



ΔΙΕΘΝΕΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΗΣ ΕΛΛΑΔΟΣ

**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ**  
**ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

Εφαρμογή ανάγνωσης εξίσωσης ή συστήματος  
εξισώσεων και επίλυσής του με μεθόδους  
Αριθμητικής Ανάλυσης



Του φοιτητή  
Πουρσανίδη Σπυρίδων  
Αρ. Μητρώου: 123920

Επιβλέπων  
Γουλιάνας Κωνσταντίνος  
Αναπληρωτής Καθηγητής

Ημερομηνία 06/04/2020

Τίτλος Π.Ε.: Εφαρμογή ανάγνωσης εξίσωσης ή συστήματος εξισώσεων και επίλυσής του με μεθόδους  
Αριθμητικής Ανάλυσης  
Κωδικός Π.Ε.: 19049

Όνοματεπώνυμο φοιτητή: Πουρσανίδης Σπυρίδων

Όνοματεπώνυμο εισηγητή: Γουλιάνας Κωνσταντίνος

Ημερομηνία ανάληψης Π.Ε.: 24/11/2019

Ημερομηνία περάτωσης Π.Ε. 15/06/2020

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Πουρσανίδη Σπυρίδων που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητα και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

# ΠΡΟΛΟΓΟΣ

Η τεχνολογία εξελίσσεται συνεχώς στις μέρες μας και έχει μπει πλέον για τα καλά στις ζωές μας. Όλο και περισσότεροι άνθρωποι κάθε ηλικίας χρησιμοποιούν καθημερινά τεχνολογικές συσκευές, κυρίως smartphones, που διευκολύνουν την επίτευξη απλών καθημερινών λειτουργιών, καθώς και πιο πολύπλοκων. Για να έρθουν σε επαφή με άλλους ανθρώπους, με τους οποίους ενδεχομένως να βρίσκονται μακριά και να μην υπάρχει η δυνατότητα άμεσης επικοινωνίας. Για να βρουν πληροφορίες που χρειάζονται, όπως χρήσιμα τηλέφωνα, διευθύνσεις, μαγειρικές συνταγές. Για να περάσουν την ώρα τους με παιχνίδια, μουσική ή βίντεο. Για το λόγο αυτό, παρατηρούμε διαρκώς καινούργιες εφαρμογές ή αναβαθμίσεις των ήδη υπάρχοντων, προκειμένου να μπορέσουν να καλύψουν όλες αυτές τις ανάγκες των ανθρώπων. Το περισσότερο διαδεδομένο λειτουργικό σύστημα που φιλοξενεί τέτοιου είδους εφαρμογές είναι το Android.

Το Android είναι ένα λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας, το οποίο τρέχει τον πυρήνα του Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους προγραμματιστές να συνθέτουν κώδικα με τη χρήση της Java, με τη βοήθεια βιβλιοθηκών ανεπτυγμένων από την Google. Η δημιουργία του ξεκίνησε στο Palo Alto της Καλιφόρνια τον Οκτώβριο του 2003 από τους Andy Rubin, Nick Sears και Chris White, ενώ η πρώτη παρουσίαση της πλατφόρμας έγινε στις 5 Νοεμβρίου 2007. Από τότε έχει ακολουθήσει μια σειρά αναβαθμίσεων καθώς και νέων εκδόσεων, ξεκινώντας από τις 15 Σεπτεμβρίου 2009 με την έκδοση 1.6 (Donut) και φτάνοντας μέχρι τις 3 Σεπτεμβρίου 2019 με την έκδοση 10, την πρώτη έκδοση που δεν έχει ως κωδική ονομασία κάποιο γλυκό. Στις 19 Φεβρουαρίου 2020 κυκλοφόρησε σε beta η έκδοση 11.

Το Google Play, παλαιότερα γνωστό ως Android Market, είναι μία ψηφιακή εφαρμογή διανομής που ανέπτυξε η Google. Λειτουργεί ως το επίσημο ηλεκτρονικό κατάστημα για συσκευές Android, επιτρέποντας στους χρήστες να αναζητήσουν και να κατεβάσουν εφαρμογές που αναπτύχθηκαν με το SDK του Android, καθώς και βιβλία, μουσική, ταινίες και σειρές. Περιέχει περισσότερες από 3.5 εκατομμύρια εφαρμογές, με χρήστες σε περισσότερες από 145 χώρες. Κυκλοφόρησε στις 22 Οκτωβρίου 2008 με το όνομα Android Market και αργότερα, τον Μάρτιο του 2012 μετονομάστηκε σε Google Play.[1]

# ΠΕΡΙΛΗΨΗ

Η εργασία αυτή έχει ως στόχο να κάνει ευκολότερη την εκμάθηση των Μαθηματικών. Αναφέρεται κυρίως σε μαθητές, αλλά και σε γονείς μαθητών, που θέλουν να βοηθήσουν τα παιδιά τους, αλλά δεν έχουν τις απαραίτητες γνώσεις. Θα ήταν χρήσιμη επίσης και σε φοιτητές που ασχολούνται με μεθόδους Αριθμητικής Ανάλυσης, προκειμένου να μάθουν πως χρησιμοποιούνται, καθώς και να λύνουν τα προβλήματα που τις απαιτούν γρηγορότερα. Πρόκειται για μια εφαρμογή σε Android, η οποία επιλύει μαθηματικές παραστάσεις, εξισώσεις πρώτου και δευτέρου βαθμού, συστήματα πινάκων και γραμμικά συστήματα.

Η εφαρμογή θα μπορούσε να χωριστεί σε 3 τρία κομμάτια. Αρχικά, δέχεται σαν είσοδο, είτε χειρόγραφα είτε σε μορφή φωτογραφίας, την μαθηματική παράσταση, την εξίσωση ή τον πίνακα που θέλουμε να επιλύσουμε. Σε περίπτωση που επιλέξουμε φωτογραφία, μας δίνεται η δυνατότητα να φορτώσουμε μια φωτογραφία από τη συλλογή, ή να τραβήξουμε μια καινούργια με την κάμερα του κινητού. Στη συνέχεια, επιλέγουμε τη μέθοδο που θέλουμε να χρησιμοποιήσουμε από τη λίστα με τις διαθέσιμες μεθόδους επίλυσης. Τέλος εμφανίζονται τα αναλυτικά βήματα επίλυσης, καθώς και το αποτέλεσμα.

Κλείνοντας, η εφαρμογή έχει δημιουργηθεί με τέτοιο τρόπο ώστε να μην ενθαρρύνει τη χρήση της ως ένα μέσο για να αντιγράψει ένας μαθητής ή για σκονάκι. Αντιθέτως, παρουσιάζει αναλυτικά τα βήματα επίλυσης, ώστε ο χρήστης, είτε αυτός είναι ένας μαθητής, ένας φοιτητής είτε ένας γονέας που προσπαθεί να βοηθήσει το παιδί του, να μαθαίνει μέσα από αυτή και να έχει μετά από μερικές χρήσεις της εφαρμογής τη δυνατότητα να είναι ανεξάρτητος. Να μπορεί να λύσει τα εκάστοτε προβλήματα που του παρουσιάζονται, χωρίς να απαιτείται η χρήση της.

# ABSTRACT

The goal of this project is to make Math easier to learn. It mostly concerns students, and also parents, which want to help their children, but don't have the required knowledge. It would also be useful to university students who use Numerical Analysis methods, in order to get to know them and also solve any mathematical problems that come up faster. It is an android application, which can solve first and second degree mathematical equations, tables and linear systems.

We can divide the application in three main parts. First of all, it receives as an input, either manually or in the form of a photograph, the mathematical equation or the table we want to solve. In case we choose to input a photograph, we can either load an existing one from the gallery or take a new one, using the phone's camera. Next, we choose the method we would like to use from a list which contains the ones available. Finally, all the necessary steps and the result are printed on the screen.

In conclusion, the application is created in such a way, so it doesn't encourage the student to copy or cheat. On the contrary, it provides all the required steps, so the user, either the student, the university student or the parent who wants to help his child, can learn through the application and become independent after a few uses of the application. To be able to solve all the possible problems that can come up, without the help of the application.

# ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	3
ΠΕΡΙΛΗΨΗ	4
ABSTRACT	5
ΠΕΡΙΕΧΟΜΕΝΑ	6
ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ	8
ΕΙΣΑΓΩΓΗ	10
1. ΜΑΘΗΜΑΤΙΚΕΣ ΜΕΘΟΔΟΙ	11
1.1 ΕΙΣΑΓΩΓΗ	11
1.2 ΕΞΙΣΩΣΕΙΣ	11
1.3 ΓΡΑΜΜΙΚΑ ΣΥΣΤΗΜΑΤΑ	12
1.3.1 Άμεσες Μέθοδοι	13
1.3.1.1 Διαγώνια Συστήματα	13
1.3.1.2 Κάτω Τριγωνικά Συστήματα	13
1.3.1.3 Άνω Τριγωνικά Συστήματα	14
1.3.1.4 Απαλοιφή Gauss με μερική οδήγηση	14
1.3.2 Επαναληπτικές Μέθοδοι Επίλυσης Γραμμικών Συστημάτων	15
1.3.2.1 Μέθοδος Jacobi	15
1.3.2.2 Μέθοδος Gauss - Seidel	16
1.3.2.3 Σύγκριση Μεθόδων Jacobi και Gauss - Seidel	17
1.4 ΑΡΙΘΜΗΤΙΚΗ ΕΠΙΛΥΣΗ ΕΞΙΣΩΣΕΩΝ	17
1.4.1 Μέθοδος της Διχοτόμησης	18
1.4.2 Μέθοδος της Εσφαλμένης Θέσης	19
1.4.3 Μέθοδος Newton - Raphson	20
1.5 ΕΠΙΛΟΓΟΣ	23
2. ΤΕΧΝΟΛΟΓΙΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ	24
2.1 ΕΙΣΑΓΩΓΗ	24
2.2 ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	24
2.2.1 Java	24
2.2.2 XML	26
2.3 ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ	28
2.3.1 Android Studio	28
2.4 ΒΙΒΛΙΟΘΗΚΕΣ	29

2.4.1 Android Image Cropper	29
2.4.2 Google Vision API	29
2.5 ΕΠΙΛΟΓΟΣ	29
3. ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	30
3.1 ΕΙΣΑΓΩΓΗ	30
3.2 ΓΕΝΙΚΑ	30
3.3 ΜΕΝΟΥ ΕΠΙΛΟΓΗΣ ΜΕΘΟΔΟΥ	36
3.3.1 Κενή Επιλογή	36
3.3.2 Gauss	49
3.3.3 Jacobi και Gauss - Seidel	51
3.3.4 Διχοτόμηση και Εσφαλμένη Θέση	53
3.3.5 Newton - Raphson	56
3.4 ΕΠΙΛΟΓΟΣ	58
ΠΡΟΤΑΣΕΙΣ ΒΕΛΤΙΩΣΗΣ	59
ΑΝΑΦΟΡΕΣ	60
ΒΙΒΛΙΟΓΡΑΦΙΑ	61

# ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

Σχήμα 1.1 Πρώτη περίπτωση αποτυχίας σύγκλισης Newton - Raphson	21
Σχήμα 1.2 Δεύτερη περίπτωση αποτυχίας σύγκλισης Newton - Raphson	22
Σχήμα 1.3 Τρίτη περίπτωση αποτυχίας σύγκλισης Newton - Raphson	22
Σχήμα 2.1 Λογότυπο της Java	24
Σχήμα 2.2 Η μέθοδος "Hello World"	26
Σχήμα 2.3 Παράδειγμα XML	27
Σχήμα 2.4 Περιβάλλον Android Studio	28
Σχήμα 3.1 Αρχική οθόνη της εφαρμογής	31
Σχήμα 3.2 Επιλογή μεθόδου επίλυσης	31
Σχήμα 3.3 Επιλογή τρόπου εισαγωγής φωτογραφίας	33
Σχήμα 3.4 Τράβηγμα καινούργιας φωτογραφίας και μενού επιλογής	33
Σχήμα 3.5 Περικοπή φωτογραφίας	34
Σχήμα 3.6 Αποτέλεσμα εισαγωγής φωτογραφίας - Μετατροπή σε κείμενο	35
Σχήμα 3.7 Προεπισκόπηση περικοπής της φωτογραφίας	36
Σχήμα 3.8 Λανθασμένη είσοδος κενής επιλογής	37
Σχήμα 3.9 Αποτέλεσμα λανθασμένης εισόδου κενής επιλογής	38
Σχήμα 3.10 Είσοδος για τη λειτουργία "κομπιουτεράκι"	38
Σχήμα 3.11 Αποτέλεσμα λειτουργίας "κομπιουτεράκι"	39
Σχήμα 3.12 Είσοδος Διαγώνιου Συστήματος	40
Σχήμα 3.13 Αποτέλεσμα Διαγώνιου Συστήματος	40
Σχήμα 3.14 Είσοδος Κάτω Τριγωνικού Συστήματος	41
Σχήμα 3.15 Αποτέλεσμα Κάτω Τριγωνικού Συστήματος	41
Σχήμα 3.16 Είσοδος Άνω Τριγωνικού Συστήματος	42
Σχήμα 3.17 Αποτέλεσμα Άνω Τριγωνικού Συστήματος	42
Σχήμα 3.18 Είσοδος εξίσωσης πρώτου βαθμού με μία λύση	43
Σχήμα 3.19 Αποτέλεσμα εξίσωσης πρώτου βαθμού με μία λύση	44
Σχήμα 3.20 Είσοδος εξίσωσης πρώτου βαθμού με καμία λύση	44
Σχήμα 3.21 Αποτέλεσμα εξίσωσης πρώτου βαθμού με καμία λύση	45
Σχήμα 3.22 Είσοδος εξίσωσης πρώτου βαθμού με άπειρες λύσεις	45
Σχήμα 3.23 Αποτέλεσμα εξίσωσης πρώτου βαθμού με άπειρες λύσεις	46
Σχήμα 3.24 Είσοδος εξίσωσης δευτέρου βαθμού με μία λύση	46
Σχήμα 3.25 Αποτέλεσμα εξίσωσης δευτέρου βαθμού με μία λύση	47
Σχήμα 3.26 Είσοδος εξίσωσης δευτέρου βαθμού με δύο λύσεις	47
Σχήμα 3.27 Αποτέλεσμα εξίσωσης δευτέρου βαθμού με δύο λύσεις	48
Σχήμα 3.28 Είσοδος εξίσωσης δευτέρου βαθμού με καμία λύση	48
Σχήμα 3.29 Αποτέλεσμα εξίσωσης δευτέρου βαθμού με καμία λύση	49
Σχήμα 3.30 Είσοδος μεθόδου απαλοιφής Gauss με μερική οδήγηση	50
Σχήμα 3.31 Αποτέλεσμα μεθόδου απαλοιφής Gauss με μερική οδήγηση	50
Σχήμα 3.32 Είσοδος μεθόδου Jacobi	51
Σχήμα 3.33 Αποτέλεσμα μεθόδου Jacobi	52
Σχήμα 3.34 Είσοδος μεθόδου Gauss - Seidel	52



Σχήμα 3.35 Αποτέλεσμα μεθόδου Gauss - Seidel	53
Σχήμα 3.36 Είσοδος μεθόδου Διχοτόμησης	54
Σχήμα 3.37 Αποτέλεσμα μεθόδου Διχοτόμησης	54
Σχήμα 3.38 Είσοδος μεθόδου Εσφαλμένης Θέσης	55
Σχήμα 3.39 Αποτέλεσμα μεθόδου Εσφαλμένης Θέσης	55
Σχήμα 3.40 Αποτέλεσμα λανθασμένης εισόδου Διχοτόμησης - Εσφαλμένης Θέσης	56
Σχήμα 3.41 Είσοδος μεθόδου Newton - Raphson	57
Σχήμα 3.42 Αποτέλεσμα μεθόδου Newton - Raphson	57
Σχήμα 3.43 Αποτέλεσμα μεθόδου Newton - Raphson - Παράγωγος μηδέν	58

# ΕΙΣΑΓΩΓΗ

Με μοναδικό σκοπό την υποστήριξη και την παροχή βοήθειας στο χρήστη να πετύχει τους στόχους του και να μάθει με σωστό τρόπο, δημιουργήθηκε η εφαρμογή με τίτλο Μαθαίνω. Μια εύχρηστη και γρήγορη εφαρμογή, που δίνει όμως αυτή την ώθηση που χρειαζόμαστε μερικές φορές για να μάθουμε κάτι που αρχικά μας φαίνεται δύσκολο. Μέσα από ένα φιλικό προς τον χρήστη περιβάλλον, η διαδικασία της εκμάθησης γίνεται πάντα πιο εύκολη. Εύκολη είναι επίσης και η πρόσβαση, από τη στιγμή που ανά πάσα στιγμή μπορούμε να ανοίξουμε την εφαρμογή στο κινητό μας, που έχουμε πάντα μαζί μας.

Μέσα από την περιγραφή της εφαρμογής θα μάθουμε πως αλληλεπιδρούν τα Μαθηματικά με τις γλώσσες προγραμματισμού και τα περιβάλλοντα που χρησιμοποιήθηκαν, με τη βοήθεια βιβλιοθηκών. Το πώς γίνεται με μια απλή είσοδο, χειρόγραφη ή σε μορφή φωτογραφίας, να προκύπτουν τόσες πολλές πληροφορίες και πάνω από όλα, το επιθυμητό αποτέλεσμα. Όλα αυτά βέβαια, χωρίς να αποκλίνουμε από τον βασικό σκοπό της εφαρμογής, τη μάθηση.

Στη συνέχεια θα δούμε αναλυτικά τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής αυτής. Το περιβάλλον εργασίας, τις γλώσσες προγραμματισμού, τις βιβλιοθήκες και τις μεθόδους Αριθμητικής Ανάλυσης που χρησιμοποιήθηκαν. Θα αναφέρουμε επίσης συμπεράσματα και προτάσεις για τη βελτίωση της εφαρμογής.

# 1. ΜΑΘΗΜΑΤΙΚΕΣ ΜΕΘΟΔΟΙ

## 1.1 ΕΙΣΑΓΩΓΗ

Όπως αναφέραμε και προηγουμένως, η εφαρμογή δέχεται μια είσοδο και έπειτα, μέσα από την κατάλληλη μαθηματική μέθοδο βρίσκει και τυπώνει το αποτέλεσμα, καθώς και τα απαραίτητα βήματα για την επίλυση του προβλήματος. Στο κεφάλαιο αυτό θα δούμε αναλυτικά όλες τις μαθηματικές μεθόδους που χρησιμοποιεί η εφαρμογή, για να λύσει τα εκάστοτε προβλήματα.

## 1.2 ΕΞΙΣΩΣΕΙΣ

Εξίσωση στα μαθηματικά ονομάζεται κάθε ισότητα που συνδέει γνωστές ποσότητες με άγνωστες, τις οποίες θέλουμε να προσδιορίσουμε. Επομένως, πρόκειται για μια μαθηματική δήλωση που βεβαιώνει την ισότητα των δύο εκφράσεων. Για παράδειγμα:

$$x + 3 = 5 \quad (1.1)$$

Η εξίσωση αυτή βεβαιώνει πως το “ $x+3$ ” είναι ίσο με το 5. Το σύμβολο της ισότητας (=) επινοήθηκε από τον Ρόμπερτ Ρέκορντ (1512-1558), ο οποίος θεώρησε ότι τίποτα δε μπορεί να είναι πιο ίσο από τις παράλληλες ευθείες γραμμές με το ίδιο μήκος.

Λύση μια εξίσωσης με  $x$  αγνώστους είναι μια  $x$ -άδα (αριθμών, συναρτήσεων) που επαληθεύει την εξίσωση. Αντικαθιστώντας τους αγνώστους στην εξίσωση με το αντίστοιχο στοιχείο της  $x$ -άδας, η ισότητα γίνεται αληθής. Μια εξίσωση που δεν έχει καμία λύση ονομάζεται αδύνατη. Μια εξίσωση που έχει άπειρες λύσεις ονομάζεται ταυτότητα. Ο άγνωστος για τον οποίο η εξίσωση είναι αληθής ονομάζεται λύση ή ρίζα της εξίσωσης.

Αν μια εξίσωση στην άλγεβρα είναι γνωστή να είναι αληθής, οι ακόλουθες πράξεις μπορούν να χρησιμοποιηθούν για να παράγουν μια άλλη αληθή εξίσωση:

- Κάθε πραγματικός αριθμός μπορεί να προστεθεί και στις δύο πλευρές
- Κάθε πραγματικός αριθμός μπορεί να αφαιρεθεί από τις δύο πλευρές
- Κάθε πραγματικός αριθμός μπορεί να πολλαπλασιαστεί και στις δύο πλευρές
- Κάθε μη μηδενικός πραγματικός αριθμός μπορεί να διαιρεθεί από τις δύο πλευρές

## ΚΕΦΑΛΑΙΟ 1

Οι αλγεβρικές αυτές ιδιότητες υπονοούν ότι η ισότητα είναι μια σχέση συμβατότητας για ένα πεδίο, στην πραγματικότητα αυτή είναι ουσιαστικά η μόνη. Το πιο γνωστό σύστημα αριθμών το οποίο επιτρέπει όλες αυτές τις διεργασίες είναι οι πραγματικοί αριθμοί. Αν η εξίσωση βασιζόταν στους φυσικούς αριθμούς, για παράδειγμα, μερικές από αυτές τις πράξεις (όπως η διαίρεση και η αφαίρεση) ίσως να μην ήταν έγκυρες, καθώς δεν επιτρέπονται αρνητικοί αριθμοί και μη ακέραιοι αριθμοί.

### 1.3 ΓΡΑΜΜΙΚΑ ΣΥΣΤΗΜΑΤΑ

Μια εξίσωση που παριστάνει μια ευθεία γραμμή ονομάζεται γραμμική εξίσωση και είναι της μορφής:

$$αχ + βγ = γ, με α \neq 0 \text{ ή } β \neq 0 \quad (1.2)$$

- Αν  $α \neq 0$ , τότε η ευθεία τέμνει και τους δύο άξονες
- Αν  $α = 0$ , τότε η εξίσωση παριστάνει μια ευθεία που είναι παράλληλη στον άξονα  $χ'χ$
- Αν  $β = 0$  (άρα  $α \neq 0$ ), τότε η εξίσωση παριστάνει μια ευθεία που είναι παράλληλη στον άξονα  $γ'γ$

Όταν έχουμε  $n$  γραμμικές εξισώσεις και ζητάμε τις κοινές λύσεις αυτών, τότε λέμε πως έχουμε να λύσουμε ένα γραμμικό σύστημα  $n$  εξισώσεων με  $n$  αγνώστους, ή πιο σύντομα, ένα γραμμικό σύστημα  $n \times n$ .

Κάθε ζεύγος αριθμών που επαληθεύει τις  $n$  εξισώσεις τους συστήματος ονομάζεται λύση του συστήματος. Η επίλυση ενός γραμμικού συστήματος γίνεται με την κατάλληλη μετατροπή του σε ένα άλλο γραμμικό σύστημα, το οποίο έχει ακριβώς τις ίδιες λύσεις με το αρχικό. Τα δύο αυτά συστήματα ονομάζονται ισοδύναμα συστήματα. Παρακάτω βλέπουμε ένα παράδειγμα γραμμικού συστήματος  $3 \times 3$ :

$$\begin{aligned} \chi + 2\gamma &= 5 \\ 2\chi + 5\gamma &= 1 \\ -\chi + 3\gamma &= 5 \end{aligned} \quad (1.3)$$

Όπως αναφέραμε προηγουμένως, το σύστημα αυτό πρέπει πρώτα να μετατραπεί σε ένα ισοδύναμο του για να μπορεί να λυθεί. Στη συνέχεια θα μελετήσουμε συστήματα που δεν απαιτούν τη μετατροπή τους για να λυθούν, αλλά έχουν άμεσες λύσεις.

# ΜΑΘΗΜΑΤΙΚΕΣ ΜΕΘΟΔΟΙ

## 1.3.1 Άμεσες Μέθοδοι

### 1.3.1.1 Διαγώνια Συστήματα

Διαγώνια ονομάζουμε τα συστήματα στα οποία όλες οι εξισώσεις έχουν μόνο έναν άγνωστο και όλοι οι συντελεστές των αγνώστων, εκτός από αυτούς που βρίσκονται πάνω στη διαγώνιο, είναι ίσοι με το μηδέν. Για παράδειγμα:

$$\begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 8 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \quad (1.4)$$

Ένα σύστημα σε αυτή τη μορφή μπορεί να λυθεί άμεσα, ως εξής:

$$x_i = \frac{b_i}{a_{ii}}, a_{ii} \neq 0 \quad (1.5)$$

### 1.3.1.2 Κάτω Τριγωνικά Συστήματα

Κάτω Τριγωνικά ονομάζουμε τα συστήματα στα οποία η πρώτη εξίσωση έχουν μόνο έναν άγνωστο και επιλύεται και όλοι οι συντελεστές των αγνώστων, που βρίσκονται πάνω από τη διαγώνιο, είναι ίσοι με το μηδέν. Για παράδειγμα:

$$\begin{bmatrix} 5 & 0 & 0 & 0 \\ 1 & 6 & 0 & 0 \\ 2 & 3 & 4 & 0 \\ 1 & 2 & 3 & 3 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \\ 9 \\ 9 \end{bmatrix} \quad (1.6)$$

Ένα σύστημα στη μορφή αυτή λύνεται ως εξής:

$$x_i = \frac{1}{a_{ii}} \cdot (b_i - \sum_{j=1}^{i-1} a_{ij} \cdot x_j), a_{ii} \neq 0 \quad (1.7)$$

# ΚΕΦΑΛΑΙΟ 1

## 1.3.1.3 Άνω Τριγωνικά Συστήματα

Άνω Τριγωνικά ονομάζουμε τα συστήματα στα οποία η τελευταία εξίσωση έχει μόνο έναν άγνωστο και επιλύεται και όλοι οι συντελεστές των αγνώστων, που βρίσκονται κάτω από τη διαγώνιο, είναι ίσοι με το μηδέν. Για παράδειγμα:

$$\begin{bmatrix} 1 & 2 & 3 & 3 \\ 0 & 2 & 3 & 4 \\ 0 & 0 & 1 & 6 \\ 0 & 0 & 0 & 5 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 9 \\ 9 \\ 7 \\ 5 \end{bmatrix} \quad (1.8)$$

Ένα σύστημα σε αυτή τη μορφή λύνεται, ξεκινώντας από τον τελευταίο άγνωστο, ως εξής:

$$x_i = \frac{1}{a_{ii}} \cdot (b_i - \sum_{j=i+1}^n a_{ij} \cdot x_j), a_{ii} \neq 0 \quad (1.9)$$

## 1.3.1.4 Απαλοιφή Gauss με μερική οδήγηση

Στις προηγούμενες μεθόδους είδαμε πως λύνονται τα γραμμικά συστήματα όταν βρίσκονται στη μορφή που πρέπει. Σε περίπτωση που δεν ισχύει αυτό όμως, πρέπει να τα μετατρέψουμε εμείς στη σωστή μορφή. Μια από τις μεθόδους που μπορεί να το κάνει αυτό είναι η απαλοιφή Gauss με μερική οδήγηση, ακολουθώντας τα παρακάτω βήματα:

### **Βήμα 1:**

Ελέγχουμε όλες τις γραμμές  $i=j, \dots, n$  για να βρούμε τη γραμμή  $i$  που περιέχει το μεγαλύτερο κατ' απόλυτη τιμή στοιχείο στην αντίστοιχη στήλη, και ανταλλάσσουμε τις γραμμές  $i$  και  $j$ .

### **Βήμα 2:**

Πολλαπλασιάζουμε την πρώτη εξίσωση του συστήματος διαδοχικά με τους συντελεστές :

$$l_{ij} = \frac{a_{ij}^{(1)}}{a_{ii}^{(1)}} \quad i=2, \dots, n \quad (1.10)$$

### **Βήμα 3:**

Στο σύστημα των  $n-1$  εξισώσεων επαναλαμβάνουμε τα βήματα 1 και 2, διατηρώντας την δεύτερη εξίσωση ως έχει, ενώ από τις  $n-2$  υπόλοιπες θα απαλειφθεί ο άγνωστος 2 x κ.ο.κ..

# ΜΑΘΗΜΑΤΙΚΕΣ ΜΕΘΟΔΟΙ

## Βήμα 4:

Το σύστημα μετατρέπεται, μετά από  $n-1$  διαδοχικούς κύκλους, σε άνω τριγωνικό σύστημα, το οποίο μπορεί να επιλυθεί χρησιμοποιώντας τη σχέση (1.9) που αναφέραμε προηγουμένως.

## 1.3.2 Επαναληπτικές Μέθοδοι Επίλυσης Γραμμικών Συστημάτων

Οι κλασικές επαναληπτικές μέθοδοι βασίζονται στην ακόλουθη διάσπαση του πίνακα των συντελεστών των αγνώστων  $A$

$$A = D + L + U \quad (1.11)$$

- Το  $D$  αναφέρεται στο διαγώνιο πίνακα με διαγώνια στοιχεία τα αντίστοιχα του  $A$
- Το  $L$  είναι ο αυστηρά κάτω τριγωνικός πίνακας
- Το  $U$  ο αυστηρά άνω τριγωνικός πίνακας

### 1.3.2.1 Μέθοδος Jacobi

Στη μέθοδο του Jacobi ο ρυθμιστής πίνακας είναι ο  $M = D$ . Σ' ότι αφορά την ικανοποίηση των δύο βασικών περιορισμών που αφορούν στο ρυθμιστή πίνακα  $M$  διαπιστώνουμε τα ακόλουθα. Καταρχάς ο  $M$  είναι αντιστρέψιμος αν ισχύει:

$$\det(M) = \det(D) = a_{11}a_{22} \cdots a_{nn} \neq 0 \quad (1.12)$$

Επομένως η μέθοδος του Jacobi μπορεί να οριστεί αν:

$$a_{ii} \neq 0, \quad i = 1..n \quad (1.13)$$

Ακόμη είναι προφανές ότι ένα γραμμικό σύστημα με πίνακα συντελεστών αγνώστων  $M = D$  είναι πολύ οικονομικότερο σε πράξεις για να λυθεί από ό,τι είναι ένα σύστημα με πίνακα συντελεστών αγνώστων  $A$ . Άρα ικανοποιείται και ο δεύτερος περιορισμός εφόσον ικανοποιείται ο πρώτος. Σε μορφή πινάκων η μέθοδος του Jacobi είναι η ακόλουθη:

$$X^{(k+1)} = D^{-1}b - D^{-1}(L + U)X^{(k)} \quad (1.14)$$

## ΚΕΦΑΛΑΙΟ 1

Αν θελήσουμε να βρούμε την οποιαδήποτε συνιστώσα του νέου διανύσματος στην  $k$  επανάληψη σε συνάρτηση γνωστών συνιστωσών του παλιού ή και του νέου διανύσματος, πολλαπλασιάζουμε τα μέλη της σχέσης (2.14) από τα αριστερά επί  $D$ , οπότε προκύπτει:

$$Dx^{(k+1)} = b - (L+U)x^{(k)} \quad (1.15)$$

Αν τώρα εκτελέσουμε τις πράξεις και στα δύο μέλη και εξισώσουμε τις  $i$ -οστές συνιστώσες των διανυσμάτων των δύο μελών και λύσουμε ως προς την  $i$ -οστή συνιστώσα παίρνουμε αμέσως ότι:

$$x_i^{(k+1)} = (b_i - \sum_{j=1, j \neq i}^n a_{ij}x_j^{(k)})/a_{ii}, \quad i = 1(1)n \quad (1.16)$$

### 1.3.2.2 Μέθοδος Gauss - Seidel

Στη μέθοδο των Gauss - Seidel και με βάση πάντα τη διάσπαση (1.11) επιλέγεται  $M = D + L$ . Για να υπάρχει η μέθοδος θα πρέπει να υπάρχει ο αντίστροφος του  $D + L$ , που είναι κάτω τριγωνικός πίνακας. Άρα αυτός θα είναι αντιστρέψιμος αν ισχύει:

$$\det(D + L) \neq 0 \quad (1.17)$$

Αυτό ισοδυναμεί με τη σχέση (2.13), όπως ακριβώς και στη μέθοδο του Jacobi. Ακόμη, ένα σύστημα με πίνακα συντελεστών αγνώστων  $D + L$ , λύνεται με προς τα μπρός αντικαταστάσεις και επομένως είναι οικονομικότερο στην επίλυσή του από ένα σύστημα με πίνακα συντελεστών  $A$ . Άρα ικανοποιείται κι ο δεύτερος περιορισμός σ' ό,τι αφορά τον  $M$  με την προϋπόθεση ότι ικανοποιείται και ο πρώτος. Επομένως, η μέθοδος των Gauss - Seidel είναι η ακόλουθη:

$$x^{(k+1)} = (D + L)^{-1}b - (D + L)^{-1}Ux^{(k)} \quad (1.18)$$

Για την εύρεση αναλυτικών εκφράσεων για τις συνιστώσες του νέου διανύσματος, ισχύει ότι ίσχυε και για τη μέθοδο του Jacobi. Δηλαδή πολλαπλασιάζουμε από τα αριστερά τα μέλη της σχέσης (2.18) επί  $D + L$ , εκτελούμε τις πράξεις και στα δύο μέλη εξισώνουμε τις  $i$ -οστές συνιστώσες των δύο μελών και τέλος έχουμε:

$$x_i^{(k+1)} = (b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)})/a_{ii}, \quad i = 1(1)n \quad (1.19)$$



## ΜΑΘΗΜΑΤΙΚΕΣ ΜΕΘΟΔΟΙ

### 1.3.2.3 Σύγκριση Μεθόδων Jacobi και Gauss - Seidel

Αν παρατηρήσει κανείς προσεκτικά τις μεθόδους των Jacobi και Gauss - Seidel θα δει ότι έχουν πάρα πολλές ομοιότητες και συγχρόνως κάποιες σημαντικές διαφορές. Ιδιαίτερα, αυτό μπορεί να φανεί, αν η σχέση (1.16) γραφτεί με τέτοιο τρόπο ώστε να άθροισμα μέσα στην παρένθεση να διασπαστεί σε δύο αθροίσματα:

$$x_i^{(k+1)} = (b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)})/a_{ii}, \quad i = 1(1)n \quad (1.20)$$

Τότε μπορεί να διαπιστωθεί το εξής. Για την εύρεση μιας συνιστώσας της νέας επανάληψης στη μέθοδο του Jacobi χρησιμοποιούνται όλες οι άλλες συνιστώσες της προηγούμενης επανάληψης ενώ στη μέθοδο των Gauss-Seidel χρησιμοποιούνται όλες οι επόμενες συνιστώσες της προηγούμενης επανάληψης και όλες οι προηγούμενες συνιστώσες της τρέχουσας επανάληψης. Πρακτικά, στην Gauss-Seidel χρησιμοποιούνται όλες οι πιο πρόσφατες διαθέσιμες συνιστώσες.

Το τελευταίο στοιχείο ίσως κάνει τη μέθοδο των Gauss-Seidel ελκυστικότερη από αυτήν του Jacobi γιατί “φαίνεται” να πλεονεκτεί στην περίπτωση σύγκλισης. Αυτό δεν είναι πάντοτε αληθές. Υπάρχουν περιπτώσεις όπου η μέθοδος του Jacobi συγκλίνει ενώ η μέθοδος των Gauss-Seidel αποκλίνει! Ακόμη, αν πάψει κανείς να σκέφτεται σειριακά (ακολουθιακά), τότε, αν έχουμε στη διάθεσή μας έναν Υπολογιστή με παράλληλη αρχιτεκτονική, στη μεν μέθοδο του Jacobi είναι δυνατόν όλες οι συνιστώσες της νέας επανάληψης να βρεθούν σε ένα χρονικό βήμα από όλες της προηγούμενης επανάληψης ενώ στη μέθοδο των Gauss-Seidel αυτό δεν είναι δυνατόν.

Τέλος, αξίζει να σημειώσουμε πως αν υπάρχει υπεροχή κατά γραμμές ή κατά στήλες, τότε η ακολουθία των διαδοχικών προσεγγίσεων που παράγεται από τις μεθόδους Jacobi και Gauss - Seidel για την επίλυση του συστήματος, συγκλίνει στη λύση  $x$  του συστήματος.

## 1.4 ΑΡΙΘΜΗΤΙΚΗ ΕΠΙΛΥΣΗ ΕΞΙΣΩΣΕΩΝ

Ένα από τα κλασικότερα προβλήματα των Μαθηματικών είναι η επίλυση της εξίσωσης  $f(x)=0$ , η εύρεση δηλαδή μιας ή περισσοτέρων τιμών που τη μηδενίζουν. Η αναλυτική έκφραση αυτών των τιμών μπορεί να δοθεί σε ελάχιστες περιπτώσεις.

Για το λόγο αυτό, είναι αναγκαία η χρησιμοποίηση μεθόδων για την προσέγγιση των ριζών μιας συνάρτησης. Στην περίπτωση αυτή, οι μέθοδοι είναι επαναληπτικές, οπότε ξεκινώντας από μια αρχική εκτίμηση, προσδιορίζουμε τον αλγόριθμο που θα δημιουργήσει μια ακολουθία αριθμών και τελικά θα συγκλίνει στη ζητούμενη ρίζα. Στη συνέχεια παρουσιάζουμε μερικές από αυτές τις μεθόδους.

# ΚΕΦΑΛΑΙΟ 1

## 1.4.1 Μέθοδος της Διχοτόμησης

Η μέθοδος της Διχοτόμησης βασίζεται κυρίως στο θεώρημα του Bolzano:

**Θεώρημα:** Αν η συνάρτηση  $f(x)$  είναι συνεχής στο κλειστό διάστημα  $[a, \beta]$  και ισχύει  $f(a)f(\beta) < 0$ , τότε υπάρχει τουλάχιστον μια ρίζα  $\rho$  της συνάρτησης  $f(x)$  στο διάστημα αυτό. Αν επιπλέον στο διάστημα  $[a, \beta]$  ισχύει  $f'(x) \neq 0$  για κάθε  $x \in [a, \beta]$ , η ρίζα αυτή είναι μοναδική.

Έστω η εξίσωση  $f(x) = 0$  με μια ρίζα  $\xi$  στο διάστημα  $I_0 = (a_0, b_0)$ . Υποθέτουμε ότι για τη συνάρτηση  $f(x)$  ισχύουν οι υποθέσεις του θεωρήματος και η εξίσωση έχει μία μοναδική ρίζα στο διάστημα  $I_0 = (a_0, b_0)$ . Για την εύρεση της ρίζας  $\xi$ , απαιτούνται τα παρακάτω βήματα:

### Βήμα 1:

Βρίσκουμε την πρώτη προσέγγιση της ζητούμενης ρίζας που είναι το σημείο:

$$x_0 = \frac{a_0 + b_0}{2} \quad (1.21)$$

### Βήμα 2:

Αν το  $x_0$  δεν είναι η ρίζα, ψάχνουμε τη ρίζα σ' ένα από τα υπο-διαστήματα  $(a_0, x_0)$ ,  $(x_0, b_0)$  στο πρώτο, αν  $f(a_0) \cdot f(x_0) < 0$  ή στο δεύτερο, αν  $f(x_0) \cdot f(b_0) < 0$ .

### Βήμα 3:

Στην πρώτη περίπτωση νέο διάστημα εντοπισμού της ρίζας είναι το  $I_1 = (a_1, b_1) = (a_0, x_0)$ , ενώ στη δεύτερη  $I_1 = (a_1, b_1) = (x_0, b_0)$ .

### Βήμα 4:

Βρίσκουμε τη νέα προσέγγιση της ζητούμενης ρίζας που είναι το σημείο:

$$x_1 = \frac{a_1 + b_1}{2} \quad (1.22)$$

και επαναλαμβάνουμε τα βήματα 2, 3, 4, μέχρι να βρούμε μια ικανοποιητική προσέγγιση για τη ρίζα  $\xi$ .

Αν συνεχίσουμε τη διαδικασία με τον ίδιο τρόπο, για  $k$  επαναλήψεις τότε η ακολουθία που προκύπτει συγκλίνει πάντοτε στη ρίζα έπειτα από συγκεκριμένες επαναλήψεις σύμφωνα με τον τύπο:

$$n \geq \frac{\log(|b_0 - a_0| \cdot 10^k)}{\log 2} \quad (1.23)$$

## ΜΑΘΗΜΑΤΙΚΕΣ ΜΕΘΟΔΟΙ

### Πλεονεκτήματα της Μεθόδου της Διχοτόμησης:

- Είναι η μόνη μέθοδος που συγκλίνει ακόμη και όταν η αρχική τιμή δε βρίσκεται κοντά στην επιθυμητή προσέγγιση της ρίζας, άρα υπάρχει πάντα σύγκλιση στο διάστημα εφαρμογής της
- Ο απαιτούμενος αριθμός επαναλήψεων που χρειάζονται για την προσέγγισή της είναι πάντοτε γνωστός, σύμφωνα με τη σχέση (2.23)
- Δε χρειάζονται περισσότερες πληροφορίες κατά την εφαρμογή της μεθόδου εκτός από το αλγεβρικό πρόσημο της  $f(x)$  σε κάθε σημείο

### Μειονεκτήματα της Μεθόδου της Διχοτόμησης:

- Συγκλίνει αργά στην προσεγγιστική τιμή
- Σε ορισμένες περιπτώσεις μπορεί να εμφανιστεί καλύτερη προσέγγιση της ρίζας σε μικρότερο αριθμό βημάτων από τον προκαθορισμένο αριθμό, οπότε τα επιπλέον βήματα είναι περιττά

### 1.4.2 Μέθοδος της Εσφαλμένης Θέσης

Ο αλγοριθμικός τύπος αυτής της μεθόδου προκύπτει από ένα ευθύγραμμο τμήμα το οποίο προσεγγίζει το τμήμα της συνάρτησης  $f(x)$  που ορίζεται στο υπό μελέτη διάστημα και πληρεί τις προϋποθέσεις του θεωρήματος Bolzano. Η τομή του ευθύγραμμου τμήματος με τον άξονα  $x$  αποτελεί την προσεγγιστική τιμή της ρίζας της συνάρτησης και χωρίζει το διάστημα σε δύο νέα διαστήματα. Σαν νέο διάστημα επιλέγεται εκείνο στο οποίο βρίσκεται η ρίζα, βάση του θεωρήματος Bolzano.

Έστω η εξίσωση  $f(x) = 0$  με μια ρίζα  $\xi$  στο διάστημα  $I_0 = (a_0, b_0)$ . Υποθέτουμε ότι για τη συνάρτηση  $f(x)$  ισχύουν οι υποθέσεις του θεωρήματος και η εξίσωση έχει μία μοναδική ρίζα στο διάστημα  $I_0 = (a_0, b_0)$ . Για την εύρεση της ρίζας  $\xi$ , απαιτούνται τα παρακάτω βήματα:

#### Βήμα 1:

Βρίσκουμε την πρώτη προσέγγιση της ζητούμενης ρίζας που είναι το σημείο:

$$x_0 = b_0 - \frac{b_0 - a_0}{f(b_0) - f(a_0)} \cdot f(b_0) \quad (1.24)$$

#### Βήμα 2:

Αν το  $x_0$  δεν είναι η ρίζα, ψάχνουμε τη ρίζα σ' ένα από τα υπο-διαστήματα  $[a_0, x_0]$ ,  $[x_0, b_0]$  στο πρώτο, αν  $f(a_0) \cdot f(x_0) < 0$  ή στο δεύτερο, αν  $f(x_0) \cdot f(b_0) < 0$ .

## ΚΕΦΑΛΑΙΟ 1

### Βήμα 3:

Στην πρώτη περίπτωση νέο διάστημα εντοπισμού της ρίζας είναι το  $I_1 = (a_1, b_1) = (a_0, x_0)$ , ενώ στη δεύτερη  $I_1 = (a_1, b_1) = (x_0, b_0)$ .

### Βήμα 4:

Βρίσκουμε τη νέα προσέγγιση της ζητούμενης ρίζας που είναι το σημείο:

$$x_1 = b_1 - \frac{b_1 - a_1}{f(b_1) - f(a_1)} \cdot f(b_1) \quad (1.25)$$

και επαναλαμβάνουμε τα βήματα 2, 3, 4, μέχρι να βρούμε μια ικανοποιητική προσέγγιση για τη ρίζα  $\xi$ .

### Πλεονεκτήματα της μεθόδου Εσφαλμένης Θέσης:

- Η ακολουθία του αλγοριθμικού τύπου συγκλίνει πάντα
- Είναι ταχύτερη από τη μέθοδο της διχοτόμησης

### Μειονεκτήματα της μεθόδου Εσφαλμένης Θέσης:

- Συγκλίνει με πιο αργό ρυθμό σε σχέση με άλλες μεθόδους

### 1.4.3 Μέθοδος Newton - Raphson

Η μέθοδος Newton-Raphson αποτελεί την ταχύτερη από τις μεθόδους προσέγγισης ρίζας μιας συνάρτησης και για αυτό το λόγο χρησιμοποιείται συχνότερα σε προβλήματα επίλυσης εξισώσεων, καθώς και σε συστήματα μη γραμμικών εξισώσεων. Σε αντίθεση με τις υπόλοιπες μεθόδους, η μέθοδος αυτή εγγυάται τουλάχιστον τετραγωνική σύγκλιση.

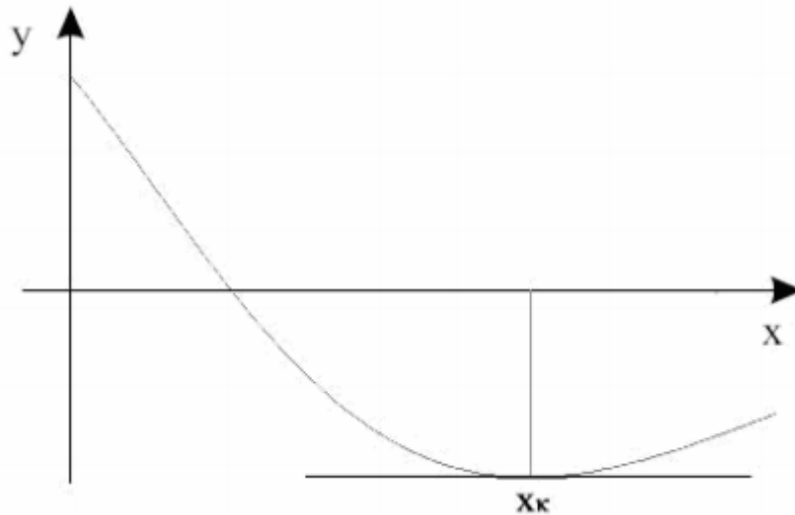
Για την εφαρμογή της απαιτείται η ύπαρξη πρώτης και δεύτερης παραγώγου της συνάρτησης  $f(x) = 0$  στο διάστημα  $I = (a, b)$  και ο γενικός της επαναληπτικός τύπος είναι ο εξής:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (1.26)$$

## ΜΑΘΗΜΑΤΙΚΕΣ ΜΕΘΟΔΟΙ

Η μέθοδος Newton-Raphson αποτυγχάνει να συγκλίνει σε τρεις περιπτώσεις:

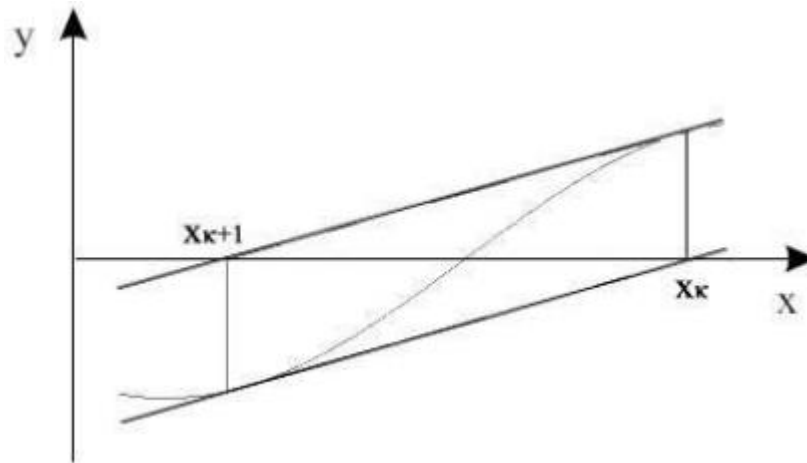
1. Όταν η παράγωγος σε κάποια επανάληψη μηδενιστεί. Τότε δεν είναι πλέον δυνατός ο προσδιορισμός του επόμενου  $x$ . (Σχήμα 1.1)



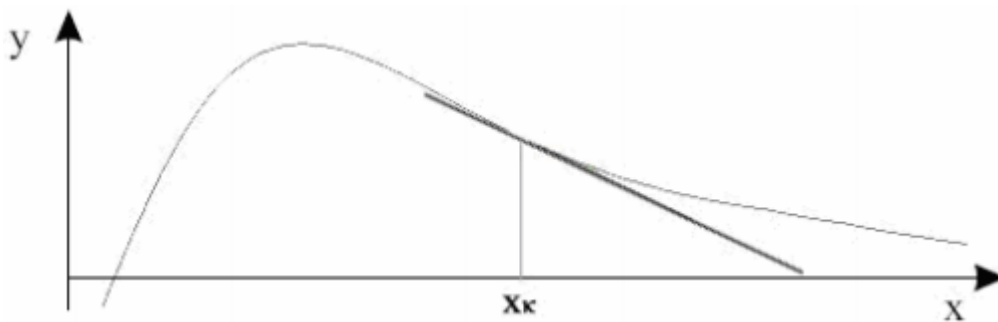
Σχήμα 1.1 Πρώτη περίπτωση αποτυχίας σύγκλισης Newton - Raphson

2. Όταν η συνάρτηση αλλάζει πρόσημο σε δύο σημεία, ενώ η παράγωγος της στα σημεία αυτά έχει την ίδια τιμή. Στην περίπτωση αυτή, η μέθοδος Newton - Raphson εισέρχεται σε ένα κλειστό βρόχο, από τον οποίο δεν μπορεί να εξέλθει, παρά μόνο όταν εξαντληθούν οι επαναλήψεις που έχει δηλώσει ο χρήστης για την εύρεση της ρίζας. (Σχήμα 1.2)
3. Όταν η συνάρτηση προσεγγίζει ασυμπτωτικά το 0 και έχει γίνει λανθασμένη επιλογή αρχικού σημείου. Σ' αυτή την περίπτωση η μέθοδος Newton μπορεί να απομακρύνεται από την ρίζα της συνάρτησης. (Σχήμα 1.3)

## ΚΕΦΑΛΑΙΟ 1



Σχήμα 1.2 Δεύτερη περίπτωση αποτυχίας σύγκλισης Newton - Raphson



Σχήμα 1.3 Τρίτη περίπτωση αποτυχίας σύγκλισης Newton - Raphson

### Πλεονεκτήματα της Μεθόδου Newton - Raphson:

- Το κυριότερο πλεονέκτημα της μεθόδου είναι η μεγάλη ταχύτητα σύγκλισης, η οποία ξεπερνά την ταχύτητα σύγκλισης πολλών άλλων μεθόδων.

### Μειονεκτήματα της Μεθόδου Newton - Raphson:

- Η μέθοδος είναι επιρρεπής σε ταλαντώσεις
- Δεν υπάρχει εγγύηση ότι η μέθοδος θα συγκλίνει
- Απαιτεί σε κάθε ρίζα, η παράγωγος να είναι μη μηδενική, αλλιώς η μέθοδος αποτυγχάνει. Αν μηδενιστεί, τότε τείνει στο άπειρο η προσεγγιστική τιμή της ρίζας και δεν μπορούμε να την επανεκκινήσουμε.
- Είναι απαραίτητος ο υπολογισμός της παραγώγου

## ΜΑΘΗΜΑΤΙΚΕΣ ΜΕΘΟΔΟΙ

Για την αντιμετώπιση παραπάνω μειονεκτημάτων, αρχικά ελέγχουμε τη μοναδικότητα της ρίζας στο διάστημα εφαρμογής της μεθόδου, και στη συνέχεια εξετάζουμε την περίπτωση σύγκλισης στη ρίζα.

### 1.5 ΕΠΙΛΟΓΟΣ

Στο κεφάλαιο αυτό είδαμε αναλυτικά όλες τις μαθηματικές μεθόδους που καθιστούν δυνατή τη λειτουργικότητα της εφαρμογής. Είδαμε τα πλεονεκτήματα και τα μειονεκτήματα της καθεμιάς, καθώς και πότε πρέπει να χρησιμοποιούνται και με ποιον τρόπο μας βοηθάνε να βρούμε τα αποτελέσματα που ψάχνουμε.

Στο επόμενο κεφάλαιο θα τις τεχνολογίες που χρησιμοποιεί η εφαρμογή. Το πως καταφέρνει να εντάξει όλες αυτές τις μεθόδους και την έξοδο που παίρνουμε, ανάλογα με την είσοδο που δίνουμε στην εφαρμογή.

# 2. ΤΕΧΝΟΛΟΓΙΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

## 2.1 ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό θα περιγράψουμε αναλυτικά τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής, καθώς και τον τρόπο με τον οποίο χρησιμοποιήθηκε η κάθε μία από αυτές. Πιο συγκεκριμένα, θα αναφερθούμε στις γλώσσες προγραμματισμού, τις απαραίτητες βιβλιοθήκες, καθώς και το προγραμματιστικό περιβάλλον στο οποίο έλαβε χώρα η υλοποίηση.

## 2.2 ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

### 2.2.1 Java

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού γενικού σκοπού που βασίζεται στις κλάσεις. Οι James Gosling, Mike Sheridan και Patrick Naughton ξεκίνησαν τη δημιουργία της τον Ιούνιο του 1991. Η γλώσσα αρχικά ονομάστηκε Oak από τον James Gosling, εμπνευσμένο από μια βελανιδιά που βρισκόταν έξω από το γραφείο του. Στη συνέχεια μετονομάστηκε σε Green, μέχρι που πήρε την τελική ονομασία Java, τον γνωστό παραδοσιακό καφέ της Ινδονησίας. [2] Η πρώτη έκδοση Java 1.0 κυκλοφόρησε το 1996 από τη Sun Microsystems μαζί με την υπόσχεση Write Once, Run Anywhere (WORA), εννοώντας πως με μια συγγραφή κώδικα θα μπορείς να τον τρέξεις παντού, σε όλες τις πλατφόρμες.



Σχήμα 2.1 Λογότυπο της Java



## ΤΕΧΝΟΛΟΓΙΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

Οι 5 βασικές αρχές για τη δημιουργία της Java ήταν οι εξής: [3]

- Πρέπει να είναι απλή, αντικειμενοστραφής και οικεία
- Πρέπει να πληρεί τους κανόνες ευρωστίας και να είναι ασφαλής
- Πρέπει να είναι αρχιτεκτονικά ουδέτερη και φορητή
- Πρέπει να εκτελείται με υψηλή απόδοση και αποτελεσματικότητα
- Πρέπει να ερμηνεύεται εύκολα και να είναι δυναμική

Το κλασικό παράδειγμα “Hello World!” που χρησιμοποιείται σε όλες τις γλώσσες προγραμματισμού παρουσιάζεται παρακάτω και πρόκειται για ένα κομμάτι κώδικα το οποίο τυπώνει στην οθόνη τη φράση “Hello World!”.

### **public class HelloWorldApp:**

Όλα τα αρχεία κώδικα της Java πρέπει να έχουν το ίδιο όνομα με την public κλάση που περιέχουν ακολουθούμενα από την κατάληξη .java. Στο συγκεκριμένο παράδειγμα θα είχαμε το αρχείο HelloWorldApp.java. Έπειτα, ο compiler (μεταγλωττιστής) της Java δημιουργεί το αντίστοιχο αρχείο .class και μόνο τότε μπορεί πλέον να εκτελεστεί το αρχείο. Κάθε αρχείο Java μπορεί να περιέχει μόνο μία public κλάση, αλλά πολλές κλάσεις που δεν είναι public.

### **public:**

Η λέξη κλειδί public, όταν χρησιμοποιείται για να περιγράψει μια μέθοδο, δηλώνει πως η μέθοδος αυτή μπορεί να κληθεί και από άλλες κλάσης. Όταν αναφέρεται σε κλάση, σημαίνει πως η κλάση αυτή μπορεί να χρησιμοποιηθεί και από άλλες κλάσης, που βρίσκονται εκτός της ιεραρχίας των κλάσεων. Άλλες λέξεις κλειδιά που αφορούν το επίπεδο πρόσβασης είναι οι λέξεις private και protected.

### **static:**

Η λέξη κλειδί static, όταν βρίσκεται πριν το όνομα μιας μεθόδου, δηλώνει μια στατική μέθοδο, η οποία σχετίζεται μόνο με αυτή την κλάση. Μόνο οι στατικές μέθοδο μπορούν να κληθούν χωρίς αναφορά σε κάποιο αντικείμενο και έχουν πρόσβαση μόνο στα μέλη της κλάσης που είναι επίσης static.

### **void:**

Η λέξη κλειδί void δηλώνει πως η μέθοδος δεν επιστρέφει καμία τιμή.

### **main:**

Με τη λέξη κλειδί main δηλώνονται οι μέθοδοι τις οποίες καλεί ο launcher της JAVA, για να παραδώσει τον έλεγχο του προγράμματος.

### **String [] args:**

Η μέθοδος main πρέπει να δεχτεί αντικειμένων String. Είναι συνηθισμένο να χρησιμοποιείται το όνομα args, παρόλο που μπορεί να αντικατασταθεί από οποιοδήποτε όνομα.

### **System.out.println(“Hello World!”):**

Είναι η μέθοδος που καλούμε όταν θέλουμε να τυπώσουμε κάτι στην οθόνη.

## ΚΕΦΑΛΑΙΟ 2

```
public class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Prints the string to the console.
    }
}
```

Σχήμα 2.2 Η μέθοδος “Hello World!”

### 2.2.2 XML

Η XML (eXtensible Markup Language) είναι μια γλώσσα σήμανσης που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων που δημιουργήθηκε από το διεθνή οργανισμό W3C (World Wide Web Consortium). Πρόκειται για μια μορφοποίηση κειμένου, με ισχυρή υποστήριξη Unicode για όλες τις γλώσσες του κόσμου και σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο διαδίκτυο. Αν και η σχεδίαση της XML εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν για παράδειγμα στις υπηρεσίες ιστού.

Υπάρχει μια ποικιλία διεπαφών προγραμματισμού εφαρμογών, που μπορούν να χρησιμοποιούν οι προγραμματιστές, για να προσπελαίνουν δεδομένα XML, αλλά και διάφορα συστήματα σχημάτων XML, τα οποία είναι σχεδιασμένα για να βοηθούν στον ορισμό γλωσσών, που προκύπτουν από την XML.

Έως το 2009, έχουν αναπτυχθεί εκατοντάδες γλώσσες που βασίζονται στην XML, συμπεριλαμβανομένων των RSS, SOAP και XHTML. Για τις περισσότερες σουίτες εφαρμογών γραφείου υπάρχουν προεπιλεγμένες κωδικοποιήσεις βασισμένες στην XML, όπως του Microsoft Office (Office Open XML), του OpenOffice.org (OpenDocument) και του iWork της Apple. [4]

#### **Χαρακτήρας Unicode:**

Ένα κείμενο XML είναι μια ακολουθία χαρακτήρων στο οποίο σχεδόν όλοι οι χαρακτήρες Unicode μπορούν να εμφανίζονται.

#### **Σήμανση και περιεχόμενο:**

Οι χαρακτήρες που απαρτίζουν ένα κείμενο XML, αποτελούν είτε τη σήμανση είτε το περιεχόμενο του. Τα δύο αυτά στοιχεία μπορούν να διακριθούν ύστερα από την εφαρμογή κάποιων απλών συντακτικών κανόνων. Οι αλφαριθμητικοί χαρακτήρες που συνιστούν τη σήμανση, ξεκινούν με τους χαρακτήρες “<” και “&” και καταλήγουν στους χαρακτήρες “>” και “;” αντίστοιχα. Οι ακολουθίες χαρακτήρων που δε συνιστούν τη σήμανση, αποτελούν το περιεχόμενο ενός κειμένου XML.

#### **Ετικέτα:**

Είναι ένα στοιχείο σήμανσης που ξεκινά με το χαρακτήρα “<” και καταλήγει στο χαρακτήρα “>”. Υπάρχουν τρία είδη ετικέτας: ετικέτες αρχής, ετικέτες τέλους και ετικέτες χωρίς περιεχόμενο.

## ΤΕΧΝΟΛΟΓΙΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

### Στοιχείο:

Στοιχείο ονομάζεται ένα λογικό απόσπασμα κειμένου, που είτε ξεκινά με μία ετικέτα αρχής και καταλήγει σε μια ετικέτα τέλους, είτε αποτελείται μόνο από μια ετικέτα χωρίς περιεχόμενο. Οι χαρακτήρες που υπάρχουν μεταξύ μιας ετικέτας αρχής και μιας ετικέτας τέλους δημιουργούν το περιεχόμενο του στοιχείου, το οποίο μπορεί να περιέχει σήμανση, καθώς και άλλα στοιχεία, που ονομάζονται στοιχεία παιδιά.

### Χαρακτηριστικό:

Είναι ένα στοιχείο σήμανσης που αποτελείται από ένα ζευγάρι ονόματος-τιμής, το οποίο υπάρχει μέσα σε μία ετικέτα αρχής ή ετικέτα χωρίς περιεχόμενο.

### Δήλωση XML:

Τα κείμενα XML συνήθως αρχίζουν με τη δήλωση κάποιων πληροφοριών σχετικών με αυτά, όπως για παράδειγμα: `<?xml version="1.0" encoding="utf-8"?>`

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    tools:context=".ResultActivity">
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:id="@+id/textRes"
            android:textColor="@color/colorPrimary"
            android:padding="5dp"
            android:textSize="20sp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Activity 2"/>
        </ScrollView>
    </RelativeLayout>
```

Σχήμα 2.3 Παράδειγμα XML

## 2.3 ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ

### 2.3.1 Android Studio

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε εξ' ολοκλήρου το Android Studio. Πρόκειται για ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για ανάπτυξη εφαρμογών στην πλατφόρμα Android, βασισμένο στο περιβάλλον IntelliJ IDEA της JetBrains. Είναι διαθέσιμο για εγκατάσταση στα λειτουργικά συστήματα Windows macOS και Linux. Ανακοινώθηκε στις 16 Μαΐου του 2013 στο Google I/O. Από τότε ακολούθησε μια σειρά αναβαθμίσεων με την τελευταία μεγάλη να κυκλοφορεί τον Μάιο του 2020. Παρακάτω παραθέτουμε μερικά από τα χαρακτηριστικά που το καθιστούν τόσο πετυχημένο:

#### Έξυπνος code editor:

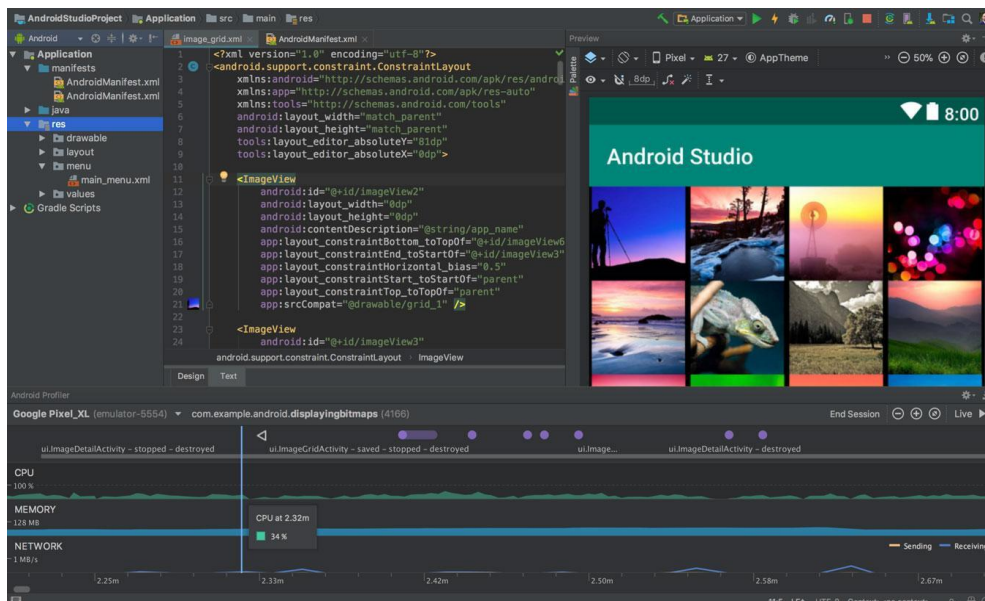
Βοηθάει στο να δουλεύεις γρηγορότερα και πιο αποτελεσματικά, προσφέροντας αυτόματης συμπλήρωσης κώδικα, ανακατασκευής και ανάλυσης κώδικα.

#### Εφαρμογή αλλαγών:

Το Android Studio επιτρέπει στο χρήστη την εφαρμογή αλλαγών στον κώδικα, ενώ η εφαρμογή τρέχει, χωρίς να απαιτείται επανεκκίνηση της εφαρμογής.

#### Προσομοιωτές Android:

Υπάρχει η δυνατότητα της εγκατάστασης Android emulator, που επιτρέπουν το άνοιγμα της εφαρμογής σε αυτό. Έτσι μπορούμε να δοκιμάσουμε αν η εφαρμογή μας τρέχει σωστά σε διαφορετικές εκδόσεις Android και διαφορετικά κινητά ή tablet.



Σχήμα 1.4 Περιβάλλον Android Studio

### 2.4 ΒΙΒΛΙΟΘΗΚΕΣ

#### 2.4.1 Android Image Cropper

Το Android Image Cropper είναι μια βιβλιοθήκη περικοπής εικόνων για Android. Παρέχει τη δυνατότητα περιστροφής εικόνας και ζουμ κατά τη διάρκεια της περικοπής, καθώς και την αλλαγή του μεγέθους της κατά την αποθήκευση. Τέλος, επιτρέπει την περικοπή εικόνας και την αποθήκευση της σε μορφή Bitmap, Resource ή Android URI. Υλοποιήθηκε από τον Arthur Terplitzki και έχει αναρτηθεί στο GitHub. [5]

#### 2.4.2 Google Vision API

Πρόκειται για ένα API το οποίο περιέχει εκπαιδευμένα μοντέλα μηχανικής μάθησης και επιτρέπει στους χρήστες του να χρησιμοποιούν την τεχνολογία Vision της Google. Το Vision API μπορεί πολύ γρήγορα να ταξινομήσει εικόνες σε πολλές κατηγορίες με λογικές ετικέτες. Μπορεί ακόμα να αναγνωρίσει αντικείμενα, πρόσωπα, καθώς και τμήματα κειμένου από μια φωτογραφία. Η συγκεκριμένη εφαρμογή χρησιμοποιεί το Optical Character Recognition (OCR) κομμάτι του Google Vision, προκειμένου να αντλεί κείμενο από μια εικόνα.

### 2.5 ΕΠΙΛΟΓΟΣ

Στο κεφάλαιο αυτό αναλύσαμε τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής, το περιβάλλον στο οποίο έλαβε χώρα, καθώς και τις βιβλιοθήκες που μας βοήθησαν να πετύχουμε τους στόχους μας. Στη συνέχεια θα δούμε αναλυτικά την περιγραφή της εφαρμογής. Τον τρόπο με τον οποίο συνδυάζονται οι μέθοδοι Αριθμητικής Ανάλυσης με τις τεχνολογίες που χρησιμοποιεί η εφαρμογή προκειμένου να εμφανίσει τα βήματα επίλυσης, καθώς και τα αποτελέσματα που μας ενδιαφέρουν.

# 3. ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

## 3.1 ΕΙΣΑΓΩΓΗ

Μέχρι στιγμής είδαμε τις τεχνολογίες που χρησιμοποιεί η εφαρμογή, καθώς και τις μαθηματικές μεθόδους που έχουμε εντάξει προκειμένου να πετύχουμε τους στόχους μας. Πάμε τώρα να δούμε τον τρόπο με τον οποίο καταφέρνει μέσω της εφαρμογής να αλληλεπιδρά η Java με τα Μαθηματικά και δεχόμενη μια απλή είσοδο να παρουσιάζει τα αποτελέσματα, όπως και τον τρόπο επίλυσης.

## 3.2 ΓΕΝΙΚΑ

Όπως έχουμε αναφέρει και σε προηγούμενα κεφάλαια, η εφαρμογή δέχεται μια είσοδο, γίνεται επιλογή της μαθηματικής μεθόδου που θα χρησιμοποιηθεί και τέλος παρουσιάζονται τα αποτελέσματα, καθώς και τα βήματα επίλυσης που ακολούθησε η κάθε μέθοδος.

Για να το πετύχουμε αυτό, δημιουργήσαμε δύο διαφορετικές οθόνες για την εφαρμογή ή αλλιώς activities. Ένα activity ουσιαστικά παρέχει το παράθυρο στο οποίο η εφαρμογή θα ζωγραφίσει το UI. Συνήθως πιάνει όλη την οθόνη, αλλά δίνεται η δυνατότητα να το μικρύνουμε, ή και να το κάνουμε να εμφανίζεται σαν pop - up πάνω από άλλα παράθυρα. Ουσιαστικά, ένα activity υλοποιεί μια οθόνη της εφαρμογής.

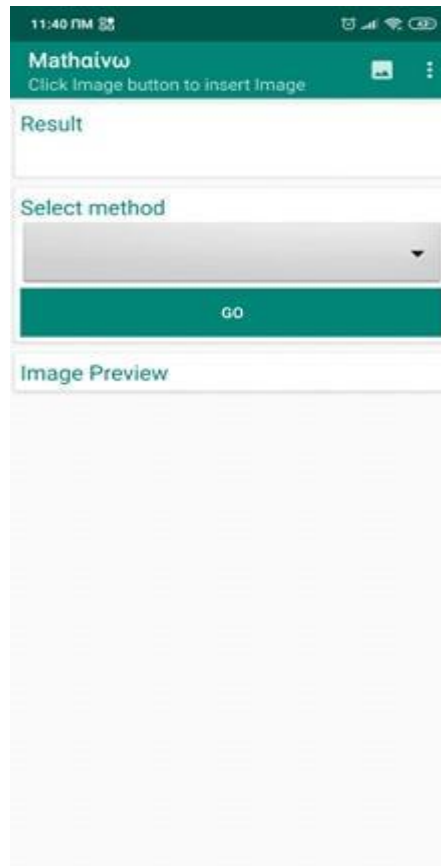
Στις περισσότερες εφαρμογές υπάρχουν πολλαπλές οθόνες, άρα και πολλαπλά Activities. Είθισται ένα activity σε κάθε εφαρμογή να δηλώνεται ως το main activity, το οποίο είναι η πρώτη οθόνη που εμφανίζεται όταν ανοίγουμε την εφαρμογή. Κάθε activity μπορεί να ανοίγει ένα άλλο, και με αυτό τον τρόπο δημιουργείται και η αλυσίδα αλληλεπιδράσεων των οθονών εντός της εφαρμογής.

Η εφαρμογή αποτελείται από δύο activities:

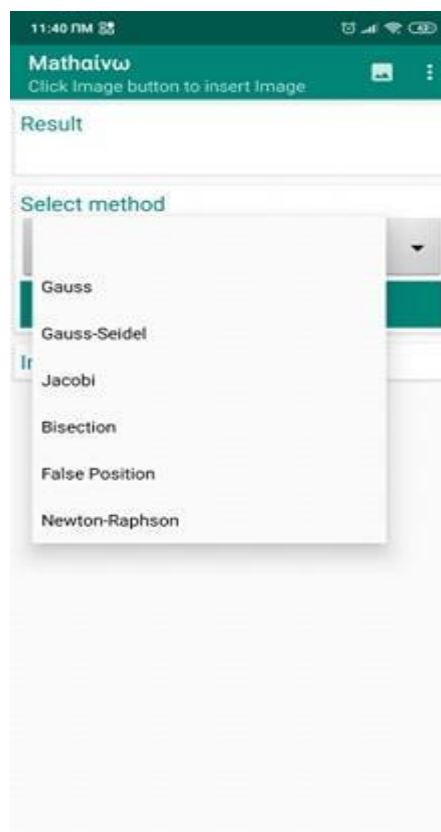
- Τη βασική οθόνη, στην οποία ο χρήστης δίνει την είσοδο που χρειάζεται και επιλέγει την κατάλληλη μέθοδο επίλυσης.
- Την οθόνη προβολής αποτελεσμάτων, στην οποία εμφανίζονται τα αποτελέσματα της εκάστοτε μεθόδου που επιλέχθηκε.

Στη συνέχεια θα δούμε αναλυτικά την εμφάνιση και το περιεχόμενο του κάθε Activity.

## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ



Σχήμα 3.1 Αρχική οθόνη της εφαρμογής



Σχήμα 3.2 Επιλογή μεθόδου επίλυσης

## ΚΕΦΑΛΑΙΟ 3

Στο Σχήμα 3.1 παρατηρούμε ότι η αρχική οθόνη είναι χωρισμένη σε διάφορα μέρη. Ας δούμε αναλυτικά τα μέρη που συνθέτουν την αρχική οθόνη:

### **Εικονίδιο φωτογραφίας:**

Όπως έχουμε αναφέρει, ο χρήστης έχει τη δυνατότητα είτε χειροκίνητα, είτε με μια φωτογραφία να παρέχει την είσοδο στην εφαρμογή, τα δεδομένα δηλαδή που θέλει να αναλύσει και χρησιμοποιώντας την κατάλληλη μέθοδο, να πάρει τα αποτελέσματα που χρειάζεται.

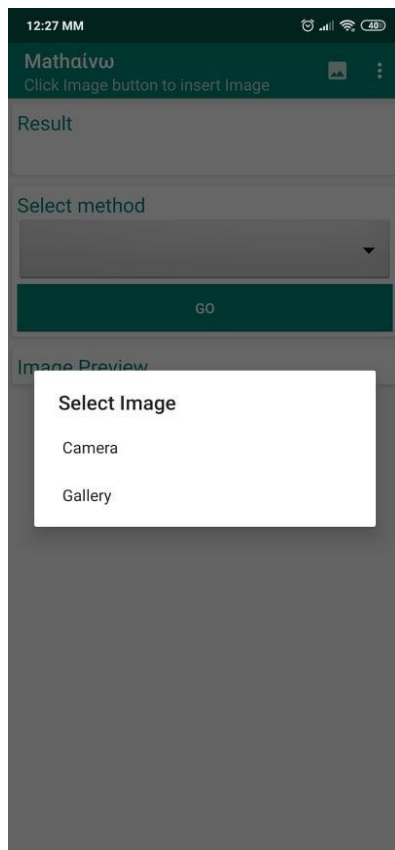
Στο πάνω δεξιά άκρο της οθόνης βρίσκουμε το εικονίδιο αυτό, το οποίο αποτελεί ένα επιπλέον εργαλείο για την εφαρμογή και δεν είναι απαραίτητη η χρήση του. Ας δούμε λοιπόν τι συμβαίνει όταν επιλέξουμε το εικονίδιο αυτό:

1. Αρχικά εμφανίζεται η επιλογή να φορτώσουμε μια φωτογραφία από τη συλλογή ή να τραβήξουμε μία καινούργια. (Σχήμα 3.3)
  - Στην περίπτωση που επιλέξουμε να φορτώσουμε από τη συλλογή, ανοίγει αυτόματα η συλλογή φωτογραφιών του κινητού μας, ώστε να διαλέξουμε τη φωτογραφία που μας ενδιαφέρει.
  - Αν επιλέξουμε να τραβήξουμε μια καινούργια, ανοίγει αυτόματα η κάμερα του κινητού. Αφού βγάλουμε μια καινούργια φωτογραφία, επιλέγουμε αν μας ικανοποιεί και θέλουμε να την χρησιμοποιήσουμε, ή αν θέλουμε να δοκιμάσουμε ξανά, τραβώντας μια καινούργια, από τα αντίστοιχα εικονίδια. (Σχήμα 3.4)
2. Αφού επιλέξουμε τη φωτογραφία που μας ενδιαφέρει, είτε από τη συλλογή, είτε τραβώντας μια καινούργια θα εμφανιστεί η οθόνη περικοπής της φωτογραφίας. Μπορούμε να επιλέξουμε ολόκληρη τη φωτογραφία ή να περικόψουμε ένα μέρος της για καλύτερα αποτελέσματα.

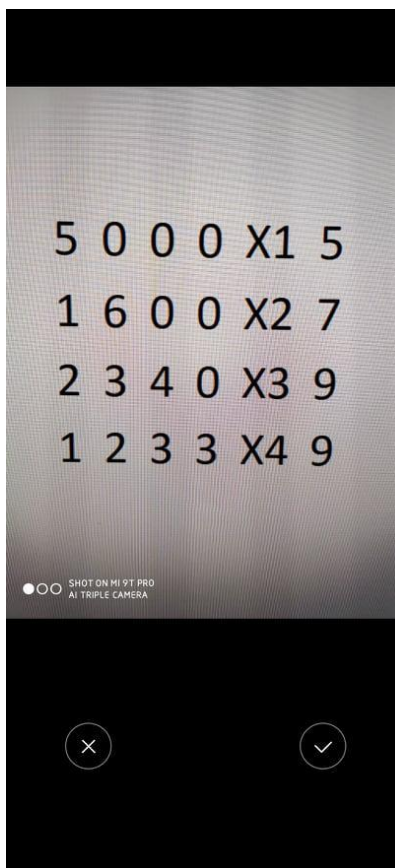
Στο σημείο αυτό παρέχονται και δυνατότητες για περιστροφή της φωτογραφίας είτε τον καθρεπτισμό της (κάθετη περιστροφή). Όταν ολοκληρώσουμε τις αλλαγές που θέλουμε να κάνουμε πατάμε το CROP που βρίσκεται στο πάνω δεξιά άκρο της οθόνης αυτής. Δίνεται επίσης η δυνατότητα επιστροφής, σε περίπτωση που αλλάξαμε γνώμη και θέλουμε να επιλέξουμε μια διαφορετική φωτογραφία. (Σχήμα 3.5)



## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

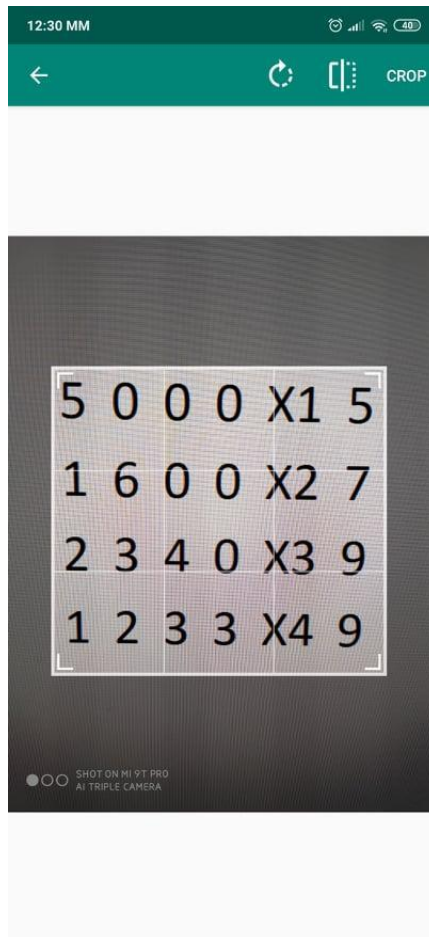


Σχήμα 3.3 Επιλογή τρόπου εισαγωγής φωτογραφίας



Σχήμα 3.4 Τράβηγμα καινούργιας φωτογραφίας και μενού επιλογής

## ΚΕΦΑΛΑΙΟ 3



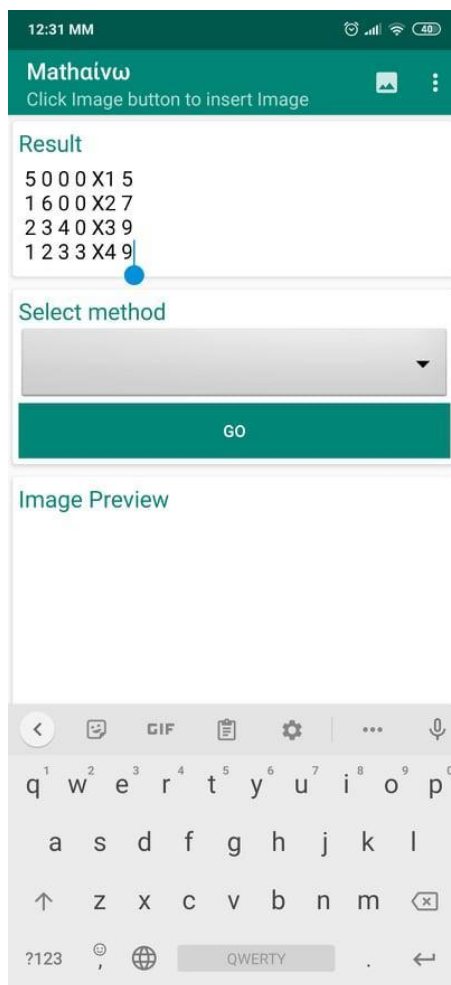
Σχήμα 3.5 Περικοπή φωτογραφίας

### Result:

Στο πάνω μέρος της οθόνης βλέπουμε το πεδίο με τίτλο Result. Πρόκειται για ένα editable Text View, το οποίο αποτελεί την είσοδο των δεδομένων που θέλουμε να επεξεργαστούμε. Στο πεδίο αυτό θα εμφανιστούν τα αποτελέσματα του OCR που χρησιμοποιούμε για να τραβήξουμε δεδομένα από μια φωτογραφία. (Σχήμα 3.6)

Έχει την ιδιότητα να είναι editable, δηλαδή επεξεργάσιμο, έτσι ώστε σε περίπτωση που η φωτογραφία δεν μετατραπεί με απόλυτη ακρίβεια στα δεδομένα που θέλουμε, να μπορούμε να κάνουμε τις μικρές αλλαγές που ίσως χρειάζονται. Επίσης κάποιος χρήστης μπορεί να μη θέλει να χρησιμοποιήσει αυτό το εργαλείο της εφαρμογής, οπότε έτσι δίνεται σε αυτόν η δυνατότητα να εισάγει τα δεδομένα χειροκίνητα.

## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ



Σχήμα 3.6 Αποτέλεσμα εισαγωγής φωτογραφίας - Μετατροπή σε κείμενο

### Select method:

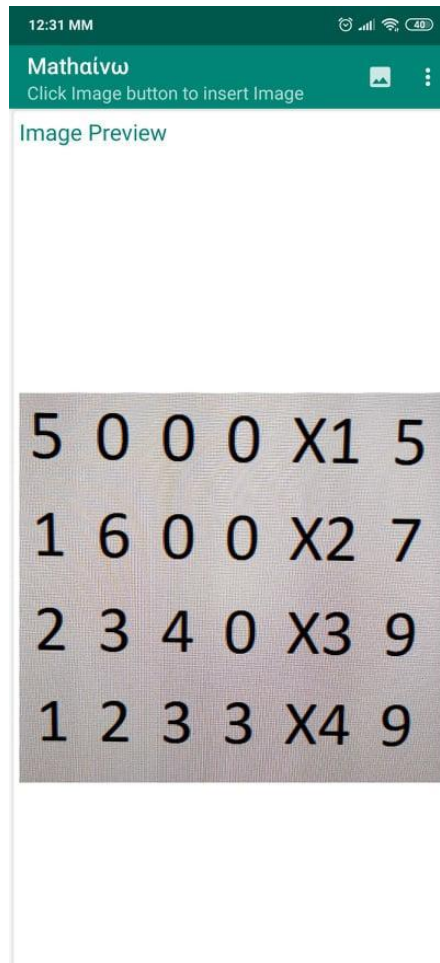
Σε αυτό το σημείο επιλέγουμε τη μέθοδο με την οποία θέλουμε να επεξεργαστούμε την είσοδο για να παράγουμε τα αποτελέσματα που επιθυμούμε. Πρόκειται για ένα αντικείμενο spinner, μέσα στο οποίο μπορούμε να αποθηκεύσουμε προκαθορισμένες επιλογές. Πατώντας πάνω στο βελάκι θα εμφανιστεί η λίστα με όλες τις διαθέσιμες μεθόδους της εφαρμογής. Συνέχεια επιλέγουμε πατώντας τη μέθοδο που μας ενδιαφέρει.

Θα παρατηρήσουμε πως υπάρχει και η δυνατότητα της κενής επιλογής. Στη συνέχεια θα δούμε αναλυτικά τι συμβαίνει όταν επιλέξουμε αυτή ή οποιαδήποτε από τις υπόλοιπες μαθηματικές μεθόδους. (Σχήμα 3.2)

### Image Preview:

Αφού επιλέξουμε, με οποιονδήποτε από τους δύο τρόπους που αναφέραμε, και περικόψουμε τη φωτογραφία που μας ενδιαφέρει θα παρατηρήσουμε πως το αποτέλεσμα της περικοπής θα εμφανιστεί σε αυτό το πεδίο. Πρόκειται για ένα Image View το οποίο με τη βοήθεια BitMaps καταφέρνει και ζωγραφίζει το αποτέλεσμα της περικοπής στο σημείο αυτό της οθόνης. (Σχήμα 3.7)

## ΚΕΦΑΛΑΙΟ 3



Σχήμα 3.7 Προεπισκόπηση περικοπής της φωτογραφίας

### 3.3 ΜΕΝΟΥ ΕΠΙΛΟΓΗΣ ΜΕΘΟΔΟΥ

Στην ενότητα αυτή θα αναλύσουμε όλες τις διαφορετικές επιλογές του μενού και θα δούμε πως λειτουργεί η εφαρμογή σε κάθε πιθανή περίπτωση.

#### 3.3.1 Κενή Επιλογή

Η πρώτη επιλογή του μενού επιλογής μεθόδου είναι η κενή επιλογή. Στην κατηγορία αυτή εντάσσονται οι απλούστερες μαθηματικές μέθοδοι. Πιο συγκεκριμένα θα δουμε τη λειτουργία της εφαρμογής ως κομπιουτεράκι, τη δυνατότητα επίλυσης εξισώσεων πρώτου και δευτέρου βαθμού, καθώς και την επίλυση Διαγώνιων Συστημάτων, Κάτω και Άνω Τριγωνικών Συστημάτων 4x4.

Ανάλογα με την είσοδο που θα δώσουμε η εφαρμογή αντιλαμβάνεται από μόνη της ποια μέθοδο πρέπει να χρησιμοποιήσει για να παράγει τα σωστά αποτελέσματα, με τη χρήση regular expressions.

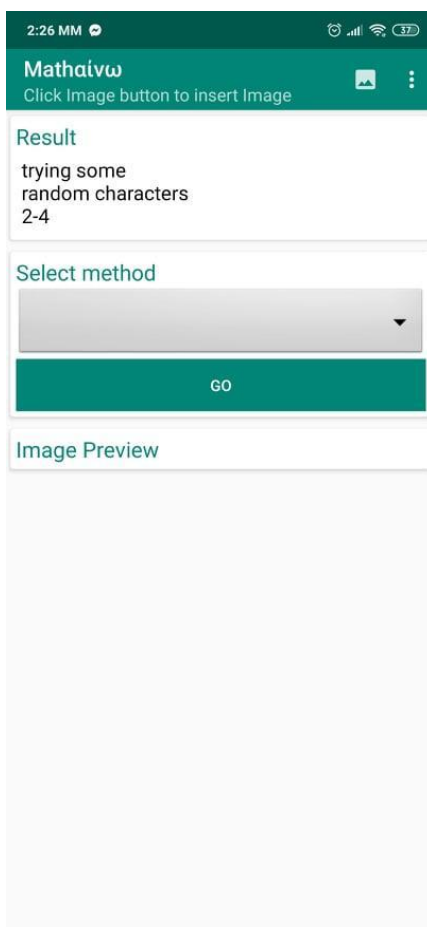
## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

### Λανθασμένη είσοδος:

Προφανώς, σε περίπτωση που η εφαρμογή λάβει ως είσοδο κάτι το οποίο δεν ανήκει σε καμία από τις κατηγορίες εισόδου που μπορεί να επιλύσει ή κάτι το οποίο δεν είναι μαθηματικά ορθό, θα εμφανίσει το αντίστοιχο μήνυμα λάθος και θα ζητήσει από τον χρήστη καινούργια είσοδο. (Σχήμα 3.8) (Σχήμα 3.9)

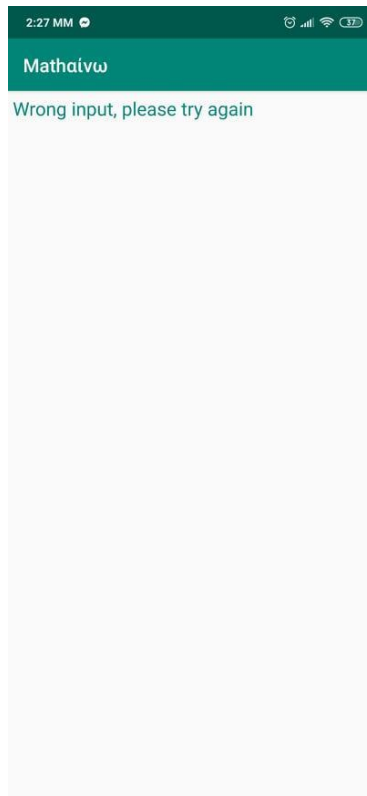
### Κομπιουτεράκι:

Αν δοθεί στην εφαρμογή ως είσοδος μια οποιαδήποτε μαθηματική παράσταση, χωρίς αγνώστους, η εφαρμογή θα υπολογίσει το αποτέλεσμα ακολουθώντας φυσικά τη σωστή προτεραιότητα των πράξεων, και θα εμφανίσει το αποτέλεσμα. Υποστηρίζονται οι κλασικές πράξεις της πρόσθεσης, της αφαίρεσης, του πολλαπλασιασμού και της διαίρεσης, καθώς και δυνάμεις. (Σχήμα 3.10) (Σχήμα 3.11)

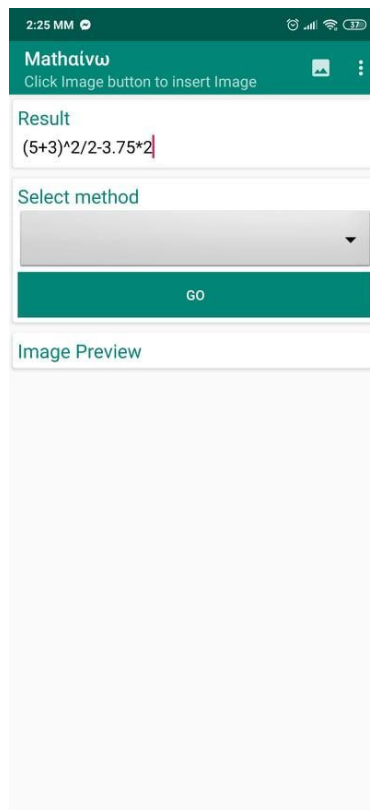


Σχήμα 3.8 Λανθασμένη είσοδος κενής επιλογής

## ΚΕΦΑΛΑΙΟ 3

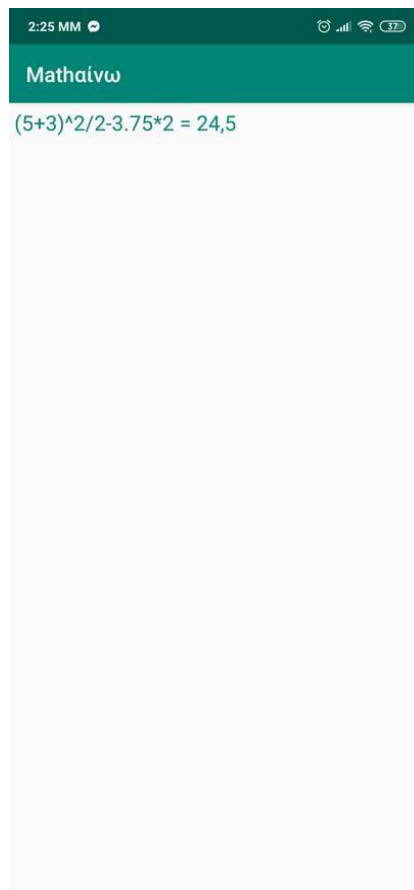


Σχήμα 3.9 Αποτέλεσμα λανθασμένης εισόδου κενής επιλογής



Σχήμα 3.10 Είσοδος για τη λειτουργία “κομπιουτεράκι”

## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ



Σχήμα 3.11 Αποτέλεσμα λειτουργίας “κομπιουτεράκι”

### **Διαγώνια Συστήματα, Κάτω Τριγωνικά Συστήματα, Άνω Τριγωνικά Συστήματα:**

Για τη διάκριση των τριών αυτών διαφορετικών Συστημάτων έχουμε δημιουργηθεί τρεις μέθοδοι, με τη βοήθεια των οποίων μπορούμε να διακρίνουμε αν η είσοδος που δόθηκε είναι ένα Διαγώνιο, ή ένα Κάτω ή Άνω Τριγωνικό Σύστημα.

Πρώτα από όλα, με το κατάλληλο κομμάτι κώδικα μετατρέπουμε την είσοδο κειμένου σε έναν πίνακα αριθμών. Στη συνέχεια ελέγχουμε σε ποια κατηγορία ανήκει η είσοδος με τη βοήθεια των μεθόδων `checkDiagonal()`, η οποία ελέγχει αν τα στοιχεία της διαγωνίου είναι μηδενικά, `checkDown()`, η οποία ελέγχει αν τα στοιχεία κάτω από τη διαγώνιο είναι μηδενικά και `checkUp()`, η οποία αντίστοιχα ελέγχει αν τα στοιχεία πάνω από τη διαγώνιο είναι μηδενικά. Όλες αυτές οι μέθοδοι επιστρέφουν `true` και `false`, σε περίπτωση που όλα τα στοιχεία είναι μηδενικά ή όχι αντίστοιχα. Πλέον είναι εύκολο να διακρίνουμε τις περιπτώσεις ως εξής:

- Διαγώνια Συστήματα: `checkDiagonal()==false`, `checkDown()==true`, `checkUp()==true` (Σχήμα 3.12) (Σχήμα 3.13)
- Κάτω Τριγωνικά Συστήματα: `checkDiagonal()==false`, `checkDown()==false`, `checkUp()==true` (Σχήμα 3.14) (Σχήμα 3.15)
- Άνω Τριγωνικά Συστήματα: `checkDiagonal()==false`, `checkDown()==true`, `checkUp()==false` (Σχήμα 3.16) (Σχήμα 3.17)

## ΚΕΦΑΛΑΙΟ 3



Σχήμα 3.12 Είσοδος Διαγώνιου Συστήματος



Σχήμα 3.13 Αποτέλεσμα Διαγώνιου Συστήματος



## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

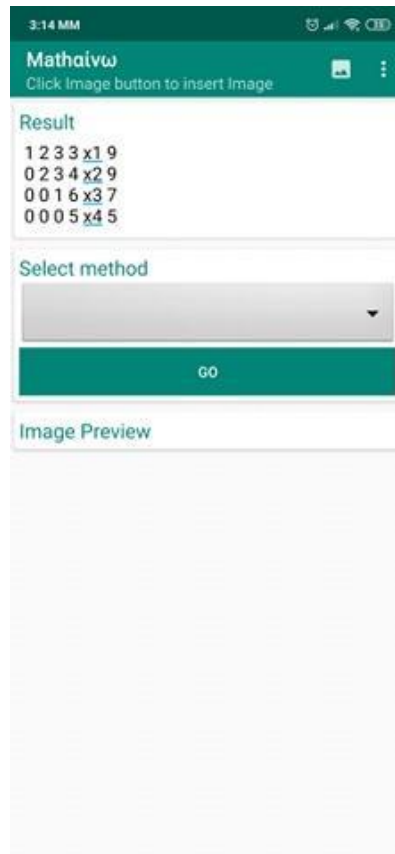


Σχήμα 3.14 Είσοδος Κάτω Τριγωνικού Συστήματος



Σχήμα 3.15 Αποτέλεσμα Κάτω Τριγωνικού Συστήματος

## ΚΕΦΑΛΑΙΟ 3



Σχήμα 3.16 Είσοδος Άνω Τριγωνικού Συστήματος



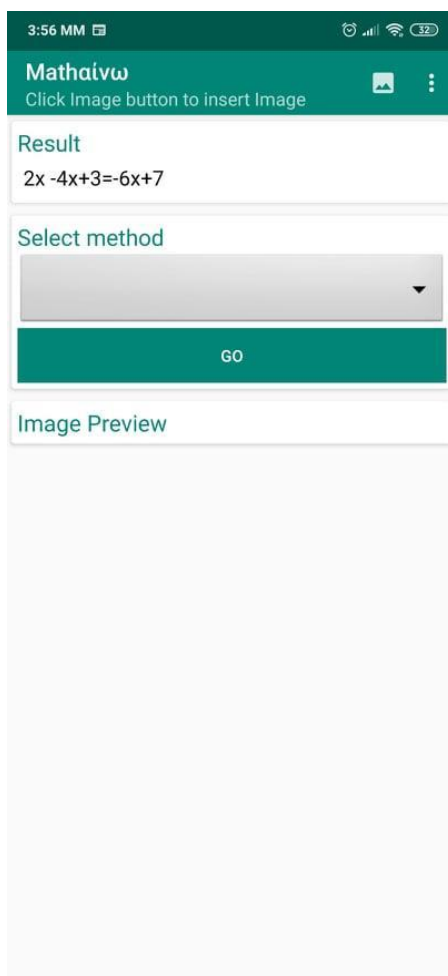
Σχήμα 3.17 Αποτέλεσμα Άνω Τριγωνικού Συστήματος

## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

### Εξισώσεις:

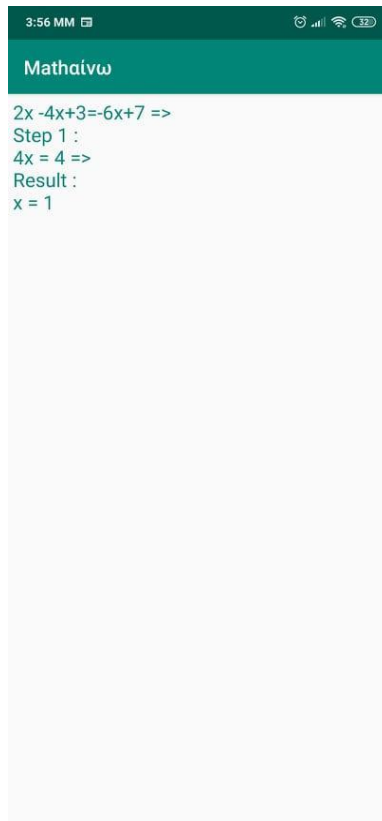
Η εφαρμογή αντιλαμβάνεται με τη βοήθεια regular expressions, αν η εξίσωση που δόθηκε είναι πρώτου ή δευτέρου βαθμού, και καλεί την κατάλληλη μέθοδο για να τη λύσει. Δεν είναι όλες οι εξισώσεις όμως ίδιες. Υπάρχουν κάποιες βασικές διαφορές μεταξύ τους και η εφαρμογή καλύπτει όλες τις πιθανές περιπτώσεις, οι οποίες παρουσιάζονται παρακάτω:

1. Πρώτου βαθμού:
  - a. Μία λύση (Σχήμα 3.18) (Σχήμα 3.19)
  - b. Καμία λύση (Σχήμα 3.20) (Σχήμα 3.21)
  - c. Άπειρες λύσεις (Σχήμα 3.22) (Σχήμα 3.23)
  
2. Δευτέρου βαθμού:
  - a. Μία λύση, αν η Διακρίνουσα είναι ίση μηδέν (Σχήμα 3.24) (Σχήμα 3.25)
  - b. Δύο λύσεις, αν η Διακρίνουσα είναι θετική (Σχήμα 3.26) (Σχήμα 3.27)
  - c. Καμία λύση στους πραγματικούς αριθμούς, αν η Διακρίνουσα είναι αρνητική. Λύνεται μόνο με τη χρήση φανταστικών αριθμών. (Σχήμα 3.28) (Σχήμα 3.29)

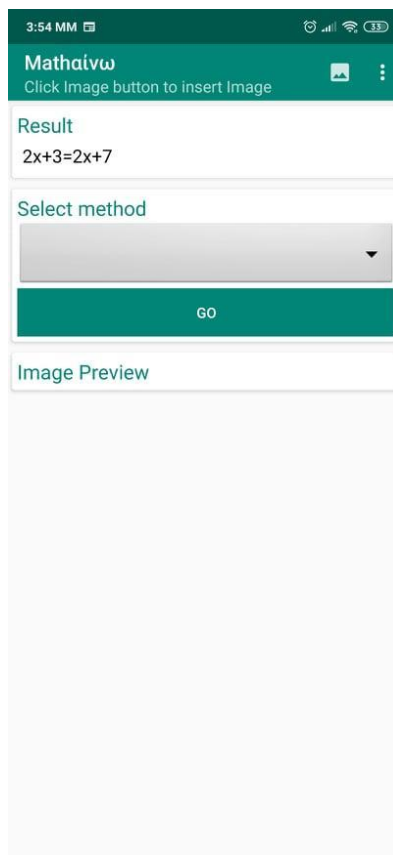


Σχήμα 3.18 Είσοδος εξίσωσης πρώτου βαθμού με μία λύση

## ΚΕΦΑΛΑΙΟ 3

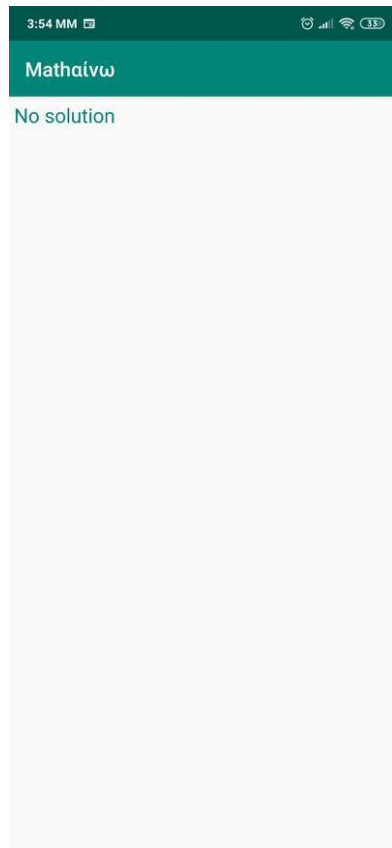


Σχήμα 3.19 Αποτέλεσμα εξίσωσης πρώτου βαθμού με μία λύση

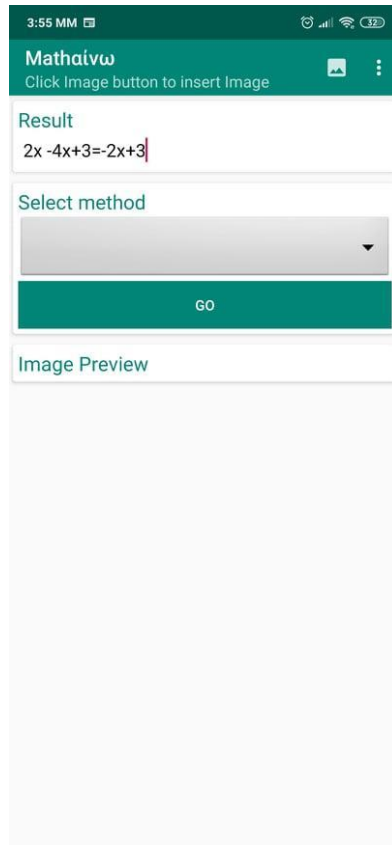


Σχήμα 3.20 Είσοδος εξίσωσης πρώτου βαθμού με καμία λύση

## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

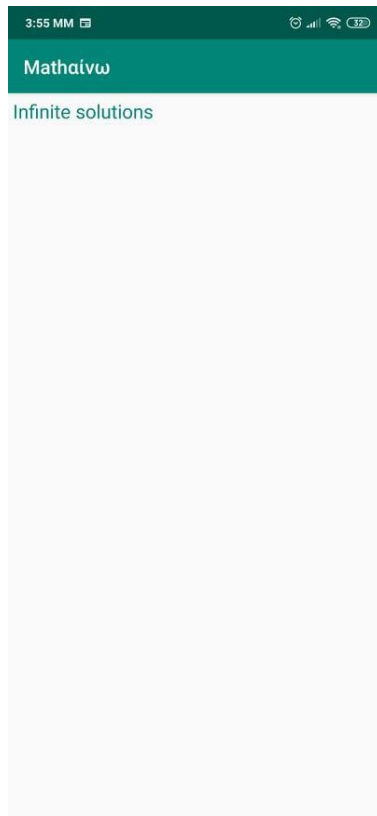


Σχήμα 3.21 Αποτέλεσμα εξίσωσης πρώτου βαθμού με καμία λύση

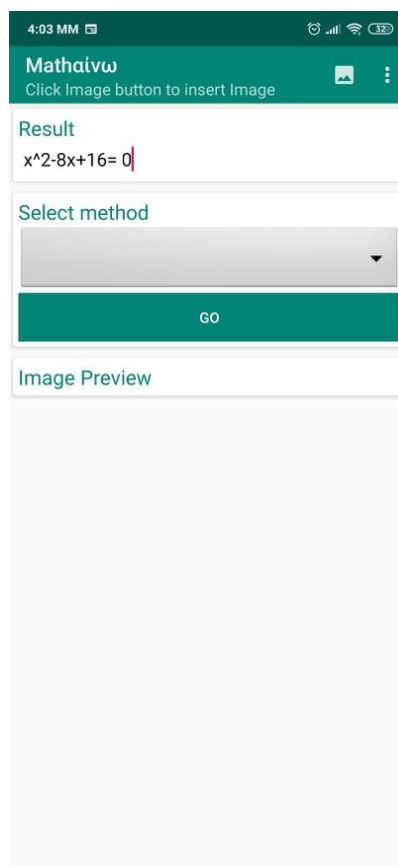


Σχήμα 3.22 Είσοδος εξίσωσης πρώτου βαθμού με άπειρες λύσεις

## ΚΕΦΑΛΑΙΟ 3

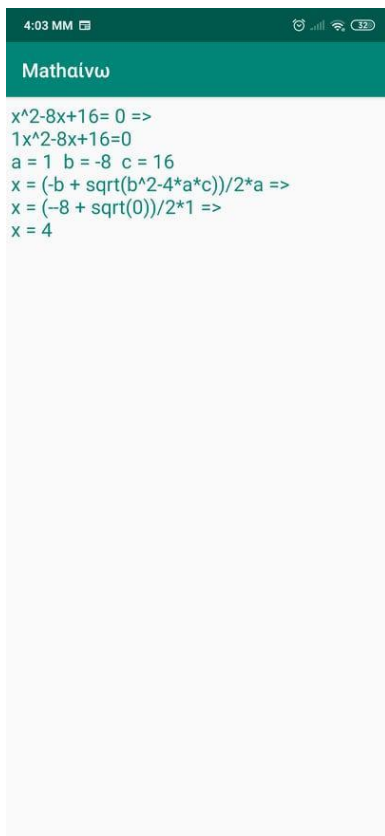


Σχήμα 3.23 Αποτέλεσμα εξίσωσης πρώτου βαθμού με άπειρες λύσεις

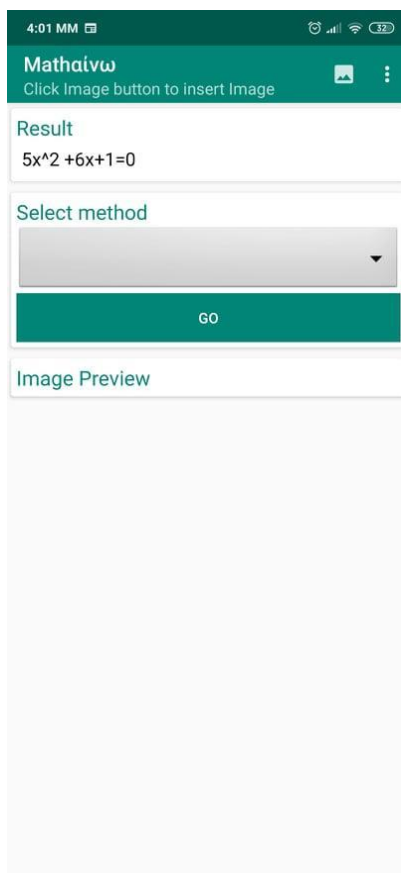


Σχήμα 3.24 Είσοδος εξίσωσης δευτέρου βαθμού με μία λύση

## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

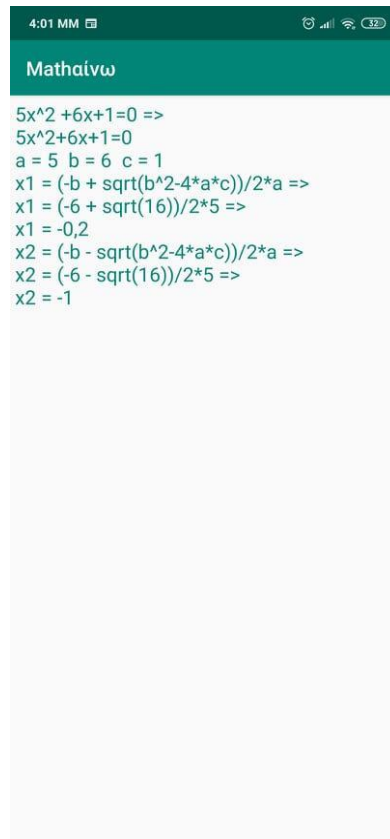


Σχήμα 3.25 Αποτέλεσμα εξίσωσης δευτέρου βαθμού με μία λύση

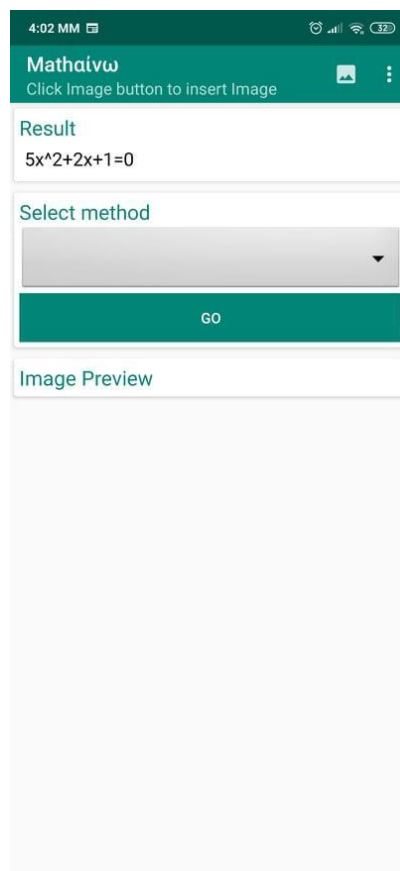


Σχήμα 3.26 Είσοδος εξίσωσης δευτέρου βαθμού με δύο λύσεις

## ΚΕΦΑΛΑΙΟ 3



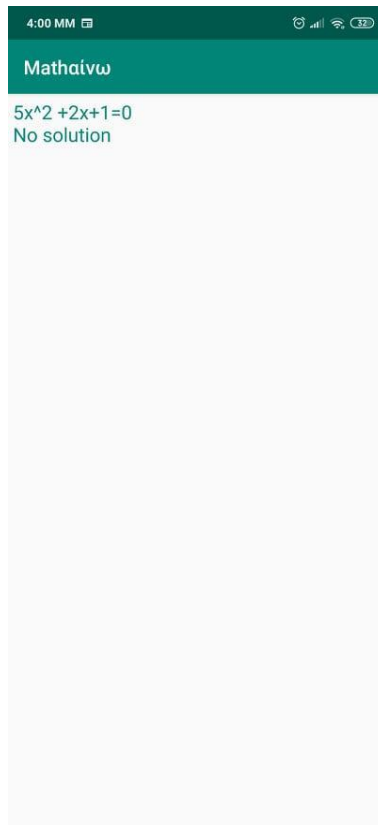
Σχήμα 3.27 Αποτέλεσμα εξίσωσης δευτέρου βαθμού με δύο λύσεις



Σχήμα 3.28 Είσοδος εξίσωσης δευτέρου βαθμού με καμία λύση



## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ



Σχήμα 3.29 Αποτέλεσμα εξίσωσης δευτέρου βαθμού με καμία λύση

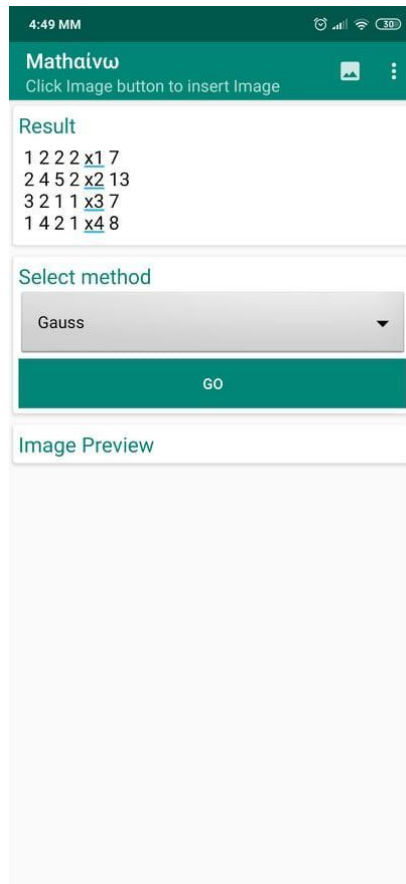
### 3.3.2 Gauss

Όπως αναφέραμε και στο Κεφάλαιο 2, η μέθοδος του Gauss λειτουργεί ως εξής:

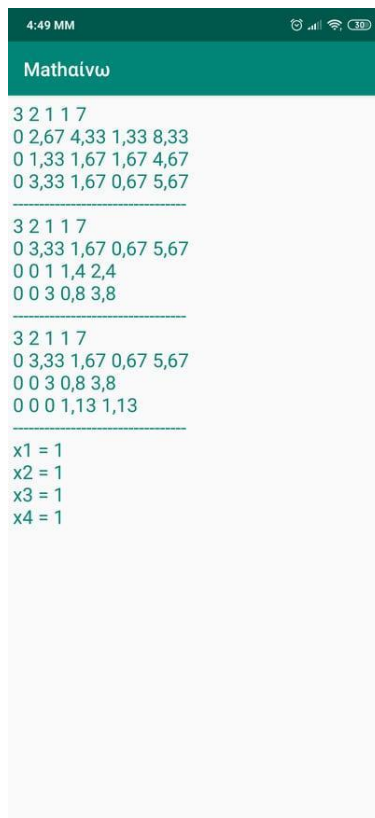
- Ξεκινώντας από την πρώτη, βρίσκουμε τη γραμμή που περιέχει τον μεγαλύτερο κατά απόλυτη τιμή συντελεστή στην αντίστοιχη στήλη και την ανταλλάζουμε με την τρέχουσα γραμμή.
- Πολλαπλασιάζουμε τις εξισώσεις με τους κατάλληλους συντελεστές
- Επαναλαμβάνουμε τα δύο πρώτα βήματα  $n-1$  φορές
- Το σύστημα έχει πλέον μετατραπεί σε Άνω Τριγωνικό Σύστημα και μπορεί να λυθεί

Στη συνέχεια θα δούμε ένα παράδειγμα της εφαρμογής που χρησιμοποιεί την απαλοιφή Gauss με μερική οδήγηση, εμφανίζοντας σε κάθε επανάληψη τον καινούργιο μετασχηματισμό του πίνακα. (Σχήμα 3.30) (Σχήμα 3.31)

### ΚΕΦΑΛΑΙΟ 3



Σχήμα 3.30 Είσοδος μεθόδου απαλοιφής Gauss με μερική οδήγηση



Σχήμα 3.31 Αποτέλεσμα μεθόδου απαλοιφής Gauss με μερική οδήγηση

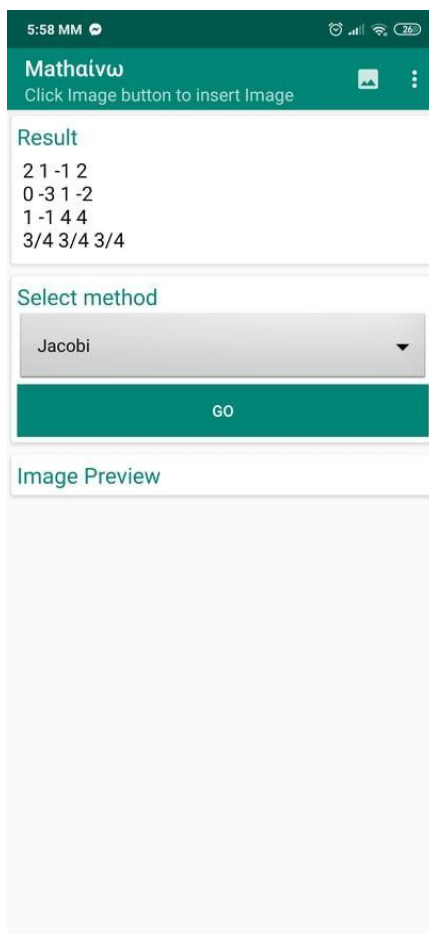
## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

### 3.3.3 Jacobi και Gauss - Seidel

Όταν επιλέγουμε από τη λίστα τη μέθοδο του Jacobi ή των Gauss - Seidel, η εφαρμογή δέχεται ως είσοδο το σύστημα εξισώσεων, καθώς και τις αρχικές τιμές του συστήματος. Έπειτα ελέγχει αν υπάρχει υπεροχή κατά γραμμές ή κατά στήλες, για να γνωρίζει αν η σύγκλιση του συστήματος είναι δεδομένη. Στην περίπτωση που δεν ισχύει καμία από τις δύο περιοχές, εμφανίζεται ο αντίστοιχο μήνυμα, και ο αλγόριθμος συνεχίζεται μέχρι να φτάσουμε το μέγιστο πλήθος επαναλήψεων.

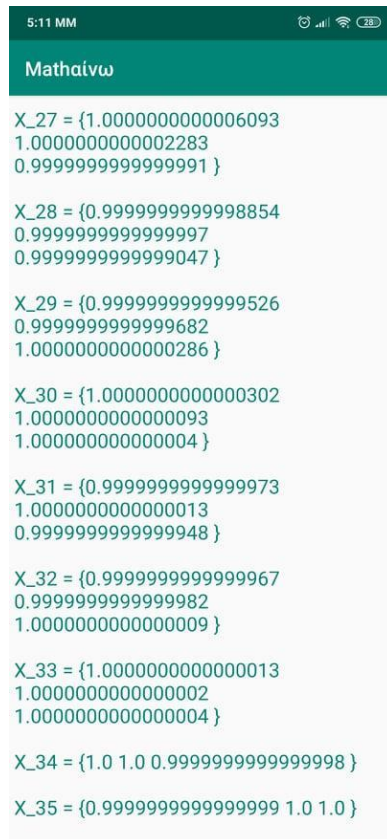
Η βασική διαφορά των δύο μεθόδων είναι ότι για την εύρεση μιας συνιστώσας της νέας επανάληψης, στη μέθοδο του Jacobi χρησιμοποιούνται όλες οι συνιστώσες της προηγούμενης επανάληψης, ενώ η μέθοδος των Gauss - Seidel χρησιμοποιεί τις πιο πρόσφατες διαθέσιμες συνιστώσες.

Στη συνέχεια θα παρουσιάσουμε τα αποτελέσματα της εφαρμογής όταν εισάγουμε τα ίδια δεδομένα και για τις δύο μεθόδους. Θα παρατηρήσουμε πως η μέθοδος των Gauss - Seidel συγκλίνει ταχύτερα από αυτή του Jacobi, πράγμα που την καθιστά πιο αποτελεσματική. (Σχήμα 3.32) (Σχήμα 3.33) (Σχήμα 3.34)(Σχήμα 3.35)



Σχήμα 3.32 Είσοδος μεθόδου Jacobi

## ΚΕΦΑΛΑΙΟ 3



```
5:11 MM
Mathαίνω

X_27 = {1.0000000000006093
1.0000000000002283
0.9999999999999991 }

X_28 = {0.9999999999998854
0.9999999999999997
0.9999999999999047 }

X_29 = {0.999999999999526
0.999999999999682
1.000000000000286 }

X_30 = {1.000000000000302
1.000000000000093
1.00000000000004 }

X_31 = {0.99999999999973
1.00000000000013
0.99999999999948 }

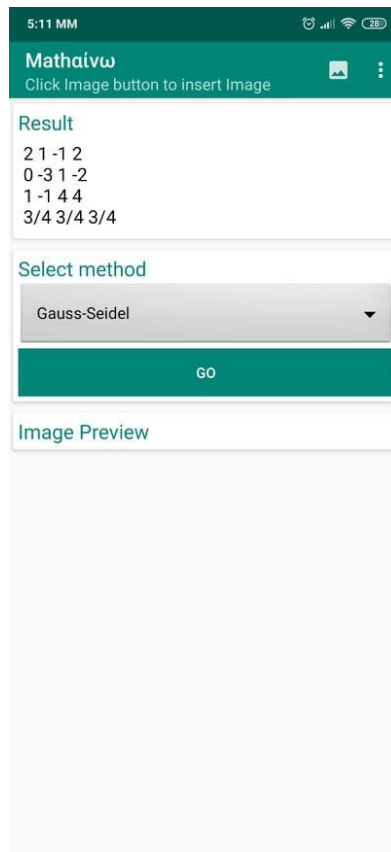
X_32 = {0.99999999999967
0.99999999999982
1.00000000000009 }

X_33 = {1.00000000000013
1.00000000000002
1.00000000000004 }

X_34 = {1.0 1.0 0.99999999999998 }

X_35 = {0.99999999999999 1.0 1.0 }
```

Σχήμα 3.33 Αποτέλεσμα μεθόδου Jacobi



5:11 MM

Mathαίνω

Click Image button to insert Image

Result

```
2 1 -1 2
0 -3 1 -2
1 -1 4 4
3/4 3/4 3/4
```

Select method

Gauss-Seidel

GO

Image Preview

Σχήμα 3.34 Είσοδος μεθόδου Gauss - Seidel

## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

```
5:11 MM
Mathaίνω
X_16 = {1.00000000001636
1.0000000000036025
0.9999999999968106 }
X_17 = {0.99999999996604
0.999999999989369
1.00000000000583 }
X_18 = {1.000000000008231
1.000000000001943
0.999999999998428 }
X_19 = {0.99999999998243
0.99999999999476
1.00000000000309 }
X_20 = {1.00000000000417
1.00000000000102
0.99999999999921 }
X_21 = {0.99999999999909
0.99999999999973
1.00000000000016 }
X_22 = {1.00000000000022
1.00000000000004
0.99999999999996 }
X_23 = {0.99999999999996
0.99999999999998 1.0 }
X_24 = {1.0 1.0 1.0 }
```

Σχήμα 3.35 Αποτέλεσμα μεθόδου Gauss - Seidel

### 3.3.4 Διχοτόμηση και Εσφαλμένη Θέση

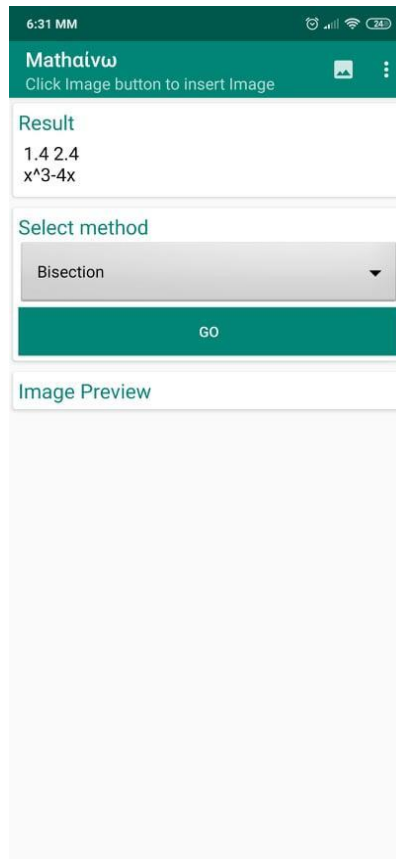
Όπως αναφέραμε και σε προηγούμενο κεφάλαιο, οι δύο μέθοδοι ισχύουν μόνο σε περίπτωση που ισχύει το θεώρημα του Bolzano. Και στις δύο περιπτώσεις, η εφαρμογή δέχεται σαν είσοδο το διάστημα των τιμών, καθώς και την εξίσωση που θέλουμε να μελετήσουμε. Σε περίπτωση που δεν ισχύει  $f(a)f(b) < 0$ , εμφανίζεται το αντίστοιχο μήνυμα λάθος και ζητείται από τον χρήστη να εισάγει καινούργιες τιμές. (Σχήμα 3.40)

Οι δύο μέθοδοι παρουσιάζουν αρκετές ομοιότητες:

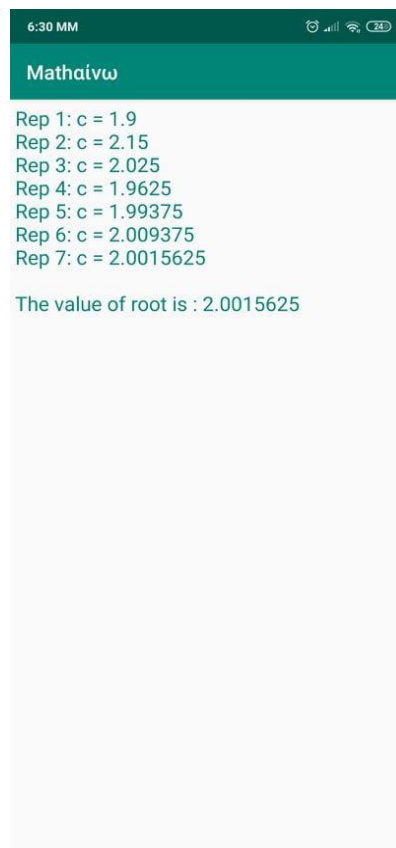
- Και οι δύο μέθοδοι θεωρούν δεδομένο ότι η συνάρτηση είναι συνεχής στο διάστημα  $[a,b]$  και ότι  $f(a)f(b) < 0$ .
- Συγκλίνουν πάντοτε: Η μέθοδος της Εσφαλμένης θέσης συνήθως αρκετά ταχύτερα, αλλά υπάρχουν και περιπτώσεις που η μέθοδος της Διχοτόμησης είναι γρηγορότερη.

Στη συνέχεια θα δούμε μια περίπτωση που η μέθοδος της Διχοτόμησης συγκλίνει όντως ταχύτερα. (Σχήμα 3.36) (Σχήμα 3.37) (Σχήμα 3.38) (Σχήμα 3.39)

## ΚΕΦΑΛΑΙΟ 3

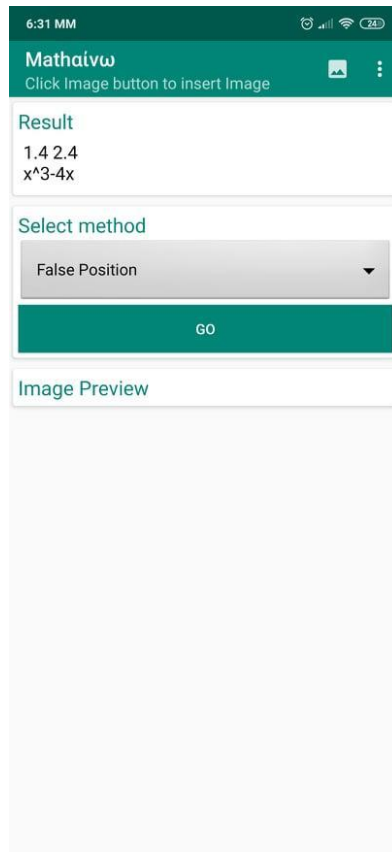


Σχήμα 3.36 Είσοδος μεθόδου Διχοτόμησης



Σχήμα 3.37 Αποτέλεσμα μεθόδου Διχοτόμησης

## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

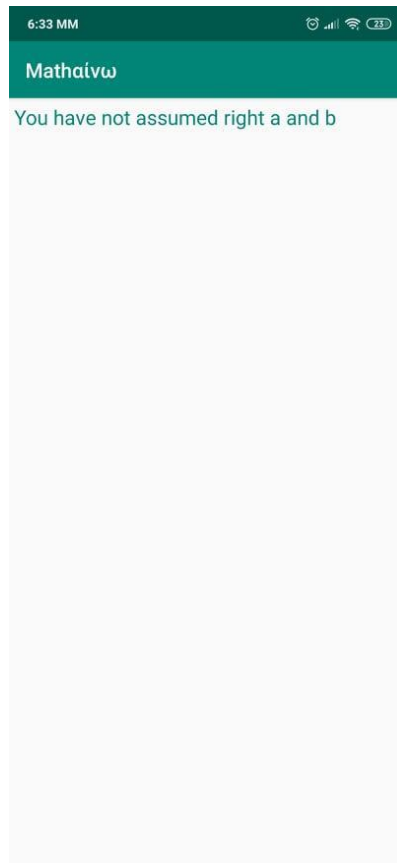


Σχήμα 3.38 Είσοδος μεθόδου Εσφαλμένης Θέσης



Σχήμα 3.39 Αποτέλεσμα μεθόδου Εσφαλμένης Θέσης

## ΚΕΦΑΛΑΙΟ 3



Σχήμα 3.40 Αποτέλεσμα λανθασμένης εισόδου Διχοτόμησης - Εσφαλμένης Θέσης

### 3.3.5 Newton - Raphson

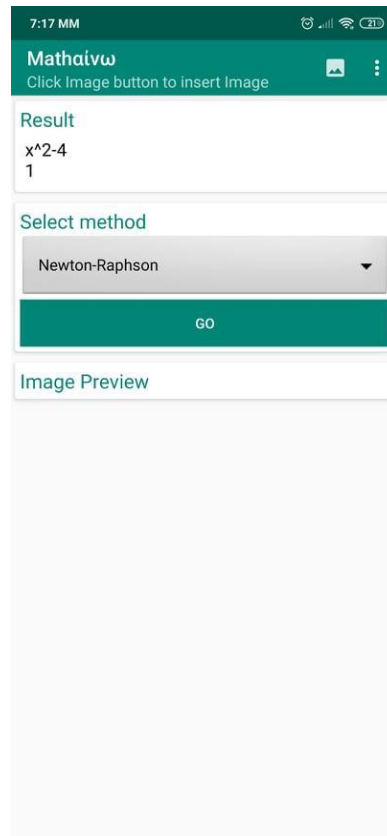
Όπως έχουμε αναφέρει, η μέθοδος Newton-Raphson αποτελεί μια από τις ταχύτερες μεθόδους προσέγγισης ρίζας μιας συνάρτησης. Για να λειτουργήσει σωστά απαιτεί την ύπαρξη πρώτης και δεύτερης παραγώγου της συνάρτησης.

Η εφαρμογή δέχεται σαν είσοδο μια συνάρτηση και μια αρχική τιμή για την προσέγγιση της ρίζας. Στη συνέχεια υπολογίζει την παράγωγο της συνάρτησης και την χρησιμοποιεί για να βρει την προσέγγιση της ρίζας.

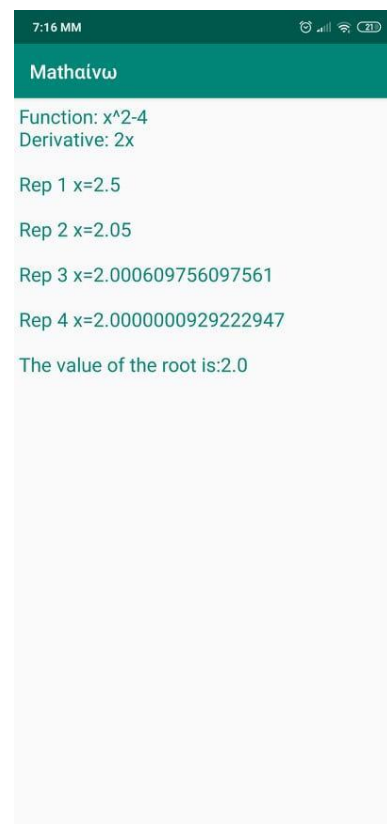
Σε περίπτωση που η παράγωγος είναι ίση με το μηδέν, η μέθοδος δε μπορεί να λειτουργήσει, οπότε εμφανίζεται το αντίστοιχο μήνυμα. (Σχήμα 3.43) Σε αντίθετη περίπτωση εμφανίζονται οι προσεγγίσεις σε κάθε επανάληψη, καθώς και το τελικό αποτέλεσμα της μεθόδου. (Σχήμα 3.41) (Σχήμα 3.42)



## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

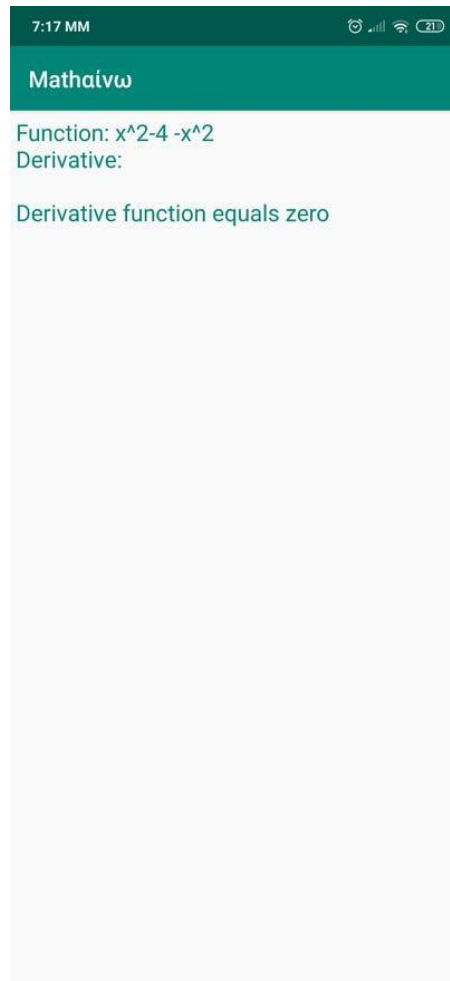


Σχήμα 3.41 Είσοδος μεθόδου Newton - Raphson



Σχήμα 3.42 Αποτέλεσμα μεθόδου Newton - Raphson

## ΚΕΦΑΛΑΙΟ 3



Σχήμα 3.43 Αποτέλεσμα μεθόδου Newton - Raphson - Παράγωγος μηδέν

### 3.4 ΕΠΙΛΟΓΟΣ

Έχοντας πλέον δει τις τεχνολογίες που χρησιμοποιεί η εφαρμογή, τις μαθηματικές μεθόδους, καθώς και πως αλληλεπιδρούν μεταξύ τους, ο αναγνώστης είναι πλέον σε θέση να αντιληφθεί το πως λειτουργεί η εφαρμογή.

Η εφαρμογή αυτή, είναι ένα πολύ καλό παράδειγμα πως κάτι το οποίο μπορεί με μια πρώτη ματιά να μοιάζει απλό, μπορεί να έχει βάθος αλλά και πολυπλοκότητα. Από τη μία, οι τεχνολογίες που χρησιμοποιεί μπορεί να είναι σχετικά απλές αν κάποιος τις δει μία προς μία. Αν όμως συνδέσεις όλες αυτές τις τεχνολογίες μεταξύ τους και με τις μαθηματικές μεθόδους που χρησιμοποιεί η εφαρμογή, δημιουργείται κάτι εντυπωσιακό. Κάτι που απλοποιεί σε μεγάλο βαθμό και αυτοματοποιεί μια διαδικασία που θα έπαιρνε αρκετή ώρα αν κάποιος την έκανε με το χέρι.

## ΠΡΟΤΑΣΕΙΣ ΒΕΛΤΙΩΣΗΣ

Έχοντας πλέον αναλύσει πλήρως τη λειτουργία και τις δυνατότητες της εφαρμογής, ας δούμε πως θα μπορούσε να γίνει ακόμα καλύτερη:

- Εισαγωγή περισσότερων μαθηματικών μεθόδων π.χ. Μέθοδος των Διαδοχικών Προσεγγίσεων, Μέθοδος της Χορδής κ.α
- Υποστήριξη περισσότερων γλωσσών. Επιπλέον θα ήταν πολύ καλό αν μπορούσε η εφαρμογή μέσω της ip του χρήστη να αντιληφθεί ποια είναι η προτιμώμενη γλώσσα του
- Δυνατότητα εναλλαγής θεμάτων και χρωμάτων της εφαρμογής με βάση την προτίμηση του χρήστη
- Δυνατότητα επικοινωνίας του χρήστη με τον Developer, προκειμένου να αναφέρει τυχόν προβλήματα που αντιμετωπίζει ή δικές του προτάσεις για βελτίωση
- Δυνατότητα εξαγωγής αποτελεσμάτων σε μορφές pdf ή xls
- Δημιουργία λογαριασμού για κάθε χρήστη και προσθήκη της δυνατότητας να μπορεί να δει το ιστορικό με τις μεθόδους που χρησιμοποίησε και τις εισόδους που είχε δώσει

## ΑΝΑΦΟΡΕΣ

[1] [https://en.wikipedia.org/wiki/Google\\_Play](https://en.wikipedia.org/wiki/Google_Play)

[2] Byous, Jon (c. 1998). "[Java technology: The early years](#)". Sun Developer Network. [Sun Microsystems](#).

[3] "[1.2 Design Goals of the Java™ Programming Language](#)". Oracle. January 1, 1999

[4] <https://el.wikipedia.org/wiki/XML>

[5] Copyright 2016, Arthur Teplitzki, 2013, Edmodo, Inc. (<https://github.com/ArthurHub/Android-Image-Cropper>)

# ΒΙΒΛΙΟΓΡΑΦΙΑ

1. <https://hipwallpaper.com/math-wallpapers/>
2. <https://logos-download.com/10695-java-logo-download.html>
3. <https://developer.android.com/studio>
4. "Google's Android OS: Past, Present, and Future". PhoneArena. August 18, 2011
5. "Getting Started with Android Studio". Android Developers
6. The Whetstone of Witte, 1557, Robert Record
7. <https://stackoverflow.com/>