

Μέθοδοι Επίλυσης με Η/Υ

10^ο Μάθημα:

*Συμβολικές πράξεις, ολοκληρώματα, παράγωγοι,
διαφορικές εξισώσεις*

Ν.Δ. Λαγαρός, Α. Στάμος, Χ. Φραγκουδάκης

Το σημερινό μάθημα περιέχει:

- 1) Σύμβολα και συμβολικές πράξεις
- 2) Συμβολικά ολοκληρώματα
- 3) Συμβολικές παραγωγίσεις
- 4) Συμβολική επίλυση διαφορικών εξισώσεων
- 5) Εφαρμογές

Σύμβολα και συμβολικές πράξεις

Στην άλγεβρα ισχύει:

$$(1+x)(1+y) = x+y+xy+1$$

Στο Matlab όμως (command window):

```
>> (1+x)*(1+y)
```

```
Undefined function or variable 'x'.
```

Δίνοντας τιμές στις μεταβλητές x, y δίνει αριθμητικό (όχι αλγεβρικό) αποτέλεσμα:

```
>> x=10; y=100; (1+x)*(1+y)
```

```
ans =
```

```
1111
```

Οι μεταβλητές x, y πρέπει να δηλωθούν ως σύμβολα (συμβολικές μεταβλητές):

```
>> clear; x=sym('x'); y=sym('y'); (1+x)*(1+y)
```

```
Ans =
```

```
(x + 1)*(y + 1)
```

Σύμβολα και συμβολικές πράξεις

Τα σύμβολα δεν έχουν κάποια αριθμητική τιμή:

```
>> x  
x =  
x
```

Πράξεις μεταξύ συμβόλων δίνει αποτέλεσμα που είναι επίσης σύμβολο:

```
>> syms x y; p=(1+x)*(1+y); p  
p =  
(x + 1)*(y + 1)
```

(η εντολή `syms x y;` είναι συντομογραφία των εντολών `x=sym('x');` `y=sym('y');`;

Το Matlab δίνει το αποτέλεσμα απλοποιημένα (με λιγότερους όρους). Για να κάνει τις (συμβολικές) πράξεις:

```
>> expand(p)  
ans =  
x + y + x*y + 1
```

Σύμβολα και συμβολικές πράξεις

Μπορεί να οριστεί συμβολική συνάρτηση:

```
>> clear; syms x a f(x); f(x) = (1+x)*(1+a); f  
f(x) =  
(a + 1)*(x + 1)
```

Εδώ a είναι μία (συμβολική) σταθερά.

Η τιμή της συνάρτησης υπολογίζεται για συμβολικές σταθερές:

```
>> syms w; f(w)  
ans =  
(a + 1)*(w + 1)  
>> f(a)  
ans =  
(a + 1)^2  
>> expand(f(w+a))  
ans =  
2*a + w + a*w + a^2 + 1
```

Σύμβολα και συμβολικές πράξεις

Η τιμή της συνάρτησης υπολογίζεται και για αριθμητικές τιμές. Αν η συνάρτηση εκτός από συμβολικές μεταβλητές περιέχει και συμβολικές σταθερές, το αποτέλεσμα είναι εν γένει συμβολικό:

```
>> f(1)
ans =
2*a + 2
>> f(-1)
ans =
0
```

Για αριθμητικό αποτέλεσμα, οι συμβολικές σταθερές πρέπει να αντικατασταθούν από αριθμητικές σταθερές:

```
>> g=subs(f, a, 10); g
g(x) =
11*x + 11
>> g(1)
ans =
22
```

Σύμβολα και συμβολικές πράξεις

Μία συμβολική συνάρτηση μπορεί να περιέχει έτοιμες συναρτήσεις του Matlab:

```
>> clear; syms x a f(x); f(x) = a*sin(a*x^2); f  
f(x) =  
a*sin(a*x^2)
```

```
>> f(pi)  
ans =  
a*sin(pi^2*a)
```

```
>> f(0)  
ans =  
0
```

```
>> f(f(x))  
ans =  
a*sin(a^3*sin(a*x^2)^2)
```

Σύμβολα και συμβολικές πράξεις

Στο τελευταίο παράδειγμα το αποτέλεσμα είναι συμβολική παράσταση, όχι συνάρτηση:

```
>> g=f(f(x)); g
g =
a*sin(a^3*sin(a*x^2)^2)
>> g(2)
Error using mupadmex
Error in MuPAD Command: Index exceeds matrix dimensions.
Error in sym/subsref (line 687)
    B = mupadmex('symobj::subsref',A.s,inds{:});
```

Για να είναι συνάρτηση πρέπει να οριστεί:

```
>> g(x)=f(f(x)); g
g(x) =
a*sin(a^3*sin(a*x^2)^2)
>> g(2)
ans =
a*sin(a^3*sin(4*a)^2)
```


Συμβολικά ολοκληρώματα

Το αόριστο ολοκλήρωμα μίας συνάρτησης μπορεί να υπολογιστεί συμβολικά (το αποτέλεσμα είναι συνάρτηση):

```
>> clear; syms t f(t); f(t)=t;  
>> g=int(f); g  
g(t) =  
t^2/2
```

Η μεταβλητή ολοκλήρωσης μπορεί να ορίζεται:

```
>> clear; syms t s f(t, s); f(t, s)=t*s;  
>> g=int(f, s)  
g(t, s) =  
(s^2*t)/2
```

Ομοίως μπορεί να υπολογιστεί το ορισμένο ολοκλήρωμα (το αποτέλεσμα είναι συνάρτηση μίας μεταβλητής):

```
>> syms a b;  
>> int(f, s, a, b)  
ans(t) =  
-(t*(a^2 - b^2))/2
```

Συμβολικά ολοκληρώματα

Οι σταθερές ολοκλήρωσης μπορεί να είναι αριθμητικές:

```
>> int(f, s, 5, 10)
ans(t) =
(75*t)/2
```

Το διπλό ολοκλήρωμα υπολογίζεται σε δύο στάδια:

```
>> clear; syms t s f(t, s); f(t, s)=t*s;
>> g=int(f, t)
g(t, s) =
(s*t^2)/2
>> int(g, s)
ans(t, s) =
(s^2*t^2)/4
```

Ή ακόμα και σε ένα στάδιο:

```
>> clear; syms t s f(t, s); f(t, s)=s*sin(10*t);
>> int(int(f, t), s)
ans(t, s) =
-(s^2*cos(10*t))/20
```

Συμβολικά ολοκληρώματα

Το ορισμένο διπλό ολοκλήρωμα είναι συμβολική παράσταση (όχι συνάρτηση):

```
>> syms a b; g=int(int(f, t, a, b), s, a, b);  
>> g  
g =  
-((cos(10*a) - cos(10*b)) * (a^2 - b^2))/20
```

Ομοίως το απλό ορισμένο ολοκλήρωμα είναι συμβολική παράσταση:

```
>> clear; syms x a b f(x); f(x)=sin(x);  
>> g=int(f, x, a, b); g  
g =  
cos(a) - cos(b)
```

Το ορισμένο ολοκλήρωμα μπορεί να οριστεί ως συνάρτηση άλλου συμβόλου:

```
>> g(a)=int(f, x, a, b); g  
g(a) =  
cos(a) - cos(b)  
>> g(pi)  
ans =  
- cos(b) - 1
```

Συμβολικά ολοκληρώματα

Μερικά “δύσκολα” ολοκληρώματα υπολογίζονται:

```
>> clear; syms x f(x); f(x)=exp(-x^2);  
>> int(f, x, 0, x)  
ans =  
(pi^(1/2)*erf(x))/2
```

Μερικά όχι:

```
>> clear; syms x f(x); f(x)=exp(sin(x));  
>> int(f, x, 0, x)  
ans =  
int(exp(sin(x)), x, 0, x)
```

Η αριθμητική ολοκλήρωση υπολογίζεται πάντα:

```
>> x=2:0.1:10; y=exp(sin(x)); trapz(x, y)  
ans =  
10.3679
```

Συμβολικές παραγωγίσεις

Η παράγωγος μίας συνάρτησης μπορεί να υπολογιστεί συμβολικά (το αποτέλεσμα είναι συνάρτηση):

```
>> clear; syms t a f(t); f(t)=sin(a*t);  
>> g=diff(f); g  
g(t) =  
a*cos(a*t)
```

Η μεταβλητή ως προς την οποία γίνεται η παραγωγή μπορεί να ορίζεται:

```
>> clear; syms t s a f(t, s); f(t, s)=s^2*sin(a*t);  
>> diff(f, t)  
ans(t, s) =  
a*s^2*cos(a*t)  
>> diff(f, s)  
ans(t, s) =  
2*s*sin(a*t)
```

Συμβολικές παραγωγίσεις

Η δεύτερη, τρίτη κλπ παράγωγος υπολογίζεται ορίζοντας το βαθμό παραγωγίσης:

```
>> diff(f, s, 2)
ans(t, s) =
2*sin(a*t)
```

Προσοχή: η συνάρτηση diff υπολογίζει διαφορές (και όχι παραγώγους) σε μη συμβολικά ορίσματα:

```
clear; x=[1 3 6 10];
>> diff(x)
ans =
     2     3     4
```

Προσεγγιστικές αριθμητικές τιμές της παραγώγου υπολογίζονται από το script::

```
H=0.001;           %Μικρό βήμα
x=0:h:2*pi;
y=sin(x);           %Αριθμητικές τιμές της συνάρτησης
dy=diff(y)/h;        %Διαφορές/βήμα -> παράγωγος
n = length(dy);     %Ένα λιγότερο από το y
plot(x, y, x(1:n), dy);
```

Συμβολική επίλυση διαφορικών εξισώσεων

Η επίλυση διαφορικών εξισώσεων μπορεί να γίνει συμβολικά (το αποτέλεσμα είναι συμβολική παράσταση, όχι συνάρτηση):

```
>> clear; syms x a y(x);  
>> dy=diff(y, x);  
>> g=dsolve(dy==a*y); g  
g =  
C3*exp(a*x)
```

Μπορεί να δοθεί αρχική συνθήκη (εδώ εξαναγκάζουμε το αποτέλεσμα σε συνάρτηση):

```
>> clear; syms x a b y(x);  
>> dy=diff(y, x);  
>> g(x)=dsolve(dy==a*y, y(0)==b); g  
g(x) =  
b*exp(a*x)
```

Η αρχική συνθήκη μπορεί να αναφέρεται στην παράγωγο:

```
>> g(x)=dsolve(dy==a*y, dy(0)==b); g  
g(x) =  
(b*exp(a*x))/a
```

Συμβολική επίλυση διαφορικών εξισώσεων

Η διαφορική εξίσωση (και η αρχική συνθήκη) μπορεί να δοθεί σε πεπλεγμένη μορφή:

```
>> clear; syms t a y(t); dy=diff(y, t);  
>> g(t)=dsolve(dy+a*t==1, y(0)+dy(0)==0); g  
g(t) =  
t - (a*t^2)/2 - 1
```

Επίλυση διαφορικών εξισώσεων δευτέρου βαθμού:

```
>> clear; syms t a b x(t); ddx=diff(x, t, 2);  
>> g(t)=dsolve(ddx==a+b*t); g  
g(t) =  
(b*t^3)/6 + (a*t^2)/2 + C2*t + C3
```

Επίλυση διαφορικών εξισώσεων δευτέρου βαθμού με αρχικές συνθήκες:

```
>> clear; syms t a b x0 v0 x(t);  
>> dx=diff(x, t); ddx=diff(x, t, 2);  
>> g(t)=dsolve(ddx==a+b*t, x(0)==x0, dx(0)==v0); g  
g(t) =  
(b*t^3)/6 + (a*t^2)/2 + v0*t + x0
```


Συμβολική επίλυση διαφορικών εξισώσεων

Μια (μη γραμμική) Δ.Ε. μπορεί να έχει πολλές λύσεις. Οι λύσεις επιστρέφονται ως πίνακας συμβόλων. Τα σύμβολα μπορούν να ανατεθούν μετά σε (συμβολικές) συναρτήσεις:

```
>> clear; syms t y(t); dy=diff(y, t);  
>> A=dsolve((dy+y)^2==1); A  
A =  
    C22*exp(-t) + 1  
    C24*exp(-t) - 1  
>> f(t)=A(1); f  
f(t) =  
    C22*exp(-t) + 1  
>> g(t)=A(2); g  
g(t) =  
    C24*exp(-t) - 1
```

Δεν επιλύονται συμβολικά όλες οι διαφορικές εξισώσεις:

```
>> clear; syms t y(t); ddy=diff(y, t, 2);  
>> dsolve(ddy==sqrt(y*t))
```

Warning: Explicit solution could not be found.
> In dsolve at 197

Αριθμητική επίλυση διαφορικών εξισώσεων

Επιλύονται Δ.Ε. της μορφής $y'=f(t, y)$ για t μεταξύ t_1 και t_2 και με αρχική συνθήκη $y(t_1)=c$

Η συνάρτηση $f(t, y)$ πρέπει να επιστρέφει την παράγωγο της $y(t)$ για χρόνο ίσο με t , με δεδομένο τον χρόνο t και την τιμή της $y(t)$ σε αυτόν το χρόνο.

Η εντολή συντάσσεται:

```
[T, Y] = ode45(@f, [t1 t2], c)
```

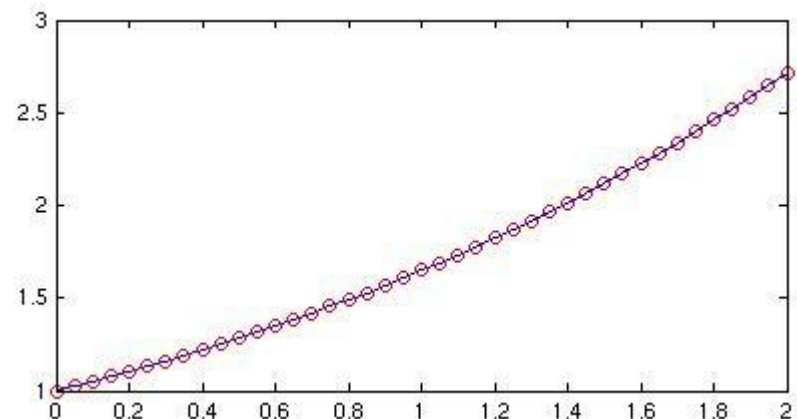
όπου **T** είναι μητρώο με τους χρόνους (μεταξύ t_1 και t_2) που υπολογίστηκε η συνάρτηση $y(t)$ και **Y** είναι πίνακας με τις τιμές της $y(t)$ σε αυτούς τους χρόνους.

Παράδειγμα: επίλυση της $y'=0.5y$ στο διάστημα $t=0$ έως 2 για $y(0)=1$ (η αναλυτική λύση είναι $y(t)=\exp(0.5t)$):

```
function dy=f(t, y)
dy=0.5*y;
end
```

Script:

```
[T, Y] = ode45(@f, [0 2], 1);
Y2 = exp(0.5*T);
plot(T, Y2, T, Y, 'ro');
```



Αριθμητική επίλυση διαφορικών εξισώσεων

Η επίλυση Δ.Ε. δευτέρου βαθμού $y''=f(t, y, y')$ και με αρχικές συνθήκες $y(t_1)=c_1$, $y'(t_1)=c_2$, γίνεται έμμεσα ορίζοντας βοηθητικές συναρτήσεις:

$$\begin{aligned} y_1(t) &= y(t) & y_1' &= f_1(t, y_1, y_2) = y_2 \\ y_2(t) &= y'(t) & y_2' &= f_2(t, y_1, y_2) = f(t, y_1, y_2) \end{aligned}$$

Στο Matlab πρέπει να συνταχθεί συνάρτηση F η οποία με δεδομένα το χρόνο t και ένα μονόστηλο πίνακα που αποτελείται από τιμές των συναρτήσεων y_1, y_2 για το χρόνο t, να επιστρέφει μονόστηλο πίνακα που αποτελείται από τις παραγώγους y_1', y_2' για το χρόνο t.

Η εντολή συντάσσεται:

$$[T, Y] = \text{ode45}(@F, [t1 \ t2], C)$$

όπου T είναι μονόστηλος πίνακας με τους χρόνους (μεταξύ t_1 και t_2) που υπολογίστηκαν οι συναρτήσεις $y_1(t), y_2(t)$, T είναι δίστηλος πίνακας με τις τιμές των $y_1(t), y_2(t)$ σε αυτούς τους χρόνους και C είναι μονόστηλος πίνακας με τις σταθερές c_1, c_2

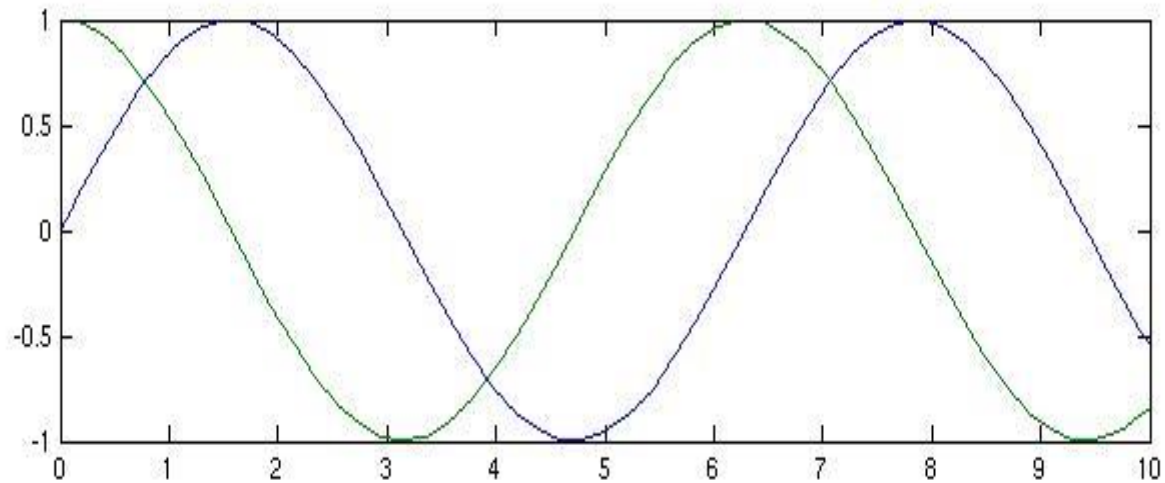
Παράδειγμα: επίλυση της $y'' = -y$ στο διάστημα $t=0$ έως 10 για $y(0)=0, y'(0)=1$ (η αναλυτική λύση είναι $y(t)=\sin(t)$).

Αριθμητική επίλυση διαφορικών εξισώσεων

```
function DY=F(t, Y)
DY=[Y(2) -Y(1)];
DY=transpose(DY);
end
```

Script:

```
[T, Y] = ode45(@F, [0 10]', [0 1]');
plot(T, Y);
```



Ερωτήσεις...