

Secțiunile 3.7 6.1 10.1 și 10.2

Exercițiile din Capitolul 3 : 24, 25, 29, 37

Exercițiile din Capitolul 10 : 1,2,6,7