Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας

Τμήμα Ηλεκτρονικών Μηχανικών

ΕΙΣΑΓΩΓΙΚΕΣ ΣΗΜΕΙΩΣΕΙΣ ΓΙΑ ΤΟ MATLAB

Επιμέλεια:
Αλεξανδρίδης Αλέξανδρος
Αναπληρωτής Καθηγητής ΤΕΙ Αθήνας

Σαρίμβεης Χαράλαμπος
Αναπληρωτής Καθηγητής ΕΜΠ

MATLAB: MATrix LABoratory, δυναμικό περιβάλλον για επιστημονικούς και αριθμητικούς υπολογισμούς. Βασίζεται στην χρήση πινάκων τα στοιχεία των οποίων μπορεί να είναι πραγματικοί ή μιγαδικοί αριθμοί. Ακόμα και μια απλή ποσότητα (scalar) θεωρείται ως πίνακας με ένα στοιχείο.

ΒΑΣΙΚΕΣ ΓΝΩΣΕΙΣ

Ανοίγοντας το MATLAB εμφανίζονται, ο χώρος εργασίας και διάφορα άλλα βοηθητικά παράθυρα (ιστορικό εντολών, μεταβλητές που έχουν σωθεί στον χώρο εργασίας και το directory στο οποίο εργαζόμαστε). Όλες οι εργασίες που θα παρουσιαστούν, εκτός από την κατασκευή προγραμμάτων και το SIMULINK πραγματοποιούνται στον χώρο εργασίας.

Ορισμός μεταβλητών

```
>> a=1;
>> b=2;
>> A=3;
```

Όταν τοποθετείται ερωτηματικό δίπλα από μία παράσταση, το αποτέλεσμα δεν παρουσιάζεται στον χώρο εργασίας, αλλά αποθηκεύεται απλώς για μελλοντικές πράξεις.

Στο MATLAB είναι σημαντικό το αν ορίζεται μία μεταβλητή με μικρά ή κεφαλαία. Αυτό σημαίνει ότι **η μεταβλητή A είναι διαφορετική από την a**.

Βασικές πράξεις: +, -, *, /, ^

```
>> c=a+b; (πρόσθεση)
>> c=A*b; (πολλαπλασιασμός)
>> c=A^b; (δύναμη του αριθμού A)
```

Βασικές εντολές

>> who (Παρουσιάζει τις μεταβλητές που είναι αποθηκευμένες στον χώρο εργασίας)

>> whos (Παρουσιάζει τις μεταβλητές που είναι αποθηκευμένες στον χώρο εργασίας και ορισμένα στοιχεία για τις μεταβλητές αυτές)

>> clear a (διαγράφει την μεταβλητή a)

>> clear (διαγράφει όλες τις μεταβλητές από τον χώρο εργασίας)

>> help + όνομα εντολής (παρουσιάζει λεπτομερή βοήθεια για την συγκεκριμένη εντολή)

Ορισμός πινάκων και διανυσμάτων

Ορισμός διανύσματος (γραμμή)
>> v=[1 2 3];

Ορισμός διανύσματος (στήλη)
>> v=[1
2
3]; ή

>> v=[1;2;3];

Ορισμός πίνακα 3x3
>> A=[1 2 3
4 5 6
2 4 9]; ή
>> A=[1 2 3;4 5 6;2 4 9];

Άλλοι τρόποι ορισμού διανυσμάτων

>> v=[1:5]; (Παράγει το διάνυσμα-ακολουθία από 1έως 5 με βήμα 1)
v =[1 2 3 4 5]

>> v=[1:2:7]; (Παράγει το διάνυσμα-ακολουθία από 1 έως 7 με βήμα 2)
v =[1 3 5 7]

>> v=linspace(0,10,5); (Παράγει διάνυσμα 5 στοιχείων ξεκινώντας από το 0, καταλήγοντας στο 10, διατηρώντας ίσα διαστήματα)
v=[0 2.5000 5.0000 7.5000 10.0000]

>> v=logspace(0,4,5); (Παράγει διάνυσμα 5 αριθμών ξεκινώντας από το 10^0 και καταλήγοντας στο $10^{4)}$

v =[1 10 100 1000 10000]

Εξειδικευμένοι πίνακες

>> I=eye(5); (Μοναδιαίος πίνακας διάστασης 5)
>> Z=zeros(5,3); (Μηδενικός πίνακας 5 γραμμών και 3 στηλών)

>> D=diag([1 4 6]) (Παράγει διαγώνιο πίνακα 3x3 όπως φαίνεται παρακάτω)

```
D =

     1     0     0
     0     4     0
     0     0     6
```

>> U=ones(2,3) (Παράγει πίνακα 2x3 με όλα τα στοιχεία μονάδα όπως φαίνεται παρακάτω)

```
U =

     1     1     1
     1     1     1
```

Διαχείριση πινάκων

Πράξεις πινάκων

Οι πράξεις είναι οι εξής:
>> C=A+B; (Άθροισμα πινάκων)
>> C=A-B; (Διαφορά πινάκων)
>> D=A*B; (Πολλαπλασιασμός πινάκων)
>> G=A/B; (Πολλαπλασιασμός του A με τον αντίστροφο του B)
>> x=A\b; (Επιλύει το γραμμικό σύστημα **A*x=b**)
>> B=A^2; (Βρίσκει την $2^{η}$ δύναμη του A. Εφαρμόζεται μόνο για τετραγωνικούς πίνακες)
>> B=A+1; (Προστίθεται σε κάθε στοιχείο του A ο αριθμός 1)
>> B=2*A; (Πολλαπλασιασμός αριθμού επί πίνακα)

Όταν πριν από το σύμβολο οποιασδήποτε πράξης μεταξύ πινάκων προηγείται τελεία, τότε αυτή δεν γίνεται μεταξύ των πινάκων, αλλά μεταξύ των στοιχείων τους. Π.χ.

>> B=A.^3; (Υψώνεται κάθε στοιχείο του A στην $3^{η}$ δύναμη)
>> C=A.*B; (Πολλαπλασιάζεται κάθε στοιχείο του A με το αντίστοιχο στοιχείο του B)
>> C=A./B; (Διαιρείται το κάθε στοιχείο του A με το αντίστοιχο στοιχείο του B)

Δείκτες στοιχείων πινάκων και διαστάσεις

>> n=size(A); (Δίνει τις διαστάσεις του πίνακα A)
>> n=length(v); (Δίνει το μήκος του διανύσματος v)

>> a=A(3,5); (Επιστρέφει το στοιχείο της $3^{ης}$ γραμμής και της $5^{ης}$ στήλης του πίνακα Α)
>> a=A(2,:); (Επιστρέφει την $2^{η}$ γραμμή του πίνακα Α)
>> a=A(:,3); (Επιστρέφει την $3^{η}$ στήλη του πίνακα Α)
>> a=A(2:4,1:3); (Επιστρέφει πίνακα ο οποίος αποτελείται από τις γραμμές 2 έως 4 και τις στήλες 1 έως 3 όπως φαίνεται παρακάτω)

A =

1	2	3	1
2	3	4	2
3	4	5	3
2	4	5	2

>> a=A(2:4,1:3) (Επιστρέφει τα στοιχεία των γραμμών 2 έως 4 και των στηλών 1 έως 3)

a =

2	3	4
3	4	5
2	4	5

>> a=A(:,2:end) (Για τον πίνακα Α που παρουσιάστηκε παραπάνω επιστρέφει τα στοιχεία όλων των γραμμών και των στηλών από την $2^{η}$ μέχρι την τελευταία)

a =

2	3	1
3	4	2
4	5	3
4	5	2

Βασικές εντολές πινάκων και διανυσμάτων

>> B=A' (Επιστρέφει τον ανάστροφο του Α)
>> B=inv(A); (Επιστρέφει τον αντίστροφο του Α)
>> a=diag(A); (Επιστρέφει διάνυσμα με τα διαγώνια στοιχεία του πίνακα Α)
>> s=eig(A); (Επιστρέφει διάνυσμα με τις ιδιοτιμές του πίνακα Α)
>> s=max(v); (Επιστρέφει το μέγιστο στοιχείο του διανύσματος v)
>> s=min(v); (Επιστρέφει το ελάχιστο στοιχείο του διανύσματος v)
>> s=norm(v); (Επιστρέφει την ευκλείδια νόρμα του διανύσματος v)
>> s=sort(v); (Διατάσσει τα στοιχεία του v κατά αύξουσα σειρά)

Έτοιμες συναρτήσεις του MATLAB

Ο πίνακας που ακολουθεί περιλαμβάνει τις βασικές μαθηματικές συναρτήσεις του MATLAB:

abs(x)	Απόλυτη τιμή
sqrt(x)	Τετρ. ριζα
exp(x)	e^x
log(x)	ln(x)
log10(x)	$\log_{10}(x)$
sin(x)	Ημίτονο
cos(x)	Συνημίτονο
tan(x)	Εφαπτομένη
cot(x)	Συνεφαπτομένη
asin(x)	Αντίστροφο ημίτονο
acos(x)	Αντίστροφο συνημίτονο
atan(x)	Αντίστροφη εφαπτομένη
acot(x)	Αντίστροφη συνεφαπτομένη
sinh(x)	Υπερβολικό ημίτονο
cosh(x)	Υπερβολικό συνημίτονο
tanh(x)	Υπερβολική εφαπτομένη
coth(x)	Υπερβολική συνεφαπτομένη
asinh(x)	Αντίστροφο υπερβολικό ημίτονο
acosh(x)	Αντίστροφο υπερβολικό συνημίτονο
atanh(x)	Αντίστροφη υπερβολική εφαπτομένη
acoth(x)	Αντίστροφη υπερβολική συνεφαπτομένη

Άλλες χρήσιμες συναρτήσεις και εντολές

Πολυώνυμα

Οι συντελεστές του πολυωνύμου π.χ. του $P(x) = x^4 + 3x^3 - 15x^2 - 2x + 9$ αποθηκεύονται στην παρακάτω array:

```
>> p=[1 3 -15 -2 9];
```

Ακολούθως υπάρχουν οι παρακάτω δυνατότητες:

```
>> y=polyval(p,2); (Υπολογίζει την τιμή του πολυωνύμου P(x) στο σημείο x=2)
>> x=roots(p); (Υπολογίζει τις ρίζες του πολυωνύμου)
>> conv(p,q); (Πολλαπλασιάζει πολυώνυμα)
>> deconv(p,q); (Διαιρεί πολυώνυμα)
>> polyfit(x,y,n); (Βρίσκει τους συντελεστές του πολυωνύμου n-οστού βαθμού το οποίο προσαρμόζεται καλύτερα στα δεδομένα x και y)
>> poly([1 2 -1]); (Επιστρέφει το πολυώνυμο το οποίο έχει τις ρίζες 1, 2, -1)
```

Εντολή solve

Η εντολή solve χρησιμεύει για την αναλυτική ή αριθμητική επίλυση εξισώσεων ή συστημάτων εξισώσεων: Π.χ.
>> x=solve('p*sin(x) = r') (Επιλύει ως προς x και αντιμετωπίζει τα p και r σαν παραμέτρους, οπότε παράγει το παρακάτω αποτέλεσμα)

x =

asin(r/p)

>> [x,y] = solve('sin(x+y)-exp(x)*y = 0','x^2-y = 2') (Επιλύει το σύστημα ως προς x και y. Καθώς δεν υπάρχει αναλυτική λύση, υπολογίζει την αριθμητική λύση)

x =

-6.017327250059306564109729711790 5

y =

34.208227234306296508646214438330

Εντολή dsolve

Χρησιμεύει για αναλυτική επίλυση (όταν αυτή είναι εφικτή) διαφορικών εξισώσεων. Π.χ. για την εξίσωση: $\frac{dx}{dt} = -ax$

>> y=dsolve('Dx = -a*x') (Με το πρόθεμα D δηλώνεται η πρώτη παράγωγος. Εάν αντί της πρώτης παραγώγου υπήρχε $2^{η}$, τότε θα ήταν D2y κτλ. Η εξίσωση λύνεται αναλυτικά και παράγεται το παρακάτω αποτέλεσμα)

y =

C1*exp(-a*t)

όπου C1 σταθερά

>> [f,g] = dsolve('Df = f + g','Dg = -f + g','f(0) = 1','g(0) = 2') (Επιλύει αναλυτικά το σύστημα των διαφορικών εξισώσεων με αρχικές τιμές)

f =

exp(t)*(cos(t)+2*sin(t))

g =

exp(t)*(-sin(t)+2*cos(t))

Αναλυτική παραγώγιση συνάρτησης

Προτού γίνει η παραγώγιση πρέπει οι μεταβλητές της συνάρτησης να δηλωθούν ως σύμβολα. Αυτό γίνεται με την εντολή syms.

>> syms x f

Ακολούθως δηλώνεται η συνάρτηση:

>> f=x^2+3*x

Η παραγώγιση γίνεται με την εντολή diff ως εξής:

>> diff(f)

ans =

2*x+3

Εάν αντί της πρώτης παραγώγου επιθυμούσαμε την δεύτερη, η εντολή θα γραφόταν: diff(f,2) κτλ.

Μιγαδικοί αριθμοί
>> z=1+3i (Ορισμός μιγαδικού αριθμού)

z =

1.0000 + 3.0000i

>> real(z) (Πραγματικό μέρος του z)
>> imag(z) (φανταστικό μέρος του z)

Εντολές load, xlsread, save

Χρησιμοποιούνται για την επικοινωνία του MATLAB με άλλα προγράμματα όπως π.χ. το excel.
>> A=xlsread('filename'); (αποθηκεύει όλα τα αριθμητικά δεδομένα του αρχείου excel στην μεταβλητή A)
>> save όνομα αρχείου x y z -ascii (Αποθηκεύει τις μεταβλητές x, y, z σε αρχείο txt)
>> s=load('όνομα αρχείου') (αποθηκεύει τα αριθμητικά δεδομένα του αρχείου στην μεταβλητή s)

ΓΡΑΦΙΚΕΣ ΠΑΡΑΣΤΑΣΕΙΣ

Γραφικές Παραστάσεις σε Δυο Διαστάσεις

Η πιο συνηθισμένη εντολή για την χάραξη γραφημάτων σε δυο διαστάσεις στο Matlab είναι η εντολή plot. Η σύνταξη της εντολής plot είναι η εξής:
Plot(x,y,'LineSpecs')
όπου *x* και *y* είναι διανύσματα με τις τιμές των μεταβλητών που θέλουμε να σχεδιάσουμε, ενώ στο σημείο που γράφουμε *LineSpecs* ορίζουμε τις ιδιότητες της γραμμής της γραφικής παράστασης.

Παράδειγμα Ι

Το παρακάτω πρόγραμμα παράγει την γραφική παράσταση της ημιτονοειδούς συνάρτησης y=sin(x) για μια περίοδο.

```
theta=[0:pi/128:2*pi]';
y=sin(theta);
plot(theta,y,'b-');
xlabel('Angle in Radians');
ylabel('Amplitude');
title('One Period of a Sine Wave');
```

Παράδειγμα ΙΙ

Σε αυτό το παράδειγμα η εντολή plot χρησιμοποιείται για την ταυτόχρονη απεικόνιση δύο συναρτήσεων στην ίδια γραφική παράσταση.

```
theta=[0:pi/128:2*pi]';
y1=sin(theta);
y2=cos(theta);
plot(theta,y1,'b-',theta,y2, 'r-');
xlabel('Angle in Radians');
ylabel('Amplitude');
title('One Period of a Sine and a Cosine Wave');
legend('Sine','Cosine')
```

Παράδειγμα III

Στο παρακάτω παράδειγμα απεικονίζεται η συνάρτηση y=exp(-x) σε διαφορετικούς τύπους αξόνων.

```
x=[0:0.1:10];
y=exp(-x);
subplot(2,2,1);
plot(x,y)
title('Regular Plot')
subplot(2,2,2);
semilogy(x,y)
title('Semilog y')
subplot(2,2,3);
semilogx(x,y)
title('Semilog x')
subplot(2,2,4);
loglog(x,y)
title('Loglog')
```

Παράδειγμα IV

Το παράδειγμα αυτό δείχνει την χρήση της εντολής pie.

```
a=[0.5, 1, 1.6, 1.2, 0.8, 2.1];
pie(a,a==max(a));
```

Παράδειγμα V

Το παράδειγμα αυτό δείχνει την χρήση της εντολής bar.

```
x=[-2.9:0.2:2.9];
y=exp(-x.*x);
bar(x,y);
```

Παράδειγμα VI

Το παράδειγμα αυτό δείχνει την χρήση της εντολής ezplot η οποία παράγει την γραφική παράσταση συναρτήσεων που εισάγονται με συμβολικό τρόπο.

```
f='sin(2*x)*exp(-x)';xmin=0;xmax=2*pi;
ezplot(f,[xmin,xmax]);
```

Γραφικές Παραστάσεις σε Τρεις Διαστάσεις

Για τρισδιάστατες γραφικές παραστάσεις χρησιμοποιούνται συνήθως οι εντολές mesh (γραφική παράσταση πλέγματος) και surf (γραφική παράσταση επιφάνειας):

Παράδειγμα VII

Στο παράδειγμα αυτό σχεδιάζουμε το πλέγμα σε τρεις διαστάσεις για την συνάρτηση

$z = f(x, y) = xe^{-(x^2+y^2)}$

```
x =[-2:0.2:2];
y =[-2:0.2:2];
[xx,yy]=meshgrid(x,y);
z=xx.*exp(-xx.^2-yy.^2);
mesh(xx,yy,z)
title('Mesh Plot');
xlabel('X Data');
ylabel('Y Data');
zlabel('z=xx.*exp(-xx.^2-yy.^2)');
```

στο παραπάνω παράδειγμα με:

- την surf(xx,yy,z), παίρνουμε την γραφική παράσταση επιφάνειαςτην cs=contour(xx,yy,z,16), παίρνουμε την γραφική παράσταση ισοϋψών (όπου 16 ο αριθμός των ισοϋψών)
- την meshc(xx,yy,z), παίρνουμε ταυτόχρονα την γραφική παράσταση πλέγματος και ισοϋψών

ΑΡΙΘΜΗΤΙΚΗ ΕΠΙΛΥΣΗ ΔΙΑΦΟΡΙΚΩΝ ΕΞΙΣΩΣΕΩΝ ΜΕ ΤΟ MATLAB

Επίλυση προβλημάτων με αρχικές συνθήκες

Η γενική σύνταξη της εντολής που επιλύει τέτοιου τύπου προβλήματα είναι:
[t,y]=solver(odefun,tspan,y0,options)

όπου solver είναι η αριθμητική μέθοδος που χρησιμοποιεί το matlab. Πιο συγκεκριμένα στην θέση solver μπορεί να χρησιμοποιηθεί μια από τις παρακάτω επιλογές:

- ode45
- ode23
- ode113
- ode15s
- ode23s
- ode23t
- ode23tb

Με odefun συμβολίζουμε το όνομα της συνάρτησης που υπολογίζει το δεξιό μέρος της διαφορικής εξίσωσης. Tspan είναι ένα διάνυσμα που ορίζει το χρονικό διάστημα της

ολοκλήρωσης. Το y0 είναι το διάνυσμα στο οποίο περιέχονται οι αρχικές συνθήκες και options είναι οι παράμετροι που μπορούμε να επιλέξουμε χρησιμοποιώντας την συνάρτηση odeset.

Παράδειγμα
Να επιλυθεί το παρακάτω σύστημα διαφορικών εξισώσεων:

$$\dot{y}_1 = y_2 y_3, \qquad y_1(0) = 0$$
$$\dot{y}_2 = -y_1 y_3, \qquad y_2(0) = 1$$
$$\dot{y}_3 = -0.51 y_1 y_2, \qquad y_3(0) = 1$$

Καταρχήν κατασκευάζουμε μια συνάρτηση με όνομα rigid που περιέχει το δεξιό μέλος των διαφορικών εξισώσεων:

```
function dy=rigid(t,y)
dy=zeros(3,1);
dy(1)=y(2)*y(3);
dy(2)=-y(1)*y(3);
dy(3)=-0.51*y(1)*y(2);
```

Στη συνέχεια μπορούμε να αλλάξουμε κάποιες από τις παραμέτρους επίλυσης χρησιμοποιώντας την συνάρτηση odeset ως εξής:

options=odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4 1e-5]);

Η επίλυση επιτυγχάνεται με την εντολή:

[T,Y]=ode45(@rigid,[0 12],[0 1 1],options);

Μπορούμε να δούμε τα αποτελέσματα με χρήση της εντολής plot:

plot(T,Y(:,1),'-',T,Y(:,2),'-',T,Y(:,3),'-')

Άσκηση

Θεωρείστε την παρακάτω διαφορική εξίσωση δεύτερης τάξης
$\ddot{y} - (1 - y^2)\dot{y} + y = 0 \qquad y(0) = 2, \quad \dot{y}(0) = 0$

Γράψτε μια συνάρτηση σε αρχείο *.m για το δεξιό μέρος της διαφορικής εξίσωσης. Χρησιμοποιείστε την ρουτίνα ode45 για να λύσετε την διαφορική εξίσωση με αριθμητικό τρόπο στο διάστημα [0 20].

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΟ MATLAB

Λογικές πράξεις στις εντολές ελέγχου ροής
& : AND
| : OR
~ : NOT

Τελεστές συσχέτισης
= = : equal
~ = : not equal
> : greater than
>= : greater than or equal
< : less than
<= : less than or equal

Η ΕΝΤΟΛΗ FOR

Χρήση: Χρησιμοποιείται για την επανάληψη ενός μπλοκ εντολών, για διάφορες τιμές ενός δείκτη *i*

Μορφή:
```
for i=a:b:c
        μπλοκ εντολών
end
```

όπου *a* αρχική τιμή του δείκτη *i*, *b* το βήμα αύξησης και *c* η τελική τιμή του (αν παραλείψουμε το βήμα, θεωρείται *b*=1).

Παράδειγμα I
Δημιουργήστε ένα διάνυσμα x=[1^2 2^2 3^2 4^2]

```
for i=1:4
        x(i) = i*i;
end
```

Πολλαπλοί βρόγχοι for (Nested for loops)
Μπορούν να δημιουργηθούν πολλοί βρόγχοι εντολών for, ο ένας μέσα στον άλλον.

Παράδειγμα II
Κάθε στοιχείο ενός πίνακα Hilbert διαστάσεως m x n δίνεται από τον τύπο: $a(i,j)=\frac{1}{i+j-1}$, *i*=1,2,…,*m* και *j=1,2,...,n*. Κατασκευάστε έναν πίνακα Hilbert 4 x 5.

```
m = 4;n = 5;
for i=1:m
    for j=1:n
      a(i,j)=1/(i+j-1);
    end
end
```

Η ΕΝΤΟΛΗ WHILE

Χρήση: Χρησιμοποιείται για την επανάληψη ενός μπλοκ εντολών, όσο ισχύει μια συγκεκριμένη συνθήκη

Μορφή:
```
while  συνθήκη
       μπλοκ εντολών
end
```

Παράδειγμα ΙΙΙ

Η ανάπτυξη σε σειρές MacLaurin της συνάρτησης ln(1+x), για $|x| < 1$, δίνεται από τον τύπο:

$$\ln(1+x) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{x^k}{k}$$

Να εκτιμήσετε την τιμή του ln(1+x) για x=0.5, αθροίζοντας διαδοχικούς όρους των σειρών, μέχρι εκείνο τον όρο που η απόλυτη τιμή του θα είναι μικρότερη από $2*10^{-6}$.

```
v=0; x=0.5; k=1;
while abs((x^k)/k) >= 2*10^-6
        v = v+(-1)^(k+1)*((x^k)/k);
        k = k+1;
end
```

Η ΕΝΤΟΛΗ IF

Χρήση: Χρησιμοποιείται για την εκτέλεση εντολών κατά συνθήκη

Μορφή:
```
if  συνθήκη
        μπλοκ εντολών
elseif  συνθήκη
        μπλοκ εντολών
else
        μπλοκ εντολών
end
```

Η ΕΝΤΟΛΗ BREAK

Χρήση: Χρησιμοποιείται για την έξοδο από ένα for ή while loop.

Παράδειγμα IV
Το Matlab έχει ένα ελάχιστο όριο ακρίβειας το οποίο ονομάζεται EPS (floating point relative accuracy constant) και ισχύει eps = $2.2204*10^{-16}$. Κατασκευάστε ένα πρόγραμμα Matlab που να υπολογίζει την τιμή του eps.

```
EPS=1;
for i=1:1000
        EPS=EPS/2;
        if(1+EPS)==1
                EPS=EPS*2;
                break;
        end
end
```

Η ΕΝΤΟΛΗ SWITCH

Χρήση: Χρησιμοποιείται όταν μια σειρά από εντολές πρέπει να εκτελούνται ελεγχόμενα, με βάση μια συνθήκη ελέγχου ισότητας μιας παράστασης με έναν σταθερό όρο.

Μορφή:
switch *τιμή αριθμητικής παράστασης*
case *σταθερός όρος 1*
μπλοκ εντολών
case *σταθερός όρος 2*
μπλοκ εντολών
otherwise
μπλοκ εντολών
end

Παράδειγμα V
Γράψτε ένα πρόγραμμα Matlab που να βρίσκει αν ένας θετικός αριθμός *n*, είναι πολλαπλάσιο του 3, αν διαιρείται με το 3 με υπόλοιπο 1 ή αν διαιρείται με το 3 με υπόλοιπο 2.

```
if n<=0
  'No positive number'
else
  switch rem(n,3)
  case 0
    'Multiple of 3'
  case 1
    'Multiple of 3+1'
  otherwise
    'Multiple of 3+2'
  end
end
```

ΠΡΟΓΡΑΜΜΑΤΑ ΤΥΠΟΥ SCRIPT ΚΑΙ FUNCTION

Για να ανοίξει κανείς τον editor της γλώσσας προγραμματισμού του Matlab χρησιμοποιεί την εντολή edit. Τα αρχεία που περιέχουν τον κώδικα της γλώσσας έχουν extension .m και ονομάζονται M-Files. Η γλώσσα προγραμματισμού του Matlab δεν απαιτεί ξεχωριστή διαδικασία compiling, αλλά το πρόγραμμα εκτελείται κατευθείαν πληκτρολογώντας το όνομα του από το Command Window.
Τα αρχεία απαρτίζονται από κανονικές εντολές του Matlab και μπορούν να περιέχουν αναφορές σε άλλα M-Files.
Χωρίζονται σε Script αρχεία και Function αρχεία.

Τα Script περιέχουν απλώς μια σειρά εντολών. Οι εντολές στα script λειτουργούν σφαιρικά στα δεδομένα που υπάρχουν στο χώρο εργασίας.
Το παρακάτω παράδειγμα είναι ένα script πρόγραμμα που υπολογίζει το άθροισμα των n πρώτων στοιχείων της σειράς $1^2+2^2+...+n^2$ για n=5:

```
n=5;
s=0;
for i=1:n
   s=s+i^2;
end
```

Με τα function αρχεία μπορούμε να προσθέσουμε δικές μας συναρτήσεις στις ήδη έτοιμες συναρτήσεις του Matlab. Οι βασικές διαφορές των function από τα script είναι:

Α) Μπορούν να τρέξουν για πολλές τιμές των μεταβλητών χωρίς να χρειαστεί αλλαγή του προγράμματος

Β) Οι μεταβλητές που ορίζουμε και χειριζόμαστε μέσα στο function αρχείο είναι τοπικές και δεν λειτουργούν σφαιρικά στο χώρο εργασίας του Matlab.

Το παράδειγμα script αρχείου που εξετάσαμε προηγουμένως μπορεί να γραφτεί ως function με τον παρακάτω τρόπο:

```
function s=sumsq(n)
s=0;
for i=1:n
   s=s+i^2;
end
```

Στη συνέχεια καλώντας το αρχείο function από την γραμμη εντολών ως sumsq(5) μπορούμε να υπολογίσουμε το άθροισμα των 5 πρώτων όρων της σειράς.

Άσκηση

Κατασκευάστε πρόγραμμα τύπου function το οποίο να υπολογίζει το n!.