ΣΗΜΕΙΩΣΕΙΣ ΕΡΓΑΣΤΗΡΙΟΥ ΑΡΙΘΜΗΤΙΚΗΣ ΑΝΑΛΥΣΗΣ ΙΙ

ΕΠΙΜΕΛΕΙΑ: ΔΗΜΗΤΡΑ ΑΝΤΩΝΟΠΟΥΛΟΥ

ΕΡΓΑΣΤΗΡΙΑ MATLAB

Πώς μπαίνουμε σε περιβάλλον matlab (command window) , κατασκευή μεταβλητών , διανυσμάτων , κάποιες βασικές εντολές .

1) ΜΕΤΑΒΛΗΤΕΣ

Στην matlab οι μεταβλητές δεν δηλώνονται από τον χρήστη , κατασκευάζονται όμως σταδιακά . Στην πραγματικότητα αυτή η γλώσσα προγραμματισμού καταλαβαίνει όλες τις μεταβλητές σαν 1x 1 μιγαδικό πίνακα .

Παράδειγμα 1ο

```
>>x=10.1        {enter}
x =
   10.1000
>>
```

Σε αυτό το παράδειγμα κατασκευάσαμε έναν πραγματικό αριθμό x και η matlab μας έδωσε στην οθόνη την τιμή του με ακρίβεια τεσσάρων δεκαδικών . Αν στο παράδειγμα είχαμε πληκτρολογήσει

```
>>x=10.1;     {enter}
```

τότε στην οθόνη δεν θα εμφανιζόταν η καταχωρημένη τιμή . Γενικά η παρουσία ερωτηματικού στο τέλος μιας εντολής αποτίμησης μεταβλητής απαγορεύει την αυτόματη εμφάνιση της τιμής στην οθόνη .

Παράδειγμα 2ο (κατασκευή μονοδιάστατων μεταβλητών)

```
>>n=21;
>>h=n*pi ;
>>c=n+h      {enter}
c=
      86.9734
>>
```

Εδώ δώσαμε ακέραια τιμή στο n , άρρητη τιμή στο h (pi= π) , άθροισμα τιμών στο c .

Αν θέλουμε να δούμε την τιμή μιας μεταβλητής στην οθόνη , πληκτρολογούμε για παράδειγμα

```
>>c            {enter}            και στην οθόνη εμφανίζεται
c=
    86.9734
>>
```

2) ΔΙΑΝΥΣΜΑΤΑ

Τα διανύσματα κατασκευάζονται σταδιακά χωρίς να χρειάζεται να δηλώσουμε την εν γένει μεταλητή διαστασή τους .

Παράδειγμα 1ο (κατασκευή διανύσματος γραμμή)

```
>>y=[1.1 2.1];
```

Κατασκευάσαμε ένα διάνυσμα γραμμή .

Παράδειγμα 2ο (κατασκευή διανύσματος στήλη)

```
>>z=[2.3 7.1 6.2]';          ή          >>z=[2.3;7.1;6.2];
```

Κατασκευάσαμε ένα διάνυσμα στήλη .

Παράδειγμα 3ο (κατασκευή διανύσματος συντεταγμένη προς συντεταγμένη)

```
>>x(1)=1;
>>x(2)=2;
>>x
x=
      1 2
>>
```

Με αυτόν τον ορισμό η matlab καταλαβαίνει το διάνυσμα ως γραμμή .

Όταν αλλάξουμε την τιμή μιας συντεταγμένης σε ένα ήδη κατασκευασμένο διάνυσμα , τότε το διάνυσμα παραμένει ή γραμμή ή στήλη σύμφωνα με την αρχική κατασκευή του για παράδειγμα πληκτρολογούμε

```
>>r=[1.7 2.3]';
>>r(2)=3.7;
>>r                          και στην οθόνη εμφανίζεται
r=
      1.7000
      3.7000
>>
```

3) ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ-ΠΡΑΞΕΙΣ-ΠΙΝΑΚΕΣ

-Επαναληπτική εντολή 'for'

```
for k=l:m:n
    εντολές
end
```

Η επανάληψη με βήμα k , l<= k <=n , k=l+i*m , i=1,2,3…

Στη συνέχεια υπάρχουν κάποια παραδείγματα ορισμού διανυσμάτων με την 'for'

Παράδειγμα 1ο

```
for k=1:10            {το m εννοείται 1}
   y(k)=1;
end                   {ορίσαμε διάνυσμα γραμμή y μήκους 10}
```

Παράδειγμα 2ο

```
n=21;
h=1/(n-1);
for k=1:n;
   x(k)=(k-1)*h;
end
length(x)             {enter}
21                    {εμφανίστηκε στην οθόνη το μήκος του διανύσματος}
```

Παράδειγμα 3ο

```
n=5
for k=n:-1:1
   z(k)=k;
end
z                    {enter}
1 2 3 4 5    {εμφανίστηκε στην οθόνη το διάνυσμα γραμμή}
z=z'                  {enter}
 1
 2
 3
 4
 5              {εμφανίστηκε στην οθόνη  το ανάστροφο διάνυσμα}
```

-ΠΙΝΑΚΕΣ

Όπως γνωρίζουμε κάθε διάνυσμα στήλη είναι n x 1 πίνακας ενώ κάθε διάνυσμα γραμμή είναι 1 x n , συνεπώς μπορούμε μιλώντας γενικώς γιά πίνακες να συμπεριλάβουμε και τα διανύσματα.

Ορισμός πίνακα

π.χ.

```
        a=[1 2 3;4 5 6;7 8 9];
a            {enter}
1   2   3
4   5   6
7   8   9
```

Πράξεις πινάκων

+ ΠΡΟΣΘΕΣΗ

- ΑΦΑΙΡΕΣΗ

*ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΣ

^ΔΥΝΑΜΗ(π.χ. A^2=A*A)

‘ ΑΝΑΣΤΡΟΦΟΣ

\ ΑΡΙΣΤΕΡΗ ΔΙΑΙΡΕΣΗ (A\B=A^{-1}*B)

/ ΔΕΞΙΑ ΔΙΑΙΡΕΣΗ (A/B=A*B $^{-1}$)

Βασικές συναρτήσεις

$C^{nxm} \rightarrow C^{nxm}$

sin asin exp abs

cos asin log sqrt

tan atan

Παράδειγμα 1ο

x=2*pi;

y=sin(x);

y {enter}

0.0000

Παράδειγμα 2ο

for k=1:5

z(k)=k*pi;

end

y=sin(z)

z {enter}

0.0000 0.0000 0.0000 0.0000 0.0000

Παράδειγμα 3ο

x=[2*pi 3*pi; 4*pi 5*pi]; {πίνακας 2 x 2 }

cos(x) {enter}

1.0000 -1.0000

1.0000 -1.0000

ΥΠΟΠΙΝΑΚΕΣ

Έστω πίνακας A m x n διάστασης τότε ένας υποπίνακας ορίζεται ώς εξής

A(k:l,o:p) όπου 1<=k<=l<=m , 1<=o<=p<=n

Παράδειγμα 1ο

```
A=[1 2 3;4 5 6;7 8 9;10 11 12]
A=1    2    3
  4    5    6
  7    8    9
  10   11   12
A(2:3,2:3)
A=5   6
  8   9

A(2:3,2:3)=[0 0;0 0];
A=1   2    3
  4   0    0
  7   0    0
  10  11   12
```

-Υπάρχουν και κάποιες άλλες συναρτήσεις πινάκων χρήσιμες στην Αρ. Γρ. Αλγ.

det	ορίζουσα
size	διαστάσεις
rank	βαθμός

Παράδειγμα 1ο

```
A=[2 1;4 2]
A=2   1
  4   2

size(A);
rank(A)
  1
det(A)
  0.0000
```

-Η εντολή ‘x=linspace(a,b,n)’ ορίζει διάνυσμα διάστασης n όπου γιά 1<=i<=n , x(i)=a+(i-1)*(b-a)/(n-1)

Παράδειγμα 1ο

```
n=21;
x=linspace(0,1,n);
y=sin(2*pi*x);
```

Παράδειγμα 2ο

```
n=50;
x=linspace(-2.5,3.7,n);
y=linspace(8.1,9,n);
z=x+y;
```

-Δημιουργία τυχαίων πινάκων ,και πινάκων μηδενικών ή μονάδων

rand(m,n) τυχαίος πίνακας m x n

rand(n) τυχαίος πίνακας n x n

ones(m,n) πίνακας m x n με στοιχεία μονάδες

zeros(m,n) πίνακας m x n με στοιχεία μηδενικά

Παράδειγμα 1ο

```
m=50;
n=50;
A=rand(m,n);
det(A);
A=ones(m,n);

A=A^2;
```

Παράδειγμα 2ο

```
n=25;
h=1/(n-1);
x=zeros(1,n);
y=zeros(n,1);
```

```
for k=1:n
  x(k)=(k-1)*h;
  y(k)=x(k);
end
```

ΣΧΟΛΙΑΣΜΟΣ

Στο παραπάνω παράδειγμα με την εντολή zeros ουσιαστικά ορίζουμε την διάσταση ενός διανύσματος γραμμή και ενός διανύσματος στήλη (τα μηδενικά δεν ενδιαφέρουν αφού αντικαθίστανται),αυτή η διαδικασία είναι χρήσιμη για να έχουμε δομημένο προγραμματισμό.

-Η εντολή disp

disp('οποιαδήποτε φράση ') ή disp(μεταβλητή)

Παράδειγμα

```
n=5;
x=linspace(0,1,n);
x=x';
y=exp(x);
disp('    το x είναι     ')
disp(x)
```

-Εντολές (που χρησιμοποιούνται στην κατασκευή γραφικών παραστάσεων) 'axis' , 'title' , 'xlabel' , 'ylabel' , 'plot' .

Γενικά η matlab χρησιμοποιεί σύστημα αξόνων με αρχή το (0,0) και x,y >= 0 . Συνήθως όμως στις εφαρμογές θέλουμε να δούμε ένα ορισμένο κομμάτι του επιπέδου . Αυτό γίνεται με την εντολή 'axis'

π.χ

axis([-5 5 -5 5])

Η παραπάνω εντολή μας δίνει την δυνατότητα να μπορούμε (με κατάλληλη βεβαίως εντολή) να σχεδιάσουμε σημεία (x,y) του επιπέδου , όπου -5<= x <=5 , -5<= y <=5 .

Ονομασία αξόνων και γραφικών παραστάσεων

π.χ

title('orbits')   {ονομάζει την γραφική παράσταση} xlabel('y1')   {ονομάζει τον άξονα των x}

ylabel('y2')   {ονομάζει τον άξονα των y}

Τα ορίσματα των πιο πάνω εντολών είναι χαρακτήρες .

Εντολή 'plot(x,y)'

x , y είναι διανύσματα ιδίας διάστασης

η παραπάνω εντολή σχεδιάζει στο επίπεδο τα σημεία

{(x(1),y(1)) , (x(2),y(2)) , ….., (x(n),y(n))} όπου n=dim x=dim y .

π.χ

plot([1.2 1.1 2.3],[1.4 2.1 2.9])

Όταν θέλουμε να έχουμε περισσότερες από μία γραφικές παραστάσεις στό ίδιο σύστημα αξόνων , χρησιμοποιούμε την εντολή 'hold on' ενώ για να σταματήσουμε την διαδικασία αυτή τυπώνουμε την εντολή 'hold off' .

ΠΑΡΑΔΕΙΓΜΑ 1

```
n=21;
x=linspace(0,1,n);
y=sin(2*pi*x);
plot(x,y)
title('The Function y=sin(2*pi*x)')
xlabel('x radians')
ylabel('y')
```

ΣΧΟΛΙΑ

Το παραπάνω πρόγραμμα στην πραγματικότητα κατασκευάζει πολυγωνική γραμμή

ΠΑΡΑΔΕΙΓΜΑ 2

Όπως το παράδειγμα 1 αλλά για n=200;

Το γράφημα είναι πιο λείο δεδομένου ότι ο διαμερισμός είναι πιο πυκνός

ΠΑΡΑΔΕΙΓΜΑ 3

```
n=40;
x=linspace(-pi/2,pi/2,n);
y=tan(x);
```

```
plot(x,y)
axis([-pi/2 9*pi/2 -10 10])
title('The Tangent Function')
xlabel('x')
ylabel('tan(x)')
hold on
 for k=1:4
      xnew=x+k*pi;
      plot(xnew,y);
 end
 hold off
```

ΣΧΟΛΙΑ

Εδώ σχεδιάζουμε την tan(x) στα διαστήματα (-κ*π/2,κ*π/2) για κ=1(1)4 στο ίδιο σύστημα κάνοντας ουσιαστικά μεταφορά κατά π της γραφικής παράστασης της tan(x) στο (-π/2,π/2), αφού η συνάρτηση είναι περιοδική με περίοδο π. Η χρήση της axis έγινε ώστε να χωράνε στην οθόνη όλα τα διαστήματα .

ΠΑΡΑΔΕΙΓΜΑ 4

```
n=200;
x=linspace(0,1,n);
y1=sin(2*pi*x);
y2=cos(2*pi*x);
plot(x,y1)
title('the sin and cos function in the same plot')
xlabel('x')
ylabel('sin(x),cos(x)')
hold on
plot(x,y2)
hold off
```

-Υπάρχει η δυνατότητα να σχεδιάσουμε συναρτήσεις f: $R \to R$,παρατηρώντας ότι η f(x)=A*VECTOR , εσωτερικό γινόμενο διανυσμάτων

π.χ

f: $R \to R$, f(x)=2sin(x)+3sin(2x)+7sin(3x)+5sin(4x)=

$$=[\sin(x)\ \sin(2x)\ \sin(3x)\ \sin(4x)]*\begin{bmatrix}2\\3\\7\\5\end{bmatrix}$$

κάνοντας διαμερισμό στο x το διάνυσμα με την μεταβλητή x μετατρέπεται σε πίνακα , παραθέτουμε ένα πρόγραμμα σχεδίασης της παραπάνω f

ΠΑΡΑΔΕΙΓΜΑ 5

```
n=200;
x=linspace(-10,10,n)';
A=[sin(x) sin(2*x) sin(3*x) sin(4*x)];
y=A*[2;3;5;7];
plot(x,y)
title('f(x)=2sin(x)+3sin(2x)+7sin(3x)+5sin(4x)')
xlabel('x')
ylabel('f(x)')
```

ΣΧΟΛΙΑ

- Ο A είναι n x n πίνακας ενώ το [2;3;5;7] είναι n x 1 διάνυσμα , το x είναι n x 1 διάνυσμα (του διαμερισμού).

-Στις επόμενες σελίδες είναι τα γραφήματα των παραδειγμάτων 1,3,4,5

-Ορισμός συναρτήσεων

π.χ

```
function f1=function1(t)
  f1=t*cos(t);
```
$\{f1:R \to R\}$

```
function f2=function2(y)
  f2=y(1)+y(2);
```
$\{f2: R^2 \to R\}$

```
function f3=function3(y)
  f3=[y(1)+y(2) y(2)]';
```
$\{f3: R^2 \to R^2$, το ' ορίζει διάνυσμα στήλη$\}$

```
function f4=function4(t,y)
  f4=[t*y(2) (t+1)*y(1)]';
```
$\{f4: R X R^2 \to R^2\}$

```
function ex6=example6(t,y)
     ex6=[-2*y(2) 2*y(1)]';

function ex7=example7(t,y)
   ex7=[4*y(1)]';
```

ΣΧΟΛΙΑ

Για τις εφαρμογές μας πρέπει να σώζουμε τις συναρτήσεις μας στο αρχείο bin (ως m.files) . Η διαδικασία που ακολουθούμε γιαυτό είναι η εξής:

1) Μπαίνουμε στη matlab .
2) $\rightarrow$ NEW $\rightarrow$ M.FILES $\rightarrow$ ορίζουμε συνάρτηση $\rightarrow$ SAVE AS $\rightarrow$ BIN $\rightarrow$ ετικέτα ,βάζουμε όνομα αρχείου π.χ function4.m $\rightarrow$ SAVE $\rightarrow [X] \rightarrow$ είμαστε στην matlab .

ΕΠΙΛΥΣΗ ΓΡΑΜΜΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Έστω το γραμμικό σύστημα

Ax=b , A nxn πίνακας , b=nx1 διάνυσμα ,det(A) διάφορο του 0 , τότε η εντολή

x=A\ b

υπολογίζει με την μέθοδο Gauss τη λύση του συστήματος .

Πιο ειδικά , εφαρμόζει την διάσπαση LU με μερική οδήγηση,όπου

PA=LU=Pb , P πίνακας μετάθεσης

L κάτω τριγωνικός πίνακας με μονάδες στη διαγώνιο

U άνω τριγωνικός πίνακας

΄ και λύνει αυτόματα τα συστήματα

Ly=Pb

Ux=y

ΠΑΡΑΤΗΡΗΣΗ: στη μερική οδήγηση το στοιχείο οδηγός έχει μέγιστη απόλυτη τιμή.

ΠΑΡΑΔΕΙΓΜΑΤΑ

1)Να λυθεί το σύστημα

$$\begin{bmatrix} 2 & 1 & 7 \\ 0 & 1 & 3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

πληκτρολογούμε

A=[2 1 7;0 1 3;1 1 1]

b=[1 1 1]’

A\b

2)Να λυθεί το σύστημα

$$\begin{bmatrix} 0.780 & 0.563 \\ 0.913 & 0.659 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0.217 \\ 0.254 \end{bmatrix}$$

Ορίζοντας κατα τα γνωστά τον πίνακα και τους σταθερούς όρους έχουμε

$\begin{bmatrix} x \\ y \end{bmatrix}$= A\b=$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$, το αποτέλεσμα είναι ακριβές παρόλο που ο πίνακας του συστήματος έχει μεγάλο δείκτη κατάστασης

3)Να λυθεί το σύστημα

$$\begin{bmatrix} 2 & 0 & 0 \\ 1 & 3 & 0 \\ 5 & 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0.2 \\ 1.3 \\ 7.1 \end{bmatrix}$$

(η λύση είναι το διάνυσμα $\begin{bmatrix} 0.1 \\ 0.4 \\ 3.1 \end{bmatrix}$)

ΕΠΙΛΥΣΗ ΜΗ ΓΡΑΜΜΙΚΩΝ ΕΞΙΣΩΣΕΩΝ ΜΙΑΣ ΜΕΤΑΒΛΗΤΗΣ

Έστω το πρόβλημα εύρεσης των ριζών της εξίσωσης

f(x)=0

όπου f: $R \to R$.

Η matlab παρέχει την συνάρτηση fzero , πιο συγκεκριμμένα

fzero('όνομα συνάρτησης f ', x_0)

όπου

'όνομα συνάρτησης f '   είναι το όνομα του m-file της f

x_0 μια τιμή κοντά στη ρίζα.

ΠΑΡΑΔΕΙΓΜΑΤΑ

1)Να υπολογιστούν οι 3 ρίζες της

$x^3 - 6$atan(x)=0 .

Λύση

Ορίζουμε το m-file

```
function ex88=example88(x)
        ex88=x^3-6*atan(x)';
```

και το σώζουμε με το όνομα example88.m .

Στη συνέχεια πληκτρολογούμε

```
fzero('example88',1.5)
```

και η matlab δίνει τη ρίζα 1.8633 .

Ομοίως fzero('example88',0.5) δίνει τη ρίζα 0, και fzero('example88',-1.5) δίνει τη ρίζα -1.8633 .

2) 3x-2sin(x)+6=0

Λύση

Ορίζουμε

```
function ex99=example99(x)
        ex99=3*x-2*sin(x)';
```

σώζουμε σαν example99.m και πληκτρολογούμε

fzero('example99',-2)

έτσι βρίσκουμε τη ρίζα -2.4335 .

3) $10e^{-3x} + 2e^{-2x} - 6 = 0$

Λύση

```
function ex222=example222(x)
        ex222=10*exp(-3*x)+2*exp(-2*x)-6';
```

πληκτρολογώντας fzero('example222',1) , λαμβάνουμε τη ρίζα 0.2462 .

ΣΧΟΛΙΟ

Η fzero απαιτει η τιμή της συνάρτησης στο m-file να ορίζεται σαν διάνυσμα στήλη ,γιαυτό και υπάρχει το ' .

ΑΣΚΗΣΗ

Σχεδιάζοντας τις γραφικές παραστάσεις των παραπάνω συναρτήσεων επαληθεύστε τις τιμές των ριζών .

Λύση

(για το 1))

πληκτρολογούμε

```
    n=40;
    x=linspace(-2,2,n);
for k=1:40
     y(k)=example(x(k));
end
plot(x,y)
title('f(x)=x^3-6atan(x)')
xlabel('x')
ylabel('y')
```

ΩΡΑ 4η

ΠΡΟΣΕΓΓΙΣΤΙΚΗ ΕΠΙΛΥΣΗ ΔΙΑΦΟΡΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΤΗΣ ΜΟΡΦΗΣ

dy(t)=f(t , y(t))

y(tinitial)=yo=γνωστό

tinitial<= t <= tfinal

όπου y:R $\rightarrow$ R^n ,δηλαδή dim y=n

Η επίλυση γίνεται με χρήση της συνάρτησης ode45 ή ode23, η κλήση αυτών των συναρτήσεων γίνεται ως εξής (τα ίδια και για την ode23)

[t,u]=ode45('όνομα συνάρτησης',[tinitial,tfinal],yo,options);

όπου

'όνομα συνάρτησης' είναι το όνομα του m-file της f=f(t,y)

[tinitial,tfinal] είναι το διάστημα του t στο οποίο ζητάμε τη λύση

yo=[y(1),…,y(n)]=γνωστό

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

με dim[1-e4 …1-e4]=dimy=n

είναι προφανές ότι πρώτα θα ορίσουμε το options , που ορίζει την σχετική και απόλυτη ακρίβεια της μεθόδου .

Μπορούμε να σχεδιάσουμε το διάγραμμα φάσεων της λύσης για dimy=2 με άξονα των x το y1(t) και άξονα των y το y2(t) με την εντολή

plot(u(:,1),u(:,2))

επίσης είναι δυνατό με την εντολή

plot(t,u(:,1))

να δούμε την συμπεριφορά της y1 συντεταγμένης ως προς τον χρόνο κ.τ.λ.

ΣΧΟΛΙΑ

Χρειάζεται προσοχή στον ορισμό της f=f(t,y) η οποία θα πρέπει να ορισθεί ως διάνυσμα στήλη , επίσης ακόμα και στην περίπτωση αυτόνομου συστήματος θα πρέπει η μεταβλητή t να αναγραφεί .

ΠΑΡΑΔΕΙΓΜΑ 1ο

Έστω το αυτόνομο διαφορικό σύστημα

$$dy=\begin{pmatrix} 0 & -2 \\ 2 & 0 \end{pmatrix} . y$$

tinitial=-3

tfinal=3

y(tinitial)=[0.1,0.1]

ορίζουμε ως m-file

```
function ex6=example6(t,y)
    ex6=[-2*y(2) 2*y(1)]';
```

με το παρακάτω βασικό πρόγραμμα 1 βρίσκουμε τη λύση και τη σχεδιάζουμε .

ΠΡΟΓΡΑΜΜΑ 1

```
options=odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4]);
axis([-3 3 -0.2 0.15])
title('solution')
xlabel('t')
ylabel('y1(t),y2(t)')
hold on
[t,u]=ode45('example6',[-3,3],[0.1,0.1],options);
plot(t,u(:,1))
plot(t,u(:,2))
hold off
```

ΣΧΟΛΙΑ

Είναι δυνατό να βρούμε με το διάγραμμα την περίοδο της λύσης .

Με το επόμενο βασικό πρόγραμμα 2 κατασκευάζουμε το διάγραμμα φάσεων για 50 τροχιές στο ίδιο διάγραμμα, το αποτέλεσμα είναι κλειστές τροχιές (περιοδικές λύσεις) δηλαδή κέντρο γύρω από το σημείο (0,0) ,

με y(tinitial)=[0.1*k,0.1*k] ,για 1<=k <=50

tinitial=-1.7

```
tfinal=1.7
```

ΠΡΟΓΡΑΜΜΑ 2

```
options=odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4]);
axis([-5 5 -5 5])
title('orbits')
xlabel('y1')
ylabel('y2')
hold on
for k=1:50
[t,u]=ode23('example6',[-1.7,1.7],[0.1*k,0.1*k],options);
plot(u(:,1),u(:,2))
end
print   {αν θέλουμε να τυπώσουμε σε εκτυπωτή}
hold off
```

ΠΑΡΑΔΕΙΓΜΑ 2ο

Έστω το πρόβλημα αρχικών τιμών

```
        dy=t*y
        tinitial=-3
        tfinal=3
        y(tinitial)=1 ,δηλαδή dim y=1
```

Ορίζουμε ως m-file

```
      function ex21=example21(t,y)
           ex21=[t*y(1) ]';
```

ΠΡΟΓΡΑΜΜΑ 3

```
options=odeset('RelTol',1e-4,'AbsTol',[1e-4 ]);
axis([-3 3 -2 2])
title('solution')
xlabel('t')
ylabel('y(t)')
hold on
[t,u]=ode45('example21',[-3,3],[1],options);
plot(t,u(:,1))
```

hold off

-Στις επόμενες σελίδες παρατίθενται τα γραφήματα των προγραμμάτων 1,2,3

ΓΕΝΙΚΑ ΣΧΟΛΙΑ

Η ode23 και ομοίως η ode45 έχει ως ορίσματα τα εξής

‘όνομα συνάρτησης’ {f=f(t,y)}

[αρχικός χρόνος,τελικός χρόνος]

[τιμή της συνάρτησης στον αρχικό χρόνο] { y(tinitial)]}

options {ορίζει την σχετική και απόλυτη ακρίβεια της μεθόδου}

{dim RelTol=1 , dim AbsTol=dim y}

Στο παραπάνω παράδειγμα βρίσκουμε πολλές τροχιές και γιαυτό χρησιμοποιούμε επαναληπτική εντολή . Το αποτέλεσμα του προγράμματος σε εκτυπωτή βρίσκεται στο σχήμα 1 .

Μια παραλλαγή του προηγουμένου προγράμματος μας επιτρέπει να δούμε τις λύσεις συναρτήσει του χρόνου , για παράδειγμα μπορούμε να πληκτρολογήσουμε τα εξής

```
  options=odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4]);
axis([-5 5 -5 5 -1.7 1.7])
title('solutions')
xlabel('y1')
ylabel('y2')
zlabel('t')
hold on
for k=1:50
[t,u]=ode23('example6',[-1.7,1.7],[0.1*k,0.1*k],options);
plot3(u(:,1),u(:,2),t)
end
```

ΣΧΟΛΙΑ

Η εντολή plot3 σχεδιάζει γραφικές παραστάσεις στις 3 διαστάσεις . Η εκτύπωση βρίσκεται στο σχήμα 2 .

ΩΡΑ 5Η

Στη συνέχεια θα δοθούν παραδείγματα τα οποία λύνουν κάποια προβλήματα και σχεδιάζουν τις λύσεις.

ΠΑΡΑΔΕΙΓΜΑ 3ο

Έστω το πρόβλημα αρχικών τιμών στο R

y'=y*y

y(0)=c

τότε ως γνωστόν y(t)=-1/(t-1/c) .

Παρατηρούμε ότι για t=1/c η λύση εκρήγνυται (απειρίζεται σε πεπερασμένο t) συνεπώς αν απαιτήσουμε t σε διάστημα περιέχον το 1/c ,θα παρατηρήσουμε το λεγόμενο blow up .

Θέτουμε c=1 ,tinitial=0 ,tfinal πολύ κοντά στο 1 τότε υπάρχει πρόβλημα , η ακριβής λύση είναι y(t)=-1/(t-1) .

Κατά τα γνωστά ορίζουμε στα m-files

```
function ex22=example22(t,y)
        ex22=[y(1)*y(1)]';
```

ΠΡΟΓΡΑΜΜΑ4

```
options=odeset('RelTol',1e-4,'AbsTol',[1e-4]);
xlabel('t')
ylabel('y(t)')
hold on
[t,u]=ode45('example22',[0,0.9],[1],options);
plot(t,u(:,1))
hold off
```

ΣΧΟΛΙΑ

Επαναλάβετε το ίδιο πρόγραμμα για tfinal=0.999 κ.λ.π τί παρατηρείτε;

Αν tfinal=2 τί θα συμβεί;

ΠΑΡΑΔΕΙΓΜΑ 4ο

Έστω το 3 x 3 διαφορικό σύστημα

$$y' = \begin{bmatrix} y1 * y1 \\ y2 * y2 \\ y3 * y3 \end{bmatrix}$$

y(0)=[1 2 3]

[tfinal,tinitial]=[0,1/4]

Ορίζουμε στα m-files

```
function ex23=example23(t,y)
        ex23=[y(1)*y(1) y(2)*y(2) y(3)*y(3)]';
```

ΠΡΟΓΡΑΜΜΑ 5

```
options=odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4 1e-4]);
xlabel('t')
ylabel('y1(t),y2(t),y3(t)')
hold on
[t,u]=ode45('example23',[0,1/4],[1,2,3],options);
plot(t,u(:,1))
plot(t,u(:,2))
plot(t,u(:,3))
hold off
```

Στη συνέχεια παρατίθενται τα γραφήματα των ΠΡΟΓΡΑΜΜΑΤΩΝ 4 ,5.