

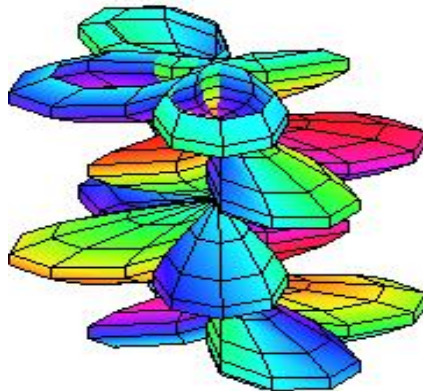


**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΑΤΡΑΣ
ΠΑΡΑΡΤΗΜΑ ΑΜΑΛΙΑΔΑΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΣΤΗ ΔΙΟΙΚΗΣΗ
ΚΑΙ ΣΤΗΝ ΟΙΚΟΝΟΜΙΑ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΕΚΜΑΘΗΣΗ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΕ MAPLE V

LEARNING AND PROGRAMMING IN MAPLE V



**ΙΟΥΛΙΑ ΑΓΓΕΛΑΚΟΠΟΥΛΟΥ
ΜΠΑΚΟΛΑ – ΜΠΑΚΟΛΟΥΛΗ ΓΛΥΚΕΡΙΑ**

ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΗΣ: ΔΡ. ΗΛΙΑΣ Κ. ΣΤΑΥΡΟΠΟΥΛΟΣ

ΑΜΑΛΙΑΔΑ, ΜΑΪΟΣ 2011

Περιεχόμενα

Περίληψη	v
Abstract	vi
Κεφάλαιο 1 Εισαγωγή	1
1.1. Ορισμός του Maple	1
1.2. Ιστορία και εξέλιξη του Maple	1
1.3. Η νέα κυκλοφορία του Maple 15	3
1.4. Βασικές λειτουργίες	7
1.5. Αρχιτεκτονική	8
1.6. Εφαρμογή του προγράμματος Maple σε άλλα προϊόντα	8
1.7. Σύγκριση του Maple με το Matlab και το Mathematica	9
Κεφάλαιο 2 Τα βασικά του Maple	13
2.1. Εκκίνηση Maple	13
2.2. Το παράθυρο του Maple	13
2.3. Η γραμμή του μενού (Menu Bar)	14
2.3.1. Η εργαλειοθήκη (Tool Bar)	20
2.3.2. Το Context Bar	23
2.3.3. Το φύλλο εργασίας (Worksheet)	24
2.3.4. Η γραμμή κατάστασης (Status bar)	26
2.4. Βασικές οδηγίες για την χρήση του προγράμματος	26
2.5. Διακοπή, επανεκκίνηση και άλλες ευκολίες	28
2.6. Σειρά των πράξεων	30
2.7. Μεταβλητές	30
2.8. Βασικές εντολές του Maple και παραδείγματα	32
Κεφάλαιο 3 Εξισώσεις	40
3.1. Εισαγωγή	40
3.2. Αλγεβρικές εξισώσεις	41
3.3. Συστήματα με περισσότερες από μια λύσεις	44
3.4. Σύνολα εξισώσεων (Sets)	47
3.5. Διαφορικές εξισώσεις	49
Κεφάλαιο 4 Εκφράσεις και Συναρτήσεις	51
4.1. Εκφράσεις	51
4.1.1. Αντικατάσταση (Substituting)	52

4.1.2. Κλασματικές δυνάμεις και μετασχηματισμοί	52
4.1.3. Ανάπτυξη (Expand)	53
4.1.4. Απλοποίηση και Παραγοντοποίηση	54
4.2. Εισαγωγή Συνάρτησης	57
4.3. Ενσωματωμένες συναρτήσεις.....	58
4.4. Παραγωγή και ολοκλήρωση συναρτήσεων	61
4.5. Δημιουργία έκφρασης από μια συνάρτηση	62
4.6. Δημιουργία συνάρτησης από μια έκφραση	62
4.7. Αντίστροφη συνάρτηση.....	63
Κεφάλαιο 5 Όρια, Παράγωγοι, Αθροίσματα και Ολοκληρώματα	65
5.1. Όρια	65
5.1.1. Όρια όταν το x τείνει σε έναν αριθμό	66
5.1.2. Όρια όταν το x τείνει στο $+\infty$ και ή στο $-\infty$	66
5.1.3. Όρια από αριστερά και από δεξιά.....	67
5.2. Παραγωγή.....	67
5.2.1. Παραγωγή μιας έκφρασης	69
5.2.2. Παραγωγή μιας συνάρτησης	71
5.3. Υπολογισμός Αθροισμάτων	72
5.4. Ολοκληρώματα	74
Κεφάλαιο 6 Πίνακες και Διανύσματα	77
6.1. Εισαγωγή πινάκων	77
6.2. Διανύσματα	81
6.3. Επίλυση γραμμικού συστήματος.....	82
6.4. Επίλυση προβλημάτων με διανύσματα	84
Κεφάλαιο 7 Γραφήματα.....	88
7.1. Εισαγωγή	88
7.2. Λίστες.....	88
7.3. Σχεδίαση Γραφημάτων	89
7.3.1. Σχεδίαση απλής έκφρασης.....	90
7.3.2. Επιλέγοντας το y -διάστημα	91
7.3.3. Κάθετες ασύμπτωτες.....	91
7.3.4. Τοπικά Ακρότατα.....	92
7.4. Παράμετροι της εντολής plot.....	92
7.4.1. Στύλ.....	93

7.4.2. Αριθμός σημείων και κλίμακα	93
7.4.3. Ασυνέχεια	94
7.4.4. Προσθήκη τίτλου και κειμένου σε γράφημα	95
7.4.5. Χρώμα και πάχος γραμμής(color and thickness).....	96
7.4.6. Το μενού plot, η εργαλειοθήκη και το context bar	96
7.5. Γενικά παραδείγματα	98
7.5.1. Σχεδιάζοντας πολλές εκφράσεις σε ένα γράφημα	98
7.5.2. Σχεδίαση γραφήματος μιας εξίσωσης	99
7.5.3. Σχεδίαση δύο συναρτήσεων σε γραφική παράσταση	100
7.5.4. Γραφική παράσταση σημείων και γραμμών που ενώνουν τα σημεία δύο ξεχωριστών γραφικών παραστάσεων.	101
7.5.5. Χάραξη δυο γραφικών παραστάσεων στους ίδιους άξονες, χρησιμοποιώντας και λεζάντα	102
7.6. Σχεδίαση Τρισδιάστατων Γραφημάτων	102
7.6.1. Σχεδίαση γραφικής παράστασης μιας επιφάνειας.....	104
7.6.2. Σχεδίαση γραφικής παράστασης μιας συνάρτησης	104
7.7. Κινούμενα Γραφήματα (animation)	105
7.7.1. Κινούμενη απεικόνιση μιας παραμετρικής καμπύλης.....	106
7.7.2. Κινούμενη τρισδιάστατη αναπαράσταση.....	107
Κεφάλαιο 8 Προγραμματισμός – Δομές Ελέγχου Ροής.....	108
8.1. Εισαγωγή	108
8.2. Χειρισμός Αλφαριθμητικών	109
8.3. Εντολές επανάληψης	110
8.3.1. Η δομή for.....	110
8.3.2. Η δομή while	114
8.4. Εντολές ελέγχου	118
8.4.1. Η δομή if	118
Κεφάλαιο 9 Τα πακέτα (packages) του Maple.....	122
9.1. Εισαγωγή	122
9.2. Το πακέτο γραμμικής άλγεβρας LinearAlgebra.....	125
9.3. Βελτιστοποίηση συναρτήσεων	127
Βιβλιογραφία	129

Κατάλογος Σχημάτων

ΣΧΗΜΑ 1: ΤΟ ΕΙΚΟΝΙΔΙΟ ΤΟΥ MAPLE	13
ΣΧΗΜΑ 2: ΤΟ ΠΑΡΑΘΥΡΟ ΤΟΥ MAPLE	14
ΣΧΗΜΑ3: MENUBAR	14
ΣΧΗΜΑ 4: TOOLBAR	20
ΣΧΗΜΑ5: CONTEXT BAR	23
ΣΧΗΜΑ 6: ΜΠΑΡΑ ΓΡΑΦΗΣ.....	23
ΣΧΗΜΑ 7: ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ.....	24
ΣΧΗΜΑ 8: ΠΑΡΑΘΥΡΟ ΑΠΟΘΗΚΕΥΣΗΣ.....	25
ΣΧΗΜΑ 9: STATUSBAR	26
ΣΧΗΜΑ 10: ΓΡΑΦΗΜΑ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ $F(x)=x^3-x$	38
ΣΧΗΜΑ 11: ΓΡΑΦΗΜΑ ΤΡΙΣΔΙΑΣΤΑΤΗΣ ΑΠΕΙΚΟΝΙΣΗΣ.....	38
ΣΧΗΜΑ 12: ΣΧΕΔΙΑΣΗ ΔΥΟ ΣΥΝΑΡΤΗΣΕΩΝ.....	39
ΣΧΗΜΑ 13: ΣΧΕΔΙΑΣΗ ΤΗΣ ΕΚΦΡΑΣΗΣ $E^{(-2x)}$	90
ΣΧΗΜΑ 14: ΣΧΕΔΙΑΣΗ ΤΗΣ ΑΠΛΗΣ ΕΚΦΡΑΣΗΣ $(x-2)^{(2/3)}$	91
ΣΧΗΜΑ 15: ΣΧΕΔΙΑΣΗ ΤΗΣ ΕΚΦΡΑΣΗΣ $E^{(-x)} \sin(3x)$	91
ΣΧΗΜΑ 16: ΣΧΕΔΙΑΣΗ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ $2x^2+1/x^2-1$	92
ΣΧΗΜΑ 17: ΣΧΕΔΙΑΣΗ ΣΥΝΑΡΤΗΣΗΣ x^3-49x	92
ΣΧΗΜΑ 18: ΣΧΕΔΙΑΣΗ ΤΗΣ ΠΑΡΑΣΤΑΣΗΣ $\sin(x)$ ΜΕ STYLE.....	93
ΣΧΗΜΑ 19: ΣΧΕΔΙΑΣΗ ΤΗΣ ΠΑΡΑΣΤΑΣΗΣ $\sqrt{1-x^2}$ ΜΕ NUMPOINTS	93
ΣΧΗΜΑ 20: ΣΧΕΔΙΑΣΗ ΤΗΣ ΠΑΡΑΣΤΑΣΗΣ $\sqrt{1-x^2}$ ΜΕ SCALLING	94
ΣΧΗΜΑ 21: ΓΡΑΦΗΜΑ ΤΗΣ $\sqrt{1-x^2}$ ΜΕ ΚΛΙΜΑΚΕΣ ΜΕΤΡΗΣΗΣ	94
ΣΧΗΜΑ 22: ΣΧΕΔΙΑΣΗ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ $2x^2+1/(x^2-1)$	95
ΣΧΗΜΑ 23: ΠΡΟΣΘΗΚΗ ΤΙΤΛΟΥ ΣΤΗΝ ΠΑΡΑΣΤΑΣΗ $2x^2+1/(x^2-1)$	95
ΣΧΗΜΑ 24: ΓΡΑΦΗΜΑ ΤΗΣ $E^{(-x)} \sin(3x)$ ΜΕ ΠΡΟΣΘΗΚΗ ΚΕΙΜΕΝΟΥ.....	96
ΣΧΗΜΑ 25: ΓΡΑΦΗΜΑ ΤΗΣ $\sin(x)$ ΜΕ ΧΡΩΜΑ ΚΑΙ ΠΑΧΟΣ ΓΡΑΜΜΗΣ	96
ΣΧΗΜΑ 26: ΓΡΑΦΗΜΑ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ $G(x)=\sqrt{1-x^2}$	97
ΣΧΗΜΑ 27: ΣΧΕΔΙΑΣΗ ΤΗΣ $G(x)=\sqrt{1-x^2}$ ΑΠΟ ΤΟ CONTEXTBAR	97
ΣΧΗΜΑ 28: ΓΡΑΦΗΜΑ ΤΩΝ $\sin(x)$ ΚΑΙ $\sin(2x)$	98
ΣΧΗΜΑ 29: ΣΧΕΔΙΑΣΗ ΤΩΝ $\sin(x)$ ΚΑΙ $\sin(2x)$ ΜΕ ΛΙΣΤΑ ΧΡΩΜΑΤΩΝ.....	98
ΣΧΗΜΑ 30: ΣΧΕΔΙΑΣΗ ΤΗΣ $x^3+y^3=6xy$ ΜΕ ΤΗΝ IMPLICITPLOT	99
ΣΧΗΜΑ 31: ΣΧΕΔΙΑΣΗ ΤΗΣ $x^3+y^3=6xy$ ΜΕ ΤΗΝ IMPLICITPLOT ΚΑΙ GRID.....	100
ΣΧΗΜΑ 32: ΣΧΕΔΙΑΣΗ ΤΩΝ x^2 ΚΑΙ $4-x^2$ ΣΕ ΜΙΑ ΓΡΑΦΙΚΗ ΠΑΡΑΣΤΑΣΗ	101
ΣΧΗΜΑ 33: ΣΧΕΔΙΑΣΗ ΓΡΑΦΗΜΑΤΟΣ ΜΕ POINTPLOT	101
ΣΧΗΜΑ 34: ΣΧΕΔΙΑΣΗ ΓΡΑΦΗΜΑΤΟΣ ΕΝΩΝΟΝΤΑΣ ΤΑ POINTPLOTS.....	102
ΣΧΗΜΑ 35: ΣΧΕΔΙΑΣΗ ΤΩΝ $\sin(x)$ ΚΑΙ $\cos(x)$ ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΛΕΖΑΝΤΑ.....	102
ΣΧΗΜΑ 36: ΣΧΕΔΙΑΣΗ ΤΡΙΣΔΙΑΣΤΑΤΟΥ ΓΡΑΦΗΜΑΤΟΣ	103
ΣΧΗΜΑ 37: ΣΧΕΔΙΑΣΗ ΤΡΙΣΔΙΑΣΤΑΤΟΥ ΓΡΑΦΗΜΑΤΟΣ ΜΕ ΑΞΟΝΕΣ ΣΕ ΜΟΡΦΗ ΚΟΥΤΙΟΥ	103
ΣΧΗΜΑ 38: ΣΧΕΔΙΑΣΗ ΤΡΙΣΔΙΑΣΤΑΤΟΥ ΓΡΑΦΗΜΑΤΟΣ ΑΠΟ ΤΟ CONTEXTBAR.....	103
ΣΧΗΜΑ 39: ΣΧΕΔΙΑΣΗ ΓΡΑΦΙΚΗΣ ΠΑΡΑΣΤΑΣΗΣ ΜΙΑΣ ΕΠΙΦΑΝΕΙΑΣ.....	104
ΣΧΗΜΑ 40: ΣΧΕΔΙΑΣΗ ΤΗΣ x^2-y^2 ΣΕ ΜΟΡΦΗ ΠΛΕΓΜΑΤΟΣ.....	104
ΣΧΗΜΑ 41: ΣΧΕΔΙΑΣΗ ΚΙΝΟΥΜΕΝΟΥ ΓΡΑΦΗΜΑΤΟΣ.....	105
ΣΧΗΜΑ 42: ANIMATION ΜΕΝΟΥ.....	105
ΣΧΗΜΑ 43: ΚΙΝΟΥΜΕΝΗ ΑΠΕΙΚΟΝΙΣΗ ΜΙΑΣ ΠΑΡΑΜΕΤΡΙΚΗΣ ΚΑΜΠΥΛΗΣ	106
ΣΧΗΜΑ 44: ΣΧΕΔΙΑΣΗ ΤΗΣ ΕΦΑΠΤΟΜΕΝΗΣ $y=x^2$	106
ΣΧΗΜΑ 45: ΚΙΝΟΥΜΕΝΗ ΤΡΙΣΔΙΑΣΤΑΤΗ ΑΝΑΠΑΡΑΣΤΑΣΗ	107
ΣΧΗΜΑ 46: ΠΕΡΙΟΔΙΚΗ ΕΠΕΚΤΑΣΗ ΤΗΣ $G(x)=x^2$	118
ΣΧΗΜΑ 47: ΓΡΑΦΗΜΑ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ $(F,-1...1)$	120
ΣΧΗΜΑ 48: ΓΡΑΦΗΜΑ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ $(G,-3..3*\pi+3)$	121

Περίληψη

Το Maple είναι ένα ισχυρό σύστημα επίλυσης μαθηματικών προβλημάτων και οπτικής αναπαράστασης των λύσεων, με ευρεία χρήση στην εκπαίδευση, έρευνα και βιομηχανία.

Το βασικό στοιχείο που καθιστά το Maple τόσο ισχυρό εργαλείο είναι η χρησιμοποίηση βιβλιοθηκών που επιτρέπουν συμβολικούς υπολογισμούς. Οι βιβλιοθήκες του αποτελούνται από ρουτίνες για συγκεκριμένες εφαρμογές, καθώς και από παραδείγματα ενδεικτικής χρήσης αυτών. Η ευκολία στη μάθηση και στη χρήση του έχουν κάνει το Maple ένα σημαντικό εργαλείο τόσο ισχυρό όσο αρκετά άλλα κατ'εξοχήν εργαλεία προγραμματισμού.

Στην παρούσα πτυχιακή εργασία παρουσιάζουμε τα βασικά χαρακτηριστικά και λειτουργίες του Maple 8. Συγκεκριμένα, αναφερόμαστε σε απλούς αριθμητικούς υπολογισμούς, επίλυση εξισώσεων, στοιχειώδεις και ειδικές συναρτήσεις, στοιχεία διαφορικού και ολοκληρωτικού λογισμού, γραμμική άλγεβρα, βελτιστοποίηση συναρτήσεων, καθώς και παρουσίαση δυσδιάστατων και τρισδιάστατων γραφικών παραστάσεων και κίνηση. Τέλος αναφερόμαστε στις διαθέσιμες βιβλιοθήκες (πακέτα) καθώς και σε βασικές προγραμματιστικές δομές του Maple 8 για την υλοποίηση και εκτέλεση αλγορίθμων.

Abstract

Maple is a powerful system for solving mathematical problems and visualization solutions, widely used in education, research and industry.

That it makes Maple a so powerful tool is the use of libraries that allow symbolic calculations. Maple libraries consist of routines for specific applications and indicative working examples. Convenience in learning and using Maple, make it a significant tool like several programming tools.

In this thesis we present the basic features and functions of Maple 8. More specific, we concern with simple arithmetic calculations, solving equations, elementary and special functions, differential and integral calculus, linear algebra, optimization, two-dimensional and three-dimensional graphical representation and animation. Finally, we refer to available libraries (packages) and to basic programming structure of Maple 8 for the implementation and execution of algorithms.

Κεφάλαιο 1

Εισαγωγή

Στο κεφάλαιο που ακολουθεί θα κάνουμε μια πρώτη γνωριμία με το πρόγραμμα MapleV,θα παρουσιάσουμετην ιστορική του εξέλιξη, τις βασικές του λειτουργίες και μια σύντομη σύγκριση του με αντίστοιχα προγράμματα.

1.1. Ορισμός του Maple

Το Maple είναι ένα πολυδύναμο εμπορικό σύστημα αλγεβρικών υπολογισμώνγια τοσύγχρονο μηχανικό,μαθηματικόκαιεπιστήμονα.Παρέχει τη δυνατότηταγια γρήγορους υπολογισμούς, ανάπτυξη σχεδίων, διδασκαλία βασικώνεννοιών ή παραγωγή σύνθετων μοντέλων εξομοίωσης υψηλής ευκρίνειας. Ο χρήστης μπορεί να εισάγει τα δεδομένα με αριθμητικά σύμβολα καθώςκαι να προσαρμόσει τις εφαρμογές στις απαιτήσεις του. Παρέχεται ηδυνατότητα για αριθμητικούς υπολογισμούς, με ακρίβεια, καθώςκαι για συμβολικούς υπολογισμούς και οπτικοποίηση. Το πρόγραμμαενσωματώνει μια γλώσσα προγραμματισμού που έχει δυναμικό χαρακτήραυπό τημορφή εντολών.

1.2. Ιστορία και εξέλιξη του Maple

Η πρώτη έκδοση του Mapleαναπτύχθηκε το Νοέμβριο του 1980 από το SymbolicComputationGroupστο Πανεπιστήμιο του Waterloo.Το όνομά του προέκυψε από το φύλλο του σφενδάμου που απεικονίζεται στην Καναδική σημαία.Η αρχική ανάπτυξη του Maple προχώρησε πολύ γρήγορα, με την πρώτη περιορισμένη έκδοση που εμφανίζεται τον Δεκέμβριο του 1980.Οι ερευνητές προσπάθησαν και συνεχώς απέρριπταν πολλές διαφορετικές ιδέες δημιουργώντας ένα συνεχώς εξελισσόμενο σύστημα.

Μέχρι το τέλος του 1983, πάνω από 50 πανεπιστήμια είχαν αντίγραφα του Maple εγκατεστημένα στον υπολογιστή τους.Το 1988, άρχισενα αναπτύσσεται και να

πωλείται στο εμπόριο από την εταιρία WaterlooMaple, πλέον γνωστή ως Maplesoft¹), μια Καναδική εταιρεία με έδρα το Waterloo.

Το 1989, η πρώτη γραφική διεπαφή(interface) για το Maple αναπτύχθηκε και περιλαμβάνεται με την έκδοση 4.3 για Macintosh². Οι προηγούμενες εκδόσεις του Maple για X11 και Windows περιελάμβαναν μόνο μια διασύνδεση γραμμής εντολών. Ακολούθησε το 1990 το MapleV, μια από τις εκδόσεις ορόσημο για την εποχή της, η οποία αποτέλεσε την τρέχουσα έκδοση για περίπου μία δεκαετία. Το 1999, κυκλοφόρησε το Maple 6 που περιελάμβανε Αριθμητικές Βιβλιοθήκες NAG που επέτρεπαν αριθμητικούς υπολογισμούς με αυθαίρετη αριθμητική ακρίβεια. Στις 16 Απριλίου του 2002 ανακοινώθηκε το Maple 8 που αποτέλεσε μία από τις πιο σημαντικές κυκλοφορίες στη μακρά ιστορία του προϊόντος, με έναν εκτεταμένο κατάλογο νέων χαρακτηριστικών. Το σημερινό γραφικό περιβάλλον εισήχθη το 2003 με την έκδοση Maple 9. Η έκδοση αυτή ήταν κατά κύριο λόγο γραμμένη σε Java³ γεγονός που την έκανε αρκετά αργή.

Το χρονικό διάστημα μεταξύ του 1995 και του 2005 το Maple έχασε σημαντικό μερίδιο αγοράς από τους ανταγωνιστές λόγω της περιορισμένης χρηστικότητας της γραφικής διεπαφής. Το 2005, το Maple 10 εισήγαγε ένα νέο έγγραφο, ως μέρος του προτύπου διεπαφής. Το κύριο χαρακτηριστικό αυτού του τρόπου είναι ότι τα μαθηματικά είναι εγγεγραμμένα με τη χρήση δύο διαστάσεων των εισροών, έτσι ώστε να μοιάζει με τύπους σε ένα βιβλίο.

Το 2008, στο Maple 12 προσθέτονται επιπλέον χαρακτηριστικά στην γραφική διεπαφή, δανεισμένα από το Mathematica⁴, συμπεριλαμβανομένων των ειδικών φύλλων στυλ, τον έλεγχο των κεφαλίδων και υποσέλιδων που ταιριάζουν στις περιοχές εκτέλεσης, την ολοκλήρωση εντολής σε πρότυπα, τον έλεγχο σύνταξης και αυτόματης αρχικοποίησης σε περιοχές. Επιπλέον χαρακτηριστικά προστέθηκαν για την ευκολότερη χρήση ως μια εργαλειοθήκη Matlab⁵.

¹<http://www.maplesoft.com>

²Κοινώς γνωστό και ως Mac είναι το εμπορικό όνομα μιας σειράς προσωπικών υπολογιστών. που σχεδιάζονται, αναπτύσσονται και προωθούνται από την εταιρεία Apple Inc. (<http://www.apple.com>)

³Αντικειμενοστραφής γλώσσα προγραμματισμού. από την εταιρεία πληροφορικής Sun Microsystems. (<http://www.oracle.com/technetwork/java/index.html>)

⁴Μια σύντομη περιγραφή του Mathematica δίνεται στην ενότητα 1.7.

⁵Μια σύντομη περιγραφή του Matlab δίνεται στην ενότητα 1.7.

Τον Σεπτέμβριο του 2009 το Maple και Maplesoft αποκτήθηκαν από την Ιαπωνική CybernetSystems, εταιρία λογισμικού για λιανική πώληση. Τέλος η τρέχουσα κύρια έκδοση είναι η έκδοση 15 που κυκλοφόρησε τον Μάιο του 2011. Για περισσότερες πληροφορίες μπορούμε να ανατρέξουμε στο (Maple).

1.3. Η νέα κυκλοφορία του Maple 15

Στις 13 Απριλίου 2011η Maplesoftανακοίνωσε μια σημαντική νέα έκδοση του Maple 15η οποία προσφέρει πρωτοφανή επίπεδα για νέες υπολογιστικές ικανότητες. Το Maple 15, εκτός από τα μαθηματικά, εφαρμόζεται και σετομείς όπως ο έλεγχος του σχεδιασμού, φυσική, στατιστική, οικονομικά και παραγωγή κώδικα. Με πάνω από 270 νέες μαθηματικές λειτουργίες και εκατοντάδες βελτιώσεις σε υφιστάμενους αλγόριθμους, οι χρήστες μπορούν να επιλύσουν πιο πολύπλοκα προβλήματα γρηγορότερα από ποτέ.Επιπλέον, διαθέτει 49 νέες διαδραστικέςμικροεφαρμογές που δείχνουν κοινά θέματα σε όλα τα μαθηματικά για την επίλυση προβλημάτων χωρίς την ανάγκη να θυμόμαστε τις εντολές ή να ξέρουμε τη σύνταξη.

Το Maple 15 δεν υστερεί, όσον αφορά την επεκτασιμότητα και την απόδοση, λόγω της υψηλής του απόδοσης. Μπορεί να υπολογίζει επίσης συμβολικά λύσεις διαφορικών εξισώσεων που άλλο σύστημα δεν μπορεί να χειριστεί, με 96% απόδοση του δείκτη αναφοράς για τις διαφορικές εξισώσεις.

Οι χρήστες μπορούν να εργάζονται αποτελεσματικά με μεγάλα σύνολα δεδομένων και να δημιουργούν ακόμη πιο καθοριστικέςδιαδραστικές εφαρμογές χρησιμοποιώντας το νέο πίνακα δεδομένων. Επίσης έχουν επέλθει σημαντικές βελτιώσειςστα προγραμματιστικά εργαλεία, στα τεχνικά χαρακτηριστικά τεκμηρίωσης καιστα χαρακτηριστικά συνδεσιμότητας.Η αποτελεσματικότητα πολλών αλγορίθμων έχει δει τεράστιες βελτιώσεις, καθώς και το εύρος και το βάθος των υπολογισμών σε τομείς όπως η διαφορική γεωμετρία έχουν εξελιχθεί περισσότερο.

Στη συνέχεια περιγράφουμε τις νέες, αναφορικά με τις προηγούμενες εκδόσεις, δυνατότητες της έκδοσης Maple 15 (Maplesoft).

Παράλληλη απόδοση(ParallelPerformance)

Το Maple 15 προσφέρει πολλές επιλογές για να εκμεταλλευτεί παράλληλους υπολογισμούς από πολυπύρηνες (multi-core)μηχανές για υπολογισμούς μεγάλης κλίμακας, που μας επιτρέπει να αντιμετωπίσουμε μεγαλύτερα προβλήματα πιο γρήγορα.Στην παράλληλη απόδοση συμπεριλαμβάνονται:

- **Αυτόματος παραλληλισμός(AutomaticParallelism):**τοMaple εκμεταλλεύεται την πλήρη ισχύ επεξεργασίας του υπολογιστή μας, αυτόματα ανιχνεύει και χρησιμοποιεί όλους τους διαθέσιμους πυρήνες του επεξεργαστή για να εκτελέσει πολλούς υπολογισμούς παράλληλα. Δεν χρειάζεται να κάνει κανένα πρόσθετο προγραμματισμό, να αλλάξει καμία επιλογή, ή ακόμα και αν ξέρουμε πόσους πυρήνες έχει ο υπολογιστής μας. Στο Maple 15, πολλές θεμελιώδεις δραστηριότητες μπορούν να γίνουν παράλληλα, έτσι ώστε τα αποτελέσματά μας είναι διαθέσιμα πολύ πιο γρήγορα και να μπορούμε να αντιμετωπίσουμε μεγαλύτερα προβλήματα.
- **Πολλές διαδικασίες προγραμματισμού για ένα τοπικό δίκτυο(Multi-processProgrammingonaLocalGrid):**μας επιτρέπει να ξεκινήσουμε πολλαπλές υπολογιστικές διεργασίες ήδη από το επίπεδο του χρήστη, χωρίς την ανάγκη για οποιαδήποτε προγενέστερη εγκατάσταση ή διοίκηση.
- **Πολυνηματικόςπρογραμματισμός(MultithreadedProgramming):**είναι το μόνο τεχνικό υπολογιστικό σύστημα που μας επιτρέπει να επωφεληθούμε από multithreading στα προγράμματα μας. Η γλώσσα προγραμματισμού του Maple προσφέρει άμεση πρόσβαση στην έναρξη και τον έλεγχο των θεμάτων. Επιπλέον, το Maple παρέχει επίσης μια εργασία με βάση το μοντέλο προγραμματισμού που απλοποιεί τη διαχείριση.
- **Υπολογισμός κλίμακας(GridComputing):**είναι ένα πακέτο που επιτρέπει να αναπτύξουμε τα προγράμματά μας παράλληλα, με μεγάλης κλίμακας υπολογισμούςσυσπειρώσεις με πλήρη αξιοποίηση όλων των διαθέσιμων επεξεργαστών ισχύς για την αντιμετώπιση των πολύ μεγάλων προβλημάτων.
- **Cuda υποστήριξη(CUDASupport):**το Mapleέχει το πλεονέκτημα να χρησιμοποιεί τιςCUDAκάρτες γραφικών, που του δίνουν τη δυνατότητα να κάνει πιο γρήγορους υπολογισμούς.

Περιβάλλον εργασίας(UserInterface)

Είναι ένα έξυπνο περιβάλλον εργασίας που μας δίνει περισσότερες επιλογές για την ανάπτυξη εφαρμογών, την εξερεύνηση μαθηματικών εννοιών και την κοινοποίηση των αποτελεσμάτων. Ακολουθούν οι εξής εφαρμογές του:

- **Πίνακες δεδομένων(DataTables):**η λειτουργία με μεγάλες ποσότητες δεδομένων είναι απρόσκοπτη στο Maple 15 αφού χρησιμοποιεί νέο πίνακα δεδομένων, τα οποία μπορούν να ενσωματωθούν απευθείας στο έγγραφό μας. Τα δεδομένα στον πίνακα είναι δυνατό να αλλάξουν τόσο αλληλεπιδραστικά όσο και με προγραμματισμό, για την παροχή δύναμης και ευελιξίας για την εφαρμογή μας.
- **Διαδραστική βοήθεια και εργασιακό πρότυπο(InteractiveAssistantsandTaskTemplates):**τοMaple έχει μια μεγάλη συλλογή από ClickabletoolsMath που μας καθοδηγούν μέσω ενός ευρέος φάσματος των κοινών καθηκόντων, από τη χάραξη γραφικής παράστασης μιας συνάρτησης μέχρι τον υπολογισμό ενός ολοκληρώματος, χωρίς να χρειάζεται να ξέρουμε τα ονόματα των εντολών ή τη σύνταξη.
- **Οπτικοποίηση και σχεδίαση γραφημάτων(VisualizationandPlotting):**μας προσφέρει μια εκτενή ακολουθία από εργαλεία οπτικοποίησης, συμπεριλαμβανομένων πάνω από 160 τύπων δισδιάστατων (2-D) και τρισδιάστατων (3-D) γραφημάτων καθώς και μιαδιαδραστική βοήθεια γραφήματος που προσφέρει άμεση πρόσβαση σε τύπους γραφημάτων που ισχύουν για δεδομένα ή εκφράσεις.
- **Δημιουργία διαδραστικών εφαρμογών(CreateInteractiveApplications):**για να δημιουργήσουμε γρήγορα και πιο εξεζητημένες εφαρμογές που περιλαμβάνουν αλληλεπιδραστικά στοιχεία όπως, κουμπιά, ρυθμιστικά, γραφήματα, πλαίσια ελέγχου, πλαίσια λίστας, κουμπιά εναλλαγής, καντράν, μετρητές, καθώς και νέα στοιχεία του πίνακα δεδομένων τα οποία εισάγονται στο Maple 15.
- **MapleClouddocumentrepository:** είναι ένα μέρος για να μοιραζόμαστε έγγραφα με άλλους χρήστες του Maple, μπορούμε να δημιουργήσουμε, να λάβουμε και να διανέμουμε τεχνικά έγγραφα όλα μέσα στο περιβάλλον του Maple.Από την έναρξή του έχουν ανταλλαγή χιλιάδες έγγραφα από την κοινότητα των χρηστών μας. Το Maple 15 επιτρέπει πλέον την αναζήτηση πλήρους κειμένου στο περιεχόμενο των

εγγράφων, στο cloudrepository, γεγονός που καθιστά ακόμη πιο εύκολο να βρούμε τα έγγραφα που παρουσιάζουν ενδιαφέρον γρήγορα.

Υπολογιστικοί αλγόριθμοι (Computational Algorithms)

Περιλαμβάνει νέους παγκόσμιου επιπέδου αλγορίθμους για συμβολικούς αλλά και αριθμητικούς υπολογισμούς, σημαντικές βελτιώσεις στην απόδοση και υποστήριξη για νέα πεδία εφαρμογών:

- **Διαφορικές εξισώσεις (Differential Equations):** το Maple είναι ο αδιαμφισβήτητος ηγέτης για τον υπολογισμό συμβολικών λύσεων διαφορικών εξισώσεων.
- **Βελτιστοποίηση (Optimization):** το πακέτο βελτιστοποίησης παρέχει πολλές τεχνικές για την εύρεση της ελάχιστης ή της μέγιστης τιμής της αντικειμενικής συνάρτησης που υπόκειται σε περιορισμούς. Λύνει γραμμικά και μη γραμμικά προβλήματα ελαχίστων τετραγώνων, που είτε περιορίζονται είτε όχι. Οι μεταβλητές μπορεί να είναι συνεχείς, ακέραιες ή δυαδικές.
- **Αριθμητική Πολυωνύμων (Polynomial Arithmetic):** έχουν γίνει αρκετές βελτιώσεις για την αποτελεσματικότητα των πολυωνύμων στους πολλαπλασιασμούς, στην ακριβή διαίρεση, στην παραγοντοποίηση και στα μονοπαραγοντικά και πολυδιάστατα πολυώνυμα με ακέραιους συντελεστές. Οι βελτιώσεις αυτές έχουν επιτευχθεί μέσω ενός συνδυασμού μιας νέας σωτηρικής δομής δεδομένων, γρήγορων αλγορίθμων που ως στόχο έχουν τη μείωση της μνήμης.
- **Διαφορική γεωμετρία (Differential Geometry):** είναι ένα πακέτο με πιο ολοκληρωμένο μαθηματικό λογισμικό, με 224 εντολές που καλύπτουν ένα ευρύ φάσμα θεμάτων βασικών υπολογισμών. Περιλαμβάνει πλήρη φάκελο ο οποίος περιέχει αναλυτικά παραδείγματα για όλες αυτές τις εντολές, 19 μαθήματα διαφορικής και 5 μαθήματα που απεικονίζουν τη χρήση του πακέτου εφαρμογών.
- **Παραμετρικές λύσεις (Parametric Solving):** περιλαμβάνει πολλές νέες διευκολύνσεις για το χειρισμό παραμετρικών εξισώσεων. Δίνει πλήρεις λύσεις σε παραμετρικές πολυωνυμικές εξισώσεις, και βρίσκει όλες τις διαφορετικές λύσεις.

Με τις νέες εντολές στο πακέτο RegularChains, μπορούμε να υπολογίσουμε πραγματικές λύσεις των πολυωνυμικών συστημάτων που αφορούν τις ανισότητες, το οποίο μας επιτρέπει και πάλι να εξερευνήσουμε παραμετρικές λύσεις.

- **Στατιστικό πακέτο(Statisticspackage):**συνδυάζει τα πλεονεκτήματα της αριθμητικής και συμβολικής προσέγγισης για την επίλυση προβλημάτων.

Συνδεσιμότητα(Connectivity)

Το Maple 15 μας δίνει δυνατότητα σύνδεσης με το Διαδίκτυο και σε άλλα προϊόντα:

- **Κώδικας γενιάς(CodeGeneration):**χρησιμοποιείC, Java, Fortran, VisualBasic, και MATLAB ως μια νέα γλώσσα για την παραγωγή κώδικα. Χρησιμοποιώντας τα εργαλεία δημιουργίας κώδικα, οι Maple εκφράσεις και τα προγράμματα που μπορούν εύκολα να μετατραπούν σε ατελώς πηγαίο κώδικα. Το Maple 15 περιλαμβάνει επίσης ένα νέο αλγόριθμο βελτιστοποίησης που δημιουργεί ακόμη πιο αποτελεσματικό και συμπαγή κώδικα από πριν.
- **HTTP συνδεσιμότητα(HTTPConnectivity):**προσφέρει ένα νέο πακέτο για τη σύνδεση των websites και την ανάκτηση των αποτελεσμάτων. Μπορούμε να συνδεθούμε με τα στοιχεία, σε απευθείας σύνδεση με τις βάσεις δεδομένων, καθώς και άλλες πηγές του διαδικτύου και εσωτερικού δικτύου, και να ενσωματώσουμε τα δεδομένα αυτόματα σε Maple εφαρμογές.
- **Excel συνδεσιμότητα(ExcelConnectivity):**το πακέτο ExcelTools στο Maple καθιστά εύκολη την εισαγωγή και εξαγωγή δεδομένων από υπολογιστικά φύλλα του MicrosoftExcel.Επιπλέον μας δίνετε η δυνατότητα να ανακτήσουμε το φύλλο και το όνομα των δεδομένων από ένα βιβλίο εργασίας του Excel.

1.4. Βασικές λειτουργίες

Οι χρήστες μπορούν να εισέλθουν στα μαθηματικά στην παραδοσιακή μαθηματική σημειογραφία. Υπάρχει ευρεία υποστήριξη για τους αριθμητικούς υπολογισμούς, με σταθερή ακρίβεια, καθώς και συμβολικούς υπολογισμούς και οπτικοποίηση. Το Maple ενσωματώνει δυναμικά δακτυλογραφημένη γλώσσα προγραμματισμού η οποία μοιάζει με Pascal. Υπάρχουν διασυνδέσεις με άλλες γλώσσες (C, Fortran, Java, Matlab και VisualBasic) όπως επίσης και μια διασύνδεση με το Excel.Το πρόγραμμα

είναι συμβατό με όλα τα ευρέως διαδεδομένα λειτουργικά συστήματα (π.χ. Windows, MacOS, Linux).

1.5. Αρχιτεκτονική

Το Maple βασίζεται σε ένα μικρό πυρήνα, γραμμένο σε γλώσσα C, ο οποίος και αποτελεί την γλώσσα προγραμματισμού Maple. Οι περισσότερες λειτουργίες που παρέχονται από τις βιβλιοθήκες, προέρχονται από διάφορες πηγές και αριθμητικούς υπολογισμούς. Πολλές εκτελούνται από την NAG Αριθμητικές Βιβλιοθήκες, ATLAS βιβλιοθήκες, ή GMP βιβλιοθήκες. Οι περισσότερες από τις βιβλιοθήκες είναι γραμμένες στη γλώσσα Maple. Αυτές έχουν τη δυνατότητα προβολής του πηγαίου κώδικα.

Οι διαφορετικές λειτουργίες σε Maple απαιτούν αριθμητικά δεδομένα σε διαφορετικές μορφές. Συμβολικά αποθηκεύονται στη μνήμη ως κατευθυνόμενα άκυκλα γραφήματα. Η βασική λειτουργία και η λειτουργία της ανάλυσης είναι γραμμένες σε Java ενώ η κλασική λειτουργία είναι γραμμένη σε C. Για περισσότερες πληροφορίες μπορούμε να ανατρέξουμε στο (Maple).

1.6. Εφαρμογή του προγράμματος Maple σε άλλα προϊόντα

Το Maple χρησιμοποιείται σε διάφορα άλλα προϊόντα που προέρχονται από Maplesoft και τα οποία αναφέρονται παρακάτω:

- MapleTA, Maplesoft's online σουίτα δοκιμής, χρησιμοποιείται από το Maple για να δημιουργήσει αλγοριθμικά ζητήματα και τον ανάλογο βαθμό στις απαντήσεις των φοιτητών.
- MapleNet επιτρέπει στους χρήστες να δημιουργήσουν JSP⁶ σελίδες και Java Applets. Από το MapleNet 12 και πάνω, επιτρέπει στους χρήστες να ανεβάζουν και να συνεργάζονται με τα Maple φύλλα εργασίας που περιέχουν διαδραστικά στοιχεία.
- MapleReader, Maplesoft πλατφόρμα για DRM όπου ελέγχονται ηλεκτρονικά βιβλία, χρησιμοποιείται το πρότυπο Mapleinterface. Δεν υπάρχουν διαθέσιμα βιβλία που να χρησιμοποιούν το προϊόν αυτό.

⁶Java τεχνολογία για την δημιουργία ιστοσελίδων. (<http://el.wikipedia.org/wiki/Java>).

- MapleSim, ένα εργαλείο προσομοίωσης μηχανικής.

Παρακάτω αναφέρονται τα τρία εμπορικά προϊόντα που δεν χρησιμοποιούν πλέον το πρόγραμμα Maple:

- Συμβολική MathToolbox σε Matlab περιείχε ένα τμήμα του Maple10, αλλά τώρα δεν το χρησιμοποιεί.
- Εκδόσεις του MathCad⁷ που εκδόθηκαν στο διάστημα μεταξύ 1994 και 2006 περιλάμβανε Maple συστήματα άλγεβρας υπολογιστών.
- Παλαιότερες εκδόσεις του συντάκτη επιστημονικών μαθηματικών περιλαμβάνονται στο Maple ως μια υπολογιστική μηχανή.

1.7. Σύγκριση του Maple με το Matlab και το Mathematica

Όπως και το Maple έτσι και το Matlab και το Mathematica, είναι υπολογιστικά προγράμματα. Βασικός τους στόχος είναι να αυτοματοποιούν κουραστικές και ορισμένες φορές δύσκολες εφαρμογές αλγεβρικών μετατροπών. Η βασική τους διαφορά με έναν υπολογιστή τσέπης είναι η δυνατότητα τους να επιλύουν τις εξισώσεις συμβολικά παρά αριθμητικά. Οι ειδικότερες χρήσεις και οπτικοποιητικές ποικίλουν σημαντικά από το ένα σύστημα στο άλλο, ωστόσο ο σκοπός τους, η βασική δηλαδή λειτουργία, παραμένει ο ίδιος, και είναι η μετατροπή-απόδοση μαθηματικών εκφράσεων σε συμβολική μορφή. Είναι διαδραστικά προγράμματα που αποβλέπουν στην ακρίβεια των υπολογισμών τους ή όταν αυτό δεν είναι εφικτό, στην όσο το δυνατόν υψηλότερη ακρίβεια. Επίσης έχουν τη δυνατότητα να μετατρέπουν μαθηματικές εκφράσεις, να σχεδιάζουν γραφήματα και διαθέτουν μεγάλο αριθμό αλγορίθμων.

Το λογισμικό Matlab

Αναπτύχθηκε την δεκαετία του 1970 από μια ομάδα ερευνητών στο πανεπιστήμιο Paderborn, στη Γερμανία. Ξεκίνησε ως ένα πρόγραμμα "Εργαστηρίου Μητρώων/Πινάκων" ("MATrixLABoratory"), ένα διαδραστικό πρόγραμμα, αμφίδρομης

⁷Λογισμικό που διευκολύνει κυρίως την επαλήθευση, την επικύρωση, την τεκμηρίωση και την επαναχρησιμοποίηση των υπολογισμών μηχανής συνδυάζοντας εξισώσεις κείμενο και γραφικά. (<http://en.wikipedia.org/wiki/Mathcad>).

επικοινωνίας, που είχε σκοπό να παρέχει ευκολότερη πρόσβαση στις βιβλιοθήκες LINPACK και EISPACK, χωρίς δηλαδή ο χρήστης να χρειάζεται να προγραμματίσει στη γλώσσα FORTRAN. Στην αρχή της κυκλοφορίας του προσφερόταν δωρεάν και κυρίως για πανεπιστημιακή χρήση. Γύρω στο 1984, όταν πλέον ήταν αρκετά διαδεδομένο και είχε αρχίσει να διαφαίνεται η εμπορική του αξία, το πρόγραμμα ξαναγράφηκε στην γλώσσα προγραμματισμού C, προστέθηκαν νέα χαρακτηριστικά, νέες βιβλιοθήκες, οι επωνομαζόμενες JACKPACK.

Το πρόγραμμα είναι συμβατό με όλα τα ευρέως διαδεδομένα λειτουργικά συστήματα (π.χ. Windows, MacOS, Linux). Η κεντρική αρχική ιδέα ήταν να παρέχει έναν εύκολο τρόπο χρήσης πολύπλοκων προγραμμάτων. Απευθύνονταν κυρίως σε επιστήμονες που είχαν ανάγκη να χρησιμοποιήσουν ένα τόσο σύνθετο λογισμικό αλλά δεν είχαν ούτε τον χρόνο και συχνά ούτε τις ικανότητες να το φτιάξουν από την αρχή. Έκτοτε το σύστημα καλύπτει ένα ευρύ φάσμα εφαρμογών και χρησιμοποιείται ως απλή και εύκολη γλώσσα προγραμματισμού. Πλέον είναι ένα πρόγραμμα που χρησιμοποιείται για αριθμητικούς υπολογισμούς, ειδικά στη γραμμική άλγεβρα (πίνακες). Έχει αναπτυχθεί αρκετά, για να γίνει ένα ισχυρότατο εργαλείο στην οπτικοποίηση (επεξεργασία εικόνας, γραφικά, animation), στον προγραμματισμό, στην έρευνα, στην επιστήμη των μηχανικών, και στις επικοινωνίες. Στο δυναμικό του συμπεριλαμβάνονται αλγόριθμοι, δυνατότητες χειρισμού τεράστιων ποσοτήτων δεδομένων, και ισχυρά προγραμματιστικά εργαλεία. Το Matlab δεν είναι σχεδιασμένο για συμβολικούς υπολογισμούς, αλλά αντισταθμίζει αυτή την αδυναμία του επιτρέποντας στο χρήστη να συνδέεται άμεσα με το Maple. Η επιφάνεια αλληλεπίδρασης βασίζεται κυρίως σε κείμενο, γεγονός που μπορεί να προκαλέσει σύγχυση σε αρκετούς χρήστες. Προσφέρεται ως πακέτο του βασικού προγράμματος, με πολλές "εργαλειοθήκες", που πωλούνται ξεχωριστά. Για περαιτέρω πληροφορίες μπορούμε να ανατρέξουμε στο (Matlab).

Το λογισμικό Mathematica

Αναπτύχθηκε το 1988 από την Wolfram Research. Το πρόγραμμα είναι συμβατό με όλα τα ευρέως διαδεδομένα λειτουργικά συστήματα (π.χ. Windows, MacOS, Linux). Χρησιμοποιείται ευρέως στον τομέα της επιστήμης, της μηχανικής και των μαθηματικών. Είναι ένα λογισμικό που χρησιμοποιείται για συμβολικούς και αριθμητικούς υπολογισμούς στην διεκπεραίωση αναλυτικών και αριθμητικών

υπολογισμών καθώς και στην δημιουργία γραφικών. Περιλαμβάνει ένα μεγάλο αριθμό εσωτερικών συναρτήσεων και έτοιμων υποπρογραμμάτων που είναι ενσωματωμένα σε μια ευέλικτη γλώσσα προγραμματισμού. Η ακρίβεια των αριθμών στο Mathematica καθορίζεται αυθαίρετα από τον χρήστη που μπορεί εύκολα να επεξεργάζεται αναλυτικά, αλγεβρικές παραστάσεις, ολοκληρώματα, διαφορικές εξισώσεις, πίνακες, γραφήματα. Για περισσότερες πληροφορίες μπορούμε να ανατρέξουμε (Τσαούση).

Ύστερα από την ανάλυση που κάναμε παραπάνω σχετικά με τα προγράμματα Maple, Matlab και Mathematica προκύπτουν τα εξής συμπεράσματα:

- Και τα τρία προγράμματα είναι συμβατά με όλα τα ευρέως διαδεδομένα λειτουργικά συστήματα.
- Το Maple χρησιμοποιείται για συμβολικούς υπολογισμούς, το Mathematica χρησιμοποιείται για συμβολικούς και αριθμητικούς υπολογισμούς, ενώ το Matlab παρότι χρησιμοποιείται για αριθμητικούς υπολογισμούς, αντισταθμίζει την αδυναμία του αυτή με σύνδεση στο Maple.
- Το Maple βασίζεται στις βιβλιοθήκες, το Matlab έχει εύκολη πρόσβαση σε αυτές, ενώ το Mathematica βασίζεται σε εσωτερικές συναρτήσεις και έτοιμα υποπρογράμματα τα οποία είναι ενσωματωμένα σε μια ευέλικτη γλώσσα προγραμματισμού.
- Το Maple και το Mathematica χρησιμοποιούν γλώσσα προγραμματισμού με τη μορφή εντολών ενώ το Matlab βασίζεται κυρίως σε κείμενο δηλαδή το όνομα που εκχωρείται είναι μόνο ένα όνομα και δεν αναπαριστά καμία μαθηματική μεταβλητή (κάθε όνομα πρέπει να έχει μια τιμή κάθε στιγμή). Όλα τα ορίσματα του τα δέχεται με τη μορφή πίνακα και όχι σαν συναρτήσεις.
- Και τα τρία προγράμματα μπορούν να επιλύσουν αλγεβρικούς υπολογισμούς, όμως το Matlab δίνει ιδιαίτερη έμφαση στους πίνακες.
- Το Matlab δεν έχει φύλλο εργασίας οπότε όταν βγαίνουμε από το πρόγραμμα δεν μπορούμε να αποθηκεύσουμε τα δεδομένα μας, όπως στο Maple και στο Mathematica.
- Το Matlab διαφέρει στα γραφήματα σε σχέση με τα άλλα δυο προγράμματα, γιατί πρέπει να χρησιμοποιεί κάποιο τερματικό που υποστηρίζει γραφικά για να τα αναπαραστήσει.

- Επίσης το Matlab δεν βρίσκει ακριβή λύση, αλλά προσεγγιστική και αυτή είναι η βασική διαφορά από τα συστήματα συμβολικών υπολογισμών όπως είναι τα άλλα δυο προγράμματα.

Κεφάλαιο 2

Τα βασικά του Maple

Στο κεφάλαιο που ακολουθεί θα παρουσιάσουμε αναλυτικά το πώς λειτουργεί το πρόγραμμα του Maple 8, όπως επίσης και τις βασικές εντολές που χρησιμοποιούνται δείχνοντας κάποια παραδείγματα για την καλύτερη κατανόηση τους. (Waterloo Maple Inc, 2002) (Λίλη, 2005) (Δημοτίκαλης, 2000).

2.1. Εκκίνηση Maple

Για να χρησιμοποιήσουμε το Maple πρέπει να το εγκαταστήσουμε πρώτα στον υπολογιστή μας. Το εικονίδιο του Maple 8 φαίνεται στο Σχήμα 1.

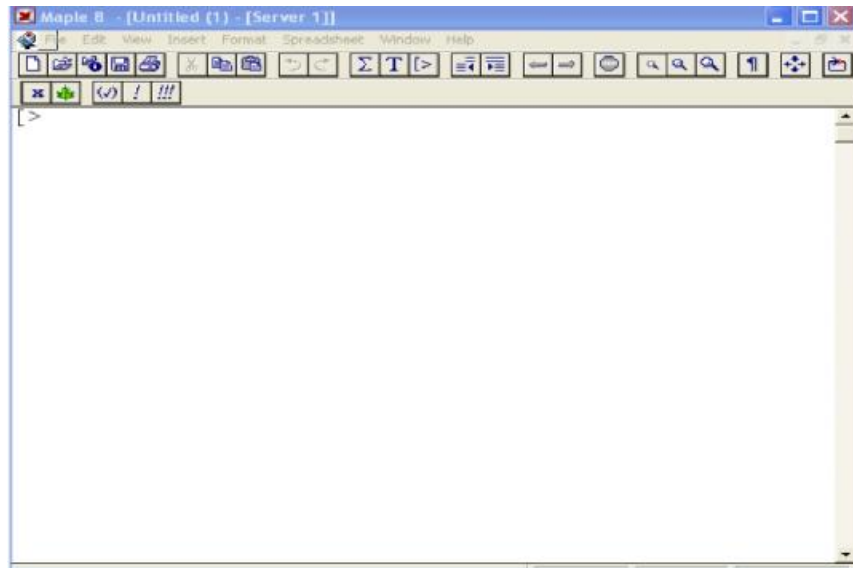


Σχήμα 1: Το εικονίδιο του Maple

Μπορούμε να ξεκινήσουμε το πρόγραμμα με διπλό κλικ πάνω στο εικονίδιο αυτό. Μετά από λίγο, αφού το πρόγραμμα φορτώσει, θα εμφανιστεί στην οθόνη μας το παράθυρο έναρξης του Maple.

2.2. Το παράθυρο του Maple

Όταν τρέξουμε το πρόγραμμα, εμφανίζεται ένα παράθυρο του Maple, όπως φαίνεται στο Σφάλμα! Άγνωστη παράμετρος αλλαγής.. Μπορούμε να του αλλάξουμε το μέγεθος όπως κάθε παράθυρο-εφαρμογή με Windows (το Maple είναι διαθέσιμο για διάφορα λειτουργικά συστήματα υπολογιστή).



Σχήμα 2: Το παράθυρο του Maple

Η έξοδος από το πρόγραμμα μπορεί να γίνει με τους εξής τρόπους:

- με κλικ στο τετραγωνίδιο [x] που βρίσκεται πάνω δεξιά στο παράθυρο του Maple, και
- με την επιλογή File → Exit στο παράθυρο εργασίας.

2.3. Η γραμμή του μενού (MenuBar)

Στην κορυφή του παραθύρου του Maple όπως φαίνεται και στο **Σφάλμα! Άγνωστη παράμετρος αλλαγής.** υπάρχει μια γραμμή από αναδυόμενα (pop-up) μενού. Μερικές φορές κάποια από τα μενού είναι γκριζα ή έχουν εξαφανιστεί. Αυτό σημαίνει ότι το συγκεκριμένο μενού (και οι επιλογές του) δεν είναι διαθέσιμο στο σημείο που βρισκόμαστε.



Σχήμα3: MenuBar

Στη συνέχεια ακολουθεί η ανάλυση του μενού **File**, το οποίο αποτελείται από κάποιες βασικές εντολές διαχείρισης αρχείων και στα οποία κάνουμε μια σύντομη αναφορά:

New: δημιουργεί νέο παράθυρο εργασίας (newmapleworksheet).

Open: ανοίγει υπάρχον παράθυρο εργασίας από οποιαδήποτε μονάδα μπορεί να έχει αυτό αποθηκευτεί (π.χ. από το σκληρό δίσκο ή τη δισκέτα).

OpenURL: ανοίγει τρέχουσα σελίδα από το διαδίκτυο (απαραίτητη προϋπόθεση να είμαστε συνδεδεμένοι με το internet).

SaveorSaveAs: αποθηκεύει το τρέχον παράθυρο εργασίας (η δεύτερη εντολή SaveAs χρησιμοποιείται αν έχουμε ήδη χρησιμοποιήσει την πρώτη αλλά θέλουμε να σώσουμε το τρέχον παράθυρο με διαφορετικό όνομα ή σε διαφορετική μονάδα). Το τρέχον παράθυρο εργασίας σώζεται σε μορφή *.mw ή σε μορφή *.mws. Στη δεύτερη περίπτωση το σωσμένο αρχείο είναι αναγνώσιμο από παλιές εκδόσεις Maple (π.χ. MapleV).

Exportas: σώζει το υπάρχον παράθυρο εργασίας σε μορφή ιστοσελίδας (δηλαδή ως *.htm ή *.html), σε μορφή LaTeX(*.tex)⁸ σε μορφή MapleText(*.txt) κ.α.

Send: στέλνει το τρέχον παράθυρο εργασίας σε μορφή συνημμένου κειμένου με ηλεκτρονικό ταχυδρομείο. Σε αυτή την περίπτωση το πρόγραμμα μάς ζητάει να σώσουμε την εργασία μας, ανοίγει καινούριο παράθυρο όπου ακολουθώντας τις οδηγίες δίνουμε το όνομα και τον κωδικό του ηλεκτρονικού μας ταχυδρομείου και μέσω του outlook ολοκληρώνεται η διαδικασία.

Close: κλείνει το τρέχον παράθυρο εργασίας. Αν δεν έχει σωθεί μας ζητείται να το σώσουμε πριν κλειστεί.

Print: τυπώνει τις εντολές και τα αποτελέσματα της εργασίας μας δηλαδή, τι υπάρχει στο παράθυρο εργασίας.

Printpreview: παρουσιάζει την εικόνα της εργασίας μας που πρόκειται να εκτυπωθεί.

Printsetup: ρυθμίζει το μέγεθος του κειμένου και τη μορφή της εκτύπωσης.

Preferences: αλλάζει τις επιλογές του Maple(δηλαδή κάνει διάφορες μορφοποιήσεις, σχετικά με το μέγεθος της παλέτας ή των γραφημάτων).

Exit: βγαίνει από το πρόγραμμα. Αν δεν έχει σωθεί η εργασία μας και πάλι μας ζητείται να γίνει.

Το μενού **Edit** μας διευκολύνει διότι διαθέτει κάποιες εντολές επεξεργασίας οι οποίες είναι χρήσιμες κατά την εγγραφή ενός κειμένου στο φύλλο εργασίας και αφορούν την αναίρεση, την αποκοπή και διάφορες άλλες ενέργειες που ακολουθούν παρακάτω:

⁸ Είναι ένα σύστημα ηλεκτρονικής στοιχειοθεσίας, κατάλληλο για την συγγραφή επιστημονικών κειμένων.

Undo: επανέρχεται στην αρχική κατάσταση, σε αυτή πριν την τελευταία μας πράξη (είτε αυτή είναι διαγραφή κειμένου είτε πληκτρολόγηση εντολής είτε οτιδήποτε έχει ολοκληρωθεί με enter).

Redo: εκτελεί την αντίστροφη διαδικασία της εντολής Undo.

Cut: διαγράφει ότι έχουμε επιλέξει και το αποθηκεύει (είναι Delete και Copy ταυτόχρονα). Η επιλογή κειμένου ή εντολής ή αποτελέσματος γίνεται πατώντας το Ctrl και κάνοντας κλικ σε ό,τι θέλουμε να επιλέξουμε ή κάνοντας κλικ στην αρχή της επιλογής και κρατώντας το σταθερά πατημένο πηγαίνουμε στο τέλος της επιλογής.

Copy: αντιγράφει ό,τι έχουμε επιλέξει και το κρατά στη μνήμη για οποιαδήποτε μελλοντική χρήση (η μνήμη κρατά κάθε φορά μόνο ένα επιλεγμένο τμήμα δηλαδή αν κάνουμε Copy σε ένα επιλεγμένο τμήμα και μετά κάνουμε Copy σε κάποιο άλλο διαφορετικό, η εντολή θα κρατήσει στη μνήμη της το τελευταίο).

Paste: επικολλάει το επιλεγμένο τμήμα που έχει στη μνήμη της από την εντολή Copy στο σημείο που βρίσκεται ο κέρσορας.

Delete Paragraph: διαγράφει ό,τι έχουμε επιλέξει ή αν δεν έχουμε επιλέξει διαγράφει όλα τα τρέχοντα στοιχεία που βρίσκονται πάνω στη γραμμή του κέρσορα.

Find: βρίσκει κείμενο ή φράση ή εντολή που έχουμε πληκτρολογήσει και μεταφέρει εκεί τον κέρσορα. Επιπλέον, μπορεί το κείμενο που ζητάμε να το αντικαταστήσει με κάτι άλλο που του δίνουμε, χωρίς να αναγκαστούμε να το ψάξουμε σε όλο το παράθυρο εργασίας.

Spellcheck: έλεγχος του τρέχοντος φύλλου εργασίας για ορθογραφικά λάθη.

Hyperlinks: αλλάζει τη μορφή (worksheet, webpage, e-mail, helptopic), το όνομα ή το αρχείο των υπερσυνδέσμων (hyperlinks) που υπάρχουν στο παράθυρο εργασίας μας. Για να δημιουργηθεί ένα hyperlink δηλαδή παραπομπή σε αρχείο ή ιστοσελίδα ή ηλεκτρονική διεύθυνση ή παροχή βοήθειας, αρκεί να επιλέξουμε το συγκεκριμένο κείμενο ή εντολή ή αποτέλεσμα και να το μετατρέψουμε σε hyperlink από τις επιλογές μορφής κειμένου στη γραμμή εργαλείων.

Unit Converter: βοηθάει στη μετατροπή των μονάδων (για παράδειγμα μετατρέπει της ώρα σε λεπτά).

Complete Command: ολοκληρώνει την τρέχουσα εντολή.

Entry Mode: μετατρέπει το κείμενό μας σε μορφή Text.

Split or Join: διαχωρίζει ή συνενώνει αντίστοιχα τμήματα της εργασίας μας (σε μορφή κειμένου ή εντολών) που είναι ενωμένα ή διαχωρισμένα αντίστοιχα με το Enter.

Execute:εκτελεί όλες τις εντολές από την αρχή μέχρι το τέλος του παραθύρου εργασίας (Worksheet) ή εκτελεί τις εντολές σε ένα προεπιλεγμένο τμήμα (Selection) ή διορθώνει συντακτικά λάθη (AutoCorrect).

RemoveOutput:διαγράφει τα αποτελέσματα των εντολών από την προεπιλεγμένη περιοχή (FromSelection) ή από όλο το παράθυρο εργασίας (FromWorksheet).

Το μενού **View** μας δίνει τη δυνατότητα να επιλέξουμε το τι μορφή θα έχει το παράθυρο μας δηλαδή ποία γραμμή εργαλείων θέλουμε να εμφανίζεται. Παρακάτω ακολουθεί η ανάλυση του:

Toolbar:εμφανίζει τη γραμμή εργαλείων. Κάνοντας κλικ στην εντολή εμφανίζουμε ή κρύβουμε την γραμμή εργαλείων.

ContextBar:εμφανίζει τη γραμμή εντολών-γραμμή κειμένου.

StatusBar:εμφανίζει τη γραμμή κατάστασης που βρίσκεται στο κάτω μέρος του παραθύρου εργασίας (χρόνος της CPU, χωρητικότητα κεντρικής μνήμης για πράξεις).

Palettes: εμφανίζει παράθυρο με έτοιμες εκφράσεις Symbol, Expression, Matrix, Vector, ShowAll δηλαδή Σύμβολα, Εκφράσεις, Πίνακες, Διανύσματα. Κάνοντας κλικ σε ένα από τα προηγούμενα εμφανίζεται στον κέρσορα η εντολή η αντίστοιχη με το σύμβολο. Το μόνο που μένει να κάνουμε είναι να δηλώσουμε τη μεταβλητή ή τα επιπλέον που θέλουμε να προσθέσουμε.

ZoomFactor:μεγεθύνει ή μικραίνει το περιβάλλον της εργασίας μας.

Bookmarks:δημιουργεί, μετονομάζει, διαγράφει και πηγαίνει το προτρεπτικό σε διάφορα προεπιλεγμένα τμήματα (σελιδοδείκτες) στο παράθυρο εργασίας. Οι εντολές `AddBookmarks` και `ModifyBookmarks` προσθέτουν και μετονομάζουν σελιδοδείκτες αντίστοιχα.

Back:επιστρέφει στο περσυνδεμένο έγγραφο.

Forward:πηγαίνει εμπρός στο περσυνδεμένο έγγραφο.

HideContent:υποκρύπτει Inputs(εντολές), Outputs(αποτελέσματα), Graphs(γραφήματα) κ.α.

ShowInvisibleCharacters:επιλέγει την επίδειξη ή όχι των χαρακτήρων.

ShowSectionRanges: επιλέγει την επίδειξη των γραμμών που οδηγούν στη σειρά των τμημάτων.

ShowGroupRange: επιλέγει την επίδειξη των γραμμών που οδηγούν στην ομάδα.

ShowObjectType:παρουσιάζει τον τύπο του αντικειμένου που είναι συνδεδεμένο.

ExpandAllSections: ανοίγει αντίστοιχα τις τυχόν υποενότητες που υπάρχουν στο παράθυρο εργασίας.

CollapseAllSections: κλείνει αντίστοιχα τις τυχόν υποενότητες που υπάρχουν στο παράθυρο εργασίας.

Το μενού **Insert** χρησιμοποιείται ανάλογα με το τι επιθυμούμε να εισάγουμε στο φύλλο εργασίας. Στη συνέχεια το παραθέτουμε αναλυτικότερα:

TextorStandardMathorMapleInputorStandardMathInput:αλλάζει την τρέχουσα μορφή του κειμένου. Όταν κάποια από τις παραπάνω μορφές είναι η τρέχουσα τότε αυτή δεν μπορεί να επιλεγεί.

ExecutionGroup:δημιουργεί νέο προτρεπτικό μια γραμμή πριν τον τρέχοντα κέρσορα (BeforeCursor) ή μια γραμμή μετά (AfterCursor).

Plot: εισάγει άξονες στις δύο ή τρεις διαστάσεις (2D, 3Dαντίστοιχα)

Spreadsheet:εισάγει πίνακα πράξεων.

Paragraph:δημιουργεί νέα γραμμή κειμένου (Text) μία γραμμή πριν τον τρέχοντα κέρσορα (BeforeCursor) ή μια γραμμή μετά (AfterCursor).

Section:εισάγει υποενότητα.

Subsection:παρεμβάλατε μια υποενότητα μετά από την τρέχουσα παράγραφο.

HyperLink: παρεμβάλατε ένας νέος υπερσύνδεσμος κειμένου.

Object: παρεμβάλατε ένα αντικείμενο.

Page Break:εισάγει προτρεπτικό με το σύμβολο PageBreak. Το πρόγραμμα λειτουργεί το ίδιο με ή χωρίς το PageBreak.

Το μενού **Format** μας βοηθά να μορφοποιούμε οποιοδήποτε κείμενο εισάγουμε στο φύλλο εργασίας. Παρακάτω ακολουθούν όλες οι επιλογές που μας επιτρέπει να κάνουμε:

Styles:δυνατότητα αλλαγής ή και προσωπικής διαμόρφωσης κειμένου ή και παραγράφου. Οι αλλαγές μπορούν να σωθούν ακόμα και να χρησιμοποιούνται ως προεπιλογή (default).

PageNumbers:εισάγει αριθμούς στις σελίδες.

Italic-Bold-Underline:διαμορφώνει τη γραμματοσειρά του κειμένου δηλαδή διαμορφώνουμε αν θα είναι τα γράμματα πλάγια, έντονα, υπογραμμισμένα.

LeftJustify: διαμορφώνει την παράγραφο αριστερά.

Center:διαμορφώνει την παράγραφο στο κέντρο.

RightJustify: διαμορφώνει την παράγραφο δεξιά.

Paragraph: διαμορφώνει τη δομή της παραγράφου.

Character: ελέγχει την εμφάνιση του κειμένου.

Indent: εισάγει μια υποενότητα χωρίς να διαγράψει τα υπάρχοντα (εντολές-αποτελέσματα).

Outdent: διαγράφει μια υποενότητα χωρίς να διαγράψει τα υπάρχοντα (εντολές-αποτελέσματα).

Convertto: μετατρέπει το επιλεγμένο κείμενο ή το κείμενο που βρίσκεται στην τρέχουσα γραμμή εργασίας σε κάποια από τις μορφές που παρέχει (StandardMath, StandardMathInput, MapleInput, PlainText).

Το μενού **Window** αναφέρεται στις ενέργειες που εφαρμόζονται στα παράθυρα του προγράμματος, τα βλέπουμε πιο κάτω:

Cascade-Tile-Horizontal-Vertical: ταξινομεί όλα τα παράθυρα, είτε το ένα δίπλα στο άλλο είτε οριζόντια, είτε κάθετα.

ArrangeIcons: τακτοποιεί τα ελαχιστοποιημένα εικονίδια των παραθύρων.

CloseAll: κλείνει όλα τα παράθυρα.

CloseAllHelp: κλείνει όλα τα παράθυρα της βοήθειας.

1 Untitled(1)-[Server 1: παρουσιάζει τα τρέχοντα παράθυρα εργασιών με το σωσμένο ή όχι όνομά τους με 1 το τρέχον και 2, 3,... τα υπόλοιπα.

Το μενού **Help** είναι το πιο σημαντικό εργαλείο του Maple. Με αυτό το εργαλείο παίρνουμε πληροφορίες για τη χρησιμότητα κάποιας εντολής και τον τρόπο εφαρμογής της ή για οποιοδήποτε πρόβλημα συναντούμε στο περιβάλλον του Maple. Οι τρόποι που μπορούμε να ανατρέξουμε στη βοήθεια είναι είτε από το μενού Help το οποίο παρουσιάζεται αναλυτικά πιο κάτω:

Introduction: άποψη της εισαγωγής του Maple 8.

Help on Context: μας εμφανίζει βοήθεια για το τρέχον σημείο που βρισκόμαστε.

New User's Tour: τρέχει το νέο γύρο του χρήστη.

What's new: καινούριες δυνατότητες.

Using Help: γενικές οδηγίες.

Glossary: γλωσσάριο βοήθειας.

Topic Search: αναζήτηση λήμματος για μια συγκεκριμένη εντολή.

Full Text Search: αναζήτηση των θεμάτων βοήθειας που περιέχουν μια λέξη ή λέξεις.

History: επιστροφή σε προηγούμενα επισκέψιμα θέματα βοήθειας.

SavetoDatabase: προσθέτει το τρέχον φύλλο εργασίας σε μια βάση δεδομένων βοήθειας.

RemoveTopic: αφαιρεί ένα θέμα από μια βάση δεδομένων.

MapleontheWeb: δυνατότητα υποστήριξης βοήθειας από το διαδίκτυο.

RegisterMaple 8: καταχωρεί αντίγραφο στο Maple 8.

AboutMaple 8: αναφέρει πληροφορίες έκδοσης επίδειξης για αυτό το αντίγραφο του Maple 8.


Ή εναλλακτικά μπορούμε να πληκτρολογήσουμε “?εντολή” (όπου εντολή = η εντολή που τον ενδιαφέρει π.χ. ?plot), ευρισκόμενος σε ένα φύλλο εργασίας, οπότε και θα εμφανιστεί αυτόματα το παράθυρο της Help στην παράγραφο με τις πληροφορίες για την συγκεκριμένη εντολή. Στην συνέχεια με την εντολή usage(εντολή) [όπου εντολή = η εντολή που τον ενδιαφέρει π.χ. usage(plot)], οπότε και θα εμφανιστεί μια περίληψη των παραμέτρων της συγκεκριμένης εντολής. Επιπλέον με την εντολή example(εντολή) [όπου εντολή = η εντολή που τον ενδιαφέρει π.χ. example (plot)], οπότε και θα εμφανιστούν παραδείγματα για την χρήση της συγκεκριμένης εντολής.


2.3.1. Η εργαλειοθήκη (ToolBar)


Η εργαλειοθήκη όπως φαίνεται και στο Σφάλμα! Άγνωστη παράμετρος αλλαγής, είναι μια γραμμή από κουμπιά (buttons) που περιέχει τα πιο σημαντικά κουμπιά. Οι παρακάτω εντολές εμφανίζονται αν τοποθετήσουμε τον κέρσορα πάνω στα αντίστοιχα εικονίδια χωρίς να κάνουμε κλικ.





Σχήμα 4: ToolBar


 **Create a new worksheet:** δημιουργεί νέο παράθυρο εργασίας επίσης ακριβώς αν κάνουμε κλικ με τη σειρά File → New από τη γραμμή του προσωπικού μενού.


 **Open an existing worksheet:** ανοίγει υπάρχον, αποθηκευμένο αρχείο σε μορφή Maple επίσης ακριβώς αν κάνουμε κλικ με τη σειρά File → Open από τη γραμμή του προσωπικού μενού.


 **OpenaspecifiedURL:** ανοίγει μια διευκρινισμένη διεύθυνση επίσης αρχείου στο διαδίκτυο (internet).


 **Savetheactiveworksheet:** αποθηκεύει το τρέχον παράθυρο εργασίας επίσης ακριβώς αν κάνουμε κλικ με τη σειρά File → Save από τη γραμμή του προσωπικού μενού.


 **Printtheactiveworksheet:** τυπώνει το τρέχον παράθυρο εργασίας επίσης ακριβώς αν κάνουμε κλικ με τη σειρά File → Print από τη γραμμή του προσωπικού μενού.


 **Cuttheselectiontotheclipboard:** διαγράφει ό,τι είναι επιλεγμένο στο παράθυρο εργασίας επίσης ακριβώς αν κάνουμε κλικ με τη σειρά Edit → Cut από τη γραμμή του προσωπικού μενού.


 **Copytheselectiontotheclipboard:** αντιγράφει ό,τι είναι επιλεγμένο στο παράθυρο εργασίας επίσης ακριβώς αν κάνουμε κλικ με τη σειρά Edit → Copy από τη γραμμή του προσωπικού μενού, αφού πρώτα βέβαια επιλέξουμε ένα τμήμα από το παράθυρο εργασίας. Υπενθυμίζουμε ότι η μνήμη του Copy είναι μόνο για ένα τμήμα τη φορά.


 **Pastetheclipboardcontentstothecurrentworksheet:** επικολλά ό,τι είναι αντιγραμμένο μέσω επίσης προηγούμενης εντολής Copy επίσης ακριβώς αν κάνουμε κλικ διαδοχικά Edit → Paste από τη γραμμή του προσωπικού μενού.


 **Undothelastoperation:** επανέρχεται στην κατάσταση που βρισκόταν πριν την τελευταία ολοκληρωμένη εντολή (ολοκληρωμένη εντολή θεωρείται το σύνολο εντολών και στοιχείων μεταξύ δύο διαδοχικών Enter). Εκτελείται επίσης ακριβώς αν κάνουμε κλικ με τη σειρά Edit → Undo από τη γραμμή του προσωπικού μενού.


 **Redothelastoperation:** εκτελεί ό,τι έχουμε επαναφέρει με την εντολή Undo επίσης ακριβώς αν κάνουμε κλικ με τη σειρά Edit → Redo από τη γραμμή του προσωπικού μενού.


 **InsertNonexecutableStandardMathinatextregion:** εισάγει το προεπιλεγμένο κείμενο από μορφή Standard Math (απενεργοποιημένο) σε εκτελέσιμο.


 **Inserttextatthecursor:** εισάγει γραμμή κειμένου κάτω από το τρέχον σημείο του κέρσορα επίσης ακριβώς αν κάνουμε κλικ με τη σειρά Insert →Text από τη γραμμή του προσωπικού μενού.


 **Insertanewexecutiongroupafterthecursor:** εισάγει καινούριο προτροπτικό κάτω από το τρέχον σημείο του κέρσορα επίσης ακριβώς αν κάνουμε κλικ με τη σειρά Insert →ExecutionGroup →AfterCursor από τη γραμμή του προσωπικού μενού.


 **Removeanysectionenclosingtheselection:** διαγράφει οτιδήποτε βρίσκεται σε μια υποενότητα μιας προεπιλεγμένης περιοχής.

 **Enclosetheselectioninasubsection:** εισάγει καινούρια υποενότητα επίσης ακριβώς αν κάνουμε κλικ με τη σειρά Insert →Section από τη γραμμή του προσωπικού μενού ή ενσωματώνει την επιλεγμένη περιοχή σε υποενότητα.


 **GobackoneorGoforwardoneinthehyperlinkhistory:** καθώς βρισκόμαστε μέσα στο περιβάλλον επίσης υπερσυνδέσμου (hyperlink) δημιουργείται ιστορία και μπορούμε να μεταβούμε με το GobackorGoforward πίσω ή μπροστά αντίστοιχα μεταξύ των εργασιών επίσης.

 **Interruptthecomputationinprogress:** διακόπτεται η διαδικασία υπολογισμού του προγράμματος σε περίπτωση που αυτή διαρκεί πολύ χρόνο ή οποτεδήποτε επιθυμούμε να μη γίνει.

 **Zoom 100% or 150% or 200%:** μεγεθύνει (150% ή 200%) το παράθυρο εργασίας ή το επαναφέρει (100%) στο αρχικό του μέγεθος επίσης ακριβώς αν κάνουμε κλικ με τη σειρά View →ZoomFactor από τη γραμμή του προσωπικού μενού.

 **Controlthedisplayofnon-printingcharacters:** ελέγχει την επίδειξη επίσης μη εκτύπωσης των χαρακτήρων.

 **Resizetheactivewindowtofilltheavailablespace:** ρυθμίζει το μέγεθος του ενεργού παράθυρου.


 **IssuethetheMaplerestartcommand:** εκδίδει την εντολή καινούριου ξεκινήματος του Maple.


2.3.2. To ContextBar


Είναι η τρέχουσα εργαλειοθήκη όπως φαίνεται και στο **Σφάλμα! Άγνωστη παράμετρος αλλαγής.** Τα κουμπιά στο contextbar αλλάζουν συχνά, ανάλογα με το τι κάνουμε σε κάθε στιγμή που χρησιμοποιούμε το πρόγραμμα. Μπορεί επίσης να περιέχει ένα πεδίο για την έκδοση και την είσοδο του κειμένου.




Σχήμα5:Context Bar

 **Toggle the input display between standard math and Maple Input:** μετατρέπει τη μορφή του τρέχοντος ή του προεπιλεγμένου κειμένου από μορφή Standard Math (απενεργοποιημένο) σε εκτελέσιμη Maple Input (ενεργοποιημένο) και αντίστροφα, όπως ακριβώς και αν κάνουμε διαδοχικά κλικ **Format** → **Convert to** → **Standard Math Input** or **Maple Input** από τη γραμμή του προσωπικού μενού.

 **Toggle the expression type between executable and non-executable:** μετατρέπει τη μορφή του τρέχοντος ή του προεπιλεγμένου κειμένου από εκτελέσιμη (ενεργοποιημένο πλήκτρο) σε μη εκτελέσιμη (απενεργοποιημένο) δηλαδή από Maple Input σε Text και αντίστροφα, όπως ακριβώς και αν κάνουμε διαδοχικά κλικ **Format** → **Convert to** → **Maple Input** or **Text** από τη γραμμή του προσωπικού μενού.

 **Auto-correct the syntax of the expression:** κάνει αυτόματα διόρθωση στη σύνταξη μιας έκφρασης.

 **Execute the current expression:** εκτελεί την τρέχουσα εντολή.

 **Execute the worksheet:** εκτελεί τις εντολές από την αρχή μέχρι το τέλος όλου του παραθύρου εργασίας, όπως ακριβώς και αν κάνουμε διαδοχικά κλικ **Edit** → **Execute** → **Worksheet** από τη γραμμή του προσωπικού μενού.

Όταν θελήσουμε να γράψουμε κάποιο κείμενο στο φύλλο εργασίας, το ContextBar αλλάζει και γίνεται όπως φαίνεται στο **Σφάλμα! Άγνωστη παράμετρος αλλαγής.**



Σχήμα 6:Μπάρα γραφής

P Normal ▾ Ορίζεται ο στυλ παραγράφου.

Times New Roman ▾ Ορίζεται η γραμματοσειρά (font).

12 ▾ Ορίζει το μέγεθος των γραμμάτων.

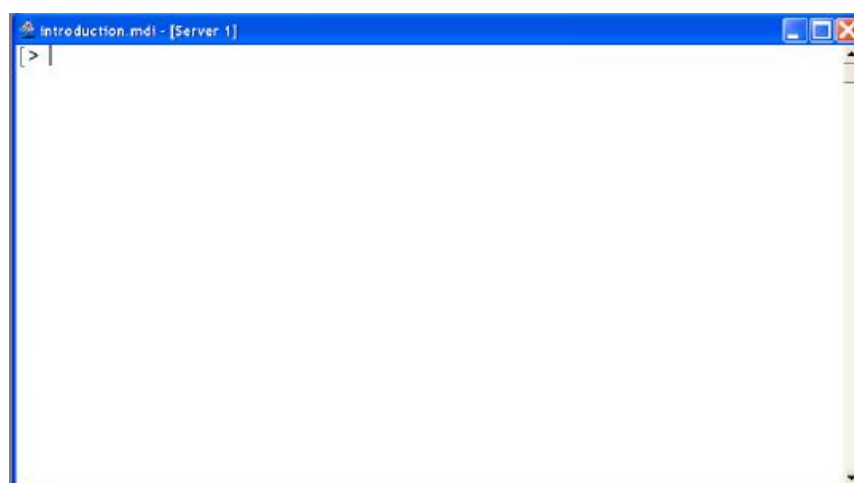
B *I* U Κάνουν bold (μαύρο, έντονο), italics (πλάγιο), και underline (υπογραμμισμένο) το επιλεγμένο (μαυρισμένο) κείμενο.

☰ ☷ ☹ Είναι για left (αριστερή), center (κέντρο), and right (δεξιά) ευθυγράμμιση της παραγράφου του επιλεγμένου κειμένου.

!!! Εκτελεί τις εντολές από την αρχή μέχρι το τέλος όλου του παραθύρου εργασίας.

2.3.3. Το φύλλο εργασίας (Worksheet)

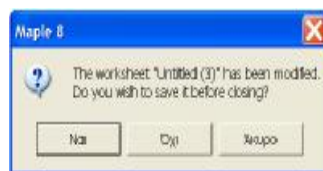
Το σημαντικότερο στο παράθυρο του Maple είναι το ενεργό φύλλο εργασίας. Το φύλλο εργασίας όπως φαίνεται **Σφάλμα! Άγνωστη παράμετρος αλλαγής.** είναι η βασική μονάδα της χρήσης του Maple, όπως το document σε ένα επεξεργαστή κειμένου. Κάθε φύλλο εργασίας του Maple που ανοίγουμε έχει το δικό του παράθυρο μέσα στο βασικό παράθυρο του Maple (που περιλαμβάνει επιπλέον τα menu και toolbar). Αυτό το παράθυρο μπορεί να μετακινηθεί και να αλλάξει μέγεθος αλλά πάντα μέσα στο βασικό παράθυρο του Maple.



Σχήμα 7: φύλλο εργασίας

Χρησιμοποιώντας τις εντολές του File μενού μπορούμε να δημιουργήσουμε ένα νέο φύλλο εργασίας (new), να ανοίξουμε (open) ένα υπάρχον φύλλο εργασίας σε ένα φάκελο που έχουμε πρόσβαση, να αποθηκεύσουμε (save) το τρέχον φύλλο εργασίας, ή να εκτυπώσουμε (print) το τρέχον φύλλο εργασίας. Υπάρχουν επίσης τα πρώτα πέντε κουμπιά στην εργαλειοθήκη. Επιπλέον υπάρχει η εντολή αποθήκευσης (saveas) στο μενού File που επιτρέπει να αποθηκεύουμε το τρέχον φύλλο εργασίας με ένα νέο όνομα σε διαφορετικό αρχείο.

Όταν αποθηκεύουμε ένα υπάρχον φύλλο εργασίας, αποθηκεύεται ακριβώς όπως εμφανίζεται εκείνη τη στιγμή. Το Maple θυμάται ακόμα και την τρέχουσα θέση του κέρσορα (cursor) μέσα στο φύλλο εργασίας.



Σχήμα 8: παράθυρο αποθήκευσης


Μια άλλη εντολή στο μενού File είναι το κλείσιμο (close) του τρέχοντος φύλλου εργασίας. Συνήθως ένα παράθυρο μηνύματος εμφανίζεται όπως φαίνεται στο **Σφάλμα! Άγνωστη παράμετρος αλλαγής.** με το μήνυμα ότι το φύλλο εργασίας έχει αλλάξει. Ρωτά αν θέλουμε να αποθηκεύσουμε τις αλλαγές. Αυτό αναφέρετε στις αλλαγές που κάναμε από τότε που ανοίξαμε το φύλλο εργασίας (εάν έχουμε κάνει αποθήκευση του φύλλου εργασίας κατά τη διάρκεια τις εργασίας, αναφέρετε στις αλλαγές που έχουν γίνει μετά από την τελευταία αποθήκευση). Η αλλαγή μπορεί να είναι τόσο ασήμαντη όσο η απλή μετακίνηση του κέρσορα μέσα στο φύλλο. Εάν επιλέξουμε "Όχι" κάθε αλλαγή που έχει γίνει θα χαθεί (δεν θα αποθηκευτεί) (Δημοτίκαλης, 2000).

Προσθήκη κειμένου—Τεκμηρίωση εργασίας

Προσθέτουμε έναν τίτλο στο φύλλο εργασίας και κάποιο κείμενο για να περιγράψουμε το πρόβλημα που λύνουμε και να τεκμηριώσουμε τα βήματα που ακολουθήσαμε για να λυθεί. Μπορούμε να προσθέσουμε το κείμενο σε διάφορες θέσεις στο φύλλο εργασίας για να περιγράψουμε πώς λύνετε το πρόβλημα. Στη

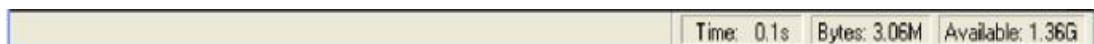
συνέχεια μπορούμε να μορφοποιήσουμε το κείμενο με τη χρήση διαφορετικών προκαθορισμένων στυλ, ή μπορούμε να ορίσουμε το στυλ που επιθυμούμε.

Για να προσθέσουμε έναν τίτλο στο φύλλο εργασίας:

- Παρεμβάλουμε μια νέα ομάδα εκτέλεσης στην κορυφή του φύλλου εργασίας, τοποθετούμε το σημείο εισαγωγής στην κορυφαία γραμμή, και από τις επιλογές ενθέτων, επιλέγουμε την ομάδα εκτέλεσης, πριν από τον κέρσορα.
- Πατάμε το ακόλουθο κουμπί  για να προσθέσουμε μια εντολή.
- Εισάγουμε το ανάλογο κείμενο, από το στυλ πλαισίου επιλέγουμε τίτλο.

2.3.4. Η γραμμή κατάστασης (Statusbar)

Στο κάτω μέρος του παραθύρου του Mapleόπως φαίνεται στο **Σφάλμα! Άγνωστη παράμετρος αλλαγής.**, βρίσκεται η γραμμή κατάστασης που εμφανίζει σημαντικές πληροφορίες: το χρόνο κάποιου υπολογισμού (σε seconds), τη μνήμη που απαιτεί το φύλλο εργασίας (σε Kilobytes), τη διαθέσιμη μνήμη στον υπολογιστή.



Σχήμα 9: Statusbar

2.4. Βασικές οδηγίες για την χρήση του προγράμματος

Το Προτρεπτικό (Prompt) του Maple

Το προτρεπτικό (prompt) είναι το σύμβολο (>) στο οποίο θα δίνουμε τις εντολές στο Maple. Το πρόγραμμα εμφανίζει τις εντολές στην οθόνη με κόκκινα γράμματα, τα αποτελέσματα με μπλε γράμματα, ενώ το κανονικό κείμενο εμφανίζεται με μαύρα γράμματα. Εάν νομίζουμε ότι πληκτρολογούμε κείμενο και τα γράμματα είναι κόκκινα, τότε πληκτρολογούμε μια εντολή στο προτρεπτικό (prompt) του Maple. Κάθε εντολή πρέπει να τελειώνει είτε με ελληνικό ερωτηματικό (;) είτε με άνω και κάτω τελεία (:). Εάν μια εντολή τελειώνει με ερωτηματικό το Maple εκτελεί την εντολή και εμφανίζει το αποτέλεσμα, ενώ αν τελειώνει με άνω και κάτω τελεία το Maple εκτελεί την εντολή αλλά δεν εμφανίζει το αποτέλεσμα στην οθόνη. Όταν πατήσουμε το πλήκτρο Return ή Enter, αλλάζει γραμμή και εμφανίζει ένα νέο prompt (>).

Για παράδειγμα μπορούμε να προσθέσουμε τους αριθμούς 2 και 3 πληκτρολογώντας **2+3;** στο προτρεπτικό του Maple και στη συνέχεια πατάμε το Return ή Enter.

```
>2+3;
```

5

Εάν δεν συμπεριλάβουμε ερωτηματικό ή άνω και κάτω τελεία στο τέλος μιας εντολής, το Maple θα εμφανίσει ένα μήνυμα προειδοποίησης μετά από ένα νέο προτρεπτικό. Στην παρακάτω γραμμή έχουμε πληκτρολογήσει την παραπάνω εντολή χωρίς άνω και κάτω τελεία. Τοποθετούμε το δείκτη οπουδήποτε στην επόμενη εντολή και πατάμε Enter ή Return για να δούμε τι θα συμβεί.

```
>2+3
```

Warning, prematureendofinput

Διάκριση του Maple μεταξύ μικρών και κεφαλαίων γραμμάτων

Οι εντολές και οι δεσμευμένες λέξεις γράφονται πάντα με μικρά γράμματα.

- Αυτός ο κανόνας ωστόσο έχει αρκετές εξαιρέσεις οι σταθερές π και i είναι: Pi και I .
- Επίσης οι εντολές του πακέτου `LinearAlgebra` (γραμμική άλγεβρα) ξεκινάνε όλες με κεφαλαίο γράμμα.
- Οι παράμετροι των εντολών τοποθετούνται σε παρενθέσεις και στην περίπτωση που είναι περισσότερες από μία χωρίζονται με κόμμα.

Έστω ότι θέλουμε να εισάγουμε την εντολή `evalf` με τη σταθερά Pi , αν γράψουμε την εντολή με κεφαλαίο γράμμα δεν αναγνωρίζει ούτε την εντολή αλλά ούτε και τη σταθερά, άρα δεν μας δίνει το αναμενόμενο αποτέλεσμα όπως βλέπουμε:

```
>Evalf(Pi);
```

Evalf(π)

Ενώ αν εισάγουμε την εντολή με μικρό γράμμα μας δίνει το σωστό αποτέλεσμα όπως διακρίνουμε παρακάτω:

```
>evalf(Pi);
```

3.141592654

2.5. Διακοπή, επανεκκίνηση και άλλες ευκολίες

Διακοπή (Stop)

Αν δεν θέλουμε να περιμένουμε την εκτέλεση μιας εντολής του Maple που παίρνει πολύ χρόνο για να εκτελεστεί κάνουμε κλικ στο κουμπί stop στην εργαλειοθήκη.

Επανεκκίνηση (Restart)

Η εντολή **restart** του Maple (επανεκκίνηση) είναι πολύ χρήσιμη. Ακυρώνει τις τιμές που έχουν αποδοθεί σε όλες τις μεταβλητές, ξεφορτώνει όλα τα φορτωμένα επιπλέον πακέτα και βάζει σε όλες τις μεταβλητές του Maple, όπως π.χ. η **Digits**, τις προεπιλεγμένες τιμές τους.

Οι χρήστες συχνά μπερδεύονται χρησιμοποιώντας το Maple γιατί αποδίδουν τιμές σε διάφορες μεταβλητές που στη συνέχεια ξεχνούν. Αυτές οι τιμές των μεταβλητών μπορεί να δώσουν ανεξήγητα αποτελέσματα στους επόμενους υπολογισμούς. Το πρόβλημα αυτό μπορεί να αποφευχθεί κάνοντας **restart** στην αρχή του κάθε φύλλου εργασίας (δίνοντας το σαν πρώτη εντολή). Εάν σε κάποιο φύλλο κάτι δεν πάει καλά μπορούμε να επαναεκτελέσουμε όλες τις εντολές από την κορυφή ξεκινώντας από το **restart**.

Μπορούμε επίσης να χρησιμοποιήσουμε το **restart** στη μέση ενός φύλλου εργασίας για να ακυρώσουμε τις τιμές όλων των μεταβλητών. Αυτό είναι ιδιαίτερα χρήσιμο αν θέλουμε να εκτελέσουμε διαφορετικά προβλήματα (ασκήσεις) στο ίδιο φύλλο εργασίας. Με το **restart** οι τιμές των μεταβλητών της μίας άσκησης δεν θα μεταφέρονται στις τιμές των ίδιων μεταβλητών της επόμενης.

Νέο Προτροπτικό (prompt) του Maple

Εάν χρειαστούμε ένα νέο προτροπτικό του Mapleprompt κάπου μέσα σε ένα φύλλο εργασίας (για να παρεμβάλλετε μια εντολή) κάνουμε κλικ στο κουμπί της

εργαλειοθήκης που έχει εικόνα [➤]. Το νέο προτροπτικό θα εμφανιστεί ακριβώς μετά το σημείο του κέρσορα.

Παρεμβολή κειμένου (Inserting text)

Για να παρεμβάλουμε ένα κείμενο σχολίου σε ένα φύλλο εργασίας του Maple, κάνουμε κλικ στο κουμπί με το «T» στην εργαλειοθήκη.

Αποκοπή (Cut), Αντιγραφή (Copy), και Επικόλληση (Paste)

Οι εντολές Cut, Copy, και Paste στο μενού Edit (επεξεργασία) μας βοηθούν να αποφύγουμε την πληκτρολόγηση κειμένου που έχουμε ήδη πληκτρολογήσει. Υπάρχουν επίσης και τα αντίστοιχα κουμπιά στην εργαλειοθήκη (το πέμπτο, έκτο και έβδομο, το ψαλίδι είναι η αποκοπή), όπως έχουμε ήδη δει στο Σχήμα 4.

Αποθήκευση και Ανάκληση Φύλλου Εργασίας

Για να ονομάσουμε ένα άτιτλο (untitled) φύλλο εργασίας που φτιάξαμε και να το αποθηκεύσουμε ακολουθούμε τα εξής βήματα:

- Πάμε στο μενού File και επιλέγουμε την εντολή Save As.
- Στο Save in του πλαισίου εντολών που εμφανίζεται, επιλέγουμε το φάκελο (folder) στον οποίο θέλουμε να αποθηκευτεί το φύλλο εργασίας που συντάξαμε. Στο Filename του ίδιου πλαισίου, πληκτρολογούμε το όνομα-τίτλο που θα δώσουμε στο φύλλο εργασίας. Τέλος, στο ίδιο πλαίσιο, επιλέγουμε την εντολή Save. (Υποθέτουμε ότι στο Save As type υπήρχε η ένδειξη Maple Worksheet την οποία και δεν αλλάζουμε, έτσι το αρχείο που δημιουργήσαμε θα είναι του τύπου Worksheet=φύλλο εργασίας).
- Όταν θα θέλουμε να ανακαλέσουμε το φύλλο εργασίας που φτιάξαμε, αρκεί να ανοίξουμε το Maple, να πάμε στο μενού File και να επιλέξουμε την εντολή Open. Αφού συμπληρώσουμε τα στοιχεία του αρχείου στο εμφανιζόμενο παράθυρο, δίνουμε την εντολή Open μέσω του ίδιου παραθύρου και το φύλλο εργασίας προβάλει στην οθόνη μας.

Σβήσιμο Αποτελεσμάτων (output)

Εάν κλείσουμε ένα φύλλο εργασίας του Maple που περιέχει αποτελέσματα (γραμμές με μπλε), την επόμενη φορά που θα το ανοίξουμε θα δούμε και τα αποτελέσματα. Για

παράδειγμα όταν ανοίξουμε ένα φύλλο εργασίας μπορούμε να δούμε την εντολή $x := 250$; που να συνεχίζεται με την γραμμή αποτέλεσμα

$x := 250$

Αυτό μπορεί να μας κάνει να πιστέψουμε ότι το Maple έχει ήδη αποδώσει την τιμή 250 στη μεταβλητή x . Στην πραγματικότητα το Maple δεν το κάνει αυτό **αλλά μόνο όταν επαναεκτελέσουμε την εντολή $x:=250$** . Αυτό μπορεί να δημιουργήσει σύγχυση.

ΠΡΟΣΟΧΗ:πρέπει πάντα πριν κλείσουμε ένα φύλλο εργασίας του Maple να σβήνουμε τα αποτελέσματα επιλέγοντας την εντολή RemoveOutput από το μενού Edit.

2.6. Σειρά των πράξεων

Το Maple εκτελεί πρώτα την ύψωση σε δύναμη, μετά τους πολλαπλασιασμούς και διαιρέσεις, μετά τις προσθέσεις και αφαιρέσεις(Δημοτίκαλης, 2000).

Εάν υπάρχουν μια σειρά από προσθέσεις και αφαιρέσεις για να γίνουν, το Maple εκτελεί από αριστερά προς τα δεξιά. Παρόμοια εάν υπάρχουν μια σειρά από πολλαπλασιασμούς και διαιρέσεις το Maple τις εκτελεί από αριστερά προς τα δεξιά.

Για να ξεπεράσουμε την κανονική σειρά εκτέλεσης εντολών του Maple πρέπει να χρησιμοποιήσουμε παρενθέσεις. Το Maple εκτελεί πρώτα τις πράξεις στις εσωτερικές παρενθέσεις.

2.7.Μεταβλητές

Μια μεταβλητή στο Maple μπορεί να έχει για όνομα οποιοδήποτε αλφαριθμητική σειράχαρακτήρων για παράδειγμα a , $total$, $exp1$, $sum1$, $sum2$ κ.α. Δεν είναι αποδεκτά ονόματα που αρχίζουν με αριθμό π.χ. $2nd$, $3total$, επίσης εκείνα που περιέχουν ειδικούς χαρακτήρες π.χ. $a\&b$, ονόματα που είναι δεσμευμένα από το Maple σαν εντολές π.χ. $plot$, $solve$ (Δημοτίκαλης, 2000).

Για να αποδώσουμε τιμή σε μια μεταβλητή στο Maple πρέπει να χρησιμοποιήσουμε τα σύμβολα :=.

Απόδοση τιμής

Όνομα μεταβλητής := τιμή ή έκφραση;

Για παράδειγμα η επόμενη εντολή αποδίδει την τιμή **897642** στην μεταβλητή **A**.

```
>A := 897642;
```

```
A := 897642
```

Βάζουμε κενά πριν και μετά από το := γιατί κάνει την εντολή ευκολότερη στην ανάγνωση. Το Maple συνήθως δεν ενδιαφέρεται, για τα κενά αυτά, ανάμεσα στα διάφορα στοιχεία μιας εντολής. Όμως δεν μπορούμε να αφήσουμε κενά μέσα στο ίδιο στοιχείο μιας εντολής, για παράδειγμα στην προηγούμενη εντολή δεν μπορούσαμε να αφήσουμε κενό μεταξύ : και =.

Ένας άλλος τρόπος για να δώσουμε τιμή σε μια μεταβλητή είναι να πληκτρολογήσουμε την απλή εξίσωση $x=5$, όμως δεν αποδίδουμε την τιμή 5 στη μεταβλητή x . Ωστόσο η εντολή **assign** επιτρέπει να χρησιμοποιήσουμε μια απλή εξίσωση για να δώσουμε τιμή σε μια μεταβλητή.

```
>assign(x=8);
```

Αξίζει να σημειωθεί ότι εκτελώντας την **assign(x=8)** δεν εμφανίζει κανένα αποτέλεσμα. Αλλά η τιμή αποδίδετε στη μεταβλητή:

```
>x;
```

```
8
```

Η εντολή **assign** μπορεί να αποδώσει τιμές σε πολλές μεταβλητές χρησιμοποιώντας ένα σύνολο απλών εξισώσεων.

```
>assign({y=4, z=5});
```

```
>y;
```

```
4
```

>z;

5

Όταν θέλουμε να μετακινηθούμε από ένα πρόβλημα σε άλλο, είναι καλύτερα να εξαφανίσουμε τις μεταβλητές που έχουμε ορίσει σε προηγούμενο πρόβλημα. Για αυτό θα πρέπει να πληκτρολογήσουμε την εντολή **restart**η οποία ακυρώνει την απόδοση της τιμής μιας μεταβλητής.

2.8. Βασικές εντολές του Maple και παραδείγματα

Στη συνέχεια δίνουμε έναν πίνακα με στοιχειώδεις εντολές του Maple και παραδείγματα χρήσης του⁹.

Εντολή	Επεξήγηση
Restart	Επανεκκίνηση –Μηδενισμός όλων των μεταβλητών
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
:	Απόκρυψη αποτελεσμάτων
;	Τελικόσύμβολο εντολών
%	Άμεση αναφορά στο προηγούμενο αποτέλεσμα
Sqrt	Τετραγωνική ρίζα
Abs	Απόλυτητιμή
Subs	Αντικατάσταση
<όνομα μεταβλητής> := <τιμή>	Ανάθεσητιμήςσεμεταβλητή
<όνομα μεταβλητής> :=<'τιμή'>	Ακύρωση απόδοσηςτιμής
<από>..<έως>	Όρια μιας μεταβλητής
evalf()	Υπολογισμός σε αριθμό κινητής υποδιαστολής
Simplify	Απλοποίηση
diff()	Παραγωγήιση

⁹Θεωρούμε πως ο αναγνώστης είναι εξοικειωμένος με στοιχειώδεις έννοιες μαθηματικών απειροστικού διαφορικού και ολοκληρωτικού λογισμού και γραμμικής άλγεβρας, καθώς επίσης με βασικές έννοιες πληροφορικής και χρήση ηλεκτρονικού υπολογιστή.

<code>int()</code>	Ολοκλήρωση
<code>convert()</code>	Μετατροπή
<code>expand()</code>	Ανάπτυξη
<code>factor()</code>	Παραγοντοποίηση
<code>limit()</code>	Όριο
<code>solve()</code>	Λύση εξισώσεων
<code>fsolve()</code>	Προσεγγιστική λύση εξισώσεων
<code>dsolve()</code>	Λύση διαφορικής εξίσωσης
<code>sum()</code>	Άθροισμα
<code>evalm()</code>	Πράξεις για πίνακες
<code>plot()</code>	Σχεδίαση γραφικής παράστασης
<code>plot3d()</code>	Τρισδιάστατη απεικόνιση
<code>display()</code>	Συνδυασμός διαγραμμάτων
<code>DEplot()</code>	Γραφική παράσταση λύσης διαφορικής εξίσωσης

Γενικά παραδείγματα

Επανεκκίνηση και μηδενισμός όλων των μεταβλητών:

`>restart;`

Απλή πρόσθεση:

`>6+2;`

8

Απλή αφαίρεση:

`>9-4;`

5

Πολλαπλασιασμός:

`>8*6;`

48

Διαίρεση:

`>28/2;`

14

Συνδυασμός όλων των παραπάνω πράξεων:

`>5*5-4^2+10/2;`

14

Η άνω και κάτω τελεία αποκρύπτει το αποτέλεσμα της εξόδου:

```
>t1:=20;
```

Τετραγωνική ρίζα:

```
>sqrt (144);
```

12

Χρήση του προηγούμενου αποτελέσματος:

```
>sqrt(49);
```

7

```
>(%)^3;
```

343

Αυτή η δυνατότητα είναι ιδιαίτερα χρήσιμη όταν κάνουμε πολλές πράξεις στη σειρά, γιατί δεν θα χρειάζεται να ξαναγράψουμε την απάντηση κάθε φορά.

Απόλυτητιμή:

```
>abs(x+5);
```

$|x + 5|$

Αντικατάσταση:

Για να κάνουμε αντικατάσταση, χρησιμοποιούμε την εντολή **subs**.

```
>A:=4*x^3-5;
```

$A := 4x^3 - 5$

```
>subs(x=4,A);
```

251

Ανάθεση τιμής σε μια μεταβλητή:

```
>B:=1987;
```

$B := 1987$

Η παραπάνω εντολή αποδίδει την τιμή 1987 στην μεταβλητή B.

Ακύρωση απόδοσης τιμής:

```
>B:='1987';
```

$B := 1987$

Η παραπάνω εντολή ακυρώνει την τιμή που έχει αποδοθεί στο B.

Υπολογισμός σε αριθμό κινητής υποδιαστολής:

```
>evalf(4^1/3);
```

1.333333333

Η **evalf** σημαίνει evaluate floating point. Με την **evalf** μπορούμε επίσης να προσδιορίσουμε τον αριθμό των δεκαδικών ψηφίων που περιέχονται στην απάντηση.

```
>evalf(4^1/3,7);
```

1.333333

Απλοποίηση:

Για να απλοποιήσουμε μια παράσταση χρησιμοποιούμε την εντολή **simplify**.

```
>simplify(3/x+4/x^2);
```

$$\frac{3x + 4}{x^2}$$

Παραγωγή:

Χρησιμοποιούμε την εντολή **diff**, για να βρούμε την παράγωγο μιας παράστασης ως προς μια μεταβλητή, η οποία παίρνει σαν δεδομένα εισόδου την παράσταση και την μεταβλητή ως προς την οποία παραγωγίζουμε.

```
>y:=x^4;
```

$$y := x^4$$

```
>diff(y,x);
```

$$4x^3$$

Επίσης μπορούμε να βάλουμε την παράσταση κατευθείαν στην εντολή.

```
>y:=x^4;
```

$$y := x^4$$

```
>diff(x^4,x);
```

$$4x^3$$

Ολοκλήρωση:

Χρησιμοποιούμε την εντολή **int**, για να υπολογίσουμε ολοκληρώματα.

```
>f:=x->x^4;
```

$$f := x \rightarrow x^4$$

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

$$\frac{x^5}{5}$$

Ανάπτυξη:

Η εντολή **expand** κάνει τις πράξεις και αναπτύσσει το γινόμενο.

```
>expand((x+2)^3);
```

$$x^3 + 6x^2 + 12x + 8$$

Παραγοντοποίηση:

Η εντολή **factor** κάνει παραγοντοποιήσεις.

```
>factor(x^4+7*x-2);
```

$$(x+2)(x^3 - 2x^2 + 4x - 1)$$

Όριο:

Η εντολή **limit** υπολογίζει το όριο μιας παράστασης ή συνάρτησης όταν μια μεταβλητή τείνει προς μία τιμή.

```
>limit((x^3-2)/(x+1),x=2);
```

$$2$$

Λύση εξισώσεων:

```
>solve(3*x-5=17);
```

$$\frac{22}{3}$$

Προσεγγιστική λύση εξισώσεων:

```
>f:=x->x^2-2*sin(x);
```

$$f := x \rightarrow x^2 - 2 \sin(x)$$

```
>solve(f(x)=1/2,x);
```

$$\text{RootOf}(2_Z^2 - 4 \sin(_Z) - 1, \text{label} = _L1)$$

Το Maple δεν κατάρθωσε να βρει την αναλυτική μορφή της λύσης της εξίσωσης. Αυτό είναι αναμενόμενο αφού η εξίσωση δεν είναι αλγεβρική. Για να βρούμε τις λύσεις αυτές μπορούμε να χρησιμοποιήσουμε την εντολή **fsolve**, η οποία προσπαθεί να

λύσει την εξίσωση αριθμητικά και όχι ακριβώς. Η εντολή χρησιμοποιείται ακριβώς όπως και η **solve**.

```
>fsolve(f(x)=1/2,x);
```

-0.2263172795

Άθροισμα:

Ας δούμε πως μπορούμε να υπολογίσουμε αθροίσματα με το Maple. Η εντολή:

```
sum(S(i), i = 0..N)
```

αντιστοιχεί στο

$$\sum_{i=0}^N s(i)$$

(Το N μπορεί να είναι άπειρο).

```
>sum(k^2,k=0..5);
```

55

Πράξεις σε πίνακες:

Πρώτα ορίζουμε τον πίνακα

```
>A:=matrix([[4,1],[2,5]]);
```

$$A := \begin{bmatrix} 4 & 1 \\ 2 & 5 \end{bmatrix}$$

Αν θέλουμε να δούμε τον πίνακα A:

```
>evalm(A);
```

$$\begin{bmatrix} 4 & 1 \\ 2 & 5 \end{bmatrix}$$

Στο κεφάλαιο 6 ακολουθούν αναλυτικότερα οι πράξεις σε πίνακες.

Σχεδίαση γραφικής παράστασης:

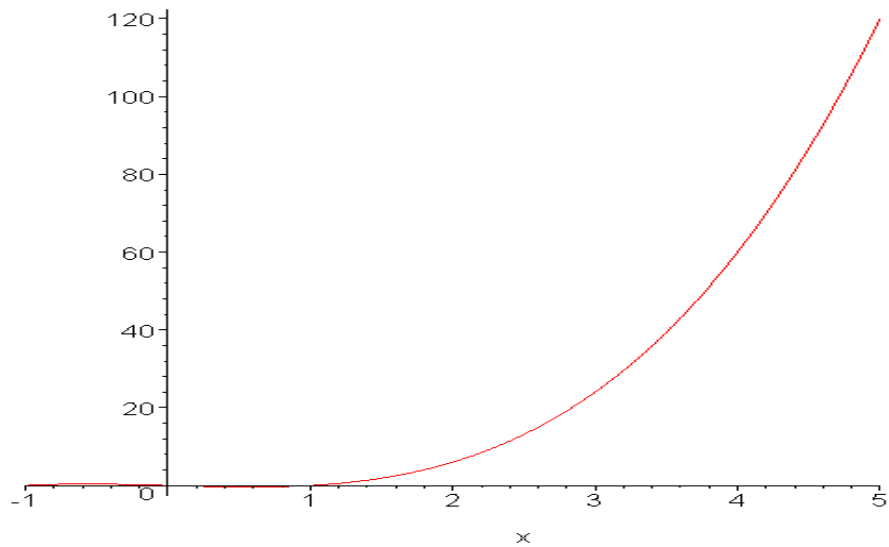
Έστω η συνάρτηση:

```
>f:=x->(x^3-x);
```

$$f := x \rightarrow x^3 - x$$

Της οποίας η γραφική παράσταση είναι:

```
plot(f(x),x=-1..5);
```

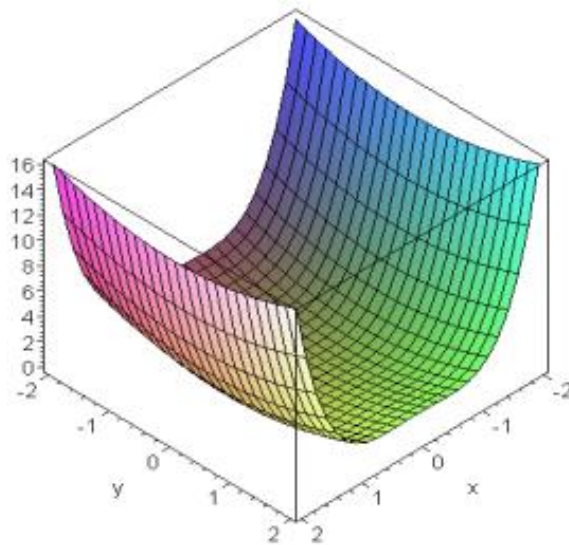


Σχήμα 10: Γράφημα της συνάρτησης $f(x)=x^3-x$

Τρισδιάστατη απεικόνιση:

Η εντολή **plot3d** χρησιμοποιείται για τρισδιάστατη αναπαράσταση.

```
>plot3d(y^2/2-x^2/2+x^4,x=-2..2,y=-2..2);
```



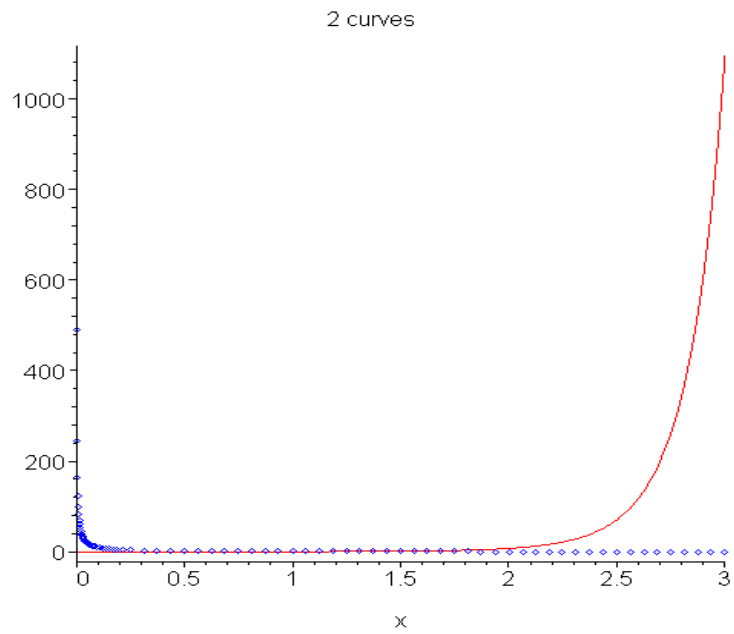
Σχήμα 11: Γράφημα τρισδιάστατης απεικόνισης

Συνδυασμός διαγραμμάτων:

Η εντολή **display** χρησιμοποιείται για να παρουσιάσει μαζί μια ή περισσότερες γραφικές παραστάσεις.

```
>F:=plot( exp(x^2-2), x=0..3, color=red,style=line):
G:=plot( 1/x, x=0..3, color=blue,style=point):
```

```
plots[display]([F,G], title="2 curves");
```



Σχήμα 12: Σχεδίαση δύο συναρτήσεων

Κεφάλαιο 3

Εξισώσεις

Σε αυτό το κεφάλαιο θα αναφερθούμε στην επίλυση γραμμικών και μη εξισώσεων και στις εντολές που χρησιμοποιούνται μέσα από μια σειρά παραδειγμάτων.

3.1. Εισαγωγή

Εξίσωση (**equation**) είναι μια ισότητα που περιέχει τουλάχιστον μια μεταβλητή. Οι εξισώσεις έχουν πάντα τη μορφή:

$$\text{έκφραση1} = \text{έκφραση2}$$

Το παρακάτω είναι μια εξίσωση:

$$y=5*x+5$$

Για να δώσουμε όνομα σε μια εξίσωση χρησιμοποιούμε την εντολή:

equation1

```
>equation1:=y=5*x+5;
```

```
equation1 := y = 5 x + 5
```

Για να εξάγουμε το αριστερό μέρος μιας εξίσωσης χρησιμοποιούμε την εντολή **lhs** (lefthandside= αριστερή πλευρά) και για το δεξιό μέρος, την εντολή **rhs** (righthandside= δεξιά πλευρά).

```
>equation1;
```

```
y = 5 x + 5
```

```
>lhs(equation1);
```

```
y
```

```
>rhs(equation1);
```

$$5x + 5$$

3.2. Αλγεβρικές εξισώσεις

Το πρόγραμμα του Maple έχει ένα σημαντικό πλεονέκτημα, καθώς έχει την ικανότητα να επιλύει αλγεβρικές εξισώσεις, γραμμικές και μη, ως προς μια μεταβλητή. Για να λύσουμε μια εξίσωση χρησιμοποιούμε την εντολή **solve** ή την εντολή **fsolve**. Η εντολή **solve** λύνει τις εξισώσεις συμβολικά, ενώ η **fsolve**, λύνει τις εξισώσεις αριθμητικά. Επίσης η **fsolve** μπορεί να χρησιμοποιηθεί όταν η **solve** δεν επιστρέφει ένα αποτέλεσμα.

Θα πρέπει να πούμε στο Maple για ποια μεταβλητή ή μεταβλητές θέλουμε λύση. Αυτό με τη σειρά του προσπαθεί να λύσει τις εξισώσεις με αλγεβρικούς χειρισμούς.

Όταν θέλουμε να λύσουμε μια μόνο εξίσωση η μορφή της είναι:

Solve (εξίσωση, μεταβλητή για την οποία θέλουμε λύση)

```
> solve( x^2+2=4, x);
```

$$\sqrt{2}, -\sqrt{2}$$

```
> fsolve( x^2+2=4, x);
```

$$-1.414213562, 1.414213562$$

Τέλος, θα ήτανε καλύτερα να δώσουμε ένα όνομα σε μια εξίσωση πριν την επιλύσουμε όπως επίσης μπορούμε να δώσουμε όνομα και στην λύση της εξίσωσης.

Παράδειγμα1

Δίνουμε ένα όνομα στην εξίσωση:

```
> MyEquation := 5*x-2=15;
```

$$MyEquation := 5x - 2 = 15$$


```
>solve (MyEquation,x);
```

$$\frac{17}{5}$$

Παράδειγμα 2

Δίνουμε ένα όνομα στη λύση της εξίσωσης:

```
>answer:=solve(MyEquation, x);
```

$$answer := \frac{17}{5}$$

Χρησιμοποιώντας την εντολή **subs** μπορούμε να επιβεβαιώσουμε την απάντηση με αντικατάσταση της λύσης στην εξίσωση με την εντολή που προαναφέραμε.

Παράδειγμα3

```
>subs(x=answer, MyEquation);
```

$$15 = 15$$

Σύστημα με μια εξίσωση:

Παράδειγμα 1

```
>restart;
```

```
>eq:=(3-a ^4)*x+b=c*y;
```

$$eq := (3 - a^4)x + b = c y$$

```
>solve(eq,x);
```

$$\frac{b - c y}{-3 + a^4}$$

Παράδειγμα 2

```
>eq1:=x=cos(4*x);
```

$$eq1 := x = \cos(4 x)$$

Αν προσπαθήσουμε να λύσουμε την εξίσωση αυτή με την εντολή **solve** τότε

παίρνουμε το εξής:

```
> solve(eq1, x);
```

$$\frac{1}{4} \text{RootOf}(-Z + 4 \cos(Z))$$

Αυτό σημαίνει ότι το Maple δεν κατόρθωσε να βρει την αναλυτική μορφή της λύσης της εξίσωσης. Αυτό είναι αναμενόμενο αφού η εξίσωση είναι μη αλγεβρική. Μπορούμε, όμως, να βρούμε την λύση (αν υπάρχει) προσεγγιστικά με την εντολή **fsolve** που παίρνει σαν δεδομένα εισόδου μια εξίσωση και την μεταβλητή ως προς την οποία θέλουμε να λύσουμε την εξίσωση την οποία γράφουμε ίση με μια αρχική εκτίμηση της λύσης. Για την πιο πάνω εξίσωση υπάρχουν 3 λύσεις: δύο μεταξύ του -1 και 0 και μία μεταξύ του 0 και 1.

```
> fsolve(eq1, x=-1);
```

-0.8988262168

```
> fsolve(eq1, x=-0.5);
```

-0.5333330629

```
> fsolve(eq1, x=1);
```

0.3130883085

Σύστημα με δύο εξισώσεις:

```
> eq1 := 2*x+4*y=6;
```

$eq1 := 2x + 4y = 6$

```
> eq2 := -x+5*y=1/2;
```

$eq2 := -x + 5y = \frac{1}{2}$

```
> solve( {eq1, eq2}, {x, y} );
```

$\{y = \frac{1}{2}, x = 2\}$

Γραμμικό Σύστημα με δύο εξισώσεις:

Ορίζουμε πρώτα τις δυο εξισώσεις:

```
>linear1:=4*x-3*y=23;
```

$$linear1 := 4x - 3y = 23$$

```
>linear2:=2*x-9*y=45;
```

$$linear2 := 2x - 9y = 45$$

Έπειτα, καταχωρούμε τη λύση του συστήματος στη μεταβλητή **soln**, χρησιμοποιώντας την εντολή **solve**, αφού στα δεδομένα εισόδου βάλουμε τις δύο εξισώσεις και τους δύο αγνώστους σε αγκύλες (**{ }**):

```
>soln:=solve({linear1,linear2},{x,y});
```

$$soln := \left\{ x = \frac{12}{5}, y = \frac{-67}{15} \right\}$$

Επαληθεύουμε τις λύσεις ως εξής:

```
>subs(soln,linear1);
```

$$23 = 23$$

```
>subs(soln,linear2);
```

$$45 = 45$$

3.3. Συστήματα με περισσότερες από μια λύσεις

Όταν έχουμε μία εξίσωση με περισσότερες από μια λύσεις ας δούμε τι γίνεται:

```
>eq:=x^2+x-5=0;
```

$$eq := x^2 + x - 5 = 0$$

```
>solve(eq,x);
```

$$-\frac{1}{2} + \frac{\sqrt{21}}{2}, -\frac{1}{2} - \frac{\sqrt{21}}{2}$$

Η απάντηση μας είναι μερικές εκφράσεις οι οποίες χωρίζονται με κόμμα. Κάτι τέτοιο το ονομάζουμε σειρά εκφράσεων. Μπορούμε να της δώσουμε ένα όνομα. Θα την ονομάσουμε λοιπόν **sol**.

```
>sol:=solve(eq,x);
```

$$\text{sol} := -\frac{1}{2} + \frac{\sqrt{21}}{2}, -\frac{1}{2} - \frac{\sqrt{21}}{2}$$

Η πρώτη έκφραση στην σειρά εκφράσεων είναι η **sol[1]** και η δεύτερη η **sol[2]**.

Άρα έχουμε:

>sol[1];

$$-\frac{1}{2} + \frac{\sqrt{21}}{2}$$

>sol[2];

$$-\frac{1}{2} - \frac{\sqrt{21}}{2}$$

Στην συνέχεια θα πολλαπλασιάσουμε την **x-sol[1]** και την **x-sol[2]** για επιβεβαίωση:

>prod:=(x-sol[1])*(x-sol[2]);

$$\text{prod} := \left(x + \frac{1}{2} - \frac{\sqrt{21}}{2} \right) \left(x + \frac{1}{2} + \frac{\sqrt{21}}{2} \right)$$

>expand(prod);

$$x^2 + x - 5$$

Τέλος θα χρησιμοποιήσουμε την εντολή **evalf** για να μετατρέψουμε το αποτέλεσμα μας σε δεκαδική μορφή:

>evalf(sol);

$$1.791287848, -2.791287848$$

Ας δούμε ένα ακόμη παράδειγμα:

>eq3:=x^2+x*y-2*y^2=0;

$$\text{eq3} := x^2 + x y - 2 y^2 = 0$$

>eq4:=4*x-y=1;

$$\text{eq4} := 4 x - y = 1$$

>sols :=solve({eq3, eq4},{x,y});

$$sols := \{x = \frac{2}{9}, y = \frac{-1}{9}\}, \{x = \frac{1}{3}, y = \frac{1}{3}\}$$

Το Maple δίνει πάλι την λύση σαν μια σειρά εκφράσεων. Αυτή τη φορά όμως κάθε όρος στην έκφραση είναι ένα σύνολο εξισώσεων. Άρα:

>**sols[1];**

$$\{x = \frac{2}{9}, y = \frac{-1}{9}\}$$

>**sols[1][1];**

$$x = \frac{2}{9}$$

Ένα επιπλέον παράδειγμα που περιλαμβάνει τη RootOf:

>**eq7:=2*x^2-3*y=5;**

$$eq7 := 2x^2 - 3y = 5$$

>**eq8:=x*y-y=1;**

$$eq8 := xy - y = 1$$

>**solutions:=solve({eq7,eq8},{x,y});**

$$solutions := \{x = 2, y = 1\}, \{y = -\frac{4}{3} - \frac{2}{3}\text{RootOf}(2_Z^2 + 2_Z - 1, \text{label} = _L1), \\ x = \text{RootOf}(2_Z^2 + 2_Z - 1, \text{label} = _L1)\}$$

Η μια λύση είναι σίγουρα η **x=2,y=1**. Τις άλλες όμως λύσεις δεν τις γνωρίζουμε, για αυτό θα χρησιμοποιήσουμε την **allvalues**. Με την **allvalues** μπορούμε να χειριστούμε εξισώσεις που περιέχουν **RootOf** ή ακόμα και ένα σύνολο εξισώσεων που περιέχουν **RootOf**. Εάν το δεύτερο σύνολο των εξισώσεων είναι αυτό με το **RootOf**, μπορούμε να βρούμε τις λύσεις.

Η **allvalues** επιστρέφει την ακριβή αλγεβρική έκφραση όταν είναι δυνατό, μετά μπορεί να χρησιμοποιηθεί η **evalf** για να πάρουμε προσεγγίσεις των αριθμητικών

τιμών. Για πολυωνυμικές εξισώσεις που δεν μπορούν να λυθούν ακριβώς η **allvalues** επιστρέφει προσεγγιστικές αριθμητικές λύσεις.

```
>solutions[2];
```

$$\{x = \text{RootOf}(2_Z^2 + 2_Z - 1, \text{label} = _L3), \\ y = -\frac{4}{3} - \frac{2}{3} \text{RootOf}(2_Z^2 + 2_Z - 1, \text{label} = _L3)\}$$

```
>allvalues(solutions[2]);
```

$$\{x = -\frac{1}{2} + \frac{\sqrt{3}}{2}, y = -1 - \frac{\sqrt{3}}{3}\}, \{x = -\frac{1}{2} - \frac{\sqrt{3}}{2}, y = -1 + \frac{\sqrt{3}}{3}\}$$

```
>evalf(solutions[2]);
```

$$\{y = -1.577350269, x = 0.3660254038\}$$

3.4. Σύνολα εξισώσεων (Sets)

Για να λύσουμε ένα σύστημα από εξισώσεις θα πρέπει να χρησιμοποιήσουμε την εντολή **solve** στην παρακάτω μορφή:

solve({εξισώσεις},{μεταβλητές για τις οποίες λύνονται})

Παρατηρούμε ότι οι εξισώσεις περικλείονται σε άγκιστρα και είναι χωρισμένες με κόμματα. Οι μεταβλητές επίσης περικλείονται με άγκιστρα και διαχωρίζονται με κόμματα. Στο Maple, τα αντικείμενα που περικλείονται σε άγκιστρα και διαχωρίζονται με κόμμα αποτελούν ένα **σύνολο** (set). Έτσι το Maple επιλύει ένα σύνολο εξισώσεων για ένα **σύνολο** μεταβλητών (Δημοτίκαλης, 2000):

Παράδειγμα 1

```
>eq1:=4*x-7*y=14;
```

$$eq1 := 4x - 7y = 14$$

```
>eq2:=14*x+2*y=19;
```

$$eq2 := 14x + 2y = 19$$

```
>soln:=solve({eq1, eq2},{x,y});
```

$$\text{soln} := \left\{ y = \frac{-60}{53}, x = \frac{161}{106} \right\}$$

Η λύση μας είναι $x=161/106$, $y=-60/53$. Το Maple δίνει την απάντηση σαν ένα σύνολο δύο εξισώσεων. Σε ένα σύνολο η σειρά με την οποία τα στοιχεία του εμφανίζονται δεν είναι σημαντική. Όμως, αν δώσουμε σε ένα σύνολο όνομα, μπορούμε να εξάγουμε τα στοιχεία του ένα προς ένα σαν να είναι μια σειρά εκφράσεων. Άρα έχουμε:

```
>soln[1];
```

$$y = \frac{-60}{53}$$

Τώρα το **soln[1]** είναι το όνομα μιας εξίσωσης.

Παράδειγμα 2

```
>eq1:=2*x-5*y=12;
```

$$eq1 := 2x - 5y = 12$$

```
>eq2:=12*x+4*y=17;
```

$$eq2 := 12x + 4y = 17$$

```
>soln:=solve({eq1,eq2},{x,y});
```

$$\text{soln} := \left\{ y = \frac{-55}{34}, x = \frac{133}{68} \right\}$$

Αν θέλουμε να αποδώσουμε στις x , y τις τιμές που βρήκαμε, πρέπει να χρησιμοποιήσουμε την εντολή **assign**. Επειδή η **soln** είναι το όνομα ενός συνόλου απλών εξισώσεων.

```
>assign(soln);
```

```
>x;
```

$$\frac{133}{68}$$

```
>y;
```

$$\frac{-55}{34}$$

3.5. Διαφορικές εξισώσεις

Για να λύσουμε μια ή περισσότερες διαφορικές εξισώσεις, θα πρέπει πρώτα να τις ορίσουμε στο Maple. Για να λυθούν θα πρέπει να χρησιμοποιήσουμε την εντολή **dsolve**, σε αυτή την εντολή μπορούμε να δώσουμε και τις αρχικές συνθήκες του προβλήματος (Παράρτημα Β, Επίλυση διαφορικών εξισώσεων με το Maple).

>restart;

Ας πάρουμε τη διαφορική εξίσωση της μορφής $\frac{dy}{dx} = f(x,y)$.

Η λύση αυτής εξίσωσης είναι η $y(x)$ η οποία ικανοποιεί την ακόλουθη εξίσωση:

$$y'(x) = f(x, y(x))$$

Για παράδειγμα, αν θέλουμε να λύσουμε $\frac{dy}{dx} = 2x - 3y$ εισάγουμε την εντολή **dsolve** στο Maple.

>dsolve(diff(y(x),x)= 2*x-3*y(x));

$$y(x) = \frac{2x}{3} - \frac{2}{9} + e^{(-3x)} _C1$$

Για να βρούμε τις ιδιαίτερες λύσεις, μπορούμε να αντικαταστήσουμε τις διαφορετικές τιμές για τη σταθερά, $_C1$. Για να λύσουμε αυτήν την διαφορική εξίσωση με έναν αρχικό όρο, εισάγουμε την ακόλουθη εντολή.

>dsolve({diff(y(x),x)=2*x-3*y(x),y(0)=5},y(x));

$$y(x) = \frac{2x}{3} - \frac{2}{9} + \frac{47}{9} e^{(-3x)}$$

Μπορούμε επίσης να βρούμε τη σχέση μεταξύ του αρχικού όρου και της σταθεράς στη γενική λύση:

>dsolve ({diff(y(x),x)=2*x-3*y(x),y(0)=C},y(x));

$$y(x) = \frac{2x}{3} - \frac{2}{9} + e^{(-3x)} \left(\frac{2}{9} + C \right)$$

```
>sol: =dsolve ({diff(y(x), x) =2*x-3*y(x), y (0) =C}, y(x));
```

$$sol := y(x) = \frac{2x}{3} - \frac{2}{9} + e^{(-3x)} \left(\frac{2}{9} + C \right)$$

Έπειτα, εξάγουμε τη δεξιά πλευρά αυτής της εξίσωσης (που χρησιμοποιεί rhs) και την επαναπροσδιορίζουμε.

```
>y: =unapply(rhs(sol), x, C);
```

$$y := (x, C) \rightarrow \frac{2}{3}x - \frac{2}{9} + e^{(-3x)} \left(\frac{2}{9} + C \right)$$

Κεφάλαιο 4

Εκφράσεις και Συναρτήσεις

Στο κεφάλαιο που ακολουθεί θα αναφερθούμε στην αντικατάσταση, απλοποίηση και παραγοντοποίηση εκφράσεων όπως επίσης και με τις συναρτήσεις, την παραγοντοποίηση και την ολοκλήρωση τους αλλά και την δημιουργία εκφράσεων από μια συνάρτηση και το αντίστροφο (Δημοτίκαλης, 2000), (Ξενοφώντος & Γεωργίου, 2007).

4.1. Εκφράσεις

Όπως γνωρίζουμε οι μεταβλητές και οι αριθμοί μπορούν να συνδυαστούν για να κάνουν **εκφράσεις** (expressions) του Maple. Ακολουθούν μερικά παραδείγματα εκφράσεων του Maple:

```
>x^2;
```

$$x^2$$

```
>4*x^2-9;
```

$$4x^2 - 9$$

```
>x;
```

$$x$$

```
>5;
```

$$5$$

```
>x*sin(2^x)-4*ln(x^2+2) ;
```

$$x \sin(2^x) - 4 \ln(x^2 + 2)$$

Στο Maple μπορούμε να δώσουμε σε μια έκφραση ένα όνομα ώστε να μπορούμε να ανατρέχουμε σε αυτό το όνομα αργότερα.

Ένα όνομα μπορεί να δοθεί σε μια έκφραση χρησιμοποιώντας τα σύμβολα :=. Η παρακάτω εντολή δίνει το όνομα **A** στην έκφραση $3*x^2-7$.

```
>A: = 3*x^2-7;
```

$$A := 3x^2 - 7$$

Τα ονόματα μπορεί να είναι μεγαλύτερα από ένα γράμμα, π.χ. dist θα είναι ένα έγκυρο όνομα για μια έκφραση.

4.1.1. Αντικατάσταση (Substituting)

Για να υπολογίσουμε μια έκφραση που περιέχει το x στο σημείο $x=5$, πρέπει να αντικαταστήσουμε το $x=5$ στην έκφραση, χρησιμοποιώντας την εντολή **subs**:

```
>A: = 3*x^2-7;
```

$$A := 3x^2 - 7$$

```
>subs(x=5, A);
```

68

ΠΡΟΣΟΧΗ: στο κόμμα και στις παρενθέσεις.

Εάν το **A** είναι μια έκφραση που περιέχει το x , δεν μπορούμε να υπολογίσουμε την τιμή της στο $x=5$ πληκτρολογώντας **A(5)**:

```
>A(5);
```

$$3x(5)^2 - 7$$

Εάν θέλουμε να χρησιμοποιήσουμε όρους όπως το **A(5)**, πρέπει να ορίσουμε ένα άλλο αντικείμενο του Maple, μια **συνάρτηση** (function).

4.1.2. Κλασματικές δυνάμεις και μετασχηματισμοί

Έστω ότι έχουμε την έκφραση $(x-2)^{(1/3)}$.

```
>expr := (x-2)^(1/3);
```

$$expr := (x - 2)^{(1/3)}$$

Αν δοκιμάσουμε να αντικαταστήσουμε το $x=-6$ στην έκφραση.

```
>subs(x=-6, expr);
```

$(-8)^{(1/3)}$

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

1.000000000 + 1.732050807 I

Αυτή βέβαια δεν είναι η απάντηση που θέλουμε. Μάθαμε ότι στο Maple δεν μπορούμε να υπολογίσουμε την πραγματική κυβική ρίζα ενός αρνητικού με την ύψωση του αριθμού στο $1/3$. Για να λάβουμε την σωστή απάντηση, πρέπει να ξαναγράψουμε την έκφραση $(x-2)^{(1/3)}$ σαν **surd(x-2,3)** πριν αντικαταστήσουμε με $x=-6$.

Ένας τρόπος να μετασχηματίσουμε την έκφραση που περιέχει δύναμη (^) σε μια έκφραση που περιέχει **surd** είναι να χρησιμοποιήσουμε την εντολή:

convert(expression, surd)

```
>newexpr := convert(expr, surd);
```

newexpr := surd(x - 2, 3)

```
>subs(x=-6, newexpr);
```

surd(-8, 3)

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

-2.

Η εντολή **convert** χρησιμοποιείται για να μετατρέψει την έκφραση του Maple από μια μορφή σε μια άλλη.

4.1.3. Ανάπτυξη (Expand)

Ακολουθεί ένα παράδειγμα χρήσης της **expand**.

Έστω ότι θέλουμε να πολλαπλασιάσουμε την έκφραση $(x^2-4)(x+1)(x-2)(x^2+x+1)$ η οποία θα είναι η είσοδος και θα μας δώσει ως έξοδο το αποτέλεσμα της σε γινόμενο. Αρχικά ας ονομάσουμε την έκφραση B.

```
>B := (x^2-4)*(x+1)*(x-2)*(x^2+x+1);
      B := (x^2 - 4) (x + 1) (x - 2) (x^2 + x + 1)
```

Τώρα με την εντολή **expand** πολλαπλασιάζουμε την έκφραση. Ονομάζουμε το αποτέλεσμα C.

```
>C := expand(B);
      C := x^6 - 6 x^4 - 3 x^3 + 6 x^2 + 12 x + 8
```

Ακολουθεί ένα ακόμη παράδειγμα που παρουσιάζει ότι το Maple αναπτύσσει, απλοποιεί, και παραγοντοποιεί εκφράσεις και όχι συναρτήσεις.

```
>h := x -> sin(2*x)*cos(4*x);
      h := x -> sin(2 x) cos(4 x)
```

```
>expand(h);
      h
```

Το **h** είναι το όνομα της συνάρτησης και το Maple δεν αναπτύσσει (ή παραγοντοποιεί ή απλοποιεί) συναρτήσεις. Αν όμως η **h(x)** είναι μια έκφραση το Maple μπορεί να την αναπτύξει χρησιμοποιώντας τριγωνομετρικές ταυτότητες.

```
>expand(h(x));
      16 sin(x) cos(x)^5 - 16 sin(x) cos(x)^3 + 2 sin(x) cos(x)
```

4.1.4. Απλοποίηση και Παραγοντοποίηση

Υπάρχουν διάφορες εντολές που μας επιτρέπουν να απλοποιήσουμε μια παράσταση. Παρακάτω θα δούμε τις πιο βασικές. Έστω η έκφραση:

```
>expression1:=x^2-6*x+9;
      expression1 := x^2 - 6 x + 9
```

Τότε η εντολή **factor** γράφει την παράσταση **expression1** σαν γινόμενο, ενώ η εντολή **expand** κάνει τις πράξεις και αναπτύσσει το γινόμενο:

```
>factor(expression1);  
          (x-3)2  
>expand(%);  
          x2-6x+9
```

Ας δούμε και κάποια άλλα παραδείγματα:

```
>expand( (x+1)5 ); factor(%);  
          x5+5x4+10x3+10x2+5x+1  
          (x+1)5
```

Μια άλλη σημαντική εντολή είναι η **simplify**, η οποία προσπαθεί να κάνει απλοποιήσεις. Ας δούμε κάποια παραδείγματα:

```
>simplify(1/x+1/x4);  
           $\frac{x^3+1}{x^4}$   
>simplify(cos(t)2+sin(t)2);  
          1  
>simplify((a2-b2)/(a+b));  
          a-b  
>simplify(4(1/2)+3);  
          5  
>simplify(exp(a+ln(b*exp(c))));  
          b e(a+c)  
>simplify(sin(x)2+cos(x)2);  
          1  
>simplify(sin(x)2);  
          1-cos(x)2  
>g := sqrt(x2);  
          g :=  $\sqrt{x^2}$ 
```

Θα πρέπει να επισημάνουμε τα παρακάτω για την ευκολότερη χρήση της απλοποίησης. Στην απλοποίηση είναι καλύτερα να χρησιμοποιούνται δύο εντολές, μια για ορισμό της έκφρασης και μια για την απλοποίησή της. Ένας λόγος για αυτό είναι ότι ορίζοντας την έκφραση, το Maple μας εμφανίζει το αποτέλεσμα, το οποίο μας δίνει τη δυνατότητα να ελέγξουμε αν είναι πραγματικά αυτό που θέλαμε να δώσουμε. Ένας δεύτερος λόγος είναι ότι μικρές εντολές του Maple είναι πιο ευανάγνωστες από τις μεγάλες σε μήκος. Γενικά είναι καλύτερα να χρησιμοποιούμε μια σειρά μικρών σε μήκος εντολών από μια μόνο πολύπλοκη εντολή. Το παρακάτω παράδειγμα δείχνει πως χρησιμοποιείται η εντολή **simplify** για την πρόσθεση κλασμάτων:

```
>A := x/(x-1) + 3/(x+1);
```

$$A := \frac{x}{x-1} + \frac{3}{x+1}$$

```
>simplify(A);
```

$$\frac{x^2 + 4x - 3}{x^2 - 1}$$

Δίνουμε τώρα ένα παράδειγμα που δείχνει ότι το Maple μπορεί επίσης να απλοποιήσει τριγωνομετρικές εκφράσεις:

```
>trig_expr := sin(x)^2+cos(x)^2;
```

$$trig_expr := \sin(x)^2 + \cos(x)^2$$

```
>simplify(trig_expr);
```

$$1$$

Παράδειγμα 1

Έστω ότι θέλουμε να παραγοντοποιήσουμε το πολυώνυμο $f=(x+y+1)^3-1$. Πρώτα ορίζουμε το πολυώνυμο και έπειτα το παραγοντοποιούμε χρησιμοποιώντας την εντολή **factor**:

```
>f:=(x+y+1)^3;
```

$$f := (x + y + 1)^3$$

```
>f:=(x+y+1)^3: factor(f-1);
(x+y)(x^2+2xy+3x+3+3y+y^2)
```

Παράδειγμα 2

Έστω ότι θέλουμε να παραγοντοποιήσουμε το γινόμενο των πολυωνύμων $p1=-3x+7x^2-3x^3+7x^4$ και $p2=5x^5+3x^3+x^2-2x+1$. Αρχικά θα τα ορίσουμε:

```
>p1 := -3*x+7*x^2-3*x^3+7*x^4;
p1 := -3x + 7x^2 - 3x^3 + 7x^4
```

```
>p2 := 5*x^5+3*x^3+x^2-2*x+1;
p2 := 5x^5 + 3x^3 + x^2 - 2x + 1
```

Και στη συνέχεια θα παραγοντοποιήσουμε το γινόμενο των πολυωνύμων $p1p2$:

```
>factor(p1*p2);
x(7x-3)(x^2+1)^2(5x^3-2x+1)
```

4.2. Εισαγωγή Συνάρτησης

Μπορούμε να ορίσουμε τις δικές μας συναρτήσεις στο Maple χρησιμοποιώντας το συνδυασμό->"μείον-μεγαλύτερο". Αυτός ο συνδυασμός συμβόλων μοιάζει με βέλος, και διαβάζεται σαν "τείνει στο". Για παράδειγμα η επόμενη εντολή ορίζει την συνάρτηση $f(x)=x^2$:

```
>f := x -> x^2;
f := x -> x^2
```

Αν γράψουμε:

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

το Maple μας λέει ότι η $f(x)$ ισούται με x^2 . Παρατηρούμε ότι αν πούμε:

```
>f;
f
```


τότε δεν παίρνουμε x^2 , μια και η f είναι συνάρτηση του x και αυτό πρέπει να διευκρινιστεί.

Μπορούμε να διαβάσουμε την προηγούμενη εντολή σαν: "η f είναι η συνάρτηση του x τείνει στο x^2 ". Ορίζοντας μια συνάρτηση με αυτό τον τρόπο επιτρέπεται να χρησιμοποιούμε εκφράσεις όπως $f(5)$ όταν εκτιμάται μια συγκεκριμένη τιμή της συνάρτησης:

```
>f (5);
```

25

Βέβαια μπορούμε να χρησιμοποιήσουμε την νέα μας συνάρτηση με περισσότερο πολύπλοκους τρόπους. Ας δούμε το επόμενο παράδειγμα:

```
>(f (5+.001)-f (5))/.001;
```

10.001

Η γενική μορφή μιας εντολής που ορίζει μια συνάρτηση στο Maple είναι:

Όνομα της συνάρτησης := όνομα της μεταβλητής -> έκφραση που περιέχει την μεταβλητή

Στο παράδειγμά μας το όνομα της συνάρτησης ήταν f , η μεταβλητή ήταν x , και η έκφραση που περιέχει την μεταβλητή ήταν x^2 .

4.3. Ενσωματωμένες συναρτήσεις

Όπως ένας υπολογιστής τσέπης, το Maple έχει ένα αριθμό από ενσωματωμένες συναρτήσεις:

εκθετική: **exp(x)**(αυτό σημαίνει e^x)

φυσικός λογάριθμος: **ln(x)**

τριγωνομετρικές: **sin(x)**, **cos(x)**, **tan(x)**, **sec(x)**, **csc(x)**, **cot(x)**

αντίστροφεστριγωνομετρικές: **arcsin(x)**, **arccos(x)**, **arctan(x)**, **arcsec(x)**, **arccsc(x)**, **arccot(x)**

απόλυτη τιμή: **abs(x)**

τετραγωνική ρίζα: **sqrt(x)**

n-οστή ρίζα: **surd(x,n)**

πρόσημο: **signum(x)** (είναι +1 εάν **x** θετικός, 0 εάν **x** είναι 0, -1 εάν **x** είναι αρνητικός)

Οι τριγωνομετρικές συναρτήσεις χρησιμοποιούν ακτίνια (**radians**), και όχι μοίρες (**degrees**).

Ας δούμε μερικά παραδείγματα:

```
>exp(1);
```

e

ΠΡΟΣΟΧΗ: Στο αποτέλεσμα το Maple χρησιμοποιεί το

e

*που είναι το **exp(1)**, όπως το γνωρίζουμε. Όμως, εάν χρησιμοποιήσουμε το σύμβολο **esainput**, το Maple δεν το καταλαβαίνει. Το γράμμα **e** είναι απλά το όνομα μιας μεταβλητής.*

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

2.718281828

```
>evalf (e);
```

e

Σημειώνουμε ότι στο αποτέλεσμα του Maple, ο αριθμός e είναι με έντονα (**e**), ενώ η μεταβλητή e είναι με πλάγια (*e*).

```

>sin (3*Pi/2);
-1
>sin (3);
sin(3)
>evalf(%);
0.1411200081

```

Η προηγούμενη απάντηση είναι σωστή, είναι το ημίτονο των 3 ακτινίων (3 **radians**).

```

>abs (-7);
7
>sqrt(7);
 $\sqrt{7}$ 
>evalf (sqrt(7));
2.645751311
>sqrt(-7);
 $\sqrt{7} I$ 
>signum(Pi);
1
>surd(8,3);
2
>surd(-8,3);
-2

```

ΠΡΟΣΟΧΗ: Η προηγούμενη εντολή υπολογίζει σωστά την κυβική ρίζα του -8. Μπορεί κάποιος να νομίζει ότι η εντολή $(-8)^{(1/3)}$ πρέπει επίσης να παράγει το αποτέλεσμα (-2). Όμως ας δούμε τι συμβαίνει:

```

>(-8)^(1/3);
 $(-8)^{(1/3)}$ 
>evalf(%);
1.000000000 + 1.732050807 I

```

Αυτή είναι η μιγαδική κυβική ρίζα του -8, η οποία δεν είναι πιθανόν αυτή που θέλουμε. Δεν χρησιμοποιούμε τη δύναμη (^) για να υπολογίσουμε τη ρίζα ενός αρνητικού αριθμού. Μόνο η συνάρτηση **surd** μπορεί να υπολογίσει σωστά μια πραγματική ρίζα ενός αρνητικού αριθμού.

ΠΡΟΣΟΧΗ: Οι άνθρωποι και τα βιβλία συνήθως γράφουν $\sin x$ αντί για $\sin(x)$. Αν το δοκιμάσουμε στο Maple δεν θα γίνει κατανοητό.

```
>sinPi;  
Error, missingoperatoror `;
```

4.4. Παραγωγή και ολοκλήρωση συναρτήσεων

Η εντολή που πληκτρολογούμε στο πρόγραμμα για να παραγωγίσουμε μια συνάρτηση είναι **diff**, ενώ για να ολοκληρώσουμε η εντολή είναι **int**. Σ' αυτές τις εντολές ορίζουμε σαν πρώτο στοιχείο την συνάρτηση την οποία θα παραγωγίσουμε ή θα ολοκληρώσουμε και έπειτα την μεταβλητή ως προς την οποία θα γίνει η παραγωγή ή η ολοκλήρωση.

Παράδειγμα 1

```
>restart;  
>f:=x^8/y^5;
```

$$f := \frac{x^8}{y^5}$$

```
>diff (f,x);
```

$$\frac{8x^7}{y^5}$$

```
>diff(f,x,x);
```

$$\frac{56x^6}{y^5}$$

```
>diff(f,x,y,y);
```

$$\frac{240x^7}{y^7}$$

Παρακάτω θα χρησιμοποιήσουμε την ίδια συνάρτηση f του πιο πάνω προβλήματος και θα την ολοκληρώσουμε.

```
>restart;
```

```
>f:=x^8/y^5;
```

$$f := \frac{x^8}{y^5}$$

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

$$\frac{x^9}{9 y^5}$$

```
>int(int(f,x),y);
```

$$-\frac{x^9}{36 y^4}$$

4.5. Δημιουργία έκφρασης από μια συνάρτηση

Μια πολύ χρήσιμη λειτουργία του Maple είναι το αν η **g** είναι συνάρτηση, ορισμένη χρησιμοποιώντας το συνδυασμό **->**, τότε η **g(x)** είναι μια έκφραση. Ας προσέξουμε την ακόλουθη σειρά εντολών.

```
>g:=x->x^2+x;
```

$$g := x \rightarrow x^2 + x$$

```
>g(x);
```

$$x^2 + x$$

```
>g(6);
```

$$42$$

```
>subs(x=6,g(x));
```

$$42$$

4.6. Δημιουργία συνάρτησης από μια έκφραση

Ας δούμε τώρα πως μετατρέπουμε μια έκφραση σε μια συνάρτηση. Έστω η έκφραση:

```
>expr := sin(x^3+x)*ln(csc(2^x));
```

$$expr := \sin(x^3 + x) \ln(\csc(2^x))$$

Ας υποθέσουμε ότι θέλουμε να μετατρέψουμε την έκφραση σε συνάρτηση, για την οποία θέλουμε το όνομα **h**. Επειδή η έκφραση είναι πολύπλοκη, δεν χρειάζεται να την επαναπληκτρολογήσουμε. Μπορούμε να μετατρέψουμε την έκφραση σε συνάρτηση με την εντολή του Maple **unapply**.

```
>h := unapply(expr, x);  
h := x → sin(x3 + x) ln(csc(2x))
```

Η μορφή της εντολής **unapply** είναι πάντα

unapply(μια έκφραση,μια μεταβλητή)

Πρέπει να προσέξουμε τις παρενθέσεις και το κόμμα. Η μεταβλητή που δίνετε είναι η ανεξάρτητη μεταβλητή της συνάρτησης που ορίζουμε.

4.7. Αντίστροφη συνάρτηση

Έστω η συνάρτηση $f(x) = \sqrt{4x+1}$. Θα προσπαθήσουμε να βρούμε την αντίστροφη συνάρτηση και να τη γράψουμε σαν συνάρτηση του x . Για να το κάνουμε αυτό υπάρχουν τέσσερα βήματα.

Βήμα 1

θα γράψουμε την εξίσωση **y = f(x)**. Και θα την ονομάσουμε "**eq**".

```
>eq:=y=sqrt(4*x+1);  
eq := y = √(4x + 1)
```

Βήμα 2

Θα επιλύσουμε την εξίσωση **y=f(x)** ως προς x . Η λύση θα είναι μια έκφραση που περιέχει το y , ή μια σειρά εκφράσεων που θα περιέχει πολλές εκφράσεις που περιέχουν το y . Θα ονομάσουμε τη λύση "**ysolution**".

```
>ysolution:=solve(eq,x);
```

$$ysolution := -\frac{1}{4} + \frac{y^2}{4}$$

Βήμα 3

Θα χρησιμοποιήσουμε την εντολή **subs** για να αλλάξουμε το y σε x στην **ysolution**.
Θα ονομάσουμε την νέα έκφραση "**xsolution**".

```
>xsolution:=subs(y=x,ysolution);
```

$$xsolution := -\frac{1}{4} + \frac{x^2}{4}$$

Βήμα 4

Θα χρησιμοποιήσουμε την εντολή **unapply** για να επαναφέρουμε την έκφραση **xsolution** σε συνάρτηση. Θα δώσουμε στην αντίστροφη συνάρτηση ένα κατάλληλο όνομα όπως "**finv**".

```
>finv:=unapply(xsolution,x);
```

$$finv := x \rightarrow -\frac{1}{4} + \frac{1}{4}x^2$$

Κεφάλαιο 5

Όρια, Παράγωγοι, Αθροίσματα και Ολοκληρώματα

Στο κεφάλαιο που ακολουθεί θα αναφερθούμε στον υπολογισμό των ορίων στην παραγωγή εκφράσεων και συναρτήσεων, στον υπολογισμό των αθροισμάτων και τον υπολογισμό των ορισμένων αλλά και αόριστων ολοκληρωμάτων. Για περισσότερες πληροφορίες μπορούμε να ανατρέξουμε (Ξενοφώντος & Γεωργίου, 2007).

5.1. Όρια

Η εντολή του Maple `limit` (όριο) χρησιμοποιείτε για να βρούμε το όριο μιας έκφρασης που περιέχει το x όταν το x τείνει σε κάποια τιμή. Η σύνταξη είναι:

limit (έκφραση που περιέχει το x , $x =$ τιμή)

```
>limit ((x^2-1)/(x+1), x=-1);
```

-2

```
>limit ((4+3/n) ^n,n=infinity);
```

∞

```
>g: =x-> (2*x^3)/ (4*x+6);
```

$$g := x \rightarrow \frac{2x^3}{4x+6}$$

```
>limit (g(x), x=4);
```

$\frac{64}{11}$

```
>f: =x->abs(x)/x-x^2;
```

$$f := x \rightarrow \frac{|x|}{x} - x^2$$

```
>limit (f(x), x=0, right);
```

1

```
>limit (f(x), x=0, left);
```


-1

5.1.1. Όρια όταν το x τείνει σε έναν αριθμό

```
>f:=x->((x^2-5*x+6)/(x-3));
```

$$f := x \rightarrow \frac{x^2 - 5x + 6}{x - 3}$$

```
>lim:=limit(f(x),x=3);
```

lim := 1

```
>value(lim);
```

1

Υπολογισμός ορίων με περισσότερες από μια μεταβλητές:

```
>B := limit( 2*x^2/(x^2+2*y^2), x=0 );
```

$$B := \lim_{x \rightarrow 0} \frac{2x^2}{x^2 + 2y^2}$$

```
>value(B);
```

0

```
>limit( 2*x^2/(x^2+2*y^2), y=0 );
```

2

5.1.2. Όρια όταν το χτείνει στο +∞ και ή στο - ∞

```
>g := x -> (2*x^3-1)/sqrt(3*x^6+1);
```

$$g := x \rightarrow \frac{2x^3 - 1}{\sqrt{3x^6 + 1}}$$

```
>lim1 := limit(g(x), x=infinity);
```

$$\text{lim1} := \frac{2\sqrt{3}}{3}$$

```
>evalf(lim1);
```

1.154700539

```
>lim2 := limit(g(x), x=-infinity);
```

$$\text{lim2} := -\frac{2\sqrt{3}}{3}$$

```
>evalf(lim2);
```

-1.154700539

5.1.3. Όρια από αριστερά και από δεξιά

```
>h := x -> abs(x)/x-x^2;
```

$$h := x \rightarrow \frac{|x|}{x} - x^2$$

```
>limit(h(x), x=0, left);
```

-1

```
>limit(h(x), x=0, right);
```

1

Επειδή τα πλευρικά όρια δεν είναι ίσα, μπορούμε να πούμε ότι το όριο δεν υπάρχει. Ας το δούμε με το Maple:

```
>limit(h(x), x=0);
```

undefined

5.2. Παραγωγήιση

Για να παραγωγίσουμε μια **έκφραση** (expression), χρησιμοποιούμε την εντολή **diff**, ενώ για να παραγωγίσουμε μια **συνάρτηση** (function) χρησιμοποιούμε τον τελεστή παραγωγίσης **D**.

Για να βρούμε την παράγωγο μιας παράστασης ως προς μια μεταβλητή, χρησιμοποιούμε την εντολή **diff**, όπως έχουμε προαναφέρει, η οποία παίρνει σαν δεδομένα εισόδου την παράσταση και την μεταβλητή ως προς την οποία παραγωγίζουμε.

Παράδειγμα 1

```
>y:=x^7;
```

$$y := x^7$$

```
>diff(y,x);
```

$$7 x^6$$

Μπορούμε να βάλουμε την παράσταση κατευθείαν στην εντολή, ως εξής:

```
>diff(x^7,x);
```

$$7 x^6$$

Ας δούμε τώρα μια διαφορά μεταξύ των παρακάτω δύο εφαρμογών της εντολής **diff**.

```
>diff(c*x^7,x);
```

$$7 c x^6$$

```
>diff(c*x^5,c);
```

$$x^5$$

Για παραγώγους πιο υψηλής τάξης χρησιμοποιούμε το σύμβολο **\$** μετά από τη μεταβλητή. Για παράδειγμα, αν θέλουμε να βρούμε την τρίτη παράγωγο του cx^5 , γράφουμε:

```
>diff(c*x^5,x$3);
```

$$60 c x^2$$

Μπορούμε επίσης, να δώσουμε μια συνάρτηση σαν δεδομένο εισόδου.

```
>f:=x->cos(x)^3;
```

$$f := x \rightarrow \cos(x)^3$$

```
>diff(f(x),x);
```

$$-3 \cos(x)^2 \sin(x)$$

```
>diff(f(x),x$5);
```

$$60 \sin(x)^3 - 183 \cos(x)^2 \sin(x)$$

```
>diff(f,x);
```

$$0$$

Παρατηρούμε ότι η τελευταία εντολή μας έδωσε 0 γιατί δεν διευκρινίσαμε ότι η f είναι συνάρτηση του x και το Maple το θεώρησε σαν σταθερά. Στη συνέχεια θα

υπολογίσουμε την παράγωγο μιας συνάρτησης σαν μια καινούργια συνάρτηση χρησιμοποιώντας τον τελεστή **D**.

Παράδειγμα 2

```
>f:=x->x^5;
```

$$f := x \rightarrow x^5$$

```
>D(f);
```

$$x \rightarrow 5x^4$$

Αν τώρα, καταχωρίσουμε την απάντηση με το όνομα, μπορούμε να βρούμε τιμές της παραγώγου εύκολα.

```
>fprime:=D(f);
```

$$fprime := x \rightarrow 5x^4$$

```
>fprime(2); fprime(3.5); fprime(-7.14);
```

80

750.3125

12994.59808

Αυτό μπορεί να γίνει και κατευθείαν με τον τελεστή **D**.

```
>D(f)(2);
```

80

```
>D(f)(3.5);
```

750.3125

```
>D(f)(x);
```

$5x^4$

5.2.1. Παραγωγή μιας έκφρασης

Θα χρησιμοποιήσουμε την εντολή **diff** για να παραγωγίσουμε μια έκφραση ως προς μια μεταβλητή. Ας παραγωγίσουμε την $\sin(t^3)\cos(e^t)$ ως προς το t και να ονομάσουμε το αποτέλεσμα **deriv**.

```
>expr := sin(t^3)*cos(exp(t));
```

$expr := \sin(t^3) \cos(e^t)$

```
>deriv := diff(expr, t);
```

$deriv := 3 \cos(t^3) t^2 \cos(e^t) - \sin(t^3) \sin(e^t) e^t$

Το αποτέλεσμα της εντολής **diff** είναι μια νέα έκφραση, η **deriv** είναι τώρα το όνομα μιας έκφρασης. Εάν θέλουμε την τιμή της παραγώγου της **sin(t³)cos(e^t)** στο σημείο=

$\frac{p}{4}$, πρέπει να αντικαταστήσουμε $tot = \frac{p}{4}$ στην έκφραση **deriv**.

```
>subs(t=Pi/4, deriv);
```

$\frac{3}{16} \cos\left(\frac{\pi^3}{64}\right) \pi^2 \cos\left(e^{\left(\frac{\pi}{4}\right)}\right) - \sin\left(\frac{\pi^3}{64}\right) \sin\left(e^{\left(\frac{\pi}{4}\right)}\right) e^{\left(\frac{\pi}{4}\right)}$

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

-1.784707437

Θα πρέπει να επισημάνουμε ότι η σύνταξη για παραγωγή της έκφρασης ως προς το t είναι:

diff(έκφραση, t)

Πρέπει να ορίσουμε ότι θέλουμε να παραγωγίσουμε ως προς t . Η αιτία φυσικά είναι ότι μια έκφραση μπορεί να περιέχει διάφορα γράμματα και το Maple πρέπει να γνωρίζει ποιο από όλα θεωρούμε σαν μεταβλητή. Τα υπόλοιπα θεωρούνται σαν σταθερές. Έτσι εάν η έκφραση δεν περιέχει t και ζητήσουμε από το Maple να παραγωγίσει ως προς t , το αποτέλεσμα είναι 0. Ακολουθεί ένα παράδειγμα:

```
>expr := a*sin(b*x);
```

$expr := a \sin(b x)$

```
>diff(expr, x);
```

$a \cos(b x) b$

```
>diff(expr, t);
```

0

Όταν παραγωγίζουμε με το Maple το αποτέλεσμα μερικές φορές μοιάζει διαφορετικό από αυτό που θα παίρναμε εάν λύναμε το πρόβλημα με το χέρι. Ας δούμε ένα παράδειγμα:

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

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

```
>deriv := diff(expr, x);
```

$$deriv := \frac{1}{(x-2)^2} - \frac{2x}{(x-2)^3}$$

Αν το λύναμε με το χέρι θα παίρναμε ένα μόνο κλάσμα. Μπορούμε να μετατρέψουμε την απάντηση του Maple σε ένα απλό κλάσμα χρησιμοποιώντας την εντολή **simplify** για να προσθέσουμε τα δύο κλάσματα.

```
>simplify(deriv);
```

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

5.2.2. Παραγωγή μιας συνάρτησης

Εάν θέλουμε να παραγωγίσουμε μια **συνάρτηση** (function) και το αποτέλεσμα να είναι μια νέα συνάρτηση, για την οποία η συνηθισμένη ονοματολογία να μπορεί να χρησιμοποιηθεί, πρέπει να χρησιμοποιήσουμε τον τελεστή **D**. Ας ορίσουμε μια συνάρτηση f με $f(t)=\sin(t^3)\cos(e^t)$, την παραγωγίζουμε, και ονομάζουμε την συνάρτηση που προκύπτει **fprime**.

```
>f := t -> sin(t^3)*cos(exp(t));
```

$$f := t \rightarrow \sin(t^3) \cos(e^t)$$

```
>fprime := D(f);
```

$$fprime := t \rightarrow 3 \cos(t^3) t^2 \cos(e^t) - \sin(t^3) \sin(e^t) e^t$$

Δεν χρειάζεται να προσδιορίσουμε τη μεταβλητή t . Για μια συνάρτηση το όνομα της μεταβλητής είναι ορισμένο. Επίσης, αυτή τη φορά, το "**fprime**" είναι το όνομα μιας συνάρτησης και όχι έκφρασης. Επομένως μπορούμε να βρούμε την τιμή της **fprime** στο $t = \frac{p}{4}$ ως εξής:

```
>fprime(Pi/4);
```

$$\frac{3}{16} \cos\left(\frac{\pi^3}{64}\right) \pi^2 \cos\left(e^{\left(\frac{\pi}{4}\right)}\right) - \sin\left(\frac{\pi^3}{64}\right) \sin\left(e^{\left(\frac{\pi}{4}\right)}\right) e^{\left(\frac{\pi}{4}\right)}$$

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

-1.784707437

5.3. Υπολογισμός Αθροισμάτων

Για να υπολογίσουμε αθροίσματα με το Maple, χρησιμοποιούμε την εντολή **sum(s(i),i=0..N)**, όπως έχουμε ήδη αναφέρει και στην ενότητα 2.8. Παρακάτω θα δούμε κάποια ενδεικτικά παραδείγματα:

Παράδειγμα 1

```
>sum(i^a,i=0..10);
```

$$1 + 2^a + 3^a + 4^a + 5^a + 6^a + 7^a + 8^a + 9^a + 10^a$$

```
>sum(k^2,k=0..5);
```

55

```
>sum((1/k),k=1..infinity);
```

∞

```
>sum(1/n!,n=0..infinity);
```

e

```
>sum(1/i^2,i=1..infinity);
```

$\frac{\pi^2}{6}$

Παράδειγμα 2

Ορίζουμε στο Maple το εξής άθροισμα $\sum_{i=0}^N F\left(\frac{i}{N}\right)$ σαν συνάρτηση του N , όπου η $F(x)$ είναι κάποια συνάρτηση:

>S:=N->sum(F(i/N),i=0...N);

$$S := N \rightarrow \sum_{i=0}^N F\left(\frac{i}{N}\right)$$

Έχουμε:

>S(1);

$$F(1)$$

>S(2);

$$\frac{1}{2}F\left(\frac{1}{2}\right) + \frac{1}{2}F(1)$$

>S(3);

$$\frac{1}{3}F\left(\frac{1}{3}\right) + \frac{1}{3}F\left(\frac{2}{3}\right) + \frac{1}{3}F(1)$$

Έστω τώρα ότι:

>F:=x->x^3;

$$F := x \rightarrow x^3$$

Τότε:

>S(1);

$$1$$

>S(2);

$$\frac{9}{16}$$

>S(3);

$$\frac{4}{9}$$

>limit(S(N),N=infinity);

$$\frac{1}{4}$$

5.4. Ολοκληρώματα

Για να υπολογίσουμε το αόριστο ολοκλήρωμα $\int f(x) dx$ γράφουμε

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

Και για το ορισμένο ολοκλήρωμα $\int_a^b f(x) dx$ γράφουμε

```
>int(f(x),x=a..b);
```

Παράδειγμα 1

```
>f:=t->t^2;
```

$$f := t \rightarrow t^2$$

```
>int(f(t),t);
```

$$\frac{t^3}{3}$$

```
>diff(%,t);
```

$$t^2$$

```
>int(f(t),t=0..2);
```

$$\frac{8}{3}$$

```
>int(t^2,t=0..2);
```

$$\frac{8}{3}$$

Με το Maple μπορούμε να υπολογίσουμε και ολοκληρώματα των οποίων οι απαντήσεις περιέχουν γνωστές (σε ορισμένους) μαθηματικές συναρτήσεις.

Παράδειγμα 2

```
>int(sin(t^2),t=0..2);
```

$$\frac{1}{2} \operatorname{FresnelS}\left(\frac{2\sqrt{2}}{\sqrt{\pi}}\right) \sqrt{2} \sqrt{\pi}$$

Για να δούμε την πιο πάνω τιμή σε δεκαδική μορφή λέμε:

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

```
0.8047764895
```

Θα μπορούσαμε να λέγαμε απευθείας

```
>evalf(int(sin(t^2),t=0..2));
```

```
0.8047764895
```

αλλά πρέπει να έχουμε υπόψη μας ότι αυτός ο συνδυασμός εντολών θα δώσει μια προσέγγιση στο ολοκλήρωμα όταν αυτό δεν μπορεί να υπολογιστεί αναλυτικά. Φυσικά, ορισμένες φορές αυτό είναι επιθυμητό, αλλά όταν δεν είναι πρέπει να είμαστε προσεχτικοί. Ας δούμε για παράδειγμα τη διαφορά στα πιο κάτω:

```
>evalf(int(sin(t^2),t=0..2));
```

```
0.8047764895
```

```
>int(sin(t^2),t=0.0..2.0);
```

```
0.8047764893
```

Όταν το Maple δεν μπορεί να υπολογίσει κάποιο ολοκλήρωμα, τότε μας επιστρέφει σαν απάντηση αυτό που ζητάμε.

Παράδειγμα 1

```
>int(sin(t)^(1/3), t=0..2);
```

$$\int_0^2 \sin(t)^{(1/3)} dt$$

Φυσικά, σε αυτή την περίπτωση, μια και έχουμε ορισμένο ολοκλήρωμα, θα μπορούσαμε να χρησιμοποιήσουμε την εντολή **evalf** που θα μας δώσει μια προσέγγιση για την τιμή του.

ΣΗΜΕΙΩΣΗ: Μπορούμε να υπολογίσουμε αόριστα και ορισμένα πολλαπλά ολοκληρώματα με την εντολή `int`, ως εξής:

$\iint f(x, y) dx dy$	<code>>int(int(f(x,y),x),y);</code>
$\int_c^d \int_a^b f(x, y) dx dy$	<code>>int(int(f(x,y),x=a..b),y=c..d);</code>

Παράδειγμα 2

`>int(int(x*y,x),y);`

$$\frac{x^2 y^2}{4}$$

`>int(int(x*exp(y),x=0..2),y=0..3);`

$$2 e^3 - 2$$

Κεφάλαιο 6

Πίνακες και Διανύσματα

Στο κεφάλαιο αυτό θα αναφερθούμε στις πράξεις που εκτελούμε στους πίνακες και στα διανύσματα, στις εντολές που χρησιμοποιούμε και επιπλέον στο πακέτο που απαιτείτε να φορτώσουμε για την επίλυση τους. Για περισσότερες πληροφορίες μπορούμε να ανατρέξουμε (Ξενοφώντος & Γεωργίου, 2007).

6.1. Εισαγωγή πινάκων

Το Maple είναι ένα λογισμικό που χρησιμοποιείται επίσης και για επεξεργασία πινάκων και διανυσμάτων. Στην συγκεκριμένη περίπτωση είναι απαραίτητο να χρησιμοποιήσουμε το πακέτο **linalg** γιατί περιέχει εξειδικευμένες εντολές και αν δεν το φορτώσουμε δεν μπορούμε να προχωρήσουμε. Στο Κεφάλαιο 9, γίνεται εκτενέστερη ανάλυση για τα πακέτα. Για να φορτώσουμε το πακέτο πληκτρολογούμε `with` (όνομα πακέτου),όπως φαίνεται και πιο κάτω:

```
>with(linalg);
```

```
Warning, the protected names norm and trace have been redefined and  
unprotected
```

[*BlockDiagonal, GramSchmidt, JordanBlock, LUdecomp, QRdecomp, Wronskian, addcol, addrow, adj, adjoint, angle, augment, backsub, band, basis, bezout, blockmatrix, charmat, charpoly, cholesky, col, coldim, colspace, colspan, companion, concat, cond, copyinto, crossprod, curl, definite, delcols, delrows, det, diag, diverge, dotprod, eigenvals, eigenvalues, eigenvectors, eigenvects, entermatrix, equal, exponential, extend, ffgausselim, fibonacci, forwardsub, frobenius, gausselim, gaussjord, geneqns, genmatrix, grad, hadamard, hermite, hessian, hilbert, htranspose, ihermite, indexfunc, innerprod, intbasis, inverse, ismith, issimilar, iszero, jacobian, jordan, kernel, laplacian, leastsqrs, linsolve, matadd, matrix, minor, minpoly, mulcol, mulrow, multiply, norm, normalize, nullspace, orthog, permanent, pivot, potential, randmatrix, randvector, rank, ratform, row, rowdim, rowspace, rowspan, rref, scalarmul, singularvals, smith, stackmatrix, submatrix, subvector, subbasis, swapcol, swaprow, sylvester, toeplitz, trace, transpose, vandermonde, vecpotent, vectdim, vector, wronskian*]

Η λίστα που παίρνουμε περιέχει όλες τις εντολές που μας είναι διαθέσιμες με αυτό το πακέτο. Αν δεν θέλουμε να τις δούμε, βάζουμε ‘ : ’ αντί για ‘;’ στο τέλος. Αφού φορτώσουμε το πακέτο, μπορούμε για παράδειγμα να ορίσουμε ένα πίνακα:

$$A = \begin{bmatrix} 2 & 1 \\ -1 & 5 \end{bmatrix}$$

με δύο ισοδύναμους τρόπους:

```
>A:=matrix([[2,1],[-1,5]]);
```

$$A := \begin{bmatrix} 2 & 1 \\ -1 & 5 \end{bmatrix}$$

```
>A:=matrix(2,2,[2,1,-1,5]);
```

$$A := \begin{bmatrix} 2 & 1 \\ -1 & 5 \end{bmatrix}$$

Αν θέλουμε να δούμε τον πίνακα A, λέμε:

```
>evalm(A);
```

$$\begin{bmatrix} 2 & 1 \\ -1 & 5 \end{bmatrix}$$

(Το **evalm** προέρχεται από το **evaluatematrix**)

Αν πούμε απλώς A θα πάρουμε:

```
>A;
```

A

Για να δούμε το στοιχείο του πίνακα A που βρίσκεται στη θέση (2,1) λέμε:

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

-1

Ανπούμε:

```
>A[3,1];
```

Error, 1st index, 3, larger than upper array bound 2

Παίρνουμε λάθος γιατί ο πίνακας μας είναι 2x2.

Ορίζουμε επίσης:

```
>B:=matrix([[a,-b],[c,1/d]]);
```

$$B := \begin{bmatrix} a & -b \\ c & \frac{1}{d} \end{bmatrix}$$

```
>C:=matrix([[7,-4,2],[9,5,4]]);
```

$$C := \begin{bmatrix} 7 & -4 & 2 \\ 9 & 5 & 4 \end{bmatrix}$$

Για να προσθέσουμε πίνακες πρέπει να πούμε:

```
>evalm(A+B);
```

$$\begin{bmatrix} 2+a & 1-b \\ -1+c & 5+\frac{1}{d} \end{bmatrix}$$

Αν πούμε απλά A+B θα πάρουμε:

```
>A+B;
```

A + B

Το ίδιο ισχύει και για τις υπόλοιπες πράξεις.

```
>evalm(C+B);
```

Error, (in linalg[matadd]) matrix dimensionw incompatible

Για να πολλαπλασιάσουμε πίνακες λέμε:

```
>multiply(A,C);
```

$$\begin{bmatrix} 23 & -3 & 8 \\ 38 & 29 & 18 \end{bmatrix}$$

>**multiply(C,A)**

Error, (in multiply) non matching dimensions for vector/matrix product

Μπορούμε, ισοδύναμα, να πούμε:

>**evalm(A&*B);**

$$\begin{bmatrix} 2a+c & -2b+\frac{1}{d} \\ -a+5c & b+\frac{5}{d} \end{bmatrix}$$

όπου τα σύμβολα **&*** υποδηλώνουν τον πολλαπλασιασμό.

Ο ανάστροφος ενός πίνακα δίνεται από την εντολή **transpose**:

>**transpose(B);**

$$\begin{bmatrix} a & c \\ -b & \frac{1}{d} \end{bmatrix}$$

Η ορίζουσα ενός πίνακα δίνεται από την εντολή **det**:

>**det(A);**

11

>**det(B);**

$$\frac{a+bcd}{d}$$

Ο αντίστροφος ενός πίνακα υπολογίζεται με διάφορους τρόπους:

>**inverse(A);**

$$\begin{bmatrix} \frac{5}{11} & \frac{-1}{11} \\ \frac{1}{11} & \frac{2}{11} \end{bmatrix}$$

```
>evalm(1/A);
```

$$\begin{bmatrix} \frac{5}{11} & \frac{-1}{11} \\ \frac{1}{11} & \frac{2}{11} \end{bmatrix}$$

```
>evalm(A^(-1));
```

$$\begin{bmatrix} \frac{5}{11} & \frac{-1}{11} \\ \frac{1}{11} & \frac{2}{11} \end{bmatrix}$$

6.2. Διανύσματα

Τα διανύσματα ορίζονται με παρόμοιο τρόπο με τους πίνακες.

```
>v:=vector(3,[1,cos(t),sin(t)]);
```

$$v := [1, \cos(t), \sin(t)]$$

```
>evalm(v);
```

$$[1, \cos(t), \sin(t)]$$

```
>u:=vector(3,[-1,cos(t),sin(t)]);
```

$$u := [-1, \cos(t), \sin(t)]$$

Το εσωτερικό γινόμενο δύο διανυσμάτων δίνεται από την εντολή **angle**:

```
>angle(u,v);
```

$$\arccos\left(\frac{-1 + \cos(t)^2 + \sin(t)^2}{1 + \cos(t)^2 + \sin(t)^2}\right)$$

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

$$\frac{\pi}{2}$$

Η 2η-όραμα ενός διανύσματος δίνεται από την εντολή **norm**:

```
>norm(u,2);
```

$$\sqrt{1 + |\cos(t)|^2 + |\sin(t)|^2}$$


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

$$\sqrt{2}$$

Το εξωτερικό δύο διανυσμάτων του R^3 δίνεται από την εντολή **crossprod**:

```
>crossprod(u,v);
```

$$[0, 2 \sin(t), -2 \cos(t)]$$

Τέλος, για επαλήθευση χρησιμοποιούμε την εντολή **multiplay**.

6.3. Επίλυση γραμμικού συστήματος

Για να λύσουμε ένα γραμμικό σύστημα θα φορτώσουμε πρώτα το πακέτο **linalg**όπως αναφέραμε παραπάνω.

Έστω ότι θέλουμε να λύσουμε είναι: $2x+2y-z=1$, $3x+4y+2z=-2$, $x+4y+3z=3$. Θα ορίσουμε τον πίνακα και το διάνυσμα και στη συνέχεια θα το λύσουμε με δύο διαφορετικούς τρόπους:

```
>A:=matrix(3,3,[2,2,-1,3,4,2,1,4,3]);
```

$$A := \begin{bmatrix} 2 & 2 & -1 \\ 3 & 4 & 2 \\ 1 & 4 & 3 \end{bmatrix}$$

```
>b:=vector(3,[1,-2,3]);
```

$$b := [1, -2, 3]$$

Η λύση του συστήματος $Ax=b$ μπορεί να βρεθεί με διάφορους τρόπους. Ο πιο απλός είναι μέσω της εντολής **linsolve**:

```
>x:=linsolve(A,b);
```

$$x := \left[\frac{-24}{7}, 3, \frac{-13}{7} \right]$$

Επαληθεύουμε:

```
>multiply(A,x);
```

[1, -2, 3]

Μπορούμε να τυπώσουμε το **x** ή κάποιο από τα στοιχεία του στην οθόνη, με τον ίδιο τρόπο όπως και πριν:

```
>evalm(x);
```

$\left[\frac{-24}{7}, 3, \frac{-13}{7} \right]$

```
>x[3];
```

$\frac{-13}{7}$

Ένας άλλος τρόπος εξεύρεσης της λύσης του συστήματος **Ax=b** είναι ο παρακάτω:

```
>x:=multiply(inverse(A),b);
```

$x := \left[\frac{-24}{7}, 3, \frac{-13}{7} \right]$

Αυτό όμως δεν συστήνεται στην πράξη καθώς, όπως βλέπουμε εισάγουμε μαζί και την εντολή multiply και την inverse με αποτέλεσμα να εμφανίζεται στην οθόνη μας μόνο το αποτέλεσμα. Η επαλήθευση ενώ γίνεται δεν φαίνεται.

Μπορούμε ακόμη να κατασκευάσουμε τον επαυξημένο πίνακα **[A|b]** με την εντολή **augment**:

```
>A_b:=augment(A,b);
```

$A_b := \begin{bmatrix} 2 & 2 & -1 & 1 \\ 3 & 4 & 2 & -2 \\ 1 & 4 & 3 & 3 \end{bmatrix}$

και να κάνουμε πρώτα απαλοιφή **Gauss** με την εντολή **gausselim**.

```
>A_b_elim:=gausselim(A_b);
```

$A_b_elim := \begin{bmatrix} 2 & 2 & -1 & 1 \\ 0 & 1 & \frac{7}{2} & \frac{-7}{2} \\ 0 & 0 & -7 & 13 \end{bmatrix}$

και μετά ανάδρομη αντικατάσταση με την εντολή **backsub**.

```
>x:=backsub(A_b_elim);
```

$$x := \left[\frac{-24}{7}, 3, \frac{-13}{7} \right]$$

Ένας άλλος τρόπος είναι μέσω της ανοιγμένης κλιμακωτής μορφής του επαυξημένου πίνακα (απαλοιφή Gauss-Jordan) και η εντολή για αυτή τη μέθοδο είναι **rref**.

>rref(A_b);

$$\begin{bmatrix} 1 & 0 & 0 & \frac{-24}{7} \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & \frac{-13}{7} \end{bmatrix}$$

6.4. Επίλυση προβλημάτων με διανύσματα

Ας δούμε μερικά παραδείγματα.

Παράδειγμα 1

Έστω $u_1=(1,2,0)$, $u_2=(1,0,2)$, $u_3=(4,-1,1)$, και θέλουμε να γράψουμε το διάνυσμα $u=(3,-2,0)$ σαν γραμμικός συνδυασμός των u_1, u_2, u_3 .

Θεωρούμε ότι το διάνυσμα u είναι γραμμικός συνδυασμός των u_1, u_2, u_3 και υπάρχουν σταθερές a, b, c τέτοιες ώστε $u=au_1+bu_2+cu_3$. Δηλαδή: $(3,-2,0)=a(1,2,0)+b(1,0,2)+c(4,-1,1)$. Εξισώνοντας τις συντεταγμένες παίρνουμε το πιο κάτω σύστημα:

>e1:=3=a+b+4*c;

$$e1 := 3 = a + b + 4c$$

>e2:=-2=2*a-c;

$$e2 := -2 = 2a - c$$

>e3:=0=2*b+c;

$$e3 := 0 = 2b + c$$

>solve({e1,e2,e3},{a,b,c});

$$\left\{ b = \frac{-1}{2}, a = \frac{-1}{2}, c = 1 \right\}$$

Άρα: $u = -1/2 u_1 - 1/2 u_2 + u_3$

Παράδειγμα 2

Έστω $u = (-1, 3, -2)$, και θέλουμε να βρούμε:

α) το διάνυσμα που έχει την ίδια κατεύθυνση με το u αλλά διπλάσιο μήκος από το u .

β) το διάνυσμα που έχει την αντίθετη κατεύθυνση με το u αλλά τριπλάσιο μήκος από το u . Θα ορίσουμε πρώτα το διάνυσμα u :

```
>u:=[-1,3,-2];
```

```
u := [-1, 3, -2]
```

α)

```
>2*u;
```

```
[-2, 6, -4]
```

β)

```
>-3*u;
```

```
[3, -9, 6]
```

Παράδειγμα 3

Έστω ότι θέλουμε να βρούμε το σημείο στο οποίο οι ευθείες $x=2-t$, $y=3t$, $z=-1+2t$ τέμνουν το επίπεδο $2y+3z=6$. Αρχικά θα ορίσουμε τις ευθείες:

```
> x:=2-t;
```

```
x := 2 - t
```

```
>y:=3*t;
```

```
y := 3 t
```

```
>z:=-1+2*t;
```

```
z := -1 + 2 t
```

Το σημείο τομής ικανοποιεί τις εξισώσεις της ευθείας του επιπέδου. Αντικαθιστούμε τις εξισώσεις της ευθείας στην εξίσωση του επιπέδου:

```
>t:=solve(2*y+3*z=6,t);
```

```
t :=  $\frac{3}{4}$ 
```

Άρα το ζητούμενο σημείο έχει συντεταγμένες:

>x;

$$\frac{5}{4}$$

>y;

$$\frac{9}{4}$$

>z;

$$\frac{1}{2}$$

Παράδειγμα 4

Έστω ότι θέλουμε να υπολογίσουμε τη γωνία των ευθειών που έχουν τις παρακάτω εξισώσεις:

$x=t$, $y=2+2t$, $z=1+2t$ και $x=3+3t$, $y=4t$, $z=3-12t$. Θα φορτώσουμε πρώτα το πακέτο:

>with(linalg):

Warning, the protected names norm and trace have been redefined and unprotected

>a:=vector([1,2,2]);

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

>b:=vector([3,4,-12]);

$$b := [3, 4, -12]$$

Άρα το συνημίτονο της γωνίας δίνεται από τον τύπο:

>cos(theta):=dotprod(a,b)/(sqrt(dotprod(a,a))*sqrt(dotprod(b,b)));

$$\cos(\theta) := \frac{-1}{3}$$

Δηλαδή $\theta = \arccos\left(\frac{-1}{3}\right)$.

Παράδειγμα 5

Έστω ότι θέλουμε να βρούμε την εξίσωση του επιπέδου που διέρχεται από το σημείο $(-1, 2, -5)$ και είναι κάθετο στα επίπεδα $2x - y + z = 1$ και $x + y - 2z = 3$. Πρώτα παίρνουμε τα κάθετα διανύσματα των επιπέδων:

```
>with(linalg):
```

```
Warning, the protected names norm and trace have been redefined and unprotected
```

```
>v1:=vector([2,-1,1]);
```

```
v1 := [2, -1, 1]
```

```
>v2:=vector([1,1,-2]);
```

```
v2 := [1, 1, -2]
```

```
>n:=crossprod(v1,v2);
```

```
n := [1, 5, 3]
```

```
>r:=vector([x,y,z]);
```

```
r := [x, y, z]
```

```
>p:=vector([-1,2,-5]);
```

```
p := [-1, 2, -5]
```

Η εξίσωση του επιπέδου είναι:

```
>dotprod(r,n)=dotprod(p,n);
```

```
x + 5 y + 3 z = -6
```

Κεφάλαιο 7

Γραφήματα

7.1.Εισαγωγή

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τις δυνατότητες που μας δίνει το Maple για να σχεδιάζουμε διάφορα γραφήματα, αυτά μπορεί να είναι απλά γραφήματα(plot), τρισδιάστατα(plot3d) ή γραφήματα με κίνηση(animation). Μπορούμε να σχεδιάζουμε περισσότερες από μια καμπύλες στο ίδιο γράφημα καθώς και άξονες, παραμετρικές καμπύλες, το σημείο τομής δύο καμπύλων αλλά και κάθετες ασύμπτωτες(Waterloo Maple Inc, 2002),(Δημοτίκαλης, 2000).

7.2. Λίστες

Πριν παρουσιάσουμε το πως δημιουργούμε γραφήματα πρέπει να ξέρουμε τις **λίστες** (lists). Στο Maple μια **λίστα** είναι μια συλλογή από αντικείμενα που χωρίζονται με κόμμα και περικλείονται σε αγκύλες:

[αντικείμενο 1, αντικείμενο 2, ...]

Πρέπει να θυμίσουμε ότι ένα **σύνολο** (set) είναι μια συλλογή αντικειμένων χωρισμένων με κόμμα και περικλειόμενα με άγκιστρα:

{αντικείμενο 1,αντικείμενο 2,...}

Μια λίστα διαφέρει από ένα σύνολο σε διάφορα σημεία. Το ένα είναι ότι η σειρά των πραγμάτων σε μια λίστα είναι σημαντική για το Maple, ενώ η σειρά των αντικειμένων σε ένα σύνολο όχι. Για παράδειγμα, **{x,y}** και **{y,x}** είναι το ίδιο σύνολο=, αλλά **[x,y]** και **[y,x]** δεν είναι η ίδια λίστα.

Μπορούμε να αναφερθούμε στα επιμέρους στοιχεία μιας λίστας με τον ίδιο τρόπο που αναφερόμαστε στα αντικείμενα ενός συνόλου. Εάν το όνομα της λίστας είναι **L**, τότε το πρώτο αντικείμενο της λίστας είναι **L[1]**, το δεύτερο είναι **L[2]**, κ.ο.κ.

Μέχρι τώρα έχουμε παρουσιάσει τρία διαφορετικά σύμβολα ομαδοποίησης: **()**, **{ }**, και **[]**. Τα οποία χρησιμοποιούνται για διαφορετικούς στόχους.

Οι κανονικές παρενθέσεις **()** χρησιμοποιούνται:

- για να ομαδοποιήσουν πράγματα σε υπολογισμούς,
- για εντολές του Maple όπως **factor(x²-1)**, και
- για ενσωματωμένες συναρτήσεις όπως **sin(Pi)**.

Τα άγκιστρα **{ }** χρησιμοποιούνται για να ομαδοποιήσουν μαζί αντικείμενα όπως **{x,y}**.

Οι αγκύλες **[]** χρησιμοποιούνται:

- για να ομαδοποιήσουν αντικείμενα σε μια λίστα, όπως **[x,y]**, και
- για να επιλέξουν ένα αντικείμενο από μια συλλογή, όπως **sol[1]**.

7.3. Σχεδίαση Γραφημάτων

Για τη σχεδίαση των γραφημάτων χρησιμοποιούμε την εντολή **plot**.

plot(έκφραση που περιέχει το x, x=a..b)

Δημιουργεί ένα γράφημα της έκφρασης όπου η τιμή του **x** είναι στο διάστημα **[a,b]**. Το Maple επιλέγει το διάστημα που θα εμφανίσει στον **y**-άξονα.

plot(έκφραση που περιέχει το x, x=a..b, y=c..d)

Είναι παρόμοια με την προηγούμενη αλλά το διάστημα στον **y**-άξονα είναι **[c,d]**.

Μπορούν να γίνουν ταυτόχρονα γραφήματα εκφράσεων που περιλαμβάνουν το x στους ίδιους άξονες χρησιμοποιώντας μια λίστα εκφράσεων στη θέση μιας έκφρασης:

[έκφραση1, έκφραση2, ...]

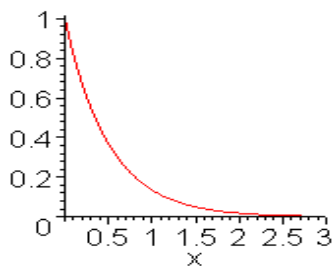
7.3.1.Σχεδίαση απλής έκφρασης

Ας σχεδιάσουμε την έκφραση e^{-2x} στο διάστημα $[0,3]$.

```
>expr := exp(-2*x);
```

$$expr := e^{(-2x)}$$

```
>plot(expr, x=0..3);
```



Σχήμα 13: Σχεδίαση της έκφρασης e^{-2x}

Ας δοκιμάσουμε να σχεδιάσουμε $y=(x-2)^{(2/3)}$ στο διάστημα $[0, 10]$.

Εκφράσεις όμως που περιέχουν κλασματικές δυνάμεις μπορεί να μην σχεδιαστούν σωστά, γιατί δεν μπορεί να υπολογιστεί η ρίζα ενός αρνητικού αριθμού χρησιμοποιώντας το $^$. Για αυτό το λόγο χρησιμοποιούμε το **surd** στη θέση του.

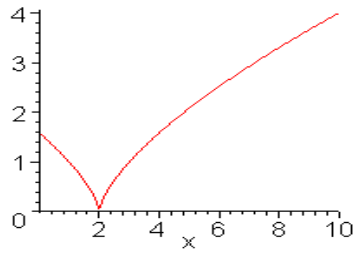
```
>expr := (x-2)^(2/3);
```

$$expr := (x - 2)^{(2/3)}$$

```
>newexpr := convert(expr, surd);
```

$$newexpr := \text{surd}(x - 2, 3)^2$$

```
>plot(newexpr, x=0..10);
```



Σχήμα 14: Σχεδίαση της απλής έκφρασης $(x-2)^{(2/3)}$

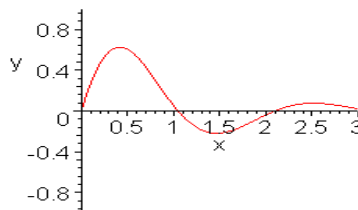
7.3.2. Επιλέγοντας το y-διάστημα

Στα προηγούμενο γράφημα, το Maple επέλεξε το διάστημα που θα εμφανίσει στον y - άξονα. Μπορούμε να κάνουμε την επιλογή εμείς αν θέλουμε.

```
>f := x ->exp(-x)*sin(3*x);
```

$$f := x \rightarrow e^{(-x)} \sin(3x)$$

```
>plot(f(x), x=0..3, y=-1..1);
```



Σχήμα 15: Σχεδίαση της έκφρασης $e^{(-x)} \sin(3x)$

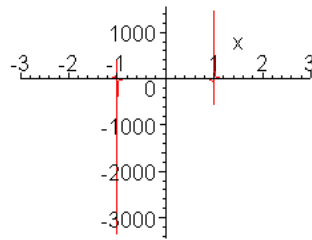
7.3.3. Κάθετες ασύμπτωτες

Ας δούμε πως το Maple σχεδιάζει μια συνάρτηση που το γράφημά της έχει κάθετες ασύμπτωτες.

```
>h := x -> (2*x^2+1)/(x^2-1);
```

$$h := x \rightarrow \frac{2x^2 + 1}{x^2 - 1}$$

```
>plot(h(x), x=-3..3);
```



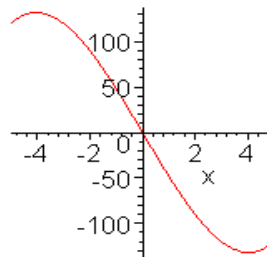
Σχήμα 16: Σχεδίαση της συνάρτησης $2x^2+1/x^2-1$

7.3.4.Τοπικά Ακρότατα

```
>f := x ->x^3-49*x;
```

$$f := x \rightarrow x^3 - 49x$$

```
>plot(f(x),x=-5..5);
```



Σχήμα 17: Σχεδίαση συνάρτησης x^3-49x

Από την άποψη του γραφήματος μπορούμε να δούμε ότι υπάρχουν τοπικά ελάχιστο και μέγιστο.

7.4.Παράμετροι της εντολής plot

Μπορούμε να χρησιμοποιήσουμε όσες επιλογές (παραμέτρους) γραφημάτων χωρισμένες με κόμμα μεταξύ τους:

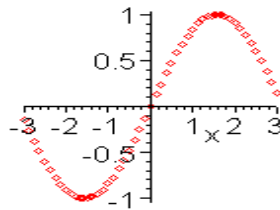
plot(έκφραση που περιέχει το x, x=a..b, y=c..d, επιλογή1, επιλογή2, ...)

Ο προσδιορισμός του διαστήματος y –άξονα είναι προαιρετικός.

7.4.1.Στύλ

Το Maple δημιουργεί τα γραφήματα υπολογίζοντας την τιμή μιας έκφρασης σε ένα αριθμό σημείων και στη συνέχεια ενώνοντας τα σημεία. Εάν θέλουμε να δούμε μόνο τα σημεία που υπολογίζει το Maple θα χρησιμοποιήσουμε την επιλογή της `plotstyle=point`. Η μόνη άλλη επιλογή στο `style` που χρειάζεται να ξέρουμε είναι η `style=line` οποία ενώνει τα σημεία και είναι η προεπιλεγμένη.

```
>plot(sin(x), x=-3..3, style=point);
```



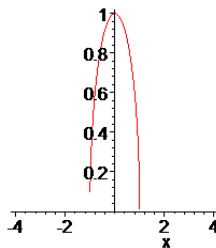
Σχήμα 18: Σχεδίαση της παράστασης $\sin(x)$ με Style

7.4.2.Αριθμός σημείων και κλίμακα

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

$$g := x \rightarrow \sqrt{1 - x^2}$$

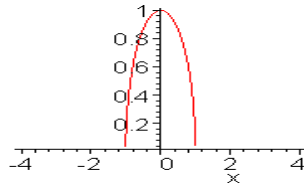
```
>plot(g(x), x=-4.2..4.2);
```



Σχήμα 19: Σχεδίαση της παράστασης $\sqrt{1-x^2}$ με `numpoints`

Ένα μέρος του γραφήματος (καμπύλης) λείπει. Μπορούμε να διορθώσουμε αυτό το πρόβλημα χρησιμοποιώντας την επιλογή της `plotnumpoints`:

```
>plot(g(x), x=-4.2..4.2, numpoints=400);
```

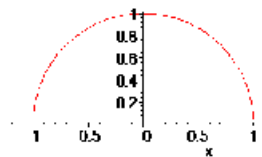


Σχήμα 20: Σχεδίαση της παράστασης $\sqrt{1-x^2}$ με **scaling**

Δίνοντας **numpoints=400** εξαναγκάζουμε το Maple να προσθέσει περισσότερα σημεία στο γράφημα. Η προεπιλεγμένη τιμή της numpoints είναι 50 (Στην πραγματικότητα, **numpoints** είναι το ελάχιστο πλήθος των σημείων που σχεδιάζει το Maple. Αυτόματα υπολογίζει περισσότερα σημεία σε περιοχές που η καμπύλη έχει μεγαλύτερη κλίση).

Παρατηρούμε ότι το γράφημα δεν μοιάζει με ημικύκλιο. Ο λόγος είναι ότι η κλίμακα στον x-άξονα και y-άξονα είναι διαφορετική. Η επιλογή γραφήματος **scaling=constrained** ορίζει ότι οι κλίμακες μέτρησης στους άξονες x και y θα είναι ίδιες.

```
>plot(g(x), x=-1.2..1.2, numpoints=100, scaling=constrained);
```

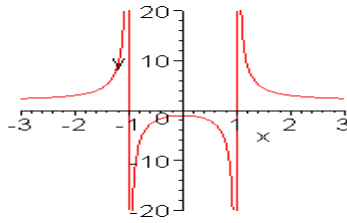


Σχήμα 21: Γράφημα της $\sqrt{1-x^2}$ με κλίμακες μέτρησης

7.4.3. Ασυνέχεια

Η εντολή `discont=true` χρησιμοποιείται όταν η συνάρτηση είναι ασυνεχής σε κάποιο σημείο έτσι ώστε να μην εμφανίζεται μια συνεχής γραμμή στο σημείο της ασυνέχειας, όπως στο παράδειγμα.

```
>plot((2*x^2+1)/(x^2-1), x=-3..3, y=-20..20);
```



Σχήμα 22: Σχεδίαση της συνάρτησης $2x^2+1/(x^2-1)$

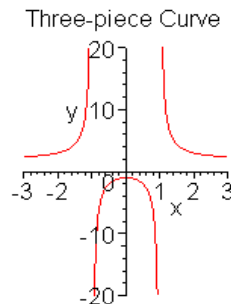
7.4.4. Προσθήκη τίτλου και κειμένου σε γράφημα

Το παρακάτω παράδειγμα δείχνει πως προσθέτουμε τίτλο (**title**) που είναι μια ακόμα επιλογή της `plot`. Ο τίτλος (το κείμενο) θα πρέπει να περικλείεται σε εισαγωγικά.

```
>h := x -> (2*x^2+1)/(x^2-1);
```

$$h := x \rightarrow \frac{2x^2 + 1}{x^2 - 1}$$

```
>plot(h(x), x=-3..3, y=-20..20, discontin=true, title=`Three-
piece Curve`);
```



Σχήμα 23: Προσθήκη τίτλου στην παράσταση $2x^2+1/(x^2-1)$

Για να τυπώσουμε ένα κείμενο μέσα σε ένα γράφημα χρησιμοποιούμε την εντολή **textplot** και **display** που βρίσκονται στο πακέτο **plots**. Το επόμενο παράδειγμα δείχνει πως γίνεται αυτό.

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

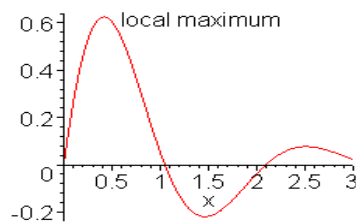
```
>f := x -> exp(-x)*sin(3*x);
```

$$f := x \rightarrow e^{-x} \sin(3x)$$

```
>Plot1 := plot(f(x), x=0..3):
```

```
>Plot2 := textplot([0.6, 0.62, `local maximum`], align=RIGHT):
```

```
>display({Plot1, Plot2});
```

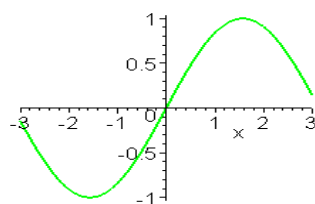


Σχήμα 24: Γράφημα της $e^{(-x)} \sin(3x)$ με προσθήκη κειμένου

ΣΗΜΕΙΩΣΗ: η τρίτη και τέταρτη εντολή τελειώνουν με άνω και κάτω τελεία, διαφορετικά το Maple θα εμφάνιζε στην οθόνη πληροφορίες για το plotstructure. Αυτές οι δύο εντολές ορίζουν δύο plotstructures και τις ονομάζουν. Η **textplot** στην δεύτερη εντολή εμφανίζει το κείμενο "localmaximum" στο σημείο (0.6,0.62) του γραφήματος. Προσοχήστη χρήση των αγκυλών και αγκίστρων. Η εντολή **align=RIGHT** ευθυγραμμίζει το κείμενο δεξιά από το σημείο (0.6, 0.62). άλλες δυνατότητες της **align**είναι **LEFT**(αριστερά), **ABOVE** (από πάνω), και **BELOW** (από κάτω). Η εντολή **display** χρησιμοποιείτε για να εμφανίσει ταυτόχρονα στο ίδιο γράφημα ένα σύνολο ή λίστα από plotstructures.

7.4.5.Χρώμακαιπάχοςγραμμής(color and thickness)

```
>plot(sin(x), x=-3..3,color=green, thickness=2);
```



Σχήμα 25: Γράφημα της $\sin(x)$ με χρώμα και πάχος γραμμής

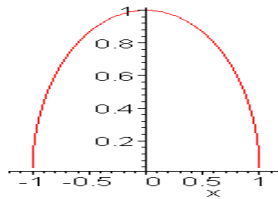
7.4.6. Το μενού plot, η εργαλειοθήκη και το contextbar

Υπάρχουν μερικές αλλαγές που μπορούμε να κάνουμε σε ένα γράφημα χωρίς να πληκτρολογήσουμε εντολές. Ας το δούμε έχοντας ένα γράφημα:

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

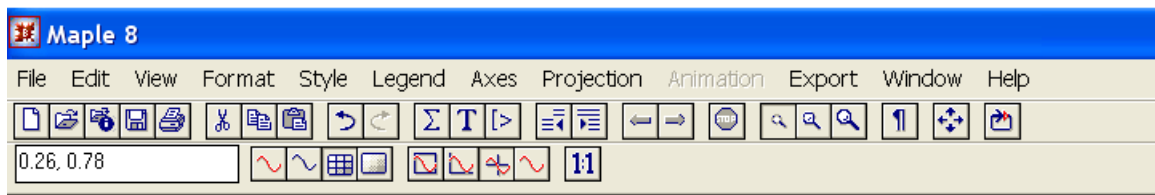
$$g := x \rightarrow \sqrt{1-x^2}$$

```
>plot(g(x), x=-1.2..1.2, numpoints=100);
```



Σχήμα 26: Γράφημα της συνάρτησης $g(x)=\sqrt{1-x^2}$

Αν τοποθετήσουμε το δείκτη του ποντικιού στο γράφημα και κάνουμε κλικ, ένα μαύρο πλαίσιο εμφανίζεται γύρω από το γράφημα και η γραμμή των μενού και η σχετική γραμμή εργαλείων (contextbar) αλλάζει και γίνεται όπως ακριβώς φαίνεται στο Σφάλμα! Άγνωστη παράμετρος αλλαγής..



Σχήμα 27:Σχεδίαση της $g(x)=\sqrt{1-x^2}$ από το contextbar

Πατώντας από το μενού το **Projection**,επιλέγουμε **Constrained**και μπορούμε να εξισώσουμε τις κλίμακες στον x-άξονα και y- άξονα.Εναλλακτικά πατάμε στο contextbar το κουμπί με ετικέτα **1:1**.

Πατώντας από το μενού το **Style**και επιλέγοντας**Point**μπορούμε να δούμε τα σημεία που υπολόγισε το Maple.Επιλέγοντας το **Symbol**μπορούμε να αλλάξουμε τα σημεία σε crosses (σταυρούς), diamonds (ρόμβους),point(σημεία)κλπ. Επιλέγοντας το **LineStyle**κάνουμε τη γραμμή dashed (διακεκομμένη), κλπ. Χρησιμοποιώντας τη **LineWidth** ελέγχουμε το πάχος.

Με το μενού **Axes** μπορούμε να αλλάξουμε την εμφάνιση των αξόνων.

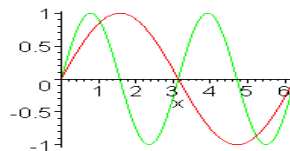
Για να επιστρέψουμε στη συνηθισμένη εμφάνιση του γραφήματος επιλέγουμε**Line** από το μενού **Style**.

7.5.Γενικά παραδείγματα

7.5.1.Σχεδιάζοντας πολλές εκφράσεις σε ένα γράφημα

Για να σχεδιάσουμε πολλές εκφράσεις σε ένα γράφημα και σύστημα συντεταγμένων, ο καλύτερος τρόπος είναι να δώσουμε στο Maple μια λίστα από εκφράσεις στην εντολή plot, παρά ένα σύνολο εκφράσεων. Η επόμενη εντολή δείχνει πως να γίνει αυτό:

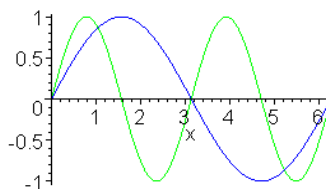
```
>plot([sin(x), sin(2*x)], x=0..2*Pi);
```



Σχήμα 28: Γράφημα των $\sin(x)$ και $\sin(2x)$

Το Maple έχει σχεδιάσει τις συναρτήσεις $\sin x$ και $\sin 2x$ στους ίδιους άξονες. Αλλά πως τις ξεχωρίζουμε? Για να είμαστε σίγουροι για την κάθε μία μπορούμε να δώσουμε μια λίστα χρωμάτων. Επειδή σε μια λίστα η σειρά είναι σημαντική, το Maple θα χρησιμοποιήσει το πρώτο χρώμα για την πρώτη συνάρτηση κ.ο.κ.

```
>plot([sin(x), sin(2*x)], x=0..2*Pi, color=[blue, green]);
```



Σχήμα 29: Σχεδίαση των $\sin(x)$ και $\sin(2x)$ με λίστα χρωμάτων

7.5.2.Σχεδίαση γραφήματος μιας εξίσωσης

Για να σχεδιάσουμε το γράφημα μιας εξίσωσης στο Maple, πρέπει να χρησιμοποιήσουμε την εντολή **implicitplot**, που είναι μέρος του πακέτου **plots**. Για να έχουμε πρόσβαση σε όλες τις εντολές του πακέτου **plots**, πρέπει πρώτα να πληκτρολογήσουμε την εντολή **with(plots)**.

```
>with(plots);
```

```
[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, cylinderplot, densityplot, display, display3d, fieldplot, fieldplot3d, gradplot, gradplot3d, graphplot3d, implicitplot, implicitplot3d, inequal, interactive, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, replot, rootlocus, semilogplot, setoptions, setoptions3d, spacecurve, sparsematrixplot, sphereplot, surfdata, textplot, textplot3d, tubeplot]
```

Το αποτέλεσμα είναι μια λίστα από όλες τις εντολές στο πακέτο **plots**.

Στην εντολή **implicitplot** πρέπει να καθορίζουμε και το x και το y-διάστημα.

Πρέπει αρχικά να δώσουμε ένα όνομα στην εξίσωσή μας, και μετά να χρησιμοποιήσουμε την εντολή **implicitplot** για να σχεδιάσουμε το γράφημα.

```
>eqn := x^3+y^3=6*x*y;
```

$$eqn := x^3 + y^3 = 6xy$$

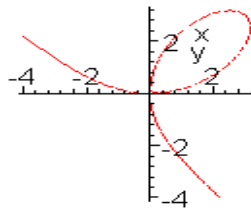
```
>implicitplot(eqn, x=-4..4, y=-4..4);
```



Σχήμα 30: Σχεδίαση της $x^3+y^3=6xy$ με την **implicitplot**

Το γράφημα είναι λίγο ασαφές, ειδικά κοντά στην αρχή των αξόνων, όπου η καμπύλη τέμνει τον εαυτό της. Η εντολή **implicitplot** στην πραγματικότητα δοκιμάζει τις τιμές της συνάρτησης σε ένα πίνακα (grid) 25x25 και χρησιμοποιεί τα αποτελέσματα για να δημιουργήσει το γράφημα. Εάν το γράφημα δεν είναι ευκρινές, μπορείτε να αυξήσετε τον αριθμό των σημείων στον πίνακα, αλλά ο υπολογισμός απαιτεί περισσότερο χρόνο. Ακολουθεί ένα παράδειγμα με αλλαγή του πίνακα (grid) σε 80x80.

```
>implicitplot(eq, x=-4..4, y=-4..4, grid=[80,80]);
```



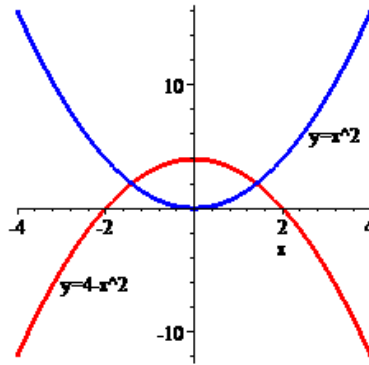
Σχήμα 31: Σχεδίαση της $x^3+y^3=6xy$ με την **implicitplot** και **grid**

Παρατηρούμε ότι τώρα οι καμπύλες γραμμές του γραφήματος είναι πιο "ομαλές".

7.5.3.Σχεδίαση δύο συναρτήσεων σε γραφική παράσταση

Για να παρουσιάσουμε μαζί μια ή περισσότερες γραφικές παραστάσεις χρησιμοποιούμε την εντολή **display**.

```
>with(plots):
>p1:=plot(x^2,x=-4..4,color=blue,thickness=3,font=[TIMES,ROMAN,18]):
>p2:=plot(4-x^2,x=-4..4,color=red,thickness=3,font=[TIMES,ROMAN,8]):
>t1:=textplot([2.6,6,`y=x^2`],align=RIGHT):
>t2:=textplot([-3,-6,`y=4-x^2`],align=RIGHT):
> display({p1,p2,t1,t2},font=[TIMES,ROMAN,8],
tickmarks=[3,3]);
```



Σχήμα 32: Σχεδίαση των x^2 και $4-x^2$ σε μια γραφική παράσταση

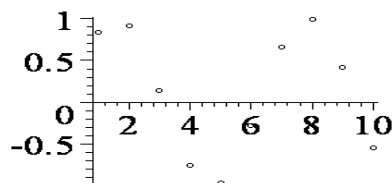
7.5.4.Γραφική παράσταση σημείων και γραμμών που ενώνουν τα σημεία δύο ξεχωριστών γραφικών παραστάσεων.

Για να σχεδιάσουμε αυτή τη γραφική παράσταση θα πρέπει να χρησιμοποιήσουμε την εντολή **pointplot**.

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

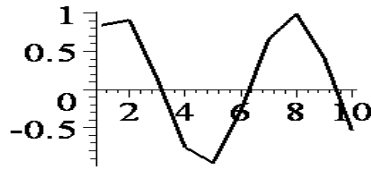
```
>points:=[[n, sin(n)]$n=1..10]:
```

```
>pointplot (points,style=point,symbol=circle, font=
[TIMES,ROMAN,18]);
```



Σχήμα33:Σχεδίασηγραφήματοςμεpointplot

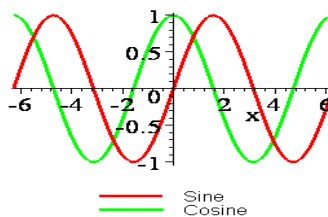
```
>pointplot(points, style=line,thickness=3, font=
[TIMES, ROMAN,18]);
```



Σχήμα 33: Σχεδίαση γραφήματος ενώνοντας τα pointplots

7.5.5.Χάραξη δυο γραφικών παραστάσεων στους ίδιους άξονες, χρησιμοποιώντας και λεζάντα

```
>plot([sin(x),cos(x)],x=-2*Pi..2*Pi,thickness=3,
font=[TIMES,ROMAN,14],legend=["Sine", "Cosine"]);
```

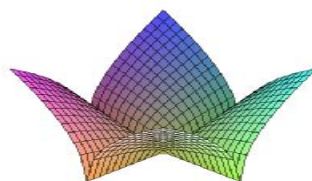


Σχήμα 34: Σχεδίαση των $\sin(x)$ και $\cos(x)$ χρησιμοποιώντας λεζάντα

7.6. Σχεδίαση Τρισδιάστατων Γραφημάτων

Για να κατασκευάσουμε τη γραφική παράσταση μιας συνάρτησης δύο μεταβλητών, χρησιμοποιούμε την εντολή **plot3d**, που δουλεύει με παρόμοιο τρόπο όπως και η **plot**, αλλά τώρα πρέπει να δώσουμε δύο πεδία ορισμού (ένα για την κάθε μεταβλητή):

```
>plot3d(sqrt(abs(x*y)),x=-2..2,y=-2..2)
```

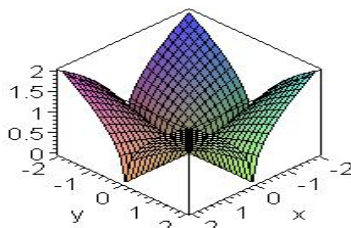


Σχήμα 35: Σχεδίαση τρισδιάστατου γραφήματος

Αν θέλουμε να βάλουμε άξονες σε μορφή κουτιού, στο γράφημα, αυτό μπορεί να γίνει με δυο τρόπους:

Τρόπος Α'

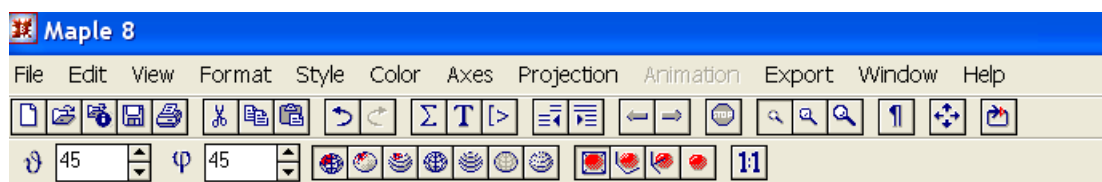
```
>plot3d(sqrt(abs(x*y)),x=-2..2,y=-2..2,axes=boxed);
```



Σχήμα 36: Σχεδίαση τρισδιάστατου γραφήματος με άξονες σε μορφή κουτιού

Τρόπος Β'

Κάνοντας κλικ στο γράφημα ένα μαύρο πλαίσιο εμφανίζεται γύρω από το αυτό και η γραμμή των μενού και η σχετική γραμμή εργαλείων (contextbar) αλλάζει και γίνεται όπως ακριβώς φαίνεται στο Σφάλμα! Άγνωστη παράμετρος αλλαγής..



Σχήμα 37: Σχεδίαση τρισδιάστατου γραφήματος από το contextbar

Πατώντας το **Axes** από το μενού, επιλέγουμε το **Boxed**.

Μπορούμε επίσης να αλλάξουμε το χρώμα του γραφήματος πατώντας από το μενού το **Color** ή ακόμη να αλλάξουμε και το ύφος των καμπύλων ή της επιφάνειας πατώντας από το μενού το **Style**.

Άλλη μια δυνατότητα που μας δίνει το Maple, είναι η περιστροφή του γραφήματος κρατώντας πατημένο το αριστερό κλικ στο ποντίκι.

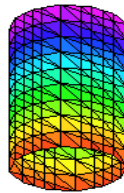
7.6.1.Σχεδίαση γραφικής παράστασης μιας επιφάνειας

Για να γίνει η γραφική παράσταση της επιφάνειας $x^2+y^2=2$, χρειάζεται να χρησιμοποιήσουμε την εντολή `implicitplot3d`.

```
>c1:=x^2+y^2=2;
```

$$c1 := x^2 + y^2 = 2$$

```
>implicitplot3d(c1,x=-3..3,y=-3..3,z=10..10,color=green,axes=framed);
```



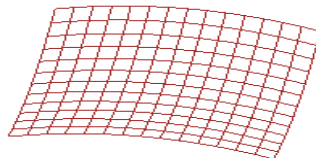
Σχήμα 38: Σχεδίαση γραφικής παράστασης μιας επιφάνειας

7.6.2.Σχεδίαση γραφικής παράστασης μιας συνάρτησης

```
>d:=x->x^2-y^2;
```

$$d := x \rightarrow x^2 - y^2$$

```
>plot3d(d(x), x=-4..4, y=-4..4, style=wireframe, grid=[13,15],  
orientation=[10,10], color=brown );
```



Σχήμα 39: Σχεδίαση της x^2-y^2 σε μορφή πλέγματος

7.7.Κινούμενα Γραφήματα(animation)

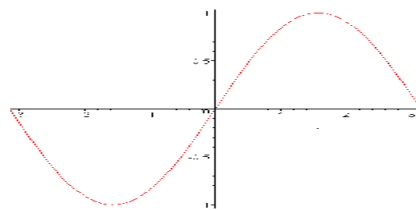
Μία από τις πιο εντυπωσιακές δυνατότητες του Maple είναι η δυνατότητα να φτιάχνουμε κινούμενες εικόνες(animation). Η εντολή που χρησιμοποιούμε είναι:

animate(έκφραση που περιέχει x και p, x=a..b, p=c..d)

Αρχικά παράγει ένα γράφημα (plot) της έκφρασης στο x-διάστημα [a,b], με τον άξονα y ίσο με το c. Εάν κάνουμε κλικ στο γράφημα, μπορούμε να παίξουμε το animation. Το Maple αναπαράγει το γράφημα της έκφρασης στο x-διάστημα [a,b], αυξάνοντας την τιμή του p από c σε d με ίσα βήματα. Ο προεπιλεγμένος αριθμός των βημάτων είναι 16. Αυτό μπορεί να αυξηθεί με την επιλογή **frames=n**.

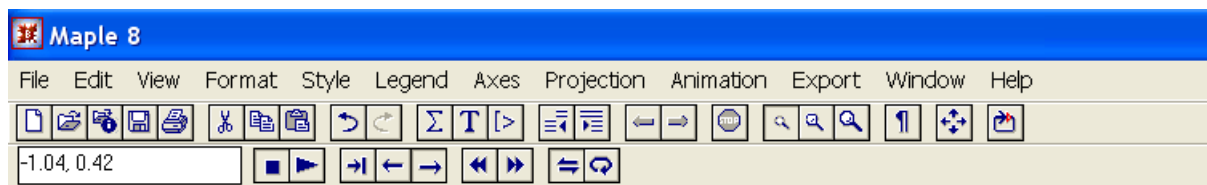
>**with(plots):**

>**animate(sin(b*x), x=-Pi..Pi, b=1..10);**



Σχήμα 40: Σχεδίαση κινούμενου γραφήματος

Κάνοντας κλικ στο γράφημα, το μενού, η εργαλειοθήκη και το contextbar αλλάζει. Υπάρχει τώρα ένα animation μενού, και το contextbar έχει μια γραμμή από νέα κουμπιά, το οποίο παρουσιάζεται στο Σχήμα 41.



Σχήμα 41: Animation μενού

Για παράδειγμα το τετράγωνο είναι το stop και το τρίγωνο το play. Πατώντας το play η γραφική παράσταση αρχίζει να κινείται. Χρησιμοποιώντας τα άλλα κουμπιά ή επιλογές

στο animation μενού, μπορούμε να παίξουμε το animation πιο αργά ή πιο γρήγορα, προς τα πίσω ή προς τα μπρος, μόνο μια φορά ή συνεχώς.

7.7.1.Κινούμενη απεικόνιση μιας παραμετρικής καμπύλης

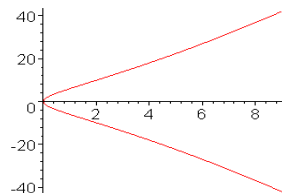
```
>f := t ->t^2;
```

$$f:=t \rightarrow t^2$$

```
>g := t ->t^3-a*t;
```

$$g:=t \rightarrow t^3 - a t$$

```
>animate([f(t), g(t), t=-3..3], a=-5..5);
```



Σχήμα 42: Κινούμενη απεικόνιση μιας παραμετρικής καμπύλης

Το παρακάτω animation δείχνει το γράφημα της $y=x^2$ μαζί με μερικές από τις εφαπτόμενές του. Η εφαπτομένη ευθεία της $y=x^2$ στο (b, b^2) έχει εξίσωση $y=b^2+2b(x-b)$. Θα δούμε το γράφημα στο $x=-2..2$, και αφήνουμε το b να μεταβάλλεται από -2 έως 2 με τα συνηθισμένα 16 frames.

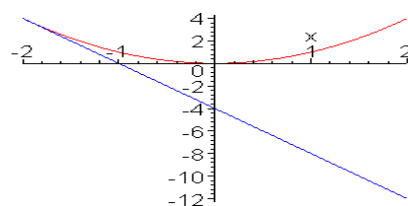
```
>P1:=plot(x^2, x=-2..2, color=red):
```

```
>tanline := b^2 + 2*b*(x-b);
```

$$tanline := b^2 + 2 b (x - b)$$

```
>P2:=animate(tanline, x=-2..2, b=-2..2, color=blue):
```

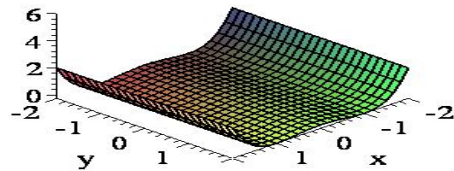
```
>display({P1,P2});
```



Σχήμα 43: Σχεδίαση της εφαπτομένης $y=x^2$

7.7.2.Κινούμενη τρισδιάστατη αναπαράσταση

```
>animate3d(t*y^2/2-x^2/2+x^4/4,x=-2..2,y=-2..2,t=  
0..2, axes=framed,thickness=2,font=[TIMES,ROMAN,14]);
```



Σχήμα 44: Κινούμενη τρισδιάστατη αναπαράσταση

Κεφάλαιο 8

Προγραμματισμός – Δομές Ελέγχου Ροής

8.1. Εισαγωγή

Σε αυτό το κεφάλαιο θα ασχοληθούμε με τον προγραμματισμό σε Maple, ο οποίος είναι αρκετά απλός. Οι ισχυρές εντολές της γλώσσας, διευκολύνουν στη διατύπωση του προβλήματος με λίγες μόνο γραμμές κώδικα (Waterloo Maple Inc, 2002).

Προκειμένου να γραφτούν τα ενδιαφέροντα παραδείγματα των διαδικασιών, πρέπει να καθορίσουμε δύο νέα είδη εντολών Maple, της επαναληπτικής δήλωσης και της υπό όρους δήλωσης. Αυτές καλούνται δηλώσεις ελέγχου, δεδομένου ότι ελέγχουν τη εντολή στην οποία οι εντολές Maple εκτελούνται. Μέχρι τώρα, όποτε έχουμε συσσωρεύσει τις εντολές Maple σε μια ομάδα, είτε σε μια ομάδα εκτέλεσης είτε σε ένα σώμα διαδικασίας, οι εντολές στην ομάδα έχουν εκτελεσθεί σε μια διαδοχική, είτε γραμμική εντολή, μια μετά από άλλη.

Επίσης θα αναφερθούμε στην εντολή `proc` οποία πραγματοποιεί τις διαδικασίες στο Maple. Παρακάτω θα περιγράψουμε τη σύνταξη και την δομή της εντολής `proc`. Μια διαδικασία στο Maple έχει την ακόλουθη γενική μορφή:

```
proc (P)  
local L;  
global G;  
options O  
description D;  
B  
endproc
```

Το B, είναι μια ακολουθία από συνθήκες που αποτελούν το σώμα της διαδικασίας. Οι επίσημοι παράμετροι είναι τα P,local,global, options και description. Ας δούμε ένα παράδειγμα για να κατανοήσουμε την διαδικασία (procedure) στο Maple:

Έστω ότι έχουμε την έκφραση $3x^3 - 4y$, οι x και y είναι οι παράμετροι και αποτελούν την μοναδική συνθήκη της διαδικασίας. Δίνουμε το όνομα Gστη διαδικασία:

```
>G:=proc(x,y) 3*x^3-4*yendproc;
```

```
G := proc(x, y) 3*x^3 - 4*y end proc
```

Δίνουμε τις τιμές 4 και 5 αντίστοιχα για το xκαι yκαι υπολογίζεται το αποτέλεσμα.

```
>G(4,5);
```

172

8.2. Χειρισμός Αλφαριθμητικών

Ένα stringείναι μια ακολουθία χαρακτήρων. Για να δημιουργήσουμε ένα string, εσωκλείουμε μια οποιαδήποτε ακολουθία χαρακτήρων σε διπλά εισαγωγικά. Γιαπαράδειγμα:

```
>" This is a string";  
" This is a string"
```

Δεν μπορούμε να αναθέσουμε μια τιμή σε ένα string.

```
>"Hello":=5;  
Error, invalid left hand side of assignment
```

Δεν υπάρχει ουσιαστικά κάποιος περιορισμός όσον αφορά το μήκος του string (αλφαριθμητικού) στοπρόγραμμα. Στις περισσότερες υλοποιήσεις τουπρογράμματος, αυτό σημαίνει ότι ένα stringμπορεί να περιέχει πάνω από μισό εκατομμύριο χαρακτήρες.

Για να καθοριστεί το μήκος του string, μπορεί να χρησιμοποιηθεί ηεντολή **length** ("Whatisthelengthofthisstring?");

Όλοι οι χαρακτήρες, πέρα από τα εισαγωγικά, προσμετρούνται. Κάθε κενό προσμετρείται ως ένας χαρακτήρας. Ένα κενό string (empty ή nullstring) αναπαρίσταται με διπλά εισαγωγικά χωρίς να εσωκλείονται χαρακτήρες, ούτε καν κενό διάστημα. Για παράδειγμα:

```
> ; "" ;  
""
```

Υπάρχουν πολλές εντολές που μπορούμε να χρησιμοποιήσουμε για το χειρισμό των strings. Προκειμένου να εξάγουμε μέρος του αρχικού string, μπορούμε είτε να χρησιμοποιήσουμε την εντολή **substring(exprString, range)**, η οποία επιστρέφει μέρος του αρχικού string το οποίο υποδεικνύουμε με την **range** είτε να χρησιμοποιήσουμε **subscripts**. Για παράδειγμα:

```
> S := " abcdef";  
S := " abcdef"
```

8.3. Εντολές επανάληψης

8.3.1. Η δομή for

Χρήση: χρησιμοποιείται όταν γνωρίζουμε τον αριθμό των επαναλήψεων.

Γενική μορφή:

```
for δείκτης from έκφραση (αρχική τιμή) by έκφραση  
    (βήμα επανάληψης) to έκφραση (τελική τιμή) do  
    εντολές  
enddo;
```

Περιγραφή της λειτουργίας: η διαδικασία επαναλαμβάνεται όσες φορές ορίζει η αρχική τιμή, η τελική τιμή και το βήμα της επανάληψης. Το do πρέπει να τελειώνει με το od.

Παράδειγμα 1

```
>forifrom 3 to 7 do 'i'=iod;
                                i=3
                                i=4
                                i=5
                                i=6
                                i=7

>i;
                                8
```

Παράδειγμα 2

```
>for i from 3 to 7 by 2 do 'i'=i od;
                                i=3
                                i=5
                                i=7

>i;
                                9
```

Παράδειγμα 3

Θα βρούμε το πρόγραμμα για το άθροισμα των φυσικών αριθμών από την τιμή 1 ως
imax.=1000

```
>i:='i':total:=0:
>imax:=1000:
> for i from 0 to imax do
> total:=i+total:
> od:
> total;

                                500500
```

Παράδειγμα 4

Με την εντολήconvert θα μετατρέψουμε σε δυαδική μορφή όλους τους ακέραιους
αριθμούς από το 100 μέχρι το 110.

```
>for i from 100 to 110 do
> i, convert(i, binary);
```

>od;

```
100, 1100100
101, 1100101
102, 1100110
103, 1100111
104, 1101000
105, 1101001
106, 1101010
107, 1101011
108, 1101100
109, 1101101
110, 1101110
```

Παράδειγμα 5

Με τις επόμενες εντολές δημιουργούμε πολυώνυμα βαθμού από 1 έως 10) και τις παραγοντοποιήσεις αυτών, και με την εντολή **print** εμφανίζουμε τα αποτελέσματα.

```
>for i from 1 to 10 do
>exp1 := add( x^j, j=0..i );
>exp2 := factor( exp1 );
>print( exp1 = exp2 );
>od:
```

$$1 + x = 1 + x$$

$$1 + x + x^2 = 1 + x + x^2$$

$$1 + x + x^2 + x^3 = (1 + x)(1 + x^2)$$

$$1 + x + x^2 + x^3 + x^4 = 1 + x + x^2 + x^3 + x^4$$

$$1 + x + x^2 + x^3 + x^4 + x^5 = (1 + x)(x^2 - x + 1)(1 + x + x^2)$$

$$1 + x + x^2 + x^3 + x^4 + x^5 + x^6 = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6$$

$$1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 = (1 + x)(1 + x^2)(1 + x^4)$$

$$1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8 = (1 + x + x^2)(x^6 + x^3 + 1)$$

$$1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8 + x^9 =$$

$$(1 + x)(x^4 - x^3 + x^2 - x + 1)(1 + x + x^2 + x^3 + x^4)$$

$$1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{10} =$$

$$1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{10}$$

Παράδειγμα 6

Η εντολή **coeffs** επιστρέφει τους συντελεστές εντός πολυωνύμου, ως μια ακολουθία.

```
>for i from 0 to 8 do
>coeffs( expand( (x+y)^i ) );
>od;
```

```
1
1, 1
1, 2, 1
1, 3, 3, 1
4, 6, 4, 1, 1
1, 1, 10, 10, 5, 5
1, 1, 6, 15, 20, 15, 6
7, 21, 35, 35, 21, 7, 1, 1
1, 1, 8, 28, 56, 70, 56, 28, 8
```

Παράδειγμα 7

Η εντολή **nops**(έκφραση), η οποία επιστρέφει το πλήθος των επιμέρους εκφράσεων και η εντολή **op** η οποία εξάγει επιμέρους εκφράσεις από μια έκφραση.

```
>L := [a, b, c]:
>for n from 1 to nops(L) do L := [n, op(L), n] od;
L := [1, a, b, c, 1]
L := [2, 1, a, b, c, 1, 2]
L := [3, 2, 1, a, b, c, 1, 2, 3]

>L := [a, b, c]:
>for n in L do L := [n, op(L), n] od;
L := [a, a, b, c, a]
L := [b, a, a, b, c, a, b]
L := [c, b, a, a, b, c, a, b, c]
```


Παράδειγμα 8

Έχουμε μια λίστα (L) από αριθμούς και θέλουμε να υπολογίσουμε το μέσο όρο τους με την εντολή `average`.

```
>L := [2,3,4,5,6,7,8];  
                                     L := [2,3,4,5,6,7,8]  
  
>total := 0;  
>for i in L do total := total+i od;  
>avg := %/nops(L);  
                                     avg := 5
```

Παράδειγμα 9

Έστω ότι θέλουμε να υπολογίσουμε το άθροισμα των n πρώτων φυσικών αριθμών.

```
>SUM:=proc(n)  
>local i, total;  
>total:=0;  
>for i from 1 to n do  
>total:=total+i;  
>end do;  
>total;  
>endproc;
```

Για παράδειγμα το άθροισμα από 1 έως 150 προκύπτει ως

```
>SUM(150);  
                                     11325
```

Χρησιμοποιώντας την εντολή `add` μπορούμε να πάρουμε να υπολογίσουμε το άθροισμα, ανεξάρτητα από την διαδικασία `SUM`.

```
>add(n, n=1..150);  
                                     11325
```

8.3.2. Η δομή `while`

Χρήση: χρησιμοποιείται όταν επιδιώκουμε την επανάληψη συγκεκριμένων εντολών όσο ισχύει μια συνθήκη.

Γενική μορφή:

```
while λογική έκφραση do  
    εντολές  
enddo;
```

Περιγραφή της λειτουργίας: Η διαδικασία επαναλαμβάνεται όσο ισχύει η λογική συνθήκη. Και στις δύο ανωτέρω δομές, υπάρχουν δύο εντολές ελέγχου του βρόχου, η εντολή **break** και η εντολή **next**. Ειδικότερα, η εντολή **break** έχει ως αποτέλεσμα να σταματά η εκτέλεση της εντολής στην οποία βρίσκεται και να συνεχίζει με την εκτέλεση της αμέσως επόμενης. Με την εντολή **next** το πρόγραμμα προχωρά αμέσως στην επόμενη επανάληψη.

Παράδειγμα 1

Ο πιο κάτω βρόχος αποφασίζει ποιοι αριθμοί ανάμεσα από το 1 έως το 10, είναι ή δεν είναι πρώτοι.

```
>i := 1; # Give i an initial value.  
  while i <= 10 do  
    i, "Is it a prime?", isprime(i);  
  i := i+1;  
  od;  
  
    i := 1  
1, "Is it a prime?", false  
    i := 2  
2, "Is it a prime?", true  
    i := 3  
3, "Is it a prime?", true  
    i := 4  
4, "Is it a prime?", false  
    i := 5  
5, "Is it a prime?", true  
    i := 6  
6, "Is it a prime?", false
```

```

        i := 7
    7, "Is it a prime?", true
        i := 8
    8, "Is it a prime?", false
        i := 9
    9, "Is it a prime?", false
        i := 10
    10, "Is it a prime?", false
        i := 11
>i;

```

11

Παράδειγμα 2

Ο βρόχος `while` βρίσκει τον πρώτο πρώτοαριθμό που είναι μεγαλύτερος από τον ακέραιο αριθμό. Επειδή δεν ξέρουμε ποιος θα είναι αυτός, δεν μπορούμε να γνωρίζουμε πόσες φορές ο βρόχος `while` θα επαναλάβει τη διαδικασία για να τον βρει. Οπότε θα πρέπει να αυξήσουμε τοποστο σώμα του βρόχου.

```

>n := 1000:
while not isprime(n) do n := n+1 od;
        n := 1001
        n := 1002
        n := 1003
        n := 1004
        n := 1005
        n := 1006
        n := 1007
        n := 1008
        n := 1009

```

Παράδειγμα 3

Ο βρόχος `while` βρίσκει τον πρωταρχικό αριθμό που είναι μικρότερος από τον ακέραιο αριθμό. Επειδή δεν ξέρουμε ποιος θα είναι ο πρωταρχικός αριθμός, δεν μπορούμε να γνωρίζουμε πόσες φορές ο βρόχος `while` θα επαναλάβει τη διαδικασία για να τον

βρει. Οπότε θα πρέπει να μειώσουμε τοποστο σώμα του βρόχου.

```
>n := 1000;
while not isprime(n) do n := n-1 od;
    n := 999
    n := 998
    n := 997
```

Παράδειγμα 4

Ο βρόχος for-in-while βρίσκει τον πρώτο αριθμό σε μια λίστα.

```
>L := [a,b,c,d,e,10,9,8,7];
    L := [a, b, c, d, e, 10, 9, 8, 7]
>for i in L while not type(i, numeric) do od;
>i;
```

10

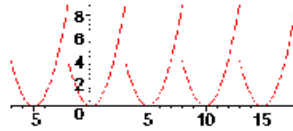
Παράδειγμα 5

Έστω ότι γείναι μια συνάρτηση και ότι ακαι bδσο αριθμοί με $a < b$. Παρακάτω καθορίζουμε την περιοδική επέκταση (η νέα συνάρτηση από την αρχική συνάρτηση) της f από τη g και χαράζουμε και τη γραφική παράσταση.

```
>f := proc(x)
  local y;
  y := x;
  while y >= b do y := y-(b-a) od;
  while y < a do y := y+(b-a) od;
  g(y);
end;
    f := proc(x)
    local y;
    y := x;
    while b ≤ y do y := y - b + a end do;
    while y < a do y := y + b - a end do;
    g(y)
end proc
>g := x -> x^2;
    g := x → x2
>a := -2; b := 3;
    a := -2
```

$b := 3$

```
>plot( f, -7..18, scaling=constrained, discontin=true,  
color=red);
```



Σχήμα 45: περιοδική επέκταση της $g(x)=x^2$

8.4. Εντολές ελέγχου

8.4.1. Η δομή if

Χρήση: χρησιμοποιείται όταν είναι απαραίτητη η λήψη αποφάσεων, ανάλογα με κάποιες συνθήκες και το **if**θα πρέπει να τελειώνει με το **fi**.

Γενική μορφή:

```
if λογική παράσταση then  
    εντολές  
elif λογική παράσταση then  
    εντολές  
else  
endif;
```

Περιγραφή της λειτουργίας: η δομή αυτή ορίζει έναν έλεγχο. Με την ανωτέρω μορφή μπορούμε να ελέγξουμε δύο λογικές παραστάσεις. Η λογική παράσταση μπορεί:

α) να είναι οποιαδήποτε παράσταση του Boolean, η οποία μπορεί να είναι true, false, FAIL, και διαμορφώνεται χρησιμοποιώντας:

- σχεσιακούς τελεστές: <, <=, >, >=, =, and, <>
- λογικούς τελεστές: and, or, not
- λογικές τιμές: true, false, FAIL

β) να είναι μια φωλιασμένη (nested) εντολή, δηλαδή το **then** ή το **else** να περιέχει μια άλλη **if** εντολή. Υπάρχουν όμως και πιο απλές εκδοχές της γενικής μορφής:

```
if λογική παράσταση then
    εντολές
endif;
if λογική παράσταση then
    εντολές
else
    εντολές
end if;
```

Παράδειγμα 1

```
>p:=4:
>if p<2 then lprint(`p is less than 2`):
>elif p>=2 then lprint(`p is not less than 2`):fi:

    `p is not less than 2`
```

Παράδειγμα 2

Παρακάτω είναι μια απλή διαδικασία, η οποία παίρνει δυο αριθμούς ως παραμέτρους και επιστρέφει το μεγαλύτερο από τους δυο.

```
>bigger := proc( a, b )
>if a >= b then a else b fi;
>end;
>
    bigger := proc(a, b) if b ≤ a then a else b end if end proc

>bigger(3, 5);
    5

>bigger(-3, -5);
    -3
```

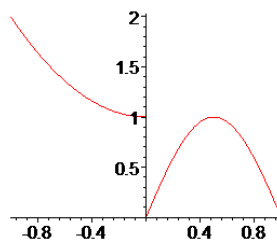
Παράδειγμα 3

Γίνεται υπολογισμός της συνάρτησης $f(x) = \begin{cases} x^2+1 & x < 0 \\ \sin(\pi x) & 0 \leq x \end{cases}$.

```
>f := proc(x)
>if x < 0 then x^2+1 else
>if x >= 0 then sin(Pi*x) fi;
>fi
>end;
```

$f := \text{proc}(x) \text{ if } x < 0 \text{ then } x^2 + 1 \text{ else if } 0 \leq x \text{ then } \sin(\pi x) \text{ end if end if end proc}$

```
>plot( f, -1..1, discontin=true, color=red );
```



Σχήμα 46: Γράφημα της συνάρτησης (f,-1...1)

Παράδειγμα 4

Γίνεται υπολογισμός της συνάρτησης $g(x) = \begin{cases} x^2+x & x \leq 0 \\ \sin(x) & 0 < x < 3\pi \\ x^2-6\pi x+9\pi^2-x+3\pi & 3\pi \leq x \end{cases}$.

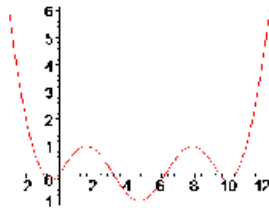
```
>g := proc( x )
> if x <= 0 then
> x^2 + x
>elif x < 3*Pi then
> sin(x)
> else
> x^2-6*x*Pi+9*Pi^2-x+3*Pi
> fi
>end;
```

```

g := proc(x)
  if x ≤ 0 then x^2 + x
  elif x < 3×π then sin(x)
  else x^2 - 6×x×π + 9×π^2 - x + 3×π end proc
end if

```

```
>plot( g, -3..3*Pi+3 );
```



Σχήμα 47:Γράφημα της συνάρτησης (g,-3..3*Pi+3)

Παράδειγμα 5

Η ακόλουθη ομάδα εκτέλεσης δημιουργεί μια λίστα από τυχαίους ακεραίους αριθμούς μεταξύ 1 και 10 και στη συνέχεια θα χρησιμοποιήσει ένα εσωτερικό if για να καθορίσει τι ποσοστό από τυχαίους αριθμούς ήταν δεκαδικοί.

```

>N := 1000:
> counter := 0:
>seq( rand(1..10)(), i=1..N ):
> for i in % do # Check for 10's.
> if i = 10 then counter := counter+1 fi
> od;
> counter/N;

```

$$\frac{12}{125}$$

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

0.09600000000

Κεφάλαιο 9

Τα πακέτα (packages) του Maple

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τα διαθέσιμα πακέτα που περιέχει το Maple με εκτενέστερη αναφορά στο πακέτο της γραμμικής άλγεβρας `LinearAlgebra` και στο πακέτο της βελτιστοποίησης `simplexpackage`. Για περισσότερες πληροφορίες μπορούμε να ανατρέξουμε στη βοήθεια του Maple.

9.1. Εισαγωγή

Η βιβλιοθήκη περιέχει την πλειοψηφία των ρουτινών του προγράμματος. Περιλαμβάνει την λειτουργικότητα που σχετίζεται με τον υπολογισμό, τη γραμμική άλγεβρα, τη στατιστική, τα γραφικά και πολλούς άλλους τομείς. Περιλαμβάνει και εντολές, οι οποίες χωρίζονται σε δύο κατηγορίες:

- τις εντολές ανώτερου επιπέδου (top-level), δηλαδή οι εντολές που χρησιμοποιούνται πιο συχνά στο πρόγραμμα και
- τα packages, τα οποία περιέχουν σχετικές εξειδικευμένες εντολές σε τομείς όπως η γραμμική άλγεβρα, ο υπολογισμός διανύσματος και η δημιουργία κώδικα. Ειδικότερα, τα packages είναι συλλογές λειτουργιών και άλλων δεδομένων, που αντιμετωπίζονται ως σύνολο.

Μπορεί επίσης να περιέχουν και άλλα packages (subpackages). Συνήθως περιλαμβάνουν λειτουργίες, οι οποίες μας δίνουν τη δυνατότητα να κάνουμε υπολογισμούς σε πολύ εξειδικευμένους τομείς προβληματισμού. Παρακάτω αναφερόμαστε στα διαθέσιμα packages του προγράμματος:

algcurves: το χρησιμοποιούμε για να δημιουργήσουμε αλγεβρικές καμπύλες.

codegen: μεταφράζει τον κώδικα του Maple σε άλλες γλώσσες.

CodeGeneration: μεταφράζει τον κώδικα του Maple σε άλλες γλώσσες.

combinat: το χρησιμοποιούμε για τις συνδυαστικές συναρτήσεις.

combstruct: το χρησιμοποιούμε για συνδυαστικές δομές.

context: συμφραζόμενα μενού.

CurverFitting: είναι λειτουργίες που υποστηρίζουν την καμπύλη-εγκατάσταση.

DEtools: είναι εργαλεία διαφορικών εξισώσεων.

diffalg: το χρησιμοποιούμε στην διαφορική άλγεβρα.

diffoms: το χρησιμοποιούμε σε διαφορικές μορφές.

Domains: δημιουργούν πεδία υπολογισμού.

finance: το εφαρμόζουμε στα οικονομικά μαθηματικά.

GaussInt: είναι ακέραιοι Gaussian.

genfunc: το χρησιμοποιούμε σε ρητές συναρτήσεις παραγωγίσις.

geom3d: το χρησιμοποιούμε στην ευκλείδεια τρισδιάστατη γεωμετρία.

geometry: το εφαρμόζουμε στην ευκλείδεια γεωμετρία.

intrans: το εφαρμόζουμε σε ολοκληρωτικούς μετασχηματισμούς.

LargeExpressions: εργαλεία για διαχείριση και δημιουργία υπολογιστικών ακολουθιών.

LibraryTools: εργαλεία για τον χειρισμό καταλόγων.

liesymm: είναι συμμετρίες Lie.

linalg: πακέτο γραμμικής άλγεβρας με βάση τα στοιχεία σε πίνακα δομών.

LinearAlgebra: είναι πακέτο γραμμικής άλγεβρας με βάση rtable δομές δεδομένων.

LinearFunctionalSystems: για να βρούμε τη λύση γραμμικών λειτουργικών συστημάτων της εξίσωσης.

LinearOperators: το χρησιμοποιούμε για την επίλυση γραμμικών λειτουργικών εξισώσεων για να δημιουργήσουμε και να εκτελέσουμε την ελάχιστη annihilators.

ListTools: είναι εργαλεία για το χειρισμό σε καταλόγους.

LRtools: το χρησιμοποιούμε για να χειριστούμε τις γραμμικές σχέσεις επανάληψης.

Maplets: το εφαρμόζουμε για την δημιουργία γραφικών διεπαφών του χρήστη στο Maple.

MathML: το χρησιμοποιούμε για την εισαγωγή και την εξαγωγή στο Maple εκφράσεων MathML.

Matlab: σύνδεση με το λογισμικό του Matlab.

MetrixPolynomialAlgebra: είναι συμβολικός χειρισμός για τις πολυωνυμικές μήτρες.

networks: το εφαρμόζουμε για γράφημα σε δίκτυα.

numapprox: το εφαρμόζουμε για αριθμητική προσέγγιση.

numtheory: το χρησιμοποιούμε στη θεωρία αριθμών.

Orealgebra: το χρησιμοποιούμε για βασικούς υπολογισμούς στις άλγεβρες των

γραμμικών χειριστών.

OrthogonalSeries: εργαλεία για τη σειρά κλασικών ορθογώνιων πολυωνύμων.

orthopoly: είναι ορθογώνια πολυώνυμα.

pedic: το εφαρμόζουμε σε αριθμούς p -adic.

PDEtools: εργαλεία για την επίλυση μερικών διαφορικών εξισώσεων.

plots: είναι γραφικά πακέτα.

plottools: το εφαρμόζουμε για βασικά γραφικά αντικείμενα.

PollynomialTools: είναι ένα εργαλείο πολυωνύμων .

powseries: είναι επίσημες δυναμικές σειρές.

process:{Unix}-πολυ-επεξεργασία.

RandomTools: εργαλεία για την εργασία με τυχαία αντικείμενα.

RealDomain: παρέχει ένα πραγματικό αριθμητικό πλαίσιο.

Rif: πολυωνυμική γραμμική ODEsandPDEs.

ScientificConstants: είναι φυσικές σταθερές και περιοδικές ιδιότητες επιτραπέζιων στοιχείων.

simplex: το εφαρμόζουμε στην γραμμική βελτιστοποίηση.

Slode: το χρησιμοποιούμε για να κατασκευάσουμε λύσεις για γραμμικές ODEs.

SNAP: είναι συμβολικοί-αριθμητικοί αλγόριθμοι για πολυωνυμική αριθμητική.

Sockets: εργαλεία για την επικοινωνία του δικτύου στο Maple.

SoftwareMetrics: είναι λειτουργίες για την μέτρηση της πολυπλοκότητας του κώδικα.

SolveTools: είναι εργαλεία για την επίλυση.

Spread: το εφαρμόζουμε για υπολογιστικά φύλλα.

stats: το εφαρμόζουμε για στατιστικές.

StringTools: το χρησιμοποιούμε για τον χειρισμό σειράς.

Student: το χρησιμοποιούμε για την συλλογή των πακέτων που καλύπτουν προπτυχιακά μαθήματα μαθηματικών.

student: είναι λογισμικό που χρησιμοποιούν οι σπουδαστές.

Student[Calculus1]: μας βοηθά με τη διδασκαλία και την εκμάθηση μίας μεταβλητής λογισμικού.

SumTools: για να βρούμε τις κλειστές μορφές αόριστων και καθορισμένων ποσών.

sumtools: το εφαρμόζουμε για τα αόριστα και καθορισμένα ποσά.

TypeTools: είναι εργαλεία για την επέκταση τύπων.

Units: το χρησιμοποιούμε για τις μετατροπές μονάδων.

VariationalCalculus: είναι εργαλεία για τον υπολογισμό των παραλλαγών.

VectorCalculus: είναι πακέτο διανυσματικού υπολογισμού.

Worksheet: είναι εργαλεία για τη δημιουργία και το χειρισμό σε φύλλα εργασίας του Maple.

XTMLTools: είναι εργαλεία για τη χρήση εγγράφων XML.

9.2. Το πακέτο γραμμικής άλγεβρας **LinearAlgebra**

Το Maple διαθέτει πολλές εντολές για τον υπολογισμό της γραμμικής άλγεβρας. (Waterloo Maple Inc, 2002) Αυτές οι εντολές είναι διαθέσιμες στο πακέτο **LinearAlgebra**. Το πακέτο περιέχει ρουτίνες για την κατασκευή και το χειρισμό πινάκων και διανυσμάτων. Οι εντολές που βλέπουμε όταν φορτώσουμε το πακέτο είναι:

```
>with(LinearAlgebra);  
[Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm,  
BilinearForm, CharacteristicMatrix, CharacteristicPolynomial, Column,  
ColumnDimension, ColumnOperation, ColumnSpace, CompanionMatrix,  
ConditionNumber, ConstantMatrix, ConstantVector, CreatePermutation,  
CrossProduct, DeleteColumn, DeleteRow, Determinant, DiagonalMatrix, Dimension,  
Dimensions, DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors,  
Equal, ForwardSubstitute, FrobeniusForm, GaussianElimination,  
GenerateEquations, GenerateMatrix, GetResultDataType, GetResultShape,  
GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm,  
HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix,  
IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary,  
JordanBlockMatrix, JordanForm, LA_Main, LUdecomposition, LeastSquares,  
LinearSolve, Map, Map2, MatrixAdd, MatrixInverse, MatrixMatrixMultiply,  
MatrixNorm, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial,  
Minor, Modular, Multiply, NoUserValue, Norm, Normalize, NullSpace,  
OuterProductMatrix, Permanent, Pivot, PopovForm, QRdecomposition,  
RandomMatrix, RandomVector, Rank, ReducedRowEchelonForm, Row,  
RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply,  
ScalarVector, SchurForm, SingularValues, SmithForm, SubMatrix, SubVector,  
SumBasis, SylvesterMatrix, ToeplitzMatrix, Trace, Transpose, TridiagonalForm,  
UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply,  
VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip]
```

Για να δημιουργήσουμε πίνακες και διανύσματα χρησιμοποιούμε τις εντολές Matrix και Vector αντίστοιχα. Εάν δεν διευκρινίζεται το διάνυσμα επιστρέφεται σε στήλες.

```
>A1:=Matrix(3, 2, [[1, 0], [0, 1], [1, 1]]);
```

$$A1 := \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

```
>V1:=Vector([1, 0, 0]);
```

$$V1 := \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

```
>V2:=Vector[row]([1, 0, 0]);
```

$$V2 := [1, 0, 0]$$

Μπορούμε να δημιουργήσουμε πίνακες και διανύσματα στο Maple. Αν εισάγουμε τα στοιχεία μας με την μορφή $\langle a, b, c \rangle$ τότε το Maple θα τα εμφανίσει σε γραμμές, ενώ αν τα εισάγουμε με την μορφή $\langle a \mid b \mid c \rangle$ θα τα εμφανίσει σε στήλες. Εάν κάποιες από τις τιμές που διευκρινίζονται ανάμεσα στα ζευγάρια που υπάρχουν σε $\langle \rangle$ δεν είναι κλιμακωτά κατασκευάζεται ένας πίνακας αλλιώς κατασκευάζεται ένα διάνυσμα. Ας δούμε κάποια παραδείγματα:

Κατασκευή μιας στήλης από ένα διάνυσμα

```
>V3:=< 1, 2, 3 >;
```

$$V3 := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Κατασκευή μιας γραμμής από ένα διάνυσμα

```
>V4:=< 1 | 2 | 3 >;
```

$$V4 := [1, 2, 3]$$

Κατασκευή πίνακα από στήλες

```
>A2:=<< 1, 2, 3 > | < 4, 5, 6 >>;
```

$$A2 := \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

Αύξηση του πίνακα με μια στήλη διανύματος

```
>A3:=< % | < x, y, z >>;
```

$$A3 := \begin{bmatrix} 1 & 4 & x \\ 2 & 5 & y \\ 3 & 6 & z \end{bmatrix}$$

Κατασκευή πίνακα από γραμμές

```
>A4:=<< 1 | 2 | 3 >, < 4 | 5 | 6 >>;
```

$$A4 := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

9.3. Βελτιστοποίηση συναρτήσεων

Ο όρος της βελτιστοποίησης αναφέρεται στην αναζήτηση βέλτιστων παραμέτρων ενός συνήθως περίπλοκου συστήματος.

Στην απλούστερη περίπτωση, αυτό σημαίνει την επίλυση των προβλημάτων στις οποίες ο ένας επιδιώκει να ελαχιστοποιήσει ή να μεγιστοποιήσει μια πραγματική συνάρτηση με τη συστηματική επιλογή των τιμών των πραγματικών ή ακέραιων αριθμών μεταβλητών μέσα από μια σειρά που επιτρέπονται. Γενικότερα, σημαίνει την εξεύρεση των "βέλτιστων διαθέσιμων" τιμών κάποιων αντικειμενικών συναρτήσεων, που δίνεται σε μια καθορισμένη περιοχή, συμπεριλαμβανομένου μια ποικιλία από διαφορετικούς τύπους αντικειμενικών συναρτήσεων και διαφορών τύπων.

Το **simplexpackage**, περιέχει εντολές για τη βελτιστοποίηση. Ας δούμε ένα παράδειγμα για την καλύτερη κατανόηση του.

Έστω ότι θέλουμε να ελαχιστοποιήσουμε την έκφραση $t=3x-6y-3z$. Αρχικά θα φορτώσουμε το πακέτο

```
>with(simplex);
```

```
[basis, convexhull, cterm, define_zero, display, dual, feasible, maximize, minimize, pivot,
pivoteqn, pivotvar, ratio, setup, standardize ]
```

```
>t:=3*x-6*y-3*z;
```

```
t := 3 x - 6 y - 3 z
```

Στη συνέχεια θα καθορίσουμε τα c_1, c_2, c_3 .

```
>c1:=3*x+5*y+z<=25;
```

$$c1 := 3x + 5y + z \leq 25$$

```
>c2:=8*x-4*y-2*z <=5;
```

$$c2 := 8x - 4y - 2z \leq 5$$

```
>c3:=6*x-7*y+8*z <=50;
```

$$c3 := 6x - 7y + 8z \leq 50$$

```
>minimize(t, {c1,c2,c3});
```

Όπως βλέπουμε το Maple δεν μας δίνει κάποια λύση, γιατί πρέπει να χρησιμοποιήσουμε την εντολή **feasible**, η οποία αποφασίζει αν τα σύνολα $c1, c2, c3$ είναι εφικτά ή όχι.

```
>feasible({c1,c2,c3});
```

true

Τέλος θα χρησιμοποιήσουμε την παράμετρο **NONNEGATIVE** και θα δούμε το αποτέλεσμα.

```
>minimize(t, {c1,c2,c3}, NONNEGATIVE);
```

$$\left\{ x = 0, y = \frac{150}{47}, z = \frac{425}{47} \right\}$$

Για περισσότερες πληροφορίες μπορούμε να ανατρέξουμε στη βοήθεια του Maple.

Βιβλιογραφία

- 1) Holmes, M. H. An Introductory Guide to Maple.
<http://eaton.math.rpi.edu/CourseMaterials/intro.to.maple>
- 2) Maple. [http://en.wikipedia.org/wiki/Maple_\(software\)](http://en.wikipedia.org/wiki/Maple_(software))
- 3) *Maplesoft*.http://www.maplesoft.com/products/maple/new_features/intex.aspx
- 4) Mathematica.<http://en.wikipedia.org/wiki/Mathematica>
- 5) Matlab.<http://en.wikipedia.org/wiki/MATLAB>
- 6) Waterloo Maple Inc. (2002). *Introduction to Maple 8*. <http://www.groups.mcs.st-and.ac.uk/~peterc/intro2Maple8>.
- 7) Waterloo Maple Inc. (2002). *Maple Getting Started Guide*.
http://www.serc.iisc.ernet.in/ComputingFacilities/software/maple/book/maple_getting_started_guide
- 8) Waterloo Maple Inc. (2002). *ProgrammingGuide*. Canada.
- 9) Δημοστίκαλης, Ι. (2000). Οδηγός του Maple V σε 5 Μαθήματα.<http://www.daedalus.edu.uoc.gr/Edu%20Pages/new/books/maple/maple-guide.doc>.
- 10) Λίλη, Λ.-Ρ. (2005). Εισαγωγή στο λειτουργικό σύστημα εφαρμογών της Maple 9.
http://daedalus.edu.uoc.gr/Edu%20Pages/new/books/maple/intro_maple
- 11) Ξενοφώντος, Χ., & Γεωργίου, Γ. (2007, Μάιος). Μια σύντομη εισαγωγή στο Maple.http://www.ucy.ac.cy/data/KEDIMAX/Xenofontos_Handout
- 12) Παράρτημα Β, *Επίλυση διαφορικών εξισώσεων με το Maple*.
http://www.physics.upatras.gr/UploadedFiles/course_182_7452
- 13) Τσαούση, Χ. Λύση παραδειγμάτων απειροστικού λογισμού με τη χρήση του Maple.
<http://www2.ucy.ac.cy/~christod/maple/mas005maple>