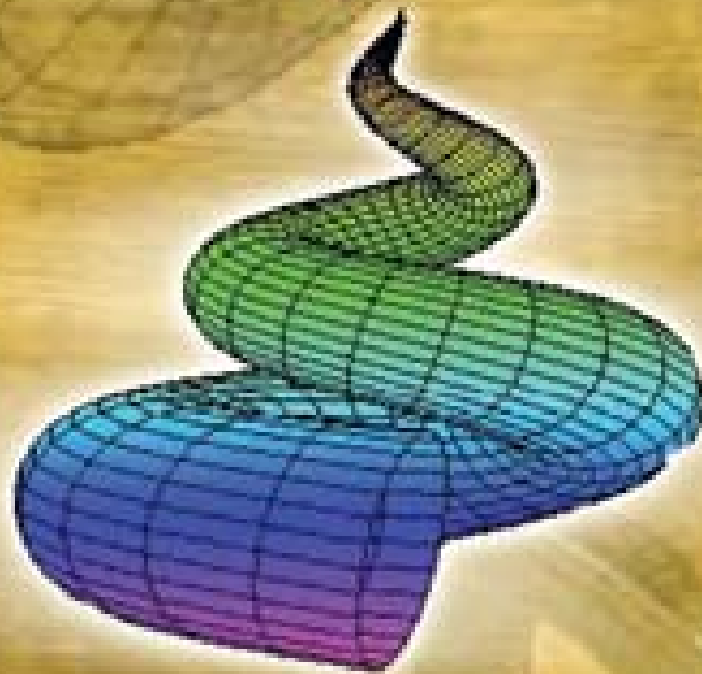


Ν. Ματζάκος

# Εισαγωγή στο Maple



ΕΚΔΟΣΕΙΣ  
ΝΕΩΝ  
ΤΕΧΝΟΛΟΓΙΩΝ



# Περιεχόμενα

<b>Ευρετήριο Πινάκων.....</b>	<b>7</b>
<b>Ευρετήριο Εικόνων.....</b>	<b>8</b>
<b>Εισαγωγή.....</b>	<b>9</b>
<b>Κεφάλαιο 1-Περιβάλλον Εργασίας - Στοιχεία Εντολών .....</b>	<b>13</b>
<b>1.1 Το Πρόγραμμα.....</b>	<b>14</b>
1.2.1 Εισαγωγή Εντολών.....	22
1.2.2 Εισαγωγή Εντολών με Χρήση του Μαθηματικού Συμβολισμού .....	23
1.2.3 Εισαγωγή και Επεξεργασία Κειμένου .....	23
1.2.4 Ενότητες.....	25
<b>1.3 Εντολές και Βασικοί Κανόνες Σύνταξης τους.....</b>	<b>26</b>
<b>1.4 Βοήθεια.....</b>	<b>30</b>
<b>1.5 Συντομεύσεις Πληκτρολόγιου .....</b>	<b>32</b>
<b>Κεφάλαιο 2-Αριθμητικοί Υπολογισμοί.....</b>	<b>35</b>
<b>2.1 Βασικές Πράξεις και Συναρτήσεις.....</b>	<b>36</b>
<b>2.2 Προσεγγιστικές και Ακριβείς Τιμές. ....</b>	<b>41</b>
<b>2.3 Αρίθμηση Αποτελεσμάτων –Συντομεύσεις.....</b>	<b>45</b>
<b>2.4 Απλοποίηση Αριθμητικών Εκφράσεων .....</b>	<b>47</b>
<b>2.5 Μιγαδικοί Αριθμοί.....</b>	<b>48</b>
<b>Κεφάλαιο 3-Μεταβλητές.....</b>	<b>51</b>
<b>3.1 Απόδοση Τιμής.....</b>	<b>52</b>
<b>3.2 Τύποι Μεταβλητών .....</b>	<b>56</b>
<b>3.3 Υποθέσεις σε Μεταβλητές .....</b>	<b>61</b>
<b>Κεφάλαιο 4-Αλγεβρικοί Υπολογισμοί .....</b>	<b>65</b>
<b>4.1 Πράξεις με Πολυωνυμικές Εκφράσεις .....</b>	<b>66</b>

4.2 Παραγοντοποίηση .....	68
4.3 Ανάπτυγμα .....	74
4.3 Απλοποίηση.....	78
4.5 Collect.....	86
4.6 Αντικατάσταση.....	87
<b>Κεφάλαιο 5-Βασικές Δομές Δεδομένων.....</b>	<b>91</b>
5.1 Ακολουθίες Εκφράσεων.....	92
5.2 Λίστες .....	94
5.3 Σύνολα .....	100
5.4 Πίνακες.....	102
<b>Κεφάλαιο 6-Διανύσματα-Μήτρες.....</b>	<b>105</b>
6.1 Διανύσματα .....	105
6.2 Μήτρες .....	108
6.3 Πράξεις με Διανύσματα και Μήτρες .....	114
6.4 Πακέτο Εντολών Linearalgebra .....	117
6.5 Λυμένες Ασκήσεις .....	132
<b>Κεφάλαιο 7-Λύση Εξισώσεων- Ανισώσεων.....</b>	<b>147</b>
7.1 Ακριβής Επίλυση Αλγεβρικών Εξισώσεων-Συστημάτων .....	148
7.2 Προσεγγιστική Επίλυση Αλγεβρικών Εξισώσεων .....	156
7.3 Λύση Ανισώσεων.....	163
<b>Κεφάλαιο 8-Ακολουθίες Αριθμών-Σειρές .....</b>	<b>167</b>
8.1 Ακολουθίες.....	167
8.2 Σειρές.....	170
8.3 Δυναμοσειρές .....	172
<b>Κεφάλαιο 9-Λογισμός Συναρτήσεων μιας Μεταβλητής.....</b>	<b>175</b>
9.1 Ορισμός Συνάρτησης .....	175
9.2 Όριο .....	180
9.2 Παραγωγή.....	184

9.3 Ανάπτυγμα Taylor .....	188
9.4 Ολοκλήρωση.....	189
<b>Κεφάλαιο 10-Γραφικές Παραστάσεις στο Επίπεδο .....</b>	<b>195</b>
10.1 Γραφική Παράσταση Συνάρτησης .....	195
10.2 Οδηγός Γραφικής Παράστασης.....	206
10.3 Γραφική Παράσταση Πολλών Συναρτήσεων .....	207
10.4 Σχεδιάζοντας Σημεία.....	208
10.5 Σημεία και Γραφικές Παραστάσεις.....	210
10.6 Γραφική Παράσταση Πεπλεγμένης Συνάρτησης.....	212
10.7 Γραφική Παράσταση Παραμετρικών Εξισώσεων .....	213
10.8 Γραφική Παράσταση Συνάρτησης σε Πολικές Συντεταγμένες .....	215
10.9 Γραφικές Παραστάσεις Βασικών Σχημάτων στο Επίπεδο.....	217
10.10 Κίνηση.....	223
<b>Κεφάλαιο 11-Λυμένες Ασκήσεις στις Συναρτήσεις μιας Μεταβλητής.....</b>	<b>227</b>
<b>Κεφάλαιο 12-Λογισμός Συναρτήσεων Πολλών Μεταβλητών ..</b>	<b>243</b>
12.1 Ορισμός Συνάρτησης .....	244
12.2 Όριο .....	246
12.3 Μερική Παράγωγος .....	248
12.4 Κατευθυνόμενη Παράγωγος.....	251
12.5 Πολλαπλή Ολοκλήρωση .....	253
<b>Κεφάλαιο 13-Γραφικές Παραστάσεις στο Χώρο .....</b>	<b>255</b>
13.1 Γραφική Παράσταση Συνάρτησης Δυο Μεταβλητών .....	255
13.2 Γραφική Παράσταση Συνάρτησης Πεπλεγμένης Μορφής.....	260
13.3 Γραφική Παράσταση Παραμετρικών Εξισώσεων .....	262
13.4 Ισοσταθμικές Καμπύλες .....	263
13.5 Εμφανίζοντας Πολλές Γραφικές Παραστάσεις.....	264
13.5 Βασικά Σχήματα στο Χώρο.....	265

<b>Κεφάλαιο 14-Λυμένες Ασκήσεις στις Συναρτήσεις Πολλών Μεταβλητών.....</b>	<b>271</b>
<b>Κεφάλαιο 15-Διαφορικές Εξισώσεις.....</b>	<b>281</b>
15.1 Συνήθεις Διαφορικές Εξισώσεις .....	281
15.2 Οικογένεια Ολοκληρωτικών Καμπυλών.....	287
15.3 Αριθμητική Επίλυση Διαφορικών Εξισώσεων.....	298
15.4 Διαφορικές Εξισώσεις με Μερικές Παραγώγους .....	303
<b>Κεφάλαιο 16-Προσεγγιστικές Μέθοδοι.....</b>	<b>311</b>
16.1 Μέθοδος Ελαχίστων Τετραγώνων .....	312
16.2 Παρεμβολή.....	316
16.3 Αριθμητική Ολοκλήρωση .....	322
<b>Κεφάλαιο 17-Στοιχεία Προγραμματισμού.....</b>	<b>329</b>
17.1 Εντολές Επανάληψης.....	330
17.2 Εντολές Ελέγχου .....	333
17.3 Ορίζοντας Διαδικασίες.....	335
17.4 Χρόνος υπολογισμών.....	339
17.5 Ανάγνωση-Εγγραφή σε Αρχεία .....	340
17.6 Εισαγωγή στα Maplets.....	341
<b>Ευρετήριο Εντολών .....</b>	<b>349</b>
<b>Βιβλιογραφία .....</b>	<b>355</b>

# Ευρετήριο Πινάκων

ΠΙΝΑΚΑΣ 1. ΣΥΝΤΟΜΕΥΣΕΙΣ ΠΛΗΚΤΡΟΛΟΓΙΟΥ	33
ΠΙΝΑΚΑΣ 2 ΒΑΣΙΚΩΝ ΠΡΑΞΕΩΝ ΚΑΙ ΣΥΝΑΡΤΗΣΕΩΝ	40
ΠΙΝΑΚΑΣ 3. ΒΑΣΙΚΩΝ ΠΡΑΞΕΩΝ ΚΑΙ ΣΥΝΑΡΤΗΣΕΩΝ	40
ΠΙΝΑΚΑΣ 4. ΕΝΤΟΛΕΣ ΓΙΑ ΤΟΥΣ ΑΚΕΡΑΙΟΥΣ	43
ΠΙΝΑΚΑΣ 5. ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ ΜΙΓΑΔΙΚΩΝ ΕΝΤΟΛΩΝ.	48
ΠΙΝΑΚΑΣ 6. ΠΡΑΞΕΙΣ ΜΕ ΠΟΛΥΩΝΥΜΑ	66
ΠΙΝΑΚΑΣ 7. ΕΝΤΟΛΕΣ ΓΙΑ ΤΑ ΠΟΛΥΩΝΥΜΑ.	66
ΠΙΝΑΚΑΣ 8. ΠΡΑΞΕΙΣ ΜΕ ΜΗΤΡΕΣ.	114
ΠΙΝΑΚΑΣ 9. ΕΙΔΙΚΕΣ ΜΗΤΡΕΣ ΚΑΙ ΔΙΑΝΥΣΜΑΤΑ.	119
ΠΙΝΑΚΑΣ 10. ΒΑΣΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ ΤΟΥ ΠΑΚΕΤΟΥ LINEARALGEBRA.	122
ΠΙΝΑΚΑΣ 11. ΒΑΣΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ ΤΟΥ ΠΑΚΕΤΟΥ LINEARALGEBRA.	123
ΠΙΝΑΚΑΣ 12. ΤΕΛΕΣΤΕΣ ΜΗΤΡΩΝ	125
ΠΙΝΑΚΑΣ 13. ΤΕΛΕΣΤΕΣ ΔΙΑΝΥΣΜΑΤΩΝ.	127
ΠΙΝΑΚΑΣ 14. ΕΙΔΙΚΕΣ ΜΟΡΦΕΣ ΜΗΤΡΩΝ ΤΟΥ ΠΑΚΕΤΟΥ LINEARALGEBRA.	129
ΠΙΝΑΚΑΣ 15. ΠΑΡΑΜΕΤΡΟΙ ΤΗΣ ΕΝΤΟΛΗ PLOT ΓΙΑ ΤΗΝ ΕΥΚΡΙΝΕΙΑ.	200
ΠΙΝΑΚΑΣ 16. ΠΑΡΑΜΕΤΡΟΙ ΤΗΣ ΕΝΤΟΛΗΣ PLOT ΓΙΑ ΚΕΙΜΕΝΟ.	205
ΠΙΝΑΚΑΣ 17. ΒΑΣΙΚΑ ΣΧΗΜΑΤΑ ΣΤΟ ΕΠΙΠΕΔΟ.	218
ΠΙΝΑΚΑΣ 18. ΠΑΡΑΜΕΤΡΩΝ ΤΗΣ PLOT3D	256
ΠΙΝΑΚΑΣ 19. ΠΑΡΑΜΕΤΡΩΝ ΤΗΣ PLOT3D	259
ΠΙΝΑΚΑΣ 20. ΠΑΡΑΜΕΤΡΟΙ ΤΗΣ PLOT3D (ΓΡΑΜΜΑΤΟΣΕΙΡΕΣ)	260
ΠΙΝΑΚΑΣ 21. ΒΑΣΙΚΑ ΣΧΗΜΑΤΑ ΣΤΟ ΧΩΡΟ	265
ΠΙΝΑΚΑΣ 22. ΔΥΝΑΤΕΣ ΤΙΜΕΣ ΤΗΣ ΠΑΡΑΜΕΤΡΟΥ METHOD ΓΙΑ ΤΗΝ ΑΡΙΘΜΗΤΙΚΗ ΕΠΙΛΥΣΗ ΜΙΑΣ Δ.Ε.	298

# Ευρετήριο Εικόνων

<i>EIKONA 1. COMMAND LINE MODE</i> .....	14
<i>EIKONA 2. STANDARD MAPLE 11, Η ΕΚΔΟΣΗ ΥΛΟΠΟΙΗΜΕΝΗ ΣΕ JAVA</i> .....	15
<i>EIKONA 3. MAPLE- ΕΙΣΑΓΩΓΗ ΕΝΤΟΛΩΝ</i> .....	16
<i>EIKONA 4. CLASSIC WORKSHEET MAPLE 11</i> .....	16
<i>EIKONA 5. FILE MENU</i> .....	17
<i>EIKONA 6. EDIT MENU</i> .....	17
<i>EIKONA 7. VIEW MENU</i> .....	18
<i>EIKONA 8. INSERT MENU</i> .....	18
<i>EIKONA 9. FORMAT MENU</i> .....	18
<i>EIKONA 10. ΕΡΓΑΛΕΙΟΘΗΚΗ</i> .....	19
<i>EIKONA 11. ΓΡΑΜΜΗ ΚΑΤΑΣΤΑΣΗΣ</i> .....	19
<i>EIKONA 12. ΣΥΝΤΟΜΕΥΣΕΩΝ ΣΤΟ STANDARD MAPLE ΠΕΡΙΒΑΛΛΟΝ</i> .....	20
<i>EIKONA 13. ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ</i> .....	21
<i>EIKONA 14. ΕΙΣΑΓΩΓΗ ΕΝΤΟΛΩΝ - 2D MAPLE INPUT</i> .....	23
<i>EIKONA 15. ΠΑΡΑΔΕΙΓΜΑ ΕΙΣΑΓΩΓΗΣ ΚΕΙΜΕΝΟΥ</i> .....	24
<i>EIKONA 16. ΟΡΙΣΜΟΣ ΑΡΧΕΙΟΥ ΕΛΛΗΝΙΚΟΥ ΟΡΘΟΓΡΑΦΟΥ</i> .....	24
<i>EIKONA 17. ΠΑΡΑΔΕΙΓΜΑ ΕΙΣΑΓΩΓΗΣ ΕΝΟΤΗΤΩΝ ΚΑΙ ΥΠΟΕΝΟΤΗΤΩΝ</i> .....	25
<i>EIKONA 18. ΠΟΛΛΑ ΦΥΛΛΑ ΕΡΓΑΣΙΑΣ</i> .....	25
<i>EIKONA 19. Ο ΚΩΔΙΚΑΣ ΥΛΟΠΟΙΗΣΗΣ (ΣΕ ΕΝΤΟΛΕΣ MAPLE) ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ ΤΟΥ ΛΟΓΑΡΙΘΜΟΥ LN</i> .....	28
<i>EIKONA 20. ΒΟΗΘΕΙΑ</i> .....	30
<i>EIKONA 21. TOPIC SEARCH</i> .....	32
<i>EIKONA 22. ΧΡΗΣΗ ΕΤΙΚΕΤΩΝ (LABEL)</i> .....	46




# Εισαγωγή

Τα συστήματα συμβολικής Άλγεβρας, Computer Algebra System (C.A.S.), είναι προηγμένα περιβάλλοντα εργασίας που μας επιτρέπουν να κάνουμε αριθμητικούς υπολογισμούς με ακριβείς ή προσεγγιστικές τιμές. Επίσης, μας επιτρέπουν να κάνουμε συμβολικούς υπολογισμούς (δηλαδή υπολογισμούς με σύμβολα των οποίων το αποτέλεσμα δεν είναι κατ'ανάγκη αριθμός).

Το 1969 στο εργαστήριο υπολογιστών του MIT δημιουργήθηκε το πρώτο C.A.S. σύστημα, γνωστό ως Macsyma, το οποίο αναπτύχθηκε σε γλώσσα προγραμματισμού Lisp. Σήμερα υπάρχει πληθώρα τέτοιων προγραμμάτων τόσο ανοικτού κώδικα, στα οποία έχει κανείς ελεύθερη πρόσβαση, όπως το Maxima, Octave κ.ά. όσο και «προστατευμένου» κώδικα όπως το Derive, Mathematica, Mathcad και το Maple.

Το Maple<sup>1</sup> ανήκει στην κατηγορία των συστημάτων συμβολικής άλγεβρας. Χρησιμοποιείται ευρέως από μαθητικούς, μηχανικούς και άλλους επιστήμονες, στα χέρια των οποίων αποτελεί ένα πολύ καλό επιστημονικό εργαλείο για έρευνα, αλλά και ένα σύγχρονο μέσο διδασκαλίας. Το Maple είναι ένα ισχυρό μαθηματικό πακέτο και μια ιδιαίτερα προηγμένη γλώσσα προγραμματισμού. Αναπτύχθηκε από το πανεπιστήμιο του Καναδά Waterloo και το ερευνητικό κέντρο ETH Zürich, ενώ η ανάπτυξή του έχει γίνει σε γλώσσα προγραμματισμού C. Στην Καναδική του

καταγωγή οφείλεται και το λογότυπό του  το οποίο μοιάζει με την καναδική σημαία. Από 1995 συνεχίζεται η ανάπτυξη και η προώθησή του σε συνεργασία με την εταιρία Waterloo Maple Software. Βρίσκεται ήδη στη 11<sup>η</sup> έκδοσή του και έχει αναπτυχθεί σε πολλές πλατφόρμες λειτουργικών συστημάτων, όπως Windows, Macintosh, Linux, Sun Solaris, HP-UX, DEC, Irix, AIX. Η ανάπτυξή του είναι συνεχής και ανοικτή σε όλους τους χρήστες του, στους οποίους παρέχεται η δυνατότητα να δημιουργήσουν βιβλιοθήκες προγραμμάτων και εντολών. Περισσότερα στοιχεία καθώς και έτοιμα φύλλα εργασίας μπορεί να βρει κανείς στο διαδίκτυο, στο Maple Application Center [www.mapleapps.com](http://www.mapleapps.com).

Στο βιβλίο αυτό παρουσιάζονται βασικά θέματα στα οποία το Maple παρέχει τη δυνατότητα για επίλυση και επεξεργασία. Έχει καταβληθεί προσπάθεια να παρουσιάζονται οι εντολές με αρκετές λεπτομέρειες, πράγμα όμως που δεν είναι πάντα δυνατόν λόγω των πολλών παραμέτρων που αυτές δέχονται οι εντολές.

Το βιβλίο αυτό αποτελείται από 17 κεφάλαια και η δομή τους είναι η εξής:

---

<sup>1</sup> © Waterloo Maple Inc.

Στο 1<sup>ο</sup> κεφάλαιο παρουσιάζεται το γραφικό περιβάλλον εργασίας, καθώς και οι πολλές ευκολίες που αυτό παρέχει μέσα από τα μενού και τις διάφορες συντομεύσεις. Επίσης, παρουσιάζονται κάποιοι βασικοί κανόνες για τη σύνταξη των εντολών.

Στο 2<sup>ο</sup> κεφάλαιο παρουσιάζεται ο τρόπος με τον οποίο μπορούμε να χρησιμοποιήσουμε το Maple ως αριθμομηχανή πολύ μεγάλης ακρίβειας.

Στο κεφάλαιο 3 παρουσιάζεται ο τρόπος με τον οποίο μπορούμε να ορίσουμε μεταβλητές και γίνεται ανάλυση των βασικών τύπων τους.

Στο κεφάλαιο 4 παρουσιάζονται οι αλγεβρικοί υπολογισμοί, καθώς και ο τρόπος με τον οποίο μπορούμε να ορίσουμε μαθηματικές εκφράσεις και να τις επεξεργαστούμε εφαρμόζοντας τις γνωστές ιδιότητες για την παραγοντοποίηση αλγεβρικών και αριθμητικών παραστάσεων, την απλοποίησή τους κ.ά.

Στα κεφάλαια 5 παρουσιάζονται οι βασικές δομές του Array, Table, λίστας και σύνολου.

Στο κεφάλαιο 6 παρουσιάζονται τα διανύσματα και οι μήτρες καθώς και το πακέτο εντολών για την Γραμμική Άλγεβρα που μας παρέχει τη δυνατότητα να κάνουμε λογισμό με Μήτρες και Διανύσματα.

Στο κεφάλαιο 7 επιλύονται εξισώσεις και συστήματα γραμμικά και μη γραμμικά, καθώς και ανισώσεις. Επίσης, παρουσιάζεται η αριθμητική επίλυση εξισώσεων και συστημάτων.

Στα κεφάλαια 8 παρουσιάζουμε τις εντολές εκείνες με τις οποίες μπορούμε να χειριστούμε ακολουθίες και σειρές.

Στα κεφάλαια 9 και 10 ασχολούμαστε με τις εντολές που μας επιτρέπουν να παρουσιάσουμε θέματα λογισμού μιας μεταβλητής. Δίνεται ο τρόπος ορισμού των συναρτήσεων και οι εντολές για τον υπολογισμό ορίων, παραγώγων και ολοκληρωμάτων. Επίσης, παρουσιάζονται αναλυτικά εντολές που μας επιτρέπουν να σχεδιάσουμε γραφικές παραστάσεις και σχήματα στο επίπεδο.

Στο κεφάλαιο 11 βλέπουμε κάποιες βασικές ασκήσεις στο λογισμό μιας μεταβλητής.

Στα κεφάλαια 12 και 13 παρουσιάζουμε τον τρόπο με τον οποίο μπορούμε να γενικεύσουμε τις εντολές του κεφαλαίου 9, ώστε να μπορούμε να αντιμετωπίσουμε θέματα από το λογισμό πολλών μεταβλητών. Επίσης, παρουσιάζονται αναλυτικά εντολές που μας επιτρέπουν να σχεδιάσουμε γραφικές παραστάσεις και σχήματα στο χώρο, ενώ παρατίθενται και κάποιες ασκήσεις για συναρτήσεις πολλών μεταβλητών.

Στο κεφάλαιο 14 βλέπουμε κάποιες βασικές ασκήσεις στο λογισμό συναρτήσεων πολλών μεταβλητών.

Στο κεφάλαιο 15 επιλύονται συνήθεις και μερικές διαφορικές εξισώσεις και παρουσιάζεται το πακέτο εντολών DEtools.

Στο κεφάλαιο 16 γίνεται μια παρουσίαση ορισμών βασικών προσεγγιστικών θεμάτων.

Τέλος στο κεφάλαιο 17 δίνονται στοιχεία προγραμματισμού και διασύνδεσης του Maple με άλλα προγράμματα.



# Κεφάλαιο 1

## Περιβάλλον Εργασίας - Στοιχεία Εντολών

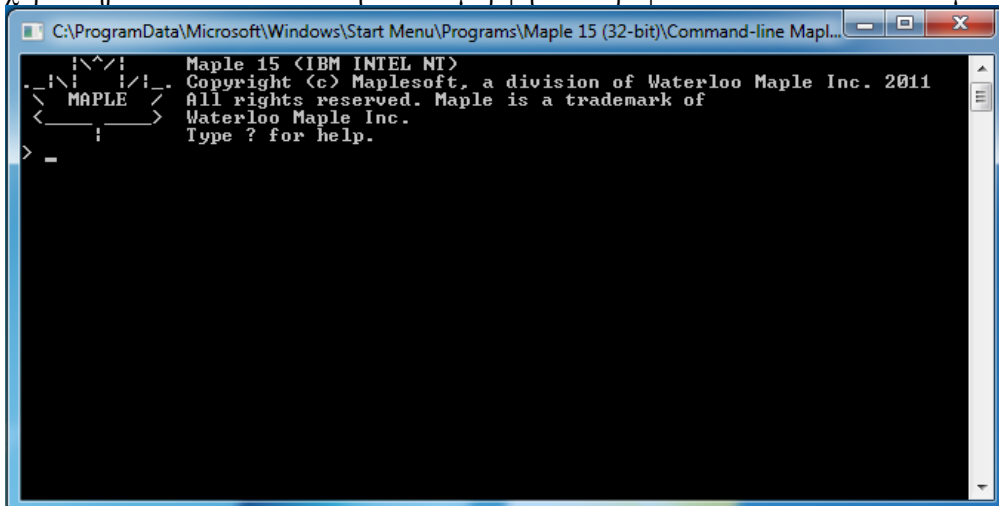
Το Maple, όπως πλέον και κάθε σύγχρονο υπολογιστικό - προγραμματιστικό περιβάλλον, είναι αρκετά φιλικό στο χρήστη και έχει τη δυνατότητα να εκτελείται εκτός από το περιβάλλον γραμμής εντολών (command line mode) και σε γραφικό («παραθυρικό») περιβάλλον. Αυτό σημαίνει ότι έχουμε όλες τις ευκολίες που παρέχει ένα γραφικό περιβάλλον επικοινωνίας, όπως τα μενού των εντολών, οι συντομεύσεις, η εύκολη πρόσβαση στα αρχεία βοήθειας, η δυνατότητα πολλαπλών παραθύρων κ.ά. Στο κεφάλαιο αυτό θα παρουσιάσουμε το περιβάλλον εργασίας του Maple και θα δούμε πώς μπορούμε να συνδυάσουμε και να μορφοποιήσουμε μαθηματικά κείμενα με το κύριο χαρακτηριστικό του Maple, που είναι οι πολύπλοκοι αλγεβρικοί και αριθμητικοί υπολογισμοί.

Το ακόλουθο υλικό έχει αναπτυχθεί χρησιμοποιώντας το Maple στην έκδοση 11 σε λειτουργικό Windows XP σε ένα σύστημα PC 4 2.66Mhz με 512 Ram μνήμης.

## 1.1 Το Πρόγραμμα

Το Maple στην έκδοση 15 δίνει τη δυνατότητα να ξεκινήσουμε το πρόγραμμα σε τρεις μορφές. Στη μορφή γραμμής εντολών Command Line Maple και σε δύο μορφές γραφικού περιβάλλοντος εργασίας, το (Standard) Maple 15 και το Classic Worksheet Maple 15.

Αν καλέσουμε το Maple σε Command Line Mode η εικόνα που θα έχουμε είναι αυτή που φαίνεται παρακάτω (εικόνα 1). Οι εντολές εδώ δίνονται αποκλειστικά με μορφή χαρακτήρων ASCII και στην ίδια μορφή επιστρέφονται και τα αποτελέσματα.



*Εικόνα 1. Command Line Mode*

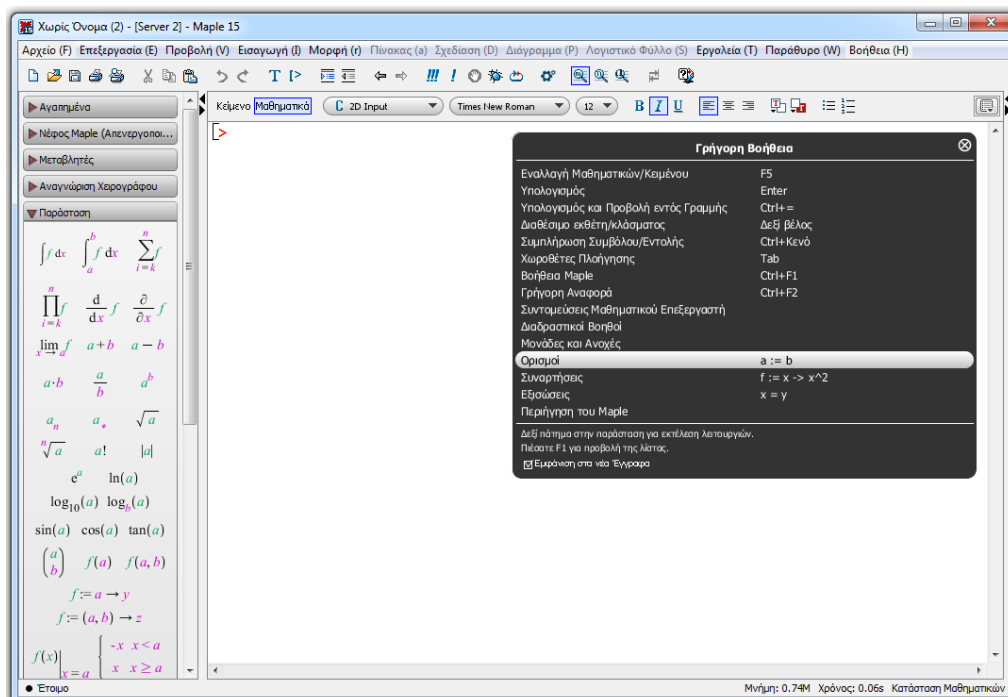
Στις επόμενες σελίδες θα ασχοληθούμε αποκλειστικά με το γραφικό περιβάλλον εργασίας. Ωστόσο, όσα θα παρουσιάσουμε ισχύουν και για την έκδοση γραμμής εντολών.

Οι δύο μορφές γραφικού περιβάλλοντος δε διαφέρουν μεταξύ τους σε τίποτα παρά μόνο στο ότι η πιο όμορφα σχεδιασμένη Standard Maple 11 είναι αναπτυγμένη σε Java και έχει περισσότερες απαιτήσεις σε πόρους του συστήματος, ενώ δίνει επιπλέον δυνατότητες σε συντομεύσεις και παλέτες εντολών.

Το γραφικό περιβάλλον εργασίας του Maple έχει πολλά κοινά στοιχεία με όλα τα γνωστά προγράμματα, όπως οι επεξεργαστές κειμένου (π.χ. Open Word, MS Word κ.α.), τα υπολογιστικά φύλλα (πχ. Excel). Αποτελείται από τη γνωστή οριζόντια μπάρα εντολών (Menu), από όπου μπορούμε να βρούμε βασικούς αλλά και προχωρημένους χειρισμούς του προγράμματος.

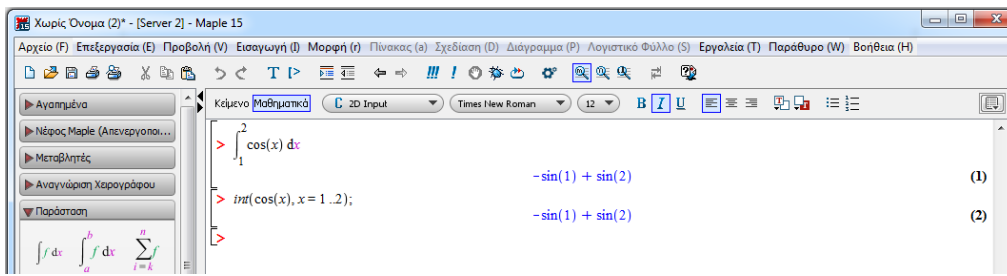
Στην παρακάτω εικόνα (Εικόνα 2) παρουσιάζεται το περιβάλλον εργασίας στη μορφή (Standard) Maple 11, όπου στο αριστερό μέρος μπορούμε να δούμε τις

παλέτες συντομεύσεων των εντολών οι οποίες δίνονται στην μαθηματική μορφή τους. Το περιβάλλον σε αυτή την έκδοση είναι εξελληνισμένο και σχεδόν κάθε επεξεργασία μπορεί να γίνει με χρήση των μενού και των υπομενού του προγράμματος.






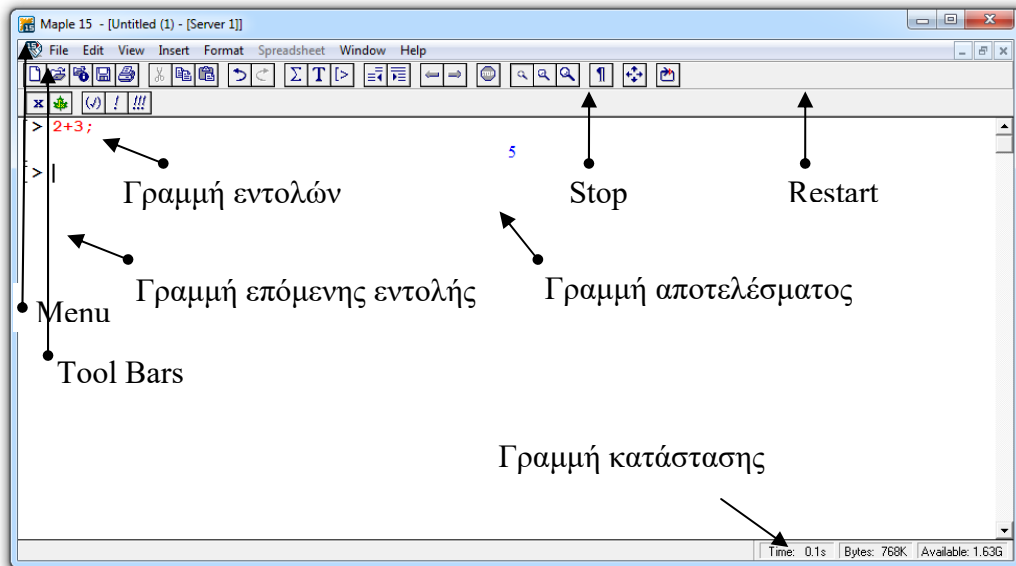
Εικόνα 2. Standard Maple 15, η έκδοση υλοποιημένη σε Java.

Στην εικόνα 3 παρουσιάζουμε ένα παράδειγμα υπολογισμού με το Maple, υπολογίζουμε το ολοκλήρωμα  $\int_1^2 \cos(x)dx$ , χρησιμοποιώντας πρώτα τις παλέτες (Document mode) που φαίνονται στο αριστερό μέρος, και στη συνέχεια το ίδιο ολοκλήρωμα, χρησιμοποιώντας την εντολή `int` (Worksheet mode).



**Εικόνα 3.** Maple- Εισαγωγή Εντολών

Στην εικόνα 4 βλέπουμε το «κλασικό» περιβάλλον του Maple (Classic Worksheet Maple), όπου παρουσιάζεται το παράθυρο με τους χειρισμούς κλεισίματος , αλλαγής μεγέθους , ελαχιστοποίησης  και μέσα σε αυτό οι εντολές αρχειοθέτησης, επεξεργασίας κ.ά. Επίσης, στην εικόνα 4 είναι ανοικτό ένα φύλλο εργασίας και παρουσιάζεται η πρόσθεση δύο αριθμών .



**Εικόνα 4.** Classic Worksheet Maple 15

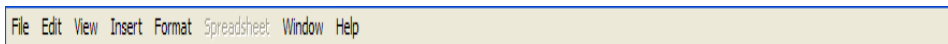
Στη συνέχεια του βιβλίου θα χρησιμοποιούμε αποκλειστικά την εισαγωγή εντολών στο περιβάλλον Maple Classic Worksheet. Όπου, ωστόσο, υπάρχουν σημαντικές διαφορές ανάμεσα στα δυο περιβάλλοντα, θα γίνεται σχετική αναφορά.

Στο σημείο αυτό θα παρουσιάσουμε εν συντομία κάποια βασικά μενού και υπομενού του περιβάλλοντος Classic Worksheet Maple.

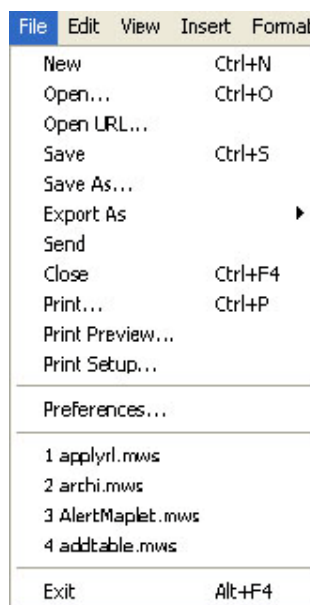
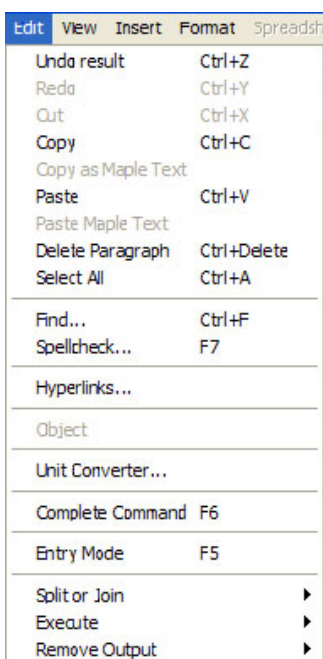


## ✚ Γραμμή Μενού (Menu Bar)

Στη γραμμή μενού υπάρχουν σε υπομενού οι εντολές για τη διαχείριση αρχείων (File), οι εντολές επεξεργασίας (Edit), οι εντολές προβολής (View), οι εντολές εισαγωγής (Insert), οι εντολές διαμόρφωσης (Format), οι εντολές Spreadsheet, οι εντολές Windows και οι εντολές βοήθειας (Help).



Στο υπομενού File (Εικόνα 5) μπορούμε να δημιουργήσουμε ένα νέο φύλλο εργασίας (New), να ανοίξουμε (Open), να αποθηκεύσουμε (Save), να μετατρέψουμε (Export As) ένα φύλλο εργασίας σε απλό κείμενο, ιστοσελίδα ή σε μορφή Latex, να εκτυπώσουμε, να θέσουμε τις ιδιότητες εκτύπωσης, να ορίσουμε τις βασικές λειτουργίες του προγράμματος και, βέβαια, να βγούμε από το πρόγραμμα.

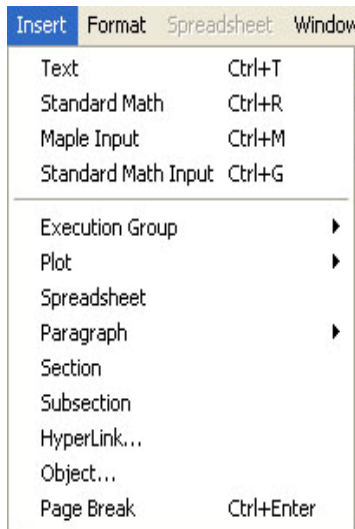


*Εικόνα 5. File Menu*

Στο υπομενού Edit (Εικόνα 6) είναι δυνατόν να εκτελέσουμε τις γνωστές λειτουργίες αντιγραφής (Copy Ctrl+C) και επικόλλησης (Paste Ctrl+V), να αναζητήσουμε κείμενο μέσα στο φύλλο εργασίας (Ctrl+F), να ενεργοποιήσουμε τον ορθογράφο (F7), να κάνουμε μετατροπή μονάδων και, κάτι ιδιαίτερα χρήσιμο, να συμπληρώσει το Maple μια εντολή που αρχίσαμε να γράφουμε (Complete Command) (F6) κ.ά.

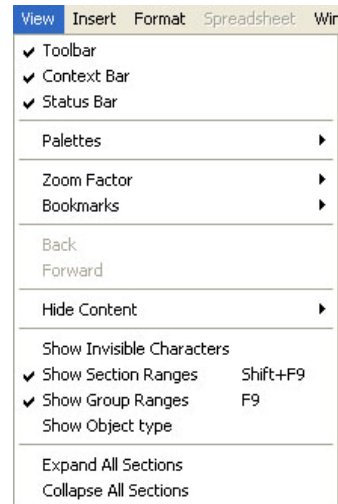
*Εικόνα 6. Edit Menu*

Στο υπομενού View (Εικόνα 6) μπορούμε να ενεργοποιήσουμε την εργαλειοθήκη, τη γραμμή περιεχομένων και τη γραμμή κατάστασης. Επίσης, μπορούμε να ανοίξουμε τις παλέτες συντομεύσεων, να ορίσουμε το μέγεθος γραμμάτων που θα βλέπουμε κ.ά.



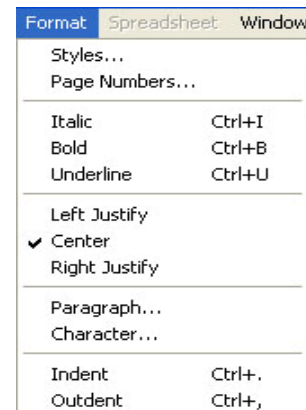
**Εικόνα 8. Insert Menu**

Από το μενού Format (Εικόνα 9) μπορούμε να μορφοποιήσουμε το κείμενο και τις εντολές που εισάγουμε, να ορίσουμε παραγράφους και να μετατρέψουμε απλό κείμενο σε εντολή του Maple κ.ά.



**Εικόνα 7. View Menu**

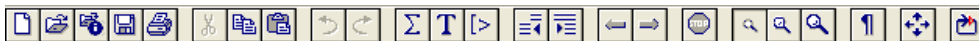
Από το υπομενού Insert (Εικόνα 8) μπορούμε να εισαγάγουμε κείμενο (Ctrl+T), να εισαγάγουμε μαθηματικά σύμβολα (Ctrl +R), να εισαγάγουμε μια εντολή του Maple (Ctrl +M), να εισαγάγουμε μια εντολή του Maple χρησιμοποιώντας των μαθηματικό συμβολισμό (Ctrl +G), να δημιουργήσουμε παραγράφους, γραφικές παραστάσεις τομής και υποτομής, καθώς και συνδέσμους με ιστοσελίδες, αλλά και να εισαγάγουμε διακοπή σελίδας κ.ά.



**Εικόνα 9. Format Menu**

### ✚ Εργαλειοθήκη (Tool Bar)

Στη γραμμή εργαλείων (Εικόνα 10) υπάρχουν συντομεύσεις εντολών, όπως η δημιουργία, το άνοιγμα, η αποθήκευση, η εκτύπωση ενός φύλλου εργασίας, οι βασικές εντολές αντιγραφής και επικόλλησης, οι εντολές δημιουργίας ενοτήτων, οι εντολές μεγέθυνσης, η εντολή επανεκκίνησης του πυρήνα του προγράμματος.



Εικόνα 10. Εργαλειοθήκη

### ✚ Γραμμή κατάστασης (Status Bar)

Από τη γραμμή κατάστασης (Εικόνα 11) μπορούμε να αντλήσουμε πληροφορίες για το χρόνο υπολογισμού, τη μνήμη που χρειάζεται το φύλλο εργασίας καθώς και τη συνολική μνήμη που απομένει.



Εικόνα 11. Γραμμή Κατάστασης

### ✚ Παλέτες Συντομεύσεων

Το Maple και στο περιβάλλον Classic Worksheet διαθέτει κάποιες παλέτες συντομεύσεων εντολών (για να εμφανιστούν οι παλέτες πρέπει να ενεργοποιηθούν από το μενού **View** ).

Αν από αυτές τις παλέτες επιλέξουμε στοιχεία, το Maple εισάγει αυτόματα την αντίστοιχη εντολή με τη μαθηματική της μορφή, χωρίς αυτή να εμφανίζεται στην οθόνη.

Για παράδειγμα, εάν από την παλέτα *expression* επιλέξουμε με το ποντίκι το



εικονίδιο τότε μπορούμε να υπολογίσουμε ένα ολοκλήρωμα χωρίς να γνωρίζουμε απαραίτητα την αντίστοιχη εντολή.

Οι διαθέσιμες παλέτες είναι:

Η παλέτα **Expression** που περιέχει συντομεύσεις για τις εντολές ορίων ολοκληρωμάτων, δυνάμεων, ριζών και βασικών συναρτήσεων.

EXPRESSION					
$\int^a$	$\int_b^c$	$\frac{d}{dx} a$	$\frac{d}{dx} a$	$\frac{\partial a}{\partial b}$	$\lim_{b \rightarrow c} a$
$a+b$	$a-b$	$a \times b$	$a/b$	$a=b$	$a \neq b$
$a^b$	$a_b$	$\sqrt{a}$	$\sqrt[n]{a}$	$a!$	$ a $
$e^a$	$\ln$	$\log$	$\sin$	$\cos$	$\tan$

ΣΥΜΒΟΛ											
α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ
ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω
A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M
N	Ξ	O	Π	P	Σ	T	Υ	Φ	X	Ψ	Ω
e	∞	π	i								

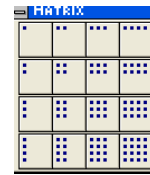
Η παλέτα **Symbols**

που περιέχει τους ελληνικούς χαρακτήρες καθώς και κάποιες σταθερές και σύμβολα  $e$ ,  $\pi$ ,  $i$ ,  $\infty$ .

Η παλέτα **Vector** η οποία δίνει τη δυνατότητα δημιουργίας διανυσμάτων.



Η παλέτα **Matrix** η οποία δίνει τη δυνατότητα δημιουργίας μητρώων οποιασδήποτε διάστασης.



### ✚ Παλέτες Συντομεύσεων στο Standard Maple περιβάλλον

Στο περιβάλλον Standard Maple διαθέτουμε επιπλέον παλέτες και συντομεύσεις όπως αυτές φαίνονται στην εικόνα 12. Για το ποιες από τις παλέτες θα εμφανίζονται μπορούμε να το ρυθμίσουμε από το μενού προβολή→παλέτες (εικόνα 2).

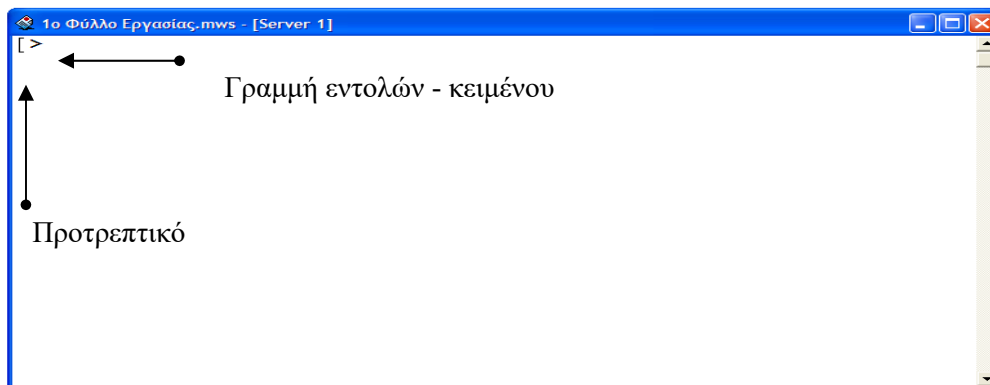


*Εικόνα 12. Συντομεύσεων στο Standard Maple περιβάλλον*

## 1.2 Φύλλο Εργασίας (Worksheet)

Όπως ήδη έχει γίνει αντιληπτό, για να είμαστε σε θέση να εργαστούμε με το Maple και να εισαγάγουμε εντολές, πρέπει πρώτα να ανοίξουμε ένα φύλλο εργασίας «worksheet». Αυτό γίνεται εύκολα από το μενού File→ New (Εικόνα 5).

Στην εικόνα 13 παρουσιάζεται ένα φύλλο εργασίας και το προτρεπτικό, το οποίο αναμένει από εμάς την εισαγωγή εντολών ή κειμένου.



*Εικόνα 13. Φύλλο Εργασίας*

Υπάρχουν τέσσερις καταστάσεις (Modes) για το προτρεπτικό ενός φύλλου εργασίας στο Classic Worksheet Maple:


- |   |                                  |
|---|----------------------------------|
| 1. Η κατάσταση εισαγωγής εντολών [ <b>&gt;</b>  | 1D Maple Input ( <b>Ctrl+M</b> ) |
| 2. Η κατάσταση εισαγωγής εντολών με εμφάνιση των αντίστοιχων μαθηματικών συμβολισμών τους. [ <b>&gt;?</b> | 2D Maple Input ( <b>Ctrl+G</b> ) |
| 3. Η κατάσταση εισαγωγής κειμένου [ <b>[</b>  | ( <b>Ctrl+T</b> )                |
| 4. Η κατάσταση εισαγωγής μαθηματικών συμβόλων [ <b>?]</b>   | ( <b>Ctrl+R</b> )                |



Στην συνέχεια θα παρουσιάσουμε αναλυτικά τις τέσσερις αυτές καταστάσεις εισαγωγής εντολών και κειμένου.

### 1.2.1 Εισαγωγή Εντολών

Για να εισαγάγουμε μια εντολή, πρέπει η γραμμή εντολής του συστήματος (system command) να βρίσκεται σε κατάσταση εισαγωγής εντολών, πρέπει δηλαδή το προτροπτικό να έχει την μορφή:

[>

Για να φέρουμε το προτροπτικό σε αυτή την κατάσταση, πρέπει να επιλέξουμε από την εργαλειοθήκη (Εικόνα 10) το εικονίδιο  ή να πατήσουμε συγχρόνως τα Ctrl+M.

-  Σε κάθε εντολή πρέπει να δηλώνουμε το τέλος της, χρησιμοποιώντας το ελληνικό ερωτηματικό “;”
-  Ο υπολογισμός γίνεται απλώς με το πάτημα του [ENTER].

Ας δούμε τώρα ένα απλό παράδειγμα:

Όταν ανοίγουμε για πρώτη φορά ένα φύλλο εργασίας, το προτροπτικό βρίσκεται ήδη στην κατάσταση εισαγωγή εντολής, οπότε μπορούμε να εισαγάγουμε την πρώτη μας εντολή.

> 2+3; [ENTER]

5


Αν θέλουμε να εισαγάγουμε περισσότερες εντολές στην ίδια ομάδα υπολογισμών, πατάμε [SHIFT] και [ENTER], ενώ το [ENTER] για τον υπολογισμό το πατάμε στο τέλος.

> 2+3; [SHIFT] και [ENTER]

6+7; [ENTER]

5

13

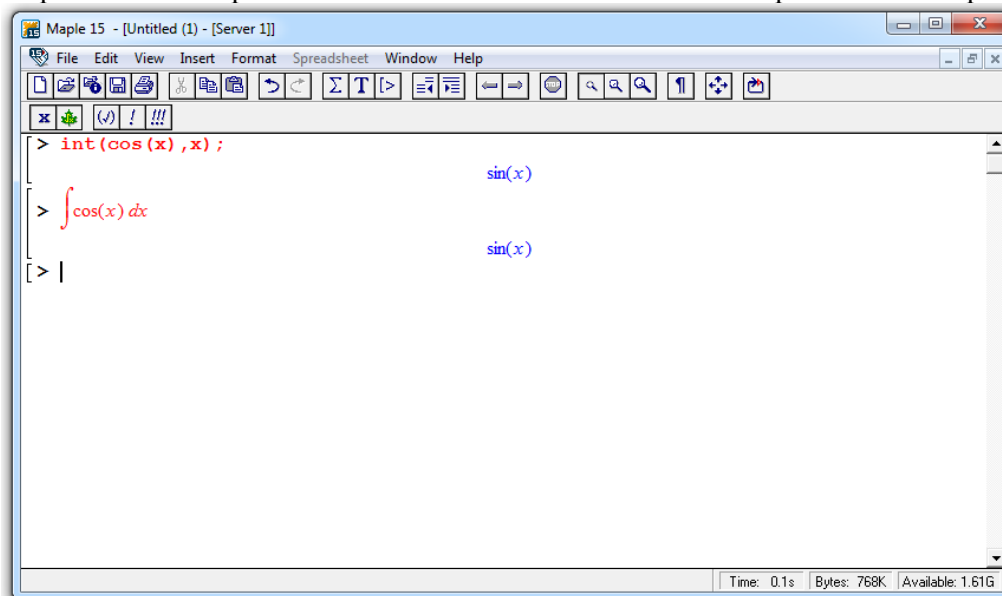
Για να διακόψουμε τους υπολογισμούς του Maple, πατάμε το κουμπί  της εργαλειοθήκης (Εικόνα 10).

#### Παρατήρηση: 1-1

Όταν εργαζόμαστε στη Standard Maple μορφή του προγράμματος, δεν είναι απαραίτητο να δηλώσουμε το τέλος της γραμμής χρησιμοποιώντας το ερωτηματικό.


## 1.2.2 Εισαγωγή Εντολών με Χρήση του Μαθηματικού Συμβολισμού


Είναι δυνατόν να εισαγάγουμε μια εντολή με χρήση του «κλασικού» μαθηματικού συμβολισμού της. Για να γίνει αυτό, πρέπει να επιλέξουμε από το μενού Insert->Standard Maple input ή να πατήσουμε συγχρόνως τα Ctrl+G. Σε αυτή την περίπτωση, το προτροπτικό έχει την μορφή [ $\int$ ? και οι εντολές δίνονται σε ειδικό χώρο, κάτω από την μπάρα του μενού, ή επιλέγονται από τις παλέτες. Στην εικόνα 14 έχουμε ένα παράδειγμα για την εισαγωγή μιας εντολής και με τους δύο τρόπους (1D Maple Input και 2D Maple Input)



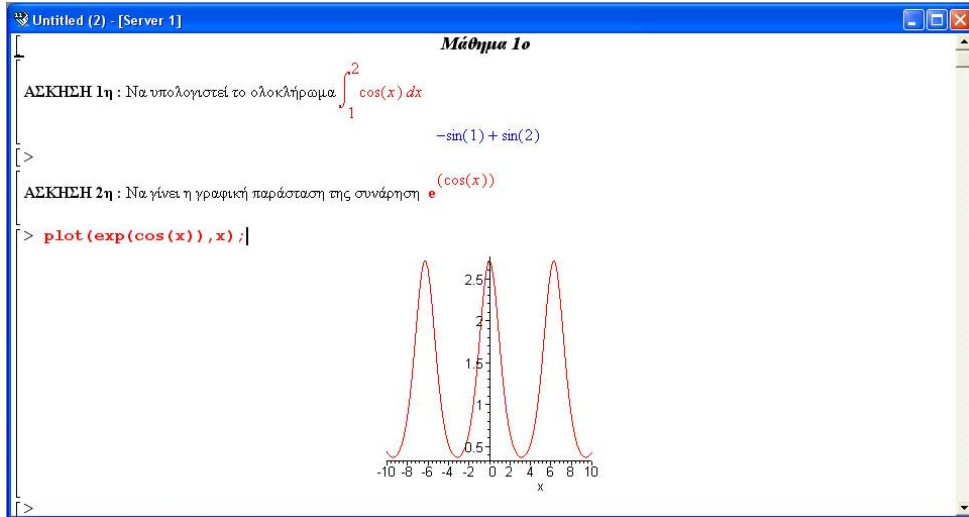
Εικόνα 14. Εισαγωγή εντολών - 2D Maple Input

## 1.2.3 Εισαγωγή και Επεξεργασία Κειμένου

Σε ένα φύλλο εργασίας είναι δυνατό να εισαγάγουμε κείμενο, το οποίο μπορούμε να το μορφοποιήσουμε (να ορίσουμε τη γραμματοσειρά, να το στοιχίσουμε κ.ά.). Για την εισαγωγή κειμένου πρέπει να επιλέξουμε από την εργαλειοθήκη (Εικόνα 10) το εικονίδιο  ή να πατήσουμε συγχρόνως τα πλήκτρα Ctrl+T για να έλθει το προτροπτικό στην κατάσταση κειμένου, δηλαδή στην μορφή [.

Επίσης, είναι δυνατόν να εισαγάγουμε μαθηματικά σύμβολα και εκφράσεις μέσα σε κείμενο, επιλέγοντας από την εργαλειοθήκη το εικονίδιο  ή πατώντας συγχρόνως

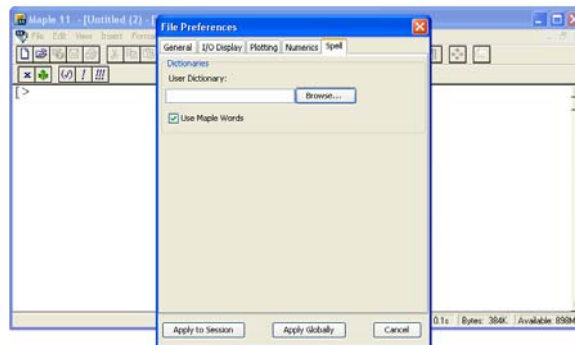
τα Ctrl+R. Στην περίπτωση αυτή, οι μαθηματικοί συμβολισμοί που παρεμβάλλονται στο κείμενο δεν υπολογίζονται. Αν, ωστόσο, θέλουμε να παρεμβάλουμε σε κείμενο μαθηματικούς συμβολισμούς, οι οποίοι και να εκτελούνται πατώντας το [ENTER], τότε τους εισάγουμε πατώντας Ctrl+G.



*Εικόνα 15. Παράδειγμα Εισαγωγής Κειμένου*

Ένα πολύ χρήσιμο εργαλείο που υπάρχει στο Maple είναι ο ορθογράφος. Σε μικτά κείμενα, που αποτελούνται από κείμενο και εντολές, μπορούμε να κάνουμε ορθογραφικό έλεγχο στο κείμενο, πατώντας το F7 ή από το μενού Edit-> SpellCheck


Για να ορίσουμε ελληνικό ορθογράφο, πρέπει από το Menu->Preferences να ορίσουμε το αρχείο του ελληνικού ορθογράφου. Το Maple δε συνοδεύεται από αρχείο Ελληνικής ορθογραφίας, αλλά μπορούμε εμείς να ορίζουμε οποιοδήποτε αρχείο έχουμε σε μορφή txt.





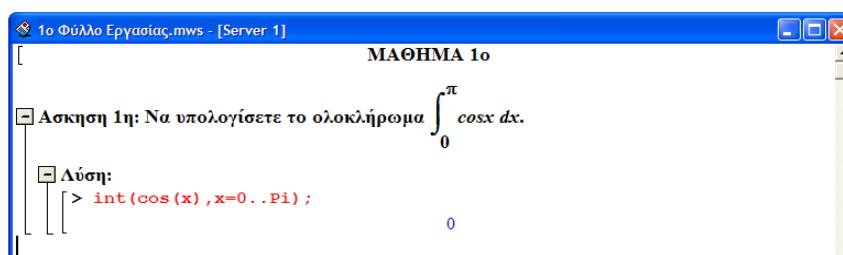
*Εικόνα 16. Ορισμός Αρχείου Ελληνικού Ορθογράφου.*



## 1.2.4 Ενότητες

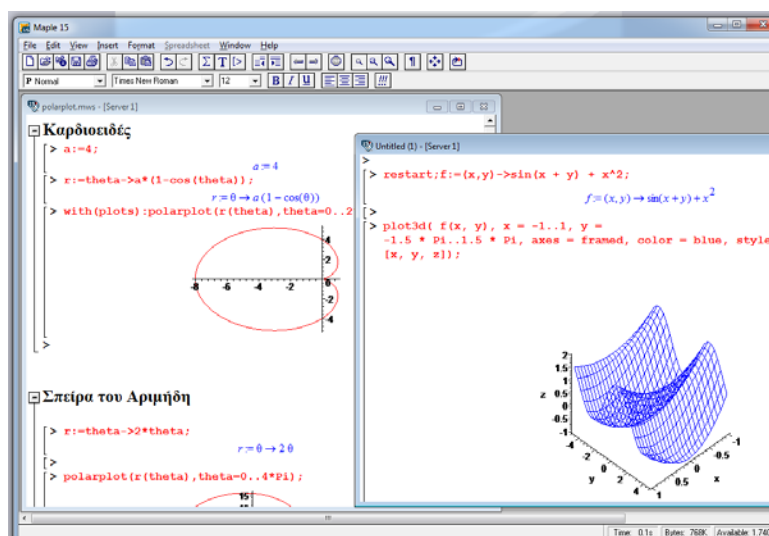
Ένα μικτό φύλλο εργασίας που αποτελείται από κείμενο αλλά και εντολές μπορούμε να το χωρίσουμε σε ενότητες (sections) επιλέγοντας από την εργαλειοθήκη το εικονίδιο  ή από το μενού Insert->Section (Εικόνα 8).

Έτσι, έχουμε την δυνατότητα να δημιουργήσουμε φύλλα εργασίας, στα οποία μπορούμε να κρύβουμε κάποια κομμάτια από το κείμενο ή τις εντολές (π.χ. τη λύση μιας άσκησης). Το  που παρουσιάζεται δείχνει ότι μια υποενότητα είναι ήδη ανοιχτή. Αντίθετα, εάν σε μια υποενότητα δεν παρουσιάζεται, τότε στη θέση του – βλέπουμε το σύμβολο .



Εικόνα 17. Παράδειγμα Εισαγωγής Ενότητας και Υποενότητας.

Επιπλέον, έχουμε τη δυνατότητα να δουλεύουμε παράλληλα περισσότερα από ένα φύλλα εργασίας, όπως φαίνεται στην εικόνα 18.



Εικόνα 18. Πολλά Φύλλα Εργασίας

**Παρατήρηση: 1-2**

Όταν δουλεύουμε στο Standard Maple, οι καταστάσεις για την εισαγωγή εντολών και κειμένου χωρίζονται σε Document mode και Worksheet mode:

Στην κατάσταση **Document mode** εισάγουμε το πρόβλημά μας στη μαθηματική του μορφή χρησιμοποιώντας τις παλέτες χωρίς να εμφανίζεται το προτρεπτικό. Για να ξεκινήσουμε σε Document mode επιλέγουμε από το μενού **File** το **New** και στη συνέχεια επιλέγουμε το **Document Mode**.

Στην κατάσταση **Worksheet mode** όπως ήδη έχουμε αναφέρει μπορούμε να κάνουμε έναν υπολογισμό είτε εισάγοντας την αντίστοιχη εντολή από το πληκτρολόγιο (1D Math Input), είτε δίνοντας το αντίστοιχο μαθηματικό (2D Math Input) συμβολισμό από τις παλέτες. Για να ξεκινήσουμε σε Document mode, επιλέγουμε από το μενού **File** το **New** και στη συνέχεια επιλέγουμε το **Worksheet mode**.

```
> int(cos(x), x);
```

1D Math Input

```
> ∫cos(x) dx
```

2D Math Input

**1.3 Εντολές και Βασικοί Κανόνες Σύνταξης τους**

Το Maple αποτελείται από χιλιάδες εντολές και συναρτήσεις. Αυτές χωρίζονται σε δύο βασικές κατηγορίες, εκείνες που είναι ενσωματωμένες (built in) στον πυρήνα (Kernel) του προγράμματος και εκείνες που βρίσκονται σε βιβλιοθήκες εντολών (library commands).

Οι εντολές που βρίσκονται στον πυρήνα του προγράμματος ενεργοποιούνται από την στιγμή της έναρξης του Maple και είναι διαθέσιμες πάντα. Είναι λίγες σε πλήθος και είναι γραμμένες σε γλώσσα προγραμματισμού C. Αν ζητήσουμε πληροφορίες για μια εντολή του πυρήνα, το Maple δε θα μας επιστρέψει τίποτα για τον τρόπο υλοποίησής της. Για παράδειγμα, αν ζητήσουμε πληροφορίες για την εντολή `evalf` (η εντολή αυτή, όπως θα δούμε στη συνέχεια, επιστρέφει την προσεγγιστική τιμή μιας παράστασης).

```
> interface( verboseproc=2 );
```

```
> print(evalf);
```

**proc() option builtin = evalf, remember; end proc**

Το Maple απλώς μας αναφέρει ότι η εντολή `evalf` είναι εντολή του πυρήνα εντολών.

Οι εντολές των βιβλιοθηκών ανήκουν είτε στην κύρια βιβλιοθήκη (Main Library) είτε σε πακέτα (packages). Οι εντολές της κύριας βιβλιοθήκης είναι διαθέσιμες, όπως και οι εντολές του πυρήνα από τη στιγμή που το Maple ξεκινά, αντίθετα οι εντολές των πακέτων είναι διαθέσιμες μόνο μετά από την ενεργοποίηση του πακέτου.

Οι εντολές των βιβλιοθηκών είναι γραμμένες στη γλώσσα προγραμματισμού Maple. Αν ζητήσουμε πληροφορίες για την εντολή `ln` (είναι εντολή της κύριας βιβλιοθήκης εντολών), τότε το Maple μας επιστρέφει το υποπρόγραμμα που την υλοποιεί σε γλώσσα Maple. Αυτό είναι ιδιαίτερα χρήσιμο αν θελήσουμε να επεκτείνουμε μια ήδη υπάρχουσα εντολή, διαβάζοντας τον κώδικά της και κάνοντας τις απαραίτητες μετατροπές.

```
> interface( verboseproc=2 );
> print(ln);
proc(x::algebraic)
local res, k, b;
option `Copyright (c) 1992 by the University of Waterloo. All rights reserved.`;
if nargs ≠ 1 then error "expecting 1 argument, got %1", nargs
elif type(x, 'complex(float)') then return evalf('ln'(x))
elif x = 0 then return NumericEvent('real_to_complex',
    NumericEvent('division_by_zero', -∞ + undefined×I))
elif type(x, '∞') then
    if type(x, 'neg_infinity') then
        return NumericEvent('real_to_complex', ∞ + π×I)
    else res := ∞ + `if` (type(x, 'pos_infinity'), 0, argument(x)×I)
    end if
elif type(x, 'undefined') then res := NumericTools:-ThrowUndefined(x)
elif type(x, 'negative') then
    return NumericEvent('real_to_complex', ln(-x) + π×I)
elif type(x, 'function') and op(0, x) = 'exp' and ℑ(op(1, x)) = 0 then
    res := op(1, x)
elif type(x, 'posint') and iperfpow(x) ≠ FAIL then
    b := iperfpow(x, 'k'); res := k×ln(b)
elif type(x, 'rational') and numer(x) = 1 and 0 < x then res := -ln(denom(x))
```

```

elif type(x, 'SymbolicInfinity') and type(signum(0, x, undefined), 'numeric') then
    if 0 < signum(0, x, undefined) then res := ∞
    else return NumericEvent('real_to_complex', ∞ + π×I)
    end if
elif type(x, 'constant^numeric') and signum(op(1, x)) = 1 then
    res := op(2, x)×ln(op(1, x))
elif type(x, '^') and (signum(0, abs(op(2, x)) - 1, 0) = -1 or
    signum(0, abs(op(2, x)) - 1, -1) = -1 and coulditbe(op(1, x) < 0) = false or
    signum(0, abs(op(2, x)) - 2, 0) = -1 and signum(0, ℑ(op(1, x)), 1) = 1 or
    signum(0, abs(op(2, x)) - 2, -1) = -1 and signum(0, ℑ(op(1, x)), 0) = 1) then
    res := op(2, x)×ln(op(1, x))
elif not type(procname, 'indexed') then res := 'ln'(x)
else return 'procname'(x)
end if ;
if type(res, 'nonreal') and not hastype([args], 'nonreal') then
    NumericEvent('real_to_complex', res)
else ln(args) := res
end if
end proc

```

*Εικόνα 19. Ο Κώδικας Υλοποίησης (σε εντολές Maple) της συνάρτησης του λογαρίθμου ln*

Σε όλες τις περιπτώσεις οι εντολές αυτές ακολουθούν κάποιους βασικούς κανόνες συντακτικού, τους εξής:

- ◇ Γίνετε διαχωρισμός κεφαλαίων και πεζών γραμμάτων.
- ◇ Οι εντολές και οι δεσμευμένες λέξεις γράφονται πάντα με μικρά γράμματα.

Αυτός ο κανόνας ωστόσο έχει αρκετές εξαιρέσεις, για παράδειγμα οι σταθερές  $\pi$  και  $i$ , όπου οι εντολές είναι  $\text{Pi}$  και  $\text{I}$  αντίστοιχα. Επίσης οι εντολές του πακέτου LinearAlgebra ξεκινάνε όλες με κεφαλαίο γράμμα βλέπε κεφάλαιο 6.4, και οι εντολές που τυπώνουν το αντίστοιχο μαθηματικό σύμβολο ξεκινάνε με κεφάλαιο γράμμα παράδειγμα  $\text{Int}$  και  $\text{int}$  βλέπε παράδειγμα 9.2 .

- ◇ Δεν επιτρέπεται να ορίσουμε μεταβλητή με το όνομα είδη δεσμευμένης λέξης (εντολής) (περισσότερα για τον ορισμό μεταβλητών στην παράγραφο 3.2).
- ◇ Οι παράμετροι των εντολών τοποθετούνται σε παρενθέσεις και στην περίπτωση που είναι περισσότερες από μία χωρίζονται με κόμμα.
- ◇ Μια εντολή για να εκτελεστεί πρέπει να ακολουθείται από ελληνικό ερωτηματικό ; ή : αν δεν θέλουμε να δούμε στην οθόνη το αποτέλεσμα της εντολής.

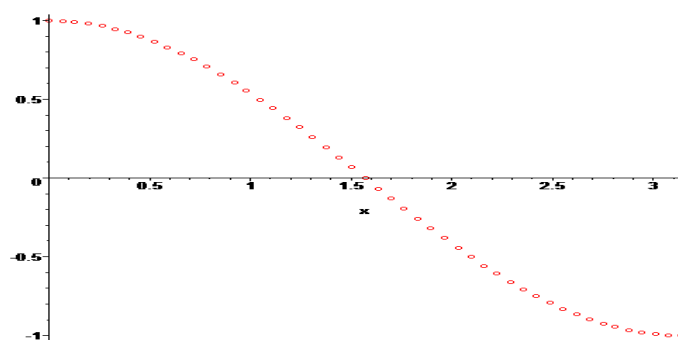
**εντολή( παράμετρος1, παράμετρος2... );**

Ας δούμε ένα παράδειγμα εντολής

```
> plot(cos(x), x=0..Pi, style=point, symbol=circle);
```

Η παραπάνω εντολή κατασκευάζει τη γραφική παράσταση της έκφρασης  $\cos(x)$  στο διάστημα  $[0, \pi]$ . Η εντολή είναι γραμμένη με μικρούς χαρακτήρες και οι παράμετροι της εντολής βρίσκονται μέσα σε παρενθέσεις και χωρίζονται με κόμματα.

Αν μετά την παραπάνω εντολή πατήσουμε [ENTER] τότε θα έχουμε το αποτέλεσμα:



### Μηνύματα Λάθους

Το Maple από τη στιγμή που θα πατήσουμε το Enter θα προσπαθήσει να εκτελέσει την εντολή που του δώσαμε. Εάν σε αυτή υπάρχει κάποιο λάθος, θα μας επιστρέψει το αντίστοιχο μήνυμα.

Έτσι, στην παρακάτω εντολή δεν έχουμε δηλώσει το τέλος γραμμής με το ; και το Maple επιστρέφει μήνυμα λάθους.

```
> plot(cos(x), x=0..Pi, style=point, symbol=circle)
```

Warning, premature end of input

Στην παρακάτω εντολή δεν έχουμε κλείσει σωστά τις παρενθέσεις και έτσι το σύμβολο ; είναι απρόσμενο για το πρόγραμμα.

```
> plot(cos(x), x=0..Pi, style=point, symbol=circle;
```

Error, `;` unexpected

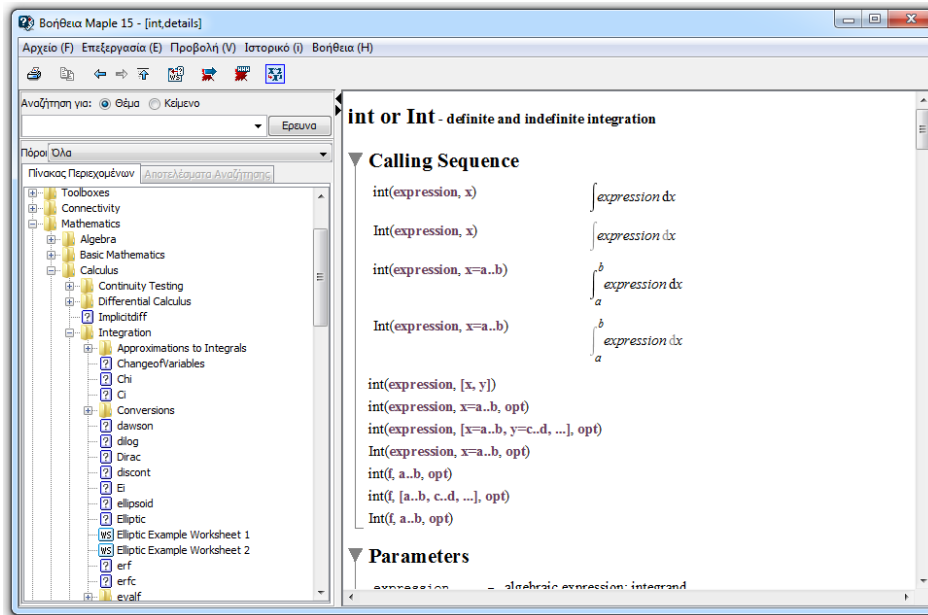
Ωστόσο, αν υπάρχει λάθος στο όνομα μιας εντολής, για παράδειγμα, έχει δοθεί με κεφαλαίους αντί για μικρούς χαρακτήρες, τότε το Maple δεν επιστρέφει κάποιο μήνυμα λάθους, αλλά επαναλαμβάνει την εντολή.

```
> Plot(cos(x), x=0..Pi, style=point, symbol=circle);
```

*Plot(cos(x), x = 0 .. π, style = point, symbol = circle)*

## 1.4 Βοήθεια

Για την ακριβή σύνταξη των εντολών, παραδείγματα, λεπτομέρειες για τις μεθόδους που χρησιμοποιούνται καθώς και νέα στοιχεία για κάθε έκδοση του προγράμματος, μπορούμε να καλέσουμε τη βοήθεια που έχει το Maple από το μενού Help.



Εικόνα 20. Βοήθεια

Σε οποιοδήποτε σημείο του φύλλου εργασίας μπορούμε να καλέσουμε τη βοήθεια για μία εντολή πληκτρολογώντας ?εντολή. Αν εκτελέσουμε αυτήν την εντολή, θα

ανοίξει αυτόματα το παράθυρο της βοήθειας στην αντίστοιχη παράγραφο, όπου περιέχονται λεπτομέρειες για την εντολή. Για παράδειγμα, αν σε ένα φύλλο εργασίας πληκτρολογήσουμε `?plot`, θα ενεργοποιηθεί η βοήθεια και θα εμφανιστεί η Εικόνα 20. Βοήθεια.

Επίσης με την εντολή `usage` (εντολή) ; μπορούμε να έχουμε μια σύνοψη των παραμέτρων της εντολής.

```
> usage (plot) ;
```

**plot - create a two-dimensional plot**

**Calling Sequence**

```
plot(f, x)
plot(f, x=x0..x1)
```

**Parameters**

- `f` - expression in independent variable `x`
- `x` - independent variable
- `x0` - left endpoint of horizontal range
- `x1` - right endpoint of horizontal range

**Description**

**Examples**

**Details**

**See Also**

Με την εντολή `example` (εντολή) ; η βοήθεια για την εντολή μας δίνει παραδείγματα της χρήσης της.

```
> example (plot) ;
```

**Examples**

```
> plot(cos(x) + sin(x), x=0..Pi);
plot(tan(x), x=-Pi..Pi);
plot([sin(t), cos(t), t=-Pi..Pi]);
plot(sin(t), t);
```

Since no domain is specified in the last example, we use `t=-10..10`.

Same four plots, but as procedures or operators:

```
> plot(cos + sin, 0..Pi);
plot(tan, -Pi..Pi);
plot([sin, cos, -Pi..Pi]);
plot(sin);
```

For expressions having discontinuities over finite intervals one can do:

```
> plot(tan(x), x = -2*Pi..2*Pi, y = -4..4, discontin = true);
plot(-1 + 2*Heaviside(x-1), x = -1..2, discontin = true);
```

Multiple plots (in a set or list):

```
> plot([sin(x), x-x^3/6], x=0..2, color=[red,blue], style=[point,line]);
```

Επίσης, μέσα από το μενού `Help` μας δίνεται η δυνατότητα να κάνουμε αναζήτηση, ώστε να βρεθούν όλες οι σχετικές σελίδες βοήθειας είτε ανά θέμα (`Help-> Topic Search`) είτε με βοήθεια κάποιων λέξεων κλειδιών (`Help-> Full Text Search`)





Alt + T	Ανοίγει το μενού εντολών Tools.
Alt + W	Ανοίγει το μενού εντολών Window.
Alt + H	Ανοίγει το μενού εντολών Help.
Alt + F + D	Επισυνάπτει το φύλλο εργασίας σε email.
Ctrl + N	Δημιουργεί ένα νέο φύλλο εργασίας.
Ctrl + O	Ανοίγει ένα φύλλο εργασίας.
Ctrl + F4	Κλείνει το ενεργό φύλλο εργασίας.
Ctrl + S	Αποθηκεύει ένα φύλλο εργασίας.
Ctrl + P	Τυπώνει ένα φύλλο εργασίας.
Ctrl + Shift + P	Page Setup.
Alt + F4	Κλεινή το Maple.
Ctrl + Z	Undo
Ctrl + Y	Redo
Ctrl + X	Αποκόβει κείμενο ή εντολές.
Ctrl + C	Αντιγράφει κείμενο ή εντολές.
Ctrl + V	Επικολλεί κείμενο ή εντολές.

**Πίνακας 1.** Συντομεύσεις Πληκτρολογίου

Στην, συνέχεια του βιβλίου θα παρουσιάζουμε κάθε εντολή συνοπτικά σε έναν πίνακα, όπου πέρα από τη περιγραφή της και τη γενική της σύνταξη θα παρουσιάζεται και η δυνατότητα επιλογή της από τις παλέτες των έτοιμων εκφράσεων και εντολών, όταν αυτή βέβαια είναι διαθέσιμη .

<b>Περιγραφή:</b>	
<b>Γενική Σύνταξη:</b>	
<b>Expression Palettes</b>	



# Κεφάλαιο 2

## Αριθμητικοί Υπολογισμοί

Στο κεφάλαιο αυτό θα δούμε πώς εισάγονται οι βασικές πράξεις και συναρτήσεις και θα χρησιμοποιήσουμε το Maple σαν ένα πολύ ισχυρό κομπιουτεράκι, που όμως έχει τη δυνατότητα υπολογισμών μεγάλης ακρίβειας. Αυτό οφείλεται στο γεγονός ότι το Maple κάνει διαχωρισμό των αριθμητικών και των συμβολικών πράξεων. Κάνει πράξεις συμβολικά, όπου αυτό είναι δυνατό, για παράδειγμα στις πράξεις κλασμάτων, στις πράξεις όπου εμφανίζονται εκφράσεις όπως  $\pi$ ,  $e$ ,  $\sqrt{2}$  και γενικά άρρητοι αριθμοί. Λέγοντας ότι κάνει τις πράξεις συμβολικά εννοούμε ότι «βλέπει» τους αριθμούς αυτούς σαν μαθηματικά αντικείμενα χωρίς να θεωρεί τις αντίστοιχες δεκαδικές προσεγγιστικές τιμές τους.

## 2.1 Βασικές Πράξεις και Συναρτήσεις

Οι βασικές πράξεις είναι η πρόσθεση που δηλώνεται με +, η αφαίρεση με -, ο πολλαπλασιασμός με \* (είναι συχνό λάθος να παραλείπεται το αστεράκι \* [Shift+8]) και η διαίρεση με /.

Ας ξεκινήσουμε ζητώντας πράξεις ανάμεσα σε ακέραιους αριθμούς.

Προσθέτουμε τους αριθμούς 2 και 4:

```
> 2+4 [ENTER]
```

Warning, premature end of input

Το μήνυμα λάθους που δεχόμαστε οφείλεται στο γεγονός ότι δε δηλώσαμε το τέλος της εντολής με το σύμβολο ;. Αυτό δεν πρέπει να το ξεχνάμε ποτέ στο τέλος μια εντολής.

Ας κάνουμε πάλι την πρόσθεση:

```
> 2+4; [ENTER]
```

6

Το γινόμενο  $-3 \cdot 4$

```
> -3*4; [ENTER]
```

-12

Συνεχίζουμε παρουσιάζοντας παραδείγματα πράξεων και υπολογισμών, όπως είναι οι δυνάμεις και ο υπολογισμός του παραγοντικού.

Ο υπολογισμός της δύναμης  $4^{21}$

```
> 4^21; [ENTER]
```

4398046511104

Το σύμβολο της δύναμης ^ εισάγεται με τα πλήκτρα [Shift+6].

Υπολογισμός του  $10!$

```
> 10!; [ENTER]
```

3628800

Βέβαια, υπάρχουν περιορισμοί στις πράξεις που μπορούμε να κάνουμε με το Maple. Για παράδειγμα, αν του ζητήσουμε να υπολογίσει την δύναμη:  $123456789^{123456789}$ , θα δεχθούμε μήνυμα λάθους.

```
> 123456789^123456789; [ENTER]
```

```
Error, numeric exception: overflow
```

Η αιτία του λάθους που το πρόγραμμα μας επιστρέφει ως απάντηση είναι ότι έχουμε βγει εκτός των ορίων των ακεραίων αριθμών που μπορεί να αναπαραστήσει ο υπολογιστής.

Με την εντολή `kernelopts(maxdigits)` μπορούμε να δούμε το μέγιστο πλήθος ψηφίων:

```
> kernelopts(maxdigits); [ENTER]
```

```
268435448
```

Ας δούμε τον υπολογισμό μιας δύναμης όπου το αποτέλεσμα είναι ένα «μεγάλος» αριθμός:

```
> 123456789^30; [ENTER]
```

```
556479685757868675258508744085500475509416318102469581179
163094403782079923208229189828895770817595030148760102
378174273722559243833793347276390947626956403726506972
769455855761863337658368312401395856008299464503576992
811687453871135433889801
```

Εδώ πρέπει να παρατηρήσουμε ότι, αν μια έξοδος - αποτέλεσμα στο Maple συνεχίζεται στην επόμενη γραμμή, αυτό δηλώνεται με το σύμβολο `\`.

Με την εντολή `length` μπορούμε να προσδιορίσουμε τον αριθμό των ψηφίων μιας αριθμητικής παράστασης.

```
> length(123456789^30); [ENTER]
```

```
243
```

<b>Περιγραφή:</b>	<b>Υπολογίζει τον αριθμό των ψηφίων ενός αριθμού.</b>
<b>Γενική σύνταξη:</b>	<b><code>length(αριθμός);</code></b>

Για τον υπολογισμό μιας αριθμητικής παράστασης το Maple πραγματοποιεί τις πράξεις σύμφωνα με τους γνωστούς κανόνες προτεραιότητας. Έτσι, υπολογίζονται πρώτα:

- Παραστάσεις μέσα στις παρενθέσεις, από την πιο μικρή παρένθεση στην πιο μεγάλη. Οι παρενθέσεις στο Maple χρησιμοποιούνται για να δηλώσουν μια ομάδα όρων.
- Δυνάμεις.
- Πολλαπλασιασμοί, διαιρέσεις.
- Προσθέσεις, αφαιρέσεις.
- Όλες οι παραπάνω πράξεις πραγματοποιούνται με προτεραιότητα από αριστερά προς τα δεξιά.

Έτσι, για τον υπολογισμό της παράστασης  $\frac{5^2}{5 \cdot 3}$  είναι λάθος να δώσουμε την εντολή:

`> 5^2/5*3;` [ENTER]

15

Πρέπει να δηλώσουμε την ομάδα των όρων που αποτελούν τον παρονομαστή χρησιμοποιώντας παρενθέσεις.

`> 5^2/(5*3);` [ENTER]

$\frac{5}{3}$

Επίσης, για τον υπολογισμό της δύναμης  $(-2)^4$ , αν παραλείψουμε την παρένθεση, το Maple θα υπολογίσει πρώτα τη δύναμη του 2.

`> -2^4;` [ENTER]

-16

Ο σωστός τρόπος εισαγωγής της παράστασης αυτής είναι:

`> (-2)^4;` [ENTER]

16

Ας δούμε τώρα τον υπολογισμό της δύναμης  $2^{ab}$ . Αν δε βάλουμε παρένθεση στο γινόμενο  $a*b$ , τότε το Maple θα εκτελέσει πρώτα τη δύναμη και στη συνέχεια τον πολλαπλασιασμό. Έτσι, η παράσταση που θα έχουμε υπολογίσει θα είναι η  $2^a \cdot b$

> 2<sup>a\*b</sup>; [ENTER]

$$2^a b$$

Αντίθετα, κάνοντας χρήση των παρενθέσεων για να δηλώσουμε τον εκθέτη θα έχουμε:

> 2<sup>(a\*b)</sup>; [ENTER]

$$2^{(a b)}$$

Αξιοπρόσεκτος είναι, επίσης, ο υπολογισμός δυνάμεων με αρνητικό εκθέτη. Για παράδειγμα, κατά τον υπολογισμό της δύναμης  $2^{-4}$  θα έχουμε μήνυμα λάθους αν δώσουμε την εντολή:

> 2<sup>-4</sup>; [ENTER]

Error, `-' unexpected

Αν επιθυμούμε, λοιπόν, να υψώσουμε σε αρνητική δύναμη, πρέπει να βάλουμε τον εκθέτη σε παρενθέσεις.

> 2<sup>(-4)</sup>; [ENTER]<sup>2</sup>

$$\frac{1}{16}$$

Στον παρακάτω πίνακα δίνονται οι βασικές πράξεις, συναρτήσεις και σταθερές που μπορούμε να χρησιμοποιήσουμε.

Πράξη ή Συνάρτηση	Συμβολισμός στο Maple	Μαθηματικός Συμβολισμός
Πρόσθεση	x+y	x+y
Αφαίρεση	x-y	x-y
Πολλαπλασιασμός	x*y	x*y
Διαίρεση	x/y	$\frac{x}{y}$
Δύναμη	x^y	$x^y$

<sup>2</sup> Είναι πλέον ξεκάθαρο ότι πρέπει να πατάμε [ENTER] για την εκτέλεση μιας εντολής. Γι' αυτό δε θα το αναφέρουμε ξανά.

Παραγοντικό	$n!$	$n!$
Απόλυτη Τιμή	$\text{abs}(x)$	$ x $
Τετραγωνική Ρίζα	$\text{sqrt}(x)$	$\sqrt{x}$
n-οστή ρίζα	$\text{surd}(x,n)$	$\sqrt[n]{x}$
Εκθετική συνάρτηση	$\text{exp}(x)$	$e^x$
Νεπέριος λογάριθμος	$\ln(x)$	$\ln(x)$
Λογάριθμος με βάση το 10	$\log_{10}(x)$	$\log_{10}(x)$
Λογάριθμος με βάση το $a$	$\log[a](x)$	$\log_a(x)$

*Πίνακας 2 Βασικών Πράξεων και Συναρτήσεων*

Πράξη ή Συνάρτηση	Συμβολισμός στο Maple	Μαθηματικός Συμβολισμός
Συνάρτηση ημιτόνου	$\sin(x)$	$\sin(x)$
Συνάρτηση συνημιτόνου	$\cos(x)$	$\cos(x)$
Συνάρτηση εφαπτομένης	$\tan(x)$	$\tan(x)$
Συνάρτηση συνεφαπτομένης	$\cot(x)$	$\cot(x)$
Συνάρτηση τέμνουσας	$\sec(x)$	$\sec(x)$
Συνάρτηση συντέμνουσας	$\csc(x)$	$\csc(x)$
Τόξο ημιτόνου	$\arcsin(x)$	$\arcsin(x)$
Τόξο συνημιτόνου	$\arccos(x)$	$\arccos(x)$
Τόξο εφαπτομένης	$\arctan(x)$	$\arctan(x)$
Η σταθερά $\pi$	Pi	$\pi$
Η σταθερά $e$	$\text{exp}(1)$	$e$
Φανταστική μονάδα	$i$ ή $\text{sqrt}(-1)$	$i$
Άπειρο	infinity	$\infty$

*Πίνακας 3. Βασικών Πράξεων και Συναρτήσεων*



## 2.2 Προσεγγιστικές και Ακριβείς Τιμές.

Αν ζητήσουμε το αποτέλεσμα της πρόσθεσης  $\frac{1}{3} + \frac{5}{7}$  το Maple θα εφαρμόσει τους γνωστούς κανόνες πρόσθεσης κλασμάτων χωρίς να μετατρέψει τους επιμέρους προσθετέους σε δεκαδικές προσεγγίσεις. Έτσι:

> 1/3+5/7;

$$\frac{22}{21}$$

Αυτό είναι μεγάλο πλεονέκτημα στους υπολογισμούς μας, γιατί δεν έχουμε ενδιάμεσα προσεγγιστικά σφάλματα. Αν την ίδια πράξη την κάναμε με μια απλή αριθμομηχανή, τότε θα προσθέταμε τις αντίστοιχες «δεκαδικές» τιμές τους με συνέπεια να μην έχουμε ακριβές αποτέλεσμα.

Ο παραπάνω τρόπος υπολογισμού είναι χαρακτηριστικό όλων των Συστημάτων Συμβολικής Άλγεβρας (C.A.System). Έτσι, καταλήγοντας μπορούμε να πούμε ότι το Maple χειρίζεται δύο ειδών αριθμητικές τιμές, τις Ακριβείς και τις Προσεγγιστικές αριθμητικές τιμές:

### Ακριβείς αριθμητικές τιμές:

- Ακέραιοι.
- Κλάσματα ακεραίων.
- Συμβολικές σταθερές π.
- sqrt(x), cos(x), sin(x), exp(x) κ.ά. ,όπου x ακριβής τιμή.

Αυτές τις ποσότητες το Maple τις χειρίζεται ως μαθηματικά αντικείμενα και εκτελεί πράξεις επιστρέφοντας ακριβή αποτελέσματα εκτός αν του ζητηθεί το αντίθετο.

Έτσι, αν του ζητήσουμε να κάνει την πράξη  $2 \cdot (3\pi - \pi)$ , όπως και στο παράδειγμα της πρόσθεσης κλασμάτων, θα μας επιστρέψει την ακριβή τιμή.

> 2\*(3\*Pi-Pi);

$$4\pi$$

Όμοια, αν ζητήσουμε την πρόσθεση μεταξύ άρρητων αριθμών, συγκεκριμένα ριζών, για παράδειγμα  $4\sqrt{2} + \sqrt{3} + \sqrt{2}$  θα έχουμε ως αποτέλεσμα ακριβή τιμή. Έτσι

> 4\*sqrt(2)+sqrt(3)+sqrt(2);

$$5\sqrt{2} + \sqrt{3}$$

Αν θέλουμε τη «δεκαδική» μορφή του αποτελέσματος, πρέπει να κάνουμε χρήση της εντολής `evalf`.

<b>Περιγραφή:</b>	<b>Επιστρέφει τη μορφή κινητής υποδιαστολής μιας έκφρασης.</b>	
<b>Γενική σύνταξη:</b>	<b><code>evalf(exp,n);</code></b>	
<b>Παράμετροι:</b>	<code>exp</code>	Η έκφραση που θέλουμε σε μορφή κινητής υποδιαστολής.
	<code>n</code>	Αριθμός δεκαδικών ψηφίων (ως προεπιλογή είναι το 10)

Ας δούμε τώρα τα παραπάνω παραδείγματα, ζητώντας το αποτέλεσμα σε μορφή κινητής υποδιαστολής.

```
> evalf(1/3+5/7);
```

1.047619048

Πράξεις με συμβολικές σταθερές:

```
> evalf(2*(3*Pi-Pi));
```

12.56637062

Αν θέλουμε τη σταθερά  $\pi$  σε προσεγγιστική τιμή με ακρίβεια 100 ψηφίων, τότε :

```
> evalf(Pi,100);
```

3.14159265358979323846264338327950288419716939937510582097  
4944592307816406286208998628034825342117068

Πράξεις με ρίζες  $4\sqrt{2} + \sqrt{3} + \sqrt{2}$  :

```
> evalf(4*sqrt(2)+sqrt(3)+sqrt(2));
```

8.803118618

Ειδικά για τους ακέραιους αριθμούς, το Maple διαθέτει ένα σύνολο εντολών, το οποίο φαίνεται στον παρακάτω πίνακα.

<i>Εντολές</i>	<i>Περιγραφή</i>
<i>ifactor(ακέραιος);</i>	Παραγοντοποίηση ενός ακεραίου.
<i>iquo(ακέραιος,n);</i>	Επιστρέφει το πηλίκο μιας ακέραιας διαίρεσης.
<i>irem(ακέραιος,n);</i>	Επιστρέφει το υπόλοιπο μιας ακέραιας διαίρεσης.
<i>iroot(ακέραιος,n);</i>	Επιστρέφει μια ακέραια προσέγγιση της n-οστής ρίζας του ακεραίου.
<i>isqrt(ακέραιος);</i>	Επιστρέφει μια ακέραια προσέγγιση της τετραγωνικής ρίζας του ακεραίου.
<i>ακέραιος mod n;</i>	Επιστρέφει το υπόλοιπο της διαίρεσης του ακεραίου με το n.
<i>igsd(a,b);</i>	Επιστρέφει το μέγιστο κοινό διαιρέτη δύο ακεραίων.
<i>isprime(ακέραιος);</i>	Ελέγχει εάν ένας ακέραιος είναι πρώτος.
<i>nextprime(ακέραιος);</i>	Επιστρέφει τον επόμενο πρώτο ενός ακεραίου αριθμού.
<i>ithprime(i);</i>	Υπολογίζει τον i-στο πρώτο αριθμό.

**Πίνακας 4.** Εντολές για τους ακέραιους

Έτσι, αν θέλουμε να παραγοντοποιήσουμε τον ακέραιο αριθμό 100, έχουμε:

```
> ifactor(100);
```

$$(2)^2 (5)^2$$

Για το πηλίκο και το υπόλοιπο της διαίρεσης του 50 με το 3 έχουμε τις εντολές:

```
> iquo(50,3);
```

16

```
> irem(50,3);
```

2

```
> 50 mod 3;
```

2

```
> iroot(4,3);
```

2



Ας κάνουμε την ίδια παρατήρηση με τον αριθμό  $\sqrt{2}$ .

Αν χρησιμοποιήσουμε την εντολή της τετραγωνικής ρίζας `sqrt`, τότε θα έχουμε την ακριβή τιμή :

```
> sqrt(2);
```

$$\sqrt{2}$$

Ενώ, αν ζητήσουμε την τετραγωνική ρίζα του 2. , ως αποτέλεσμα θα έχουμε την προσεγγιστική τιμή:

```
> sqrt(2.);
```

1.414213562

Όμοια και στα παραδείγματα:

Η ακριβής τιμή:

```
> sin(2);
```

sin(2)

Η προσεγγιστική τιμή:

```
> sin(2.);
```

0.9092974268

### 2.3 Αρίθμηση Αποτελεσμάτων –Συντομεύσεις

Το Maple διαθέτει τη δυνατότητα να αναφερόμαστε σε αποτελέσματα του φύλλου εργασίας που έχουν ήδη υπολογιστεί χρησιμοποιώντας κάποιες συντομεύσεις για αναφορά σε αυτά.

Με χρήση του συμβόλου % αναφερόμαστε στο αποτέλεσμα της τελευταίας εξόδου.

Έτσι, ας ζητήσουμε την πρόσθεση κάποιων κλασμάτων, για παράδειγμα:

```
> 3/5+5/9+7/12;
```

$\frac{313}{180}$

Για να έχουμε την προσεγγιστική τιμή του παραπάνω αποτελέσματος, μπορούμε να χρησιμοποιήσουμε την εντολή `evalf` που είδαμε παραπάνω με όρισμα το σύμβολο %.

```
> evalf(%);
```

1.738888889

Αν χρησιμοποιήσουμε το `%%`, αναφερόμαστε όχι στο τελευταίο αποτέλεσμα αλλά στο προτελευταίο.

```
> 2+1;
```

3

```
> 3+3;
```

6

Προσθέτουμε τα δύο αποτελέσματα:

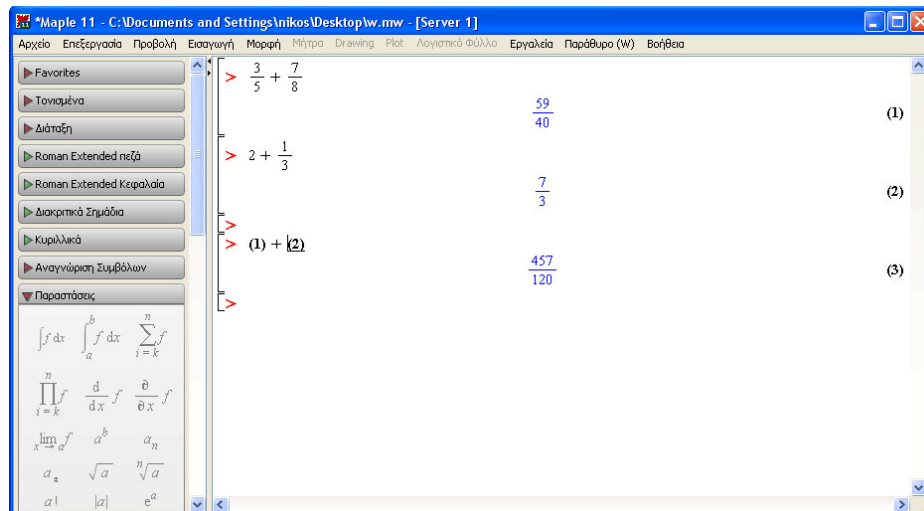
```
> %+%;
```

9

### Παρατήρηση 2.1

Αν χρησιμοποιούμε το Standard Maple περιβάλλον, κάθε αποτέλεσμα αριθμείται αυτόματα από το πρόγραμμα και μπορούμε να το καλέσουμε σε άλλο σημείο του φύλλου εργασίας μας, χρησιμοποιώντας τις ετικέτες (Labels). Για να χρησιμοποιήσουμε μια ετικέτα, επιλέγουμε από το μενού Insert την επιλογή Label (Ctrl+L) και εισάγουμε τον αριθμό ετικέτας στην οποία θέλουμε να αναφερθούμε.

Έτσι, αν θέλουμε να κάνουμε χρήση του αποτελέσματος της πράξης  $3/5+7/8$  και στην συνέχεια την πράξη  $2+1/3$ , μπορούμε να συνδυάσουμε τα δύο αυτά αποτελέσματα π.χ. να τα προσθέσουμε.



Εικόνα 22. Χρήση Ετικετών (Label)

## 2.4 Απλοποίηση Αριθμητικών Εκφράσεων

Με το Maple, μέχρι αυτό το σημείο, έχουμε δει πώς μπορούμε να υπολογίσουμε αριθμητικές παραστάσεις χρησιμοποιώντας τις βασικές πράξεις. Μια ακόμα δυνατότητα που μας δίνεται σε αριθμητικές εκφράσεις είναι να εφαρμόσουμε «γνωστές» ιδιότητες και σχέσεις ώστε να απλοποιήσουμε ή να αναπτύξουμε τις παραστάσεις αυτές.

Έτσι, εάν έχουμε την αριθμητική έκφραση:

```
> (2/3-sqrt(3))^4;
```

$$\left(\frac{2}{3}-\sqrt{3}\right)^4$$

με την εντολή `simplify` μπορούμε να τη γράψουμε σε απλούστερη μορφή.

```
> simplify(%);
```

$$\frac{(-2+3\sqrt{3})^4}{81}$$

Ενώ, με την εντολή `expand`, μπορούμε να την αναπτύξουμε.

```
> expand(%%);
```

$$\frac{1393}{81} - \frac{248\sqrt{3}}{27}$$

Τις εντολές αυτές θα τις εξετάσουμε περισσότερο αναλυτικά στο κεφάλαιο 4, όπου δε θα αφορούν μόνο αριθμητικές παραστάσεις αλλά γενικότερα αλγεβρικές εκφράσεις.

<b>Περιγραφή:</b>	<i>Απλοποιεί μια αριθμητική έκφραση.</i>
<b>Γενική σύνταξη:</b>	<i>simplify(έκφραση);</i>

<b>Περιγραφή:</b>	<i>Αναπτύσσει μια αριθμητική έκφραση.</i>
<b>Γενική σύνταξη:</b>	<i>expand(έκφραση);</i>

## 2.5 Μιγαδικοί Αριθμοί

Το Maple δίνει τη δυνατότητα να χειριστούμε μιγαδικούς αριθμούς και τις βασικές πράξεις τους. Έτσι, η πρόσθεση δύο μιγαδικών αριθμών συμβολίζεται με + και η αφαίρεση με -, ενώ ο πολλαπλασιασμός με \*.

Ένας μιγαδικός στο Maple ορίζεται με δύο τρόπους:

✚ Με τη μαθηματική μορφή του  $a + bi$

> 2+3\*I;

$2 + 3 I$

✚ Με την εντολή Complex (προσοχή: η εντολή αυτή γράφεται με κεφάλαιο το γράμμα C)

> Complex(2, 3);

$2 + 3 I$

Ας δούμε ένα παράδειγμα πρόσθεσης και πολλαπλασιασμού δύο μιγαδικών αριθμών.

> (3+2\*I) + (2+5\*I);

$5 + 7 I$

> (3+2\*I) \* (2+5\*I);

$-4 + 19 I$

Στον παρακάτω πίνακα παρουσιάζουμε βασικές εντολές που εφαρμόζονται στους μιγαδικούς αριθμούς.

Εντολή	Περιγραφή
<i>Re(αριθμός);</i>	Επιστρέφει το πραγματικό μέρος του αριθμού.
<i>Im(αριθμός);</i>	Επιστρέφει το φανταστικό μέρος του αριθμού.
<i>conjugate(αριθμός);</i>	Επιστρέφει το συζυγή ενός μιγαδικού αριθμού.
<i>abs(αριθμός);</i>	Επιστρέφει την απόλυτη τιμή ενός αριθμού.
<i>arguments( αριθμός);</i>	Επιστρέφει το πρώτο όρισμα ενός αριθμού.

Πίνακας 5. Βασικές εντολές Μιγαδικών εντολών.



Έτσι, αν θέλουμε μόνο το πραγματικό μέρος του μιγαδικού  $3+2i$ , έχουμε:

> Re (3+2\*I) ;

3

Για το φανταστικό μέρος του μιγαδικού έχουμε:

> Im (3+2\*I) ;

2

Για το συζυγή μιγαδικό αριθμό έχουμε:

> conjugate (3+2\*I) ;

$3 - 2I$

Για την απόλυτη τιμή του έχουμε:

> abs (3+2\*I) ;

$\sqrt{13}$

Ενώ, για ένα όρισμα του έχουμε:

> argument (3+2\*I) ;

$\arctan\left(\frac{2}{3}\right)$

Επίσης, μπορούμε να κάνουμε συμβολικούς υπολογισμούς με μιγαδικούς αριθμούς χρησιμοποιώντας την εντολή `evalc`.

Έτσι

> (a+b\*I) ^2;

$(a + bI)^2$

Εδώ η εφαρμογή της εντολής κάνει τις πράξεις.

> evalc (%);

$a^2 + 2Iab - b^2$

<b>Περιγραφή:</b>	<b>Πραγματοποιεί συμβολικούς υπολογισμούς σε εκφράσεις με μιγαδικούς.</b>
<b>Γενική σύνταξη:</b>	<b>evalc (έκφραση με μιγαδικό μέρος);</b>

Ένα ακόμα παράδειγμα:

```
> 1/(a+b*I);
```

$$\frac{1}{a + bI}$$

Εδώ η εφαρμογή της εντολής φέρνει τον παραπάνω μιγαδικό στην κανονική μορφή αναπαράστασης μιγαδικών  $a + bi$ .

```
> evalc(%);
```

$$\frac{a}{a^2 + b^2} - \frac{bI}{a^2 + b^2}$$

Ας δούμε ένα ακόμα παράδειγμα:

```
> abs(a+b*I);
```

$$|a + bI|$$

Εδώ η εφαρμογή της εντολής φέρνει τον παραπάνω μιγαδικό στην κανονική μορφή αναπαράστασης μιγαδικών  $a + bi$ .

```
> evalc(%);
```

$$\sqrt{a^2 + b^2}$$

# Κεφάλαιο 3

## Μεταβλητές

Σε ένα φύλλο εργασίας είναι χρήσιμο να δίνουμε ονόματα σε αριθμητικές ή συμβολικές παραστάσεις, ώστε να μπορούμε να τις καλούμε σε άλλο σημείο του φύλλου εργασίας. Έτσι είμαστε σε θέση να δημιουργούμε πιο σύνθετους υπολογισμούς.

Μια μεταβλητή ή σταθερά στο Maple μπορεί να έχει για όνομα οποιαδήποτε αλφαριθμητική σειρά χαρακτήρων, για παράδειγμα `a`, `total`, `exp1`, `sum1`, `sum2` κ.ά. Δεν είναι αποδεκτά ονόματα που αρχίζουν με αριθμό π.χ. `2nd`, `3total`, ή εκείνα που περιέχουν ειδικούς χαρακτήρες, όπως τους `@`, `&`, `*`, `_` κ.ά. τα ονόματα δηλαδή `a&b`, `total*sum`, `_sum`, ονόματα που είναι δεσμευμένα από το Maple ως εντολές π.χ. `plot`, `solve` και ονόματα που έχουν ιδιαίτερη σημασία στη γλώσσα του Maple. Υπάρχουν 46 τέτοια ονόματα που είναι δεσμευμένα. Έναν κατάλογο αυτών των ονομάτων μπορούμε να έχουμε, πληκτρολογώντας την εντολή:

```
> ?reserved;
```

and assuming break by catch description do done elif else  
 end error export fi finally for from global if implies  
 in intersect local minus mod module next not od option  
 options or proc quit read return save stop subset then  
 to try union use while xor

Επίσης, πρέπει να σημειωθεί ότι το Maple κάνει διαχωρισμό μεταξύ μικρών και κεφαλαίων γραμμάτων. Επομένως, άλλη είναι η μεταβλητή  $a$  και άλλη η μεταβλητή  $A$ .

### 3.1 Απόδοση Τιμής

Το Maple μας παρέχει τη δυνατότητα να αποδώσουμε σε μια μεταβλητή μια «τιμή». Η τιμή αυτή μπορεί να είναι ένας αριθμός, μια έκφραση, μια συνάρτηση, μια εξίσωση, μια γραφική παράσταση κ.ά. Αυτό μπορεί να γίνει με δύο τρόπους, είτε χρησιμοποιώντας τον τελεστή απόδοσης τιμής  $:=$ , είτε χρησιμοποιώντας την εντολή `assign`.

<i>Γενική σύνταξη:</i>	<i>Όνομα μεταβλητής:=αριθμητική τιμή ή έκφραση</i>
------------------------	--

Αποδίδουμε στη μεταβλητή  $a$  την τιμή 2:

```
> a:=2;
```

$a := 2$

Αν στο φύλλο εργασίας καλέσουμε το  $a$ , το πρόγραμμα θα μας επιστρέψει την τιμή 2.

```
> a;
```

2

Αν καλέσουμε τη μεταβλητή  $A$ , τότε δε θα μας επιστραφεί τίποτα, αφού, όπως έχουμε αναφέρει, γίνεται διαχωρισμός μεταξύ μικρών και κεφαλαίων χαρακτήρων.

```
> A;
```

$A$

Επίσης, μπορούμε να κάνουμε πράξεις με τις μεταβλητές.

```
> a+3;
```

5

**Παράδειγμα: 3-1**

Αν έχουμε ορίσει μια μεταβλητή, αυτή ισχύει σε όλο το φύλλο εργασίας και όσο δε σβήνουμε το περιεχόμενο της μνήμης του προγράμματος. Στην περίπτωση που θέλουμε να αλλάξουμε την τιμή της, απλώς πρέπει να την ορίσουμε ξανά δίνοντας τη νέα τιμή που θέλουμε. Έτσι:

```
> total:=200;
                                     total := 200

>total;
                                     200

>total:=sin(Pi);
                                     total := 0
```

□

Για την αποφυγή λαθών είναι σκόπιμο να μηδενίζονται όλες οι μεταβλητές όταν ξεκινάμε νέους υπολογισμούς. Αυτό γίνεται με την εντολή `restart`.

<i>Περιγραφή:</i>	<i>Καθαρίζει όλες τις μεταβλητές ενός φύλλου εργασίας.</i>
<i>Γενική σύνταξη:</i>	<i>restart</i>

Αν στο παραπάνω παράδειγμα εκτελέσουμε την εντολή `restart` και στη συνέχεια ζητήσουμε την τιμή `total`, τότε θα μας επιστραφεί το όνομα της μεταβλητής χωρίς να έχει καταχωρισμένη καμιά τιμή.

```
> restart;

> total;
                                     total
```

**Παράδειγμα: 3-2**

Αν προσπαθήσουμε να δώσουμε τιμή σε ήδη δεσμευμένα ονόματα, θα έχουμε μήνυμα λάθους. Έτσι η προσπάθεια να αποδώσουμε τιμές στις μεταβλητές `gamma`, `Beta`, `Chi`, `Pi` οδηγεί με μηνύματα σφάλματος.

```
> gamma := 0;
Error, attempting to assign to `gamma` which is protected

> Beta := 0;
Error, attempting to assign to `Beta` which is protected
```

```

> Chi := 0;
Error, attempting to assign to `Chi` which is protected
> Pi := 0;
Error, attempting to assign to `Pi` which is protected
> Psi := 0;
Error, attempting to assign to `Psi` which is protected
> plot:=4;
Error, attempting to assign to `plot` which is protected

```

Αν, ωστόσο, θέλουμε να χρησιμοποιήσουμε μια ήδη δεσμευμένη λέξη μπορούμε να χρησιμοποιήσουμε την εντολή `unprotect`.

```

> unprotect(plot);
> plot:=4;

```

*plot := 4*

□

Για να δούμε ποιες μεταβλητές έχουμε ορίσει σε ένα φύλλο εργασίας, χρησιμοποιούμε την εντολή `anames`.

<i>Περιγραφή:</i>	<i>Επιστρέφει τις μεταβλητές που έχουν ορισθεί σε ένα φύλλο εργασίας.</i>
<i>Γενική σύνταξη:</i>	<i>anames(παράμετρος);</i>

### Παράδειγμα: 3-3

Σε ένα καινούργιο φύλλο εργασίας ορίζουμε τις μεταβλητές  $x$  και  $y$ .

```
> x:=2.2;
```

*x := 2.2*

```
> y:=3;
```

*y := 3*

```
> anames();
```

*type/interfaceargs, x, y, interface*

□

**Παράδειγμα: 3-4**

Θα ορίσουμε την εξίσωση  $2x^2=3x+2$ , την οποία θα ονομάσουμε eq1.

```
> eq1 := 2*x^2=3*x+2;
```

$$eq1 := 2x^2 = 3x + 2$$

<b>Περιγραφή:</b>	<b>Επιστρέφει το αριστερό μέρος μιας έκφρασης (εξίσωση, ανίσωση ή σχέση).</b>
<b>Γενική σύνταξη:</b>	<b>rhs(έκφραση)</b>

Το δεξιό μέρος της εξίσωσης eq1 είναι:

```
> rhs (eq1) ;
```

$$3x + 2$$

<b>Περιγραφή:</b>	<b>Επιστρέφει το αριστερό μέρος μιας έκφρασης (εξίσωση, ανίσωση ή σχέση).</b>
<b>Γενική σύνταξη:</b>	<b>lhs(έκφραση)</b>

Το αριστερό μέρος της εξίσωσης eq1 είναι:

```
> lhs (eq1) ;
```

$$2x^2$$

□

Εκτός από τη χρήση του := για την απόδοση τιμής σε μια έκφραση, το Maple μας δίνει έναν ακόμα τρόπο με τη χρήση της εντολής assign.

<b>Περιγραφή:</b>	<b>Εκχωρεί τιμή σε μια μεταβλητή.</b>
<b>Γενική σύνταξη:</b>	<b>assign(όνομα, έκφραση)</b>

```
> assign (total, 300) ;
```

```
> total;
```

$$300$$

Αν θέλουμε να αποδεσμεύσουμε μια μεταβλητή, χρησιμοποιούμε την εντολή unassign.

<b>Περιγραφή:</b>	<b>Αποδεσμεύει μια μεταβλητή.</b>
<b>Γενική σύνταξη:</b>	<b>unassign(όνομα μεταβλητής)</b>

Στο παραπάνω παράδειγμα μπορούμε να αποδεσμεύσουμε το όνομα `total` ως εξής:

```
> unassign('total');
```

Αν τώρα ζητήσουμε τη μεταβλητή `total`, το Maple μας επιστρέφει το όνομα της μεταβλητής χωρίς τιμή.

```
> total;
```

*total*

### 3.2 Τύποι Μεταβλητών

Οι μεταβλητές που μπορούμε να χρησιμοποιήσουμε στο Maple είναι διαφόρων ειδών. Στον παρακάτω πίνακα περιγράφουμε κάποιες από αυτές:

ΤΥΠΟΣ	ΠΕΡΙΓΡΑΦΗ
<code>integer</code>	Ακέραιοι, των οποίων το εύρος καθορίζεται κάθε φορά από το σύστημα του υπολογιστή.
<code>fraction</code>	Κλάσματα
<code>float</code>	Αριθμοί κινητής υποδιαστολής
<code>complex</code>	Μιγαδικοί αριθμοί
<code>string</code>	Αλφαριθμητικοί
<code>nonreal</code>	Μη πραγματική
<code>boolean</code>	Λογική (True ή False)
<code>rational</code>	Ρητός
<code>sfloat</code> ,	SFloat(M, E) Αριθμοί κινητής υποδιαστολής (M η μαντίσα και E ο εκθέτης).
<code>{...}</code>	Σύνολα
<code>[...]</code>	Λίστα
<code>(...)</code>	Ακολουθία
<code>indexed</code>	Δείκτες

Με την εντολή `whattype` μπορούμε να βρούμε το είδος της μεταβλητής που χρησιμοποιούμε.



<i>Περιγραφή:</i>	<i>Επιστρέφει τον τύπο της μεταβλητής.</i>
<i>Γενική σύνταξη:</i>	<i>whattype(μεταβλητή)</i>

Ακολουθως παρατίθενται ορισμένα παραδείγματα των βασικών τύπων μεταβλητών για την εντολή `whattype`.

Ορίζουμε το `x`

```
> x:=2;
```

*x := 2*

Εξετάζουμε ποιος είναι ο τύπος της μεταβλητής `x`

```
> whattype (x) ;
```

*integer*

```
> whattype (2/3) ;
```

*fraction*

```
> x:=22.3;
```

*x := 22.3*

```
> whattype (x) ;
```

*float*

```
> whattype (2-3*I) ;
```

*complex*

```
>whattype ("Κείμενο") ;
```

*string*

```
> x:=exp(1) ;
```

*x := e*

```
> whattype (x) ;
```

*function*

```
>x:=pi;
```

*x := π*

```
> whattype (x) ;
```

*symbol*

```
> whattype ({1, 2, 3});
```

*set*

```
> whattype ([1, 2, 3]);
```

*list*

```
> whattype ((1, 2, 3));
```

*exprseq*

```
> whattype (x[1]);
```

*indexed*

Η εντολή `type` ελέγχει τον τύπο μιας έκφρασης και επιστρέφει τιμή αληθή ή ψευδή.

<i>Περιγραφή:</i>	<i>Ελέγχει τον τύπο μιας έκφρασης.</i>	
<i>Γενική σύνταξη:</i>	<i>type (έκφραση, τύπος έκφρασης)</i>	
<i>Παράμετροι:</i>	τύπος έκφρασης	integer, rational, fraction, float, sfloat, numeric, extended_ ,numeric, nonreal

### Παράδειγμα: 3-5

Θα ορίσουμε η μεταβλητή `x` να έχει την τιμή 2.3 και θα εξετάσουμε εάν η μεταβλητή `x` είναι ακέραιος ή αριθμός κινητής υποδιαστολής.

```
> x := 2.3;
```

*x := 2.3*

Εξετάζουμε αν το `x` είναι ακέραιος:

```
> type (x, integer);
```

*false*

Εξετάζοντας αν το `x` είναι αριθμός κινητής υποδιαστολής, θα έχουμε τιμή αληθή.

```
> type (x, float);
```

*true*

Ακολουθούν κάποια ακόμα παραδείγματα για τη χρήση της εντολής `type`.

```
> type (2, integer);
```

*true*

```

>type(1/2, integer);
false

>type(exp(1), integer);
false

>type(exp(1), float);
false

>type(1.5, float);
true

>type(2-5*I, nonreal);
true

```

□

Μπορούμε να μετατρέψουμε μια μεταβλητή από έναν τύπο σε άλλον χρησιμοποιώντας την εντολή `convert`.

<i>Περιγραφή:</i>	<i>Μετατρέπει μια μεταβλητή από έναν τύπο σε άλλον.</i>	
<i>Γενική σύνταξη:</i>	<i>convert(expr, form)</i>	
<i>Παράμετροι:</i>	expr	Η έκφραση που μετατρέπουμε.
	form	fraction, Vector, set, list, binary, octal, hex.

Οι μετατροπές που μπορούν να γίνουν με τη χρήση της εντολής `convert` είναι πολλές. Όλες τις δυνατότητες που δίνει αυτή η εντολή μπορούμε να τις δούμε πληκτρολογώντας `?convert`.

### Παράδειγμα: 3-6

Αφού αποδοθεί η τιμή 0.2 στο `x`, θα μετατρέψουμε το `x` σε κλάσμα.

Αποδίδουμε στο `x` την τιμή 0.2:

```

> x:=0.2;
x := 0.2

```

Ρωτάμε τι τύπου είναι το `x` και το Maple μας επιστρέφει την απάντηση `float`, το `x` είναι δηλαδή αριθμός κινητής υποδιαστολής.

```

> whattype(x);
float

```

Μετατρέπουμε το  $x$  σε κλάσμα:

```
> x:=convert(x, fraction);
```

$$x := \frac{1}{5}$$

Αν ρωτήσουμε τώρα τον τύπο της μεταβλητής  $x$ , το πρόγραμμα θα μας απαντήσει `fraction`.

```
> whattype(x);
```

*fraction*

□

### Παράδειγμα: 3-7

Στο παράδειγμα αυτό θα ορίσουμε μια λίστα αριθμών, στη συνέχεια θα τη μετατρέψουμε σε διάνυσμα και τέλος σε σύνολο αριθμών.

```
> a:=[1,2,3];
```

$$a := [1, 2, 3]$$

```
> whattype(a);
```

*list*

```
> a:=convert(a, Vector);
```

$$a := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

```
> whattype(a);
```

*Vector*<sub>column</sub>

```
> a:=convert(a, set);
```

$$a := \{1, 2, 3\}$$

```
> whattype(a);
```

*set*

□

**Παράδειγμα: 3-8**

Η εντολή `convert` μπορεί να χρησιμοποιηθεί γενικότερα για μετατροπές. Τη χρησιμοποιούμε για παράδειγμα, αν θέλουμε να μετατρέψουμε τον αριθμό 0.2 σε δυαδική μορφή :

```
> convert(0.2, binary);
```

0.001100110011

ή τον αριθμό 170 από τη δεκαδική στην οκταδική μορφή:

```
> convert(170, decimal, octal);
```

120

ή τον αριθμό 100 σε δεκαεξαδική μορφή:

```
> convert(100, hex);
```

64

□

**3.3 Υποθέσεις σε Μεταβλητές**

Το Maple δίνει τη δυνατότητα να θέσουμε υποθέσεις στις μεταβλητές μας με την εντολή `assume`.

<i>Περιγραφή:</i>	<i>Θέτει σε μια μεταβλητή μια υπόθεση.</i>
<i>Γενική σύνταξη:</i>	<i>assume(μεταβλητή, υπόθεση);</i>

Υποθέτουμε ότι η μεταβλητή  $x$  είναι θετικός αριθμός.

```
> assume(x>0);
```

```
> sign(x);
```

1

Αν ζητήσουμε τη  $x$ , το Maple θα επιστρέψει τη μεταβλητή με το σύμβολο  $\sim$ , το οποίο σημαίνει ότι έχουμε υποθέσεις για τη μεταβλητή  $x$ .

```
> x;
```

$x\sim$

Με την εντολή `about` μπορούμε να δούμε τις υποθέσεις για μια μεταβλητή.

<i>Περιγραφή:</i>	<i>Επιστρέφει τις υποθέσεις για μια μεταβλητή.</i>
<i>Γενική σύνταξη:</i>	<i><code>about(μεταβλητή);</code></i>

Για τη μεταβλητή  $x$ :

```
> about(x);
```

Originally  $x$ , renamed  $x\sim$ :

is assumed to be: `RealRange(Open(0),infinity)`

### Παράδειγμα: 3-9

Υποθέτουμε ότι το  $a$  είναι ακέραιος.

```
> assume(a, integer);
```

```
> about(a);
```

Originally  $a$ , renamed  $a\sim$ :

is assumed to be: `integer`

Υποθέτουμε ότι το  $x > -1$ .

```
> assume(x > -1);
```

```
> about(x);
```

Originally  $x$ , renamed  $x\sim$ :

is assumed to be: `RealRange(Open(-1),infinity)`

□

### Παράδειγμα: 3-10

Θα δούμε τώρα πώς διαφοροποιείται ο υπολογισμός της  $\sqrt{x^2}$  ανάλογα με τις υποθέσεις που έχουμε για το  $x$ .

Έτσι, εάν ζητήσουμε την  $\sqrt{x^2}$ , το Maple δε θα μπορέσει να κάνει κανέναν υπολογισμό.

```
> sqrt(x^2);
```

$\sqrt{x^2}$

Αν τώρα υποθέσουμε ότι το  $x > 0$ :

```
> assume(x, positive);
```

```
> about( x );
```

Originally  $x$ , renamed  $x\sim$ :

is assumed to be: `RealRange(Open(0),infinity)`

το Maple θα επιστρέψει το  $x$ .

```
> sqrt(x^2);
```

$x\sim$

Ισοδύναμα για την παραπάνω υπόθεση μπορούμε να έχουμε την υπόθεση:

```
> assume( x, RealRange(Open(0),infinity) );
```

```
> about( x );
```

Originally  $x$ , renamed  $x\sim$ : is assumed to be:  
`RealRange(Open(0),infinity)`

□





# Κεφάλαιο 4

## Αλγεβρικοί Υπολογισμοί

Το Maple, όπως και κάθε σύστημα συμβολικής άλγεβρας, γνωρίζει τους περισσότερους μετασχηματισμούς και κανόνες άλγεβρας. Σε αυτό το κεφάλαιο θα δούμε τις εντολές που μας επιτρέπουν να χειριστούμε μαθηματικές αλγεβρικές παραστάσεις.

Θα ξεκινήσουμε παρουσιάζοντας τη δυνατότητα να εκτελούμε πράξεις μεταξύ πολυωνύμων και στη συνέχεια θα δούμε πώς μπορούμε να παραγοντοποιήσουμε, να απλοποιήσουμε και γενικά να κάνουμε «πράξεις» με εκφράσεις πολυωνυμικές και μη.

Οι βασικές εντολές που θα αναλυθούν είναι: `+`, `-`, `*`, `quo`, `rem`, `subs`, `expand`, `factor`, `simplify` και η εντολή `collect`.

## 4.1 Πράξεις με Πολυωνυμικές Εκφράσεις

Μια σημαντική κατηγορία εκφράσεων, στις οποίες μπορούμε να εφαρμόσουμε αλγεβρικούς υπολογισμούς, είναι τα πολυώνυμα. Για την πρόσθεση, την αφαίρεση και τον πολλαπλασιασμό πολυωνύμων μπορούμε να χρησιμοποιήσουμε τους γνωστούς τελεστές  $+$ ,  $-$ ,  $*$  αντίστοιχα. Στον παρακάτω πίνακα περιγράφονται οι σημαντικότερες εντολές που αφορούν τα πολυώνυμα.

<i>Εντολή</i>	<i>Περιγραφή</i>
<i>sort(πολυώνυμο);</i>	Ταξινομεί ένα πολυώνυμο κατά φθίνουσα σειρά τάξεως.
<i>divide(πολυώνυμο1, πολυώνυμο2, μεταβλητή);</i>	Εξετάζει αν δυο πολυώνυμα διαιρούνται.
<i>rem(πολυώνυμο1, πολυώνυμο2, μεταβλητή);</i>	Επιστρέφει το υπόλοιπο της διαίρεσης δύο πολυωνύμων.
<i>quo(πολυώνυμο1, πολυώνυμο2, μεταβλητή);</i>	Επιστρέφει το πηλίκο της διαίρεσης δύο πολυωνύμων.

*Πίνακας 6. Πράξεις με Πολυώνυμα*

<i>Εντολή</i>	<i>Περιγραφή</i>
<i>coeff(πολυώνυμο, όρος);</i>	Εξάγει το συντελεστή του όρου.
<i>coeffs(πολυώνυμο);</i>	Εξάγει τους συντελεστές των όρων του πολυωνύμου.
<i>lcoeffs(πολυώνυμο);</i>	Εξάγει το συντελεστή του σταθερού όρου.
<i>tcoeffs(πολυώνυμο);</i>	Εξάγει το συντελεστή του μεγιστοβάθμιου όρου.
<i>degree(πολυώνυμο);</i>	Επιστρέφει το βαθμό του πολυωνύμου.

*Πίνακας 7. Εντολές για τα πολυώνυμα.*

**Παράδειγμα: 4-1**

Να ορισθούν τα πολυώνυμα  $p_1 = x^3 - 3x + 2$ ,  $p_2 = x^2 - 5x + 6$  και να βρεθούν τα  $p_1 + p_2$ ,  $p_1 - p_2$ ,  $p_1 * p_2$ , το υπόλοιπο και το ηλίκο της διαίρεσης  $p_1$  και  $p_2$ .

**Λύση:**

Ορίζουμε τα πολυώνυμα  $p_1$  και  $p_2$ .

> p1 := x^3 - 3\*x + 2;

$$p1 := x^3 - 3x + 2$$

> p2 := x^2 + 5\*x + 6;

$$p2 := x^2 + 5x + 6$$

Τα προσθέτουμε:

> p1+p2;

$$x^3 + x^2 + 2x + 8$$

Αν θέλουμε να διατάξουμε το πολυώνυμο σε φθίνουσα σειρά δυνάμεων του  $x$ , χρησιμοποιούμε την εντολή `sort`.

> sort(%);

$$x^3 + x^2 + 2x + 8$$

Η διαφορά των δύο πολυωνύμων είναι:

> p1-p2;

$$x^3 - 8x - 4 - x^2$$

Παρουσιάζουμε πάλι το πολυώνυμο κατά φθίνουσα σειρά δυνάμεων.

> sort(%);

$$x^3 - x^2 - 8x - 4$$

Το γινόμενο τους είναι:

> p1\*p2;

$$(x^3 - 3x + 2)(x^2 + 5x + 6)$$

Βλέπουμε ότι το πρόγραμμα επιστρέφει ένα νέο πολυώνυμο χωρίς ωστόσο να εφαρμόζει την προσημαστική ιδιότητα μεταξύ των δύο πολυωνύμων. Αυτό είναι

δυνατό με την εντολή `expand` ή την εντολή `collect` που θα δούμε παρακάτω. Ωστόσο, είναι δυνατό να βρούμε το βαθμό αυτού του πολυωνύμου με την εντολή `degree`.

```
> degree (%);
```

5

Το υπόλοιπο της διαίρεσης των  $p_1$  και  $p_2$  είναι:

```
> rem(p1, p2, x);
```

$32 + 16x$

Το πηλίκο της διαίρεσης των  $p_1$  και  $p_2$  είναι:

```
> quo(p1, p2, x);
```

$x - 5$

□

## 4.2 Παραγοντοποίηση

Για την παραγοντοποίηση μιας πολυωνυμικής έκφρασης, δηλαδή τη μετατροπή της σε γινόμενο παραγόντων, μπορούμε να χρησιμοποιήσουμε την εντολή `factor`. Επίσης, ένα είδος παραγοντοποίησης μπορεί να θεωρηθεί και η χρήση της εντολής `combine`, που αφορά μη πολυωνυμικές εκφράσεις και θα τη δούμε παρακάτω.

<i>Περιγραφή:</i>	<i>Παραγοντοποιεί μια έκφραση.</i>	
<i>Γενική σύνταξη:</i>	<i>factor(έκφραση,option)</i>	
<i>Παράμετροι:</i>	option	<ul style="list-style-type: none"> <li>▪ <b>real</b> ή <b>complex</b>, αν η παραγοντοποίηση θέλουμε να γίνει στους πραγματικούς ή στους μιγαδικούς αριθμούς αντίστοιχα.</li> <li>▪ ο παράγοντας (ρίζα) που θέλω να παρουσιάζεται στο γινόμενο.</li> </ul>

### Παράδειγμα: 4-2

Να παραγοντοποιηθούν οι εκφράσεις  $a^2 + 2ab + b^2$ ,  $6x^2 + 18x - 24$ ,  $x^3 + 7x^2 - 21x - 27$ .

**Λύση:**

```
> factor(a^2+2*a*b+b^2);
```

$$(a + b)^2$$

```
> factor(6*x^2+18*x-24);
```

$$6(x + 4)(x - 1)$$

```
> factor(x^3+7*x^2-21*x-27);
```

$$(x + 9)(x - 3)(x + 1)$$

□

**Παράδειγμα: 4-3**

Να παραγοντοποιηθεί η παράσταση  $x^2 + 4$ .

**Λύση:**

Προσπαθώντας να παραγοντοποιήσουμε την παραπάνω παράσταση βλέπουμε ότι το Maple δεν επιστρέφει κάποιο αποτέλεσμα αλλά την ίδια την έκφραση.

```
> factor(x^2+4);
```

$$x^2 + 4$$

Αυτό συμβαίνει γιατί δεν μπορεί να γίνει η παραγοντοποίηση στους πραγματικούς αριθμούς. Έτσι, αν χρησιμοποιήσουμε την παράμετρο `complex`, τότε έχουμε το αποτέλεσμα:

```
> factor(x^2+4, complex);
```

$$(x + 2.000000000 I)(x - 2.000000000 I)$$

**Παράδειγμα: 4-4**

Να παραγοντοποιηθεί η παράσταση  $x^4 - 2$ .

**Λύση:**

```
> factor(x^4-2);
```

$$x^4 - 2$$

Ζητώντας την παραγοντοποίηση του παραπάνω πολυωνύμου δεν έχουμε αποτέλεσμα.

Εάν θέλουμε να εμφανίζεται ένας όρος που να περιέχει τη ρίζα  $\sqrt{2}$ , τότε:

```
> factor(x^4-2, sqrt(2));
```

$$(x^2 + \sqrt{2})(x^2 - \sqrt{2})$$

□

### Παράδειγμα: 4-5

Να παραγοντοποιηθεί η παράσταση  $x^4 - 8x^2 + 15$ .

**Λύση:**

```
> factor(x^4-8*x^2+15);
```

$$(x^2 - 5)(x^2 - 3)$$

Αν θέλουμε περαιτέρω παραγοντοποίηση, μπορούμε να ζητήσουμε να γίνει παραγοντοποίηση χρησιμοποιώντας όρους που περιέχουν τη,  $\sqrt{3}$  (είναι ρίζα του όρου  $(x^2 - 3)$ ).

```
> factor(x^4-8*x^2+15, sqrt(3));
```

$$(x^2 - 5)(x - \sqrt{3})(x + \sqrt{3})$$

ή χρησιμοποιώντας όρους που περιέχουν την  $\sqrt{5}$ ,

```
> factor(x^4-8*x^2+15, sqrt(5));
```

$$(x^2 - 3)(x + \sqrt{5})(x - \sqrt{5})$$

ή και τις δύο συγχρόνως.

```
> factor(x^4-8*x^2+15, {sqrt(3), sqrt(5)});
```

$$(x - \sqrt{3})(x + \sqrt{3})(x + \sqrt{5})(x - \sqrt{5})$$

□

### Η Εντολή `Combine`

Η εντολή `combine` είναι ένα είδος «παραγοντοποίησης» για μη πολυωνυμικές εκφράσεις, όπως εκφράσεις που περιέχουν τριγωνομετρικούς ή εκθετικούς όρους. Συγκεκριμένα, η εντολή `combine` εφαρμόζει κάποιους συγκεκριμένους μετασχηματισμούς. Ορισμένοι από αυτούς φαίνονται παρακάτω:

$$\left. \begin{array}{l} \text{Μετασχηματισμοί δυνάμεων} \\ x^y \cdot x^z \rightarrow x^{y+z} \\ x^{y^z} \rightarrow x^{y \cdot z} \\ x^n \cdot y^n \rightarrow (x \cdot y)^n \end{array} \right\} \textit{power}$$

$$\text{Radical μετασχηματισμοί} \quad \left. (x^n \cdot y)^{\frac{m}{n}} \rightarrow (x^m y^n)^{\frac{1}{d}} \right\} \textit{radical}$$

Τριγωνομετρικοί

$$\left. \begin{array}{l} \sin x \cdot \sin y \rightarrow \frac{1}{2} \cos(x-y) - \frac{1}{2} \cos(x+y) \\ \sin x \cdot \cos y \rightarrow \frac{1}{2} \sin(x-y) + \frac{1}{2} \sin(x+y) \\ \cos x \cdot \cos y \rightarrow \frac{1}{2} \cos(x-y) + \frac{1}{2} \cos(x+y) \end{array} \right\} \textit{trig}$$

$$\left. \begin{array}{l} \text{Εκθετικοί} \\ e^x \cdot e^y \rightarrow e^{x+y} \\ (e^x)^y \rightarrow e^{x \cdot y} \\ e^{x+n \cdot \ln y} \rightarrow y^n e^x \end{array} \right\} \textit{exp}$$

$$\left. \begin{array}{l} \text{Λογαριθμικοί} \\ y \cdot \ln x \rightarrow \ln(x^y) \\ \ln x + \ln y \rightarrow \ln(x \cdot y) \end{array} \right\} \textit{ln}$$

<b>Περιγραφή:</b>	Συνενώνει αν είναι δυνατόν εκφράσεις έτσι ώστε να αποτελούνται από έναν όρο.
<b>Γενική σύνταξη:</b>	<i>combine(έκφραση,παράμετρος)</i>
<b>Παράμετροι:</b>	power, radical, exp, ln, trig, arctan,abs, arctan conjugate, errors, icombine, piecewise, polylog, product ,Psi,range, signum, units.

**Παράδειγμα: 4-6**

Ας δούμε κάποιους από τους παραπάνω μετασχηματισμούς σε συμβολική μορφή.

```
> combine( ln(x)+ln(y), ln, symbolic );
```

$$\ln(xy)$$

```
> combine( ln(x)+ln(y), symbolic );
```

$$\ln(xy)$$

```
> combine( exp(x)^y );
```

$$(e^x)^y$$

```
> combine( exp(x)^y, symbolic );
```

$$e^{(xy)}$$

```
> combine( (x^y)^z );
```

$$(x^y)^z$$

```
> combine( (x^y)^z, symbolic );
```

$$x^{(yz)}$$

□

**Παράδειγμα: 4-7**

Να γραφούν σε απλούστερη μορφή οι εκφράσεις  $\sin x \cdot \cos y + \cos x \cdot \sin y$ ,

$$\ln|x| - \ln|x+1|.$$

**Λύση:**

Ορίζουμε την έκφραση expr.

```
> expr:=sin(x)*cos(y)+cos(x)*sin(y);
```

$$expr := \sin(x) \cos(y) + \cos(x) \sin(y)$$

Αν ζητήσουμε την παραγοντοποίηση της έκφρασης, με την εντολή factor δε θα έχουμε κανένα αποτέλεσμα ,

```
> factor(expr);
```

$$\sin(x) \cos(y) + \cos(x) \sin(y)$$



ενώ χρησιμοποιώντας την εντολή `combine` έχουμε:

```
> combine (expr) ;
```

$$\sin(x + y)$$

□

### Παράδειγμα: 4-8

Να εφαρμοσθεί η εντολή `combine` στην έκφραση

$$e^x e^y + \sin x \cdot \sin y + \sqrt{2} \sqrt{x+1}.$$

#### Λύση:

Καλούμε την εντολή `restart` για να σβήσουμε το περιεχόμενο των μεταβλητών από το προηγούμενο παράδειγμα.

```
> restart ;
```

Ορίζουμε την έκφραση `expr`.

```
> expr := exp(x) * exp(y) + sin(x) * sin(y) + sqrt(2) *
sqrt(x+1) ;
```

$$expr := e^x e^y + \sin(x) \sin(y) + \sqrt{2} \sqrt{x+1}$$

Αρχικά, θα εφαρμόσουμε μόνο τριγωνομετρικούς μετασχηματισμούς στην παραπάνω έκφραση:

```
> combine (expr, trig) ;
```

$$e^x e^y + \frac{1}{2} \cos(x-y) - \frac{1}{2} \cos(x+y) + \sqrt{2} \sqrt{x+1}$$

Μπορούμε, επίσης, να εφαρμόσουμε μόνο `radical` μετασχηματισμούς:

```
> combine (expr, radical) ;
```

$$e^x e^y + \sin(x) \sin(y) + \sqrt{2x+2}$$

Μπορούμε, όμως, να εφαρμόσουμε τριγωνομετρικούς μαζί με `radical` μετασχηματισμούς:

```
> combine (expr, [exp, radical]) ;
```

$$e^{(x+y)} + \sin(x) \sin(y) + \sqrt{2x+2}$$

Έχουμε, τέλος, τη δυνατότητα να εφαρμόσουμε όλους τους δυνατούς μετασχηματισμούς:

> combine(expr);

$$e^{(x+y)} + \frac{1}{2} \cos(x-y) - \frac{1}{2} \cos(x+y) + \sqrt{2x+2}$$

□

### 4.3 Ανάπτυγμα

Η εντολή `expand` αναπτύσσει μια έκφραση, είναι ουσιαστικά η αντίστροφη διαδικασία της εντολής `factor`.

Περιγραφή:	Αναπτύσσει μια έκφραση.
Γενική σύνταξη:	<code>expand (έκφραση)</code>

Η εντολή `expand` μπορεί να εφαρμοστεί σε πολλές μαθηματικές εκφράσεις εκτός των πολυωνυμικών, όπως οι τριγωνομετρικές και αυτές που περιέχουν λογαρίθμους.

Αναπτύσσει μια έκφραση εφαρμόζοντας κανόνες. Κάποιοι από αυτούς φαίνονται παρακάτω:

$\left. \begin{array}{l} x^{y+z} \rightarrow x^y \cdot x^z \\ x^{y \cdot z} \rightarrow x^{y^z} \\ (x \cdot y)^n \rightarrow x^n \cdot y^n \end{array} \right\} \textit{power}$	$\left. \begin{array}{l} \cos 2x \rightarrow 2 \cos^2 x - 1 \\ \sin 2x \rightarrow 2 \sin x \cdot \cos x \\ \cos(x+y) \rightarrow \cos x \cdot \cos y - \sin x \cdot \sin y \\ \sin(x+y) \rightarrow \sin x \cdot \cos y + \cos x \cdot \sin y \\ \tan(x+y) = \frac{\tan x + \tan y}{1 - \tan x \cdot \tan y} \end{array} \right\} \textit{trig}$
$\left. \begin{array}{l} e^{x+y} \rightarrow e^x \cdot e^y \\ e^{x \cdot y} \rightarrow (e^x)^y \\ e^{x+n \cdot \ln y} \rightarrow y^n e^x \end{array} \right\} \textit{exp}$	$\left. \begin{array}{l} \ln x^y \rightarrow y \cdot \ln x \\ \ln(x \cdot y) \rightarrow \ln x + \ln y \end{array} \right\} \textit{ln}$

Σε όλους τους παραπάνω κανόνες ισχύουν βέβαια οι περιορισμοί που απαιτούνται ώστε αυτοί να ορίζονται.

### Παράδειγμα: 4-9

Να αναπτυχθούν οι εκφράσεις  $(a+b)^2$ ,  $\frac{x+1}{x+2} \sin(x+y)$ ,  $\tan(2x)$ ,  $\tan(x+y)$ ,  $x^2(x^3+2x^2+7x+12)$ ,  $(u+v)^4$ ,  $(x+1)(x+2)(x-9)$  και  $\sin(7x)$ .

#### Λύση:

> expand((a+b)^2);

$$a^2 + 2 a b + b^2$$

> expand((x+1)/(x+2));

$$\frac{x}{x+2} + \frac{1}{x+2}$$

> expand(sin(x+y));

$$\sin(x) \cos(y) + \cos(x) \sin(y)$$

> expand(tan(2\*x));

$$\frac{2 \tan(x)}{1 - \tan(x)^2}$$

> expand(tan(x+y));

$$\frac{\tan(x) + \tan(y)}{1 - \tan(x) \tan(y)}$$

> expand(x^2\*(x^3+2\*x^2+7\*x+12));

$$x^5 + 2 x^4 + 7 x^3 + 12 x^2$$

> expand((u+v)^4);

$$u^4 + 4 u^3 v + 6 u^2 v^2 + 4 u v^3 + v^4$$

> expand((x+1)\*(x+2)\*(x-9));

$$x^3 - 6x^2 - 25x - 18$$

> expand(sin(7\*x));

$$64 \sin(x) \cos(x)^6 - 80 \sin(x) \cos(x)^4 + 24 \sin(x) \cos(x)^2 - \sin(x)$$

□

#### Παράδειγμα: 4-10

Να αναπτυχθεί έκφραση  $\ln(x \cdot y)$ .

#### Λύση:

Εφαρμόζοντας την εντολή `expand` δεν παίρνουμε αποτέλεσμα, γιατί δεν ισχύουν οι απαραίτητες υποθέσεις για τα  $x, y$ .

> expand(ln(x\*y));

$$\ln(xy)$$

Αν τώρα υποθέσουμε ότι  $x > 0$  και  $y > 0$ , εφαρμόζεται το ανάπτυγμα.

> assume(x>0 and y>0);

> expand(ln(x\*y));

$$\ln(x) + \ln(y)$$

Μια άλλη υλοποίηση των παραπάνω εντολών θα μπορούσε να είναι η ακόλουθη:

> expand(ln(x\*y) ) assuming positive;

$$\ln(x) + \ln(y)$$

□

#### Παράδειγμα: 4-11

Να αναπτυχθεί έκφραση  $\ln\left(\frac{|x|}{|x+1|}\right)$ .

#### Λύση:

Ορίζουμε την έκφραση `expr`.

> expr:=ln(abs(x)/abs(x+1));

$$expr := \ln\left(\frac{|x|}{|x+1|}\right)$$

Εφαρμόζουμε στην έκφραση  $\text{expr}$  την εντολή  $\text{expr}$ .

`> expand (expr) ;`

$$\ln(|x|) - \ln(|x + 1|)$$

□

### Παράδειγμα: 4-12

Να αναπτυχθεί η έκφραση  $\sin(x + y) + e^{x+y} + e^{z+w} + x^{n+m}$ .

#### Λύση:

Ορίζουμε την έκφραση  $\text{expr}$ .

`> expr := sin(x+y) + exp(x+y) + exp(z+w) + x^(n+m) ; ;`

$$\text{expr} := \sin(x + y) + e^{(x+y)} + e^{(z+w)} + x^{(n+m)}$$

Καλώντας την εντολή  $\text{expand}$  γίνονται όλα τα δυνατά αναπτύγματα.

`> expand (expr) ;`

$$\sin(x) \cos(y) + \cos(x) \sin(y) + e^x e^y + e^z e^w + x^n x^m$$

Αν προσθέσουμε μια ακόμα παράμετρο, όπως την παράμετρο  $\text{exp}$ , στην εντολή, τότε το πρόγραμμα αναπτύσσει τα πάντα εκτός από τους όρους που περιέχουν εκθετικά.

`> expand (expr, exp) ;`

$$\sin(x) \cos(y) + \cos(x) \sin(y) + e^{(x+y)} + e^{(z+w)} + x^n x^m$$

Αν θελήσουμε με την εντολή  $\text{expand}$  το πρόγραμμα να αναπτύξει τα πάντα εκτός από τα εκθετικά της μορφής  $e^{x+y}$ , έχουμε:

`> expand (expr, exp (x+y) ) ;`

$$\sin(x) \cos(y) + \cos(x) \sin(y) + e^{(x+y)} + e^z e^w + x^n x^m$$

Αν θελήσουμε με την εντολή  $\text{expand}$  το πρόγραμμα να αναπτύσσει τα πάντα εκτός από τα εκθετικά της μορφής  $e^{z+w}$  και το ημίτονο, έχουμε:

`> expand (expr, exp (z+w), sin) ;`

$$\sin(x + y) + e^x e^y + e^{(z+w)} + x^n x^m$$

Αν θελήσουμε το πρόγραμμα να αναπτύσσει τα πάντα εκτός από τις δυνάμεις  $x^{n+m}$ , έχουμε:

> expand(expr, x^(n+m) );

$$\sin(x) \cos(y) + \cos(x) \sin(y) + e^x e^y + e^z e^w + x^{(n+m)}$$

□

### 4.3 Απλοποίηση

Η εντολή `simplify` απλοποιεί μια έκφραση. Η γενική σύνταξη της εντολής εμφανίζεται στον παρακάτω πίνακα.

<b>Περιγραφή:</b>	<i>Απλοποιεί μια έκφραση εφαρμόζοντας κάποιους βασικούς κανόνες.</i>	
<b>Γενική σύνταξη:</b>	<i>simplify (έκφραση)</i> <i>simplify(έκφραση, n1, n2, ... )</i> <i>simplify(έκφραση, assume=prop)</i>	
<b>Παράμετροι:</b>	n1, n2	trig, hypergeom, radical, power, ln.
	prop	Οι υποθέσεις που θέλουμε να έχουμε υπ'όψιν κατά την απλοποίηση.

Οι κανόνες που έχει υπ'όψιν του το Maple για να κάνει τις απλοποιήσεις είναι:

$$\left. \begin{array}{l} x^y \cdot x^z \rightarrow x^{y+z} \\ x^{y^z} \rightarrow x^{y \cdot z} \\ x^n \cdot y^n \rightarrow (x \cdot y)^n \\ \sqrt{x^2} \rightarrow \text{sgn } x \cdot x \end{array} \right\} \text{power}$$

$$\left. (x^n \cdot y)^{\frac{m}{n}} \rightarrow (x^m y^n)^{\frac{1}{d}} \right\} \text{radical με } x, y \text{ θετικούς, } m, n, d \text{ ακέραιους με } |m| < d$$

$$\left. \begin{array}{l} e^x \cdot e^y \rightarrow e^{x+y} \\ (e^x)^y \rightarrow e^{x \cdot y} \\ e^{x+n \cdot \ln y} \rightarrow y^n e^x \end{array} \right\} \text{exp}$$

$$\left. \begin{array}{l} \ln x^y \rightarrow y \cdot \ln x \\ \ln(x \cdot y) \rightarrow \ln x + \ln y \\ \ln(e^x) \rightarrow x \end{array} \right\} \text{ln}$$

$$\left. \begin{array}{l} \sin^2 x + \cos^2 x = 1 \\ \cosh^2 x - \sinh^2 x = 1 \end{array} \right\} trig$$

$$\left. \begin{array}{l} \arcsin(\sin(x)) \rightarrow x \\ \arccos(\cos(x)) \rightarrow x \\ \arctan(\tan(x)) \rightarrow x \end{array} \right\} hypergeom$$

Σε όλους τους παραπάνω κανόνες ισχύουν βέβαια οι περιορισμοί που απαιτούνται ώστε αυτοί να ορίζονται.

### Παράδειγμα: 4-13

Να απλοποιηθούν οι εκφράσεις  $\frac{(x-2)^2}{x^2-5x+6}$ ,  $\frac{7}{x+2} + \frac{3x}{(x+2)^2}$ .

#### Λύση:

Για τις παραπάνω δυο εκφράσεις έχουμε:

> simplify((x-2)^2/(x^2-5\*x+6));

$$\frac{x-2}{x-3}$$

> simplify(7/(x+2)+(3\*x)/(x+2)^2);

$$\frac{2(5x+7)}{(x+2)^2}$$

□

### Παράδειγμα: 4-14

Να απλοποιηθεί η έκφραση  $\frac{x(x+y)^2 - x - (x+y+1)x^2y}{x^2 + xy + x}$ .

#### Λύση:

Ορίζουμε την έκφραση expr.

> expr := (x\*(x+y)^2 - x - (x+y+1)\*x^2\*y)/(x^2+x\*y+x);

$$expr := \frac{x(x+y)^2 - x - (x+y+1)x^2y}{x^2 + xy + x}$$

```
> simplify(expr);
```

$$x - xy - 1 + y$$

□

### Παρατήρηση: 4-1

Απαιτείται ιδιαίτερη προσοχή όταν ζητάμε μια απλοποίηση. Για να γίνει θα πρέπει να δώσουμε τις σωστές υποθέσεις. Στα παρακάτω παραδείγματα χρησιμοποιούμε και την παράμετρο `assume` για να δώσουμε τις απαραίτητες υποθέσεις.

### Παράδειγμα: 4-15

Έστω ότι ζητάμε την απλοποίηση της έκφρασης  $\sqrt{x^2}$ .

#### Λύση:

Ορίζουμε την έκφραση `expr`.

```
> expr:=sqrt(x^2);
```

$$expr := \sqrt{x^2}$$

```
> simplify(expr);
```

$$\text{csgn}(x) x$$

Η παραπάνω προσπάθεια απλοποίησης της έκφρασης με χρήση της εντολής `simplify` επιστρέφει το `x` με το πρόσημό του κάνοντας χρήση της εντολής `sgn`.

Αν τώρα ζητήσουμε την απλοποίηση της έκφρασης στους πραγματικούς αριθμούς, θα έχουμε:

```
> simplify(expr, assume=real);
```

$$|x|$$

Ενώ, αν κάνουμε την υπόθεση ότι το `x` είναι θετικός αριθμός, παίρνουμε ως αποτέλεσμα το `x`.

```
> simplify(expr, assume=positive);
```

$$x$$

□



**Παράδειγμα: 4-16**

Να απλοποιηθούν οι παραστάσεις  $\ln x^2$  και  $\ln(xy)$ .

**Λύση:**

Αν ζητήσουμε την απλοποίηση, το Maple δεν εφαρμόζει τους κανόνες που αναφέρονται παραπάνω. Αυτό συμβαίνει γιατί δεν έχουμε τις σωστές υποθέσεις για το  $x$ .

```
> simplify(ln(x^2), ln);
```

$$\ln(x^2)$$

Αν τώρα κάνουμε την υπόθεση ότι βρισκόμαστε στους θετικούς αριθμούς, τότε εφαρμόζεται ο κανόνας.

```
> simplify(ln(x^2), ln, assume=positive);
```

$$2 \ln(x)$$

```
> simplify(ln(x*y), ln);
```

$$\ln(xy)$$

Όμοια, αν και εδώ υποθέσουμε ότι  $x > 0$ .

```
> simplify(ln(x*y), ln, assume=positive);
```

$$\ln(x) + \ln(y)$$

□

**Παράδειγμα: 4-17**

Να απλοποιηθεί η παράσταση  $\arcsin(\sin x)$ .

**Λύση:**

Ονομάζουμε την παραπάνω έκφραση  $expr$ .

```
> expr:=arcsin(sin(x));
```

$$expr := \arcsin(\sin(x))$$

Εφαρμόζοντας την εντολή `simplify` στην παραπάνω έκφραση, δεν έχουμε κάποια απλοποίησή της.

```
> simplify(expr);
```

$$\arcsin(\sin(x))$$

Αν κάνουμε τώρα την υπόθεση ότι το διάστημα στο οποίο κινείται το  $x$  είναι το  $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ , τότε:

```
> simplify(expr, assume=RealRange(-Pi/2, Pi/2));
```

$$x$$

□

#### Παράδειγμα: 4-18

Να απλοποιηθεί η παράσταση  $\cos^2 x + \sin^2 x + \frac{1}{(2^x)^3}$ .

#### Λύση:

Ονομάζουμε την παραπάνω έκφραση με τη μεταβλητή  $expr$ .

```
> expr:=cos(x)^2+sin(x)^2 + (2^x)^(-3);
```

$$expr := \cos(x)^2 + \sin(x)^2 + \frac{1}{(2^x)^3}$$

Ζητάμε την απλοποίησή της.

```
> simplify(expr);
```

$$1 + 8^{(-x)}$$

Ζητάμε την απλοποίησή της εφαρμόζοντας μόνο τους τριγωνομετρικούς κανόνες.

```
> simplify(expr, trig);
```

$$\frac{(2^x)^3 + 1}{(2^x)^3}$$

Ζητάμε την απλοποίησή της εφαρμόζοντας μόνο τους κανόνες των δυνάμεων.

```
> simplify(expr, power);
```

$$\cos(x)^2 + \sin(x)^2 + 8^{(-x)}$$

Ζητάμε την απλοποίησή της εφαρμόζοντας τους τριγωνομετρικούς κανόνες και αυτούς των δυνάμεων.

> simplify(expr, trig, power);

$$1 + 8^{(-x)}$$

Παρατηρούμε ότι το αποτέλεσμα είναι ίδιο με αυτό που δίνει η εντολή simplify χωρίς να δώσουμε παραμέτρους.

Τέλος, ας κάνουμε την απλοποίηση ζητώντας μορφή radical.

> simplify(expr, radical);

$$\frac{\cos(x)^2 (2^x)^3 + \sin(x)^2 (2^x)^3 + 1}{(2^x)^3}$$

□

#### Παρατήρηση: 4-2

Μια ακόμα δυνατότητα που έχουμε με την εντολή simplify είναι να θεωρήσουμε ότι ισχύουν κάποιες συγκεκριμένες σχέσεις ανάμεσα στις μεταβλητές που παρουσιάζονται στην προς απλοποίηση έκφραση.

#### Παράδειγμα: 4-19

Αν ζητήσουμε την απλοποίηση της έκφρασης  $xy + xyz - 1$  με την υπόθεση ότι ισχύει  $xy = 1$ , τότε θα έχουμε:

#### Λύση:

Ορίζουμε την έκφραση:

> expr := x\*y + x\*y\*z - 1;

$$expr := xy + xyz - 1$$

Έτσι, θεωρώντας και την υπόθεση  $xy = 1$  έχουμε την απλοποίηση:

> simplify(expr, {x\*y=1});

z

□

#### Παράδειγμα: 4-20

Αν ζητήσουμε την απλοποίηση της έκφρασης  $x^5 + y^5 = 1$  με την υπόθεση ότι ισχύει  $x^2 + y^2 = 1$ , τότε θα έχουμε:

**Λύση:**

Ορίζουμε την έκφραση:

```
> expr:=x^5+y^5;
```

$$expr := x^5 + y^5$$

```
> simplify(expr, {x^2+y^2=1}, [x, y]);
```

$$y^5 + xy^4 - 2xy^2 + x$$

Στην παραπάνω περίπτωση το Maple λύνει την υπόθεση ως προς  $x^2$  και αντικαθιστά. Αν όμως θέλουμε να λύσουμε πρώτα ως προς  $y^2$ , έχουμε:

```
> simplify(expr, {x^2+y^2=1}, [y, x]);
```

$$x^5 + yx^4 - 2yx^2 + y$$

□

#### Παράδειγμα: 4-21

Να απλοποιηθεί η παράσταση  $\sqrt{(x-9)^2}$ .

**Λύση:**

Θέτουμε με  $a$  την παράσταση  $\sqrt{(x-9)^2}$ .

```
> a:=sqrt((x-9)^2);
```

$$a := \sqrt{(x-9)^2}$$

Αν ζητήσουμε την απλοποίησή της, τότε το Maple θα μας επιστρέψει μια έκφραση στην οποία μπροστά είναι το πρόσημο του  $x-9$ .

```
> simplify(a) ;
```

$$csgn(x-9)(x-9)$$

Αν κάνουμε την υπόθεση ότι  $x \geq 9$ , τότε:

```
> simplify(a) assuming x>=9;
```

$$x-9$$

Αν κάνουμε την υπόθεση ότι  $x \leq 9$ , τότε:

```
> simplify(a) assuming x<=-9;
      -x + 9
```

□

### Παράδειγμα: 4-22

Να απλοποιηθεί η παράσταση  $\sqrt{(1-x^2)^2}$ .

#### Λύση:

Ορίζουμε την παράσταση.

```
> expr:=sqrt((1-x^2)^2);
```

$$expr := \sqrt{(1-x^2)^2}$$

Αν κάνουμε την απλοποίηση χωρίς καμία υπόθεση, έχουμε:

```
> simplify(expr);
```

$$csgn(-1+x^2)(-1+x^2)$$

Αν θέλουμε την απλοποίηση της παράστασης με την υπόθεση ότι  $-1 \leq x \leq 1$

```
> simplify(expr) assuming -1<=x<=1;
```

Error, unexpected relational operator

Το μήνυμα λάθους προκύπτει γιατί πρέπει να δώσουμε την υπόθεση στην μορφή  $x \geq -1$  και  $x \leq 1$ , έτσι έχουμε:

```
> simplify( expr ) assuming x>=-1 and x<=1;
```

$$1-x^2$$

Αν ζητήσουμε την απλοποίηση με την υπόθεση ότι  $x \in (-\infty, -1) \cup (1, \infty)$ , τότε θα έχουμε μήνυμα λάθους.

```
> simplify( expr ) assuming x<=-1 and x>=1;
```

Error, (in assuming) contradictory assumptions

Η σωστή σύνταξη για αυτή την υπόθεση είναι:

```
> simplify(expr) assuming OrProp( RealRange(-infinity,-1), RealRange(1,infinity) );
```

$$-1 + x^2$$

□

## 4.5 Collect

Η εντολή `collect` προσπαθεί να φέρει μια παράσταση σε πιο «συνοπτική» μορφή εξάγοντας κοινούς παράγοντας αλλά και κάνοντας κάποιες πράξεις.

<b>Περιγραφή:</b>	<b>Κάνει τις «πράξεις» σε μία έκφραση.</b>	
<b>Γενική σύνταξη:</b>	<b>collect(έκφραση, x);</b>	
<b>Παράμετροι:</b>	x	Μεταβλητή.

### Παράδειγμα: 4-23

Να γίνουν οι πράξεις στην έκφραση  $6(x+4) \cdot (x-1)$ .

**Λύση:**

```
> collect (6* (x+4) * (x-1) , x) ;
```

$$-24 + 6x^2 + 18x$$

□

### Παράδειγμα: 4-24

Να ορισθεί η έκφραση `expr` ως ανάπτυγμα της  $(x-y+z)^2 \cdot (x+z)$  και στη συνέχεια να δοθεί σε απλούστερη μορφή ως προς  $x$ ,  $y$  και  $z$ .

**Λύση:**

Ορίζουμε την έκφραση.

```
> expr := expand ( (x-y+z) ^2 * (x+z) ) ;
```

$$expr := x^3 + 3x^2z - 2x^2y - 4xyz + 3xz^2 + y^2x + y^2z - 2yz^2 + z^3$$

Κάνουμε τις πράξεις ως προς τη μεταβλητή  $z$ .

```
> collect (expr, z) ;
```

$$z^3 + (3x - 2y)z^2 + (-4xy + 3x^2 + y^2)z + x^3 + y^2x - 2x^2y$$

Κάνουμε τις πράξεις ως προς τη μεταβλητή  $y$ .

```
> collect (expr, y) ;
```

$$(x+z)y^2 + (-2x^2 - 4xz - 2z^2)y + x^3 + 3x^2z + 3xz^2 + z^3$$

Κάνουμε τις πράξεις ως προς τη μεταβλητή  $x$ .

> collect (expr, x) ;

$$x^3 + (3z - 2y)x^2 + (y^2 + 3z^2 - 4yz)x + z^3 + y^2z - 2yz^2$$

□

## 4.6 Αντικατάσταση

Δύο πολύ χρήσιμες εντολές, με τις οποίες μπορούμε να κάνουμε αντικατάσταση σε εκφράσεις που περιέχουν μεταβλητές, είναι η `subs` και η εντολή `eval`.

Περιγραφή:	Κάνει αντικατάσταση σε μια έκφραση.	
Γενική σύνταξη:	<code>subs (x=a, έκφραση)</code> <code>eval (έκφραση, x=a)</code>	
Παράμετροι:	x:	Η μεταβλητή που θα αντικατασταθεί με την τιμή $a$ .

### Παράδειγμα: 4-25

Να βρεθεί η τιμή της παράστασης  $x^2 + x + 1$  για  $x=2$ .

**Λύση:**

> subs (x=2, x^2+x+1) ;

7

Το ίδιο αποτέλεσμα θα έχουμε και με χρήση της εντολής `eval`.

> eval (x^2+x+1, x=2) ;

7

Οι αντικαταστάσεις μιας μεταβλητής σε μια έκφραση δεν είναι κατ' ανάγκη αριθμητικές. Μπορούμε να κάνουμε αντικατάσταση χρησιμοποιώντας άλλες μεταβλητές. Αν στην παραπάνω έκφραση θέλουμε να κάνουμε την αντικατάσταση  $x=a+2$ , τότε:

> subs (x=a+2, x^2+x+1) ;

$$(a+2)^2 + a + 3$$

□

**Παράδειγμα: 4-26**

Δίνεται η εξίσωση  $(x-2)^2 + (y+3)^2 = 25$ . Να γίνει η αλλαγή των μεταβλητών  $z = x-2$ ,  $w = y+3$ .

**Λύση:**

Ορίζουμε την εξίσωση eqn.

```
> eqn := (x-2)^2 + (y+3)^2 = 25;
```

$$eqn := (x-2)^2 + (y+3)^2 = 25$$

Κάνουμε την αντικατάσταση στην eqn ως εξής: όπου  $x-2$  το  $z$  και όπου  $y+3$  το  $w$ .

```
> subs (x-2=z, y+3=w, eqn) ;
```

$$z^2 + w^2 = 25$$

□

**Παράδειγμα: 4-27**

Δίνεται η έκφραση  $\frac{e^x}{1-e^x}$ . Να γίνει η αντικατάσταση  $e^x = y$ .

**Λύση:**

Ορίζουμε την έκφραση expr.

```
> expr := exp(x) / (1-exp(x)) ;
```

$$expr := \frac{e^x}{1-e^x}$$

```
> subs (exp(x)=y, expr) ;
```

$$\frac{y}{1-y}$$

□



**Παράδειγμα: 4-28**

Να απλοποιηθεί η έκφραση  $\frac{a}{ab-b^2} - \frac{b}{a^2-ab}$  και στη συνέχεια να γίνει η αντικατάσταση  $a = \sin x$ ,  $b = \cos x$ .

**Λύση:**

Ορίζουμε την έκφραση με το όνομα `expr`.

```
> expr:=a/(a*b-b^2)-b/(a^2-a*b);
```

$$expr := \frac{a}{ab-b^2} - \frac{b}{a^2-ab}$$

Απλοποιούμε

```
> simplify(expr);
```

$$\frac{a+b}{ab}$$

και με την εντολή `subs` κάνουμε την αντικατάσταση.

```
> subs(a=sin(x),b=cos(x),%);
```

$$\frac{\sin(x) + \cos(x)}{\sin(x) \cos(x)}$$

□



# Κεφάλαιο 5

## Βασικές Δομές Δεδομένων

Το Maple έχει 29 βασικούς τύπους δεδομένων. Στο κεφάλαιο 3 είδαμε μερικούς από αυτούς, όπως τους ακεραίους, τα κλάσματα, τους αριθμούς κινητής υποδιαστολής, τα strings κ.ά. Στο κεφάλαιο αυτό θα παρουσιάσουμε πιο σύνθετους τύπους δεδομένων.

Μια δομή δεδομένων (data structure) είναι μια συλλογή δεδομένων που επιπλέον είναι εφοδιασμένη με μια δομή. Οι δομές που θα παρουσιάσουμε εδώ είναι η ακολουθία εκφράσεων, η λίστα, το σύνολο, το array και το table.

## 5.1 Ακολουθίες Εκφράσεων

Η ακολουθία εκφράσεων, που αποτελεί βασική δομή στο Maple, είναι οποιαδήποτε συλλογή εκφράσεων. Για να ορίσουμε μια ακολουθία εκφράσεων απλώς παραθέτουμε τα στοιχεία της χωρίζόμενα με κόμμα. Έτσι, για παράδειγμα, θα ορίσουμε την ακολουθία εκφράσεων  $px$  ως εξής:

```
> px:=2, sum, abc, Pi;
```

$px := 2, sum, abc, \pi$

Αν ζητήσουμε τον τύπο της μεταβλητής  $px$ , έχουμε:

```
> whattype (px) ;
```

$exprseq$

Για να αναφερθούμε σε έναν από τους όρους της ακολουθίας χρησιμοποιούμε τις τετράγωνες αγκύλες. Έτσι, αν θέλουμε το δεύτερο όρο της παραπάνω ακολουθίας, δίνουμε την εντολή:

```
> px[2] ;
```

$sum$

Εκτός από το να παραθέσουμε τα στοιχεία μιας ακολουθίας, μπορούμε, να τη δημιουργήσουμε, χρησιμοποιώντας το γενικό όρο της με την εντολή  $seq$ .

<i>Περιγραφή:</i>	<i>Δημιουργεί μια ακολουθία.</i>	
<i>Γενική σύνταξη:</i>	$seq(f, i = m..n)$	
<i>Παράμετροι:</i>	f	Γενικός όρος της ακολουθίας.
	i	Ο δείκτης.
	m, n	Αρχική και τελική αριθμητική τιμή.

Έτσι, θα δημιουργήσουμε την ακολουθία εκφράσεων που ως όρους έχει τις δυνάμεις του  $x$  με τον εκθέτη να παίρνει τιμές από  $-3$  έως  $6$ .

```
> seq(x^i, i=-3..6) ;
```

$\frac{1}{x^3}, \frac{1}{x^2}, \frac{1}{x}, 1, x, x^2, x^3, x^4, x^5, x^6$

```
> whattype (%) ;
```

*exprseq*

Μια από τις πιο σημαντικές δυνατότητες του Maple σχετικά με τις ακολουθίες εκφράσεων είναι ότι μπορούμε να εισαγάγουμε ένα στοιχείο που ήδη υπάρχει. Έστω ότι ορίζουμε την ακολουθία εκφράσεων:

```
> px:=2, sum, abc, Pi;
```

*px := 2, sum, abc, π*

```
> px:=px, sum;
```

*px := 2, sum, abc, π, sum*

```
> px, px;
```

*2, sum, abc, π, sum, 2, sum, abc, π, sum*

**Παράδειγμα: 5-29**

Δίνεται το πολυώνυμο  $x^3 - 2x^2 + x + 12$ . Θα δημιουργήσουμε μια ακολουθία εκφράσεων με όρους τους όρους του πολυωνύμου.

**Λύση:**

Ορίζουμε το πολυώνυμο.

```
> polynomial:=x^3-2*x^2+x+12;
```

*polynomial := x<sup>3</sup> - 2 x<sup>2</sup> + x + 12*

Εξάγουμε όλους τους όρους του πολυωνύμου.

```
> op(polynomial);
```

*x<sup>3</sup>, -2 x<sup>2</sup>, x, 12*

<i>Περιγραφή:</i>	<i>Εξάγει της επιμέρους εκφράσεις από μια έκφραση.</i>	
<i>Γενική σύνταξη:</i>	<i>op(i,j,expr)</i>	
<i>Παράμετροι:</i>	expr	η έκφραση
	i,j	Ακέραιοι που δείχνουν ένα στοιχείο της ακολουθίας.

Με την εντολή `whattype` βλέπουμε ότι το προηγούμενο αποτέλεσμα είναι ακολουθία εκφράσεων.

```
> whattype(%);
```

*exprseq*

Εξάγουμε μόνο το δεύτερο όρο της ακολουθίας.

```
> op(2, polynomial);
```

$$-2x^2$$

<b>Περιγραφή:</b>	<b>Επιστρέφει το πλήθος των επιμέρους εκφράσεων.</b>
<b>Γενική σύνταξη:</b>	<b>nops(έκφραση)</b>

```
> nops(polynomial);
```

4

□

## 5.2 Λίστες

Για να ορίσουμε μια λίστα από αριθμούς και όχι μόνο, χρησιμοποιούμε τις τετράγωνες παρενθέσεις. Λίστα είναι μια παράθεση στοιχείων όχι απαραίτητα αριθμητικών που έχουν μια σειρά.

<b>Περιγραφή:</b>	<b>Δημιουργία λίστας στοιχείων.</b>
<b>Γενική σύνταξη:</b>	<b>[στοιχείο1, στοιχείο2,...]</b>

```
> category := [cat, dog];
```

$$category := [cat, dog]$$

```
> whattype(category);
```

*list*

Για να προσπελάσουμε ένα από τα στοιχεία μιας λίστας, πρέπει να καλέσουμε το όνομα της λίστας και μέσα σε αγκύλες τον αριθμό που δείχνει τη θέση του στοιχείου.

```
> category[1];
```

*cat*

<b>Περιγραφή:</b>	<b>Εξετάζει εάν ένα στοιχείο ανήκει σε μια λίστα, σε ένα σύνολο ή συνάρτηση.</b>
<b>Γενική σύνταξη:</b>	<b>member(στοιχείο, λίστα)</b>

**Παράδειγμα: 5-30**

Ορίζουμε μια λίστα colors.

```
> colors := [black, white, red, green, blue];  
           colors := [black, white, red, green, blue]
```

```
> whattype (colors);
```

*list*

Ο αριθμός των στοιχείων μιας λίστας.

```
> nops (colors);
```

*5*

Τα στοιχεία μιας λίστας.

```
> op (colors);
```

*black, white, red, green, blue*

Το δεύτερο στοιχείο μιας λίστας.

```
> colors [2];
```

*white*

Το στοιχείο red ανήκει στη λίστα colors.

```
> member (red, colors);
```

*true*

Το στοιχείο magenda δεν ανήκει στην λίστα colors.

```
> member (magenta, colors);
```

*false*

Αν θέλουμε να εισαγάγουμε ένα στοιχείο σε μια λίστα, τότε:

```
> colors := [magenta, op (colors)];  
           colors := [magenta, black, white, red, green, blue]
```

```
> member (magenta, colors);
```

*true*

□

**Παράδειγμα: 5-31**

Θα δημιουργήσουμε μια λίστα με στοιχεία τους όρους της ακολουθίας  $n^2$ , με  $n = -3..3$ .

```
> seq1 := [ seq( n^2, n=-3..3 ) ];
           seq1 := [9, 4, 1, 0, 1, 4, 9]

> seq1[2];
           4

> seq1[3];
           1
```

□

<b>Περιγραφή:</b>	<b>Ταξινομεί τα στοιχεία μια λίστας.</b>
<b>Γενική σύνταξη:</b>	<b>sort(λίστα)</b>

```
> numbers := [2, 4, -2, 64, 0.3];

           numbers := [2, 4, -2, 64, 0.3]

> sort(numbers);
           [-2, 0.3, 2, 4, 64]
```

Με τις εντολές `select` και `remove` μπορούμε να επιλέξουμε και να αφαιρέσουμε ένα στοιχείο από μια λίστα.

<b>Περιγραφή:</b>	<b>Επιλέγει από μια λίστα ένα ή περισσότερα στοιχεία.</b>
<b>Γενική σύνταξη:</b>	<b>select(opt, λίστα)</b>

<b>Περιγραφή:</b>	<b>Μετακινεί από μια λίστα ένα ή περισσότερα στοιχεία.</b>
<b>Γενική σύνταξη:</b>	<b>remove(opt, λίστα)</b>

Δημιουργούμε τη λίστα των ακεραίων αριθμών από το 10 έως το 20.

```
> integers := [seq(i, i=10..20)];
           integers := [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
```



Επιλέγουμε όσους από τη λίστα είναι πρώτοι αριθμοί.

```
> select(isprime, integers);
      [11, 13, 17, 19]
```

Αφαιρούμε τους πρώτους αριθμούς από τη λίστα.

```
> remove(isprime, integers);
      [10, 12, 14, 15, 16, 18, 20]
```

Σε μια λίστα αριθμών, με χρήση των εντολών `max` και `min`, μπορούμε να βρούμε τη μέγιστη και ελάχιστη τιμή αντίστοιχα.

<i>Περιγραφή:</i>	<i>Βρίσκει το μέγιστο ή το ελάχιστο αριθμών.</i>
<i>Γενική σύνταξη:</i>	<code>max(x1, x2, ... )</code> <code>min(x1, x2, ... )</code>

```
> num := -2, 3, 4, 234, 5, 23, -43, 23, 423, 43, 6, 64;
```

```
      num := -2, 3, 4, 234, 5, 23, -43, 23, 423, 43, 6, 64
```

```
> min(num);
      -43
```

□

Για να δημιουργήσουμε τη γραφική παράσταση μιας λίστας ή ενός συνόλου αριθμών μπορούμε να χρησιμοποιήσουμε την εντολή `listplot` (η εντολή αυτή είναι μέρος του πακέτου `plots`).

<b>Περιγραφή:</b>	<b>Παριστάνει γραφικά μια λίστα σημείων.</b>	
<b>Γενική σύνταξη:</b>	<b><i>listplot</i>( λίστα αριθμών, παράμετροι )</b>	
<b>Παράμετροι:</b>	color=	aquamarine black blue navy coral cyan brown gold green gray grey khaki magenta maroon orange pink plum red sienna tan turquoise violet wheat white yellow
	axes =	frame, boxed, normal, none
	style=	line, point, patch, or patchnogrid. προεπιλογή είναι η γραμμή (line).
	symbol=	box, circle, cross, diamond, or point.
	labels=	[x, y]
	title=	Τίτλος της γραφικής παράστασης.
<b>Πακέτο:</b>	<b><i>plots</i></b>	

Η εντολή `listplot` έχει ίδιες παραμέτρους με την εντολή `plot` (δες Κεφάλαιο 5)

#### Παράδειγμα: 5-32

Να κατασκευαστεί η ακολουθία  $a_n = n^2$  και να παρασταθεί γραφικά για  $n = -4, \dots, 4$ .

#### Λύση:

Ορίζουμε την ακολουθία `seq1`.

```
> seq1 := [ seq( n^2, n=-4..4 ) ];
```

`seq1 := [ 16, 9, 4, 1, 0, 1, 4, 9, 16 ]`

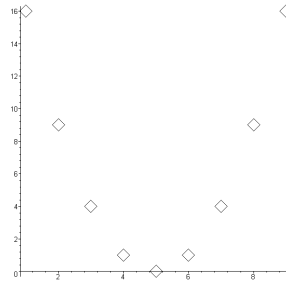
Ενεργοποιούμε το πακέτο εντολών `plots`.

```
> with(plots) :
```

Αναπαριστούμε σε ένα σύστημα αξόνων τα σημεία της λίστας χρησιμοποιώντας σταυρούς.

```
>listplot(seq1,style=point,
symbolsize=50);
```

```
symbol=diamond,
```



□

### Παράδειγμα: 5-33

Να κατασκευάσετε την ακολουθία  $a_n = \sin(n)$  και να την παραστήσετε γραφικά για  $n = 0, \dots, 20$  χρησιμοποιώντας γραμμές (line).

#### Λύση:

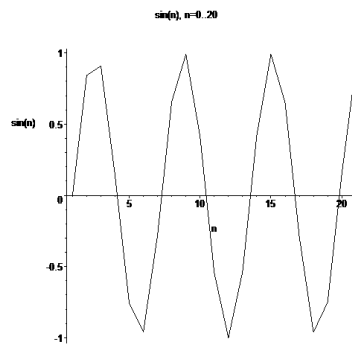
Ορίζουμε την ακολουθία seq2.

```
> seq2:= [ seq( sin(n), n=0..20 ) ];
```

```
seq2 := [0, sin(1), sin(2), sin(3), sin(4), sin(5), sin(6), sin(7), sin(8), sin(9), sin(10),
sin(11), sin(12), sin(13), sin(14), sin(15), sin(16), sin(17), sin(18), sin(19),
sin(20)]
```

```
> with(plots):
```

```
listplot( seq2, style=line, labels=[n, "sin(n)"]
,title="sin(n), n=0..20");
```



□

### 5.3 Σύνολα

Οι σγουρές παρενθέσεις χρησιμοποιούνται για να δημιουργήσουμε σύνολα από στοιχεία, όχι απαραίτητα αριθμητικά, που δεν έχουν σειρά.

<b>Περιγραφή:</b>	<b>Δημιουργεί ένα σύνολο στοιχείων.</b>
<b>Γενική σύνταξη:</b>	<b>{ στοιχείο1, στοιχείο2,... }</b>

**Παράδειγμα: 5-34**

```
> A := {seq( n^2, n=-10..10 ) };
      A := {0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100}
```

```
> whattype (A) ;
```

*set*

Και στα σύνολα μπορούμε να χρησιμοποιήσουμε τις εντολές `op` και `nops` που είδαμε στις ακολουθίες και στις λίστες.

Άρα, ο αριθμός των στοιχείων του A είναι:

```
> nops (A) ;
```

11

Ενώ, το τρίτο στοιχείο του A είναι:

```
> op (3, A) ;
```

4

Το κενό σύνολο είναι:

```
> keno := { } ;
```

*keno := { }*

□

#### Σχέσεις-Πράξεις Σύνολων

Θα δούμε τώρα τις βασικές σχέσεις και πράξεις μεταξύ συνόλων.

Για να εξετάσουμε αν ένα στοιχείο ανήκει σε ένα σύνολο, χρησιμοποιούμε την εντολή `in`.

<b>Γενική σύνταξη:</b>	<b><i>In</i></b>
------------------------	------------------

Για να εξετάσουμε αν ένα σύνολο είναι υποσύνολο ενός άλλου συνόλου, χρησιμοποιούμε την εντολή `subset`

<b>Γενική σύνταξη:</b>	<b><i>subset</i></b>
------------------------	----------------------

Η ένωση συνόλων γίνεται με την εντολή `union`.

<b>Γενική σύνταξη:</b>	<b><i>union</i></b>
------------------------	---------------------

> A union B;

$$A \cup B$$

Η διαφορά συνόλων γίνεται με την εντολή `minus`.

<b>Γενική σύνταξη:</b>	<b><i>minus</i></b>
------------------------	---------------------

>A minus B;

$$A \setminus B$$

Η τομή συνόλων γίνεται με την εντολή `intersect`.

<b>Γενική σύνταξη:</b>	<b><i>intersect</i></b>
------------------------	-------------------------

>A intersect B;

$$A \cap B$$

### Παράδειγμα: 5-35

Ορίζουμε δύο σύνολα  $A = \{1, 3, 5, 7\}$ ,  $B = \{0, 2, 4, 6\}$ . Θα υπολογίσουμε την ένωση, την τομή κ.ά.

#### Λύση:

Ορίζουμε τα σύνολα A και B.

>A := {1, 3, 5, 7};

$$A := \{1, 3, 5, 7\}$$

>B := {0, 2, 4, 6};

$$B := \{0, 2, 4, 6\}$$

Το σύνολο C ως η ένωση των A και B, ορίζεται ως εξής:

>C :=A union B;

$$C := \{0, 1, 2, 3, 4, 5, 6, 7\}$$

```
> E:=A minus B;
```

$E := \{1, 3, 5, 7\}$

```
> H:=A intersect B;
```

$H := \{ \}$

Εξετάζουμε αν το σύνολο A είναι υποσύνολο του συνόλου B.

```
> A subset B;
```

*false*

Εξετάζουμε αν το στοιχείο 2 ανήκει στα σύνολα A και B.

```
> member(2, A);
```

*false*

```
> member(2, B);
```

*true*

Στα σύνολα, ένας άλλος τρόπος να εξετάσουμε εάν ένα στοιχείο ανήκει σε κάποιο από αυτά είναι με χρήση της εντολής `evalb`:

```
> evalb(2 in B);
```

*true*

## 5.4 Πίνακες

Η δομή του πίνακα (`table`) είναι μια γενίκευση της δομής λίστας. Μια λίστα περιέχει εκφράσεις και, για να αναφερθούμε σε κάθε μια έκφραση, χρησιμοποιούμε έναν ακέραιο, που αντιστοιχεί στη θέση της έκφρασης στη λίστα. Στη δομή πίνακα αναφερόμαστε σε ένα στοιχείο του χρησιμοποιώντας όχι απαραίτητα έναν ακέραιο αλλά οποιαδήποτε άλλη έκφραση. Επίσης, στη δομή `table` δεν καθορίζουμε εξαρχής το μέγεθός της. Αυτό καθορίζεται δυναμικά με την πρόσθεση ή την αφαίρεση στοιχείων. Η δομή `table`, λοιπόν, ομοιάζει περισσότερο με τη δομή της εγγραφής (`Record`) που έχουμε στις γλώσσες, όπως η C και η Pascal.

Η εντολή `table` δημιουργεί έναν πίνακα.

<i>Περιγραφή:</i>	<i>Δημιουργεί μια δομή τύπου table.</i>
<i>Γενική σύνταξη:</i>	<i>table( )</i>

Ορίζουμε με *t* έναν πίνακα.

```
> t:=table();
```

```
t := table()
```

Καταχωρούμε στην μεταβλητή *t* το στοιχείο *blue* με δείκτη *sky*,

```
> t[sky]:=blue;
```

```
tsky := blue
```

το στοιχείο *brown* με δείκτη *ground*,

```
> t[ground]:=brown;
```

```
tground := brown
```

το στοιχείο *black* με δείκτη *night*.

```
> t[night]:=black;
```

```
tnight := black
```

```
> t();
```

```
table([ground = brown, night = black, sky = blue])()
```

Η δομή *array* είναι και αυτή μια γενίκευση της δομής της λίστας. Κάθε στοιχείο της έχει ένα δείκτη που είναι οπωσδήποτε ένας ακέραιος αριθμός. Το *array* έχει συγκεκριμένο μέγεθος και αυτό δε δίνεται δυναμικά, αλλά ορίζεται εξ αρχής.

<i>Περιγραφή:</i>	<i>Δημιουργεί μια δομή τύπου array.</i>
<i>Γενική σύνταξη:</i>	<i>Array(διάσταση)</i>

Κατασκευάζουμε ένα *Array* μιας γραμμής και 10 στηλών.

```
> a:=Array(1..10);
```

```
a := array(1 .. 10, [ ])
```

Για να το εμφανίσουμε χρησιμοποιούμε την εντολή *print*.

```
> print(a);
```

```
[a1, a2, a3, a4, a5, a6, a7, a8, a9, a10]
```

Κατασκευάζουμε ένα *array* 5 γραμμών και 6 στηλών.

```
>b:=Array( 1..5, 1..6 );
```

```
      b := array(1 .. 5, 1 .. 6, [ ])
```

```
>print(b);
```

$$\begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} & b_{1,5} & b_{1,6} \\ b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} & b_{2,5} & b_{2,6} \\ b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} & b_{3,5} & b_{3,6} \\ b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} & b_{4,5} & b_{4,6} \\ b_{5,1} & b_{5,2} & b_{5,3} & b_{5,4} & b_{5,5} & b_{5,6} \end{bmatrix}$$

Κατασκευάζουμε ένα array 2 γραμμών και 3 στηλών ορίζοντας και τα στοιχεία του.

```
>c:=Array(1..2,1..3, [[1,2,3],[2,-2,3]]);
```

$$a := \begin{bmatrix} 1 & 2 & 3 \\ 2 & -2 & 3 \end{bmatrix}$$

Αν θέλουμε το στοιχείο που βρίσκεται στην (1,2) θέση, έχουμε:

```
>c[1,2];
```



# Κεφάλαιο 6

## Διανύσματα-Μήτρες

Ιδιαίτερες περιπτώσεις της δομής `array` είναι οι δομές διάνυσμα (vector) και μήτρα (matrix). Στο κεφάλαιο αυτό θα εξετάσουμε εντολές που αφορούν τα διανύσματα και τις μήτρες και θα δούμε πώς μπορούμε να κάνουμε πράξεις με αυτές. Επιπλέον, θα παρουσιάσουμε το πακέτο για τη Γραμμική Άλγεβρα (LinearAlgebra packages), το οποίο μας δίνει τη δυνατότητα να κάνουμε πολύπλοκους υπολογισμούς με μήτρες και διανύσματα καθώς και να υπολογίζουμε διάφορα χαρακτηριστικά μεγέθη τους.

### 6.1 Διανύσματα

Ένα μονοδιάστατο `array`, του οποίου η αρίθμηση ξεκινάει από το 1, είναι ένα διάνυσμα. Ας δούμε ένα παράδειγμα διανύσματος ορισμένου με τη χρήση `array`.

Ορίζουμε ένα `array`, του οποίου η αρίθμηση ξεκινάει από το 1.

```
> a:=array( 1..3 );
```

```
a := array(1 .. 3, [ ])
```

Αν εξετάσουμε τον τύπο του με την εντολή `type`, τότε το πρόγραμμα θα μας επιστρέψει την τιμή «αληθής» και στις δύο προπτώσεις, αφού βέβαια είναι και `vector` και `array`.

```
> type( a, vector );
```

*true*

```
> type( a, array );
```

*true*

Ας ορίσουμε τώρα, για παράδειγμα, το `array` `b`, του οποίου η αρίθμηση ξεκινά από το 0.

```
> b:=array( 0..3 );
```

*b := array(0 .. 3, [ ])*

Αν τώρα εξετάσουμε τον τύπο του `b`, τότε ο πρόγραμμα θα μας επιστρέψει την τιμή «ψευδής» για την περίπτωση `vector` και «αληθής» για την περίπτωση `array`.

```
> type( b, vector );
```

*false*

```
> type( b, array );
```

*true*

Τα διανύσματα μπορούν εύκολα να εισαχθούν και χωρίς χρήση του `array`, με χρήση των συμβόλων `<, >` ή την εντολή `Vector`.

<b>Περιγραφή:</b>	<i>Δημιουργεί ένα διάνυσμα.</i>
<b>Γενική σύνταξη:</b>	<i>&lt; στοιχείο1, στοιχείο2,... &gt;</i>

```
> s:=<1, 2, 3>;
```

$$s := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

```
> whattype(s);
```

*Vector*  
*column*

Αν θέλουμε το δεύτερο στοιχείο του διανύσματος στήλης, τότε:

```
>s [2];
```

2

<i>Περιγραφή:</i>	<i>Δημιουργεί ένα διάνυσμα.</i>	
<i>Γενική σύνταξη:</i>	<i>Vector( n,m,L,f)</i>	
<i>Παράμετροι:</i>	n,m	Η διάστασή του.
	L	Οι λίστες με τα στοιχεία του διανύσματος .
	f	Η συνάρτηση που χρησιμοποιείται για να δημιουργηθούν τα στοιχεία του διανύσματος.

```
>Vector ([1,2,3]);
```

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Ένα μηδενικό διάνυσμα διαστάσεως 4.

```
>Vector (4, 'shape'='zero');
```

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Το διάνυσμα που δημιουργείται από τη συνάρτηση f.

```
> f:=i->x^i
```

$$f:=i \rightarrow x^i$$

```
>Vector (3, f);
```

$$\begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}$$

## 6.2 Μήτρες

Ένα διδιάστατο array, η αρίθμηση του οποίου ξεκινάει από το 1 και για τις δύο διαστάσεις, είναι μια μήτρα. Ας το δούμε στην πράξη με το παρακάτω παράδειγμα:

```
> a:=array( 1..2, 1..2 );
```

```
a := array(1 .. 2, 1 .. 2, [ ])
```

Αν εξετάσουμε τον τύπο του a, θα έχουμε την τιμή «αληθής» και στις δύο περιπτώσεις, και για matrix και για array.

```
> type( a, matrix );
```

```
true
```

```
> type(a, array );
```

```
true
```

Αν τώρα εξετάσουμε τον τύπο του b, τότε το πρόγραμμα θα μας επιστρέψει την τιμή «ψευδής» για την περίπτωση matrix και «αληθής» για την περίπτωση array.

```
> b:=array( 0..2, 1..2 );
```

```
b := array(0 .. 2, 1 .. 2, [ ])
```

```
> type( b, matrix );
```

```
false
```

```
> type( b, array );
```

```
true
```

Ειδικότερα, για τη δημιουργία μιας μήτρας, το Maple διαθέτει την εντολή Matrix.

<i>Περιγραφή:</i>	<i>Δημιουργεί μια Μήτρα.</i>	
<i>Γενική σύνταξη:</i>	<i>Matrix( n,m,L,f)</i> <i>Matrix( n, m, init,shape)</i>	
<i>Παράμετροι:</i>	n	Το πλήθος των γραμμών της μήτρας.
	m	Το πλήθος των στηλών της μήτρας.
	L	Οι λίστες ή τα διανύσματα των στοιχείων.

	f	Η συνάρτηση που χρησιμοποιείται για να δημιουργηθούν τα στοιχεία του πίνακα.
	init	Maple procedure, table, array, list, Array, Matrix, Vector, set of equations, or expression of type algebraic; initial values for the Matrix.
	shape=	identity, scalar[x], zero, constant[x], diagonal, band[b], band[b1,b2], symmetric, skewsymmetric, antisymmetric, hermitian, skewhermitian, antihermitian, triangular, Hessenberg.

Περισσότερες παραμέτρους για την εντολή Matrix μπορούμε να βρούμε στη βοήθεια πληκτρολογώντας ? Matrix

### Παράδειγμα: 6-1

Παρουσιάζουμε κάποια χαρακτηριστικά παραδείγματα δημιουργίας μήτρας με την εντολή matrix.

Θα δημιουργήσουμε μια 2x2 μήτρα, της οποίας τα στοιχεία είναι τα στοιχεία της λίστας [1,2,3,4].

```
> Matrix(2, 2, [1, 2, 3, 4]);
```

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Αν οι γραμμές της μήτρας είναι οι λίστες [10,20] και [30,40], τότε η μήτρα δημιουργείται ως εξής:

```
> Matrix([[10, 20], [30, 40]]);
```

$$\begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix}$$

Μια 2x2 μηδενική μήτρα:

```
> Matrix(2, 2, 0);
```

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Η μήτρα A είναι 2x2 και τα στοιχεία της δημιουργούνται από τη συνάρτηση f.

```
> f := (i, j) -> x^(i+j);
```

```
A := Matrix(2,2,f);
```

$$A := \begin{bmatrix} x^2 & x^3 \\ x^3 & x^4 \end{bmatrix}$$

Μια 3x2 μήτρα με όλα τα στοιχεία της μονάδες:

```
> Matrix(3,2,1);
```

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

□

### Παράδειγμα: 6-2

Δημιουργούμε μια 3x3 ταυτοτική μήτρα.

```
> Matrix(3,3,shape=identity);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Κατασκευάζουμε δύο διαγώνιες μήτρες. Η πρώτη έχει στην κύρια διαγώνιο τον αριθμό της γραμμής  $i$ .

```
> Matrix(3,3,(i,j)->i, shape=diagonal);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

Στη δεύτερη διαγώνια μήτρα τοποθετούμε στην κύρια διαγώνιο τα στοιχεία 1,2,3,4.

```
> Matrix(1..4,1..4,<1,2,3,4>,shape=diagonal);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

**Παράδειγμα: 6-3**

Δημιουργούμε μια διαγώνια μήτρα 11x11 με διαγώνια στοιχεία τον αριθμό της γραμμής  $i$ .

```
> Matrix(11,11, (i,j)->i, shape=diagonal);
```

```

11 x 11 Matrix
Data Type: anything
Storage: diagonal
Shape: diagonal
Order: Fortran_order

```

Το Maple δεν μας εμφανίζει τη μήτρα, αλλά μόνο δίνει κάποια στοιχεία για τη δομή της. Για να εμφανισθούν μεγάλες μήτρες, ορίζουμε στην εντολή `interface` το μέγεθος των μητρών που θα εμφανίζονται.

```
> interface(rtablesizer=infinity);
```

Έτσι, έχουμε:

```
> Matrix(11,11, (i,j)->i+j, shape=diagonal);
```

```

2  0  0  0  0  0  0  0  0  0  0
0  4  0  0  0  0  0  0  0  0  0
0  0  6  0  0  0  0  0  0  0  0
0  0  0  8  0  0  0  0  0  0  0
0  0  0  0  10 0  0  0  0  0  0
0  0  0  0  0 12  0  0  0  0  0
0  0  0  0  0  0 14  0  0  0  0
0  0  0  0  0  0  0 16  0  0  0
0  0  0  0  0  0  0  0 18  0  0
0  0  0  0  0  0  0  0  0 20  0
0  0  0  0  0  0  0  0  0  0 22

```

Τη συμμετρική μήτρα A

```
> A:=Matrix(3,3,shape=symmetric);
```

$$A := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Με στοιχείο  $a_{12} = 2$

```
> A[1,2]:=2;
```

$$A_{1,2} := 2$$

```
> A;
```

$$\begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Την αντισυμμετρική μήτρα A με στοιχείο  $a_{12} = x$

```
> A:=Matrix(2,2,shape=antisymmetric,storage=sparse):A[1,2]:=x:A;
```

$$\begin{bmatrix} 0 & x \\ -x & 0 \end{bmatrix}$$

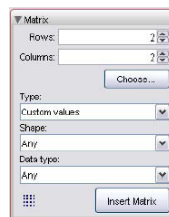
Την κάτω τριγωνική μήτρα

```
>Matrix(3,[[x],[y,y],[z,z,z]],shape=triangular[lower]);
```

$$\begin{bmatrix} x & 0 & 0 \\ y & y & 0 \\ z & z & z \end{bmatrix}$$

□

Επίσης, μπορούμε να εισαγάγουμε μια μήτρα από τις παλέτες.





Ένας εναλλακτικός τρόπος ορισμού μιας μήτρας είναι η χρήση των συμβόλων  $\langle, \rangle, |$ . Ουσιαστικά ορίζουμε τη μήτρα από τα διανύσματα από τα οποία αποτελείται. Το σύμβολο  $|$  καθορίζει τις στήλες της μήτρας.

<i>Περιγραφή:</i>	<i>Δημιουργεί μια Μήτρα κατά στήλες.</i>
<i>Γενική σύνταξη:</i>	$\langle \langle a_{11}, a_{21}, \dots, a_{m1} \rangle   \langle a_{12}, a_{22}, \dots, a_{m2} \rangle   \dots   \langle a_{1n}, a_{2n}, \dots, a_{mn} \rangle \rangle$

Έτσι, για να ορίσουμε μια 3x3 μήτρα κατά στήλες, έχουμε:

$\text{>A} := \langle \langle 2, 3, -1 \rangle | \langle 0.1, 3.3, 1 \rangle | \langle 12, 3.2, 2 \rangle \rangle;$

$$A := \begin{bmatrix} 2 & 0.1 & 12 \\ 3 & 3.3 & 3.2 \\ -1 & 1 & 2 \end{bmatrix}$$

Για να ορίσουμε τη μήτρα B από τα διανύσματα v και u, ορίζουμε αρχικά τα v και u.

$\text{>v} := \langle 1, 2, 3 \rangle;$

$$v := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$\text{>u} := \langle 4, 5, 6 \rangle;$

$$u := \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

Ορίζουμε τη μήτρα B.

$\text{>B} := \langle v | u \rangle;$

$$B := \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

Επίσης, μπορούμε να ορίσουμε τη μήτρα κατά γραμμές ως εξής:

<i>Περιγραφή:</i>	<i>Δημιουργεί μια Μήτρα κατά γραμμές.</i>
<i>Γενική σύνταξη:</i>	$\langle \langle a_{11}   a_{21}   \dots   a_{m1} \rangle, \langle a_{12}   a_{22}   \dots   a_{m2} \rangle, \dots, \langle a_{1n}   a_{2n}   \dots   a_{mn} \rangle \rangle$

Θα δημιουργήσουμε μια μήτρα οριζοντάς την κατά γραμμές.

```
> <<1|2|3>, <4|5|6>>;
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

### Παρατήρηση: 6-1

Η εντολή Matrix μπορεί να γραφεί και με το m μικρό. Η εντολή matrix είχε αναπτυχθεί το 1990 με σκοπό να συνεργάζεται με το πακέτο γραμμικής άλγεβρας linalg. Ωστόσο, από την έκδοση Maple 6 και μετά, το πακέτο αυτό έχει βελτιωθεί από το πακέτο LinearAlgebra, το οποίο παρέχει μια σειρά εντολών για τη Γραμμική Άλγεβρα. Έτσι, παρουσιάστηκε το 1998 η εντολή Matrix (η διαφορά τους το κεφαλαίο M) η οποία συνεργάζεται καλύτερα με το νέο πακέτο εντολών. Στην τρέχουσα έκδοση Maple 10 συνυπάρχουν και οι δύο εντολές.

## 6.3 Πράξεις με Διανύσματα και Μήτρες

Οι πράξεις με μήτρες ορίζονται κατά το συνήθη φυσιολογικό τρόπο. Με + συμβολίζουμε την πρόσθεση και με . (τελεία) τον πολλαπλασιασμό.

ΠΡΑΞΗ	ΠΕΡΙΓΡΑΦΗ
A+B	Προσθέτει δυο Μήτρες ή δύο Διανύσματα.
A.B	Πολλαπλασιάζει Μήτρες ή Διανύσματα.
λ*A A*λ	Βαθμωτός πολλαπλασιασμός αριθμού με Μήτρα ή διάνυσμα.
A^n	Υπολογίζει δυνάμεις Μητρών.

*Πίνακας 8. Πράξεις με μήτρες.*

### Παράδειγμα: 6-4

Ορίζουμε δύο διανύσματα v και u.

```
> v := <1, 2, 3>;
```

$$v := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

```
> u := <4, 5, 6>;
```

$$u := \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

Η πρόσθεσή τους είναι:

> v+u;

$$\begin{bmatrix} 5 \\ 7 \\ 9 \end{bmatrix}$$

Το γινόμενο τους είναι:

> v . u;

32

Το βαθμωτό γινόμενο είναι:

> 3\*v;

$$\begin{bmatrix} 3 \\ 6 \\ 9 \end{bmatrix}$$

### Παράδειγμα: 6-5

Στο παράδειγμα αυτό θα δούμε τις βασικές πράξεις σε δύο 2x2 πίνακες σε γενική

μορφή  $A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$ ,  $B = \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix}$ .

### Λύση:

Ορίζουμε δυο μήτρες 2x2, την A και B.

> A:=Matrix(2,2,[[a1,a2],[a3,a4]]);

$$A := \begin{bmatrix} a1 & a2 \\ a3 & a4 \end{bmatrix}$$

> B:=Matrix(2,2,[[b1,b2],[b3,b4]]);

$$B := \begin{bmatrix} b1 & b2 \\ b3 & b4 \end{bmatrix}$$

Η πρόσθεσή τους θα είναι:

> A+B;

$$\begin{bmatrix} a1 + b1 & a2 + b2 \\ a3 + b3 & a4 + b4 \end{bmatrix}$$

Το γινόμενο τους:

> A.B;

$$\begin{bmatrix} a1 b1 + a2 b3 & a1 b2 + a2 b4 \\ a3 b1 + a4 b3 & a3 b2 + a4 b4 \end{bmatrix}$$

Αριθμός k επί A:

> k\*A;

$$\begin{bmatrix} k a1 & k a2 \\ k a3 & k a4 \end{bmatrix}$$

Ο γραμμικός συνδυασμός  $k \cdot A + l \cdot B$  είναι:

> k\*A+l\*B;

$$\begin{bmatrix} k a1 + l b1 & k a2 + l b2 \\ k a3 + l b3 & k a4 + l b4 \end{bmatrix}$$

Ορίζουμε και μια μήτρα 2x3.

> C:=Matrix(2, 3, [[c1, c2, c3], [c4, c5, c6]]);

$$C := \begin{bmatrix} c1 & c2 & c3 \\ c4 & c5 & c6 \end{bmatrix}$$

Όπως είναι αναμενόμενο, η προσπάθεια να προσθέσουμε τον A και το C δίνει μήνυμα λάθους.

> A+C;

Error, (in rtable/Sum) invalid arguments

Μήνυμα λάθους βέβαια παίρνουμε και στην προσπάθεια πολλαπλασιασμού των C και A.

> C.A;

Error, (in LinearAlgebra:-MatrixMatrixMultiply) first matrix column dimension (3) <> second matrix row dimension (2)

Ο πολλαπλασιασμός των  $A$  και  $C$  ορίζεται:

> A.C;

$$\begin{bmatrix} a1 c1 + a2 c4 & a1 c2 + a2 c5 & a1 c3 + a2 c6 \\ a3 c1 + a4 c4 & a3 c2 + a4 c5 & a3 c3 + a4 c6 \end{bmatrix}$$

Η δύναμη  $A^2$  είναι:

> A^2;

$$\begin{bmatrix} a1^2 + a2 a3 & a1 a2 + a2 a4 \\ a3 a1 + a4 a3 & a2 a3 + a4^2 \end{bmatrix}$$

Ο υπολογισμός της δύναμης  $A^{-1}$ , που είναι ο αντίστροφος του  $A$ , γίνεται ως εξής:

> A^(-1);

$$\begin{bmatrix} -\frac{a4}{-a1 a4 + a2 a3} & \frac{a2}{-a1 a4 + a2 a3} \\ \frac{a3}{-a1 a4 + a2 a3} & -\frac{a1}{-a1 a4 + a2 a3} \end{bmatrix}$$

□

## 6.4 Πακέτο Εντολών LinearAlgebra

Για τη χρήση Μητρών και Διανυσμάτων στη Γραμμική Άλγεβρα έχει αναπτυχθεί ένα πακέτο εντολών που μας δίνει τη δυνατότητα να κάνουμε εύκολα πολύπλοκες διαδικασίες με Μήτρες και Διανύσματα. Στην ενότητα αυτή θα παρουσιάσουμε το πακέτο Γραμμικής Άλγεβρας *LinearAlgebra*.

Για να ενεργοποιήσουμε τις εντολές του πακέτου πληκτρολογούμε:

> with(LinearAlgebra);

Εδώ πρέπει να σημειώσουμε ότι όλες οι εντολές του πακέτου ξεκινάνε με κεφαλαίο γράμμα, για να μην υπάρχει σύγχυση με εντολές του πακέτου `linalg`, που, όπως έχουμε αναφέρει στην Παρατήρηση 6.2, είναι η προηγούμενη υλοποίηση των εντολών της Γραμμικής Άλγεβρας.

Για ό,τι ακολουθεί στο κεφάλαιο αυτό θεωρούμε ότι έχουμε «ενεργοποιήσει» το πακέτο με την εντολή:

>with(LinearAlgebra) :

Για την πρόσθεση και πολλαπλασιασμό έχουμε εκτός από τους τελεστές +,· που αναφέραμε νωρίτερα και τις εντολές:

<b>Περιγραφή:</b>	<b>Προσθέτει, Πολλαπλασιάζει Μήτρες-Διανύσματα.</b>	
<b>Γενική σύνταξη:</b>	<i>MatrixAdd(A,B) ή Add(A,B)</i> <i>MatrixMatrixMultiply(A,B) ή Multiply(A,B)</i>	
<b>Παράμετροι:</b>	<i>A,B</i>	<i>Μήτρες, Διανύσματα</i>

### Παράδειγμα: 6-6

Δίνονται οι μήτρες  $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$  και  $B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$ . Να υπολογιστούν οι  $A+B$ ,  $A \cdot B$ ,

$3A$ .

#### Λύση:

Ορίζουμε τις μήτρες χρησιμοποιώντας την εντολή Matrix.

> A:=Matrix(2,2,[[1,2],[3,4]]);

$$A := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

> B:=Matrix(2,2,[[5,6],[7,8]]);

$$B := \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

Το άθροισμα των A και B είναι:

> Add(A,B);

$$\begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$$

Το γινόμενο των A και B είναι:

> Multiply(A,B);

$$\begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

Το βαθμωτό γινόμενο  $3A$  είναι:

> Multiply(3,A);

$$\begin{bmatrix} 3 & 6 \\ 9 & 12 \end{bmatrix}$$

□

Ακολουθεί ένας πίνακας με τις ειδικές μήτρες και τα διανύσματα που μπορούμε να χρησιμοποιήσουμε σε αυτό το πακέτο.

ΕΝΤΟΛΗ	ΠΕΡΙΓΡΑΦΗ
<i>ZeroMatrix(n,m )</i> ;	Δημιουργεί μια μηδενική μήτρα.
<i>IdentityMatrix( διάσταση )</i> ;	Δημιουργεί μια ταυτοτική μήτρα.
<i>HilbertMatrix( n,m,expr )</i> ;	Δημιουργεί τη μήτρα του Hilbert.
<i>HankelMatrix( L,n)</i> ;	Δημιουργεί τη μήτρα του Hankel.
<i>BezoutMatrix( p(x), q(x), x)</i> ;	Δημιουργεί τη μήτρα του Bazout.
<i>HouseholterMatrix</i>	Δημιουργεί τη μήτρα του Householder.
<i>JordanBlockMatrix</i>	Δημιουργεί τη μήτρα του Jordan.
<i>SylvesterMatrix</i>	Δημιουργεί τη μήτρα του Sylvester.
<i>ToeplitzMatrix</i>	Δημιουργεί τη μήτρα του Toeplitz.
<i>VandermondeMatrix</i>	Δημιουργεί τη μήτρα του Vandermonde.
<i>CompanionMatrix( πολυώνυμο)</i>	Δημιουργεί το συνοδό πίνακα ενός πολυωνύμου.
<i>GivensRotation( διάνυσμα)</i> ;	Κατασκευάζει τη μήτρα για δοσμένη περιστροφή.
<i>UnitVector</i>	Μοναδιαίο Διάνυσμα
<i>RandomMatrix(n,m)</i> ;	Δημιουργεί μια μήτρα με τυχαίους αριθμούς.
<i>RandomVector (αριθμός στοιχείων)</i> ;	Δημιουργεί ένα διάνυσμα με τυχαίους αριθμούς.

*Πίνακας 9. Ειδικές Μήτρες και Διανύσματα.*

**Παράδειγμα: 6-7**

Παραθέτουμε παραδείγματα ειδικών μητρών και διανυσμάτων.

✚ Μηδενική τετραγωνική Μήτρα τάξης 2.

> ZeroMatrix(2);

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

✚ Μηδενική Μήτρα 2x3.

> ZeroMatrix(2, 3);

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

✚ Ταυτοτική Μήτρα τάξης 3.

> IdentityMatrix(3);

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

✚ Μήτρα Hilbert τάξεως 3.

> HilbertMatrix(3);

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}$$

✚ Η 2x3 μήτρα Hilbert του x είναι :

> HilbertMatrix(2, 3, x);

$$\begin{bmatrix} \frac{1}{2-x} & \frac{1}{3-x} & \frac{1}{4-x} \\ \frac{1}{3-x} & \frac{1}{4-x} & \frac{1}{5-x} \end{bmatrix}$$



✚ Η Μήτρα Bezout των πολυωνύμων  $ax^2 + bx + c$  και  $cx + d$  είναι:

> BezoutMatrix(a\*x^2+b\*x+c, c\*x+d, x);

$$\begin{bmatrix} d & a-bc & -c^2 \\ & c & d \end{bmatrix}$$

✚ Μοναδιαίο διάνυσμα.

> UnitVector(2, 3);

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

✚ Τυχαία μήτρα 3x3.

> RandomMatrix(3, 3);

$$\begin{bmatrix} -21 & -50 & -79 \\ -56 & 30 & -71 \\ -8 & 62 & 28 \end{bmatrix}$$

✚ Τυχαίο διάνυσμα διάστασης 2.

> RandomVector(2);

$$\begin{bmatrix} -62 \\ -90 \end{bmatrix}$$

✚ Περιστροφή του διανύσματος V κατά 1,3:

> V := <1, 2, 3>;

$$V := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

> GivensRotationMatrix(V, 1, 3);

$$\begin{bmatrix} \frac{\sqrt{10}}{10} & 0 & \frac{3\sqrt{10}}{10} \\ 0 & 1 & 0 \\ -\frac{3\sqrt{10}}{10} & 0 & \frac{\sqrt{10}}{10} \end{bmatrix}$$

✚ Η συνοδός μήτρα του πολυωνύμου  $2x^2 - x + 1$  είναι:

> CompanionMatrix(2\*x^2-x+1);

$$\begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

ΕΝΤΟΛΗ	ΠΕΡΙΓΡΑΦΗ
<b>Determinant(Μήτρα);</b>	Βρίσκει την ορίζουσα μιας Μήτρας.
<b>Adjoint( Μήτρα);</b>	Βρίσκει τη συμπληρωματική μήτρα μιας μήτρας.
<b>Minor(Μήτρα, n,m);</b>	Υπολογίζει την ελάχιστη ορίζουσα ως προς τη n γραμμή και m στήλη.
<b>Rank( Μήτρα);</b>	Βρίσκει την τάξη μιας μήτρας
<b>ConditionNumber( Μήτρα);</b>	Υπολογίζει τον «condition number» μιας μήτρας : $\ A^{-1}\  \cdot \ A\ $
<b>MatrixInverse( Μήτρα);</b>	Βρίσκει την αντίστροφη μήτρα.
<b>NullSpace( Μήτρα);</b>	Υπολογίζει μια βάση του πυρήνα μιας μήτρας.
<b>Permanent (Τετραγωνική Μήτρα);</b>	Υπολογίζει τον «permanent» μια τετραγωνικής μήτρας.
<b>SingularValues( Μήτρα);</b>	Υπολογίζει τις χαρακτηριστικές τιμές μιας Μήτρας.
<b>Trace( Μήτρα);</b>	Βρίσκει το Ίχνος μιας τετραγωνικής Μήτρας.
<b>Transpose( Μήτρα ή Διάνυσμα);</b>	Επιστρέφει την ανάστροφη Μήτρα.
<b>HermitianTranspose( Μήτρα ή Διάνυσμα);</b>	Υπολογίζει την ερμητιανή ανάστροφη μήτρα μιας μήτρας ή διανύσματος.

*Πίνακας 10. Βασικές Συναρτήσεις του Πακέτου LinearAlgebra.*

ΕΝΤΟΛΗ	ΠΕΡΙΓΡΑΦΗ
<i>CharacteristicMatrix (Τετραγωνική Μήτρα, Μεταβλητή);</i>	Βρίσκει τη χαρακτηριστική Μήτρα μιας Μήτρας
<i>CharacteristicPolynomial(Μήτρα, Μεταβλητή);</i>	Βρίσκει το χαρακτηριστικό πολυώνυμο μιας Μήτρας.
<i>Eigenvalues( Μήτρα);</i>	Βρίσκει τις ιδιοτιμές μιας μήτρας.
<i>Eigenvectors ( Μήτρα);</i>	Υπολογίζει τις ιδιοτιμές και τα ιδιοδιανύσματα μιας μήτρας.
<i>MinimalPolynomial( Μήτρα, Μεταβλητή);</i>	Βρίσκει το ελάχιστο πολυώνυμο μιας μήτρας.
<i>Basis(Μήτρα);</i>	Υπολογίζει μια βάση του χώρου των διανυσμάτων.
<i>ColumnSpace(Μήτρα);</i>	Υπολογίζει μια βάση του χώρου των στηλών.
<i>RowSpace(Μήτρα);</i>	Υπολογίζει μια βάση του χώρου των γραμμών.
<i>NullSpace(Μήτρα);</i>	Υπολογίζει μια βάση του πυρήνα μιας μήτρας.
<i>MatrixNorm(Μήτρα);</i>	Υπολογίζει την p-νόρμα.

Πίνακας 11. Βασικές Συναρτήσεις του Πακέτου *LinearAlgebra*.**Παράδειγμα: 6-8**

Δίνεται η μήτρα  $A = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$ . Να βρεθεί το ίχνος και η ορίζουσά της.

**Λύση:**

Ορίζουμε τη μήτρα κατά στήλες:

$$\gg A := \langle\langle a, b, c \rangle \mid \langle d, e, f \rangle \mid \langle g, h, i \rangle \rangle;$$

$$A := \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

Το ίχνος της μήτρας A είναι:

> Trace (A) ;

$$a + e + i$$

Η ορίζουσα της A είναι:

> Determinant (A) ;

$$a e i - a h f + b f g - b d i + c d h - c e g$$

Οι ελάσσονες ορίζουσες είναι:

> Minor (A, 3, 3) ;

$$a e - b d$$

> Minor (A, 2, 3) ;

$$a h - b g$$

> Minor (A, 1, 3) ;

$$d h - e g$$

□

Ακολουθούν κάποιες ακόμα εντολές για την επεξεργασία μιας μήτρας.

ΕΝΤΟΛΗ	ΠΕΡΙΓΡΑΦΗ
<b>Column(Μήτρα, n);</b>	Επιστρέφει τη στήλη μίας Μήτρας.
<b>ColumnDimension(Μήτρα);</b>	Υπολογίζει τη διάσταση στηλών μιας Μήτρας.
<b>ColumnOperation(Μήτρα,K);</b>	Εφαρμόζει μια στοιχειώδη πράξη στηλών.
<b>DeleteColumn(Μήτρα,L);</b>	Σβήνει μια στήλη σε μια Μήτρα.
<b>DeleteRow(Μήτρα,L);</b>	Σβήνει μια γραμμή σε μια Μήτρα.
<b>Dimension(Μήτρα);</b>	Υπολογίζει τη διάσταση μιας Μήτρας.
<b>Row(Μήτρα, n);</b>	Επιστρέφει γραμμή μίας Μήτρας.
<b>RowDimension(Μήτρα);</b>	Υπολογίζει τη διάσταση γραμμών μιας

	Μήτρας.
<i>RowOperation(Μήτρα,K);</i>	Εφαρμόζει μια στοιχειώδη πράξη γραμμών.
<i>SubMatrix(Μήτρα,r,c);</i>	Δημιουργεί μια υπο – μήτρα.
<i>SubVector(Μήτρα,i);</i>	Δημιουργεί ένα υπο διάνυσμα.

Πίνακας 12. Τελεστές Μητρών

**Παράδειγμα: 6-9**

Ορίζουμε την μήτρα  $A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$ . Θα εφαρμόσουμε τους τελεστές του παραπάνω πίνακα σε αυτήν.

**Λύση:**

Ορίζουμε τη μήτρα.

```
> A := <<a|b|c>, <d|e|f>, <g|h|i>>;
```

$$A := \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Εξάγουμε τη 2<sup>η</sup> στήλη της A.

```
> Column(A, 2);
```

$$\begin{bmatrix} b \\ e \\ h \end{bmatrix}$$

Η διάσταση στηλών της A είναι:

```
> ColumnDimension(A);
```

3

```
> ColumnOperation(A, 2);
```

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Σβήνουμε τη δεύτερη στήλη της.

```
> DeleteColumn(A, 2);
```

$$\begin{bmatrix} a & c \\ d & f \\ g & i \end{bmatrix}$$

Σβήνουμε τη δεύτερη γραμμή της.

```
> DeleteRow(A, 2);
```

$$\begin{bmatrix} a & b & c \\ g & h & i \end{bmatrix}$$

Η διάσταση της μήτρας A είναι:

```
> Dimension(A);
```

3, 3

Η τρίτη γραμμή της είναι:

```
> Row(A, 3);
```

[g, h, i]

Η διάσταση γραμμών της είναι:

```
> RowDimension(A);
```

3

```
> RowOperation(A, 2);
```

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Ο υποπίνακας που δημιουργείται από την πρώτη γραμμή και δεύτερη στήλη είναι:.

```
> SubMatrix(A, 1, 2);
```

[ b ]

□

ΕΝΤΟΛΗ	ΠΕΡΙΓΡΑΦΗ
<i>CrossProduct(v,u);</i>	Υπολογίζει το εξωτερικό γινόμενο δύο διανυσμάτων.
<i>DotProduct(v,u);</i>	Υπολογίζει το εσωτερικό γινόμενο δύο διανυσμάτων.
<i>GramSchmidt(Διανύσματα);</i>	Υπολογίζει ένα ορθογώνιο σύνολο διανυσμάτων.
<i>IntersectonBasis(Μήτρα);</i>	return a basis for the intersection of vector space(s)
<i>Norm</i>	Υπολογίζει την p-νόρμα ενός διανύσματος ή μήτρας.
<i>Normalize(Διάνυσμα);</i>	Κανονικοποιεί ένα διάνυσμα.
<i>SumBasis(Μήτρα);</i>	Υπολογίζει μια βάση για ευθύ άθροισμα δυο υποχώρων
<i>VectorAngle(v,u);</i>	Υπολογίζει τη γωνία ανάμεσα σε δύο διανύσματα v,u της ίδιας διάστασης.
<i>VectorNorm</i>	Υπολογίζει την p-νόρμα ενός διανύσματος.

Πίνακας 13. Τελεστές Διανυσμάτων.

**Παράδειγμα: 6-10**

Ορίζουμε δύο διανύσματα  $v = (x_1, y_1, z_1)$  και  $u = (x_2, y_2, z_2)$  ως εξής:

> v := <x1, y1, z1>;

$$v := \begin{bmatrix} x1 \\ y1 \\ z1 \end{bmatrix}$$

> u := <x2, y2, z2>;

$$u := \begin{bmatrix} x2 \\ y2 \\ z2 \end{bmatrix}$$

Η γωνία που σχηματίζουν είναι:

> VectorAngle (v, u) ;

$$\arccos\left(\frac{\overline{x1} x2 + \overline{y1} y2 + \overline{z1} z2}{\sqrt{|x1|^2 + |y1|^2 + |z1|^2} \sqrt{|x2|^2 + |y2|^2 + |z2|^2}}\right)$$

Οι νόρμες είναι:

> VectorNorm (v) ;

$$\max(|x1|, |y1|, |z1|)$$

Η 1- νόρμα του v είναι:

> VectorNorm (v, 1) ;

$$|x1| + |y1| + |z1|$$

Η 2- νόρμα του v είναι:

> VectorNorm (v, 2) ;

$$\sqrt{|x1|^2 + |y1|^2 + |z1|^2}$$

Το εξωτερικό του γινόμενο είναι:

> CrossProduct (v, u) ;

$$\begin{bmatrix} y1 z2 - z1 y2 \\ z1 x2 - x1 z2 \\ x1 y2 - y1 x2 \end{bmatrix}$$

Το εσωτερικό τους γινόμενο είναι:

> DotProduct (v, u) ;

$$\overline{x1} x2 + \overline{y1} y2 + \overline{z1} z2$$

Θα δημιουργήσουμε ένα ορθοκανονικό σύνολο από αυτά τα διανύσματα.

> GramSchmidt ({v, u}) ;



$$\left\{ \begin{array}{l} x2 \\ y2 \\ z2 \end{array} \right\}, \left\{ \begin{array}{l} -\frac{(x1 \bar{x2} + y1 \bar{y2} + z1 \bar{z2}) x2}{x2 \bar{x2} + y2 \bar{y2} + z2 \bar{z2}} + x1 \\ -\frac{(x1 \bar{x2} + y1 \bar{y2} + z1 \bar{z2}) y2}{x2 \bar{x2} + y2 \bar{y2} + z2 \bar{z2}} + y1 \\ -\frac{(x1 \bar{x2} + y1 \bar{y2} + z1 \bar{z2}) z2}{x2 \bar{x2} + y2 \bar{y2} + z2 \bar{z2}} + z1 \end{array} \right\}$$

ΕΝΤΟΛΗ	ΠΕΡΙΓΡΑΦΗ
<i>BidiagonalForm(Μήτρα);</i>	Ανάγει μια μήτρα στην bidiagonal μορφή της.
<i>FrobeniusForm(Μήτρα);</i>	Ανάγει μια μήτρα στην Frobenius μορφή της.
<i>GaussianElimination(Μήτρα);</i>	Εφαρμόζει «Gaussian elimination» σε μια μήτρα
<i>HermiteForm(Μήτρα);</i>	Υπολογίζει την Hermite κανονική μορφή μιας μήτρας.
<i>HessenbergForm(Μήτρα);</i>	Ανάγει μια τετραγωνική μήτρα στην Hessenberg μορφή της
<i>JordanForm(Μήτρα);</i>	Ανάγει μια μήτρα στη μορφή Jordan.
<i>LUdecomposition(Μήτρα);</i>	Υπολογίζει την Cholesky, PLU ή PLU1R ανάλυση μιας μήτρας.
<i>PopovForm(Μήτρα, Μεταβλητή);</i>	Υπολογίζει την Popov κανονική μορφή μιας μήτρας.
<i>QRdecomposition(Μήτρα);</i>	Παραγοντοποιεί μια μήτρα.
<i>ReducedRowEchelonForm(Μήτρα);</i>	Εφαρμόζει την Gauss-Jordan μορφή μιας μήτρας.
<i>SchurForm(Τετραγωνική Μήτρα);</i>	Ανάγει μια τετραγωνική μήτρα στην Schur μορφή της.
<i>SmithForm(Μήτρα);</i>	Ανάγει μια μήτρα στην Smith κανονική μορφή της.
<i>TridiagonalForm(Τετραγωνική Μήτρα);</i>	Ανάγει μια τετραγωνική μήτρα σε τριγωνική μορφή.

Πίνακας 14. Ειδικές Μορφές Μητρών του πακέτου *LinearAlgebra*.

**Παράδειγμα: 6-11**

```
> A := RandomMatrix(3, generator=-5..5);
```

$$A := \begin{bmatrix} 4 & -1 & -5 \\ 1 & 4 & 1 \\ -5 & -5 & 0 \end{bmatrix}$$

```
> BidiagonalForm(A);
```

$$\begin{bmatrix} -6.48074069840786038, 4.84522347020139766, 0. \\ 0., 1.99859894903996182, -5.06236984847990356 \\ 0., 0., -3.86028797914189780 \end{bmatrix}$$

```
> FrobeniusForm(A);
```

$$\begin{bmatrix} 0 & 0 & -50 \\ 1 & 0 & 3 \\ 0 & 1 & 8 \end{bmatrix}$$

```
> GaussianElimination(A);
```

$$\begin{bmatrix} 4 & -1 & -5 \\ 0 & \frac{17}{4} & \frac{9}{4} \\ 0 & 0 & \frac{-50}{17} \end{bmatrix}$$

```
> HermiteForm(A);
```

$$\begin{bmatrix} 1 & 0 & 43 \\ 0 & 1 & 27 \\ 0 & 0 & 50 \end{bmatrix}$$

```
> HessenbergForm(A);
```

$$\begin{bmatrix} -2., -0.444089209850062616 \cdot 10^{-15}, -4.47213595499957872 \\ 2.23606797749978981, 4., -0.99999999999999956 \\ 0., -2.99999999999999912, 1.99999999999999934 \end{bmatrix}$$

> LUdecomposition(A) ;

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ -1 & \frac{-5}{6} & 1 \end{bmatrix}, \begin{bmatrix} -2 & -4 & -2 \\ 0 & 6 & -2 \\ 0 & 0 & \frac{-5}{3} \end{bmatrix}$$

> QRdecomposition(A) ;

$$\begin{bmatrix} \frac{-2}{3} & \frac{32\sqrt{293}}{879} & \frac{-7\sqrt{293}}{293} \\ \frac{-1}{3} & \frac{38\sqrt{293}}{879} & \frac{-10\sqrt{293}}{293} \\ \frac{2}{3} & \frac{-13\sqrt{293}}{879} & \frac{-12\sqrt{293}}{293} \end{bmatrix}, \begin{bmatrix} 3 & \frac{2}{3} & \frac{11}{3} \\ 0 & \frac{\sqrt{293}}{3} & \frac{-76\sqrt{293}}{879} \\ 0 & 0 & \frac{20\sqrt{293}}{293} \end{bmatrix}$$

> ReducedRowEchelonForm(A) ;

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

> SchurForm(A) ;

[-0.900504906720115628 , -4.76508454900225686 ,  
0.676772452212909714]  
[0.553351464748722432 , -0.900504906720115628 ,  
-0.506179232646233324]  
[0. , 0. , 5.80100981344022681]

> SmithForm(A) ;

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 20 \end{bmatrix}$$

## 6.5 Λυμένες Ασκήσεις

Στις παρακάτω ασκήσεις έχουμε ενεργοποιήσει το πακέτο LinearAlgebra.

### Άσκηση: 6-1

Να βρεθεί ο βαθμός της μήτρας  $A = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 4 & 4 \\ 9 & 7 & 18 \end{pmatrix}$ .

#### Λύση:

Ορίζουμε τη μήτρα  $A$  κατά γραμμές.

> A := <<1|3|2>, <2|4|4>, <9|7|18>>;

$$A := \begin{bmatrix} 1 & 3 & 2 \\ 2 & 4 & 4 \\ 9 & 7 & 18 \end{bmatrix}$$

Ο βαθμός της μήτρας είναι:

> Rank(A);

2

Πράγματι, αν ζητήσουμε τους γραμμικούς υποχώρους των χώρων των γραμμών και των στηλών, βλέπουμε ότι η διάστασή τους είναι 2.

> RowSpace(A);

$$[[1, 0, 2], [0, 1, 0]]$$

> ColumnSpace(A);

$$\left[ \begin{bmatrix} 1 \\ 0 \\ -11 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 10 \end{bmatrix} \right]$$

□

### Άσκηση: 6-2

Να βρεθεί η αντίστροφη μήτρα, η ορίζουσα και η τάξη της μήτρας  $\begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}$ .

**Λύση:**

Ορίζουμε τη μήτρα κατά στήλες:

>A :=< <1, 3> | <2, 5> > ;

$$A := \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}$$

Η ορίζουσα είναι:

>Determinant (A) ;

-1

Ο αντίστροφος είναι:

> MatrixInverse (A) ;

$$\begin{bmatrix} -5 & 2 \\ 3 & -1 \end{bmatrix}$$

Η τάξης της είναι:

> Rank (A) ;

2

□

**Άσκηση: 6-3**

Να βρεθεί η αντίστροφη της μήτρας  $A = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 2 \\ 3 & 2 & 4 \end{pmatrix}$ .

**Λύση:**

Θα προσδιορίσουμε την αντίστροφη της μήτρας A, δημιουργώντας τη διαμερισμένη μήτρα  $(A|I_n)$  και στη συνέχεια μετατρέποντας τη μήτρα αυτή στη μορφή Gauss-Jordan.

Ορίζουμε τη μήτρα A κατά γραμμές.

> A :=<<1 | 1 | 1>, <2 | 1 | 2>, <3 | 2 | 2>>;

$$A := \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 2 \\ 3 & 2 & 2 \end{bmatrix}$$

Δημιουργούμε τη μήτρα B, η οποία είναι η διαμερισμένη  $(A|I_n)$ .

> B := <A | IdentityMatrix(3)>;

$$B := \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 2 & 0 & 1 & 0 \\ 3 & 2 & 2 & 0 & 0 & 1 \end{bmatrix}$$

Ζητάμε την Gauss-Jordan μορφή της B.

> ReducedRowEchelonForm(B);

$$\begin{bmatrix} 1 & 0 & 0 & -2 & 0 & 1 \\ 0 & 1 & 0 & 2 & -1 & 0 \\ 0 & 0 & 1 & 1 & 1 & -1 \end{bmatrix}$$

Στο δεξιό μέρος της B είναι η αντίστροφη μήτρα.

Το ίδιο αποτέλεσμα βέβαια θα έχουμε αν χρησιμοποιήσουμε τη γνωστή από την προηγούμενη άσκηση εντολή MatrixInverse.

> MatrixInverse(A);

$$\begin{bmatrix} -2 & 0 & 1 \\ 2 & -1 & 0 \\ 1 & 1 & -1 \end{bmatrix}$$

□

#### Άσκηση: 6-4

Να βρεθεί ο αντίστροφος της μήτρας  $A = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \\ 9 & 7 & 0 \end{pmatrix}$ .

#### Λύση:

Σε αυτή την άσκηση θα υπολογίσουμε την αντίστροφη μήτρα εφαρμόζοντας τον τύπο

$$A^{-1} = \frac{adjA}{\det(A)} \text{ όπου } adjA \text{ είναι η συμπληρωματική μήτρα } adjA = \begin{bmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \end{bmatrix}$$

όπου  $A_{ij}$ ,  $i, j = 1, 2, 3$  τα αλγεβρικά συμπληρώματα της μήτρας  $A$ .

Έτσι, αρχικά ορίζουμε τη μήτρα :

> A := <<1 | 3 | 5>, <2 | 4 | 6>, <9 | 7 | 0>>;

$$A := \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \\ 9 & 7 & 0 \end{bmatrix}$$

Υπολογίζουμε την ορίζουσα

> Determinant (A) ;

10

η οποία είναι διάφορη του μηδενός. Έτσι προχωράμε στον υπολογισμό του συμπληρωματικού.

> adj := <<Minor (A, 1, 1) | Minor (A, 2, 1) | Minor (A, 3, 1)>, <Minor (A, 1, 2) | Minor (A, 2, 2) | Minor (A, 3, 2)>, <Minor (A, 1, 3) | Minor (A, 2, 3) | Minor (A, 3, 3)>>;

$$adj := \begin{bmatrix} -42 & -35 & -2 \\ -54 & -45 & -4 \\ -22 & -20 & -2 \end{bmatrix}$$

> antristrofos := adj \* 1 / Determinant (A) ;

$$antristrofos := \begin{bmatrix} \frac{-21}{5} & \frac{-7}{2} & \frac{-1}{5} \\ \frac{-27}{5} & \frac{-9}{2} & \frac{-2}{5} \\ \frac{-11}{5} & -2 & \frac{-1}{5} \end{bmatrix}$$

□

**Άσκηση: 6-5**

Δίνεται  $A = \begin{bmatrix} 4 & -5 \\ 2 & -3 \end{bmatrix}$ . Να βρεθούν οι ιδιοτιμές και τα ιδιοδιανύσματα της  $A$ .

**Λύση:**

Ορίζουμε την  $A$  κατά στήλες.

```
> A := <<4, 2> | <-5, -3>>;
```

$$A := \begin{bmatrix} 4 & -5 \\ 2 & -3 \end{bmatrix}$$

Οι ιδιοτιμές είναι:

```
> Eigenvalues (A) ;
```

$$\begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

Με την εντολή `Eigenvectors` παίρνουμε τις ιδιοτιμές και τα αντίστοιχα ιδιοδιανύσματα :

```
> l, x := Eigenvectors (A) ;
```

$$l, x := \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & \frac{5}{2} \\ 1 & 1 \end{bmatrix}$$

Το πρώτο ιδιοδιάνυσμα μπορούμε να το απομονώσουμε ως εξής:

```
> x1 := Column (x, 1) ;
```

$$x1 := \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Ενώ, το δεύτερο ιδιοδιάνυσμα είναι:

```
> x2 := Column (x, 2) ;
```

$$x2 := \begin{bmatrix} \frac{5}{2} \\ 1 \end{bmatrix}$$

□



**Άσκηση: 6-6**

Δίνεται η μήτρα  $A = \begin{pmatrix} 1 & 1 & 2 \\ 4 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ . Να βρεθεί το χαρακτηριστικό πολυώνυμο της A

και στη συνέχεια να επαληθευτεί το θεώρημα Cayley-Hamilton.

**Λύση:**

Ορίζουμε την A κατά στήλες.

> A := <<1 | 1 | 2>, <4 | 1 | 0>, <0 | 0 | 0>>;

$$A := \begin{bmatrix} 1 & 1 & 2 \\ 4 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Η χαρακτηριστική μήτρα είναι:

> CharacteristicMatrix(A, lambda);

$$\begin{bmatrix} \lambda - 1 & -1 & -2 \\ -4 & \lambda - 1 & 0 \\ 0 & 0 & \lambda \end{bmatrix}$$

Έτσι, το χαρακτηριστικό πολυώνυμο είναι η ορίζουσα της παραπάνω μήτρας.

> Determinant(%);

$$\lambda^3 - 2\lambda^2 - 3\lambda$$

Στο ίδιο αποτέλεσμα μπορούμε να οδηγηθούμε κατευθείαν χρησιμοποιώντας την εντολή CharacteristicPolynomial. Έτσι, ορίζουμε το χαρακτηριστικό πολυώνυμο:

> xarakteristikh := CharacteristicPolynomial(A, lambda);

$$xarakteristikh := \lambda^3 - 2\lambda^2 - 3\lambda$$

Αν αντικαταστήσουμε στο παραπάνω πολυώνυμο όπου λ τη μήτρα A, θα έχουμε:

> subs(lambda=A, xarakteristikh);

$$\begin{bmatrix} 1 & 1 & 2 \\ 4 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}^3 - 2 \begin{bmatrix} 1 & 1 & 2 \\ 4 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}^2 - 3 \begin{bmatrix} 1 & 1 & 2 \\ 4 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Ζητάμε να γίνουν οι πράξεις.

```
> eval(%);
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Και έτσι, πράγματι, έχουμε ότι  $p_n(A) = O$ , με  $p_n(\lambda)$  το χαρακτηριστικό πολυώνυμο της  $A$ .

□

### Άσκηση: 6-7

Να λυθεί το σύστημα  $\begin{cases} x+3y=-3 \\ 2x+5y=2 \end{cases}$  με χρήση μητρών μέσω εύρεσης της αντίστροφης μήτρας.

**Λύση:**

```
> restart;
```

```
with(LinearAlgebra):
```

Ορίζουμε τη μήτρα του συστήματος  $A$ .

```
> A := <<1, 2>|<3, 5> >;
```

$$A := \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}$$

Ορίζουμε, επίσης, τη μήτρα των σταθερών  $B$ .

```
> B := <-3, 2>;
```

$$B := \begin{bmatrix} -3 \\ 2 \end{bmatrix}$$

Υπολογίζουμε την αντίστροφη μήτρα της  $A$ .

```
> invA := MatrixInverse(A);
```

$$\text{inv}A := \begin{bmatrix} -5 & 3 \\ 2 & -1 \end{bmatrix}$$

Όπως είναι γνωστό, η λύση του συστήματος είναι  $X = A^{-1} \cdot B$ . Έτσι, έχουμε:

> X:=invA.B;

$$X := \begin{bmatrix} 21 \\ -8 \end{bmatrix}$$

□

### Άσκηση: 6-8

Να λυθεί το σύστημα  $\begin{cases} x+3y=-3 \\ 2x+5y=2 \end{cases}$  με την μέθοδο του Grammer.

#### Λύση:

> restart;

with(LinearAlgebra):

Ορίζουμε τη μήτρα A του συστήματος,

> A := <<1, 2>|<3, 5> >;

$$A := \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}$$

Και τη μήτρα B των σταθερών του συστήματος.

> B := <<-3, 2>;

$$B := \begin{bmatrix} -3 \\ 2 \end{bmatrix}$$

Η μήτρα ως προς x είναι:

> Dx := <B|Column(A, 2)>;

$$Dx := \begin{bmatrix} -3 & 3 \\ 2 & 5 \end{bmatrix}$$

Η μήτρα ως προς y είναι:

> Dy := <Column(A, 1)|B>;

$$Dy := \begin{bmatrix} 1 & -3 \\ 2 & 2 \end{bmatrix}$$

> Determinant (Dx) ;

-21

> Determinant (Dy) ;

8

Η λύση του συστήματος είναι  $x = \frac{D_x}{D}$ ,  $y = \frac{D_y}{D}$ . Έτσι, έχουμε:

> x := Determinant (Dx) / Determinant (A) ;

x := 21

> y := Determinant (Dy) / Determinant (A) ;

y := -8

□

### Άσκηση: 6-9

Να λυθεί το σύστημα  $\begin{cases} x + 3y = -3 \\ 2x + 5y = 2 \end{cases}$  με τη μέθοδο του απαλοιφής του Gauss.

#### Λύση:

> restart;

with (LinearAlgebra) :

Ορίζουμε τη μήτρα A του συστήματος,

> A := <<1, 2> | <3, 5> >;

$$A := \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}$$

Και τη μήτρα B των σταθερών του συστήματος.

> B := <-3, 2>;

$$B := \begin{bmatrix} -3 \\ 2 \end{bmatrix}$$

Η επαυξημένη μήτρα είναι:

> C := <A | B>;

$$C := \begin{bmatrix} 1 & 3 & -3 \\ 2 & 5 & 2 \end{bmatrix}$$

Μετασχηματίζουμε την επαυξημένη μήτρα C σε Gauss-Jordan μορφή.

> ReducedRowEchelonForm(C) ;

$$\begin{bmatrix} 1 & 0 & 21 \\ 0 & 1 & -8 \end{bmatrix}$$

Από την παραπάνω μήτρα συμπεραίνουμε ότι η λύση του συστήματος είναι:  
 $x = 21, y = -8$

□

### Άσκηση: 6-10

Να λυθεί το σύστημα  $\begin{cases} x - y + z = 3 \\ x + y + k \cdot z = 1 \\ x + k \cdot y + z = k \end{cases}$  για τις διάφορες τιμές τις παραμέτρους

$k \in \mathbb{R}$ . Σε κάθε περίπτωση να προσδιορίσετε το χώρο των λύσεων και να βρείτε μια βάση του.

#### Λύση:

> restart;

> with(LinearAlgebra) :

Ορίζουμε την επαυξημένη μήτρα του συστήματος.

> A := <<1 | -1 | 1 | 3>, <1 | 1 | k | 1>, <1 | k | 1 | k>>;

$$A := \begin{bmatrix} 1 & -1 & 1 & 3 \\ 1 & 1 & k & 1 \\ 1 & k & 1 & k \end{bmatrix}$$

Με την εντολή GaussianElimination παίρνουμε τη μορφή Gauss της επαυξημένης μήτρας.

> G := GaussianElimination(A) ;

$$G := \begin{bmatrix} 1 & -1 & 1 & 3 \\ 0 & 2 & k-1 & -2 \\ 0 & 0 & -\frac{(k+1)(k-1)}{2} & 2k-2 \end{bmatrix}$$

Βρίσκουμε για ποιες τιμές του  $k$  το στοιχείο  $(3,3)$  της παραπάνω μήτρας γίνεται μηδέν (την εντολή `solve` την εξετάζουμε αναλυτικά στο κεφάλαιο 7).

```
> sol:=solve(G[3,3]=0, k);
```

```
sol := 1, -1
```

Οι τιμές που προέκυψαν με τη χρήση της εντολής `solve` είναι το  $+1$  και το  $-1$ .

- Αν  $k \neq -1$  και  $k \neq 1$ , τότε  $\text{rank}(A) = \text{rank}(A_\varepsilon) = 3 = n = 3$  και το σύστημα έχει μια μοναδική λύση.

Αυτή θα την έχουμε, αν ζητήσουμε την Gauus-Jordan μορφή της μήτρας  $A$ .

```
> ReducedRowEchelonForm(A);
```

$$\begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & \frac{k-3}{k+1} \\ 0 & 0 & 1 & -\frac{4}{k+1} \end{bmatrix}$$

Άρα  $x=4$ ,  $y = \frac{k-3}{k+1}$  και  $z = -\frac{4}{k+1}$

- Αν  $k = -1$ , δηλαδή:

```
> k:=sol[2];
```

```
k := -1
```

Τότε, η επαυξημένη μήτρα είναι:

```
> G;
```

$$\begin{bmatrix} 1 & -1 & 1 & 3 \\ 0 & 2 & -2 & -2 \\ 0 & 0 & 0 & -4 \end{bmatrix}$$

Και το σύστημα είναι αδύνατο ( $0 \cdot z = 4$ ).

- Αν  $k = 1$ , δηλαδή:

```
> k:=sol[1];
```

```
k := 1
```

τότε, η επαυξημένη μήτρα είναι:

> G;

$$\begin{bmatrix} 1 & -1 & 1 & 3 \\ 0 & 2 & 0 & -2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

και το σύστημα έχει άπειρες λύσεις.

□

### Άσκηση: 6-11

Δίνεται  $\{x_1, x_2, x_3\}$  βάση του  $\mathbb{R}^3$  με  $x_1 = (1, 1, 1)$ ,  $x_2 = (0, 1, 1)$  και  $x_3 = (1, 0, 1)$ . Να βρεθεί μια ορθοκανονική βάση.

**Λύση:**

> restart;

with(LinearAlgebra):

Ορίζουμε τα διανύσματα.

> v1:=<1, 1, 1>;

$$v1 := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

> v2:=<0, 1, 1>;

$$v2 := \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

> v3:=<1, 0, 1>;

$$v3 := \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Από τη Γραμμική Άλγεβρα έχουμε ότι, εάν  $\{x_1, x_2, x_3\}$  μια βάση ενός γραμμικού χώρου  $E$ , τότε η διαδικασία κατασκευής ορθοκανονικής βάσης είναι η εξής:

$$\text{Θέτουμε } y_1 = \frac{1}{\|x_1\|} \cdot x_1$$

$$\text{Θέτουμε } y_2 = \frac{z_2}{\|z_2\|} \text{ με } z_2 = x_2 - \langle x_2, y_1 \rangle y_1, \quad y_1, z_2 \text{ είναι ορθογώνια.}$$

$$\text{Θέτουμε } y_3 = \frac{z_3}{\|z_3\|} \text{ με } z_3 = x_3 - \langle x_3, y_1 \rangle y_1 - \langle x_3, y_2 \rangle y_2, \quad y_1, y_2, z_3 \text{ είναι ορθογώνια. Έτσι:}$$

> y1:=v1/Norm(v1,2);

$$y1 := \begin{bmatrix} \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} \end{bmatrix}$$

> z2:=v2-DotProduct(v2,y1)\*y1;

$$z2 := \begin{bmatrix} -\frac{2}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}$$

> y2:=z2/Norm(z2,2);

$$y2 := \begin{bmatrix} -\frac{\sqrt{6}}{3} \\ \frac{\sqrt{6}}{6} \\ \frac{\sqrt{6}}{6} \end{bmatrix}$$



```
> z3:=v3-DotProduct(v3,y2)*y2-DotProduct(v3,y1)*y1;
```

$$z3 := \begin{bmatrix} 0 \\ -\frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$$

```
> y3:=z3/Norm(z3,2);
```

$$y3 := \begin{bmatrix} 0 \\ -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix}$$

Το ίδιο αποτέλεσμα βέβαια θα έχουμε και αν χρησιμοποιήσουμε την εντολή GramSchmidt.

```
> GramSchmidt({v1,v2,v3},normalized);
```

$$\left\{ \begin{bmatrix} \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} \end{bmatrix}, \begin{bmatrix} -\frac{\sqrt{6}}{3} \\ \frac{\sqrt{6}}{6} \\ \frac{\sqrt{6}}{6} \end{bmatrix}, \begin{bmatrix} 0 \\ -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} \right\}$$

□



# Κεφάλαιο 7

## Λύση Εξισώσεων- Ανισώσεων

Στο κεφάλαιο αυτό θα παρουσιάσουμε τον τρόπο με τον οποίο επιλύουμε αλγεβρικές εξισώσεις και συστήματα. Με την εντολή `solve`, το Maple προσπαθεί να λύσει μια εξίσωση ή ένα σύστημα αναλυτικά. Επίσης, στις περιπτώσεις όπου η εντολή `solve` δεν μπορεί να δώσει αναλυτική λύση, θα δούμε την εντολή `fsolve` που δίνει προσεγγιστικές λύσεις μιας εξίσωσης ή ενός συστήματος. Στη συνέχεια, θα παρουσιάσουμε τον τρόπο με τον οποίο μπορούμε να λύνουμε ανισώσεις και συστήματα ανισώσεων καθώς και τον τρόπο γραφικής τους αναπαράστασης.

## 7.1 Ακριβής Επίλυση Αλγεβρικών Εξισώσεων-Συστημάτων

Για να υπολογίσουμε την ακριβή λύση μιας εξίσωσης ή ενός συστήματος εξισώσεων συμβολικά με χρήση αναλυτικών μεθόδων, το Maple έχει την εντολή `solve`.

Περιγραφή:	Λύνει μια εξίσωση ή ένα σύστημα εξισώσεων.	
Γενική σύνταξη:	<code>solve(eqns, vars)</code>	
Παράμετροι:	eqns	Εξίσωση ή σύστημα εξισώσεων {εξίσωση1, εξίσωση2,...}
	vars	Μεταβλητή ή μεταβλητές {μεταβλητή1, μεταβλητή2,...} ως προς τις οποίες θέλω τη λύση (αυτή η παράμετρος μπορεί πολλές φορές να παραλειφθεί).

### Παράδειγμα: 7-1

Να λυθεί η εξίσωση  $x^2 + 5x + 6 = 0$ .

#### Λύση:

Λύνουμε την εξίσωση ως προς  $x$  χρησιμοποιώντας την εντολή `solve`.

```
> solve ({x^2+5*x+6=0}, {x});
```

$$\{x = -2\}, \{x = -3\}$$

Το Maple μας επιστρέφει δύο λύσεις, το -2 και το -3.

□

Το Maple, όπως ήδη έχουμε δει, μπορεί να χειρισθεί αλγεβρικές εκφράσεις. Έτσι, αν του δώσουμε μια εξίσωση σε γενική μορφή και ζητήσουμε τη λύση της, θα μας επιστρέψει τον αναλυτικό τύπο της λύσης, εάν βέβαια αυτός υπάρχει.

### Παράδειγμα: 7-2

Να λυθεί η εξίσωση  $ax^2 + bx + c = 0$ .

#### Λύση:

Ορίζουμε την εξίσωση:

```
> eq := a*x^2+b*x+c=0;
```

$$eq := ax^2 + bx + c = 0$$

Λύνουμε την εξίσωση ως προς τη μεταβλητή  $x$ .

> solve (eq, {x}) ;

$$\left\{x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}\right\}, \left\{x = \frac{-b - \sqrt{b^2 - 4ac}}{2a}\right\}$$

Αν ως δεύτερη παράμετρο στην εντολή solve χρησιμοποιήσουμε το  $a$ , τότε θα λύσουμε την εξίσωση eq θεωρώντας ως άγνωστο το  $a$ .

> solve (eq, {a}) ;

$$\left\{a = -\frac{bx+c}{x^2}\right\}$$

□

### Παράδειγμα: 7-3

Να λυθεί η εξίσωση  $-x^3 - x^2 = 4$ .

#### Λύση:

Θα ονομάσουμε την εξίσωση με τη μεταβλητή eqn.

> eqn := -x^3 - x^2 = 4 ;

$$eqn := -x^3 - x^2 = 4$$

Λύνουμε την εξίσωση ως προς  $x$ .

> solve ({eqn}, {x}) ;

$$\left\{x = -2\right\}, \left\{x = \frac{1}{2} + \frac{1}{2}I\sqrt{7}\right\}, \left\{x = \frac{1}{2} - \frac{1}{2}I\sqrt{7}\right\}$$

Οι ρίζες δίνονται σε μορφή ακολουθίας συνόλων.

Εδώ έχουμε 3 λύσεις. Αν θέλουμε να αναφερθούμε σε μία από αυτές, πρέπει να δώσουμε ένα όνομα στις λύσεις. Έστω ότι τις ονομάζουμε sol.

> sol := solve ({eqn}, {x}) ;

$$sol := \left\{x = -2\right\}, \left\{x = \frac{1}{2} + \frac{1}{2}I\sqrt{7}\right\}, \left\{x = \frac{1}{2} - \frac{1}{2}I\sqrt{7}\right\}$$

Για να αναφερθούμε σε μια από αυτές, απλώς καλούμε τη μεταβλητή που συμβολίζει τις λύσεις με τον αριθμό της σειράς της λύσης.

Για τη δεύτερη λύση, για παράδειγμα, χρησιμοποιούμε:

```
> sol[2];
```

$$\left\{x = \frac{1}{2} + \frac{1}{2}I\sqrt{7}\right\}$$

Για να πάρουμε τη δεύτερη από τις λύσεις σε μορφή κινητής υποδιαστολής, δεν έχουμε παρά να χρησιμοποιήσουμε την εντολή `evalf`.

```
> evalf(sol[2]);
```

$$\{x = 0.5000000000 + 1.322875656 I\}$$

Η πρώτη λύση της εξίσωσης είναι το  $-2$ . Ας το επαληθεύσουμε κάνοντας αντικατάσταση στην εξίσωση.

```
> subs(sol[1], eqn);
```

$$4 = 4$$

Αν αντικαταστήσουμε το αποτέλεσμα της δεύτερης λύσης στην εξίσωση, δε φαίνεται κατευθείαν ότι αυτό επαληθεύει την εξίσωση. Γι' αυτό πρέπει να κάνουμε απλοποίηση της παράστασης.

```
> subs(sol[2], eqn);
```

$$-\left(\frac{1}{2} + \frac{1}{2}I\sqrt{7}\right)^3 - \left(\frac{1}{2} + \frac{1}{2}I\sqrt{7}\right)^2 = 4$$

```
> simplify(%);
```

$$4 = 4$$

□

#### Παράδειγμα: 7-4

Να λυθεί το σύστημα των εξισώσεων  $\begin{cases} x^2 \cdot y^2 = 0 \\ x + y = 1 \end{cases}$ .

#### Λύση:

Θα χρησιμοποιήσουμε την εντολή `solve` για την επίλυση του συστήματος και τις λύσεις θα τις ονομάσουμε `sol`. Έτσι:

```
> sol:=solve( {x^2*y^2=0, x+y=1}, {x,y} );
```

$$sol := \{y = 0, x = 1\}, \{y = 0, x = 1\}, \{x = 0, y = 1\}, \{x = 0, y = 1\}$$

Όπως και στο προηγούμενο παράδειγμα, εάν θέλουμε να αναφερθούμε σε μια από τις λύσεις, για παράδειγμα στη 2<sup>η</sup>, γράφουμε:

```
> sol [2] ;
```

$$\{y = 0, x = 1\}$$

Ενώ, αν θέλουμε να έχουμε μόνο την τιμή του x ή του y, για παράδειγμα στη δεύτερη λύση, επιλέγουμε από αυτή τη λίστα το 1<sup>ο</sup> ή 2<sup>ο</sup> στοιχείο της.

```
> sol [2] [1] ;
```

$$y = 0$$

```
> sol [2] [2] ;
```

$$x = 1$$

Αν, επιπλέον, έχουμε και έναν περιορισμό, για παράδειγμα το x να είναι διαφορετικό του μηδενός, τότε έχουμε:

```
> sol := solve ( {x^2*y^2=0, x+y=1, x<>0} ) ;
```

$$sol := \{y = 0, x = 1\}, \{y = 0, x = 1\}$$

Βλέπουμε ότι οι λύσεις που υπολογίστηκαν έχουν περιοριστεί σε 2, ώστε να ισχύει ο περιορισμός που θέσαμε.

□

### Παράδειγμα: 7-5

Να λυθεί η εξίσωση  $\cos(x) = 0$ .

**Λύση:**

Όπως είναι γνωστό, η εξίσωση έχει για λύσεις όλα τα  $x = 2k\pi + \frac{\pi}{2}$ , με  $k \in \mathbb{Z}$ .

Ας δούμε τι αποτέλεσμα επιστρέφει η εντολή solve.

```
> solve ( {cos (x)=0} , {x} ) ;
```

$$\{x = \frac{\pi}{2}\}$$

Δηλαδή, παίρνουμε ως λύση μόνο το πρώτο όρισμα των λύσεων. Αν θέλουμε να έχουμε τις άπειρες λύσεις της εξίσωσης, πρέπει να θέσουμε τη μεταβλητή περιβάλλοντος EnvAllSolutions «αληθή».

```
> _EnvAllSolutions:=true;
```

*`_EnvAllSolutions := true`*

Τώρα, αν ζητήσουμε πάλι τη λύση της εξίσωσης,

```
> solve({cos(x)=0}, {x});
```

έχουμε:

$$\{x = \frac{1}{2}\pi + \pi \_Z1\sim\}$$

Με το συμβολισμό  $\_Z1\sim$  συμβολίζονται οι ακέραιες τιμές.

□

### Παράδειγμα: 7-6

Να λυθεί η εξίσωση  $\frac{x^2 - 2x + 1}{x - 1} = 0$ .

**Λύση:**

Ορίζουμε την εξίσωση:

```
> eqn := (x^2 - 2*x + 1) / (x - 1) = 0;
```

$$eqn := \frac{x^2 - 2x + 1}{x - 1} = 0$$

Η λύση της ως προς x είναι:

```
> sol := solve({eqn}, {x});
```

$$sol := \{x = 1\}$$

Το Maple μας επέστρεψε ως λύση την τιμή 1, ενώ ουσιαστικά η εξίσωση δεν ορίζεται στο 1.

Αν κάνουμε επαλήθευση για τη λύση αυτή, θα έχουμε:

```
> subs(sol, eqn);
```

Error, numeric exception: division by zero

Το μήνυμα λάθους στην επαλήθευση δημιουργήθηκε, γιατί το Maple, πριν επιχειρήσει να λύσει μια εξίσωση, απομακρύνει τις ιδιομορφίες (singularity) και ουσιαστικά λύνει την εξίσωση κάνοντας προηγουμένως όλες τις δυνατές απλοποιήσεις ( Η παραπάνω ιδιομορφία δεν παρουσιάζεται στις εκδόσεις 10.5 και μετά, έτσι το πρόγραμμα δεν επιστέφει καμία λύση για την παραπάνω εξίσωση).

□



**Παρατήρηση: 7-1**

Η εντολή `solve` λύνει πάντα μια πολυωνυμική εξίσωση βαθμού  $n$ , αλλά ορισμένες από τις λύσεις δεν τις δίνει σε σαφή μορφή, αλλά τις παρουσιάζει μέσω της έκφρασης `RootOf`.

Η έκφραση `RootOf(expr, index)` συμβολίζει όλες τις ρίζες της `expr` και η παράμετρος `index` δείχνει τον αριθμό και τη σειρά των λύσεων.

Η έκφραση `RootOf` χρησιμοποιείται όταν το Maple δεν μπορεί να δώσει μια «κομψή» λύση ή όταν δεν μπορεί να βρει τη λύση, παρόλο που αυτή υπάρχει. Σ' αυτή την περίπτωση μπορούμε να προσδιορίσουμε τη λύση αριθμητικά.

Περιγραφή:	Αναπαριστά λύσεις μιας έκφρασης ή εξίσωσης.
Γενική σύνταξη:	<code>RootOf(έκφραση)</code>

Ας δούμε τα δύο επόμενα παραδείγματα.

**Παράδειγμα: 7-7**

Να λυθεί η εξίσωση  $x^4 + x^3 + x^2 + x - 1 = 0$ .

**Λύση:**

Ορίζουμε την πολυωνυμική εξίσωση  $x^4 + x^3 + x^2 + x - 1 = 0$ .

```
> eqn := x^4 + x^3 + x^2 + x - 1 = 0;
```

$$eqn := x^4 + x^3 + x^2 + x - 1 = 0$$

Αν προσπαθήσουμε να λύσουμε την εξίσωση με χρήση της εντολής `solve`, θα έχουμε τις τέσσερις λύσεις της εκφρασμένες με τη `RootOf`.

```
> sol := solve(eqn, {x});
```

$$\begin{aligned} sol := \{ & x = \text{RootOf}(\_Z^4 + \_Z^3 + \_Z^2 + \_Z - 1, index = 1) \}, \\ & \{ x = \text{RootOf}(\_Z^4 + \_Z^3 + \_Z^2 + \_Z - 1, index = 2) \}, \\ & \{ x = \text{RootOf}(\_Z^4 + \_Z^3 + \_Z^2 + \_Z - 1, index = 3) \}, \\ & \{ x = \text{RootOf}(\_Z^4 + \_Z^3 + \_Z^2 + \_Z - 1, index = 4) \} \end{aligned}$$

Αν θέλουμε να δούμε τη λύση που είναι εκφρασμένη με χρήση της `RootOf` εντολής, θα πρέπει να χρησιμοποιήσουμε την εντολή `allvalues`.

<b>Περιγραφή:</b>	<b>Υπολογίζει όλες τις δυνατές τιμές μιας έκφρασης που περιέχει την RootOf.</b>
<b>Γενική σύνταξη:</b>	<b><i>allvalues</i>(έκφραση)</b>

Αν ζητήσουμε την 1<sup>η</sup> λύση σε οποιαδήποτε δυνατή μορφή, τότε:

```
> allvalues(sol[1]);
```

$$\left\{ \begin{array}{l} x = -\frac{1}{4} \\ -\frac{\sqrt{3} \sqrt{\frac{-5(260 + 12\sqrt{1689})^{(1/3)} + 2(260 + 12\sqrt{1689})^{(2/3)} - 112}{(260 + 12\sqrt{1689})^{(1/3)}}}}{12} \\ + \left( \left( -30(260 + 12\sqrt{1689})^{(1/3)} \right. \right. \\ \left. \left. \sqrt{\frac{-5(260 + 12\sqrt{1689})^{(1/3)} + 2(260 + 12\sqrt{1689})^{(2/3)} - 112}{(260 + 12\sqrt{1689})^{(1/3)}}} - 6 \right. \right. \\ \left. \left. \sqrt{\frac{-5(260 + 12\sqrt{1689})^{(1/3)} + 2(260 + 12\sqrt{1689})^{(2/3)} - 112}{(260 + 12\sqrt{1689})^{(1/3)}}} \right. \right. \\ \left. \left. (260 + 12\sqrt{1689})^{(2/3)} \right. \right. \\ \left. \left. + 336 \sqrt{\frac{-5(260 + 12\sqrt{1689})^{(1/3)} + 2(260 + 12\sqrt{1689})^{(2/3)} - 112}{(260 + 12\sqrt{1689})^{(1/3)}}} \right) \right\}$$

$$\left. \left( +90\sqrt{3} (260 + 12\sqrt{1689})^{(1/3)} \right) / \left( (260 + 12\sqrt{1689})^{(1/3)} \right) \right\} \\ \left. \sqrt[12]{ \frac{-5(260+12\sqrt{1689})^{(1/3)} + 2(260+12\sqrt{1689})^{(2/3)} - 112}{(260+12\sqrt{1689})^{(1/3)}} } \right\} / 12$$

Η παραπάνω μορφή βέβαια είναι αρκετά άκομψη και δύσχρηστη. Έτσι εδώ είναι καλύτερο να ζητήσουμε τις λύσεις προσεγγιστικά με την εντολή `evalf`.

`> evalf(sol);`

$$\{x = 0.5187900637\}, \{x = -0.1140706312 + 1.216746004 I\}, \\ \{x = -1.290648801\}, \{x = -0.1140706312 - 1.216746004 I\}$$

□

### Παράδειγμα: 7-8

Να λυθεί η εξίσωση  $x^6 + x^5 + x^3 + x^2 + x + 1 = 0$ .

#### Λύση:

Ορίζουμε την εξίσωση.

`> eqn := x^6 + x^5 + x^3 + x^2 + x + 1 = 0;`

$$eqn := x^6 + x^5 + x^3 + x^2 + x + 1 = 0$$

Αν ζητήσουμε τις λύσεις της εξίσωσης, η εντολή `solve` θα μας επιστρέψει κάποιες από αυτές χρησιμοποιώντας την έκφραση `RootOf`.

`> sol := solve(eqn, {x});`

$$sol := \{x = -1\}, \{x = \text{RootOf}(\_Z^5 + \_Z^2 + 1, \text{index} = 1)\}, \\ \{x = \text{RootOf}(\_Z^5 + \_Z^2 + 1, \text{index} = 2)\}, \\ \{x = \text{RootOf}(\_Z^5 + \_Z^2 + 1, \text{index} = 3)\}, \\ \{x = \text{RootOf}(\_Z^5 + \_Z^2 + 1, \text{index} = 4)\}, \\ \{x = \text{RootOf}(\_Z^5 + \_Z^2 + 1, \text{index} = 5)\}$$

Αν εδώ ζητήσουμε τη 2<sup>η</sup> λύση χωρίς να μας ενδιαφέρει η μορφή της, το Maple θα μας την επιστρέψει πάλι σε μορφή `RootOf`, γιατί δεν μπορεί να εκφράσει τη λύση ούτε σε απλή ούτε σε πολύπλοκη μορφή.

```
> allvalues(sol[2]);
```

$$\{x = \text{RootOf}(\_Z^5 + \_Z^2 + 1, \text{index} = 1)\}$$

Επομένως, το μόνο που μπορούμε να κάνουμε είναι να ζητήσουμε την αριθμητική προσέγγιση των λύσεων με την εντολή `evalf`.

```
> evalf(sol);
```

$$\{x = -1.\}, \{x = 0.7515192324 + 0.7846159210 I\},$$

$$\{x = -0.1545896767 + 0.8280741332 I\}, \{x = -1.193859111\},$$

$$\{x = -0.1545896767 - 0.8280741332 I\},$$

$$\{x = 0.7515192324 - 0.7846159210 I\}$$

□

## 7.2 Προσεγγιστική Επίλυση Αλγεβρικών Εξισώσεων

Η εντολή `solve` επιστρέφει πάντα τη λύση μιας πολυωνυμικής εξίσωσης έστω και εάν χρησιμοποιεί την έκφραση `RootOf`. Υπάρχουν, όμως, μη γραμμικές εξισώσεις, οι οποίες δεν επιδέχονται ακριβή λύση. Στην περίπτωση που μια εξίσωση ή ένα σύστημα εξισώσεων δεν επιδέχεται ακριβή λύση μπορούμε να έχουμε λύση αριθμητικά με την εντολή `fsolve` (η εντολή δίνει την αριθμητική προσέγγιση των λύσεων, όπως και η εντολή `evalf` που είδαμε στο προηγούμενο παράδειγμα).

Περιγραφή:	Λύνει μια εξίσωση ή ένα σύστημα εξισώσεων αριθμητικά.	
Γενική σύνταξη:	<i>fsolve</i> ( <i>eqns</i> , <i>vars</i> , <i>options</i> )	
Παράμετροι:	<i>eqns</i>	Εξίσωση ή σύστημα εξισώσεων {εξίσωση1, εξίσωση2,...}
	<i>vars</i>	Μεταβλητή ή μεταβλητές {μεταβλητή1, μεταβλητή2,...} ως προς τις οποίες θέλω τη λύση (αυτή η παράμετρος μπορεί πολλές φορές να παραλειφθεί).
	<i>maxsols</i> =	Μέγιστος αριθμός λύσεων σε πολυωνυμικές εξισώσεις.
	<i>complex</i>	Με την παράμετρο αυτή αναζητούμε εκτός από πραγματικές και μιγαδικές λύσεις.
	<i>min..max</i>	Το διάστημα στο οποίο θα αναζητηθεί λύση.

**Παράδειγμα: 7-9**

Να λυθεί η εξίσωση  $x^5 + x^2 + 1 = 0$ .

**Λύση:**

Ορίζουμε την εξίσωση.

```
> eqn := x^5 + x^2 + 1;
```

$$eqn := x^5 + x^2 + 1$$

Η χρήση της εντολής solve δίνει :

```
> sol := solve (eqn, {x});
```

$$\begin{aligned} sol := \{ & x = \text{RootOf}(\_Z^5 + \_Z^2 + 1, index = 1) \}, \\ & \{ x = \text{RootOf}(\_Z^5 + \_Z^2 + 1, index = 2) \}, \\ & \{ x = \text{RootOf}(\_Z^5 + \_Z^2 + 1, index = 3) \}, \\ & \{ x = \text{RootOf}(\_Z^5 + \_Z^2 + 1, index = 4) \}, \\ & \{ x = \text{RootOf}(\_Z^5 + \_Z^2 + 1, index = 5) \} \end{aligned}$$

Και εδώ η εντολής solve επιστρέφει τις ρίζες συμβολικά χρησιμοποιώντας την RootOf δομή. Χρησιμοποιώντας την εντολή fsolve έχουμε την 1 πραγματική ρίζα της εξίσωσης.

```
> fsolve (eqn, {x});
```

$$\{x = -1.193859111\}$$

Ενώ, για να δούμε και τις μιγαδικές λύσεις, πρέπει να προσθέσουμε την παράμετρο complex.

```
> fsolve (eqn, {x}, complex);
```

$$\begin{aligned} \{x = -1.193859111\}, \{x = -0.1545896767 - 0.8280741332 I\}, \\ \{x = -0.1545896767 + 0.8280741332 I\}, \\ \{x = 0.7515192324 - 0.7846159210 I\}, \\ \{x = 0.7515192324 + 0.7846159210 I\} \end{aligned}$$

Το ίδιο αποτέλεσμα θα είχαμε, αν ζητούσαμε προσέγγιση της sol με την εντολή evalf.

```
> evalf (sol);
```

$$\{x = 0.7515192324 + 0.7846159210I\}, \{x = -0.1545896767 + 0.8280741332I\}, \\ \{x = -1.193859111\}, \{x = -0.1545896767 - 0.8280741332I\}, \\ \{x = 0.7515192324 - 0.7846159210I\}$$

□

**Παρατήρηση: 7-2**

Αν χρησιμοποιηθεί η εντολή `fsolve` σε μια εξίσωση στην οποία μπορούμε να έχουμε ακριβή λύση, τότε το πρόγραμμα θα μας επιστρέψει τη λύση σε μορφή κινητής υποδιαστολής.

**Παράδειγμα: 7-10**

Να λυθεί η εξίσωση  $2 \sin x = 1$ .

**Λύση:**

Η εντολή `solve` δίνει την ακριβή λύση, που εδώ υπάρχει.

```
> solve(2*sin(x)=1);
```

$$\frac{\pi}{6}$$

Η εντολή `fsove` δίνει την αριθμητική της προσέγγιση.

```
> fsolve(2*sin(x)=1);
```

$$0.5235987756$$

□

**Παράδειγμα: 7-11**

Να λυθεί η εξίσωση  $\frac{x^5}{2} - \cos x^9 = 0$ .

**Λύση:**

Ορίζουμε την εξίσωση.

```
> eqn := (x^5)/2 - cos(x^9) = 0;
```

$$eqn := \frac{x^5}{2} - \cos(x^9) = 0$$

Η χρήση της εντολής `solve` δε δίνει κανένα αποτέλεσμα. Ωστόσο, αυτό δε σημαίνει ότι δεν υπάρχει λύση της εξίσωσης.

```
> solve (eqn, {x});
```

*Warning, solutions may have been lost*

Αν χρησιμοποιήσουμε την εντολή `SolutionsMayBeLost`, το πρόγραμμα θα μας επιστρέψει «αληθή» τιμή, πράγμα που σημαίνει ότι πράγματι υπάρχουν λύσεις και η εντολή `solve` δεν τις υπολόγισε.

```
> _SolutionsMayBeLost;
```

*true*

Ένας άλλος τρόπος για να έχουμε πληροφορίες για το αν υπάρχουν λύσεις που δεν υπολογίστηκαν είναι να χρησιμοποιήσουμε την εντολή `infolevel`. Αν ορίσουμε το επίπεδο πληροφοριών που επιστρέφει το Maple στο 1 και προσπαθήσουμε να λύσουμε πάλι την εξίσωση, θα έχουμε:

```
> infolevel[solve] :=1;
```

```
> solve (eqn, {x});
```

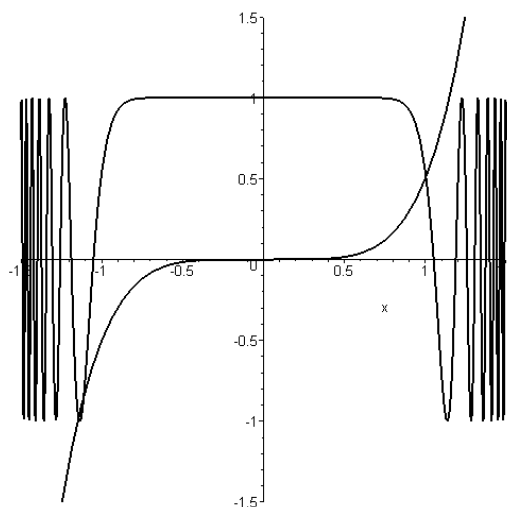
```
solve: Warning: no solutions found
```

```
solve: Warning: solutions may have been lost
```

Αν κάνουμε τις γραφικές παραστάσεις των δύο συναρτήσεων που αποτελούν την εξίσωση, βλέπουμε ότι αυτές τέμνονται σε 3 σημεία. Άρα, υπάρχουν 3 λύσεις της εξίσωσης, μια κοντά στην τιμή 1 και δύο λύσεις κοντά στο -1.

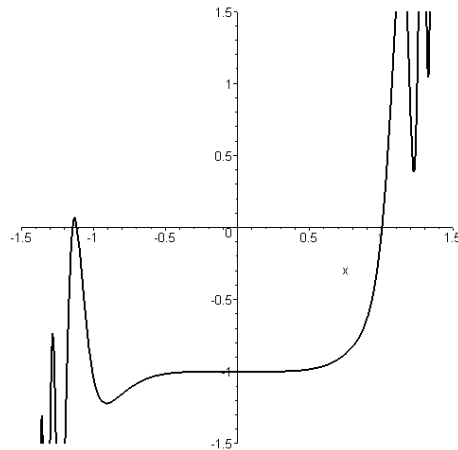
(περισσότερα για την εντολή `plot` στο κεφάλαιο 9).

```
> plot ([ (x^5)/2, cos (x^9) ], x=-1.5..1.5, -1.5..1.5 );
```



Το ίδιο μπορούμε να δούμε από τη γραφική παράσταση της συνάρτησης που ορίζει την εξίσωση.

```
> plot( (x^5)/2-cos(x^9), x=-1.5..1.5, -1.5..1.5 );
```



Αν ζητήσουμε μια προσεγγίσιμη λύση χρησιμοποιώντας την εντολή fsolve, θα έχουμε:

```
> fsolve(eqn, x);
```

1.003913143

Τη δεύτερη λύση την αναζητούμε στο διάστημα [-1.5,-1].

```
> fsolve(eqn, x, -1.5..-1);
```

-1.111072980

Την τρίτη λύση την αναζητούμε στο διάστημα [-1.5,-1.1].

```
> fsolve(eqn, x, -1.5..-1.1);
```

-1.143731386

□

### Παράδειγμα: 7-12

Να λυθεί το σύστημα των εξισώσεων  $\begin{cases} y = 2 \sin x \\ x^2 + y^2 = 1 \end{cases}$ .



**Λύση:**

Ορίζουμε τις εξισώσεις.

```
> eq1 := y = 2 * sin(x) ;
```

$$eq1 := y = 2 \sin(x)$$

```
> eq2 := x^2 + y^2 = 1 ;
```

$$eq2 := x^2 + y^2 = 1$$

Ζητάμε τη λύση του συστήματος με την εντολή Solve.

```
> solve({eq1, eq2}, {x, y}) ;
```

$$\{x = \text{RootOf}(\_Z^2 + 4 \sin(\_Z)^2 - 1, \text{label} = \_L11), \\ y = 2 \sin(\text{RootOf}(\_Z^2 + 4 \sin(\_Z)^2 - 1, \text{label} = \_L11))\}$$

Η εντολή solve δίνει ένα ζεύγος λύσεων εκφρασμένο με την RootOf. Έτσι, με την εντολή fsolve παίρνουμε την προσεγγιστική λύση.

```
> fsolve({eq1, eq2}, {x, y}) ;
```

$$\{x = 0.4600098267, y = 0.8879138243\}$$

□

**Παράδειγμα: 7-13**

Να λυθεί το σύστημα  $\begin{cases} x^2 + y^2 = 9 \\ x^y = 2 \end{cases}$ .

**Λύση:**

Ορίζουμε τις εξισώσεις.

```
> eq1 := x^2 + y^2 = 9 ;
```

$$eq1 := x^2 + y^2 = 9$$

```
> eq2 := x^y = 2 ;
```

$$eq2 := x^y = 2$$

Ζητάμε τη λύση του συστήματος με την εντολή solve.

```
> solve({eq1, eq2}, {x, y}) ;
```

$$\left\{ \begin{array}{l} y = \frac{\ln(2)}{\text{RootOf}((e^{-z})^2 - z^2 + \ln(2)^2 - 9 - z^2)}, \\ x = e^{\text{RootOf}((e^{-z})^2 - z^2 + \ln(2)^2 - 9 - z^2)} \end{array} \right\}$$

Και η προσεγγιστική τιμή της είναι:

```
> evalf(%);
```

$$\{x = 2.929883880, y = 0.6448103938\}$$

Την ίδια προσεγγιστική λύση παίρνουμε και με την εντολή `fsolve`.

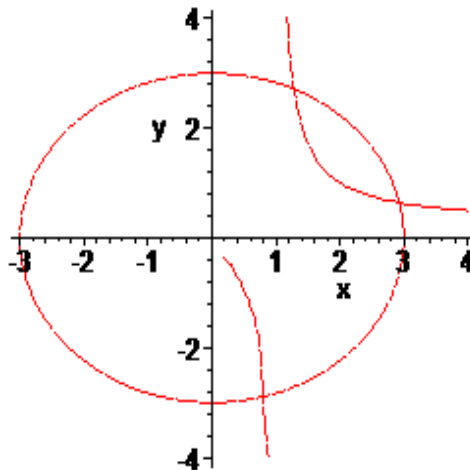
```
> fsolve( {eq1, eq2}, {x, y} );
```

$$\{x = 2.929883881, y = 0.6448103937\}$$

Σχηματίζουμε τη γραφική παράσταση των δύο αυτών εξισώσεων με τη βοήθεια του πακέτου `plots` και της εντολής `implicitplot` (δες περισσότερα για την εντολή αυτή στο κεφάλαιο 9).

```
> with(plots):
```

```
> implicitplot( {x^2+y^2=9, x^y=2}, x=-4..4, y=-4..4 );
```



Από τη γραφική παράσταση βλέπουμε ότι πρέπει να αναζητήσουμε δύο ακόμα λύσεις του συστήματος, μια για  $x \in [0, 1]$  και μια για  $x \in [1, 2]$ .

```
> fsolve({eq1, eq2}, {x, y}, x=0..1);
      {x = 0.7870713992, y = -2.894912540}
> fsolve({eq1, eq2}, {x, y}, x=1..2);
      {x = 1.291746892, y = 2.707653960}
```

□

### 7.3 Λύση Ανισώσεων

Χρησιμοποιώντας την εντολή `solve` μπορούμε να λύσουμε και ανισώσεις.

Ας δούμε κατευθείαν κάποια παραδείγματα:

#### Παράδειγμα: 7-14

Να λυθεί η ανίσωση  $x^2 + 5x + 6 < 0$ .

#### Λύση:

Ορίζουμε ως `ineq` την ανίσωση.

```
> ineq := x^2+5*x+6 < 0;
      ineq := x^2 + 5 x < -6
> solve( ineq, {x} );
      {-3 < x, x < -2}
```

□

#### Παράδειγμα: 7-15

Να λυθεί η ανίσωση  $x^3 - 6x^2 + 11x - 6 > 0$ .

#### Λύση:

Ορίζουμε ως `ineq` την ανίσωση.

```
> ineq := x^3-6*x^2+11*x-6>0;
      ineq := 0 < x^3 - 6 x^2 + 11 x - 6
> solve( ineq, {x} );
      {1 < x, x < 2}, {3 < x}
```

□

**Παράδειγμα: 7-16**

Να συναληθευθούν οι ανισώσεις  $\begin{cases} x^2 - 3x + 2 \geq 0 \\ 2x - 8 < 0 \end{cases}$ .

**Λύση:**

Ορίζουμε ως `ineq1` και `ineq2` τις δύο ανισώσεις.

```
> ineq1:=x^2-3*x+2>=0;
```

$$\text{ineq1} := 0 \leq x^2 - 3x + 2$$

```
> ineq2:=2*x-8<0;
```

$$\text{ineq2} := 2x < 8$$

Ζητάμε τη συναλήθευσή τους ως προς  $x$ .

```
> solve({ineq1, ineq2}, {x});
```

$$\{x \leq 1\}, \{x < 4, 2 \leq x\}$$

□

Ιδιαίτερα χρήσιμο είναι στην περίπτωση που έχουμε να συναληθεύσουμε κάποιες γραμμικές ανισώσεις, να έχουμε μια εικόνα για την περιοχή του επιπέδου, στην οποία ισχύουν οι ανισώσεις. Αυτό το επιτυγχάνουμε με την εντολή `inequal`.

<b>Περιγραφή:</b>	<i>Σχεδιάζει την περιοχή που ορίζεται από γραμμικές ανισώσεις.</i>	
<b>Γενική σύνταξη:</b>	<i><code>inequal(ineqs, x, y, options)</code></i>	
<b>Παράμετροι:</b>	<code>ineqs</code>	Η ανίσωση ή οι ανισώσεις που θα σχεδιάσουμε. Αν είναι περισσότερες από μία γράφονται μέσα σε <code>{ }</code> .
	<code>x</code>	<code>x</code> = ελάχιστη .. μέγιστη τιμή του <code>x</code>
	<code>y</code>	<code>y</code> = ελάχιστη .. μέγιστη τιμή του <code>y</code>
<b>Πακέτο:</b>	<b>plots</b>	

**Παράδειγμα: 7-17**

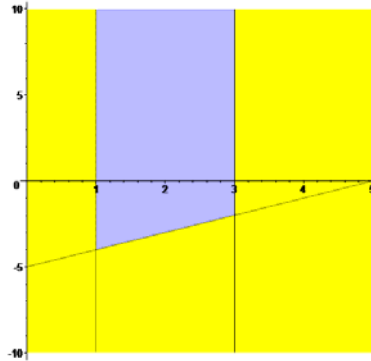
Να σχεδιαστεί η περιοχή που περικλείεται από τις ανισώσεις  $\begin{cases} x - y < 5 \\ 1 < x \leq 3 \end{cases}$ .

**Λύση:**

Ενεργοποιούμε αρχικά το πακέτο plots.

```
> with(plots):
```

```
> inequal( {x-y<5, 1<x, x<=3}, x=0..5, y=-10..10,
optionsexcluded= (color=yellow) );
```



□

**Παράδειγμα: 7-18**

Να σχεδιαστεί η περιοχή που περικλείεται από τις ανισώσεις

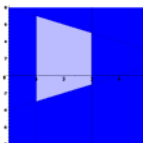
$$\begin{cases} x - y < 4 \\ x + y < 8 \\ x \leq 3 \\ x > 1 \end{cases}.$$
**Λύση:**

Ενεργοποιούμε το πακέτο plots.

```
> with(plots):
```

Χρησιμοποιούμε την εντολή inequal και την παράμετρο color για να ορίσουμε το χρώμα της περιοχής σε μπλε.

```
> inequal( {x-y<4, x+y<8, x<=3, x>1}, x=0..5, y=-8..8,
optionsexcluded= (color=blue) );
```



□



# Κεφάλαιο 8

## Ακολουθίες Αριθμών-Σειρές

Στο κεφάλαιο αυτό θα δούμε τον τρόπο με τον οποίο μπορούμε να υπολογίσουμε αθροίσματα και γινόμενα ακολουθιών. Επίσης, θα δούμε με παραδείγματα τον υπολογισμό γνωστών σειρών και δυναμοσειρών.

### 8.1 Ακολουθίες

Για να ορίσουμε μια ακολουθία αριθμών χρησιμοποιούμε την εντολή `seq` που είδαμε και στο κεφάλαιο 5.

#### Παράδειγμα: 8-1

Δημιουργούμε την ακολουθία  $n^2$  για  $n = -4, \dots, 4$  και την ονομάζουμε ως `seq1`.

```
> seq1:=seq( n^2, n=-4..4 ) ;  
seq1 := 16, 9, 4, 1, 0, 1, 4, 9, 16
```

```
> whattype (seq1) ;
```

*exprseq*

□

**Παράδειγμα: 8-2**

Στο παράδειγμα αυτό θα δημιουργήσουμε την ακολουθία των 20 πρώτων περιττών αριθμών.

**Λύση:**

```
> seq( 2*n+1, n=0..19 ) ;
```

1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39

Αν θέλουμε να αναφερθούμε σε κάποιον συγκεκριμένο όρο της ακολουθίας, πρέπει να την ονομάσουμε. Εδώ θα χρησιμοποιήσουμε τη μεταβλητή s.

```
> s := seq( 2*n+1, n=0..19 ) ;
```

s := 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39

Ο πρώτος όρος είναι:

```
> s[1] ;
```

1

Και ο δεύτερος:

```
> s[2] ;
```

Οι πρώτοι τέσσερις όροι είναι:

```
> s[1..4] ;
```

1, 3, 5, 7

Για να αναφερθούμε στους όρους της ακολουθίας, χρησιμοποιήσαμε τις τετράγωνα αγκύλες [ ]. Παρατηρούμε ότι το ίδιο αποτέλεσμα θα είχαμε και με την εντολή op.

□

**✚ Άθροισμα Όρων Ακολουθίας**

Με την εντολή sum μπορούμε να υπολογίσουμε το άθροισμα των όρων μιας ακολουθίας εκφράσεων ή αριθμών.

<i>Περιγραφή:</i>	<i>Υπολογίζει το άθροισμα μιας έκφρασης.</i>	
<i>Γενική σύνταξη:</i>	sum(expr,x=a..b)	
<i>Παράμετροι:</i>	expr:	Η έκφραση.
	x:	Ο δείκτης.
	a..b:	Τα όρια του δείκτη.



**Παράδειγμα: 8-3**

Να υπολογιστούν τα αθροίσματα  $\sum_{n=0}^5 n^2$ ,  $\sum_{n=0}^k n^2$ .

**Λύση:**

Για να υπολογίσουμε το άθροισμα των τετραγώνων των ακεραίων αριθμών από το 0 έως το 5, γράφουμε :

```
> sum(n^2, n=0..5);
```

55

το άθροισμα  $\sum_{n=0}^k n^2$  είναι:

```
> sum(n^2, n=0..k);
```

$$\frac{(k+1)^3}{3} - \frac{(k+1)^2}{2} + \frac{k}{6} + \frac{1}{6}$$

□

**🚩 Γινόμενο Όρων μιας Ακολουθίας**

<i>Περιγραφή:</i>	<i>Υπολογίζει το γινόμενο.</i>	
<i>Γενική σύνταξη:</i>	<i>product(expr,x=a..b)</i>	
<i>Παράμετροι:</i>	expr:	Η έκφραση.
	x:	Ο δείκτης.
	a..b:	Τα όρια του δείκτη.

Θα υπολογίσουμε το γινόμενο  $\prod_{k=1}^4 k^2$

```
> product( k^2, k=1..4 );
```

576

**Παρατήρηση: 8.1**

Αν τις εντολές `sum` και `product` τις γράψουμε με κεφαλαίο το πρώτο γράμμα, τότε το maple εμφανίζει αντίστοιχο άθροισμα ή γινόμενο χωρίς να το υπολογίζει. Στη συνέχεια μπορεί να υπολογιστεί το όριο με τη χρήση της εντολής `value()`.

> `Sum(n, n=0..k);`

$$\sum_{n=0}^k n$$

> `value(%);`

$$\frac{(k+1)^2}{2} - \frac{k}{2} - \frac{1}{2}$$

□

**8.2 Σειρές****Παράδειγμα: 8-4**

Να υπολογιστούν οι σειρές:

a)  $\sum_{n=1}^{\infty} \frac{1}{n}$  (αρμονική σειρά)

b)  $\sum_{n=1}^{\infty} \frac{(-1)^n}{n}$  (εναλλάσσουσα σειρά)

c)  $\sum_{n=1}^{\infty} r^n$  για τις διάφορες τιμές του  $r$  (γεωμετρική σειρά)

d)  $\sum_{n=0}^{\infty} \frac{1}{(n+1)(n+2)}$  (Τηλεσκοπική σειρά)

**Λύση:**

Για τον υπολογισμό της σειράς θα χρησιμοποιήσουμε την εντολή `sum`, αρχικά με κεφαλαίο `S` για να εμφανισθεί το σύμβολο του αθροίσματος και στη συνέχεια με μικρό για τον υπολογισμό του.

a) `Sum(1/n, n=1..infinity)=sum(1/n, n=1..infinity);`

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty$$

b)  $\text{Sum}((-1)^n/n, n=1..infinity) = \text{sum}((-1)^n/n, n=1..infinity);$

$$\sum_{n=1}^{\infty} \frac{(-1)^n}{n} = -\ln(2)$$

c) Αρχικά υποθέτουμε ότι το  $|r| > 1$ . Έτσι:

$> \text{assume}(\text{abs}(r) > 1);$

$> \text{Sum}(r^n, n=1..infinity) = \text{sum}(r^n, n=1..infinity);$

$$\sum_{n=1}^{\infty} r^n = \infty$$

Αν υποθέσουμε ότι  $|r| < 1$ , έχουμε:

$> \text{assume}(\text{abs}(r) < 1);$

$> \text{Sum}(r^n, n=1..infinity) = \text{sum}(r^n, n=1..infinity);$

$$\sum_{n=1}^{\infty} r^n = -\frac{r}{r-1}$$

Ως εφαρμογή των παραπάνω θα υπολογίσουμε τις σειρές  $\sum_{n=1}^{\infty} \left(\frac{3}{2}\right)^n$  και

$$\sum_{n=1}^{\infty} \left(\frac{1}{2}\right)^n.$$

$> \text{Sum}((3/2)^n, n=1..infinity) = \text{sum}((3/2)^n, n=1..infinity);$

$$\sum_{n=1}^{\infty} \left(\frac{3}{2}\right)^n = \infty$$

$> \text{Sum}((1/2)^n, n=1..infinity) = \text{sum}((1/2)^n, n=1..infinity);$

$$\sum_{n=1}^{\infty} \left(\frac{1}{2}\right)^n = 1$$

d)

> Sum(1/((n+1)\*(n+2)), n=0..infinity) = sum(1/((n+1)\*(n+2)), n=0..infinity);

$$\sum_{n=0}^{\infty} \frac{1}{(n+1)(n+2)} = 1$$

□

### 8.3 Δυναμοσειρές

Για τον υπολογισμό του αναπτύγματος μιας συνάρτησης σε άθροισμα δυνάμεων της ανεξάρτητης μεταβλητής, το Maple διαθέτει την εντολή `series`.

<i>Περιγραφή:</i>	<i>Υπολογίζει το ανάπτυγμα σε σειρά μιας συνάρτησης.</i>	
<i>Γενική σύνταξη:</i>	<i>series(expr, eqn, n)</i>	
<i>Παράμετροι:</i>	expr:	Η συνάρτηση της οποίας ζητάμε το ανάπτυγμα.
	eqn:	Το κέντρο $x = x_0$ γύρω από το οποίο θα υπολογιστεί το ανάπτυγμα.
	n:	Μη αρνητικός ακέραιος.

#### Παράδειγμα: 8-4

Να αναπτυχθεί σε σειρά με κέντρο το 0 η συνάρτηση  $e^x$  (σειρά Maclaurin).

**Λύση:**

> series( exp(x), x=0, 5 );

$$1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + O(x^5)$$

> whattype(%);

*series*

Μετατρέπουμε το παραπάνω ανάπτυγμα σε πολώνυμο.

> convert(%%, polynom) ;

$$1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4$$

□

### Παρατήρηση: 8-2

Το σύμβολό  $O(x^5)$  που εμφανίζεται στο παραπάνω παράδειγμα σημαίνει ότι ακολουθούν όροι 5ης τάξης.

### Παράδειγμα: 8-5

Να αναπτυχθεί σε σειρά με κέντρο το  $\pi$  η συνάρτηση  $\sin(x)$ .

#### Λύση:

> series(sin(x), x=Pi, 6) ;

$$-(x - \pi) + \frac{1}{6}(x - \pi)^3 - \frac{1}{120}(x - \pi)^5 + O((x - \pi)^6)$$

Μετατρέπουμε το παραπάνω ανάπτυγμα σε πολυώνυμο.

> convert(%, polynom) ;

$$-x + \pi + \frac{(x - \pi)^3}{6} - \frac{(x - \pi)^5}{120}$$

□



# Κεφάλαιο 9

## Λογισμός Συναρτήσεων μιας Μεταβλητής

Στο κεφάλαιο αυτό θα δούμε τις βασικότερες εντολές που μπορούμε να χρησιμοποιήσουμε στο λογισμό συναρτήσεων μιας μεταβλητής. Θα ξεκινήσουμε με τον τρόπο ορισμού μιας συνάρτησης και στη συνέχεια θα δούμε τη βασική για το λογισμό έννοια του ορίου. Στην παράγραφο 9.3 θα δούμε ποιες είναι οι εντολές για τον υπολογισμό των παραγώγων μιας συνάρτησης, ενώ στην παράγραφο 9.4 θα δούμε την εντολή για τον υπολογισμό ολοκληρωμάτων.

### 9.1 Ορισμός Συνάρτησης

Είναι χρήσιμο να αναφερόμαστε σε εκφράσεις χρησιμοποιώντας διάφορους συμβολισμούς για αυτές. Έτσι, μπορούμε να ορίσουμε μια συνάρτηση με τον εξής τρόπο:

<i>Γενική σύνταξη:</i>	<i>Όνομα συνάρτησης:= ανεξάρτητη μεταβλητή-&gt;τύπος</i>
------------------------	--

**Παράδειγμα: 9-1**

Να ορισθεί η συνάρτηση  $f(x) = x^3$  και να βρεθούν  $f(2)$ ,  $f(x+2)$ .

**Λύση:**

Ορίζουμε τη συνάρτηση και τη θέτουμε με τη μεταβλητή f.

```
> f := x -> x^3;
```

$$f := x \rightarrow x^3$$

Ζητάμε την τιμή της στο 2 καθώς και στο x+2.

```
> f(2);
```

8

```
> f(x+2);
```

$$(x+2)^3$$

□

**Παράδειγμα: 9-2**

Να ορισθεί η συνάρτηση  $g(x) = \sin \frac{x}{2}$  και να βρεθούν  $g(1)$ .

**Λύση:**

Ορίζουμε τη συνάρτηση g.

```
> g := x -> sin(x/2);
```

$$g := x \rightarrow \sin\left(\frac{1}{2}x\right)$$

Ζητάμε την τιμή της συνάρτησης στο  $\frac{\pi}{2}$  και στο 1.

```
> g(Pi/2);
```

$$\frac{\sqrt{2}}{2}$$

Η προσεγγιστική τιμή  $g(1)$  είναι:

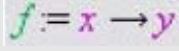
```
> evalf(%);
```

0.7071067810

□



Ένας άλλος τρόπος για να ορίσουμε μια συνάρτηση είναι η εντολή `unapply`.

<i>Περιγραφή:</i>	<i>Ορίζει μια Συνάρτηση.</i>
<i>Γενική σύνταξη:</i>	<i>Όνομα συνάρτησης :=unapply (τύπος, ανεξάρτητη μεταβλητή)</i>
<i>Expression Palette</i>	

### Παράδειγμα: 9-3

Να ορισθεί η συνάρτηση  $f(x) = |x^2 - 4|$  και να γίνει η γραφική της παράσταση στο διάστημα  $[-4,4]$ .

#### Λύση:

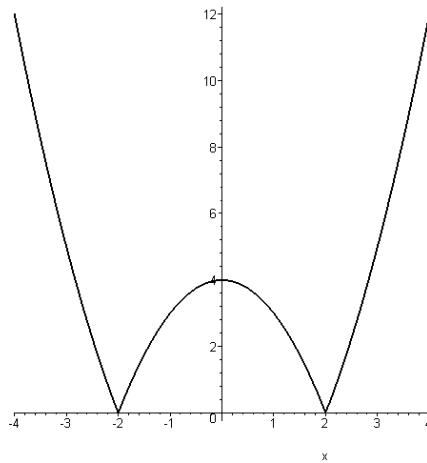
Ορίζουμε τη συνάρτηση χρησιμοποιώντας την εντολή `unapply`. Θυμίζουμε ότι η απόλυτη τιμή δίνεται με την εντολή `abs`.

```
> f := unapply( abs(x^2-4), x );
```

$$f := x \rightarrow |x^2 - 4|$$

Ας δούμε σε αυτό το παράδειγμα τη γραφική παράσταση της συνάρτησης  $f$  χρησιμοποιώντας την εντολή `plot`. Την εντολή αυτή θα την παρουσιάσουμε αναλυτικά σε επόμενη παράγραφο.

```
> plot( f(x), x=-4..4 );
```



□

Για τον ορισμό πολύκλαδων συναρτήσεων μπορούμε να χρησιμοποιήσουμε την εντολή `piecewise`.

<b>Περιγραφή:</b>	<b>Ορίζει μια πολύκλαδη συνάρτηση.</b>
<b>Γενική σύνταξη:</b>	<b>Όνομα συνάρτησης:=<code>piecewise</code>(συνθήκη 1, τύπος 1, συνθήκη2, τύπος2)</b>

#### Παράδειγμα: 9-4

Να ορισθεί η δίκλαδη  $f(x) = \begin{cases} -x & x \leq -1 \\ x^2 & x \leq 1 \\ \frac{\sin(x-1)}{x-1} & 1 < x \end{cases}$  συνάρτηση και να γίνει η γραφική

της παράσταση στο διάστημα  $[-2,2]$ .

#### Λύση:

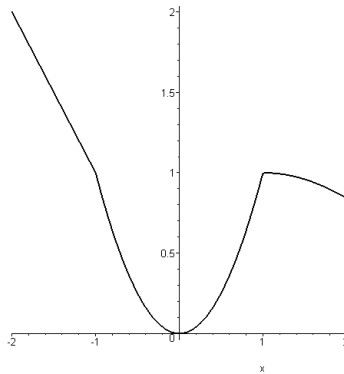
Ορίζουμε τη συνάρτηση.

```
> f:=piecewise(x<=-1, -x, x<=1, x*x, x>1, sin(x-1)/(x-1) );
```

$$f := \begin{cases} -x & x \leq -1 \\ x^2 & x \leq 1 \\ \frac{\sin(x-1)}{x-1} & 1 < x \end{cases}$$

Η γραφική παράστασή της στο διάστημα  $[-2,2]$  είναι:

```
> plot(f(x), x=-2..2);
```



□

Οι πράξεις ανάμεσα σε συναρτήσεις ορίζονται με τους γνωστούς τελεστές +, -, \*, / και η σύνθεση με το σύμβολο @.

Περιγραφή	Γενική σύνταξη:
Πρόσθεση Συναρτήσεων	$f+g$ ;
Αφαίρεση Συναρτήσεων	$f-g$ ;
Πολλαπλασιασμός Συναρτήσεων	$f*g$ ;
Πηλίκο Συναρτήσεων	$f/g$ ;
Σύνθεση συναρτήσεων	$f@g$ ;

### Παράδειγμα: 9-5

Δίνονται οι συναρτήσεις  $f(x) = \cos x$  και  $g(x) = \frac{x}{2}$ . Να υπολογιστούν οι  $f + g$ ,

$f \cdot g$ ,  $\frac{f}{g}$  και οι συνθέσεις  $f \circ g$  και  $g \circ f$ .

#### Λύση:

Ορίζουμε τις συναρτήσεις  $f$  και  $g$ .

>  $f := x \rightarrow \cos(x)$  ;

$$f := x \rightarrow \cos(x)$$

>  $g := x \rightarrow x/2$  ;

$$g := x \rightarrow \frac{1}{2}x$$

Η πρόσθεση των συναρτήσεων είναι:

>  $(f+g)(x)$  ;

$$\cos(x) + \frac{x}{2}$$

Το γινόμενο των συναρτήσεων είναι:

>  $(f*g)(x)$  ;

$$\frac{1}{2} \cos(x) x$$

Το πηλίκο των συναρτήσεων είναι:

> (f/g) (x) ;

$$\frac{2 \cos(x)}{x}$$

Η σύνθεση  $f \circ g$  είναι:

> (f@g) (x) ;

$$\cos\left(\frac{x}{2}\right)$$

Η σύνθεση  $g \circ f$  είναι:

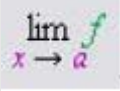
> (g@f) (x) ;

$$\frac{1}{2} \cos(x)$$

□

## 9.2 Όριο

Όρια μπορούμε εύκολα να υπολογίσουμε με τη βοήθεια του Maple χρησιμοποιώντας την εντολή `limit`.

<i>Περιγραφή:</i>	<i>Υπολογίζει το Όριο μιας έκφρασης.</i>	
<i>Γενική σύνταξη:</i>	<i>limit(f, x=a, dir)</i>	
<i>Παράμετροι:</i>	f:	Η συνάρτηση της οποίας αναζητούμε το όριο.
	x:	Το σημείο στο οποίο αναζητούμε το όριο.
	dir	right, left ανάλογα αν θέλουμε το από δεξιά ή το από αριστερά όριο.
<i>Expression Palette</i>		

### Παράδειγμα: 9-6

Να υπολογισθούν τα όρια  $\lim_{x \rightarrow 3} (x^2 + 2x + 1)$  και  $\lim_{x \rightarrow \pi} \cos x$ .

**Λύση:**

Τα όρια είναι:

```
> limit (x^2+2*x+2, x=3) ;
```

17

```
> limit (cos (x) , x=Pi) ;
```

-1

□

**Παράδειγμα: 9-7**

Να υπολογισθεί το όριο  $\lim_{x \rightarrow 0} \frac{|x|}{x}$ .

**Λύση:**

Ορίζουμε τη συνάρτηση:

```
> f := x -> abs (x) / x ;
```

$$f := x \rightarrow \frac{|x|}{x}$$

Το όριο θα είναι:

```
> limit (f (x) , x=0) ;
```

*undefined*

Το παραπάνω όριο, όπως είναι γνωστό, δεν υπάρχει. Θέτοντας τώρα ως παραμέτρους το `left`, `right` θα αναζητήσουμε τα πλευρικά όρια.

Το από δεξιά όριο  $\lim_{x \rightarrow 0^+} \frac{|x|}{x}$  είναι:

```
> limit (f (x) , x=0, left) ;
```

-1

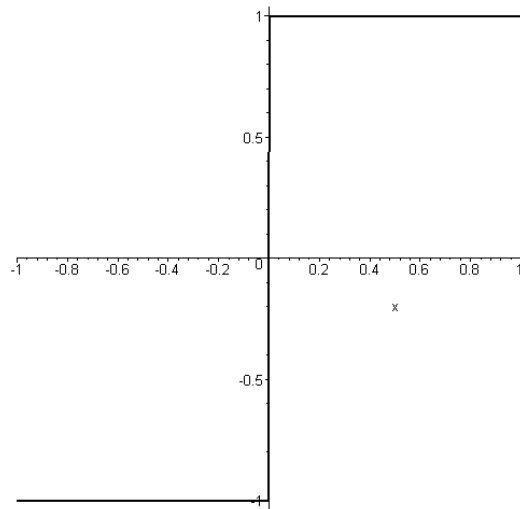
Το από αριστερά όριο  $\lim_{x \rightarrow 0^-} \frac{|x|}{x}$  είναι:

```
> limit (f (x) , x=0, right) ;
```

1

Το όριο αυτό βέβαια δεν υπάρχει. Αυτό μπορούμε να το δούμε και από τη γραφική παράσταση της συνάρτησης.

```
> plot (f (x) , x=-1..1) ;
```



□

### Παράδειγμα: 9-8

Να υπολογισθούν τα όρια  $\lim_{x \rightarrow \infty} \frac{\sin(x)}{x}$  και  $\lim_{x \rightarrow -\infty} \frac{\sin(x)}{x}$ .

#### Λύση:

Εδώ ζητάμε το όριο στο  $\pm\infty$ . Υπενθυμίζουμε ότι η εντολή του είναι `infinity`.

Ορίζουμε τη συνάρτηση:

```
> f:=x->sin(x)/x;
```

$$f := x \rightarrow \frac{\sin(x)}{x}$$

```
> limit (f (x) , x=infinity) ;
```

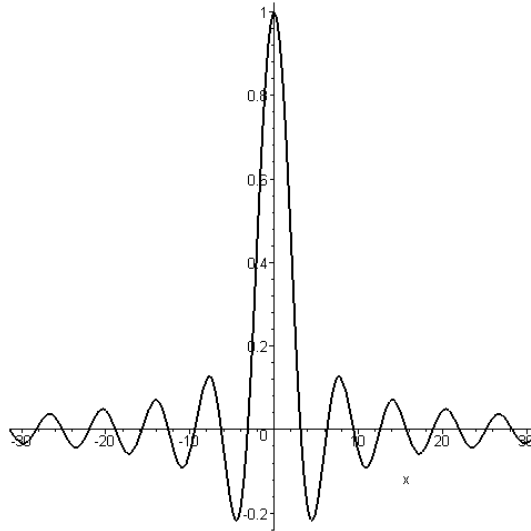
0

```
> limit (f (x) , x=-infinity) ;
```

0

Το αποτέλεσμα αυτό μπορούμε να το δούμε σχεδιάζοντας τη γραφική παράσταση της συνάρτησης.

```
> plot(f(x), x=-10*Pi..10*Pi);
```



□

### Παρατήρηση: 9-1

Αν η εντολή `Limit` γραφεί με το γράμμα `L` κεφαλαίο, τότε απλώς τυπώνεται το όριο χωρίς να υπολογίζεται. Στη συνέχεια μπορεί να υπολογιστεί το όριο με τη χρήση της εντολής `value()`.

### Παράδειγμα: 9-9

Ας δούμε ορισμένα όρια χρησιμοποιώντας τις εντολές `Limit` και `limit`.

#### Λύση:

Αν θέλουμε να παρουσιάσουμε ένα όριο χωρίς να υπολογιστεί η τιμή του, τότε:

```
> Limit((x+3)/(x^2-2), x=2);
```

$$\lim_{x \rightarrow 2} \frac{x+3}{x^2-2}$$

Με την εντολή `value` μπορούμε να ζητήσουμε την τιμή του παραπάνω ορίου.

```
> value(%);
```

$$\frac{5}{2}$$

Επίσης, μπορούμε να έχουμε και την προσεγγιστική τιμή του, ως εξής:

```
> evalf(%);
```

2.500000000

Κάνοντας χρήση της εντολής `Limit` μπορούμε να δημιουργήσουμε μαθηματικές εκφράσεις της μορφής:

```
> Limit((x^2-4)/(x-2), x=2)=limit((x^2-4)/(x-2), x=2);
```

$$\lim_{x \rightarrow 2} \frac{x^2 - 4}{x - 2} = 4$$

```
> Limit((1+Pi/x)^x, x=infinity):
```


```
% = value(%);
```

$$\lim_{x \rightarrow \infty} \left(1 + \frac{\pi}{x}\right)^x = e^\pi$$

□

## 9.2 Παραγωγήιση

Για τον υπολογισμό της παραγώγου μιας έκφρασης διαθέτουμε την εντολή `diff`.

Περιγραφή:	Υπολογίζει την παράγωγο μιας έκφρασης.	
Γενική σύνταξη:	<code>diff(expr, x\$n)</code>	
Παράμετροι:	expr:	Η προς παραγωγήιση έκφραση.
	x:	Η μεταβλητή ως προς την οποία παραγωγίζουμε.
	n:	Η τάξη της παραγώγου.
Expression Palette		



**Παράδειγμα: 9-10**

Να βρεθεί η πρώτη και η δεύτερη παράγωγος του  $\cos(x)$ .

**Λύση:**

Η πρώτη παράγωγος είναι:

```
> diff(cos(x), x);
```

$$-\sin(x)$$

και η δεύτερη:

```
> diff(cos(x), x$2);
```

$$-\cos(x)$$

□

**Παρατήρηση: 9-2**

Αν η εντολή `Diff` γραφεί με `D` κεφαλαίο, τότε απλώς τυπώνεται το μαθηματικό σύμβολο της παραγώγου χωρίς να υπολογίζεται. Στη συνέχεια μπορεί να υπολογιστεί το όριο με τη χρήση της εντολής `value()`.

```
> Diff(cos(x), x);
```

$$\frac{d}{dx} \cos(x)$$

```
> value(%);
```

$$-\sin(x)$$

Έτσι, μπορούμε να δημιουργήσουμε εκφράσεις της μορφής:

```
> Diff(cos(x), x) = diff(cos(x), x);
```

$$\frac{d}{dx} \cos(x) = -\sin(x)$$

Το Maple μας παρέχει ακόμα μια εντολή για τον υπολογισμό της παραγώγου, την εντολή `D`.

<b>Περιγραφή:</b>	<b>Υπολογίζει την παράγωγο μιας συνάρτησης και επιστρέφει συνάρτηση.</b>
<b>Γενική σύνταξη:</b>	<b>D(συνάρτηση)</b>

**Παράδειγμα: 9-11**

Να υπολογιστούν η πρώτη, η δεύτερη και η τρίτη παράγωγος της συνάρτησης  $f(x) = x^4$ .

**Λύση:**

Ορίζουμε τη συνάρτηση  $f$ .

```
> f := x -> x^4;
```

$$f := x \rightarrow x^4$$

Η συνάρτηση της πρώτης παραγώγου είναι:

```
> D(f);
```

$$x \rightarrow 4x^3$$

Οι συναρτήσεις της δεύτερης και τρίτης παραγώγου είναι:

```
> (D@@2)(f);
```

$$x \rightarrow 12x^2$$

```
> (D@@3)(f);
```

$$x \rightarrow 24x$$

□

**Παρατήρηση: 9-3**

Η διαφορά των εντολών  $D$  και  $\text{diff}$  είναι ότι η εντολή  $D$  υπολογίζει την παράγωγο μιας συνάρτησης και επιστρέφει συνάρτηση, ενώ η εντολή  $\text{diff}$  υπολογίζει την παράγωγο μιας έκφρασης και επιστρέφει έκφραση. Οι δύο αυτές εντολές μπορούν να δώσουν το ίδιο αποτέλεσμα, αν το αποτέλεσμα της εντολής  $\text{diff}$  το μετατρέψουμε σε συνάρτηση με την εντολή  $\text{unapply}$ , όπως θα δούμε στο παρακάτω παράδειγμα.

**Παράδειγμα: 9-12**

Δίνεται η συνάρτηση  $f(x) = x \ln(x)$ . Να βρεθεί το  $f'(2)$ .

**Λύση:**

Ορίζουμε τη συνάρτηση  $f$ .

```
> f := x -> x * ln(x);
```

$$f := x \rightarrow x \ln(x)$$

Ορίζουμε τη συνάρτηση της παραγώγου και την ονομάζουμε  $g$  (εδώ για τον ορισμό της δε χρησιμοποιούμε την εντολή `unapply`, αλλά χρησιμοποιούμε τα σύμβολα `:=`, αφού η  $D(f)$  είναι ήδη συνάρτηση).

>  $g := D(f)$  ;

$$g := x \rightarrow \ln(x) + 1$$

Η τιμή της παραγώγου στο 2 θα είναι:

>  $g(2)$  ;

$$\ln(2) + 1$$

Ισοδύναμα χρησιμοποιώντας την εντολή `diff` μπορούμε να δημιουργήσουμε τη συνάρτηση της πρώτης παραγώγου.

> `diff(f(x), x)` ;

$$\ln(x) + 1$$

Δημιουργούμε τη συνάρτηση  $g$  με τύπο την πρώτη παράγωγο της  $f$ .

>  $g := \text{unapply}(\%, x)$  ;

$$g := x \rightarrow \ln(x) + 1$$

Η τιμή της παραγώγου στο 2 θα είναι:

>  $g(2)$  ;

$$\ln(2) + 1$$

□

### Παράδειγμα: 9-13

Δίνεται η συνάρτηση  $f(x) = |x|$ . Να ορισθεί η συνάρτηση της πρώτης παραγώγου και να γίνει η γραφική της παράσταση .

#### Λύση:

Ορίζουμε τη συνάρτηση.

>  $f := x \rightarrow \text{abs}(x)$  ;

$$f := x \rightarrow |x|$$

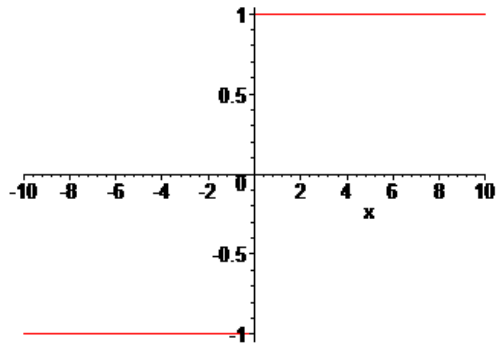
Ορίζουμε τη συνάρτηση της παραγώγου ως εξής:

```
> df := D(f);
```

$$df := x \rightarrow \text{abs}(1, x)$$

Ακολουθεί η γραφική παράσταση της παραγώγου της  $f$ , στην οποία βλέπουμε την ασυνέχεια στο 0.

```
> plot(df(x), x, discontin=true);
```



Αν βέβαια ζητήσουμε την τιμή της παραγώγου στο 0, θα έχουμε μήνυμα λάθους, αφού η συνάρτηση δεν είναι παραγωγίσιμη στο 0.

```
> df(0);
```

```
Error, (in simpl/abs) abs is not differentiable at 0
```

□

### 9.3 Ανάπτυγμα Taylor

Για την εύρεση του αναπτύγματος Taylor μιας συνάρτησης, το Maple διαθέτει την εντολή `taylor`.

<b>Περιγραφή:</b>	<i>Υπολογίζει την παράγωγο μιας έκφρασης.</i>	
<b>Γενική σύνταξη:</b>	<i>taylor(expr, x=a, n)</i>	
<b>Παράμετροι:</b>	expr:	Η έκφραση ή συνάρτηση που θα αναπτυχθεί.
	a	Το κέντρο της σειράς.
	n:	Η τάξη της σειράς.

**Παράδειγμα: 9-14**

Να βρεθεί η σειρά Taylor της συνάρτησης  $f(x) = e^x$  με κέντρο το 0 μέχρι 4<sup>ης</sup> τάξης.

**Λύση:**

> taylor (exp (x) , x=0 , 4) ;

$$1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + O(x^4)$$

□

**9.4 Ολοκλήρωση**

Ο υπολογισμός ενός ορισμένου ή αορίστου ολοκληρώματος γίνεται με την εντολή int.

<i>Περιγραφή:</i>	<i>Υπολογίζει το Ολοκλήρωμα μιας έκφρασης.</i>	
<i>Γενική σύνταξη:</i>	<i>int(expr, x=a...b)</i>	
<i>Παράμετροι:</i>	expr:	Η προς ολοκλήρωση έκφραση.
	x:	Η μεταβλητή ως προς την οποία ολοκληρώνουμε.
	a,b:	Τα άκρα της ολοκλήρωσης.
<i>Expression Palette</i>		

**Παράδειγμα: 9-15**

Να υπολογισθούν τα ολοκληρώματα  $\int \cos(x)dx$  ,  $\int_0^1 \frac{1}{x} dx$  ,  $\int_0^1 x^2 dx$  ,  $\int_1^\infty \sqrt{x} dx$  ,

$$\int_0^\infty \frac{1}{1+e^x} dx .$$

**Λύση:**

Το αόριστο ολοκλήρωμα  $\int \cos(x)dx$  είναι:

> int (cos (x) , x) ;

$$\sin(x)$$

Το ορισμένο ολοκλήρωμα  $\int_0^1 x^2 dx$  είναι:

`>int (x^2, x=0..1) ;`

$$\frac{1}{3}$$

Το καταχρηστικό ολοκλήρωμα  $\int_0^1 \frac{1}{x} dx$  είναι:

`>int (1/x, x=0..1) ;`

$\infty$

Το καταχρηστικό ολοκλήρωμα  $\int_1^{\infty} \sqrt{x} dx$  είναι:

`>int (sqrt (x) , x=1..infinity) ;`

$\infty$

Το καταχρηστικό ολοκλήρωμα  $\int_0^{\infty} \frac{1}{1+e^x} dx$  είναι:

`>int (1/(1+exp (x)) , x=0..infinity) ;`

$$\ln(2)$$

□

### Παράδειγμα: 9-16

Να υπολογισθεί το ολοκλήρωμα  $\int_a^2 \frac{1}{x} dx$ .

( το ολοκλήρωμα στην έκδοση 8 του Maple έδινε λάθος αποτέλεσμα)

#### Λύση:

Αν ζητήσουμε το ολοκλήρωμα αυτό από το Maple, δε μας επιστρέφει καμία απάντηση:

`>int (1/x, x=a..2) ;`

$$\int_a^2 \frac{1}{x} dx$$

Αν όμως χρησιμοποιήσουμε την παράμετρο `AllSolutions`, τότε:

```
> int( 1/x, x=a..2, AllSolutions );
```

$$\begin{cases} \text{undefined} & a < 0 \\ \infty & a = 0 \\ -\ln(a) + \ln(2) & 0 < a \end{cases}$$

□

### Παράδειγμα: 9-17

Να υπολογισθεί το ολοκλήρωμα  $\int x \ln(x) dx$ .

#### Λύση:

Αν θέσουμε το επίπεδο πληροφοριών που παρουσιάζει το Maple στο 5, θα δούμε πληροφορίες για τον τρόπο με τον οποίο το πρόγραμμα μπορεί να υπολογίσει ένα ολοκλήρωμα.

```
> restart;
```

```
> infolevel[int]:=5;
```

$$\text{infolevel}_{int} := 5$$

```
> int(x*ln(x), x);
```

```
int/indef1: first-stage indefinite integration
```

```
int/indef2: second-stage indefinite integration
```

```
int/ln: case of integrand containing ln
```

$$\frac{1}{2}x^2 \ln(x) - \frac{x^2}{4}$$

□

### Παρατήρηση: 9-4

Όπως και στις εντολές `limit` και `diff`, έτσι και στην εντολή για τον υπολογισμό ολοκληρωμάτων `int`, αν αυτή γραφεί με **I** κεφαλαίο, τότε απλώς τυπώνεται το όριο

χωρίς να υπολογίζεται. Στη συνέχεια μπορεί να υπολογιστεί το όριο με τη χρήση της εντολής `value()`.

```
> Int (cos (x) , x) ;
```

$$\int \cos(x) dx$$

```
> value (%) ;
```

$$\sin(x)$$

Έτσι και εδώ, μπορούμε να δημιουργήσουμε μαθηματικές εκφράσεις στις οποίες εμφανίζονται τα ολοκληρώματα με το συνηθισμένο μαθηματικό συμβολισμό τους.

```
> Int (cos (x) , x=1..2) =int (cos (x) , x=Pi..2*Pi) ;
```

$$\int_1^2 \cos(x) dx = 0$$

### Παρατήρηση: 9-5

Πολλά ολοκληρώματα δεν υπολογίζονται αναλυτικά και το Maple επιστρέφει ως αποτέλεσμα ειδικές συναρτήσεις, όπως:

Error Function -  $\text{erf}(x)$ , Euler's Constant,  $\gamma$ , Exponential Integral -  $\text{Ei}(x)$ , Elliptic Integral Function -  $\text{EllipticF}(k,z)$ , Gamma Function -  $\Gamma(x)$ , Zeta Function -  $\zeta(s)$ , Heaviside step function, Dirac Delta Function, Bessel and modified Bessel functions.

Στις περιπτώσεις αυτές, πρέπει να υπολογίσουμε το ολοκλήρωμα χρησιμοποιώντας αριθμητικές μεθόδους. Αυτό μπορεί να γίνει με την εντολή `evalf`.

<b>Περιγραφή:</b>	<b>Υπολογίζει το Ολοκλήρωμα μιας έκφρασης προσεγγιστικά.</b>	
<b>Γενική σύνταξη:</b>	<b><code>evalf(int(expr, x=a..b,method,digits,epsilon))</code></b>	
<b>Παράμετροι:</b>	<code>expr:</code>	Η προς ολοκλήρωση έκφραση.
	<code>x:</code>	Η μεταβλητή ως προς την οποία ολοκληρώνουμε.
	<code>a,b:</code>	Τα άκρα της ολοκλήρωσης.
	<code>method =</code>	Η μέθοδος αριθμητικής ολοκλήρωσης που θα χρησιμοποιηθεί. <i>Προεπιλογή:</i> η μέθοδος <code>hybrid symbolic-n</code>



		tolerance numeric.
	digits =	Αριθμός των ψηφίων που θα έχουμε στο αποτέλεσμα.
	epsilon =	Το σφάλμα $\text{eps} = 0.5 * 10^{(1-\text{digits})}$

Υπάρχουν διάφορες μέθοδοι για τον προσεγγιστικό υπολογισμό ενός ορισμένου ολοκληρώματος. Το ποια μέθοδο θα χρησιμοποιήσουμε μπορεί να οριστεί από την παράμετρο `method`.

### Παράδειγμα: 9-18

Να υπολογισθεί το ολοκλήρωμα  $\int_0^1 \frac{e^{-x^3}}{x^2+1} dx$ .

#### Λύση:

Αν ζητήσουμε το ολοκλήρωμα αυτό χρησιμοποιώντας την εντολή `int`, τότε:

```
> int( exp(-x^3)/(x^2+1), x = 0..1 );
```

$$\int_0^1 \frac{e^{-x^3}}{x^2+1} dx$$

Στο παραπάνω ολοκλήρωμα βλέπουμε ότι ως αποτέλεσμα παίρνουμε το ίδιο το ολοκλήρωμα. Αυτό σημαίνει ότι το Maple δεν μπορεί να βρει τη λύση σε κλειστή μορφή. Γι' αυτό το λόγο, για τον υπολογισμό αυτού του ορισμένου ολοκληρώματος, πρέπει να καταφύγουμε σε αριθμητική επίλυση, προσέγγιση, χρησιμοποιώντας την εντολή `evalf`.

```
> evalf( int( exp(-x^3)/(x^2+1), x = 0..1 ) );
```

0.6649369431

Αν ζητήσουμε το αποτέλεσμα με 20 ψηφία, έχουμε:

```
> evalf(int(exp(-x^3)/(x^2+1), x=0..1, digits=20));
```

0.66493694307392745613

□

**Παράδειγμα: 9-19**

Να υπολογισθεί το ορισμένο ολοκλήρωμα  $\int_0^1 \frac{1}{\sqrt{1+x^4}} dx$ .

**Λύση:**

```
> int( 1/sqrt( 1 + x^4 ) , x=0..1 );
```

$$\frac{1}{2} \text{EllipticK}\left(\frac{\sqrt{2}}{2}\right)$$

```
> evalf(%);
```

0.9270373385

□

**Παράδειγμα: 9-20**

Να υπολογισθεί το αόριστο ολοκλήρωμα  $\int \frac{1}{\sqrt{1-x^4}} dx$  και το ορισμένο

$$\int_0^1 \frac{1}{\sqrt{1-x^4}} dx.$$

**Λύση:**

```
> f:=x->1/sqrt(1-x^4);
```

$$f:=x \rightarrow \frac{1}{\sqrt{1-x^4}}$$

```
> int(f(x), x);
```

$$\frac{\sqrt{1-x^2} \sqrt{x^2+1} \text{EllipticF}(x, I)}{\sqrt{1-x^4}}$$

```
> int(f(x), x=0..1);
```

$$\frac{1}{4} \text{B}\left(\frac{1}{4}, \frac{1}{2}\right)$$

```
> evalf(%);
```

1.311028777

□