# ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ Τμήμα Μηχανικών Παραγωγής και Διοἰκησης

Εισαγωγή στην Τεχνητή Νοημοσύνη (ΜΠΔ306)

Χειμερινό εξάμηνο 2020-2021 Εργαστηριακός βοηθός: Γιαννακάκη Κατερίνα

# Εισαγωγή στο MATLAB Ασκήσεις εξάσκησης

# Εισαγωγή

Οι ασκήσεις που ακολουθούν σκοπό έχουν την εξοικείωση με το περιβάλλον και τη γλώσσα προγραμματισμού MATLAB.

Το κείμενο που βρίσκεται δεξιά από το σύμβολο >> μπορεί να γραφεί στο Command Window του MATLAB και να εκτελεστεί. Κείμενο που βρίσκεται δεξιά από το σύμβολο % είναι κείμενο σχολίων. Σχόλια γραμμένα με bold είναι εκφωνήσεις ασκήσεων για τις οποίες πρέπει να δώσετε μόνοι σας την εντολή.

# Αριθμητικές πράξεις

Εκτελέστε τις παρακάτω αριθμητικές πράξεις και παρατηρήστε το αποτέλεσμα που δίνουν. Προσέξτε την προτεραιότητα των πράξεων και το αποτέλεσμα στην περίπτωση χρήσης παρενθέσεων.

>>	4*2^3	% η ύψωση σε δύναμη έχει μεγαλύτερη προτεραιότητα % από τον πολλαπλασιασμό
>>	(4*2)^3	% οι παρενθέσεις έχουν μεγαλύτερη προτεραιότητα
>>	4*(2^3)	% οι παρενθέσεις εδώ δεν χρειάζονται, δίνουν % προτεραιότητα στην πράξη με την εξ' ορισμού % μεγαλύτερη προτεραιότητα
>>	20-5^2	
>>	20^5-2	
>>	36/3^2	

```
>> 6^2/4
>> (4*(10-8))^3 % η πράξη στην πιο εσωτερική παρένθεση έχει
% προτεραιότητα
>> 20-5+3-8+10-2^3
```

Όταν ο κέρσορας βρίσκεται μπροστά από το prompt (>>) στο Command Window, και χωρίς να έχετε γράψει κάτι άλλο, αν πατήσετε το **πάνω βελάκι**, θα εμφανιστεί το **ιστορικό** των εντολών που έχετε δώσει και μπορείτε να επιλέξετε μια από αυτές για να την επαναλάβετε. Αν γράψετε τους πρώτους χαρακτήρες από κάποια προηγούμενη εντολή που έχετε δώσει και μετά πατήσετε το πάνω βελάκι, τότε θα εμφανιστεί το ιστορικό μόνο για τις εντολές που ξεκινούν με αυτούς τους χαρακτήρες.

#### Διανύσματα/Πίνακες

```
>> x = [2 \ 3 \ 8 \ 12]
                      % δημιουργεί διάνυσμα-γραμμή
                      % (προσοχή στα κενά (spaces))
>> x = [2 3 8 12];
                     % semicolon (;)!
                      % δημιουργεί διάνυσμα-γραμμή με τιμές από
>> x = [3:9]
                      % 3 μέχρι 9 (το default βήμα είναι 1)
>> x = [9:-1:1]
                      % δημιουργεί διάνυσμα-γραμμή με τιμές από
                      % 9 μέχρι 1 με βήμα -1
>> x = [0:0.1:1]
                      % προσοχή στη χρήση της τελείας για
                      % υποδιαστολή
>> % δημιουργείστε διάνυσμα-γραμμή με τιμές 10, 14, 18, 22, 26, 30
>> x = [10:4:30]
>> x
                      % εμφανίζει τα περιεχόμενα του διανύσματος χ
>> x = linspace(0, 2, 11)
>> % δημιουργείστε διάνυσμα-γραμμή με 6 τιμές, με αρχική τιμή το -10
   % και τελική τιμή το 12
>> x = linspace(-10,12,6)
>> y = linspace(0,10);
>> length(x)
                % εμφανίζει το μήκος του διανύσματος χ
>> length(y)
                % εμφανίζει το μήκος του διανύσματος γ
>> size(x)
                % εμφανίζει τις διαστάσεις του διανύσματος χ
>> size(y)
                🖇 εμφανίζει τις διαστάσεις του διανύσματος γ
>> y(5)
                % εμφανίζει το 5° στοιχείο του διανύσματος y
```

>>	y(1) %	εμφανίζει το 1° στοιχείο του διανύσματος γ	
>>	y(100)		
>>	y(end) % १	εμφανίζει το τελευταίο στοιχείο του διανύσματος γ	
>>	y(1:10) %	εμφανίζει τα πρώτα 10 στοιχεία του διανύσματος γ	
>>	y(5:20)		
>>	y([1 3 20 8 10	12]) % εμφανίζει το 1°, το 3°, το 20°, το 8°, το 10° % και το 12° στοιχείο του διανύσματος γ	
>>	y([1,3,20,8,10,	12]) % το ίδιο με πριν	
>>	x '	% ανάστροφος	
>>	x, x'		
>>	x; x'		
>>	3*[1 2 5]		
>>	3*x		
>>	x/2		
>>	x^2 %	σφάλμα	
>>	x.^2 %	προσοχή στην τελεία	
>>	x-1		
>>	x = x + 1		
>>	a = [10 14 27;2	3 6 0] % πίνακας διαστάσεων 2x3	
>>	b = zeros(2, 5)		
>>	c = ones(2, 5)		
>>	ones(size(a))		
>>	z = randi(25,5)		
>>	z = randi(25, 4, 6)		
>>	z = rand(3, 5)		
>>	z = rand(4)		
>>	% δημιουργείστε πίνακα R με 5 γραμμές και 8 στήλες που να περιέχει τυχαίους πραγματικούς αριθμούς με τιμές στο διάστημα [20,25]		
>>	R = 25 + (25-20)	) * rand(5,8)	

```
>> % στο διάνυσμα b: αντικαταστήστε την τιμή του στοιχείου που
    βρίσκεται στη 2<sup>η</sup> γραμμή και 3<sup>η</sup> στήλη με την τιμή 5
>> b(2,3) = 5
% στο διάνυσμα C: αντικαταστήστε τις τιμές των στοιχείων που
  βρίσκονται στην 2<sup>η</sup> στήλη και στις δυο γραμμές με την τιμή Ο
>> c(:,2) = 0
% στον πίνακα z: αντικαταστήστε τις τιμές των στοιχείων που
  βρίσκονται στη 2<sup>η</sup>, 3<sup>η</sup> και 4<sup>η</sup> γραμμή και στην 1<sup>η</sup>, 2<sup>η</sup> και 4<sup>η</sup> στήλη με
  την τιμή 100
>> z([2:4],[1 2 4]) = 100
>> g = [1; 2; 3; 4]
>> q = [1 2 3 4; 5 6 7 8]
>> g - 2
>> 2*g-1
>> g = [1 2 3 4; 5 6 7 8; 9 10 11 12]
>> h = [1 1 1 1; 2 2 2 2; 3 3 3]
>> g-h
>> g*h
          % σφάλμα! γιατί;
>> h*g
            % τώρα γιατί έγινε ο πολλαπλασιασμός; τι άλλαξε;
>> g*h'
>> q'*h
>> g.*h
            % πολλαπλασιασμός στοιχείο προς στοιχείο
>> g./h
          % διαίρεση στοιχείο προς στοιχείο
>> e = [ 1 2 3; 4 5 6; 7 8 0]
>> f = [9 8 7; 6 5 4; 3 2 1]
>> e*f
>> f*e % διαφορετικό αποτέλεσμα
>> eye(4)
>> eye(2,4)
>> eye(4,2)
>> q = 1:4
>> diag(q)
>> repmat(q,4,3)
```

ΣΗΜΑΝΤΙΚΟ: Η αρχή της αρίθμησης (π.χ. πρώτο στοιχείο του πίνακα) για το ΜΑΤLAB είναι το 1, και όχι το 0 όπως σε άλλες γλώσσες προγραμματισμού.

#### help

>> help who
>> help whos
>> help size
>> help clc
>> help clear
>> help length
>> help linspace
>> help help

# Δεσμευμένες λέξεις

Οι παρακάτω λέξεις είναι δεσμευμένες λέξεις που χρησιμοποιούνται από το MATLAB και δεν θα πρέπει να χρησιμοποιούνται ως ονόματα μεταβλητών: for, end, if, while, function, return, elseif, case, otherwise, switch, continue, else, try, catch, global, persistent, break. Μπορείτε ωστόσο να χρησιμοποιήσετε την ίδια λέξη μετατρέποντας κάποιο από τα γράμματά της σε κεφαλαίο: το MATLAB το αναγνωρίζει ως διαφορετικό όνομα.

Επίσης, τα ονόματα: ans, beep, pi, eps, inf, NaN, nan, i, j, nargin, nargout, realmin, realmax, bitmax, varargin, vararout είναι ονόματα που χρησιμοποιεί το ίδιο το MATLAB για μεταβλητές και καλό θα ήταν να αποφεύγονται να δίνονται ως ονόματα μεταβλητών από το χρήστη. Δοκιμάστε τις παρακάτω εντολές:

```
>> end = 1 % σφάλμα

>> End = 1 % μια χαρά

>> pi % π

>> pi = 3 % αντικατάσταση της default τιμής

>> pi

>> clear pi

>> pi
```

## Output

Ό,τι θέλουμε να εμφανιστεί ως αποτέλεσμα ή μήνυμα στο command Window

>> disp('The disp command writes text') >> disp(' $\xi \alpha \nu \dot{\alpha}$  : The disp command writes text') >> t = [0:0.1:1]; >> x = 0.1234567890123456\*t;>> format short % ρυθμίζει τον τρόπο εμφάνισης των τιμών, ειδικά των % πολύ μεγάλων & πολύ μικρών αριθμών και των % δεκαδικών αριθμών με πολλά δεκαδικά ψηφία >> t >> x >> format long >> x >> format long e >> x >> format short e >> x >> format short g >> x >> format long g

>> x

Χρησιμοποιώντας το προσδιοριστικό μετατροπής % και το κατάλληλο γράμμα, ρυθμίζουμε την μορφή εμφάνισης της τιμής κάποιας μεταβλητής. Επιπλέον, χρησιμοποιώντας αριθμούς μαζί με το προσδιοριστικό μετατροπής, ρυθμίζουμε πόσα ψηφία της τιμής θέλουμε να εμφανιστούν.

Γενική μορφή: %Χ.Υα

όπου στη θέση του α επιλέγουμε ένα από τα f, e, g για δεκαδική, εκθετική ή συντμημένη μορφή αντίστοιχα

στη θέση Χ επιλέγουμε των αριθμό των συνολικών θέσεων (ακέραιο και δεκαδικό μέρος και υποδιαστολή) που θέλουμε να δεσμευτούν για τη συγκεκριμένη τιμή στη θέση Υ επιλέγουμε τον αριθμό δεκαδικών ψηφίων που θέλουμε να εμφανιστούν π.χ. αν γράψουμε %8.2f

θα εμφανιστεί ο αριθμός ως δεκαδικός, θα δεσμευτεί χώρος για συνολικά 6 ψηφία και θα εμφανιστούν μόνο τα 2 από τα δεκαδικά ψηφία του αριθμού. Δηλαδή από τον χώρο των 8 ψηφίων, τη μια θέση θα καταλάβει η υποδιαστολή, 2 θέσεις θα καταλάβουν τα δυο δεκαδικά ψηφία και μένουν και 5 θέσεις για το ακέραιο μέρος του αριθμού. Αν το ακέραιο μέρος του αριθμού αποτελείται από π.χ. 1 ψηφίο, τότε τα υπόλοιπα 4 ψηφία θα εμφανιστούν ως κενά αριστερά του αριθμού. Για τον αριθμό 2.4598, με το προσδιοριστικό μετατροπής %8.2f:

```
>> disp(sprintf('%8.2f', 2.4598))
θα εμφανιστεί:
     2.46
Παρατηρούμε ότι έγινε και στρογγυλοποίηση, δεν εμφανίστηκε 2.45.
Η sprintf () συντάσσεται με περίπου τον ίδιο τρόπο όπως η fprintf ().
>> disp(sprintf('The 10th element of x is x(10) = %6.3f', x(10)))
>> disp(sprintf('The 10th element of x is x(10) = \$10.3f', x(10)))
>> disp(sprintf('The 10th element of x is x(10) = \$12.3f', x(10)))
>> disp(sprintf('The 10th element of x is x(10) = \$8.3e', x(10)))
>> disp(sprintf('The 10th element of x is x(10) = \$8.3e', x(10)))
>> disp(sprintf('The 10th element of x is x(10) = \$8.12e', x(10)))
>> disp(sprintf('The 10th element of x is x(10) = \$24.15e', x(10)))
>> disp(sprintf('The 10th element of x is x(10) = \$1.4g', x(10)))
>> disp(sprintf('The 10th element of x is x(10) = \$16.12q', x(10)))
>> disp(sprintf('The 10th element of x is x(10) = \$24.15g', x(10)))
>> format short
>> x
```

#### Σχεσιακοί τελεστές

### Λογικοί τελεστές

Χρησιμοποιούνται για να σχηματίσουν λογικές εκφράσεις. Π.χ. αν θέλουμε να ελέγξουμε αν ισχύουν ταυτόχρονα δυο ανισότητες/ισότητες:

ή αν ισχύει η αντίθετή της:

>>  $\sim$  (a == b) & (c > d)

#### for

```
>> for i=1:5
i
end
>> for i=1:5
disp(sprintf('Aυτή είναι η %dη επανάληψη',i))
end
>> for i=1:5
disp(sprintf('Aυτή είναι η %dη επανάληψη',i))
end
>> k=10;
for i=1:5
k=k+2
end
```

# if / then / else / elseif

```
>> a = 5; b = -2; c = 5; d = 2;
>> if(a>b)
а
end
>> if(a<b)
а
end
>> if(a<b)
b
end
>> if(a>b)
а
else b
end
>> if(a>c)
а
elseif(a<c)</pre>
С
elseif(a==c)
a, c
else
d
end
>> if(a+1>c)
а
elseif(a<c)</pre>
С
elseif(a==c)
a, c
else
d
end
```

Παρατηρήστε σε αυτή την περίπτωση ότι ισχύει και η έκφραση a+1>c και η έκφραση a=c. Όμως από τη στιγμή που πρώτα έχουμε γράψει την a+1>c, και αφού τη βρήκε αληθή, εκτελεί την εντολή που αντιστοιχεί σε αυτή τη συνθήκη και μετά τερματίζει, δεν ελέγχει τις υπόλοιπες συνθήκες. Οι υπόλοιπες θα ελεγχθούν μόνο αν η πρώτη έκφραση είναι ψευδής. Και ακόμα και αυτές θα ελεγχθούν διαδοχικά. Δηλαδή πρώτα η a<c. Αν αυτή ισχύει, δεν θα ελεγχθεί η επόμενη, a==c.

#### while

```
>> k=1; while(k<10) disp(sprintf('Autń είναι η %dη επανάληψη του while',k)), k=k+1; end
```

Είναι σημαντικό να εξασφαλίσουμε ότι θα αποφύγουμε τις ατέρμονες επαναλήψεις. Στη συγκεκριμένη περίπτωση, αυτό γίνεται γιατί έχουμε ην έκφραση k=k+1, συνεπώς το k αυξάνει κατά 1 σε κάθε επανάληψη και σίγουρα θα φτάσει να πάρει την τιμή 10, ώστε να τερματίσει το while. Στην επόμενη περίπτωση όμως:

>> k=1; while(k<10) disp(sprintf('Autń είναι η %dη επανάληψη του while',k)), end

Δεν υπάρχει η έκφραση k=k+1, άρα το k θα έχει πάντα την τιμή 1, και οι επαναλήψεις δεν θα τερματίσουν ποτέ. Σε τέτοιες περιπτώσεις, μπορούμε να σταματήσουμε απότομα το "τρέξιμο" με το συνδυασμό των πλήκτρων "Ctrl" και "C".

### Άλλες συναρτήσεις

```
>> z = randi(5,1,10), w = randi(5,3,10)
>> sum(z)
                      % άθροισμα
>> sum(w,1)
>> sum(w,2)
>> sum(z, 1)
>> sum(z, 2)
>> sum([4 4 2])
>> sum([4 4 2],1)
>> sum([4 4 2],2)
>> sum([4 4 2 z])
>> mean(z)
                      % μέσος όρος
>> mean(w)
>> mean(w,1)
>> mean(w, 2)
>> mean([4 4 2])
>> mean([4 4 2],1)
>> mean([4 4 2],2)
>> mean([4 4 2 z])
```

# Γραφήματα (plot)

Μπορείτε να γράψετε τον παρακάτω κώδικα στο Command Window, γραμμήγραμμή (στο τέλος κάθε γραμμής πατάτε Enter) ή μπορείτε να το γράψετε ολόκληρο (χωρίς τα >>) στον Editor (καρτέλα EDITOR ► New ► Script), να το αποθηκεύσετε με ένα όνομα (καρτέλα EDITOR ► Save ► Save As...) και να το τρέξετε με το Run (καρτέλα EDITOR ► κουμπί Run).

```
>> x = linspace(0,1);
>> y = 2 - x + x.^3/2 - x.^5/7;
>> expo = exp(x);
>> figure; % δημιουργεί ένα παράθυρο για να φιλοξενήσει το γράφημα
>> plot(x,y);
>> figure % δημιουργεί ένα νέο παράθυρο
>> plot(x,expo);
>> hold on % ό,τι γράφημα δημιουργηθεί μετά το hold on, θα
% εμφανιστεί στο ίδιο παράθυρο με το προηγούμενο
% γράφημα (πληροφορίες: hold)
>> plot(x,y,'r-.'); % δείτε τη σύνταξη: plot
>> hold off % τερματίζει την ενέργεια που ξεκίνησε το hold on
```