



ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

“ΘΕΩΡΗΤΙΚΗ ΠΛΗΡΟΦΟΡΙΚΗ & ΘΕΩΡΙΑ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΛΕΓΧΟΥ”

Ανάπτυξη διαδραστικού περιβάλλοντος (GUI) για την Ανάλυση και Σύνθεση Πολυμεταβλητών Συστημάτων Αυτομάτου Ελέγχου

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτριος Ν. Βαρσάμης

Επιβλέπων: Νικόλαος Καραμπετάκης
Επικ. Καθηγητής Α.Π.Θ.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

.....
Ν. Καραμπετάκης
Επικ. Καθηγητής Α.Π.Θ.

.....
Α. Βαρδουλάκης
Καθηγητής Α.Π.Θ.

.....
Μ. Γουσίδου
Επικ. Καθηγήτρια Α.Π.Θ.

Θεσσαλονίκη, Απρίλιος 2006

.....
Δημήτριος Ν. Βαρσάμης

Πτυχιούχος Μαθηματικός Πανεπιστημίου Ιωαννίνων

Copyright © Δημήτριος Ν. Βαρσάμης, 2006.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευτεί ότι εκφράζουν τις επίσημες θέσεις του Α.Π.Θ.

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία έχει ως στόχο τη χρησιμοποίηση του πολυωνυμικού πακέτου POLYX, που έχει εφαρμογή στην ανάλυση και σύνθεση πολυμεταβλητών συστημάτων αυτομάτου ελέγχου, με την βοήθεια ενός διαδραστικού περιβάλλοντος. Η ανάπτυξη του διαδραστικού περιβάλλοντος υλοποιήθηκε με την χρήση του graphical user interface του λογισμικού πακέτου MATLAB. Το πλεονέκτημα χρήσης ενός παραθυρικού περιβάλλοντος σε μια εφαρμογή είναι πολύ μεγάλο, επειδή η εφαρμογή γίνεται πιο φιλική στον τελικό χρήστη και δεν είναι απαραίτητη προϋπόθεση ο χρήστης να γνωρίζει αρκετά καλά το λογισμικό πακέτο MATLAB όσο και το πακέτο POLYX.

Η εργασία αποτελείται από τρία κεφάλαια εκ των οποίων στο πρώτο γίνεται μια αναφορά στο πολυωνυμικό πακέτο POLYX, στο δεύτερο γίνεται μια εισαγωγή στο graphical user interfaces του MATLAB και αναπτύσσεται ένα απλό παράδειγμα και τέλος στο τρίτο γίνεται μια λεπτομερή περιγραφή της εφαρμογής.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Πολυωνυμικοί πίνακες, POLYX, MATLAB, GUI

ABSTRACT

This thesis aim at the use of polynomial toolbox POLYX that has application in analysis and design multivariable systems automatic control, with the help of a graphical user interface. The development of this graphical user interface with the use of software MATLAB. The advantage of use a graphical user interface in a programming application is huge, because the use is more user friendly in the final user and it is not necessary requirement for the user to know the software MATLAB and the toolbox POLYX.

The project consists of three chapters, in the first of which there is report for the polynomial toolbox POLYX, in the second there is an introduction in the graphical user interface of MATLAB and a simple example is developed and finally in the third there is a detailed description of the application.

KEY WORDS

Polynomial matrix, POLYX, MATLAB, GUI

**Στην μνήμη του πατέρα μου
Νίκο**

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT.....	4
ΠΕΡΙΕΧΟΜΕΝΑ.....	7
1 ^ο ΚΕΦΑΛΑΙΟ MATLAB - POLYX	9
1.1 ΕΙΣΑΓΩΓΗ ΣΤΟ MATLAB	10
1.2 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΑΚΕΤΟΥ POLYX	11
1.2.1 Συναρτήσεις	12
1.2.2 Κλάσεις	13
1.2.3 Μεταβλητές.....	14
2 ^ο ΚΕΦΑΛΑΙΟ GRAPHICAL USER INTERFACE	15
2.1 ΓΡΑΦΙΚΗ ΔΙΕΠΙΦΑΝΕΙΑ ΧΡΗΣΤΗ.....	16
2.1.1 Εισαγωγή στο GUIDE.....	17
2.1.2 Γραμμή εργαλείων	19
2.1.3 Εργαλειοθήκη	22
2.1.4 Προγραμματισμός στο m-file.....	28
2.2 ΜΙΑ ΑΠΛΗ ΕΦΑΡΜΟΓΗ GUI.....	30
2.2.1 Θεωρητικός σχεδιασμός εφαρμογής	30
2.2.2 Σχεδιασμός εφαρμογής με τη βοήθεια του GUIDE.....	31
2.2.3 Προγραμματισμός εφαρμογής	39
2.2.4 Εκτέλεση εφαρμογής	47
3 ^ο ΚΕΦΑΛΑΙΟ ΑΝΑΠΤΥΞΗ ΔΙΑΔΡΑΣΤΙΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ (GUI) ΓΙΑ ΤΗΝ ΑΝΑΛΥΣΗ ΚΑΙ ΣΥΝΘΕΣΗ ΠΟΛΥΜΕΤΑΒΛΗΤΩΝ ΣΥΣΤΗΜΑΤΩΝ ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ	53

3.1 ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	54
3.2 ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΦΑΡΜΟΓΗΣ	59
3.2.1 Σχεδιασμός και προγραμματισμός στοιχείων ελέγχου...	59
3.2.2 Αρχικές συναρτήσεις εφαρμογής.....	62
3.2.3 Αντικείμενα, δομές και παράθυρα διαλόγου της εφαρμογής	62
3.2.4 Εμφάνιση αποτελεσμάτων	64
3.2.5 Αρχικοποίηση στοιχείων ελέγχου.....	66
3.2.6 Προγραμματισμός σταθερών στοιχείων ελέγχου.....	68
3.2.7 Προγραμματισμός προσωρινών στοιχείων ελέγχου.....	77
3.2.8 Προγραμματισμός μενού επιλογών.....	79
3.2.8.1 Μενού file.....	79
3.2.8.2 Μενού basic operations.....	85
3.2.8.3 Μενού advanced operations.....	105
3.2.8.4 Μενού equations.....	133
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	149
ΠΑΡΑΡΤΗΜΑΤΑ.....	151

1^ο ΚΕΦΑΛΑΙΟ

MATLAB - POLYX

Το MATLAB είναι μια δυναμική εφαρμογή για υπολογισμούς και έχει γίνει πλέον το εργαλείο των περισσότερων επιστημονικών εφαρμογών. Το πολυωνυμικό πακέτο POLYX, το οποίο είναι ένα toolbox του MATLAB, προσφέρει στον χρήστη τη δυνατότητα να υλοποιεί πράξεις, απλές και σύνθετες, μεταξύ πολυωνυμικών πινάκων.

1.1 ΕΙΣΑΓΩΓΗ ΣΤΟ MATLAB

Το πακέτο λογισμικού MATLAB (MathWorks Inc.) παρέχει ένα δυναμικό, εύχρηστο και ανοικτό υπολογιστικό περιβάλλον για υλοποίηση επιστημονικών εφαρμογών σε ένα μεγάλο φάσμα γνωστικών πεδίων, όπως στη Γραμμική Άλγεβρα, τη Στατιστική, τα Εφαρμοσμένα Μαθηματικά, την Αριθμητική Ανάλυση, την Επεξεργασία Σημάτων και Εικόνας, τη Θεωρία Ελέγχου. Έχει υλοποιηθεί σε πολλές λειτουργικές πλατφόρμες (όπως Windows, Macintosh OS και Unix) και σε δύο βασικές εκδόσεις, την επαγγελματική και την εκπαιδευτική (student edition).

Το περιβάλλον του MATLAB υποστηρίζει ένα μεγάλο αριθμό ενδογενών λειτουργιών και συναρτήσεων, καθώς και εξωτερικές βιβλιοθήκες (Toolboxes) για εξειδικευμένες περιοχές εφαρμογών. Υποστηρίζει επίσης μία ευέλικτη, απλή και δομημένη γλώσσα προγραμματισμού (script language), η οποία έχει πολλές ομοιότητες με τη γλώσσα C, και παρέχει δυνατότητες εύκολης δημιουργίας, διασύνδεσης και χρήσης βιβλιοθηκών σε κώδικα γραμμένο στη γλώσσα αυτή.

Το MATLAB εκτελεί από απλούς μαθηματικούς υπολογισμούς έως και προγράμματα με εντολές παρόμοιες εκείνων που υποστηρίζει μία γλώσσα προγραμματισμού υψηλού επιπέδου. Συγκεκριμένα, εκτελεί απλές μαθηματικές πράξεις αλλά εξίσου εύκολα χειρίζεται μιγαδικούς αριθμούς, δυνάμεις, ειδικές μαθηματικές συναρτήσεις, πίνακες, διανύσματα και πολυώνυμα. Μπορεί επίσης να αποθηκεύει και να ανακαλεί δεδομένα, να δημιουργεί και να εκτελεί ακολουθίες εντολών που αυτοματοποιούν διάφορους υπολογισμούς και να σχεδιάζει γραφικά.

Το MATLAB μπορεί να δημιουργήσει και γραφικές διεπιφάνειες χρήστη, δηλαδή να δημιουργήσει παραθυρικό περιβάλλον στο οποίο να μπορεί ο χρήστης να χρησιμοποιήσει όλες τις δυνατότητες αυτής της εφαρμογής. Αυτό είναι και το αντικείμενο της παρούσης διπλωματικής εργασίας.

Οι λειτουργίες του MATLAB διακρίνονται στις τυποποιημένες, δηλαδή σε αυτές που χειρίζονται αριθμητικά δεδομένα και εξάγουν αριθμητικά

αποτελέσματα, και στις συναρτήσεις του Symbolic Toolbox, οι οποίες χειρίζονται και υπολογίζουν συμβολικές εκφράσεις, δηλαδή επεξεργάζονται μαθηματικά σύμβολα. Το POLYX εισάγει στο MATLAB ένα νέο πακέτο με συναρτήσεις, το polynomial toolbox, το οποίο χειρίζεται πολυωνυμικές εκφράσεις.

1.2 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΑΚΕΤΟΥ POLYX

Στην ενότητα αυτή θα περιγράψουμε το πακέτο POLYX και τις συναρτήσεις του POLYX που χρησιμοποιούνται στην εφαρμογή. Το πακέτο POLYX είναι ένα νέο toolbox του MATLAB, το Polynomial Toolbox, το οποίο έχει εφαρμογή στα συστήματα, στα σήματα, στον έλεγχο στην ανάλυση και στον σχεδιασμό συστημάτων με βάση προχωρημένες πολυωνυμικές μεθόδους. Το POLYX περιέχει πάνω από 200 m - files σε κώδικα MATLAB. Πιο συγκεκριμένα το POLYX έχει τα παρακάτω χαρακτηριστικά και δυνατότητες:

- Απλές πράξεις μεταξύ πολυωνυμικών πινάκων και ευανάγνωστη παρουσίαση αυτών με την βοήθεια ενός νέου αντικειμένου (pol-object).
- Σύνθετες πράξεις και συναρτήσεις για πολυωνυμικούς πίνακες καθώς και αρκετές ρουτίνες για την επίλυση γραμμικών και τετραγωνικών εξισώσεων με πολυωνυμικούς πίνακες.
- Πολυωνυμικούς πίνακες με μιγαδικούς συντελεστές για εφαρμογή στην επεξεργασία σήματος.
- Σήματα και συστήματα σε συνεχή ή διακριτό χρόνο βασισμένα σε κλασματικούς (ρητούς) πολυωνυμικούς πίνακες
- Κλασσική και εύρωστη ανάλυση των γραμμικά χρονικά ανεξάρτητων LTI συστημάτων και φίλτρων.
- Κλασσικός και βέλτιστος σχεδιασμός στην επανατοποθέτηση πόλων και στους σταθεροποιητικούς ελεγκτές.

- Εύκολη μετατροπή ενός αντικειμένου σε μορφή LTI object του MATLAB και από μορφή pol object σε sym object του Symbolic Math Toolbox του MATLAB.

Στην συνέχεια αυτού του κεφαλαίου αναφέρονται οι συναρτήσεις, οι κλάσεις των αντικειμένων και οι μεταβλητές του POLYX που χρησιμοποιούνται στην εφαρμογή.

1.2.1 Συναρτήσεις

Στα πλαίσια της εφαρμογής μας, επικεντρωθήκαμε σε συναρτήσεις οι οποίες έχουν εφαρμογή στις πράξεις πολυωνύμων ή πολυωνυμικών πινάκων, για παράδειγμα η πρόσθεση μεταξύ δυο πινάκων.

Επίσης, επικεντρωθήκαμε σε συναρτήσεις οι οποίες έχουν εφαρμογή στις ιδιότητες των πολυωνύμων ή πολυωνυμικών πινάκων, για παράδειγμα ο υπολογισμός της οριζουσας καθώς και σε συναρτήσεις που έχουν εφαρμογή στις ειδικές μορφές πινάκων, για παράδειγμα ο υπολογισμός της Companion μορφής ενός πολυωνυμικού πίνακα.

Τέλος, επικεντρωθήκαμε σε συναρτήσεις οι οποίες έχουν εφαρμογή στην επίλυση εξισώσεων με πολυώνυμα ή πολυωνυμικούς πίνακες, για παράδειγμα η επίλυση διοφαντικών εξισώσεων αλλά και η επίλυση εξισώσεων Lyapunov.

Στο παράρτημα δίνεται αναλυτικότερα ο τρόπος σύνταξης καθώς και η περιγραφή όλων των συναρτήσεων του POLYX που χρησιμοποιούνται από την εφαρμογή.

1.2.2 Κλάσεις

Οι κλάσεις των αντικειμένων (objects) του POLYX που χρησιμοποιούνται στην εφαρμογή είναι οι εξής:

Η πολυωνυμική μορφή **pol-object** που έχει ως στοιχεία του πίνακα απλά πολυώνυμα, όπως φαίνεται και παρακάτω.

$\begin{matrix} s & s \\ -1 + s & 1 + s \end{matrix}$

pol-object

Η ρητή πολυωνυμική μορφή **sdf-object** που έχει ως στοιχεία του πίνακα πολώνυμα και ένα πολώνυμο ως παρονομαστή, όπως φαίνεται και παρακάτω.

$\begin{matrix} 0.5 + 0.5s & -0.5s \\ 0.5 - 0.5s & 0.5s \\ \hline s \end{matrix}$

sdf-object

Η ρητή πολυωνυμική μορφή **rdf-object** με αριθμητή πολυωνυμικό πίνακα και δεξιό παρονομαστή πολυωνυμικό πίνακα, όπως φαίνεται και παρακάτω.

$\begin{matrix} s & s^2 & / & s & s \\ 1 & 1 + s^3 & / & -1 + s & 1 + s \end{matrix}$

rdf-object

Η ρητή πολυωνυμική μορφή **ldf-object** με αριθμητή πολυωνυμικό πίνακα και αριστερό παρονομαστή πολυωνυμικό πίνακα, όπως φαίνεται και παρακάτω.

s	s^2	\backslash	s	s
1	$1 + s^3$	\backslash	$-1 + s$	$1 + s$

ldf-object

1.2.3 Μεταβλητές

Οι μεταβλητές του POLYX που χρησιμοποιούνται στην εφαρμογή είναι οι εξής:

- Η μεταβλητή s που χρησιμοποιείται από το POLYX για σήματα και συστήματα συνεχή χρόνου.
- Η μεταβλητή z που χρησιμοποιείται από το POLYX για σήματα και συστήματα διακριτού χρόνου.

ΚΕΦΑΛΑΙΟ 2^ο

GRAPHICAL USER INTERFACE (ΓΡΑΦΙΚΗ ΔΙΕΠΙΦΑΝΕΙΑ ΧΡΗΣΤΗ)

Το MATLAB προσφέρει στο χρήστη τη δυνατότητα να κατασκευάσει δικές του γραφικές διεπιφάνειες, (GUI¹). Η χρησιμότητα της λειτουργίας αυτής είναι μεγάλη, επειδή τα προγράμματα - εφαρμογές που περιέχουν γραφική διεπιφάνεια γίνονται πιο φιλικά στον τελικό χρήστη. Στο κεφάλαιο αυτό περιγράφονται πρώτα τα εργαλεία που προσφέρονται από το MATLAB για τη δημιουργία γραφικών διεπιφανειών και στη συνέχεια παρουσιάζεται ένα απλό παράδειγμα ενός GUI .

¹ Graphical User Interface

2.1 ΓΡΑΦΙΚΗ ΔΙΕΠΙΦΑΝΕΙΑ ΧΡΗΣΤΗ

Σκοπός αυτού του κεφαλαίου είναι να κατανοήσει ο χρήστης πως μπορεί να δημιουργήσει μια παραθυρική εφαρμογή με τη βοήθεια του MATLAB. Υπάρχουν πολλά πλεονεκτήματα από τη χρήση παραθυρικού περιβάλλοντος. Σημαντικό ρόλο όμως παίζει και η αρχική σωστή σχεδίαση μιας διεπιφάνειας χρήστη, έτσι ώστε να δημιουργηθεί ένα απλό και λειτουργικό GUI, χωρίς περιττά στοιχεία που θα περιπλέκουν τον απλό χρήστη.

Η ανάπτυξη μιας εφαρμογής σε GUI χωρίζεται σε τέσσερα στάδια:

1. Θεωρητική σχεδίαση της εφαρμογής πριν την υλοποίηση της σε MATLAB.
2. Σχεδιασμός της εφαρμογής στο κατάλληλο περιβάλλον (GUIDE Layout Editor) και δημιουργία των απαιτούμενων αντικειμένων.
3. Καθορισμός ιδιοτήτων του GUI και του κάθε αντικειμένου.
4. Προγραμματισμός του κάθε αντικειμένου, εάν χρειάζεται.

Το MATLAB προσφέρει μια ικανοποιητική εργαλειοθήκη, η οποία διευκολύνει πολύ τη δημιουργία ενός GUI. Αυτή η εργαλειοθήκη ή αλλιώς GUIDE (Graphical User Interface Design Environment), όπως θα δούμε στη συνέχεια, περιέχει μια πληθώρα χρήσιμων εργαλείων ελέγχου όπως κουμπιά, πλαίσια κ.α. Για να αξιοποιήσει πλήρως ο χρήστης τις δυνατότητες του GUIDE, είναι απαραίτητο να γνωρίζει πρώτα κάποιες βασικές εντολές της γλώσσας MATLAB.

Ο χρήστης έχει τη δυνατότητα να μη χρησιμοποιήσει τον GUIDE και αντί αυτού να δημιουργήσει μόνος του κάποιο figure προγραμματίζοντας κατάλληλα σε ένα m-file. Επίσης μπορεί να εισάγει και να προγραμματίσει τα διάφορα στοιχεία ελέγχου εκ του μηδενός. Με αυτή την τεχνική προγραμματισμού, η οποία ονομάζεται χαμηλού επιπέδου, θα ασχοληθούμε ιδιαίτερα στο κομμάτι παρουσίασης της εφαρμογής. Στην επόμενη ενότητα θα περιγράψουν τα στάδια ανάπτυξης μιας εφαρμογής με την χρήση του GUIDE, το οποίο αποτελεί μια τεχνική προγραμματισμού υψηλού επιπέδου.

2.1.1 Εισαγωγή στο GUIDE

Ένα GUI αποτελείται συνήθως από διάφορα παράθυρα, τα οποία περιέχουν ποικίλα στοιχεία ελέγχου όπως πεδία κειμένου, γραμμές κύλισης, λίστες, πεδία εισαγωγής κειμένου κ. α. Τα παράθυρα αυτά είναι δυνατόν να καλούν το ένα το άλλο, να δέχονται δεδομένα από τον χρήστη, να μεταβιβάζουν πιθανώς τα δεδομένα από το ένα παράθυρο στο άλλο και γενικά να επιτελούν διάφορες λειτουργίες. Το GUIDE για να το πετύχει αυτό δημιουργεί για κάθε νέο παράθυρο δύο αρχεία. Τα αρχεία αυτά είναι το fig - file και το m-file.

Το fig-file, ουσιαστικά είναι το παράθυρο-figure, όπου το MATLAB αποθηκεύει τα στοιχεία ελέγχου και την ακριβή θέση τους. Εδώ ο προγραμματιστής σχεδιάζει την εμφάνιση του παραθύρου.

Το m-file, όπου ο προγραμματιστής πρέπει να γράψει τον κώδικα που θα ενσωματωθεί στα στοιχεία ελέγχου (π.χ. κουμπιά ή φόρμες εισαγωγής δεδομένων), ώστε αυτά να επιτελέσουν τις επιθυμητές λειτουργίες.

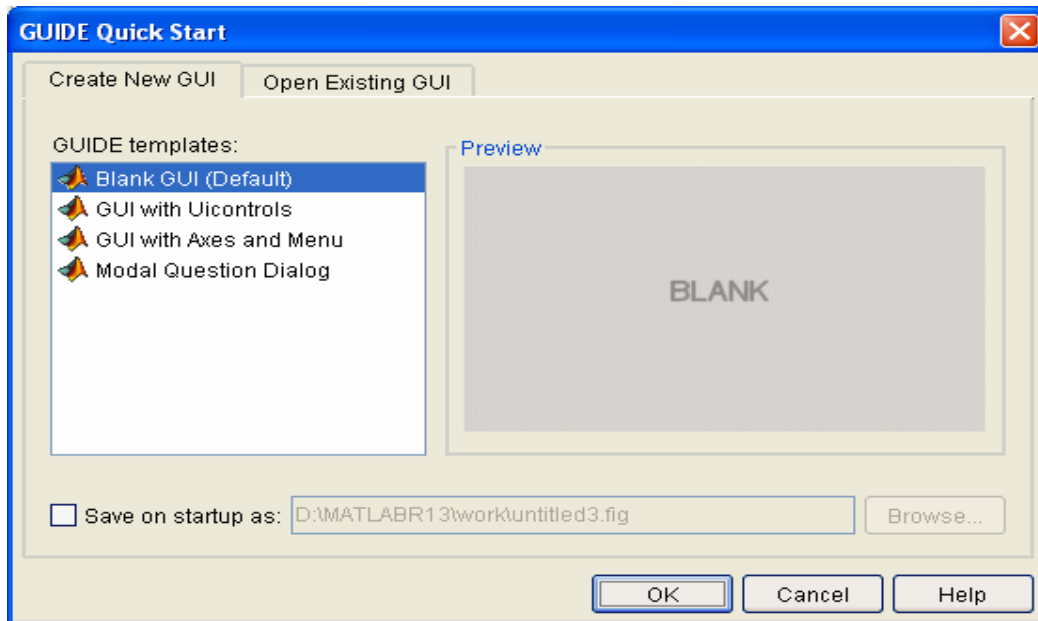
Πρέπει να σημειωθεί ότι κάθε αρχείο *.fig πρέπει να συνοδεύεται από το αντίστοιχο αρχείο *.m (με το ίδιο όνομα). Αν για κάποιο λόγο, χαθεί ή καταστραφεί το *.m αρχείο, τότε το παράθυρο *.fig δεν θα είναι λειτουργικό. Κάθε φορά που κάποιος χρήστης δημιουργεί ένα νέο παράθυρο (figure), το GUIDE δημιουργεί αυτομάτως και τους δυο προαναφερθέντες τύπους αρχείων. Στη συνέχεια θα περάσουμε στην πράξη, δείχνοντας πως γίνεται η εκκίνηση του GUIDE. Επίσης θα περιγράψουμε τις ιδιότητες μερικών βασικών στοιχείων ελέγχου.

Η εκκίνηση του GUIDE γίνεται εύκολα, με δυο τρόπους. Πρώτον, με την κλήση της ομώνυμης συνάρτησης από τη γραμμή εντολών του MATLAB.

» guide

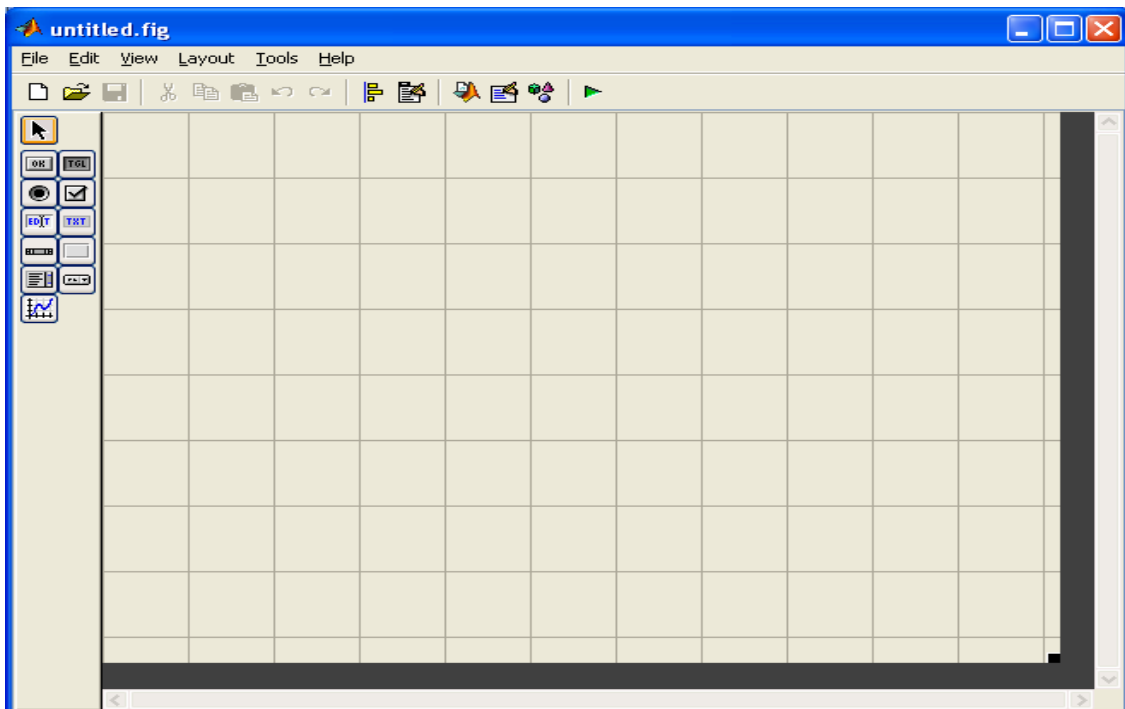
Δεύτερον, επιλέγοντας από τη γραμμή μενού File → New → GUI . Όποιο τρόπο και να επιλέξει ο χρήστης, θα ξεκινήσει ένας οδηγός που θα τον καθοδηγεί στη δημιουργία ενός παραθύρου, όπως στην εικόνα 2.1. Ο οδηγός ρωτάει τον χρήστη αν θέλει να δημιουργήσει ένα κενό παράθυρο (Blank GUI), αν θέλει να

δημιουργήσει ένα παράθυρο βασισμένο σε κάποια πρότυπα, (π.χ. με άξονες και μενού) ή αν θέλει να ανοίξει ένα έτοιμο παράθυρο.



Εικόνα 2.1 Ο αρχικός οδηγός του GUIDE.

Αν ο χρήστης αφήσει την προεπιλεγμένη επιλογή και πατήσει το κουμπί OK, τότε θα δημιουργηθεί ένα νέο άδειο παράθυρο, όπως στην εικόνα 2.2



Εικόνα 2.2. Ένα νέο άδειο παράθυρο.

Ο χρήστης μπορεί να δει ότι το περιβάλλον δημιουργίας του παραθύρου αποτελείται από μια κεντρική γραμμή επιλογών, μια γραμμή εργαλείων, καθώς και μια κάθετη εργαλειοθήκη στα αριστερά. Η γκριζα περιοχή με το πλέγμα το οποίο καλύπτει το παράθυρο, είναι το φόντο, η ταπετσαρία του παραθύρου.

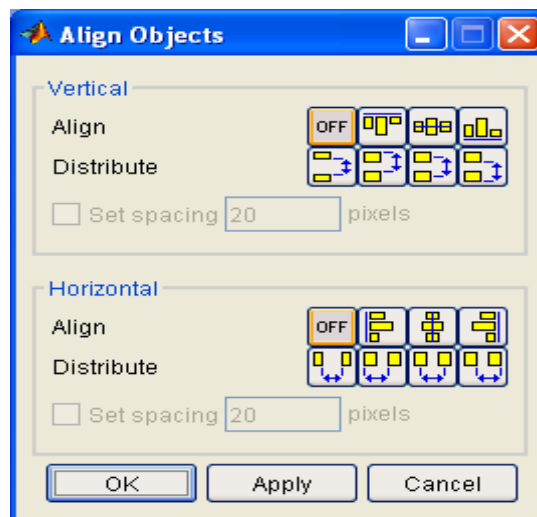
2.1.2 Γραμμή εργαλείων

Ιδιαίτερης σημασίας είναι τα τελευταία έξι κουμπιά της γραμμής εργαλείων, που φαίνονται σε μεγέθυνση στην εικόνα 2.3.



Εικόνα 2.3. Κουμπιά από τη γραμμή εργαλείων.

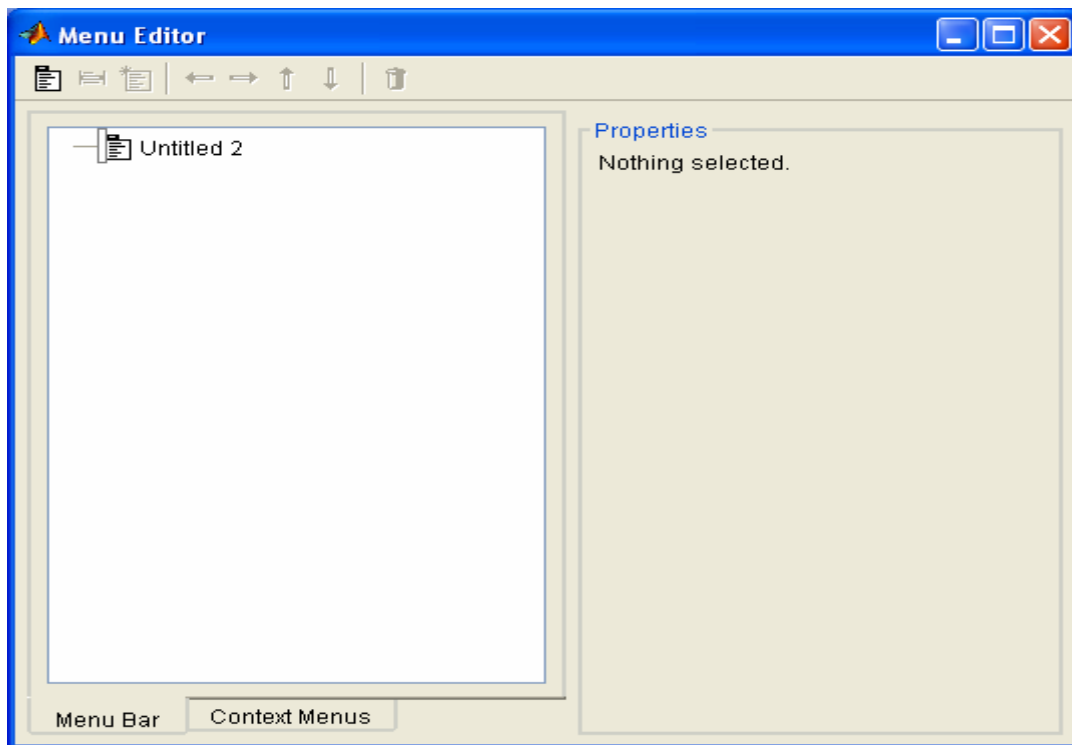
Στη συνέχεια θα δοθεί μια περιγραφή των λειτουργιών που προσφέρονται από αυτήν τη γραμμή εργαλείων. Το **πρώτο** εικονίδιο, από τα αριστερά, ενεργοποιεί το παράθυρο της εικόνας 2.4



Εικόνα 2.4. Στοιχισι αντικειμένων.

Ο χρήστης, μέσω αυτού του παραθύρου, μπορεί να στοιχίσει είτε κατακόρυφα είτε οριζόντια, όποια στοιχεία ελέγχου επιθυμεί. Κλασικό παράδειγμα είναι η εισαγωγή στοιχείων σε φόρμες, όπου συνηθίζεται τα πεδία εισαγωγής κειμένου να είναι ομοιόμορφα τοποθετημένα και όχι σε τυχαία θέση.

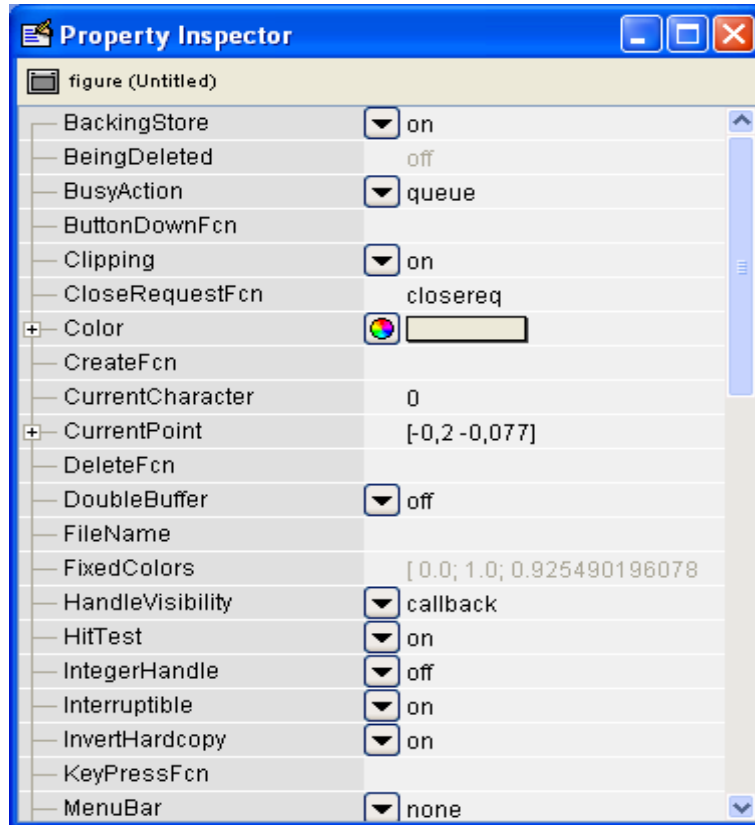
Το **δεύτερο** από τα κουμπιά της εικόνας 2.3 μας επιτρέπει να δημιουργήσουμε γραμμή μενού στην παραθυρική μας εφαρμογή. Η γραμμή μενού είναι δυνατό να περιέχει και πτυσσόμενα υπό-μενού. Στην επόμενη εικόνα 2.5 φαίνεται το παράθυρο που ενεργοποιείται με το πάτημα του σχετικού κουμπιού.



Εικόνα 2.5. Δημιουργία μενού και υπό - μενού.

Στο παράδειγμα που ακολουθεί θα εξετάσουμε αναλυτικότερα την εισαγωγή μενού ή και υπό-μενού. Το πάτημα του **τρίτου** κουμπιού της εικόνας 2.3 εκκινεί τον γνωστό m-file Editor, ανοίγοντας ταυτόχρονα το αντίστοιχο m-file

του παραθύρου μας. Το **τέταρτο** κουμπί εκκινεί τον Property Inspector όπως φαίνεται στην εικόνα 2.6 .















Εικόνα 2.6. Ρύθμιση ιδιοτήτων των στοιχείων ελέγχου.

Από το προηγούμενο παράθυρο, και έχοντας επιλέξει πρώτα κάποιο στοιχείο ελέγχου, ή και το ίδιο το παράθυρο της εφαρμογής μας, αλλάζουμε όποιες ιδιότητες επιθυμούμε. Το **πέμπτο** κουμπί της εικόνας 2.3 ανοίγει ένα παράθυρο, που μας δείχνει πόσα και ποια αντικείμενα υπάρχουν στο συγκεκριμένο παράθυρο της εφαρμογής μας. Τέλος, το **έκτο** κουμπί της εικόνας 2.3 εκτελεί την εφαρμογή μας αλλά ταυτόχρονα μας ενημερώνει ότι πρέπει να αποθηκευτεί προηγουμένως.

2.1.3 Εργαλειοθήκη

Η εργαλειοθήκη που βρίσκεται κάθετα αριστερά στην εικόνα 2.2, περιέχει όλα τα στοιχεία ελέγχου, τα οποία είναι διαθέσιμα στον χρήστη.

Η εισαγωγή τους στο παράθυρο είναι απλή και γίνεται με απλό σύρσιμο (drag and drop). Στην συνέχεια θα αναφερθούμε εκτενέστερα για κάθε στοιχείο ελέγχου.

	Select
	Push Button
	Toggle Button
	Radio Button
	Checkbox
	Edit Text
	Static Text
	Slider
	Frame
	List box
	Popup Menu
	Axes

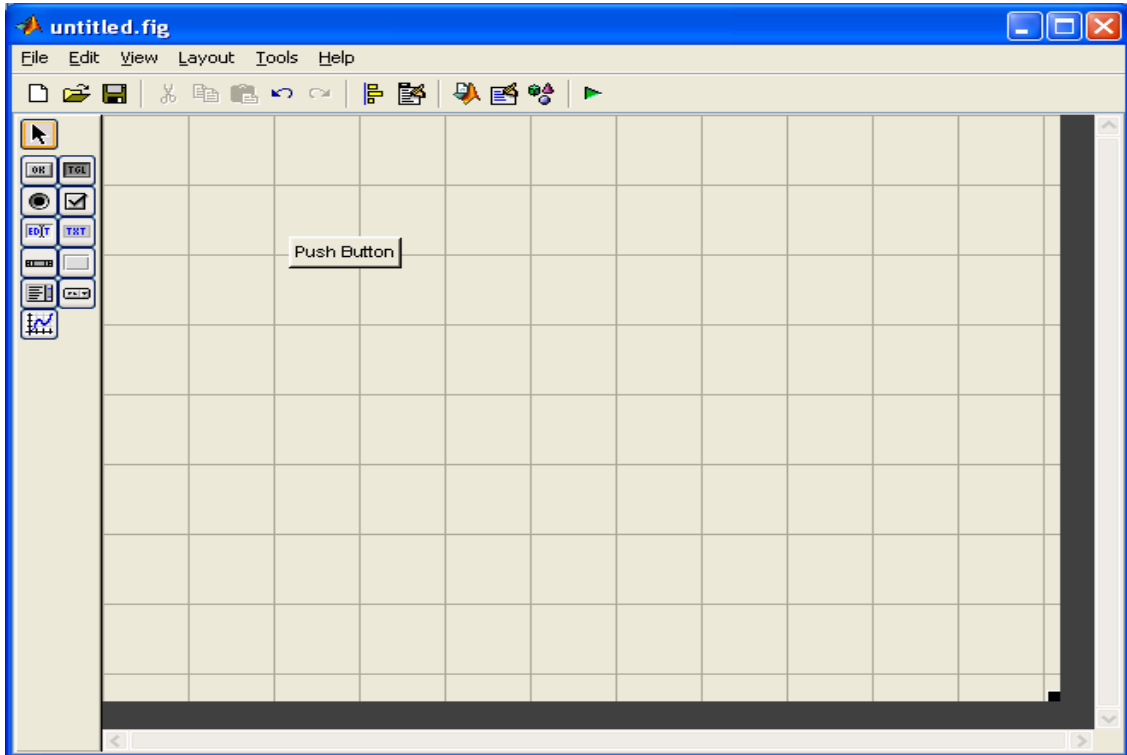
Πίνακας στοιχείων ελέγχου

Select

Όταν είναι ενεργοποιημένο το κουμπί επιλογής έχουμε τη δυνατότητα να επιλέξουμε οποιοδήποτε στοιχείο ελέγχου, αλλά και να κάνουμε πολλαπλή επιλογή έτσι ώστε να δημιουργήσουμε μια ομάδα στοιχείων ελέγχου.

Push Button

Αφού εισάγουμε στο παράθυρο ένα push button τότε αυτόματα δημιουργείται ένα, σε συγκεκριμένες διαστάσεις, το οποίο έχει κάποιες ιδιότητες.



Εικόνα 2.7. Παράθυρο με το στοιχείο ελέγχου push button.

Χρησιμοποιώντας τον Property Inspector μπορούμε να αλλάξουμε κάποιες ιδιότητες. Στον παρακάτω πίνακα παρουσιάζονται η ιδιότητες που είναι κοινές σε όλα τα στοιχεία ελέγχου και χρησιμοποιούνται περισσότερο. Στην πρώτη στήλη έχουμε το όνομα της ιδιότητας, στην δεύτερη στήλη έχουμε τις επιλογές για κάθε ιδιότητα και στην τρίτη στήλη έχουμε την περιγραφή.

Στις επιλογές έχουμε τις προκαθορισμένες τιμές (default) για κάθε ιδιότητα ή, όταν μπορούμε να διαλέξουμε μεταξύ κάποιων επιλογών, σε άγκιστρα { } είναι η προκαθορισμένη επιλογή (default) ενώ σε παρενθέσεις () είναι οι άλλες επιλογές .

ΙΔΙΟΤΗΤΑ	ΕΠΙΛΟΓΕΣ	ΠΕΡΙΓΡΑΦΗ
Background Color	Default	Ορίζουμε το χρώμα του φόντου
Enable	{on} (off)	Ορίζουμε αν είναι ενεργό ή όχι το στοιχείο ελέγχου
FontName	MS Sans Serif	Ορίζουμε την γραμματοσειρά
FontSize	8.0	Ορίζουμε το μέγεθος των γραμμάτων
FontWeight	{normal} (light) (demi) (bold)	Ορίζουμε το στυλ των γραμμάτων
HorizontalAlignment	{center} (right) (left)	Οριζόντια στοίχιση
Position		Ορίζουμε τις διαστάσεις και την θέση του στοιχείου ελέγχου
	X	Οριζόντια απόσταση από την αριστερή κάτω γωνία
	Y	Κάθετη απόσταση από την αριστερή κάτω γωνία
	Width	Πλάτος στοιχείου ελέγχου
	Height	Ύψος στοιχείου ελέγχου
String	pushbutton	Ορίζουμε το κείμενο που θα είναι πάνω στο στοιχείο ελέγχου
Tag	pushbutton1	Ορίζουμε την ετικέτα του στοιχείου ελέγχου
Units	{characters} (points) (normalized) (pixels) (centimeters) (inches)	Ορίζουμε την μονάδα μέτρησης του στοιχείου ελέγχου
Visible	{on} (off)	Ορίζουμε αν είναι ορατό ή όχι το στοιχείο ελέγχου

Πίνακας ιδιοτήτων στοιχείων ελέγχου

Το κουμπί `pushbutton1` συνοδεύεται από μια συνάρτηση στο αντίστοιχο `m-file` του παραθύρου την

```
function pushbutton1_Callback(hObject, eventdata, handles)
```

στην οποία ενσωματώνουμε τις εντολές που θέλουμε να εκτελεί το κουμπί αφού το πατήσουμε .

Toggle Button

Το στοιχείο ελέγχου `togglebutton` μας δίνει τη δυνατότητα επιλογής ή όχι μιας λειτουργίας. Εάν το έχουμε πατημένο θα εκτελεστεί η λειτουργία που θα περιγράφεται στο αντίστοιχο `m-file`, αλλιώς, αν δεν είναι πατημένο, δεν επηρεάζει το πρόγραμμα.

Το στοιχείο ελέγχου `togglebutton` συνοδεύεται από μια συνάρτηση στο αντίστοιχο `m-file` του παραθύρου:

```
function togglebutton_Callback(hObject, eventdata, handles)
```

Μπορούμε βέβαια, να αλλάξουμε τις ιδιότητες του στοιχείου ελέγχου με τη βοήθεια του `Property Inspector`, όπως περιγράφονται στον πίνακα ιδιοτήτων .

Radio Button

Το στοιχείο ελέγχου `radiobutton` μας δίνει τη δυνατότητα να επιλέξουμε μια μόνο επιλογή από ένα πλήθος επιλογών. Διαλέγοντας μια αναιρείται η προηγούμενη.

Το στοιχείο ελέγχου `radiobutton` συνοδεύεται από μια συνάρτηση στο αντίστοιχο `m-file` του παραθύρου:

```
function radiobutton_Callback(hObject, eventdata, handles)
```

Μπορούμε βέβαια, να αλλάξουμε τις ιδιότητες του στοιχείου ελέγχου με τη βοήθεια του `Property Inspector`, όπως περιγράφονται στον πίνακα ιδιοτήτων.

Checkbox

Το στοιχείο ελέγχου `checkbox` μας δίνει τη δυνατότητα να επιλέξουμε μια ή περισσότερες επιλογές, δηλαδή να τσεκάρουμε τις επιλογές που θέλουμε.

Το στοιχείο ελέγχου checkbox συνοδεύεται από μια συνάρτηση στο αντίστοιχο m-file του παραθύρου:

```
function checkbox_Callback(hObject, eventdata, handles)
```

Μπορούμε βέβαια, να αλλάξουμε τις ιδιότητες του στοιχείου ελέγχου με τη βοήθεια του Property Inspector, όπως περιγράφονται στον πίνακα ιδιοτήτων.

Edit Text

Το στοιχείο ελέγχου edittext μας δίνει τη δυνατότητα να πάρουμε την τιμή που έχει εισάγει μέσα σ' αυτό ο χρήστης .

Το στοιχείο ελέγχου edittext συνοδεύεται από δυο συναρτήσεις στο αντίστοιχο m-file του παραθύρου:

```
function edit_Callback(hObject, eventdata, handles) και  
function edit_CreateFcn(hObject, eventdata, handles) .
```

Η δεύτερη είναι η συνάρτηση που ορίζει παραμέτρους κατά την δημιουργία του edittext .

Μπορούμε βέβαια, να αλλάξουμε τις ιδιότητες του στοιχείου ελέγχου με τη βοήθεια του Property Inspector, όπως περιγράφονται στον πίνακα ιδιοτήτων.

Static Text

Το στοιχείο ελέγχου statictext μας δίνει τη δυνατότητα να εισάγουμε στατικά κείμενα σε οποιοδήποτε σημείο του παραθύρου .

Το στοιχείο ελέγχου statictext δε συνοδεύεται από καμιά συνάρτηση στο αντίστοιχο m-file του παραθύρου.

Μπορούμε βέβαια, να αλλάξουμε τις ιδιότητες του στοιχείου ελέγχου με τη βοήθεια του Property Inspector, όπως περιγράφονται στον πίνακα ιδιοτήτων .

Slider

Το στοιχείο ελέγχου slider μας δίνει τη δυνατότητα να μεταβάλουμε κάποια τιμή μιας μεταβλητής με τη βοήθεια της μπάρας (slider).

Το στοιχείο ελέγχου slider συνοδεύεται από δυο συναρτήσεις στο αντίστοιχο m-file του παραθύρου:

function slider_Callback(hObject, eventdata, handles) και
function slider_CreateFcn(hObject, eventdata, handles) .

Η δεύτερη είναι η συνάρτηση που ορίζει παραμέτρους κατά την δημιουργία του slider .

Μπορούμε βέβαια, να αλλάξουμε τις ιδιότητες του στοιχείου ελέγχου με τη βοήθεια του Property Inspector, όπως περιγράφονται στον πίνακα ιδιοτήτων .

Frame

Το στοιχείο ελέγχου frame χρησιμοποιείται για την ομαδοποίηση πολλών στοιχείων ελέγχου.

Το στοιχείο ελέγχου frame δε συνοδεύεται από καμία συνάρτηση στο αντίστοιχο m-file του παραθύρου.

Μπορούμε βέβαια, να αλλάξουμε τις ιδιότητες του στοιχείου ελέγχου με τη βοήθεια του Property Inspector, όπως περιγράφονται στον πίνακα ιδιοτήτων .

List box

Το στοιχείο ελέγχου list box μας δίνει τη δυνατότητα να διαλέξουμε μέσα από μια λίστα μια λειτουργία ή να παρουσιάσουμε ένα αποτέλεσμα μιας λειτουργίας.

Το στοιχείο ελέγχου list box συνοδεύεται από δυο συναρτήσεις στο αντίστοιχο m-file του παραθύρου:

function listbox_Callback(hObject, eventdata, handles) και
function listbox_CreateFcn(hObject, eventdata, handles) .

Η δεύτερη είναι η συνάρτηση που ορίζει παραμέτρους κατά τη δημιουργία του list box.

Μπορούμε βέβαια, να αλλάξουμε τις ιδιότητες του στοιχείου ελέγχου με τη βοήθεια του Property Inspector, όπως περιγράφονται στον πίνακα ιδιοτήτων .

Popup Menu

Το στοιχείο ελέγχου popup menu μας δίνει τη δυνατότητα να διαλέξουμε μέσα από ένα μενού πολλαπλών επιλογών την λειτουργία που επιθυμούμε .

Το στοιχείο ελέγχου `popup menu` συνοδεύεται από δυο συναρτήσεις στο αντίστοιχο `m-file` του παραθύρου:

*function popupmenu_Callback(hObject, eventdata, handles) και
function popupmenu_CreateFcn(hObject, eventdata, handles) .*

Η δεύτερη είναι η συνάρτηση που ορίζει παραμέτρους κατά τη δημιουργία του `popup menu` .

Μπορούμε βέβαια, να αλλάξουμε τις ιδιότητες του στοιχείου ελέγχου με τη βοήθεια του `Property Inspector`, όπως περιγράφονται στον πίνακα ιδιοτήτων.

Axes

Το στοιχείο ελέγχου `axes` μας δίνει τη δυνατότητα να εισάγουμε μια γραφική παράσταση στο παράθυρο .

Το στοιχείο ελέγχου `axes` δε συνοδεύεται από καμία συνάρτηση στο αντίστοιχο `m-file` του παραθύρου .

Μπορούμε βέβαια, να αλλάξουμε τις ιδιότητες του στοιχείου ελέγχου με τη βοήθεια του `Property Inspector`, όπως περιγράφονται στον πίνακα ιδιοτήτων, όμως το στοιχείο ελέγχου `axes` έχει περισσότερες ιδιότητες από τα άλλα, οι οποίες είναι για την γραφική παράσταση.

2.1.4 Προγραμματισμός στο m-file

Όπως είδαμε τα περισσότερα στοιχεία ελέγχου συνοδεύονται από μια συνάρτηση `callback`, στην οποία προσθέτουμε τις εντολές που θέλουμε να εκτελεί η ενεργοποίηση του κάθε στοιχείου ελέγχου.

Κάθε στοιχείο ελέγχου μπορεί να δημιουργηθεί και να οριστούν οι ιδιότητες του από το `m-file` του παραθύρου. Και σε ένα έτοιμο στοιχείο ελέγχου, που δημιουργήθηκε με την πιο πάνω διαδικασία, μπορούμε με προγραμματισμό στο `m-file` να αλλάξουμε τις ιδιότητες του.

Κάθε στοιχείο ελέγχου είναι ένα αντικείμενο (`handles`) στο `m-file` και καλείται ως `handles.tag_of_button`. Οι εντολές `set` και `get` δίνουν τη

δυνατότητα να αλλάξουμε κάποια ιδιότητα σε ένα στοιχείο ελέγχου ή να πάρουμε κάποια τιμή από στοιχεία ελέγχου που επιστρέφουν τιμές (Toggle Button, Radio Button, Checkbox, Edit Text, Slider, List box ,Popur Menu). Για παράδειγμα η εντολή

```
set(handles.text1,'string','hello');
```

αλλάζει το κείμενο που υπήρχε στο text1 και τοποθετεί το κείμενο hello, ενώ η εντολή

```
temp=get(handles.edit1,'string');
```

παίρνει ότι υπάρχει μέσα στο αντικείμενο edit1 και το καταχωρεί στην μεταβλητή temp.

Οι ιδιότητες που χρησιμοποιούνται περισσότερο στον προγραμματισμό του m-file είναι οι εξής : string,value .

Η ιδιότητα **string** δίνει τη δυνατότητα να εισάγουμε σε όλα τα αντικείμενα εκτός του axes, με τη βοήθεια της set, κάποιο κείμενο πάνω στο αντικείμενο, καθώς και να πάρουμε κάποιο κείμενο από ένα αντικείμενο με τη βοήθεια της get.

Η ιδιότητα **value** δίνει τη δυνατότητα να ορίσουμε μια τιμή ή να πάρουμε μια τιμή από ένα αντικείμενο. Στον παρακάτω πίνακα έχουμε τα αντικείμενα, τα οποία έχουν την ιδιότητα value και τις τιμές που παίρνει η ιδιότητα, αλλά και μια περιγραφή.

ΑΝΤΙΚΕΙΜΕΝΟ	ΤΙΜΕΣ	ΠΕΡΙΓΡΑΦΗ
Toggle Button	1	Όταν έχει πατηθεί
	0	Όταν δεν έχει πατηθεί
Radio Button	1	Όταν έχει επιλεγθεί
	0	Όταν δεν έχει επιλεγθεί
Checkbox	1	Όταν έχει τσεκαριστεί
	0	Όταν δεν έχει τσεκαριστεί
Slider	1	Μας επιστρέφει την θέση που βρίσκεται το
	0	slider (αριθμός μεταξύ 0 και 1)

List box	1	Όταν έχει επιλεγθεί η πρώτη γραμμή
	2	Όταν έχει επιλεγθεί η δεύτερη γραμμή
	3	Όταν έχει επιλεγθεί η τρίτη γραμμή
	.	.
	.	.
Popup Menu	1	Όταν έχει επιλεγθεί η πρώτη επιλογή
	2	Όταν έχει επιλεγθεί η δεύτερη επιλογή
	3	Όταν έχει επιλεγθεί η τρίτη επιλογή
	.	.
	.	.

Πίνακας τιμών ιδιότητας value

Ο τρόπος με τον οποίο μπορούμε να ορίσουμε ή να πάρουμε κάποια τιμή από την ιδιότητα value είναι ο ίδιος με εκείνο της ιδιότητας string.

2.2 ΜΙΑ ΑΠΛΗ ΕΦΑΡΜΟΓΗ GUI

2.2.1 Θεωρητικός σχεδιασμός εφαρμογής

Στην εφαρμογή που ακολουθεί θα αναλυθεί ο προγραμματισμός σε χαμηλό επίπεδο των GUI. Η εφαρμογή που θα δημιουργήσουμε είναι ένας matrix editor, στον οποίο θα δημιουργούμε αριθμητικούς τετραγωνικούς πίνακες στους οποίους θα μπορούμε να κάνουμε κάποιες πράξεις. Οι πράξεις που θα υπολογίζει η εφαρμογή θα είναι η εύρεση ανάστροφου πίνακα, ο υπολογισμός της ορίζουσας, η εύρεση αντιστρόφου πίνακα, ο υπολογισμός του ίχνους, ο υπολογισμός αθροισμάτων γραμμών ή στηλών και ο υπολογισμός νορμών. Για λόγους απλότητας το συγκεκριμένο πρόγραμμα θα υποστηρίζει τετραγωνικούς πίνακες με διαστάσεις έως 5 x 5. Ο χρήστης θα χειρίζεται το

πρόγραμμα, μέσω μιας διεπιφάνειας, η οποία θα περιέχει τα απαραίτητα στοιχεία ελέγχου:

- 3 κουμπιά εκτέλεσης εντολών (pushbutton) για να εκτελούν τις λειτουργίες calculate, refresh, save .
- Ένα μενού πολλαπλών επιλογών (popup menu), από το οποίο θα επιλέγεται η επιθυμητή πράξη και ένα static text, το οποίο θα ενημερώνει το χρήστη.
- Δημιουργία μενού για την επιλογή νέου πίνακα και για την έξοδο από την εφαρμογή. Δημιουργία των αντιστοίχων static text που θα ενημερώνουν τον χρήστη.
- Κάθε πράξη μπορεί να συνοδεύεται από checkbox για τον διαχωρισμό των επιμέρους πράξεων .

2.2.2 Σχεδιασμός εφαρμογής με τη βοήθεια του GUIDE

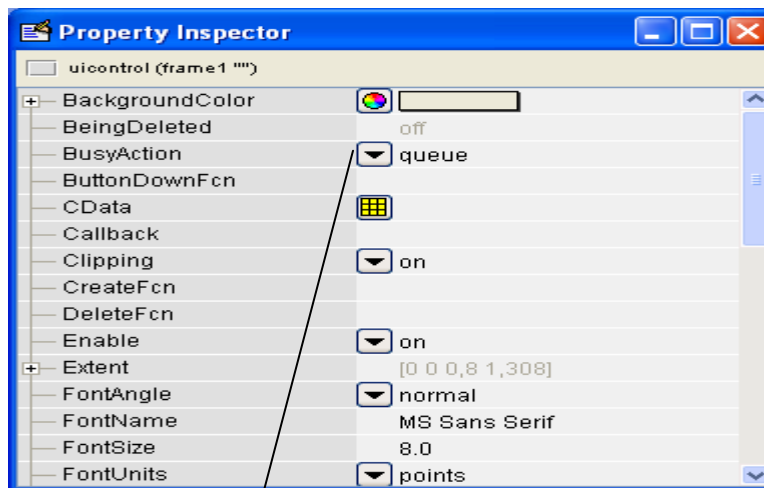
Αρχικά εκκινούμε το GUIDE και δημιουργούμε ένα νέο παράθυρο-figure. Το αποθηκεύουμε δίνοντάς του ταυτόχρονα το όνομα *matrix_editor*. Σε αυτό το σημείο το MATLAB αυτομάτως εκκινεί τον m-file editor και δημιουργεί το αντίστοιχο m-file. Παρατηρήστε ότι στο m-file δημιουργούνται κάποιες ενσωματωμένες συναρτήσεις, όπως είναι οι ακόλουθες:

```
function varargout = matrix_editor(varargin)
function matrix_editor_OpeningFcn(hObject, eventdata, handles, varargin)
function varargout = matrix_editor_OutputFcn(hObject, eventdata, handles)
```

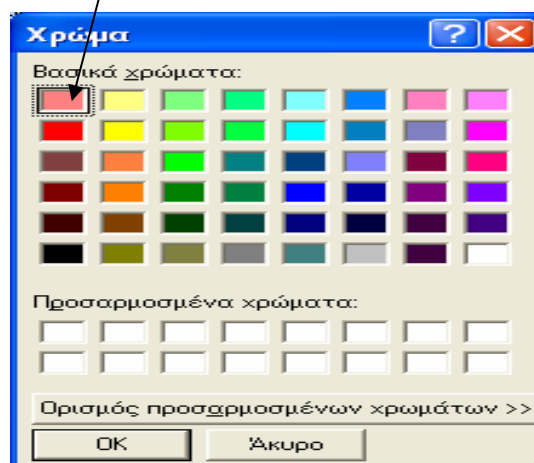
Οι παραπάνω συναρτήσεις που ενσωματώθηκαν στο m-file, είναι απαραίτητες για τη σωστή λειτουργία του παραθύρου. Το MATLAB για κάθε μια από αυτές συμπεριλαμβάνει κάποια επεξηγηματικά σχόλια. Συγκεκριμένα, η πρώτη συνάρτηση χρειάζεται ώστε να μπορεί ο χρήστης να καλεί το παράθυρο από τη γραμμή εντολών πληκτρολογώντας το όνομα του (» *matrix_editor*). Επιπλέον, ο κώδικας που περιλαμβάνεται στο σώμα εντολών της δεύτερης συνάρτησης, εκτελείται ακριβώς πριν εκκινήσει το παράθυρο. Τέλος, οι εντολές που υπάρχουν στην τρίτη συνάρτηση, εξάγουν τα αποτελέσματα τους στη γραμμή εντολών του

MATLAB. Στη συνέχεια, θα δούμε ότι για κάθε ένα στοιχείο ελέγχου που θα εισάγουμε θα δημιουργούνται επίσης κάποιες άλλες συναρτήσεις.

Έπειτα θα αρχίσουμε να εισάγουμε από την αριστερή κάθετη γραμμή εργαλείων τα διάφορα στοιχεία ελέγχου. Εισάγουμε πρώτα ένα πλαίσιο (frame) και αλλάζουμε το μέγεθος του από τις χαρακτηριστικές λαβές στα άκρα του, έτσι ώστε να καλύπτει τον χώρο στον οποίο θα βάλουμε τα push button. Για αισθητικούς λόγους, θα το χρωματίσουμε ανοιχτό κίτρινο. Αυτό θα γίνει αφού το επιλέξουμε (κάνοντας ένα αριστερό κλικ πάνω του και επιλέγοντας property inspector) και μετά μέσω της ιδιότητας Background Color, ενεργοποιούμε την παλέτα χρωμάτων όπως φαίνεται στις εικόνες 2.8 και 2.9.



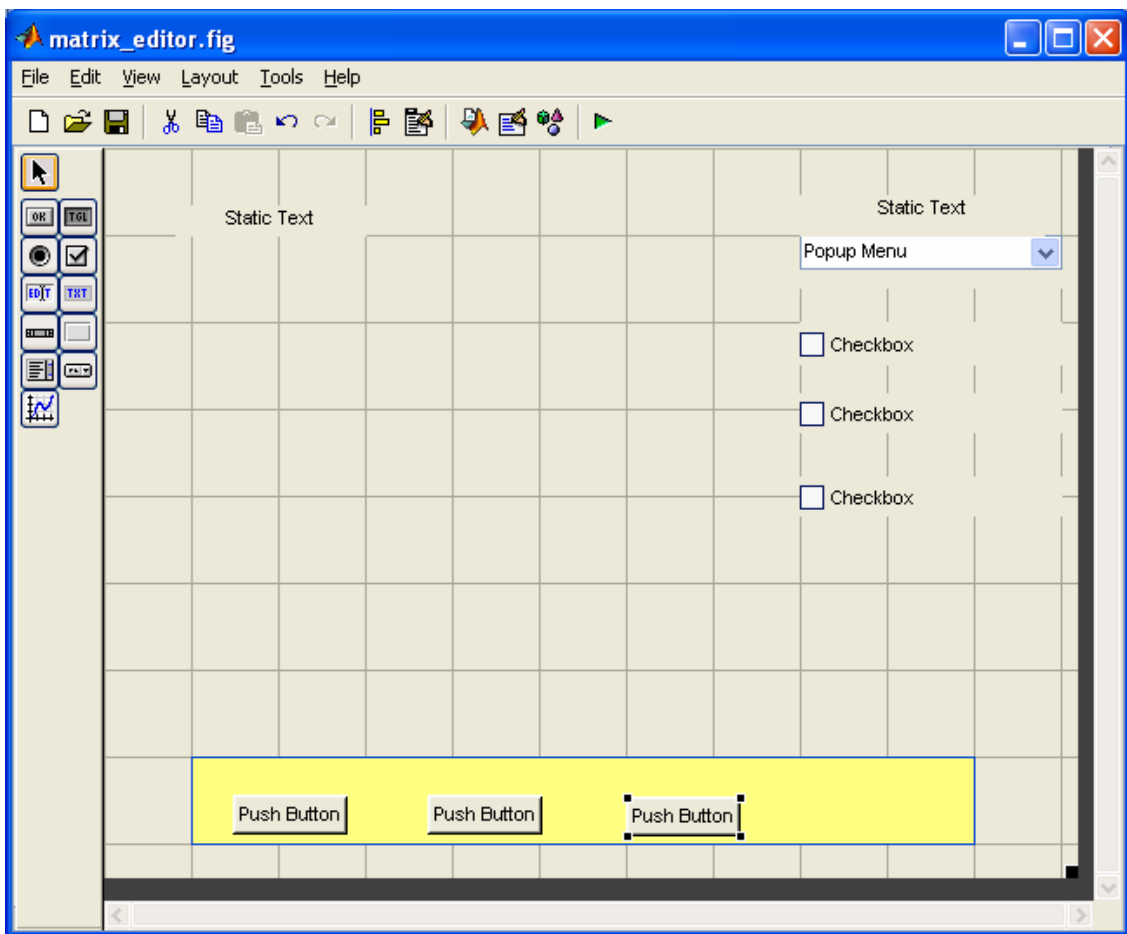
Εικόνα 2.8. Ιδιότητα Background Color



Εικόνα 2.9. Παλέτα χρωμάτων

Πιο συγκεκριμένα, επιλέγουμε το κουμπί της εικόνας 2.8 όπως φαίνεται και από το βέλος, το οποίο ενεργοποιεί το παράθυρο της εικόνας 2.9. Μέσω της παλέτας των διαφορετικών χρωμάτων ο χρήστης μπορεί εύκολα να επιλέξει το επιθυμητό χρώμα.

Τώρα εισάγουμε όλα στοιχεία ελέγχου που θα μας χρειαστούν. Αναλυτικά, τρία push button, ένα popup menu, τρία check boxes και τα static text που θα μας χρειαστούν.

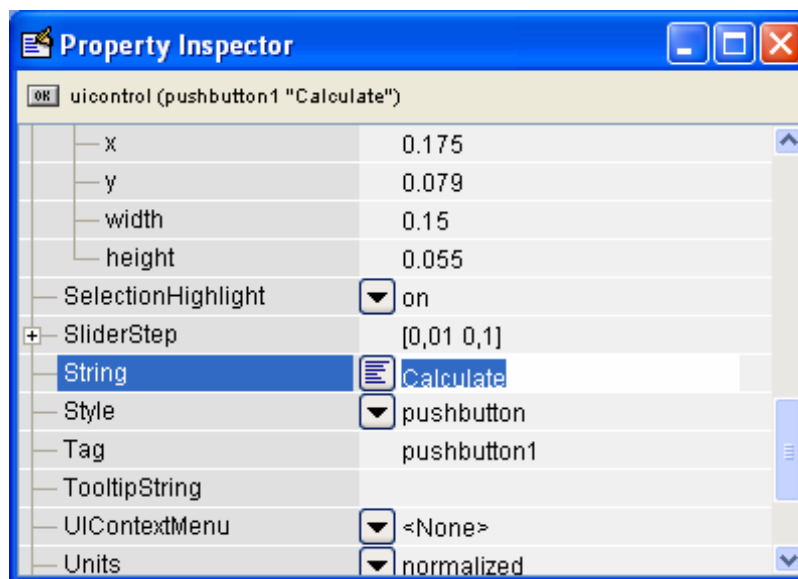


Εικόνα 2.10. Εισαγωγή των στοιχείων ελέγχου χωρίς αλλαγή ιδιοτήτων και χωρίς στοίχιση .

Συνεχίζουμε, αλλάζοντας, εάν θέλουμε, για όλα αυτά τα στοιχεία που έχουμε εισάγει το χρώμα τους κατά τη βούληση μας, καθώς επίσης και το κείμενο που θα φαίνεται επάνω τους αλλά και το μέγεθος των στοιχείων ελέγχου.

Επιπλέον, όπως προαναφέραμε, μπορούμε να τα στοιχίσουμε για να πετύχουμε ομοιόμορφη εμφάνιση.

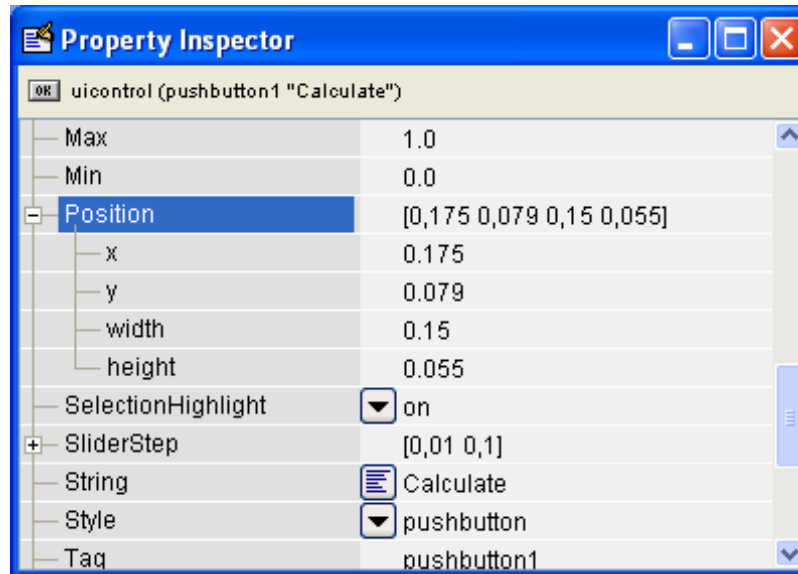
Είναι ιδιαίτερα σημαντικό να έχουμε υπόψη μας ότι για να αλλάξουμε τις ιδιότητες οποιουδήποτε στοιχείου ελέγχου, πρέπει προηγουμένως να το επιλέξουμε και έπειτα να ανοίξουμε το μενού property inspector. Στην εικόνα 2.11 φαίνεται από ποια ιδιότητα αλλάζουμε το κείμενο που φιλοξενείται επάνω σε κάποιο πεδίο κειμένου.



Εικόνα 2.11. Ιδιότητα string.

Μπορούμε να ομαδοποιήσουμε κάποια στοιχεία ελέγχου και να αλλάξουμε κάποιες κοινές ιδιότητες, όπως το μέγεθος των στοιχείων ελέγχου με την ιδιότητα position αλλά και την στοιχισή με τη βοήθεια του Align Objects .

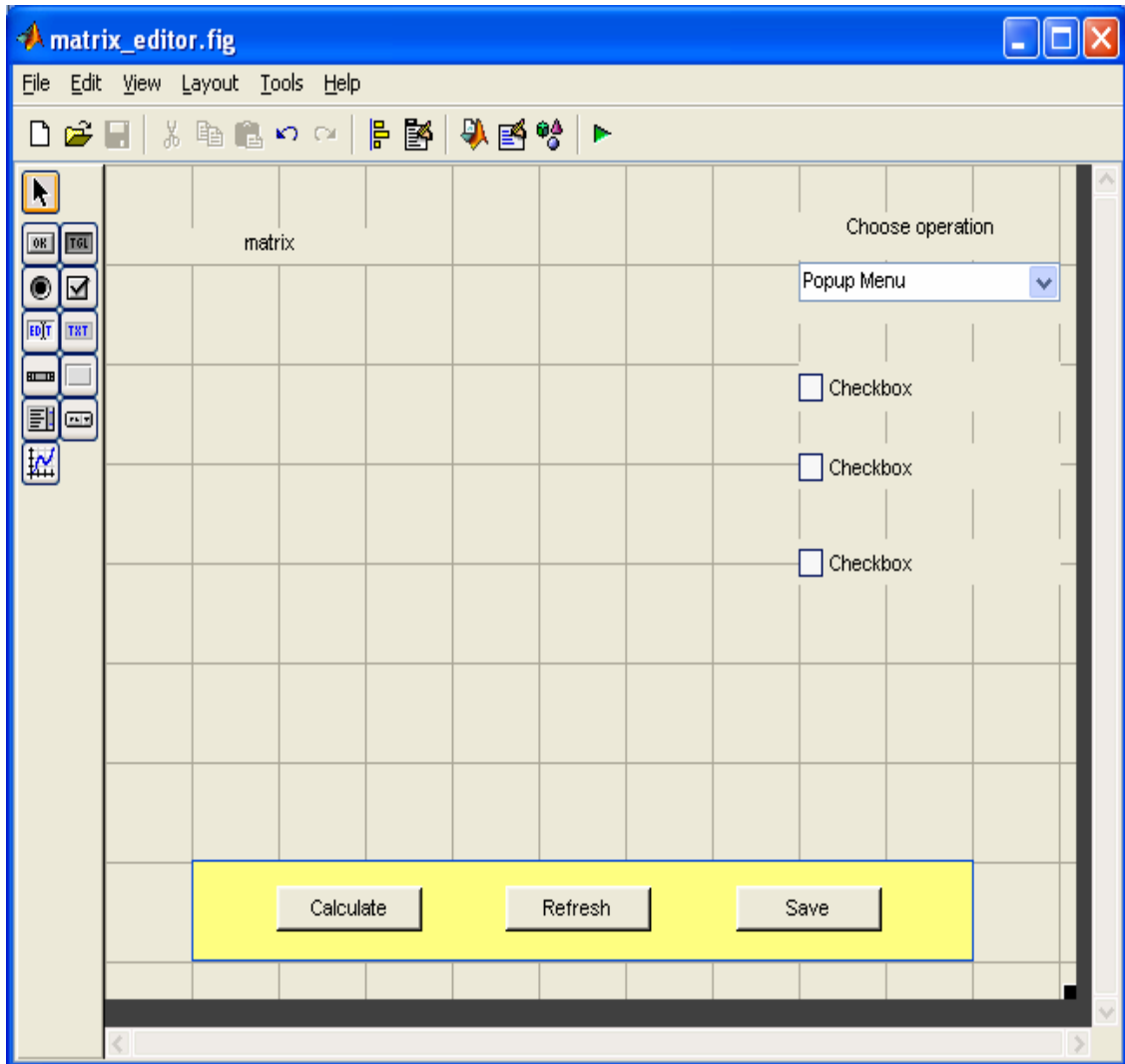
Η ιδιότητα position είναι εύκολο να χρησιμοποιηθεί αλλάζοντας και την ιδιότητα units, επιλέγοντας την τιμή normalized. Η ιδιότητα position έχει τέσσερις παραμέτρους x, y, width, height. Οι δυο πρώτες ρυθμίζουν την θέση του στοιχείου ελέγχου μέσα στο παράθυρο και οι επόμενες δυο ρυθμίζουν το μέγεθος του στοιχείου ελέγχου σε αναλογία με το παράθυρο. Η ιδιότητα units με επιλογή normalized ορίζει ως μονάδα μέτρησης ποσοστό επί του παραθύρου, δηλαδή εάν έχουμε τις τιμές που βλέπουμε στην εικόνα 2.12



Εικόνα 2.12. Ιδιότητα position.

το x είναι 0.175, το οποίο σημαίνει ότι η κάτω αριστερή γωνία του στοιχείου ελέγχου απέχει 17.5 % σε οριζόντια απόσταση από την κάτω αριστερή γωνία του παραθύρου. Το y είναι 0.079, το οποίο σημαίνει ότι η κάτω αριστερή γωνία του στοιχείου ελέγχου απέχει 7.9 % σε κατακόρυφη απόσταση από την κάτω αριστερή γωνία του παραθύρου. Αντίστοιχα το width είναι 0.15, το οποίο σημαίνει ότι το πλάτος του στοιχείου ελέγχου είναι 15% του πλάτους του παραθύρου. Τέλος, το height είναι 0.055, το οποίο σημαίνει ότι το ύψος του στοιχείου ελέγχου είναι 5.5 % του ύψους του παραθύρου.

Έως εδώ έχουμε ολοκληρώσει τη σχεδίαση του παραθύρου της εφαρμογής μας. Στην εικόνα 2.13, φαίνεται το παράθυρο μας έως αυτό το σημείο.



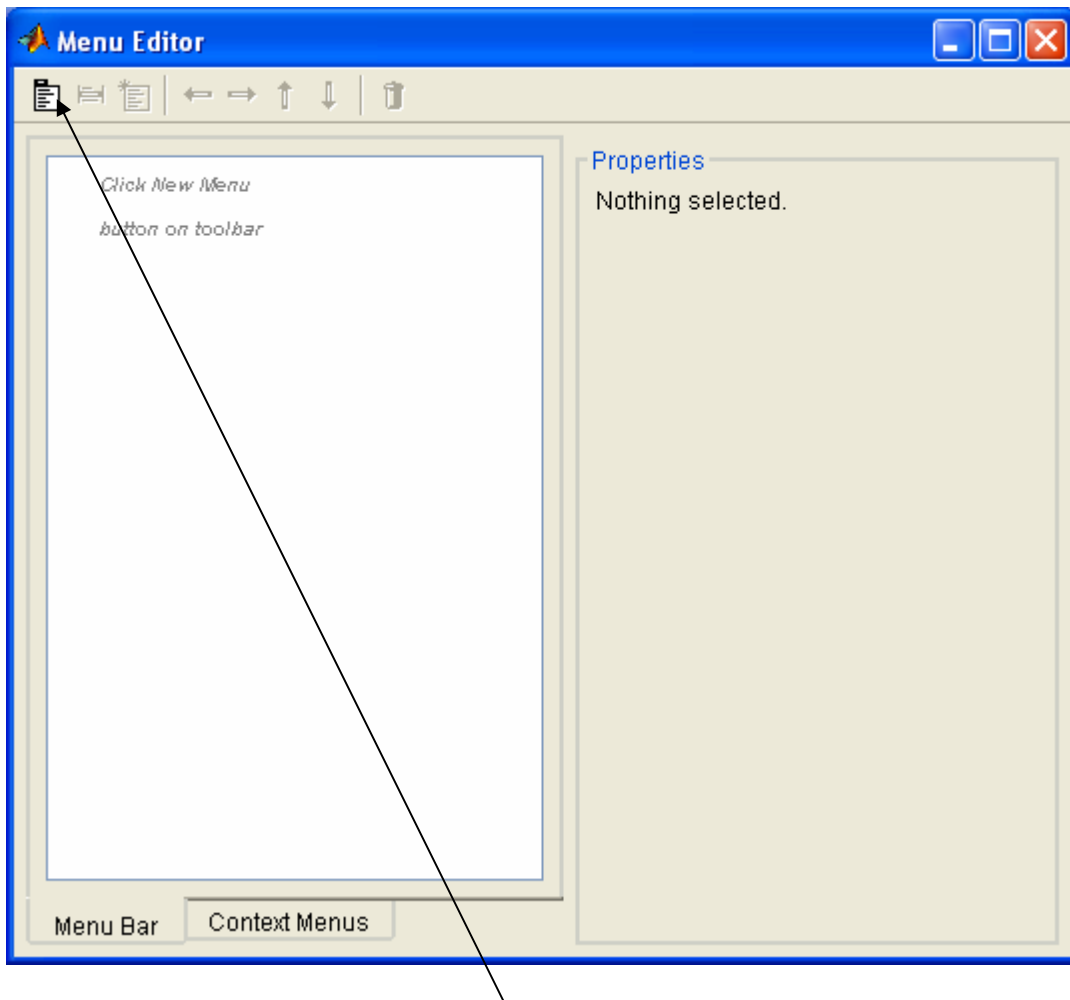
Εικόνα 2.13. Εισαγωγή των στοιχείων ελέγχου.

Μπορούμε για να επιταχύνουμε την εισαγωγή των στοιχείων ελέγχου, εάν για παράδειγμα κάνουμε αντιγραφή ενός κουμπού και επικόλληση σε ένα διπλανό σημείο. Φυσικά το παράθυρο δεν είναι ακόμη λειτουργικό. Πρέπει να προγραμματιστούν κατάλληλα τα διάφορα στοιχεία ελέγχου ώστε να μπορέσουν να λειτουργήσουν.

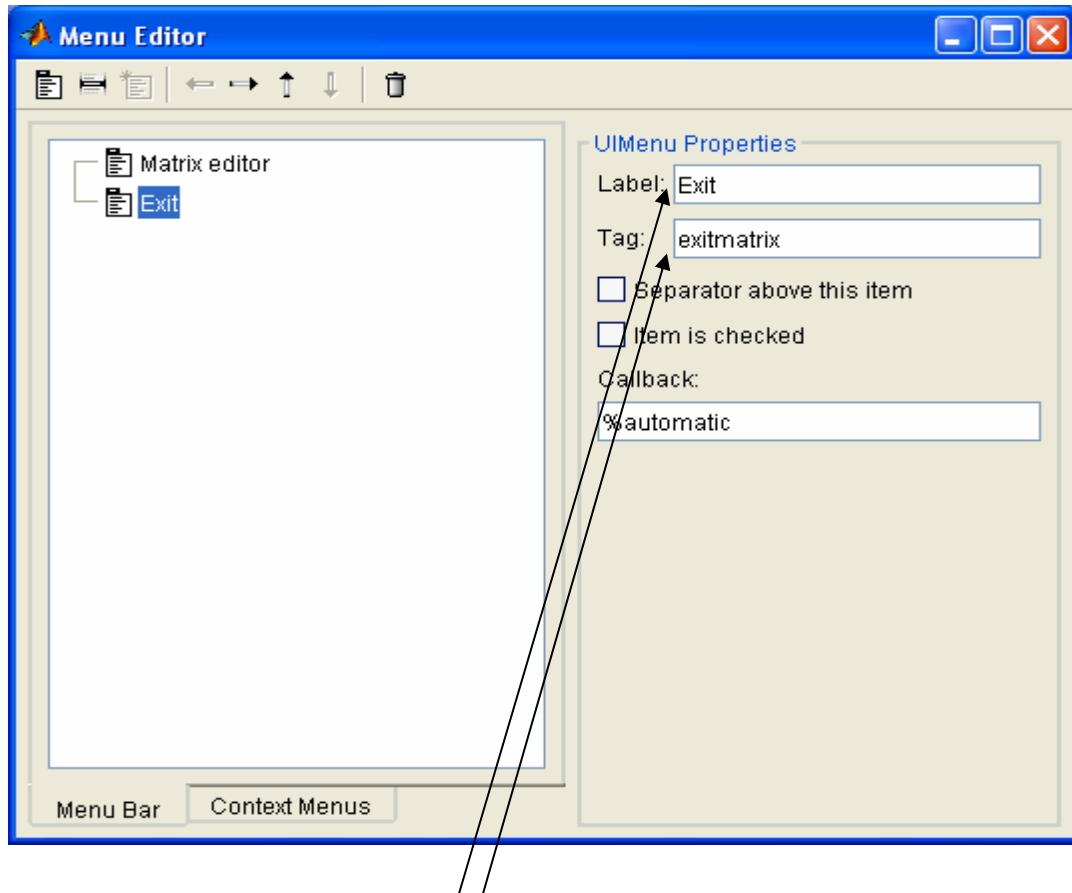
Εδώ θα πρέπει να δημιουργήσουμε και το μενού που μας χρειάζεται, όπως επίσης και τις επιλογές που θα έχουμε στο popup menu. Θα πρέπει να ορίσουμε την ιδιότητα `visible` στην επιλογή `off` στα `checkbox`, γιατί η εμφάνιση τους δε θα είναι σε όλες τις επιλογές, και την ιδιότητα `enable` στην τιμή `off` στα στοιχεία ελέγχου

push button και στο popup menu, τα οποία θα ενεργοποιούνται ανάλογα με τις επιλογές του χρήστη.

Το μενού το δημιουργούμε με τον τρόπο που έχουμε προαναφερθεί, ενεργοποιώντας τον menu editor (Εικόνα 2.14) και επιλέγοντας το κουμπί που φαίνεται στην εικόνα 2.14, δημιουργούμε τα μενού που θέλουμε, ορίζοντας τις ιδιότητες label και tag όπως φαίνεται στην εικόνα 2.15 .

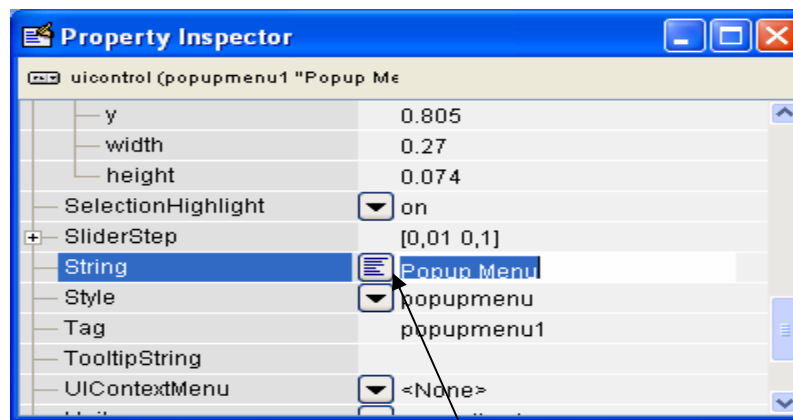


Εικόνα 2.14 Menu Editor



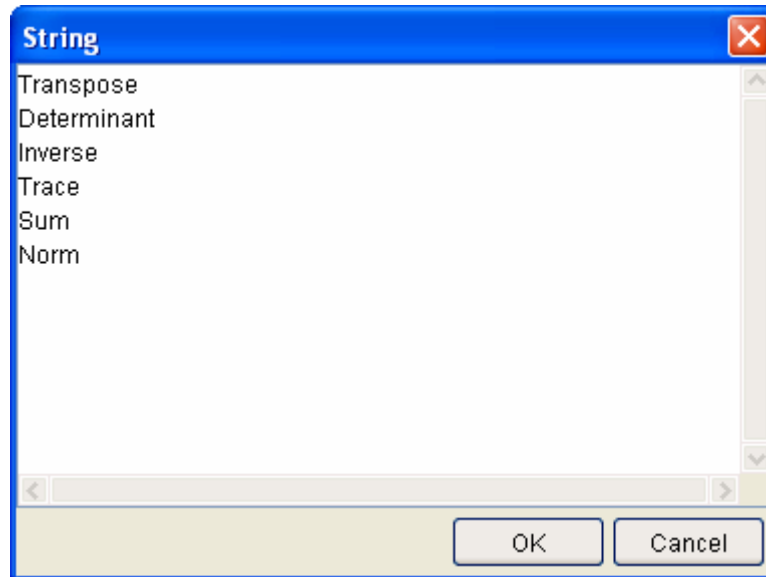
Εικόνα 2.15 Δημιουργία και ορισμός ιδιοτήτων ενός Menu

Επίσης στο popup menu θέλουμε να εισάγουμε όλες τις επιλογές που μας χρειάζονται (πράξεις που θα εκτελεί ο matrix editor). Επιλέγουμε την ιδιότητα string και ειδικότερα το κουμπί που βρίσκεται σ' αυτή την ιδιότητα (Εικόνα 2.16)



Εικόνα 2.16 Ιδιότητα string

και στο παράθυρο που μας εμφανίζεται συμπληρώνουμε της επιλογές που θέλουμε (Εικόνα 2.17) .



Εικόνα 2.17 Ιδιότητα string

2.2.3 Προγραμματισμός εφαρμογής

Βλέποντας σε αυτό το σημείο το αντίστοιχο m-file, παρατηρεί κανείς ότι έχουν ενσωματωθεί πολλές ακόμη συναρτήσεις. Θυμίζουμε στον αναγνώστη ότι για το κάθε ένα στοιχείο ελέγχου έχουν εισέλθει κάποιες συναρτήσεις, όπως περιγράφονται παραπάνω.

Θα ξεκινήσουμε από τον προγραμματισμό των συναρτήσεων για το μενού, επειδή πρέπει πρώτα να δημιουργηθεί ένας πίνακας, στον οποίο θα γίνονται οι διάφορες πράξεις .

Μέσα στο m-file υπάρχουν οι αντίστοιχες συναρτήσεις:

1. *function matrixeditor_Callback(hObject, eventdata, handles)*
2. *function exitmatrix_Callback(hObject, eventdata, handles)*

οι οποίες είναι οι συναρτήσεις επιστροφής για το μενού matrix editor και για το μενού exit αντίστοιχα.

Στην πρώτη θα βάλουμε τις εντολές που χρειάζονται για να δημιουργηθεί ένας τετραγωνικός πίνακας διαστάσεων το πολύ έως 5 x 5.

Πρώτα θα δημιουργηθεί ένα παράθυρο διαλόγου (εισόδου τιμών `inputdlg`), στο οποίο θα ζητείται από τον χρήστη να εισάγει τη διάσταση του τετραγωνικού πίνακα. Αυτό υλοποιείται με την παρακάτω εντολή

```
answer=inputdlg('give the dimension of square matrix ','define of dimension');
```

Στην μεταβλητή `answer` έχει καταχωρηθεί η διάσταση του τετραγωνικού πίνακα σε μορφή cell array. Έτσι με την εντολή

```
dim=str2num(answer{1});
```

παίρνουμε την διάσταση του τετραγωνικού πίνακα .

Στην συνέχεια με προγραμματισμό θα δημιουργήσουμε τα edit box στα οποία θα εισάγει ο χρήστης τις τιμές του πίνακα.

```
for i=1:dim
    for j=1:dim
        a(i,j)=uicontrol('style','edit','backgroundcolor','w','units','normalized',...
            'position',[0.09*j 0.8-0.1*i 0.08 0.05]);
    end
end
handles.a=a;
set(handles.pushbutton1,'enable','on');
set(handles.pushbutton2,'enable','off');
set(handles.pushbutton3,'enable','off');
set(handles.popupmenu1,'enable','on');
guidata(hObject,handles);
```

Η εντολή `uicontrol` δημιουργεί ένα στοιχείο ελέγχου ορίζοντας ταυτόχρονα και κάποιες ιδιότητες του. Επίσης έχει δημιουργηθεί το αντικείμενο `handles.a`, το οποίο μπορούμε να το χρησιμοποιήσουμε σε όποια συνάρτηση θέλουμε. Αλλάξαμε την ιδιότητα `enable` σε κάποια στοιχεία ελέγχου και με την εντολή `guidata(hObject,handles);` αποθηκεύουμε στην προσωρινή μνήμη (όσο το αρχικό παράθυρο είναι ανοικτό) τα `handles` που έχουν δημιουργηθεί ή και τις όποιες αλλαγές έχουν γίνει στις ιδιότητες τους .

Στην δεύτερη συνάρτηση θα βάλουμε τις εντολές που χρειάζονται για την έξοδο από την εφαρμογή. Θα δημιουργηθεί ένα παράθυρο διαλόγου (ερώτησης `questdlg`) στο οποίο θα ερωτάται ο χρήστης εάν θέλει να τερματίσει την εφαρμογή .

```
h=questdlg('exit from program?');  
switch h  
    case 'Yes'  
        closereq;  
    end
```

Η εφαρμογή τερματίζεται με την εντολή `closereq` .

Συνεχίζουμε με τον προγραμματισμό του `popup menu`, αλλά και των `check boxes`, τα οποία καθορίζουν την πράξη που θα εκτελεστεί (`popup menu`) καθώς και τις διάφορες παραμέτρους κάθε πράξης (`check boxes`), εάν υπάρχουν.

Μέσα στο `m-file` υπάρχουν οι αντίστοιχες συναρτήσεις :

1. `function popupmenu1_CreateFcn(hObject, eventdata, handles)`
2. `function popupmenu1_Callback(hObject, eventdata, handles)`
3. `function checkbox1_Callback(hObject, eventdata, handles)`
4. `function checkbox2_Callback(hObject, eventdata, handles)`
5. `function checkbox3_Callback(hObject, eventdata, handles)`

Η πρώτη είναι συνάρτηση δημιουργίας του `popup menu` ενώ οι άλλες είναι συναρτήσεις επιστροφής .

Στην δεύτερη συνάρτηση θα βάλουμε τις εντολές που χρειάζονται για την εμφάνιση ή όχι των `check boxes`, ανάλογα με την πράξη που έχει επιλέξει ο χρήστης.

```
pop=get(handles.popupmenu1,'value');  
switch pop  
    case 1  
        set(handles.checkbox1,'visible','off');  
        set(handles.checkbox2,'visible','off');
```

```

    set(handles.checkbox3,'visible','off');
case 2
    set(handles.checkbox1,'visible','off');
    set(handles.checkbox2,'visible','off');
    set(handles.checkbox3,'visible','off');
case 3
    set(handles.checkbox1,'visible','off');
    set(handles.checkbox2,'visible','off');
    set(handles.checkbox3,'visible','off');
case 4
    set(handles.checkbox1,'visible','off');
    set(handles.checkbox2,'visible','off');
    set(handles.checkbox3,'visible','off');
case 5
    set(handles.checkbox1,'string','sum of rows');
    set(handles.checkbox1,'visible','on');
    set(handles.checkbox2,'string','sum of columns');
    set(handles.checkbox2,'visible','on');
case 6
    set(handles.checkbox1,'string','1-norm ');
    set(handles.checkbox1,'visible','on');
    set(handles.checkbox2,'string','infinity norm');
    set(handles.checkbox2,'visible','on');
    set(handles.checkbox3,'string','2-norm');
    set(handles.checkbox3,'visible','on');
end
guidata(hObject, handles);

```

Τα check boxes θα εμφανιστούν στην πράξη των αθροισμάτων, με επιλογή για γραμμές ή στήλες και στην πράξη των νορμών, με επιλογή 1-norm, infinity norm και 2-norm. Στις παραπάνω εντολές κάνουμε ορατά ή μη ορατά τα

αντίστοιχα checkboxes και τοποθετούμε τα κατάλληλα κείμενα στο κάθε ένα από αυτά, ανάλογα την πράξη.

Τελειώνουμε με τον προγραμματισμό των push button, τα οποία, εφόσον πατηθούν, θα εκτελούν κάποιες λειτουργίες .

Μέσα στο m - file υπάρχουν οι αντίστοιχες συναρτήσεις :

1. *function pushbutton1_Callback(hObject, eventdata, handles)*
2. *function pushbutton2_Callback(hObject, eventdata, handles)*
3. *function pushbutton3_Callback(hObject, eventdata, handles)*

Και οι τρεις συναρτήσεις είναι οι αντίστοιχες συναρτήσεις επιστροφής των push button .

Στην **πρώτη** θα βάλουμε τις εντολές που χρειάζονται για να εκτελεί την λειτουργία calculate, δηλαδή την εκτέλεση της πράξης που επιλέχθηκε από τον χρήστη και την ενεργοποίηση των λειτουργιών refresh και save .

```
a=handles.a;
dim=length(a);
for i=1:dim
    for j=1:dim
        mat(i,j)=str2num(get(handles.a(i,j),'string'));
    end
end
handles.mat=mat;
pop=get(handles.popupmenu1,'value');
switch pop
    case 1
        result=mat.';
    case 2
        result=det(mat);
    case 3
        result=inv(mat);
```

```
case 4
    result=trace(mat);
case 5
    ans1=get(handles.checkbox1,'value');
    ans2=get(handles.checkbox2,'value');
    if ans1 & ~ans2
        result=sum(mat,2)
    elseif ~ans1 & ans2
        result=sum(mat,1)
    else
        warndlg('Must check one box');
        return;
    end
case 6
    ans1=get(handles.checkbox1,'value');
    ans2=get(handles.checkbox2,'value');
    ans3=get(handles.checkbox3,'value');
    if ans1 & ~ans2 & ~ans3
        result=norm(mat,1);
    elseif ~ans1 & ans2 & ~ans3
        result=norm(mat,inf);
    elseif ~ans1 & ~ans2 & ans3
        result=norm(mat,inf);
    else
        warndlg('Must check one box');
        return;
    end
end
handles.result=result;
set(handles.a,'visible','off');
[rows,columns]=size(result);
```

```
if rows==1 & columns==1
    b=uicontrol('style','text','backgroundcolor','w','units','normalized',...
        'position',[0.09 0.7 0.08 0.05]);
    handles.b=b;
    set(handles.b,'string',num2str(result));
else
    for i=1:rows
        for j=1:columns
            b(i,j)=uicontrol('style','text','backgroundcolor','w','units','normalized',...
                'position',[0.09*j 0.8-0.1*i 0.08 0.05]);
            handles.b=b;
            set(handles.b(i,j),'string',num2str(result(i,j)));
        end
    end
end
end
set(handles.pushbutton1,'enable','off');
set(handles.pushbutton2,'enable','on');
set(handles.pushbutton3,'enable','on');
set(handles.popupmenu1,'enable','off');
guidata(hObject, handles);
```

Τα αποτελέσματα εμφανίζονται σε static text στο handles.b ενώ γίνεται έλεγχος εάν έχει επιλεγθεί (στις πράξεις που υπάρχουν) κάποιο check box. Στην περίπτωση που δεν έχει επιλεγθεί αλλά και στην περίπτωση που έχουν επιλεγθεί δυο ή και περισσότερα, θα δημιουργείται ένα παράθυρο διαλόγου (προσοχής warndlg) στο οποίο θα ενημερώνεται ο χρήστης για το λάθος του.

Στην **δεύτερη** θα βάλουμε τις εντολές που χρειάζονται για να εκτελεί την λειτουργία refresh, δηλαδή να σβήνει το αποτέλεσμα της πράξης, να εξαφανίζει τα check boxes, να επαναφέρει στο παράθυρο τον πίνακα που έχει εισάγει ο χρήστης και να ενεργοποιεί τα κατάλληλα στοιχεία ελέγχου.

```
set(handles.b,'visible','off');
```

```

set(handles.checkbox1,'visible','off');
set(handles.checkbox2,'visible','off');
set(handles.checkbox3,'visible','off');
set(handles.a,'visible','on');
set(handles.pushbutton1,'enable','on');
set(handles.pushbutton2,'enable','off');
set(handles.pushbutton3,'enable','off');
set(handles.popupmenu1,'enable','on');
guidata(hObject, handles);

```

Στην **τρύτη** θα βάλουμε τις εντολές που χρειάζονται για να εκτελεί την λειτουργία save, δηλαδή να αποθηκεύει τον πίνακα και το αποτέλεσμα σε ένα αρχείο κειμένου (matrix.txt) .

```

fid=fopen('matrix.txt','w');
fprintf(fid,' INPUT MATRIX \n\n');
mat=handles.mat;
[rows,columns]=size(mat);
for i=1:rows
    for j=1:columns
        fprintf(fid,' %6.f',mat(i,j));
    end
    fprintf(fid,'\n');
end
fprintf(fid,'\n RESULT \n');
result=handles.result;
[rows,columns]=size(result);
for i=1:rows
    for j=1:columns
        fprintf(fid,' %6.f',mat(i,j));
    end
    fprintf(fid,'\n');
end

```

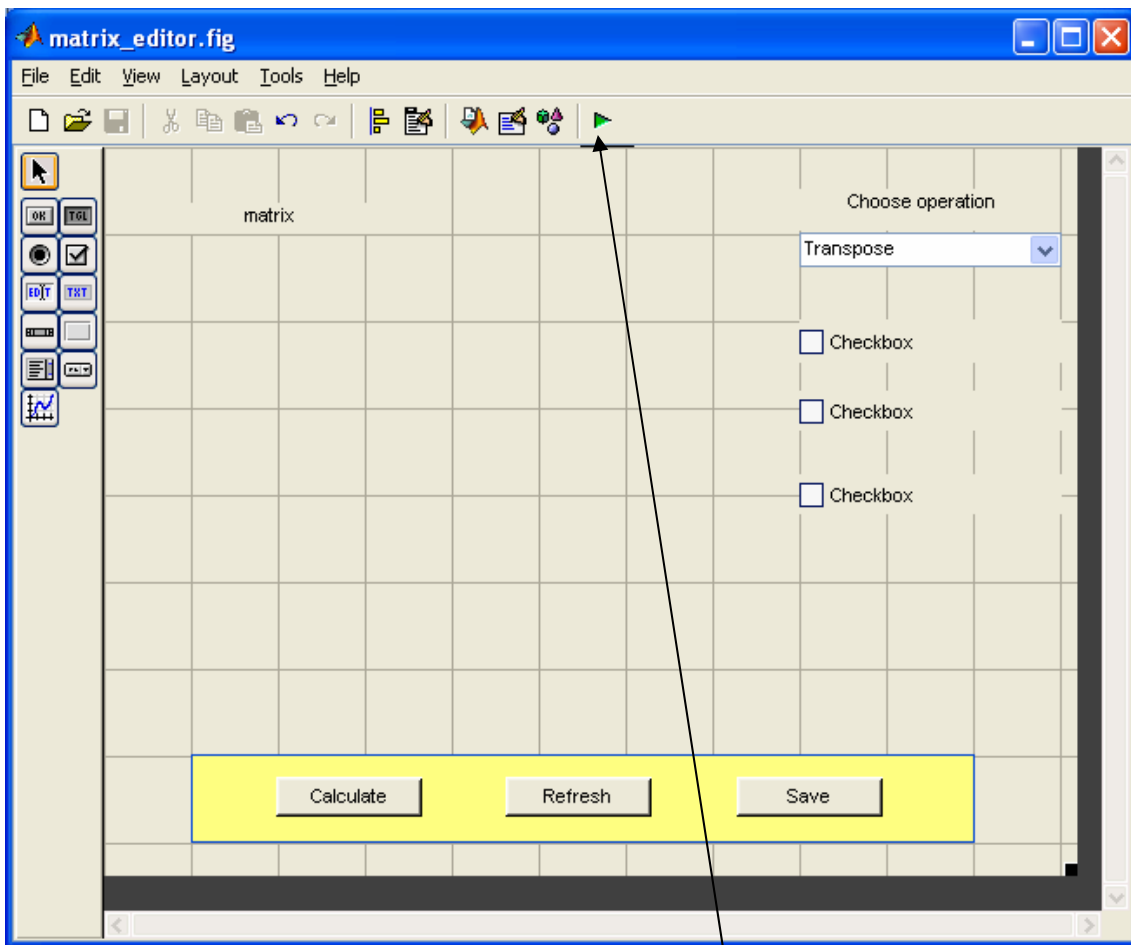
end

fclose(fid);

2.2.4 Εκτέλεση εφαρμογής

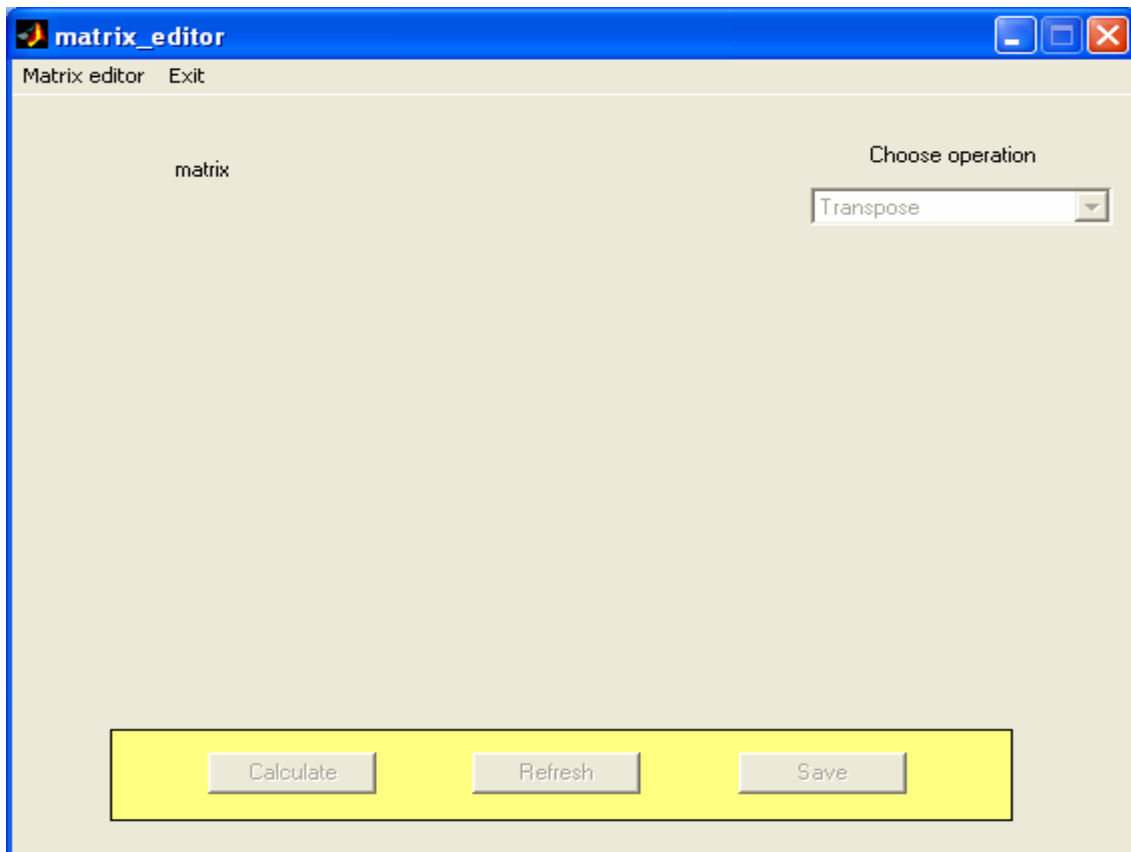
Η εκτέλεση της εφαρμογής μπορεί να γίνει με δυο τρόπους :

- α) Γράφοντας στο command line του MATLAB το όνομα του αρχείου
>>matrix_editor
- β) Με το κουμπί run στην γραμμή εργαλείων του παραθύρου όπως φαίνεται στην εικόνα 2.18



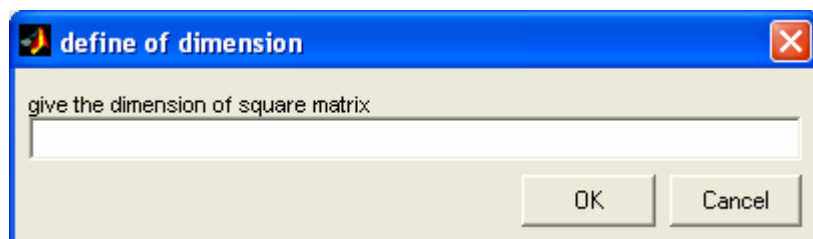
Εικόνα 2.18 Εκτέλεση εφαρμογής

Εκτελώντας την εφαρμογή εμφανίζεται το αρχικό παράθυρο, στο οποίο έχουμε δυνατότητα επιλογής μονό στο μενού -όλα τα αλλά στοιχεία ελέγχου είναι ανενεργά- και ο χρήστης οδηγείται στην επιλογή κάποιου από τα δυο μενού ή της εξόδου ή της εισαγωγής νέου τετραγωνικού πίνακα όπως φαίνεται στην εικόνα 2.19



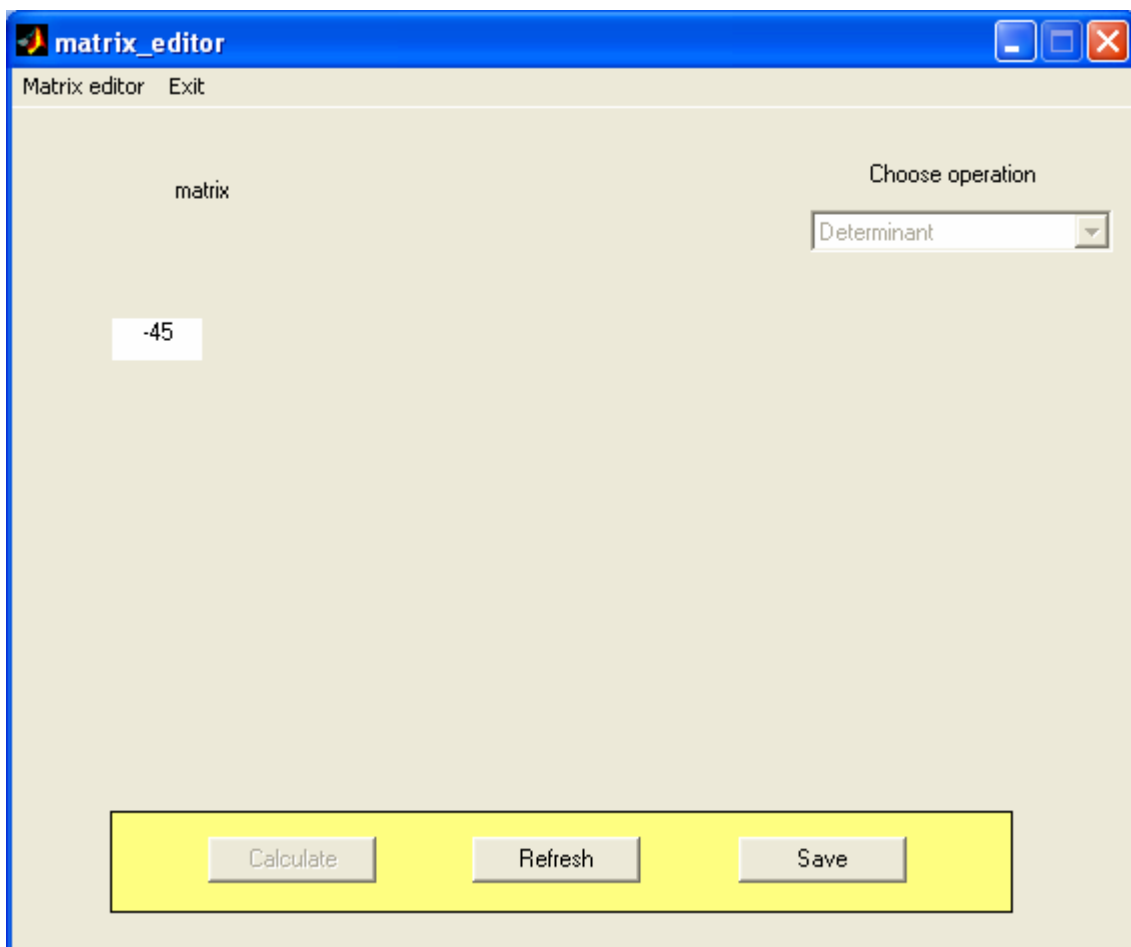
Εικόνα 2.19 Αρχικό παράθυρο

Επιλέγοντας ο χρήστης το μενού matrix editor εμφανίζεται ένα νέο παράθυρο, στο οποίο ορίζει την διάσταση του τετραγωνικού πίνακα (εικόνα 2.20).



Εικόνα 2.20 Ορισμός διάστασης πίνακα

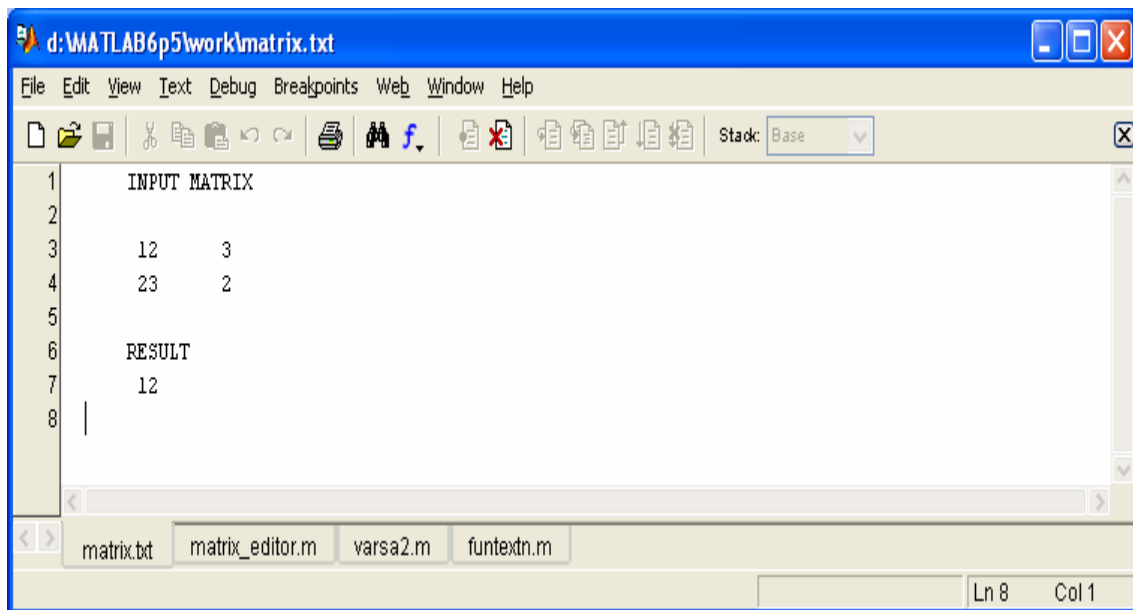
Αφού ο χρήστης εισάγει έναν αριθμό και πατήσει OK, επιστρέφουμε στο αρχικό παράθυρο, στο οποίο έχουν εμφανιστεί τα edit box που χρειάζονται για την εισαγωγή τιμών στον πίνακα και καθίσταται ενεργό το push button calculate και το popup menu στο οποίο επιλέγει ο χρήστης την πράξη που επιθυμεί. Εφόσον εισαγάγει τιμές στα edit box και επιλέξει κάποια πράξη, μπορεί να πατήσει το push button calculate .



Εικόνα 2.21 Calculate

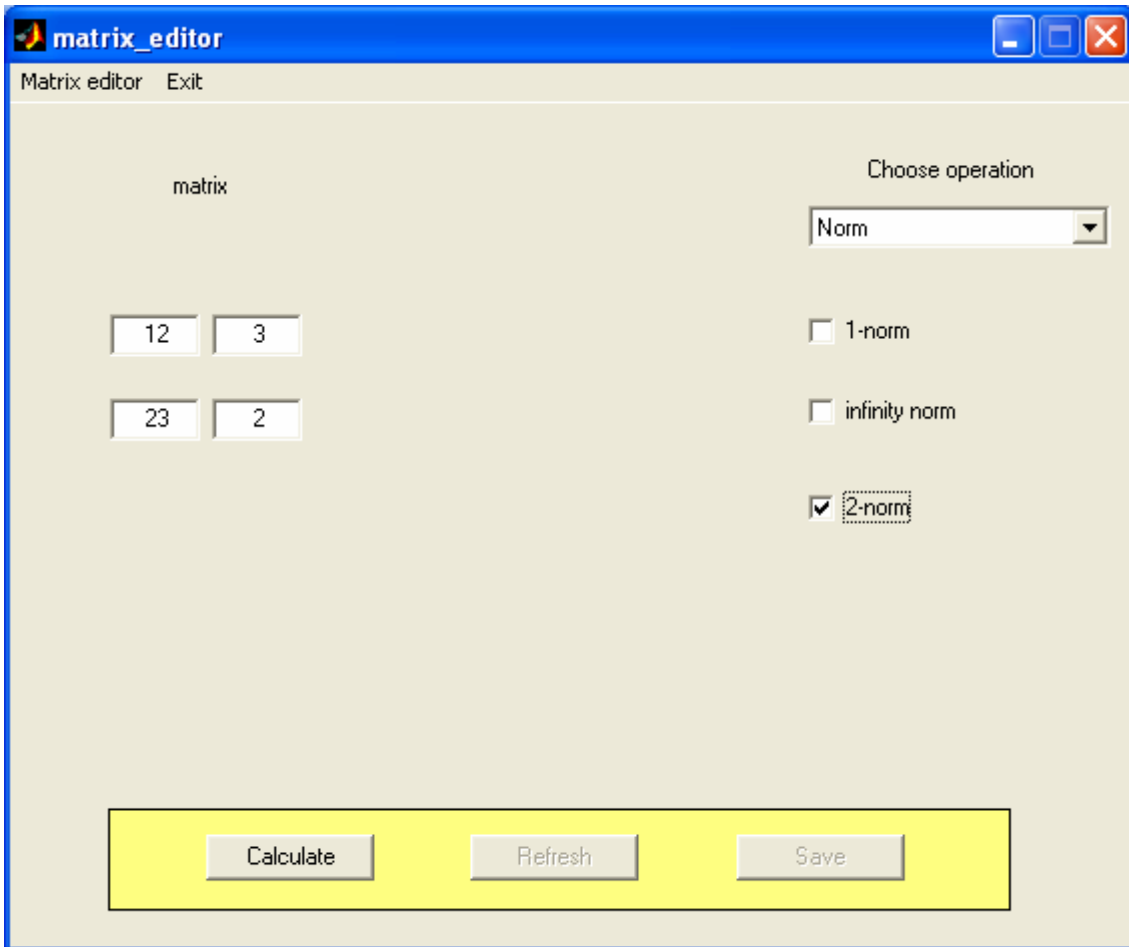
Στην εικόνα 2.21 βλέπουμε ότι μετά το πάτημα του push button calculate εμφανίζεται σε ένα static text το αποτέλεσμα της πράξης και το push button calculate και το popup menu είναι ανενεργά ενώ τα push button refresh και save ενεργά. Το push button save μας δίνει τη δυνατότητα να αποθηκεύσουμε τον

πίνακα και το αποτέλεσμα σε αρχείο κειμένου (matrix.txt)



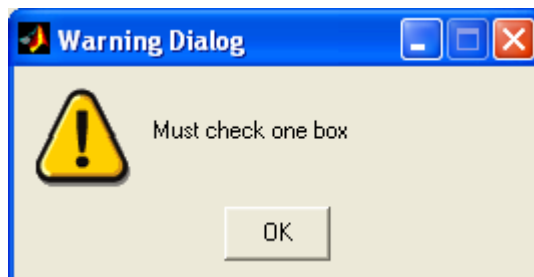
Εικόνα 2.22 Αρχείο matrix.txt

Το push button refresh σβήγει το αποτέλεσμα, επαναφέρει τον πίνακα στο παράθυρο, κάνει ενεργά το push button calculate και το popup menu, ενώ καθιστά ανενεργά τα push button refresh και save. Πλέον υπάρχει η δυνατότητα επιλογής νέας πράξης με τον ίδιο πίνακα ή η εισαγωγή νέου πίνακα. Στην περίπτωση που ο χρήστης επιλέξει sum ή norm θα εμφανιστούν και τα αντίστοιχα check boxes (εικόνα 2.23).



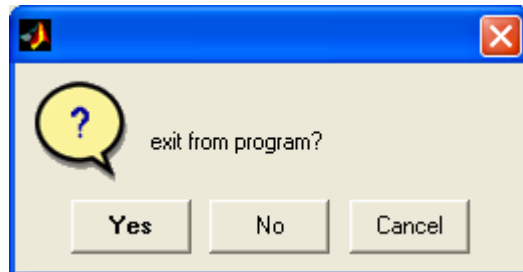
Εικόνα 2.23 Επιλογή νόρμας

Στην περίπτωση που ο χρήστης δεν επιλέξει κάποιο check box ή επιλέξει δυο εμφανίζεται ένα μήνυμα λάθους (εικόνα 2.24)



Εικόνα 2.24

Τέλος η εφαρμογή τερματίζει όταν ο χρήστης επιλέξει το μενού exit, στο οποίο όμως θα εμφανιστεί ένα παράθυρο διαλόγου στο οποίο ερωτάτε ο χρήστης εάν θέλει να τερματίσει την εφαρμογή (εικόνα 2.25).



Εικόνα 2.25

3^ο ΚΕΦΑΛΑΙΟ

ΑΝΑΠΤΥΞΗ ΔΙΑΔΡΑΣΤΙΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ (GUI) ΓΙΑ ΤΗΝ ΑΝΑΛΥΣΗ ΚΑΙ ΣΥΝΘΕΣΗ ΠΟΛΥΜΕΤΑΒΛΗΤΩΝ ΣΥΣΤΗΜΑΤΩΝ ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ

Η εφαρμογή αυτή έχει γίνει με σκοπό την εύκολη χρήση σε διαδραστικό (παραθυρικό) περιβάλλον MATLAB (GUI) των συναρτήσεων του πολυωνυμικού πακέτου POLYX.

Στις παρακάτω ενότητες δίνονται οι οδηγίες χρήσης της εφαρμογής και γίνεται μια λεπτομερή ανάλυση του προγραμματισμού της εφαρμογής.

3.1 ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

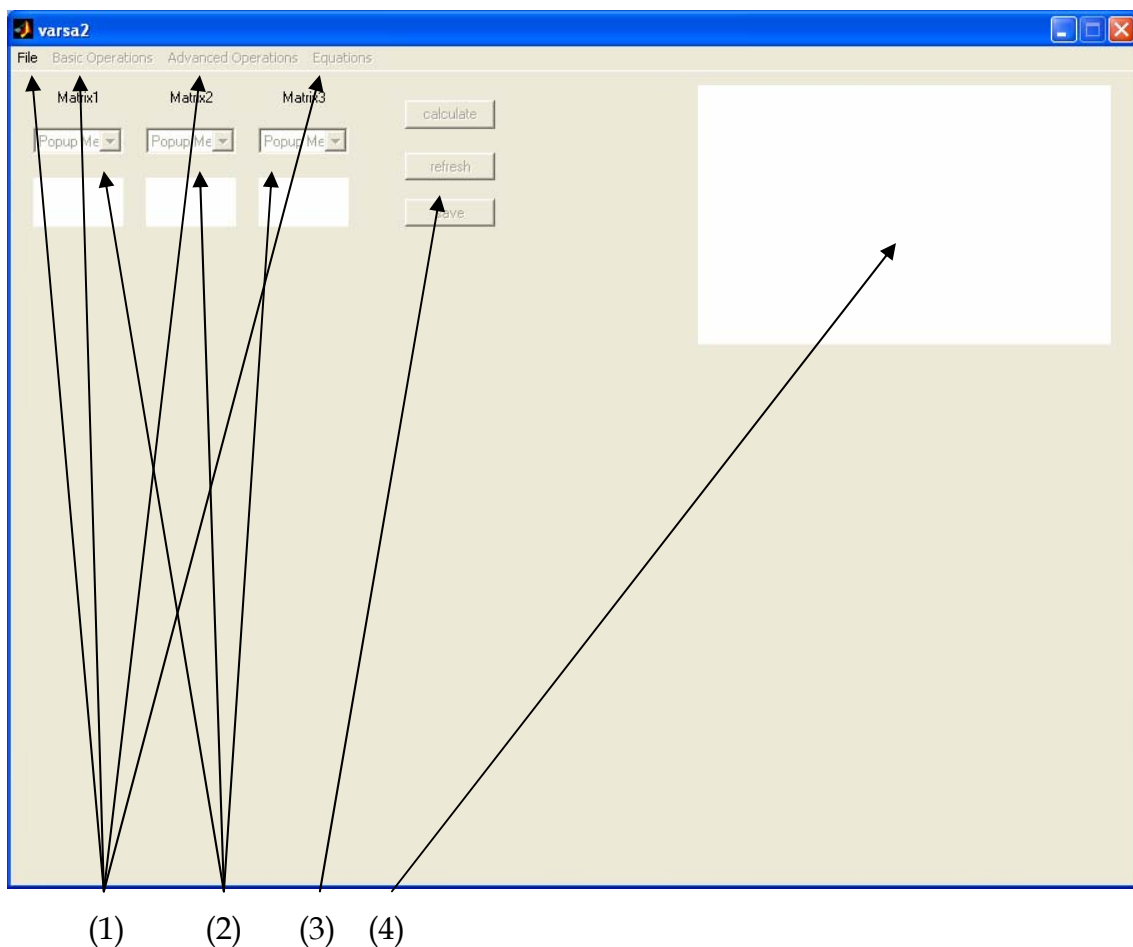
Η εφαρμογή είναι κατασκευασμένη έτσι ώστε να οδηγεί τον χρήστη εύκολα στην επιλογή που επιθυμεί χρησιμοποιώντας τα μενού επιλογών και ανάλογα την επιλογή του να εμφανίζεται ή να γίνεται ενεργό όποιο στοιχείο ελέγχου είναι απαραίτητο για την λειτουργία που επέλεξε. Μετά την εκκίνηση της εφαρμογής ο χρήστης έχει τη δυνατότητα να επιλέξει μόνο το μενού file και τα υπό-μενού Matrix Editor, Workspace, Load Work και Exit. Με την εκκίνηση της εφαρμογής οδηγείται στις παρακάτω λειτουργίες :

- Με την επιλογή Matrix Editor να δημιουργήσει νέους πολυωνυμικούς ή αριθμητικούς πίνακες .
- Με την επιλογή Workspace να εισάγει πίνακες από το workspace του Command Window του MATLAB.
- Με την επιλογή Load Work να φορτώσει τους πίνακες που είχαν αποθηκευτεί από την τελευταία αποθηκευμένη εργασία .
- Με την επιλογή Exit να τερματίσει την εφαρμογή .

Αφού ο χρήστης εισάγει κάποιους πολυωνυμικούς ή αριθμητικούς πίνακες μπορεί να εκτελέσει τις πράξεις που περιγράφονται στα άλλα τρία μενού επιλογών (εικόνα 3.1 (1)). Για κάθε πράξη που επιλέγει ο χρήστης, ενεργοποιούνται, δηλαδή, γίνονται ορατά ή ενεργά όλα τα στοιχεία ελέγχου που είναι απαραίτητα για την εκτέλεση της πράξης. Ο χρήστης, μετά την επιλογή πράξης από το αντίστοιχο μενού ή υπό-μενού, έχει τη δυνατότητα να επιλέξει μέσα από τα popup menu που είναι ενεργά τον πίνακα ή τους πίνακες που επιθυμεί (εικόνα 3.1 (2)), να καθορίσει τις διάφορες παραμέτρους της πράξης (αν υπάρχουν), να εκτελέσει την επιθυμητή λειτουργία πατώντας τα αντίστοιχα push button (εικόνα 3.1 (3)) καθώς και να βλέπει την αντίστοιχη βοήθεια για την πράξη που επέλεξε στο static text (εικόνα 3.1 (4)). Για κάθε πίνακα που επιλέγει ο χρήστης εμφανίζεται το μέγεθος και η κλάση του επιλεγμένου πίνακα στο αντίστοιχο static text (εικόνα 3.1 (2)). Το push button calculate γίνεται ενεργό με την επιλογή της πράξης ενώ τα push button save και

refresh γίνονται ενεργά αφού πρώτα πατηθεί το push button calculate. Όταν το αποτέλεσμα της πράξης είναι πίνακας, δίνεται η δυνατότητα στον χρήστη να αποθηκεύει το αποτέλεσμα με το πάτημα του push button save, έτσι ώστε να μπορεί να το χρησιμοποιήσει σε κάποια άλλη πράξη. Ο χρήστης έχει και τη δυνατότητα του ανανέωσης (refresh) των δεδομένων κάποιας πράξης με το πάτημα του push button refresh. Επίσης, ο χρήστης έχει τη δυνατότητα να διαχειριστεί τους πίνακες, δηλαδή να εισάγει νέους πίνακες, να διαγράψει πίνακες και να αποθηκεύσει την εργασία του ώστε να μπορεί να την δουλέψει κάποια άλλη στιγμή.

Η εφαρμογή είναι κατασκευασμένη έτσι ώστε εάν ο χρήστης κάνει κάποιο λάθος είτε ως προς την λειτουργία της εφαρμογής είτε ως προς την ορθή καταχώρηση πινάκων στις πράξεις, να εμφανίζεται ένα παράθυρο διαλόγου το οποίο θα καθοδηγεί τον χρήστη προς την σωστή κατεύθυνση .



Εικόνα 3.1 Αρχικό παράθυρο

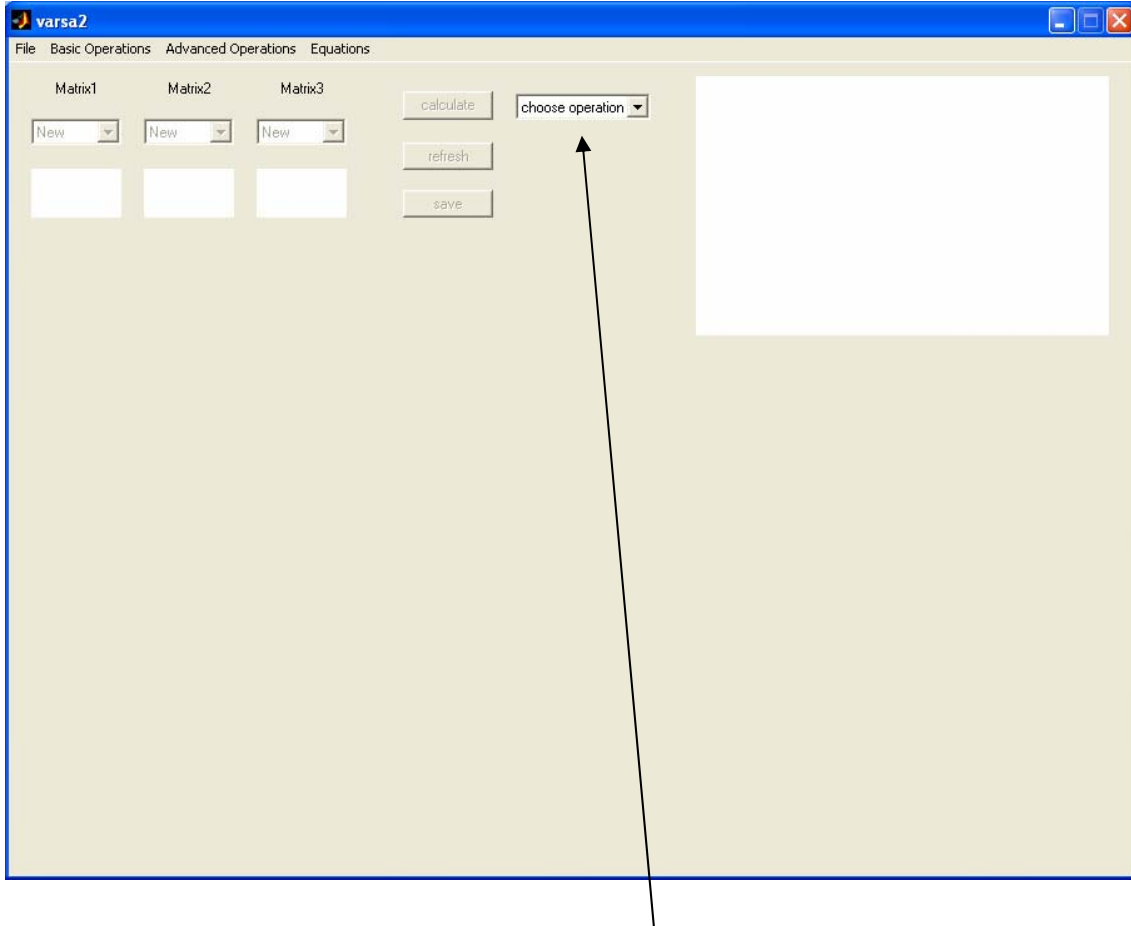
Το αρχικό παράθυρο της εφαρμογής αποτελείται από τέσσερις ομάδες στοιχείων ελέγχου. Η πρώτη (1) ομάδα στοιχείων ελέγχου είναι μενού επιλογών, η δεύτερη (2) ομάδα στοιχείων ελέγχου είναι τριάδες στοιχείων ελέγχου που αποτελούνται από δυο static text και ένα popup menu, η τρίτη (3) ομάδα στοιχείων ελέγχου είναι μια τριάδα push button και το τέταρτο (4) στοιχείο ελέγχου είναι ένα static text .

Τα μενού (η πρώτη (1) ομάδα στοιχείων ελέγχου) είναι ο κύριος κορμός της εφαρμογής μέσα από τα οποία γίνεται η εισαγωγή δεδομένων και η επιλογή των πράξεων, τα οποία είναι οι βασικότερες λειτουργίες της εφαρμογής και είναι πάντα ορατά .

Τα δυο static text και το ένα popup menu (η δεύτερη (2) ομάδα στοιχείων ελέγχου) είναι αυτά τα οποία διαχειρίζονται τα δεδομένα, δηλαδή τους αριθμητικούς ή πολυωνυμικούς πίνακες και είναι πάντα ορατά. Όταν χρήστης εισάγει δεδομένα τότε τα popup menu έχουν επιλογές και ανάλογα την πράξη που επιλέγει ο χρήστης στην οποία μπορεί να χρειάζεται ένας, δυο ή και τρεις πίνακες, γίνονται ενεργά τα αντίστοιχα popup menu. Όταν ο χρήστης επιλέξει κάποιο πίνακα αυτόματα βλέπουμε στο κάτω static text του αντίστοιχου popup menu την διάσταση και την κλάση (αριθμητικός ή πολυωνυμικός (double or pol)) του πίνακα .

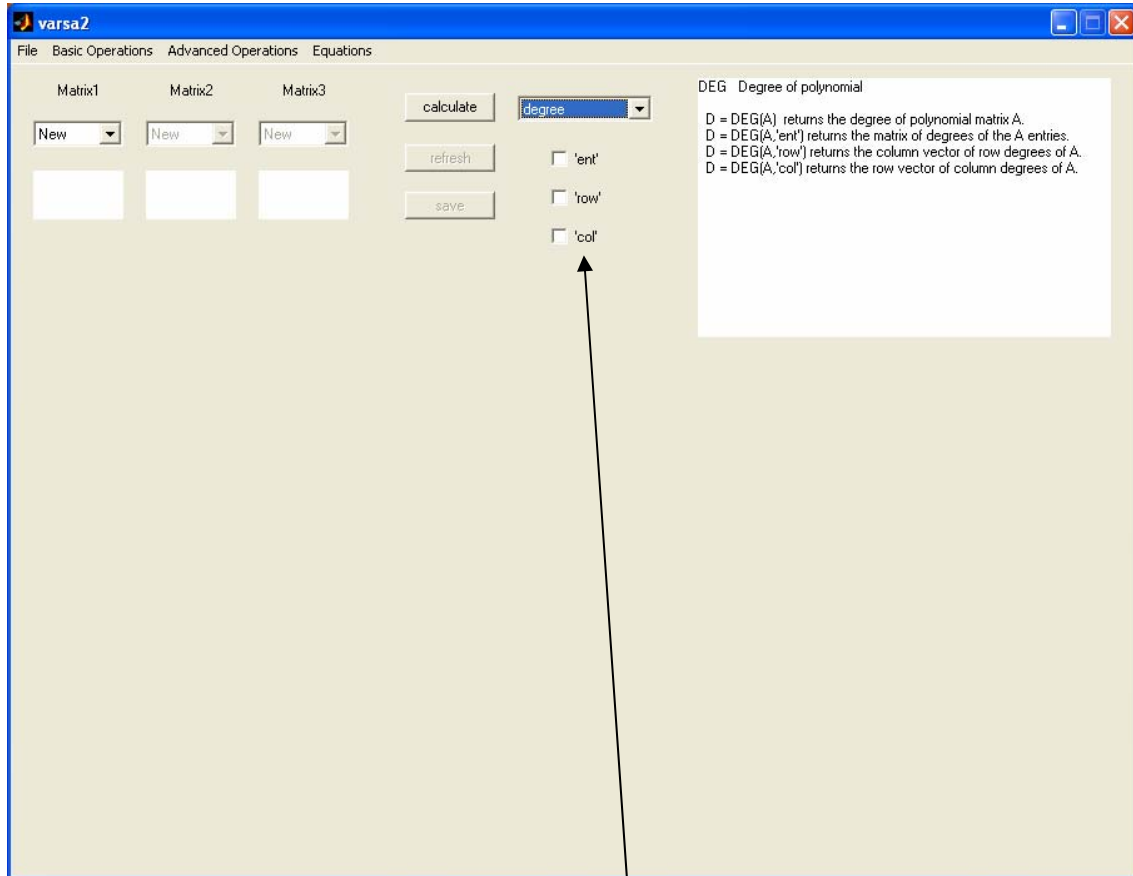
Τα τρία push button (η τρίτη (3) ομάδα στοιχείων ελέγχου) είναι αυτά που εκτελούν, εφόσον πατηθούν, τρεις βασικές λειτουργίες : calculate, refresh, save. Στο αρχικό παράθυρο, αυτά τα τρία στοιχεία ελέγχου είναι ανενεργά αλλά είναι πάντα ορατά. Όταν ο χρήστης επιλέξει κάποια πράξη τότε γίνονται ενεργά, στην αρχή το calculate και αφού πατηθεί το calculate, το refresh, ενώ το save γίνεται ενεργό μονό όταν το αποτέλεσμα είναι πίνακας. Ιδιαίτερη αναφορά πρέπει να κάνουμε στο στοιχείο ελέγχου calculate το οποίο δεν είναι ένα στοιχείο ελέγχου αλλά είναι πολλά τα οποία είναι στην ίδια θέση στο παράθυρο και αυτό γίνεται με προγραμματισμό, γιατί η κάθε πράξη έχει διαφορετικό τρόπο εκτέλεσης και δεν μπορεί ένα στοιχείο ελέγχου push button να εκτελέσει όλες τις πράξεις.

Το στοιχείο ελέγχου static text (το τέταρτο (4) στοιχείο ελέγχου) είναι αυτό στο οποίο εμφανίζεται η βοήθεια (help) κάθε πράξης που επιλέγει ο χρήστης.



Εικόνα 3.2 Εμφάνιση νέου popup menu στο παράθυρο

Σε κάποιες επιλογές των μενού και των υπό-μενού είναι απαραίτητη η χρήση ενός επιπλέον popup menu όπως φαίνεται στην εικόνα 3.2. Το popup menu αυτό δημιουργείται με προγραμματισμό και όπως και το push button calculate δεν είναι ένα στοιχείο ελέγχου, αλλά πολλά στοιχεία ελέγχου τα οποία έχουν διαφορετική λειτουργία ανάλογα την πράξη στην οποία εμφανίζονται.



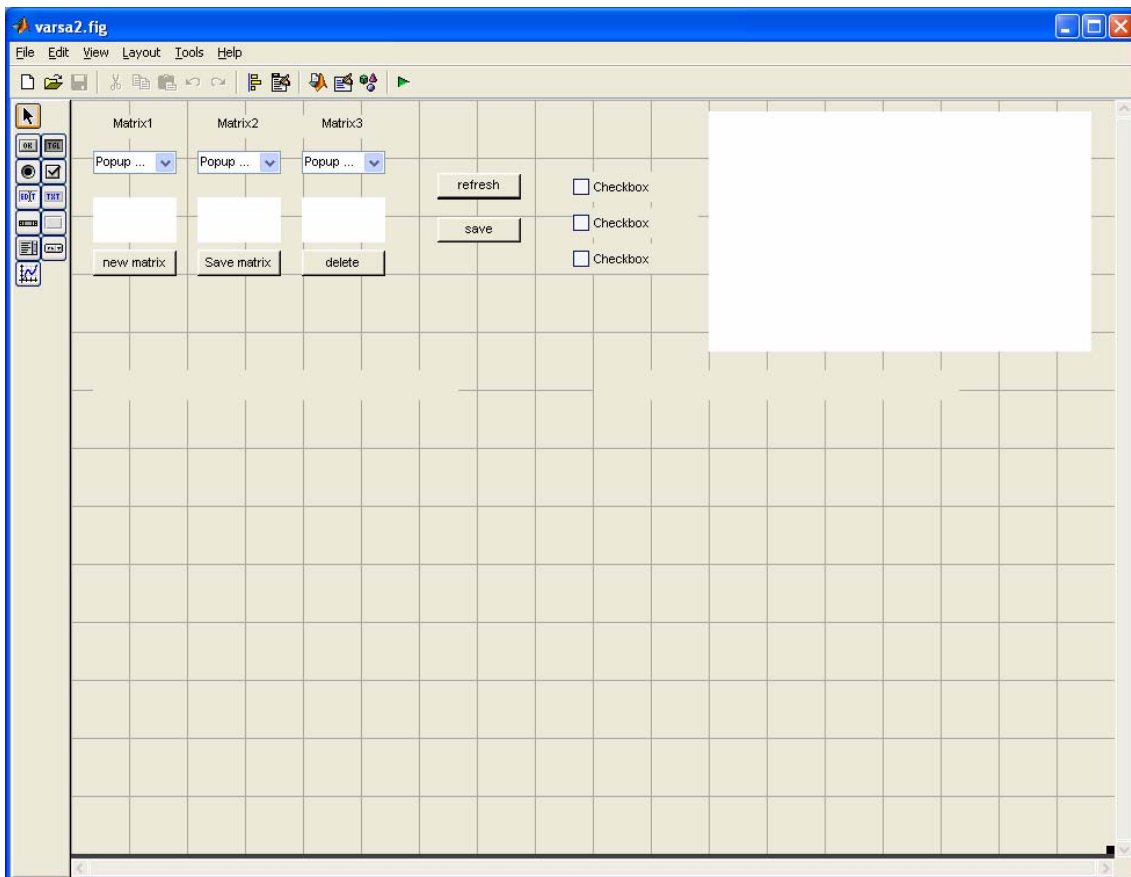
Εικόνα 3.3 Εμφάνιση check box στο παράθυρο

Τέλος, σε κάποιες πράξεις είναι απαραίτητη η χρήση των check boxes όπως φαίνεται στην εικόνα 3.3. Με την βοήθεια των check boxes ο χρήστης έχει τη δυνατότητα να επιλέξει μια επιμέρους λειτουργία για την πράξη που επέλεξε ή να καθορίσει μια συγκεκριμένη παράμετρο.

3.2 ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΦΑΡΜΟΓΗΣ

3.2.1 Σχεδιασμός και προγραμματισμός στοιχείων ελέγχου

Στην ενότητα αυτή θα αναφερθούμε στον ιδιαίτερο σχεδιασμό της εφαρμογής ο οποίος χωρίζεται σε δυο στάδια : 1) στα σταθερά στοιχεία ελέγχου που έχουν σχεδιαστεί στο layout editor του GUIDE και 2) στα προσωρινά στοιχεία ελέγχου που η δημιουργία και ο έλεγχος γίνεται με προγραμματισμό. Όπως βλέπουμε λοιπόν έχουμε δυο κατηγορίες στοιχείων ελέγχου τα σταθερά και τα προσωρινά.



Εικόνα 3.4 Σταθερά στοιχεία ελέγχου

Η επιλογή των στοιχείων ελέγχου έγινε έτσι ώστε σταθερά να είναι αυτά τα στοιχεία ελέγχου τα οποία επιτελούν μια συγκεκριμένη λειτουργία σε πολλές πράξεις, για παράδειγμα τα push button refresh και save κάνουν την ίδια

λειτουργία σε κάθε πράξη που είναι ενεργά, όπως και τα popup menu με τα οποία επιλέγουμε πίνακες. Προσωρινά είναι τα στοιχεία ελέγχου που επιτελούν μια συγκεκριμένη λειτουργία σε μια πράξη αλλά για κάθε διαφορετική πράξη έχουν διαφορετικό προγραμματισμό, ένα τέτοιο στοιχείο ελέγχου είναι το push button calculate το οποίο έχει διαφορετικό προγραμματισμό σε κάθε πράξη.

Τα στοιχεία ελέγχου που φαίνονται στην εικόνα 3.4, είναι τα σταθερά στοιχεία ελέγχου που σχεδιάστηκαν με την βοήθεια του GUIDE και του Layout Editor. Όλα τα άλλα στοιχεία ελέγχου που χρησιμοποιούνται μέσα στην εφαρμογή δημιουργούνται και ορίζονται οι ιδιότητές τους με προγραμματισμό.

Σταθερά στοιχεία ελέγχου, όπως φαίνεται στην εικόνα 3.4, είναι: τα push button new matrix, save matrix, delete, refresh και save, τα popup menu και τα static text για τους πίνακες, το static text για την βοήθεια, τα check box καθώς και τα μενού file, basic operations, advanced operations, equations με τα αντίστοιχα υπό-μενού.

Στοιχείο ελέγχου	Ετικέτα (tag)	Χρήση (ή ιδιότητα string)
Popup menu	popupmenu1	Επιλογή πίνακα 1
	popupmenu2	Επιλογή πίνακα 2
	popupmenu3	Επιλογή πίνακα 3
Push Button	pushbutton1	Λειτουργία new matrix
	pushbutton2	Λειτουργία save matrix
	pushbutton3	Λειτουργία delete
	refresh1	Λειτουργία refresh
	refresh2	Λειτουργία refresh
	savepb	Λειτουργία save
Static Text	text1	Matrix 1
	text2	Matrix 2
	text3	Matrix 3
	text4	Πληροφορίες του πίνακα 1
	text5	Πληροφορίες του πίνακα 2

	text6	Πληροφορίες του πίνακα 3
	text7	Βοήθεια
	text8	Πληροφορίες αποτελέσματος
	text9	Πληροφορίες αποτελέσματος
Check box	checkbox1	Επιλογή παραμέτρων
	checkbox2	Επιλογή παραμέτρων
	checkbox3	Επιλογή παραμέτρων

Πίνακας σταθερών στοιχείων ελέγχου

Προσωρινά στοιχεία ελέγχου είναι το popup menu p1, το οποίο εμφανίζεται σε κάποιες πράξεις (εικόνα 3.2), το push button p2, το οποίο είναι το στοιχείο ελέγχου calculate, τα static text a1, a2 τα οποία είναι τα στοιχεία ελέγχου όπου γίνεται η εμφάνιση των αποτελεσμάτων καθώς και τα edit text a1 στα οποία γίνεται η εισαγωγή τιμών στους πίνακες.

Στοιχείο ελέγχου	Ετικέτα (tag)	Χρήση (ή ιδιότητα string)
Popup menu	p1	Επιλογή επιμέρους πράξεων
Push Button	p2	Λειτουργία calculate
Static Text	a1	Εμφάνιση αποτελεσμάτων
	a2	Εμφάνιση αποτελεσμάτων
Edit text	a1	Εισαγωγή τιμών

Πίνακας προσωρινών στοιχείων ελέγχου

3.2.2 Αρχικές συναρτήσεις εφαρμογής

Οι πρώτες συναρτήσεις που υπάρχουν στο m-file είναι:

1. *function varargout = polyxgui(varargin)*
2. *function polyxgui_OpeningFcn(hObject, eventdata, handles, varargin)*
3. *function varargout = polyxgui_OutputFcn(hObject, eventdata, handles)*

Η πρώτη χρειάζεται ώστε να μπορεί ο χρήστης να καλεί το παράθυρο από τη γραμμή εντολών πληκτρολογώντας το όνομα του (» polyxgui). Επιπλέον, ο κώδικας που περιλαμβάνεται στο σώμα εντολών της δεύτερης συνάρτησης, εκτελείται ακριβώς πριν εκκινήσει το παράθυρο. Τέλος, οι εντολές που υπάρχουν στην τρίτη συνάρτηση, εξάγουν τα αποτελέσματα τους στη γραμμή εντολών του MATLAB.

Αναλυτικότερα η δεύτερη συνάρτηση περιέχει εντολές αρχικοποίησης των προσωρινών στοιχείων ελέγχου. Σ' αυτή την συνάρτηση δημιουργούνται όλα τα προσωρινά στοιχεία ελέγχου και ορίζονται κάποιες απαραίτητες ιδιότητες για το καθένα από αυτά. Για τα σταθερά στοιχεία ελέγχου η αρχικοποίηση γίνεται στο GUIDE ορίζοντας τις απαραίτητες ιδιότητες μέσω του property inspector.

3.2.3 Αντικείμενα, δομές και παράθυρα διαλόγου της εφαρμογής

Εκτός των σταθερών και των προσωρινών στοιχείων ελέγχου που υπάρχουν στην εφαρμογή, χρησιμοποιούνται και κάποια αντικείμενα (handles) απαραίτητα για την λειτουργία της εφαρμογής. Σ' αυτά τα αντικείμενα καταχωρούνται κάποιες τιμές οι οποίες μπορούν να χρησιμοποιηθούν από οποιαδήποτε συνάρτηση του m-file.

Αντικείμενο	Χρήση
handles.mat1	Καταχώρηση του πίνακα 1
handles.mat2	Καταχώρηση του πίνακα 2

handles.mat3	Καταχώρηση του πίνακα 3
handles.result	Καταχώρηση αποτελέσματος (όταν η πράξη επιστρέφει ένα αποτέλεσμα)
handles.result1	Καταχώρηση πρώτου αποτελέσματος (όταν η πράξη επιστρέφει δυο αποτελέσματα)
handles.result2	Καταχώρηση δεύτερου αποτελέσματος (όταν η πράξη επιστρέφει δυο αποτελέσματα)

Πίνακας αντικειμένων

Στην εφαρμογή χρησιμοποιείται και η δομή `User_struct` με την βοήθεια της ιδιότητας `User Data` του παραθύρου. Η ιδιότητα `User Data` δίνει τη δυνατότητα να αποθηκεύονται σε μια δομή δεδομένα αλλά και να μεταφέρονται αυτά, τα δεδομένα, μεταξύ των συναρτήσεων. Η δομή (μεταβλητή) που χρησιμοποιείται είναι η `user_struct` με πεδία `user_struct.name` και `user_struct.matr` στα οποία καταχωρούνται το όνομα και οι τιμές αντίστοιχα όλων των πινάκων.

Η εφαρμογή έχει προγραμματιστεί έτσι ώστε ο χρήστης να οδηγείται εύκολα στην πράξη - λειτουργία που επιθυμεί. Αυτό επιτυγχάνεται με την βοήθεια των ιδιοτήτων `enable` και `visible` οι οποίες κάνουν, αντίστοιχα, ενεργά ή μη και ορατά ή μη τα στοιχεία ελέγχου που είναι απαραίτητα για κάθε πράξη που έχει επιλεγεί. Επίσης χρησιμοποιούνται και παράθυρα διαλόγου προσοχής και λάθους τα οποία ενημερώνουν τον χρήστη τις τυχόν λάθος επιλογές του. Τα παράθυρα διαλόγου προσοχής (`warnDlg`) χρησιμοποιούνται για την ενημέρωση του χρήστη στην περίπτωση που δεν έχει ορίσει σωστά τις παραμέτρους της πράξης που επέλεξε, για παράδειγμα ζητά τον υπολογισμό της πρόσθεσης δυο πινάκων και δεν έχει επιλέξει το δεύτερο πίνακα, ενώ τα παράθυρα διαλόγου λάθους (`errorDlg`) χρησιμοποιούνται για την ενημέρωση του χρήστη στην περίπτωση που δεν έχει επιλέξει σωστά τα δεδομένα της πράξης που επέλεξε, για παράδειγμα ζητά τον υπολογισμό της πρόσθεσης δυο

πινάκων και δεν έχει επιλέξει πίνακες με ίδια διάσταση. Για κάθε διαφορετικό παράθυρο διαλόγου εμφανίζεται και το κατάλληλο μήνυμα στον χρήστη.

3.2.4 Εμφάνιση αποτελεσμάτων

Για την εμφάνιση των αποτελεσμάτων χρησιμοποιείται από την εφαρμογή, το m-file (συναρτήση) `funtext`. Η συνάρτηση αυτή περιέχεται σε όλες τις συναρτήσεις του m-file που εξάγουν αποτελέσματα. Η συνάρτηση `funtext` περιέχει τις εντολές:

```
function a=funtext(mat,str)
if str==1
    n1=0.02;
    n2=0.60;
elseif str==2
    n1=0.20;
    n2=0.60;
else
    n1=0.50;
    n2=0.60;
end
[rows,columns]=size(mat);
if rows==1 & columns==1
    a=uicontrol('style','text','backgroundcolor','w','units','normalized',...
        'FontSize',10,'position',[n1 0.5 0.20 0.03]);
    handles.a=a;
    if isnumeric(mat)
        set(handles.a,'string',num2str(mat));
    else
        set(handles.a,'string',char(mat));
    end
else
```



```
if isnumeric(mat)
    m=char(pol(mat));
else
    m=char(mat);
end
len=zeros(rows,columns);
for i=1:rows
    for j=1:columns
        len(i,j)=length(m{i,j});
    end
end
lenmax=max(max(len))+1;
n3=lenmax*0.0055555555555555;
if n3 <= 0.12
    n4=0.03;
elseif n3 <= 0.24
    n3=0.12;
    n4=0.05;
else
    n3=0.12;
    n4=0.07
end
for i=1:rows
    for j=1:columns
        a(i,j)=uicontrol('style','text','backgroundcolor','w',
            'units','normalized','FontSize',10,
            'position',[n1+(n3+0.01)*(j-1) n2-(n4+0.02)*i n3 n4]);
    end
end
handles.a=a;
if isnumeric(mat)
```

```
for i=1:rows
    for j=1:columns
        set(handles.a(i,j),'string',num2str(mat(i,j)));
    end
end
else
    mat=char(mat);
    for i=1:rows
        for j=1:columns
            set(handles.a(i,j),'string',mat(i,j))
        end
    end
end
end
```

Η συνάρτηση έχει παραμέτρους το `mat` το οποίο είναι ο πίνακας που θέλουμε να εμφανίσουμε και το `str` το οποίο καθορίζει την θέση που θα εμφανιστεί ο πίνακας στο παράθυρο. Στη συνάρτηση δημιουργείται το `handles.a` το οποίο είναι `static text` και σ' αυτό καταχωρούνται οι τιμές του πίνακα. Επίσης, γίνεται έλεγχος εάν ο πίνακας (`mat`) είναι αριθμητικός ή πολυωνυμικός καθώς επίσης και έλεγχος εάν ο πίνακας είναι `1x1`, δηλαδή, ένα στοιχείο στο οποίο έχουμε διαφορετική εμφάνιση. Τέλος, το πλάτος των στοιχείων ελέγχου καθορίζεται από το μήκος σε χαρακτήρες του μεγαλύτερου στοιχείου του πίνακα.

3.2.5 Αρχικοποίηση στοιχείων ελέγχου

Σε κάθε μενού γίνεται αρχικοποίηση των σταθερών και των προσωρινών στοιχείων ελέγχου με την βοήθεια των `m-files (script)` `delete` και `propinit`, τα οποία αρχικοποιούν το παράθυρο, δηλαδή, κάνουν ορατά (`visible`) και μη ενεργά (`enable`) τα στοιχεία ελέγχου που έχει το αρχικό παράθυρο. Επομένως όταν ο χρήστης επιλέγει κάποιο υπό-μενού από ένα μενού πρώτα γίνεται

αρχικοποίηση των στοιχείων ελέγχου. Αυτό μας βοηθάει στο να μην επαναλάβουμε την διαδικασία της αρχικοποίησης σε κάθε υπό-μενού.

Το m-file **dele** περιέχει όλες τις απαραίτητες εντολές για την αρχικοποίηση όλων των στοιχείων ελέγχου (σταθερών και προσωρινών).

```
set(handles.p1,'visible','off');
set(handles.p2,'visible','on','enable','off');
set(handles.popupmenu1,'enable','off');
set(handles.popupmenu2,'enable','off');
set(handles.popupmenu3,'enable','off');
set(handles.pushbutton1,'visible','off');
set(handles.pushbutton2,'visible','off');
set(handles.pushbutton3,'visible','off');
set(handles.savepb,'enable','off');
set(handles.refresh1,'visible','on','enable','off');
set(handles.refresh2,'visible','off');
set(handles.a1,'visible','off');
set(handles.a2,'visible','off');
set(handles.a3,'visible','off');
set(handles.a4,'visible','off');
set(handles.text1,'string','Matrix1');
set(handles.text2,'string','Matrix2');
set(handles.text3,'string','Matrix3');
set(handles.text4,'string','');
set(handles.text5,'string','');
set(handles.text6,'string','');
set(handles.text7,'string','');
set(handles.text8,'string','');
set(handles.text9,'string','');
set(handles.checkbox1,'visible','off','value',0);
set(handles.checkbox2,'visible','off','value',0);
set(handles.checkbox3,'visible','off','value',0);
```

Το m-file **popinit** περιέχει όλες τις απαραίτητες εντολές για την αρχικοποίηση των popup menu που χρησιμοποιούνται για την επιλογή των πινάκων.

```
set(handles.popupmenu1,'value',1);
set(handles.popupmenu2,'value',1);
set(handles.popupmenu3,'value',1);
user_struct=get(handles.figure1,'UserData');
k1=length(user_struct);
uu1=[];
for i=1:k1
    temp=cellstr(user_struct(i).name);
    uu1=[uu1;temp];
end
set(handles.popupmenu1,'string',uu1);
set(handles.popupmenu2,'string',uu1);
set(handles.popupmenu3,'string',uu1);
```

3.2.6 Προγραμματισμός σταθερών στοιχείων ελέγχου

Τα στοιχεία ελέγχου **popupmenu1-2-3** χρησιμοποιούνται για την επιλογή πινάκων. Οι συναρτήσεις που συνοδεύουν αυτά τα στοιχεία ελέγχου είναι:

1. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
2. *function popupmenu1_Callback(hObject, eventdata, handles)*
3. *function popupmenu2_CreateFcn(hObject, eventdata, handles)*
4. *function popupmenu2_Callback(hObject, eventdata, handles)*
5. *function popupmenu3_CreateFcn(hObject, eventdata, handles)*
6. *function popupmenu3_Callback(hObject, eventdata, handles)*

Οι συναρτήσεις (1), (3) και (5) είναι συναρτήσεις δημιουργίας των popup menu, ενώ οι συναρτήσεις (2), (4) και (6) είναι συναρτήσεις επιστροφής στις οποίες έχουν προστεθεί οι παρακάτω εντολές

```
set(handles.a1,'visible','off');
set(handles.a2,'visible','off');
user_struct=get(handles.figure1,'UserData');
pop1=get(handles.popupmenu1,'value');
mat1=user_struct(pop1).matr;
if isempty(mat1)
    return;
end
[rows,columns]=size(mat1);
r=num2str(rows);
c=num2str(columns);
cc=char(class(mat1));
e1={['class: ',cc,['size: ',r,'x',c]};
set(handles.text4,'string',e1);
a1=funttext(mat1,1);
handles.a1=a1;
handles.mat1=mat1;
```

στις οποίες, στη μεταβλητή pop1 καταχωρείται η επιλογή πίνακα και στην μεταβλητή mat1 καταχωρείται ο επιλεγμένος πίνακας ο οποίος ανακτάται από την δομή user_struct. Τέλος γίνεται έλεγχος εάν ο πίνακας είναι άδειος και καταχωρούνται στο text4 οι πληροφορίες του πίνακα που επιλέχθηκε καθώς επίσης δημιουργούνται τα αντικείμενα handles.a1 και handles.mat1.

Οι συναρτήσεις (4) και (6) περιέχουν τις ίδιες εντολές με την (2) με μόνη διαφορά στην αρίθμηση των μεταβλητών (mat2, pop2,mat3,pop3).

Το στοιχείο ελέγχου **pushbutton1** χρησιμοποιείται για την δημιουργία νέου πίνακα. Η συνάρτηση που συνοδεύει αυτό το στοιχείο ελέγχου είναι:

```
function pushbutton1_Callback(hObject, eventdata, handles)
```

η οποία είναι συνάρτηση επιστροφής στην οποία έχουν προστεθεί οι παρακάτω εντολές

```
set(handles.a1,'visible','off');
```

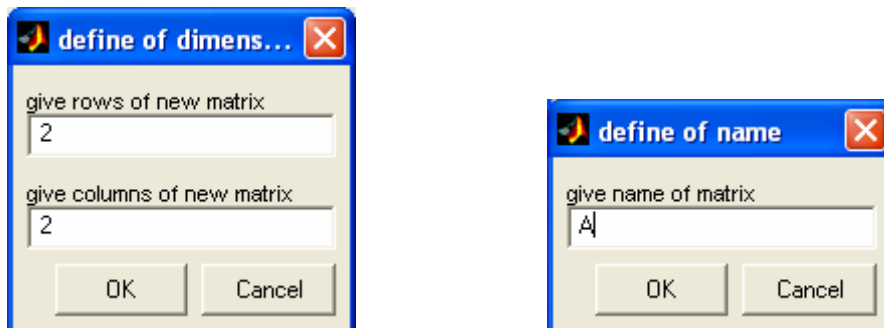
```

set(handles.text4,'string','');
popinit;
user_struct=get(handles.figure1,'UserData');
k=length(user_struct);
answer=inputdlg({'give rows of new matrix ','...
                ['give columns of new matrix '],'...
                'define of dimension',[1,30]});
if isempty(answer)
    return;
end
rows=str2num(answer{1});
columns=str2num(answer{2});
if isempty(rows) | isempty(columns)
    warndlg;
    return;
end
done=0;
while done==0
    answer1=inputdlg(['give name of matrix ','define of name',[1,30]});
    an=answer1{1};
    for i=1:k
        if strcmp(an,user_struct(i).name)
            done=0;
            h=warndlg('this name is already exist');
            waitfor(h);
            break;
        end
    done=1;
    end
end
handles.answer1=answer1{1};

```

```
for i=1:rows
    for j=1:columns
        a1(i,j)=uicontrol('style','edit','backgroundcolor','w',...
            'units','normalized','position',[0.09*j 0.60-0.1*i 0.08 0.04]);
    end
end
handles.a1=a1;
set(handles.popupmenu1,'enable','off');
set(handles.pushbutton1,'enable','off');
set(handles.pushbutton2,'enable','on');
```

στις οποίες, με τα παράθυρα διαλόγου εισαγωγής (inputdlg), όπως φαίνεται στην εικόνα 3.5, καταχωρούνται οι διαστάσεις του πίνακα στην μεταβλητή answer καθώς και το όνομα του πίνακα καταχωρείται στην μεταβλητή answer1. Στην περίπτωση που υπάρχει πίνακας με ίδιο όνομα ζητείται από τον χρήστη να εισάγει νέο όνομα.



Εικόνα 3.5 Παράθυρα διαλόγου εισαγωγής

Στην συνέχεια δημιουργούνται τα προσωρινά edit text στα οποία ο χρήστης θα εισαγάγει τις τιμές που επιθυμεί και τα δεδομένα θα καταχωρηθούν στο αντικείμενο handles.a1. Τέλος, ακολουθούν εντολές που ρυθμίζουν την ιδιότητα enable στα απαραίτητα στοιχεία ελέγχου.

Το στοιχείο ελέγχου **pushbutton2** χρησιμοποιείται για την αποθήκευση του πίνακα που έχει εισαγάγει ο χρήστης. Η συνάρτηση που συνοδεύει αυτό το στοιχείο ελέγχου είναι:

```
function pushbutton2_Callback(hObject, eventdata, handles)
```

η οποία είναι συνάρτηση επιστροφής στην οποία έχουν προστεθεί οι παρακάτω εντολές

```
user_struct=get(handles.figure1,'UserData');
k=length(user_struct);
ans1=handles.answer1;
user_struct(k+1).name=ans1;
[rows,columns]=size(handles.a1);
for i=1:rows
    for j=1:columns
        aa(i,j)=sym(get(handles.a1(i,j),'string'));
    end
end
aa=pol(aa);
user_struct(k+1).matr=aa;
set(handles.a1,'visible','off');
set(handles.figure1,'UserData',user_struct);
set(handles.basic_operations,'enable','on');
set(handles.advanoper,'enable','on');
set(handles.equations,'enable','on');
set(handles.savework,'enable','on');
set(handles.deletematrix,'enable','on');
set(handles.popupmenu1,'enable','on');
set(handles.pushbutton1,'enable','on');
set(handles.pushbutton2,'enable','off');
popinit;
```

στις οποίες, στην δομή `user_struct.name` καταχωρείται το όνομα του πίνακα και στην δομή `user_struct.matr` καταχωρούνται οι τιμές του πίνακα οι οποίες

ανακτώνται από το `handles.a1`. Έπειτα ακολουθούν εντολές που ρυθμίζουν την ιδιότητα `enable` στα απαραίτητα στοιχεία ελέγχου.

Το στοιχείο ελέγχου **pushbutton3** χρησιμοποιείται για την διαγραφή ενός πίνακα. Η συνάρτηση που συνοδεύει αυτό το στοιχείο ελέγχου είναι:

```
function pushbutton3_Callback(hObject, eventdata, handles)
```

η οποία είναι συνάρτηση επιστροφής στην οποία έχουν προστεθεί οι παρακάτω εντολές

```
temp_struct=get(handles.figure1,'UserData');  
n=length(temp_struct);  
pop1=get(handles.popupmenu1,'value');  
if pop1==1  
    warndlg('must choice matrix');  
    return;  
end  
user_struct=struct('name','new','matr',[]);  
set(handles.figure1,'UserData',user_struct);  
if pop1==n  
    user_struct(1:n-1)=temp_struct(1:n-1);  
else  
    user_struct(1:pop1-1)=temp_struct(1:pop1-1);  
    user_struct(pop1:n-1)=temp_struct(pop1+1:n);  
end  
set(handles.popupmenu1,'enable','off');  
set(handles.pushbutton3,'enable','off');  
set(handles.figure1,'UserData',user_struct);  
popinit;  
set(handles.a1,'visible','off');  
set(handles.text4,'string','');
```

στις οποίες, αφαιρείται από την δομή `user_struct.name` και `user_struct.matr` ο πίνακας που επέλεξε ο χρήστης. Έπειτα ακολουθούν εντολές που ρυθμίζουν τις ιδιότητες `enable` και `string` στα απαραίτητα στοιχεία ελέγχου.

Το στοιχείο ελέγχου **refresh1** χρησιμοποιείται για την ανανέωση των δεδομένων μιας πράξης-λειτουργίας, στην οποία δεν ενεργό το προσωρινό popup menu `p1`. Η συνάρτηση που συνοδεύει αυτό το στοιχείο ελέγχου είναι:

```
function refresh1_Callback(hObject, eventdata, handles)
```

η οποία είναι συνάρτηση επιστροφής στην οποία έχουν προστεθεί οι παρακάτω εντολές

```
set(handles.p1,'value',1);  
set(handles.p2,'enable','on');  
set(handles.refresh1,'enable','off');  
set(handles.savepb,'enable','off');  
set(handles.a1,'visible','off');  
set(handles.a2,'visible','off');  
set(handles.a3,'visible','off');  
set(handles.text4,'string','');  
set(handles.text5,'string','');  
set(handles.text6,'string','');  
set(handles.text8,'string','');  
set(handles.text9,'string','');  
set(handles.checkbox1,'value',0);  
set(handles.checkbox2,'value',0);  
set(handles.checkbox3,'value',0);  
popinit;
```

στις οποίες, ρυθμίζονται οι ιδιότητες `visible`, `enable`, `string` και `value` στα απαραίτητα στοιχεία ελέγχου έτσι ώστε να ανανεώνονται τα δεδομένα της επιλεγμένης πράξης-λειτουργίας.

Το στοιχείο ελέγχου **refresh2** χρησιμοποιείται για την ανανέωση των δεδομένων μιας πράξης-λειτουργίας, στην οποία δεν ενεργό το προσωρινό popup menu p1. Η συνάρτηση που συνοδεύει αυτό το στοιχείο ελέγχου είναι:

```
function refresh2_Callback(hObject, eventdata, handles)
```

η οποία είναι συνάρτηση επιστροφής στην οποία έχουν προστεθεί οι παρακάτω εντολές

```
set(handles.p1,'value',1);  
set(handles.p1,'enable','on');  
set(handles.refresh2,'enable','off');  
set(handles.savepb,'enable','off');  
set(handles.a1,'visible','off');  
set(handles.a2,'visible','off');  
set(handles.a3,'visible','off');  
set(handles.text4,'string','');  
set(handles.text5,'string','');  
set(handles.text6,'string','');  
set(handles.text7,'string','');  
set(handles.text8,'string','');  
set(handles.text9,'string','');  
set(handles.checkbox1,'visible','off','value',0);  
set(handles.checkbox2,'visible','off','value',0);  
set(handles.checkbox3,'visible','off','value',0);  
popinit;
```

στις οποίες, ρυθμίζονται οι ιδιότητες *visible*, *enable*, *string* και *value* στα απαραίτητα στοιχεία ελέγχου έτσι ώστε να ανανεώνονται τα δεδομένα της επιλεγμένης πράξης-λειτουργίας.

Το στοιχείο ελέγχου **savepb** χρησιμοποιείται για την αποθήκευση του αποτελέσματος μιας πράξης-λειτουργίας. Η συνάρτηση που συνοδεύει αυτό το στοιχείο ελέγχου είναι:

```
function savepb_Callback(hObject, eventdata, handles)
```

η οποία είναι συνάρτηση επιστροφής στην οποία έχουν προστεθεί οι παρακάτω εντολές

```

user_struct=get(handles.figure1,'UserData');
result=handles.result;
k=length(user_struct);
done=0;
while done==0
    answer1=inputdlg(['give name to saved result ','define of name',[1,30]);
    an=answer1{1};
    for i=1:k
        if strcmp(an,user_struct(i).name)
            done=0;
            h=warndlg('this name is already exist');
            waitfor(h);
            break;
        end
        done=1;
    end
end
user_struct(k+1).name=answer1{1};
user_struct(k+1).matr=result;
set(handles.figure1,'UserData',user_struct);
set(handles.savepb,'enable','off');
set(handles.p1,'enable','on');
set(handles.p2,'enable','on');
set(handles.refresh1,'enable','off');
set(handles.refresh2,'enable','off');
set(handles.a1,'visible','off');
set(handles.text4,'string','');
set(handles.text5,'string','');
set(handles.text6,'string','');

```

```
set(handles.text8,'string','');
```

```
set(handles.text9,'string','');
```

```
popinit;
```

στις οποίες, με το παράθυρο διαλόγου εισαγωγής (inputdlg) καταχωρείται στην μεταβλητή answer1 το όνομα του πίνακα και στο handles.result είναι καταχωρημένες οι τιμές του πίνακα. Στην περίπτωση που υπάρχει πίνακας με το ίδιο όνομα, ζητείται από τον χρήστη να εισάγει νέο όνομα. Τέλος ρυθμίζονται οι ιδιότητες visible, enable και string στα απαραίτητα στοιχεία ελέγχου.

3.2.7 Προγραμματισμός προσωρινών στοιχείων ελέγχου

Τα προσωρινά στοιχεία ελέγχου συνοδεύονται από διαφορετικές συναρτήσεις επιστροφής το καθένα, ανάλογα με την πράξη που έχει επιλέξει ο χρήστης.

Το στοιχείο ελέγχου p1 το οποίο είναι το popup menu που εμφανίζεται στις πράξεις-λειτουργίες equations και operations for 1 matrix δίνει τη δυνατότητα στο χρήστη να επιλέξει την επιμέρους πράξη που επιθυμεί. Για τις συναρτήσεις που συνοδεύουν το p1 θα αναφερθούμε αναλυτικότερα στον προγραμματισμό των μενού επιλογών.

Το στοιχείο ελέγχου p2 το οποίο είναι το push button calculate εκτελεί όλες τις πράξεις-λειτουργίες και συνοδεύεται από 27 συναρτήσεις επιστροφής. Στις συναρτήσεις επιστροφής ο προγραμματισμός χωρίζεται σε τρία στάδια. Στο πρώτο στάδιο γίνεται έλεγχος εάν ο χρήστης έχει επιλέξει σωστά τα δεδομένα (πίνακες, πράξεις, επιμέρους επιλογές). Οι εντολές που υλοποιούν το πρώτο στάδιο είναι κοινές σε όλες τις συναρτήσεις επιστροφής που συνοδεύουν το p2 και η μόνη διαφοροποίησή τους είναι στους πίνακες που είναι απαραίτητοι για την εκτέλεση της πράξης-λειτουργίας που έχει επιλεγεί. Οι εντολές είναι:

```
set(handles.a1,'visible','off');
```

```
pop1=get(handles.popupmenu1,'value');
```

```
if pop1==1
```

```
    warndlg('must choose matrix 1');  
    return;  
end  
pop2=get(handles.popupmenu2,'value');  
if pop2==1  
    warndlg('must choose matrix 2');  
    return;  
end
```

με τις οποίες στις μεταβλητές pop1, pop2 και pop3 (ανάλογα τους πίνακες που είναι απαραίτητοι) έχουν καταχωρηθεί οι επιλογές του χρήστη στα popup menu 1, 2 και 3 αντίστοιχα και γίνεται έλεγχος για την ορθή επιλογή αυτών.

Στο δεύτερο στάδιο γίνεται έλεγχος, εάν χρειάζεται, για την ορθότητα των δεδομένων και εκτελείται η πράξη. Για το δεύτερο στάδιο τις κάθε μιας συνάρτησης επιστροφής των προσωρινών στοιχείων ελέγχου θα αναφερθούμε αναλυτικότερα στον προγραμματισμό των μενού επιλογών.

Τέλος, στο τρίτο στάδιο εμφανίζονται τα αποτελέσματα στο παράθυρο και ρυθμίζονται κάποιες ιδιότητες των στοιχείων ελέγχου που είναι απαραίτητα για την επιλεγμένη πράξη. Οι εντολές που υλοποιούν το τρίτο στάδιο είναι κοινές σε όλες τις συναρτήσεις επιστροφής που συνοδεύουν το p2 και η διαφοροποίησή τους είναι στα αποτελέσματα, δηλαδή εάν έχουμε ένα ή δυο αποτελέσματα, καθώς και στα στοιχεία ελέγχου που είναι απαραίτητα για την εκτέλεση της πράξης-λειτουργίας που έχει επιλεγθεί. Οι εντολές είναι:

```
set(handles.text8,'string',' TEXT');  
a1=funtext(result1,1);  
handles.a1=a1;  
set(handles.text9,'string',' TEXT');  
a2=funtext(result2,3);  
handles.a2=a2;  
set(handles.popupmenu1,'enable','off');  
set(handles.popupmenu2,'enable','off');  
set(handles.popupmenu3,'enable','off');
```

```
set(handles.p1,'enable','off');  
set(handles.p2,'enable','off');  
set(handles.refresh2,'enable','on');
```

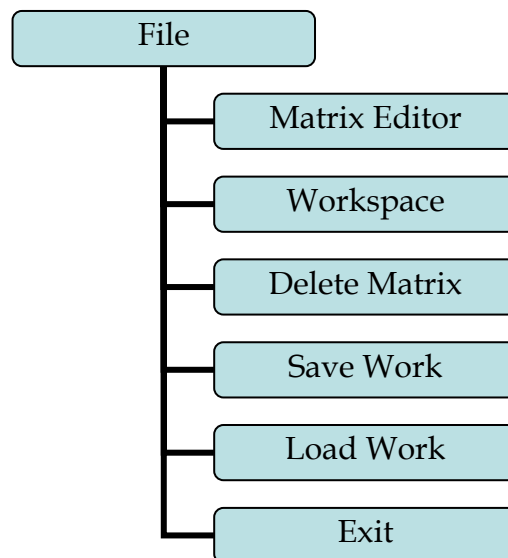
οι οποίες, με την χρήση της funtext εμφανίζουν το ή τα αποτελέσματα, ανάλογα την πράξη, στο παράθυρο καθώς και στα text8 και text9 από ένα ενημερωτικό μήνυμα. Επίσης ορίζεται η ιδιότητα enable στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

3.2.8 Προγραμματισμός μενού επιλογών

Στην εφαρμογή υπάρχουν τέσσερα μενού (file, basic operations, advanced operations, equations) και τα αντίστοιχα υπό-μενού τους. Σε κάθε μενού γίνεται αρχικοποίηση των σταθερών και των προσωρινών στοιχείων ελέγχου με την βοήθεια των m-files (script) dele και popinit.

3.2.8.1 Μενού file

Το μενού file έχει την παρακάτω δομή:



Μενού File

Το μενού **File** δίνει τη δυνατότητα στον χρήστη να επιλέξει ένα από τα υπό-μενού, όπως φαίνεται στο διάγραμμα. Η συνάρτηση που χρησιμοποιείται στο m-file για το συγκεκριμένο μενού είναι η εξής:

```
function file_Callback(hObject, eventdata, handles)
```

Στην συνάρτηση αυτή, η οποία είναι συνάρτηση επιστροφής του μενού, έχουν προστεθεί οι εντολές

```
delete;
```

```
popinit;
```

Για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Matrix Editor** ο χρήστης έχει τη δυνατότητα να εισάγει νέους πίνακες χρησιμοποιώντας τα σταθερά στοιχεία ελέγχου push button new matrix και save matrix τα οποία γίνονται ενεργά και ορατά μονό στο συγκεκριμένο υπό-μενού. Τέλος, ενεργοποιημένο είναι και το σταθερό στοιχείο ελέγχου popupmenu1 από το οποίο μπορεί ο χρήστης να επιλέξει τον πίνακα που θέλει να εμφανίσει στο παράθυρο.

Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (handles) που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function matrix_editor_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function pushbutton1_Callback(hObject, eventdata, handles)*
5. *function pushbutton2_Callback(hObject, eventdata, handles)*
6. *function refresh1_Callback(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.popupmenu1,'enable','on');
```

```
set(handles.pushbutton1,'visible','on');
```

```
set(handles.pushbutton2,'visible','on','enable','off');
```


οι οποίες ορίζουν τις ιδιότητες `visible` και `enable` στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού.

Οι συναρτήσεις (2),(3),(4),(5) και (6) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Workspace** ο χρήστης έχει τη δυνατότητα να εισάγει τις μεταβλητές που υπάρχουν στο workspace του command window του MATLAB. Η διαδικασία αυτή είναι αυτοματοποιημένη και απλά, με ένα παράθυρο διαλόγου απλού μηνύματος (`msgbox`), ενημερώνεται ο χρήστης για τον αριθμό των μεταβλητών που καταχωρηθήκαν ή με ένα παράθυρο διαλόγου προσοχής (`warnDlg`) αν δεν έχει γίνει καμιά καταχώρηση μεταβλητής. Η συνάρτηση που χρησιμοποιείται στο m-file για το συγκεκριμένο υπό-μενού είναι η εξής:

```
function workspace_Callback(hObject, eventdata, handles)
```

Στην συνάρτηση αυτή, η οποία είναι η συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.basic_operations,'enable','on');  
set(handles.advanoper,'enable','on');  
set(handles.equations,'enable','on');  
set(handles.savework,'enable','on');  
set(handles.deletematrix,'enable','on');
```

οι οποίες ορίζουν την ιδιότητα `enable` στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού. Επίσης έχουν προστεθεί οι εντολές

```
user_struct=get(handles.figure1,'UserData');  
k2=length(user_struct);  
evalin('base','save matr1.mat');  
matlist=who('-file','matr1.mat');  
k=length(matlist);  
if k>0  
for i=1:k
```

```

temp1=['load matr1 ',matlist{i}];
eval(temp1);
matnum=[];
temp2=['matnum=',matlist{i},'];
eval(temp2);
user_struct(i+k2).name=matlist{i};
user_struct(i+k2).matr=matnum;
end
str=num2str(k);
msgbox(['variables in workspace ',str],'attention');
end
set(handles.figure1,'UserData',user_struct);
if k==0
    warndlg('empty workspace','attention');
end

```

στις οποίες ορίζονται οι μεταβλητές από το workspace με την χρήση της εντολής `evalin` και αποθηκεύονται στην δομή `user_struct`. Τέλος χρησιμοποιείται και το m-file `popinit` για το οποίο έχουμε αναφερθεί παραπάνω.

Με την κλήση του υπό-μενού **Delete Matrix** ο χρήστης έχει τη δυνατότητα να διαγράψει έναν πίνακα. Αφού πρώτα ο χρήστης επιλέξει κάποιο πίνακα, χρησιμοποιώντας το push button `delete matrix` διαγράφει τον επιλεγμένο πίνακα. Οι συναρτήσεις που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. `function deletematrix_Callback(hObject, eventdata, handles)`
2. `function popupmenu1_CreateFcn(hObject, eventdata, handles)`
3. `function popupmenu1_Callback(hObject, eventdata, handles)`
4. `function pushbutton3_Callback(hObject, eventdata, handles)`

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.popupmenu1,'enable','on');  
set(handles.pushbutton3,'enable','on','visible','on');
```

οι οποίες ορίζουν τις ιδιότητες `visible` και `enable` στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού.

Οι συναρτήσεις (2),(3) και (4) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Save work** ο χρήστης έχει τη δυνατότητα να αποθηκεύσει την δουλειά του σε ένα `mat-file` έτσι ώστε να μπορεί να την φορτώσει οποιαδήποτε στιγμή θέλει. Η συνάρτηση που χρησιμοποιείται στο `m-file` για το συγκεκριμένο υπό-μενού είναι η εξής:

```
function savework_Callback(hObject, eventdata, handles)
```

Στην συνάρτηση αυτή, η οποία είναι η συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
user_struct=get(handles.figure1,'UserData');  
save 'dimitris.mat' user_struct
```

οι οποίες αποθηκεύουν την δομή `user_struct` στο `mat-file` `dimitris.mat`.

Με την κλήση του υπό-μενού **Load work** ο χρήστης έχει τη δυνατότητα να φορτώσει την δουλειά που είχε αποθηκεύσει σε ένα `mat-file`. Η συνάρτηση που χρησιμοποιείται στο `m-file` για το συγκεκριμένο υπό-μενού είναι η εξής:

```
function loadwork_Callback(hObject, eventdata, handles)
```

Στην συνάρτηση αυτή, η οποία είναι η συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.basic_operations,'enable','on');  
set(handles.advanoper,'enable','on');  
set(handles.equations,'enable','on');  
set(handles.savework,'enable','on');  
set(handles.deletematrix,'enable','on');
```

οι οποίες ορίζουν την ιδιότητα `enable` στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού. Επίσης, έχουν προστεθεί οι εντολές

```
if ~exist('dimitris.mat')
    warndlg(' saved file not exist');
    return;
end
load dimitris
set(handles.figure1,'UserData',user_struct);
```

με τις οποίες ελέγχεται πρώτα αν υπάρχει το αρχείο `dimitris.mat` και έπειτα φορτώνεται στην μνήμη του MATLAB. Τέλος η δομή `user_struct` αποθηκεύεται στην ιδιότητα `User Data`.

Με την κλήση του υπό-μενού **Exit** ο χρήστης έχει τη δυνατότητα να τερματίσει την εφαρμογή. Η συνάρτηση που χρησιμοποιείται στο `m-file` για το συγκεκριμένο υπό-μενού είναι η εξής:

```
function exit_Callback(hObject, eventdata, handles)
```

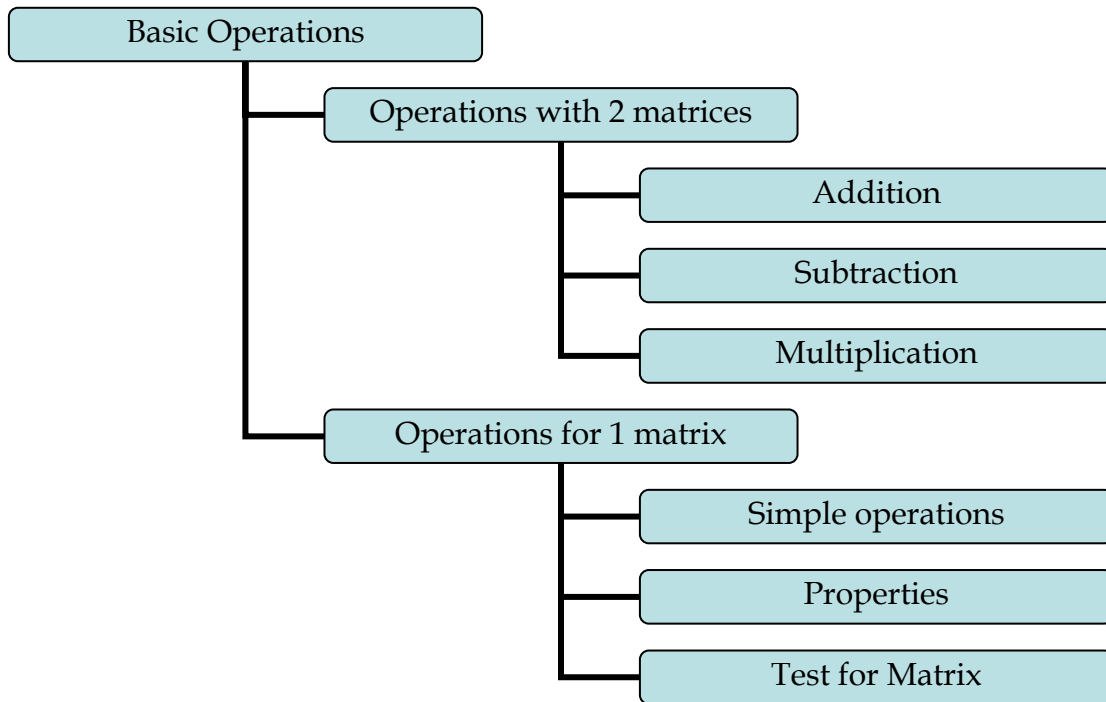
Στην συνάρτηση αυτή, η οποία είναι η συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
h=questdlg('exit from program?');
switch h
    case 'Yes'
        closereq;
end
```

οι οποίες δημιουργούν ένα παράθυρο διαλόγου (`questdlg`) και ανάλογα την επιλογή του χρήστη τερματίζεται ή όχι η εφαρμογή.

3.2.8.2 Μενού basic operations

Το μενού basic operations έχει την παρακάτω δομή:



Μενού Basic Operations

Το μενού **Basic Operations** δίνει τη δυνατότητα στον χρήστη να επιλέξει ένα από τα υπό-μενού όπως φαίνεται στο διάγραμμα. Η συνάρτηση που χρησιμοποιείται στο m-file για το συγκεκριμένο μενού είναι η εξής:

```
function basic_operations_Callback(hObject, eventdata, handles)
```

Στην συνάρτηση αυτή, η οποία είναι συνάρτηση επιστροφής του μενού, έχουν προστεθεί οι εντολές

```
delete;
```

```
popinit;
```

Για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Operations with 2 matrices** ο χρήστης οδηγείται, όπως φαίνεται στο διάγραμμα, σε άλλα τρία υπό-μενού τα οποία

εκτελούν διάφορες πράξεις με δυο πολυωνυμικούς πίνακες. Η συνάρτηση που χρησιμοποιείται στο m-file για το συγκεκριμένο υπό-μενού είναι η εξής:

```
function oper2_Callback(hObject, eventdata, handles)
```

Στην συνάρτηση αυτή, η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.refresh2,'visible','off');  
set(handles.refresh1,'visible','on','enable','off');
```

οι οποίες ορίζουν τις ιδιότητες visible και enable στα σταθερά στοιχεία ελέγχου refresh1-2, ανάλογα με το αντικείμενο που θα χρησιμοποιηθεί.

Με την κλήση του υπό-μενού **Addition** ο χρήστης έχει τη δυνατότητα να υπολογίσει το άθροισμα των δυο πινάκων που έχει ήδη επιλέξει. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (handles) που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function addition_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function popupmenu2_CreateFcn(hObject, eventdata, handles)*
5. *function popupmenu2_Callback(hObject, eventdata, handles)*
6. *function refresh1_Callback(hObject, eventdata, handles)*
7. *function savepb_Callback(hObject, eventdata, handles)*
8. *function push4(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.popupmenu1,'enable','on');  
set(handles.popupmenu2,'enable','on');  
set(handles.p2,'string','addition',...  
    'Callback','polyxgui('push4',gcbo,[],guidata(gcbo))',...  
    'enable','on');  
set(handles.text7,'string',{'-----HELP TEXT----- '})
```

οι οποίες ορίζουν την ιδιότητα `enable` στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού. Επίσης, ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (`text7`) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου `push button p2`.

Στην συνάρτηση (8), η οποία είναι συνάρτηση επιστροφής του `pushbutton p2`, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;  
mat2=handles.mat2;  
[r1,c1]=size(mat1);  
[r2,c2]=size(mat2);  
if r1==r2 & c1==c2  
    result=mat1+mat2;  
else  
    errordlg('Matrices not of the same dimensions.');
```

```
return;  
end  
handles.result=result;
```

με τις οποίες στις μεταβλητές `mat1` και `mat2` έχουν καταχωρηθεί οι επιλεγμένοι πίνακες και εφόσον γίνει έλεγχος στις διαστάσεις των πινάκων, εκτελείται η πρόσθεση και το αποτέλεσμα καταχωρείται στην μεταβλητή `result`.

Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας `enable` στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5),(6),(7) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Subtraction** ο χρήστης έχει τη δυνατότητα να υπολογίσει τη διαφορά των δυο πινάκων που έχει ήδη επιλέξει. Οι

συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (handles) που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function subtraction_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function popupmenu2_CreateFcn(hObject, eventdata, handles)*
5. *function popupmenu2_Callback(hObject, eventdata, handles)*
6. *function refresh1_Callback(hObject, eventdata, handles)*
7. *function savepb_Callback(hObject, eventdata, handles)*
8. *function push5(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.popupmenu1,'enable','on');
set(handles.popupmenu2,'enable','on');
set(handles.p2,'string','subtract',...
    'Callback','polyxgui('push5',gcbo,[],guidata(gcbo))',...
    'enable','on');
set(handles.text7,'string',{'-----HELP TEXT-----'});
```

οι οποίες ορίζουν την ιδιότητα enable στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού. Επίσης, ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2.

Στην συνάρτηση (8), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;
mat2=handles.mat2;
[r1,c1]=size(mat1);
[r2,c2]=size(mat2);
if r1==r2 & c1==c2
    result=mat1-mat2;
```



```
else
    errordlg('Matrices not of the same dimensions. ');
    return;
end
handles.result=result;
```

με τις οποίες στις μεταβλητές `mat1` και `mat2` έχουν καταχωρηθεί οι επιλεγμένοι πίνακες και εφόσον γίνει έλεγχος στις διαστάσεις των πινάκων, εκτελείται η αφαίρεση και το αποτέλεσμα καταχωρείται στην μεταβλητή `result`. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας `enable` στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5),(6),(7) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Multiplication** ο χρήστης έχει τη δυνατότητα να υπολογίσει το γινόμενο των δυο πινάκων που έχει ήδη επιλέξει. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (`handles`) που χρησιμοποιούνται στο `m-file` για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. `function multiplication_Callback(hObject, eventdata, handles)`
2. `function popupmenu1_CreateFcn(hObject, eventdata, handles)`
3. `function popupmenu1_Callback(hObject, eventdata, handles)`
4. `function popupmenu2_CreateFcn(hObject, eventdata, handles)`
5. `function popupmenu2_Callback(hObject, eventdata, handles)`
6. `function refreshpb_Callback(hObject, eventdata, handles)`
7. `function savepb_Callback(hObject, eventdata, handles)`
8. `function push6(hObject, eventdata, handles)`

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.popupmenu1,'enable','on');
set(handles.popupmenu2,'enable','on');
set(handles.p2,'string','subtract',...
```

```
'Callback','polyxgui('push6',gcbo,[],guidata(gcbo))',...
'enable','on');
set(handles.text7,'string',{'-----HELP TEXT-----'});
```

οι οποίες ορίζουν την ιδιότητα enable στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού. Επίσης, ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2.

Στην συνάρτηση (8), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;
mat2=handles.mat2;
[r1,c1]=size(mat1);
[r2,c2]=size(mat2);
if c1==r2
    result=mat1*mat2;
else
    errordlg('Matrices not of the same dimensions.');
```

```
return;
end
handles.result=result;
```

με τις οποίες στις μεταβλητές mat1 και mat2 έχουν καταχωρηθεί οι επιλεγμένοι πίνακες και εφόσον γίνει έλεγχος στις διαστάσεις των πινάκων, εκτελείται ο πολλαπλασιασμός και το αποτέλεσμα καταχωρείται στην μεταβλητή result. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας enable στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5),(6),(7) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Operations for 1 matrix** ο χρήστης οδηγείται, όπως φαίνεται στο διάγραμμα, σε άλλα τρία υπό-μενού τα οποία εκτελούν διάφορες πράξεις σε έναν πολυωνυμικό πίνακα. Η συνάρτηση που χρησιμοποιείται στο m-file για το συγκεκριμένο υπό-μενού είναι η εξής:

```
function oper1_Callback(hObject, eventdata, handles)
```

Στην συνάρτηση αυτή, η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.refresh1, 'visible', 'off');
```

```
set(handles.refresh2, 'visible', 'on', 'enable', 'off');
```

οι οποίες ορίζουν τις ιδιότητες *visible* και *enable* στα σταθερά στοιχεία ελέγχου *refresh1-2*, ανάλογα με το αντικείμενο που θα χρησιμοποιηθεί.

Με την κλήση του υπό-μενού **Simple operations** ο χρήστης έχει τη δυνατότητα να εκτελέσει την πράξη που επιθυμεί σε ένα πολυωνυμικό πίνακα χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου *popup menu p1* για την επιλογή της επιθυμητής πράξης και το προσωρινό στοιχείο ελέγχου *push button calculate* για την εκτέλεση της επιλεγμένης πράξης. Επίσης, γίνονται ενεργά τα σταθερά στοιχεία ελέγχου *push button refresh* και *save* και το σταθερό στοιχείο ελέγχου *popup menu 1*. Οι πράξεις που μπορεί να επιλέξει ο χρήστης είναι:

- Υπολογισμός ανάστροφου (*transpose*)
- Υπολογισμός συζυγή (*conjugate*)
- Υπολογισμός προσαρτημένου (*adjoint*)
- Υπολογισμός αντιστρόφου (*inverse*)
- Υπολογισμός ψευδόαντιστρόφου (*pseudoinverse*)
- Υπολογισμός μηδενικής βάσης (*nullspace*)
- Υπολογισμός ελάχιστης βάσης (*minimal basis*)

Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (*handles*) που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function simpleoper_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*

4. *function refresh2_Callback(hObject, eventdata, handles)*
5. *function savepb_Callback(hObject, eventdata, handles)*
6. *function popup1(hObject, eventdata, handles)*
7. *function push1(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p1,'visible','on',...
    'string','choose operation | transpose | conjugate | adjoint | inverse | ...
    pseudoinverse | nullspace | minimal basis',...
    'Callback','polyxgui(''popup1'',gcbo,[],guidata(gcbo))');
set(handles.p2,'string','calculate','Callback',...
    'polyxgui(''push1'',gcbo,[],guidata(gcbo))', 'enable','on');
```

οι οποίες ορίζουν διάφορες ιδιότητες στα προσωρινά στοιχεία ελέγχου popup menu p1 και push button p2.

Στην συνάρτηση (6), η οποία είναι συνάρτηση επιστροφής του popup menu p1, έχουν προστεθεί οι εντολές

```
set(handles.popupmenu1,'enable','on');
set(handles.p2,'enable','on');
pop=get(handles.p1,'value');
switch pop
case 1
    set(handles.text7,'string','must choice operation');
case 2
    set(handles.text7,'string',{'-----HELP TEXT-----'});
case 3
    set(handles.text7,'string',{'-----HELP TEXT-----'});
case 4
    set(handles.text7,'string',{'-----HELP TEXT-----'});
case 5
    set(handles.text7,'string',{'-----HELP TEXT-----'});
case 6
```

```
    set(handles.text7,'string',{'-----HELP TEXT-----'});  
case 7  
    set(handles.text7,'string',{'-----HELP TEXT-----'});  
case 8  
    set(handles.text7,'string',{'-----HELP TEXT-----'});  
end
```

με τις οποίες στην μεταβλητή pop καταχωρείται η επιλογή του χρήστη και με την εντολή switch ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) για κάθε πράξη που επιλέγεται. Επίσης ορίζεται η ιδιότητα enable στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού.

Στην συνάρτηση (7), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;  
pop=get(handles.p1,'value');  
switch pop  
case 1  
    warndlg('must choose some operation');  
    return;  
case 2  
    set(handles.text8,'string','Transpose of Matrix1 =');  
    result=mat1.';  
case 3  
    set(handles.text8,'string','Conjugate of Matrix1 =');  
    result=conj(mat1);  
case 4  
    [d1,d2]=size(mat1);  
    if d1==d2  
        set(handles.text8,'string','Adjoint of Matrix1 =');  
        result=adj(mat1);
```

```
    else
        errordlg('Matrix must be square')
        return;
    end
case 5
    [d1,d2]=size(mat1);
    if d1==d2
        set(handles.text8,'string','Inverse of Matrix1 =');
        result=inv(mat1);
    else
        errordlg('Matrix must be square')
        return;
    end
case 6
    result=pinv(mat1);
case 7
    set(handles.text8,'string','Null Space of Matrix1 =');
    result=null(mat1);
case 8
    set(handles.text8,'string','Minimal basis of Matrix1 =');
    result=minbasis(mat1);
end
handles.result=result;
```

με τις οποίες στην μεταβλητή *mat1* έχει καταχωρηθεί ο επιλεγμένος πίνακας και στην μεταβλητή *por* έχει καταχωρηθεί η επιλεγμένη πράξη και εφόσον γίνει έλεγχος στη διάσταση του πίνακα, αν χρειάζεται, εκτελείται η επιλεγμένη πράξη, εμφανίζεται ένα κατάλληλο μήνυμα στο *text8* και το αποτέλεσμα καταχωρείται στην μεταβλητή *result*. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας *enable* στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **properties** ο χρήστης έχει τη δυνατότητα να υπολογίσει την ιδιότητα που επιθυμεί σε ένα πολυωνυμικό πίνακα χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου popup menu p1 για την επιλογή του επιθυμητής ιδιότητας και το προσωρινό στοιχείο ελέγχου push button calculate για τον υπολογισμό της επιλεγμένης ιδιότητας. Επίσης, γίνεται ενεργό το σταθερό στοιχείο ελέγχου push button refresh και το σταθερό στοιχείο ελέγχου popup menu 1. Οι ιδιότητες που μπορεί να επιλέξει ο χρήστης είναι:

- Υπολογισμός ορίζουσας (determinant)
- Υπολογισμός τάξης (rank)
- Υπολογισμός ριζών (roots)
- Υπολογισμός βαθμού (degree)

Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (handles) που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function propertiesmat_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function refresh2_Callback(hObject, eventdata, handles)*
5. *function popup2(hObject, eventdata, handles)*
6. *function push2(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p1,'visible','on','enable','on',...  
    'string','choose property | determinant | rank | roots | degree',...  
    'Callback','polyxgui(''popup2'',gcbo,[],guidata(gcbo))');  
set(handles.p2,'string','calculate',  
    'Callback','polyxgui(''push2'',gcbo,[],guidata(gcbo))');
```

οι οποίες ορίζουν διάφορες ιδιότητες στα προσωρινά στοιχεία ελέγχου popup menu p1 και push button p2.

Στην συνάρτηση (5), η οποία είναι συνάρτηση επιστροφής του popup menu p1, έχουν προστεθεί οι εντολές

```

set(handles.popupmenu1,'enable','on');
set(handles.p2,'enable','on');
pop=get(handles.p1,'value');
switch pop
  case 1
    set(handles.text7,'string','must choice operation');
    set(handles.checkbox1,'visible','off');
    set(handles.checkbox2,'visible','off');
    set(handles.checkbox3,'visible','off');
  case 2
    set(handles.checkbox1,'visible','off');
    set(handles.checkbox2,'visible','off');
    set(handles.checkbox3,'visible','off');
    set(handles.text7,'string',{' -----HELP TEXT-----'});
  case 3
    set(handles.checkbox1,'visible','off');
    set(handles.checkbox2,'visible','off');
    set(handles.checkbox3,'visible','off');
    set(handles.text7,'string',{' -----HELP TEXT-----'});
  case 4
    set(handles.checkbox1,'visible','on','string','all');
    set(handles.checkbox2,'visible','on','string','det');
    set(handles.checkbox3,'visible','on','string','eig');
    set(handles.text7,'string',{' -----HELP TEXT-----'});
  case 5
    set(handles.checkbox1,'visible','on','string','ent');
    set(handles.checkbox2,'visible','on','string','row');

```



```
set(handles.checkbox3,'visible','on','string','col');  
set(handles.text7,'string',{' -----HELP TEXT-----'});
```

end

με τις οποίες στην μεταβλητή pop καταχωρείται η επιλογή του χρήστη και με την εντολή switch ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) για κάθε πράξη που επιλέγεται. Επίσης ορίζεται η ιδιότητα enable στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού και επιπλέον οι ιδιότητες visible και string στα check box τα οποία είναι απαραίτητα στις συγκεκριμένες πράξεις.

Στην συνάρτηση (6), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;  
pop=get(handles.p1,'value');  
switch pop  
case 1  
    warndlg('must choose property');  
    return;  
case 2  
    [d1,d2]=size(mat1);  
    if d1==d2  
        set(handles.text8,'string','Determinant of Matrix1 = ');  
        result=det(mat1);  
    else  
        errordlg('Matrix must be square','WRONG')  
        return;  
    end  
case 3  
    set(handles.text8,'string','Rank of Matrix1 = ');  
    result=rank(mat1);  
case 4
```

```

if all(all(deg(mat1,'ent')==0))
    errordlg('no roots in numeric matrix');
    return;
end
ch1=get(handles.checkbox1,'value');
ch2=get(handles.checkbox2,'value');
ch3=get(handles.checkbox3,'value');
if ch1 & ch2 & ch3
    warndlg('wrong');
    return;
elseif ch2 & ch3 & ~ch1
    warndlg('wrong');
    return;
end
if ch1 & ~ch2 & ~ch3
    result=roots(mat1,'all');
elseif ~ch1 & ch2 & ~ch3
    result=roots(mat1,'det');
elseif ~ch1 & ~ch2 & ch3
    result=roots(mat1,'eig');
elseif ch1 & ch2 & ~ch3
    result=roots(mat1,'det','all');
elseif ch1 & ~ch2 & ch3
    result=roots(mat1,'eig','all');
elseif ~ch1 & ~ch2 & ~ch3
    result=roots(mat1);
end
if isempty(result)
    errordlg('no roots')
    return;
end

```

```
set(handles.text8,'string','Roots of Matrix1 =');  
case 5  
ch1=get(handles.checkbox1,'value');  
ch2=get(handles.checkbox2,'value');  
ch3=get(handles.checkbox3,'value');  
if ch1 & ch2 & ch3  
    warndlg('wrong');  
    return;  
elseif ~ch1 & ch2 & ch3  
    warndlg('wrong');  
    return;  
elseif ch1 & ~ch2 & ch3  
    warndlg('wrong');  
    return;  
elseif ch1 & ch2 & ~ch3  
    warndlg('wrong');  
    return;  
end  
set(handles.text8,'string','Degree of Matrix1 =');  
if ch1 & ~ch2 & ~ch3  
    result=deg(mat1,'ent');  
elseif ~ch1 & ch2 & ~ch3  
    result=deg(mat1,'row');  
elseif ~ch1 & ~ch2 & ch3  
    result=deg(mat1,'col');  
elseif ~ch1 & ~ch2 & ~ch3  
    result=deg(mat1);  
end  
end
```

με τις οποίες στην μεταβλητή mat1 έχει καταχωρηθεί ο επιλεγμένος πίνακας, στην μεταβλητή rop έχει καταχωρηθεί η επιλεγμένη πράξη και στις μεταβλητές

ch1,ch2 και ch3 οι επιλογές των check box και εφόσον γίνει έλεγχος στη διάσταση του πίνακα, αν χρειάζεται, αλλά και στην σωστή επιλογή των check box, εκτελείται η επιλεγμένη πράξη, εμφανίζεται ένα κατάλληλο μήνυμα στο text8 και το αποτέλεσμα καταχωρείται στην μεταβλητή result. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας enable στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Test for Matrix** ο χρήστης έχει τη δυνατότητα να εκτελέσει το test που επιθυμεί σε ένα πολυωνυμικό πίνακα χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου popup menu p1 για την επιλογή του επιθυμητού test και το προσωρινό στοιχείο ελέγχου push button calculate για την εκτέλεση του επιλεγμένου test. Επίσης, γίνεται ενεργό το σταθερό στοιχείο ελέγχου push button refresh και το σταθερό στοιχείο ελέγχου popup menu 1. Τα test που μπορεί να επιλέξει ο χρήστης είναι:

- Έλεγχος εάν ο πίνακας έχει πλήρη τάξη
- Έλεγχος εάν ο πίνακας είναι μη ομαλός (singular)
- Έλεγχος εάν ο πίνακας είναι unimodular
- Έλεγχος εάν ο πίνακας είναι proper
- Έλεγχος εάν ο πίνακας είναι πρώτος (prime)
- Έλεγχος εάν ο πίνακας είναι ευσταθής (stable)

Οι συναρτήσεις που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function testmat_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function refresh2_Callback(hObject, eventdata, handles)*
5. *function popup3(hObject, eventdata, handles)*
6. *function push3(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p1,'visible','on',...  
    'string','choose test | rank | singular | unimodular | proper | prime | stable',...  
    'Callback','polyxgui(''popup3'',gcbo,[],guidata(gcbo))');  
set(handles.p2,'string','calculate',...  
    'Callback','polyxgui(''push3'',gcbo,[],guidata(gcbo))',...  
    'enable','on');
```

στις οριζόντιες διάφορες ιδιότητες στα προσωρινά στοιχεία ελέγχου popup menu p1 και push button p2.

Στην συνάρτηση (5), η οποία είναι συνάρτηση επιστροφής του popup menu p1, έχουν προστεθεί οι εντολές

```
pop=get(handles.p1,'value');  
switch pop  
    case 1  
        set(handles.text7,'string',{' must choice test for selected matrix' });  
    case 2  
        set(handles.text7,'string',{' -----HELP TEXT-----' });  
    case 3  
        set(handles.text7,'string',{' -----HELP TEXT-----' });  
    case 4  
        set(handles.text7,'string',{' -----HELP TEXT-----' });  
    case 5  
        set(handles.text7,'string',{' -----HELP TEXT-----' });  
    case 6  
        set(handles.text7,'string',{' -----HELP TEXT-----' });  
    case 7  
        set(handles.text7,'string',{' -----HELP TEXT-----' });  
end
```

με τις οποίες στην μεταβλητή `pop` καταχωρείται η επιλογή του χρήστη και με την εντολή `switch` ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου για κάθε `test` που επιλέγεται.

Στην συνάρτηση (6), η οποία είναι συνάρτηση επιστροφής του `pushbutton p2`, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```

mat1=handles.mat1;
pop=get(handles.p1,'value');
switch pop
case 1
    warndlg('must choose some test');
    return;
case 2
    result=isfullrank(mat1);
    if result==1
        set(handles.text8,'string','The matrix1 is fullrank');
    else
        set(handles.text8,'string','The matrix1 is not fullrank');
    end
case 3
    [d1,d2]=size(mat1);
    if d1==d2
        result=issingular(mat1);
        if result==1
            set(handles.text8,'string','The matrix1 is singular');
        else
            set(handles.text8,'string','The matrix1 is not singular');
        end
    else
        errordlg('Matrix must be square')
    return;

```

```
end
case 4
[d1,d2]=size(mat1);
if d1==d2
    result=isunimod(mat1);
    if result==1
        set(handles.text8,'string','The matrix1 is unimodular');
    else
        set(handles.text8,'string','The matrix1 is not unimodular');
    end
else
    errorDlg('Matrix must be square')
    return;
end
case 5
result=isproper(mat1);
if result==1
    set(handles.text8,'string','The matrix1 is proper');
else
    set(handles.text8,'string','The matrix1 is not proper');
end
case 6
result=isprime(mat1);
if result==1
    set(handles.text8,'string','The matrix1 is prime');
else
    set(handles.text8,'string','The matrix1 is not prime');
end
case 7
[d1,d2]=size(mat1);
if d1==d2
```

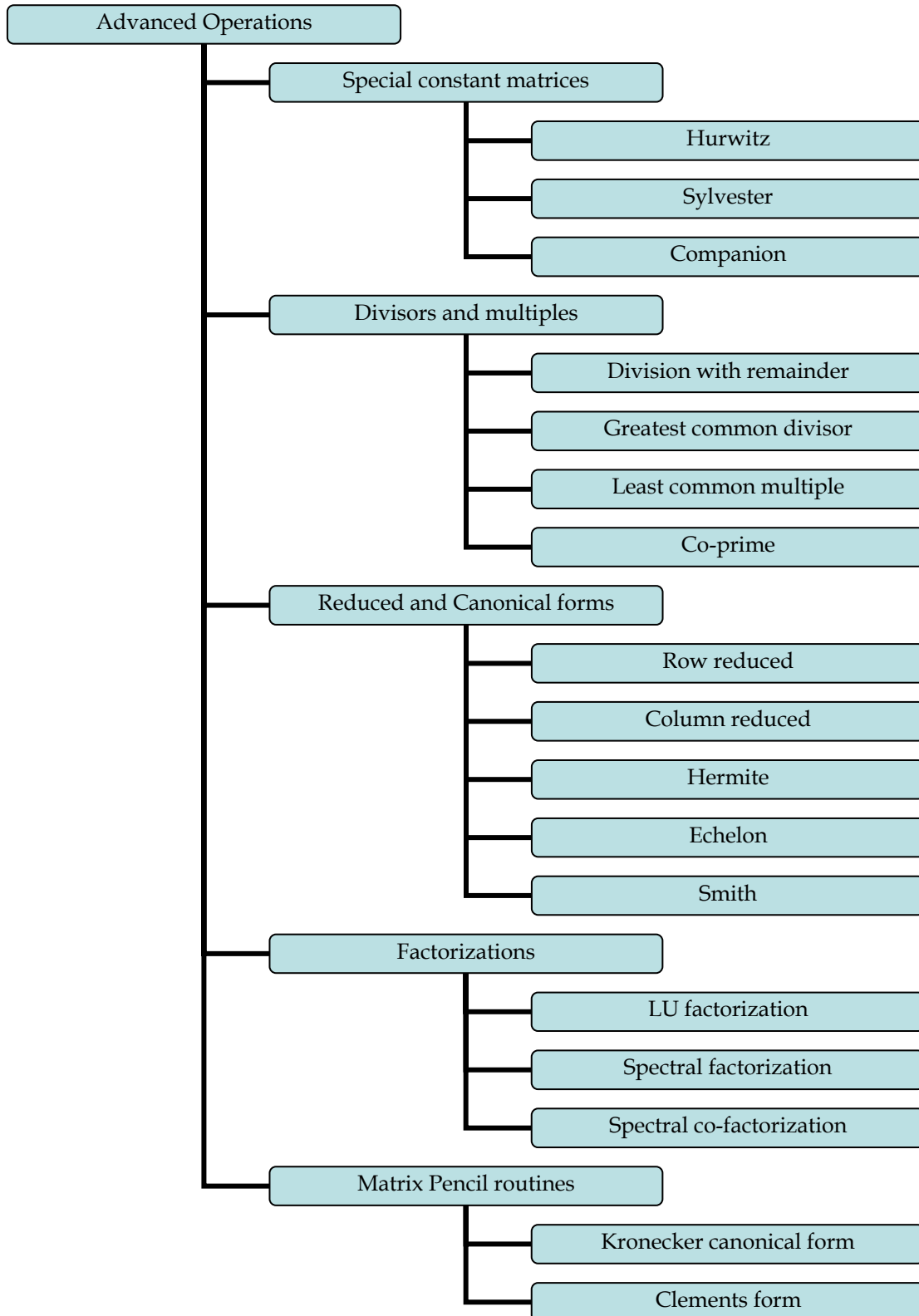
```
result=isstable(mat1);
if result==1
    set(handles.text8,'string','The matrix1 is stable');
else
    set(handles.text8,'string','The matrix1 is not stable');
end
else
    errordlg('Matrix must be square')
    return;
end
end
```

με τις οποίες στην μεταβλητή `mat1` έχει καταχωρηθεί ο επιλεγμένος πίνακας και στην μεταβλητή `pop` έχει καταχωρηθεί το επιλεγμένο `test` και εφόσον γίνει έλεγχος στη διάσταση του πίνακα, αν χρειάζεται, εκτελείται το επιλεγμένο `test` και το αποτέλεσμα καταχωρείται στη μεταβλητή `result` και εμφανίζεται στο σταθερό στοιχείο ελέγχου `text8`. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο που στην συγκεκριμένη πράξη είναι ο ορισμός της ιδιότητας `enable` στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

3.2.8.3 Μενού advanced operations

Το μενού advanced operations έχει την παρακάτω δομή:



Μενού Advanced Operations

Το μενού **Advanced Operations** δίνει τη δυνατότητα στον χρήστη να επιλέξει ένα από τα υπό-μενού όπως φαίνεται στο διάγραμμα. Η συνάρτηση που χρησιμοποιείται στο m-file για το συγκεκριμένο μενού είναι η εξής:

```
function advanoper_Callback(hObject, eventdata, handles)
```

Στην συνάρτηση αυτή, η οποία είναι συνάρτηση επιστροφής του μενού, έχουν προστεθεί οι εντολές

```
delete;  
set(handles.refresh1,'visible','on','enable','off');  
set(handles.refresh2,'visible','off');  
popinit;
```

στις οποίες ορίζονται οι ιδιότητες `visible` και `enable` στα σταθερά στοιχεία ελέγχου `refresh1-2`, ανάλογα με το αντικείμενο που θα χρησιμοποιηθεί. Επίσης γίνεται και αρχικοποίηση με τις `delete` και `popinit` για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Special constant matrices** ο χρήστης οδηγείται, όπως φαίνεται στο διάγραμμα, σε άλλα τρία υπό-μενού τα οποία υπολογίζουν ειδικούς σταθερούς πίνακες ενός πολυωνυμικού πίνακα. Η συνάρτηση που χρησιμοποιείται στο m-file για το συγκεκριμένο υπό-μενού είναι η εξής:

```
function compranmat_Callback(hObject, eventdata, handles)
```

Στην συνάρτηση αυτή, η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.popurpmenu1,'enable','on');  
set(handles.popurpmenu2,'enable','off');  
set(handles.popurpmenu3,'enable','off');
```

οι οποίες ορίζουν την ιδιότητα `enable` στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού.

Με την κλήση του υπό-μενού **Hurwitz** ο χρήστης έχει τη δυνατότητα να υπολογίσει τον Hurwitz πίνακα ενός πολυωνυμικού πίνακα που έχει ήδη

επιλέξει χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου p2. Επίσης, γίνονται ενεργά τα σταθερά στοιχεία ελέγχου push button refresh και save και το σταθερό στοιχείο ελέγχου popup menu 1. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (handles) που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function hurwitz_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function refresh1_Callback(hObject, eventdata, handles)*
5. *function savepb_Callback(hObject, eventdata, handles)*
6. *function push101(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string','calculate',...  
    'Callback','polyxgui('push101',gcbo,[],guidata(gcbo))','enable','on');  
set(handles.text7,'string',{' -----HELP TEXT-----' });
```

στις οποίες ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2.

Στην συνάρτηση (6), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;  
result=hurwitz(mat1);  
handles.result=result;
```

με τις οποίες στην μεταβλητή mat1 έχει καταχωρηθεί ο επιλεγμένος πίνακας, γίνεται ο υπολογισμός και το αποτέλεσμα καταχωρείται στην μεταβλητή result. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας enable στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Sylvester** ο χρήστης έχει τη δυνατότητα να υπολογίσει τον Sylvester πίνακα ενός πολυωνυμικού πίνακα που έχει ήδη επιλέξει χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου p2. Επίσης, γίνονται ενεργά τα σταθερά στοιχεία ελέγχου push button refresh και save και το σταθερό στοιχείο ελέγχου popup menu 1. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (handles) που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function sylvester_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function refresh1_Callback(hObject, eventdata, handles)*
5. *function savepb_Callback(hObject, eventdata, handles)*
6. *function push102(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string',...
    'calculate','Callback','polyxgui('push102',gcbo,[],guidata(gcbo))',...
    'enable','on');
set(handles.text7,'string',{' -----HELP TEXT-----'});
```

στις οποίες ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2.

Στην συνάρτηση (6), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;
result=sylv(mat1);
handles.result=result;
```

με τις οποίες στην μεταβλητή mat1 έχει καταχωρηθεί ο επιλεγμένος πίνακας, γίνεται ο υπολογισμός και το αποτέλεσμα καταχωρείται στην μεταβλητή result. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την

εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας `enable` στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Companion** ο χρήστης έχει τη δυνατότητα να υπολογίσει τον Companion πίνακα ενός πολυωνυμικού πίνακα που έχει ήδη επιλέξει χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου `p2`. Επίσης, γίνονται ενεργά τα σταθερά στοιχεία ελέγχου `push button refresh` και `save` και το σταθερό στοιχείο ελέγχου `popup menu 1`. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (`handles`) που χρησιμοποιούνται στο `m-file` για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function companion_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function refresh1_Callback(hObject, eventdata, handles)*
5. *function savepb_Callback(hObject, eventdata, handles)*
6. *function push103(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string',...  
    'calculate','Callback','polyxgui('push103',gcbo,[],guidata(gcbo))',...  
    'enable','on');  
set(handles.text7,'string',{'-----HELP TEXT-----'});
```

στις οποίες ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (`text7`) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου `push button p2`.

Στην συνάρτηση (6), η οποία είναι συνάρτηση επιστροφής του `pushbutton p2`, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;  
result=compan(mat1);
```

handles.result=result;

με τις οποίες στην μεταβλητή *mat1* έχει καταχωρηθεί ο επιλεγμένος πίνακας, γίνεται ο υπολογισμός και το αποτέλεσμα καταχωρείται στην μεταβλητή *result*. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας *enable* στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Divisors and multiples** ο χρήστης οδηγείται, όπως φαίνεται στο διάγραμμα, σε άλλα τέσσερα υπό-μενού τα οποία εκτελούν διάφορες πράξεις με δυο πολυωνυμικούς πίνακες. Η συνάρτηση που χρησιμοποιείται στο *m-file* για το συγκεκριμένο υπό-μενού είναι η εξής:

function divmult_Callback(hObject, eventdata, handles)

Στην συνάρτηση αυτή, η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

set(handles.popupmenu1,'enable','on');

set(handles.popupmenu2,'enable','on');

set(handles.popupmenu3,'enable','off');

οι οποίες ορίζουν την ιδιότητα *enable* στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού.

Με την κλήση του υπό-μενού **Division with remainder** ο χρήστης έχει τη δυνατότητα να υπολογίσει το ηλίκο και το υπόλοιπο της διαίρεσης (αριστερής ή δεξιάς) δυο πινάκων που έχει ήδη επιλέξει. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (*handles*) που χρησιμοποιούνται στο *m-file* για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function divrem_Callback(hObject, eventdata, handles)*

2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*

3. *function popupmenu1_Callback(hObject, eventdata, handles)*

4. *function popupmenu2_CreateFcn(hObject, eventdata, handles)*

5. *function popurpmenu2_Callback(hObject, eventdata, handles)*
6. *function refresh1_Callback(hObject, eventdata, handles)*
7. *function push121(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string','calculate',  
    'Callback','polyxgui('push121',gcbo,[],guidata(gcbo))',...  
    'enable','on');  
set(handles.checkbox1,'visible','on','string','right division');  
set(handles.checkbox2,'visible','on','string','left division');  
set(handles.text7,'string',{'-----HELP TEXT----- '})
```

οι οποίες ορίζουν τις ιδιότητες visible και string στα check box που είναι απαραίτητα για το συγκεκριμένο υπό-μενού. Επίσης, ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2.

Στην συνάρτηση (7), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;  
mat2=handles.mat2;  
[rmat1,cmat1]=size(mat1);  
[rmat2,cmat2]=size(mat2);  
ch1=get(handles.checkbox1,'value');  
ch2=get(handles.checkbox2,'value');  
if (~ch1 & ~ch2) | (ch1 & ch2)  
    warndlg('must choice one checkbox')  
    return;  
end  
if ch1 & ~ch2  
    if cmat1 ~= cmat2  
        errordlg('Matrix1 and matrix2 must have the same numbers of columns');
```

```
        return;
    else
        [result1,result2]=rdiv(mat1,mat2);
    end
elseif ~ch1 & ch2
    if rmat1 ~= rmat2
        error('Matrix1 and matrix2 must have the same numbers of rows!');
        return;
    else
        [result1,result2]=ldiv(mat1,mat2);
    end
end
```

με τις οποίες στις μεταβλητές `mat1` και `mat2` έχουν καταχωρηθεί οι επιλεγμένοι πίνακες και στις μεταβλητές `ch1` και `ch2` οι επιλογές των `check box` και εφόσον γίνει έλεγχος στις διαστάσεις των πινάκων, αλλά και στην σωστή επιλογή των `check box`, εκτελείται η διαίρεση και τα αποτελέσματα καταχωρούνται στις μεταβλητές `result1` και `result2`. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας `enable` στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5),(6),(7) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Greatest common divisor** ο χρήστης έχει τη δυνατότητα να υπολογίσει τον μέγιστο κοινό διαιρετή (αριστερό ή δεξιό) δυο πινάκων που έχει ήδη επιλέξει. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (`handles`) που χρησιμοποιούνται στο `m-file` για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function* `divgre_Callback(hObject, eventdata, handles)`
2. *function* `popupmenu1_CreateFcn(hObject, eventdata, handles)`
3. *function* `popupmenu1_Callback(hObject, eventdata, handles)`

4. *function popupmenu2_CreateFcn(hObject, eventdata, handles)*
5. *function popupmenu2_Callback(hObject, eventdata, handles)*
6. *function refresh1_Callback(hObject, eventdata, handles)*
7. *function push122(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string','calculate',  
    'Callback','polyxgui('push122',gcbo,[],guidata(gcbo)),...  
    'enable','on');  
set(handles.checkbox1,'visible','on','string','right GCD');  
set(handles.checkbox2,'visible','on','string','left GCD');  
set(handles.text7,'string',{'-----HELP TEXT----- '})
```

οι οποίες ορίζουν τις ιδιότητες *visible* και *string* στα *check box* που είναι απαραίτητα για το συγκεκριμένο υπό-μενού. Επίσης, ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (*text7*) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου *push button p2*.

Στην συνάρτηση (7), η οποία είναι συνάρτηση επιστροφής του *pushbutton p2*, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;  
mat2=handles.mat2;  
[rmat1,cmat1]=size(mat1);  
[rmat2,cmat2]=size(mat2);  
ch1=get(handles.checkbox1,'value');  
ch2=get(handles.checkbox2,'value');  
if (~ch1 & ~ch2) | (ch1 & ch2)  
    warndlg('must choice one checkbox')  
    return;  
end  
if ch1 & ~ch2  
    if cmat1 ~= cmat2
```

```
        errordlg('Matrix1 and matrix2 must have the same numbers of columns');
        return;
    else
        result=grd(mat1,mat2);
    end
elseif ~ch1 & ch2
    if rmat1 ~= rmat2
        errordlg('Matrix1 and matrix2 must have the same numbers of rows');
        return;
    else
        result=gld(mat1,mat2);
    end
end
```

με τις οποίες στις μεταβλητές `mat1` και `mat2` έχουν καταχωρηθεί οι επιλεγμένοι πίνακες και στις μεταβλητές `ch1` και `ch2` οι επιλογές των `check box` και εφόσον γίνει έλεγχος στις διαστάσεις των πινάκων, αλλά και στην σωστή επιλογή των `check box`, εκτελείται η διαίρεση και τα αποτελέσματα καταχωρούνται στις μεταβλητές `result1` και `result2`. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας `enable` στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5),(6),(7) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Least common multiple** ο χρήστης έχει τη δυνατότητα να υπολογίσει το ελάχιστο κοινό πολλαπλάσιο (αριστερό ή δεξιό) δυο πινάκων που έχει ήδη επιλέξει. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (`handles`) που χρησιμοποιούνται στο `m-file` για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function dioleast_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*

3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function popupmenu2_CreateFcn(hObject, eventdata, handles)*
5. *function popupmenu2_Callback(hObject, eventdata, handles)*
6. *function refresh1_Callback(hObject, eventdata, handles)*
7. *function push123(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string','calculate',  
    'Callback','polyxgui('push123',gcbo,[],guidata(gcbo))',...  
    'enable','on');  
set(handles.checkbox1,'visible','on','string','right LCM');  
set(handles.checkbox2,'visible','on','string','left LCM');  
set(handles.text7,'string',{'-----HELP TEXT----- '})
```

οι οποίες ορίζουν τις ιδιότητες visible και string στα check box που είναι απαραίτητα για το συγκεκριμένο υπό-μενού. Επίσης, ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2.

Στην συνάρτηση (7), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;  
mat2=handles.mat2;  
[rmat1,cmat1]=size(mat1);  
[rmat2,cmat2]=size(mat2);  
ch1=get(handles.checkbox1,'value');  
ch2=get(handles.checkbox2,'value');  
if (~ch1 & ~ch2) | (ch1 & ch2)  
    warndlg('must choice one checkbox')  
    return;  
end  
if ch1 & ~ch2
```

```
if rmat1 ~= rmat2
    errordlg('Matrix1 and matrix2 must have the same numbers of rows');
    return;
else
    result=rlm(mat1,mat2);
end
elseif ~ch1 & ch2
    if cmat1 ~= cmat2
        errordlg('Matrix1 and matrix2 must have the same numbers of columns');
        return
    else
        result=llm(mat1,mat2);
    end
end
```

με τις οποίες στις μεταβλητές `mat1` και `mat2` έχουν καταχωρηθεί οι επιλεγμένοι πίνακες και στις μεταβλητές `ch1` και `ch2` οι επιλογές των `check box` και εφόσον γίνει έλεγχος στις διαστάσεις των πινάκων, αλλά και στην σωστή επιλογή των `check box`, εκτελείται η διαίρεση και τα αποτελέσματα καταχωρούνται στις μεταβλητές `result1` και `result2`. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας `enable` στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5),(6),(7) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Co-prime** ο χρήστης έχει τη δυνατότητα να ελέγξει εάν δυο πολυωνυμικοί πίνακες είναι πρώτοι μεταξύ τους. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (`handles`) που χρησιμοποιούνται στο `m-file` για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function dioprri_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*

3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function popupmenu2_CreateFcn(hObject, eventdata, handles)*
5. *function popupmenu2_Callback(hObject, eventdata, handles)*
6. *function refresh1_Callback(hObject, eventdata, handles)*
7. *function push124(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string','calculate',  
    'Callback','polyxgui('push124',gcbo,[],guidata(gcbo))',...  
    'enable','on');  
set(handles.text7,'string',{'-----HELP TEXT----- '})
```

οι οποίες ορίζουν το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2.

Στην συνάρτηση (7), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;  
mat2=handles.mat2;  
result=isprime([mat1,mat2]);  
if result==1  
    set(handles.text8,'string','The matrices are coprime');  
else  
    set(handles.text8,'string','The matrices are not coprime');  
end
```

με τις οποίες στις μεταβλητές mat1 και mat2 έχουν καταχωρηθεί οι επιλεγμένοι πίνακες εκτελείται το test και το αποτέλεσμα καταχωρείται στην μεταβλητή result και εμφανίζεται στο σταθερό στοιχείο ελέγχου text8 . Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο που στην συγκεκριμένη πράξη είναι ο ορισμός της ιδιότητας enable στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5),(6),(7) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Reduced and Canonical forms** ο χρήστης οδηγείται, όπως φαίνεται στο διάγραμμα, σε άλλα έξι υπό-μενού τα οποία υπολογίζουν διάφορες ειδικές μορφές πινάκων ενός πολυωνυμικού πίνακα. Η συνάρτηση που χρησιμοποιείται στο m-file για το συγκεκριμένο υπό-μενού είναι η εξής:

```
function matrixform_Callback(hObject, eventdata, handles)
```

Στην συνάρτηση αυτή, η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.popupmenu1,'enable','on');
```

```
set(handles.popupmenu2,'enable','off');
```

```
set(handles.popupmenu3,'enable','off');
```

οι οποίες ορίζουν την ιδιότητα `enable` στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού.

Με την κλήση του υπό-μενού **Row reduced** ο χρήστης έχει τη δυνατότητα να υπολογίσει την Row reduced μορφή ενός πολυωνυμικού πίνακα που έχει ήδη επιλέξει χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου `p2`. Επίσης, γίνονται ενεργά τα σταθερά στοιχεία ελέγχου `push button refresh` και `save` και το σταθερό στοιχείο ελέγχου `popup menu 1`. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (`handles`) που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function rowredform_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function refresh1_Callback(hObject, eventdata, handles)*
5. *function savepb_Callback(hObject, eventdata, handles)*
6. *function push111(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string',...  
    'calculate','Callback','polyxgui('push111',gcbo,[],guidata(gcbo))',...  
    'enable','on');  
set(handles.text7,'string',{' -----HELP TEXT-----'});
```

στις οποίες ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2.

Στην συνάρτηση (6), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;  
result=rowred(mat1);  
handles.result=result;
```

με τις οποίες στην μεταβλητή mat1 έχει καταχωρηθεί ο επιλεγμένος πίνακας, γίνεται ο υπολογισμός και το αποτέλεσμα καταχωρείται στην μεταβλητή result. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας enable στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Column reduced** ο χρήστης έχει τη δυνατότητα να υπολογίσει την Column reduced μορφή ενός πολυωνομικού πίνακα που έχει ήδη επιλέξει χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου p2. Επίσης, γίνονται ενεργά τα σταθερά στοιχεία ελέγχου push button refresh και save και το σταθερό στοιχείο ελέγχου popup menu 1. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (handles) που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function colredform_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*

3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function refresh1_Callback(hObject, eventdata, handles)*
5. *function savepb_Callback(hObject, eventdata, handles)*
6. *function push112(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string',...
    'calculate','Callback','polyxgui('push112',gcbo,[],guidata(gcbo))',...
    'enable','on');
set(handles.text7,'string',{' -----HELP TEXT-----'});
```

στις οποίες ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2.

Στην συνάρτηση (6), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;
result=colred(mat1);
handles.result=result;
```

με τις οποίες στην μεταβλητή mat1 έχει καταχωρηθεί ο επιλεγμένος πίνακας, γίνεται ο υπολογισμός και το αποτέλεσμα καταχωρείται στην μεταβλητή result. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας enable στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Hermite** ο χρήστης έχει τη δυνατότητα να υπολογίσει την Hermite μορφή ενός πολυωνυμικού πίνακα που έχει ήδη επιλέξει χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου p2. Επίσης, γίνονται ενεργά τα σταθερά στοιχεία ελέγχου push button refresh και save και το σταθερό στοιχείο ελέγχου popup menu 1. Οι συναρτήσεις και τα αντίστοιχα

στοιχεία ελέγχου (*handles*) που χρησιμοποιούνται στο *m-file* για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function hermiteform_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function refresh1_Callback(hObject, eventdata, handles)*
5. *function savepb_Callback(hObject, eventdata, handles)*
6. *function push113(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string',...  
.....'calculate','Callback','polyxgui('push113',gcb0,[],guidata(gcb0))',...  
'enable','on');  
set(handles.text7,'string',{' -----HELP TEXT-----'});
```

στις οποίες ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (*text7*) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button *p2*.

Στην συνάρτηση (6), η οποία είναι συνάρτηση επιστροφής του pushbutton *p2*, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;  
result=hermite(mat1);  
handles.result=result;
```

με τις οποίες στην μεταβλητή *mat1* έχει καταχωρηθεί ο επιλεγμένος πίνακας, γίνεται ο υπολογισμός και το αποτέλεσμα καταχωρείται στην μεταβλητή *result*. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας *enable* στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Echelon** ο χρήστης έχει τη δυνατότητα να υπολογίσει την Echelon μορφή ενός πολυωνυμικού πίνακα που έχει ήδη επιλέξει χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου p2. Επίσης, γίνονται ενεργά τα σταθερά στοιχεία ελέγχου push button refresh και save και το σταθερό στοιχείο ελέγχου popup menu 1. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (handles) που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function echelonform_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function refresh1_Callback(hObject, eventdata, handles)*
5. *function savepb_Callback(hObject, eventdata, handles)*
6. *function push114(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string',...
    'calculate','Callback','polyxgui('push114',gcbo,[],guidata(gcbo))',...
    'enable','on');
set(handles.text7,'string',{' -----HELP TEXT-----'});
```

στις οποίες ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2.

Στην συνάρτηση (6), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;
[rmat1,cmat1] = size(mat1);
if rmat1 ~= cmat1
    errordlg('Matrix is not square. ');
    return;
elseif rank(mat1) < rmat1
    errordlg('Matrix is singular. ');
```

```
return  
end;  
mat1col = lcoef(mat1, 'col');  
if (rank(mat1col) < rmat1)  
    errordlg('Matrix is not column reduced.');
```

```
return;  
end;  
result=echelon(mat1);  
handles.result=result;
```

με τις οποίες στην μεταβλητή `mat1` έχει καταχωρηθεί ο επιλεγμένος πίνακας, στη συνέχεια γίνεται έλεγχος εάν ο πίνακας `mat1` είναι τετραγωνικός, εάν είναι μη ομαλός (singular) και τέλος εάν είναι column reduced. Έπειτα γίνεται ο υπολογισμός και το αποτέλεσμα καταχωρείται στην μεταβλητή `result`. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας `enable` στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Smith** ο χρήστης έχει τη δυνατότητα να υπολογίσει την Smith μορφή ενός πολυωνομικού πίνακα που έχει ήδη επιλέξει χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου `p2`. Επίσης, γίνονται ενεργά τα σταθερά στοιχεία ελέγχου `push button refresh` και `save` και το σταθερό στοιχείο ελέγχου `popup menu 1`. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (`handles`) που χρησιμοποιούνται στο `m-file` για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. `function smithform_Callback(hObject, eventdata, handles)`
2. `function popupmenu1_CreateFcn(hObject, eventdata, handles)`
3. `function popupmenu1_Callback(hObject, eventdata, handles)`
4. `function refresh1_Callback(hObject, eventdata, handles)`
5. `function savepb_Callback(hObject, eventdata, handles)`

6. *function push115(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string',...
    'calculate','Callback','polyxgui('push115',gcbo,[],guidata(gcbo))',...
    'enable','on');
set(handles.text7,'string',{' -----HELP TEXT-----'});
```

στις οποίες ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2.

Στην συνάρτηση (6), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;
result=smith(mat1);
handles.result=result;
```

με τις οποίες στην μεταβλητή mat1 έχει καταχωρηθεί ο επιλεγμένος πίνακας, γίνεται ο υπολογισμός και το αποτέλεσμα καταχωρείται στην μεταβλητή result. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας enable στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Factorizations** ο χρήστης οδηγείται, όπως φαίνεται στο διάγραμμα, σε άλλα τρία υπό-μενού τα οποία υπολογίζουν διάφορες παραγοντοποιήσεις ενός πολυωνυμικού πίνακα. Η συνάρτηση που χρησιμοποιείται στο m-file για το συγκεκριμένο υπό-μενού είναι η εξής:

```
function factorizations_Callback(hObject, eventdata, handles)
```

Στην συνάρτηση αυτή, η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.popupmenu1,'enable','on');
```

```
set(handles.popupmenu2,'enable','off');  
set(handles.popupmenu3,'enable','off');
```

οι οποίες ορίζουν την ιδιότητα `enable` στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού.

Με την κλήση του υπό-μενού **LU factorization** ο χρήστης έχει τη δυνατότητα να υπολογίσει την LU παραγοντοποίηση ενός πολυωνυμικού πίνακα που έχει ήδη επιλέξει χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου `p2`. Επίσης, γίνεται ενεργό το σταθερό στοιχείο ελέγχου `push button refresh` και το σταθερό στοιχείο ελέγχου `popup menu 1`. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (`handles`) που χρησιμοποιούνται στο `m-file` για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. `function factlu_Callback(hObject, eventdata, handles)`
2. `function popupmenu1_CreateFcn(hObject, eventdata, handles)`
3. `function popupmenu1_Callback(hObject, eventdata, handles)`
4. `function refresh1_Callback(hObject, eventdata, handles)`
5. `function push131(hObject, eventdata, handles)`

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string',...  
    'calculate','Callback','polyxgui('push131',gcbo,[],guidata(gcbo))',...  
    'enable','on');  
set(handles.text7,'string',{[ -----HELP TEXT----- ]});
```

στις οποίες ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (`text7`) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου `push button p2`.

Στην συνάρτηση (5), η οποία είναι συνάρτηση επιστροφής του `pushbutton p2`, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;  
[rmat1,cmat1]=size(mat1);  
if rmat1 ~= cmat1
```

```
    errordlg('Matrix is not square');  
    return;  
elseif rank(polyval(mat1, 0)) < rmat1  
    errordlg('Invalid matrix; constant term must be nonsingular.');
```

```
    return;  
end;  
[result1,result2]=lu(mat1);
```

με τις οποίες στην μεταβλητή `mat1` έχει καταχωρηθεί ο επιλεγμένος πίνακας, στη συνέχεια γίνεται έλεγχος εάν ο πίνακας `mat1` είναι τετραγωνικός και εάν ο `constant term` πίνακας είναι μη ομαλός (`singular`). Έπειτα γίνεται ο υπολογισμός και τα αποτελέσματα καταχωρούνται στις μεταβλητές `result1` και `result2`. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας `enable` στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Spectral factorization** ο χρήστης έχει τη δυνατότητα να υπολογίσει την Spectral factorization ενός πολυωνομικού πίνακα που έχει ήδη επιλέξει χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου `p2`. Επίσης, γίνεται ενεργό το σταθερό στοιχείο ελέγχου `push button refresh` και το σταθερό στοιχείο ελέγχου `popup menu 1`. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (`handles`) που χρησιμοποιούνται στο `m-file` για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. `function factspf_Callback(hObject, eventdata, handles)`
2. `function popupmenu1_CreateFcn(hObject, eventdata, handles)`
3. `function popupmenu1_Callback(hObject, eventdata, handles)`
4. `function refresh1_Callback(hObject, eventdata, handles)`
5. `function push132(hObject, eventdata, handles)`

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string',...  
    'calculate','Callback','polyxgui('push132',gcbo,[],guidata(gcbo))',...  
    'enable','on');  
set(handles.text7,'string',{' -----HELP TEXT-----'});
```

στις οποίες ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2.

Στην συνάρτηση (5), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
at1=handles.mat1;  
[rmat1,cmat1]=size(mat1);  
if rmat1 ~= cmat1  
    errordlg('Matrix is not square');  
    return;  
elseif norm(mat1-mat1')>10^-6  
    errordlg('Invalid matrix; is not para-Hermitian.');
```

```
return;  
end;  
[result1,result2]=spf(mat1);
```

με τις οποίες στην μεταβλητή mat1 έχει καταχωρηθεί ο επιλεγμένος πίνακας, στη συνέχεια γίνεται έλεγχος εάν ο πίνακας mat1 είναι τετραγωνικός και εάν είναι para-Hermitian. Έπειτα γίνεται ο υπολογισμός και τα αποτελέσματα καταχωρούνται στις μεταβλητές result1 και result2. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας enable στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Spectral co-factorization** ο χρήστης έχει τη δυνατότητα να υπολογίσει την Spectral co-factorization ενός πολυωνομικού πίνακα που έχει ήδη επιλέξει χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου p2. Επίσης, γίνεται ενεργό το σταθερό στοιχείο ελέγχου push button refresh και το σταθερό στοιχείο ελέγχου popup menu 1. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (handles) που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function factcospf_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function refresh1_Callback(hObject, eventdata, handles)*
5. *function push133(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string',...
    'calculate','Callback','polyxgui('push133',gcbo,[],guidata(gcbo))',...
    'enable','on');
set(handles.text7,'string',{' -----HELP TEXT-----'});
```

στις οποίες ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2.

Στην συνάρτηση (5), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
at1=handles.mat1;
[rmat1,cmat1]=size(mat1);
if rmat1 ~= cmat1
    errordlg('Matrix is not square');
    return;
elseif norm(mat1-mat1')>10^-6
    errordlg('Invalid matrix; is not para-Hermitian.');
```


end;

[result1,result2]=spcof(mat1);

με τις οποίες στην μεταβλητή *mat1* έχει καταχωρηθεί ο επιλεγμένος πίνακας, στη συνέχεια γίνεται έλεγχος εάν ο πίνακας *mat1* είναι τετραγωνικός και εάν είναι *para-Hermitian*. Έπειτα γίνεται ο υπολογισμός και τα αποτελέσματα καταχωρούνται στις μεταβλητές *result1* και *result2*. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας *enable* στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Matrix Pencil routines** ο χρήστης οδηγείται, όπως φαίνεται στο διάγραμμα, σε άλλα δυο υπό-μενού τα οποία υπολογίζουν ειδικές μορφές πινάκων ενός πολυωνυμικού *pencil* πίνακα. Η συνάρτηση που χρησιμοποιείται στο *m-file* για το συγκεκριμένο υπό-μενού είναι η εξής:

function penc_Callback(hObject, eventdata, handles)

Στην συνάρτηση αυτή, η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

set(handles.popupmenu1,'enable','on');

set(handles.popupmenu2,'enable','off');

set(handles.popupmenu3,'enable','off');

οι οποίες ορίζουν την ιδιότητα *enable* στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού.

Με την κλήση του υπό-μενού **Kronecker canonical form** ο χρήστης έχει τη δυνατότητα να υπολογίσει την *Kronecker canonical* μορφή ενός πολυωνυμικού *pencil* πίνακα που έχει ήδη επιλέξει χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου *p2*. Επίσης, γίνονται ενεργά τα σταθερά στοιχεία ελέγχου *push button refresh* και *save* και το σταθερό στοιχείο ελέγχου *popup*

menu 1. Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (handles) που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function penckro_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function refresh1_Callback(hObject, eventdata, handles)*
5. *function savepb_Callback(hObject, eventdata, handles)*
6. *function push141(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string',...
    'calculate','Callback','polyxgui('push141',gcbo,[],guidata(gcbo))',...
    'enable','on');
set(handles.text7,'string',{' -----HELP TEXT-----'});
```

στις οποίες ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2.

Στην συνάρτηση (6), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;
[rmat1,cmat1]=size(mat1);
if rmat1 ~= cmat1
    errordlg('Matrix is not square');
    return;
elseif all(all(deg(mat1,'ent')==0))
    errordlg('Invalid matrix; is numeric not pencil');
    return;
elseif any(any(deg(mat1,'ent')>1))
    errordlg('Invalid matrix; is polynomial not pencil');
    return;
end;
```

```
[result,Q,Z,dims]=pencan(mat1);
```

με τις οποίες στην μεταβλητή `mat1` έχει καταχωρηθεί ο επιλεγμένος πίνακας, στη συνέχεια γίνεται έλεγχος εάν ο πίνακας `mat1` είναι τετραγωνικός και εάν είναι `pencil` (υπάρχει διαχωρισμός μεταξύ αριθμητικού και πολυωνυμικού). Έπειτα γίνεται ο υπολογισμός και τα αποτελέσματα καταχωρούνται στις μεταβλητές `result` και `dims`. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας `enable` στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Clements form** ο χρήστης έχει τη δυνατότητα να υπολογίσει την Clements μορφή ενός πολυωνυμικού `pencil` πίνακα που έχει ήδη επιλέξει χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου `p2`. Επίσης, γίνονται ενεργά τα σταθερά στοιχεία ελέγχου `push button refresh` και `save` και το σταθερό στοιχείο ελέγχου `popup menu 1`. Οι συναρτήσεις που χρησιμοποιούνται στο `m-file` για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. `function penccle_Callback(hObject, eventdata, handles)`
2. `function popupmenu1_CreateFcn(hObject, eventdata, handles)`
3. `function popupmenu1_Callback(hObject, eventdata, handles)`
4. `function refresh1_Callback(hObject, eventdata, handles)`
5. `function savepb_Callback(hObject, eventdata, handles)`
6. `function push142(hObject, eventdata, handles)`

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p2,'string',...  
    'calculate','Callback','polyxgui('push141',gcbo,[],guidata(gcbo))',...  
    'enable','on');  
set(handles.text7,'string',{' -----HELP TEXT-----'});
```

στις οποίες ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) καθώς και διάφορες ιδιότητες στο προσωρινό στοιχείο ελέγχου push button p2. Στην συνάρτηση (6), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

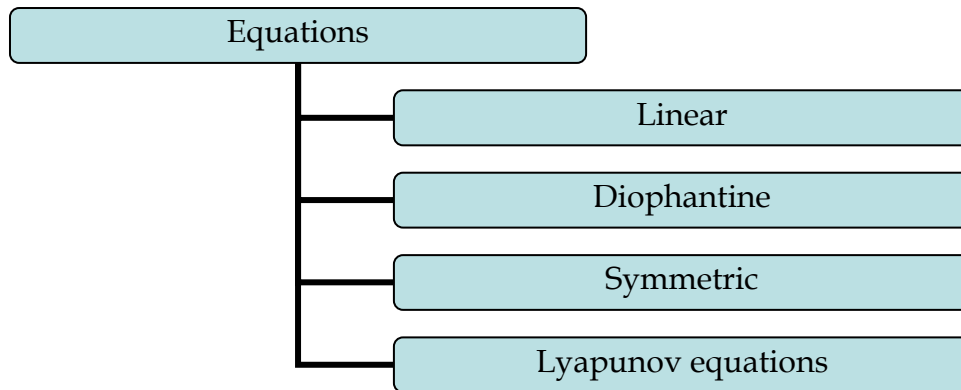
```
[rmat1,cmat1]=size(mat1);
if rmat1 ~= cmat1
    errordlg('Matrix is not square');
    return;
elseif norm(mat1-mat1')>10^-6
    errordlg('Invalid matrix; is not para-Hermitian. ');
    return;
elseif all(all(deg(mat1,'ent')==0))
    errordlg('Invalid matrix; is numeric not pencil');
    return;
elseif any(any(deg(mat1,'ent')>1))
    errordlg('Invalid matrix; is polynomial not pencil');
    return;
end;
result=clements(mat1);
```

με τις οποίες στην μεταβλητή mat1 έχει καταχωρηθεί ο επιλεγμένος πίνακας, στη συνέχεια γίνεται έλεγχος εάν ο πίνακας mat1 είναι τετραγωνικός, εάν είναι pencil (υπάρχει διαχωρισμός μεταξύ αριθμητικού και πολυωνυμικού) και τέλος εάν είναι para-Hermitian. Έπειτα γίνεται ο υπολογισμός και το αποτέλεσμα καταχωρείται στην μεταβλητή result. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας enable στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

3.2.8.4 Μενού equations

Το μενού equations έχει την παρακάτω δομή:



Μενού Equations

Το μενού **Equations** δίνει τη δυνατότητα στον χρήστη να επιλέξει ένα από τα τέσσερα υπό-μενού όπως φαίνεται στο διάγραμμα. Η συνάρτηση που χρησιμοποιείται στο m-file για το συγκεκριμένο μενού είναι η εξής:

```
function equations_Callback(hObject, eventdata, handles)
```

Στην συνάρτηση αυτή, η οποία είναι συνάρτηση επιστροφής του μενού, έχουν προστεθεί οι εντολές

```
delete;  
set(handles.text1,'string','Matrix A');  
set(handles.text2,'string','Matrix B');  
set(handles.text3,'string','Matrix C');  
set(handles.refresh1,'visible','off');  
set(handles.refresh2,'visible','on','enable','off');  
popinit;
```

στις οποίες ορίζονται οι ιδιότητες `visible` και `enable` στα σταθερά στοιχεία ελέγχου `refresh1-2`, ανάλογα με το αντικείμενο που θα χρησιμοποιηθεί. Επίσης ορίζονται κάποια κείμενα στα `text1-2-3` και γίνεται αρχικοποίηση με τις `delete` και `popinit` για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Linear** ο χρήστης έχει τη δυνατότητα να επιλύσει την γραμμική εξίσωση που επιθυμεί χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου popup menu p1 για την επιλογή της επιθυμητής εξίσωσης και το προσωρινό στοιχείο ελέγχου push button calculate για την εκτέλεση της επιλεγμένης εξίσωσης. Επίσης, γίνεται ενεργό το σταθερό στοιχείο ελέγχου push button refresh και τα σταθερά στοιχεία ελέγχου popup menu 1, popup menu 2 και popup menu 3. Οι εξισώσεις που μπορεί να επιλέξει ο χρήστης είναι:

- $A \cdot X = B$
- $X \cdot A = B$
- $A \cdot X \cdot B = C$

Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (handles) που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function linear_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function popupmenu2_CreateFcn(hObject, eventdata, handles)*
5. *function popupmenu2_Callback(hObject, eventdata, handles)*
6. *function popupmenu3_CreateFcn(hObject, eventdata, handles)*
7. *function popupmenu3_Callback(hObject, eventdata, handles)*
8. *function refresh2_Callback(hObject, eventdata, handles)*
9. *function popup20(hObject, eventdata, handles)*
10. *function push20(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p1,'string','choose equation | A*X=B | X*A=B | A*X*B=C',
    'enable','on',...
    'Callback','polyxgui(''popup20'',gcbo,[],guidata(gcbo))','visible','on');
set(handles.p2,'Callback','polyxgui(''push20'',gcbo,[],guidata(gcbo))');
```

στις οποίες ορίζονται διάφορες ιδιότητες στα προσωρινά στοιχεία ελέγχου popup menu p1 και push button p2.

Στην συνάρτηση (9), η οποία είναι συνάρτηση επιστροφής του popup menu p1, έχουν προστεθεί οι εντολές

```
pop=get(handles.p1,'value');
switch pop
  case 1
    set(handles.text7,'string',{'must choice equation'});
  case 2
    set(handles.popupmenu1,'enable','on');
    set(handles.popupmenu2,'enable','on');
    set(handles.popupmenu3,'enable','off');
    set(handles.p2,'enable','on');
    set(handles.text7,'string',{'----- HELP TEXT-----'});
  case 3
    set(handles.popupmenu1,'enable','on');
    set(handles.popupmenu2,'enable','on');
    set(handles.popupmenu3,'enable','off');
    set(handles.p2,'enable','on');
    set(handles.text7,'string',{'----- HELP TEXT-----'});
  case 4
    set(handles.popupmenu1,'enable','on');
    set(handles.popupmenu2,'enable','on');
    set(handles.popupmenu3,'enable','on');
    set(handles.p2,'enable','on');
    set(handles.text7,'string',{'----- HELP TEXT-----'});
End
```

με τις οποίες στην μεταβλητή pop καταχωρείται η επιλογή του χρήστη και με την εντολή switch ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) για κάθε πράξη που επιλέγεται. Επίσης ορίζεται η ιδιότητα enable στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού.

Στην συνάρτηση (10), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;
mat2=handles.mat2;
[rmat1,cmat1]=size(mat1);
[rmat2,cmat2]=size(mat2);
switch pop
case 1
    warndlg('must choose equation');
    return;
case 2
    if rmat1~=rmat2
        errordlg('Matrix A must have the same number of rows with matrix B');
        return;
    end;
    set(handles.text8,'string','X =');
    result=axb(mat1,mat2);
case 3
    if cmat1~=cmat2
        errordlg('Matrix A must have same number of columns with matrix B');
        return;
    end;
    set(handles.text8,'string','X =');
    result=xab(mat1,mat2);
case 4
    pop3=get(handles.popupmenu3,'value');
    if pop3==1
        warndlg('must choose matrix C');
        return;
    end
```



```
mat3=handles.mat3;  
[rmat3,cmat3] = size(mat3);  
if (rmat1~=rmat3) | (cmat2~=cmat3)  
    errordlg({'Matrix A must have the same number of rows with matrix C'});  
    return;  
end;  
set(handles.text8,'string','X =');  
result=axbc(mat1,mat2,mat3);  
end  
handles.result=result;
```

με τις οποίες στις μεταβλητές *mat1* και *mat2* έχουν καταχωρηθεί οι επιλεγμένοι πίνακες και εφόσον γίνει έλεγχος στις διαστάσεις των πινάκων, υπολογίζεται η λύση της εξίσωσης που επιλέχθηκε και το αποτέλεσμα καταχωρείται στην μεταβλητή *result*. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας *enable* στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5),(6),(7),(8) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Diophantine** ο χρήστης έχει τη δυνατότητα να επιλύσει την διοφαντική εξίσωση που επιθυμεί χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου *popup menu p1* για την επιλογή της επιθυμητής εξίσωσης και το προσωρινό στοιχείο ελέγχου *push button calculate* για την εκτέλεση της επιλεγμένης εξίσωσης. Επίσης, γίνεται ενεργό το σταθερό στοιχείο ελέγχου *push button refresh* και τα σταθερά στοιχεία ελέγχου *popup menu 1*, *popup menu 2* και *popup menu 3*. Οι εξισώσεις που μπορεί να επιλέξει ο χρήστης είναι:

- $A \cdot X + B \cdot Y = C$
- $X \cdot A + Y \cdot B = C$
- $A \cdot X + Y \cdot B = C$

- $A \cdot X + B \cdot Y = 0$
- $X \cdot A + Y \cdot B = 0$

Οι συναρτήσεις που χρησιμοποιούνται στο m-file για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function diophantine_Callback(hObject, eventdata, handles)*
2. *function popupmenu1_CreateFcn(hObject, eventdata, handles)*
3. *function popupmenu1_Callback(hObject, eventdata, handles)*
4. *function popupmenu2_CreateFcn(hObject, eventdata, handles)*
5. *function popupmenu2_Callback(hObject, eventdata, handles)*
6. *function popupmenu3_CreateFcn(hObject, eventdata, handles)*
7. *function popupmenu3_Callback(hObject, eventdata, handles)*
8. *function refresh2_Callback(hObject, eventdata, handles)*
9. *function popup1(hObject, eventdata, handles)*
10. *function push1(hObject, eventdata, handles)*

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p1,'string',
    'choose equation | A*X+B*Y=C | X*A+Y*B=C |
    A*X+Y*B=C | A*X+B*Y=0 | X*A+Y*B=0',
    'Callback',
    'polyxgui(''popup21'',gcbo,[],guidata(gcbo))','visible','on','enable','on');
set(handles.p2,'Callback','polyxgui(''push21'',gcbo,[],guidata(gcbo))');
```

στις οποίες ορίζονται διάφορες ιδιότητες στα προσωρινά στοιχεία ελέγχου popup menu p1 και push button p2.

Στην συνάρτηση (9), η οποία είναι συνάρτηση επιστροφής του popup menu p1, έχουν προστεθεί οι εντολές

```
pop=get(handles.p1,'value');
switch pop
    case 1
        set(handles.text7,'string',{'must choice equation'});
    case 2
```

```
set(handles.popupmenu1,'enable','on');  
set(handles.popupmenu2,'enable','on');  
set(handles.popupmenu3,'enable','on');  
set(handles.p2,'enable','on');  
set(handles.text7,'string',{'-----HELP TEXT-----'});
```

case 3

```
set(handles.popupmenu1,'enable','on');  
set(handles.popupmenu2,'enable','on');  
set(handles.popupmenu3,'enable','on');  
set(handles.p2,'enable','on');  
set(handles.text7,'string',{'-----HELP TEXT-----'});
```

case 4

```
set(handles.popupmenu1,'enable','on');  
set(handles.popupmenu2,'enable','on');  
set(handles.popupmenu3,'enable','on');  
set(handles.p2,'enable','on');  
set(handles.text7,'string',{'-----HELP TEXT-----'});
```

case 5

```
set(handles.popupmenu1,'enable','on');  
set(handles.popupmenu2,'enable','on');  
set(handles.popupmenu3,'enable','off');  
set(handles.p2,'enable','on');  
set(handles.text7,'string',{'-----HELP TEXT-----'});
```

case 6

```
set(handles.popupmenu1,'enable','on');  
set(handles.popupmenu2,'enable','on');  
set(handles.popupmenu3,'enable','off');  
set(handles.p2,'enable','on');  
set(handles.text7,'string',{'-----HELP TEXT-----'});
```

end

με τις οποίες στην μεταβλητή `pop` καταχωρείται η επιλογή του χρήστη και με την εντολή `switch` ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (`text7`) για κάθε πράξη που επιλέγεται. Επίσης ορίζεται η ιδιότητα `enable` στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού.

Στην συνάρτηση (10), η οποία είναι συνάρτηση επιστροφής του `pushbutton p2`, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```

mat1=handles.mat1;
mat2=handles.mat2;
[rmat1,cmat1] = size(mat1);
[rmat2,cmat2] = size(mat2);
pop=get(handles.p1,'value');
switch pop
    case 1
        warndlg('must choose equation');
        return;
    case 2
        pop3=get(handles.popupmenu3,'value');
        if pop3==1
            warndlg('must choose matrix C');
            return;
        end
        mat3=handles.mat3;
        [rmat3,cmat3] = size(mat3);
        if (rmat1~=rmat2) | (rmat1 ~= rmat3)
            errordlg({'Matrix A must have the same number of rows with matrix B'});
        end;
        [result1,result2]=axbyc(mat1,mat2,mat3);
    case 3
        pop3=get(handles.popupmenu3,'value');

```

```
if pop3==1
    warndlg('must choose matrix C');
    return;
end
mat3=handles.mat3;
[rmat3,cmat3] = size(mat3);
if (cmat1 ~= cmat2) | (cmat1 ~= cmat3)
    errordlg({'Matrix A must have same number of columns with matrix B'});
end;
[result1,result2]=xaybc(mat1,mat2,mat3);
case 4
    pop3=get(handles.popupmenu3,'value');
    if pop3==1
        warndlg('must choose matrix C');
        return;
    end
    mat3=handles.mat3;
    [rmat3,cmat3] = size(mat3);
    if (rmat1 ~= rmat3) | (cmat2 ~= cmat3)
        errordlg({'Matrix A must have the same number of rows with matrix C'});
    end;
    [result1,result2]=axybc(mat1,mat2,mat3);
case 5
    [result1,result2]=axby0(mat1,mat2);
case 6
    [result1,result2]=axby0(mat1,mat2);
end
handles.result1=result1;
handles.result2=result2;
```

με τις οποίες στις μεταβλητές mat1 και mat2 έχουν καταχωρηθεί οι επιλεγμένοι πίνακες και εφόσον γίνει έλεγχος στις διαστάσεις των πινάκων, υπολογίζεται η

λύση της εξίσωσης που επιλέχθηκε και τα αποτελέσματα καταχωρούνται στις μεταβλητές `result1` και `result2`. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας `enable` στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5),(6),(7),(8) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Symmetric** ο χρήστης έχει τη δυνατότητα να επιλύσει την συμμετρική εξίσωση που επιθυμεί χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου `popup menu p1` για την επιλογή της επιθυμητής εξίσωσης και το προσωρινό στοιχείο ελέγχου `push button calculate` για την εκτέλεση της επιλεγμένης εξίσωσης. Επίσης, γίνεται ενεργό το σταθερό στοιχείο ελέγχου `push button refresh` και τα σταθερά στοιχεία ελέγχου `popup menu 1` και `popup menu 2`. Οι εξισώσεις που μπορεί να επιλέξει ο χρήστης είναι:

- $A'X + X'A = B$
- $X'A' + A'X' = B$

Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (`handles`) που χρησιμοποιούνται στο `m-file` για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. `function symmetric_Callback(hObject, eventdata, handles)`
2. `function popupmenu1_CreateFcn(hObject, eventdata, handles)`
3. `function popupmenu1_Callback(hObject, eventdata, handles)`
4. `function popupmenu2_CreateFcn(hObject, eventdata, handles)`
5. `function popupmenu2_Callback(hObject, eventdata, handles)`
6. `function refresh2_Callback(hObject, eventdata, handles)`
7. `function popup22(hObject, eventdata, handles)`
8. `function push22(hObject, eventdata, handles)`

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p1,'string',
    'choose equation | A''*X+X''*A=B | X*A''+A*X''=B',...
```

```
'Callback',  
'polyxgui('popup22',gcbo,[],guidata(gcbo))','visible','on','enable','on');  
set(handles.p2,'Callback','polyxgui('push22',gcbo,[],guidata(gcbo))');
```

στις οποίες ορίζονται διάφορες ιδιότητες στα προσωρινά στοιχεία ελέγχου popup menu p1 και push button p2.

Στην συνάρτηση (7), η οποία είναι συνάρτηση επιστροφής του popup menu p1, έχουν προστεθεί οι εντολές

```
set(handles.popupmenu1,'enable','on');  
set(handles.popupmenu2,'enable','on');  
set(handles.p2,'enable','on');  
pop=get(handles.p1,'value');  
switch pop  
case 1  
    set(handles.text7,'string',{'must choice equation'});  
case 2  
    set(handles.text7,'string',{'-----HELP TEXT-----'});  
case 3  
    set(handles.text7,'string',{'-----HELP TEXT-----'});  
end
```

με τις οποίες στην μεταβλητή pop καταχωρείται η επιλογή του χρήστη και με την εντολή switch ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) για κάθε πράξη που επιλέγεται. Επίσης ορίζεται η ιδιότητα enable στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπόμενο.

Στην συνάρτηση (8), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;  
mat2=handles.mat2;  
[rmat1,cmat1] = size(mat1);  
[rmat2,cmat2] = size(mat2);
```

```
pop=get(handles.p1,'value');
switch pop
case 1
    warndlg('must choose equation');
    return;
case 2
    if (rmat1 ~= cmat1) | (rmat1 ~= rmat2)
        errordlg('Matrix is not square. ');
        return;
    elseif (rmat1 ~= cmat1)
        errordlg('Matrices of inconsistent dimensions. ');
        return;
    end;
    result=axxab(mat1,mat2);
case 3
    if (rmat1 ~= cmat1) | (rmat1 ~= rmat2)
        errordlg('Matrix is not square. ');
        return;
    elseif (rmat1 ~= cmat1)
        errordlg('Matrices of inconsistent dimensions. ');
        return;
    end;
    result=xaaxb(mat1,mat2);
end
handles.result=result;
```

με τις οποίες στις μεταβλητές `mat1` και `mat2` έχουν καταχωρηθεί οι επιλεγμένοι πίνακες και εφόσον γίνει έλεγχος εάν ο πρώτος πίνακας είναι τετραγωνικός και εάν συμφωνούν οι διαστάσεις των πινάκων, υπολογίζεται η λύση της εξίσωσης που επιλέχθηκε και το αποτέλεσμα καταχωρείται στην μεταβλητή `result`. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την

εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας `enable` στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5),(6) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

Με την κλήση του υπό-μενού **Lyapunov equations** ο χρήστης έχει τη δυνατότητα να επιλύσει την εξίσωση Lyapunov που επιθυμεί χρησιμοποιώντας το προσωρινό στοιχείο ελέγχου `popup menu p1` για την επιλογή της επιθυμητής εξίσωσης και το προσωρινό στοιχείο ελέγχου `push button calculate` για την εκτέλεση της επιλεγμένης εξίσωσης. Επίσης, γίνεται ενεργό το σταθερό στοιχείο ελέγχου `push button refresh` και τα σταθερά στοιχεία ελέγχου `popup menu 1`, `popup menu 2` και `popup menu 3`. Η εξίσωση που μπορεί να επιλέξει ο χρήστης είναι:

$$A * X + Y * B = C$$

Οι συναρτήσεις και τα αντίστοιχα στοιχεία ελέγχου (`handles`) που χρησιμοποιούνται στο `m-file` για το συγκεκριμένο υπό-μενού είναι οι εξής:

1. *function* `penclya_Callback(hObject, eventdata, handles)`
2. *function* `popupmenu1_CreateFcn(hObject, eventdata, handles)`
3. *function* `popupmenu1_Callback(hObject, eventdata, handles)`
4. *function* `popupmenu2_CreateFcn(hObject, eventdata, handles)`
5. *function* `popupmenu2_Callback(hObject, eventdata, handles)`
6. *function* `popupmenu3_CreateFcn(hObject, eventdata, handles)`
7. *function* `popupmenu3_Callback(hObject, eventdata, handles)`
8. *function* `refresh2_Callback(hObject, eventdata, handles)`
9. *function* `popup23(hObject, eventdata, handles)`
10. *function* `push23(hObject, eventdata, handles)`

Στην συνάρτηση (1), η οποία είναι συνάρτηση επιστροφής του υπό-μενού, έχουν προστεθεί οι εντολές

```
set(handles.p1,'string',  
'choose equation | A * X + Y * B = C',...  
'Callback',
```

```
'polyxgui('popup23',gcbo,[],guidata(gcbo))','visible','on','enable','on');
set(handles.p2,'Callback','polyxgui('push23',gcbo,[],guidata(gcbo)))');
```

στις οποίες ορίζονται διάφορες ιδιότητες στα προσωρινά στοιχεία ελέγχου popup menu p1 και push button p2.

Στην συνάρτηση (9), η οποία είναι συνάρτηση επιστροφής του popup menu p1, έχουν προστεθεί οι εντολές

```
pop=get(handles.p1,'value');
set(handles.text7,'horizontalAlignment','left');
switch pop
case 1
    set(handles.text7,'string',{' must choice equation'});
case 2
    set(handles.popupmenu1,'enable','on');
    set(handles.popupmenu2,'enable','on');
    set(handles.popupmenu3,'enable','on');
    set(handles.p2,'enable','on');
    set(handles.text7,'string',{'-----HELP TEXT-----'});
end
```

με τις οποίες στην μεταβλητή pop καταχωρείται η επιλογή του χρήστη και με την εντολή switch ορίζεται το κείμενο βοήθειας στο αντίστοιχο στοιχείο ελέγχου (text7) για κάθε πράξη που επιλέγεται. Επίσης ορίζεται η ιδιότητα enable στα σταθερά στοιχεία ελέγχου που είναι απαραίτητα για το συγκεκριμένο υπό-μενού.

Στην συνάρτηση (10), η οποία είναι συνάρτηση επιστροφής του pushbutton p2, έχουν προστεθεί οι εντολές που υλοποιούν το πρώτο στάδιο (επιλογή και έλεγχος πινάκων). Επίσης, έχουν προστεθεί οι εντολές

```
mat1=handles.mat1;
mat2=handles.mat2;
mat3=handles.mat3;
[rmat1,cmat1] = size(mat1);
[rmat2,cmat2] = size(mat2);
```

```
[rmat3,cmat3] = size(mat3);  
pop=get(handles.p1,'value');  
switch pop  
    case 1  
        warndlg('must choose equation');  
        return;  
    case 2  
        if rmat1 ~= cmat1  
            errordlg('Matrix is not square');  
            return;  
        elseif any(any(deg(mat1,'ent')>1))  
            errordlg('Invalid matrix; is polynomial not pencil');  
            return;  
        end;  
        if rmat2 ~= cmat2  
            errordlg('Matrix is not square');  
            return;  
        elseif any(any(deg(mat2,'ent')>1))  
            errordlg('Invalid matrix; is polynomial not pencil');  
            return;  
        end  
        if any(any(deg(mat3,'ent')>1))  
            errordlg('Invalid matrix; is polynomial not pencil');  
            return;  
        end  
        if rmat3 ~= rmat1 | cmat3 ~= cmat2  
            errordlg('3rd argument is not a matrix of correct dimensions');  
            return  
        end  
        [result1,result2]=plyap(mat1,mat2,mat3);  
end
```

```
handles.result1=result1;
```

```
handles.result2=result2;
```

με τις οποίες στις μεταβλητές `mat1` `mat2` και `mat3` έχουν καταχωρηθεί οι επιλεγμένοι πίνακες και εφόσον γίνει έλεγχος εάν οι δυο πρώτοι πίνακες είναι τετραγωνικοί και οι διαστάσεις του τρίτου είναι συμβατές αλλά και εάν είναι pencil πίνακες, υπολογίζεται η λύση της εξίσωσης που επιλέχθηκε και τα αποτελέσματα καταχωρούνται στις μεταβλητές `result1` και `result2`. Τέλος, έχουν προστεθεί οι εντολές που υλοποιούν το τρίτο στάδιο, δηλαδή, την εμφάνιση αποτελεσμάτων αλλά και τον ορισμό της ιδιότητας `enable` στα στοιχεία ελέγχου που είναι απαραίτητα για την συγκεκριμένη πράξη.

Οι συναρτήσεις (2),(3),(4),(5),(6),(7),(8) είναι συναρτήσεις για σταθερά στοιχεία ελέγχου, για τις οποίες έχουμε ήδη αναφερθεί.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] MATLAB Creating Graphical User Interfaces, The MATH WORKS Inc, 1999.
- [2] POLYX The polynomial toolbox for MATLAB, Polyx Ltd, 1999

ΠΑΡΑΡΤΗΜΑ Α

Οι συναρτήσεις που χρησιμοποιούμε στην εφαρμογή είναι οι εξής :

plus (+)

Σύνταξη $A + B$

Περιγραφή Η εντολή plus(+) επιστρέφει το άθροισμα των πολυωνυμικών πινάκων A και B .

minus(-)

Σύνταξη $A - B$

Περιγραφή Η εντολή minus(-) επιστρέφει την διαφορά των πολυωνυμικών πινάκων A και B .

mtimes(*)

Σύνταξη $A * B$

Περιγραφή Η εντολή mtimes(*) επιστρέφει το γινόμενο των πολυωνυμικών πινάκων A και B .

num

Σύνταξη $num(A)$

Περιγραφή Η εντολή num επιστρέφει τον αριθμητή ενός πολυωνυμικού πίνακα A .

den

Σύνταξη $den(A)$

Περιγραφή Η εντολή den επιστρέφει τον παρονομαστή ενός πολυωνυμικού πίνακα A .

transpose

Σύνταξη $transpose(A)$

Περιγραφή Η εντολή transpose επιστρέφει τον ανάστροφο ενός πολυωνυμικού πίνακα A .

conj

Σύνταξη $conj(A)$

Περιγραφή Η εντολή conj επιστρέφει τον συζυγή ενός

πολυωνυμικού πίνακα A , δηλαδή , τους συζυγείς συντελεστές των πολυωνόμων (στοιχείων) του πίνακα A .

adj

Σύνταξη $[adjA, detA] = adj(A)$

Περιγραφή Η εντολή adj μας τον προσαρτημένο πολυωνυμικό πίνακα και την ορίζουσα ενός τετραγωνικού πολυωνυμικού πίνακα A . Ο προσαρτημένος πίνακας ορίζεται ως

$$adj(A(s)) = (-1)^{i+j} det(A^{ij}(s))$$

όπου $A^{ij}(s)$ είναι ένας πίνακας από τον οποίο λείπει η i στήλη και η j γραμμή και χρησιμοποιείται στην εύρεση αντίστροφου πίνακα.

inv

Σύνταξη $inv(A)$

Περιγραφή Η εντολή inv επιστρέφει τον αντίστροφο ρητό πολυωνυμικό πίνακα ενός πολυωνυμικού τετραγωνικού και ομαλού (non-singular) πίνακα A .

Ο αντίστροφος ρητός πολυωνυμικός πίνακας δίνεται από την σχέση

$$A^{-1}(s) = \frac{1}{det(A(s))} adj(A(s)).$$

pinv

Σύνταξη $[Q, d] = pinv(A)$

Περιγραφή Η εντολή $pinv$ επιστρέφει έναν πολυωνυμικό πίνακα Q και ένα πολυώνυμο d έτσι ώστε ο ρητός πολυωνυμικός πίνακας $\frac{Q}{d}$ να είναι ο ψευδοαντίστροφος ενός μη τετραγωνικού πολυωνυμικού πίνακα A , ο οποίος πρέπει να έχει πλήρη τάξη (full rank) .

Οι πίνακες επαληθεύουν τις εξής σχέσεις :

$$A \cdot Q \cdot A = A \cdot d$$

$$Q \cdot A \cdot Q = Q \cdot d$$

null

Σύνταξη $N = null(A)$

Περιγραφή Η εντολή null επιστρέφει μια πολυωνυμική βάση (null space) για τον δεξιό μηδενικό χώρο ενός πολυωνυμικού πίνακα A έτσι ώστε να ισχύει $A(s) \cdot N(s) = 0$, όπου $N(s)$ η πολυωνυμική βάση .

minbasis

Σύνταξη $minbasis(A)$

Περιγραφή Η εντολή minbasis επιστρέφει μια ελάχιστη πολυωνυμική βάση (minimal basis) ενός πολυωνυμικού πίνακα A .

det

Σύνταξη $det(A)$

Περιγραφή Η εντολή det μας την ορίζουσα ενός τετραγωνικού πολυωνυμικού πίνακα A .

rank

Σύνταξη $rank(A)$

Περιγραφή Η εντολή rank επιστρέφει τον αριθμό των γραμμικά ανεξάρτητων γραμμών ή στηλών ενός πολυωνυμικού πίνακα A .

roots

Σύνταξη $roots(A)$
 $roots(A, 'all')$
 $roots(A, 'det')$
 $roots(A, 'eig')$

Περιγραφή Η εντολή roots, στην πρώτη περίπτωση, επιστρέφει τις πεπερασμένες ρίζες ενός πολυωνυμικού πίνακα A . Στην δεύτερη περίπτωση επιστρέφει όλες τις ρίζες, πεπερασμένες ή μη, ενός πολυωνυμικού πίνακα A . Στην

τρίτη περίπτωση επιστρέφει τις ρίζες της ορίζουσας ενός πολυωνυμικού πίνακα A . Τέλος, στην τέταρτη περίπτωση επιστρέφει ρίζες οι οποίες είναι οι γενικευμένες ιδιοτημές του Companion πίνακα ενός πολυωνυμικού πίνακα A .

deg

Σύνταξη $deg(A)$
 $deg(A, 'ent')$
 $deg(A, 'col')$
 $deg(A, 'row')$

Περιγραφή Η εντολή `deg`, στην πρώτη περίπτωση, επιστρέφει τον βαθμό ενός πολυωνυμικού πίνακα A . Στην δεύτερη περίπτωση επιστρέφει τον βαθμό του κάθε στοιχείου του πολυωνυμικού πίνακα A . Στην τρίτη και στην τέταρτη περίπτωση επιστρέφει, σε διάνυσμα, τον βαθμό της κάθε γραμμής ή στήλης αντίστοιχα, του πολυωνυμικού πίνακα A .

isfullrank

Σύνταξη $isfullrank(A)$

Περιγραφή Η εντολή `isfullrank` ελέγχει αν ένας πολυωνυμικός πίνακας A έχει πλήρη τάξη (full rank) , δηλαδή εάν η τάξη (rank) του πολυωνυμικού πίνακα A είναι ίση με την μικρότερη διάσταση του πίνακα .

issingular

Σύνταξη $issingular(A)$

Περιγραφή Η εντολή `issingular` ελέγχει αν ένας τετραγωνικός πολυωνυμικός πίνακας A είναι μη-ομαλός (singular) .

isunimod

Σύνταξη $isunimod(A)$

Περιγραφή Η εντολή `isunimod` ελέγχει αν ένας πολυωνυμικός πίνακας A είναι (unimodular) .

isproper

Σύνταξη $isproper(A)$

Περιγραφή Η εντολή `isproper` ελέγχει αν ένας ρητός πολυωνυμικός πίνακας είναι (proper) .

isprime

Σύνταξη $isprime(A)$
 $isprime([N,D])$

Περιγραφή Η εντολή `isprime` ελέγχει αν ένας πολυωνυμικός πίνακας είναι αριστερά πρώτος (left prime) ή δεξιά πρώτος (right prime) . Επίσης, στην δεύτερη περίπτωση, ελέγχει αν δυο πολυωνυμικοί πίνακες είναι πρώτοι μεταξύ τους .

isstable

Σύνταξη $isstable(A)$

Περιγραφή Η εντολή `isstable` ελέγχει αν ένας πολυωνυμικός πίνακας είναι ευσταθής (stable) ανάλογα με την μεταβλητή που έχει οριστεί (συνεχή ή διακριτή) .

hurwitz

Σύνταξη $H = hurwitz(P)$

Περιγραφή Η εντολή `hurwitz` επιστρέφει τον Hurwitz matrix ενός πολυωνυμικού πίνακα P . Ο πολυωνυμικός πίνακας P γράφεται με την βοήθεια των συντελεστών και ο Hurwitz matrix είναι ένας σταθερός πίνακας με blocks τους πίνακες συντελεστές του P , δηλαδή, εάν

$$P(s) = P_0 + P_1 \cdot s + P_2 \cdot s^2 + \dots + P_d \cdot s^d$$

τότε

$$H = \begin{bmatrix} P_{d-1} & P_{d-3} & P_{d-5} & \dots & \dots & \dots & \dots \\ P_d & P_{d-2} & P_{d-4} & \dots & \dots & \dots & \dots \\ 0 & P_{d-1} & P_{d-3} & P_{d-5} & \dots & \dots & \dots \\ 0 & P_d & P_{d-2} & P_{d-4} & \dots & \dots & \dots \\ 0 & 0 & P_{d-1} & P_{d-3} & P_{d-5} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{matrix} \text{block row1} \\ \text{block row2} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \text{block rowd} \end{matrix}$$

sylv

Σύνταξη $S = \text{sylv}(P)$

Περιγραφή Η εντολή `sylv` επιστρέφει τον Sylvester matrix του πολυωνυμικού πίνακα P . Ο πολυωνυμικός πίνακας P γράφεται με την βοήθεια των συντελεστών και ο Sylvester matrix είναι ένας σταθερός πίνακας με blocks τους πίνακες συντελεστές του P , δηλαδή, εάν

$$P(s) = P_0 + P_1 \cdot s + P_2 \cdot s^2 + \dots + P_d \cdot s^d$$

τότε

$$S = \begin{bmatrix} P_0 & P_1 & \dots & P_d & 0 & \dots & \dots & 0 \\ 0 & P_0 & P_1 & \dots & P_d & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & P_0 & P_1 & \dots & P_d \end{bmatrix} \begin{matrix} \text{block row1} \\ \text{block row2} \\ \vdots \\ \text{block rowd+1} \end{matrix}$$

compan

Σύνταξη $C = \text{compan}(P)$

Περιγραφή Η εντολή `compan` επιστρέφει τον Companion matrix του τετραγωνικού πολυωνυμικού πίνακα P . Ο πολυωνυμικός πίνακας P γράφεται με την βοήθεια των συντελεστών και ο Companion matrix είναι ένας σταθερός πίνακας με blocks τους πίνακες συντελεστές του P , δηλαδή, εάν

$$P(s) = P_0 + P_1 \cdot s + P_2 \cdot s^2 + \dots + P_d \cdot s^d$$

τότε

$$C = \begin{bmatrix} 0 & I & 0 & \dots & \dots \\ 0 & 0 & I & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & I \\ \frac{P_0}{-P_d} & \frac{P_1}{-P_d} & \dots & \dots & \frac{P_{d-1}}{-P_d} \end{bmatrix}$$

ldiv

Σύνταξη $[Q,R] = \text{ldiv}(N,D)$

Περιγραφή Η εντολή `ldiv` επιστρέφει το αριστερό πηλίκο (quotient)

και το υπόλοιπο (remainder) της ευκλείδειας διαίρεσης δυο πολυωνυμικών πινάκων N και D για τους οποίους ισχύει $N = D \cdot Q + R$.

Ο D πρέπει να είναι τετραγωνικός και ο N πρέπει να έχει ίδιο πλήθος γραμμών με τον D .

rdiv

Σύνταξη $[Q, R] = rdiv(N, D)$

Περιγραφή Η εντολή rdiv επιστρέφει το δεξιό πηλίκο (quotient) και το υπόλοιπο (remainder) της ευκλείδειας διαίρεσης δυο πολυωνυμικών πινάκων N και D για τους οποίους ισχύει $N = Q \cdot D + R$.

Ο D πρέπει να είναι τετραγωνικός και ο N πρέπει να έχει ίδιο αριθμό στηλών με τον D .

gld

Σύνταξη $G = gld(N_1, N_2, \dots, N_k)$

Περιγραφή Η εντολή gld επιστρέφει τον μέγιστο αριστερό κοινό διαιρετή (greatest left common divisor) των πολυωνυμικών πινάκων N_1, N_2, \dots, N_k οι οποίοι πρέπει να έχουν ίδιο πλήθος γραμμών.

grd

Σύνταξη $G = grd(N_1, N_2, \dots, N_k)$

Περιγραφή Η εντολή grd επιστρέφει τον μέγιστο δεξιό κοινό διαιρετή (greatest right common divisor) των πολυωνυμικών πινάκων N_1, N_2, \dots, N_k οι οποίοι πρέπει να έχουν ίδιο πλήθος στηλών.

llm

Σύνταξη $L = llm(N_1, N_2, \dots, N_k)$

Περιγραφή Η εντολή llm επιστρέφει το ελάχιστο αριστερό κοινό πολλαπλάσιο (least left common multiple) των

πολυωνυμικών πινάκων N_1, N_2, \dots, N_k οι οποίοι πρέπει να έχουν ίδιο πλήθος στηλών.

lrm

Σύνταξη $R = \text{lrm}(N_1, N_2, \dots, N_k)$

Περιγραφή Η εντολή `lrm` επιστρέφει το ελάχιστο δεξιό κοινό πολλαπλάσιο (least right common multiple) των πολυωνυμικών πινάκων N_1, N_2, \dots, N_k οι οποίοι πρέπει να έχουν ίδιο πλήθος γραμμών.

rowred

Σύνταξη $[D, rk, U, Ui] = \text{rowred}(A)$

Περιγραφή Η εντολή `rowred` επιστρέφει την row reduced form ενός πολυωνυμικού πίνακα A . Θα ισχύει $D = U \cdot A$, όπου U ένας unimodular πίνακας και D πολυωνυμικός πίνακας με γραμμές τοποθετημένες έτσι ώστε να οι βαθμοί (degree) των γραμμών να έχουν φθίνουσα σειρά.
 Το rk μας δείχνει τις μη μηδενικές γραμμές του D .
 Το Ui είναι ο αντίστροφος του U .

colred

Σύνταξη $[D, rk, U, Ui] = \text{colred}(A)$

Περιγραφή Η εντολή `colred` επιστρέφει την column reduced form ενός πολυωνυμικού πίνακα A . Θα ισχύει $D = A \cdot U$, όπου U ένας unimodular πίνακας και D πολυωνυμικός πίνακας με στήλες τοποθετημένες έτσι ώστε να οι βαθμοί (degree) των στηλών να έχουν φθίνουσα σειρά.
 Το rk μας δείχνει τις μη μηδενικές στήλες του D .
 Το Ui είναι ο αντίστροφος του U .

tri

Σύνταξη $[T, U, rowind] = \text{tri}(A, 'col')$

$[T, U, colind] = \text{tri}(A, 'row')$

Περιγραφή Η εντολή `tri`, στην πρώτη περίπτωση, επιστρέφει την

column staircase form ενός πολυωνυμικού πίνακα A . Θα ισχύει $T = A \cdot U$, όπου U ένας unimodular πίνακας.

Στην δεύτερη περίπτωση, η εντολή `tri`, επιστρέφει την row staircase form ενός πολυωνυμικού πίνακα A . Θα ισχύει $T = U \cdot A$, όπου U ένας unimodular πίνακας.

lu

Σύνταξη $[L,U] = lu(A)$

Περιγραφή Η εντολή `lu` υλοποιεί την παραγοντοποίηση ενός τετραγωνικού ομαλού (non-singular) πολυωνυμικού πίνακα A έτσι ώστε $A = L \cdot U$, όπου U ένας άνω τριγωνικός πολυωνυμικός πίνακας που τα στοιχεία της διαγώνιου έχουν μη μηδενικό πραγματικό μέρος και L ένας unimodular πίνακας.

hermite

Σύνταξη $[H,U,ind] = hermite(A)$

Περιγραφή Η εντολή `hermite` επιστρέφει την hermite column μορφή ενός πολυωνυμικού πίνακα A . Ο U είναι ένας unimodular πίνακας που ικανοποιεί την σχέση $A \cdot U = H$.

echelon

Σύνταξη $[E,U,ind] = echelon(A)$

Περιγραφή Η εντολή `echelon` επιστρέφει την echelon column μορφή ενός τετραγωνικού και column-reduced πολυωνυμικού πίνακα A . Ο U είναι ένας unimodular πίνακας που ικανοποιεί την σχέση $A \cdot U = E$.

smith

Σύνταξη $[S,U,V] = smith(A)$

Περιγραφή Η εντολή `smith` επιστρέφει την smith μορφή ενός πολυωνυμικού πίνακα A . Οι U και V είναι unimodular πίνακες που ικανοποιούν την σχέση $U \cdot A \cdot V = S$.

Η smith μορφή του πίνακα A είναι:

$$S = \begin{bmatrix} s_1 & 0 & \dots & \dots & 0 \\ 0 & s_2 & 0 & \dots & \dots \\ \dots & \dots & \ddots & \dots & \dots \\ 0 & \dots & 0 & s_k & 0 \\ 0 & \dots & \dots & \dots & 0 \end{bmatrix}$$

k είναι η τάξη (rank) του S , και τα πολυώνυμα s_1, s_2, \dots, s_k που βρίσκονται στην διαγώνιο έχουν την ιδιότητα το s_j να είναι διαιρέτης του s_{j+1} για $j=1,2,\dots,k-1$

axb

Σύνταξη $X = axb(A, B)$

Περιγραφή Η εντολή `axb` επιλύει το γραμμικό πολυωνυμικό σύστημα εξισώσεων $A \cdot X = B$, όπου A και B πολυωνυμικοί πίνακες.

xab

Σύνταξη $X = xab(A, B)$

Περιγραφή Η εντολή `xab` επιλύει το γραμμικό πολυωνυμικό σύστημα εξισώσεων $X \cdot A = B$, όπου A και B πολυωνυμικοί πίνακες.

axbc

Σύνταξη $X = axbc(A, B, C)$

Περιγραφή Η εντολή `axbc` επιλύει το γραμμικό πολυωνυμικό σύστημα εξισώσεων $A \cdot X \cdot B = C$, όπου A , B και C πολυωνυμικοί πίνακες.

axbyc

Σύνταξη $[X, Y] = axbyc(A, B, C)$
 $[X, Y] = axbyc(A, B, C, 'minx')$
 $[X, Y] = axbyc(A, B, C, 'miny')$

Περιγραφή Η εντολή `axbyc` επιλύει την διοφαντική πολυωνυμική εξίσωση $A \cdot X + B \cdot Y = C$, όπου A , B και C πολυωνυμικοί

πίνακες . Οι παράμετροι $minx$, $miny$ καθορίζουν ποιος άγνωστος θα έχει τον μικρότερο βαθμό .

axby0

Σύνταξη $[X,Y] = axby0(A,B,C)$
 $[X,Y] = axby0(A,B,C,'minx')$
 $[X,Y] = axby0(A,B,C,'miny')$

Περιγραφή Η εντολή $axby0$ επιλύει την διοφαντική πολυωνυμική εξίσωση $A \cdot X + B \cdot Y = 0$, όπου A , B και C πολυωνυμικοί πίνακες . Οι παράμετροι $minx$, $miny$ καθορίζουν ποιος άγνωστος θα έχει τον μικρότερο βαθμό .

xaabc

Σύνταξη $[X,Y] = xaabc(A,B,C)$
 $[X,Y] = xaabc(A,B,C,'minx')$
 $[X,Y] = xaabc(A,B,C,'miny')$

Περιγραφή Η εντολή $xaabc$ επιλύει την διοφαντική πολυωνυμική εξίσωση $X \cdot A + Y \cdot B = C$, όπου A , B και C πολυωνυμικοί πίνακες. Οι παράμετροι $minx$, $miny$ καθορίζουν ποιος άγνωστος θα έχει τον μικρότερο βαθμό.

xaabc0

Σύνταξη $[X,Y] = xaabc0(A,B,C)$
 $[X,Y] = xaabc0(A,B,C,'minx')$
 $[X,Y] = xaabc0(A,B,C,'miny')$

Περιγραφή Η εντολή $xaabc0$ επιλύει την διοφαντική πολυωνυμική εξίσωση $X \cdot A + Y \cdot B = 0$, όπου A , B και C πολυωνυμικοί πίνακες. Οι παράμετροι $minx$, $miny$ καθορίζουν ποιος άγνωστος θα έχει τον μικρότερο βαθμό.

axybc

Σύνταξη $[X,Y] = axybc(A,B,C)$

Περιγραφή Η εντολή $axybc$ επιλύει την διοφαντική πολυωνυμική εξίσωση $A \cdot X + Y \cdot B = C$, όπου A , B και C πολυωνυμικοί πίνακες.

axxab

Σύνταξη $X = axxab(A, B)$

Περιγραφή Η εντολή `axxab` επιλύει την αμφίπλευρη συμμετρική πολυωνυμική εξίσωση $A' \cdot X + X' \cdot A = B$, όπου A πολυωνυμικός πίνακας και B ένας para-Hermitian (εάν ο B είναι σε συνεχές χρόνο τότε πρέπει $B(s) = B^T(-s)$ ενώ σε διακριτό χρόνο πρέπει $B(z) = B^T(1/z)$) πολυωνυμικός πίνακας.

xaaxb

Σύνταξη $X = xaaxb(A, B)$

Περιγραφή Η εντολή `xaaxb` επιλύει την αμφίπλευρη συμμετρική πολυωνυμική εξίσωση $X \cdot A' + A \cdot X' = B$, όπου A πολυωνυμικός πίνακας και B ένας para-Hermitian (εάν ο B είναι σε συνεχές χρόνο τότε πρέπει $B(s) = B^T(-s)$ ενώ σε διακριτό χρόνο πρέπει $B(z) = B^T(1/z)$) πολυωνυμικός πίνακας.

axyab

Σύνταξη $[X, Y] = axyab(A, BL, BR)$

Περιγραφή Η εντολή `axyab` επιλύει την μη συμμετρική πολυωνυμική εξίσωση $A' \cdot X + Y' \cdot A = BL' + BR$, όπου A ευσταθής (stable) πολυωνυμικός πίνακας και BL, BR δυο πολυωνυμικοί πίνακες.

spf

Σύνταξη $[A, J] = spf(B)$

Περιγραφή Η εντολή `spf` υλοποιεί την φασματική παραγοντοποίηση (spectral factorization) ενός τετραγωνικού para-Hermitian (εάν ο B είναι σε συνεχές χρόνο τότε πρέπει $B(s) = B^T(-s)$ ενώ σε διακριτό χρόνο πρέπει

$B(z) = B^T\left(\frac{1}{z}\right)$) πολυωνυμικού πίνακα B.

$$B(s) = A^T(-s) \cdot J \cdot A(s)$$

$$B(z) = A^T\left(\frac{1}{z}\right) \cdot J \cdot A(z)$$

όπου A ένας ευσταθής (stable) τετραγωνικός πολυωνυμικός πίνακας και J ένας διαγώνιος σταθερός πίνακας .

sprcof

Σύνταξη $[A, J] = sprcof(B)$

Περιγραφή Η εντολή sprcof υλοποιεί την φασματική παραγοντοποίηση (spectral co-factorization) ενός τετραγωνικού para-Hermitian (εάν ο B είναι σε συνεχές χρόνο τότε πρέπει $B(s) = B^T(-s)$ ενώ σε διακριτό χρόνο πρέπει $B(z) = B^T\left(\frac{1}{z}\right)$) πολυωνυμικού πίνακα B.

$$B(s) = A(s) \cdot J \cdot A^T(-s)$$

$$B(z) = A(z) \cdot J \cdot A^T\left(\frac{1}{z}\right)$$

όπου A ένας ευσταθής (stable) τετραγωνικός πολυωνυμικός πίνακας και J ένας διαγώνιος σταθερός πίνακας .

pencan

Σύνταξη $[C, Q, Z, dims] = pencan(P)$

Περιγραφή Η εντολή pencan μετασχηματίζει έναν ομαλό τετραγωνικό πραγματικό pencil πίνακα $P(s)$ σε Kronecker κανονική μορφή .

$$C(s) = Q \cdot P(s) \cdot Z = \begin{bmatrix} a + sI & 0 \\ 0 & I + se \end{bmatrix}$$

όπου Q και Z σταθεροί ορθογώνιοι πίνακες , a σταθερός πίνακας με ιδιοτημές τις αρνητικές ρίζες του $P(s)$ και e ένας σταθερός πίνακας

clements

Σύνταξη $[C, u, p] = clements(P)$

Περιγραφή Η εντολή `clements` μετασχηματίζει έναν para-Hermitian (εάν ο P είναι σε συνεχές χρόνο τότε πρέπει $P(s) = P^T(-s)$ ενώ σε διακριτό χρόνο πρέπει $P(z) = P^T\left(\frac{1}{z}\right)$) πραγματικό pencil πίνακα $P(s) = sE + A$, όπου E ένας μη συμμετρικός και A ένας συμμετρικός πίνακας, σε Clements μορφή.

$$C(s) = uP(s)u^T = \begin{bmatrix} 0 & 0 & sE_1 + A_1 \\ 0 & A_2 & sE_3 + A_3 \\ -sE_1^T + A_1^T & -sE_3^T + A_3^T & sE_4 + A_4 \end{bmatrix}$$

όπου u ένας σταθερός ορθογώνιος πίνακας.

plyap

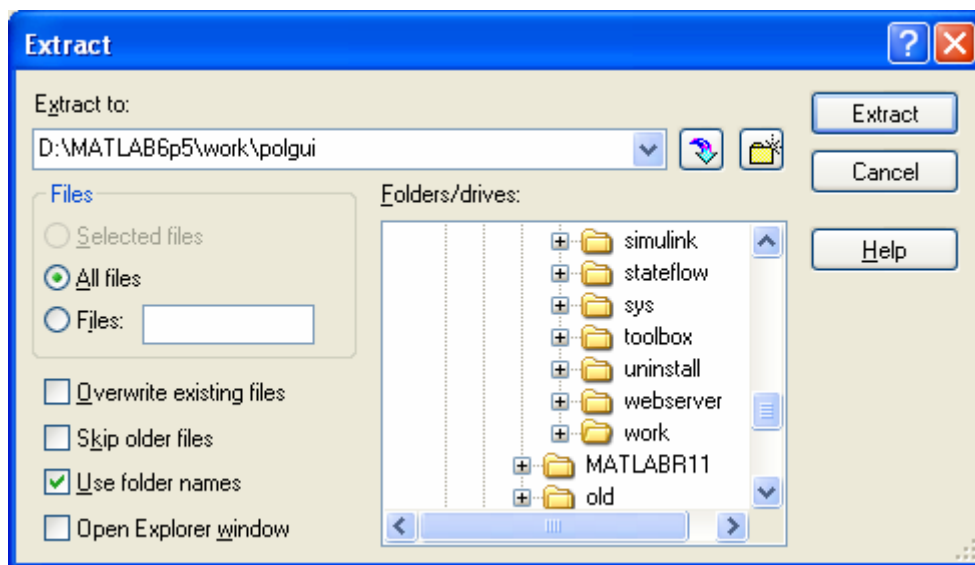
Σύνταξη $[X, Y] = plyap(A, B, C)$

Περιγραφή Η εντολή `plyap` επιλύει την εξίσωση $A(s) \cdot X + Y \cdot B(s) = C(s)$, όπου A και B δυο τετραγωνικοί pencil πίνακες και C pencil πίνακας με πλήθος γραμμών ίδιο με το πλήθος γραμμών του A και πλήθος στηλών ίδιο με το πλήθος στηλών του B . X και Y είναι σταθεροί πίνακες

ΠΑΡΑΡΤΗΜΑ Β

Οδηγίες εγκατάστασης εφαρμογής

Μέσα από το (polgui.zip) αντιγραφούμε (extract) τα αρχεία στον φάκελο εργασίας του MATLAB δημιουργώντας τον φάκελοmatlabr6p5\work\polgui



Η εφαρμογή καλείται με την εντολή polychgui

Το CD περιέχει:

- Το κείμενο της διπλωματικής σε μορφή MS-Word (ΒΑΡΣΑΜΗΣ_ΠΤΥΧΙΑΚΗ.doc), σε pdf αρχείο (Varsamis_ptixiaki.pdf) και σε zip αρχείο (varsamis_ptyxiaki.zip)
- Τα m-files και fig-files της εφαρμογής στον φάκελο \polgui αλλά και σε zip αρχείο (polgui.zip)
- Τα m-files και fig-files του παραδείγματος σε zip αρχείο (matrix_editor.zip)
- Το πακέτο POLYX σε zip αρχείο (polynomial.zip)
- Το MATLAB documentation for GUI (buildgui.pdf)