ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΟΙΚΟΝΟΜΙΚΩΝ ΚΑΙ ΚΟΙΝΩΝΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Συγκριτική Μελέτη Λογισμικών Computer Algebra Systems (CAS)

Διπλωματική Εργασία

Ανέστη Σιδηρόπουλου (Α.Μ. 25/07)

Επιβλέπων Καθηγητής:
Νικόλαος Σαμαράς, Επίκουρος Καθηγητής

Θεσσαλονίκη 2009

Περίληψη

Η παρούσα διπλωματική εργασία αφορά στη συγκριτική μελέτη μιας ειδικής κατηγορίας λογισμικών, των Computer Algebra Systems και συγκεκριμένα των ακολούθων: Gauss, Maple, Mathematica, Matlab, Scilab. Μελετήθηκαν οι δομές ελέγχου, οι δομές διακλάδωσης, οι βρόχοι, οι τρόποι διαχείρισης αρχείων, τα είδη των δομών δεδομένων, οι συναρτήσεις και οι βιβλιοθήκες κάθε προγράμματος. Ως βιβλιογραφική αφετηρία, χρησιμοποιήθηκε η τεκμηρίωση (Documentation), που συνοδεύει τα λογισμικά. Απώτερος σκοπός της εργασίας είναι να παρέχει τις απαραίτητες πληροφορίες στον μελλοντικό τελικό χρήστη, ώστε να τον διευκολύνει στην επιλογή του καταλληλότερου προγράμματος, που θα ανταποκρίνεται στις ανάγκες του.

Abstract

The present thesis concerns the comparative study of a specific category of software, Computer Algebra Systems, and in particular the following: Gauss, Maple, Mathematica, Matlab, Scilab. The analysis focused on flow control, loops, file manipulation, data structures, functions and libraries of each program. The documentation of each program was used as a bibliographic start. The purpose of the thesis is to provide the future user with the necessary information in order to make an informed decision on which program will meet his needs.

Λέξεις κλειδιά/ Key words: Gauss, Maple, Mathematica, Matlab, Scilab

Θερμές ευχαριστίες για την πολύτιμη βοήθεια
και καθοδήγησή του στην εκπόνηση της παρούσας εργασίας
στον επιβλέποντα καθηγητή μου κ. Νικόλαο Σαμαρά

ΕΙΣΑΓΩΓΗ

Σύστημα υπολογιστικής άλγεβρας (*Computer Algebra System*, εφεξής *CAS*) είναι ένα πρόγραμμα λογισμικού το οποίο διευκολύνει τα συμβολικά μαθηματικά (*symbolic mathematics*). Αντίστοιχο πρόγραμμα μπορεί επίσης να είναι γνωστό και ως συμβολική υπολογιστική (*Symbolic Computation*) ή υπολογιστική άλγεβρα (*Computational Algebra*)[1].

Τα CAS εν γένει πρωτοεμφανίστηκαν στις αρχές της δεκαετίας του 1970 και προέκυψαν από την έρευνα στον τομέα της τεχνητής νοημοσύνης.

Ο βασικός στόχος ενός CAS είναι να αυτοματοποιήσει κουραστικές και ορισμένες φορές δύσκολες εφαρμογές (*tasks*) αλγεβρικών μετατροπών (*manipulation*). Η βασική διαφορά ανάμεσα σε ένα CAS και σε έναν υπολογιστή τσέπης (*calculator*) είναι η δυνατότητα του πρώτου να επιλύει τις εξισώσεις συμβολικά παρά αριθμητικά. Οι ειδικότερες χρήσεις και οι προοπτικές ενός CAS ποικίλουν σημαντικά από το ένα σύστημα στο άλλο, ωστόσο ο σκοπός, η βασική του δηλαδή λειτουργία, παραμένει ο ίδιος: (*manipulation of symbolic equations*) μετατροπή-απόδοση (*manipulation*) μαθηματικών εκφράσεων σε συμβολική μορφή.[2] Το CAS είναι διαδραστικό και αποβλέπει κυρίως στην ακρίβεια των υπολογισμών του ή όταν αυτό δεν είναι εφικτό, στην όσο το δυνατόν υψηλότερη ακρίβεια. Άλλα πλεονεκτήματα είναι οι δυνατότητες να μετατρέπει μαθηματικές εκφράσεις, ο μεγάλος αριθμός διαθέσιμων αλγορίθμων και οι ικανότητες για plotting. Το CAS συχνά ενσωματώνει εφαρμογές για γραφικές εξισώσεις και παρέχει στον χρήστη την γλώσσα προγραμματισμού, ώστε να καθορίσει τις δικές του παραμέτρους. Βρίσκει εφαρμογή σε πολλούς τομείς της επιστήμης, όπως τα μαθηματικά, την φυσική, την χημεία, την μηχανική, την πληροφορική, την εκπαίδευση κτλ., γεγονός που το καθιστά ολοένα και πιο διαδεδομένο στην διδασκαλία, την έρευνα και την βιομηχανία[3].

[1] http://www.nationmaster.com/encyclopedia/Computer-algebra-system#History

[2] http://www.math.wpi.edu/IQP/BVCalcHist/calc5.html

[3] http://personales.unican.es/iglesias/CASA2006/

Τα CAS εν γένει κατηγοριοποιούνται σε συστήματα γενικού σκοπού και ειδικού σκοπού. Τα πρώτα είναι συστήματα σχεδιασμένα να αντιμετωπίζουν ένα ευρύ φάσμα προβλημάτων ενώ τα δεύτερα δημιουργούνται για την επίλυση προβλημάτων σε επιμέρους τομείς των μαθηματικών, της φυσικής και της μηχανικής.

Ειδικότερα, οι κατηγορίες των CAS είναι οι ακόλουθες[4]:

Α. Συστήματα γενικού σκοπού:

- Εμπορικά (*Derive, DoCon, Maple, Mathematica, MuMath, MuPAD, MathCAD, Reduce, Macsyma, Matlab*)
- Ανοιχτού Κώδικα (*Axiom, Scilab, Eigenmath, GiNaC, Maxima, Yacas, Dcas*)

Β. Συστήματα ειδικού σκοπού:

- αλγεβρικής γεωμετρίας (*CoCoA, Macaulay, SINGULAR*)
- θεωρίας γραφημάτων (*VEGA*)
- θεωρίας συνόλων (*GAP, Magma*)
- θεωρίας των αριθμών (*PARI-GP*)
- αλγεβρικών συστημάτων για υπολογιστές (*calculator*) (TI-89 (βασισμένο στην τεχνολογία του *Derive*), TI-92 (βασισμένο στην τεχνολογία του *Derive*), HP 49G+)
- εκπαιδευτικά (*Algebrator*)

Στην παρούσα εργασία θα ασχοληθούμε με τα συστήματα γενικού σκοπού, τόσο εμπορικά όσο και ανοιχτού κώδικα, και ειδικότερα με τα *Gauss, Maple, Mathematica, Matlab* και *Scilab.*

[4] http://www.onpedia.com/encyclopedia/Computer-algebra-system

1. Gauss

Έκανε την εμφάνισή του για πρώτη φορά το 1984 από την Aptech Systems. Το πρόγραμμα είναι συμβατό με όλα τα ευρέως διαδεδομένα λειτουργικά συστήματα (π.χ. Windows, Mac OS, Linux, Sun SPARC). Η τελευταία του έκδοση 9.0 ανακοινώθηκε το 2008.

Είναι ένα μαθηματικό και στατιστικό περιβάλλον ανάλυσης δεδομένων, βασισμένο στην γλώσσα προγραμματισμού πινάκων Gauss. Χρησιμοποιείται ευρέως από επιστήμονες, μηχανικούς, στατιστικολόγους, βιομέτρες, οικονομέτρες και οικονομικούς αναλυτές για την επίλυση πραγματικών προβλημάτων και προβλημάτων ανάλυσης δεδομένων εξαιρετικά μεγάλης κλίμακας στην στατιστική, οικονομετρία, επεξεργασία σήματος, βελτιστοποίηση και 2D και 3D οπτικοποίηση[5]. Έχει σχεδιαστεί για εφαρμογές με σύνθετους υπολογισμούς και απευθύνεται σε ερευνητές που δεν έχουν τον χρόνο που απαιτείται για να αναπτύξουν προγράμματα σε γλώσσα C ή FORTRAN[6].

2. Maple

Πρωτοεμφανίστηκε το 1980 από το Symbolic Computation Group του Πανεπιστημίου Waterloo (Καναδάς). Λόγω της ευρείας αποδοχής του, από το 1988 και έκτοτε, κυκλοφορεί στην αγορά από την Waterloo Maple Inc. (γνωστή και ως Maplesoft)[7]. Το όνομά του προέκυψε από το φύλλο του σφένδαμου, που απεικονίζεται στην καναδική σημαία. Το πρόγραμμα είναι συμβατό με όλα τα ευρέως διαδεδομένα λειτουργικά συστήματα (π.χ. Windows, Mac OS, Linux). Η τελευταία του έκδοση 12 ανακοινώθηκε τον Μάιο 2008.

Το Maple είναι το βασικό πρακτικό υπολογιστικό λογισμικό για τον σύγχρονο μηχανικό, μαθηματικό και επιστήμονα. Παρέχει την δυνατότητα

[5] http://en.wikipedia.org/wiki/GAUSS

[6] http://www.aptech.com/gauss.html

[7] http://en.wikipedia.org/wiki/Maple_(software)

για γρήγορους υπολογισμούς, ανάπτυξη σχεδίων, διδασκαλία βασικών εννοιών ή παραγωγή σύνθετων μοντέλων εξομοίωσης υψηλής ευκρίνειας.[8] Ο χρήστης μπορεί να εισάγει τα δεδομένα με αριθμητικά σύμβολα καθώς και να προσαρμόσει τις εφαρμογές στις απαιτήσεις του. Παρέχεται η δυνατότητα για αριθμητικούς υπολογισμούς, με αυθαίρετη ακρίβεια, καθώς και για συμβολικούς υπολογισμούς και οπτικοποίηση. Το πρόγραμμα ενσωματώνει μια γλώσσα προγραμματισμού που έχει δυναμικό χαρακτήρα υπό την μορφή εντολών. Οι εντολές μπορούν να τροποποιηθούν μέσω λεκτικών αλλαγών. Παρουσιάζει επίσης συμβατότητα και με άλλες γλώσσες (C, Fortran, Java, Matlab, Visual Basic). Υπάρχει επίσης συμβατότητα και με το Excel.

Το πρόγραμμα αναπτύσσεται γύρω από έναν πυρήνα, γραμμένο σε C, ο οποίος και αποτελεί την γλώσσα προγραμματισμού Maple. Η λειτουργικότητα του προγράμματος βασίζεται στις βιβλιοθήκες, οι οποίες προέρχονται από ποικίλες πηγές, όπως για παράδειγμα τις βιβλιοθήκες GMP και Atlas. Οι περισσότερες από αυτές είναι γραμμένες σε γλώσσα Maple και κατ' αποτέλεσμα έχουν προσβάσιμο πηγαίο κώδικα. Η διαφορετική λειτουργικότητα απαιτεί αριθμητικά δεδομένα σε διάφορες μορφές. Οι συμβολικές εκφράσεις αποθηκεύονται στην μνήμη ως directed acyclic γραφήματα. Η βασική λειτουργία και η λειτουργία της ανάλυσης είναι γραμμένες σε Java ενώ η κλασική λειτουργία είναι γραμμένη σε C.

Αποτελεί τον κύριο ανταγωνιστή του Mathematica.

3. Mathematica

Πρωτοεμφανίστηκε το 1988 από την Wolfram Research. Το πρόγραμμα είναι συμβατό με όλα τα ευρέως διαδεδομένα λειτουργικά συστήματα (π.χ. Windows, Mac OS, Linux). Η τελευταία του έκδοση 7.0.1. ανακοινώθηκε την 03.03.2009.

Είναι ένα υπολογιστικό λογισμικό που χρησιμοποιείται ευρέως στον τομέα της επιστήμης, της μηχανικής και των μαθηματικών. Χρησιμοποιείται

[8] http://www.maplesoft.com/products/Maple/features/index.aspx

στην διεκπεραίωση αναλυτικών και αριθμητικών υπολογισμών καθώς και στην δημιουργία γραφικών με χρήση ηλεκτρονικού υπολογιστή. 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 το χρήστη που μπορεί εύκολα να επεξεργάζεται αναλυτικά, αλγεβρικές παραστάσεις, ολοκληρώματα, διαφορικές εξισώσεις, πίνακες, γραφήματα κλπ[9].

4. Matlab

Πρωτοεμφανίστηκε την δεκαετία του 1970. Ξεκίνησε ως ένα πρόγραμμα "Εργαστηρίου Μητρών/Πινάκων" ("MATrixLABoratory"), ένα διαδραστικό πρόγραμμα, αμφίδρομης επικοινωνίας, που είχε σκοπό να παρέχει ευκολότερη πρόσβαση στις βιβλιοθήκες LINPACK και EISPACK - χωρίς δηλαδή ο χρήστης να χρειάζεται να προγραμματίσει στην γλώσσα FORTRAN. Σε εκείνο το στάδιο προσφερόταν ακόμη 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 ξαναγράφτηκε στην γλώσσα προγραμματισμού C, προστέθηκαν τα M-files, νέα χαρακτηριστικά, νέες βιβλιοθήκες – οι επονομαζόμενες JACKPACK – καθώς και η εταιρία Mathworks Inc, η οποία και το εκμεταλλεύεται εμπορικά.

Το πρόγραμμα είναι συμβατό με όλα τα ευρέως διαδεδομένα λειτουργικά συστήματα (π.χ. Windows, Mac OS, Linux). Η τελευταία του έκδοση R2009a ανακοινώθηκε στις 06.03.2009.

Η κεντρική αρχική ιδέα ήταν να παρέχει έναν εύκολο τρόπο χρήσης πολύπλοκων προγραμμάτων, που έκρυβαν όμως καλά την πολυπλοκότητά τους. 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

[9] http://en.wikipedia.org/wiki/Mathematica

το σύστημα καλύπτει ένα ευρύ φάσμα εφαρμογών και χρησιμοποιείται ως απλή και εύκολη γλώσσα προγραμματισμού, όπου η κάθε γραμμή κώδικα μοιάζει με την μαθηματική πρόταση που αποσκοπεί να αποδώσει[10]. Πλέον είναι ένα πρόγραμμα υπολογιστών για ανθρώπους που χρησιμοποιούν 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 μοντέρνοι 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. Από τον Σεπτέμβριο 2008, στην εργαλειοθήκη συμβόλων του Matlab προστέθηκε ως add –on το Mupad[11]. Το πρόγραμμα αυτό αποσύρθηκε από την κυκλοφορία, ως αυτοτελές προϊόν, την 28η Σεπτεμβρίου 2008, καθώς η εταιρία που το εμπορευόταν, η SciFace Software GmbH & Co KG, εξαγοράστηκε από την Mathworks Inc. Αρχικά είχε αναπτυχθεί από ομάδα ερευνητών στο πανεπιστήμιο Paderborn, Γερμανία και κατόπιν, από το 1997, το δικαίωμα εκμετάλλευσής του αποκτήθηκε από την SciFace Software GmbH & Co KG, η οποία συνέχισε να το αναπτύσσει σε συνεργασία με την ερευνητική ομάδα Mupad και συνεργάτες από άλλα πανεπιστήμια. Το πρόγραμμα ήταν συμβατό με όλα τα ευρέως διαδεδομένα λειτουργικά συστήματα (π.χ. Windows, Mac OS, Linux, Sun SPARC). Η τελευταία του έκδοση είχε ανακοινωθεί στις 02.01.2008.

[10] http://people.bath.ac.uk/masigg/a_manual.pdf

[11] http://en.wikipedia.org/wiki/MuPAD

5. Scilab

Διατίθεται δωρεάν από το 1994 μαζί με τον πηγαίο του κώδικα μέσω του Διαδικτύου. Αναπτύχθηκε από ερευνητές στο INRIA (Institut National de Recherche en Informatique et en Automatique) και το ENPC (École Nationale des Ponts et Chaussées)[12]. Το πρόγραμμα είναι συμβατό με όλα τα ευρέως διαδεδομένα λειτουργικά συστήματα (π.χ. Windows, UNIX, Linux). Η τελευταία του έκδοση 5.1.0. ανακοινώθηκε στις 12.02.2009.

Χρησιμοποιείται παγκοσμίως στην εκπαίδευση και την βιομηχανία. Είναι ένα λογισμικό για αριθμητικούς υπολογισμούς, το οποίο προσφέρει ένα ανοιχτό υπολογιστικό περιβάλλον για μηχανικές και επιστημονικές εφαρμογές.[13] Επιτρέπει την εύκολη χρήση μαθηματικών συναρτήσεων, στατιστικών μεθόδων καθώς και τη δημιουργία τυχαίων αριθμών για προσομοίωση. Χρησιμοποιείται κυρίως για επεξεργασία σήματος, στατιστική ανάλυση, επεξεργασία εικόνας κτλ. και προσφέρει μεγάλη ευκολία στην κατασκευή γραφημάτων και γραφικών παραστάσεων κάθε είδους.

Είναι μια υψηλού επιπέδου γλώσσα προγραμματισμού με κύριο χαρακτηριστικό ότι επιτρέπει στον χρήστη να κάνει πολλούς υπολογισμούς σε ελάχιστες μόνο γραμμές κώδικα μέσω της μετατροπής πρωτογενών μορφών δεδομένων σε λειτουργικά ανάλογες μήτρες. Δεν χρειάζεται compiler και ως εκ τούτου είναι πολύ εύκολο το debugging.

Παρουσιάζει λειτουργικές και συντακτικές ομοιότητες με το Matlab, καθώς σε μεγάλο βαθμό στηρίζεται στην γλώσσα προγραμματισμού του τελευταίου. Επιπλέον το Scilab έχει ενσωματωμένο μετατροπέα για μετατροπή πηγαίου κώδικα από το Matlab. Ωστόσο τα δύο αυτά προγράμματα δεν είναι πλήρως συμβατά. Ενδεικτικό είναι το γεγονός ότι το Scilab έχει λιγότερα αρχεία βοήθειας (help) από το Matlab.

Επίσης το πρόγραμμα περιέχει το πακέτο Scicos για modeling και προσομοίωση μη πεπλεγμένων (explicit) και πεπλεγμένων (implicit) δυναμικών συστημάτων, που περιέχουν τόσο συνεχή όσο και διακριτά υπο-

[12] http://en.wikipedia.org/wiki/Scilab

[13] http://www.scilab.org/platform/index_platform.php?page=overview

συστήματα, και επιπλέον το screenshot εμφανίζει το Scicos block diagram editor στο επάνω δεξί παράθυρο.

ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

Ι. ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Τα δεδομένα που διαχειρίζεται και επεξεργάζεται κάθε χρήστης μπορεί να τα αποθηκεύσει διαλέγοντας μέσα από ποικιλία δομών που διαθέτουν τα διάφορα προγράμματα.

Προκειμένου να προβεί ένα πρόγραμμα στην επεξεργασία μιας συμβολικής εξίσωσης, πρέπει πρώτα να την αποθηκεύσει σε κάποια θέση μνήμης. Στην καρδιά κάθε προγράμματος βρίσκεται η δομή δεδομένων (ή συνδυασμός δομών δεδομένων) που είναι υπεύθυνη για την περιγραφή της μαθηματικής εξίσωσης. Υπάρχουν εξισώσεις με διάφορες μεταβλητές, άλλες που περιέχουν αναφορές σε άλλες συναρτήσεις και ορισμένες που είναι οι ίδιες λογικές συναρτήσεις. Καμία δομή δεδομένων συνεπώς δεν μπορεί να θεωρηθεί ιδανική επιλογή για την αναπαράσταση μιας εξίσωσης. (Μια αναπαράσταση μπορεί να είναι επαρκής για ορισμένες μαθηματικές λειτουργίες αλλά ανεπαρκής για άλλες. Ομοίως κάποια άλλη μπορεί να είναι ανεπαρκής σε χρονική και χωρική πολυπλοκότητα αλλά εύκολη στον προγραμματισμό.)

1. Gauss

Στο πρόγραμμα υπάρχουν τέσσερις βασικές δομές δεδομένων: οι μήτρες, τα N-διάστατα arrays, τα strings και τα string arrays.

Α. Μήτρες[14]

Οι μήτρες είναι διδιάστατα arrays με αριθμούς διπλής ακρίβειας. Όλες οι μήτρες είναι εξ ορισμού μιγαδικές, αν και εάν αποτελείται μόνο από μηδέν, το φανταστικό μέρος μπορεί να μην καταλαμβάνει καθόλου χώρο. Οι μήτρες αποθηκεύονται σε γραμμή με σειρά προτεραιότητας. Μια 2×3 πραγματική

[14] *Aptech Systems Inc:* User Guide, Version 9.0, July 2008, κεφ. 10.6.2., σελ 10-11 επ.

μήτρα αποθηκεύεται με τον ακόλουθο τρόπο από το μικρότερο στο μεγαλύτερο στοιχείο της:

[1, 1] [1, 2] [1, 3] [2, 1] [2, 2] [2, 3]

Μια 2×3 μιγαδική μήτρα αποθηκεύεται με τον εξής τρόπο από μικρότερο στο μεγαλύτερο στοιχείο της:

(πραγματικό μέρος) [1, 1] [1, 2] [1, 3] [2, 1] [2, 2] [2, 3]

(φανταστικό μέρος) [1, 1] [1, 2] [1, 3] [2, 1] [2, 2] [2, 3]

Μετατροπή από μιγαδική σε πραγματική μήτρα γίνεται αυτομάτως και είναι τις περισσότερες φορές εμφανής στον χρήστη.

Μήτρες με ένα μόνο στοιχείο (1×1 μήτρα) ονομάζονται scalars και μήτρες με μία γραμμή ή στήλη (1×N ή N×1 μήτρα) ονομάζονται διανύσματα.

Για πιο αναλυτική περιγραφή βλ. και υπό IV Μήτρες, 1. Gauss, σελ. 51 επ.

Β. Ν-διάστατα Arrays[15]

Πολλές εντολές του προγράμματος υποστηρίζουν arrays με Ν διαστάσεις. Οι κάτωθι εντολές μπορούν να χρησιμοποιηθούν από τον χρήστη για την δημιουργία και επεξεργασία Ν-διάστατα array:

- *aconcat*	Συνενώνει μήτρες και arrays σε μια διάσταση που έχει καθοριστεί από τον χρήστη
- *aeye*	Δημιουργεί ένα Ν-διάστατο array στο οποίο τα επίπεδα που ορίζονται από τις δύο διαστάσεις του array είναι ίσα με την ταυτότητα
- *areshape*	Αναδιαμορφώνει ένα scalar, μια μήτρα ή ένα array σε ένα array μεγέθους σύμφωνα με τους ορισμούς του χρήστη
- *arrayalloc*	Δημιουργεί ένα Ν-διάστατο array με απροσδιόριστο περιεχόμενο.
- *arrayinit*	Δημιουργεί ένα Ν-διάστατο array με καθορισμένη τιμή
- *mattoarray*	Μετατρέπει μια μήτρα σε ένα type array.

[15] *Aptech Systems Inc:* User Guide, Version 9.0, July 2008, κεφ. 10.6.4., σελ 10-19 επ.

Γ. Strings

Για μια αναλυτική περιγραφή βλ. και υπό IV Μήτρες, 1. Gauss, σελ. 39 επ.

Δ. String arrays[16]

Τα String arrays είναι N×K μήτρες strings. Η λίστα που ακολουθεί είναι ενδεικτική των συναρτήσεων που υπάρχουν για την επεξεργασία string arrays:

$|	Vertical string array concatenation operator.
$_	Horizontal string array concatenation operator.
[ ]	Extract subarrays or individual strings from their corresponding array, or assign their values.
0	Transpose operator.
.0	Bookkeeping transpose operator.
declare	Initialize variables at compile time.
delete	Delete specified global symbols.
fgetsa	Read multiple lines of text from a file.
fgetsat	Reads multiple lines of text from a file, discarding newlines.
format	Define output format for matrices, string arrays, and strings.
fputs	Write strings to a file.
fputst	Write strings to a file, appending newlines.
let	Initialize matrices, strings, and string arrays.
loads	Load a string or string array file (.fst file).
lprint	Print expressions to the printer.
lshow	Print global symbol table to the printer.
print	Print expressions on window and/or auxiliary output.
reshape	Reshape a matrix or string array to new dimensions.
save	Save matrix, string array, string, procedure, function or keyword to disk and gives the disk file either a .fmt, .fst or .fcg extension.
show	Display global symbol table.

[16] *Aptech Systems Inc:* User Guide, Version 9.0, July 2008, κεφ. 10.6.6., σελ 10-24 επ.

sortcc	Quick-sort rows of matrix or string array based on character column.
type	Indicate whether variable passed as argument is matrix, string, or string array.
typecv	Indicate whether variables named in argument are strings, string arrays, matrices, procedures, functions or keywords.
varget	Access the global variable named by a string array.
varput	Assign the global variable named by a string array.
vec	Stack columns of a matrix or string array to form a column vector.
vecr	Stack rows of a matrix or string array to form a column vector.

Τα String arrays δημιουργούνται με την χρήση τελεστών συνένωσης string array. Ακολουθεί μια σύγκριση των τελεστών συνένωσης του οριζόντιου string και οριζόντιου string array.

```
x = "age";
y = "pay";
n = "sex";
s = x $+ y $+ n;
sa = x $_ y $_ n;
s = agepaysex
sa = age pay sex
```

2. Maple[17]

Το πρόγραμμα διαθέτει περισσότερες από είκοσι εννέα διαφορετικές δομές δεδομένων. Περίπου το 25% αυτών των δομών αφορά στις εντολές της γλώσσας προγραμματισμού, όπως για παράδειγμα η δομή *for, read*. Οι υπόλοιπες αφορούν στις διάφορες μορφές εντολών, συμπεριλαμβανομένων και όσων προέκυψαν από την χρήση αριθμητικών και λογικών τελεστών καθώς και δομών για αριθμούς, λίστες, σύνολα, πίνακες, εξισώσεις, σειρές. Στο σύνολό τους αναπαρίστανται εσωτερικά ως δυναμικοί πίνακες

[17] Maple User Manual, κεφ. 7.2., σελ. 289

(συναρτήσεις). Οι κυριότερες δομές του προγράμματος είναι: ακολουθία εκφράσεων (expression sequence), σύνολο (set), λίστα (list), πίνακα (table), array και strings.

Α. Ακολουθία Εκφράσεων

Η ακολουθία εκφράσεων που αποτελεί την πιο βασική δομή στο πρόγραμμα, είναι οποιαδήποτε συλλογή εκφράσεων. Για να ορίσει ο χρήστης μια ακολουθία εκφράσεων απλώς παραθέτει τα στοιχεία της χωριζόμενα με κόμμα. Έτσι, για παράδειγμα, θα ορίσει την ακολουθία εκφράσεων px ως εξής:

> *S:= 2, y, sin(x^2), I:*

S:= 2, y, , sin(x^2), I

Αν ζητήσει τον τύπο της μεταβλητής S, έχει:

>*whattype(S);*

exprseq

Για να αναφερθεί σε έναν από τους όρους της ακολουθίας χρησιμοποιεί τις τετράγωνες αγκύλες. Έτσι, αν θέλει το δεύτερο όρο της παραπάνω ακολουθίας, δίνει την εντολή:

> *S[2];*

y

Χρησιμοποιώντας αρνητικούς ακεραίους, μπορεί ο χρήστης να επιλέξει μια έκφραση από το τέλος της ακολουθίας:

> *S[-2]*

sin(x^2)

Για επιλογή πολλαπλών εκφράσεων χρησιμοποιεί τον τελεστή διαστήματος (range operator) (..) :

> *S[2..-2]*

y,sin(x^2)

Εκτός από το να παραθέτει τα στοιχεία της ακολουθίας, μπορεί να τη δημιουργήσει, χρησιμοποιώντας το γενικό όρο της με την εντολή *seq*.

seq(f, i=m..n)

όπου f= γενικός όρος της ακολουθίας, όπου i= ο δείκτης, όπου m,n= αρχική και τελική αριθμητική τιμή

Μια από τις πιο σημαντικές δυνατότητες του προγράμματος σχετικά με τις ακολουθίες εκφράσεων είναι ότι μπορεί ο χρήστης να εισάγει ένα στοιχείο που ήδη υπάρχει. Έστω ότι ορίζει την ακολουθία εκφράσεων:

> S:= 2, y, sin(x^2), I;

S:= 2, y, sin(x^2), I

>S:= S, sum;

S:= 2, y, sin(x^2), I, sum

>S, S;

2, y, sin(x^2), I, sum, 2, y, sin(x^2), I, sum

Για να εξάγει επιμέρους εκφράσεις από μια έκφραση χρησιμοποιεί την εντολή

op(i,j,expr)

όπου expr= η έκφραση, όπου i,j= ακέραιοι που δείχνουν ένα στοιχείο της ακολουθίας.

Μια άλλη εντολή την οποία μπορεί ο χρήστης μπορεί να χρησιμοποιήσει είναι η εντολή *nops*(έκφραση), η οποία επιστρέφει το πλήθος των επιμέρους εκφράσεων.

Β. Σύνολο

Το σύνολο είναι μια μη διατεταγμένη ακολουθία από μοναδικές εκφράσεις ενσωματωμένες σε άγκιστρα {…}. Η γενική του σύνταξη είναι *{στοιχείο1,στοιχείο2,…}*. Για παράδειγμα:

> {4, 12i, sin(⅔)}:

Μια κενή ακολουθία συμβολίζεται με { } .

Το πρόγραμμα αφαιρεί δεδομένα που έχουν εισαχθεί δύο φορές και τα οργανώνει ξανά με τρόπο ευνοϊκό προς την εξωτερική τους αποθήκευση:

> {c, a, a, a, b, c, a}

{a, b, c}

Εντός του συνόλου, υπάρχει μια πληθώρα εντολών με τις οποίες εκτελούνται οι λειτουργίες.

Β.1. Σχέσεις – Πράξεις Συνόλων:

Για να εξετάσει ο χρήστης αν ένα στοιχείο ανήκει σε ένα σύνολο, χρησιμοποιεί την εντολή *in*. Για να εξετάσει αν ένα σύνολο είναι υποσύνολο ενός άλλου συνόλου χρησιμοποιεί την εντολή *subset*.

Η ένωση των συνόλων γίνεται με την εντολή *union*:

> {2, 6, 5, 1}U{2, 8, 6, 7}

{1, 2, 5, 6, 7, 8}

Η διαφορά επιτυγχάνεται με την εντολή *minus*, και η τομή με την εντολή *intersect*.

Γ. Λίστα

Είναι μια διατεταγμένη ακολουθία εκφράσεων ενσωματωμένη σε αγκύλες [ ]. Η σειρά των στοιχείων της λίστας είναι ίδια με εκείνη της ακολουθίας (καθορισμένη από τον χρήστη):

> L:= [2, 3, 3, 1, 0]

L:= [2, 3, 3, 1, 0]

Αν ο χρήστης ζητήσει τον τύπο της μεταβλητής L, παίρνει:

>whattype(L);

list

Αντίθετα με τα σύνολα, οι διπλές εισαγωγές δεδομένων δεν αφαιρούνται από τη λίστα. Στην περίπτωση που η ακολουθία είναι άδεια, οι αγκύλες [ ] αντιπροσωπεύουν την κενή λίστα. Τα στοιχεία μια λίστας μπορούν να είναι κάθε είδους έκφρασης, ακόμη και άλλες λίστες.

Για να προσπελάσει ο χρήστης ένα από τα στοιχεία μιας λίστας πρέπει να καλέσει το όνομα της λίστας και μέσα σε αγκύλες τον αριθμό που δείχνει τη θέση του στοιχείου:

> L[-2..-1]

[1, 0]

Για να εξετάσει αν ένα στοιχείο ανήκει σε μια λίστα, σε ένα σύνολο ή συνάρτηση χρησιμοποιεί την εντολή *member* (στοιχείο, λίστα).

Για να ταξινομήσει τα στοιχεία μιας λίστας χρησιμοποιεί την εντολή *sort* (λίστα).

Για να επιλέξει ένα ή περισσότερα στοιχεία από μια λίστα χρησιμοποιεί την εντολή *select* (opt, λίστα).

Για να μετακινήσει ένα ή περισσότερα στοιχεία από μια λίστα χρησιμοποιεί την εντολή *remove* (opt, λίστα).

Σε μια λίστα αριθμών με την χρήση των εντολών *max* και *min* μπορεί ο χρήστης να βρει τη μέγιστη και ελάχιστη τιμή αντίστοιχα. Για παράδειγμα:

max(x1,x2,…)

min(x1,x2,…)

Για να δημιουργήσει τη γραφική παράσταση μιας λίστας ή ενός συνόλου αριθμών μπορεί να χρησιμοποιήσει την εντολή *listplot* (μέρος του πακέτου plots).

Δ. Πίνακες

Η δομή πίνακας (table) είναι γενίκευση της δομής λίστας. Μια λίστα περιέχει εκφράσεις και για να αναφερθεί ο χρήστης σε κάθε μια έκφραση, χρησιμοποιεί έναν ακέραιο, που αντιστοιχεί στη θέση της έκφρασης στη λίστα. Στη δομή πίνακα αναφέρεται σε ένα στοιχείο του χρησιμοποιώντας όχι απαραίτητα έναν ακέραιο άλλα οποιαδήποτε άλλη έκφραση. Επίσης, στη δομή table δεν καθορίζει εξαρχής το μέγεθος της. Αυτό καθορίζεται δυναμικά με την πρόσθεση ή την αφαίρεση στοιχείων. Η δομή πίνακας, λοιπόν μοιάζει περισσότερο με τη δομή εγγραφής (Record) που υπάρχει στις γλώσσες, όπως η C και η Pascal.

Η εντολή *table( )* δημιουργεί πίνακα:

> Greek:= table([a = α, b = β, c = γ]):

> Greek[b]

β

E. Arrays

Η δομή array είναι και αυτή γενίκευση της δομής λίστας. Κάθε στοιχείο της έχει ένα δείκτη που είναι οπωσδήποτε ένας ακέραιος αριθμός. Η δομή array έχει συγκεκριμένο μέγεθος και αυτό δε δίνεται δυναμικά, αλλά ορίζεται εξαρχής. Κάθε στοιχείο έχει ένα δείκτη τον οποίο μπορεί να χρησιμοποιήσει ο χρήστης για να έχει πρόσβαση σε αυτό. Ο δείκτης μπορεί να είναι οποιοσδήποτε ακέραιος. Ένα μονοδιάστατο array δημιουργείται χρησιμοποιώντας την εντολή *array*

array(*m..n*);

Ο array δημιουργεί μια δομή τύπου array βασικά ορίσματα της οποίας είναι:

• Ακολουθίες εκφράσεων διαστημάτων – Προσδιορισμός δεικτών για κάθε διάσταση.

• Φωλιασμένες Λίστες (Nested lists) – Προσδιορισμός περιεχομένου.

Για παράδειγμα:

> a:= Array(1..3, 1..3, [[1, 2, 3],[4, 5, 6], [7, 8, 9]])

a:= [1 2 3
4 5 6
7 8 9]

>a[1,1]

1

>a[2,3]

6

ΣΤ. Strings

Ένα string είναι μια ακολουθία χαρακτήρων που εσωκλείονται σε διπλά εισαγωγικά (" "). Για παράδειγμα:

> S:= “This is a sequence of characters.”:

“This is a sequence of characters”

Μπορεί ο χρήστης να επιλέξει συγκεκριμένους χαρακτήρες ενός string με τη χρήση των αγκυλών:

>S[8..-2]

“sequence of characters”

Το StringTools package είναι ένα εξελιγμένο σύνολο εργαλείων για το χειρισμό των strings. Για παράδειγμα:

with(StringTools):

Random(9)

¨□□d8p!<v,, ¨

Stem(“impressive”)

¨ impress¨

Split(“Create a list of strings from the words in a string”)

[“Create”, “a”, “list”, “of”, “strings”, “from”, “the”, “words”, “in”, “a”, “string”]

3. Mathematica[18]

Τα δεδομένα στο πρόγραμμα αναπαρίστανται με τη χρήση λίστας, η οποία αποτελεί θεμελιώδη δομή δεδομένων του. Χρησιμοποιείται για την αναπαράσταση συλλογών, arrays, συνόλων και ακολουθιών κάθε είδους, τα οποία μπορούν να έχουν οποιαδήποτε δομή και μέγεθος. Για την επεξεργασία και ανάλυση των λιστών το πρόγραμμα διαθέτει μια ποικιλία από ενσωματωμένες συναρτήσεις, από απλές λειτουργίες, όπως η μετακίνηση στοιχείων μέσα σε μια λίστα, αλλά και πιο σύνθετες λειτουργίες, όπως η εφαρμογή μιας συνάρτησης σε μία λίστα, η ταξινόμηση (sort) οποιουδήποτε συνόλου δεδομένων, εξαγωγή στοιχείων από μια λίστα βάση ορισμένων κριτηρίων κτλ.

Οι λίστες δημιουργούνται με τη χρήση της ενσωματωμένης συνάρτησης *List*, η οποία έχει την τυπική μορφή εισαγωγής μιας ακολουθίας από ορίσματα που χωρίζονται με κόμμα και εσωκλείονται σε άγκιστρα.

$$List\{arg_1, arg_2, \ldots, arg_n\}$$

Τα στοιχεία της συνάρτησης λίστας μπορούν να είναι οποιουδήποτε τύπου έκφρασης, συμπεριλαμβανομένων και αριθμών, συμβόλων, συναρτήσεων, χαρακτήρων, strings, ακόμη και άλλες λίστες. Τα στοιχεία μιας λίστας μπορούν να επανατοποθετηθούν, ταξινομηθούν, αφαιρεθούν,

[18] *Wellin, Paul R./Gaylord, Richard J./Kamin, Samuel N.:* An Introduction to Programming with Mathematica. Third Edition, Cambridge 2005, κεφ. 3 σελ.53

μπορεί ακόμη και να προστεθούν καινούρια στην υπάρχουσα λίστα. Επίσης μπορούν να εκτελεστούν λειτουργίες σε συγκεκριμένα στοιχεία ή σε ολόκληρες λίστες.

4. Matlab[19]

Δομικό στοιχείο του προγράμματος είναι η μήτρα (*matrix*). Όλες οι μεταβλητές είναι εξ ορισμού μήτρες, ακόμη και αυτές που ανήκουν στην κλάση char. Αν και το πρόγραμμα επιτρέπει στον χρήστη να δημιουργεί προηγμένες δομές δεδομένων, στις περισσότερες επιστημονικές εφαρμογές είναι αποτελεσματικότερη η χρήση μητρών (βλ. και υπό IV Μήτρες, 4. Matlab, σελ. 53 επ.)

Εκτός από τη βασική δομή δεδομένων, το πρόγραμμα περιέχει και τις ακόλουθες:

Α. Πολυδιάστατα Arrays

Είναι τα arrays με περισσότερα από δύο subscripts. Ένας τρόπος για να δημιουργήσει ο χρήστης ένα πολυδιάστατο array είναι καλώντας τις συναρτήσεις *zeros*, *ones*, *rand* ή *randn* με περισσότερα από δύο ορίσματα. Για παράδειγμα:

R = randn(3,4,5);

δημιουργεί μια 3x4x5 array με συνολικά 3*4*5 = 60 τυχαία στοιχεία κανονικά κατανεμημένα. Ένα τρισδιάστατο array μπορεί να αναπαριστά α) τρισδιάστατα φυσικά δεδομένα, όπως την θερμοκρασία ενός δωματίου ή β) μια ακολουθία μητρών ή γ) δείγματα μιας εξαρτώμενης από τον χρόνο μήτρας.

Β. Cell Arrays

Τα Cell arrays είναι arrays των οποίων τα στοιχεία είναι αντίγραφα από άλλα arrays. Ο χρήστης μπορεί με την εντολή *cell* να δημιουργήσει ένα cell array με άδειες μήτρες. Αν και τις περισσότερες φορές ένα cell array δημιουργείται εσωκλείοντας μια ανάμεικτη συλλογή από στοιχεία σε άγκιστρα. Τα άγκιστρα χρησιμοποιούνται επίσης με subscripts για να

[19]http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/learn_matlab/f4-2137.html

αποκτήσει ο χρήστης πρόσβαση στο περιεχόμενο διαφόρων κελιών. Για παράδειγμα:

C = {A sum(A) prod(prod(A))}

δημιουργεί ένα cell array 1x3.

Ο χρήστης μπορεί να χρησιμοποιήσει τρισδιάστατα arrays για να αποθηκεύσει μια αλληλουχία μητρών του ίδιου μεγέθους. Αντίθετα τα cell arrays χρησιμοποιούνται για να αποθηκεύσουν μια ακολουθία μητρών με διαφορετικό μέγεθος.

Γ. Χαρακτήρες και κείμενο

Ο χρήστης εισάγει κείμενο στο πρόγραμμα χρησιμοποιώντας μονά εισαγωγικά. Για παράδειγμα:

s = 'Hello'

Είναι ένα array χαρακτήρων 1x5. Εσωτερικά, οι χαρακτήρες αποθηκεύονται ως αριθμοί αλλά όχι κινητής υποδιαστολής. Η δήλωση

a = double(s)

μετατρέπει το array χαρακτήρων σε αριθμητική μήτρα που περιέχει για κάθε χαρακτήρα αναπαραστάσεις κωδικού ASCII κινητής υποδιαστολής. Το αποτέλεσμα είναι:

```
a =
   72   101   108   108   111
```

Η δήλωση

s = char(a)

αναστρέφει την μετατροπή.

Συνένωση με αγκύλες ενώνει μεταβλητές κειμένου σε μεγαλύτερα strings. Η δήλωση

h = [s, ' world']

ενώνει τα strings οριζόντια και δίνει

```
h =
  Hello world
```

Η δήλωση

v = [s; 'world']

ενώνει τα strings κάθετα και δίνει

```
v =
  Hello
  world
   'rolling'
   'stone'
   'gathers'
   'momentum.'
```

Δ. Δομές

Είναι πολυδιάστατα arrays με στοιχεία αποτελούμενα από συλλογές άλλων τύπων δεδομένων (συμπεριλαμβανομένης και της ίδιας της δομής) οι οποίοι ονομάζονται πεδία (fields). Για παράδειγμα:

```
S.name = 'Ed Plum';
S.score = 83;
S.grade = 'B+'
```

δημιουργεί μια δομή scalar με τρία πεδία:

```
S =
    name: 'Ed Plum'
   score: 83
   grade: 'B+'
```

Αφού οι δομές είναι arrays, ο χρήστης μπορεί να εισάγει και πρόσθετα στοιχεία. Σε αυτήν την περίπτωση κάθε στοιχείο του array είναι μια δομή με περισσότερα πεδία. Τα πεδία μπορούν να εισάγονται ένα κάθε φορά

```
S(2).name = 'Toni Miller';
S(2).score = 91;
S(2).grade = 'A-';
```

ή ένα ολόκληρο στοιχείο μπορεί να προστεθεί με μία μόνο δήλωση:

```
S(3) = struct('name','Jerry Garcia',...
              'score',70,'grade','C')
```

5. Scilab[20]

Α. Σταθερές μήτρες

[20] Introduction to Scilab, User's Guide διαθέσιμο σε: http://www.scilab.org/doc/intro/index.html

Το πρόγραμμα αντιμετωπίζει ορισμένες μορφές δεδομένων ως μήτρες. Τα Scalars και διανύσματα θεωρούνται όλα μήτρες.

Α.1. Scalars

Τα Scalars είναι είτε πραγματικός είτε μιγαδικός αριθμός. Οι τιμές των scalars μπορούν να ανατεθούν από τον χρήστη σε ονόματα μεταβλητών.

Α.2. Διανύσματα

Ο συνηθέστερος τρόπος για να δημιουργεί ο χρήστης διανύσματα είναι ο ακόλουθος, δηλαδή χρησιμοποιώντας κόμματα (ή κενά) και ελληνικό ερωτηματικό (;):

```
--> v=[2,-3+%i,7]
v =
! 2. - 3. + i 7. !
```

Α.3. Μήτρες

Τα στοιχεία των γραμμών χωρίζονται με κόμματα και κενά διαστήματα και τα στοιχεία των στηλών με ελληνικό ερωτηματικό (;).

Πολλαπλασιασμός μητρών με scalars, διανύσματα και άλλες μήτρες γίνεται με τον συνήθη τρόπο. Αφαίρεση και πρόσθεση μητρών γίνεται στοιχείο με στοιχείο ενώ και ο πολλαπλασιασμός και η διαίρεση στοιχείο με στοιχείο μπορούν να επιτευχθούν με τους τελεστές .* και . /

```
--> A=[2 1 4;5 -8 2]
A =
! 2. 1. 4. !
! 5. - 8. 2. !
```

Β. Μήτρες με χαρακτήρες Strings

Χαρακτήρες strings μπορεί να δημιουργήσει ο χρήστης χρησιμοποιώντας μονά ή διπλά εισαγωγικά. Συνένωση των strings πραγματοποιείται με την λειτουργία +. Οι μήτρες χαρακτήρων strings δημιουργούνται όπως οι απλές μήτρες π.χ. χρησιμοποιώντας αγκύλες. Ένα πολύ βασικό χαρακτηριστικό τους είναι η δυνατότητα να επεξεργάζονται και να δημιουργούν – αυτομάτως - συναρτήσεις. Επίσης μέσω αυτών μπορεί να υλοποιηθεί και συμβολική επεξεργασία μαθηματικών αντικειμένων.

Γ. Πολυώνυμα και πολυωνυμικές μήτρες

Αυτά δημιουργούνται και μπορούν να τύχουν επεξεργασίας πολύ εύκολα στο πρόγραμμα. Η επεξεργασία είναι όμοια με αυτή των σταθερών μητρών. Το πρόθεμα πολύ- μπορεί να χρησιμοποιηθεί για να εξειδικεύσει τους συντελεστές του πολυώνυμου ή τις ρίζες του. Με τα πολυώνυμα ο χρήστης μπορεί να κάνει πρόσθεση, αφαίρεση, πολλαπλασιασμό και διαίρεση κατά τον συνήθη τρόπο αλλά μόνο μεταξύ πολυωνύμων με τους ίδιους τύπους μεταβλητών. Τα πολυώνυμα, όπως οι πραγματικές και μιγαδικές σταθερές, μπορούν να αποτελέσουν στοιχείο μιας μήτρας.

Γ.1. Ρητή πολυωνυμική απλούστευση

Το πρόγραμμα εκτελεί αυτομάτως απλουστεύσεις pole-zero όταν η ενσωματωμένη εντολή *simp* βρίσκει κοινό συντελεστής στον αριθμητή και τον παρονομαστή ενός ρητού πολυωνύμου *num/den*.

Δ. Μήτρες Boolean

Οι σταθερές Boolean είναι `%t` και `%f`. Αυτές μπορούν να χρησιμοποιηθούν σε μήτρες Boolean. Η σύνταξη είναι η ίδια όπως και στις απλές μήτρες, δηλαδή μπορούν να συνενωθούν κτλ. Τα σύμβολα που χρησιμοποιούνται στις μήτρες Boolean ή για να τις δημιουργήσουν είναι τα `==` και `~`. Εάν *B* είναι μια μήτρα με Boolean *or (B)* και *and (B)* εκτέλεσε τα λογικά *or* και *and*.

Ε. Λίστες

Η λίστα είναι μια συλλογή δεδομένων όχι απαραίτητα του ίδιου τύπου. Η λίστα μπορεί να περιέχει όλους τους τύπους δεδομένων του προγράμματος (ακόμα και συναρτήσεις) καθώς και άλλες λίστες. Η λίστα είναι χρήσιμη για να ορίζονται δομημένα δεδομένα.

Υπάρχουν δύο είδη λίστας: η ordinary και η typed λίστα. Μια λίστα μπορεί να οριστεί με την εντολή *list*. Για παράδειγμα:

```
-->L=list(1,'w',ones(2,2)) //L is a list made of 3 entries
L =
```

L(1)

1.

L(2)

w

L(3)

! 1. 1. !

! 1. 1. !

-->L(3) //extracting entry 3 of list L

ans =

! 1. 1. !

! 1. 1. !

-->L(3)(2,2) //entry 2,2 of matrix L(3)

ans =

1.

Οι Typed λίστες έχουν ένα συγκεκριμένο πρώτο δεδομένο που εισάγεται, το οποίο πρέπει να είναι χαρακτήρας string (the type) ή διάνυσμα χαρακτήρα string – συνεπώς το πρώτο συστατικό είναι ο τύπος (type) και τα υπόλοιπα στοιχεία τα ονόματα που δίδονται στα περιεχόμενα στη λίστα δεδομένα. Τα δεδομένα αυτά μπορούν να τύχουν επεξεργασίας, αν ο χρήστης χρησιμοποιήσει χαρακτήρες strings, ως ακολούθως:

-->L=tlist(['Car';'Name';'Dimensions'],'Nevada',[2,3])

L =L(1)

!Car !

! !

!Name !

! !

!Dimensions !

L(2)

Nevada

L(3)

! 2. 3. !

-->L.Name //same as L(2)

ans =

Nevada

-->L.Dimensions(1,2)=2.3

L =

L(1)

!Car !

! !

!Name !

! !

!Dimensions !

L(2)

Nevada

L(3)

! 2. 2.3 !

-->L(3)(1,2)

ans =

2.3

-->L(1)(1)

ans =

Car

Σημαντικό χαρακτηριστικό αυτής της κατηγορίας είναι ότι αυτές οι λίστες μπορούν να ορίσουν τελεστές που λειτουργούν σε αυτές, δηλαδή είναι δυνατό να ορίσουν π.χ. τον πολλαπλασιασμό L1*L2 των δύο typed λιστών L1 και L2.

II. ΜΕΤΑΒΛΗΤΕΣ

1. Gauss

Όλες οι μεταβλητές αντιμετωπίζονται ως μήτρες στο πρόγραμμα. Μια scalar είναι απλά μια μήτρα 1x1. Ένα διάνυσμα είναι μια μήτρα (Nx1) ή (1xN). Ο χρήστης μπορεί να χρησιμοποιήσει τον Editor του προγράμματος για να βλέπει και να παρακολουθεί την τιμή κάθε μεταβλητής. Όταν τον χρησιμοποιεί για να βλέπει, επεξεργάζεται και παρακολουθεί μικρότερες μήτρες, τότε μπορεί να μειώνει το χώρο που καταλαμβάνει στην οθόνη επιλέγοντας το Minimal View από το μενού View. Επίσης μπορεί ανά πάσα

στιγμή να ανανεώσει την τιμή της μεταβλητής χρησιμοποιώντας την εντολή *Reload*.

Κάθε μεταβλητή που ορίζεται στο βασικό πρόγραμμα είναι εξ ορισμού καθολική. Οι μεταβλητές είναι δύο τύπων: strings και μήτρες, οι οποίες περιλαμβάνουν ως υπο-ομάδα τα διανύσματα (στήλες και γραμμές) και τις *scalars* – αν και όλα αντιμετωπίζονται από το πρόγραμμα με τον ίδιο τρόπο. Υπάρχουν επίσης δύο τρόποι για να κατηγοριοποιηθούν οι μεταβλητές: οι δομές και τα string arrays[21]. Για παράδειγμα:

a = b + c;

είναι έγκυρο, εφόσον τα a, b και c είναι scalars, διανύσματα ή μήτρες, υποθέτοντας ότι οι μεταβλητές είναι συμβατές. Ωστόσο τα αποτελέσματα της λειτουργίας μπορεί να είναι ελαφρώς διαφορετικά ανάλογα με τον τύπο της μεταβλητής.

Ειδικότερα:

- Οι μήτρες μπορεί να περιέχουν αριθμητικά δεδομένα ή δεδομένα σε χαρακτήρες ή και τα δύο. Τα αριθμητικά δεδομένα αποθηκεύονται σε επιστημονική γλώσσα σε περίπου 12 σημεία ακρίβειας με ένα εύρος περίπου $10^{\pm 35}$. Τα δεδομένα σε χαρακτήρες είναι ακολουθίες έως οκτώ χαρακτήρων, που θεωρούνται ως ένα στοιχείο της μήτρας. Εάν ο χρήστης εισάγει κείμενο με περισσότερους από οκτώ χαρακτήρες σε κελιά της μήτρας τότε το κείμενο θα εμφανιστεί προβληματικό.
- Τα strings είναι κομμάτια κειμένου με απεριόριστο μέγεθος. Χρησιμοποιούνται για να δίνουν πληροφορίες στον χρήστη. Εάν ο χρήστης προσπαθήσει να ορίσει μια τιμή string για ένα στοιχείο μιας μήτρας, τότε μόνο οι οκτώ πρώτοι χαρακτήρες θα διατηρηθούν.
- Τα string arrays είναι ένας βολικός τρόπος για την οργάνωση των strings. Έχουν ομοιότητες με τις μήτρες χαρακτήρων, μόνο που τα

[21] Ritchie, Felix: Programming in GAUSS, 09.10.2002 διαθέσιμο σε: http://www.trigconsulting.co.uk/gauss/man_basics.html

strings που περιέχουν μπορεί να έχουν απεριόριστο μέγεθος. Για τον λόγο αυτό το ακόλουθο είναι ένα έγκυρο string array:

Aberdeen	Dundee
Edinburgh	Glasgow
Heriot-Watt	St. Andrews
Stirling	Strathclyde

Τα string arrays είναι πιο εύχρηστα για την αποθήκευση χαρακτήρων. Ωστόσο έχουν κάποια μειονεκτήματα:

- αποθηκεύουν μόνο strings, οπότε ο χρήστης δεν μπορεί να αναμείξει χαρακτήρες και αριθμητικά δεδομένα,
- επειδή το μέγεθός τους ποικίλει, το πρόγραμμα θα τα διαχειριστεί λιγότερο αποτελεσματικά. Αν όλα τα strings χαρακτήρων αποτελούνται από οκτώ ή λιγότερους χαρακτήρες, τότε η αποθήκευσή τους σε μήτρες χαρακτήρων θα έχει οριακά ταχύτερα αποτελέσματα και
- καταλαμβάνουν περισσότερο χώρο στην μνήμη.
- Ο χρήστης πρέπει να δημιουργεί και να δίνει μια αρχική τιμή στις μεταβλητές πριν τις καταχωρήσει, δηλαδή πριν δεσμευθεί για αυτές χώρος στην μνήμη. Τα αποδεκτά ονόματα πρέπει να έχουν οκτώ χαρακτήρες, μπορούν να περιέχουν αλφαριθμητικά δεδομένα και κάτω παύλα (underscore "_") και δεν πρέπει να ξεκινούν με αριθμό. Δεσμευμένες λέξεις δεν μπορούν να χρησιμοποιηθούν. Ονόματα βασικών λειτουργιών μπορούν να ξαναχρησιμοποιηθούν, αν και καλό θα ήταν να αποφευχθεί.

Οι λειτουργίες του προγράμματος ωστόσο επιτρέπουν την χρήση τοπικών μεταβλητών, δηλαδή μεταβλητών που είναι ορατές μόνο στο πλαίσιο της εκάστοτε λειτουργίας. Οτιδήποτε έξω από αυτήν την λειτουργία δεν μπορεί να διαβάσει ή να έχει πρόσβαση σε αυτές τις μεταβλητές, δηλαδή είναι σα να μην υπάρχουν για το υπόλοιπο πρόγραμμα.

Καθώς οι λειτουργίες δεν μπορούν να δουν τις μεταβλητές που δημιουργούνται από άλλες λειτουργίες, μεταβλητές με το ίδιο όνομα μπορούν να χρησιμοποιηθούν σε περισσότερες λειτουργίες. Εάν ωστόσο τα ονόματα των μεταβλητών συγκρούονται (π.χ. μια καθολική μεταβλητή έχει το ίδιο όνομα με μια τοπική μεταβλητή) τότε η τοπική μεταβλητή πάντα έχει προτεραιότητα.

Εφόσον λοιπόν οι τοπικές μεταβλητές υπάρχουν εντός μιας λειτουργίας, όταν η λειτουργία ολοκληρωθεί και ο έλεγχος επιστρέψει στον κώδικα που καλείται, όλες αυτές οι τοπικές μεταβλητές θα διαγραφούν από την μνήμη. Εάν η λειτουργία ξαναξεκινήσει, οι τοπικές μεταβλητές θα δημιουργηθούν από την αρχή.

Οι καθολικές μεταβλητές δεν μπορούν να δηλωθούν εντός μιας λειτουργίας. Μπορούν να χρησιμοποιηθούν, να τροποποιηθεί το μέγεθός τους αλλά δεν μπορούν να δηλωθούν από την αρχή. Κάθε μεταβλητή που χρησιμοποιείται σε μια λειτουργία πρέπει είτε να έχει δηλωθεί ρητά ως τοπική μεταβλητή είτε να είναι μια προϋπάρχουσα καθολική μεταβλητή.

2. Maple[22]

Μια μεταβλητή ή σταθερά μπορεί να έχει όνομα οποιαδήποτε αλφαριθμητική σειρά χαρακτήρων για παράδειγμα a, total, exp1, sum1, sum2 κ.α.

Δεν είναι αποδεκτά ονόματα που αρχίζουν με αριθμό πχ 2nd, 3total ή εκείνα που περιέχουν ειδικούς χαρακτήρες όπως τους @,&,* κ.α. τα ονόματα δηλαδή a&b, total*sum , ονόματα που είναι δεσμευμένα από το maple ως εντολές πχ plot,solve και ονόματα που έχουν ιδιαίτερη σημασία στη γλώσσα του maple. Υπάρχουν 46 τέτοια ονόματα που είναι δεσμευμένα. Έναν κατάλογο αυτών των ονομάτων μπορούμε να έχουμε, πληκτρολογώντας την εντολή:

> *?reserved;*

[22] Ματζάκος, Νίκος: Συστήματα Συστήματα Συμβολικής Συμβολικής Άλγεβρας Άλγεβρας (Computer Algebra System Μέρος 1ο, διαθέσιμο σε:
http://dtps.unipi.gr/files/notes/2006- 2007/eksamino_1/grammikh_algebra/cas-maple1.pdf

Επίσης πρέπει να σημειωθεί ότι το πρόγραμμα κάνει διαχωρισμό μεταξύ πεζών και κεφαλαίων γραμμάτων. Επομένως άλλη είναι η μεταβλητή a και άλλη η μεταβλητή A.

Παρέχει τη δυνατότητα να αποδώσουμε σε μια μεταβλητή μια «τιμή». Η τιμή αυτή μπορεί να είναι ένας αριθμός, μια έκφραση, μια συνάρτηση, μια εξίσωση, μια γραφική παράσταση. Αυτό μπορεί να γίνει με δύο τρόπους, είτε χρησιμοποιώντας τον τελεστή απόδοσης := , είτε χρησιμοποιώντας την εντολή *assign*.

Ο χρήστης μπορεί να κάνει πράξεις με τις μεταβλητές.

Αν έχει ορίσει μια μεταβλητή αυτή ισχύει σε όλο το φύλλο εργασίας και όσο δε σβήνει το περιεχόμενο της μνήμης του προγράμματος. Στην περίπτωση που θέλει να αλλάξει την τιμή της, απλώς πρέπει να την ορίσει ξανά δίνοντας τη νέα τιμή που θέλει.

Είναι σκόπιμο για την αποφυγή λαθών να μηδενίζονται όλες οι μεταβλητές όταν ξεκινά νέους υπολογισμούς και αυτό γίνεται με την εντολή *restart*.

Α. Τύποι μεταβλητών

Οι μεταβλητές που μπορούν να χρησιμοποιηθούν στο πρόγραμμα είναι διαφόρων τύπων. Στον παρακάτω πίνακα περιγράφονται κάποιες από αυτές:

τύπος	Περιγραφή
integer	Ακέραιοι των οποίων το εύρος καθορίζεται κάθε φορά από το σύστημα του υπολογιστή.
fraction	Κλάσματα
float	Αριθμοί κινητής υποδιαστολής
complex	Μιγαδικοί Αριθμοί
string	Αλφαριθμητικοί
nonreal	Μη πραγματική
Boolean	Λογική(True ή False)
rational	Ρητός
sfloat	SFloat(M,E)

	Αριθμοί κινητής υποδιαστολής (M η μάντισσα και E ο εκθέτης)
{…}	Σύνολα
[…]	Λίστα
(…)	Ακολουθία
indexed	Δείκτες

Με την εντολή *whattype* μπορεί ο χρήστης να δει το είδος της μεταβλητής που χρησιμοποιεί.

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

Με την εντολή *convert (expr,form)* μπορεί να μετατρέψει μια μεταβλητή από έναν τύπο σε έναν άλλον. Οι μετατροπές που μπορούν να γίνουν είναι πολλές και μπορεί ο χρήστης να τις δει πληκτρολογώντας *?convert*.

Το πρόγραμμα δίνει τη δυνατότητα να θέσει υποθέσεις στις μεταβλητές του με την εντολή *assume(μεταβλητή, υπόθεση).*

3. Mathematica[23]

Α. Ορίζοντας μεταβλητές

Προκειμένου ο χρήστης να διευκολυνθεί σε μακροσκελείς υπολογισμούς, μπορεί να δίνει ονόματα σε ενδιάμεσα αποτελέσματα που προκύπτουν, δηλαδή να ορίζει μεταβλητές. Για παράδειγμα, ο χρήστης ορίζει την τιμή της μεταβλητής x να είναι 5:

```
x =5
5
```

Σημειωτέον ότι οι τιμές που ορίζονται στις μεταβλητές είναι προσωρινές. Επειδή το πρόγραμμα θεωρεί ότι ορίζοντας την τιμή μιας μεταβλητής ο χρήστης επιθυμεί η μεταβλητή αυτή να έχει διαρκώς την ίδια τιμή. εκτός εάν ο χρήστης το δηλώσει ρητά, θα πρέπει ο χρήστης να διαγράφει την τιμή που όρισε για την εκάστοτε μεταβλητή, ώστε να αποφεύγονται τα λάθη.

[23] http://reference.wolfram.com/mathematica/tutorial/DefiningVariables.html

Το όνομα που δίνει ο χρήστης στην μεταβλητή μπορεί να είναι οτιδήποτε και χωρίς κανέναν περιορισμό στο μέγεθος. Ο μόνος περιορισμός είναι ότι το όνομα της μεταβλητής δεν μπορεί να αρχίζει με αριθμό. Για παράδειγμα x2 μπορεί να είναι μια μεταβλητή, όμως 2x σημαίνει 2*x.

Το πρόγραμμα δεν διακρίνει μεταξύ πεζών και κεφαλαίων γραμμάτων. Υπάρχει ωστόσο μια συνθήκη ότι τα ονόματα των ενσωματωμένων στο πρόγραμμα στοιχείων αρχίζουν με κεφαλαίο γράμμα. Συνεπώς συνίσταται ο χρήστης να δίνει ονόματα στις μεταβλητές τους που ξεκινούν με μικρό γράμμα ώστε να αποφεύγονται οι συγχύσεις.

Β. Ορίζοντας μεταβλητές και συναρτήσεις.

Στο πρόγραμμα μια μεταβλητή δεν δέχεται απλώς μια τιμή αλλά μπορεί να χρησιμοποιηθεί καθαρά συμβολικά. Συνεπώς δημιουργώντας ο χρήστης συναρτήσεις στο πρόγραμμα μπορεί να τις κάνει όχι μόνο να δέχονται ορίσματα αλλά και να μετατρέπουν μορφές με κάθε δομή.

Γ. Παραδείγματα αναθέσεων.

Set (=)	άμεση ανάθεση (η δεξιά πλευρά να υπολογιστεί άμεσα)
SetDelayed (:=)	ανάθεση με υστέρηση (η δεξιά πλευρά να υπολογιστεί μόνο όταν χρησιμοποιηθεί)
Unset (=.)	ανάκληση ορισμού μεταβλητής
Clear	διαγραφή ορισμού συνάρτησης

4. Matlab[24],[25]

Όλες οι μεταβλητές είναι εξ ορισμού μήτρες, ακόμη και αυτές που ανήκουν στην κλάση char. Η γενική σύνταξη της εντολής καταχώρησης είναι η ακόλουθη:

όνομα_μεταβλητής = τιμή ή μεταβλητή ή αποτέλεσμα πράξεων.

Το πρόγραμμα έχει ευαισθησία στη διάκριση κεφαλαίων και πεζών γραμμάτων. Το «Α» και το «α» δεν είναι ίδιες μεταβλητές. Παρόλο που το

[24] *The MathWorks:* Matlab 7 Programming Tips, 2008, κεφ. 1 σελ. 1-26
[25] *The MathWorks:* Matlab 7 Getting Started, 2008, σελ. 2-11

όνομα μιας μεταβλητής, στην ουσία μια θέση μνήμης στην οποία καταγράφεται η τιμή, δεν περιορίζεται σε μέγεθος, το πρόγραμμα χρησιμοποιεί μόνο τους N χαρακτήρες από το όνομα (όπου N είναι ο αριθμός που επιστρέφεται με την χρήση της συνάρτησης *namelengthmax*) και αγνοεί τους υπόλοιπους χαρακτήρες.

Το όνομα της μεταβλητής αποτελείται από ένα γράμμα ακολουθούμενο από οσαδήποτε γράμματα, ψηφία ή κάτω παύλες (underscores), και να μην περιλαμβάνουν δεσμευμένους χαρακτήρες. Στην περίπτωση που ο χρήστης σε μια καταχώρηση εντολής δεν δώσει κάποιο όνομα μεταβλητής τότε το πρόγραμμα χρησιμοποιεί μια δικιά του μεταβλητή, την *ans*. Δεν χρειάζεται να δηλώνεται από πριν ο τύπος ή η διάσταση μιας μεταβλητής, δηλαδή δεν χρειάζεται ο χρήστης να δηλώσει εκ των προτέρων αν μια μεταβλητή θα περιέχει ακέραιους, πραγματικούς ή μιγαδικούς αριθμούς. Το πρόγραμμα θεωρεί όλες τις μεταβλητές ότι είναι μιγαδικοί αριθμοί, κάτι που επιτρέπει την ανάμιξη πραγματικών και μιγαδικών αριθμών στις διάφορες παραστάσεις.

Πριν χρησιμοποιήσει ο χρήστης ένα καινούριο όνομα μεταβλητής μπορεί να σιγουρευτεί ότι είναι έγκυρο κάνοντας χρήση της συνάρτησης *isvarname*. Η συνάρτηση αυτή δεν θεωρεί ονόματα με περισσότερους από τους namelengthmax χαρακτήρες ως έγκυρα.

Για να ελέγξει εάν το όνομα μιας μεταβλητής χρησιμοποιείται ήδη από μια συνάρτηση, χρησιμοποιεί την *which -all name*.

Επίσης καλό είναι να αποφεύγει την χρήση των χαρακτήρων i και j για το όνομα μεταβλητών, μιας και το πρόγραμμα τους χρησιμοποιεί για να αναπαραστήσει φανταστικές μονάδες. Όταν το πρόγραμμα συναντά ένα καινούριο όνομα μεταβλητής, αυτόματα δημιουργεί την μεταβλητή και διαμορφώνει τον απαραίτητο χώρο αποθήκευσης. Εάν η μεταβλητή υπάρχει ήδη, τότε το πρόγραμμα αλλάζει το περιεχόμενό της και εάν είναι αναγκαίο, δημιουργεί επιπλέον χώρο αποθήκευσης. Για παράδειγμα:

num_students = 25

δημιουργεί έναν 1-by-1 πίνακα με το όνομα num_students και αποθηκεύει την τιμή 25 στο μοναδικό του στοιχείο.

Για να δει τον πίνακα που έχει ανατεθεί σε μια μεταβλητή απλώς εισάγει το όνομα της μεταβλητής.

Ο χρήστης προκειμένου να μην οδηγηθεί σε λάθη θα πρέπει να αποφεύγει να ονομάζει μεταβλητές κάνοντας χρήση του ονόματος κάποιων σταθερών τιμών (constant values), όπως των μεταβλητών pi (ο αριθμός π), eps (μικρότερος θετικός αριθμός) κ.α.

Η τιμή μιας μεταβλητής μπορεί να αλλάξει οποιαδήποτε στιγμή. Για παράδειγμα έστω

x=5

x=

5

y=6

y=

6

z=x+y

z=

11

Στη συνέχεια ο χρήστης μπορεί να ορίσει νέα τιμή για την μεταβλητή x, έστω

x=7

όμως η μεταβλητή z θα εξακολουθεί να είναι z=11. Η ιδιότητα αυτή διακρίνει το πρόγραμμα από όλες τις άλλες γλώσσες προγραμματισμού.

Για να μπορέσει ο χρήστης να δει τα ονόματα των μεταβλητών του χώρου εργασίας χρησιμοποιεί την εντολή *who*. Ενώ για περισσότερες πληροφορίες την εντολή *whos*.

Για την αποθήκευση των μεταβλητών σε αρχεία, ο χρήστης χρησιμοποιεί την εντολή *save*, ενώ για να φορτώσει τα αρχεία, την εντολή *load*.

5. Scilab

Μια μεταβλητή ανήκει σε προκαθορισμένο τύπο. Μπορεί να αλλάξει την τιμή της στον ίδιο τύπο. Μπορεί να αλλάξει τον τύπο της. Μια κενή μεταβλητή, είναι εκείνη που δεν περιέχει καμία τιμή και συμβολίζεται με [ ], αντιστοιχώντας σε μια άδεια μήτρα με μηδέν γραμμές και μηδέν στήλες. Η προεπιλεγμένη μεταβλητή ans (answer) δημιουργείται αυτόματα κάθε φορά

που το αποτέλεσμα μιας έκφρασης δεν επηρεάζεται από μια μεταβλητή. Οι προεπιλεγμένες μεταβλητές συνήθως έχουν το πρόθεμα %[26]. Είναι προστατευμένες, δηλαδή δεν μπορεί να επαναπροσδιοριστούν.

Οι μεταβλητές Boolean είναι %T, %F και αντιστοιχούν στις δηλώσεις «true» και «false».

III. ΧΕΙΡΙΣΜΟΣ ΑΛΦΑΡΙΘΜΗΤΙΚΩΝ (Strings)

1. Gauss[27]

Τα Strings είναι τμήματα κειμένου απεριόριστου μήκους. Χρησιμοποιούνται για να δίνουν πληροφορίες στον χρήστη, όπως για να αποθηκεύουν τα ονόματα των αρχείων που θα ανοιχτούν, μηνύματα που εκτυπώνονται, ολόκληρα αρχεία ή οτιδήποτε άλλο μπορεί να χρειαστεί ο χρήστης. Αν προσπαθήσει ο χρήστης να αναθέσει μια τιμή string σε ένα στοιχείο μήτρας, όλα εκτός από τους πρώτους οχτώ χαρακτήρες θα χαθούν.

Το μηδενικό string (¨ ¨) είναι ένα έγκυρο τμήμα κειμένου τόσο για τα strings όσο και για τις μήτρες.

Τα Strings μπορούν να δημιουργηθούν ως εξής:

x = "example string";

ή

x = cons; / keyboard input */*

ή

x = getf("myfile",0); / read a file into a string */*

Μπορούν να εκτυπωθούν ως εξής:

print x;

Ένα string μπορεί να αποθηκευτεί στον δίσκο με την εντολή save σε ένα αρχείο με επέκταση .fst και στην συνέχεια να φορτωθεί με την εντολή load:

save x;

---

[26] *Urroz, Gilberto E.:* Comparison of SCILAB Syntax and Functions to MATLAB. 2001, διαθέσιμο σε *infoclearinghouse.com*.

[27] *Aptech Systems Inc:* User Guide, Version 9.0, July 2008, κεφ. 10.6.5., σελ 10-20 επ.

loads x;

ή

loads x=x.fst;

Κάθε τιμή byte από 0 - 255 είναι έγκυρη για ένα string. Το buffer στο οποίο ένα string αποθηκεύεται πάντα περιλαμβάνει ένα τελικό byte του ASCII το 0. Αυτό επιτρέπει στις συναρτήσεις C να περνούν τα strings ως όρισμα μέσω του Foreign Language Interface

Ακολουθεί μια συνοπτική λίστα συναρτήσεων για τον χειρισμό των strings:

- *$+* ενώνει δύο strings σε ένα ενιαίο string
- *ˆ* μεταγλωττίζει το όνομα που ακολουθεί σε μεταβλητή
- *chrs* μετατρέπει διάνυσμα κωδικών ASCII σε χαρακτήρες string
- *dttostr* μετατρέπει έναν πίνακα που περιέχει ημερομηνίες σε διαβαθμούμενη DT μορφή σε string array
- *ftocv* χαρακτήρες που αναπαριστούν αριθμούς σε N×K πίνακα
- *ftos* χαρακτήρες που αναπαριστούν αριθμούς σε 1×1 πίνακα
- *ftostrC* μετατρέπει έναν πίνακα σε string array χρησιμοποιώντας παραμέτρους σε μορφή της γλώσσας C
- *getf* φορτώνει αρχεία ASCII ή δυαδικά σε string
- *indcv* βρίσκει ευρετήριο στοιχείων σε χαρακτήρα διανύσματος
- *lower* μετατρέπει σε πεζά γράμματα
- *stof* μετατρέπει string σε κινητή υποδιαστολή
- *strindx* βρίσκει δείκτες ενός string σε ένα δεύτερο string
- *strlen* μήκος ενός string.
- *strsect* εξάγει ένα substring από ένα string.
- *strsplit* διασπά ένα N×1 διάνυσμα string σε ένα

N×K string array ανεξάρτητων συμβόλων

- *strsplitPad* διασπά ένα διάνυσμα string σε ένα string array ανεξάρτητων συμβόλων
- *strtodt* μετατρέπει ένα string array ημερομηνιών σε ένα πίνακα της διαβαθμούμενης μορφής DT
- *strtof* μετατρέπει ένα string array σε έναν αριθμητικό πίνακα
- *strtofcplx* μετατρέπει ένα string array σε μιγαδικό αριθμητικό πίνακα
- *upper* μετατρέπει σε κεφαλαία
- *vals* μετατρέπει από string σε αριθμητικό διάνυσμα κωδικών ASCII

2. Maple[28]

Ένα string είναι μια ακολουθία χαρακτήρων. Για να δημιουργήσει ο χρήστης ένα string, εσωκλείει μια οποιαδήποτε ακολουθία χαρακτήρων σε διπλά εισαγωγικά. Για παράδειγμα

> " This is a string";

"This is a string"

Ο χρήστης δεν μπορεί να αναθέσει 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) στο πρόγραμμα. Στις περισσότερες υλοποιήσεις του προγράμματος, αυτό σημαίνει ότι ένα string μπορεί να περιέχει πάνω από μισό εκατομμύριο χαρακτήρες.

Για να καθοριστεί το μήκος του string, μπορεί να χρησιμοποιηθεί η εντολή *length* ("What is the length of this string?");

Όλοι οι χαρακτήρες, πέρα από τα εισαγωγικά, προσμετρώνται. Κάθε κενό προσμετράται ως ένας χαρακτήρας. Ένα κενό string (empty ή null string)

[28] Monagan, M. B./Geddes, K. O./ Heal, K. M./Labahn, G./Vorkoetter S. M./ McCarron J./DeMarco P.: Maple Introductory Programming Guide. Maplesoft 2005, κεφ 1.2 σελ 7, κεφ 2.2 σελ 38.

αναπαρίσταται με διπλά εισαγωγικά χωρίς να εσωκλείονται χαρακτήρες, ούτε καν κενό διάστημα. Για παράδειγμα

> "";

“ ”

Υπάρχουν πολλές εντολές που μπορούν να χρησιμοποιηθούν για το χειρισμό των strings. Προκειμένου να εξάγει ο χρήστης μέρος του αρχικού string, μπορεί είτε να χρησιμοποιήσει την εντολή *substring(exprString, range);*, η οποία επιστρέφει μέρος του αρχικού string το οποίο υποδεικνύουμε με την range είτε να χρησιμοποιήσει subscripts. Για παράδειγμα

> S:=” abcdef”;

“abcdef”

> substring(S,3..7);

“cdef”

Α. Αλληλουχία strings

Όπως τα ονόματα έτσι και τα strings μπορούν να σχηματιστούν μέσω αλληλουχιών χρησιμοποιώντας την εντολή *cat* ή τον superseded concatenation τελεστής || .

Ο χρήστης μπορεί να δημιουργήσει ένα string χρησιμοποιώντας την εντολή *cat*, όπου η ακολουθία περιλαμβάνει οποιοδήποτε αριθμό εκφράσεων, χωριζόμενων με κόμμα.

cat(ακολουθία)

Η εντολή χρησιμοποιείται κατά κύριο λόγο για να ενώσει strings με ονόματα και ακέραιους και το αποτέλεσμα που επιστρέφει έχει την μορφή (όνομα ή string) του πρώτου ορίσματος της εντολής.

> cat("a", b);

"ab"

> cat("a", 2);

"a2"

> i := 5;

i := 5

> cat("The value of i is ", i, ".");

"The value of i is 5. "

B. The Concatenation Operator ||

Μπορεί επίσης ο χρήστης να concatenate strings χρησιμοποιώντας τον τελεστή || σε μια από τις ακόλουθες μορφές:

string || *name*

string || *naturalInteger*

string || *string*

string || *(expression)*

Ο τελεστής αυτός είναι ένας δυαδικός τελεστής ο οποίος χρειάζεται ένα string (ή ένα όνομα) ως αριστερό operand. Καθώς το string μπορεί να τεθεί στο αριστερό μέρος κάθε τέτοιου τελεστή, το πρόγραμμα δέχεται την διαδοχή των αλληλουχιών.

> "The "|| "value of i is " || i;

"The value of i is 5"

Γ. Ειδικοί χαρακτήρες στα strings

Για να εμφανιστούν σε ένα string τα διπλά εισαγωγικά θα πρέπει να εισαχθεί ο χαρακτήρας backslash (\) ακολουθούμενος από διπλό εισαγωγικό ("), στο σημείο που επιθυμεί ο χρήστης να εμφανιστεί το διπλό εισαγωγικό. Αυτό χρειάζεται να γίνει γιατί το πρόγραμμα δεν αναγνωρίζει ποιο διπλό εισαγωγικό ολοκληρώνει το string και το backslash λειτουργεί ως χαρακτήρας διαφυγής στο πρόγραμμα.

> "a\"b";

"a"b"

Ομοίως για να εμφανιστεί το backslash ως ένας από τους χαρακτήρες του string, ο χρήστης εισάγει δύο διαδοχικά backslashes.

> "a\\b";

"a\b"

Μια δεσμευμένη λέξη που εσωκλείεται σε διπλά εισαγωγικά είναι ένα έγκυρο string για το πρόγραμμα, που διακρίνεται από την χρήση της ως σύμβολο.

> "while";

"while"

Δ. Parsing Strings

Η εντολή *parse* δέχεται οποιοδήποτε string του προγράμματος και το αναλύει όπως εάν το string είχε εισαχθεί ή είχε διαβαστεί από ένα αρχείο.

parse(exprString, option);

Το string πρέπει να αποτελείται από μία και μόνο έκφραση. Η έκφραση αναλύεται και επιστρέφει χωρίς να έχει υπολογιστεί.

> parse("a+b");

a + b

> parse("a+b;");

a + b

Εάν το string είναι συντακτικά εσφαλμένο, τότε η εντολή επιστρέφει ως σφάλμα με την μορφή *"incorrect syntax in parse: ... (number)"*.

> parse("a++b");

Error, incorrect syntax in parse: `+` unexpected (4)

Εάν η option statement ορίζεται, το string πρέπει να αποτελείται από μία και μόνο δήλωση. Σε αυτήν την περίπτωση, η δήλωση αναλύεται και υπολογίζεται και μετά επιστρέφει το αποτέλεσμα.

> parse("sin(Pi)");

sin(¹)

> parse("sin(Pi)", statement);

0

Ε. Αναζητώντας ένα string

Για να εκτελεστεί μια αναζήτηση με ή χωρίς διάκριση κεφαλαίων πεζών γραμμάτων, ο χρήστης χρησιμοποιεί τις εντολές SearchText και searchtext αντιστοίχως.

SearchText(pattern, exprString, range);

searchtext(pattern, exprString, range);

ΣΤ. Μετατρέποντας εκφράσεις σε strings

Για να μετατρέψει ο χρήστης μια έκφραση σε string, χρησιμοποιεί την εντολή *convert > convert(a, string);*

"a"

*> convert(a+b-c*d/e, string);*

*"a+b-c*d/e"*

3. Mathematica

Πεζά και κεφαλαία γράμματα, αριθμοί, σημεία στίξης και κενό διάστημα σχηματίζουν τους βασικούς χαρακτήρες. Μια ακολουθία από χαρακτήρες μέσα σε εισαγωγικά ονομάζεται string[29].

In[1]:= "This is a string."

Out[1]:= This is a string.

Όταν το πρόγραμμα εμφανίζει στην οθόνη ένα string, το εμφανίζει χωρίς τα εισαγωγικά. Ο χρήστης μπορεί να δει τα εισαγωγικά ζητώντας την μορφή εισόδου των δεδομένων του string

In[1]:= InputForm[%]

Out[1]//InputForm= "This is a string."

Ένα string είναι μια τιμή και όπως άλλες τιμές(πχ αριθμοί, λίστες) αποτελεί και αυτή μια ενσωματωμένη συνάρτηση διαθέσιμη για την επεξεργασία strings. Οι λειτουργίες τους υποδηλώνονται από τα ονόματά τους, όπως φαίνεται και παρακάτω[30]:

- *In[1]:= StringLength["This is a test."]*

  Out[1]= 15

- *In[1]:= StringReverse["end"]*

  Out[1]= dne

[29] http://reference.wolfram.com/mathematica/ref/String.html
[30] http://reference.wolfram.com/mathematica/tutorial/OperationsOnStrings.html

- *In[1]:= StringTake["numerical", 5]*
  Out[1]= numer

- *In[1]:= StringDrop["numerical", -2]*
  Out[1]= numeric

- *In[1]:= StringPosition["numeric", "um"]*
  Out[1]= {{2, 3}}

- *In[1]:= StringInsert["numeric", "f", 5]*
  Out[1]= numefric

- *In[1]:= StringReplace["numeric", "ric"->"ro"]*
  Out[1]= numero

Μπορεί κανείς επίσης να μετατρέψει ένα string σε μια λίστα από χαρακτήρες με την ενσωματωμένη συνάρτηση *Characters*:

In[1]:= Characters["numeric"]
Out[1]= {n, u, m, e, r, i, c}

ή μπορεί να μετατρέψει την λίστα στην αρχική της μορφή δηλαδή σε string.

4. Matlab[31]

Το string είναι ένα array χαρακτήρων. Εσωτερικά οι χαρακτήρες αποθηκεύονται ως αριθμοί αλλά όχι κινητής υποδιαστολής. Έστω ότι ο χρήστης εισάγει κείμενο στο πρόγραμμα χρησιμοποιώντας μονά εισαγωγικά:

str = 'I am learning MATLAB this semester.'
str =

[31] Neuman, Edward: Programming in MATLAB, Tutorial 2 διαθέσιμο σε: http://www.math.siu.edu/matlab/tutorial2.pdf

I am learning MATLAB this semester.

Κάθε χαρακτήρας αναπαρίσταται εσωτερικά από την ASCII τιμή του. Για να δει ο χρήστης αυτήν την αναπαράσταση, χρησιμοποιεί την συνάρτηση *double*

str1 = double(str)

str1 =

Columns 1 through 12

73 32 97 109 32 108 101 97 114 110 105

110

Columns 13 through 24

103 32 77 65 84 76 65 66 32 116 104

105

Columns 25 through 35

115 32 115 101 109 101 115 116 101 114 46

Μπορεί ο χρήστης να μετατρέψει το array str1 σε μορφή χαρακτήρα χρησιμοποιώντας την συνάρτηση *char*

str2 = char(str1)

str2 =

I am learning MATLAB this semester.

Για να συγκρίνει δύο strings ως προς την ισότητά τους χρησιμοποιεί την συνάρτηση *strcmp*

iseq = strcmp(str, str2)

iseq =

1

Δύο strings μπορούν να σχηματίσουν αλληλουχία χρησιμοποιώντας την συνάρτηση *ctrcat*

strcat(str,str2)

ans =

I am learning MATLAB this semester. I am learning MATLAB this semester.

Ας σημειωθεί ότι τα strings σε αλληλουχία δεν χωρίζονται με κενό διάστημα.

Ο χρήστης μπορεί να δημιουργήσει δισδιάστατο array από strings. Για τον σκοπό αυτό είναι προτιμότερο να χρησιμοποιήσει το *cell array* παρά το δισδιάστατο array, για το λόγο ότι το αριθμητικό array πρέπει να έχει τον ίδιο αριθμό στηλών σε κάθε γραμμή. Για παράδειγμα:

carr = {'first name'; 'last name'; 'hometown'}

carr =

'first name'

'last name'

'hometown'

Ας σημειωθεί ότι ο χρήστης πρέπει να χρησιμοποιεί άγκιστρα αντί για αγκύλες.

Το πρόγραμμα έχει δύο συναρτήσεις για να κατηγοριοποιεί τους χαρακτήρες: *isletter* και *isspace*.

Υπάρχουν ακόμα δύο σημαντικές συναρτήσεις που χρησιμοποιούνται για την μετατροπή αριθμών σε strings, δηλαδή οι συναρτήσεις *int2str* και *num2str*. Η *int2str* στρογγυλοποιεί το όρισμά του (πίνακα) σε ακεραίους και μετατρέπει το αποτέλεσμα σε string πίνακα. Για παράδειγμα:

Έστω

A = randn(3)

A =

-0.4326	*0.2877*	*1.1892*
-1.6656	*-1.1465*	*-0.0376*
0.1253	*1.1909*	*0.3273*

Τότε

B = int2str(A)

B =

0	*0*	*1*
-2	*-1*	*0*
0	*1*	*0*

Η συνάρτηση *num2str* παίρνει ένα array και το μετατρέπει σε array string. Εκτελώντας την συνάρτηση αυτή στον πίνακα Α που ορίστηκε παραπάνω, θα προκύψει:

```
C = num2str(A)
C =
-0.43256    0.28768    1.1892
-1.6656     -1.1465    -0.037633
 0.12533    1.1909     0.32729
```

Η συνάρτηση επίσης χρησιμοποιείται συχνά και για την ονομασία plots σε συνδυασμό με τις εντολές *title, xlabel, ylabel* και *text*.

Α. Δημιουργώντας strings με αλληλουχία

Τα strings συχνά δημιουργούνται από αλληλουχία μικρότερων στοιχείων (π.χ. strings, τιμές κτλ) τόσο οριζόντια όσο και κατακόρυφα. Δύο συνήθεις μέθοδοι για αλληλουχία είναι να χρησιμοποιεί ο χρήστης είτε τον τελεστή αλληλουχίας ([]) είτε την συνάρτηση *sprintf*. Για παράδειγμα:

```
numChars = 28;
s = ['There are ' int2str(numChars) ' characters here']
s = sprintf('There are %d characters here\n', numChars)
```

Β. Σύγκριση μεθόδων αλληλουχίας

Όταν δημιουργεί ο χρήστης strings αλληλουχίας είναι προτιμότερη συχνά η συνάρτηση sprintf σε σχέση με τον τελεστή αλληλουχίας [] για τους εξής λόγους:

- Είναι ευκολότερη η ανάγνωση, ειδικά όταν σχηματίζονται σύνθετες εκφράσεις
- Παρέχει στον χρήστη μεγαλύτερο έλεγχο στην μορφή των αποτελεσμάτων
- Εκτελείται πιο γρήγορα
- Υπάρχει επίσης και η δυνατότητα να χρησιμοποιήσει ο χρήστης την συνάρτηση strcat, η οποία όμως δεν συνιστάται για απλές αλληλουχίες.

Γ. Αποθήκευση Strings Arrays σε Cell Array

Είναι συχνά προτιμότερο να αποθηκεύει ο χρήστης τα strings arrays σε cell array αντί σε character array, ειδικά όταν τα strings έχουν διαφορετικό μήκος. Διότι στα character array τα strings πρέπει να είναι ίσου μήκους, γεγονός το οποίο υποχρεώνει τον χρήστη να τα εξομοιώνει με την χρήση του χαρακτήρα του κενού διαστήματος.

Δ. Μετατροπή Strings σε Cell Arrays

Ο χρήστης μπορεί να μετατρέψει βασικά character arrays σε cell arrays χρησιμοποιώντας τις συναρτήσεις *cellstr* και *char*

Ε. Αναζήτηση και αντικατάσταση μέσω κανονικών εκφράσεων

Η χρήση κανονικών εκφράσεων προσφέρει έναν ευέλικτο τρόπο αναζήτησης και αντικατάστασης χαρακτήρων ή φράσεων μέσα σε ένα string. Περιγραφή συνάρτησης:

regexp	ταίριασμα κανονικών εκφράσεων
regexpi	ταίριασμα κανονικών εκφράσεων, αγνοώντας πεζά κεφαλαία
regexprep	αντικατάσταση string χρησιμοποιώντας κανονική έκφραση

5. Scilab[32],[33]

Τα strings θεωρούνται ένας πίνακας 1x1. Μπορούν να δημιουργηθούν με την χρήση είτε μονών είτε διπλών εισαγωγικών. Τα στοιχεία των strings οριοθετούνται με διπλά κόμματα ή διπλά εισαγωγικά. Κάθε πίνακας string που εισάγεται έχει το δικό του μέγεθος. Αλληλουχία strings επιτυγχάνεται με την λειτουργία + . Πίνακες χαρακτήρων strings δημιουργούνται όπως οι κανονικοί πίνακες π.χ. χρησιμοποιώντας αγκύλες. Ένα βασικό χαρακτηριστικό των πινάκων με χαρακτήρες strings είναι η δυνατότητα επεξεργασίας και δημιουργίας συναρτήσεων. Μπορεί επίσης να ενσωματωθεί στο πρόγραμμα η συμβολική επεξεργασία μαθηματικών αντικειμένων μέσω πινάκων με χαρακτήρες strings. Οι χαρακτήρες strings

[32] Scilab Help Browser, Scilab Manual, Strings, κεφ. XLIX, σελ. 2129
[33] http://www.spas.cnrs-gif.fr/ScilabPasapas-Ch10Sc01.html

μπορούν να χρησιμοποιηθούν για την αυτόματη δημιουργία νέων συναρτήσεων.

Το μήκος ενός string είναι μεταβλητό. Η συνάρτηση *length* επιστρέφει το μήκος ενός string ως αριθμό χαρακτήρων. Η συνάρτηση *emptystr* δημιουργεί ένα κενό string, του οποίου το μήκος είναι μηδενικό.

IV. ΜΗΤΡΕΣ

1. Gauss

Οι μήτρες μπορούν να περιέχουν αριθμητικά δεδομένα ή χαρακτήρες ή και τα δύο. Το πρόγραμμα δεν μπορεί να κάνει διάκριση ανάμεσα στα δύο. Είναι δισδιάστατα arrays διπλής ακρίβειας. Όλα τα στοιχεία μιας μήτρας αποθηκεύονται σε μορφή κινητής υποδιαστολής διπλής ακρίβειας, και καταλαμβάνουν 8 bytes μνήμης.

Κάθε κελί σε μια μήτρα μπορεί να περιέχει μέχρι 8 χαρακτήρες κειμένου ή αριθμητικά δεδομένα με διάστημα περίπου 1.0E ± 35[34]. Αν εισάγει ο χρήστης κείμενο μεγαλύτερο των 8 χαρακτήρων σε κελιά μιας μήτρας, το κείμενο θα περικοπεί.

Νέες μήτρες μπορούν να οριστούν κάθε στιγμή. Ο πιο απλός τρόπος είναι να αναθέσει ο χρήστης μια τιμή σε μια μήτρα, πράγμα που επιτυγχάνεται με δύο τρόπους, δηλαδή αναθέτοντας είτε μια σταθερή μεταβλητή είτε το αποτέλεσμα μιας λειτουργίας.

Μήτρες με ένα μόνο στοιχείο αναφέρονται ως αριθμητικές (scalars) (1×1 μήτρες), και μήτρες με μία μόνο γραμμή ή στήλη ως διανύσματα (1×N ή N×1 μήτρες).

Κάθε μήτρα ή διάνυσμα μπορεί να έχει δύο δείκτες. Τα διανύσματα μπορούν να έχουν και έναν δείκτη. Τα scalars μπορεί να έχουν έναν ή δύο δείκτες - καθώς scalars, διανύσματα και μήτρες είναι του ίδιου τύπου δεδομένα για το πρόγραμμα.

Η πλειονότητα των συναρτήσεων και τελεστών του προγράμματος δέχονται ως ορίσματα μήτρες. Οι ακόλουθες συναρτήσεις και τελεστές χρησιμοποιούνται για να ορίσουν, αποθηκεύσουν και να φορτώσουν μήτρες:

[ ] Ορισμός δεικτών σε μήτρες

[34] *Rossi, Eduardo:* Introduction to Gauss Programming Language. University of Pavia, October 2006. κεφ. 3, σελ. 21.

=	Τελεστής ανάθεσης
|	Κάθετη αλληλουχία (concatenation)
_	Οριζόντια αλληλουχία (concatenation)
con	Εισαγωγή αριθμητικού από το πληκτρολόγιο
cons	Εισαγωγή χαρακτήρα από το πληκτρολόγιο
load	Φόρτωση μήτρας (ίδια με loadm).
readr	Ανάγνωση από μια μήτρα του προγράμματος ή από αρχείο συνόλου δεδομένων
save	Αποθήκευση μητρών, διαδικασιών και strings στον δίσκο
saved	Μετατροπή μήτρας σε σύνολο δεδομένων του προγράμματος
stof	Μετατροπή string σε μήτρα
submat	Εξαγωγή υπομήτρας
writer	Εγγραφή δεδομένων σε ένα σύνολο δεδομένων του προγράμματος

Α. Οι βασικοί τελεστές[35]

Το πρόγραμμα έχει οκτώ μαθηματικούς τελεστές και έξι σχεσιακούς.

Οι μαθηματικοί τελεστές είναι οι ακόλουθοι:

+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
’	Μετάθεση
%	Διαίρεση modulo
!	Παραγοντοποίηση
^	Εκθετοποίηση

Οι σχεσιακοί τελεστές είναι οι ακόλουθοι: [36]

== EQ	ισούται
> GT	μεγαλύτερο από
/= NE	δεν ισούται

[35] *Aptech Systems Inc:* User Guide, Version 9.0, July 2008, κεφ. 11.2., σελ 11-4 επ.
[36] *Aptech Systems Inc:* User Guide, Version 9.0, July 2008, κεφ. 11.3., σελ 11.9 επ.

< LT	μικρότερο από
>= GE	μεγαλύτερο από/ίσο
<= LE	μικρότερο από/ίσο

Μπορεί να χρησιμοποιήσει ο χρήστης είτε το σύμβολο είτε το ακρωνύμιο με τα δύο γράμματα.

Υπάρχουν επίσης και πέντε λογικοί τελεστές:[37]

NOT

AND

OR

XOR

EQV

Οι πράξεις γίνονται από τα αριστερά προς τα δεξιά και ακολουθούν την εξής προτεραιότητα:

- άγκιστρο ή αγκύλη
- αναστροφή
- παραγοντοποίηση
- εκθετοποίηση
- άρνηση
- πολλαπλασιασμός και διαίρεση
- πρόσθεση και αφαίρεση
- σχεσιακός τελεστής dot
- λογικός τελεστής dot
- σχεσιακοί τελεστές
- λογικοί τελεστές
- δείκτες γραμμών και στηλών

Έστω οι μήτρες a,b,c,d, οπότε ισχύουν οι ακόλουθες πράξεις μεταξύ μητρών:

a = b+c-d;

[37] *Aptech Systems Inc:* User Guide, Version 9.0, July 2008, κεφ. 11.4., σελ 11-13 επ.

*a = b'*c';*

a = (b+c)(d-e);*

a = ((b+c)(d+e))/((b-c)*(d-e));*

*a = (b*c)';*

Η διαίρεση μεταξύ αριθμού και μήτρας γίνεται στοιχείο προς στοιχείο ενώ αντίθετα στην περίπτωση διαίρεσης μήτρας με μήτρα, το πρόγραμμα υπολογίζει μια γενικευμένη αντιστροφή, η οποία είναι της μορφής a = b/c → ca = b που οδηγεί στην μορφή

a = b/c => a = c^{r-1}b (c square) ή a = $(c'c)^{-1}$c'b (c μη τετραγωνική)

Β. Συνένωση

Υπάρχουν δύο τελεστές συνένωσης:

~ οριζόντιος

| κάθετος

Με την συνένωση προστίθεται μια μήτρα, στα δεξιά ή στο κάτω μέρος μιας άλλης. Προφανώς θα πρέπει οι σχετικές γραμμές και στήλες να ταιριάζουν. Για παράδειγμα

έστω ότι ο χρήστης χρησιμοποιεί τις εξής μήτρες:

a και b, με ra και rb γραμμές και ca και cb στήλες και το αποτέλεσμα των αλληλουχιών φαίνεται στην μήτρα c:

Διαστάσεις a	Διαστάσεις b	Πράξη	Διαστάσεις c	Συνθήκη
ra × ca	rb × cb	c = a~b	ra × (ca + cb)	ra = rb
ra × ca	rb × cb	c = a | b	(ra + rb) × ca	ca = cb

Το πρόγραμμα επιτρέπει όλους τους μαθηματικούς και λογικούς τελεστές να έχουν ως πρόθεμα την τελεία (dot):

a = b.+c; a = (b+c).*d'; a = b.==c;

Αυτό ενημερώνει το πρόγραμμα ότι οι πράξεις θα γίνουν στοιχείο με στοιχείο.

Το πρόγραμμα διαθέτει πληθώρα χρήσιμων συναρτήσεων, εύκολες σε χρήση και πιο ακριβής από το να προσπαθήσει ο χρήστης να τις δημιουργήσει από την αρχή. Αυτές είναι:

A = INV(B);	Η μήτρα Α είναι η αντίστροφη της μήτρας Β.
A = SQRT(B);	Η μήτρα Α περιέχει την τετραγωνική ρίζα των στοιχείων της μήτρας Β.
A = DIAG(B);	Η μήτρα Α περιέχει τα στοιχεία της κύριας διαγωνίου της μήτρας Β.
A = SUMC(B);	Η μήτρα Α περιέχει το άθροισμα των στοιχείων κάθε στήλης της μήτρας Β.
A = SUMC(B');	Η μήτρα Α περιέχει το άθροισμα των στοιχείων κάθε γραμμής της μήτρας Β.
A = STDC(B);	Η μήτρα Α περιέχει την τυπική απόκλιση των στοιχείων κάθε στήλης της μήτρας Β.
A = LAG1(B);	Η μήτρα Α περιέχει τιμές της μήτρας Β με υστέρηση μια περιόδου.
A = LAGN(B,N);	Η μήτρα Α περιέχει τιμές της μήτρας Β με υστέρηση Ν περιόδων.
A = PACKR(B);	Η μήτρα Α περιέχει μόνο εκείνες τις γραμμές της μήτρας Β οι οποίες δεν έχουν χαμένες τιμές σε κανένα από τα στοιχεία τους.
A = ONES(10,1);	Η Α είναι μια 10x1 μήτρα και όλα τα στοιχεία της είναι ίσα με 1.
A = ZEROS(10,1);	Η Α είναι μια 10x1 μήτρα και όλα τα στοιχεία της είναι ίσα με 0.
A = EYE(10);	Η Α είναι μια 10x10 μοναδιαία μήτρα.
A = RNDN(r,c);	Η Α είναι μια rxc μήτρα αποτελούμενη από τυχαίους αριθμούς έχοντας αριθμητικό μέσο μηδέν και τυπική απόκλιση ένα.
A = ABS(B);	Η μήτρα Α περιέχει τις απόλυτες τιμές της μήτρας Β.
Y = ROWS(B);	Υ είναι ο αριθμός των σειρών της μήτρας Β.
Y = COLS(B);	Υ είναι ο αριθμός των στηλών της μήτρας Β.
Y = DET(B);	Υ είναι η ορίζουσα της μήτρας Β.

2. Maple

Ιδιαίτερες περιπτώσεις της δομής array είναι οι δομές διάνυσμα και μήτρα.[38]

Α.Διανύσματα

Διάνυσμα είναι ένα μονοδιάστατο array του οποίου η αρίθμηση ξεκινάει από το 1. Τα διανύσματα μπορούν εύκολα να εισαχθούν και χωρίς τη χρήση του array, με τη χρήση των συμβόλων <, > ή την εντολή *Vector*:

[38] Monagan, Michael: Programming in Maple: The Basics, κεφ. 4.1, σελ. 30 διαθέσιμο σε: http://theor.jinr.ru/Documents/MapleV/program/program.html

<στοιχείο1, στοιχείο2,..>

ή

Vector(n,m,L,f),

όπου n,m= η διάσταση του, L= οι λίστες του διανύσματος, f= η συνάρτηση που χρησιμοποιείται για να δημιουργηθούν τα στοιχεία του διανύσματος.

Β. Μήτρες

Μήτρα είναι ένα δισδιάστατο array του οποίου η αρίθμηση ξεκινάει από το 1 και για τις δυο διαστάσεις. Για τη δημιουργία μιας μήτρας το πρόγραμμα διαθέτει την εντολή *Matrix*:

Matrix(n,m,L,f)

Matrix(n,m,init,shape)

όπου n= το πλήθος των γραμμών της μήτρας, m=το πλήθος των στηλών της μήτρας, L= οι λίστες ή τα διανύσματα των στοιχείων, f = η συνάρτηση που χρησιμοποιείται για να δημιουργηθούν τα στοιχεία της μήτρας

init= Maple procedure, table, array, list, Array, Matrix, Vector, set of equations, or expression of type algebraic; initial values for the Matrix, shape=identity,scalar[x],zero,constant[x],diagonal,band[b],band[b1,b2], symmetric, skewsymmetric, antisymmetric, hermitian, skewhermitian, antihermitian, triangular, Hessenberg.

Ένας εναλλακτικός τρόπος ορισμού μιας μήτρας είναι η χρήση των συμβόλων <,>,|. Ουσιαστικά, ο χρήστης ορίζει τη μήτρα από τα διανύσματα που την αποτελούν. Το σύμβολο | καθορίζει τις στήλες της μήτρας:

<<a11,a21,…,am1>|<a12,a22,…am2>|..|<a1n,a2n,…amn>>

Γ. Πράξεις με Διανύσματα και Μήτρες[39]

πράξη	Περιγραφή

[39] *Maplesoft*: Maple User Manual, 2008, σελ. 146

A+B	Προσθέτει δύο Μήτρες ή δύο Διανύσματα
A.B	Πολλαπλασιάζει Μήτρες ή Διανύσματα
λ*A	Βαθμωτός πολλαπλασιασμός αριθμού με Μήτρα ή διάνυσμα
A^n	Υπολογίζει δυνάμεις Μητρών

Δ. Ειδικές μήτρες και διανύσματα[40]

Εντολή	Περιγραφή
ZeroMatrix(n,m);	Δημιουργεί μια μηδενική μήτρα
IdentityMatrix(διάσταση);	Δημιουργεί μια ταυτοτική μήτρα
HilbertMatrix(n,m,expr);	Δημιουργεί την μήτρα του Hilbert
HankelMatrix(L,n);	Δημιουργεί τη μήτρα του Hankel
BezoutMatrix(p(x), q(x), x);	Δημιουργεί τη μήτρα του Bazout
HouseholterMatrix	Δημιουργεί τη μήτρα του Householder
JordanBlockMatrix	Δημιουργεί τη μήτρα του Jordan
SylvesterMatrix	Δημιουργεί τη μήτρα του Sylvester
ToeplitzMatrix	Δημιουργεί τη μήτρα του Toeplitz
VandermondeMatrix	Δημιουργεί τη μήτρα του Vandermonde
CompanionMatrix(πολυώνυμο)	Δημιουργεί το συνοδό

[40] *Maplesoft*: Maple User Manual, 2008, σελ. 151

	πίνακα ενός πολυωνύμου
GivensRotation(διάνυσμα)	Κατασκευάζει τη μήτρα για δοσμένη περιστροφή
UnitVector	Μοναδιαίο Διάνυσμα
RandomMatrix(n,m)	Δημιουργεί μια μήτρα με τυχαίους αριθμούς
RandomVector(αριθμός στοιχείων)	Δημιουργεί ένα διάνυσμα με τυχαίους αριθμούς

3. Mathematica

Παραδοσιακά οι μήτρες δηλώνονται με κεφαλαία γράμματα. Στο πρόγραμμα θα δηλώνονται με μικρά γράμματα, αφού τα κεφαλαία είναι δεσμευμένα για τις ενσωματωμένες συναρτήσεις. Για την εισαγωγή μιας μήτρας στο πρόγραμμα, ο χρήστης πρώτα πληκτρολογεί το όνομα της μήτρας και μετά το σύμβολο της ισότητας ¨ = ¨ .

Το πρόγραμμα θεωρεί μια μήτρα ως μια λίστα από λίστες. Κάθε σειρά είναι ενσωματωμένη σε άγκιστρα {}, και τα στοιχεία χωρίζονται με κόμμα. Οι σειρές είναι χωρισμένες με κόμμα και ολόκληρη η μήτρα εσωκλείεται σε άγκιστρα.

Α. Πράξεις μητρών[41]

Αν δυο μήτρες έχουν τις ίδιες διαστάσεις, μπορεί ο χρήστης:

- να υπολογίσει το άθροισμά τους, με τη χρήση του συμβόλου (+)
- να βρει τη διαφορά τους, με χρήση του συμβόλου (-)
- να κάνει αριθμητικό πολλαπλασιασμό
- να πολλαπλασιάσει μήτρες (εάν ο αριθμός των στηλών της πρώτης μήτρας είναι ίσος με τον αριθμό των γραμμών της δεύτερης). Στο πρόγραμμα ως τελεστή πολλαπλασιασμού για μήτρες χρησιμοποιείται η τελεία (dot)
- να βρει τον ανάστροφο πίνακα με την εντολή *transpose*, δηλαδή την μήτρα που προκύπτει αλλάζοντας τις γραμμές και τις στήλες της αρχικής μήτρας

[41] http://reference.wolfram.com/mathematica/guide/MatrixOperations.html

- να βρει την δύναμη μιας μήτρας με την εντολή *MatrixPower*, το πρώτο όρισμα της εντολής είναι η μήτρα και το δεύτερο η επιθυμητή δύναμη
- να βρει την αντίστροφη μήτρα μιας τετραγωνικής μήτρας, εάν αυτή υπάρχει, που είναι η μήτρα της οποίας το γινόμενο με την αρχική μήτρα δίνει την μοναδιαία μήτρα
- να βρει την ορίζουσα μιας τετραγωνικής μήτρας, που είναι ένας αριθμός μη μηδενικός αν και μόνο αν η μήτρα είναι μη μοναδιαία. Ο υπολογισμός της ορίζουσας είναι πολύ εύκολος με την χρήση της εντολής *Det*.

Κάθε πράξη μεταξύ μητρών μπορεί να πραγματοποιηθεί σε μια μήτρα η οποία δέχεται ως είσοδο είτε αριθμητικά είτε καθαρά συμβολικά δεδομένα.

4. Matlab[42],[43]

Ο χρήστης μπορεί να εισάγει μια μήτρα με έναν από τους παρακάτω τρόπους:

- Εισάγοντας λίστα από στοιχεία
- Φορτώνοντας μήτρες από αρχεία
- Χρησιμοποιώντας ενσωματωμένες συναρτήσεις
- Χρησιμοποιώντας M-files που δημιούργησε ο ίδιος

Η λίστα των στοιχείων που εσωκλείονται σε αγκύλες σχηματίζουν μια μήτρα. Τα στοιχεία της γραμμής μιας μήτρας, χωρίζονται με το κενό διάστημα ή με κόμμα. Με τη χρήση του ελληνικού ερωτηματικού « ; », δηλώνεται το τέλος μια γραμμής. Τα στοιχεία μπορεί να είναι αριθμοί, μεταβλητές ή αριθμητικές παραστάσεις. Για παράδειγμα:

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

A =

16 3 2 13

[42] *The MathWorks:* Matlab 7 Getting Started Guide, 2008, κεφ. 2 σελ. 2.1
[43] http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/math/f1-84864.html

5 10 11 8

9 6 7 12

4 15 14 1

Όταν ένα από τα στοιχεία της μήτρας είναι μιγαδικός αριθμός τότε ολόκληρη η μήτρα μετατρέπεται σε μιγαδική.

Το σύμβολο της άνω και κάτω τελείας, : , μπορεί να χρησιμοποιηθεί από τον χρήστη σε αρκετές περιπτώσεις. Για παράδειγμα,

Η έκφραση 1:10 είναι ένα διάνυσμα γραμμή το οποίο περιέχει ακεραίους από το 1 έως το10:

1 2 3 4 5 6 7 8 9 10

Επίσης , ο χρήστης μπορεί να προσθέσει ένα βήμα αύξησης:

x=100:-7:50

x=

100 93 86 79 72 65 58 51

Μπορεί να χρησιμοποιηθεί και για να παραστήσει όλες τις γραμμές ή όλες τις στήλες μιας μήτρας.

Το πρόγραμμα διαθέτει μια ποικιλία συναρτήσεων για την κατασκευή σχεδόν όλων των γνωστών μητρών. Από τις πιο βασικές είναι:

Το πρόγραμμα διαθέτει τέσσερις συναρτήσεις οι οποίες δημιουργούν βασικούς τύπους μητρών και είναι οι εξής:

- zeros(m,n) - όλα τα στοιχεία της μήτρας είναι 0
- ones(m,n) - όλα τα στοιχεία της μήτρας είναι 1
- eyes(m,n) - μοναδιαίος πίνακας
- rand(m,n) - τυχαία στοιχεία ομοιόμορφα κατανεμημένα
- randn(m,n) - τυχαία στοιχεία κανονικά κατανεμημένα

Για παράδειγμα:

Z = zeros(2,4)

Z =

0 0 0 0

0 0 0 0

F = 5*ones(3,3)

```
F =
     5 5 5
     5 5 5
     5 5 5

N = fix(10*rand(1,10))
N =
      9 2 6 4 8 7 4 0 8 4

R = randn(4,4)
R =
    0.6353    0.0860   -0.3210   -1.2316
   -0.6014   -2.0046    1.2366    1.0556
    0.5512   -0.4931   -0.6313   -0.1132
   -1.0998    0.4620   -2.3252    0.3792
```

Επίσης το πρόγραμμα διαθέτει μια ενσωματωμένη συνάρτηση για την δημιουργία μαγικής τετραγωνικής μήτρας σχεδόν οποιουδήποτε μεγέθους, την *magic*:

```
B = magic(4)
B =
    16   2   3  13
     5  11  10   8
     9   7   6  12
     4  14  15   1
```

Συνένωση

Για την ένωση δύο μητρών σε μια μεγαλύτερη, ο χρήστης μπορεί αν χρησιμοποιήσει τον τελεστή συνένωσης, που είναι αγκύλες []. Για παράδειγμα:

```
B = [A  A+32; A+48  A+16]
```

Το αποτέλεσμα είναι μια μήτρα 8x8, η οποία προκύπτει από την ένωση των τεσσάρων υπομητρών:

B =

16	3	2	13	48	35	34	45
5	10	11	8	37	42	43	40
9	6	7	12	41	38	39	44
4	15	14	1	36	47	46	33
64	51	50	61	32	19	18	29
53	58	59	56	21	26	27	24
57	54	55	60	25	22	23	28
52	63	62	49	20	31	30	17

Πράξεις Μητρών

Το πρόγραμμα εκτελεί μεταξύ των μητρών όλες τις γνωστές πράξεις:

+ Πρόσθεση, C = A + B

- Αφαίρεση, C = A - B

* Πολλαπλασιασμό, C = A * B

/ Διαίρεση από δεξιά, επιλύει το σύστημα A*x = b, (x = A / b)

\ Διαίρεση από αριστερά, επιλύει το σύστημα y*B = d,(y =d \ B)

^ Ύψωση σε δύναμη

Οι πράξεις της πρόσθεσης και της αφαίρεσης γίνονται εξ ορισμού στοιχείο με στοιχείο. Για να μπορέσει ο χρήστης να εκτελέσει και τις υπόλοιπες πράξεις στοιχείο με στοιχείο, το πρόγραμμα διαθέτει τα ακόλουθα σύμβολα πράξεων:

.* για τον πολλαπλασιασμό

./ για την διαίρεση από δεξιά

.\ για την διαίρεση από αριστερά

.^ για την ύψωση σε δύναμη

Μια ακόμα πράξη είναι και η εύρεση ανάστροφης μήτρας. Για την πράξη αυτή ο χρήστης χρησιμοποιεί το σύμβολο ΄ μετά το όνομα της μήτρας.

5. Scilab[44]

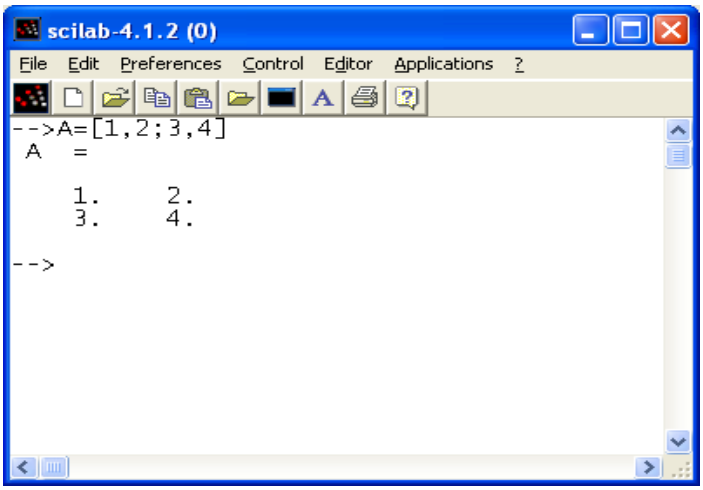
Στο πρόγραμμα οι αριθμοί κατά κύριο λόγο αποθηκεύονται σε μήτρες, οι οποίες θεωρούνται πίνακας. Μήτρες με μόνο μία στήλη ή με μία μόνο γραμμή ονομάζονται διανύσματα (οι μήτρες και τα διανύσματα μοιάζουν πολύ με τα arrays, όπως αυτά χρησιμοποιούνται σε γλώσσες προγραμματισμού όπως οι C, Visual Basic κτλ).

Α. Δημιουργία μητρών. Ανάκτηση δεδομένων από μήτρες.

Για να δημιουργήσει ο χρήστης μια μήτρα 2x2, θα δώσει την ακόλουθη εντολή:

A=[1,2;3,4]

Γενικά, το κόμμα (ή το κενό διάστημα) χρησιμοποιείται για να διαχωρίσει στοιχεία σε μια σειρά (ή γραμμή) και η αγγλική άνω τελεία (ελληνικό ερωτηματικό) για να διαχωρίσει τις στήλες.



Β. Ορισμένες ειδικές μήτρες

Δημιουργία μοναδιαίας μήτρας 2x2:

C=eye(2,2)

Δημιουργία μήτρας 3x2 με όλα τα στοιχεία της «1»:

D=ones(3,2)

Δημιουργία μήτρας 3x2 με όλα τα στοιχεία της «0»:

[44]Haugen, Finn: Master Scilab! 2008 διαθέσιμο σε: http://home.hit.no/~finnh/scilab_scicos/scilab/index.htm#matrix

E=zeros(3,2)

Γ. Πράξεις με μήτρες

Το πρόγραμμα εκτελεί μεταξύ μητρών όλες τις γνωστές πράξεις

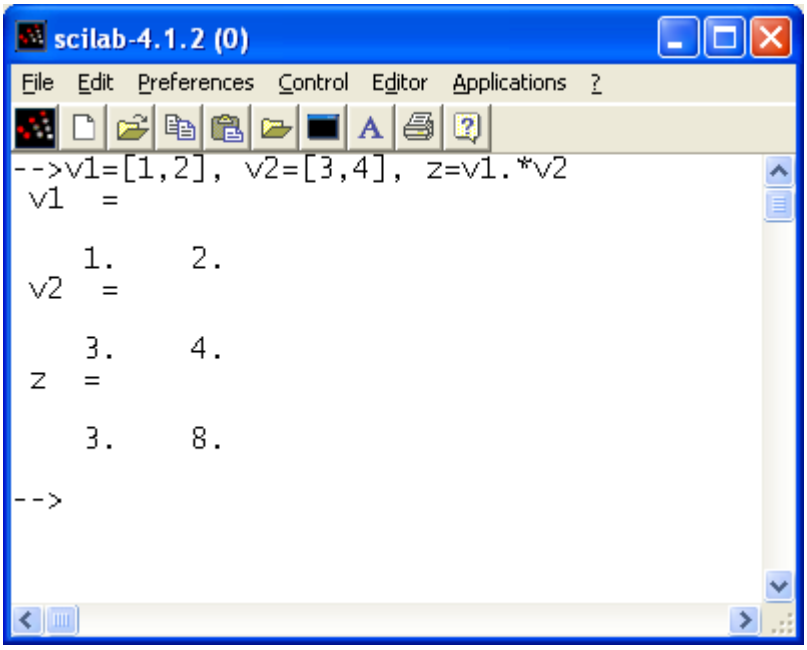
’	αναστροφή
+	πρόσθεση
-	αφαίρεση
*	πολλαπλασιασμό
\	διαίρεση από αριστερά
/	διαίρεση από δεξιά
ˆ	ύψωση σε δύναμη

Πρόσθεση δύο μητρών:	F=A+C
Πολλαπλασιασμός δύο μητρών:	G=A*C
Εύρεση ανάστροφης μήτρας:	H=inv(A)
Αφαίρεση δύο μητρών:	F=A-C

Για πράξεις στοιχείο με στοιχείο, ο χρήστης χρησιμοποιεί το τελεστή τελεία (dot). Για παράδειγμα:

έστω τα διανύσματα

v1= [1,2], c2=[3,4], z=v1.*v2



Τα σύμβολα που έχει στη διάθεση του ο χρήστης για πράξεις στοιχείο με στοιχείο είναι τα ακόλουθα:

.* για τον πολλαπλασιασμό

.\ για την διαίρεση από αριστερά

./ για την διαίρεση από δεξιά

.^ για την ύψωση σε δύναμη

.*. για το γινόμενο kronecker

./. για την διαίρεση kronecker από δεξιά

.\. για την διαίρεση kronecker από αριστερά

ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ

I. ΑΡΧΕΙΑ INPUT/OUTPUT

1. Gauss

Το πρόγραμμα διαχειρίζεται δεδομένα σε διάφορους τύπους. Μπορεί να διαβάσει και να δημιουργήσει βασικά αρχεία text και προηγούμενες μορφές υπολογιστικών φύλλων καθώς και, μέσω των δικών του τύπων, να αποθηκεύσει μήτρες, datasets ή δείγματα κώδικα.

GAUSS File Types	File Extension
GAUSS datasets	.dat, .dht (files come in pairs)
GAUSS matrices	.fmt
ASCII files (normal text)	anything

Ειδικότερα:[45]

A. Data Sets

Σε αυτήν την μορφή είναι προτιμότερο να αποθηκεύονται δεδομένα που περιέχονται σε μια μήτρα για να χρησιμοποιηθούν μέσα στο πρόγραμμα. Η χρήση αυτών των data sets παρέχει στον χρήστη ταχύτητα στην ανάγνωση και γραφή δεδομένων. Πολλές βιβλιοθήκες συναρτήσεων είναι σχεδιασμένες να διαβάζουν δεδομένα από τέτοια data sets.

B. ASCII Files

Το πρόγραμμα παρέχει την δυνατότητα ο χρήστης να μπορεί να γράψει και να διαβάσει αρχεία ASCII. Το γεγονός αυτό είναι ιδιαίτερα σημαντικό για την ανταλλαγή αρχείων ASCII μεταξύ του προγράμματος αυτού και άλλων προγραμμάτων, μιας και τα περισσότερα από αυτά επιτρέπουν την δημιουργία και ανάγνωση αρχείων ASCII.

B.1. Matrix Data

[45] *Aptech Systems Inc:* User Guide, Version 9.0, July 2008, κεφ. 21, σελ 21-1 επ.

- Ανάγνωση: με την εντολή *load* μπορούν να αναγνωστούν αρχεία που περιέχουν αριθμητικά δεδομένα που χωρίζονται με κενά διαστήματα ή κόμματα και είναι αρκετά μικρά ώστε να χωρούν σε μια μήτρα ή string.

- Γραφή: ο χρήστης μπορεί να γράψει σε ένα αρχείο ASCII, το οποίο με την εντολή *print* ή *printfm* θα εμφανιστεί σε βοηθητική έξοδο. Τα αρχεία που θα προκύψουν είναι βασικά αρχεία ASCII και μπορούν να επεξεργαστούν με τον editor του προγράμματος είτε οποιονδήποτε άλλο editor κειμένου. Οι εντολές *output* και *outwidth* χρησιμοποιούνται για τον έλεγχο της βοηθητικής εξόδου. Οι εντολές *print* ή *printfm* χρησιμοποιούνται για να ελέγχουν το τι στέλνεται στο αρχείο εξόδου. Με την εντολή *screen* το παράθυρο ελαχιστοποιείται και μεγεθύνεται.

Γ. General File I/O

Η εντολή *getf* θα διαβάσει ένα αρχείο και θα το επιστρέψει σε μια μεταβλητή string. Με αυτόν τον τρόπο μπορεί να διαβαστεί οποιοδήποτε αρχείο αρκεί να χωρά σε μια μεταβλητή μονού string.

Για να διαβάσει ο χρήστης αρχεία σειριακά, χρησιμοποιεί την εντολή *fopen* για να ανοίξει το αρχείο και τις εντολές *fgets*, *fputs* και συναφείς εντολές για να διαβάσει και να γράψει το αρχείο. Με την εντολή *ftell* εντοπίζεται η τρέχουσα θέση σε ένα αρχείο.

Δ. GAUSS Data Archives

Τα GAUSS Data Archive (GDA) είναι πολύ ισχυρά και ευέλικτα, δίνοντας στον χρήστη μεγαλύτερο έλεγχο ως προς τον τρόπο αποθήκευσης των δεδομένων. Δεν υπάρχει περιορισμός στον αριθμό των μεταβλητών που μπορούν να αποθηκευτούν σε ένα GDA, οπότε ο μόνος περιορισμός είναι ο διαθέσιμος χώρος αποθήκευσης στον δίσκο. Επίσης μπορούν να αποθηκεύσουν οποιοδήποτε τύπο δεδομένων. Ο χρήστης μπορεί να γράψει μήτρες, arrays, strings, string arrays, sparse μήτρες και δομές σε ένα GDA και αυτό θα τηρεί αρχείο του τύπου, μεγέθους και θέσης κάθε μεταβλητής που περιέχεται σε αυτό.

Ε. Αρχεία μητρών

Αυτά τα αρχεία δημιουργούνται με την εντολή *save*. Η εντολή αυτή κρατά στην μνήμη μια μήτρα, της δίνει έναν τίτλο που περιέχει πληροφορίες για τον αριθμό των γραμμών και στηλών που έχει η μήτρα και την αποθηκεύει στον δίσκο. Οι αριθμοί αποθηκεύονται ως διπλής ακριβείας όπως ακριβώς και στις αποθηκευμένες μήτρες. Αυτά τα αρχεία έχουν την κατάληξη .fmt. Τα αρχεία αυτά δεν μπορούν να είναι μεγαλύτερα από μια μήτρα. Ονόματα μεταβλητών δεν μπορούν να συνδεθούν με τα αρχεία αυτά.

Τα αρχεία μητρών μπορούν να φορτωθούν στην μνήμη με τις εντολές *load* (με την οποία φορτώνεται ολόκληρη η μήτρα) ή *loadm*. Μπορούν να ανοιχτούν με την εντολή *open* και να διαβαστούν με την εντολή *readr*. Όμως είναι διαθέσιμα μόνο για ανάγνωση και όχι για συνένωση ή ανανέωση.

Περαιτέρω με την εντολή *import*, η οποία είναι ένα multi-line GDT (gauss data tool) αρχεία δεδομένων μετατρέπονται σε data sets του προγράμματος. Στην τρέχουσα εκδοχή του υποστηρίζει μετατροπή τόσο από αρχεία ASCII όσο και Excel.

2. Maple[46]

Το πρόγραμμα υποστηρίζει την εισαγωγή και εξαγωγή αρχείων στις ακόλουθες μορφές:

- Σύνδεση με το CAD
  - Ανάκτηση παραμέτρων από ένα σχέδιο CAD και αποστολή πίσω νέων τιμών για να ενσωματωθούν αυτομάτως στο σχέδιο.
  - Συμπεριλαμβάνει το διαδραστικό CAD Link Assistant.
- Δημιουργία κώδικα
  - Δημιουργία κώδικα σε Visual Basic, MATLAB®, Java, C και Fortran.
  - Υπερκαλύπτει ή εμπλουτίζει τις μεταφράσεις κώδικα, όπως αυτές προσδιορίζονται στον υπάρχοντα ορισμό του ή ορίζει έναν καθόλα νέο ορισμό της γλώσσας.

[46] Product Specifications διαθέσιμο σε:
www.maplesoft.com/view.aspx?SF.../Maple12_Product_Spec.pdf

- Σύνδεση με το MATLAB
  - Μεταφραστής κώδικα MATLAB σε Maple για εντολές και .m αρχεία.
  - Απευθύνεται στο MATLAB για να υπολογίσει και να ανακτήσει τα αποτελέσματα των εκφράσεών του.
- Εύκολη εισαγωγή και εξαγωγή αρχείων Microsoft® Excel®.
  - Προγραμματιστική και διαδραστική χρήση μέσω του Data Import Assistant.
  - Υποστήριξη μερικής εισαγωγής αρχείων.
- Σύνδεση με το Microsoft Excel 2000, Excel XP, Excel 2003 και Excel 2007 για Windows.
  - Πρόσβαση στο Maple kernel μέσα από το Excel.
  - Δυνατότητα αντιγραφής και επικόλλησης μεταξύ Maple και Excel.
  - Λειτουργία του Wizard μέσω της δημιουργίας μιας συνάρτησης Maple.
- Σύνδεση με βάση δεδομένων.
  - Εργαλεία για την αναζήτηση, ανανέωση και δημιουργία βάσεων δεδομένων μέσα από το Maple.
- Εργαλεία για την μετατροπή και την μεταγλώττιση του Mathematica® Notebook.
- Πρόσβαση στους Maple αλγόριθμους και δομές δεδομένων σε μεταγλωττισμένα προγράμματα C, Java και Visual Basic μέσω του OpenMaple™.
- Εξωτερική κλήση σε Java, C και Fortran.
- Παρουσίαση και υποστήριξη περιεχομένου σε MathML 2.0
- Εισαγωγή και εξαγωγή αρχείων XML.
- Σύνδεση με TCP/IP.
- Εξαγωγή φύλλων εργασίας σε HTML, XML, MathML, LaTeX και RTF.
- Εξαγωγή γραφικών σε BMP, DXF, EPS, GIF, HPGL, JPEG, PCX, POV, TEK και WMF.
- Εισαγωγή, επεξεργασία και εξαγωγή δεδομένων από αρχεία WAV, JPEG και TIFF.
- Εισαγωγή δεδομένων από ASCII, CSV, Matrix Market, MATLAB κ.α.

- Απευθείας σύνδεση με το Maple Application Center™, Teacher Resource Center, Student Help Center κ.α.

3. Mathematica[47]

Το πρόγραμμα υποστηρίζει την εισαγωγή και εξαγωγή αρχείων στις ακόλουθες περιπτώσεις και με τις παρακάτω μορφές:

Basic Formats	Table
	List
	String
	Text
	Binary"
Raster Image Formats	GIF
	JPEG
	TIFF
	PNG
	BMP
	PICT
	WMF
	SCT"
Vector Graphics Formats	SVG
	WMF
	EPS
	PDF
	DXF
3D Geometry & Modeling Formats	PLY
	OFF
	OBJ
	X3D
	Maya
	POV

[47] Mathematica,Documentation Center, Data Manipulation, Importing & Exporting διαθέσιμο σε: http://reference.wolfram.com/mathematica/guide/ImportingAndExporting.html

	LWO
	STL
Audio Formats	WAV
	AIFF
	MIDI
	SND
	FLAC
	Wave64
Multimedia Formats	AVI
	FLV
	QuickTime
	SWF
Tabular & Spreadsheet Formats	Table
	CSV
	TSV
	XLS
	ODS
	SXC
Database Formats	MDB
	DBF
	DIF
	XLS
Scientific & Medical Data Formats	HDF
	CDF
	FITS
	DICOM
	EDF
Chemical & Biomolecular Formats	MOL
	SDF
	SMILES
	PDB
	GenBank

	FASTA
Geospatial Formats	SHP
	USGSDEM
	GTOPO30
	SDTS
	TIGER
	SP3
Numerical Data Formats	XPORT
	MAT
	MTX
	HarwellBoeing
	MPS
Mathematical Data Formats	Graph6
	Sparse6
Document Formats	PDF
	HTML
	NB
	RTF
	TeX
	Text
Web Formats	HTML
	GIF
	JPEG
	SWF
	XHTML
	X3D
Print Formats	PDF
	EPS
	TeX
	SCT
	ACO
XML Formats	XML

	XHTML
	MathML
	SVG
	X3D
	ODS
Systems & Utility Formats	Directory
	ApacheLog
	MBOX
	VCF
	RSS
Compression & Archive Formats	Base64
	BZIP2
	GZIP
	TAR
	UUE
	ZIP
	WDX
Binary Formats	Bit
	Byte
	Integer16
	Real32
	TerminatedString

4. Matlab[48]

Ο παρακάτω πίνακας παρουσιάζει τους τύπους αρχείων που ο χρήστης μπορεί να γράψει και να διαβάσει στο πρόγραμμα καθώς και τις συναρτήσεις που υποστηρίζουν καθέναν από τους τύπους αυτούς.

File Format	File Content	Extension	Functions
MATLAB	Saved MATLAB workspace	.mat	load, save

[48] http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/ref/fileformats.html

File Format	File Content	Extension	Functions
formatted			
Text	Text	any	textscan
	Text	any	textread
	Delimited text	any	dlmread, dlmwrite
	Comma-separated numbers	.csv	csvread, csvwrite
Extended Markup Language	XML-formatted text	.xml	xmlread, xmlwrite
Audio	NeXT/SUN sound	.au	auread, auwrite
	Microsoft WAVE sound	.wav	wavread, wavwrite
Video	Audio/video	.avi	aviread, avifile
		.avi .mpg, .mpeg, .wmv, .asf, .asx, .mov, .mp4, .m4v, .3gp, .3g2, .dv	mmreader
Scientific data	Data in Common Data Format	.cdf	cdfread, cdfwrite
	Flexible Image Transport System data	.fits	fitsread
	Data in Hierarchical Data Format (HDF4)	.hdf	hdf
	Data in Hierarchical Data Format (HDF5)	.h5	hdf5
Spreadsheet	Microsoft Excel® worksheet	.xls	xlsread, xlswrite
	Lotus 123 worksheet	.wk1	wk1read, wk1write

File Format	File Content	Extension	Functions
Graphics	TIFF image	.tiff	imread, imwrite
	PNG image	.png	same
	HDF image	.hdf	same
	BMP image	.bmp	same
	JPEG image	.jpeg	same
	GIF image	.gif	same
	PCX image	.pcx	same
	XWD image	.xwd	same
	Cursor image	.cur	imread
	Icon image	.ico	imread

5. Scilab[49]

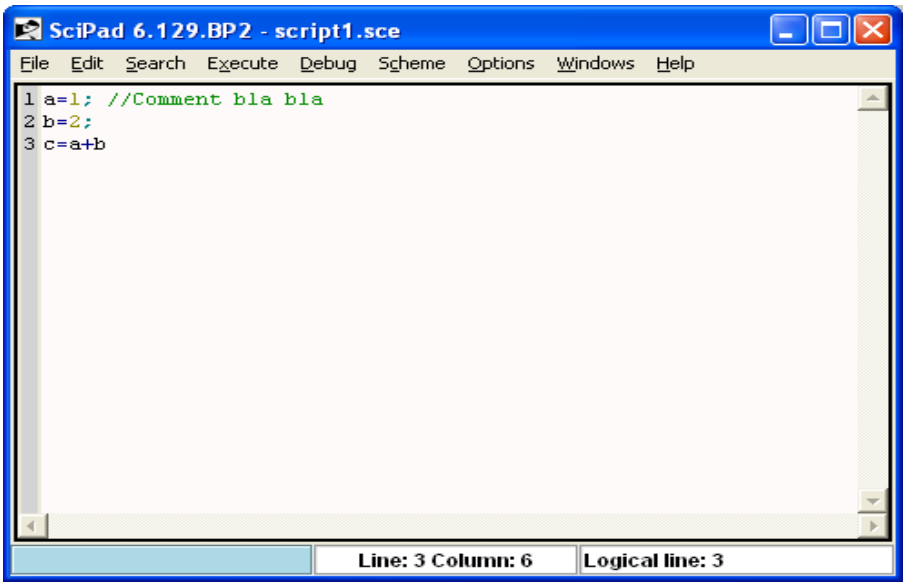
Α. Δουλεύοντας με αρχεία.

- Το αρχείο script είναι ένα αρχείο με όνομα της μορφής *.sce το οποίο περιέχει εντολές του προγράμματος. Ο χρήστης μπορεί να επεξεργαστεί το αρχείο χρησιμοποιώντας τον ενσωματωμένο στο πρόγραμμα editor. Ακόμα και για μικρές εφαρμογές είναι προτιμότερη η χρήση των scripts καθώς ο χρήστης μπορεί α) να έχει όλες τις εργασίες του αποθηκευμένες σε αρχεία, γεγονός που διευκολύνει την τεκμηρίωση και β) να εκτελέσει όλες τις εντολές του ακόμα και μετά από κάποιες τροποποιήσεις τους. Για να δημιουργήσει και να τρέξει ο χρήστης ένα απλό script ακολουθεί τα εξής βήματα. Ας σημειωθεί ότι το να τρέξει ένα τέτοιο αρχείο είναι σαν να εκτελεί όλες τις εντολές που υπάρχουν στην γραμμή εντολών μία προς μία.

– ανοίγει τον editor (όπως στην εικόνα που ακολουθεί), είτε επιλέγοντάς τον από το μενού είτε εκτελώντας την εντολή *scipad*, και εισάγει τις εντολές που τον ενδιαφέρουν. Οι δύο γραμμές (//) χρησιμεύουν για την είσοδο δεδομένων στο script. Υπάρχει επίσης η δυνατότητα να

[49]Haugen, Finn: Master Scilab! 2008 διαθέσιμο σε: http://home.hit.no/~finnh/scilab_scicos/scilab/index.htm#scripts

ανοίξει περισσότερα από ένα scripts στο ίδιο παράθυρο εργασίας από το μενού File / New.



Ο χρήστης αποθηκεύει το script με το όνομα που επιθυμεί (έστω script1.sce) στον φάκελο που επιλέγει

υπάρχουν δύο τρόποι για να τρέξει το αρχείο που δημιούργησε:

- από το Scipad μενού Execute / Load into Scilab, οπότε το αποτέλεσμα θα φανεί στο παράθυρο εντολών

- εκτελώντας την εντολή exec (όνομα αρχείου) script1.sce στην γραμμή εντολών. Το πρόγραμμα απαντά error!

Αυτό συνέβη διότι το script1.sce (το αρχείο που ο χρήστης θέλει να τρέξει) δεν βρίσκεται στο ενεργό directory προγράμματος, δηλαδή στο χώρο όπου θα ψάξει πρώτα το πρόγραμμα για να βρει το αρχείο που του ζητά ο χρήστης να εκτελέσει με την εντολή exec. Εάν ο χρήστης θέλει να δει ποιο είναι το ενεργό (current) directory, μπορεί να ανατρέξει στο μενού File / Get Current Directory. Αν προκύψει ότι είναι διαφορετικό από εκείνο στο οποίο αυτός έχει αποθηκεύσει το script που θέλει να τρέξει, τότε θα πρέπει να θέσει εκείνο ως ενεργό (current) directory και να ξαναδώσει την εντολή εκτέλεσης, ως εξής:

- ανατρέχει στο μενού File / Change Directory, οπότε ανοίγει ένα παράθυρο από το οποίο επιλέγει το directory που θέλει να ορίσει ως ενεργό (current)

- δίνει την εντολή exec script1.sce στην γραμμή εντολών, οπότε το script θα τρέξει χωρίς προβλήματα.

Τα αρχεία Function είναι της μορφής *.sci και μοιάζουν με τα αρχεία script, εκτός από το γεγονός ότι περιέχουν έναν ή περισσότερους ορισμούς συναρτήσεων. Είναι πιο σύνηθες ο χρήστης να βάζει συναρτήσεις σε αρχεία παρά απευθείας στο πρόγραμμα, διότι α) αποθηκεύονται ώστε να χρησιμοποιηθούν πάλι στην συνέχεια και β) είναι εύκολο να διορθώσει ή να τροποποιήσει μια συνάρτηση μέσω της επεξεργασίας του αρχείου. Ομοίως όπως τα αρχεία script, φορτώνονται με την εντολή exec ή από το μενού Execute

Β. Input

Η εντολή *read* χρησιμοποιείται για να διαβαστούν (εισαχθούν) δεδομένα από εξωτερικά αρχεία σε μήτρες του προγράμματος. Η εντολή είναι της μορφής:

x = read(filename, nrows, ncols)

Η εντολή υποθέτει ότι τα δεδομένα στο αρχείο που διαβάζει είναι οργανωμένα σε στήλες. Παρόλα αυτά ο αριθμός των σειρών, nrows και στηλών, ncols, στη διατύπωσή της δεν χρειάζεται να ταιριάζουν με τη διάταξη των δεδομένων του αρχείου.

Η εντολή *save* (“…”) σώζει το περιβάλλον εργασίας στο αρχείο work.dat. Το αρχείο που δημιουργείται είναι ένα binary αρχείο που μπορεί να χρησιμοποιηθεί μόνο από το πρόγραμμα.

Γ. Output

Με την εντολή *write* αποθηκεύονται δεδομένα του προγράμματος σε διάφορα αρχεία. Κάθε φορά μόνο μια μήτρα μπορεί να γραφεί σε ένα αρχείο.

Η εντολή *Load(“…”),* φορτώνει τα δεδομένα του αρχείου.

Για να έχει ο χρήστης μια λίστα της τωρινής του session μπορεί να χρησιμοποιήσει την εντολή/λειτουργία *diary*. Η διατύπωσή της είναι

diary(output_filename)

όπου το όνομα αρχείου γράφεται μέσα σε εισαγωγικά.

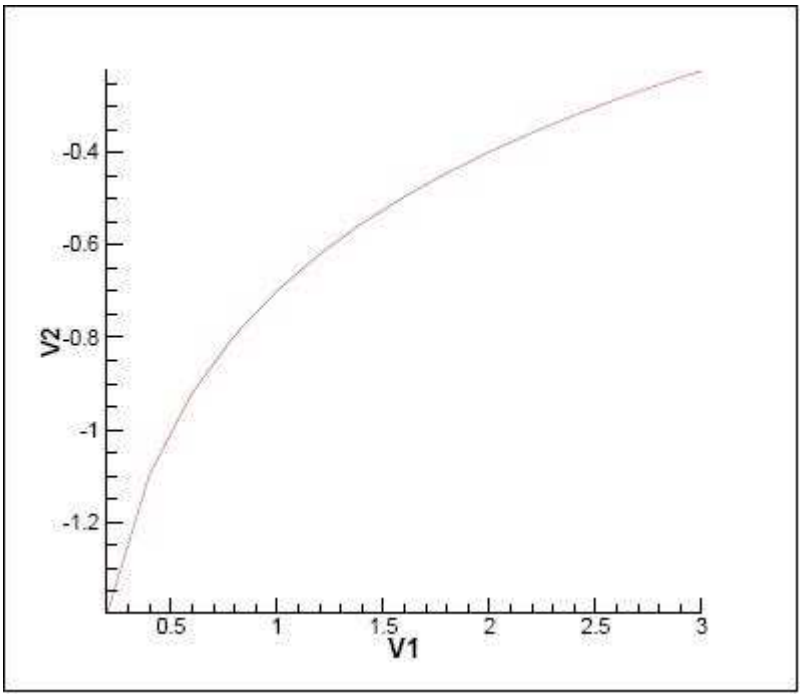
Στην συνέχεια, με την χρήση notepad ή text editor, μπορεί ο χρήστης να ανοίξει το αρχείο session.txt και να δει τα περιεχόμενα του αρχείου αυτού. Να σημειωθεί ότι το worksheet του προγράμματος δεν επιτρέπει την αποκοπή και επικόλληση, γι' αυτό και είναι απαραίτητη η χρήση της εντολή diary, αφού είναι ο μοναδικός τρόπος να αντιγράψει ο χρήστης τα εξαγόμενα δεδομένα σε text αρχείο.

II. ΓΡΑΦΙΚΑ

1. Gauss

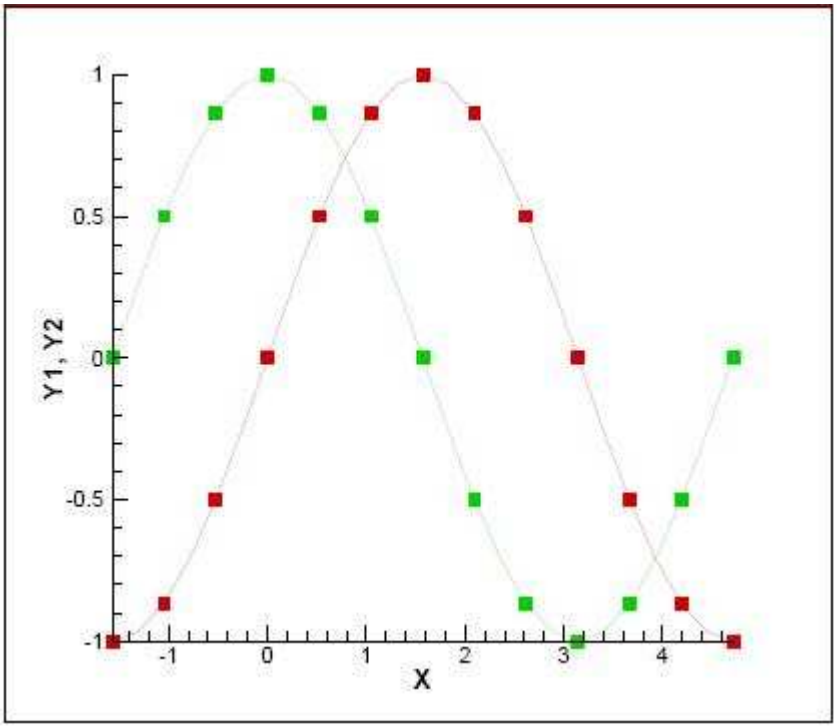
Το πρόγραμμα περιέχει τέσσερεις βασικές μορφές συναρτήσεων για την δημιουργία γραφικών παραστάσεων[50]. Οι απλές και εμπλουτισμένες μορφές τους είναι οι ακόλουθες:

α. η συνάρτηση XY Line Plot (απλή) και XY Line Plot και Bar Plot (εμπλουτισμένες), οι οποίες χρησιμοποιούν καρτερσιανές συντεταγμένες

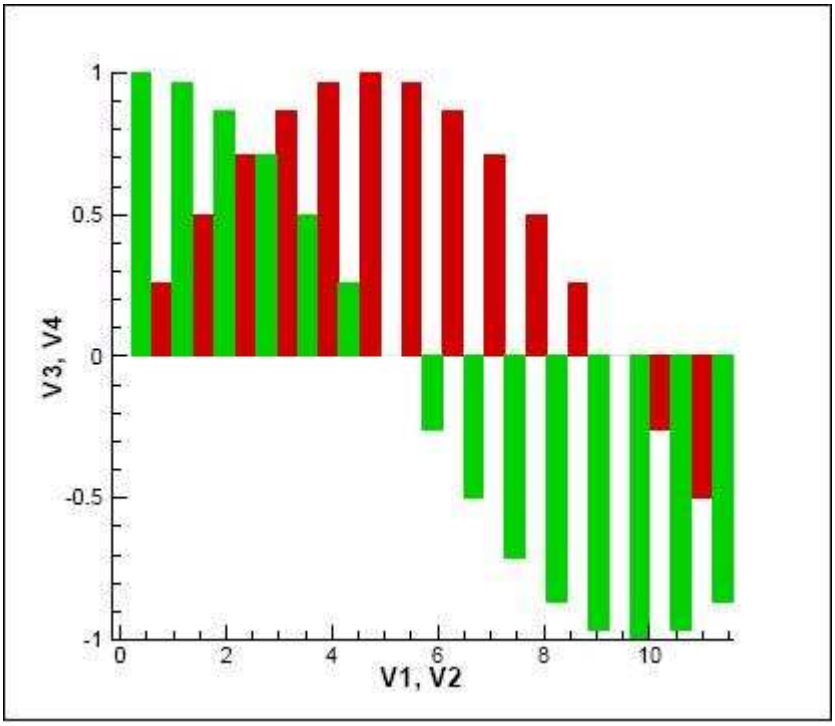


XY Line Plot

[50] Aptech Systems Inc: GAUSSplot, Professional Graphics Cookbook, Powered by Tecplot, 2005, σελ. 5 επ.

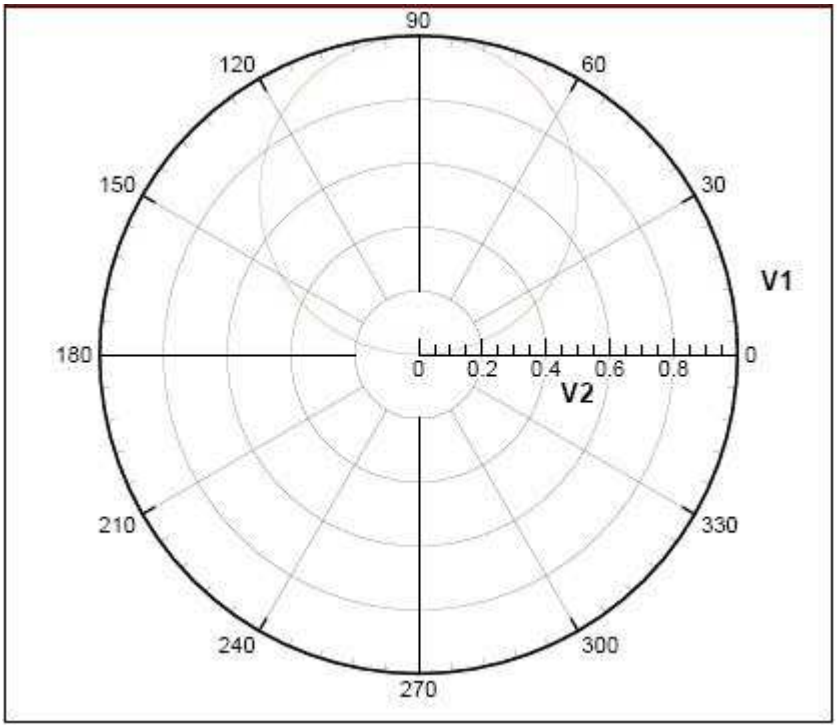


XY Line Plot



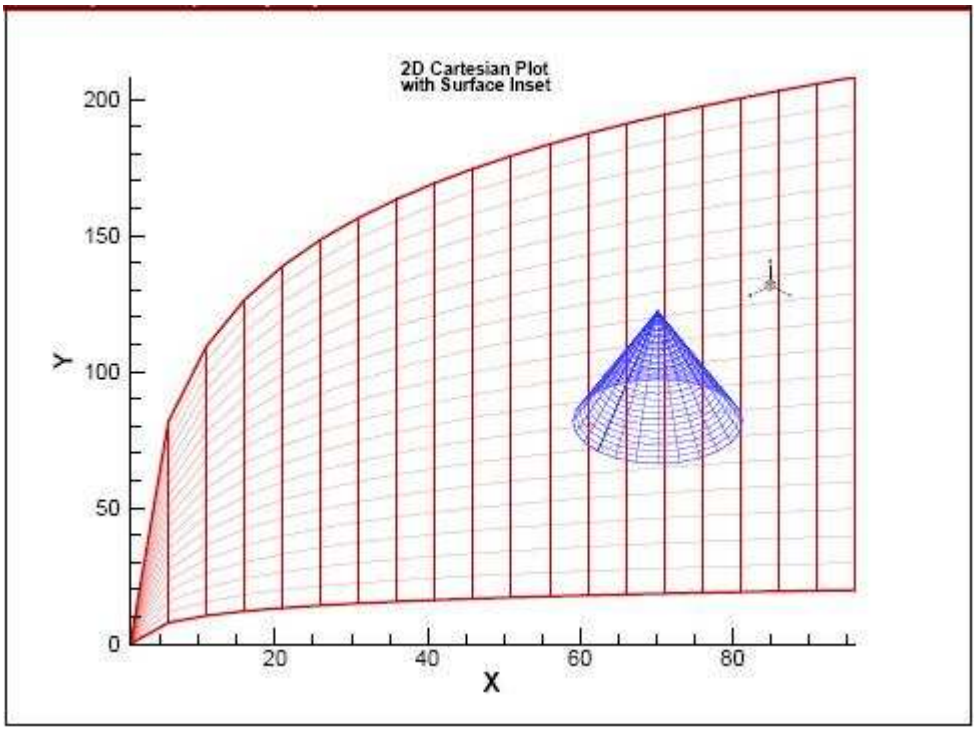
Barplot

β. η συνάρτηση Polar Line Plot (απλή) και Polar Line Plot (εμπλουτισμένη), οι οποίες χρησιμοποιούν πολικές συντεταγμένες

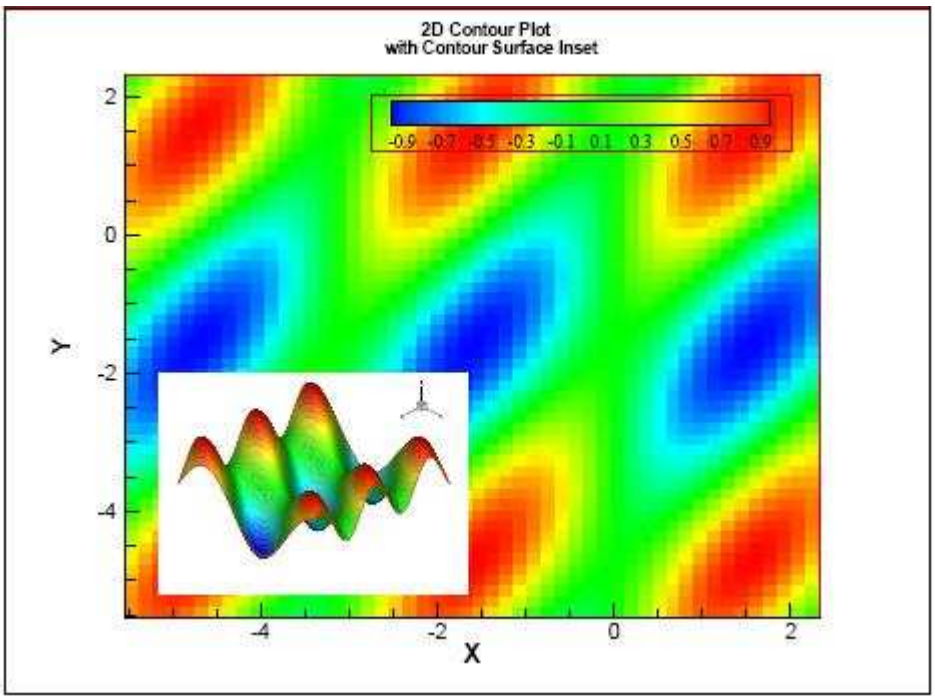


Polarlineplot

γ. η συνάρτηση 2D Cartesian Plot (απλή) και 2D Cartesian Plot και 2D Contour Plot (εμπλουτισμένες)

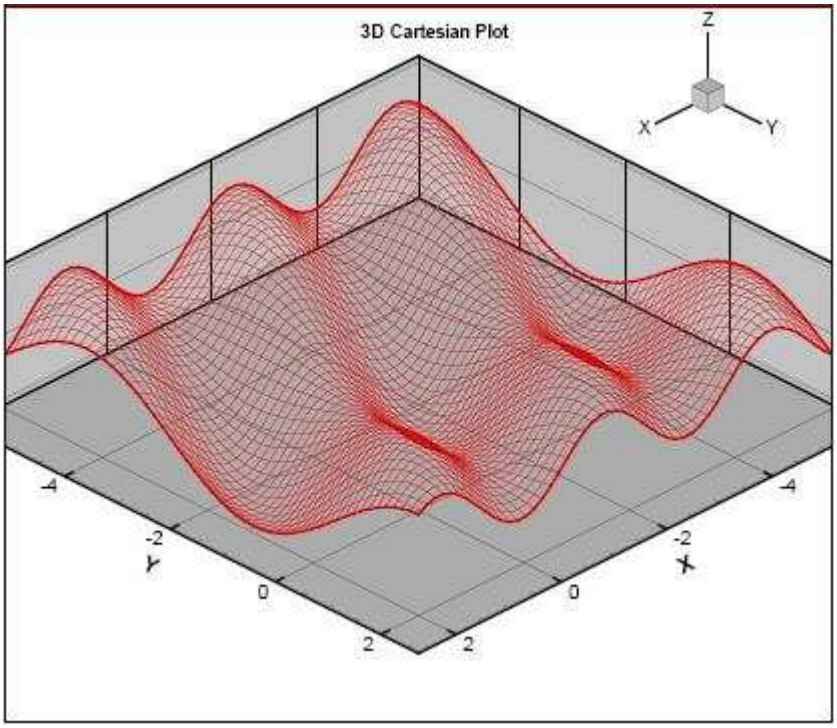


2D Cartesianplot

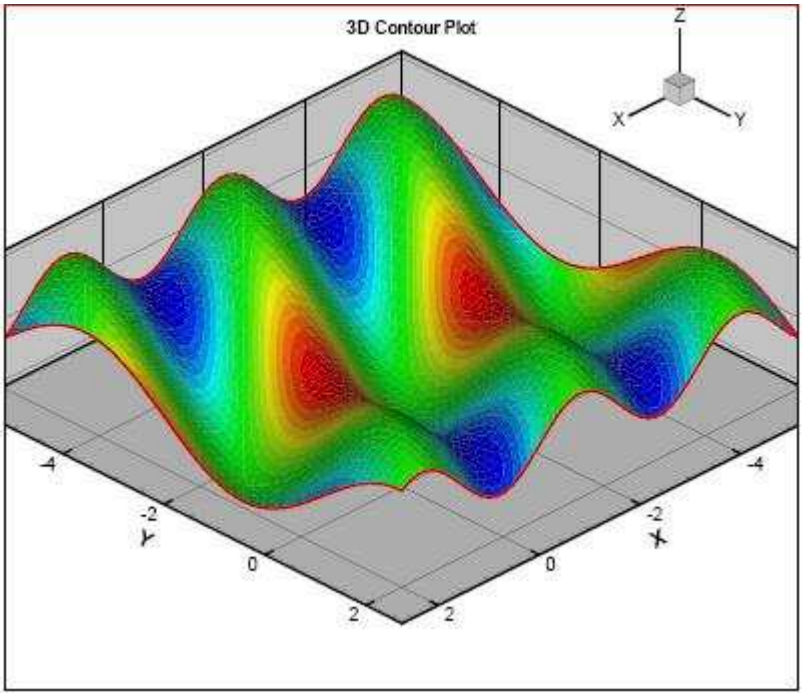


2D Contourplot

δ. η συνάρτηση 3D Cartesian Plot (απλή) και 3D Cartesian Plot και 3D Contour Plot (εμπλουτισμένες).



3D Cartesianplot



3D Contourplot

Για να δημιουργήσει ο χρήστης μια γραφική παράσταση πρέπει να ακολουθήσει τα εξής βήματα:

Κατ' αρχάς η δημιουργία πραγματοποιείται σε μια επιφάνεια εργασίας (workspace), όπου υπάρχει ένα πλαίσιο (frame). Κάθε γραφική παράσταση μπορεί να περιέχει περισσότερα του ενός πλαίσια, καθένα από τα οποία αποτελείται από δύο άξονες. Ο χρήστης έχει την δυνατότητα να αποκρύπτει την εμφάνιση των αξόνων.

Βήμα 1ο: ενεργοποίηση της βιβλιοθήκης (pgraph library)

Βήμα 2ο: τα δεδομένα πρέπει να βρίσκονται σε μια μήτρα

Βήμα 3ο: το πρόγραμμα έχει εξ ορισμού κάποιες ρυθμίσεις για τις επιμέρους παραμέτρους της γραφικής παράστασης, τις οποίες ο χρήστης μπορεί να διαμορφώσει κατά βούληση μέσω των διαφόρων γραφιστικών μεθόδων και των κοινών-γενικών μεταβλητών. Οι μεταβλητές που αρχίζουν με το πρόθεμα “_p” είναι οι κοινές-γενικές μεταβλητές που χρησιμοποιούνται από graphics routines.

Βήμα 4ο: Οι graphics routines επεξεργάζεται τα δεδομένα που εισάγει ο χρήστης και τις κοινές μεταβλητές που έχουν οριστεί προηγουμένως.

Αντί των βημάτων 2 και 3 μπορεί ο χρήστης να χρησιμοποιήσει την εντολή *gpMakePlotTypePlot.*

Οι επιλογές του χρήστη είναι πολλές: μπορεί να χωρίσει την επιφάνεια εργασίας σε περισσότερα από ένα παράθυρα, τα οποία είτε θα αλληλοκαλύπτονται είτε θα στοιβάζονται το ένα πάνω στο άλλο (εντολή *Windows*), μπορεί διαλέξει μεταξύ τεσσάρων εναλλακτικών φόντων (εντολή *Fonts*), μπορεί να ονοματίσει τους άξονες (εντολή *XLabel*, *YLabel*) αλλά και την ίδια την γραφική παράσταση (εντολή *Title*).

2. Maple[51]

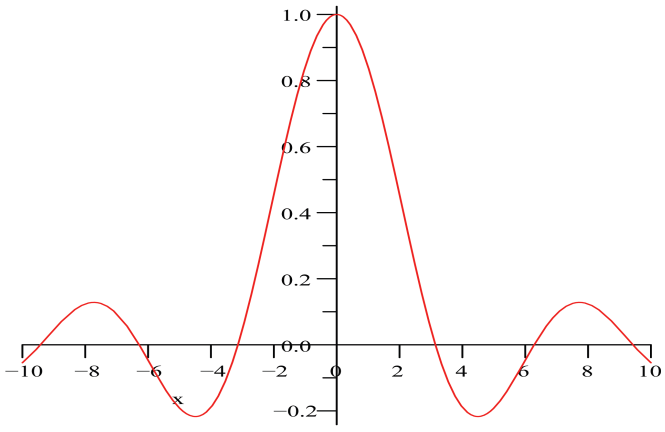
Χαρακτηριστικά. Το πρόγραμμα περιέχει ένα ευρύ φάσμα πακέτων και διαδικασιών για την δημιουργία διδιάστατων και τρισδιάστατων γραφικών παραστάσεων μέσω: α. των συναρτήσεων plot και plot3d, β. των πακέτων plots και γ. των πακέτων plottools.

Α. *Δημιουργία γραφικού στο επίπεδο.*

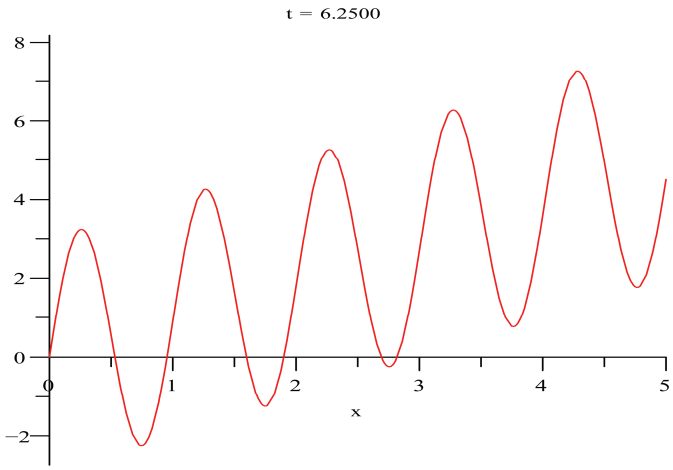
Η βασική συνάρτηση για την δημιουργία διδιάστατων γραφικών είναι η *plot*.

plot(plotexpression, x=a..b, ...)

plot3d(plotexpression, x=a..b, y=a..b, ...)



[51] Maplesoft: Maple User Manual, 2008, κεφ. 5, σελ. 189



Παρέχεται η δυνατότητα

α. γραφικές παραστάσεις: ύπαρξη περισσοτέρων στο ίδιο σύστημα αξόνων, απεικόνιση σημείων (εντολή *listplot* και *plot*), συνδυασμός επιμέρους γραφικών παραστάσεων (εντολή *display* από το πακέτο plots), καθορισμός χρώματος (εντολή *color)*, μορφής (σημείο, συνεχόμενη γραμμή, σύμβολο) (εντολή *style*) και μεγέθους συμβόλων (εντολή *symbolize*), ορισμός τίτλου (εντολή *title)*, κειμένου (εντολή *text*), υποσημειώσεων (εντολή *legend*), απεικόνιση πεπλεγμένης συνάρτησης (εντολή implicitplot), συνάρτησης σε πολικές συντεταγμένες (εντολή polarplot) και σχημάτων στο επίπεδο (εντολή display από το πακέτο plottools), κίνηση (εντολή *animate*) και

β. άξονες: καθορισμός του τύπου τους (εντολή *axes*) και του τίτλου τους (εντολή *labels*), και με μια πληθώρα επιμέρους εντολών για την προσαρμογή κάθε γραφικής παράστασης στις ανάγκες του εκάστοτε χρήστη. Προς την κατεύθυνση αυτή, το πρόγραμμα παρέχει επίσης την δυνατότητα χρησιμοποίησης ενός οδηγού γραφικών παραστάσεων (εντολή *interactive* από το πακέτο plots).

Β. Δημιουργία γραφικού στο χώρο.

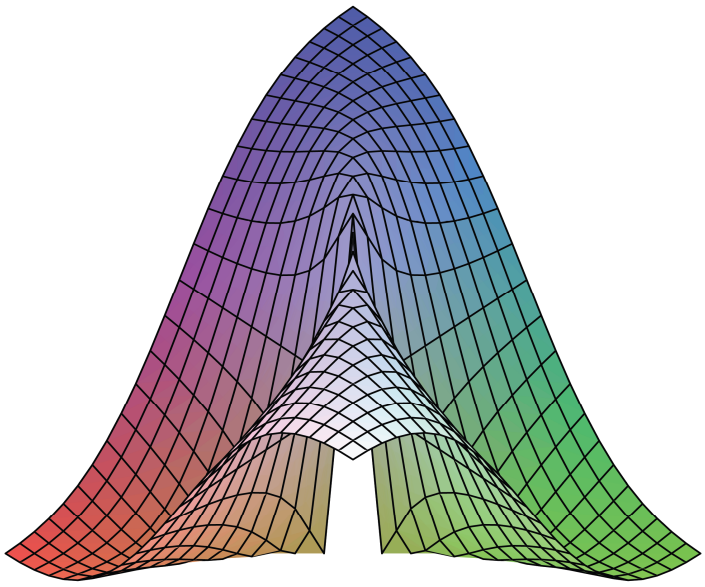
Η βασική συνάρτηση για την δημιουργία τρισδιάστατων γραφικών είναι η plot3d, η οποία προσφέρεται με τις ίδιες παραμέτρους όπως και η plot.

plot3d(expr, x=a..b, y=c..d, opts)

plot3d(f, a..b, c..d, opts)

plot3d([exprf, exprg, exprh], s=a..b, t=c..d, opts)

plot3d([f, g, h], a..b, c..d, opts)



3. Mathematica[52]

Χαρακτηριστικά. Η βασική εντολή για την γραφική παράσταση μιας συνάρτηση είναι η *Plot*, η οποία εμφανίζει πολλές ομοιότητες με την εντολή Table, του ίδιου προγράμματος καθώς η λειτουργία αυτή μοιάζει κατά πολύ με την δημιουργία ενός πίνακα των τιμών μιας συνάρτησης. Η εντολή αυτή έχει δύο μεταβλητές: η πρώτη είναι η συνάρτηση που θα απεικονιστεί γραφικά – και αντιστοιχεί στον οριζόντιο άξονα - και η δεύτερη αφορά στο όνομα της συνάρτησης και στην περιοχή των τιμών που η μεταβλητή μπορεί να λάβει (iterator) – και αντιστοιχεί στον κάθετο άξονα. Το πρόγραμμα έχει εξ ορισμού κάποιες ρυθμίσεις όμως μπορεί ο χρήστης να αξιοποιήσει τις παραμέτρους της βασικής εντολής και να πετύχει το άριστο αποτέλεσμα στην γραφική του παράσταση.

Το πρόγραμμα εξ ορισμού επιλέγει μόνο του σε ποιο σημείο θα τέμνονται οι άξονες και συνήθως αυτό δεν συμπίπτει με την αρχή τους (εντολή *AxesOrigin→{0,0]*). Εάν ο χρήστης το επιθυμεί, του παρέχεται η δυνατότητα να τέμνονται στην αρχή τους. Επίσης εξ ορισμού το πρόγραμμα

[52] http://reference.wolfram.com/mathematica/guide/VisualizationAndGraphicsOverview.html

επιλέγει να προβάλλει στον χρήστη μόνο ένα τμήμα της γραφικής παράστασης, δηλαδή εκείνο που εμφανίζει κάποιο ενδιαφέρον. Και πάλι βέβαια εάν ο χρήστης το επιθυμεί, μπορεί να δει ολόκληρη την γραφική παράσταση (εντολή *PlotRange→All*). Ο χρήστης μπορεί επίσης:

α. να ρυθμίσει την μορφή της γραφικής παράστασης, όπως το χρώμα (εντολή *PlotStyle*), τον φόντο της (εντολή *DefaultFont→{...}*), τον τίτλο της (εντολή *PlotLabel→"..."*), ένα επεξηγηματικό κείμενο (έχοντας κατεβάσει το πακέτο *Graphics 'Legend'* και χρησιμοποιώντας την αντίστοιχη εντολή)

β. να απεικονίσει μια τμηματική συνάρτηση

γ. να απεικονίσει μια πεπλεγμένη συνάρτηση (έχοντας κατεβάσει το πακέτο *Graphics 'ImplicitPlot'* και χρησιμοποιώντας την αντίστοιχη εντολή)

δ. να εμφανίσει περισσότερες γραφικές παραστάσεις εντός των ίδιων αξόνων (στην εντολή *Plot* εισάγεται μια λίστα των συναρτήσεων ως πρώτη μεταβλητή σε συνδυασμό με την εντολή *Evaluate*)

ε. να συνδυάσει δύο ξεχωριστές γραφικές παραστάσεις (εντολή *Show*)

στ. να δώσει κίνηση σε μία ή περισσότερες γραφικές παραστάσεις (εντολή *Table* σε συνδυασμό με την παράμετρο *PlotRange*)

Αναφορικά με τους άξονες, έχει την δυνατότητα:

α. να πετύχει την ίδια διαβάθμιση και στους δύο (εντολή *AspectRatio→Automatic)*

β. να τους απομακρύνει (εντολή *Axes→False*)

γ. να προσθέσει πλαίσιο (εντολή *Frame→True*)

δ. να τους δώσει τίτλο (εντολή *AxesLabel→{...}*)

Με την βοήθεια του ποντικιού μπορεί ο χρήστης να βρει τις συντεταγμένες οποιουδήποτε σημείου της γραφικής παράστασης καθώς και να μεγεθύνει ή να σμικρύνει μια γραφική παράσταση.

4. Matlab[53]

Η εύκολη απεικόνιση και δημιουργία γραφικών παραστάσεων αποτελεί μέρος της υπολογιστικής διαδικασίας του προγράμματος. Περιέχει συναρτήσεις για διδιάστατα και τρισδιάστατα γραφικά ενώ εμμέσως μπορεί να δημιουργηθούν και τετραδιάστατα γραφικά.

[53] http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/creating_plots/bqrw9tj.html

Χαρακτηριστικά. Δεδομένου ότι το πρόγραμμα είναι σχεδιασμένο να χρησιμοποιεί πίνακες και όχι συναρτήσεις, όλες οι εντολές για γραφήματα δέχονται πίνακες ως ορίσματά τους. Επίσης το πρόγραμμα παρέχει την δυνατότητα το διάγραμμα να εμφανίζεται σε ένα παράθυρο, όπου οι διαδοχικές εντολές θα προστίθενται στο ίδιο διάγραμμα μέχρι να δημιουργηθεί αυτό που ανταποκρίνεται στις απαιτήσεις μας, εκτός εάν εξ αρχής επιλέξουμε διαφορετικά.

Α. Δημιουργία γραφικού στο επίπεδο.

Γενικά. Για την απλούστερη μορφή διδιάστατων γραφικών παραστάσεων χρησιμοποιούμε την συνάρτηση plot. Μπορούμε να αφήσουμε το πρόγραμμα να εμφανίσει την γραφική παράσταση όπως είναι ορισμένο να την εκτελέσει, οπότε έστω ότι x και y είναι δύο διανύσματα ίδιου μήκους, τότε η συνάρτηση plot (x, y) ενώνει τα διαδοχικά σημεία (xi, yi) με συνεχή ευθύγραμμα τμήματα. Αλλιώς αν χρησιμοποιήσουμε σημάδια (markers) αντί των σημείων (xi, yi), να προσδιορίσουμε τη μορφή της γραμμής (line style) που θα ενώνει τα σημάδια και τις γραμμές από μια γκάμα χρωμάτων που μας προσφέρεται. Το πρόγραμμα εκλαμβάνει τις γραμμές ως αντικείμενα με δικές τους ιδιότητες (το όνομα καθεμιάς αναπαρίσταται είτε με κεφαλαία είτε με μικρά γράμματα, χωρίς την δυνατότητα εναλλαγής τους), καθεμιά από τις οποίες έχει μια τιμή, που δεν είναι αναγκαστικά ένας αριθμός. Υπάρχει λοιπόν η λειτουργία εκείνη του προγράμματος που μας επιτρέπει να δούμε ποιες είναι οι ιδιότητες κάθε αντικειμένου, ποιες τιμές μπορούν να πάρουν και πώς αλλάζουν αυτές οι τιμές. Η λειτουργία αυτή είναι τμήμα του αντικειμενοστραφούς προγραμματισμού του προγράμματος. Εδώ αξίζει να σημειωθεί ότι εκτός από την γραμμή άλλα αντικείμενα του προγράμματος είναι το κείμενο (text) και οι άξονες (axes), και όλα μαζί απεικονίζονται στο παράθυρο γραφικών, την ρίζα (root), η οποία είναι και αυτή ένα αντικείμενο. Επόμενα στην ιεραρχία μετά την ρίζα είναι τα αντικείμενα unicontrols, aces, unimenus. Όταν τροποποιούνται οι ιδιότητες ενός αντικειμένου, αλλάζουν ταυτόχρονα και οι ιδιότητες των ιεραρχικά κατώτερων αντικειμένων.

Α.1. Παράθυρο γραφικών. Η σχηματιζόμενη γραφική παράσταση θα εμφανιστεί σε ένα νέο παράθυρο, που θα ανοίξει αυτόματα, το παράθυρο

γραφικών (graphics window). Αν χρησιμοποιήσουμε την συνάρτηση plot για να κάνουμε και δεύτερη γραφική παράσταση, τότε η προηγούμενη θα διαγραφεί. Το πρόγραμμα βέβαια μας δίνει την δυνατότητα να εμφανίσουμε περισσότερα παράθυρα γραφικών, αξιοποιώντας την εντολή *figure(k)*. Επιπλέον μπορούμε να εμφανίσουμε περισσότερες από μία γραφικές παραστάσεις στο ίδιο παράθυρο γραφικών και μάλιστα με τρεις εναλλακτικούς τρόπους: είτε ως γραφική παράσταση ζευγαριών διανυσμάτων με την χρήση διαφορετικών χρωματισμών, είτε ως γραφική παράσταση των αντίστοιχων στηλών των εκάστοτε μητρών, είτε με την χρήση των εντολών hold on και hold off, όταν οι γραφικές παραστάσεις προκύπτουν μετά από υπολογισμούς. Στην περίπτωση αυτή, οι περισσότερες γραφικές παραστάσεις μπορούν να εμφανίζονται είτε σε ένα ενιαίο παράθυρο είτε με την εντολή *subplot*, να δημιουργούνται κελιά μέσα στο ενεργό παράθυρο είτε με την εντολή *axes*, να προκύπτει ένα νέο παράθυρο μέσα στο ενεργό.

Η εντολή *hold on* επιτρέπει να διατηρούνται οι προηγούμενες γραφικές παραστάσεις, όταν νέες κατευθύνονται στο ενεργό παράθυρο ενώ το αντίθετο αποτέλεσμα επέρχεται με την εντολή *hold off*.

Η εντολή *clf* διαγράφει μια παράσταση από το ενεργό παράθυρο.

Η εντολή *pause* επιφέρει κίνηση στην οθόνη. Η λειτουργία αυτή διακρίνεται σε πρωτόγονη (primitive) και βελτιωμένη (enhanced).

Η εντολή *print* αποθηκεύει και εκτυπώνει τις εικόνες των γραφικών παραστάσεων.

Η εντολή *close* κλείνει το ενεργό παράθυρο.

Α.2. Τίτλος και επεξηγήσεις. Η γραφική παράσταση που θα σχηματιστεί μπορεί να έχει τίτλο. Η εντολή είναι *title* και μπορεί να είναι οποιαδήποτε έκφραση αρκεί να εσωκλείεται σε αριστερές αποστρόφους. Κείμενο όμως μπορεί να υπάρξει και μέσα στην γραφική παράσταση, συνηθέστερα με την εντολή *text*. Όταν όμως υπάρχουν περισσότερες γραφικές παραστάσεις εντός των ίδιων αξόνων, προκειμένου να προσδιοριστεί επακριβώς το σημείο όπου θα εμφανιστεί το κείμενο, η εντολή grid on δημιουργεί ένα πλέγμα στην γραφική παράσταση, μέσω του οποίου τοποθετείται το κείμενο στο επιθυμητό σημείο και κατόπιν απομακρύνεται με την εντολή *grid off*. Σε

πιο σύνθετες περιπτώσεις υπάρχει η εντολή *gtext*, με την οποία τοποθετούμε το κείμενο στο επιθυμητό σημείο με την βοήθεια του ποντικιού. Εναλλακτικά, υπάρχει και η εντολή *legend*, με την οποία εμφανίζεται ένα πλαίσιο, εντός του οποίου γράφεται το κείμενο, και τοποθετείται είτε εντός είτε εκτός των αξόνων.

Α.3. Άξονες. Αναφορικά με τους άξονες της γραφικής παράστασης. Μπορούμε να τους δώσουμε επιγραφές, με τις εντολές *xlabel* (επιγραφή του άξονα των x) και *ylabel* (επιγραφή του άξονα των y). Μπορούμε να τους μορφοποιήσουμε με την εντολή *axis*, η οποία ορίζει μεταξύ άλλων παραμέτρων το μέγεθος της μονάδας τους και τα όριά τους. Ενδεικτικά:

η εντολή *axis equal* – θέτει ίσες μονάδες μέτρησης και στους δύο άξονες,

η εντολή *axis square* – εμφανίζει την γραφική παράσταση μέσα σε τετράγωνο.

η εντολή *box off* – εξαφανίζει το κουτί μέσα στο οποίο εμφανίζονται εξ αρχής οι άξονες.Το αντίθετο αποτέλεσμα επέρχεται με την εντολή box on.

η εντολή *axis auto* – επαναφέρει τις εξ ορισμού ρυθμίσεις.

η εντολή *axis off* – εμφανίζει την γραφική παράσταση χωρίς άξονες.Το αντίθετο αποτέλεσμα επέρχεται με την εντολή *axis on*.

Α.4. Ειδικές γραφικές παραστάσεις. Μπορούμε να τις επιτύχουμε χρησιμοποιώντας είτε

α. Συναρτήσεις που χρησιμοποιούν καρτεσιανές συντεταγμένες, ενδεικτικά line, fplot, plotyy, semilogx, semilogy, loglog, hist, bar, barh, pareto, stem, stairs, errorbar.

β. Συναρτήσεις που χρησιμοποιούν πολικές ή σφαιρικές συντεταγμένες, ενδεικτικά polar, rose.

γ. Συναρτήσεις που ζωγραφίζουν βέλη, ενδεικτικά compass, feather, quiver.

δ. Συναρτήσεις που χρωματίζουν, ενδεικτικά area.

Β. Δημιουργία γραφικού στον χώρο.

Β.1. Καμπύλες. Για την απλούστερη μορφή τρισδιάστατων γραφικών παραστάσεων χρησιμοποιούμε την συνάρτηση plot3. Λειτουργεί ακριβώς

όπως και η συνάρτηση plot3 – η οποία περιγράφηκε αμέσως παραπάνω – με την μόνη διαφορά ότι πλέον υπάρχει και ένα τρίτο διάνυσμα που αναπαριστά την τρίτη συνιστώσα των σημείων. Δεύτερη σημαντική εντολή σε αυτήν την κατηγορία είναι η εντολή *view*, με την οποία καθορίζεται η θέση από την οποία ο παρατηρητής βλέπει την γραφική παράσταση.

Β.2. Επιφάνειες. Για την γραφική παράσταση μιας επιφάνειας στον τρισδιάστατο χώρο χρησιμοποιούμε τις εξής συναρτήσεις:

α. η συνάρτηση mesh και οι παραλλαγές της και

β. η συνάρτηση surf και οι παραλλαγές της.

Β.3. Ισοβαρείς ή ισοσταθμικές καμπύλες. Για την γραφική παράσταση μιας ισοβαρούς ή ισοσταθμικής καμπύλης στον τρισδιάστατο χώρο χρησιμοποιούμε τις εξής συναρτήσεις:

α. η συνάρτηση τύπου contour,

β. η συνάρτηση pcolor και

γ. η συνάρτηση τύπου graph3d.

Γ. Δημιουργία γραφικής διεπιφάνειας χρήστη.

Αξιοποιώντας την εργαλειοθήκη που παρέχει το πρόγραμμα ο χρήστης μπορεί να δημιουργήσει την δική του γραφική διεπιφάνεια είτε ξεκινώντας από το μηδέν (from scratch) είτε δημιουργώντας κάποιο figure προγραμματίζοντας σε m-αρχείο – οπότε και θα εφαρμόσει τεχνική χαμηλού επιπέδου προγραμματισμού - είτε χρησιμοποιώντας το GUIDE – που αποτελεί μια υψηλού επιπέδου τεχνική προγραμματισμού.

5. Scilab[54]

Χαρακτηριστικά. Αν και το πρόγραμμα εμφανίζει πολλές λειτουργικές ομοιότητες με το Matlab, στο πεδίο της δημιουργίας γραφικών παραστάσεων δεν είναι απολύτως συμβατά. Εξαίρεση συνιστούν οι απλές συναρτήσεις plot και mesh του Matlab και plot3d του Scilab – η οποία παράγει τα ίδια αποτελέσματα με την mesh. Ωστόσο υπάρχει μια σειρά από συναρτήσεις οι

[54]Haugen, Finn: Master Scilab! 2008 διαθέσιμο σε: http://home.hit.no/~finnh/scilab_scicos/scilab/index.htm#help

οποίες γράφτηκαν ακριβώς για να είναι συμβατές με τις αντίστοιχες του Matlab και οι οποίες έχουν μπροστά το πρόθεμα mtlb_ , π.χ. mtlb_ plot, mtlb_ subplot, mtlb_ axes. Αν και για τις συναρτήσεις αυτές δεν παρέχεται η λειτουργία “help” - ενδεχομένως σε μια προσπάθεια να κατευθύνονται οι χρήστες στην χρήση των αντίστοιχων λειτουργιών του εν λόγω προγράμματος - είναι και πάλι ευκολότερες στην χρήση. Αξίζει να σημειωθεί ότι υπάρχει και η συνάρτηση mtlb_e χωρίς όμως να υπάρχει και η αντίστοιχη συνάρτηση στο Matlab (η οποία θα εμφανιζόταν ως e).

Α. Παράθυρα γραφικών. Το πρόγραμμα δίνει την δυνατότητα στον χρήστη να χρησιμοποιεί περισσότερα του ενός παράθυρα γραφικών ScilabGraphicx (όπου x είναι ο αριθμός κάθε παραθύρου ώστε να διευκολύνεται ο χειρισμός τους) όμως κάθε φορά μόνο ένα από αυτά είναι ενεργό. Υπό αυτήν την έννοια, ο χρήστης μπορεί να δημιουργήσει απευθείας το παράθυρο γραφικών αριθμός 7 ενώ παράλληλα να δημιουργεί ή διαγράφει άλλα παράθυρα. Η εκτέλεση της εντολής *plot* επιφέρει αυτομάτως την δημιουργία ενός νέου παραθύρου, εάν αυτό κριθεί αναγκαίο. Κάθε παράθυρο γραφικών έχει συγκεκριμένο περιεχόμενο και ως εκ τούτου η ίδια εντολή *plot* μπορεί να έχει διαφορετικά αποτελέσματα σε διαφορετικά παράθυρα.

Υπάρχουν τέσσερα κουμπιά στο παράθυρο γραφικών:

- 3D Rot.: ο χρήστης μπορεί με το ποντίκι να προσδώσει περιστροφική κίνηση σε ένα τρισδιάστατο γραφικό.
- 2D Zoom: ο χρήστης μπορεί να ζουμάρει σε ένα δισδιάστατο γραφικό.
- UnZoom: ο χρήστης επιστρέφει στο αρχικό γραφικό (όχι στο προηγούμενο γραφικό σε περίπτωση διαδοχικών ζουμ)
- File: ο χρήστης μπορεί να ανοίξει διάφορες εντολές και μενού.

Η εντολή *print*: ανοίγει a selection panel για εκτύπωση

Η εντολή *export* ανοίγει a selection panel για αντιγραφή ενός γραφικού σε ένα αρχείο με μια συγκεκριμένη μορφή (Postscript, Postscript-Latex, Xfig).

Η εντολή *save* αποθηκεύει απευθείας το γραφικό σε ένα αρχείο με ένα συγκεκριμένο όνομα. Αυτό το αρχείο μπορεί στην συνέχεια ο χρήστης να το επαναφέρει στο παράθυρο για να ξανασχεδιάσει το γραφικό.

Η εντολή *close* είναι όμοια με την εντολή *delete graphic window* - η οποία βρίσκεται στο μενού του βασικού παραθύρου εντολών του προγράμματος – όμως αυτή εκτελείται στο συγκεκριμένο παράθυρο (με αποτέλεσμα να διαγραφεί το περιεχόμενο του γραφικού).

Β. Η εντολή plot. Η λειτουργία της εντολής απεικονίζεται μέσα από το ακόλουθο παράδειγμα:

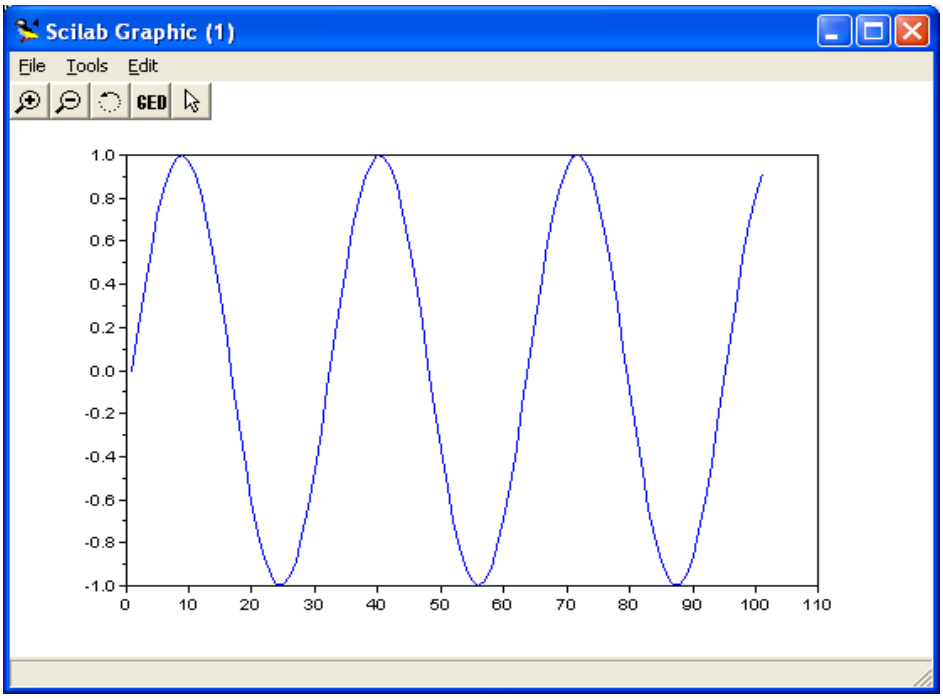
Έστω ότι τα δεδομένα που χρησιμοποιεί ο χρήστης είναι τα εξής:

```
t=[0:.1:10]';
u=-1+0.2*t;
y=sin(2*t);
```

1. Δίνοντας την ακόλουθη εντολή:

```
scf(1);
plot(y)
```

Θα προκύψει το παρακάτω διάγραμμα, όπου παράλληλα με τον άξονα του x είναι οι δείκτες του διανύσματος y, οι οποίοι είναι ακέραιοι από το 1 μέχρι 101.



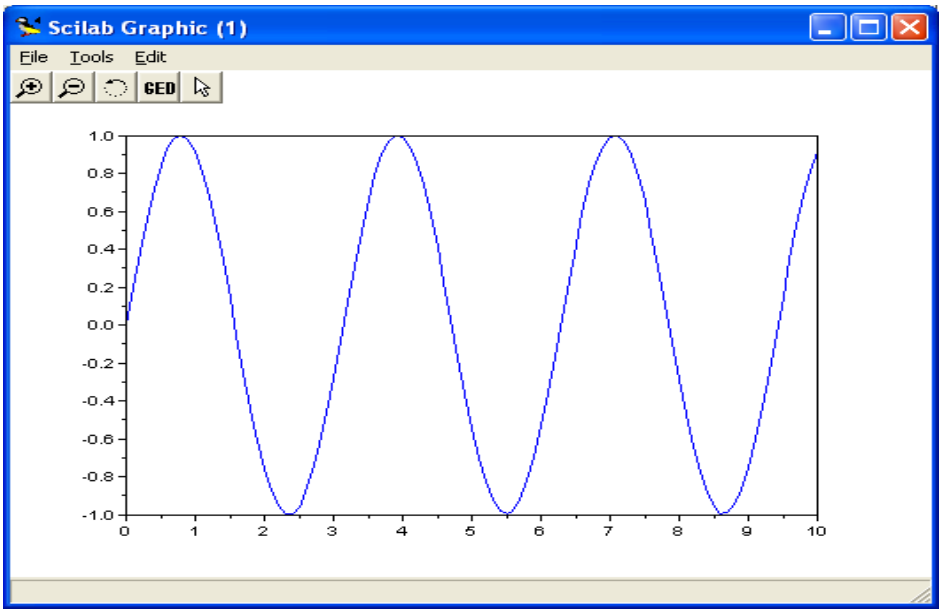
2. Δίνοντας την εντολή *plot(t,y)*, όπου τα διανύσματα t και y έχουν το ίδιο μήκος (ίδιο αριθμό στοιχείων), τότε το πρόγραμμα θα προσθέσει στο προηγούμενο γραφικό δύο ή περισσότερες καμπύλες. Συνεπώς προκειμένου να διαγράψει το προηγούμενο γραφικό, χρησιμοποιεί την εντολή *clf* (clear figure) πριν την εντολή *plot*

scf(1);

clf;

plot(t,y)

Οπότε πλέον στο άξονα του x ορίζονται οι τιμές του t



3. Εάν ωστόσο ο χρήστης θέλει το νέο αυτό γραφικό να δημιουργηθεί σε ένα νέο παράθυρο γραφικών και όχι να αντικαταστήσει το προηγούμενο γραφικό, τότε δίνει τις παρακάτω εντολές:

scf(2);

plot(t,y)

4. Για να βάλει ο χρήστης πλαίσιο ή τίτλους και επικεφαλίδες, δίνει τις ακόλουθες εντολές:

scf(1);

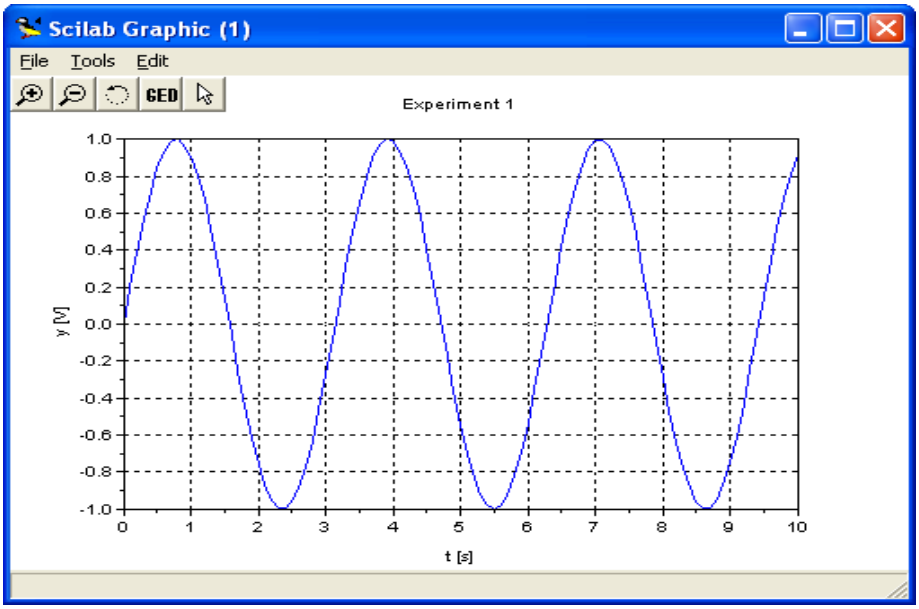
plot(t,y)

ax1=gca();ax1.grid=[0,0];

title('Experiment1')

xlabel('t[s]')

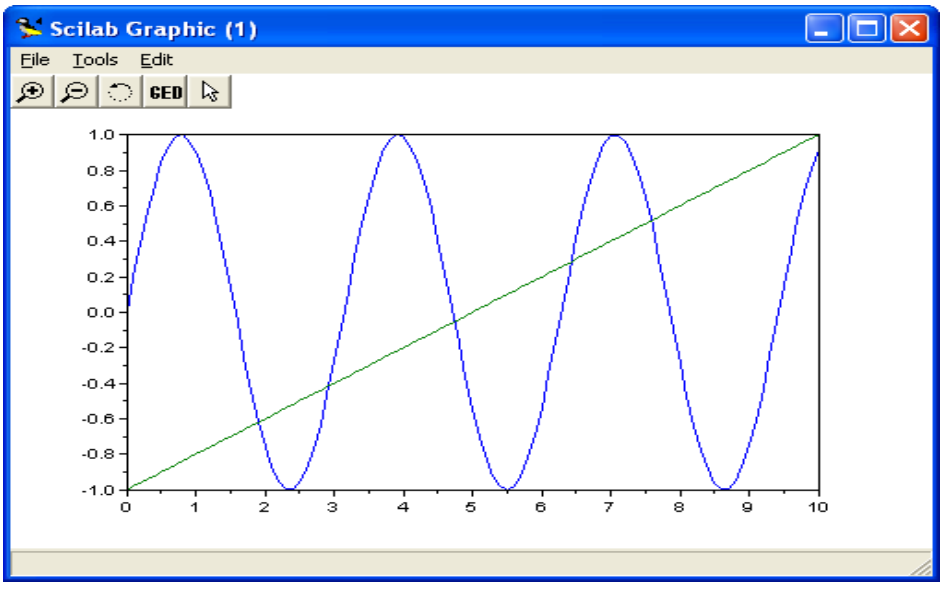
ylabel('y [V]')



5. Προκειμένου να σχεδιάσει δύο καμπύλες στο ίδιο γραφικό:

scf(1);clf;

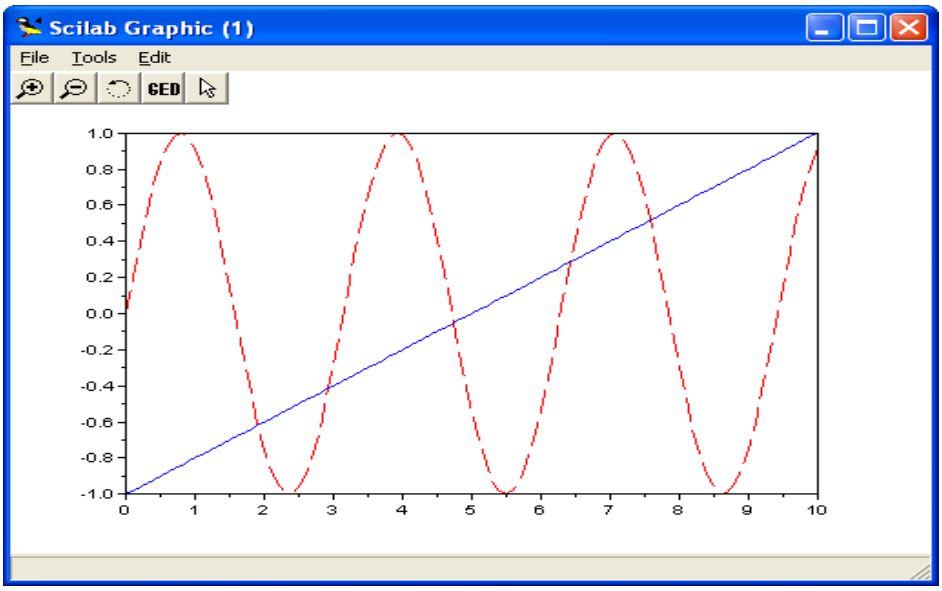
plot(t,y,t,u)



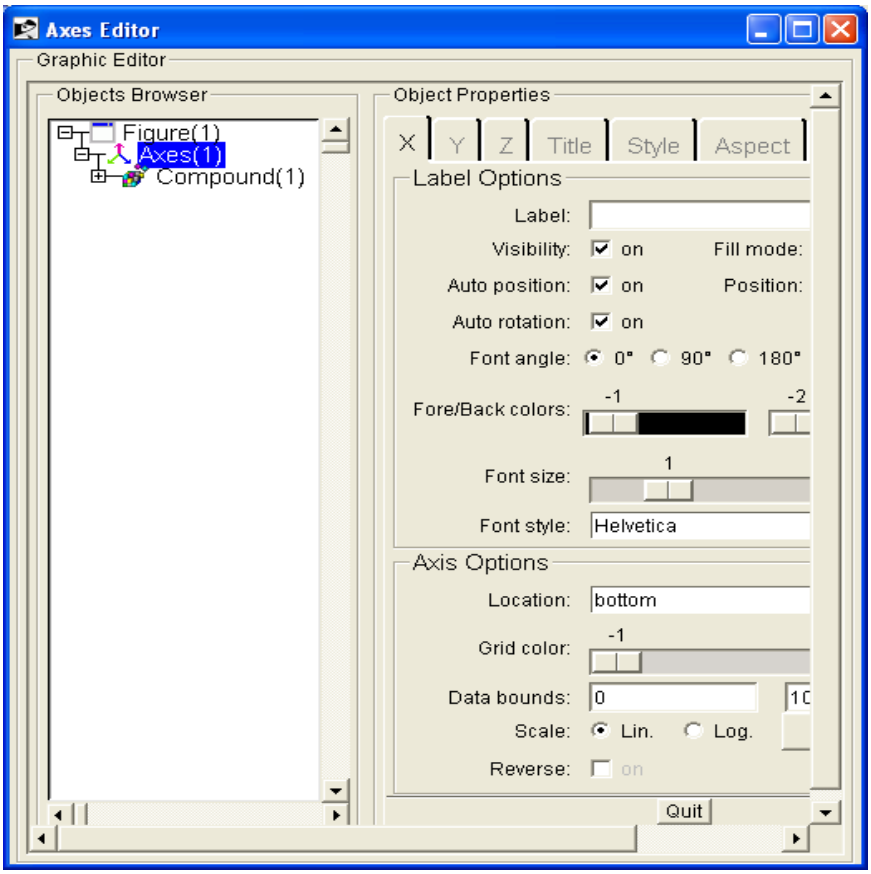
6. Και περαιτέρω εάν θέλει να ορίσει διαφορετικά χαρακτηριστικά για την κάθε καμπύλη:

scf(1); clf;

plot(t,y,'r--',t,u,'b')



Για το ίδιο αποτέλεσμα, αντί να ορίζει τις παραμέτρους της εντολής, ο χρήστης μπορεί από το παράθυρο του γραφικού να κάνει κλικ στο κουμπί GED, οπότε βρίσκει τον Graphics Editor. Από εκεί ο χρήστης μπορεί να αλλάζει το χρώμα και την μορφή των γραμμών, να βάζει τίτλους στους άξονες κτλ, όπως φαίνεται και στην εικόνα που ακολουθεί.



7. Επίσης μπορεί να χωρίσει το παράθυρο σχεδιάζοντας ταυτόχρονα περισσότερα από ένα γραφικά σε επιμέρους τμήματα του παραθύρου.

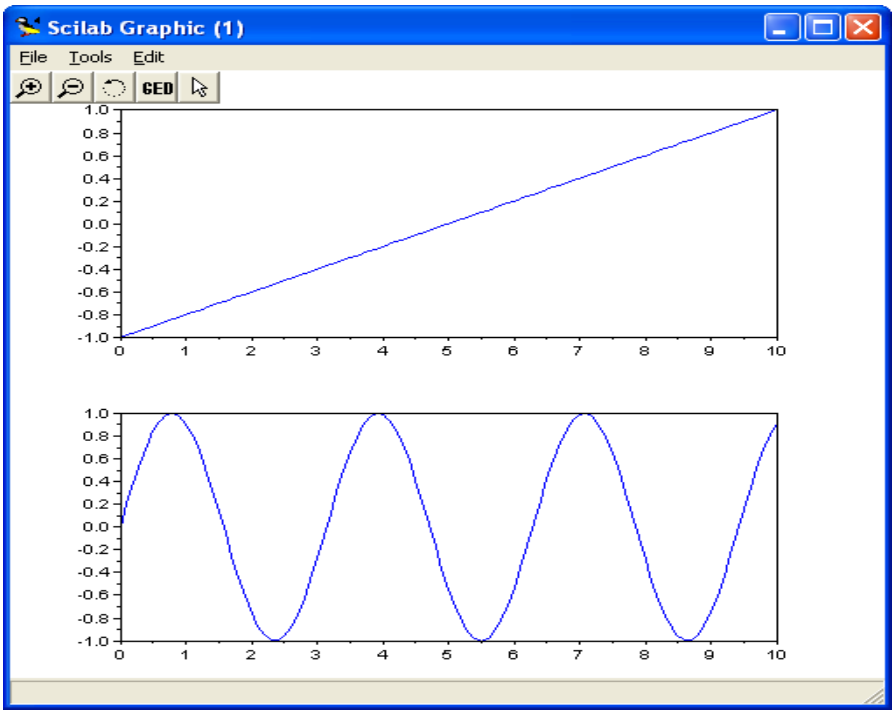
scf(1);clf;

subplot(211)

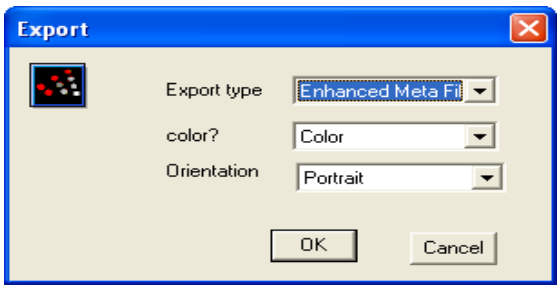
plot(t,u)

subplot(212)

plot(t,y)



Γ. Δημιουργία αρχείων γραφικών. Από το παράθυρο του γραφικού, ο χρήστης επιλέγει από το μενού File / Export, οπότε και θα ανοίξει το παρακάτω παράθυρο, οπότε και έχει την δυνατότητα να ορίσει τις παραμέτρους που τον ενδιαφέρουν ανάλογα με το περιεχόμενο του αρχείου που θέλει να δημιουργήσει, π.χ. αν θέλει να αποθηκεύσει στο αρχείο ένα κείμενο MS Word τότε είναι προτιμότερο να επιλέξει την μορφή αρχείου EMF (Enhanced Meta File).



ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ - ΔΟΜΕΣ ΕΛΕΓΧΟΥ ΡΟΗΣ

Οι δομές ελέγχου ροής είναι σύνολα εντολών με ειδικό τρόπο σύνταξης που αποσκοπούν στην ενσωμάτωση των υπολογιστικών αλγορίθμων στην εκάστοτε γλώσσα προγραμματισμού.

1. Gauss

Α. Εντολές ελέγχου

Α.1. Η δομή if[55]

Γενική σύνταξη: *if* συνθήκη1;
εντολή1;
endif;

Σύνθετη μορφή: *if* συνθήκη1;
εντολή1;
elseif συνθήκη2;
εντολή2;
elseif συνθήκη3;
εντολή3;
…
else συνθήκη4;
εντολή4;
endif;

Περιγραφή λειτουργίας: Κάθε συνθήκη συνδέεται με ένα σύνολο εντολών. Κάθε συνθήκη ελέγχεται με την σειρά που εμφανίζεται στη γραμμή εντολών. Εάν η συνθήκη είναι αληθής (true), τότε θα εκτελεστεί το σύνολο των εντολών. Μόλις ολοκληρωθεί η εκτέλεση των εντολών, το πρόγραμμα θα μεταπηδήσει στο τέλος του βρόχου και θα συνεχίσει την εκτέλεση από εκεί. Δηλαδή το πρόγραμμα θα εκτελέσει το πολύ ένα σύνολο εντολών. Για τον λόγο αυτό, ακόμα και αν υπάρχουν περισσότερες αληθείς συνθήκες, το

[55] *Aptech Systems Inc:* User Guide, Version 9.0, July 2008, κεφ. 10.8.2., σελ 10-34 επ.

πρόγραμμα θα εκτελέσει την πρώτη από αυτές και θα αγνοήσει τις υπόλοιπες. Αν καμία συνθήκη δεν ικανοποιείται, τότε το πρόγραμμα δεν θα εκτελέσει καμία εντολή εκτός εάν υπάρχει και εντολή else. Η εντολή αυτή δεν συνδέεται με καμία συνθήκη και συνεπώς αν το πρόγραμμα φτάσει σε αυτήν, τότε την εκτελεί πάντα. Το γεγονός ότι το πρόγραμμα φτάνει στην εντολή else σημαίνει ότι καμία συνθήκη δεν ήταν αληθής αλλά και ότι σίγουρα θα εκτελεστεί κάποια εντολή.

Β. Εντολές επανάληψης

Β.1. Η δομή for[56]

Χρήση: ο βρόχος σταματά να επαναλαμβάνεται όταν μια συνθήκη ικανοποιηθεί και στην συγκεκριμένη περίπτωση αφορά στον αριθμό των επαναλήψεων που είναι προκαθορισμένος και γνωστός εκ των προτέρων.

Γενική μορφή: *for i (start, stop, interval)*;
:
endfor;

Περιγραφή λειτουργίας: ο βρόχος σταματά μόνο όταν ξεπεραστεί το όριο, που έχει τεθεί. Οι μεταβλητές μέσα στην παρένθεση ορίζουν τον αριθμό των επαναλήψεων. Ο δείκτης *i* μπορεί να αναφέρεται σε στοιχείο μέσα στο βρόχο και να αποτελεί και στοιχείο του ίδιου του βρόχου, με την έννοια ότι αν δεν υπάρχει όταν ο βρόχος ξεκινά, μπορεί να δημιουργηθεί. Το *for* είναι ταχύτερο από τις δύο άλλες δομές (*while/until***)** ωστόσο έχει πιο περιορισμένη λειτουργικότητα. Ο βρόχος δέχεται μόνο ακέραιους ως στοιχεία ενώ αντίθετα τα *while/until* δέχονται τα πάντα, όπως ακέραιους, μήτρες, string.

Β.2. Η δομή while/until[57]

Χρήση: χρησιμοποιείται όταν οι συνθήκες για να μπει ή να βγει από τον βρόχο πρέπει να αξιολογούνται διαρκώς. Η χρήση των δύο αυτών δομών είναι ίδια εκτός από το εξής: οι βρόχοι *do while* εκτελούν τις εντολές μέχρι η συνθήκη να είναι ψευδής (false) ενώ οι βρόχοι *do until* μέχρι η συνθήκη να

[56] *Aptech Systems Inc:* User Guide, Version 9.0, July 2008, κεφ. 10.8.1., σελ 10-34 επ.
[57] *Aptech Systems Inc:* User Guide, Version 9.0, July 2008, κεφ. 10.8.1., σελ 10-32 επ.

είναι αληθής (true). Συνεπώς οι εντολές *do while* συνθήκη και *do until* μη συνθήκη είναι όμοιες.

Γενική μορφή: *do while* συνθήκη;
εντολή;
endo;

do until συνθήκη;
εντολή;
endo;

Περιγραφή λειτουργίας: κατ' αρχάς ελέγχει την συνθήκη. Αν είναι αληθής, εκτελεί τις εντολές του βρόχου και επιστρέφει στο βήμα 1, το οποίο και επαναλαμβάνει. Αν η συνθήκη είναι ψευδής, τότε προσπερνά όλες τις εντολές του βρόχου και συνεχίζει την εκτέλεση της πρώτης εντολής μετά τον βρόχο. Πρέπει να σημειωθεί ότι η συνθήκη ελέγχεται πριν μπει στο βρόχο, συνεπώς αν βρεθεί ψευδής, δεν θα εισέλθει καθόλου. Επίσης δεν υπάρχει καμία ένδειξη στον ορισμό του βρόχου που να ορίζει πώς θα καθοριστεί ή θα τροποποιηθεί η συνθήκη, οπότε εναπόκειται στον χρήστη να εξασφαλίσει ότι η συνθήκη έχει οριστεί σωστά σε κάθε βήμα.

Προκειμένου να σταματήσει ο χρήστης την επανάληψη (τόσο στην δομή *for* όσο και στην δομή *while/until*) έχει τρεις επιλογές:

- την εντολή *break*, με την οποία βγαίνει αμέσως από τον βρόχο,
- την εντολή *continue*, με την οποία η εκτέλεση μεταφέρεται στην αρχή του βρόχου, όπου η υπό εξέταση συνθήκη επανεκτιμάται,
- την δομή *if*, εξασφαλίζει πιο ορθολογική έξοδο από τον βρόχο (βλ. αναλυτικά παραπάνω).

2. Maple[58]

Α. Εντολές επανάληψης

Α. 1. Η δομή for

Χρήση: χρησιμοποιείται όταν ο χρήστης γνωρίζει τον αριθμό των επαναλήψεων.

Γενική μορφή:

[58] Maplesoft: Maple User Manual, 2008, κεφ. 8.2 σελ. 326

for δείκτης *from* έκφραση (αρχική τιμή) *by* έκφραση (βήμα επανάληψης) *to* έκφραση (τελική τιμή) *do*

εντολές

end do;

Περιγραφή της λειτουργίας: η διαδικασία επαναλαμβάνεται όσες φορές ορίζει η αρχική τιμή, η τελική τιμή και το βήμα της επανάληψης.

Α.2. Η δομή while

Χρήση: χρησιμοποιείται όταν ο χρήστης επιδιώκει την επανάληψη συγκεκριμένων εντολών όσο ισχύει μια συνθήκη.

Γενική μορφή:

while λογική έκφραση *do*

εντολές

end do;

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.

Β. Εντολές ελέγχου

Β. 1. Η δομή if

Χρήση: χρησιμοποιείται όταν είναι απαραίτητη η λήψη αποφάσεων, ανάλογα με κάποιες συνθήκες.

Γενική μορφή:

if λογική παράσταση *then*

εντολές

elif λογική παράσταση *then*

εντολές

else

εντολές

end if;

Περιγραφή της λειτουργίας: η δομή αυτή ορίζει έναν έλεγχο. Με την ανωτέρω μορφή ο χρήστης μπορεί να ελέγξει 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

και 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
end if;

και

if λογική παράσταση *then*
εντολές
else
εντολές
end if;

3. Mathematica[59]

Το πρόγραμμα έχει ενσωματωμένη μια ξεχωριστή και δυναμική γλώσσα προγραμματισμού, που πολλοί υποστηρίζουν ότι έχει ομοιότητες με τις γλώσσες προγραμματισμού C++ και Scheme. Οι εντολές για την ροή δεδομένων ποικίλουν από τις πλέον απλές έως τις πιο περίπλοκες.

Α. Η δομή do

Γενική σύνταξη: *Do*[εντολή,{…}];

Περιγραφή της λειτουργίας: είναι από τις πιο βασικές εντολές. Για παράδειγμα: Do[Body,{i, imin, imax, step}] το πρόγραμμα εκτελεί την διαδρομή Body, η οποία εξαρτάται από την παράμετρο i, ξεκινώντας από το

[59] http://reference.wolfram.com/mathematica/guide/ProceduralProgramming.html

imin και ολοκληρώνοντας στο imax σε βήματα step. Για την ολοκλήρωση δεν χρειάζεται να ικανοποιηθούν όλες οι παράμετροι.

Β. Η δομή for

Γενική σύνταξη: *for[*εντολή*]*

Περιγραφή της λειτουργίας: η εκτέλεση είναι πιο αργή και η σύνταξη πιο δύσκολη από την δομή *do*. Για παράδειγμα: For[start, test, incr, body] το πρόγραμμα εκτελεί το start κατόπιν επαναλαμβάνει διαρκώς το body μέχρι το test να αποτύχει να επιστρέψει True.

Γ. Η δομή if

Γενική σύνταξη: *if[*εντολή*]*

Περιγραφή της λειτουργίας: η δομή αυτή έχει τρία στοιχεία. Το πρώτο είναι ο έλεγχος. Αν από τον έλεγχο προκύψει ότι η συνθήκη είναι αληθής, τότε εκτιμάται και το δεύτερο στοιχείο. Αν από τον έλεγχο προκύψει ότι η συνθήκη είναι ψευδής, τότε εκτιμάται το τρίτο στοιχείο. Για παράδειγμα: If[condition, t, f, u] το πρόγραμμα επιστρέφει t εάν η συνθήκη είναι αληθής και f εάν η συνθήκη είναι ψευδής και u σε κάθε άλλη περίπτωση. Αν το u δεν οριστεί τότε δεν γίνεται τίποτε.

Δ. Η δομή Module

Γενική σύνταξη: Module [{...}, ...]

Περιγραφή λειτουργίας: Είναι το βασικό εργαλείο προγραμματισμού. Τα στοιχεία της λίστας {...} αποκαλούνται τοπικές μεταβλητές ή αναθέσεις, οι οποίες χρησιμοποιούνται μόνο εντός της Module. Το τμήμα εντός των αγκυλών και εκτός των αγκίστρων είναι μια μορφή ακολουθίας εντολών του προγράμματος, οι οποίες πρόκειται να εκτελεστούν.

Ε. Η δομή Block

Γενική σύνταξη: Block [{...}, ...]

Περιγραφή λειτουργίας: Η μόνη διαφορά από την Module είναι ο τρόπος που εισάγονται οι τοπικές μεταβλητές. Στην Module ο χρήστης δίνει στις τοπικές μεταβλητές ένα μοναδικό όνομα, η Module ξαναγράφεται βάσει αυτών των μοναδικών ονομάτων και η νέα Module υπολογίζεται επί τη

βάσει των τιμών που έχει ορίσει ο χρήστης. Αντίθετα στην Block, οι τοπικές μεταβλητές δεν παίρνουν μοναδικά ονόματα.

ΣΤ. Η δομή With

Γενική σύνταξη: With [{...}, ...]

Περιγραφή λειτουργίας: Υπάρχουν δύο εκδοχές της δομής: α) With[{x = a, y= b}, expr], οπότε υπολογίζει το expr με αντικαταστάσεις εντός των αγκίστρων. Στοιχεία όπως τα x, y παραμένουν μεταβλητές εκτός της With και β) With[{x,y, … }, body], οπότε λειτουργεί όπως η Module ή Block αλλά επιτρέπει την είσοδο και έξοδο αναθέσεων ή τοπικών μεταβλητών.

Ζ. Η δομή while

Γενική σύνταξη: *while*[συνθήκη, εντολή];

Περιγραφή λειτουργίας: Για παράδειγμα: While[test, body] ενόσω το test είναι αληθές, το body υπολογίζεται.

Η. Η δομή Which

Γενική σύνταξη: *Which[*συνθήκη, εντολή*];*

Περιγραφή λειτουργίας: Για παράδειγμα: Which[test1, value1, test2, value2, …], οπότε υπολογίζει το test1 και υπολογίζει και το value1, εάν το test1 είναι αληθές. Εάν το test1 είναι ψευδές, υπολογίζει το test2 και εάν το test2 είναι αληθές, υπολογίζει το value2.Συνεχίζει μέχρι το πρώτο ζευγάρι που θα είναι αληθές.

Θ. Η δομή Switch

Γενική σύνταξη: *Switch[*συνθήκη, εντολή*];*

Περιγραφή λειτουργίας: Για παράδειγμα: Switch[expr, form1, value1, form2, value2, …], οπότε υπολογίζει το expr και το συγκρίνει με το form1 και υπολογίζει το value1, εφόσον υπάρχει ταύτιση. Συνεχίζει μέχρι να βρεθεί το πρώτο ζευγάρι.

Ι. Η δομή OptionQ

Γενική σύνταξη: opt___?

Περιγραφή λειτουργίας: χρησιμοποιείται για να μπορεί το πρόγραμμα να αναγνωρίζει τις παραμέτρους των συναρτήσεων που έχει δημιουργήσει ο χρήστης. Αυτή θα είναι η τελευταία μεταβλητή στην συνάρτηση και αποτελείται από έναν ή περισσότερους κανόνες. FilterOptions χρησιμοποιούνται για να δημιουργηθεί μια υπολίστα παραμέτρων της συνάρτησης που έχει δημιουργήσει ο χρήστης ώστε να εισαχθεί σε μια συνάρτηση του προγράμματος στο Utilities package.

4. Matlab [60]

Ο χρήστης του προγράμματος χρησιμοποιεί τις ακόλουθες δομές ανάλογα εάν θέλει να επιτύχει:

Α. Σύγκριση

Αυτή επιτυγχάνεται με την χρήση σχεσιακών τελεστών και ειδικότερα:

<, <=, >, >=, ~=

Το αποτέλεσμα που θα προκύψει είναι μια λογική τιμή, δηλαδή το αποτέλεσμα θα είναι true ή false. Στο συγκεκριμένο πρόγραμμα μόνο το 0 είναι ίσο με false. Όλες οι υπόλοιπες τιμές είναι true. Οι τελεστές αυτοί μπορεί να χρησιμοποιηθούν και σε μήτρες με την ίδια μορφή.

Ωστόσο το αποτέλεσμα της σύγκρισης αυτής μπορεί να τροποποιηθεί με την χρήση λογικών τελεστών. Αυτοί χρησιμοποιούνται για να συνδέσουν λογικές εκφράσεις [με το & (and και) και το | (or ή)] ή να αλλάξουν μια λογική τιμή, [με το ~ (not όχι)], με σκοπό να προκύψει μια νέα λογική τιμή.

Β. Εκτέλεση υπό όρους[61]

Η σύγκριση ή κάποιος άλλος λογικός έλεγχος έχουν ως αποτέλεσμα συγκεκριμένα τμήματα του κώδικα να εκτελούνται ή να προσπερνώνται. Η εκτέλεση υπό όρους περιλαμβάνει τις ακόλουθες δομές:

Β. 1. Η δομή if

Χρήση: χρησιμοποιείται για να εκτελεστούν επί μέρους εντολές ανάλογα με τις συνθήκες που ικανοποιούνται.

Γενική μορφή: *if* συνθήκη_0

[60] Παπαρρίζος: MATLAB 6.5. Εκδόσεις Ζυγός, Θεσσαλονίκη 2004, κεφ. 3 σελ. 93 επ.
[61] http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/learn_matlab/f4-1931.html

σύνολο εντολών_0

elseif συνθήκη_1

σύνολο εντολών_1

:

:

elseif συνθήκη_k

σύνολο εντολών_k

else

σύνολο εντολών_(k+1)

end

Περιγραφή της λειτουργίας: τα μπλοκ των εντολών *elseif* και *else* είναι προαιρετικά. Τα σύνολα των εντολών μπορούν να περιέχουν οποιεσδήποτε εντολές ακόμη και την ίδια την εντολή *if*. Σε περίπτωση που μέσα σε ένα *if* περιλαμβάνονται και άλλα *if*, αυτά ονομάζονται φωλιασμένα (*nested if*). Ο προγραμματισμός με φωλιασμένα *if* είναι μεν πιο αποτελεσματικός από άποψη χρόνου αλλά ενέχει περισσότερες πιθανότητες λογικών λαθών. Το σύνολο των εντολών θα εκτελεστεί μόνο εάν η συνθήκη του *if* είναι αληθής (true).

Β. 2. Η δομή switch

Χρήση: χρησιμοποιείται για να εκτελεστεί κυρίως σε περιπτώσεις όπου το αποτέλεσμα ενός ελέγχου χρησιμοποιείται σε άλλους ελέγχους καθώς και σε αυτές που, ανάλογα με την τιμή μιας μεταβλητής ή παράστασης εκτελούνται και ανάλογες εντολές.

Γενική μορφή: *switch* παράσταση

case τιμή_1

σύνολο εντολών

case {τιμή_2, τιμή_3,…}

σύνολο εντολών

:

:

otherwise

σύνολο εντολών

end

Περιγραφή της λειτουργίας: Αν η παράσταση πάρει την τιμή τιμή_1, θα εκτελεστεί το σύνολο των εντολών που βρίσκεται ακριβώς κάτω από την έκφραση case τιμή_1. Αν πάρει μία από τις τιμές τιμή_2, τιμή_3…θα εκτελεστεί το σύνολο των εντολών κάτω από την έκφραση case {τιμή_2, τιμή_3,…}. Οι τιμές οπωσδήποτε εσωκλείονται σε αγκύλες. Αν η παράσταση δεν πάρει κάποια από τις τιμές που ακολουθούν το μπλοκ της εντολής *case*, τότε εκτελείται το σύνολο των εντολών του μπλοκ της εντολής *otherwise*, το οποίο όμως είναι προαιρετικό, ενώ υπάρχει η δυνατότητα να συμπεριληφθεί και οποιοσδήποτε αριθμός από το μπλοκ της εντολής *case*. Σημειωτέον ότι η παράσταση μπορεί να είναι μεταβλητή, αριθμητική και συμβολοσειρά.

Γ. Επανάληψη[62]

Γ. 1. Η δομή for

Χρήση: χρησιμοποιείται για να εκτελεστεί ένα σύνολο εντολών, όταν είναι προκαθορισμένος ο αριθμός των επαναλήψεων.

Γενική μορφή: *for* δείκτης = μήτρα
σύνολο εντολών
end

Περιγραφή της λειτουργίας: για κάθε στήλη της μήτρας του δείκτη εκτελείται το σύνολο των εντολών μέχρι την εντολή *end*. Το σύνολο των εντολών του βρόχου μπορεί να περιέχει οποιαδήποτε εντολή του προγράμματος, ακόμη και τον ίδιο τον βρόχο του *for*. Οι εντολές *for* ζευγαρώνουν με εντολές *end*, χαρακτηριστικό που παρατηρείται σε όλες τις δομές ελέγχου ροή του προγράμματος. Επίσης η εκτέλεση των εντολών του σώματος του βρόχου δεν μπορεί να διακοπεί πρόωρα με εντολές καταχώρησης στην τιμή του δείκτη του βρόχου. Αν και σε ορισμένες περιπτώσεις η χρήση του *for* είναι αναπόφευκτη, είναι δυνατόν να αντικατασταθεί με πράξεις διανυσμάτων, ενέργεια ταχύτερη, η οποία είναι γνωστή και ως διανυσματοποίηση (vectorization).

62 http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/learn_matlab/f4-1931.html

Γ.2. Η δομή while

Χρήση: χρησιμοποιείται για να εκτελεστεί ένα σύνολο εντολών, όταν δεν είναι προκαθορισμένος ο αριθμός των επαναλήψεων, που απαιτούνται για την ικανοποίηση μιας συνθήκης.

Γενική μορφή: *while* συνθήκη

σύνολο εντολών

end

Περιγραφή της λειτουργίας: οι εντολές εκτελούνται όσο η συνθήκη είναι αληθής. Δηλαδή, ελέγχεται πρώτα η συνθήκη και αν είναι αληθής εκτελούνται όλες οι εντολές της δομής. Μετά ελέγχεται και πάλι η συνθήκη και αν βρεθεί αληθής, επαναλαμβάνεται η εκτέλεση των εντολών. Η διαδικασία επαναλαμβάνεται μέχρι να βρεθεί ότι η συνθήκη είναι ψευδής. Συνεπώς αν στο σώμα των εντολών δεν υπάρχουν εντολές που να τροποποιούν την συνθήκη, η δομή θα εκτελείται επ' αόριστον. Για να σταματήσει η εκτέλεση του προγράμματος, ο χρήστης πληκτρολογεί Ctrl+c ταυτόχρονα.

Και στις δύο ανωτέρω δομές, η εντολή *continue* προσπερνά τις τιμές του δείκτη – για τις οποίες ο χρήστης δεν θέλει να εκτελεστούν οι εντολές - στον βρόχο μέσα στον οποίο βρίσκεται και ο έλεγχος περνά στην επόμενη σειρά επαναλήψεων του ίδιου βρόχου. Η εντολή *break* σταματά την εκτέλεση του βρόχου μέσα στον οποίο βρίσκεται και, στην περίπτωση των φωλιασμένων βρόχων, μεταφέρει τον έλεγχο της ροής στον αμέσως εξωτερικό βρόχο. Αντίθετα, η εντολή *return* σταματά την εκτέλεση της συνάρτησης μέσα στην οποία βρίσκεται και μεταφέρει τον έλεγχο της ροής στην αμέσως εξωτερική συνάρτηση, που μπορεί να είναι και το ίδιο το πρόγραμμα.

Ομοίως λειτουργεί η εντολή *return* και μέσα σε αρχεία script. Γενικά αξίζει να σημειωθεί πως ό,τι ισχύει για τις εντολές του παραθύρου εντολών ισχύει και για τις εντολές ενός αρχείου *script*. Το αρχείο αυτό είναι ένα εκτελέσιμο αρχείο του προγράμματος, είναι δηλαδή ένα *m-αρχείο* (η άλλη κατηγορία m-αρχείου είναι το αρχείο συναρτήσεων. Σε αυτό το αρχείο μπορεί να γραφεί κάθε σύνολο εντολών που μπορεί να εκτελεστεί από το παράθυρο εντολών και ειδικότερα όταν οι εντολές αυτές πρέπει να εκτελεστούν κατ' επανάληψη με μικρές κάθε φορά αλλαγές.

5. Scilab

Ο χρήστης του προγράμματος χρησιμοποιεί τις ακόλουθες δομές ανάλογα εάν θέλει να επιτύχει[63]:

Α. Σύγκριση

Υπάρχουν πέντε τρόποι για να μπορεί ο χρήστης να κάνει συγκρίσεις των τιμών των δεδομένων του προγράμματος. Αυτοί οι τελεστές είναι οι εξής:

== ίσο με

< μικρότερο από

> μεγαλύτερο από

<= μικρότερο ή ίσο με

>= μεγαλύτερο ή ίσο με

<> or ~= όχι ίσο με

Β. Εκτέλεση υπό όρους

Β.1. Η δομή if

Χρήση: χρησιμοποιείται ώστε ο χρήστης, ανάλογα με το αποτέλεσμα ενός ελέγχου, να μπορεί εναλλακτικά να εκτελεί κάποιες εντολές.

Γενική μορφή: *if…then*

εντολές

end

Σύνθετη μορφή: *if…then*

εντολές

else

εντολές

…

end

if…then

εντολές

elseif…then

[63] Introduction to Scilab, User's Guide διαθέσιμο σε: http://www.scilab.org/doc/intro/index.html

εντολές

…

else

εντολές

end

Β.2. Η δομή while

Χρήση: χρησιμοποιείται για να δημιουργήσει ο χρήστης βρόχους προγραμματισμού, οι οποίοι αξιοποιούνται στον υπολογισμό αθροισμάτων, εκτελώντας μια σειρά από εντολές μέχρι να ικανοποιηθεί μια συνθήκη.

Γενική μορφή: *while…end*

Β.3. Η δομή for

Χρήση: χρησιμοποιείται για να εκτελεί μια σειρά εντολών, αναφορικά με ένα σύνολο τιμών, σε προκαθορισμένες επαναλήψεις.

Γενική μορφή: *for μεταβλητή = γραμμή_διάνυσμα*

εντολή

…

εντολή

end

Περιγραφή λειτουργίας: ο χρήστης μπορεί να χρησιμοποιήσει την for τόσο με διανύσματα όσο και με μήτρες, θέτοντας ως τιμές της τα στοιχεία του διανύσματα ή τις στήλες της μήτρας. Ομοίως με τις μήτρες, μπορεί να χρησιμοποιηθεί με στις λίστες, δεχόμενη ως τιμές τα δεδομένα που εισάγονται στην λίστα.

Φωλιασμένοι βρόχοι *(nested loops)*, είναι οι βρόχοι for που εμπεριέχονται σε βρόχους for και μπορούν να χρησιμοποιηθούν για τον υπολογισμό διπλών αθροισμάτων καθώς και για να επεξεργαστεί ο χρήστης μεμονωμένα στοιχεία μιας μήτρας ή να δημιουργήσει μήτρες που ακολουθούν κάποιο συγκεκριμένο μοτίβο.

Β.4. Η δομή case

Χρήση: χρησιμοποιείται ώστε ο χρήστης να συγκρίνει μια έκφραση με περισσότερες πιθανές εκφράσεις και να επιλέξει να εκτελέσει τις οδηγίες

που έπονται της πρώτης περίπτωσης, οι οποίες και αντιστοιχούν στην αρχική έκφραση. Υπάρχει η δυνατότητα να θέσει ο χρήστης και μια δήλωση *else* σε περίπτωση που καμία από τις υποθέσεις δεν ικανοποιείται.

Γενική μορφή: *case…select*

Γ. Ειδικές εντολές συναρτήσεων

Το πρόγραμμα διαθέτει εντολές που χρησιμοποιούνται σχεδόν αποκλειστικά σε συναρτήσεις και οι οποίες είναι:

- η εντολή *argn*: επιστρέφει τον αριθμό των εισόδων και εξόδων δεδομένων στην συνάρτηση
- η εντολή *error*: αναστέλλει την λειτουργία μιας συνάρτησης, τυπώνει ένα μήνυμα σφάλματος και επιστρέφει στο προηγούμενο στάδιο της εκτέλεσης όταν εντοπιστεί ένα σφάλμα
- οι εντολές *warning* και *pause*: αναστέλλουν προσωρινά την λειτουργία μιας συνάρτησης
- η εντολή *break*: τερματίζει βίαια έναν βρόχο
- οι εντολές *return* και *resume*: χρησιμοποιούνται για επιστροφή σε περιβάλλον calling και για να περάσουν τοπικές μεταβλητές από περιβάλλον λειτουργίας σε περιβάλλον calling.

ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ

I. ΒΙΒΛΙΟΘΗΚΕΣ

1. Gauss

Α. Βιβλιοθήκες[64]

Η βιβλιοθήκη του προγράμματος είναι ένα αρχείο text, το οποίο λειτουργεί ως λεξικό για τα αρχεία πηγαίου κώδικα, όπου περιέχονται οι συμβολικοί ορισμοί. Η εντολή *LIBRARY* χρησιμοποιείται για την ενεργοποίηση των βιβλιοθηκών. Οι βιβλιοθήκες ως αρχεία έχουν την κατάληξη .lcg. Το σύστημα βιβλιοθηκών του προγράμματος επιτρέπει την δημιουργία και διατήρηση προγραμμάτων modular. Ο χρήστης μπορεί να δημιουργήσει βιβλιοθήκες με συχνά χρησιμοποιούμενες συναρτήσεις, τις οποίες το σύστημα του προγράμματος θα βρίσκει και θα μεταγλωττίζει αυτομάτως, κάθε φορά που θα χρησιμοποιούνται σε ένα πρόγραμμα.

Β. Εφαρμογές GAUSS[65]

Είναι προγράμματα που έχουν ήδη γραφτεί και τα οποία μπορούν να προσαρμοστούν. Είναι το αντίστοιχο των ToolBoxes του MATLAB. Οι εφαρμογές GAUSS είναι οι ακόλουθες:

Algorithmic Derivatives	A program for generating GAUSS procedures for computing algorithmic derivatives.
Constrained Maximum Likelihood MT	Solves the general maximum likelihood problem subject to general constraints on the parameters.
Constrained Optimization	Solves the nonlinear programming problem subject to general constraints on the parameters.
CurveFit	Nonlinear curve fitting.
Descriptive Statistics	Basic sample statistics including means, frequencies and crosstabs. This application is backwards compatible with programs written with Descriptive Statistics 3.1
Descriptive Statistics MT	Basic sample statistics including means, frequencies and crosstabs. This application is thread-safe and takes

[64] *Rossi, Eduardo:* Introduction to Gauss Programming Language. University of Pavia, October 2006, κεφ. 9.1., σελ. 155 επ.
[65] http://www.aptech.com/gauss_apps.html

	advantage of structures.
Discrete Choice	A statistical package for estimating discrete choice and other models in which the dependent variable is qualitative in some way.
FANPAC MT	Comprehensive suite of GARCH (Generalized AutoRegressive Conditional Heteroskedastic) models for estimating volatility.
Linear Programming MT	Solves small and large scale linear programming problems
Linear Regression MT	Least squares estimation.
Loglinear Analysis MT	Analysis of categorical data using loglinear analysis.
Maximum Likelihood MT	Maximum likelihood estimation of the parameters of statistical models.
Nonlinear Equations MT	Solves systems of nonlinear equations having as many equations as unknowns.
Optimization	Unconstrained optimization.
Time Series MT	Exact ML estimation of VARMAX, VARMA, ARIMAX, ARIMA, and ECM models subject to general constraints on the parameters. Panel data estimation. Unit root and cointegration tests.

Γ. Πρόσθετα (Add-on) packages[66]

Επειδή η βασική έκδοση του προγράμματος είναι μια σχετικά χαμηλού επιπέδου γλώσσα επεξεργασίας μητρών, προσφέρονται πλέον αυτά που ονομάζονται "add-ons". Αυτά είναι λειτουργίες γραμμένες από πριν και οι οποίες επιτρέπουν σχετικά σύνθετες λειτουργίες να εκτελούνται με βασικές γνώσεις του προγράμματος και χωρίς πολύ κόπο. Για παράδειγμα, υπάρχουν add-ons που περιέχουν πακέτα για:

- παλινδρόμηση OLS
- μη γραμμική και υπό συνθήκες εκτίμηση
- οικονομική και τεχνική ανάλυση
- εξομοίωση

[66] Ritchie, Felix: Programming in GAUSS, 09.10.2002 διαθέσιμο σε: http://www.trigconsulting.co.uk/gauss/man_basics.html

- ανάλυση δεδομένων
- πρόβλεψη

2. Maple[67]

Η βιβλιοθήκη περιέχει την πλειοψηφία των ρουτινών του προγράμματος. Περιλαμβάνει την λειτουργικότητα που σχετίζεται με τον υπολογισμό, την γραμμική άλγεβρα, την 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 διανυσμάτος 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 δίνουν τη δυνατότητα στον χρήστη να κάνει υπολογισμούς σε πολύ εξειδικευμένους τομείς προβληματισμού. Ο ακόλουθος πίνακας αφορά στα διαθέσιμα packages του προγράμματος.

algcurves	tools for studying one-dimensional algebraic curves defined by multi-variate polynomials
Algebraic	commands for performing computations with algebraic numbers
ArrayTools	tools used for low level manipulation of Matrices, Vectors, and Arrays
AudioTools	commands for audio file I/O and manipulation
Bits	commands for performing bit-wise operations efficiently
Cache	commands for cache table manipulation
CAD	tools to connect with CAD applications
codegen	tools for translating Maple procedures to other languages
CodeGeneration	tools for translating Maple code to other languages
CodeTools	commands for analyzing and profiling

[67] http://www.maplesoft.com/support/help/view.aspx?path=index/package

	Maple code
combinat	combinatorial functions, including commands for calculating permutations and combinations of lists, and partitions of integers
combstruct	commands for generating and counting combinatorial structures
ContextMenu	tools for building and modifying context-sensitive menus
CurveFitting	commands that support curve-fitting
Database	commands and Maplet applications for using databases
DEtools	tools for manipulating, solving, and plotting systems of differential equations
diffalg	commands for manipulating systems of differential polynomial equations (ODEs or PDEs)
difforms	commands for handling differential forms
DiscreteTransforms	commands for computing transforms of discrete data
DocumentTools	commands that allow programmatic access to Maple documents and components
Domains	commands for creating domains of computation
DynamicSystems	commands for creating, manipulating, simulating, and plotting linear systems objects
ExcelTools	commands that allow access to stored data in Microsoft Excel format
ExternalCalling	tools for calling external functions from Maple
FileTools	commands for file manipulation and processing
finance	commands for financial computations
GaloisField	procedures and constants for doing arithmetic in the finite field GF(p^k)
GaussInt	commands for working with Gaussian integers
genfunc	commands for manipulating rational generating functions
geom3d	commands for three-dimensional Euclidean geometry
geometry	commands for two-dimensional Euclidean geometry
gfun	commands for generating function manipulation
GraphTheory	collection of routines for creating, drawing, manipulating, and testing graphs

Groebner	commands for Groebner basis calculations in skew algebras
group	commands for working with permutation and finitely-presented groups
hashmset	commands for multisets
ImageTools	tools for image processing
InstallerBuilder	create an installer for a Maple toolbox
IntegerRelations	commands for approximating floating numbers by integer linear combinations of symbolic constants
IntegrationTools	tools used for manipulation of integrals
inttrans	commands for working with integral transforms and their inverses
LargeExpressions	tools for managing creation of computation sequences
LibraryTools	commands for library manipulation and processing
liesymm	commands for characterizing the contact symmetries of systems of partial differential equations
LinearAlgebra	commands for manipulating Matrices and Vectors as rtable data structures
LinearFunctionalSystems	commands for constructing solutions of linear functional systems of equations
LinearOperators	tools for solving linear functional equations, building annihilators and minimal annihilators, and performing accurate integration
ListTools	tools for manipulating lists
Logic	commands for manipulating expressions by using Boolean logic
LREtools	commands for manipulating, plotting, and solving linear recurrence equations
Maplets	tools to create graphical user interfaces for Maple
MathematicalFunctions	tools providing information about mathematical functions
MathML	commands for importing and exporting Maple expressions as MathML
Matlab	commands to facilitate a MATLAB Link
MatrixPolynomialAlgebra	tools for symbolic manipulation of polynomial matrices
MmaTranslator	tools for translating from Mathematica to Maple, expressions, command operations and notebooks
MultiSeries	commands for performing asymptotic and series expansions in general asymptotic scales

numapprox	commands for calculating polynomial approximations to functions on a given interval
numtheory	commands for classic number theory
Optimization	commands for numerically solving optimization theory problems
OreTools	tools for performing basic arithmetic in pseudo-linear (ore) algebra
Ore_algebra	routines for basic calculations in algebras of linear operators
OrthogonalSeries	tools for series of classical orthogonal polynomials
orthopoly	commands for generating various types of orthogonal polynomials
padic	commands for computing p-adic approximations to real numbers
PDEtools	tools for solving partial differential equations
Physics	a package implementing the standard mathematical physics computational objects and their operations
plots	commands for displaying graphical representations
plottools	commands for generating and manipulating graphical objects
PolynomialIdeals	commands for computing with polynomial ideals
PolynomialTools	commands for manipulating polynomial objects
powseries	commands for creating and manipulating formal power series represented in general form
priqueue	functions on priority queues
ProcessControl	commands for computing and visualizing statistical process control
QDifferenceEquations	commands for constructing solutions of linear q-difference equations
queue	commands on queue data structures
RandomTools	tools for working with random objects
RationalNormalForms	tools for using rational normal forms as a basis for constructing minimal representations and decomposing hypergeometric terms
RealDomain	provides a real number context
RegularChains	tools for solving systems of algebraic equations symbolically
RootFinding	advanced commands for finding roots numerically
ScientificConstants	commands for accessing physical constants

	and Periodic Table Element properties
ScientificErrorAnalysis	commands for representation and construction of numerical quantities with a value and error
Security	tools for Maple engine security
simplex	commands for linear optimization using the simplex algorithm
Slode	commands for finding formal power series solutions of linear ODEs
SNAP	symbolic-numeric algorithms for polynomial arithmetic
Sockets	tools for network communication in Maple
SoftwareMetrics	functions for quantifying code complexity
SolveTools	commands for solving systems of algebraic equations
Spread	tools for working with spreadsheets in Maple
stack	commands on stack data structures
Statistics	tools for mathematical statistics and data analysis
StringTools	optimized commands for string manipulation
Student	collection of packages covering undergraduate mathematics courses
Student[Calculus1]	commands to assist with the teaching and learning of single-variable calculus
Student[LinearAlgebra]	commands to assist with the teaching and learning of basic linear algebra
Student[MultivariateCalculus]	commands to assist with the teaching and learning of multivariate calculus
Student[Precalculus]	commands to assist with the teaching and learning of precalculus
Student[VectorCalculus]	commands to assist with the teaching and learning of vector calculus
sumtools	commands for computing indefinite and definite sums
SumTools	tools for finding closed forms of indefinite and definite sums
tensor	commands for calculating with tensors and their applications in General Relativity Theory
Threads	user level commands for using threads in Maple
Tolerances	provides computations with tolerances
ToolboxInstaller[Data]	access data during installation of a Maple toolbox
Typesetting	tools for programmatic access to Standard Worksheet Typeset and 2-D equation Parsing options

TypeTools	commands for extending the set of recognized types in the type command
Units	commands for converting values between units, and environments for performing calculations with units
VariationalCalculus	tools for Calculus of Variations computations
VectorCalculus	commands for performing multivariate and vector calculus operations
Worksheet	tools for generating and manipulating Maple worksheets
XMLTools	tools for using XML documents

3. Mathematica[68]

Γενικά. Το πρόγραμμα διαθέτει ποικίλα βασικά πρόσθετα packages, τα οποία παρέχουν πρόσθετη λειτουργικότητα σε συγκεκριμένα ζητήματα. Ο συγκεκριμένος κώδικας αυτών των packages μπορεί να είναι εκτελέσιμος και σε μελλοντικές εκδοχές του προγράμματος αλλά η γενική λειτουργικότητά τους θα είναι συχνά ενσωματωμένη απευθείας στον πυρήνα του συστήματος του προγράμματος.

Τα packages είναι αρχεία text τα οποία περιέχουν εντολές του προγράμματος. Είναι σχεδιασμένα ώστε να διευκολύνουν την διανομή των προγραμμάτων του χρήστη σε άλλους χρήστες αλλά και να παρέχουν έναν μηχανισμό στο χρήστη να δημιουργεί προγράμματα που είναι συμβατά με το ίδιο το πρόγραμμα.

Ειδικά. Η υπό έρευνα έκδοση του προγράμματος διαθέτει τα ακόλουθα packages:

Statistical Packages	Analysis of Variance
	Hierarchical Clustering
	Hypothesis Testing
	Multivariate Statistics
	Statistical Plots
Plotting, Charting, and Graphing Packages	Error Bar Plotting
	Plot Legends
	Splines
Discrete Math Packages	Combinatorica
	Computational Geometry

[68] http://reference.wolfram.com/mathematica/guide/StandardExtraPackages.html

	Graph Utilities
Calculus Packages	Equation Trekker
	Fourier Series
	Function Approximations
	Numerical Calculus
	Numerical Differential Equation Analysis
	Variational Methods
	Vector Analysis
Algebra Packages	Finite Fields
	Quaternions
Polyhedra & Polytopes Packages	Polyhedron Operations
	Polytopes
Cartography & Dates Packages	Calendar
	Geodesy
	World Plotting
Sound Packages	Audio
	Music
Physical Quantities & Properties Packages	Black-Body Radiation
	Physical Constants
	Resonance Absortion Lines
	Standard Athmosphere
	Units
Utility Packages	Benchmarking
	Developer Utilities
	Experimental Functions
	Notation
	XML
Miscellaneous Mathematical Packages	Computer Arithmetic
	Primality Proving

4. Matlab[69]

Το πρόγραμμα για την διευκόλυνση του χρήστη διαθέτει τις ακόλουθες εργαλειοθήκες (toolboxes), όπως αυτές ομαδοποιούνται σε κατηγορίες ανάλογα με την λειτουργία με την οποία σχετίζονται:

- Math and Optimization
    - Optimization Toolbox
    - Symbolic Math Toolbox
    - Partial Differential Equation Toolbox
    - Genetic Algorithm and Direct Search Toolbox
- Statistics and Data Analysis
    - Statistics Toolbox
    - Neural Network Toolbox
    - Curve Fitting Toolbox
    - Spline Toolbox
    - Model-Based Calibration Toolbox
- Control System Design and Analysis
    - Control System Toolbox
    - System Identification Toolbox
    - Fuzzy Logic Toolbox
    - Robust Control Toolbox
    - Model Predictive Control Toolbox
    - Aerospace Toolbox
- Signal Processing and Communications
    - Signal Processing Toolbox
    - Communications Toolbox
    - Filter Design Toolbox
    - Filter Design HDL Coder
    - Wavelet Toolbox
    - Fixed-Point Toolbox
    - RF Toolbox
- Image Processing
    - Image Processing Toolbox

[69] http://www.mathworks.com/products/product_listing/index.html?ref=pfomain

- Image Acquisition Toolbox
- Mapping Toolbox

- Test & Measurement
  - Data Acquisition Toolbox
  - Instrument Control Toolbox
  - Image Acquisition Toolbox
  - SystemTest
  - OPC Toolbox
  - Vehicle Network Toolbox
- Computational Biology
  - Bioinformatics Toolbox
  - SimBiology
- Financial Modeling and Analysis
  - Financial Toolbox
  - Financial Derivatives Toolbox
  - Datafeed Toolbox
  - Fixed-Income Toolbox
  - Econometrics Toolbox
- Application Deployment
  - MATLAB Compiler
  - Spreadsheet Link EX *(for Microsoft Excel)*
- Application Deployment Targets
  - MATLAB Builder EX *(for Microsoft Excel)*
  - MATLAB Builder NE *(for Microsoft .NET Framework)*
  - MATLAB Builder JA *(for Java language)*
- Database Connectivity and Reporting
  - Database Toolbox
  - MATLAB Report Generator

5. Scilab

Α. Βιβλιοθήκες

Είναι συλλογές συναρτήσεων, οι οποίες μπορούν να φορτωθούν αυτομάτως σε περιβάλλον του προγραμμάτος όταν αυτό χρησιμοποιείται ή να φορτωθούν όταν το επιλέγει ο χρήστης. Ο χρήστης μπορεί να δημιουργεί

βιβλιοθήκες με την εντολή *lib* ή και την εντολή *genlib*. Επιπλέον με την εντολή *s=librarieslist()* εμφανίζεται η ακόλουθη λίστα:

compatibility_ functilib	tclscilib	signal_ processinglib	jvmlib
umfpacklib	metanetgraph_ toolslib	statisticslib	integerlib
spreadsheetlib	metaneteditorlib	linear _algebralib	dynamic_linklib
development_ toolslib	scicos_menuslib	graphic _exportlib	timelib
scilab2fortranlib	scicos_utilslib	graphicslib	functionslib
maple2scilablib	scicos_autolib	parameterslib	elementary_ functionlib
matiolib	differential_ equationlib	simulated _annealinglib	data_structureslib
m2scilib	sparselib	genetic _algorithmslib	iolib
texmacslib	interpolationlib	stringlib	fileiolib
soundlib	polynomialslib	special _functionslib	demo_toolslib
scipadinternalslib	optimizationlib	overloadinglib	helptoolslib
scipadlib	cacsdlib	guilib	corelib

Το πρόγραμμα δημιουργήθηκε με πολλές υπάρχουσες βιβλιοθήκες, οι οποίες είναι ενδεικτικά[70]:

- Linear algebra LINPACK, EISPACK, LAPACK and BLAS
- control:
    - dsubsp and exchnqz.
    - rpoly.
    - lybsc, lydsr, lybad,sydsr and sybad.

[70] http://www.scilab.org/platform/index_platform.php?page=acknowledgement

- sszer.
- syhsc.
- rilac, ricd.
- dexpm1, pade, dclmat, coef, cerr, wexpm1, wpade, wclmat.
- polmc.
- bdiag.
- ereduc,fstair.
- a selection of SLICOT routines issued from NICONET european project work.

- Cumulative Distribution: DCDFLIB.
- ODE, DAE:

  - lsode, lsodar; lsodi, lsoda.
  - dassl
  - daspk, daskr
  - colnew: (boundary problems).
- Networks and graphs: routines for network analysis.
- Optimization:

  - non linear optimization routines.
  - Linear and quadratic programming solver: routines/optim/plcbas.f.
  - Semidefinite programming
  - zero of nonlinear functions: hybrd.

- Signal processing: routines.
- Sparse matrix:

  - sparse Lu factorization and resolution comes from Sparse 1.3.
  - sparse Cholesky factorization codes.
  - routines to read .mps file (sparse linear programming).
  - sparse matrix eigenvalue; eigenvector computation comes from ARPACK.

- Random Number Generator: randlib,

- Statistics functions and a few others: ACM algorithms numbers 330, 493, 523, 582, 595, 597, 599, 632, 678, 708, 715 and 750.
- Bessel functions are based on the Slatec library.
- qsort: qsort.c
- Scipad: an embedded text editor derived.
- Scilab includes f2c from AT&T Bell Laboratories and Bellcore to compile fortran programs under Windows.
- Command aquisition: zzledt.c .
- Sounds: SoX (Lance Norskog, Chris Bagwell) has been adapted to read and write sound files.
- Data encoding: Scilab uses XDR (Sun Microsystems, Inc.) for file exchange data encoding .
- Parallel computing: interface routines with PVM.
- Gui:

    - Unix, based on Xaw and Xmu routines (from MIT X11 distribution) and derived from xterm, xxgdb, xfig codes.
    - Windows, derived from gnuplot/gsview.
    - TCL/TK interface.
    - Scilab uses GD to produce bitmaps outputs for its graphics.
    - A tcl/tk environment used for the help browser, waitbar and editvar.

- FreeBSD port by The FreeBSD Community.
- The Scilab functions for reading Excel files use ripOLE code (from PLDaniels Software) to extract Excel streams out of .xls files.
- Scilab Windows binary version uses Inno Setup: a free installer for Windows programs.
- PDE Scicos palette.
- Linked list (list.c & list.h).

Επιπλέον ένα βασικό πλεονέκτημα του προγράμματος είναι η δυνατότητα που παρέχει στον χρήστη να προσθέτει νέα στοιχεία μέσω των toolboxes, τα οποία είναι plugin. Ωστόσο η ποιότητά των toolboxes ποικίλει πολύ και η

συστηματοποίησή τους υστερεί. Τα διαθέσιμα toolboxes του προγράμματος είναι:

- 2-D and 3-D graphics, animation
- Linear algebra, sparse matrices
- Polynomials and rational functions
- Interpolation, approximation
- Simulation: ODE solver and DAE solver
- Scicos: a hybrid dynamic systems modeler and simulator
- Classic and robust control, LMI optimization
- Differentiable and non-differentiable optimization
- Signal processing
- Metanet: graphs and networks
- Parallel Scilab
- Statistics
- Interface with Computer Algebra: Maple package for Scilab code generation
- Interface with Fortran, Tcl/Tk, C, C++, Java, LabVIEW
- And a large number of contributions for various domains.

Ανά κατηγορία[71]:

GUI TOOLS	IMAGE TOOLS	FINITE ELEMENTS TOOLS
OPTIMIZATION TOOLS	SCICOS	DATA HANDLING TOOLS
MODELING AND CONTROL TOOLS	DATA ANALYSIS AND STATISTICS	CONTRIBUTED SCILAB BINARIES
EDITOR STYLES	MANUALS	DATA ACQUISITION AND REAL-TIME
MATH TOOLS	SPECIALIZED TOOLBOXES	OTHER TOOLS

[71] http://www.scilab.org/contrib/index_contrib.php?page=downlaod&category

II. ΒΟΗΘΕΙΑ (HELP)

1. Gauss

Α. Μενού Help[72]

Ο χρήστης μέσω του μενού Help μπορεί να αποκτήσει πρόσβαση στις πληροφορίες του συστήματος Help του προγράμματος. Ειδικότερα το μενού Help περιλαμβάνει τις ακόλουθες εντολές:

- User's Guide (Οδηγός χρήστη): παρέχει πρόσβαση στον online οδηγό χρήστη του προγράμματος,

- Reference (Παραπομπή): παρέχει πρόσβαση σε online γλωσσική παραπομπή (language reference), η οποία περιέχει την σύνταξη κάθε εντολής του προγράμματος (οι δύο αυτές βοήθειες είναι αποθηκευμένες σε αρχεία PDF και παρέχεται η δυνατότητα εκτύπωσης, κατόπιν συνεννόησης με την Aptech Systems)

- Keyboard (Πληκτρολόγιο): παρέχει πρόσβαση στην λίστα των πλήκτρων που αντιστοιχούν στις κινήσεις του κέρσορα, στην επεξεργασία (editing) και την επιλογή κειμένου (text selection),

- Tip of the Day (Συμβουλή της ημέρας): προβάλλει συμβουλές για την καλύτερη χρήση των συμβόλων που είναι διαθέσιμα στα διαδραστικά παράθυρα του προγράμματος και

- About Gauss (Σχετικά με το Gauss): παρέχει πληροφορίες σχετικά με την έκδοση του προγράμματος που χρησιμοποιείται, με το είδος της άδειας που έχει ο χρήστης και τον αριθμό ταυτοποίησής του καθώς και πληροφορίες προστασίας πνευματικών δικαιωμάτων.

Β. Context – sensitive Help

Για ταχύτερα αποτελέσματα, η Help του προγράμματος είναι ρυθμισμένη να ανταποκρίνεται, χωρίς ο χρήστης να χρειάζεται να πλοηγηθεί στο σύστημά της, απλώς πατώντας F1 (context – sensitive Help). Αρκεί δηλαδή να θέσει ο χρήστης τον κέρσορα πάνω στην λέξη κλειδί και πατώντας F1, θα μεταφερθεί στο αντίστοιχο χωρίο της Help για την λέξη αυτή. Οι περιοχές του προγράμματος που διαθέτουν αυτή την λειτουργία είναι: τα παράθυρα,

[72] Aptech Systems Inc: User Guide, Version 9.0, July 2008, κεφ. 9, σελ 9-1 επ.

τα κουμπιά της εργαλειοθήκης, τα μενού και η γλώσσα προγραμματισμού. Διαφορετικά, αν απλώς πατήσει F1, θα αποκτήσει πρόσβαση στην εισαγωγή του οδηγού χρήστη του προγράμματος (User's Guide). Για πιο αναλυτική περιγραφή κάθε εντολής, ο χρήστης θα πρέπει να ανατρέξει στην language reference του προγράμματος.

Γ. Άλλοι τρόποι υποστήριξης

Schift + F1: πληκτρολογώντας Schift + F1 ή πατώντας το «βέλος + ?» στην εργαλειοθήκη, ο κέρσορας παίρνει την μορφή «βέλος + ?» και ανάλογα αν το στοιχείο που ενδιαφέρει τον χρήστη είναι context – sensitive ή όχι, μεταφέρεται είτε στο αντίστοιχο χωρίο της Help για το στοιχείο αυτό είτε εμφανίζεται το μενού της Help.

Ctrl + F1: θέτοντας τον κέρσορα πάνω στο στοιχείο που ενδιαφέρει τον χρήστη και πατώντας Ctrl + F1, ο χρήστης μπορεί να ψάξει σε όλες τις ενεργές βιβλιοθήκες για οποιοδήποτε παγκόσμιο σύμβολο. Το πρόγραμμα αναζητά σε όλες τις ενεργές βιβλιοθήκες το αρχείο, στο οποίο περιέχεται το αναζητούμενο σύμβολο. Αν βρεθεί, το αρχείο που περιέχει τον πηγαίο κώδικα ανοίγει σε ένα παράθυρο επεξεργασίας, Αν το αρχείο περιέχει το string **>όνομα_συμβόλου στην αρχή μιας σειράς του σχολιασμένου κώδικα, τότε ο κέρσορας θα βρεθεί στην αρχή αυτής της σειράς. Αντίθετα, αν δεν βρεθεί το string, τότε ο κέρσορας θα βρεθεί στην αρχή αυτού του αρχείου.

Δ. ToolTip

Εύχρηστη είναι επίσης η λειτουργία ToolTip (συμβουλή για τα εργαλεία), η οποία προσφέρει μια μικρή περιγραφή της λειτουργίας κάθε συμβόλου του προγράμματος, εφόσον ο κέρσορας βρεθεί πάνω στο αντίστοιχο σύμβολο.

Ε. Γενική υποστήριξη

Περαιτέρω, ο χρήστης έχει την δυνατότητα να επικοινωνήσει με άλλους χρήστες του προγράμματος μέσω του Gaussians mail list, για την οποία υπάρχουν αναλυτικές πληροφορίες στην ιστοσελίδα του προγράμματος http://www.aptech.com, στο πεδίο Resource Library. Επίσης ο χρήστης που

έχει εν ισχύ Premier Support, μπορεί να απευθυνθεί και στο support@aptech .com.

2. Maple[73]

Ο χρήστης που έρχεται για πρώτη φορά σε επαφή με το πρόγραμμα μπορεί μέσω του μενού της Help, να αποκτήσει μια πρώτη εικόνα για την λειτουργία και τα χαρακτηριστικά του προγράμματος. Με άλλα λόγια μπορεί:

- να κάνει μια εικονική περιήγηση στο πρόγραμμα μέσω του Help> Take a Tour of Maple,

- να διαβάσει τα εγχειρίδια οδηγιών του προγράμματος μέσω του Help> Manuals, Dictionary and More> Manuals,

- να συμβουλευτεί την λίστα με τις βασικές εντολές και έννοιες του προγράμματος μέσω του Help> Quick Help ή εναλλακτικά πατώντας F1

- να συμβουλευτεί τον πίνακα με τις εντολές και τις πληροφορίες για νέους χρήστες, που εμφανίζεται σε ξεχωριστό παράθυρο και περιέχει υπερσυνδέσμους στις αντίστοιχες σελίδες της Help, όπου και παρέχονται περισσότερες πληροφορίες, μέσω του Help> Quick Reference ή εναλλακτικά Ctrl+F2 (Command+ F2 για Macintosh).

Περαιτέρω όταν ο χρήστης χρησιμοποιεί το πρόγραμμα, μπορεί να αναζητήσει πληροφορίες για την ακριβή σύνταξη των εντολών, παραδείγματα, λεπτομέρειες για τις μεθόδους που χρησιμοποιεί το σύστημα καθώς και νέα στοιχεία για κάθε έκδοση του προγράμματος μέσω της Help. Ειδικότερα μπορεί να ανατρέξει στην Help και εναλλακτικά:

- να προβεί σε αναζήτηση ανά θέμα, επιλέγοντας από το μενού της εργαλειοθήκης Help>Search for>Topic,

- να προβεί σε αναζήτηση βάσει λέξεων-κλειδιά, επιλέγοντας από το μενού της εργαλειοθήκης Help>Search for>Text,

- να πληκτρολογήσει "?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

[73] Maplesoft: Maple User Manual, 2008, κεφ. 1.9 σελ. 33

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.

Επιπλέον, ο χρήστης μπορεί να έχει πρόσβαση :

- σε λεξικό μαθηματικών και μηχανικής, με χιλιάδες ορισμούς, σύμβολα και γραφήματα μέσω της Help>Manuals, Dictionary and more>Dictionary

- σε μια λίστα εντολών με θεματικές ενότητες, τις οποίες μπορεί να χρησιμοποιήσει για να εκτελέσει απευθείας μια εφαρμογή, μέσω του Tools>Tasks>Browse και για το οποίο μπορεί να πάρει πληροφορίες , μέσω του Help>Manuals, Dictionary and more>Tasks.

3. Mathematica[74]

Ο χρήστης μπορεί να αποκτήσει πρόσβαση στο documentation του προγράμματος με ποικίλους τρόπους, ανάλογα με τα δεδομένα που έχει στην διάθεσή του:

- αν γνωρίζει το όνομα της συνάρτησης αλλά αγνοεί την σύνταξή της ή την ακριβή λειτουργία της, ο ευκολότερος τρόπος είναι να πληκτρολογήσει ? και το όνομα της συνάρτησης,

- αν γνωρίζει το όνομα της εντολής αλλά αγνοεί την ορθογραφία της, τότε μπορεί να πληκτρολογήσει ? και μέρος της εντολής (για το οποίο είναι σίγουρος) ακολουθούμενο από το * (π.χ. ?Integ* όταν αναζητά την εντολή Integrate), οπότε εμφανίζεται μια λίστα όλων των σχετικών ενσωματωμένων στο πρόγραμμα εντολών.

Αξιοσημείωτο είναι το γεγονός ότι ο χρήστης μπορεί να φτιάξει το δικό του documentation για τις δικές του συναρτήσεις και να προγραμματίσει με τέτοιον τρόπο, ώστε οι εκάστοτε χρήστες να έχουν πρόσβαση σε αυτό (το documentation), ακριβώς με τον ίδιο τρόπο, όπως και στο αντίστοιχο των ενσωματωμένων στο βασικό πρόγραμμα συναρτήσεων.

4. Matlab[75]

[74] http://reference.wolfram.com/mathematica/guide/Mathematica.html

Α. Ο φυλλομετρητής Help

Το MATLAB έχει ενσωματωμένες πληροφορίες για τις εντολές του, τις μεταβλητές, τις συναρτήσεις και πολλά άλλα στοιχεία. Ο χρήστης έχει την δυνατότητα να βρει όλες αυτές τις πληροφορίες μέσα από την επιλογή Help.

Ο φυλλομετρητής Help είναι ένα HTML viewer, ενσωματωμένο στην επιφάνεια εργασίας του προγράμματος. Για να ανοίξει ο χρήστης την επιλογή Help έχει τρεις εναλλακτικές:

- να κάνει κλικ στο αντίστοιχο εικονίδιο με το "?" στην εργαλειοθήκη,

- να επιλέξει από το μενού Help > Product Help,

- να πληκτρολογήσει την λέξη doc στην γραμμή εντολών.

Η Help αποτελείται από δύο παράθυρα: α) τον πλοηγητή Help, τον οποίο χρησιμοποιεί ο χρήστης για να βρει τις πληροφορίες που ψάχνει και β) το παράθυρο παρουσίασης, όπου προβάλλονται οι πληροφορίες.

Ειδικότερα, ο πλοηγητής Help διαθέτει τις ακόλουθες επιλογές:

- το πεδίο Search for: ο χρήστης μπορεί να αναζητήσει λέξεις τόσο στο online documentation όσο και στα Demos (δοκιμαστικά), είτε:

- ορίζοντας συγκεκριμένες λέξεις-κλειδιά, τις οποίες γράφει εντός εισαγωγικών π.χ. "word1 word2",

- χρησιμοποιώντας το * στην θέση γραμμάτων π.χ. wo*d1,

- θέτοντας τελεστές Boolean (λογικούς) ανάμεσα στις λέξεις-κλειδιά π.χ. word1 NOT word2,

- την επιλογή Search Results: παρουσιάζει τα αποτελέσματα της αναζήτησης (όπως περιγράφηκε ανωτέρω), διαχωρίζοντας τα αποτελέσματα του documentation από αυτά των Demos,

- την επιλογή Demos: ο χρήστης βλέπει και μπορεί να τρέξει δοκιμαστικά των προϊόντων της MathWorks και να μάθει τα βασικά στοιχεία της λειτουργίας τους και των εργαλείων τους. Περιλαμβάνουν και κώδικα, που μπορεί να αποτελέσει τη βάση για την δημιουργία m-αρχείων από τον χρήστη,

- την επιλογή Contents (περιεχόμενα): ο χρήστης βλέπει τους τίτλους και των πίνακα περιεχομένων του documentation. Εξ ορισμού, η επιλογή συγχρονίζεται με την σελίδα που παρουσιάζεται. Εάν ο χρήστης βλέπει μια

[75] http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/matlab_prog/f10-60119.html

σελίδα, ως αποτέλεσμα αναζήτησης ή ακολουθώντας κάποιον σύνδεσμο, κάνοντας κλικ στην επιλογή Contents βλέπει το περιεχόμενο της σελίδας, όπως αυτό περιγράφεται στο σύνολο του documentation.

- την επιλογή Index: ο χρήστης βρίσκει συγκεκριμένες λέξεις-κλειδιά στο documentation.

Καθώς ο χρήστης συμβουλεύεται το documentation, μπορεί

- να φυλλομετρά άλλες σελίδες
- να ορίζει την σελίδα στα "Αγαπημένα"
- να εκτυπώνει την σελίδα
- να βρίσκει όρους στην σελίδα
- να αντιγράφει ή να εκτελεί ένα συγκεκριμένο κομμάτι του.

Β. Φυλλομετρητής Help και συναρτήσεις

Όταν ο χρήστης αναζητά πληροφορίες για τις συναρτήσεις του προγράμματος έχει τις ακόλουθες δύο εναλλακτικές μέσω του φυλλομετρητή Help:

- πληκτρολογώντας στην γραμμή εντολών doc και το όνομα της συνάρτησης π.χ. doc format → ο χρήστης βρίσκει documentation για την συνάρτηση format είτε

- επιλέγοντας Contents, ανοίγει την επιλογή Matlab, η οποία περιέχει τις ακόλουθες δύο υποεπιλογές

- Συναρτήσεις – Λίστα κατηγοριών
- Συναρτήσεις – Αλφαβητική Λίστα.

Γ. Help για συναρτήσεις από την γραμμή εντολών.

Ο χρήστης έχει τις ακόλουθες επιλογές από την γραμμή εντολών αναφορικά με το Help:

- να πληκτρολογήσει Help, οπότε θα εμφανιστούν όλες οι κατηγορίες για τις οποίες μπορεί να αναζητήσει πληροφορίες,

- να πληκτρολογήσει Help και το όνομα μιας κατηγορίας, οπότε θα εμφανιστεί μια λίστα με τις συναρτήσεις της κατηγορίας αυτής καθώς και μια σύντομη περιγραφή του documentation κάθε επιμέρους συνάρτησης,

- να πληκτρολογήσει Help και το όνομα μιας συνάρτησης, οπότε θα εμφανιστεί το documentation της. Το όνομα της συνάρτησης είναι σε

κεφαλαία γράμματα ώστε να ξεχωρίζει από το υπόλοιπο κείμενο. Εδώ πρέπει να σημειωθεί ότι ο χρήστης πρέπει να πληκτρολογεί την αναζητούμενη συνάρτηση με μικρά γράμματα αλλιώς δεν θα εκτελεστεί η εντολή. Στο κείμενο της Help φαίνεται ξεκάθαρα ποια συνάρτηση γράφεται με κεφαλαία, ποια με μικρά και ποια με συνδυασμό και των δύο, γεγονός που πρέπει να τηρεί και ο χρήστης.

Δ. Help και topics

Ο χρήστης μπορεί να λάβει πληροφορίες για κάθε επιμέρους topic πληκτρολογώντας στην γραμμή εντολών Help και το αντίστοιχο όνομα του topic που τον ενδιαφέρει (π.χ. Datatypes, java, debug).

Ε. Εναλλακτικοί τρόποι

Διαφορετικά ο χρήστης μπορεί να απευθυνθεί για βοήθεια στην Τεχνική Υποστήριξη (www.mathworks.com\support) και να συμμετέχει στο Usenet newsgroup for Matlab users (comp.soft-sys.matlab) στην Matlab Central.

Στ. Δημιουργώντας την Help

Όταν ο χρήστης προγραμματίζει χρησιμοποιώντας Matlab, μπορεί να δημιουργήσει την δική του Help. Ειδικότερα, μπορεί να ξεκινήσει κάποιο πρόγραμμα με ένα τμήμα του κειμένου, που θα παρέχει βοήθεια αναφορικά με το πώς και το πότε θα χρησιμοποιείται μια συνάρτηση ή μια υποσυνάρτηση (subfunction) ή μια προσωπική συνάρτηση (private function). Το κείμενο θα προβάλλεται μέσω της εντολής help και το όνομα της συνάρτησης ή της υποσυνάρτησης ή της προσωπικής συνάρτησης.

Ζ. Help και methods και overloaded συναρτήσεις.

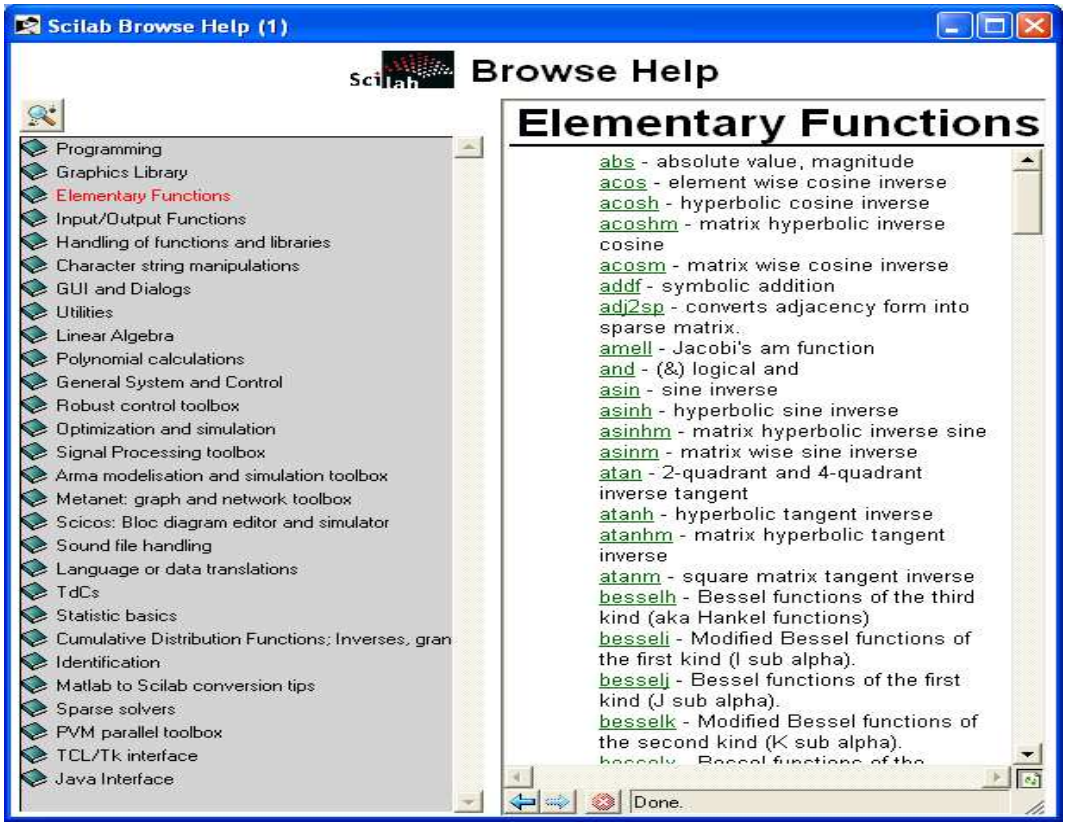
Ο χρήστης μπορεί να γράψει το κείμενο Help για αντικειμενοστραφείς μεθόδους κλάσης, που ενσωματώνονται στα m-αρχεία. Το κείμενο θα εμφανιστεί πληκτρολογώντας help και το όνομα της κλάσης/μεθόδου, π.χ. help classname/methodname. Επίσης το ίδιο μπορεί να κάνει και για overloaded Matlab συναρτήσεις.

Η. Γενικά

Επειδή τα αποτελέσματα που θα προκύψουν μπορεί να είναι περισσότερα από αυτά που μπορούν να εμφανιστούν ταυτόχρονα στην οθόνη, ο χρήστης χρησιμοποιεί την εντολή more on, ώστε να εμφανίζονται ανά σελίδα. Ο χρήστης επανέρχεται στην προηγούμενη κατάσταση πληκτρολογώντας more off.

5. Scilab[76]

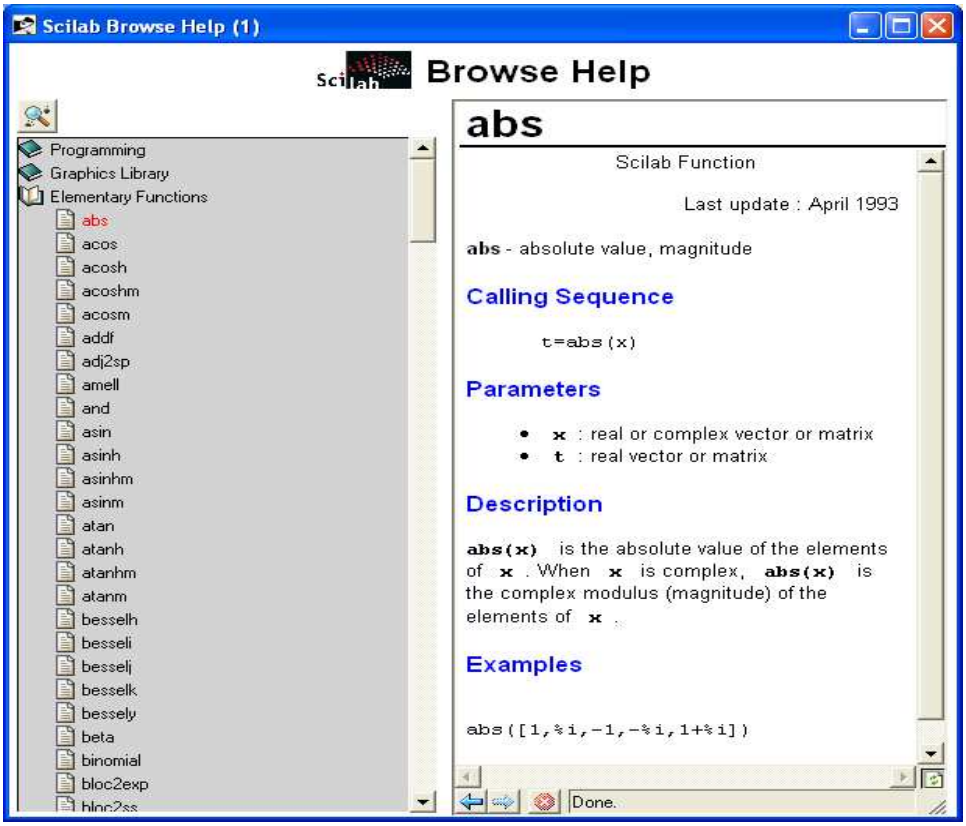
Ο χρήστης μπορεί να αποκτήσει πρόσβαση στην ηλεκτρονική βοήθεια του προγράμματος είτε πατώντας το κουμπί με το λατινικό ερωτηματικό (?) στην γραμμή εργαλείων του παραθύρου, είτε μπαίνοντας στο help μιας συνάρτησης. Έστω ότι ο χρήστης πατά το κουμπί με το λατινικό ερωτηματικό (?) στην γραμμή εργαλείων του παραθύρου, οπότε εμφανίζεται η οθόνη που ακολουθεί:



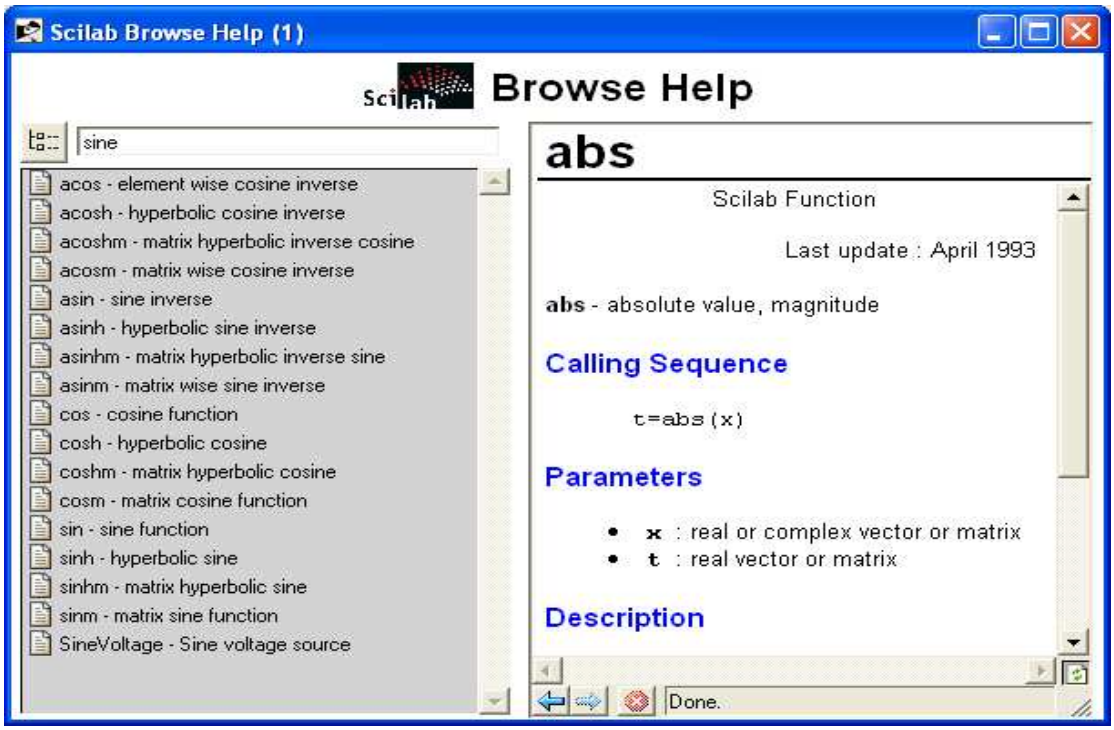
[76]Haugen, Finn: Master Scilab! 2008 διαθέσιμο σε: http://home.hit.no/~finnh/scilab_scicos/scilab/index.htm#help

Όπως φαίνεται, οι εντολές και οι συναρτήσεις είναι οργανωμένες σε ποικίλες κατηγορίες. Για να βρει ο χρήστης περισσότερες πληροφορίες για μια συγκεκριμένη συνάρτηση:

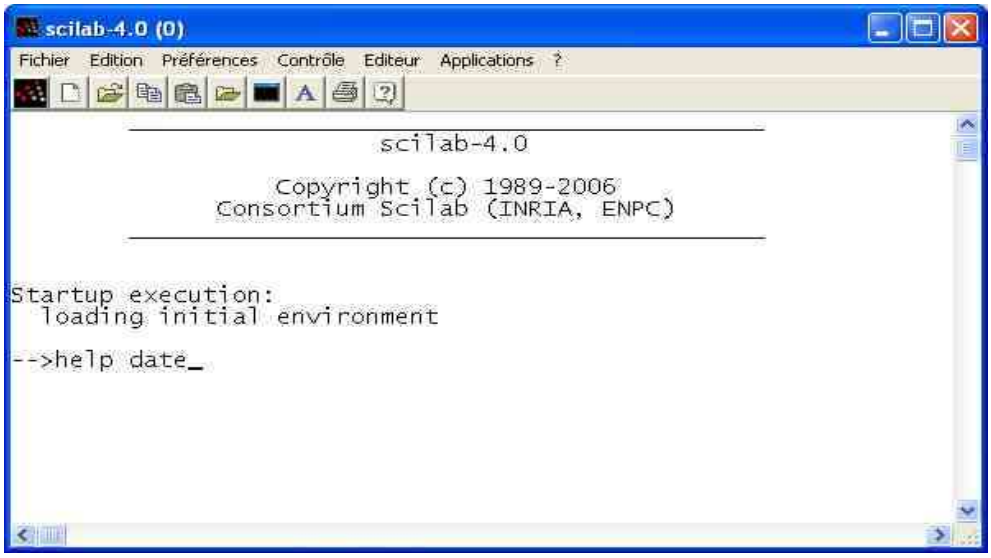
- είτε κάνει κλικ πάνω σε αυτήν. Για παράδειγμα κάνει κλικ στην συνάρτηση *abs* στην κατηγορία Βασικές Συναρτήσεις οπότε του εμφανίζεται η παρακάτω οθόνη:



- είτε κάνει κλικ πάνω στο κουμπί αναζήτηση (ο μεγεθυντικός φακός στην οθόνη αμέσως παραπάνω). Για παράδειγμα αν αναζητά την συνάρτηση *sine* πληκτρολογεί την λέξη στο κουτάκι που εμφανίζεται και προκύπτει η ακόλουθη οθόνη:



Ο χρήστης μπορεί επίσης εισάγοντας την λέξη-κλειδί στο κουτί apropos ενός αρχείου whatis να βρει την πληροφορία που τον ενδιαφέρει. Επιπλέον μπορεί εύκολα να προσθέτει νέα στοιχεία ή λέξεις – κλειδιά, δημιουργώντας ένα .cat ASCII αρχείο (όπου θα περιγράφει το στοιχείο που θέλει να εισάγει) και ένα whatis αρχείο στο ευρετήριό του. Κατόπιν, στην μεταβλητή %help προσθέτει την διαδρομή του ευρετηρίου του (και έναν τίτλο), χρησιμοποιώντας την βασική μορφοποίηση του εγχειριδίου του προγράμματος. Το εγχειρίδιο Scilab LATEX είναι αυτομάτως προσβάσιμο από τα στοιχεία του εγχειριδίου με ένα Makefile. Όταν βρίσκεται σε ένα παράθυρο εργασίας μπορεί να πληκτρολογήσει την λέξη help ακολουθούμενη από την λέξη - κλειδί που τον ενδιαφέρει και να βρει όλες τις σχετικές πληροφορίες.



Ειδικότερα, όταν το πρόγραμμα φορτώνεται σε περιβάλλον windows τότε παρέχεται στον χρήστη η δυνατότητα online Help η οποία είτε είναι ενσωματωμένη στο πρόγραμμα είτε μπορεί να την κατεβάσει είτε να αποκτήσει πρόσβαση σε αυτήν μέσω του διαδικτύου

ΚΕΦΑΛΑΙΟ ΠΕΜΠΤΟ

ΣΥΓΚΡΙΣΗ

Δεδομένα της σύγκρισης: Για την σύγκριση των παραπάνω CAS χρησιμοποιήθηκαν 8 συναρτήσεις (matrix inverse, matrix multiplication, transpose matrix, matrix rank, matrix determinant, matrix norm, matrix LU, matrix trace) σε τυχαίους πίνακες με διαστάσεις που ξεκινούσαν από 500x500 και συνέχιζαν με βήμα 250 έως 2000x2000[77].

Τα προγράμματα έτρεξαν σε υπολογιστή με επεξεργαστή 2.0 GHz Intel® Core™ 2 Duo processor T7200 με Intel® Centrino® Duo Mobile Technology, μνήμη 2 GB DDR2 667 MHz και λειτουργικό Microsoft Windows XP Home Edition.

Πίνακας 1:

Τα αποτελέσματα εμφανίζονται σε δευτερόλεπτα και δηλώνουν τον χρόνο που χρειάστηκε κάθε πρόγραμμα για τον υπολογισμό της κάθε συνάρτησης σε κάθε διάσταση. Η σκιασμένη στήλη στο Gauss υποδηλώνει ότι το πρόγραμμα δεν διαθέτει έτοιμη συνάρτηση για τον υπολογισμό της.

			TIMING				
1	**Inverse**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500	0.219	0.469	0.047	0.203	0.078	
	750x750	0.735	1.375	0.125	0.375	0.172	
	1000x1000	1.812	2.891	0.328	0.516	0.500	
	1250x1250	3.547	5.438	0.594	1.047	0.891	
	1500x1500	6.141	8.156	0.922	1.656	1.563	
	1750x1750	9.953	13.235	1.500	2.703	2.563	
	2000x2000	14.969	18.031	2.203	3.984	3.813	
2	**Multiplication**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500	0.235	0.531	0.016	0.047	-0.017	
	750x750	0.719	1.250	0.016	0.141	0.125	
	1000x1000	1.687	4.188	0.031	0.328	0.313	
	1250x1250	3.328	7.203	0.016	0.672	0.594	
	1500x1500	5.735	10.937	0.031	1.094	1.000	
	1750x1750	9.031	19.672	0.078	1.766	1.609	
	2000x2000	13.516	28.500	0.078	2.625	2.406	
3	**Transpose**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500	0.000	0.000	0.016	0.000	-0.094	

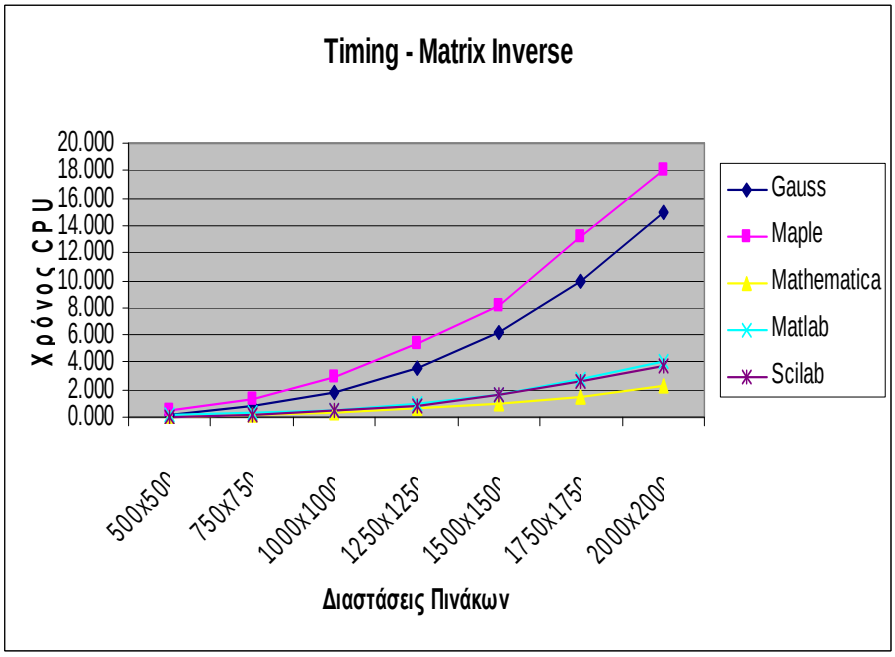
[77] Στο Παράρτημα δίνονται αναλυτικά οι κώδικες που χρησιμοποιήθηκαν για την σύγκριση.

	750x750	0.000	0.000	0.000	0.016	0.000	
	1000x1000	0.016	0.016	0.016	0.016	0.016	
	1250x1250	0.016	0.016	0.000	0.016	0.016	
	1500x1500	0.031	0.031	0.032	0.031	-0.016	
	1750x1750	0.047	0.046	0.046	0.047	0.016	
	2000x2000	0.063	0.062	0.063	0.047	0.031	
4	**Rank**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500	0.328	0.546	0.156	0.359	-0.219	
	750x750	1.094	1.266	0.360	0.922	0.578	
	1000x1000	3.812	2.875	1.172	2.297	2.234	
	1250x1250	8.766	5.672	2.625	4.438	4.922	
	1500x1500	16.016	9.906	5.016	7.781	9.484	
	1750x1750	26.047	15.844	8.313	12.172	15.828	
	2000x2000	39.312	25.922	12.687	17.969	24.563	
5	**Determinant**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500	0.079	1.125	0.016	0.031	-0.172	
	750x750	0.312	2.484	0.047	0.078	0.078	
	1000x1000	1.266	4.422	0.094	0.172	0.125	
	1250x1250	3.109	9.766	0.188	0.297	0.234	
	1500x1500	5.906	13.828	0.281	0.563	0.391	
	1750x1750	9.781	17.954	0.375	0.859	0.672	
	2000x2000	15.015	27.484	0.671	1.203	1.078	
6	**Norm**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500	0.015	1.407	0.125	0.359	-0.031	
	750x750	0.000	5.672	0.359	0.984	0.547	
	1000x1000	0.000	7.453	1.203	2.391	2.203	
	1250x1250	0.000	11.032	2.766	4.609	4.984	
	1500x1500	0.000	17.922	5.125	7.813	9.531	
	1750x1750	0.000	26.125	8.265	12.188	15.797	
	2000x2000	0.000	39.843	12.922	18.109	24.656	
7	**LU**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500	0.156	1.062	0.031	0.063	-0.172	
	750x750	0.703	4.829	0.078	0.063	0.125	
	1000x1000	2.047	6.281	0.125	0.172	0.234	
	1250x1250	4.562	9.797	0.281	0.344	0.438	
	1500x1500	8.515	13.875	0.360	0.516	0.688	
	1750x1750	14.578	22.468	0.547	0.891	1.047	
	2000x2000	21.922	33.250	0.812	1.406	1.484	
8	**Trace**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500		0.000	0.000	0.000	-0.172	
	750x750		0.000	0.000	0.000	-0.016	
	1000x1000		0.000	0.000	0.000	-0.031	
	1250x1250		0.000	0.000	0.000	-0.047	
	1500x1500		0.000	0.000	0.000	-0.063	
	1750x1750		0.000	0.000	0.000	-0.156	
	2000x2000		0.000	0.000	0.000	-0.141	

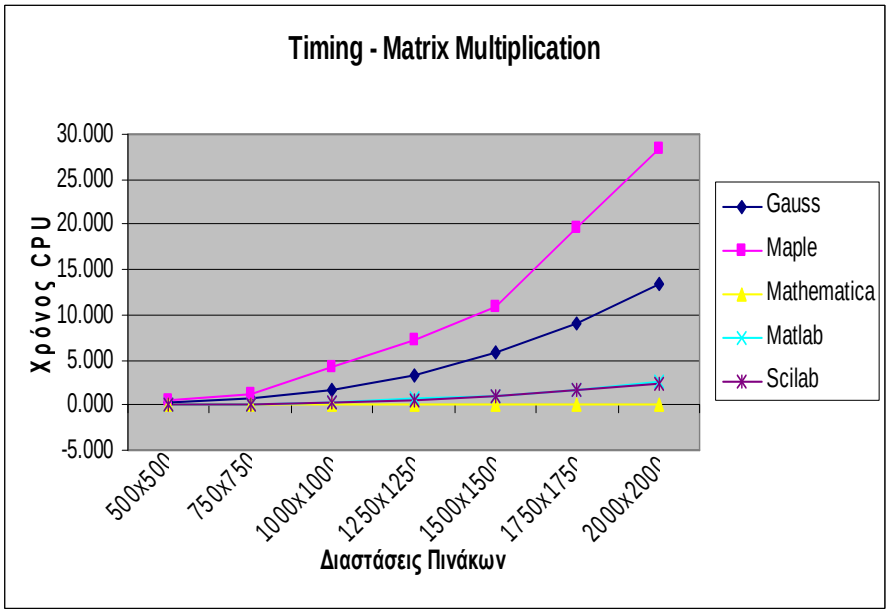
Πίνακας 1

Γραφήματα:

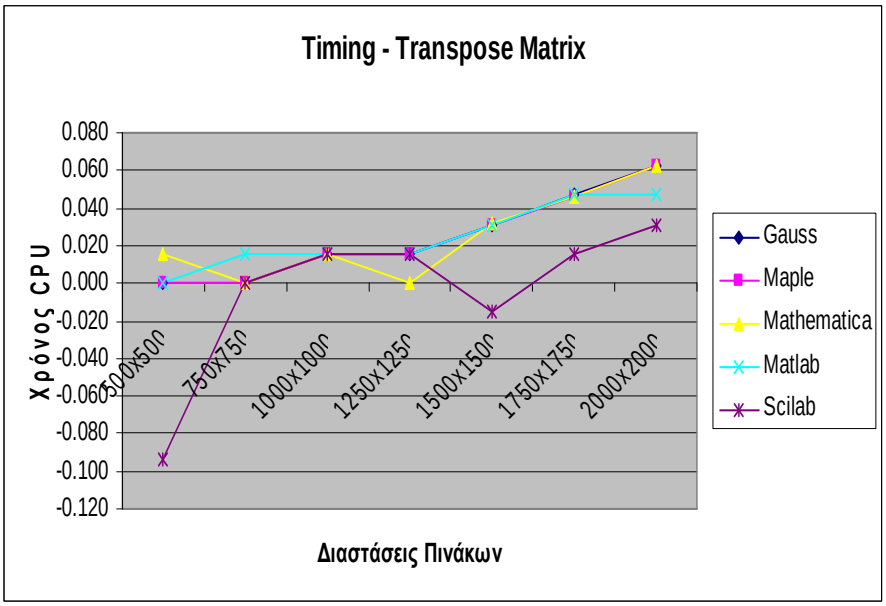
Τα ακόλουθα 8 γραφήματα (1.1 – 1.8) αναπαριστούν τους χρόνους που χρειάστηκε κάθε πρόγραμμα στις διαστάσεις των τυχαίων πινάκων που μελετήθηκαν.



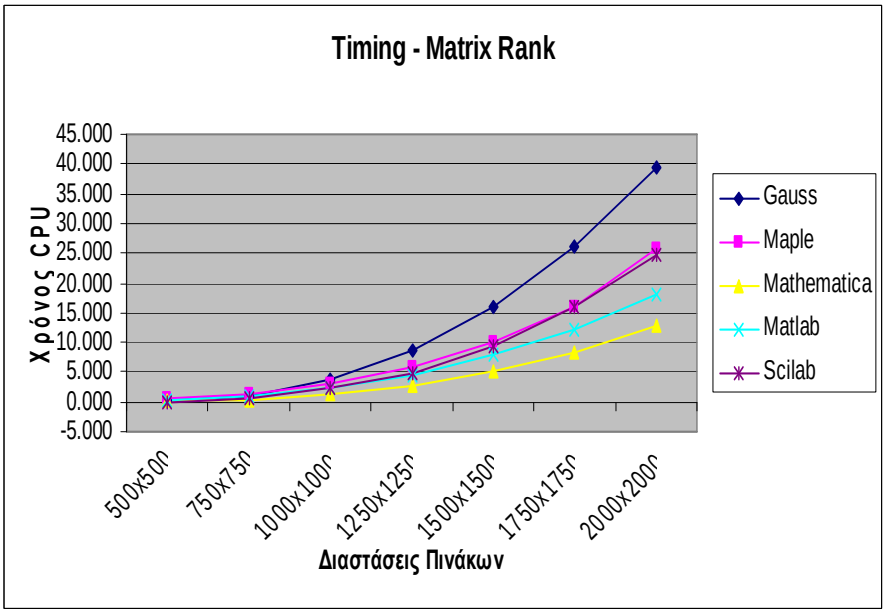
Γράφημα 1.1



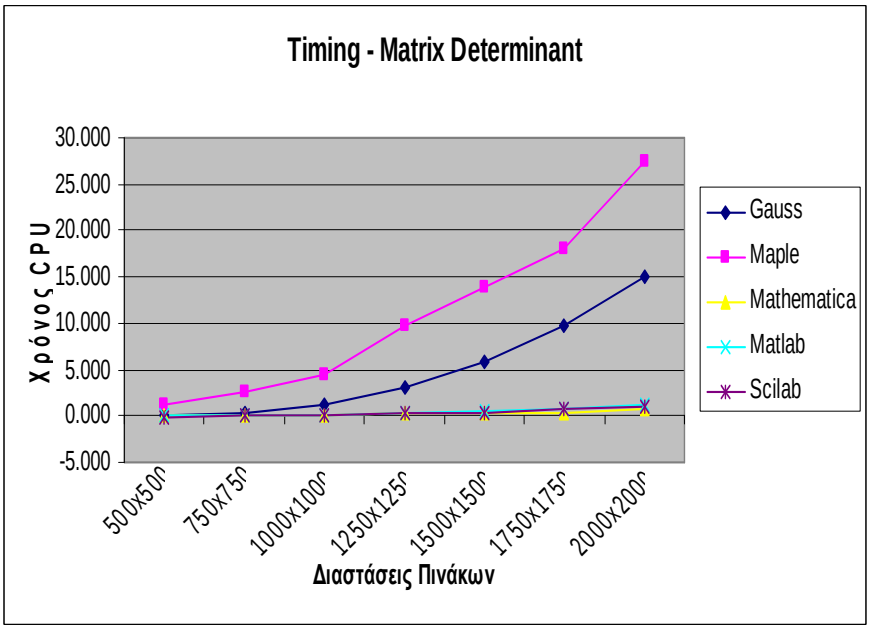
Γράφημα 1.2



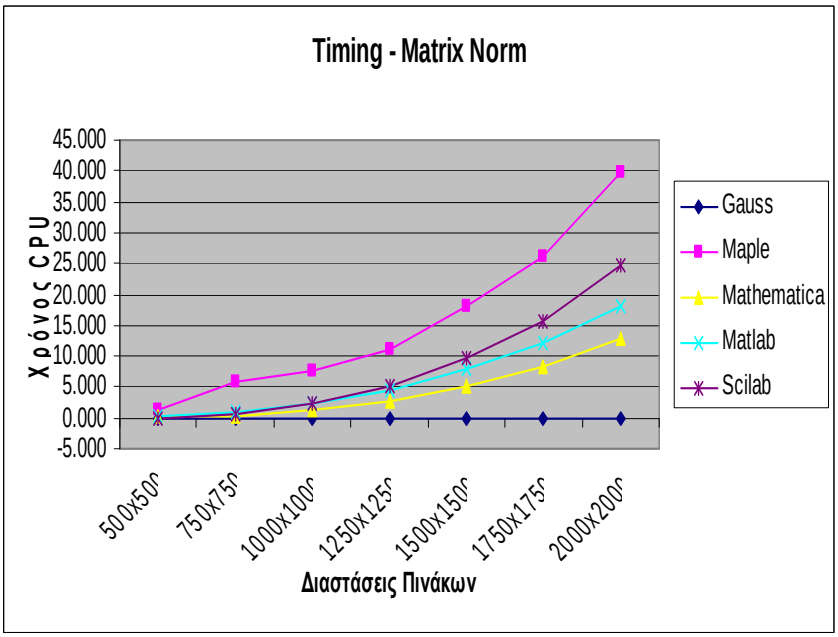
Γράφημα 1.3



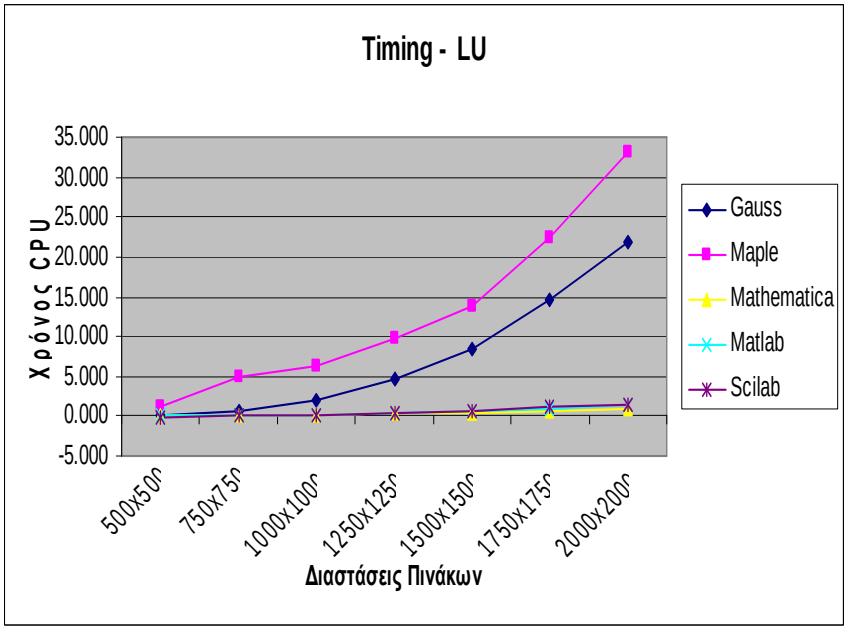
Γράφημα 1.4



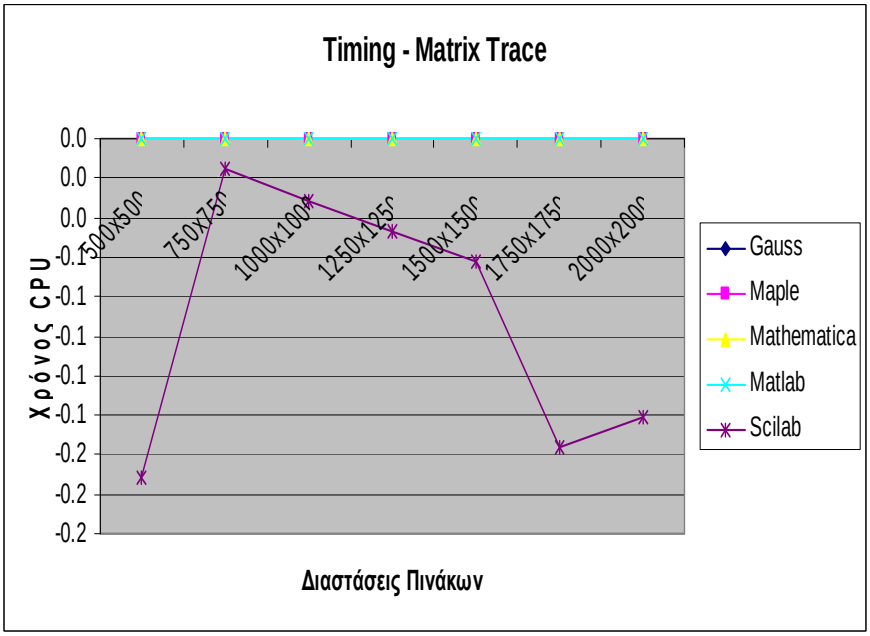
Γράφημα 1.5



Γράφημα 1.6



Γράφημα 1.7



Γράφημα 1.8

Πίνακας 2:

Το σύμβολο + υποδηλώνει την απουσία αποτελέσματος. Η σκιασμένη στήλη στο Gauss υποδηλώνει ότι το πρόγραμμα δεν διαθέτει έτοιμη συνάρτηση για τον υπολογισμό της. Η μονάδα (1.0) αντιπροσωπεύει το πρόγραμμα με τον καλύτερο χρόνο. Τα επιμέρους αποτελέσματα αφορούν στο κάτα πόσο βραδύτερο είναι καθένα από τα υπόλοιπα προγράμματα σε σχέση με το ταχύτερο, δηλαδή την μονάδα (1.0).

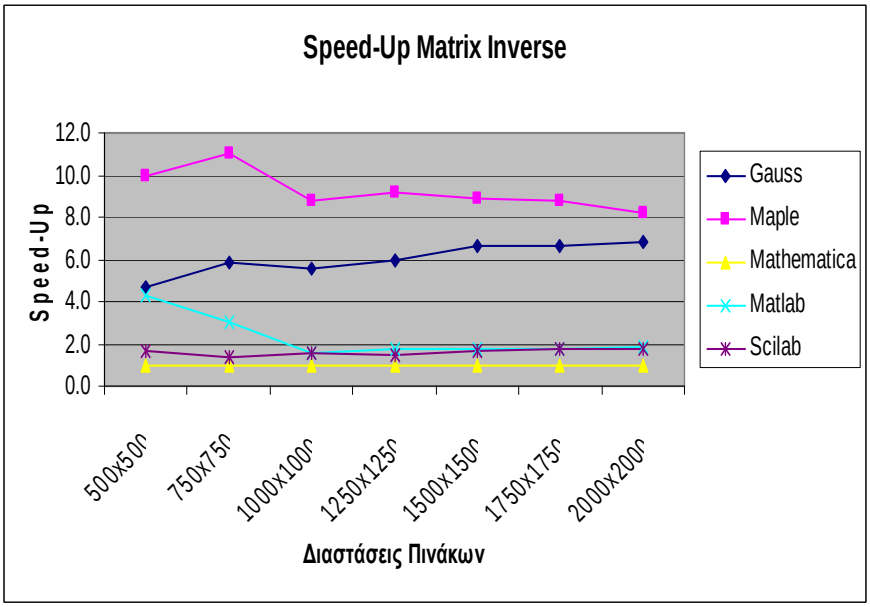
			SPEED – UP				
1	**Inverse**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500	4.7	10.0	1.0	4.3	1.7	
	750x750	5.9	11.0	1.0	3.0	1.4	
	1000x1000	5.5	8.8	1.0	1.6	1.5	
	1250x1250	6.0	9.2	1.0	1.8	1.5	
	1500x1500	6.7	8.8	1.0	1.8	1.7	
	1750x1750	6.6	8.8	1.0	1.8	1.7	
	2000x2000	6.8	8.2	1.0	1.8	1.7	
2	**Multiplication**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500	13.8	31.2	0.9	2.8	1.0	
	750x750	44.9	78.1	1.0	8.8	7.8	
	1000x1000	54.4	135.1	1.0	10.6	10.1	
	1250x1250	208.0	450.2	1.0	42.0	37.1	
	1500x1500	185.0	352.8	1.0	35.3	32.3	
	1750x1750	115.8	252.2	1.0	22.6	20.6	
	2000x2000	173.3	365.4	1.0	33.7	30.8	
3	**Transpose**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500	0.0	0.0	0.2	0.0	1.0	
	750x750	1.0	1.0	1.0	+	1.0	
	1000x1000	1.0	1.0	1.0	1.0	1.0	
	1250x1250	+	+	1.0	+	+	
	1500x1500	1.9	1.9	2.0	2.0	1.0	
	1750x1750	3.0	2.9	2.9	3.0	1.0	
	2000x2000	2.0	2.0	2.0	1.5	1.0	
4	**Rank**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500	1.5	2.5	0.7	1.6	1.0	
	750x750	3.0	3.5	1.0	2.6	1.6	
	1000x1000	3.3	2.5	1.0	2.0	1.9	
	1250x1250	3.3	2.2	1.0	1.7	1.9	
	1500x1500	3.2	2.0	1.0	1.6	1.9	

	1750x1750	3.1	1.9	1.0	1.5	1.9	
	2000x2000	3.1	2.0	1.0	1.4	1.9	
5	**Determinant**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500	0.5	6.5	0.1	0.2	1.0	
	750x750	6.6	52.9	1.0	1.7	1.7	
	1000x1000	13.5	47.0	1.0	1.8	1.3	
	1250x1250	16.5	51.9	1.0	1.6	1.2	
	1500x1500	21.0	49.2	1.0	2.0	1.4	
	1750x1750	26.1	47.9	1.0	2.3	1.8	
	2000x2000	22.4	41.0	1.0	1.8	1.6	
6	**Norm**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500	0.5	45.4	4.0	11.6	1.0	
	750x750	1.0	+	+	+	+	
	1000x1000	1.0	+	+	+	+	
	1250x1250	1.0	+	+	+	+	
	1500x1500	1.0	+	+	+	+	
	1750x1750	1.0	+	+	+	+	
	2000x2000	1.0	+	+	+	+	
7	**LU**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500	0.9	6.2	0.2	0.4	1.0	
	750x750	11.2	77.3	1.2	1.0	2.0	
	1000x1000	16.4	50.2	1.0	1.4	1.9	
	1250x1250	16.2	34.9	1.0	1.2	1.6	
	1500x1500	23.7	38.5	1.0	1.4	1.9	
	1750x1750	26.7	41.1	1.0	1.6	1.9	
	2000x2000	27.0	40.9	1.0	1.7	1.8	
8	**Trace**	**Gauss**	**Maple**	**Mathematica**	**Matlab**	**Scilab**	
	500x500		0.0	0.0	0.0	1.0	
	750x750		0.0	0.0	0.0	1.0	
	1000x1000		0.0	0.0	0.0	1.0	
	1250x1250		0.0	0.0	0.0	1.0	
	1500x1500		0.0	0.0	0.0	1.0	
	1750x1750		0.0	0.0	0.0	1.0	
	2000x2000		0.0	0.0	0.0	1.0	

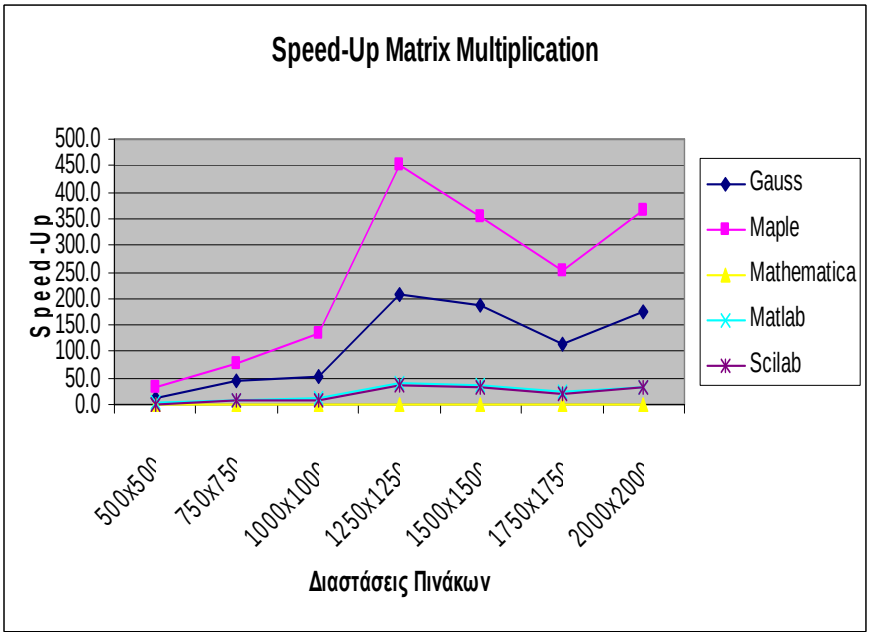
Πίνακας 2

Γραφήματα:

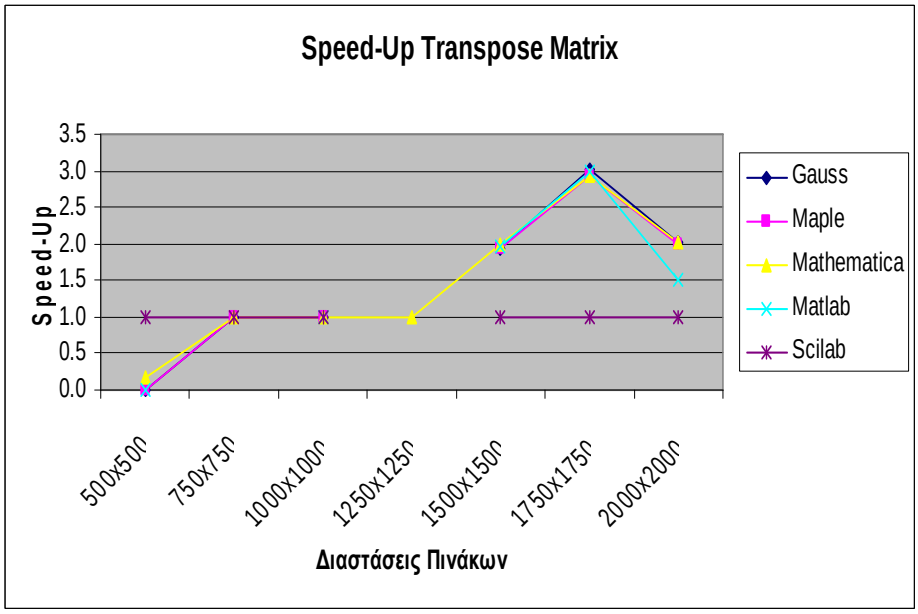
Τα ακόλουθα 8 γραφήματα (2.1 – 2.8) αναπαριστούν τα αποτελέσματα του πίνακα 2, δηλαδή για κάθε συνάρτηση κατά πόσο καθένα πρόγραμμα είναι βραδύτερο από το ταχύτερο πρόγραμμα.



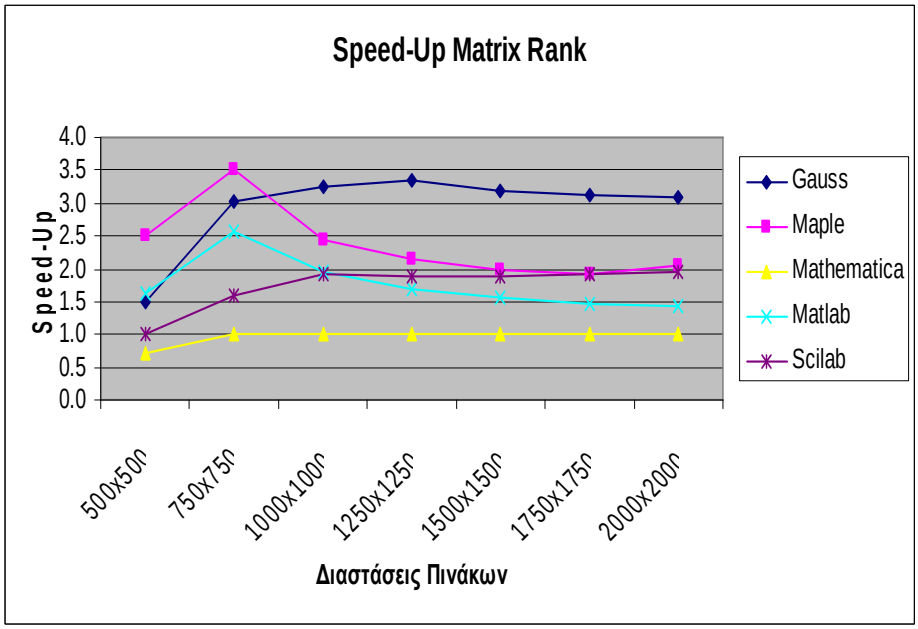
Γράφημα 2.1



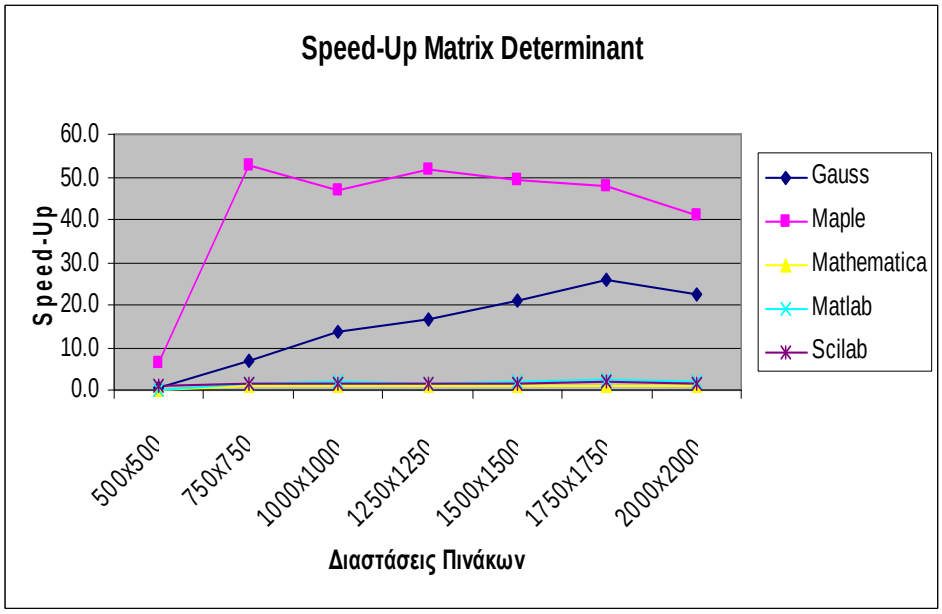
Γράφημα 2.2



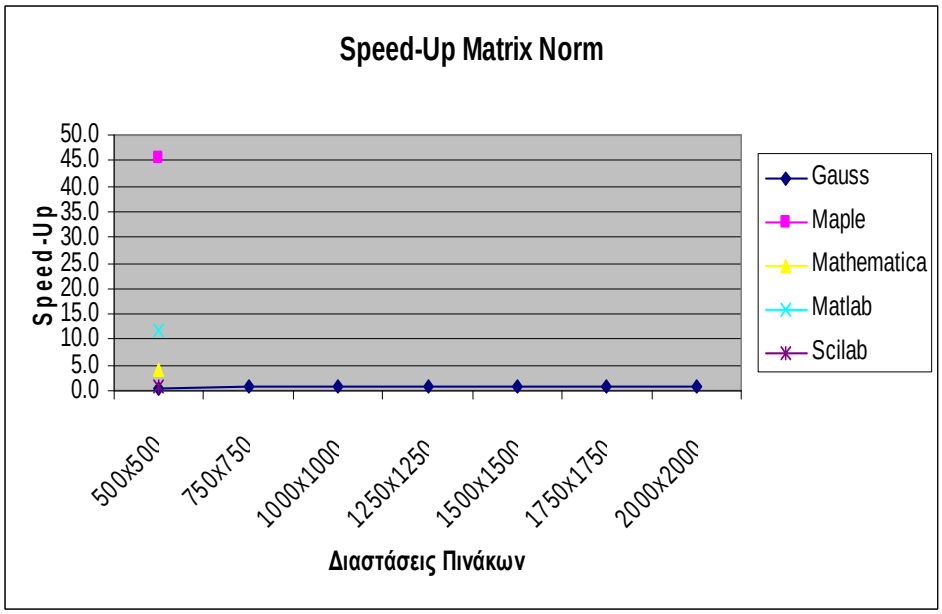
Γράφημα 2.3



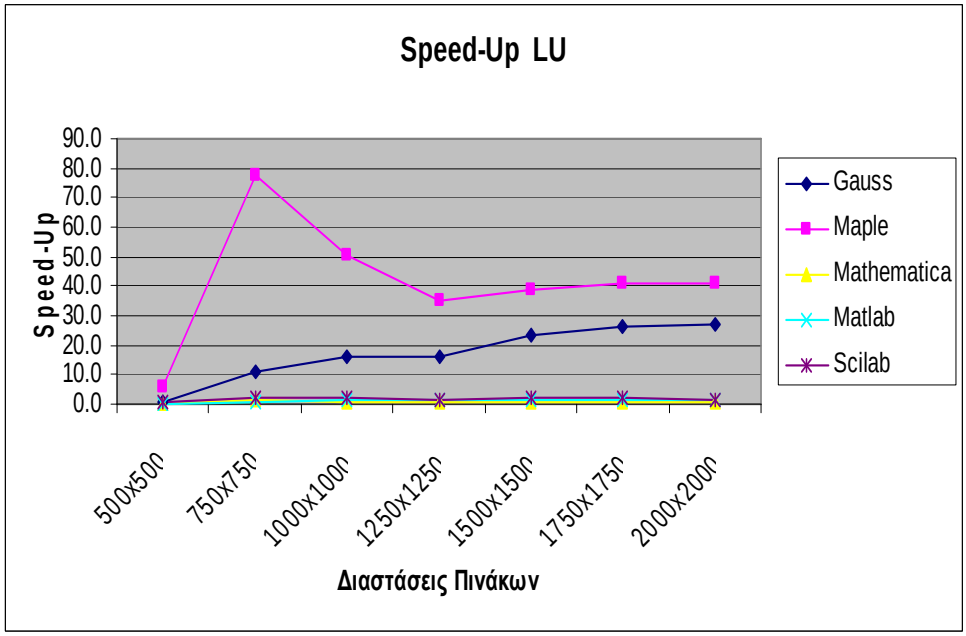
Γράφημα 2.4



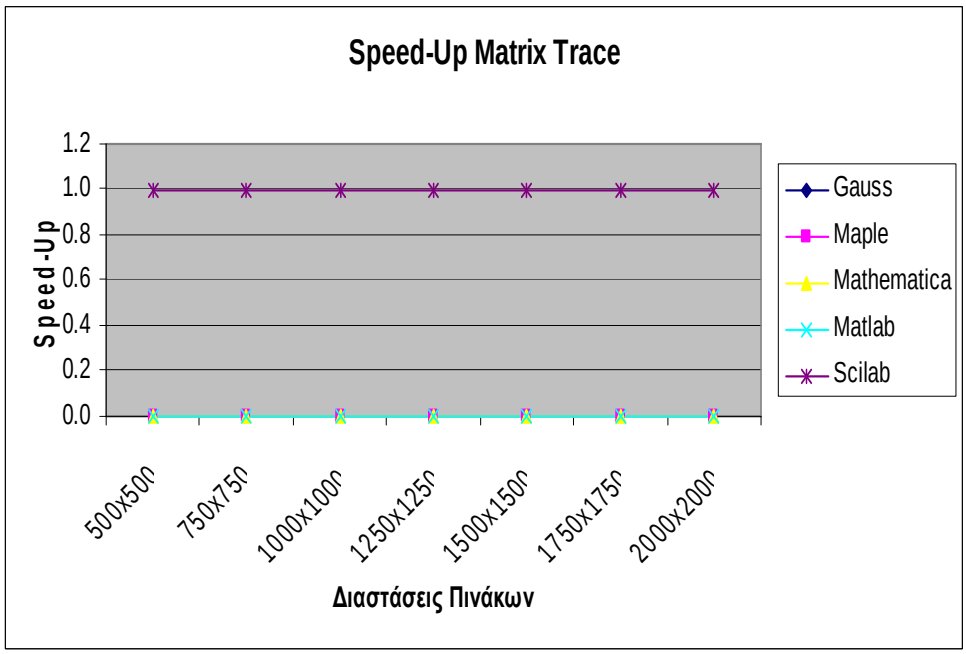
Γράφημα 2.5



Γράφημα 2.6



Γράφημα 2.7



Γράφημα 2.8

ΣΥΜΠΕΡΑΣΜΑΤΑ

Κατά την διαδικασία της συγκριτικής αξιόλογησης δεν αντιμετωπίστηκε κανένα αξιοσημείωτο πρόβλημα. Όλα τα προγράμματα διαθέτουν έτοιμες τις συναρτήσεις που χρησιμοποιήθηκαν, πλην του Gauss, το οποίο δεν διαθέτει την συνάρτηση matrix trace. Συνεπώς ήταν απλή η δημιουργία του κώδικα για τον υπολογισμό των συναρτήσεων.

Όπως προκύπτει αναλυτικά και από τους παραπάνω πίνακες 1 και 2 και τα σχετικά γραφήματα (1.1 – 1.8 και 2.1 – 2.8), όλα τα προγράμματα βραδύνουν καθώς αυξάνεται η διάσταση του τυχαίου πίνακα. Βάσει των αποτελεσμάτων της σύγκρισης ταχύτερο πρόγραμμα είναι το Mathematica, ακολουθεί το Scilab χωρίς όμως μεγάλες διαφορές με το επόμενο που είναι το Matlab, έπεται το Gauss και τελευταίο το Maple.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Α. Εισαγωγή

- http://www.nationmaster.com/encyclopedia/Computer-algebra-system#History
- http://www.math.wpi.edu/IQP/BVCalcHist/calc5.html
- http://www.onpedia.com/encyclopedia/Computer-algebra-system
- http://personales.unican.es/iglesias/CASA2006/

Β. Gauss

- http://www.aptech.com/gauss.html
- http://en.wikipedia.org/wiki/GAUSS
- http://www.aae.wisc.edu/aae637/gausscode.htm
- *Goertzen, Simon:* The Gauss Package Manual. Extented Gauss Functionality for GAP. Version 2009.01.05. January 2009.
- *Aptech Systems Inc:* User Guide, Version 9.0, July 2008.
- *Aptech Systems Inc:* Gauss Language Reference, Version 9.0, July 2008.
- *Rossi, Eduardo:* Introduction to Gauss Programming Language. University of Pavia, October 2006.
- *Aptech Systems Inc:* GAUSSplot, Professional Graphics Cookbook, Powered by Tecplot, 2005.
- *Ritchie, Felix:* Programming in GAUSS, 09.10.2002 διαθέσιμο σε: http://www.trigconsulting.co.uk/gauss/man_basics.html
- *του ιδίου:* GAUSS A Beginner's Guide. Department of Economics University of Stirling, February 1994, (last revised) April 1997.

Γ. Maple

- http://www.maplesoft.com/products/Maple/features/index.aspx
- http://en.wikipedia.org/wiki/Maple_(software)
- *Maplesoft*: Maple User Manual, 2008.
- *Product Specifications* διαθέσιμο σε: www.maplesoft.com/view.aspx?SF.../Maple12_Product_Spec.pdf
- *Ματζάκος, Ν.:* Εισαγωγή στο Maple. Εκδόσεις Νέων Τεχνολογιών. Αθήνα 2007.

- *του ιδίου*: Συστήματα Συστήματα Συμβολικής Συμβολικής Άλγεβρας Άλγεβρας (Computer Algebra System Μέρος 1ο, διαθέσιμο σε: http://dtps.unipi.gr/files/notes/2006- 2007/eksamino_1/grammikh_algebra/cas-maple1.pdf
- *Monagan, M. B./Geddes, K. O./ Heal, K. M./Labahn, G./Vorkoetter S. M./ McCarron J./DeMarco P.:* Maple Introductory Programming Guide. Maplesoft 2005.
- *Garvan, Frank*: The Maple book. Chapman & Hall/CRC, 2002.
- *Graham, David I.*: Review of Maple 7. University of Plymouth.
- *Monagan, Michael:* Programming in Maple: The Basics διαθέσιμο σε: http://theor.jinr.ru/Documents/MapleV/program/program.html

Δ. Mathematica

- http://en.wikipedia.org/wiki/Mathematica
- http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/index.html
- http://reference.wolfram.com/mathematica/guide/ImportingAndExporting.html
- *MacMahon, David/Topa, Daniel M.*: A Beginner's Guide to Mathematica. Chapman and Hall/CRC, Boca Raton, London, New York 2006.
- *Wellin, Paul R./Gaylord, Richard J./Kamin, Samuel N.:* An Introduction to Programming with Mathematica. Third Edition, Cambridge 2005.
- *Liska Richard, Drska Ladislav, Limpouch Jiri, Sinor Milan, Wester Michael, Winkler Franz*: Υπολογιστική Άλγεβρα Αλγόριθμοι Συστήματα και Εφαρμογές: Μετάφραση: Μαρκομανώλης Γεώργιος (6954), Μπουκουβάλα Βασιλική (6969), Μυλωνάς Θεοδόσιος (6974), Παπαδάτος Μάρκος (6486), Ιωάννινα 2003.
- *Trott, Michael*: The Mathematica guidebook for programming. New York, 2004
- *Torrence, Bruce F./Torrence, Eve A.:* The student's introduction to Mathematica. A handbook for Precalculus, Calculus, and Linear Algebra. Cambridge University Press, Cambridge 1999.

E. Matlab

- http://people.bath.ac.uk/masigg/a_manual.pdf
- http://en.wikipedia.org/wiki/MuPAD
- *Hart, David (1η έκδ)/Wolfe, Clinton (αναθ.): Βάβαλη Μ./Κατελανή, Τ.* (απόδοση στα ελληνικά): Ξεκινώντας με το MATLAB. Πανεπιστήμιο Κρήτης, Τμήμα Μαθηματικών.
- *The MathWorks:* Matlab 7 Getting Started Guide, 2008.
- *The MathWorks:* Matlab 7 Programming Tips, 2008.
- *Hahn, Brian D./Valentine, Daniel T.:* Essential Matlab for Engineers and Scientists. Third Edition, Elsevier 2007.
- *Graham, Ivan*: Matlab Manual and Introductory Tutorials with some revisions by Nick Britton, Mathematical Sciences, University of Bath 2005.
- *Παπαρρίζος, Κωνσταντίνος:* MATLAB 6.5. Εκδόσεις Ζυγός, Θεσσαλονίκη 2004.
- *Recktenwald, Gerald W.:* Matlab Programming. Department of Mechanical Engineering Portland State University 2001.
- *Στεφανίδης, Γιώργος/Σαμαράς, Νικόλαος:* Υπολογιστικές μέθοδοι με το MATLAB. Εκδόσεις Ζυγός, Θεσσαλονίκη 1999.
- *Gkioulekas Eleftherios*: Programming with Matlab. Mathematical Sciences Computing Center University of Washington, December 1996.
- *Neuman, Edward*: Programming in MATLAB, Tutorial 2 διαθέσιμο σε: http://www.math.siu.edu/matlab/tutorial2.pdf

ΣΤ. Scilab

- http://www.scilab.org
- http://en.wikipedia.org/wiki/Scilab
- Scilab Manual, διαθέσιμο σε: http://www.scilab.org/download/index_download.php?page=documentation
- Introduction to Scilab, User's Guide διαθέσιμο σε: http://www.scilab.org/doc/intro/index.html
- http://www.spas.cnrs-gif.fr/ScilabPasapas-Ch10Sc01.html
- *Haugen, Finn:* Master Scilab! 2008 διαθέσιμο σε: http://home.hit.no/~finnh/scilab_scicos/scilab/index.htm#help

- *Graeme, Chandler/Stephen Roberts:* Introduction to Scilab 2002 διαθέσιμο σε: http://comptlsci.anu.edu.au/Scilab/primer.pdf
- *Urroz, Gilberto E.:* Programming with SCILAB. 2002, διαθέσιμο σε: http://studenti.ptfos.hr/Preddiplomski_studij/Mjerenje_i_upravljanje_procesima/ProgrammingWithSCILAB.pdf
- *του ιδίου:* Comparison of SCILAB Syntax and Functions to MATLAB. 2001, διαθέσιμο σε *infoclearinghouse.com*.

ΠΑΡΑΡΤΗΜΑ

Παρακάτω παρατίθενται οι κώδικες που χρησιμοποιήθηκαν για τον υπολογισμό των Matrix Inverse, Matrix Multiplication, Transpose Matrix, Matrix Rank, Matrix Determinant, Matrix Norm, Matrix LU και Matrix Trace ώστε να εξαχθούν τα αποτελέσματα της σύγκρισης των προγραμμάτων βάσει των παραμέτρων που είχαν τεθεί.

A. Gauss

1. Matrix Inverse

```
local zeit, timing;
timing=0;
zeit=hsec;
for i(500,2000,250);
        A=rndn(i,i);
        timing=(hsec-zeit)/100;
        B=inv(A);
        timing1=((hsec-zeit)/100)-timing;
        print timing1;
endfor;
```

2. Matrix Multiplication

```
local zeit, timing;
timing=0;
zeit=hsec;
for i(500,2000,250);
        A=rndn(i,i);
        B=rndn(i,i);
        timing=(hsec-zeit)/100;
        C=A*B;
        timing1=((hsec-zeit)/100)-timing;
        print timing1;
endfor;
```

3. Transpose Matrix

```
local zeit, timing;
timing=0;
zeit=hsec;
for i(500,2000,250);
        A=rndn(i,i);
        timing=(hsec-zeit)/100;
        B=A ';
        timing1=((hsec-zeit)/100)-timing;
        print timing1;
endfor;
```

4. Matrix Rank

```
local zeit, timing;
timing=0;
zeit=hsec;
for i(500,2000,250);
        A=rndn(i,i);
        timing=(hsec-zeit)/100;
        k=rank(A);
        timing1=((hsec-zeit)/100)-timing;
        print timing1;
endfor;
```

5. Matrix Determinant

```
local zeit, timing;
timing=0;
zeit=hsec;
for i(500,2000,250);
        A=rndn(i,i);
        timing=(hsec-zeit)/100;
        d=det(A);
        timing1=((hsec-zeit)/100)-timing;
        print timing1;
endfor;
```

6. Matrix Norm

```
local zeit, timing;
timing=0;
zeit=hsec;
for i(500,2000,250);
        A=rndn(i,i);
        timing=(hsec-zeit)/100;
        proc infnorm(A);
        retp(maxc(sumc(abs(A '))));
        endp;
        timing1=((hsec-zeit)/100)-timing;
        print timing1;
endfor;
```

7. Matrix LU

```
local zeit, timing;
timing=0;
zeit=hsec;
for i(500,2000,250);
        A=rndn(i,i);
        timing=(hsec-zeit)/100;
        {l,u}=lu(A);
        timing1=((hsec-zeit)/100)-timing;
        print timing1;
endfor;
```

8. Matrix Trace

το πρόγραμμα δεν διαθέτει έτοιμη συνάρτηση για τον υπολογισμό της

B. Maple

1. Matrix Inverse

```
with(LinearAlgebra):
for i from 500 by 250 to 2000 do
    A:=RandomMatrix(i, generator=-0.5..0.5);
    t:=time();
    B:=MatrixInverse(A);
    t1:=time()-t;
    print(t1);
end do:
```

2. Matrix Multiplication

```
with(LinearAlgebra):
for i from 500 by 250 to 2000 do
    A:=RandomMatrix(i,generator=-0.5..0.5);
    B:=RandomMatrix(i, generator=-0.5..0.5);
    t:=time();
    C:=A. B;
    t1:=time()-t;
    print(t1);
end do:
```

3. Transpose Matrix

```
with(LinearAlgebra):
for i from 500 by 250 to 2000 do
    A:=RandomMatrix(i,generator=-0.5..0.5);
    t:=time();
    B:=Transpose(A);
    t1:=time()-t;
    print(t1);
end do:
```

4. Matrix Rank

```
with(LinearAlgebra):
```

```
for i from 500 by 250 to 2000 do
    A:=RandomMatrix(i, generator=-0.5..0.5);
    t:=time();
    B:=Rank(A);
    t1:=time()-t;
    print(t1);
end do:
```

5. Matrix Determinant

```
with(LinearAlgebra):
for i from 500 by 250 to 2000 do
    A:=RandomMatrix(i,generator=-0.5..0.5);
    t:=time();
    B:=Determinant(A);
    t1:=time()-t;
    print(t1);
end do:
```

6. Matrix Norm

```
with(LinearAlgebra):
for i from 500 by 250 to 2000 do
    A:=RandomMatrix(i,generator=-0.5..0.5);
    t:=time();
    B:=norm(A);
    t1:=time()-t;
    print(t1);
end do:
```

7. Matrix LU

```
with(LinearAlgebra):
for i from 500 by 250 to 2000 do
    A:=RandomMatrix(i,generator=-0.5..0.5);
    t:=time();
    B:=LUDecomposition(A);
```

```
        t1:=time()-t;
        print(t1);
end do:
```

8. Matrix Trace

```
with(LinearAlgebra):
for i from 500 by 250 to 2000 do
        A:=RandomMatrix(i,generator=-0.5..0.5);
        t:=time();
        B:=Trace(A);
        t1:=time()-t;
        print(t1);
end do:
```

Γ. Mathematica

1. Matrix Inverse

For[i=500,i<2001,i=i+250,a=RandomReal[{-1,1},{i,i}];t=TimeUsed[];b=Inverse[a];t1=TimeUsed[]-t;Print[t1]]

2. Matrix Multiplication

*For[i=500,i<2001,i=i+250,a=RandomReal[{-1,1},{i,i}];b=RandomReal[{-1,1},{i,i}];t=TimeUsed[];c=a*b;t1=TimeUsed[]-t;Print[t1]]*

3. Transpose Matrix

For[i=500,i<2001,i=i+250,a=RandomReal[{-1,1},{i,i}];t=TimeUsed[];b=Transpose[a];t1=TimeUsed[]-t;Print[t1]]

4. Matrix Rank

For[i=500,i<2001,i=i+250,a=RandomReal[{-1,1},{i,i}];t=TimeUsed[];k=MatrixRank[a];t1=TimeUsed[]-t;Print[t1]]

5. Matrix Determinant

For[i=500,i<2001,i=i+250,a=RandomReal[{-1,1},{i,i}];t=TimeUsed[];d=Det[a];t1=TimeUsed[]-t;Print[t1]]

6. Matrix Norm

For[i=500,i<2001,i=i+250,a=RandomReal[{-1,1},{i,i}];t=TimeUsed[];n=Norm[a];t1=TimeUsed[]-t;Print[t1]]

7. Matrix LU

For[i=500,i<2001,i=i+250,a=RandomReal[{-1,1},{i,i}];t=TimeUsed[];LUDecomposition[a];t1=TimeUsed[]-t;Print[t1]]

8. Matrix Trace

For[i=500,i<2001,i=i+250,a=RandomReal[{-1,1},{i,i}];t=TimeUsed[];z=Tr[a];t1=TimeUsed[]-t;Print[t1]]

Δ. Matlab

1. Matrix Inverse

```
for i=500:250:2000
        A=rand(i,i);
        t=cputime;
        B=inv(A);
        t1=cputime-t
end
```

2. Matrix Multiplication

```
for i=500:250:2000
        A=rand(i,i);
        B=rand(i,i);
        t=cputime;
        C=A*B;
        t1=cputime-t
end
```

3. Transpose Matrix

```
for i=500:250:2000
        A=rand(i,i);
        t=cputime;
        B= A';
        t1=cputime-t
end
```

4. Matrix Rank

```
for i=500:250:2000
        A=rand(i,i);
        t=cputime;
        k=rank(A);
        t1=cputime-t
end
```

5. Matrix Determinant

```
for i=500:250:2000
        A=rand(i,i);
        t=cputime;
        d=det(A);
        t1=cputime-t
end
```

6. Matrix Norm

```
for i=500:250:2000
        A=rand(i,i);
        t=cputime;
        n=norm(A);
        t1=cputime-t
end
```

7. Matrix LU

```
for i=500:250:2000
        A=rand(i,i);
        t=cputime;
        [L,U]=lu(A);
        t1=cputime-t
end
```

8. Matrix Trace

```
for i=500:250:2000
        A=rand(i,i);
        t=cputime;
        z=trace(A);
        t1=cputime-t
end
```

E. Scilab

1. Matrix Inverse

```
stacksize('max')
for i=500:250:2000
        A=rand(i,i);
        t=timer();
        B=inv(A);
        t1=timer()-t
end
```

2. Matrix Multiplication

```
stacksize('max')
for i=500:250:2000
        A=rand(i,i);
        B=rand(i,i);
        t=timer();
        C=A*B;
        t1=timer()-t
end
```

3. Transpose Matrix

```
stacksize('max')
for i=500:250:2000
        A=rand(i,i);
        t=timer();
        B= A';
        t1=timer()-t
end
```

4. Matrix Rank

```
stacksize('max')
for i=500:250:2000
        A=rand(i,i);
        t=timer();
```

```
        k=rank(A);
        t1=timer()-t
end
```

5. Matrix Determinant

```
stacksize('max')
for i=500:250:2000
        A=rand(i,i);
        t=timer();
        d=det(A);
        t1=timer()-t
end
```

6. Matrix Norm

```
stacksize('max')
for i=500:250:2000
        A=rand(i,i);
        t=timer();
        n=norm(A);
        t1=timer()-t
end
```

7. Matrix LU

```
stacksize('max')
for i=500:250:2000
        A=rand(i,i);
        t=timer();
        [L,U]=lu(A);
        t1=timer()-t
end
```

8. Matrix Trace

```
stacksize('max')
for i=500:250:2000
```

```
        A=rand(i,i);
        t=timer();
        z=trace(A);
        t1=timer()-t
end
```