



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΤΟΠΟΓΡΑΦΙΑΣ ΚΑΙ ΓΕΩΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή/ Διπλωματική Εργασία

**Ανάπτυξη δικτυακών εφαρμογών σε περιβάλλον matlab :
Εφαρμογή στην ανάλυση και επεξεργασία σεισμολογικών
δεδομένων**

Συγγραφέας/είς

Σουπιωνάς Αλέξανδρος

AM: 509130940227

Επιβλέπων:

Γεώργιος Χλούπης

Αθήνα, Οκτώβριος 2022

Π.Α.Δ.Α – Σχολή Μηχανικών

Τμήμα Μηχανικών Τοπογραφίας και Γεωπληροφορικής

Ανάπτυξη δικτυακών εφαρμογών σε περιβάλλον matlab : Εφαρμογή στην ανάλυση και επεξεργασία
σεισμολογικών δεδομένων



UNIVERSITY OF WEST ATTICA

SCHOOL ENGINEERING

DEPARTMENT OF SURVEYING AND GEOINFORMATICS ENGINEERING

Diploma Thesis

**Development of web applications in matlab environment:
application in analysis and processing of seismological data**

Student name and surname:

Soupionas Alexandros

Registration Number: 509130940227

Supervisor name and surname:

Hloupis George

Athens, October 2022

Π.Α.Δ.Α – Σχολή Μηχανικών

Τμήμα Μηχανικών Τοπογραφίας και Γεωπληροφορικής

Ανάπτυξη δικτυακών εφαρμογών σε περιβάλλον matlab : Εφαρμογή στην ανάλυση και επεξεργασία σεισμολογικών δεδομένων



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΤΟΠΟΓΡΑΦΙΑΣ ΚΑΙ ΓΕΩΠΛΗΡΟΦΟΡΙΚΗΣ

**Ανάπτυξη δικτυακών εφαρμογών σε περιβάλλον matlab :
Εφαρμογή στην ανάλυση και επεξεργασία σεισμολογικών
δεδομένων**

Μέλη Εξεταστικής Επιτροπής συμπεριλαμβανομένου και του Εισηγητή

Η πτυχιακή/διπλωματική εργασία εξετάστηκε επιτυχώς από την κάτωθι Εξεταστική Επιτροπή:

Α/α	ΟΝΟΜΑ ΕΠΩΝΥΜΟ	ΒΑΘΜΙΔΑ/ΙΔΙΟΤΗΤΑ	ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ
1	ΒΑΣΙΛΗΣ ΑΝΔΡΙΤΣΑΝΟΣ	ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗ	
2	ΓΙΩΡΓΟΣ ΧΛΟΥΠΗΣ	ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ	
3	ΠΑΝΑΓΙΩΤΗΣ ΑΠΑΝΤΩΝΙΟΥ	ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ	

Π.Α.Δ.Α – Σχολή Μηχανικών

Τμήμα Μηχανικών Τοπογραφίας και Γεωπληροφορικής

Ανάπτυξη δικτυακών εφαρμογών σε περιβάλλον matlab : Εφαρμογή στην ανάλυση και επεξεργασία
σεισμολογικών δεδομένων

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος **Σουπιωνάς Αλέξανδρος** του **Ιωάννη**, με αριθμό μητρώου **509130940227** φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Τοπογραφίας και Γεωπληροφορικής, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών



*** Ονοματεπώνυμο /Ιδιότητα**

(Υπογραφή)

Ψηφιακή Υπογραφή Επιβλέποντα

*** Σε εξαιρετικές περιπτώσεις και μετά από αιτιολόγηση και έγκριση του επιβλέποντα, προβλέπεται χρονικός περιορισμός πρόσβασης (embargo) 6-12 μήνες. Στην περίπτωση αυτή θα πρέπει να υπογράψει ψηφιακά ο/η επιβλέπων/ουσα καθηγητής/τρια, για να γνωστοποιεί ότι είναι ενημερωμένος/η και συναινεί. Οι λόγοι χρονικού αποκλεισμού πρόσβασης περιγράφονται αναλυτικά στις πολιτικές του Ι.Α. (σελ. 6):**

[https://www.uniwa.gr/wp-](https://www.uniwa.gr/wp-content/uploads/2021/01/%CE%A0%CE%BF%CE%BB%CE%B9%CF%84%CE%B9%CE%BA%CE%B5%CC%81%CF%82_%CE%99%CE%B4%CF%81%CF%85%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CE%BF%CF%85%CC%81_%CE%91%CF%80%CE%BF%CE%B8%CE%B5%CF%84%CE%B7%CF%81%CE%B9%CC%81%CE%BF%CF%85_final.pdf)

[content/uploads/2021/01/%CE%A0%CE%BF%CE%BB%CE%B9%CF%84%CE%B9%CE%BA%CE%B5%CC%81%CF%82_%CE%99%CE%B4%CF%81%CF%85%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CE%BF%CF%85%CC%81_%CE%91%CF%80%CE%BF%CE%B8%CE%B5%CF%84%CE%B7%CF%81%CE%B9%CC%81%CE%BF%CF%85_final.pdf](https://www.uniwa.gr/wp-content/uploads/2021/01/%CE%A0%CE%BF%CE%BB%CE%B9%CF%84%CE%B9%CE%BA%CE%B5%CC%81%CF%82_%CE%99%CE%B4%CF%81%CF%85%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CE%BF%CF%85%CC%81_%CE%91%CF%80%CE%BF%CE%B8%CE%B5%CF%84%CE%B7%CF%81%CE%B9%CC%81%CE%BF%CF%85_final.pdf)).

Περιεχόμενα

ΠΕΡΙΛΗΨΗ	8
ABSTRACT	9
Κατάλογος πινάκων.....	10
Κατάλογος εικόνων	10
Λέξεις-κλειδιά.....	11
1. Εισαγωγή	12
1.1. Αντικείμενο.....	12
1.2. Δομή	12
1.3. Διαδικτυακές εφαρμογές	13
2. Θεωρητικό Μέρος.....	15
2.1. MATLAB Web Apps.....	15
2.2. MATLAB App Building.....	15
2.3. Χρήση λειτουργιών του App Designer	18
2.4. Design View και Code View	20
2.4.1. Διάταξη εφαρμογών στην προβολή σχεδίασης εφαρμογών (Design View).....	20
2.4.2. Διαχείριση κώδικα στην προβολή κώδικα εφαρμογής σχεδιαστή (Code View)	21
3. Πειραματικό Μέρος	25
3.1. Δημιουργία εφαρμογής υπολογισμού κανονικοποιημένου φάσματος ισχύος	25
3.1.1. Ανάλυση σεισμολογικών δεδομένων σε φυσικό χρόνο	25
3.1.2. Σχεδιασμός της εφαρμογής στο App Designer	29
3.1.3. Προγραμματισμός της εφαρμογής στο App Designer	31
3.1.4. Αποτελέσματα	36
3.1.5. Δημιουργία διαδικτυακής εφαρμογής.....	37
3.2. Δημιουργία εφαρμογής υπολογισμού κατανομής Gutenberg-Richter (frequency-magnitude distribution GR-FMD) και b-value	39
3.2.1. Νόμος του Gutenberg – Richter	39
3.2.2. Σχεδιασμός της εφαρμογής στο App Designer	41
3.2.3. Προγραμματισμός της εφαρμογής στο App Designer	42
3.2.4. Αποτελέσματα	45
3.2.5. Δημιουργία διαδικτυακής εφαρμογής.....	46

3.3. Υπολογισμός μεγέθους πληρότητας (magnitude of completeness - Mc) και b-value από σεισμολογικούς καταλόγους με χρήση Gutenberg-Richter FMD	46
3.3.1. Σχεδιασμός της εφαρμογής στο App Designer	47
3.3.2. Προγραμματισμός της εφαρμογής στο App Designer	48
3.3.3. Αποτελέσματα	50
4. Συμπεράσματα	52
Βιβλιογραφία	54

ΠΕΡΙΛΗΨΗ

Στην παρούσα εργασία δημιουργούμε διαδικτυακές εφαρμογές μέσω των εργαλείων του λογισμικού matlab. Τα βασικά εργαλεία που χρησιμοποιούμε είναι το MATLAB App Designer και το MATLAB Web App Compiler που μας βοηθάνε στο σχεδιασμό και τη δημιουργία διαδικτυακών εφαρμογών. Σαν είσοδο κάθε εφαρμογής που δημιουργούμε δέχεται αρχεία σεισμολογικών δεδομένων με σκοπό την ανάλυση, υπολογισμό και τη δημιουργία διαγραμμάτων. Οι εφαρμογές που δημιουργούμε αφορούν τον υπολογισμό του κανονικοποιημένου φάσματος φυσικού χρόνου και τον υπολογισμό της κατανομής Gutenberg-Richter (frequency-magnitude distribution GR-FMD) και b-value.

ABSTRACT

In this work we create online applications through the tools of Matlab software. The main tools we use are MATLAB App Designer and MATLAB Web App Compiler that help us design and create web applications. As an input to each application we create, it accepts seismological data files for the purpose of analysis, calculation and the creation of diagrams. The applications we create concern the calculation of the normalized spectrum of natural time and the calculation of the Gutenberg-Richter distribution (frequency-magnitude distribution GR-FMD) and b-value.

Κατάλογος πινάκων

Πίνακας 2.1. Τύποι Διεπαφών -Interface

Κατάλογος εικόνων

Εικόνα 2.1. Το Interface του App Designer

Εικόνα 2.2.Καρτέλα βιβλιοθήκης λειτουργιών – Component Library

Εικόνα 2.3.Βασικοί παράμετροι στο Code View

Εικόνα 2.4.Ιδιότητες παραμέτρων στο Code View

Εικόνα 2.5.Καρτέλα κώδικα περιήγησης – Code Browser

Εικόνα 2.6.Επιλογές στη βιβλιοθήκη περιήγησης – Component Library

Εικόνα 3.1.Επιλογές πεδίων στο Component Browser

Εικόνα 3.2.Οπτική προβολή της εφαρμογής στο App Designer

Εικόνα 3.3.Αυτόματη δημιουργία εντολών

Εικόνα 3.4.Έλεγχος έγκυρων τιμών

Εικόνα 3.5.Λανθασμένες τιμές σαν είσοδο μεταβλητών

Εικόνα 3.6.Μύνημα λάθους

Εικόνα 3.7.Κώδικας εισαγωγής αρχείου και μετατροπής δεδομένων

Εικόνα 3.8.Κώδικας υπολογισμού φάσματος ισχύος και εμφάνιση διαγράμματος

Εικόνα 3.9.Διάγραμμα κανονικοποιημένου φάσματος ισχύος συναρτήσει της φυσικής συχνότητας των δεδομένων

Εικόνα 3.10.Κεντρικό Μενού του εργαλείου Compiler

Εικόνα 3.11.Φόρτωμα εφαρμογής στο Compiler

Εικόνα 3.12.Δημοσίευση της εφαρμογής στο διαδίκτυο

Εικόνα 3.13. Νόμος του Gutenberg – Richter για $b=1$

Εικόνα 3.14. Νόμος του Gutenberg – Richter για διάφορες τιμές του b

Εικόνα 3.15.Συγκρινοντας το Roll of με το ιδανικό Gutenberg– Richter με $b=1$

Εικόνα 3.16.Οπτική προβολή της 2ης εφαρμογής στο App Designer

Π.Α.Δ.Α – Σχολή Μηχανικών, Τμήμα Μηχανικών Τοπογραφίας και Γεωπληροφορικής
Ανάπτυξη δικτυακών εφαρμογών σε περιβάλλον matlab : Εφαρμογή στην ανάλυση και επεξεργασία σεισμολογικών δεδομένων

Εικόνα 3.17.Κώδικας υπολογισμού κατανομής Gutenberg-Richter και b-value

Εικόνα 3.18.Κώδικας συνάρτησης Bval

Εικόνα 3.19.Διάγραμμα κατανομής Gutenberg-Richter και b-value

Εικόνα 3.20.Οπτική προβολή της 3ης εφαρμογής στο App Designer

Εικόνα 3.21.Κώδικας υπολογισμού μεγέθους πληρότητας και ένα b-value από σεισμολογικούς καταλόγους με χρήση Gutenberg-Richter FMD

Εικόνα 3.22.Διάγραμματα μεγέθους πληρότητας και b-value από σεισμολογικούς καταλόγους με χρήση Gutenberg-Richter FMD

Λέξεις-κλειδιά

Μεγέθους σεισμού: Η μέτρηση του μεγέθους ενός σεισμού βασίζεται σε μετρήσεις των πλατών των σεισμικών κυμάτων.

Διακομιστής : είναι υλικό ή / και λογισμικό που αναλαμβάνει την παροχή διάφορων υπηρεσιών, «εξυπηρετώντας» αιτήσεις άλλων προγραμμάτων, γνωστών ως πελάτες (clients) που μπορούν να τρέχουν στον ίδιο υπολογιστή ή σε σύνδεση μέσω δικτύου

Πεδίο : μεταβλητές των εφαρμογών που δημιουργούνται και δέχονται τιμές

Διεπαφή (Interface) : αποκαλείται το αντικείμενο εκείνο ή τμήμα αυτού (π.χ. κάποιο τμήμα λογισμικού, μια συσκευή κλπ.) που παρεμβάλλεται μεταξύ ενός αντικειμένου και του περιβάλλοντός του και διαμέσω του οποίου πραγματοποιείται κάθε επικοινωνία του αντικειμένου με το περιβάλλον του.

1. Εισαγωγή

1.1. Αντικείμενο

Η παρούσα διπλωματική εργασία έχει στόχο την ανάπτυξη δικτυακών εφαρμογών σε περιβάλλον matlab : Εφαρμογή στην ανάλυση και επεξεργασία σεισμολογικών δεδομένων . Συγκεκριμένα θα δημιουργήσουμε διαδικτυακές εφαρμογές χρησιμοποιώντας το λογισμικό matlab που θα αλληλεπιδρά με το χρήστη και θα εμφανίζει αποτελέσματα διαγράμματα και αναλύσεις .Συγκεκριμένα θα δημιουργήσουμε τρεις εφαρμογές χρησιμοποιώντας τα εργαλεία του λογισμικού matlab που αφορούν τον υπολογισμό του κανονικοποιημένου φάσματος φυσικού χρόνου και τον υπολογισμό της κατανομής Gutenberg-Richter (frequency-magnitude distribution GR-FMD) και παραμέτρου b-value καθώς και την εμφάνιση διαγραμμάτων τους.

1.2. Δομή

Η παρούσα εργασία αποτελείται από τρία κεφάλαια :

- Το θεωρητικό μέρος που περιέχει την ανάλυση των εργαλείων του matlab για το σχεδιασμό και την δημιουργία διαδικτυακών εφαρμογών
- Το πειραματικό μέρος που περιέχει αναλυτικά τη δημιουργία τριών διαδικτυακών εφαρμογών χρησιμοποιώντας τα εργαλεία του matlab, με θέμα τον υπολογισμό του κανονικοποιημένου φάσματος φυσικού χρόνου και τον υπολογισμό της κατανομής Gutenberg-Richter (frequency-magnitude distribution GR-FMD) και παραμέτρου b-value καθώς και την γραφική αναπαράσταση τους μέσω διαγραμμάτων..
- Τα συμπεράσματα της δημιουργίας διαδικτυακών εφαρμογών με τη χρήση των εργαλείων του matlab.

1.3. Διαδικτυακές εφαρμογές

Τι είναι :

Οι διαδικτυακές εφαρμογές είναι κατανεμημένες εφαρμογές που αναπτύσσονται στο διαδίκτυο και δημιουργούνται με τη χρήση adhoc και εμπειρικών διαδικασιών ανάπτυξης και σχεδιασμού εφαρμογών . Κατά τη δημιουργία διαδικτυακών εφαρμογών υπάρχει μεγάλη πιθανότητα αποτυχίας της λειτουργίας της εφαρμογής λόγω προβλημάτων που συναντάμε στο διαδικτυακό περιβάλλον .

Στην εποχή μας έχουν αναπτυχθεί διάφορες διαδικτυακές εφαρμογές σε πολλούς κλάδους όπως στο εμπόριο , στην εκπαίδευση , στη ψυχαγωγία κ.α. Οι ήδη υπάρχον βάσεις δεδομένων και εφαρμογές τροποποιούνται με σκοπό να εξυπηρετούν και τους χρήστες του διαδικτύου . Σύμφωνα με μελέτες , η πολυπλοκότητα των διαδικτυακών εφαρμογών αυξάνεται συνεχώς και η ανάπτυξη τους είναι αρκετά μικρή σε σχέση με τα απλά συστήματα λογισμικού , εξαιτίας των χρονικών απαιτήσεων της αγοράς . Έτσι οι διαδικτυακές εφαρμογές έχουν μεγαλύτερη πιθανότητα αποτυχίας από ότι οι κοινές εφαρμογές [16].

Οι εφαρμογές web καταφέρνουν να προσφέρουν τη λειτουργία τους σε οποιοδήποτε σημείο του κόσμου υπάρχει πρόσβαση στο διαδίκτυο , δίνοντας τους μεγάλη ευχρηστία . Ωστόσο, οι εφαρμογές web δεν χρησιμοποιούν τις κλασικές client-servers τεχνολογίες αλλά προγράμματα περιήγησης και web servers . Οι web εφαρμογές προτιμούνται περισσότερο από τις απλές εφαρμογές για τους εξής λόγους [16]:

- Η ευχρηστία των web εφαρμογών . Οι εφαρμογές web χρησιμοποιούν ένα κλασικό πρωτόκολλο HTTP που για να λειτουργήσει απαιτείται μόνο ένα πρόγραμμα περιήγησης απτή μεριά του πελάτη-χρήστη της εφαρμογής καθώς και η μεγάλη διαθεσιμότητα τους σε πολλές πλατφόρμες του διαδικτύου.
- Μικρότερο κόστος συντήρησης και εγκατάστασης . Δεν απαιτείται εγκατάσταση κάποιου λογισμικού ή εφαρμογής από τη πλευρά του πελάτη-χρήστη διότι τρέχουν μέσω του διαδικτύου , χρειάζεται μόνο η πρόσβαση στο διαδίκτυο. Η συντήρηση των διαδικτυακών εφαρμογών γίνεται με τη τροποποίηση του κώδικα των εφαρμογών αυτών που βρίσκεται σε ένα server και όχι σε κάθε μηχάνημα ξεχωριστά, που σημαίνει ότι μειώνεται σημαντικά ο χρόνος και το κόστος συντήρησης των διαδικτυακών εφαρμογών σε σχέση με τις απλές client/server εφαρμογές .

Οι web εφαρμογές κυμαίνονται από απλές στατικές ιστοσελίδες μέχρι και πιο σύνθετες εφαρμογές. Έτσι χωρίζονται σε κατηγορίες σύμφωνα με τα στοιχεία και την πολυπλοκότητα τους [16] :

1. Brochure Web εφαρμογές . Πρόκειται κυρίως για προσωπικές ιστοσελίδες που μπορεί να περιέχουν βιογραφικά και πληροφορίες ή ιστοσελίδες προϊόντων . Επικεντρώνονται στην ανάπτυξη περιεχομένου , στη διάταξη γραφικών και κειμένων .
2. Service orientted εφαρμογές . Πρόκειται για υπηρεσίες webmail ή online συστήματα επεξεργασίας κειμένου . Αυτές οι εφαρμογές παρέχουν κάποιες υπηρεσίες στο χρήστη και προγραμματιστικά εστιάζουν κυρίως στη δημιουργία της παρεχόμενης υπηρεσίας.
3. Data intensive εφαρμογές . Πρόκειται για online καταλόγους βιβλιοθηκών . Αυτές οι εφαρμογές παρέχουν ένα interface για περιήγηση και επεξεργασία μεγάλου όγκου δεδομένων στο χρήστη .
4. Information's system εφαρμογές. Πρόκειται κυρίως για ηλεκτρονικά καταστήματα ή ηλεκτρονικές τραπεζικές συναλλαγές. Αυτές οι εφαρμογές συνδυάζουν Service orientted (2) και Data intensive (3) εφαρμογές ενώ προγραμματιστικά εστιάζουν στη ροή των δεδομένων και τον έλεγχο ροής .

2. Θεωρητικό Μέρος

2.1. MATLAB Web Apps

Οι διαδικτυακές εφαρμογές είναι εφαρμογές του MATLAB που μπορούν να εκτελεστούν σε πρόγραμμα περιήγησης ιστού. Ο σχεδιασμός μιας διαδραστικής εφαρμογής MATLAB επιτυγχάνεται χρησιμοποιώντας το App Designer, ενώ η δημιουργία και δημοσίευση της διαδικτυακά γίνεται χρησιμοποιώντας το Web App Compiler . Το MATLAB Compiler™ διατίθεται με την έκδοση ανάπτυξης του διακομιστή εφαρμογών Web MATLAB.

Κάθε εφαρμογή Ιστού έχει μια μοναδική διεύθυνση URL και μπορεί να προσπελαστεί από ένα πρόγραμμα περιήγησης ιστού χρησιμοποιώντας πρωτόκολλα HTTP ή HTTPS. Ο διακομιστής έχει μια αρχική σελίδα με όλες τις διαθέσιμες φιλοξενούμενες εφαρμογές Ιστού. Μοιράζεστε εφαρμογές Ιστού κοινοποιώντας τη μοναδική διεύθυνση URL σε μια εφαρμογή Ιστού ή τη διεύθυνση URL στην αρχική σελίδα του διακομιστή.

Οι εφαρμογές Ιστού έχουν σχεδιαστεί για να εκτελούνται μόνο σε ένα αξιόπιστο περιβάλλον intranet και όχι στο ανοιχτό Διαδίκτυο. Μόνο οι εφαρμογές που έχουν σχεδιαστεί με χρήση του App Designer μπορούν να αναπτυχθούν ως εφαρμογές ιστού. Οι εφαρμογές Ιστού υποστηρίζονται σε Windows®, Linux® και macOS. [2]

2.2. MATLAB App Building

Χρησιμοποιούμε το λογισμικό MATLAB για τη δημιουργία διαδραστικών interfaces (διεπαφών) με το χρήστη που μπορούν να ενσωματωθούν σε διάφορα περιβάλλοντα .

Υπάρχουν δύο τύποι interface που μπορούμε να δημιουργήσουμε :

- Εφαρμογές – Με αυτόνομες διεπαφες που εκτελούνται βάση των επιλογών του χρήστη .
- Εργασίες Live Editor – Με διεπαφές που μπορούν να ενσωματωθούν σε ένα ζωντανό σενάριο και δημιουργούν κώδικα ως χρήστες εξερευνώντας διάφορους παραμέτρους . [1]

Πίνακας 2.1. Τύποι Διεπαφών – Interface[1]

Τύπος	Τρόποι Δημιουργίας	Τύποι αρχείων	Επιλογές δημοσιεύσεων
Εφαρμογές	Διαδραστικά χρησιμοποιώντας το App Designer	.mlapp	<ul style="list-style-type: none"> • Πακέτο ως ενιαίο αρχείο. • Ανάπτυξη ως διαδικτυακής εφαρμογής που μπορεί να εκτελεστεί σε μια ιστοσελίδα από ένα πρόγραμμα περιήγησης (απαιτείται το MATLAB Compiler • Δημιουργία αυτόνομης εφαρμογής (απαιτείται το MALAB Compiler) • Διανομή του κύριου αρχείου διεπαφής και υποστηριζόμενων αρχείων
	Προγραμματιστικά χρησιμοποιώντας λειτουργίες του MATLAB	.m	<ul style="list-style-type: none"> • Πακέτο ως ενιαίο αρχείο. • Δημιουργία αυτόνομης εφαρμογής (απαιτείται το MALAB Compiler) • Διανομή του κύριου αρχείου διεπαφής και υποστηριζόμενων αρχείων
Εργασίες Live Editor	Προγραμματιστικά χρησιμοποιώντας λειτουργίες του matlab.task.LiveTask	.m	<ul style="list-style-type: none"> • Διανομή του κύριου αρχείου διεπαφής και υποστηριζόμενων αρχείων

Για να δημιουργήσουμε μια αυτόνομη διεπαφή (Interface) μπορούμε να δημιουργήσουμε μια εφαρμογή με δύο τρόπους

- Διαδραστικά , χρησιμοποιώντας το App Designer
- Προγραμματιστικά με τη χρήση λειτουργιών και συναρτήσεων του MATLAB

Κάθε μια από αυτές τις μεθόδους προσφέρει διαφορετική ροή εργασίας και ένα διαφορετικό σύνολο λειτουργιών , Η Καλύτερη επιλογή εξαρτάται απ' τις απαιτήσεις του έργου μας .

Στη συγκεκριμένη εργασία επιλέξαμε να δημιουργήσουμε την εφαρμογή μας με το 1ο τρόπο χρησιμοποιώντας το App Designer για το σχεδιασμό της εφαρμογής και το MATLAB Compiler για την δημιουργία της και την δημοσίευση της διαδικτυακά.

- Χρήση του App Designer για τη δημιουργία διαδραστικών εφαρμογών

Το App Designer είναι ένα πλούσιο διαδραστικό περιβάλλον που εισήχθη στο MATLAB R2016a για πρώτη φορά και είναι προτεινόμενο για τη δημιουργία εφαρμογών σε περιβάλλον MATLAB . Περιλαμβάνει μια πλήρως ενσωματωμένη έκδοση του MATLAB Editor . Η Σχεδίαση διάταξης(Design view) και οι προβολή κώδικα (Code view) συνδέονται μεταξύ τους έτσι ώστε κάθε αλλαγή επηρεάζει η μία την άλλη .

Επιπλέον , διατίθεται ένα μεγάλο σύνολο διαδραστικών στοιχείων συμπεριλαμβανομένου της ημερομηνίας (date picker) και εικόνας (image components)κ.α.. Λειτουργίες όπως η διαχείριση διάταξης του πλέγματος σε χειροκίνητη ή αυτόματη (grid layout manager) και η ανίχνευση και προσαρμογή στο μέγεθος της οθόνης .

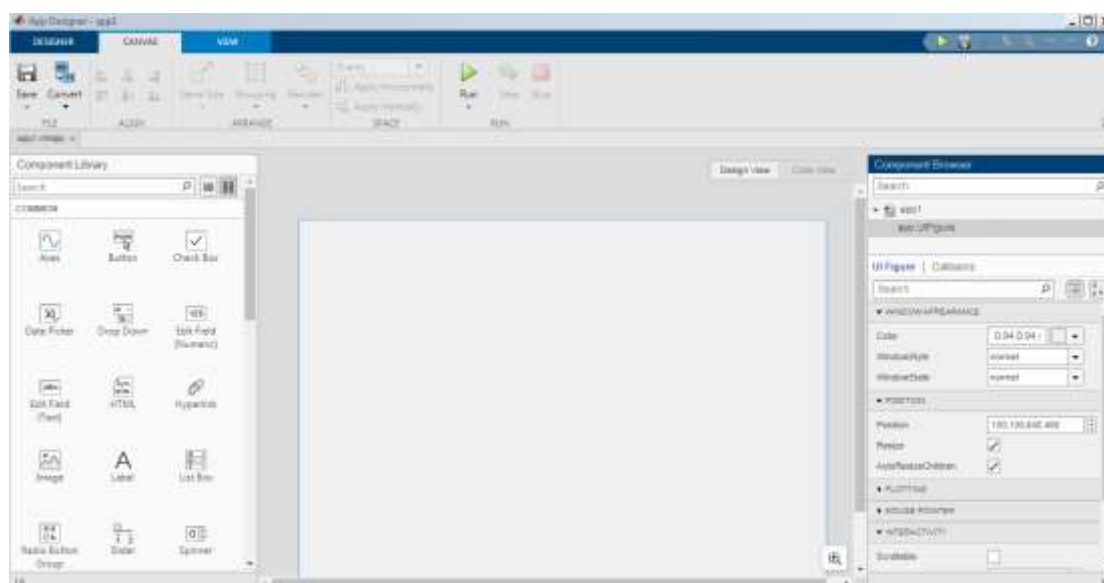
- Χρήση λειτουργιών και εντολών του MATLAB για τη δημιουργία εφαρμογών προγραμματιστικά

Μπορούμε να κωδικοποιήσουμε τη διάταξη και τη συμπεριφορά της εφαρμογής ,ας χρησιμοποιώντας μόνο λειτουργίες του MATLAB (functions) . Σε αυτό τον τρόπο αλλάζει η συμπεριφορά της λειτουργίας figure για εμφανίσεις διαγραμμάτων , εικόνων, πινάκων, γραφημάτων και χρησιμοποιούμε την εντολή UIfigure και αντίστοιχα σε ότι έχει να κάνει με την εμφάνιση UIAxes κλπ.

2.3. Χρήση λειτουργιών του App Designer

Για να μπούμε στο περιβάλλον του App Designer ανοίγουμε το MATLAB και γράφουμε “appdesigner” στο Command Line .

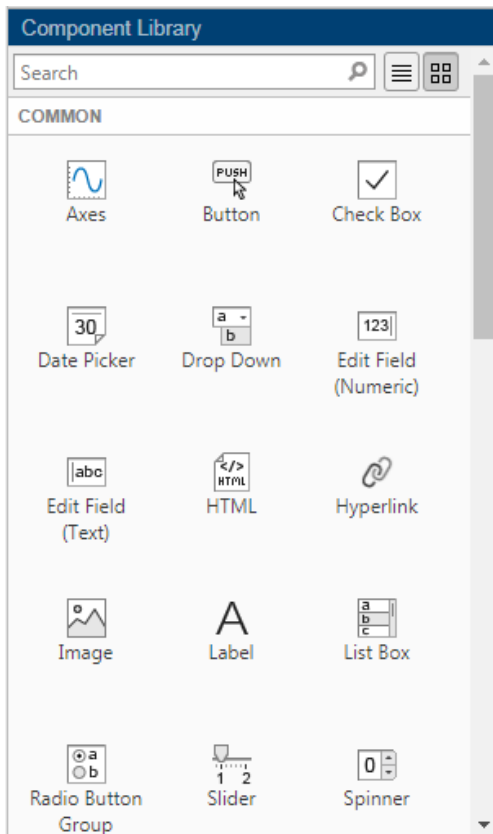
Ανοίγοντας το AppDesigner επιλέγουμε μια κενή εφαρμογή (Create new blank App) .Παρατηρώντας το περιβάλλον του App Designer , στο κέντρο βρίσκεται το Interface μας το οποίο θα βλέπει και ο κάθε χρήστης που θα τρέχει διαδικτυακά την εφαρμογή . Στα αριστερά βρίσκεται η βιβλιοθήκη λειτουργιών (Components Library) εκεί υπάρχουν διάφορες λειτουργίες που τοποθετούμε στο interface μας για να τρέχει ο χρήστης . Δεξιά υπάρχει μια βιβλιοθήκη περιήγησης (Component Browser) των λειτουργιών που μας δείχνει αναλυτικές επιλογές που μπορούμε να ρυθμίσουμε για κάθε λειτουργία .



Εικόνα 2.1. Το Interface του App Designer

Οι λειτουργίες είναι drag and drop δηλαδή τραβάμε και αφήνουμε στο interface της εφαρμογής κάθε λειτουργία που θέλουμε . Το App Designer μας βοηθάει σημαντικά στη στοίχιση των λειτουργιών που τοποθετούμε στο interface ώστε να φαίνονται σωστά η διάταξη .

Αναλυτικότερα κάποιες λειτουργίες (Component Library) που μας δίνει το App Designer είναι :



Εικόνα 2.2.Καρτέλα βιβλιοθήκης λειτουργιών – Component Library

- **Axes** : Εμφάνιση γραφικής οπτικοποίησης δεδομένων , δημιουργείται ένα διάγραμμα μέσα στο οποίο όταν πατάει ο χρήστης μια λειτουργία μπορεί αν εμφανίζεται ένα διάγραμμα διαφόρων δεδομένων
- **Button** : Εκτέλεση εντολής όταν πατηθεί το κουμπί . Είναι η βασική λειτουργία που χρησιμοποιούμε για την εκτέλεση οποιαδήποτε λειτουργίας όπως να φορτώσουμε δεδομένα , να εμφανίσουμε μια γραφική παράσταση κ.α.
 - **Check Box** : Επιλογή κατάστασης δυαδικής επιλογής . Είναι η λειτουργία που χρησιμοποιείται δίνοντας τη δυνατότητα στο χρήστη να επιλέξει αν θα χρησιμοποιεί ή θα βλέπει κάποια αποτελέσματα αναλόγως τη λειτουργία
 - **Date Picker** : Επιλογή και εμφάνιση ημερομηνίας σε συγκεκριμένη μορφή , δίνει τη δυνατότητα στο χρήστη να επιλέξει από ένα ημερολόγιο συγκεκριμένη ημερομηνία
 - **Drop Down** : Επιλογή κάποιας επιλογής απτη συγκεκριμένη λίστα , τις επιλογές της λίστας τις βάζουμε εμείς και ο χρήστης απλώς επιλέγει μία από αυτές
 - **Edit Field (Numeric)** :Εισαγωγή αριθμητικού

δεδομένου

- **Edit Field (Text)** : Εισαγωγή δεδομένου κειμένου
- **HTML** : Εμφάνιση σήμανσης ή ενσωματωμένου αρχείου τύπου HTML
- **Hyperlink** : Άνοιγμα ιστοσελίδας ή εκτέλεση κώδικα MATLAB
- **Image**: Εμφάνιση εικόνας
- **Label**: Ετικέτα, χρησιμοποιείται για ετικέτες των λειτουργιών καθώς και για να γράψουμε διάφορα σχόλια που πρέπει να είναι γνωστά στ χρήστη της εφαρμογής
- **List Box**: Επιλογή μιας ή περισσότερων επιλογών απτη λίστα , τις επιλογές της λίστας τις βάζουμε εμείς και ο χρήστης απλώς επιλέγει μία από αυτές.
- **Radio Button Group** : Επιλογή μιας επιλογής των κουμπιών επιλογής .
- **Slider**: Προσαρμογή τιμής ενός συγκεκριμένου εύρους
- **Spinner**: Εισαγωγή αριθμητικών δεδομένων και προσαρμογή των τιμών με κουμπιά αύξησης ή μείωσης
- **State Button** : Εναλλαγή μεταξύ καταστάσεων
- **Table** :Εμφάνιση πίνακα δεδομένων
- **Text Area** :Εισαγωγή κειμένου πολλαπλών γραμμών

- **Toggle Button Group** : Επιλογή μονής επιλογής από την ομάδα κουμπιών εναλλαγής (toggle buttons)
- **Tree** : Εμφάνιση ιεραρχικής λίστας στοιχείων
- **Tree(Check box)** : Εμφάνιση ιεραρχικής λίστας πλαισίου στοιχείων

Επιπλέον υπάρχουν και κάποιες άλλες λειτουργίες με διάφορα όργανα και διακόπτες που μπορούμε να χρησιμοποιήσουμε καθώς και κάποιες εργαλεία σχημάτων .

Κάθε λειτουργία τοποθετείται και εμφανίζεται στο κέντρο του Interface από εκεί καλείται ή εκτελείται όταν τρέχουμε την εφαρμογή.

2.4. Design View και Code View

Στο κεντρικό παράθυρο του App Designer βλέπουμε το Interface της εφαρμογής μας, υπάρχει μια επιλογή 2 κατηγοριών πάνω δεξιά (Design View και Code View) .Συγκεκριμένα στο Design View βλέπουμε οπτικά το περιβάλλον που θα βλέπει και ο χρήστης όταν τρέχει την εφαρμογή , δηλαδή κουμπιά και άλλες λειτουργίες που εκτελούν κάποιες διαδικασίες. Ενώ στο Code View βλέπουμε το κώδικα που εκτελεί το App Designer , κάθε φορά που αλλάζουμε κάτι προσθέτοντας μια λειτουργία στο Design view επηρεάζεται και το Code View δημιουργώντας αυτόματα κώδικα για κάθε λειτουργία που προσθέτουμε . Προγραμματιστικά όμως πρέπει να γράψουμε στο Code view τι θα κάνει η κάθε λειτουργία χρησιμοποιώντας εντολές του MATLAB.

2.4.1. Διάταξη εφαρμογών στην προβολή σχεδίασης εφαρμογών (Design View)

Το Design View στο App Designer παρέχει ένα πλούσιο σύνολο εργαλείων διάταξης για το σχεδιασμό σύγχρονων εφαρμογών με επαγγελματική εμφάνιση. Παρέχει επίσης μια εκτενή βιβλιοθήκη στοιχείων διεπαφής χρήστη, ώστε να μπορούμε να δημιουργήσουμε διάφορες διαδραστικές λειτουργίες. Οποιοσδήποτε αλλαγές κάνουμε στην Προβολή σχεδίασης (Design View) αντικατοπτρίζονται αυτόματα στην προβολή κώδικα. Έτσι, μπορούμε να διαμορφώσουμε πολλές πτυχές της εφαρμογής σας χωρίς να γράψουμε κώδικα.

Για να προσθέσουμε ένα στοιχείο στην εφαρμογή μας, χρησιμοποιούμε μία από τις παρακάτω μεθόδους:

1. Σύρετε ένα στοιχείο από τη Βιβλιοθήκη στοιχείων (Component Library) και αποθέστε το στον καμβά .
2. Κάντε κλικ σε ένα στοιχείο στη Βιβλιοθήκη στοιχείων και, στη συνέχεια, μετακινήστε τον κέρσορα πάνω από τον καμβά. Ο κέρσορας αλλάζει σε

σταυρόνημα. Κάντε κλικ στο ποντίκι σας για να προσθέσετε το στοιχείο στον καμβά στο προεπιλεγμένο του μέγεθος ή κάντε κλικ και σύρετε για να το μέγεθος του στοιχείου καθώς το προσθέτετε. Ορισμένα στοιχεία μπορούν να προστεθούν μόνο στο προεπιλεγμένο μέγεθος.

2.4.2. Διαχείριση κώδικα στην προβολή κώδικα εφαρμογής σχεδιαστή (Code View)

Το Code View παρέχει τις περισσότερες από τις ίδιες δυνατότητες προγραμματισμού που παρέχει το MATLAB Editor. Παρέχει επίσης ένα πλούσιο σύνολο λειτουργιών που μας βοηθούν να πλοηγηθούμε στον κώδικά σας και να αποφύγουμε πολλές κουραστικές εργασίες. Για παράδειγμα, μπορούμε να αναζητήσουμε μια επιστροφή κλήσης (Callback) πληκτρολογώντας μέρος του ονόματός της σε μια γραμμή αναζήτησης. Κάνοντας κλικ σε ένα αποτέλεσμα αναζήτησης, το πρόγραμμα επεξεργασίας μετακινείται στον ορισμό αυτής της επανάκλησης(Callback). Και αν αλλάξουμε το όνομα μιας επιστροφής κλήσης, το App Designer ενημερώνει αυτόματα όλες τις αναφορές σε αυτό στον κώδικά μας.

Παρακάτω φαίνεται τις παραμέτρους που έχουν δημιουργηθεί αυτόματα στο Code View απτή στιγμή που τις έχουμε προσθέσει στο Design View . Για κάθε παράμετρο από αυτές που θέλουμε να χρησιμοποιήσουμε αναφερόμαστε ως “app.ονομασία παραμέτρου “ για να τη δηλώσουμε σε κάποια λειτουργία .

```
classdef appexample2 < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure          matlab.ui.Figure
        X2EditField        matlab.ui.control.NumericEditField
        X2EditFieldLabel   matlab.ui.control.Label
        X1EditField        matlab.ui.control.NumericEditField
        X1EditFieldLabel   matlab.ui.control.Label
        Label_3            matlab.ui.control.Label
        Label_2            matlab.ui.control.Label
        Label              matlab.ui.control.Label
        loadButton         matlab.ui.control.Button
    end
```

Εικόνα 2.3.Βασικοί παράμετροι στο Code View

Στη συνέχεια βλέπουμε τις θέσεις των παραμέτρων που έχουν δημιουργηθεί αυτόματα καθώς και τις ιδιότητες τους .

Π.Α.Δ.Α – Σχολή Μηχανικών, Τμήμα Μηχανικών Τοπογραφίας και Γεωπληροφορικής
Ανάπτυξη δικτυακών εφαρμογών σε περιβάλλον matlab : Εφαρμογή στην ανάλυση και επεξεργασία σεισμολογικών δεδομένων

```

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 631 656];
app.UIFigure.Name = 'MATLAB App';

% Create loadButton
app.loadButton = uibutton(app.UIFigure, 'push');
app.loadButton.ButtonPushedFcn = createCallbackFcn(app, @loadButtonPushed, t
app.loadButton.Position = [151 339 100 22];
app.loadButton.Text = {'load'; ''};

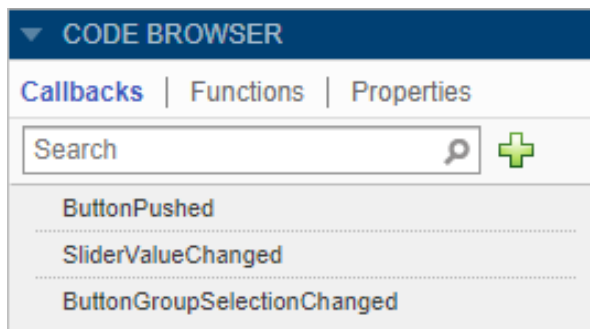
% Create Label
app.Label = uilabel(app.UIFigure);
app.Label.Position = [308 511 140 22];
app.Label.Text = 'X1,X2 : Εύρος δείγματος ';

% Create Label_2
app.Label_2 = uilabel(app.UIFigure);
app.Label_2.Position = [317 482 118 22];
app.Label_2.Text = ' X1:το πρώτο δείγμα ';

```

Εικόνα 2.4.Ιδιότητες παραμέτρων στο Code View

- Code Browser (Callbacks, Functions, and Properties tabs)



Εικόνα 2.5.Καρτέλα κώδικα περιήγησης – Code Browser

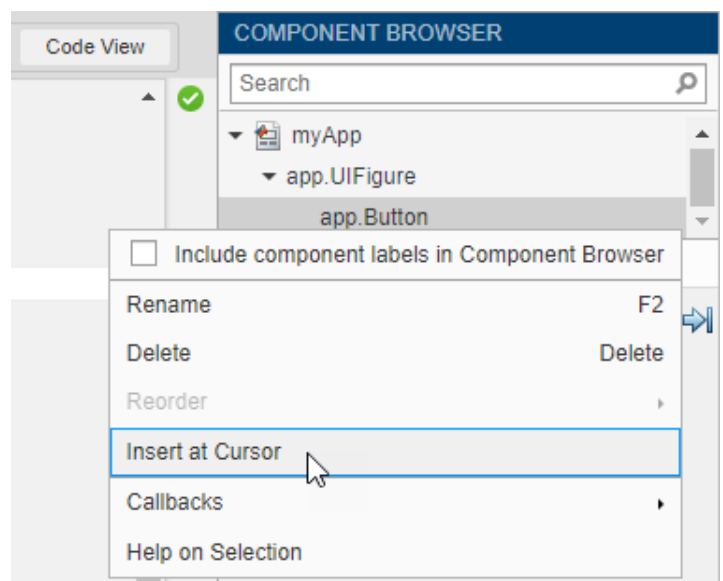
Χρησιμοποιούμε αυτές τις καρτέλες για να προσθέσουμε, να διαγράψουμε ή να μετονομάσουμε οποιαδήποτε από τις επιστροφές κλήσεων (Callbacks), τις βοηθητικές λειτουργίες ή τις προσαρμοσμένες ιδιότητες στην εφαρμογή μας. Κάνοντας κλικ σε ένα στοιχείο στην καρτέλα Επιστροφές κλήσεων ή Λειτουργίες, το πρόγραμμα επεξεργασίας μετακινείται στην αντίστοιχη ενότητα του κώδικά σας. Αναδιατάζετε τη σειρά των επανακλήσεων επιλέγοντας την επιστροφή κλήσης που θέλουμε να μετακινήσουμε και, στη συνέχεια, σύρουμε και αποθέτουμε την επανάκληση στη νέα της θέση στη λίστα. Αυτό επανατοποθετεί επίσης την επιστροφή κλήσης στο πρόγραμμα επεξεργασίας.

Το Callback είναι μια πολύ χρήσιμη επιλογή που την ενεργοποιούμε από Code Browser προσθέτοντας τη με το εικονίδιο ή από Component Browser, πατώντας δεξιά κλικ σε κάποιο στοιχείο και επιλέγοντας Callbacks.

Είναι η εντολή που μας επιτρέπει κάθε φορά που ενεργοποιεί ο χρήστης μια λειτουργία ,όπως να πατήσει ένα κουμπί, να εκτελείται μια συγκεκριμένη διαδικασία που θα του

δηλώσουμε .

Δεξιά βρίσκεται η βιβλιοθήκη περιήγησης (Component Browser) των λειτουργιών που μας δείχνει αναλυτικές επιλογές που μπορούμε να ρυθμίσουμε για κάθε λειτουργία . Υπάρχουν διάφορες επιλογές που μπορούμε να επιλέξουμε για κάθε λειτουργία όπως η διάταξη στοίχιση , ετικέτες, χρώματα , θέση , callabacks, ιδιότητες .



Εικόνα 2.6.Επιλογές στη βιβλιοθήκη περιήγησης – Component Library

Στη δεύτερη καρτέλα του Code Browser είναι τα functions(συναρτήσεις) λειτουργούν όπως και στο MATLAB . Κάθε φορά που προσθέτουμε μια συνάρτηση δημιουργείται στο Code View πάνω από όλες τις παραμέτρους έτσι ώστε όταν τη καλούμε αργότερα σε κάποια λειτουργία να μπορεί να τη καλέσει .

Τα Properties είναι διάφορες ιδιότητες των παραμέτρων που μπορούμε να επιλέξουμε όπως να θέσουμε σε μια μεταβλητή εξαρχής μια γνωστή διότι πρέπει να είναι γνωστή για να δουλέψει σωστά η εφαρμογή.

Το μειονέκτημα του App Designer είναι ότι δεν αποθηκεύει μεταβλητές τις οποίες τρέχει , δηλαδή κάθε φορά που θα τρέξει μια λειτουργία δεν αποθηκεύεται το αποτέλεσμα του , έτσι αν θέλουμε αργότερα να χρησιμοποιήσουμε αυτό το αποτέλεσμα δεν μπορούμε . Δεν υπάρχει workspace όπως στο Matlab , άρα χρησιμοποιούμε τα functions για να καλούμε και να έχουμε αποθηκευμένες μεταβλητές που θέλουμε να χρησιμοποιήσουμε.

3. Πειραματικό Μέρος

3.1. Δημιουργία εφαρμογής υπολογισμού κανονικοποιημένου φάσματος ισχύος

Στην εργασία μας θέλουμε να δημιουργήσουμε ένα αρχείο .mlapp στο AppDesigner , το οποίο δέχεται ένα αρχείο .txt από το χρήστη που περιέχει μεγέθη σεισμολογικών δεδομένων σε ριχτερ. Το αποτέλεσμα της εφαρμογής θα είναι ένα διάγραμμα κανονικοποιημένου φάσματος ισχύος συναρτήσει της φυσικής συχνότητας των μετρήσεων. Το αρχείο αυτό το προσαρμόζουμε στη μορφή ενός μονόστηλου πίνακα που περιέχει μόνο τα μεγέθη των σεισμών σε τοπικό μέγεθος. Στη συγκεκριμένη εργασία δημιουργήσαμε εμείς ένα αρχείο τύπου text παίρνοντας σεισμολογικά δεδομένα απτην ιστοσελίδα του Εθνικού Αστεροσκοπείου Αθηνών (<https://bbnet.gein.noa.gr/HL/>) και το χρησιμοποιήσαμε ως παράδειγμα για την εφαρμογή μας.[5]

3.1.1. Ανάλυση σεισμολογικών δεδομένων σε φυσικό χρόνο

Το 2001 προτάθηκε από τους καθηγητές Βαρώτσο Π., Σαρλή Ν. και Σκορδά Ε. η έννοια του φυσικού χρόνου χ [13]. Ειδικότερα, βρέθηκε ότι σημαντικές δυναμικές ιδιότητες που δεν αποκαλύπτονται με συνήθεις τεχνικές εξόρυξης δεδομένων από τις χρονοσειρές σημάτων που εκπέμπονται από πολύπλοκα συστήματα, μπορούν να αναδυθούν από την ανάλυση τους στο νέο αυτό πεδίο χρόνου (στο φυσικό χρόνο). Αυτό οδηγεί σε μείωση της αβεβαιότητας με αποτέλεσμα να εξάγεται η μέγιστη δυνατή πληροφορία από ένα σήμα [14]. Η ανάλυση στο φυσικό χρόνο έχει το πλεονέκτημα ότι επιτρέπει τη μελέτη της δυναμικής εξέλιξης ενός πολύπλοκου συστήματος και έτσι εκτιμά ικανοποιητικά το πότε το σύστημα προσεγγίζει (ένα) κρίσιμο σημείο. Για το λόγο αυτό, ο φυσικός χρόνος αποτελεί μια σημαντικά εναλλακτική τεχνική στην πρόβλεψη επικείμενων καταστροφικών γεγονότων. Σχετικά παραδείγματα ανάλυσης δεδομένων, σε αυτό το νέο πεδίο του χρόνου, έχουν δημοσιευθεί σε ερευνητικά πεδία διαφόρων επιστημών όπως Καρδιολογία .[15]

Στην παρούσα μελέτη, γίνεται χρήση της μεθοδολογίας όπως εφαρμόζεται στο πεδίο της Σεισμολογίας. Με δεδομένο ότι ο φυσικός χρόνος επιτρέπει την εκτίμηση του χρόνου ενός επικείμενου μεγάλου γεγονότος, όπως προαναφέρθηκε, μπορεί με τον τρόπο αυτό να προσδιορίσει πότε ένα πολύπλοκο σύστημα προσεγγίζει ένα κρίσιμο σημείο. Βρέθηκε ότι όταν η διασπορά του φυσικού χρόνου, που συμβολίζεται με (κ_1), γίνει ίση με 0.070 (που υποδηλώνει ότι το σύστημα προσεγγίζει στο κρίσιμο σημείο) και οι εντροπίες φυσικού χρόνου (ορθή και αντίστροφη) είναι μικρότερες από 0.0966, ο σεισμός συμβαίνει μετά από λίγες ώρες έως μία εβδομάδα. Αυτό έχει επιβεβαιωθεί μέχρι σήμερα για πολλούς σημαντικούς σεισμούς στην Ελλάδα. Στην παρούσα μελέτη ως σημαντικό γεγονός οριοθετείται η κατάρρευση του δοκιμίου μαρμάρου και εξετάζεται η χρονική εξέλιξη της διασποράς και της εντροπίας φυσικού χρόνου. Ο μετασχηματισμός μίας χρονοσειράς από το συμβατικό πεδίο χρόνου στο πεδίο του φυσικού χρόνου, γίνεται αν αγνοήσουμε τον συμβατικό χρόνο κάθε δείγματος της χρονοσειράς και διατάξουμε το κάθε δείγμα με βάση την σειρά εμφάνισης του. Με Π.Α.Δ.Α – Σχολή Μηχανικών, Τμήμα Μηχανικών Τοπογραφίας και Γεωπληροφορικής Ανάπτυξη δικτυακών εφαρμογών σε περιβάλλον matlab : Εφαρμογή στην ανάλυση και επεξεργασία σεισμολογικών δεδομένων

τον τρόπο αυτό διατηρείται το ενεργειακό περιεχόμενο της χρονοσειράς αλλά αγνοείται η χρονική πληροφορία.

Σε μία χρονοσειρά αποτελούμενη από N διακριτά γεγονότα ο φυσικός χρόνος του k -στου γεγονότος που χαρακτηρίζεται από ενέργεια A_k , ορίζεται [7, 8] ως $\chi_k = k/N$. Το Σχήμα 2.1 είναι ενδεικτικό της απεικόνισης μίας χρονοσειράς σε συμβατικό και σε φυσικό χρόνο. Η ανάλυση στο πεδίο του φυσικού χρόνου, επικεντρώνεται στην μελέτη την εξέλιξη του ζεύγους (χ_k, p_k) , όπου:

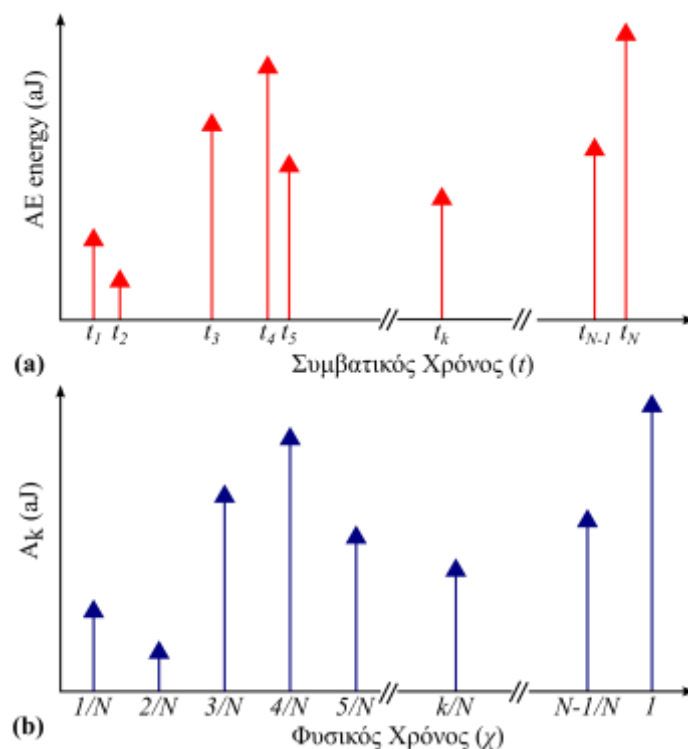
$$p_k = \frac{A_k}{\sum_{n=1}^N A_n} \quad (1.1)$$

είναι η κανονικοποιημένη ενέργεια του k -στού γεγονότος [7, 9].

Από τη Σχέση (2.1) προκύπτει:

$$\sum_{k=1}^N p_k = 1 \quad (1.2)$$

Θα πρέπει να σημειωθεί πως η εξέλιξη των ζευγών (χ_k, A_k) και (χ_k, p_k) θεωρούνται



Σχήμα 0.1 (α) Μια χρονοσειρά ενέργειας ΑΕ στο συμβατικό χρόνο (t). (β) Η ίδια χρονοσειρά στο φυσικό χρόνο (χ).

ισοδύναμες στην ανάλυση του φυσικού χρόνου.

Σε μία χρονοσειρά αποτελούμενη από N γεγονότα, η επίδραση του συντελεστή \hat{T} της αντιστροφής χρόνου στα A_k δίδεται από τη σχέση:

$$\hat{T}A_k = A_{N-k+1} \quad (1.3)$$

έτσι ώστε ο πρώτος παλμός ($k=1$) να τοποθετείται τελευταίος στην χρονικά αντεστραμμένη χρονοσειρά, ο δεύτερος παλμός ($k=2$) προτελευταίος κ.ο.κ. Κατά συνέπεια, ο συντελεστής \hat{T} της χρονικής αντιστροφής στον φυσικό χρόνο επιδρά στις πιθανότητες P_k ως εξής:

$$\hat{T}P_k = P_{N-k+1} \quad (1.4)$$

Για τις ανάγκες της ανάλυσης σε φυσικό χρόνο ορίστηκε η συνεχής συνάρτηση $\Phi(\omega)$ [7, 9]

$$\Phi(\omega) = \frac{\sum_{k=1}^N A_k e^{i\omega \frac{k}{N}}}{\sum_{n=1}^N A_n} = \sum_{k=1}^N p_k e^{i\omega \frac{k}{N}} = \sum_{k=1}^N p_k e^{i\omega x_k} \quad (1.5)$$

όπου, $\omega = 2\pi\varphi$ και φ η συχνότητα στο φυσικό χρόνο, που ονομάζεται φυσική συχνότητα. Το κανονικοποιημένο φάσμα ισχύος $\Pi(\omega)$ υπολογίζεται ως εξής:

$$\Pi(\omega) = |\Phi(\omega)|^2 = \left| \sum_{k=1}^N p_k e^{i\omega \frac{k}{N}} \right|^2 \quad (1.6)$$

Θεωρώντας ότι $\omega \rightarrow 0$, η $\Phi(\omega)$ δίνει τις στατιστικές ιδιότητες της συνάρτησης $p(\chi)$, όπως η μέση τιμή (μ), η τυπική απόκλιση (σ), η διασπορά κ.τ.λ. Κάνοντας το ανάπτυγμα Taylor της Εξίσωσης (1.6) για χαμηλές τιμές του ω γύρω από το μηδέν (0) έχουμε:

$$\Pi(\omega) = 1 + \kappa_1 \omega^2 + \kappa_2 \omega^4 + \kappa_3 \omega^6 + \kappa_4 \omega^8 + \dots \quad (1.7)$$

όπου

$$\kappa_1 = -\frac{1}{2} \left. \frac{d^2 \Pi(\omega)}{d\omega^2} \right|_{\omega=0} \quad (1.8)$$

Αναλύοντας το $\frac{d^2\Pi(\omega)}{d\omega^2}$ ως ακολούθως έχουμε:

$$\frac{d^2\Pi(\omega)}{d\omega^2} = \Phi^*(\omega) \frac{d^2\Phi(\omega)}{d\omega^2} + \Phi(\omega) \frac{d^2\Phi^*(\omega)}{d\omega^2} + 2 \frac{d\Phi(\omega)}{d\omega} \frac{d\Phi^*(\omega)}{d\omega} \quad (1.9)$$

Λαμβάνοντας υπόψη την Εξίσωση (1.5) καθώς και ότι $\Phi(0)=1$, προκύπτει:

$$\kappa_1 = -\frac{1}{2} \left[-\sum_k p_k \chi_k^2 - \sum_k p_k \chi_k^2 - 2 \left(\sum_k p_k \chi_k \right)^2 \right] = \langle \chi^2 \rangle - \langle \chi \rangle^2 \quad (1.10)$$

όπου

$$\langle \chi^n \rangle = \sum_{k=1}^N p_k \chi_k^n \quad (1.11)$$

Η Εξίσωση (1.11) υποδηλώνει τις ροπές του φυσικού χρόνου που σταθμίζονται από το P_k . Έτσι, η ποσότητα κ_1 αντιστοιχεί στην διασπορά του φυσικού χρόνου χ , ανηγμένη όπως αυτή προκύπτει για P_k , και δίδεται από τη σχέση:

$$\kappa_1 = \langle \chi^2 \rangle - \langle \chi \rangle^2 = \sum_{k=1}^N p_k \left(\frac{k}{N} \right)^2 - \left(\sum_{k=1}^N \frac{k}{N} p_k \right)^2 \quad (1.12)$$

Σημειώνεται πως καθώς το κανονικοποιημένο φάσμα ισχύος $\Pi(\omega)$ δεν αλλάζει κατά την αντιστροφή του χρόνου, το ίδιο ισχύει και για την ποσότητα κ_1 .

Όπως υποδείχθηκε [8], η ποσότητα κ_1 μπορεί να χρησιμεύσει ως παράμετρος τάξης [10] που μπορεί να προσδιορίσει την προσέγγιση του δυναμικού συστήματος σε ένα κρίσιμο σημείο [47]. Για το λόγο αυτό, η προϋπόθεση:

$$\kappa_1 = 0.070 \quad (1.13)$$

έχει βρεθεί ότι ισχύει για μια σειρά κρίσιμων συστημάτων συμπεριλαμβανομένου του μοντέλου σεισμών Olami-Feder-Christensen [12] και μοντέλο τρένου Burridge-Knoppoff (βλ. Πίνακας 8.1 σελ. 343 της αναφοράς [8]).

Στο κανονικοποιημένο φάσμα ισχύος η συνθήκη $\kappa_1=0.07$ εξετάζεται συγκρίνοντας την καμπύλη του φάσματος για $\varphi \in [0.05]$ των πειραματικών δεδομένων με την καμπύλη του φάσματος που αντιστοιχεί στην κρίσιμη κατάσταση (θεωρητικά υπολογιζόμενη). Θεωρούμε ότι το σύστημα προσεγγίζει την κρίσιμη Π.Α.Δ.Α – Σχολή Μηχανικών, Τμήμα Μηχανικών Τοπογραφίας και Γεωπληροφορικής Ανάπτυξη δικτυακών εφαρμογών σε περιβάλλον matlab : Εφαρμογή στην ανάλυση και επεξεργασία σεισμολογικών δεδομένων

κατάσταση αν η πειραματική καμπύλη προσεγγίζει «από κάτω προς τα επάνω» (και ταυτιστεί) με την θεωρητική.

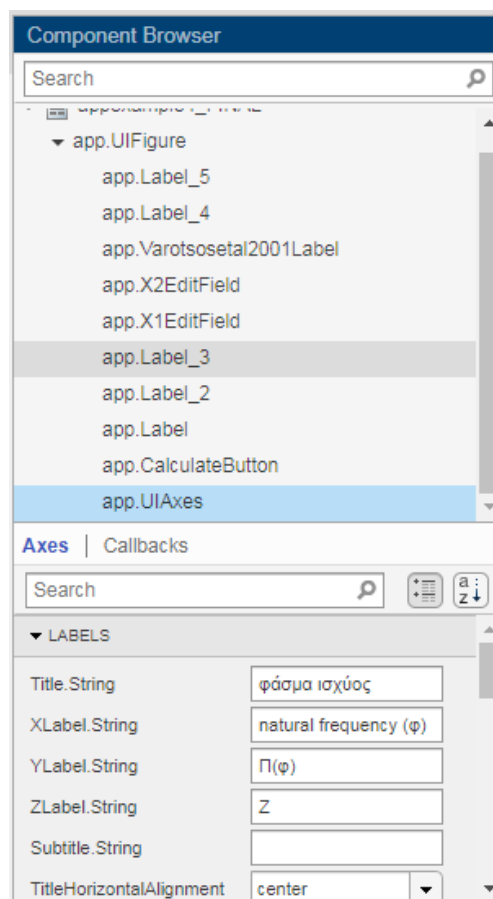
3.1.2. Σχεδιασμός της εφαρμογής στο App Designer

Η εφαρμογή θα ζητάει από χρήστη ένα συγκεκριμένο εύρος του δείγματος των σεισμολογικών δεδομένων που θα πρέπει να πληκτρολογήσει. Έτσι εισάγουμε δύο πεδία “Edit Filed:Numeric” σέρνοντας τα από Component Library στο interface της εφαρμογής μας. Επιλέγοντας τώρα το Component Browser βλέπουμε τα δύο πεδία που δημιουργήσαμε, επιλέγοντας πάνω τους μπορούμε να ρυθμίσουμε κάποιες ιδιότητες. Ονομάζουμε τα πεδία μας X1 και X2 για το πρώτο και το τελευταίο δείγμα αντίστοιχα, ρυθμίζουμε χρώματα γραμματοσειράς σε κόκκινο και χρώμα περιγράμματος σε γκρι χρώμα καθαρά για την εμφάνιση. Παρατηρούμε ότι στις ιδιότητες των πεδίων στη καρτέλα “Value” έχει τοποθετηθεί μηδέν αυτόματα με τη δημιουργία του πεδίου διότι είναι πεδίο τύπου “numeric” και πάντα πρέπει να υπάρχει μια αριθμητική τιμή.

Έπειτα, εισάγουμε άλλα 3 πεδία ετικετών Labels από Component Library σέρνοντας τα στο interface της εφαρμογής μας που γράφουμε σχόλια (σε μορφή κειμένου) για τις τιμές X1, X2 για να γνωρίζει ο χρήστης που θα το τρέχει τι είναι. Τροποποιούμε και εδώ τα χρώματα περιγραμμάτων των πεδίων και το χρώμα γραμματοσειράς των κειμένων και τα στοιχίζουμε καθέτως αντίστοιχα.

Στη συνέχεια προσθέτουμε άλλα 3 πεδία ετικετών Labels από Component Library σέρνοντας τα στο interface της εφαρμογής μας που στο ένα γράφουμε σχόλια του διαγράμματος μας και τα άλλα δύο τα χρησιμοποιούμε σαν υπόμνημα γράφοντας τη θεωρητική και τη πειραματική καμπύλη στα αντίστοιχα χρώματα των γραφικών παραστάσεων του διαγράμματος δηλαδή κόκκινο και μπλε αντίστοιχα.

Παρατηρώντας στο Component Browser υπάρχει ένα πεδίο που περιέχει όλα τα υπόλοιπα πεδία και ονομάζεται app.UIFigure, είναι το κεντρικό παράθυρο που εμφανίζεται στην οθόνη μας και δημιουργείται αυτόματα απτή στιγμή που θα ανοίξουμε ένα νέο project στο App Designer. Επιλέγουμε στις ιδιότητες του πεδίου αυτού ένα γαλάζιο χρώμα για το φόντο του παραθύρου για την εφαρμογή μας. Κάθε λειτουργία και νέο πεδίο που εισάγουμε στην εφαρμογή μας έχει μπροστά από όνομα του το τύπο app., είναι Π.Α.Δ.Α – Σχολή Μηχανικών, Τμήμα Μηχανικών Τοπογραφίας και Γεωπληροφορικής Ανάπτυξη δικτυακών εφαρμογών σε περιβάλλον matlab : Εφαρμογή στην ανάλυση και επεξεργασία σεισμολογικών δεδομένων



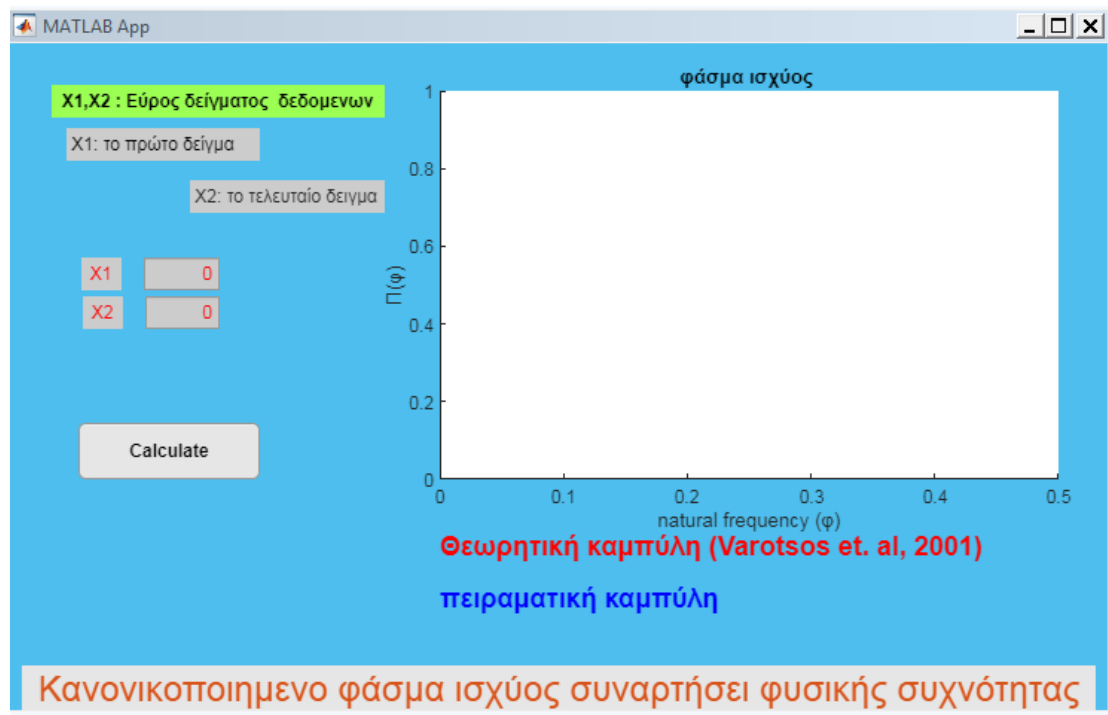
Εικόνα 3.1.Επιλογές πεδίων στο Component Browser

ιδιότητα του App Designer που δημιουργεί για κάθε λειτουργία , έτσι αργότερα στο κώδικα που θα δούμε παρακάτω όταν χρειαζόμαστε τη τιμή κάποιου πεδίου το καλούμε με αυτό το όνομα

Μετά, προσθέτουμε ένα κενό διάγραμμα (λειτουργία Axes) απο Component Library σέρνοντας το πάλι στο interface μας και φτιάχνουμε τις ετικέτες των αξόνων μας X και Y σε natural frequency(φ) και $\Pi(\varphi)$ αντίστοιχα καθώς και την ετικέτα του τίτλου διαγράμματος. Όλες τις ετικέτες την ρυθμίζουμε απο Component Browser επιλέγοντας το διάγραμμα που προσθέσαμε.

Τέλος προσθέτουμε τη κύρια λειτουργία της εφαρμογής για τον υπολογισμό ,την εμφάνιση διαγράμματος του φάσματος ισχύος και το φόρτωμα του αρχείου των σεισμολογικών δεδομένων χρησιμοποιώντας τη λειτουργία button απ'τό Component Library και δίνοντας του την ονομασία “calculate” .

Η τελική εμφάνιση της εφαρμογής:



Εικόνα 3.2.Οπτική προβολή της εφαρμογής στο App Designer

3.1.3. Προγραμματισμός της εφαρμογής στο App Designer

Προχωράμε στο προγραμματιστικό μέρος της εφαρμογής , ως στιγμή έχουμε δημιουργήσει απλώς την εμφάνιση της εφαρμογής μας αν τρέξουμε για δοκιμή το πρόγραμμα δεν θα δούμε κάποια ενέργεια . Έτσι λοιπόν πατώντας δεξιά κλικ πάνω στο πεδίο app.Button με όνομα “calculate” στο Component Browser και επιλέγοντας Callbacks->Add ButtonPushedFnc callback ενεργοποιούμε την ιδιότητα “Callback “, πρόκειται για την επανάκληση που αναφερθήκαμε στο θεωρητικό μέρος της εργασίας , δίνοντας του αυτή την ιδιότητα το App Designer δέχεται ότι όταν πατηθεί αυτή η λειτουργία δηλαδή το κουμπί “calculate” , θα κάνει κάποια ενέργεια . Παρατηρούμε ότι απτη στιγμή που θα επιλέξουμε την ιδιότητα Callback δημιουργείται μια ένας κώδικας στο code view που περιμένει κάποιες εντολές για να εκτελέσει μια ενέργεια όταν πατηθεί το κουμπί από χρήστη .

```
% Callbacks that handle component events
methods (Access = private)

% Button pushed function: loadButton
function loadButtonPushed(app, event)
```

Εικόνα 3.3.Αυτόματη δημιουργία εντολών

Ξεκινώντας με τις δύο μεταβλητές που δηλώσαμε X1 και X2 που θα εισάγει ο χρήστης τις δηλώνουμε ως A και B στη μορφή που φαίνεται παρακάτω δηλαδή

```
A=app.X1EditField.Value;
```

```
B=app.X2EditField.Value;
```

Προσοχή αν δεν βάλουμε στο τέλος .Value δεν θα δέχεται τη τιμή αλλά θα το δέχεται όλο το πεδίο σαν μεταβλητή και δεν θα δουλεύει σωστά . Με αυτό τον τρόπο έχουμε τις δύο μεταβλητές που θα πρέπει αν πληκτρολογήσει ο χρήστης στο interface ,

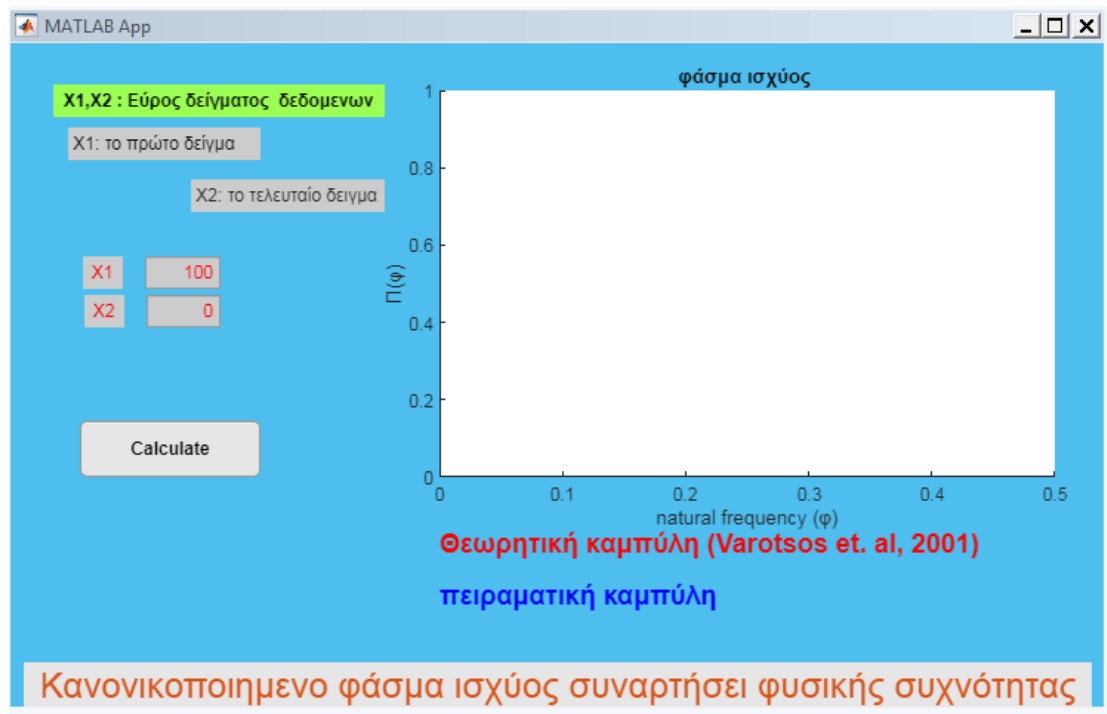
Στη συνέχεια , βάζουμε ένα if..else..end για έλεγχο των τιμών που πρέπει να βάλει ο χρήστης , αν τυχόν δεν βάλει τιμές και πατήσει κατευθείαν το κουμπί calculate δηλαδή με μηδενικές τιμές, να εμφανίζει στο χρήστη το μήνυμα “Δεν έχετε εισάγει τιμές για το εύρος δείγματος “ . Αντίστοιχα αν δώσει τιμή μόνο σε μία απ τις δύο μεταβλητές θα εμφανίζεται το ίδιο μήνυμα . Χρησιμοποιήσαμε τις εντολές AND(&&) και OR(||) για να καλύψουμε κάθε περίπτωση μηδενικής τιμής από χρήση .

```
% Callbacks that handle component events
methods (Access = private)

% Button pushed function: loadButton
function loadButtonPushed(app, event)
    A=app.X1EditField.Value;
    B=app.X2EditField.Value;
    if A==0 && B==0 || B==0 || A==0
        f = app.UIFigure;
        mess = "Δεν έχετε εισάγει τιμές για το εύρος δείγματος";
        title = "Error 101";
        uialert(f, mess, title);
```

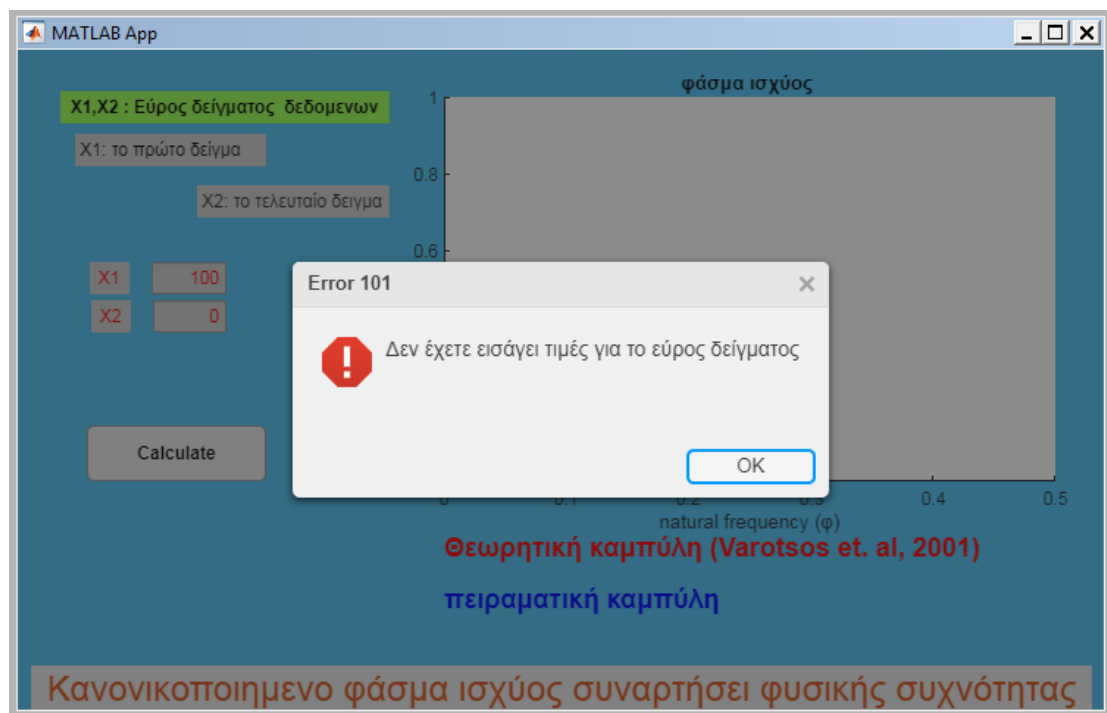
Εικόνα 3.4. Έλεγχος έγκυρων τιμών

Οπότε ελέγχουμε τι εμφανίζει σε αυτή τη περίπτωση που ο χρήστης κάνει λάθος στις τιμές , τρέχοντας την εφαρμογή .



Εικόνα 3.5.Λανθασμένες τιμές σαν είσοδο μεταβλητών

Αφήνουμε εσκεμμένα μηδενική τη δεύτερη τιμή και πατάμε το load .



Εικόνα 3.6.Μήνυμα λάθους

Έπειτα η πατάμε OK στο μήνυμα του προβλήματος και η εφαρμογή περιμένει το χρήστη να πληκτρολογήσει τιμές .

Συνεχίζουμε στη περίπτωση του else , δηλαδή στη περίπτωση που χρήστης έχει πληκτρολογήσει τιμές στις 2 μεταβλητές και πατάει το κουμπί “calculate” . Τότε τρέχει μια σειρά εντολών , αρχικά ζητάει απτο χρήστη σαν είσοδο της εφαρμογής να επιλέξει απτον υπολογιστή του ένα αρχείο που περιέχει τα σεισμολογικά δεδομένα. Το αρχείο αυτό πρέπει να είναι της εξής μορφής : μια στήλη με τιμές των σεισμών σε ρίχτερ , ξεκινώντας η 1η τιμή απτή 1η σειρά του αρχείου και να είναι τύπος αρχείο text file.

Δηλώνουμε σε μια μεταβλητή το αρχείο που θα φορτώσει ο χρήστης στη σωστή μορφή χρησιμοποιώντας τις εντολές fopen και textscan για να αναγνωρίσει το τύπο της μεταβλητής που είναι αριθμητικός και μετατρέπουμε τις τιμές σε αριθμητική τιμή τύπου double δηλαδή τύπο αριθμητικών δεδομένων που παρέχει επαρκή ακρίβεια για τις περισσότερες υπολογιστικές εργασίες .

```
else
Ar=uigetfile;          %ανοιγμα αρχείου μονοστηλου με μετρήσεις , import
Ar2=fopen(Ar);
Ar3=textscan(Ar2, '%f ');
Ar4= cell2mat(Ar3);
```

Εικόνα 3.7.Κώδικας εισαγωγής αρχείου και μετατροπής δεδομένων

Έπειτα χρησιμοποιούμε μια σειρά εντολών που είναι ο υπολογισμός του φάσματος και η εμφάνιση του διαγράμματος του . Προσοχή, στην εμφάνιση του διαγράμματος πρέπει να δηλώσουμε να εμφανίζεται στο τρέχον διάγραμμα(axes) της εφαρμογής και όχι να ανοίγει σε νέο παράθυρο. Αυτό επιτυγχάνεται δηλώνοντας ,σε κάθε εντολή που περιέχει εντολές τύπου plot και εμφάνισης , app.UIAxes δηλαδή να εμφανίσει τα πάντα στο τρέχον παράθυρο της εφαρμογής . Το app.UIAxes στην ουσία είναι η μεταβλητή που περιέχει το διάγραμμα.

```

56     h=3;
57     Mcatalog=Ar4(A:B);
58     Ncat=length(Mcatalog);
59     k=(1:length(Mcatalog))';
60     xk=k/Ncat;
61     %naturalpspectrum function
62     Ek=Mcatalog;
63     pk=Ek./sum(Ek);
64     mResult = [];
65     fi=(0:0.01:2)';
66     nCnt = 1;
67     for i=1:length(fi);
68         Pfi=(sum(pk.*cos(2*pi*fi(i)*xk)).^2+(sum(pk.*sin(2*pi*fi(i)*xk)).^2);
69         mResult = [mResult; Pfi];
70         nCnt = nCnt + 1;
71     end
72     % Plot natural spectrum
73     g=2*pi*fi;
74     Pfi2=(18./(5*(g.^2)))-(6*cos(g)./(5*(g.^2)))-(12*sin(g)./(5*(g.^3)));
75     if h==1
76         g=2*pi*fi;
77         Pfi2=(18./(5*(g.^2)))-(6*cos(g)./(5*(g.^2)))-(12*sin(g)./(5*(g.^3)));
78     else
79
80     plot(app.UIAxes,fi,mResult);
81     hold(app.UIAxes, 'on');
82     plot(app.UIAxes,fi,Pfi2,'r');
83     hold(app.UIAxes, 'off');

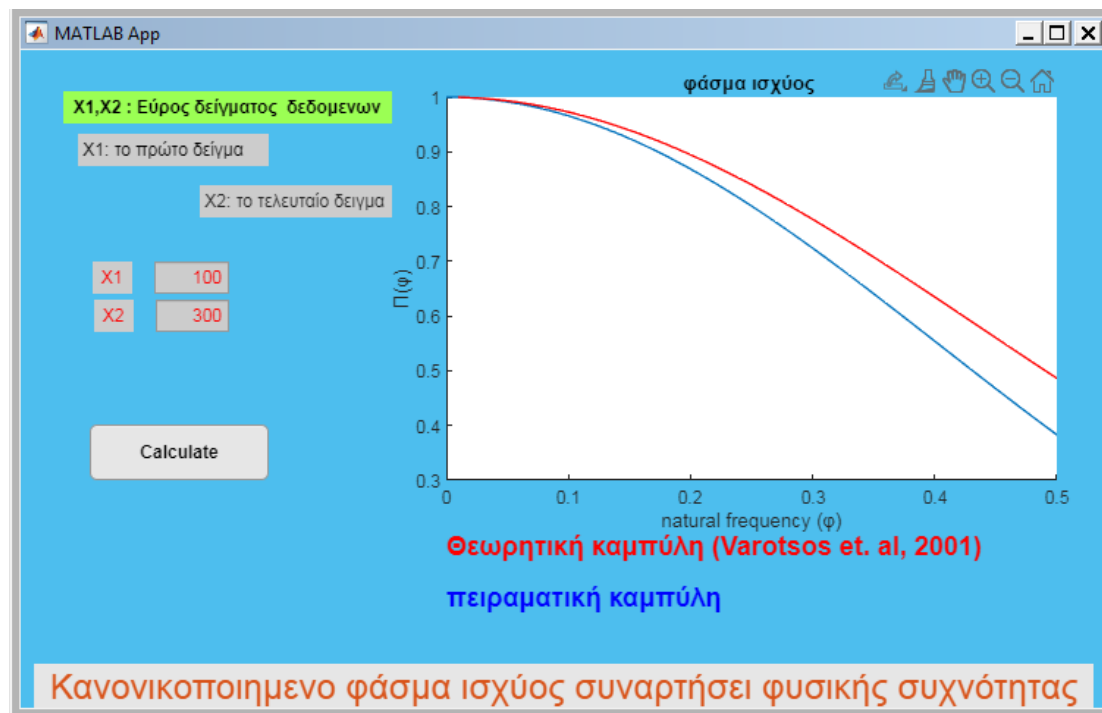
```

Εικόνα 3.8.Κώδικας υπολογισμού φάσματος ισχύος και εμφάνιση διαγράμματος

Τέλος , κλείνουμε το if και το callback function και τρέχουμε την εφαρμογή μας .

3.1.4. Αποτελέσματα

Αφού ο χρήστης τρέξει την λειτουργία “calculate” εμφανίζεται σε ένα νέο παράθυρο το διάγραμμα του κανονικοποιημένου φάσματος ισχύος συναρτήσεως της φυσικής συχνότητας των δεδομένων .



Εικόνα 3.9. Διάγραμμα κανονικοποιημένου φάσματος ισχύος συναρτήσεως της φυσικής συχνότητας των δεδομένων

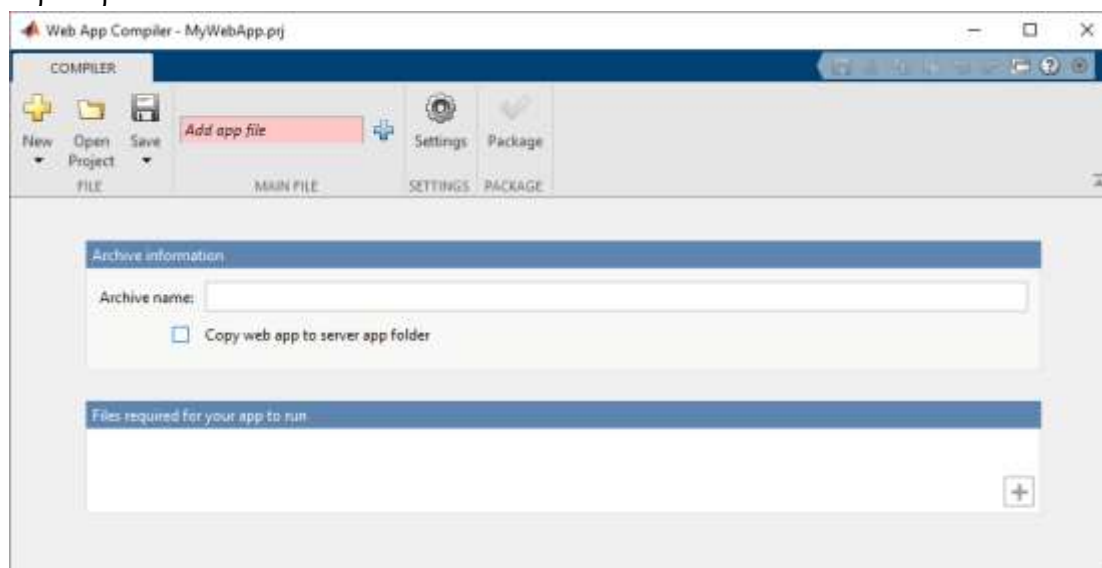
Το αποτέλεσμα που λαμβάνουμε αποτελείται από δύο καμπύλες φάσματος όπου η κόκκινη έχει υπολογιστεί για $\kappa_1 \rightarrow 0.070$ δηλαδή αποτελεί την θεωρητικά υπολογιζόμενη καμπύλη που αντιστοιχεί στο κρίσιμο σημείο, ενώ η μπλε καμπύλη αντιστοιχεί στην υπολογιζόμενη με βάση τα πειραματικά δεδομένα. Ο αναλυτής λοιπόν μπορεί να εκτελεί αλληπάλληλους υπολογισμούς (αλλάζοντας το X2) και να εξετάζει εάν η μπλε καμπύλη τείνει (από κάτω προς τα πάνω) προς την κόκκινη, γεγονός που θα υποδήλωνε την είσοδο του συστήματος σε κρίσιμη κατάσταση

Σε αυτό το παράθυρο του διαγράμματος υπάρχει η δυνατότητα απ'τό χρήστη να :

- Επιλέξει συγκεκριμένα σημεία πάνω στις γραφικές παραστάσεις
- Μεγέθυνση και σμίκρυνση στο διάγραμμα (Zoom In, Zoom Out)
- Αποθηκεύσει το διάγραμμα σε αρχείο μορφής εικόνων. Επιλέγοντας στο παράθυρο τη καρτέλα File->Save as->JPEG ή TIF ή οποιασδήποτε άλλης μορφής εικόνων.

3.1.5. Δημιουργία διαδικτυακής εφαρμογής

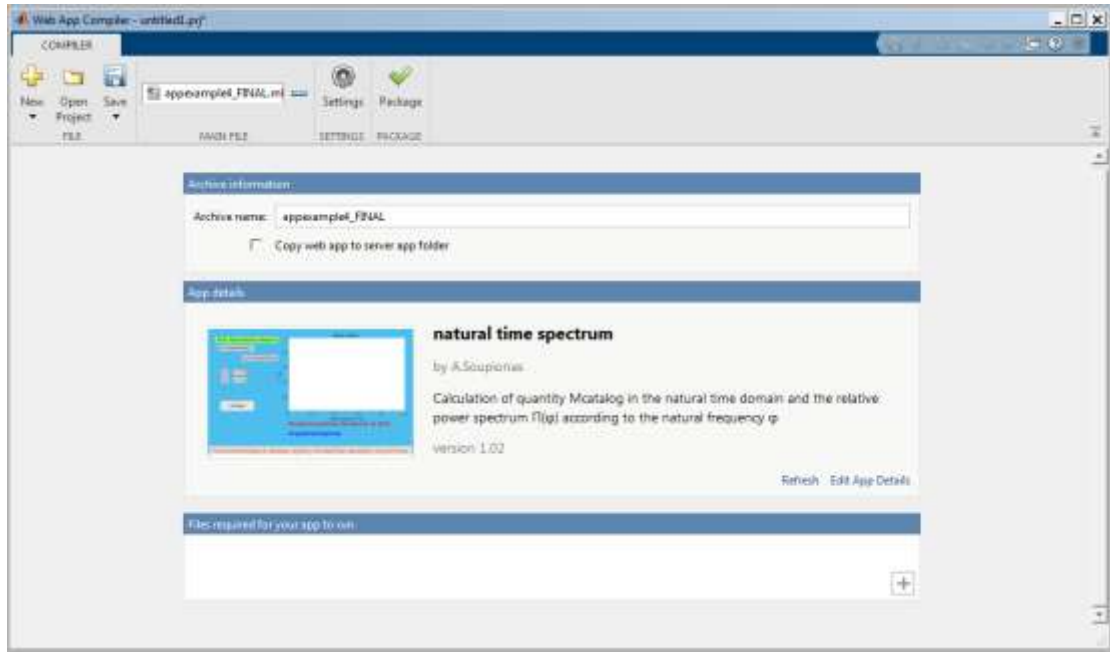
Ανοίγοντας το MATLAB πληκτρολογούμε “WebAppCompiler” στο Command window και ανοίγει το Matlab Compiler . Σε αυτή τη καρτέλα εμφανίζεται ένα παράθυρο



Εικόνα 3.10.Κεντρικό Μενού του εργαλείου Compiler[3]

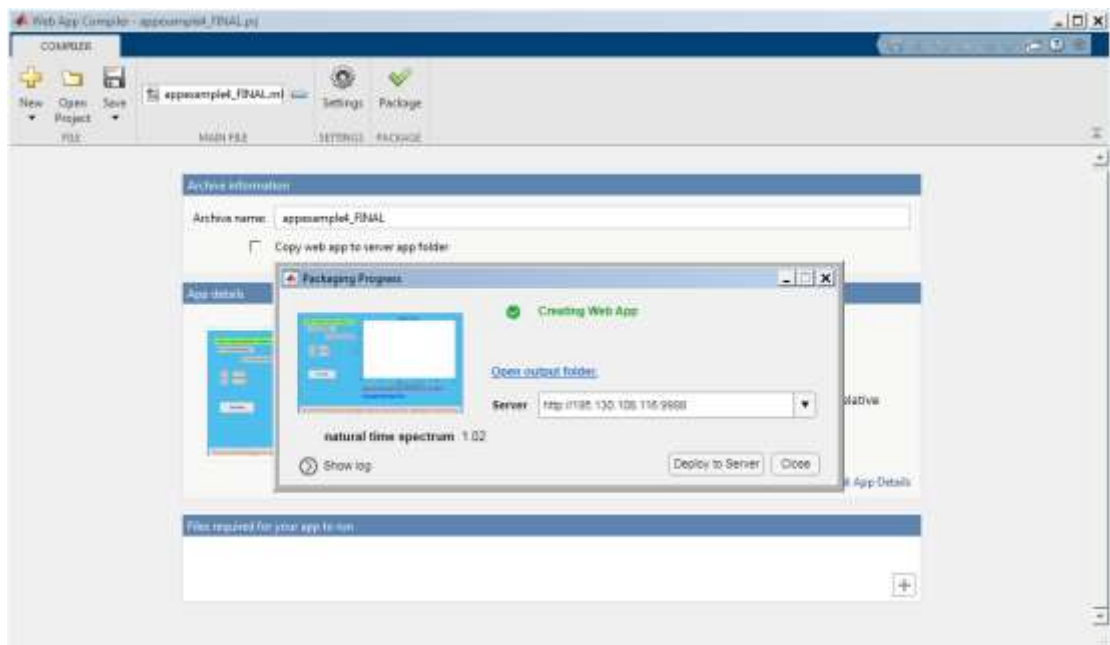
Στην επιλογή Add app file προσθέτουμε το αρχείο του σχεδιασμού εφαρμογής που δημιουργούμε στο App Designer και είναι σε μορφή .mlapp

Όταν φορτώνουμε το αρχείο της εφαρμογής που θα δημιουργήσουμε στο App Designer , στην επιλογή Files required for your App to run εμφανίζονται αυτόματα τα εξωτερικά αρχεία που χρησιμοποιούνται για να τρέξει εφαρμογή όπως functions του MATLAB . Στη περίπτωση μας δε χρησιμοποιούμε εξωτερικές συναρτήσεις ή κώδικες που καλούμε γιατί τα έχουμε βάλει όλα μεστό κώδικα της εφαρμογής .



Εικόνα 3.11. Φόρτωμα εφαρμογής στο Compiler

Έπειτα, επιλέγοντας το Package γίνεται η δημιουργία της διαδικτυακής εφαρμογής και δημιουργείται ένα αρχείο .ctf file. Μας ζητείται να πληκτρολογήσουμε το σύνδεσμο της ιστοσελίδας που θα δημοσιευτεί η εφαρμογή μας. Πατώντας το deploy to server δημοσιεύεται η εφαρμογή στο διαδίκτυο.



Εικόνα 3.12. Δημοσίευση της εφαρμογής στο διαδίκτυο

3.2. Δημιουργία εφαρμογής υπολογισμού κατανομής Gutenberg-Richter (frequency-magnitude distribution GR-FMD) και b-value

Σε αυτή την εφαρμογή θέλουμε να δημιουργήσουμε ένα αρχείο .mlapp στο AppDesigner , το οποίο δέχεται ένα αρχείο .txt απτο χρήστη που περιέχει σεισμολογικά δεδομένα. Σκοπός της εφαρμογής είναι ο υπολογισμός κατανομής Gutenberg-Richter (frequency-magnitude distribution GR-FMD) και b-value και η εμφάνιση διαγράμματος της κατανομής αυτής.

3.2.1. Νόμος του Gutenberg – Richter

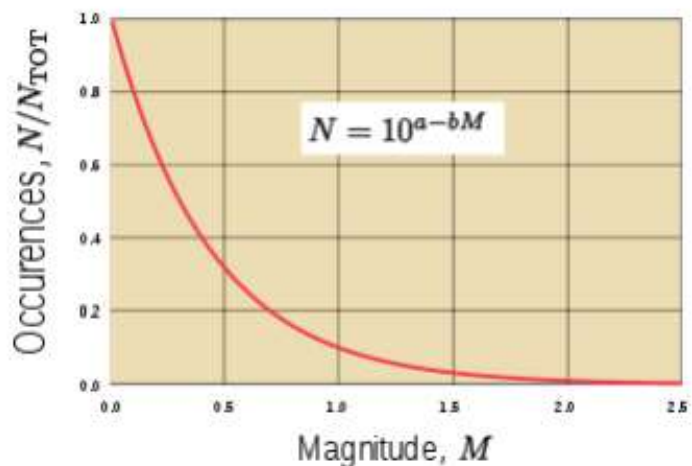
Στη σεισμολογία ο νόμος του Gutenberg– Richter εκφράζει τη σχέση μεταξύ του μεγέθους του συνολικού αριθμού των σεισμών σε κάθε δεδομένη περιοχή και χρονική περίοδος του μικρότερου μεγέθους.

$$\text{Log}_{10}N=a-bM$$

Η

$$N=10^{a-bM}$$

- Όπου N είναι ο αριθμός των γεγονότων που έχουν μέγεθος μεγαλύτερο ή ίσο του N
- a και b είναι σταθερές



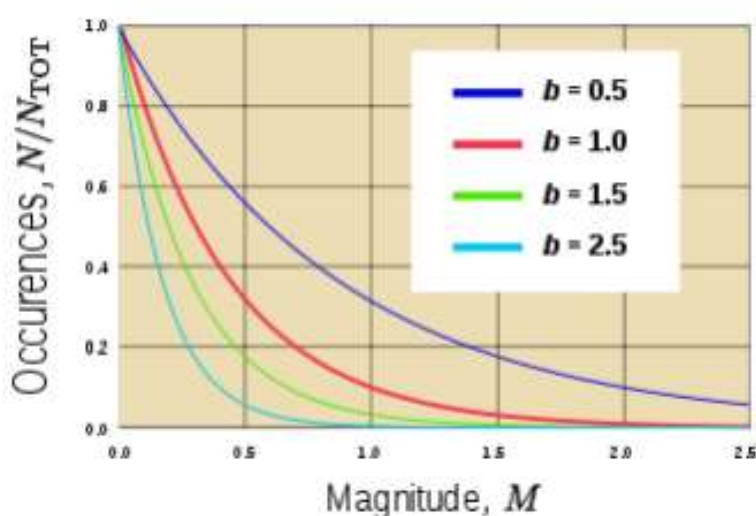
Gutenberg–Richter law for $b = 1$

Εικόνα 3.13. Νόμος του Gutenberg – Richter για $b=1$

Η σχέση μεταξύ του μεγέθους ενός σεισμού και της συχνότητας παρατηρήθηκε αρχικά απ'τόν Charles Francis Richter και Beno Gutenberg το 1956.

Αυτή η σχέση μεταξύ του μεγέθους ενός σεισμικού γεγονότος και της συχνότητας εμφάνισης του είναι γνωστό μέσω των σταθερών a και b οι οποίες μπορεί να διαφοροποιούνται από περιοχή σε περιοχή ή σε διαφορετικά χρονικά διαστήματα

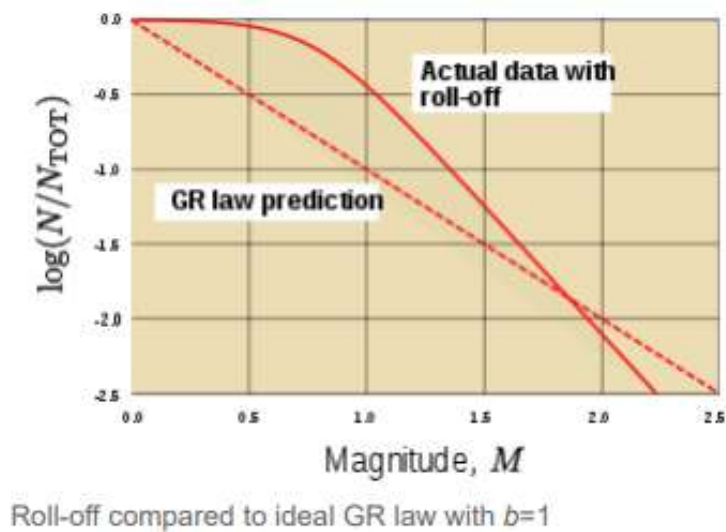
Η παράμετρος b (συχνά αναφερόμενη b -value) είναι συνήθως κοντά στο 1.0 στη ενεργά σεισμικά περιοχές . Αυτό σημαίνει ότι για μια δεδομένη συχνότητα μεγέθους 4.0 ή μεγαλύτερη θα είναι 10 φορές περισσότερο από ένα σεισμό μεγέθους 3.0 και 100 φορές μεγαλύτερη από έναν μεγέθους 2.0



GR law plotted for various b -values

Εικόνα 3.14. Νόμος του Gutenberg – Richter για διάφορες τιμές του b

Υπάρχει μια εμφανής μείωση της παραμέτρου b -value για μικρότερα μεγέθη με εμβέλεια σε όλους τους εμπειρικούς καταλόγους των σεισμών . Αυτό το φαινόμενο περιγράφεται ως “roll-off” της b -value μια περιγραφή λόγω της πλοκής της λογαριθμικής εκδοχής του Gutenberg– Richter , δηλαδή όσο πιο χαμηλό μέγεθος τόσο πιο επίπεδο .



Εικόνα 3.15. Συγκρίνοντας το Roll of με το ιδανικό Gutenberg– Richter με $b=1$

Αυτό μπορεί σε μεγάλο βαθμό να οφείλεται στα ελλιπή δεδομένα λόγω αδυναμίας εντοπισμού. Δηλαδή πολλοί σεισμοί χαμηλού μεγέθους δεν ανιχνεύονται επειδή λιγότεροι σταθμοί τους ανιχνεύουν και τους καταγράφουν λόγω της μείωσης του σήματος των οργάνων σε επίπεδο θορύβου .

Μερικά σύγχρονα μοντέλα δυναμικής σεισμού, ωστόσο , προβλέπει μια φυσική μετατόπιση στην κατανομή μεγέθους του σεισμού .

Η παράμετρος a -value καθορίζει το ποσοστό σεισμικότητας κάθε περιοχής

$$N = N_{TOT} 10^{-bM}$$

Όπου $N_{TOT} 10^a$ Είναι το συνολικό νούμερο των γεγονότων. Εφόσον 10^a είναι το συνολικό νούμερο των γεγονότων , 10^{-bM} θα πρέπει να είναι η πιθανότητα αυτών των γεγονότων .[4]

3.2.2. Σχεδιασμός της εφαρμογής στο App Designer

Ο σχεδιασμός της εφαρμογής θα είναι παρόμοιος με τη προηγούμενη εφαρμογή που δημιουργήθηκε , μόνο που εδώ ο χρήστης δεν εισάγει κάποιο εύρος δείγματος και στο τέλος εμφανίζεται και το αποτέλεσμα του υπολογισμού της κατανομής εκτός από διάγραμμα.

Έτσι , εισάγουμε

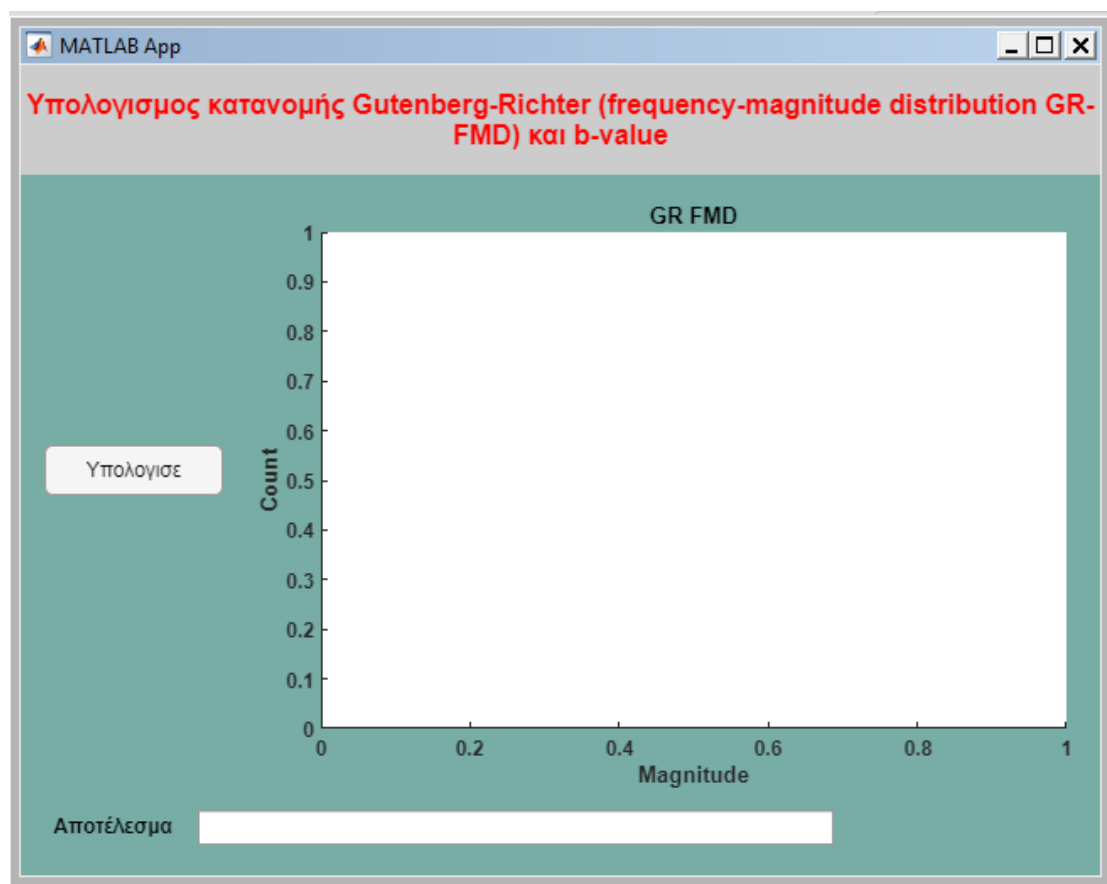
- ένα κενό διάγραμμα (Axes) απο component Library
- ένα πεδίο (τύπου text Area) απο component library που θα εμφανίζεται το αποτέλεσμα

Π.Α.Δ.Α – Σχολή Μηχανικών, Τμήμα Μηχανικών Τοπογραφίας και Γεωπληροφορικής
Ανάπτυξη δικτυακών εφαρμογών σε περιβάλλον matlab : Εφαρμογή στην ανάλυση και επεξεργασία σεισμολογικών δεδομένων

- ένα πεδίο (Label) απτο Component Library με το τίτλο της εφαρμογής
- ένα κουμπί “Υπολόγισε” (τύπου button) απτο Component Library

Στο πεδίο που θα εμφανίζεται το αποτέλεσμα ρυθμίζουμε το όνομα του πεδίου ως «αποτέλεσμα» απτο το Component Browser όπως κάναμε και στη προηγούμενη εφαρμογή , καθώς και τα χρώματα περιγραμμάτων πεδίων , το φόντο της εφαρμογής μας και το τύπο γραμματοσειράς κειμένων των πεδίων και λειτουργιών .

Η τελική εμφάνιση της εφαρμογής:



Εικόνα 3.16.Οπτική προβολή της 2^{ης} εφαρμογής στο App Designer

3.2.3. Προγραμματισμός της εφαρμογής στο App Designer

Προχωράμε στο προγραμματιστικό μέρος της εφαρμογής , ως στιγμής έχουμε δημιουργήσει απλώς την εμφάνιση της εφαρμογής μας αν τρέξουμε για δοκιμή το πρόγραμμα δεν θα δούμε κάποια ενέργεια . Έτσι λοιπόν πατώντας δεξιά κλικ πάνω στο πεδίο app.Button με ετικέτα “Υπολόγισε” στο Component Browser και επιλέγοντας Callbacks->Add ButtonPushedFnc callback ενεργοποιούμε την ιδιότητα “Callback “.

Π.Α.Δ.Α – Σχολή Μηχανικών, Τμήμα Μηχανικών Τοπογραφίας και Γεωπληροφορικής
Ανάπτυξη δικτυακών εφαρμογών σε περιβάλλον matlab : Εφαρμογή στην ανάλυση και επεξεργασία σεισμολογικών δεδομένων

Γράφουμε το κώδικα για την ενέργεια που θα κάνει το κουμπί “υπολόγισε” Χρησιμοποιούμε τον ίδιο τρόπο στο κώδικα για να ζητήσουμε απτο χρήστη να επιλέξει απτον υπολογιστή του ένα αρχείου δεδομένων τύπου .txt. και συνεχίζουμε στις εντολές του κώδικα για τον υπολογισμό και την εμφάνιση της κατανομής.

Στο κώδικα όμως καλείται μια εξωτερική συνάρτηση (function) που πρέπει να την προσθέσουμε στο κωδικά μας , επιλέγουμε στο Code Browser (Δες στο Θεωρητικό Μέρος) τα functions , τις συναρτήσεις και προσθέτουμε μία private function . Οι συναρτήσεις δημιουργούνται πριν από κάθε ενέργεια στο κώδικα έτσι ώστε να μπορούμε να τις καλούμε οποιαδήποτε στιγμή μέσα στο πρόγραμμα. Η συνάρτηση που χρησιμοποιούμε είναι η Bval.m .

Τέλος, γράφουμε στο κώδικα το αποτέλεσμα του υπολογισμού που θέλουμε να μας εμφανίζεται στο πεδίο TextArea που έχουμε προσθέσει . Προσοχή στην εμφάνιση μηνύματος η εντολή fprintf δεν λειτουργεί σωστά και χρησιμοποιούμε την sprint

```

% Button down function: UIAxes
function UIAxesButtonDown(app, event)
% Load in variables and define some constants.
M=load(uigetfile);

Mc=1.3; % See scrip_Mc.m for the determination of 1.3 as the Mc.
dM=0.05;
GREY=[0.85,0.85,0.85];

% GR-FMD fit parameters from function call.
[b, b_err, a, R2_b,~,Mgr,Ngr,ngr]=Bval(M, Mc,dM);

% Define some values for plotting.
po=[-b,a];
Mgr_fit=Mc:dM:max(M);
Ngr_fit=10.^polyval(po,Mgr_fit);

% Output fit statistics to user prompt.
answer = sprintf('b-value fit: %0.2f ± %0.2f (R² %0.3f)\n',b,b_err, R2_b);
app.TextArea.Value = answer;

% Plot GR-FMD and fit.
semilogy(app.UIAxes,Mgr, Ngr, 'o', 'Color', 'k');
hold(app.UIAxes, 'on')
bar(app.UIAxes,Mgr,ngr, 'FaceColor', GREY);
semilogy(app.UIAxes,Mgr_fit, Ngr_fit, '-', 'Color', 'black');

end
end

```

Εικόνα 3.17.Κώδικας υπολογισμού κατανομής Gutenberg-Richter και b-value

```

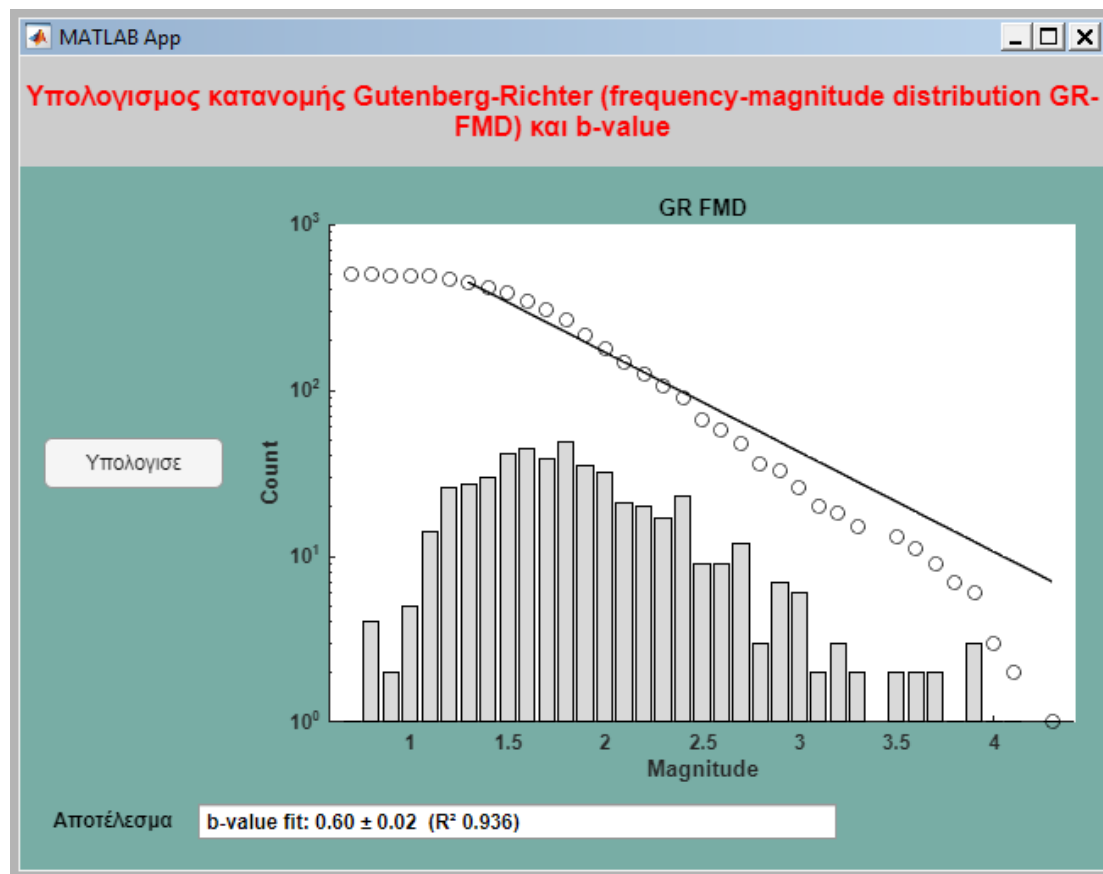
17 function results = Bval(app,M_list,Mc,dM)
18 % Ensure that input is a row vector.
19 if(iscolumn(M_list))
20     M_list=M_list';
21 end
22
23 % Check to see there is enough input data points.
24 if( length(M_list)<=3 )
25     b=NaN; b_err=NaN; R2=NaN; X2=NaN; L=NaN; a=NaN; Mbins=[]; Ncum=[];
26     fprintf('Error: Insufficient input data.\n');
27     return;
28 end
29
30 % Define magnitude bin widths based on the on input error value (dM).
31 Mbins=min(M_list)-dM:dM/2:max(M_list)+dM;
32 [Nbin,~]=histcounts(M_list, Mbins);
33
34 % Get cumulative number of events, but only keep unique bins.
35 Ncum=fliplr(cumsum(fliplr(Nbin)));
36 diffN=[diff(Ncum),0];
37 Ncum=Ncum(diffN~=0);
38 Mbins=Mbins(diffN~=0);
39 Nbin=Nbin(diffN~=0);
40 Ntot=max(Ncum(Mbins>=Mc));
41
42 % Define axis for parameter fitting.
43 logN=log10(Ncum);
44 Y=logN(Mbins>=Mc);
45
46 X=Mbins(Mbins>=Mc);
47 % Ensure there is still enough data to fit.
48 if( length(Y)<=3 )
49     b=NaN; b_err=NaN; R2=NaN; X2=NaN; L=NaN; a=NaN; Mbins=[]; Ncum=[];
50     fprintf('Error: Insufficient data after magnitude truncation.\n');
51     return;
52 end
53 % Find the mean and standard deviation of the magnitude value.
54 temp=[abs(diff(10.^Y)), 10^Y(end)];
55 M_avg=sum( X.*temp )/Ntot;
56 M_std=sqrt( sum((X.^2).*temp)/Ntot - M_avg^2 );
57
58 % Determine most likely b-value, its error, and the a-value.
59 b=1/( log(10)*(M_avg-(Mc-dM/4)) );
60 b_err=2.3*M_std*b^2/sqrt(Ntot);
61 w=(Nbin(Mbins>=Mc)); w=w/sum(w);
62 a=sum((b*X+Y).*w);
63
64 % Compute goodness-of-fit statistics (in linear space).
65 po=[-b, a];
66 Y=10.^Y;
67 Yfit=10.^polyval(po,X);
68 Ybar=sum(Y.*w);
69 SSres=sum(w.*((Y-Yfit).^2));
70 SSres=sum(w.*((Y-Ybar).^2));
71 R2=1-(SSres/SSres);
72 X2=SSres/(b_err);
73 return

```

Εικόνα 3.18.Κώδικας συνάρτησης Bval

3.2.4. Αποτελέσματα

Τρέχουμε την εφαρμογή και βλέπουμε ότι δουλεύει σωστά η εμφάνιση της κατανομής καθώς και ο το αποτέλεσμα του υπολογισμού ότι εμφανίζεται στο πεδίο του Αποτελέσματος.



Εικόνα 3.19. Διάγραμμα κατανομής Gutenberg-Richter και b-value

Στα αποτελέσματα που λαμβάνουμε περιλαμβάνονται όλες οι πληροφορίες που απαιτούνται για να έχουμε μια ολοκληρωμένη εκτίμηση της σεισμικότητας με χρήση GR FMD. Ειδικότερα, λαμβάνουμε

- ✓ Την βέλτιστη ευθεία παλινδρόμησης (μαύρη συμπαγής γραμμή) για τον σεισμικό κατάλογο. Η κλίση της ευθείας είναι η b-value η τιμή της οποίας εμφανίζεται στο πεδίο «Αποτέλεσμα»
- ✓ Την αθροιστική κατανομή (ανοιχτοί κύκλοι) και την κατανομή ανά μέγεθος (μπάρες)
- ✓ Το εκτιμώμενο μέγεθος πληρότητας (magnitude of completeness) του καταλόγου που αντιστοιχεί στην αρχή της βέλτιστης ευθείας παλινδρόμησης

Σε αυτό το παράθυρο του διαγράμματος υπάρχει η δυνατότητα από χρήστη να :

- Επιλέξει συγκεκριμένα σημεία πάνω στις γραφικές παραστάσεις
- Μεγέθυνση και σμίκρυνση στο διάγραμμα(Zoon In , Zoom Out)
- Αποθηκεύσει το διάγραμμα σε αρχείο μορφής εικόνων . Επιλέγοντας στο παράθυρο τη καρτέλα File->Save as->JPEG ή TIF ή οποιασδήποτε άλλης μορφής εικόνων .
- Δυνατότητα αντιγραφής αποτελέσματος επιλέγοντας το και πατώντας Ctrl+C και σε μορφή κειμένου .

3.2.5. Δημιουργία διαδικτυακής εφαρμογής

Ανοίγοντας το MATLAB πληκτρολογούμε “WebAppCompiler” στο Command window και ανοίγει το Matlab Compiler . Σε αυτή τη καρτέλα εμφανίζεται ένα παράθυρο .

Στην επιλογή Add app file προσθέτουμε το αρχείο του σχεδιασμού εφαρμογής που δημιουργούμε στο App Designer και είναι σε μορφή .mlapp

Όταν φορτώνουμε το αρχείο της εφαρμογής που θα δημιουργήσουμε στο App Designer , στην επιλογή Files required for your App to run εμφανίζονται αυτόματα τα εξωτερικά αρχεία που χρησιμοποιούνται για να τρέξει εφαρμογή όπως functions του MATLAB . Στη περίπτωση μας δε χρησιμοποιούμε εξωτερικές συναρτήσεις ή κώδικες που καλούμε γιατί τα έχουμε βάλει όλα μεστό κώδικα της εφαρμογής .

Η διαδικασία που ακολουθείται στο εργαλείο Web App Compiler θα είναι ίδια σε κάθε εφαρμογή που θα δημιουργήσουμε.

Έπειτα , επιλέγοντας το Package γίνεται η δημιουργία της διαδικτυακής εφαρμογής και δημιουργείται ένα αρχείο .ctf file . Μας ζητείται να πληκτρολογήσουμε το σύνδεσμο της ιστοσελίδας που θα δημοσιευτεί η εφαρμογή μας . Πατώντας το deploy to server δημοσιεύεται η εφαρμογή στο διαδίκτυο .

3.3. Υπολογισμός μεγέθους πληρότητας (magnitude of completeness - Mc) και b-value από σεισμολογικούς καταλόγους με χρήση Gutenberg-Richter FMD

Σε αυτή την εφαρμογή θέλουμε να δημιουργήσουμε ένα αρχείο .mlapp στο AppDesigner , το οποίο δέχεται ένα αρχείο .txt από χρήστη που περιέχει σεισμολογικά δεδομένα. Σκοπός της εφαρμογής είναι ο υπολογισμός του μεγέθους

πληρότητας (magnitude of completeness - M_c) και b-value από σεισμολογικούς καταλόγους με χρήση Gutenberg-Richter FMD και η εμφάνιση διαγραμμάτων τους .

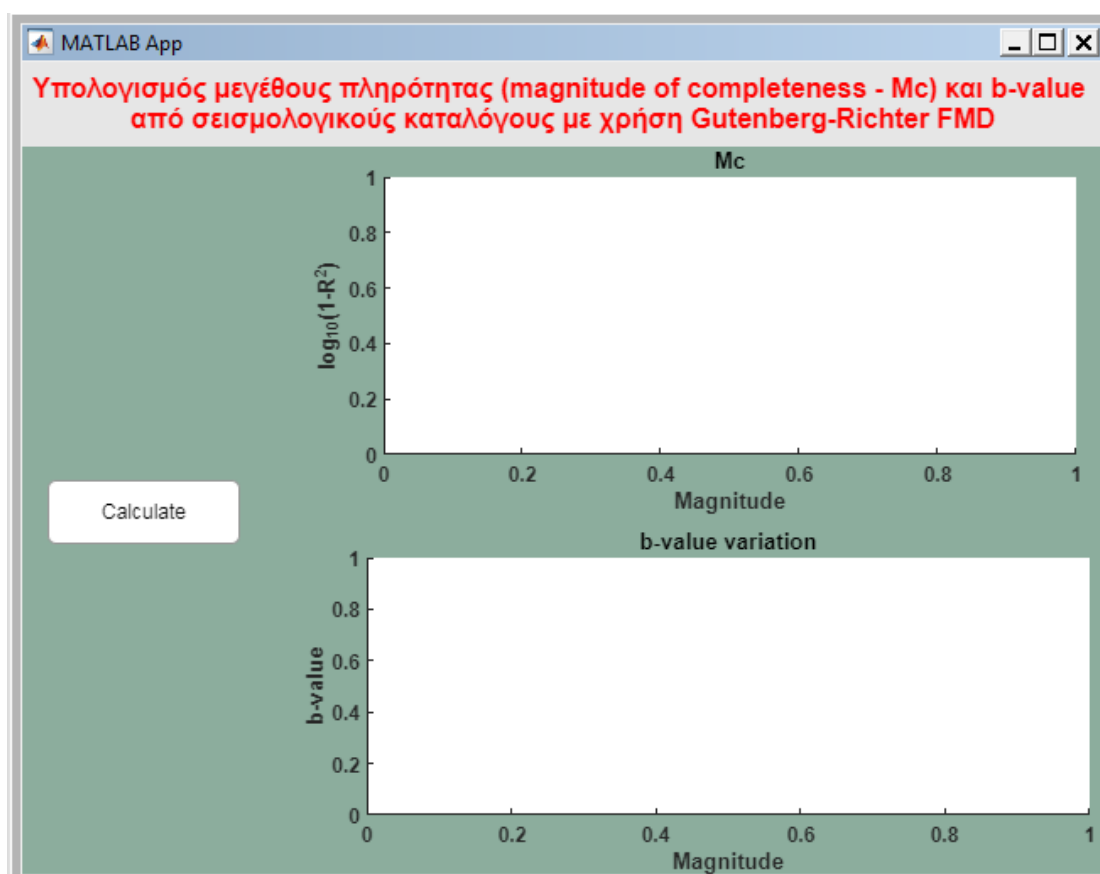
3.3.1. Σχεδιασμός της εφαρμογής στο App Designer

Ο σχεδιασμός της εφαρμογής ακολουθεί την ίδια ροή με τις προηγούμενες εφαρμογές που δημιουργήθηκαν . Σε αυτήν την εφαρμογή θα χρειαστούν 2 διαγράμματα, ένα για το μέγεθος πληρότητας και ένα b-value από σεισμολογικούς καταλόγους με χρήση Gutenberg-Richter FMD.

Έτσι , εισάγουμε

- Δύο κενά διαγράμματα (Axes) από component Library
- ένα πεδίο (Label) από Component Library με το τίτλο της εφαρμογής
- ένα κουμπί “Calculate” (τύπου button) από Component Library

Ρυθμίζουμε τις ετικέτες των αξόνων των δύο διαγραμμάτων καθώς και το τίτλο τους , όπως και το φόντο της εφαρμογής μας και το χρώμα και το στυλ γραμματοσειράς του πεδίου(Label) με το τίτλο της εφαρμογής.



Εικόνα 3.20. Οπτική προβολή της 3^{ης} εφαρμογής στο App Designer

3.3.2. Προγραμματισμός της εφαρμογής στο App Designer

Προχωράμε στο προγραμματιστικό μέρος της εφαρμογής , πατώντας δεξιά κλικ πάνω στο πεδίο `app.Button` με ετικέτα “Calculate” στο `Component Browser` και επιλέγοντας `Callbacks->Add ButtonPushedFnc callback` ενεργοποιούμε την ιδιότητα “Callback “.

Γράφουμε το κώδικα για την ενέργεια που θα κάνει το κουμπί “Calculate” Χρησιμοποιούμε τον ίδιο τρόπο στο κώδικα για να ζητήσουμε απτο χρήστη να επιλέξει απτον υπολογιστή του ένα αρχείου δεδομένων τύπου `.txt`. και συνεχίζουμε στις εντολές του κώδικα για τον υπολογισμό και την εμφάνιση του μεγέθους πληρότητας και `b-value` από σεισμολογικούς καταλόγους με χρήση `Gutenberg-Richter FMD`.

Στο κώδικα καλείται μια εξωτερική συνάρτηση (`function`) που πρέπει να την προσθέσουμε στο κώδικα μας ,είναι η ίδια που χρησιμοποιήσαμε και για τον υπολογισμό κατανομής στη προηγούμενη εφαρμογή. Επιλέγουμε στο `Code Browser` τα `functions` , τις συναρτήσεις και προσθέτουμε μία `private function` . Οι συναρτήσεις δημιουργούνται πριν από κάθε ενέργεια στο κώδικα έτσι ώστε να μπορούμε να τις καλούμε οποιαδήποτε στιγμή μέσα στο πρόγραμμα. Η συνάρτηση που χρησιμοποιούμε είναι η `Bval.m` .

Τέλος , γράφουμε το κύριο κώδικα της λειτουργίας “Calculate” .Ο κώδικας της συνάρτησης είναι ακριβός ίδιος με αυτόν του υπολογισμού κατανομής της προηγούμενης εφαρμογής.


```

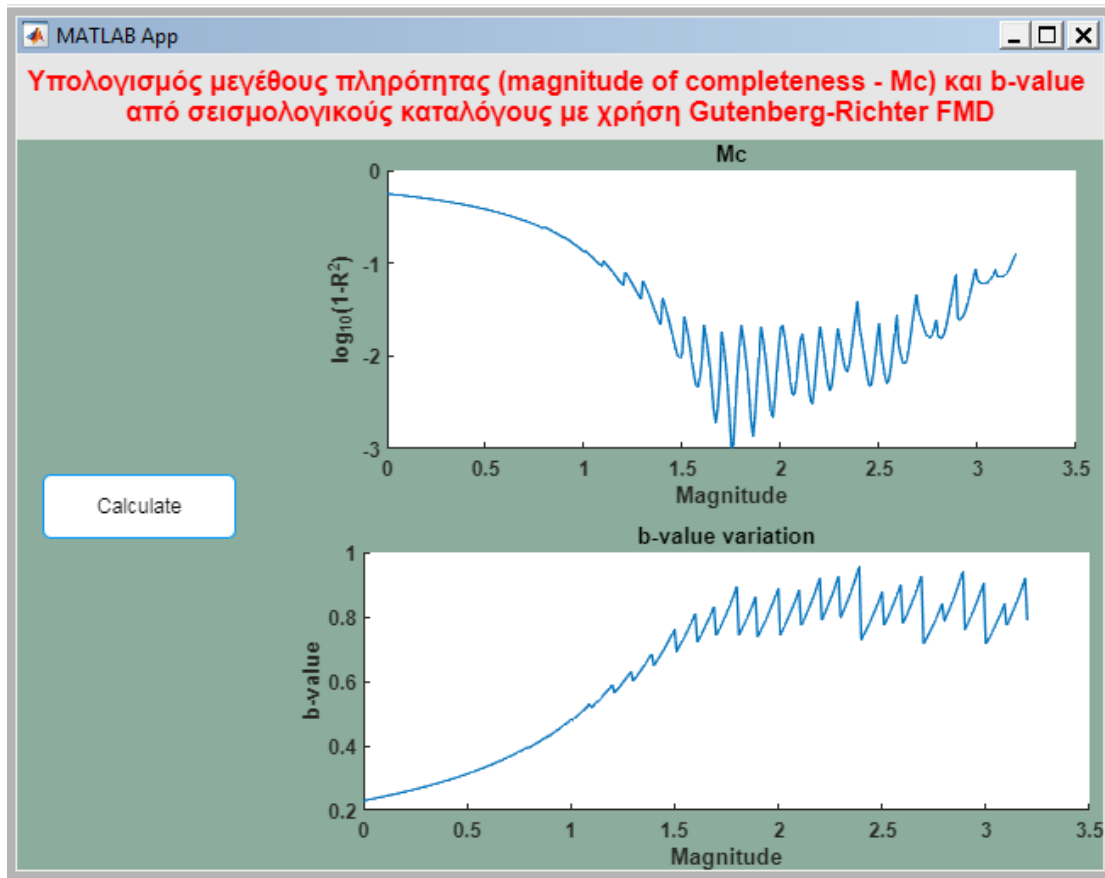
81
82     % Button pushed function: CalculateButton
83     function CalculateButtonPushed(app, event)
84
85         M=load(uigetfile);
86         Mc_list=0:0.01:3.2;
87         dM=0.05;
88         GREY=[0.85,0.85,0.85];|
89         % Preallocate space in memory.
90         R=zeros(size(Mc_list));
91         X=R;
92         b=R;
93         % Loop through every Mc truncation value and fit GR-FMD.
94         for i=1:length(Mc_list)
95             [temp,~,~,temp2, temp3, ~,~,~]=Bval(M, Mc_list(i), dM);
96             b(i)=temp;
97             R(i)=temp2;
98             X(i)=temp3;
99         end
100        % Some error handling.
101        R(R>1)=NaN;
102        R(R<0)=NaN;
103
104        % Plot GR-FMD fit statistics, visually determine fit stability and Mc.
105        plot(app.UIAxes,Mc_list, log10(1-R));
106        plot(app.UIAxes2,Mc_list, b);
107        end
108    end
109

```

Εικόνα 3.21. Κώδικας υπολογισμού μεγέθους πληρότητας και ένα b-value από σεισμολογικούς καταλόγους με χρήση Gutenberg-Richter FMD

3.3.3. Αποτελέσματα

Τρέχουμε την εφαρμογή και βλέπουμε ότι δουλεύει σωστά ο υπολογισμός του μεγέθους πληρότητας και η εμφάνιση των αντίστοιχων διαγραμμάτων.



Εικόνα 3.22. Διάγραμματα μεγέθους πληρότητας και b-value από σεισμολογικούς καταλόγους με χρήση Gutenberg-Richter FMD

Ακολουθώντας τη μέθοδο προσδιορισμού των Wiemer & Wyss [6] τα αποτελέσματα που λαμβάνουμε απεικονίζονται σε δυο διαγράμματα:

Στο άνω διάγραμμα έχουμε τη γραφική παράσταση της συνάρτησης $\log_{10}(1-R^2)$ συναρτήσει μεγέθους καταλόγου. Εδώ αναζητούμε το ελάχιστο της συνάρτησης το οποίο ταυτίζεται αριθμητικά με την τιμή του μεγεθους πληρότητας (Mc). Στην περίπτωση μας έχουμε $Mc=1.65$

Στο κάτω διάγραμμα απεικονίζεται η μεταβολή της b-value συναρτήσει του μεγέθους. Η τιμή b-value βρίσκεται για $x=Mc$ όπως υπολογίστηκε από το προηγούμενο διάγραμμα. Στην περίπτωση μας είναι $Mc=1.65$ άρα $b=0.82$

Σε αυτό το παράθυρο του διαγράμματος υπάρχει η δυνατότητα από χρήστη να :

- Επιλέξει συγκεκριμένα σημεία πάνω στις γραφικές παραστάσεις
- Μεγέθυνση και σμίκρυνση στο διάγραμμα(Zoon In , Zoom Out)
- Αποθηκεύσει το διάγραμμα σε αρχείο μορφής εικόνων . Επιλέγοντας στο παράθυρο τη καρτέλα File->Save as->JPEG ή TIF ή οποιασδήποτε άλλης μορφής εικόνων .

Για τη δημοσίευση της εφαρμογής στο διαδίκτυο χρησιμοποιούμε το εργαλείο Compiler ακριβώς με τον ίδιο τρόπο που χρησιμοποιήθηκε και για τις 2 πλοηγούμενες εφαρμογές

4. Συμπεράσματα

Στην παρούσα διπλωματική επιχειρήσαμε μια ολοκληρωμένη προσέγγιση ανάπτυξης διαδικτυακών εφαρμογών (webapps) του προγραμματιστικού περιβάλλοντος matlab. Δεδομένου ότι πρόκειται για ένα εργαλείο που αν και προσφέρει πλήθος δυνατοτήτων για τον σύγχρονο μηχανικό, έχει μια τιμή που το καθιστά μη αποκτήσιμο από όλους, είναι επιτακτική η ανάγκη της εύρεσης ενός τρόπου συνεργατικής αξιοποίησης του. Με τη χρήση των webapps έχουμε τη δυνατότητα να προσφέρουμε σε χρήστες που χρειάζονται μόνο τα αποτελέσματα υπολογισμών, τη δυνατότητα να τα έχουν, χρησιμοποιώντας εφαρμογές που αναπτύσσονται για διαδικτυακό περιβάλλον. Με τον τρόπο αυτό, κάθε χρήστης με απλή πρόσβαση στην ιστοσελίδα ενός αδειοδοτημένου matlab web server μπορεί να εκτελεί matlab εφαρμογές σε γραφικό περιβάλλον, με προφανή τα οφέλη τόσο όσον αφορά την ταχύτητα υπολογισμών όσο και την ανυπαρξία ανάγκης εγκατάστασης οποιοδήποτε πακέτου. Η μόνη απαίτηση από το χρήστη είναι να διαθέτει το αρχείο δεδομένων του σε format που απαιτεί η συγκεκριμένη εφαρμογή αλλά αυτό είναι κάτι που πολύ εύκολα καθορίζεται κατά περίπτωση.

Για τις ανάγκες της παρούσας διπλωματικής υλοποιήσαμε τρεις διαδικτυακές εφαρμογές matlab που αφορούν υπολογισμούς σε σεισμολογικούς καταλόγους:

- ✓ Υπολογισμό του κανονικοποιημένου φάσματος σεισμικού καταλόγου με βάση τη θεωρία του «φυσικού χρόνου»
- ✓ Υπολογισμός της κατανομής Gutenberg – richter και προσδιορισμός της b-value
- ✓ Υπολογισμός του μεγέθους πληρότητας (Mc) και τη b-value σύμφωνα με της μέθοδο Wiemer & Wyss [6]

Από την εμπειρία που αποκτήθηκε κατά τη διάρκεια της υλοποίησης της διπλωματικής μπορούμε να συνοψίσουμε στα παρακάτω κομβικά σημεία:

- Οι Web Apps αποτελούν ένα εύχρηστο εργαλείο για την εκτέλεση γραφικών εφαρμογών matlab σε διαδικτυακό περιβάλλον
- ΔΕΝ χρειάζεται άδεια χρήσης ο χρήστης – μόνο στον server
- Παρέχεται η δυνατότητα εκτέλεσης υπολογισμών σε matlab ακόμα και από κινητό/tablet
- Υπάρχει περιορισμός στα αναδυόμενα παράθυρα που πρακτικά σημαίνει ότι τα πάντα πρέπει να εμφανίζονται σε μία οθόνη
- Το subplotting δεν υποστηρίζεται ακόμα
- Υπάρχει έλλειψη documentation σχετικά με τα δικαιώματα χρήσης που αποκτούν οι εντολές I/O (π.χ textscan, fopen κ.λ.π)

Συνοψίζοντας μπορούμε να καταλήξουμε, ότι ακολουθώντας τις επιταγές τη σύγχρονης προγραμματιστικής υλοποίησης σε περιβάλλοντα cloud έχουμε τη δυνατότητα να το εφαρμόσουμε και για εφαρμογές επιστημονικού/τεχνικού ενδιαφέροντος με χρήση matlab. Όμως, το επίπεδο ωριμότητας του περιβάλλοντος για web development υπολείπεται ακόμα (δεν υποστηρίζονται όλες οι δυνατότητες του matlab), γεγονός όμως που σίγουρα θα αλλάξει σε επόμενες εκδόσεις

Βιβλιογραφία

- [1] MATLAB App Building Documentation
https://www.mathworks.com/help/pdf_doc/matlab/creating_guis.pdf
- [2] <https://www.mathworks.com/help/compiler/web-apps.html>
- [3] <https://www.mathworks.com/help/compiler/webapps/create-and-deploy-a-web-app.html>
- [4] https://www.cs.uoi.gr/~pvassil/supervision/diplomatikes/2021-2022/extraMaterial/Seismo_GutenbergRichter/2020_Gutenberg%E2%80%93Richter_law_Wikipedia.pdf
- [5] <https://bbnet.gein.noa.gr/HL/>
- [6] Wiemer & Wyss (2000). Bulletin of Seismological Society of America. 90, pp. 859-869
- [7] P. A. Varotsos, N. V. Sarlis, and E. S. Skordas, "Long-range correlations in the electric signals that precede rupture," *Phys Rev E Stat Nonlin Soft Matter Phys*, vol. 66, no. 1 Pt 1, p. 011902, 2002, doi: 10.1103/PhysRevE.66.011902.
- [8] P. A. Varotsos, N. V. Sarlis, and E. S. Skordas, *Natural Time Analysis: The New View of Time: Precursory Seismic Electric Signals, Earthquakes and Other Complex Time Series*. 2011.
- [9] P. A. Varotsos, N. V. Sarlis, and E. S. Skordas, "Spatio-temporal complexity aspects on the interrelation between seismic electric signals and seismicity," *Practica of Athens Academy*, vol. 76, pp. 294-231, 2001.
- [10] N. V. Sarlis, E. S. Skordas, and P. A. Varotsos, "Similarity of fluctuations in systems exhibiting Self-Organized Criticality," *EPL (Europhysics Letters)*, vol. 96, no. 2, p. 28006, 2011, doi: 10.1209/0295-5075/96/28006.
- [11] P. Varotsos, N. V. Sarlis, E. S. Skordas, S. Uyeda, and M. Kamogawa, "Natural time analysis of critical phenomena," *Proceedings of the National Academy of Sciences*, vol. 108, no. 28, pp. 11361-11364, 2011, doi: 10.1073/pnas.1108138108.
- [12] Z. Olami, H. J. S. Feder, and K. Christensen, "Self-organized criticality in a continuous, nonconservative cellular automaton modeling earthquakes," *Physical Review Letters*, vol. 68, no. 8, pp. 1244-1247, 1992, doi: 10.1103/PhysRevLett.68.1244.
- [13] P. Varotsos, N. Sarlis and E. Skordas, *Practica of Athens Academy* 76, 294-321, 2001; *Phys. Rev. E* 66, 011902, 2002).
- [14] *Phys. Rev. Lett.* 94, 170601 (2005)
- [15] P. Varotsos et al., *Phys. Rev. E* 70, 011106 (2004), P. Varotsos et al., *Appl. Phys. Lett.* 91, 064106 (2007)] και η Σεισμολογία [P. Varotsos et al., *Phys. Rev. E* 72, 041103 (2005); *Phys. Rev. E* 73, 031114 (2006); S. Uyeda et al., *J. Geophys. Res.* 114, B02310 (2009)

- [16] “Διαδικτυακή εφαρμογή καταγραφής και διεκπεραίωσης ιατρικών εξετάσεων σε μονάδες ιατρικής φροντίδας και περιθαλψης” .[Μεταπτυχιακή Διατριβή, Ιούλιος 2012, Γεώργιος Ε. Λιτίνας]