Κεφάλαιο 8ο: Διαφορικές εξισώσεις

8.1 Η *εντολή* DSolve

Σε αυτό το κεφάλαιο θα ασχοληθούμε με την λύση διαφορικών εξισώσεων δηλ. εξισώσεων με παραγώγους στις οποίες είναι άγνωστη *μια* ή *περισσότερες συναρτήσεις* y[x]*και που η παράγωγος τους* y'[x] *ικανοποιεί τις εξισώσεις*. Π.*χ* x^2 y'[x]-y[x]==0 (*άγνωστη η* y[x]) *και το σύστημα* y'[x]+y[x]-z[x]==Sin[x],z'[x]+y[x]+z[x]==Cos[x] *με άγνωστες συναρτήσεις τις* y[x],z[x]. *Δυο* είναι *οι βασικές συναρτήσεις η* DSolve *και η* NDSolve. Η *δεύτερη μας παρέχει την δυνατότητα αριθμητικής* επίλυσης *διαφορικών εξισώσεων και συστημάτων και χρησιμοποιεί προσεγγιστικές μεθόδους ενώ η πρώτη προσπαθεί να βρεί μια ακριβή* λύση *των εξισώσεων*.

```
?DSolve
```

```
DSolve[eqn, y, x] solves a differential equation for
   the function y, with independent variable x. DSolve[
   {eqn1, eqn2, ...  }, {y1, y2, ...  }, x] solves a
   list of differential equations. DSolve[eqn, y, {x1,
   x2, ... }] solves a partial differential equation. More...
```

δηλ. η DSolve[eqn, y, x] *λύνει την διαφορική εξίσωση* eqn *και βρίσκει την συνάρτηση* y, *η οποία* έχει *ανεξάρτητη μεταβλητή την* x. Η DSolve[{eqn1, eqn2, ... }, {y1, y2, ... }, x] *λύνει το σύστημα* eqn1, eqn2, ... *απο διαφορικές εξισώσεις με άγνωστες τις συναρτήσεις* y1, y2, ... *που* είναι *συναρτήσεις της* x. Τέλος *η* DSolve[eqn, y, {x1, x2, ... }] *λύνει την εξίσωση μερικών παραγώγων* eqn *ως προς την* y *που* είναι *συνάρτηση των μεταβλητών* x_1, x_2, ...
Ας λύσουμε για παράδειγμα την διαφ. εξίσωση eqn με τύπο $x^2\, y' - y = 0$ ως προς την μεταβλητή x.

```
Clear[y, x, eqn]
eqn = x^2 y'[x] - y[x] == 0
sol1 = DSolve[eqn, y, x]
```

$-y[x] + x^2\, y'[x] == 0$

$\{\{y \to (e^{-\frac{1}{\#1}}\, C[1]\ \&)\}\}$

Βλέπουμε *ότι η* γενική *λύση περιέχει μια σταθερά* C[1]. Το C[1] να το βλέπουμε σαν σταθερά και όχι σαν συνάρτηση C[ ]. Α*ν περιλαμβάνονται και* άλλες *σταθερές στην λύση αυτές θα απαριθμούνται ως*

C[1],C[2],...*Μπορούμε να αλλάξουμε τον συμβολισμό των σταθερών* όπως *μας αρέσει π.χ το* C *με* sta8era *με την εντολή* DSolveConstants->sta8era *π.χ*

```
soll = DSolve[eqn, y, x, DSolveConstants → sta8era]
```

$\left\{\left\{y \to \left(e^{-\frac{1}{\#1}}\ \text{sta8era}[1]\ \&\right)\right\}\right\}$

Παρατηρείστε ότι η γενική λύση μας δόθηκε με μορφή ενός κανόνα που βρίσκεται μέσα σε δύο {} οπότε πρέπει να βρούμε κάποιο τρόπο να την ξετρυπώσουμε. Προσέξτε παρακάτω με πόσους διαφορετικούς τρόπους βρίσκουμε 1) την τιμή της λύσης y για x=3, 2) την λύση ως συνάρτηση π.χ της μεταβλητής z 3)την παράγωγο της y ως προς την μεταβλητή z και 4)την τιμή της παραγώγου που βρήκαμε για z=3 5)πως κάνουμε την δοκιμή

```
y[3] /. soll[[1, 1]]
soll[[1, 1, 2]][3]
soll[[1, 1, 2]][x] /. x → 3

y[z] /. soll[[1, 1]]
soll[[1, 1, 2]][z]

D[y[z] /. soll[[1, 1]], z]
D[soll[[1, 1, 2]][z], z]

D[soll[[1, 1, 2]][z], z] /. z → 3

eqn /. soll
```

$\frac{\text{sta8era[1]}}{e^{1/3}}$

$\frac{\text{sta8era[1]}}{e^{1/3}}$

$\frac{\text{sta8era[1]}}{e^{1/3}}$

$e^{-1/z}\ \text{sta8era[1]}$

$e^{-1/z}\ \text{sta8era[1]}$

$\frac{e^{-1/z}\ \text{sta8era[1]}}{z^2}$

$\frac{e^{-1/z}\ \text{sta8era[1]}}{z^2}$

$\frac{\text{sta8era[1]}}{9\ e^{1/3}}$

$\frac{e^{-1/z}\ \text{sta8era[1]}}{z^2}[3]$

{True}

Μπορούμε να ορίσουμε νέες συναρτήσεις με χρήση της soll. Π.χ f[z_]:= (soll[[1,1,2]][z])^2 για να ορίσουμε το τερτάγωνο της λύσης ή την παράγωγό της: g[z_]:= D[soll[[1,1,2]][z],z] . Όμως θα χρειαστεί να είμαστε αρκετά προσεκτικοί διότι μερικές φορές τα πράγματα δεν δουλεύουν πάντα ομαλά. Δείτε και το παρακάτω ...

```
f[z_] := soll[[1, 1, 2]][z]
```

```
g[z_] := D[f[z], z]
g[3]
```

General::ivar : 3 is not a valid variable.

$$\partial_3 \frac{C[1]}{e^{1/3}}$$

Αυτό που συμβαίνει εδώ είναι ότι κατα τον υπολογισμό του g[3] αντικαθιστά το Mathematica το z με 3 και πάει να υπολογίσει το D[f[3],3] που φυσικά δεν το καταλαβαίνει!! Εμείς θέλουμε βέβαια, πρώτα να υπολογίσουμε την παράγωγο και μετά να γίνει η αντικατάσταση! αυτό επιτυγχάνεται με το Evaluate ή την χρήση του = αντί του :=.

```
Clear[z, g]
```

```
g[x_] := Evaluate[D[f[x], x]]
g1[x_] = D[f[x], x]
g[3]
g1[3]
g'[z](*παράγωγος*)
g1'[z](*παράγωγος*)
```

$$\frac{e^{-1/x}\,\text{sta8era}[1]}{x^2}$$

$$\frac{\text{sta8era}[1]}{9\,e^{1/3}}$$

$$\frac{\text{sta8era}[1]}{9\,e^{1/3}}$$

$$\frac{e^{-1/z}\,\text{sta8era}[1]}{z^4}-\frac{2\,e^{-1/z}\,\text{sta8era}[1]}{z^3}$$

$$\frac{e^{-1/z}\,\text{sta8era}[1]}{z^4}-\frac{2\,e^{-1/z}\,\text{sta8era}[1]}{z^3}$$

Σχόλιο για το := και το =. Προσέξτε το := κάνει **πρώτα** την αντικατάσταση της τιμής της x στο τύπο που βρίσκεται στα δεξιά του := και στην συνέχεια υπολογίζεται ο τύπος στα δεξιά. Δηλ. κάθε φορά που καλούμε :=

ξαναυπολογίζεται ο τύπος στα δεξιά! Αυτό όμως έχει σαν αποτέλεσμα να μην μπορεί να υπολογιστεί σωστά η παράγωγος διότι η προς παραγώγιση μεταβλητή (δηλ. x) έχει αντικατασταθεί με μια τιμή. Το Evaluate[f] αναγκάζει το *Mathematica* να λάβει υπόψη του το f (δηλ. να το υπολογίσει)πριν από οτιδήποτε άλλο πρόκειται να κάνει στη συνέχεια. Στην περίπτωσή μας αυτό σημαίνει να υπολογίσει πρώτα την παράγωγο και μετά να την αντικαταστήσει με την τιμή του x. Το = σε αντίθεση με το := υπολογίζει τον τύπο στα δεξιά μόνο την πρώτη φορά (εδώ κατά την εισαγωγή του ορισμού της g1 δηλ. της παραγώγου) και το αντικαθιστά στα αριστερά του = και στην συνέχεια το αποτέλεσμα που βρήκε(εδώ η παράγωγος) χρησιμοποιείται το ίδιο σε όλες της επόμενες φορές που θα καλέσουμε τον τύπο στα αριστερά(εδώ το g1[3]).

Στο παράδειγμα παρακάτω *για διδακτικούς λόγους γράψαμε κατά* **"λάθος"** DSolve[eqn,y[x],x] *αντί του σωστού* DSolve[eqn,y,x].

```
Remove[eqn, y, x]
eqn = x^2 y'[x] - y[x] == 0;
soll = DSolve[eqn, y[x], x, DSolveConstants → sta8era]

{{y[x] → e^(-1/x) sta8era[1]}}
```

Προσέξτε ότι *η λύση* y[x] *δεν δίνεται απευθείας αλλά μέσα σε* {{}} *δηλ. μέσα σε δύο λίστες* όπως *φαίνεται και απο την* FullForm:

```
FullForm[soll]

List[List[
  Rule[y[x], Times[Power[E, Times[-1, Power[x, -1]]], sta8era[1]]]]]
```

Για *να "ξετρυπώσουμε" την* y[x] *απο κει μέσα για να βρούμε για παράδειγμα το τετράγωνό της* θα *αναφέρουμε δύο τρόπους*:

Πρώτος τρόπος *χρησιμοποιώντας την* Last ή *ισοδύναμα παίρνοντας το στοιχείο* 2ο *της* soll[[1,1]] *δηλ. το* soll[[1,1]][[2]] *που γράφεται απλούστερα* soll[[1,1,2]]. Παρατηρείστε τα παρακάτω:

```
soll[[1]]
soll[[1, 1]]
soll[[1, 1, 2]]
Last[soll[[1, 1]]]
```

$\{y[x] \to e^{-1/x}\, \text{sta8era}[1]\}$

$y[x] \to e^{-1/x}\, \text{sta8era}[1]$

$e^{-1/x}\, \text{sta8era}[1]$

$e^{-1/x}\, \text{sta8era}[1]$

Δεύτερος τρόπος: *χρησιμοποιώντας το σύμβολο /. της αντικατάστασης*:

expr /. rules εφαρμόζει τους κανόνες που βρίσκονται στα rules *για να μετασχηματίσει κάθε τμήμα της expr.*

Παρατηρείστε τα παρακάτω:

```
y[x] /. soll
y[x] /. soll[[1, 1]]
y[x] /. (soll[[1]])
```

$\{e^{-1/x}\, \text{sta8era}[1]\}$

$e^{-1/x}\, \text{sta8era}[1]$

$e^{-1/x}\, \text{sta8era}[1]$

Το y[x] το *Mathematica* το βλέπει σαν κάποιο αντικείμενο με **δυο** όρους το x και το y και όχι σαν συνάρτηση της y με μεταβλητή την x. Για αυτό για να κάνουμε δοκιμή αν πράγματι ικανοποιείται η διαφορική εξίσωση θα πρέπει να αντικαταστήσουμε και την παράγωγο y'[x]. Παρατηρείστε τα παρακάτω:

```
lysh = soll[[1, 1]](*η λύση σε μορφή  κανόνα*)
paragwgos = D[soll, x][[1, 1]](*η παράγωγος σε μορφή κανόνα*)
eqn /. lysh
eqn /. lysh /. paragwgos
```

$y[x] \to e^{-1/x}$ sta8era[1]

$y'[x] \to \frac{e^{-1/x}\,\text{sta8era[1]}}{x^2}$

$-e^{-1/x}\,\text{sta8era[1]} + x^2\,y'[x] == 0$

True

Με $y[x] \to e^{-1/x}$ sta8era[1] *εννοούμε* ***αντικατάσταση*** **ολοκλήρου** ***του*** y[x] *με* $e^{-1/x}$ sta8era[1] ***και*** *τίποτα παραπάνω*. Επαναλαμβάνουμε ότι με *αυτην την αντικατάσταση* ***δεν μπορεί*** *να καταλάβει* ***το*** Mathematica ότι *με* y[x] *εννοούμε* ***συνάρτηση του*** x. Οπότε *δεν* έχει *καμμιά ιδέα για* ***το*** *πως θα βρεί μια* ***τιμ****ή* ***της*** y π.χ την y[3] ή *πως θα παραγωγίσει την* y. Το *μόνο που* ξέρει *είναι* ***η τιμ****ή* ***της*** y[ ] *όταν* έχουμε *θέσει* ***το*** *γράμμα* x. Οπότε *θα πρέπει* ***πρώτα*** *να κάνουμε την αντικάτάσταση* ***της*** y[x]*με* τον κανόνα lysh και της y'[x] με τον κανόνα paragwgos ***και μετά*** *να αντικαταστ*ήσουμε *στην* eqn *συγχρόνως τις αντικαταστράτιες των* y[x],y'[x] *για να κάνουμε την δοκιμ*ή. *Ας δούμε* και *τα παρακάτω αποτελέσματα*. Στο πρώτο υπολογίζουμε το y[3] αφού βέβαια κάνουμε αντικατάσταση και του y(με την lysh) **και** του x(με το 3). Στο συνέχεια υπολογίζουμε την παράγωγο της soll ως προς x και μαζί με την λύση τις αντικαθιστούμε συγχρόνως στην eqn για δοκιμή.

```
y[3] = y[x] /. lysh /. x → 3
paragwgoslyshs = D[soll, x][[1, 1]]
eqn /. {lysh, paragwgos}(*dokimi*)
```

$\frac{\text{sta8era[1]}}{e^{1/3}}$

$y'[x] \to \frac{e^{-1/x}\,\text{sta8era[1]}}{x^2}$

True

Σχόλιο για το /. Το /.{lysh, paragwgos} *σημαίνει* ότι ***συγχρόνως*** *αντικαθιστούμε την* y[x] *με* $e^{-1/x}$ sta8era[1] ***και την*** y'[x] *με* την παράγωγό της *μέσα στην* eqn. *Ας κάνουμε και ένα* άλλο *απλό παράδειγμα για να καταλάβουμε την διαφορά*:

```
{a, b, c} /. a → b /. b → d

{d, d, c}
```

```
{a, b, c} /. {a → b, b → d}

{b, d, c}
```

Στο 1*ο παράδειγμα αντικαταστήσαμε διαδοχικά ενώ στο* 2*ο κάναμε μια αντικατάσταση συγχρόνως*! *Γιαυτό εξάλλου προέκυψαν και διαφορετικά αποτελέσματα*!

Για να καταλάβει το *Mathematica* *ότι με* y[x] *εννοούμε μια* ***συνάρτηση του* x** *θα μπορούσαμε για παράδειγμα να ορίσουμε στη θέση της* y *την* f[z_]:=y[x]/.soll[[1]]/.x→z *και στη συνέχεια όπου υπάρχει η* y *να τη αντικαθιστούμε με την* f π.χ.:

```
f[z_] := y[x] /. soll[[1]] /. x → z
f[3]
f[z]
f'[z]
eqn /. y → f
```

$$\frac{\text{sta8era[1]}}{e^{1/3}}$$

$$e^{-1/z}\,\text{sta8era[1]}$$

$$\frac{e^{-1/z}\,\text{sta8era[1]}}{z^2}$$

```
True
```

Σχόλιο: *Αν αντί* y[x] *θέσουμε απλά* y *μέσα στην* DSolve *τότε τα πράγματα απλουστεύονται.* Η DSolve[eqn,y,x] *είναι πολύ χρήσιμη διότι μας δίνει* ***το*** y *να είναι* ***συνάρτηση του*** x ***και ο τύπος της*** y ***είναι*** y[x]/soll[[1,1]] *δηλ.* $=e^{-1/x}\,C[1]$. Η *παράγωγος ως προς* z ***είναι*** D[y[z]/.soll[[1,1]], z] *δηλ.* $\frac{e^{-1/z}\,C[1]}{z^2}$. *Παρατηρείστε* ό***τι το*** *Mathematica για* ***την*** *αντίστοιχη απάντηση που δίνει η* DSolve[eqn,y[x],x] , ***μπορεί να βρεί την παράγωγο ως προς*** x *αλλά* ***δεν*** *γνωρίζει πως να βρει* ***το*** y[z] *ούτε* ***την παράγωγο ως προς*** z(*όπως ήδη προαναφέραμε*)!!!

Το *πλεονέκτημα είναι φανερό*: ***μπορούμε με την λύση για την*** *άγνωστη* y *που δίνει η* DSolve[eqn,y,x] *να* ***κάνουμε*** *εύκολα* και ***απλά την δοκιμή***:

```
Clear[y, x, eqn]
eqn = x^2 y'[x] - y[x] == 0
sol1 = DSolve[eqn, y, x]
lysh1 = sol1[[1, 1]]
eqn /. lysh1
```

$-y[x] + x^2\, y'[x] == 0$

$\{\{y \to (e^{-\frac{1}{\#1}}\, C[1]\ \&)\}\}$

$y \to (e^{-\frac{1}{\#1}}\, C[1]\ \&)$

```
True
```

8.2 *Οι δυνατότητες της* DSolve

Γενικά *για την* DSolve *πρέπει να ξέρουμε* ότι *επιλύει* 1) όλες *τις γραμ. εξισώσεις με σταθερούς συντελεστές ποιασδήποτε τάξης* 2)*ένα ευρύ φάσμα γραμμικών εξισώσεων με μη σταθερούς συντελεστές μέχρι* 2*ης τάξης και* 3) *ένα ευρύ φάσμα μη γραμμικών διαφορικών εξισώσεων*. Μπορεί *επίσης να επιλύσει διαφορικές εξισώσεις με μερικές παραγώγους αρκει να δώσουμε τις ανεξάρτητες μεταβλητές απο τις οποίες εξαρτάται η ζητούμενη συνάρτηση, υπό μορφή λίστας*. Ακολουθούν *παραδείγματα*:

Γ*ραμμικ*ή *διαφορικ*ή *εξίσωση* 1*ης τάξης* με μη σταθερούς συντελεστές

$$eqn1 = y'[x] + y[x]/x + \frac{Cos[x] - E^x}{x} == 0$$

```
sol11 = DSolve[eqn1, y, x]
```

$$\frac{-e^x + Cos[x]}{x} + \frac{y[x]}{x} + y'[x] == 0$$

$$\left\{\left\{y \to Function\left[\{x\}, \frac{C[1]}{x} + \frac{e^x - Sin[x]}{x}\right]\right\}\right\}$$

Εύρ*εση μιας τιμής της* y *π.χ της* y[3]

```
y[3] /. soll1[[1, 1]]
```

$$\frac{C[1]}{3}+\frac{1}{3}\left(e^3-\mathrm{Sin}[3]\right)$$

*Όταν υπάρχουν **αρχικές συνθήκες** τις αναγράφουμε μέσα στην εντολή* DSolve *με μορφή λίστας στη μορφή*:

DSolve[{eqn, αρχικές ***συνθήκες***},y,x]

Π. *χ για να λυθεί η* eqn1 *με την αρχική συνθήκη* y[2]==1 *τότε θα γράψουμε*:

```
DSolve[{eqn1, y[2] == 1}, y, x]
```

$$\left\{\left\{y\to \mathrm{Function}\left[\{x\},\ \frac{2-e^2+e^x+\mathrm{Sin}[2]-\mathrm{Sin}[x]}{x}\right]\right\}\right\}$$

Γραμμική διαφορική εξίσωση 2ης τάξης με σταθερούς συντελεστές

```
eqn2 = y''[x] - 2 y'[x] + y[x] == 0
soll1 = DSolve[eqn2, y, x]
```

$$y[x]-2\,y'[x]+y''[x]==0$$

$$\{\{y\to \mathrm{Function}[\{x\},\ e^x\,C[1]+e^x\,x\,C[2]]\}\}$$

Εδώ εμφανίζονται δύο σταθερές στην λύση. Για να εξαφανίσουμε την μια απο τις δύο θα πρέπει να δώσουμε κάποιες αρχικές συνθήκες στην y *ή στην παράγωγο* y'. *Αν δώσουμε αρχικές συνθήκες συγχρόνως και στις δύο τότε παίρνουμε μια λύση* y *χωρίς σταθερές*:

```
DSolve[{eqn2, y[0] == 1}, y, x]
DSolve[{eqn2, y'[1] == 0}, y, x]
DSolve[{eqn2, y[0] == 1, y'[1] == 0}, y, x]
```

$\{\{y \to \text{Function}[\{x\}, e^x (1 + x\, C[2])]\}\}$

$\{\{y \to \text{Function}[\{x\}, e^x (-2 + x)\, C[2]]\}\}$

$\left\{\left\{y \to \text{Function}\left[\{x\}, -\frac{1}{2} e^x (-2 + x)\right]\right\}\right\}$

Γραμμική διαφορική εξίσωση 2ης τάξης με μη σταθερούς συντελεστές

```
eqn3 = x y''[x] - (2 x + 1) y'[x] + (x + 1) y[x] == x √x E^x
DSolve[eqn3, y, x]
```

$(1 + x)\, y[x] - (1 + 2\, x)\, y'[x] + x\, y''[x] == e^x x^{3/2}$

$\left\{\left\{y \to \text{Function}\left[\{x\}, \frac{4}{5} e^x x^{5/2} + e^x C[1] + \frac{1}{2} e^x x^2 C[2]\right]\right\}\right\}$

Εξίσωση με μερικές παραγώγους ως προς x *και* y *και άγνωστη συνάρτηση την* z[x,y]

```
eqn4 = x D[z[x, y], x] + y D[z[x, y], y] == z[x, y]
soll4 = DSolve[eqn4, z, {x, y}]
```

$y\, z^{(0,1)}[x, y] + x\, z^{(1,0)}[x, y] == z[x, y]$

$\left\{\left\{z \to \text{Function}\left[\{x, y\}, x\, C[1]\left[\frac{y}{x}\right]\right]\right\}\right\}$

Προσοχή εδώ με $C[1]\left[\frac{y}{x}\right]$ *εννοούμε μια* ***συνάρτηση*** C[1] *μιας μεταβλητής έστω* t *στην οποία έχουμε αντικαταστήσει την μεταβλητή* t *με το λόγο* y/x. *Ας κάνουμε και δοκιμή*

```
soll4[[1, 1]]
eqn4 /. soll4[[1, 1]]
eqn4 /. soll4[[1, 1]] // Simplify
```

$z \to \text{Function}\left[\{x, y\}, x\, C[1]\left[\frac{y}{x}\right]\right]$

$y\, C[1]'\left[\frac{y}{x}\right] + x\left(C[1]\left[\frac{y}{x}\right] - \frac{y\, C[1]'\left[\frac{y}{x}\right]}{x}\right) == x\, C[1]\left[\frac{y}{x}\right]$

```
True
```

Η Simplify *κάνει τους απαιτούμενους υπολογισμούς και απλοποιεί την παράσταση*

Ομογενής γραμμική εξίσωση 2ης τάξης (με σταθερούς συντελεστές και)με μερικές παραγώγους και άγνωστη συνάρτηση την z[x,y]

```
eqn5 =
 D[z[x, y], {x, 2}] - 5 D[z[x, y], x, y] + 6 D[z[x, y], {y, 2}] == 0
soll5 = DSolve[eqn5, z, {x, y}]
```

$6\, z^{(0,2)}[x, y] - 5\, z^{(1,1)}[x, y] + z^{(2,0)}[x, y] == 0$

```
{{z → Function[{x, y}, C[1][3 x + y] + C[2][2 x + y]]}}
```

Εδώ *βλέπουμε ότι η γενική λύση περιέχει δυο συναρτήσεις* C[1] *και* C[2] *στις οποίες έχουμε αντικαταστήσει την μεταβλητή τους με* 3 x+y *και με* 2 x+y *αντίστοιχα*. Η *δοκιμή γίνεται εύκολα*:

```
soll5[[1, 1]]
eqn5 /. soll5[[1, 1]]
eqn5 /. soll5[[1, 1]] // Simplify
```

```
z → Function[{x, y}, C[1][3 x + y] + C[2][2 x + y]]
```

```
9 C[1]''[3 x + y] + 4 C[2]''[2 x + y] + 6 (C[1]''[3 x + y] + C[2]''[2 x + y]) -
  5 (3 C[1]''[3 x + y] + 2 C[2]''[2 x + y]) == 0
```

```
True
```

8.3 Συστήματα διαφορικών εξισώσεων και η εντολή FullSimplify

Με *την*DSolve *μπορούμε να επιλύσουμε και συστήματα διαφορικών εξισώσεων, απλά θα προσέξουμε να δώσουμε μαζί με τις διαφορικές εξισώσεις και τις αρχικές* ή *οριακές συνθήκες μέσα σε μια λίστα και τις άγνωστες συναρτήσεις μέσα σε άλλη λίστα.*Π.*χ*

```
system6 =
  {y'[x] + y[x] - z[x] == Sin[x], z'[x] + y[x] + z[x] == Cos[x]};
agnwstesSynarthseis6 = {y[x], z[x]};
soll6 = DSolve[system6, agnwstesSynarthseis6, x]
soll6 // Simplify
soll6 // FullSimplify
```

$$\{\{y[x]\to e^{-x}C[1]\,Cos[x]+e^{-x}C[2]\,Sin[x]+e^{-x}Sin[x]\,(e^{x}Cos[x]^{2}+e^{x}Sin[x]^{2}),\ z[x]\to e^{-x}C[2]\,Cos[x]-e^{-x}C[1]\,Sin[x]+e^{-x}Cos[x]\,(e^{x}Cos[x]^{2}+e^{x}Sin[x]^{2})\}\}$$

$$\{\{y[x]\to e^{-x}(C[1]\,Cos[x]+(e^{x}+C[2])\,Sin[x]),\ z[x]\to e^{-x}((e^{x}+C[2])\,Cos[x]-C[1]\,Sin[x])\}\}$$

$$\{\{y[x]\to Sin[x]+e^{-x}(C[1]\,Cos[x]+C[2]\,Sin[x]),\ z[x]\to Cos[x]+e^{-x}(C[2]\,Cos[x]-C[1]\,Sin[x])\}\}$$

Εδώ *χρησιμοποιήσαμε και την συνάρτηση* FullSimplify *για να κάνουμε* όσο *το δυνατόν περισσότερες απλοποιήσεις.* Αλλιώς *η λυση* όπως *βλέπουμε* έχει *περίπλοκη μορφή.* Το *ίδιο θα κάνουμε* όπου *χρειάζεται και παρακάτω.* Για *την δοκιμή θα πρέπει να βρούμε και τις παραγώγους* y'[x] *και* z'[x] *και να αντικαταστήσουμε:*

```
system6 /. soll6[[1]] /. D[soll6[[1]], x]
system6 /. soll6[[1]] /. D[soll6[[1]], x] // Simplify
```

$$\{e^{-x}Sin[x]\,(e^{x}Cos[x]^{2}+e^{x}Sin[x]^{2}) == Sin[x],\ e^{-x}Cos[x]\,(e^{x}Cos[x]^{2}+e^{x}Sin[x]^{2}) == Cos[x]\}$$

```
{True, True}
```

Σχόλιο Αν *είχαμε θέσει παραπάνω* agnwstesSynarthseis6={y, z} *αντί* agnwstesSynarthseis6={y[x],z[x]} *θα* είχαμε *αποφύγει να βρούμε και τις παραγώγους των* y *και* z *για να κάνουμε την δοκιμή* . Κοιτάξτε και τα

```
system6 = {y'[x] + y[x] - z[x] == Sin[x], z'[x] + y[x] + z[x] == Cos[x]};
agnwstesSynarthseis6 = {y, z};
soll6 = DSolve[system6, agnwstesSynarthseis6, x]
system6 /. soll6 // FullSimplify

{{y -> (e^-#1 (C[1] Cos[#1] + e^#1 Sin[#1] + C[2] Sin[#1]) &),
  z -> (e^-#1 (e^#1 Cos[#1] + C[2] Cos[#1] - C[1] Sin[#1]) &)}}
```

```
{{True, True}}
```

Αν στο παραπάνω σύστημα έχουμε και αρχικές συνθήκες π.χ y[π/2]=1 *και* z[π/2]=0 *τότε αυτές προστίθονται πολύ απλά στο σύστημα*:

```
DSolve[{system6, y[Pi / 2] == 1, z[Pi / 2] == 0},
  agnwstesSynarthseis6, x] // FullSimplify

{{y[x] -> Sin[x], z[x] -> Cos[x]}}
```

Άσκηση. Προσπαθήστε χρησιμοποιώντας την Part να βρείτε την τιμή z[3] της z και την παράγωγο της y.

8.4 H *εντολή* NDSolve

Κλ*είνουμε το κεφάλαιο με την* NDSolve *η οποία παρέχει τη δυνατότητα αριθμητικής επίλυσης διαφορικών εξισώσεων και συστημάτων. Συνήθως χρησιμοποιούμε την* NDSolve *όπου μια ακριβή λύση δεν μπορεί να βρεθεί με την* DSolve. Π.*χ*

```
Clear[eqn, soll]
eqn = y''[x] + 5 Log[y[x]] == 0;
soll = DSolve[{eqn, y'[0] == 1, y[0] == 1}, y[x], x]
```

```
InverseFunction::ifun :
 Inverse functions are being used. Values may be lost for multivalued inverses.

InverseFunction::ifun :
 Inverse functions are being used. Values may be lost for multivalued inverses.

InverseFunction::ifun :
 Inverse functions are being used. Values may be lost for multivalued inverses.

General::stop :
 Further output of InverseFunction::ifun will be suppressed during this calculation.

Solve::tdep : The equations appear to involve the
   variables to be solved for in an essentially non-algebraic way.

Solve::tdep : The equations appear to involve the
   variables to be solved for in an essentially non-algebraic way.

DSolve::dsing : Unable to fit initial/boundary conditions {y'[0] == 1, y[0] == 1} .
```

soll ==

$$\left\{\left\{\text{Solve}\left[-\int_{C[2]}^{y[x]} \frac{1}{\sqrt{10\,K\$81 - C[1] - 10\,K\$81\,\text{Log}[K\$81]}}\, dK\$81 == x,\ y[x]\right],\right.\right.$$
$$\text{Solve}\left[\int_{C[2]}^{y[x]} \frac{1}{\sqrt{10\,K\$3120 - C[1] - 10\,K\$3120\,\text{Log}[K\$3120]}}\, dK\$3120 == x,\right.$$
$$\left.\left. y[x]\right]\right\},\ \{y'[0] == 1,\ y[0] == 1\}\right\}$$

δεν βγάζει κάτι...Η *σύνταξη της εντολής είναι*

NDSolve[*διαφορικές εξισώσεις*,y[x],{x,xmin,xmax}]

Αν υπάρχουν περισσότερες απο μια διαφ. εξισώσεις μπαίνουν σε λίστα μαζί με τις αρχικές συνθήκες αν αυτές βέβαια υπάρχουν. Αντι y[x] *μπορούμε να γράψουμε πιο απλά* y. Φυσικά *όπως* ήδη *ξέρουμε στην δεύτερη περίπτωση θα πάρουμε την* y *ως συνάρτηση του* x *ενώ στην πρώτη περίπτωση θα πάρουμε την τιμή* y[x] *της* y *σε μορφή κανόνα. Στο τέλος πρέπει να δηλώσουμε και το διάστημα* {x,xmin,xmax}, *μέσα στο οποίο θέλουμε να* βρεθεί *η (προσεγγιστική)* λύση. Π.χ

```
Clear[soll]
soll = NDSolve[{eqn, y'[0] == 1, y[0] == 1}, y, {x, 0, 4}]
soll1 = NDSolve[{eqn, y'[0] == 1, y[0] == 1}, y[x], {x, 0, 4}]
```

```
{{y → InterpolatingFunction[{{0., 4.}}, <>]}}
```

```
{{y[x] → InterpolatingFunction[{{0., 4.}}, <>][x]}}
```

Η NDSolve *ενεργεί ως εξής*: *Υπολογίζει τη λύση* y, *προσεγγιστικά, πρώτα για ένα μικρό πλήθος σημείων του διαστήματος* {xmin,xmax}. *Αυτά τα σημεία μαζί με τις προκύπτουσες τιμές τα βάζει σε μια λίστα που την ονομάζει για συντομία* <>. *Στη συνέχεια κάνει κατάλληλη παρεμβολή στις παραπάνω τιμές της λίστας* <> *για να μπορέσει να βρεί την τιμή της* y[x] *για οποιοδήποτε άλλο σημείο* x *του διαστήματος* που εμείς θα επιλέξουμε. *Για αυτό άλλωστε στην απάντηση η συνάρτηση* y *αναφέρεται ως* InterpolatingFunction *δηλαδή μια συνάρτηση που οι τιμές της* y[x] *υπολογίζονται με την* ***μέθοδο παρεμβολής***. *Αν θέλουμε να βρούμε για παράδειγμα την τιμή* y[0.02] *θα γράψουμε ένα απο τα παρακάτω*

```
(y[x] /. soll1[[1]]) /. x → 0.02
(y /. soll[[1, 1]])[0.02]
soll[[1, 1, 2]][0.02]
(y /. soll[[1]])[x] /. x → 0.02
(y /. soll[[1]])[0.02]
```

```
1.01999
```

```
1.01999
```

```
1.01999
```

```
1.01999
```

```
1.01999
```

Προσοχή δεν γράψαμε σκέτα y/.soll[[1,1]][0.02] ή y/.soll[[1]]/.x->0.02 *διότι δεν θα δουλέψει*! *Γενικά με* `InterpolatingFunction[ ... ][x]` *βρίσκουμε την τιμή της συνάρτησης παρεμβολής* y *στο* x. *Επίσης μπορούμε να βρούμε και όσες άλλες τιμές θάλουμε και να κάνουμε και γραφική παράσταση της (προσεγγιστικής) λύσης της διαφορικής εξίσωσης*. Π.χ *αν θελουμε να βρούμε τις τιμές* y[x] *για* x=0, 0.08, 2 0.08, 3 0.08, ... *θα γράψουμε* ενα απο τα παρακάτω

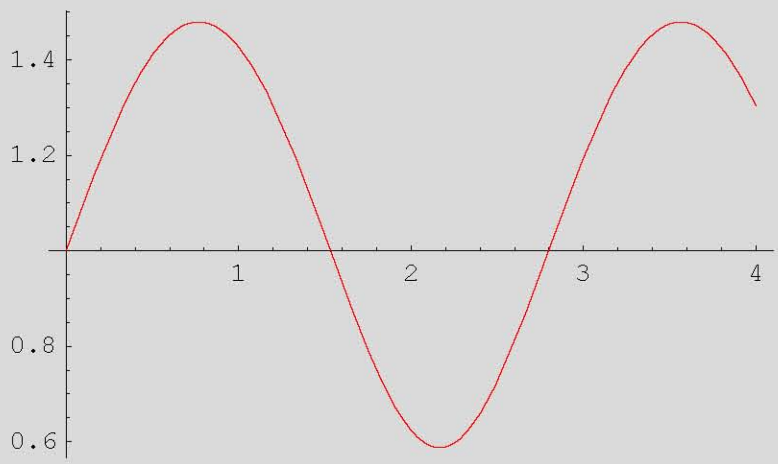
```
pinakas1 = Table[y[x] /. soll1, {x, 0, 4, .08}]
pinakas = Table[soll[[1, 1, 2]][x], {x, 0, 4, .08}]
```

```
{{1.}, {1.07958}, {1.15673}, {1.22925}, {1.29518}, {1.35286},
 {1.40089}, {1.43816}, {1.46382}, {1.47731}, {1.47834}, {1.46688},
 {1.44318}, {1.40776}, {1.36143}, {1.30524}, {1.24056}, {1.169},
 {1.09246}, {1.01311}, {0.933368}, {0.855841}, {0.783302},
 {0.718571}, {0.664395}, {0.623265}, {0.597208}, {0.587575},
 {0.594882}, {0.618737}, {0.657895}, {0.710411}, {0.773845},
 {0.845475}, {0.92248}, {1.00208}, {1.08163}, {1.15869}, {1.23106},
 {1.2968}, {1.35424}, {1.40201}, {1.43898}, {1.46433}, {1.4775},
 {1.4782}, {1.46642}, {1.44241}, {1.4067}, {1.3601}, {1.30367}}
```

```
{1., 1.07958, 1.15673, 1.22925, 1.29518, 1.35286, 1.40089,
 1.43816, 1.46382, 1.47731, 1.47834, 1.46688, 1.44318,
 1.40776, 1.36143, 1.30524, 1.24056, 1.169, 1.09246, 1.01311,
 0.933368, 0.855841, 0.783302, 0.718571, 0.664395, 0.623265,
 0.597208, 0.587575, 0.594882, 0.618737, 0.657895, 0.710411,
 0.773845, 0.845475, 0.92248, 1.00208, 1.08163, 1.15869,
 1.23106, 1.2968, 1.35424, 1.40201, 1.43898, 1.46433,
 1.4775, 1.4782, 1.46642, 1.44241, 1.4067, 1.3601, 1.30367}
```

Οι τιμές δεν διαφέρουν σε κάθε περίπτωση απλώς στον πρώτο πίνακα εμφανίζονται με την μορφή λίστας. Με *την* Plot *μπορούμε να κάνουμε την γραφικη της παράσταση:*

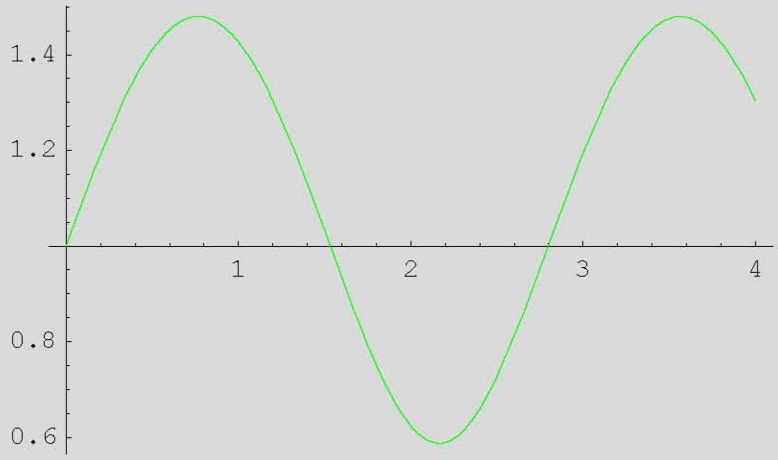
```
Plot[y[x] /. soll1, {x, 0, 4}, PlotStyle → RGBColor[1, 0, 0]]
```




```
- Graphics -
```

Φυσικά για την δεύτερη περίπτωση θα μπορούσαμε να γράψουμε

```
Plot[soll[[1, 1, 2]][x], {x, 0, 4}, PlotStyle → RGBColor[0, 1, 0]]
```



- Graphics -

Αν τώρα κάνουμε κλίκ πάνω στο σχήμα και στην συνέχεια πατήσουμε συνεχώς το πλήκτρο Alt *μπορούμε να δούμε στην αριστερή γωνία της οθόνης μας τις συντεταγμένες των σημείων της καμπύλης μας*. Η *εντολή* NDSolve *εφαρμόζεται εξίσου αποτελεσματικά και σε συστήματα διαφορικών εξισώσεων*.

Άσκηση. Σχεδιάσετε την παράγωγο της soll[[1,1,2]] στο διάστημα [0,4]. Υπόδειξη: Ορίσετε πρώτα την συνάρτηση f[x_]:=soll[[1,1,2]][x] και στην συνέχεια κάντε το γράφημα της f'[x].

***Άσκηση*:** Να *λυθεί αριθμητικά το σύστημα των διαφορικών εξισώσεων* y'[x]==Log[z[x]]+y[x],z'[x]==-Log[y[x]]-2 z[x]*με αρχικές συνθήκες* y[2]==z[2]==1 *στο διάστημα* [2,5]. Να *βρεθούν οι τιμές των συναρτήσεων* y[x] *και* z[x] *στα σημεία* x=2,2.5,3,3.5,...,5 *και να γίνουν οι γραφικές τους παραστάσεις*. Επίσης να παραγωγίσετε και να τις ολοκληρώσετε.