



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Σχολή Πολιτικών Μηχανικών  
Καθίσμα Δομοστατικής

ΑΝΑΠΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΡΑΜΜΙΚΗΣ  
ΑΝΑΛΥΣΗΣ ΧΩΡΙΚΩΝ ΦΟΡΕΩΝ ΣΕ ΓΛΩΣΣΑ  
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΡΥΘΜΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΚΕΣΚΙΝΗΣ ΗΛΙΑΣ

Επιβλέπων: Νικόλαος Λαγαρός

Συνεπιβλέπων: Αθανάσιος Στάμος

Αθήνα, Οκτώβριος 2018

## Περίληψη

Με την παρούσα εργασία υλοποιείται η ανάπτυξη προγράμματος επίλυσης χωρικών φορέων με τη μέθοδο της γραμμικής ανάλυσης, με τη χρήση της γλώσσας προγραμματισμού Python, που θα περιλαμβάνει και γραφικό περιβάλλον. Επίσης, εκτός από τα αρχεία, που δημιουργήθηκαν αρχικά και που λειτουργούν με απαραίτητη προϋπόθεση την εγκατάσταση της γλώσσας Python, καθώς και διαφόρων βιβλιοθηκών, επιχειρείται να εξαχθεί ένα αρχείο σε εκτελέσιμη μορφή, σε περιβάλλον Windows 10. Η μέθοδος ανάλυσης που χρησιμοποιήθηκε είναι η Μέθοδος της Άμεσης Δυσκαμψίας-Στιβαρότητας (Direct Stiffness Method), ενώ για το γραφικό περιβάλλον, χρησιμοποιήθηκε το PyQt5, που αποτελεί ένα ολοκληρωμένο και ευρέως χρησιμοποιούμενο πλαίσιο ανάπτυξης εφαρμογών σε γλώσσα Python, και βασίζεται στο Qt5. Άλλες βιβλιοθήκες που χρησιμοποιήθηκαν είναι το NumPy και το Pandas. Το πρόγραμμα αποτελείται από δύο αρχεία εκτέλεσης υπολογισμών, ένα αρχείο που πρωταρχικός στόχος αποτελεί η συλλογή και αποθήκευση δεδομένων από τον χρήστη και τέλος, το αρχείο που υλοποιεί το γραφικό περιβάλλον. Η επιβεβαίωση της ορθότητας των αποτελεσμάτων της ανάλυσης έγινε με σύγκριση των αποτελεσμάτων της επίλυσης ενός συγκεκριμένου φορέα με το πρόγραμμα SkyCiv. Σε γενικές γραμμές, τα αποτελέσματα αποδεικνύονται αξιόπιστα, παρ'όλ'αυτά, κρίνεται ότι απαιτείται περαιτέρω έλεγχος, καθώς διαφορετικές περιπτώσεις και ιδιαιτερότητες φορέων μπορεί να οδηγήσουν σε σφάλματα που δεν έχουν προβλεφτεί έως τώρα.

## **Abstract**

This paper presents the developing of a structural analysis program (both the calculation part and the graphical user interface) by using the programming language Python. Besides Python files, that require the installment of the language and some of its libraries, there has been an attempt to build an executable file that runs on Windows 10. The analysis is conducted with the direct stiffness method. The graphical user interface was built with PyQt5, a fully featured and broadly used framework for GUI development. Numpy and Pandas are Python libraries that were also used extensively. The program consists of two files that perform the analysis. One file whose primary function is the collection and management of data and another one that constructs the graphical user interface. The confirmation of the outcome of the program was carried out in respect to the results of a commercial program named SkyCiv. In general, the values of the two programs coincide, however, there is need for further assessment, as the distinctiveness of each construction that is being analyzed in the program, may lead to incorrect results.

## Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα καθηγητή της παρούσας διπλωματικής εργασίας, κ. Νικόλαο Λαγαρό, για την ευκαιρία που έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον για εμένα θέμα, που συνδυάζει το αντικείμενο του προγραμματισμού με τη στατική ανάλυση των κατασκευών. Επίσης, ιδιαίτερες ευχαριστίες οφείλω στον διδάκτορα κ. Αθανάσιο Στάμο, για την ουσιώδη ενθάρρυνση και καθοδήγηση που μου προσέφερε, καθώς και σε όλο το διδακτικό προσωπικό του τμήματος των Πολιτικών Μηχανικών τόσο για τις γνώσεις, όσο και για τα ερεθίσματα που αποκόμισα. Τέλος, θα ήθελα να ευχαριστήσω την σύζυγό μου, Ελένη, και τα παιδιά μου, που με υπομονή με στήριξαν καθ'όλη τη διάρκεια των σπουδών μου και τον συνάδελφο και συμφοιτητή μου, Σπαγγούρο Κώστα, για την άριστη συνεργασία μας όλο αυτό το διάστημα.



# Πίνακας Περιεχομένων

## Περίληψη

## Abstract

## Ευχαριστίες

## Πίνακας Περιεχομένων

### 1. Εισαγωγή

1.1 Πρόλογος. . . . .	9
1.2 Αντικείμενο της Διπλωματικής. . . . .	9
1.3 Μεθοδολογία. . . . .	10

### 2. Θεωρητικό Υπόβαθρο

2.1 Στάδια ανάλυσης κατασκευών . . . . .	13
2.2 Στατική ανάλυση. . . . .	13
2.3 Προσομοίωση φέροντος οργανισμού. . . . .	14
2.4 Διαδικασία ελαστικής ανάλυσης. . . . .	14
2.4.1 Γενικά. . . . .	14
2.4.2 Αναλυτική περιγραφή Μεθόδου Άμεσης Στιβαρότητας. . . . .	15
2.5 Γλώσσα προγραμματισμού Python. . . . .	20
2.5.1 Ορισμός. . . . .	20
2.5.2 Ιστορικά στοιχεία. . . . .	22
2.5.3 Χαρακτηριστικά της Python. . . . .	22
2.6 Εργαλεία - Βιβλιοθήκες. . . . .	23
2.6.1 Γενικά. . . . .	23
2.6.2 NumPy. . . . .	23
2.6.3 Pandas. . . . .	24
2.6.4 Sys. . . . .	24
2.6.5 PyQt5. . . . .	24
2.6.6 CxFreeze. . . . .	25
2.6.7 Άλλες Βιβλιοθήκες. . . . .	26

### 3. Ανάλυση Προγράμματος

3.1 Γενικά. . . . .	27
3.1.1 Εκτέλεση του προγράμματος σε περιβάλλον Windows 10. . . . .	27
3.1.2 Εκτέλεση του προγράμματος σε περιβάλλον Python. . . . .	27
3.1.3 Συστήματα συντεταγμένων. . . . .	27

3.1.4 Μονάδες μέτρησης. . . . .	29
3.2 Υλοποίηση Προγράμματος. . . . .	29
3.2.1 Τα αρχεία του προγράμματος . . . . .	29
3.2.2 Το αρχείο analysis3Dframe.py. . . . .	29
3.2.3 Το αρχείο frame3d.py. . . . .	32
3.2.3.1 Γενικά στοιχεία. . . . .	32
3.2.3.2 Αναλυτική Περιγραφή. . . . .	33
3.2.3.3 Η δομή ‘Loads’. . . . .	37
3.2.3.4 Λειτουργίες (functions) . . . . .	39
3.2.4 Το αρχείο ireact3d.py. . . . .	42
3.2.5 Το αρχείο GUI.py. . . . .	44
3.2.5.1 Γενικά. . . . .	44
3.2.5.2 Σύντομη περιγραφή. . . . .	44
3.2.5.3 Η κλάση MainWindow. . . . .	47
3.2.5.3.1 Γραμμή εργαλείων και μενού. . . . .	54
3.2.5.3.1 Πλαίσιο Σύνοψης Δεδομένων. . . . .	55
3.2.5.3.1 Παράθυρο τρισδιάστατης προβολής. . . . .	56
3.2.5.3.1 Καρτέλες Εισαγωγής Δεδομένων. . . . .	57
3.2.5.4 Η κλάση StructureViewer. . . . .	59
3.2.5.5 Η κλάση AnalysisWindow. . . . .	62
3.2.5.6 Η κλάση ViewResultsWindow. . . . .	63
3.2.5.7 Η λειτουργία main. . . . .	63
3.2.6 Το αρχείο setup.py. . . . .	64
<b>4. Χρήση του προγράμματος</b>	
4.1 Εγκατάσταση. . . . .	65
4.2 Έναρξη του προγράμματος. . . . .	66
4.2.1 Γενικά. . . . .	66
4.2.2 Δημιουργία νέας εργασίας. . . . .	67
4.2.3 Άνοιγμα αποθηκευμένης εργασίας. . . . .	69
4.3 Εισαγωγή δεδομένων. . . . .	69
4.3.1 Διατομές. . . . .	69
4.3.2 Μέλη πλαισίου. . . . .	71
4.3.3 Στηρίξεις. . . . .	73
4.3.4 Μετατοπίσεις – Στροφές κόμβων. . . . .	74

4.3.5	Εσωτερικές ελευθερώσεις. . . . .	75
4.3.6	Στερεοί κόμβοι. . . . .	75
4.3.7	Φορτία κόμβων. . . . .	76
4.3.8	Φορτία μελών. . . . .	77
4.4	Προβολή μοντέλου. . . . .	79
4.5	Ανάλυση κατασκευής. . . . .	81
4.6	Επισκόπηση αποθηκευμένων αναλύσεων. . . . .	83
4.7	Μενού κύριας λίστας. . . . .	83
4.8	Αποθήκευση εργασίας. . . . .	84
4.9	Τερματισμός προγράμματος. . . . .	84
4.10	Ανεπιθύμητες λειτουργίες. . . . .	84
<b>5. Εφαρμογή</b>		
5.1	Γενικά. . . . .	86
5.2	Περιγραφή του προγράμματος SkyCiv. . . . .	86
5.3	Περιγραφή του μοντέλου του φορέα. . . . .	86
5.4	Ανάλυση του φορέα και εξαγωγή αποτελεσμάτων. . . . .	88
<b>6. Επίλογος</b>		
6.1	Γενικά. . . . .	96
6.2	Δυσχέρειες και Αντιμετώπιση . . . . .	96
6.3	Συμπεράσματα . . . . .	97
6.4	Προτάσεις. . . . .	98
Βιβλιογραφία / Παραπομπές. . . . .		99



## Κεφάλαιο 1: Εισαγωγή

### 1.1 Πρόλογος

Ιστορικά, οι ανθρώπινες ανάγκες για ολοένα και πιο σύνθετες δομικές κατασκευές, αποτέλεσαν τον μοχλό ανάπτυξης εκείνων των μεθόδων, που θα του έδιναν τη δυνατότητα να επιτύχει τον σκοπό του, με τον όσο το δυνατόν μεγαλύτερο έλεγχο σε ότι αφορά το τελικό επίπεδο ασφάλειας αλλά και την πρόβλεψη του κόστους κατασκευής. Έτσι, προέκυψε μια ακολουθία σχεδιασμού, ανάλυσης, αξιολόγησης και κατασκευής, προκειμένου να επιτευχθεί η περάτωση μιας κατασκευής σε χρονικά οικονομικά και χρονικά πλαίσια, που να ικανοποιεί προεπιλεγμένα κριτήρια ασφαλείας και λειτουργικότητας και να εξυπηρετεί συγκεκριμένες ανάγκες.

Η ανάλυση των κατασκευών έχει πρωταρχικό σκοπό τον υπολογισμό της απόκρισης ενός δομικού φορέα σε ένα σύνολο εξωτερικών δράσεων. Είναι δηλαδή η διαδικασία καθορισμού της συνολικής εντατικής και παραμορφωσιακής κατάστασης μιας κατασκευής, καθώς και των εσωτερικών εντάσεων και παραμορφώσεων του κάθε μέλους ξεχωριστά, που συμμετέχει στον φέροντα οργανισμό, με απώτερο στόχο να εξασφαλίσει ο μηχανικός ότι οι επιβαλλόμενες δράσεις δεν θα υπερβούν τις επιτρεπόμενες των δομικών στοιχείων που απαρτίζουν την κατασκευή.

### 1.2 Αντικείμενο της Διπλωματικής

Στην παρούσα εργασία επιχειρείται η ανάπτυξη προγράμματος ΗΥ, με δυνατότητα ανάλυσης χωρικών φορέων. Περιλαμβάνει γραφικό περιβάλλον (Graphical User Interface - GUI), παρουσίαση του φορέα με χρήση τρισδιάστατων γραφικών και τελική παρουσίαση αποτελεσμάτων.

### 1.3 Μεθοδολογία

Η μέθοδος ανάλυσης που χρησιμοποιήθηκε είναι η Μέθοδος της Άμεσης Δυσκαμψιάς-Στιβαρότητας (Direct Stiffness Method), η οποία αποτελεί την εφαρμογή της Μεθόδου των Πεπερασμένων Στοιχείων σε ραβδωτούς φορείς. Είναι η βάση για τους υπολογισμούς στο σύνολο σχεδόν των λογισμικών που εκτελούν ανάλυση κατασκευών.

Η γλώσσα προγραμματισμού με την οποία συντάχθηκε το λογισμικό είναι η Python, ένα σύγχρονο εργαλείο με πληθώρα βιβλιοθηκών που καλύπτουν μεγάλο εύρος αναγκών. Πιο συγκεκριμένα, το πακέτο βιβλιοθηκών που χρησιμοποιήθηκε, είναι το Anaconda 3, που κάνει χρήση της Python 3 και περιέχει μια τεράστια συλλογή εργαλείων. Η μοναδική βιβλιοθήκη που εγκαταστάθηκε επιπλέον είναι το PyQt5 που χρησιμοποιήθηκε για την ανάπτυξη του περιβάλλοντος εργασίας. Στην περίπτωση που δεν χρησιμοποιηθεί το Anaconda, θα πρέπει να συλλεχθούν και εγκατασταθούν όλες οι βιβλιοθήκες ξεχωριστά.

Η μέθοδος εργασίας που ακολουθήθηκε, οργανώθηκε εξ αρχής σε επιμέρους βήματα, με κατάτμηση της συνολικής εργασίας σε μικρότερες, κυρίως λόγω του όγκου του κώδικα που απαιτήθηκε αλλά και της διαφορετικότητας του κάθε υποτμήματος.

Τα διακριτά βήματα που ακολουθήθηκαν είναι τα εξής:

α. Διαχωρισμός της συνολικής εργασίας σε πέντε επιμέρους τμήματα:

- Διαδικασία εισαγωγής δεδομένων από τον χρήστη.
- Ανάπτυξη προγράμματος εκτέλεσης υπολογισμών.
- Ανάπτυξη γραφικού περιβάλλοντος προγράμματος.
- Παρουσίαση φορέα σε τρεις διαστάσεις.
- Εξαγωγή και παρουσίαση αποτελεσμάτων.

Ενδεικτικά, λόγοι που οδήγησαν στην παραπάνω κατάτμηση, είναι:

(1) Η δυνατότητα ανεξαρτητοποίησης της κάθε διαδικασίας από τις υπόλοιπες, τουλάχιστον στα αρχικά στάδια της συγγραφής. Για παράδειγμα, η

εκτέλεση των υπολογισμών μπορεί να αποτελέσει μέχρι το τέλος εντελώς ανεξάρτητη εργασία. Επίσης, η συλλογή δεδομένων από τον χρήστη δεν αποτελεί ακριβώς ξεχωριστή διαδικασία ανάπτυξης, αλλά λόγω του κατά περίπτωση μεγάλου πλήθους δεδομένων, είναι σημαντικό και, όπως εκτελέστηκε στο παρόν, αυτοτελές τμήμα. Καθορίζει τους γενικούς και ειδικούς κανόνες σύνταξης του περιβάλλοντος εργασίας, τη μορφή των δεδομένων που θα εισαχθούν και τους ελέγχους που θα εκτελούνται ώστε να αποφευχθεί η πιθανότητα σφαλμάτων εισαγωγής και να βελτιωθεί η χρηστικότητα του προγράμματος.

(2) Η λογική που διέπει την ανάπτυξη του κάθε τμήματος. Έτσι, η σύνταξη του περιβάλλοντος εργασίας αποτελεί κατά βάση οργανωτική διαδικασία, απαιτεί μακροσκελή συγγραφή κώδικα και εξασφαλίζει την ευκολονόητη σειρά ενεργειών του χρήστη. Εμπεριέχει τόσο πρακτικές όσο και αισθητικές λεπτομέρειες που σκοπό έχουν να διευκολύνουν και όχι να καθυστερήσουν ή επιβαρύνουν νοητικά τον χρήστη, καθώς ο μοναδικός στόχος του τελευταίου είναι η ορθή μόρφωση του μοντέλου του φορέα, ώστε να καταλήξει σε ορθά αποτελέσματα. Αντίθετα, η εκτέλεση των υπολογισμών αποτελεί διαδικασία ως επί το πλείστον μαθηματικής βάσης, που εμπεριέχει πλήθος ελέγχων. Απαιτεί τόσο αναλυτική όσο και διαισθητική γνώση της στατικής, προκειμένου να εκτιμώνται τα αποτελέσματα και να εκτελείται ταχύτερα η αποσφαλμάτωση του κώδικα, καθώς και πολύ καλή γνώση γραμμικής άλγεβρας και αναλυτικής γεωμετρίας.

(3) Η χρήση τελείως διαφορετικών μεταξύ τους βιβλιοθηκών της γλώσσας προγραμματισμού για το κάθε τμήμα.

β. Ανάπτυξη του κάθε τμήματος ξεχωριστά, μέχρι το σημείο που αυτό ήταν δυνατόν και στη συνέχεια παράλληλη εργασία με κάποιο άλλο. Έτσι, ενώ η παρουσίαση του τρισδιάστατου φορέα αποτέλεσε μέχρι το πέρας απολύτως αυτοτελή διαδικασία, το γραφικό περιβάλλον από νωρίς σχετικά εξαρτήθηκε από τις ανάγκες της συλλογής δεδομένων, η οποία με τη σειρά της χρειάστηκε να είναι σύμφωνη με τους κανόνες που έθετε το τμήμα της ανάλυσης.

γ. Ξεχωριστό βήμα, που εκτελούνταν παράλληλα και ανά τακτά χρονικά διαστήματα, αποτέλεσε ο έλεγχος και η αποσφαλμάτωση του κώδικα, με σκοπό την εξασφάλιση αρμονικής λειτουργίας μεταξύ των επιμέρους τμημάτων.

Πέρα από τα παραπάνω, ιδιαίτερη βαρύτητα δόθηκε καθ'όλη τη διάρκεια της ανάπτυξης του λογισμικού στα εξής:

- Εύληπτη σειρά ενεργειών που θα ακολουθήσει ο χρήστης, καθώς και κατανοητό οπτικά περιβάλλον, ώστε να επικεντρωθεί στο μοντέλο του φορέα αυτό καθαυτό.
- Περιορισμός των σφαλμάτων ενεργειών του χρήστη, τόσο με την απαγόρευση εσφαλμένων, βάση λογικής, τιμών όσο και με την κατά το δυνατόν αυξημένη οπτικοποίηση των δεδομένων και χρήση κατάλληλων ειδοποιήσεων.
- Αυξημένη δυνατότητα επιλογών κατά τη διαδικασία μόρφωσης του μοντέλου του φορέα (π.χ. δυνατότητες διαγραφής δεδομένων ή και αλλαγής τους) για την ταχύτερη και ευχερέστερη πορεία προς τη φάση της ανάλυσης.
- Η μελλοντική δυνατότητα επέκτασης των δυνατοτήτων του προγράμματος για ανάλυση διαφορετικών τύπων φορέων.

## Κεφάλαιο 2: Θεωρητικό Υπόβαθρο

### 2.1 Στάδια Ανάλυσης Κατασκευών

Οι κυριότερες φάσεις της ανάλυσης των κατασκευών περιλαμβάνουν πρώτα από όλα την αρχική σχεδιαστική σύλληψη της προς υλοποίηση κατασκευής. Σε αυτό το στάδιο επιδρούν γενικά κριτήρια (αρχιτεκτονικά, οργανωτικά, θέματα νομοθεσίας και ασφάλειας κ.α.), ενώ τα κατασκευαστικά θέματα, όπως η διαστασιολόγηση δομικών στοιχείων, βασίζονται τόσο σε εμπειρικές όσο και σε προσεγγιστικές αριθμητικές εκτιμήσεις. Το δεύτερο στάδιο, αποτελεί έναν κύκλο εργασιών, ο οποίος περιλαμβάνει τη μόρφωση του μοντέλου του φορέα, τη στατική και δυναμική ανάλυση και την αξιολόγηση των αποτελεσμάτων με βάση το προδιαστασιολογημένο μοντέλο. Ο κύκλος επαναλαμβάνεται όσες φορές χρειαστεί, προκειμένου τα αποτελέσματα να ικανοποιούν τις απαιτήσεις που θέτει ο μελετητής. Τέλος, αφού ολοκληρωθεί επιτυχώς η αξιολόγηση, ακολουθεί ο λεπτομερής σχεδιασμός και τα τελικά σχέδια που θα χρησιμοποιηθούν στην κατασκευή.

### 2.2 Στατική Ανάλυση

Καθώς τα τελευταία χρόνια όλες οι υπολογιστικές διαδικασίες εκτελούνται με λογισμικά ακόμη και σε προσωπικούς ηλεκτρονικούς υπολογιστές, η πλειοψηφία των σφαλμάτων της ανάλυσης αφορούν τόσο ακατάλληλους στη σύλληψή τους φέροντες οργανισμούς, όσο και λανθασμένη προσομοίωση του φορέα. Η εξάλειψη των σφαλμάτων αυτών, προϋποθέτει συσσωρευμένη γνώση και εμπειρία καθώς και ορθή κρίση του μηχανικού, αφού παρά τους προσεκτικούς υπολογισμούς, η ακατάλληλη προσομοίωση φορέα, υπό ορισμένες συνθήκες μπορεί να παρουσιάσει εγγενείς αδυναμίες.

## 2.3 Προσομοίωση του Φέροντος Οργανισμού

Ο φέρων οργανισμός κάθε κατασκευής αποτελείται από δομικά στοιχεία, ποικίλων διαστάσεων και υλικών κατασκευής, κατάλληλα συνδεδεμένων μεταξύ τους. Επειδή η ανάλυση γενικά τρισδιάστατων σωμάτων παρουσιάζει μεγάλες δυσκολίες, δημιουργείται ένα προσομοίωμα του φορέα, κατάλληλα μορφοποιημένο ώστε να συγκλίνει προς την πραγματική συμπεριφορά της κατασκευής. Καθώς ο σκοπός είναι η απλοποίηση των προς κατασκευή στοιχείων, τα τελευταία ανάγονται στα παρακάτω:

α. Γραμμικά δομικά στοιχεία, που αποτελούν οι ράβδοι, οι δοκοί, τα τόξα, καλώδια και γενικά οτιδήποτε έχει διαστάσεις τέτοιες που να δικαιολογούν την αναγωγή του σε μονοδιάστατο στοιχείο. Στο μοντέλο, τα στοιχεία αυτά αντικαθίστανται από μία γραμμή, η οποία αποτελεί τον κεντροβαρικό τους άξονα.

β. Επιφανειακά δομικά στοιχεία, όπως είναι οι πλάκες και τα κελύφη, τα οποία χαρακτηρίζονται από δύο πολύ μεγαλύτερες διαστάσεις σε σύγκριση με την τρίτη. Τα στοιχεία αυτά απεικονίζονται με την μέση επιφάνειά τους.

γ. Χωρικά δομικά στοιχεία, όπως τα πέδιλα, οι θόλοι κτλ, εκείνα δηλαδή που είναι εξίσου ανεπτυγμένα και στις τρεις διαστάσεις, χωρίς να αφήνουν περιθώρια απλοποίησης.

## 2.4 Διαδικασία Ελαστικής Ανάλυσης

### 2.4.1 Γενικά

Την πλέον διαδεδομένη αριθμητική μέθοδο ανάλυσης αποτελεί η Μέθοδος των Πεπερασμένων Στοιχείων, της οποίας η πιο κοινή εφαρμογή είναι η γνωστή ως Μέθοδος Άμεσης Δυσκαμψίας. Έτσι, ο φέρων οργανισμός μοντελοποιείται ως ένα σύστημα απλούστερων εξιδανικευμένων στοιχείων (όπως περιγράφηκε στα προηγούμενα) διασυνδεδεμένων σε κόμβους. Οι άγνωστες μετακινήσεις και δυνάμεις του συστήματος υπολογίζονται από τις ιδιότητες ακαμψίας

των υλικών των στοιχείων, με τη χρήση μητρικών εξισώσεων. Πιο αναλυτικά, τα βήματα της μεθόδου είναι τα εξής:

α. Χρήση βασικών σχέσεων θεωρίας ελαστικότητας, που περιλαμβάνουν τις θεμελιώδεις εξισώσεις ισορροπίας, συμβιβαστού παραμορφώσεων και καταστατικές σχέσεις της τρισδιάστατης θεωρίας ελαστικότητας.

β. Επίλυση διαφορικών εξισώσεων των δομικών στοιχείων που αναφέρθηκαν παραπάνω, δηλαδή εξαγωγή των ακριβών ή προσεγγιστικών λύσεων των εξισώσεων που περιγράφουν τη συμπεριφορά τους.

γ. Μόρφωση συστήματος αλγεβρικών εξισώσεων, που αφορά στον κατάλληλο συνδυασμό των εξισώσεων του προηγούμενου βήματος με τη χρήση αυτών του πρώτου.

δ. Επίλυση του συστήματος αλγεβρικών εξισώσεων.

ε. Παρουσίαση αποτελεσμάτων σε μορφή κατάλληλη για τον σχεδιασμό.

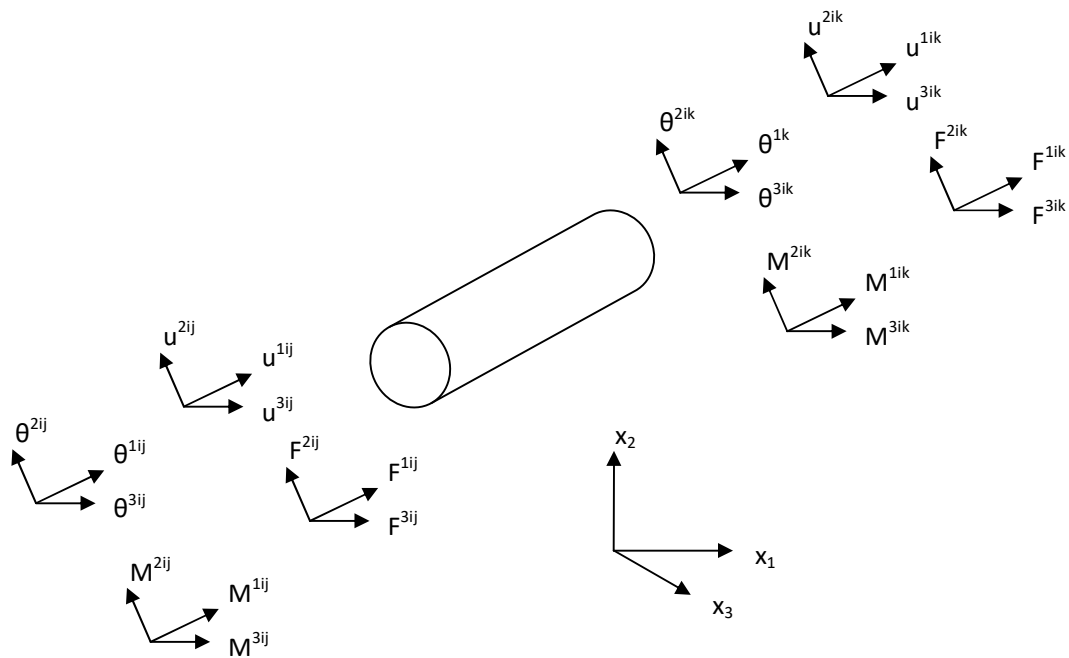
Η μέθοδος Άμεσης Δυσκαμψίας αποτελεί τη βάση για την πλειονότητα των εμπορικών και μη λογισμικών εκτέλεσης στατικής ανάλυσης κατασκευών.

#### 2.4.2. Αναλυτική περιγραφή της Μεθόδου Άμεσης Δυσκαμψίας

Η ανάλυση της μεθόδου θα γίνει για την περίπτωση του χωρικού πλαισίου, που άλλωστε αποτελεί και την γενική περίπτωση, με τους υπόλοιπους τύπους φορέων να προκύπτουν απλώς από τη διαγραφή των κατάλληλων γραμμών και στηλών από τα μητρώα.

Πρώτη ενέργεια κατά την εφαρμογή της μεθόδου είναι η αρίθμηση των κόμβων και των μελών του πλαισίου. Επίσης, επιλέγεται το καθολικό σύστημα αναφοράς, το οποίο πρέπει να είναι δεξιόστροφο ορθογώνιο και είναι αυτό ως προς το οποίο θα γίνει ο υπολογισμός των αντιδράσεων των στηρίξεων και των κινηματικών μεγεθών των κόμβων. Στη συνέχεια, ορίζεται το τοπικό σύστημα αναφοράς για κάθε ένα από τα μέλη του πλαισίου. Αυτό θα είναι επίσης δεξιόστροφο

ορθογώνιο, με τον άξονα  $x$  να είναι ο διαμήκης άξονας του μέλους με διεύθυνση από τον μικρότερο κόμβο (σε αύξοντα αριθμό) προς τον μεγαλύτερο. Οι υπόλοιποι άξονες καθορίζονται ανάλογα με την επιθυμητή διάταξη της διατομής. Τέλος, ορίζονται οι βαθμοί ελευθερίας κίνησης των κόμβων (που στην συγκεκριμένη περίπτωση είναι 6, 3 μετατοπίσεις και 3 στροφές, κατά τους αντίστοιχους άξονες). Προς το παρόν δεν λαμβάνονται υπόψη τυχόν στηρίξεις των μελών, ενώ οι βαθμοί ελευθερίας καθορίζονται στο καθολικό σύστημα αναφοράς (καθώς τα τοπικά συστήματα δεν συμπίπτουν).



Στο παραπάνω σχήμα φαίνονται τα διανύσματα των ακραίων δράσεων και μετακινήσεων όπως ισχύουν για κάθε μέλος του χωρικού πλαισίου, στο τοπικό σύστημα αναφοράς του. Επίσης, φαίνονται οι άξονες του καθολικού συστήματος, όπως επιλέχθηκε. Με  $u$  συμβολίζονται οι μετατοπίσεις, με  $\theta$  οι στροφές και με  $F$  και  $M$  οι αντίστοιχες δράσεις. Υπό διανυσματική μορφή, τα παραπάνω μπορούν να γραφτούν:

$$\{A^{ij}\} = \{F^{1ij}, F^{2ij}, F^{3ij}, M^{1ij}, M^{2ij}, M^{3ij}\}$$

$$\{D^{ij}\} = \{u^{1ij}, u^{2ij}, u^{3ij}, \theta^{1ij}, \theta^{2ij}, \theta^{3ij}\}$$

$$\{A^{ik}\} = \{F^{1ik}, F^{2ik}, F^{3ik}, M^{1ik}, M^{2ik}, M^{3ik}\}$$

$$\{D^{ik}\} = \{u^{1ik}, u^{2ik}, u^{3ik}, \theta^{1ik}, \theta^{2ik}, \theta^{3ik}\}$$

ή συνδυάζοντάς τα:

$$\{A^i\} = \{F^{1ij}, F^{2ij}, F^{3ij}, M^{1ij}, M^{2ij}, M^{3ij}, F^{1ik}, F^{2ik}, F^{3ik}, M^{1ik}, M^{2ik}, M^{3ik}\}$$

$$\{D^i\} = \{u^{1ij}, u^{2ij}, u^{3ij}, \theta^{1ij}, \theta^{2ij}, \theta^{3ij}, u^{1ik}, u^{2ik}, u^{3ik}, \theta^{1ik}, \theta^{2ik}, \theta^{3ik}\}$$

Τα παραπάνω ισχύουν τόσο για τα διανύσματα στο τοπικό σύστημα αναφοράς όσο και για εκείνα στο καθολικό ( $\underline{A}^i, \underline{D}^i$ ). Τα προαναφερθέντα διανύσματα συνδέονται μεταξύ τους με τη σχέση:

$$\{A^i\} = [\Lambda^i] \{\underline{A}^i\} \text{ ή αντίστροφα } \{\underline{A}^i\} = [\Lambda^i]^T \{A^i\}$$

$$\{D^i\} = [\Lambda^i] \{\underline{D}^i\} \text{ ή αντίστροφα } \{\underline{D}^i\} = [\Lambda^i]^T \{D^i\}$$

όπου το  $[\Lambda^i]$  είναι το μητρώο μετασχηματισμού του μέλους  $i$ .

Επίσης, τα τοπικά διανύσματα των ακραίων δράσεων και μετακινήσεων δεν είναι ανεξάρτητα μεταξύ τους, αλλά θεωρώντας ότι οι μετακινήσεις είναι μικρές, συνδέονται με την τοπική σχέση στιβαρότητας:

$\{A^i\} = [k^i] \{D^i\}$ , όπου  $k^i$  είναι το τοπικό μητρώο στιβαρότητας του μέλους.

Δεδομένου ότι το μέλος του χωρικού πλαισίου έχει δώδεκα βαθμούς ελευθερίας, το τοπικό μητρώο στιβαρότητας θα έχει την μορφή:

$$\begin{bmatrix} F_1^{\bar{u}} \\ F_2^{\bar{u}} \\ F_3^{\bar{u}} \\ M_1^{\bar{u}} \\ M_2^{\bar{u}} \\ M_3^{\bar{u}} \\ F_1^{ik} \\ F_2^{ik} \\ F_3^{ik} \\ M_1^{ik} \\ M_2^{ik} \\ M_3^{ik} \end{bmatrix} = \begin{bmatrix} k_{11}^i & 0 & 0 & 0 & 0 & 0 & k_{17}^i & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{22}^i & 0 & 0 & 0 & k_{26}^i & 0 & k_{28}^i & 0 & 0 & 0 & k_{2,12}^i \\ 0 & 0 & k_{33}^i & 0 & k_{35}^i & 0 & 0 & 0 & k_{39}^i & 0 & k_{3,11}^i & 0 \\ 0 & 0 & 0 & k_{44}^i & 0 & 0 & 0 & 0 & 0 & k_{4,10}^i & 0 & 0 \\ 0 & 0 & k_{53}^i & 0 & k_{55}^i & 0 & 0 & 0 & k_{59}^i & 0 & k_{5,11}^i & 0 \\ 0 & k_{62}^i & 0 & 0 & 0 & k_{66}^i & 0 & k_{68}^i & 0 & 0 & 0 & k_{6,12}^i \\ \hline k_{71}^i & 0 & 0 & 0 & 0 & 0 & k_{77}^i & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{82}^i & 0 & 0 & 0 & k_{86}^i & 0 & k_{88}^i & 0 & 0 & 0 & k_{8,12}^i \\ 0 & 0 & k_{93}^i & 0 & k_{95}^i & 0 & 0 & 0 & k_{99}^i & 0 & k_{9,11}^i & 0 \\ 0 & 0 & 0 & k_{10,4}^i & 0 & 0 & 0 & 0 & 0 & k_{10,10}^i & 0 & 0 \\ 0 & 0 & k_{11,3}^i & 0 & k_{11,5}^i & 0 & 0 & 0 & k_{11,9}^i & 0 & k_{11,11}^i & 0 \\ 0 & k_{12,2}^i & 0 & 0 & 0 & k_{12,6}^i & 0 & k_{12,8}^i & 0 & 0 & 0 & k_{12,12}^i \end{bmatrix} \begin{bmatrix} u_1^{\bar{u}} \\ u_2^{\bar{u}} \\ u_3^{\bar{u}} \\ \mathcal{G}_1^{\bar{u}} \\ \mathcal{G}_2^{\bar{u}} \\ \mathcal{G}_3^{\bar{u}} \\ u_1^{ik} \\ u_2^{ik} \\ u_3^{ik} \\ \mathcal{G}_1^{ik} \\ \mathcal{G}_2^{ik} \\ \mathcal{G}_3^{ik} \end{bmatrix}$$

Οι δείκτες στιβαρότητας  $k^i$  του παραπάνω μητρώου υπολογίζονται με βάση τις μοναδιαίες μετακινήσεις και τις δράσεις παγίωσης στο κάθε μέλος. Επίσης, για τον υπολογισμό των δεικτών απαιτούνται μόνο εκείνοι του τριγωνικού υπομητρώου, καθώς οι υπόλοιποι προκύπτουν από την εφαρμογή των συνθηκών συμμετρίας και ισορροπίας.

Έτσι, το τοπικό μητρώο στιβαρότητας ενός μέλους θα είναι:

$$[k^i] = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_3}{L^3} & 0 & 0 & 0 & \frac{6EI_3}{L^2} & 0 & -\frac{12EI_3}{L^3} & 0 & 0 & 0 & \frac{6EI_3}{L^2} \\ 0 & 0 & \frac{12EI_2}{L^3} & 0 & -\frac{6EI_2}{L^2} & 0 & 0 & 0 & -\frac{12EI_2}{L^3} & 0 & -\frac{6EI_2}{L^2} & 0 \\ 0 & 0 & 0 & \frac{GI_1}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GI_1}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_2}{L^2} & 0 & \frac{4EI_2}{L} & 0 & 0 & 0 & \frac{6EI_2}{L^2} & 0 & \frac{2EI_2}{L} & 0 \\ 0 & \frac{6EI_3}{L^2} & 0 & 0 & 0 & \frac{4EI_3}{L} & 0 & -\frac{6EI_3}{L^2} & 0 & 0 & 0 & \frac{2EI_3}{L} \\ \hline -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_3}{L^3} & 0 & 0 & 0 & -\frac{6EI_3}{L^2} & 0 & \frac{12EI_3}{L^3} & 0 & 0 & 0 & -\frac{6EI_3}{L^2} \\ 0 & 0 & -\frac{12EI_2}{L^3} & 0 & \frac{6EI_2}{L^2} & 0 & 0 & 0 & \frac{12EI_2}{L^3} & 0 & \frac{6EI_2}{L^2} & 0 \\ 0 & 0 & 0 & -\frac{GI_1}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GI_1}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_2}{L^2} & 0 & \frac{2EI_2}{L} & 0 & 0 & 0 & \frac{6EI_2}{L^2} & 0 & \frac{4EI_2}{L} & 0 \\ 0 & \frac{6EI_3}{L^2} & 0 & 0 & 0 & \frac{2EI_3}{L} & 0 & -\frac{6EI_3}{L^2} & 0 & 0 & 0 & \frac{4EI_3}{L} \end{bmatrix}$$

όπου τα  $E, G$  είναι το μέτρο ελαστικότητας και διάτμησης του υλικού, αντίστοιχα, το  $L$  το μήκος του μέλους και τα  $A, I_1, I_2, I_3$  το εμβαδόν της διατομής, η στρεπτική και οι καμπτικές ροπές αδρανείας.

Για την εύρεση του καθολικού μητρώου στιβαρότητας, προκειμένου να υπάρχει κοινό σύστημα αναφοράς για τις δράσεις και μετακινήσεις όλων των μελών, θα χρησιμοποιηθεί η σχέση

$$\{A^i\} = [k^i]\{D^i\}$$

που συνδέει τις δράσεις με τις μετακινήσεις στο τοπικό σύστημα αναφοράς. Η σχέση μπορεί να γραφτεί με τη βοήθεια των μητρώων μετασχηματισμού από το ένα σύστημα στο άλλο

$$[\Lambda^i]\{A^i\} = [k^i][\Lambda^i]\{D^i\}$$

$$\{A^i\} = [\Lambda^i]^T [k^i] [\Lambda^i]\{D^i\}$$

Επειδή όμοια με όσα ισχύουν για το τοπικό σύστημα, ισχύουν και για το καθολικό, ισχύει η σχέση που συνδέει δράσεις και μετακινήσεις

$$\{A^i\} = [K^i]\{D^i\}$$

συνεπάγεται ότι

$$[K^i] = [\Lambda^i]^T [k^i] [\Lambda^i]$$

Συνεπώς το καθολικό μητρώο στιβαρότητας του κάθε μέλους προκύπτει ως γινόμενο του τοπικού μητρώου και του μητρώου μετασχηματισμού του, όπως παραπάνω.

Για τη σύνθεση του καθολικού μητρώου στιβαρότητας πλαισίου με  $N$  κόμβους, λαμβάνουμε υπόψη μας ότι αυτό θα είναι τετραγωνικό με διάσταση  $6 \cdot N$ , όσοι δηλαδή και οι βαθμοί ελευθερίας του. Έπειτα, αντιστοιχίζουμε τα στοιχεία του κάθε τοπικού μητρώου στις θέσεις του καθολικού, σύμφωνα με την αρίθμηση των βαθμών ελευθερίας κίνησης.

Στην συνέχεια, χρειάζεται να γραφτούν σε διανυσματική μορφή, στο καθολικό σύστημα, οι συνιστώσες των επικόμβιων δράσεων του πλαισίου και των μετακινήσεων των κόμβων του. Για κάθε μέλος λοιπόν θα είναι:

$$\{P^i\} = \{P_1^i, P_2^i, P_3^i, M_1^i, M_2^i, M_3^i\} \text{ και}$$

$$\{\Delta^i\} = \{\underline{\Delta}_1^i, \underline{\Delta}_2^i, \underline{\Delta}_3^i, \underline{\Theta}_1^i, \underline{\Theta}_2^i, \underline{\Theta}_3^i\}$$

ενώ για όλο το πλαίσιο (θεωρώντας ότι έχει N κόμβους):

$$\{P\} = \{P_1^1, P_2^1, P_3^1, M_1^1, M_2^1, M_3^1, P_1^2, P_2^2, P_3^2, M_1^2, M_2^2, M_3^2, \dots, P_1^N, P_2^N, P_3^N, M_1^N, M_2^N, M_3^N\}$$

και

$$\{\Delta\} = \{\underline{\Delta}_1^1, \underline{\Delta}_2^1, \underline{\Delta}_3^1, \underline{\Theta}_1^1, \underline{\Theta}_2^1, \underline{\Theta}_3^1, \underline{\Delta}_1^2, \underline{\Delta}_2^2, \underline{\Delta}_3^2, \underline{\Theta}_1^1, \underline{\Theta}_2^1, \underline{\Theta}_3^1, \dots, \underline{\Delta}_1^N, \underline{\Delta}_2^N, \underline{\Delta}_3^N, \underline{\Theta}_1^N, \underline{\Theta}_2^N, \underline{\Theta}_3^N\}$$

Ακολουθεί η αναδιάταξη όλων των παραπάνω μητρώων, έτσι ώστε να ληφθεί υπόψη η στήριξη του πλαισίου, δηλαδή να διαχωριστούν οι δεσμευμένοι βαθμοί ελευθερίας από τους ελεύθερους. Πιο συγκεκριμένα, επιζητούμε να προηγούνται οι ελεύθεροι βαθμοί ελευθερίας και να έπονται οι δεσμευμένοι και αυτό γίνεται σύμφωνα με τις σχέσεις

$$\{\underline{P}_m\} = [V]^T \{P\}$$

$$\{\underline{\Delta}_m\} = [V]^T \{\Delta\}$$

$$[\underline{K}_m] = [V] [K] [V]^T$$

όπου το μητρώο [V] είναι το μητρώο αναδιάταξης.

Τέλος, από την επίλυση της εξίσωσης

$$\{\underline{P}_m\} = [\underline{K}_m] \{\underline{\Delta}_m\}$$

προκύπτουν οι σχέσεις

$$\{\underline{\Delta}_f\} = [\underline{K}_{ff}]^{-1} (\{\underline{P}_f\} - [\underline{K}_{fs}]\{\underline{\Delta}_s\})$$

$$\{\underline{P}_s\} = [\underline{K}_{sf}]\{\underline{\Delta}_f\} + [\underline{K}_{ss}]\{\underline{\Delta}_s\}$$

όπου το μητρώο  $\{\underline{\Delta}_f\}$  αφορά τις άγνωστες μετακινήσεις των κόμβων, το  $\{\underline{P}_s\}$  τις άγνωστες επικόμβιες δράσεις (δηλαδή τις αντιδράσεις στις στηρίξεις), το  $\{\underline{P}_f\}$  τις γνωστές επικόμβιες δράσεις και το  $\{\underline{\Delta}_s\}$  τις γνωστές μετακινήσεις των κόμβων (στις στηρίξεις). Τα υπόλοιπα μητρώα είναι αποτέλεσμα του επιμερισμού του καθολικού μητρώου στιβαρότητας του πλαισίου (αφού υπέστη αναδιάταξη):

$$\begin{bmatrix} \left[ \begin{array}{c|c} \overline{K}_{ff} & \overline{K}_{fs} \\ \hline \overline{K}_{sf} & \overline{K}_{ss} \end{array} \right] \end{bmatrix}$$

## 2.5 Γλώσσα προγραμματισμού Python

### 2.5.1. Ορισμός

Η Python είναι μια υψηλού επιπέδου γλώσσα προγραμματισμού, που δημιουργήθηκε το 1991 από τον Ολλανδό Guido van Rossum. Το κυριότερο χαρακτηριστικό της φιλοσοφίας με βάση την οποία σχεδιάστηκε είναι η αναγνωσιμότητα του κώδικα και η ευκολία χρήσης και εκμάθησης. Η ταχύτητα γραφής και το συντακτικό της, δίνουν τη δυνατότητα στον προγραμματιστή να εκφράσει έννοιες σε λιγότερες γραμμές κώδικα από ότι σε γλώσσες όπως η C++ ή η Java, καθώς και να υλοποιήσει αποτελεσματικότερα συντήρηση λογισμικού και αποσφαλμάτωση.

Η Python αναπτύσσεται ως ανοιχτό λογισμικό (open source) και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Ο κώδικας διανέμεται με την άδεια Python Software Foundation License η οποία είναι συμβατή με την GPL.

### 2.5.2. Ιστορικά στοιχεία

Αρχικά η Python ήταν γλώσσα σεναρίων (scripting language) που χρησιμοποιούνταν στο λειτουργικό σύστημα Amoeba, ικανή και για κλήσεις συστήματος.

Η έκδοση 2.0 κυκλοφόρησε στις 16 Οκτωβρίου του 2000. Στις 3 Δεκεμβρίου 2008 κυκλοφόρησε η έκδοση 3.0 (γνωστή και ως py3k ή python 3000). Πολλά από τα καινούργια χαρακτηριστικά αυτής της έκδοσης έχουν μεταφερθεί στις εκδόσεις 2.6 και 2.7 που είναι προς τα πίσω συμβατές.

Η Python 3.0 είναι ιστορικά η πρώτη γλώσσα προγραμματισμού που σπάει την προς τα πίσω συμβατότητα με προηγούμενες εκδόσεις ώστε να διορθωθούν κάποια λάθη που υπήρχαν σε προγενέστερες εκδόσεις και να καταστεί ακόμα πιο σαφής ο απλός τρόπος με τον οποίο μπορούν να γίνουν κάποια πράγματα.

### 2.5.3. Χαρακτηριστικά της Python

Η Python έχει μια αρκετά μεγάλη κύρια βιβλιοθήκη (standard library) καθώς και πολύ μεγάλο αριθμό δευτερευόντων βιβλιοθηκών που διευκολύνει τη χρήση της και την ανάπτυξη ποικίλων εφαρμογών.

Η αναγνωσιμότητά της σε σχέση με άλλες γλώσσες χαμηλότερου επιπέδου καθώς και το γεγονός ότι τα προγράμματα γραμμένα σε Python μπορεί να είναι έως και 3-5 φορές μικρότερα, επιτρέπει την ταχύτερη ανάπτυξη εφαρμογών και την αποτελεσματικότερη αποσφαλμάτωση. Βέβαια, το γεγονός ότι είναι γλώσσα υψηλότερου επιπέδου από άλλες, έχει κάποιο αντίκτυπο στην ταχύτητα των διεργασιών (οι εφαρμογές σε Python είναι πιο αργές). Το τελευταίο, όμως, είναι ένα μειονέκτημα που τα τελευταία χρόνια αρχίζει να παραβλέπεται, καθώς παρουσιάζεται σταδιακή στροφή σε γλώσσες που ευνοούν την απόδοση του προγραμματιστή σε βάρος της απόδοσης των εφαρμογών.

Η Python είναι πλέον μια ώριμη γλώσσα προγραμματισμού, με ευρέως διαδεδομένη χρήση από μεγάλα εκπαιδευτικά ιδρύματα (MIT, Stanford κ.α.), εταιρίες

όπως οι Google και Yahoo, τη NASA, όλες σχεδόν τις διανομές Linux και πολλούς άλλους.

## 2.6. Εργαλεία – Βιβλιοθήκες

### 2.6.1. Γενικά

Όπως προαναφέρθηκε, η κύρια βιβλιοθήκη της Python είναι αρκετά μεγάλη ώστε να καλύπτει ευρύ φάσμα απαιτήσεων, όμως επιλέγεται συνήθως η χρήση και άλλων βιβλιοθηκών, πιο εξειδικευμένων, οι οποίες συνεισφέρουν τόσο στην ταχύτητα των διεργασιών, όσο και στην αναγνωσιμότητα του κώδικα και τον περιορισμό του μεγέθους του προγράμματος. Παρακάτω παρουσιάζονται οι βιβλιοθήκες που χρησιμοποιήθηκαν στην παρούσα διπλωματική, καθώς και κάποιες που αξίζει να αναφερθούν.

### 2.6.2. NumPy

Το NumPy είναι το θεμελιώδες πακέτο της Python που αφορά επιστημονικές εφαρμογές και υπολογισμούς. Περιέχει μεταξύ άλλων:

- Ισχυρά εργαλεία  $n$ -διάστατων διατάξεων που ουσιαστικά αποτελούν πίνακες (matrices).
- Εργαλεία για την ενσωμάτωση κώδικα C / C ++ και Fortran.
- Εργαλεία γραμμικής άλγεβρας, μετασχηματισμό Fourier και δυνατότητες τυχαίων αριθμών.

Εκτός από τις προφανείς επιστημονικές του χρήσεις, το NumPy μπορεί επίσης να χρησιμοποιηθεί ως αποτελεσματική πολυδιάστατη δεξαμενή γενικών δεδομένων. Μπορούν να οριστούν αυθαίρετοι τύποι δεδομένων. Αυτό επιτρέπει στο NumPy να ενσωματώνεται απρόσκοπτα και γρήγορα με μια μεγάλη ποικιλία βάσεων δεδομένων.

Στην παρούσα διπλωματική έχει γίνει εκτεταμένη χρήση της συγκεκριμένης βιβλιοθήκης. Το σημαντικότερο πλεονέκτημά της, είναι η ταχύτητα που επιτυγχάνει σε ότι αφορά πράξεις με πίνακες, κάτι το οποίο σε εφαρμογές μικρής κλίμακας δεν μπορεί να γίνει αντιληπτό από τον χρήστη, αλλά όσο αυξάνεται το

μέγεθος και το πλήθος των πινάκων μεταξύ των οποίων εκτελούμε πράξεις, ο χρόνος αρχίζει να παίζει σημαντικό ρόλο.

### 2.6.3. Pandas

Είναι από τις πιο χρήσιμες βιβλιοθήκες της Python. Παρέχει γρήγορες, ευέλικτες και εκφραστικές δομές δεδομένων σχεδιασμένες να κάνουν την εργασία με "σχεσιακά" ή "επισημασμένα" δεδομένα τόσο εύκολη όσο και διαισθητική. Στόχος του είναι να αποτελέσει τη θεμελιώδη δομή υψηλού επιπέδου για την πραγματοποίηση πρακτικών αναλύσεων δεδομένων σε γλώσσα Python. Επιπλέον, έχει τον ευρύτερο στόχο να γίνει το πιο ισχυρό και ευέλικτο εργαλείο στατιστικής ανάλυσης και επεξεργασίας δεδομένων ανοικτού κώδικα διαθέσιμο σε οποιαδήποτε γλώσσα.

Δεν θα αναφερθούν περαιτέρω λεπτομέρειες σχετικά με τις δυνατότητες του pandas σε ότι αφορά την επεξεργασία και ανάλυση δεδομένων, που είναι και η κύρια χρήση του, καθώς στην παρούσα διπλωματική χρησιμοποιήθηκε κυρίως για την αποθήκευση και ανάκτηση δεδομένων από τον χρήστη. Ο λόγος που δεν έγινε χρήση της κύριας βιβλιοθήκης για τις συγκεκριμένες διεργασίες είναι ότι το pandas προσφέρει εργαλεία ευκολότερης και γρηγορότερης διαχείρισης μεγάλου όγκου αλληλοσυσχετιζόμενων δεδομένων.

### 2.6.4. Sys

Αποτελεί εργαλείο πρόσβασης σε μεταβλητές, λειτουργίες και μεθόδους, απαραίτητες για τον διερμηνέα της Python.

### 2.6.5 PyQt5

Το PyQt είναι ίσως η πιο δημοφιλής διασύνδεση της Python με το Qt (προφέρεται cute). Το Qt ένα ολοκληρωμένο πλαίσιο ανάπτυξης εφαρμογών σε C++, ανεξάρτητο πλατφόρμας, που χρησιμοποιείται ευρέως για την ανάπτυξη εφαρμογών γραφικού περιβάλλοντος (GUI) καθώς επίσης και για την ανάπτυξη προγραμμάτων χωρίς το τελευταίο, όπως εργαλεία κονσόλας και servers. Το Qt αποτελεί επαγγελματικό εργαλείο, με πάνω από 620 κατηγορίες αντικειμένων (classes), που καλύπτουν αντικείμενα γραφικού περιβάλλοντος, χειρισμό XML, επικοινωνία μέσω

διαδικτύου, βάσεις δεδομένων SQL, περιήγηση στο Web και άλλες τεχνολογίες. Το PyQt υποστηρίζει πλήρως την αντίστοιχη έκδοση του Qt.

Οι διαθέσιμες εκδόσεις είναι το PyQt4 και το PyQt5 (σε πλήρη αντιστοιχία με τα Qt4 και Qt5). Το PyQt4 εκτελείται σε Windows, Linux, Mac OS X και σε διάφορες πλατφόρμες UNIX. Το PyQt5 τρέχει επίσης σε Android και iOS. Η τελευταία έκδοση κατά την έναρξη δημιουργίας του παρόντος ήταν η v.5.10.1, η οποία και χρησιμοποιήθηκε.

Τέλος, επισημαίνεται εδώ και αναλύεται παρακάτω, ότι οι οδηγίες χρήσης και επεξήγησης (documentation) του PyQt είναι περιορισμένες, σε αντίθεση με το Qt, που όμως χρησιμοποιεί την C++. Επομένως ο προγραμματιστής θα πρέπει να αποκτήσει τουλάχιστον τις πολύ βασικές γνώσεις σύνταξης της συγκεκριμένης γλώσσας.

Εναλλακτικές βιβλιοθήκες δημιουργίας GUI είναι το Tkinter, το Kivy, το Pyforms, το PyGUI και το wxPython. Για διάφορους λόγους, καθώς και με βάση κριτικές και απόψεις εμπειρών χρηστών στο διαδίκτυο, επιλέχθηκε η χρήση του PyQt. Η επιλογή έγινε κυρίως με γνώμονα τις απαιτήσεις λειτουργιών, που αρχικά κρίθηκε ότι μπορεί να είναι αυξημένες, και μια επιφανειακή εξέταση όλων των βιβλιοθηκών ως προς τη χρήση τους.

#### 2.6.6 CxFreeze

Το CxFreeze είναι εργαλείο εξαγωγής εκτελέσιμων αρχείων, τα οποία αποτελούν αυτοτελείς εφαρμογές και δεν απαιτούν την παρουσία της Python στον υπολογιστή του χρήστη. Είναι ανεξάρτητο πλατφόρμας, ενώ το αρχείο – πρόγραμμα που εξάγεται είναι συμβατό με το λειτουργικό μέσα στο οποίο δημιουργήθηκε, δηλαδή αν ο χρήστης εξάγει το πρόγραμμα σε Windows, αυτό θα λειτουργεί μόνο σε Windows, αντίστοιχα και για τα υπόλοιπα.

Παρότι δοκιμάστηκε η χρήση και άλλων ανάλογων εργαλείων εξαγωγής εκτελέσιμων αρχείων (όπως το py2exe), κρίθηκε ευχερέστερη η χρήση του CxFreeze. Βέβαια, χρειάζεται σε κάθε περίπτωση η παρέμβαση του χρήστη για να προσθέσει αρχεία στον εξαγόμενο φάκελο, ώστε να είναι το πρόγραμμα λειτουργικό, κάτι που

μάλλον οφείλεται στην ίδια τη βιβλιοθήκη. Στο 5<sup>ο</sup> κεφάλαιο εξηγούνται αναλυτικότερα όλες οι λεπτομέρειες της εξαγωγής ώστε να μπορεί μελλοντικά να επαναληφθεί η διαδικασία, αν χρειαστεί.

#### 2.6.7 Άλλες Βιβλιοθήκες

Μια βιβλιοθήκη που τελικά δεν χρησιμοποιήθηκε αλλά αξίζει αναφοράς λόγω της χρησιμότητάς της είναι το SymPy, που επιτρέπει χρήση συμβολικών μαθηματικών. Παρέχει δυνατότητες εύρεσης ορισμένων και αόριστων ολοκληρωμάτων, παραγωγίσεις, παραγοντοποιήσεις, εύρεση ορίων κ.α. με χρήση συμβόλων και μπορεί να δώσει αποτελέσματα ακόμη και με τη μορφή παραστάσεων. Χρησιμοποιήθηκε στο πρόγραμμα στα πρώτα στάδια της ανάπτυξης του τμήματος των υπολογισμών, με εφαρμογή στην εύρεση των ακραίων δράσεων των φορέων, όμως η αδυναμία ενσωμάτωσής της στο εξαγόμενο πρόγραμμα με τη χρήση του CxFreeze, οδήγησε στην εγκατάλειψή του.

## Κεφάλαιο 3: Ανάλυση Προγράμματος

### 3.1 Γενικά

#### 3.1.1 Εκτέλεση του προγράμματος σε περιβάλλον Windows 10

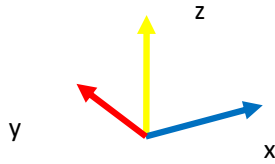
Το πρόγραμμα, στην εκτελέσιμη σε περιβάλλον Windows μορφή του, αποτελείται από έναν φάκελο στον οποίο εμπεριέχονται όλα τα απαραίτητα αρχεία για τη λειτουργία του, συμπεριλαμβανομένων των αρχείων εικόνων, δεδομένων διατομών κτλ. Επίσης, στον ίδιο φάκελο αποθηκεύονται και όλα τα αρχεία που δημιουργούνται από τον χρήστη, κατά προεπιλογή. Ο φάκελος μπορεί να τοποθετηθεί σε οποιοδήποτε σημείο του υπολογιστή, με μοναδικό περιορισμό να μην αλλάζουν θέση τα αρχεία των αποτελεσμάτων των αναλύσεων, ώστε να μπορούν να εντοπιστούν από το πρόγραμμα. Το τελευταίο δεν επηρεάζει το πρόγραμμα στην λειτουργία του, απλώς ο χρήστης δεν θα μπορεί να εντοπίσει παλιότερες αναλύσεις που πιθανώς να τον ενδιαφέρουν.

#### 3.1.2 Εκτέλεση του προγράμματος σε περιβάλλον Python

Στην εκτελέσιμη σε περιβάλλον Python μορφή του, το πρόγραμμα αποτελείται καταρχήν από τέσσερα αρχεία γραμμένα σε γλώσσα Python, από τα αρχεία με τα δεδομένα των διατομών, εικόνες και άλλα, τα οποία θα αναφερθούν αναλυτικότερα παρακάτω. Επιπλέον, συμπεριλαμβάνεται και ένα αρχείο Python με την ονομασία `setup.py`, που χρησιμοποιείται για την εξαγωγή του προγράμματος. Ο χρήστης θα πρέπει να έχει εγκατεστημένη την τελευταία έκδοση της Python, το `numpy 1.12.1`, το `pandas 0.20.1`, καθώς και το `PyQt5`. Νεότερες ή παλαιότερες εκδόσεις των παραπάνω δεν είναι σίγουρο ότι θα λειτουργήσουν κανονικά. Επίσης, θα πρέπει όλα τα αρχεία να βρίσκονται στον ίδιο φάκελο, ανεξάρτητα από τη θέση που έχει τοποθετηθεί ο τελευταίος.

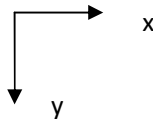
#### 3.1.3 Συστήματα συντεταγμένων

Το καθολικό σύστημα συντεταγμένων, όπως φαίνεται αρχικά στην οθόνη απεικόνισης και πριν οποιαδήποτε στροφή του φορέα, είναι το παρακάτω:



Με την ίδια διάταξη χρησιμοποιούνται οι άξονες και στο τοπικό σύστημα του κάθε μέλους, με τον άξονα x να είναι ο διαμήκης άξονάς του.

Για την καλύτερη κατανόηση του τμήματος της γραφικής αναπαράστασης της κατασκευής αναφέρεται ότι στο περιβάλλον εργασίας δεν υποστηρίζεται τρισδιάστατο σύστημα συντεταγμένων, με αποτέλεσμα να απαιτείται πρότερος μετασχηματισμός των εισαγόμενων συντεταγμένων στο υπάρχον σύστημα, το οποίο είναι:



Επιπλέον πληροφορίες σχετικά με τα συστήματα συντεταγμένων που χρησιμοποιεί το PyQt (είναι περισσότερα από ένα ανάλογα με το επίπεδο ή το αντικείμενο εργασίας) μπορούν να βρεθούν στο διαδίκτυο στην ιστοσελίδα του Qt. Οι πληροφορίες αυτές απαιτούνται αφενός για την κατανόηση του περιβάλλοντος εργασίας και αφετέρου στην περίπτωση που επιχειρηθεί αλλαγή ή επέκταση των λειτουργιών του παρόντος.

### 3.1.4 Μονάδες μέτρησης

Πλην των μεγεθών που εκφράζουν πίεση και δύναμη, όπου χρησιμοποιούνται ΚΡα και ΚΝ, αντίστοιχα, οι υπόλοιπες αποτελούν θεμελιώδη ή συμπληρωματικά μεγέθη του Διεθνούς Συστήματος Μονάδων (SI).

## 3.2 Υλοποίηση Προγράμματος

### 3.2.1 Τα αρχεία του προγράμματος

Το πρόγραμμα αποτελείται από τα παρακάτω αρχεία:

- GUI.py
- analysis3Dframe.py
- frame3d.py
- ireact3d.py
- setup.py
- 9 αρχεία τύπου csv με τα στοιχεία προτύπων διατομών (έχουν προβλεφθεί HE, IPE, IPN, HL, HP, HD, U, UPE, UPN)
- αρχεία εικόνων τύπου png

Το πιο εκτενές είναι το GUI.py, που καταλαμβάνει περίπου 4.500 γραμμές κώδικα (σε σύνολο περίπου 6000 περίπου γραμμών). Ακλουθεί η ανάλυση του κάθε αρχείου ξεχωριστά, με το GUI.py να περιγράφεται τελευταίο ώστε να γίνει πιο κατανοητή στον αναγνώστη η συσχέτιση της λειτουργίας τους.

### 3.2.2 Το αρχείο analysis3Dframe.py

Αποτελεί το τμήμα του προγράμματος, το οποίο αξιοποιεί τα εισαχθέντα από τον χρήστη δεδομένα και εκτελεί τη διαδικασία της ανάλυσης. Εδώ λαμβάνει χώρα το σύνολο σχεδόν της υπολογιστικής διαδικασίας, με τα υπόλοιπα αρχεία να είναι απαλλαγμένα από υπολογισμούς, με εξαίρεση το frame3d.py, που περιλαμβάνει κάποιους από αυτούς απολύτως απαραίτητους για τη δική του λειτουργία και κυρίως δευτερεύουσας σημασίας.

Αποτελείται από ένα σύνολο λειτουργιών που δεν ανήκουν σε κλάση. Η μοναδική λειτουργία που καλείται από εξωτερικό παράγοντα είναι το ‘analysis’ το οποίο ξεκινάει και τη διαδικασία της ανάλυσης. Εν συντομία η λειτουργία του έχει ως εξής:

- Υπολογισμός μετατοπίσεων κόμβων και αντιδράσεων με παραμέτρους συντελεστές για τον συνδυασμό φόρτισης για μόνιμα και κινητά φορτία και τη δυνατότητα συμπερίληψης των ιδίων βαρών των μελών.
- Υπολογισμός ακραίων δράσεων και αποθήκευσή τους σε ένα λεξικό.
- Επιλογή των φορτίων ανάλογα με το αν επιθυμείται ο συνυπολογισμός των ίδιων βαρών των μελών.
- Υπολογισμός των διαγραμμάτων των εσωτερικών δυνάμεων, αποθήκευσή τους σε ένα λεξικό και επιστροφή του στο γραφικό περιβάλλον από όπου και ζητήθηκε η ανάλυση.

Αναλυτικότερα, οι λειτουργίες του αρχείου είναι:

analysis	Εκτελεί συγκεντρωτικά την διαδικασία της ανάλυσης μέσω των υπολοίπων λειτουργιών. Αρχικά υπολογίζει μέσω της ‘dispsReacts’ τις αντιδράσεις και τις μετατοπίσεις των κόμβων στους βαθμούς ελευθερίας που έχουν περιορισμό ή όχι, αντίστοιχα. Στη συνέχεια, μέσω της ‘calculateEdgeForces’ υπολογίζει τις ακραίες δράσεις των μελών και, τέλος, υπολογίζει τις εσωτερικές δυνάμεις και τις αποδίδει σε ένα λεξικό.
dispsReacts	Υπολογίζει τις αντιδράσεις και τις μετατοπίσεις των κόμβων στους βαθμούς ελευθερίας που έχουν ή όχι περιορισμό, αντίστοιχα, και τις αποδίδει για χρήση στο ‘analysis’.
thereIsElasticNode	Ελέγχει αν υπάρχει κάποια ελατηριακή στήριξη και αν ναι, εκτελείται η ‘calculateElasticNode’.
returnDisplacements	Το μητρώο μετακινήσεων που υπολογίζεται από το

	πρόγραμμα περιέχει μόνο τους βαθμούς ελευθερίας που είναι ελεύθεροι να εκτελέσουν κίνηση. Η συγκεκριμένη λειτουργία αναδιατάσσει το μητρώο και προσθέτει τις μηδενικές τιμές που αντιστοιχούν στους δεσμευμένους βαθμούς ελευθερίας.
returnReactions	Επιστρέφει ένα λεξικό όπου έχουν αντιστοιχηθεί οι τιμές των αντιδράσεων με τον ακριβή βαθμό ελευθερίας στον οποίο ανήκουν.
calculateElasticNode	Εκτελείται στο τέλος του 'dispsReacts' και επιστρέφει τις τιμές των αντιδράσεων και μετακινήσεων αν υπάρχει κάποια ελατηριακή στήριξη
calculateEdgeForces	Επιστρέφει ένα λεξικό με κλειδιά τα ονόματα των μελών και τιμές τις ακραίες δράσεις τους.
K_global	Δίνει το καθολικό μητρώο δυσκαμψίας του φορέα.
forceMatrix	Δίνει τα μητρώο δυνάμεων
supportRotationMatrix	Δίνει το μητρώο μετασχηματισμού για κεκλιμένη στήριξη.
displacementMatrix	δημιουργεί το μητρώο μετακινήσεων των κόμβων (αν υπάρχουν)
transformMatrix	Δημιουργεί το μητρώο μετασχηματισμού του φορέα
nodeContinuation	Ελέγχει την συνέχεια των κόμβων, καθώς υπάρχει περίπτωση κατά τη διάρκεια μόρφωσης του φορέα να έχει σβηστεί κάποιος κόμβος.
arrangedNodes	Σύνολο λειτουργιών που έχουν σκοπό την αναδιάταξη των ονομάτων των κόμβων και την απόδοση νέων ονομάτων με τη σειρά που περάστηκαν αυτοί. (αν δηλαδή υπάρχουν μόνο οι κόμβοι n1, n2, n4, n5... επειδή διαγράφηκε κάποια στιγμή ο n3, θα αποκατασταθεί η συνέχειά τους)
renameNode	
renameAllNodes	

Με την εντολή για ανάλυση, που θα δοθεί από το αντίστοιχο παράθυρο του GUI, εκτελείται το 'analysis', που με τη σειρά του εκτελεί το 'dispsReacts', όπου αρχικά γίνεται αποκατάσταση της σειράς των κόμβων ('renameAllNodes' - στην περίπτωση που κάποιος έχει διαγραφεί κατά τη διάρκεια

της μόρφωσης του μοντέλου του φορέα). Στη συνέχεια, καλείται το 'initValues' του αρχείου frame3D και υπολογίζονται τα μοναδιαία διανύσματα των τοπικών αξόνων, τα μητρώα περιστροφής και τα μητρώα δυσκαμψίας (τοπικό και καθολικό), για κάθε μέλος ξεχωριστά. Επίσης, λαμβάνεται υπόψη τυχόν ύπαρξη στερεού κόμβου και ελευθέρωσης άκρων και μετασχηματίζονται αντίστοιχα τα μητρώα δυσκαμψίας ('getRigidK' και 'releaseEdge'). Έπειτα, κατασκευάζεται το καθολικό μητρώο δυσκαμψίας του πλαισίου ('K\_global'), το μητρώο μετασχηματισμού ('transformMatrix') προκειμένου να προηγούνται τα στοιχεία των μητρώων που αφορούν τους ελεύθερους βαθμούς ελευθερίας, τα μητρώα των επικόμβιων δράσεων ('forceMatrix' - υπολογίζονται ξεχωριστά οι μόνιμες από τις κινητές και στη συνέχεια προστίθενται πολλαπλασιασμένα με τον αντίστοιχο συντελεστή που έχει καθορίσει ο χρήστης). Αφού υπολογιστεί και το μητρώο των μετακινήσεων ('displacementMatrix') και μετασχηματιστούν όλα τα παραπάνω, επιλύονται οι γνωστές σχέσεις από τις οποίες προκύπτουν οι άγνωστες μετακινήσεις και δράσεις (αντιδράσεις). Σε περίπτωση που υπάρχει ελαστική στήριξη και πριν δοθούν οι τελικές τιμές των μετακινήσεων και στηρίξεων, υπολογίζονται τα τελευταία για την ελαστική στήριξη με τη χρήση του 'calculateElasticNode'. Το επόμενο βήμα είναι να υπολογιστούν τα εσωτερικά εντατικά μεγέθη των μελών. Αυτό γίνεται με το ('calculateEdgeForces'). Τέλος, για να υπολογιστούν τα διαγράμματα των εντατικών μεγεθών, χρησιμοποιείται ένα βήμα 1cm κατά μήκος του κάθε μέλους, και οι τιμές αποθηκεύονται σε κατάλληλη δομή δεδομένων.

Όλα τα παραπάνω αποτελέσματα, στέλνονται στο παράθυρο από το οποίο εκτελέστηκε η ανάλυση, και αποθηκεύονται αυτόματα σε αρχείο txt.

### 3.2.3 Το αρχείο frame3d.py

#### 3.2.3.1 Γενικά στοιχεία

Είναι το κομμάτι του οποίου σχεδόν αποκλειστική λειτουργία είναι η λήψη των δεδομένων από το GUI.py, επεξεργασία και αποθήκευσή τους στις κατάλληλες δομές δεδομένων, με στόχο να λαμβάνονται από το αρχείο της ανάλυσης,

αλλά και να εκτελείται η αποθήκευση της εργασίας με τον αποτελεσματικότερο δυνατό τρόπο.

Κάνει χρήση του numpy και του pandas καθώς και του αρχείου που υπολογίζει τις δράσεις παγίωσης. Αποτελείται από μία κλάση (Beam) και από έναν μικρό αριθμό λειτουργιών (functions) εκτός κλάσης. Πριν την σύντομη ανάλυση της λειτουργίας του αρχείου κρίνεται σκόπιμο, για την καλύτερη κατανόηση, να αναφερθεί ότι δεν έγινε πλήρης εκμετάλλευση των χαρακτηριστικών της γλώσσας Python που της προσδίδουν αντικειμενοστρέφεια. Έτσι, δημιουργήθηκε μόνο μια κλάση, η οποία συγκεντρώνει όλα τα χαρακτηριστικά του φορέα. Παρά το γεγονός ότι ο τύπος των μελών είναι μόνο ένας (μέλη χωρικού φορέα χωρίς διάκριση μεταξύ υποστυλωμάτων - δοκών) θα ήταν ίσως δυνατόν με μια διαφορετική αντιμετώπιση να χρησιμοποιηθούν περισσότερες π.χ. για κόμβους, για ομάδες διατομών κτλ. Μια τέτοια αλλαγή όμως, στο μέσο της συγγραφής του προγράμματος, θα επηρέαζε εκτενώς όλα τα αρχεία, και κρίθηκε ιδιαίτερα χρονοβόρα για να εκτελεστεί. Επιπλέον, κρίθηκε ότι θα βελτίωνε περισσότερο την αναγνωσιμότητα του κώδικα και λιγότερο την λειτουργία του.

### 3.2.3.2 Αναλυτική Περιγραφή

Με την εισαγωγή ενός νέου μέλους από τον χρήστη, η κλάση Beam κατασκευάζει ένα αντικείμενο ‘μέλος’, με τα παρακάτω αρχικά χαρακτηριστικά:

start_node	Κόμβος αρχής
stop_node	Κόμβος πέρατος
start	Συντεταγμένες αρχής
stop	Συντεταγμένες πέρατος
name	Όνομα μέλους
group	Ομάδα διατομής
E,A,G,v,Ix,Iy,Iz, selfload, sectiontype	Χαρακτηριστικά της διατομής του μέλους
L	Μήκος
theta	Γωνία στροφής περί τον τοπικό άξονα x
axes	Μοναδιαία διανύσματα των τοπικών αξόνων στο καθολικό

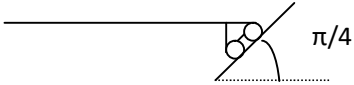
	σύστημα συντεταγμένων
rotv	3x3 πίνακας στροφής του μέλους
rotMatrix	Το μητρώο περιστροφής του μέλους

Εδώ σημειώνεται ότι ο χρήστης δεν έχει καμία εμπλοκή με τα ονόματα τόσο των μελών όσο και των κόμβων. Κάθε φορά που προσθέτει νέο μέλος, ονομάζονται αυτόματα αυτό και οι τυχόν νέοι κόμβοι που δημιουργούνται. Τέλος, δίνεται και το χαρακτηριστικό 'release', που περιγράφει τυχόν εσωτερικές ελευθερώσεις του μέλους.

Με την εισαγωγή κάθε νέου μέλους οι μοναδικοί υπολογισμοί που εκτελούνται στο παρόν είναι αυτοί που απαιτούνται για τα χαρακτηριστικά του πίνακα (μήκος, νέα ονόματα, μητρώα στροφής κτλ.)

Επίσης, το Beam περιλαμβάνει εκείνες τις δομές δεδομένων που απαιτούνται για την πλήρη περιγραφή της γεωμετρίας του φορέα:

Beams	Λεξικό (dictionary) με κλειδιά (keys) τα ονόματα των μελών και τιμές (values) τα αντικείμενα (objects) της Python που δημιουργούνται από την κλάση Beam
Nodes	Λεξικό με κλειδιά τα ονόματα των κόμβων και τιμές λίστες με τις συντεταγμένες τους. Οι τιμές των συντεταγμένων δίνονται σε μέτρα.
NodalSupports	Λεξικό με κλειδιά τα ονόματα των κόμβων και τιμές λίστες με τους περιορισμούς ή μη των βαθμών ελευθερίας. Η λίστα αποτελείται από 6 στοιχεία, ένα για κάθε βαθμό ελευθερίας, με τα τρία πρώτα να παριστάνουν μετατοπίσεις και τα τρία τελευταία στροφές. Ελευθερία βαθμού δίνεται από την τιμή 0, ενώ περιορισμός από την τιμή 1. Έτσι, ένας κόμβος χωρίς καμία στήριξη θα έχει τιμή τη λίστα [0,0,0,0,0,0], μια πλήρης πάκτωση την [1,1,1,1,1,1] και μια άρθρωση την [1,1,1,0,0,0]. Τονίζεται ότι το

	λεξικό περιέχει όλους τους κόμβους ανεξαρτήτως στήριξης.
RotatedSupports	Λεξικό με κλειδιά τα ονόματα των κόμβων στους οποίους υπάρχει στροφή σε στήριξη και τιμές λίστες τριών στοιχείων. Το κάθε στοιχείο της λίστας περιγράφει τη δεξιόστροφη γωνία κατά την οποία περιστρέφεται η στήριξη, κατά τους άξονες x, y, z, αντίστοιχα. Οι γωνίες δίνονται σε rad.
Displacements	Λεξικό με κλειδιά τα ονόματα των κόμβων οι οποίοι υπόκεινται σε μετατοπίσεις ή στροφές και τιμές λίστες έξι στοιχείων που παριστάνουν αντίστοιχα τις μετατοπίσεις και τις στροφές κατά τους τρεις άξονες. Οι μετατοπίσεις εισάγονται σε μέτρα και οι στροφές σε rad.
ElasticNode	Λεξικό με κλειδιά τα ονόματα των κόμβων που διαθέτουν ελατηριακή στήριξη και τιμές λίστες έξι στοιχείων που παριστάνουν αντίστοιχα τις δυσκαμψίες κατά τους έξι βαθμούς ελευθερίας (μεταφορικούς και στροφικούς). Οι τιμές δίνονται σε KN/m.
RigidNode	<p>Λεξικό με κλειδιά τα ονόματα των μελών που έχουν κάποια ελευθέρωση και τιμές λίστες δύο στοιχείων. Το κάθε στοιχείο είναι μία υπολίστα τριών στοιχείων που έχει τιμές τις γωνίες κατά τις οποίες λαμβάνει χώρα η ελευθέρωση. Οι γωνίες είναι δεξιόστροφες και υπολογίζονται στο τοπικό σύστημα συντεταγμένων του κάθε μέλους. Η πρώτη κύρια λίστα αφορά την αρχή του μέλους ενώ η δεύτερη το πέρας. Έτσι, το μέλος της εικόνας:</p> 

	<p>θα έχει τιμή <math>[[0,0,0],[0,-\pi/4,0]]</math>, αν θεωρήσουμε τον άξονα y με κατεύθυνση από τον αναγνώστη προς το κείμενο.</p>
Loads	<p>Είναι η κύρια δομή δεδομένων της βιβλιοθήκης pandas και ονομάζεται Dataframe και ουσιαστικά αποτελεί έναν πίνακα με πλήθος στηλών, που η καθεμία περιέχει ένα συγκεκριμένο χαρακτηριστικό του φορτίου. Οι λεπτομέρειες της συγκεκριμένης δομής αναλύονται παρακάτω.</p>
Groups	<p>Λεξικό με κλειδιά τα ονόματα των ομάδων διατομών (group) που δημιουργούνται από τον χρήστη και τιμές λίστες δύο στοιχείων. Το πρώτο στοιχείο της λίστας αποτελεί μια νέα λίστα που έχει αποθηκευμένες τις τιμές των χαρακτηριστικών της διατομής E,A,G,v,Ix,Iy,Iz και selfload. Το δεύτερο στοιχείο της λίστας είναι ο τύπος της διατομής. Έτσι αν πρόκειται για πρότυπη διατομή, το δεύτερο στοιχείο θα είναι το π.χ. 'HE100' και οι τιμές των χαρακτηριστικών αυτές που υπάρχουν αποθηκευμένες και ανασύρονται αυτόματα από το πρόγραμμα. Αντίθετα, αν η διατομή είναι τυχαία, θα ονομάζεται 'custom' και οι τιμές των χαρακτηριστικών της θα είναι αυτές ακριβώς που θα έχει εισάγει ο χρήστης. Η χρήση της συγκεκριμένης δομής εξυπηρετεί αποκλειστικά την αποθήκευση δεδομένων και εμφάνισής τους στο παράθυρο, καθώς τα παραπάνω χαρακτηριστικά αποτελούν έτσι κι αλλιώς μεταβλητές του κάθε μέλους από την κατασκευή του.</p>
LoadGroups	<p>Λίστα με τα ονόματα των τύπων φορτίων (μόνιμα, κινητά κτλ). Ενσωματώθηκε στο πρόγραμμα με</p>

	σκοπό να μπορεί ο χρήστης να δημιουργεί και να ονομάζει τύπους φορτίσεων ώστε να τους χρησιμοποιεί στους συνδυασμούς που επιθυμεί. Εδώ η χρήση είναι περιορισμένη και περιλαμβάνει μόνο μόνιμα και κινητά φορτία.
--	---

### 3.2.3.3 Η δομή 'Loads'

Ο παρακάτω πίνακας είναι μέρος του πίνακα που συνθέτει το 'Loads' και απεικονίζει ακριβώς τη μορφή της δομής δεδομένων Dataframe:

	ITEM	node	member	function	self_load	vector	group	...
0	LOAD	n1		Nodal load	NO	[0,0,-50]	G	...
1	LOAD	-	e1	Po	NO	[10,30,-20]	Q	...
2	LOAD	-	e3	FE	YES	[0,0,-100]	G	...
...	...	...	...	...	...	...	...	

Όλες οι στήλες του πίνακα και η χρήση τους φαίνονται παρακάτω:

ITEM	Λαμβάνει την ίδια τιμή για όλα τα φορτία και χρησιμοποιείται για την αποθήκευση των δεδομένων κατά το κλείσιμο του προγράμματος
node	Αφορά μόνο τα επικόμβια φορτία και φέρει τα ονόματα των κόμβων που φορτίζονται
member	Αφορά μόνο τα φορτία των μελών και φέρει τα ονόματα αυτών που δέχονται φόρτιση
function	Περιγράφει το είδος του φορτίου (σημειακό, κατανεμημένο

	ομοιόμορφα ή τριγωνικά κτλ) και παίρνει τιμές τους κωδικούς με τους οποίους σημαίνονται τα φορτία και περιγράφονται αναλυτικότερα παρακάτω στο αρχείο ireact3d.py
a	Η απόσταση (αν υπάρχει) από την αρχή του μέλους έως την έναρξη του φορτίου. Για ένα σημειακό φορτίο είναι πάντα διάφορη του μηδενός, με τοποθέτηση περιορισμών στις τιμές που εισάγονται, ενώ για τα υπόλοιπα θα είναι πάντα μεγαλύτερη ή ίση του μηδενός και μικρότερη του μήκους του μέλους. Για τα επικόμβια φορτία δεν υφίσταται τιμή.
b	Η απόσταση (αν υπάρχει) από το πέρας του φορτίου έως το πέρας του μέλους. Για ένα σημειακό φορτίο υπολογίζεται αυτόματα, ενώ για τα υπόλοιπα θα είναι πάντα μεγαλύτερη ή ίση του μηδενός και μικρότερη του μήκους του μέλους. Επίσης, τίθενται περιορισμοί κατά την εισαγωγή των δύο αποστάσεων ώστε το άθροισμά τους να είναι μικρότερο του μήκους του μέλους. Για τα επικόμβια φορτία δεν υφίσταται τιμή.
group	Ο τύπος του φορτίου (μόνιμο G ή κινητό Q)
nodestart	Ο κόμβος αρχής του μέλους
loadsstart	Οι δράσεις παγίωσης στην αρχή του μέλους, όπως αυτές υπολογίστηκαν από το αρχείο ireact3d.py και όπως αναλύεται παρακάτω. Οι δράσεις παγίωσης είναι από τα ελάχιστα στοιχεία που υπολογίζονται κατά τη διάρκεια εισαγωγής των δεδομένων και όχι στη φάση της ανάλυσης. Αν ένα φορτίο αλλάξει από τον χρήστη, αλλάζει κ η αντίστοιχη τιμή των δράσεων.
nodestop	Ο κόμβος πέρατος του μέλους
loadsstop	Ομοίως με τις δράσεις παγίωσης στην αρχή του μέλους, δίνονται οι δράσεις παγίωσης στο πέρας του.
loadsstart_release	Οι δράσεις στην αρχή και το πέρας του μέλους στην περίπτωση που αυτό έχει κάποια ελευθέρωση
self_load	Παίρνει την τιμή 'YES' μόνο αν το φορτίο αποτελεί το ίδιο

	βάρος του μέλους, διαφορετικά 'NO'
vector	Δίνει το διάνυσμα του φορτίου, το οποίο εισάγεται στο καθολικό σύστημα συντεταγμένων

### 3.2.3.4 Λειτουργίες (functions)

Το αρχείο frame3d.py περιέχει τις λειτουργίες που παρουσιάζονται στον παρακάτω πίνακα. Οι πρώτες ανήκουν στην κλάση Beam (με το πρόθεμα 'Beam.') ενώ οι τελευταίες είναι ανεξάρτητες:

Beam.initValues	Καλείται αν υπάρχει κάποια αλλαγή στα στοιχεία του μέλους (γωνία στροφής διαμήκους άξονα) κατά τη διάρκεια μόρφωσης του φορέα και οπωσδήποτε κατά την έναρξη της φάσης της ανάλυσης, όπου συνοψίζονται όλα τα στοιχεία του φορέα. Αποθηκεύει την νέα γωνία στροφής (αν υπάρχει), τους τοπικούς άξονες του μέλους, τα μητρώα περιστροφής και προκαλεί μέσω άλλης λειτουργίας την αποθήκευση του τοπικού και καθολικού μητρώου δυσκαμψίας. Αν υπάρχει στερεός κόμβος ή κάποια εσωτερική ελευθέρωση, υπολογίζει μέσω άλλων λειτουργιών και αποθηκεύει τα κατάλληλα μητρώα δυσκαμψίας.
Beam.groupVals	Αποθηκεύει ως χαρακτηριστικά του κάθε μέλους τα χαρακτηριστικά της διατομής που έχει αποδοθεί σε αυτό.
Beam.getKEI	Αποθηκεύει το τοπικό και καθολικό μητρώο δυσκαμψίας του μέλους.
Beam.getKEIReleased	Αποθηκεύει το τοπικό και καθολικό μητρώο δυσκαμψίας του μέλους με τους κατάλληλους μετασχηματισμούς, αν υπάρχει εσωτερική ελευθέρωση.
Beam.getRigidK	Αποθηκεύει το τοπικό και καθολικό μητρώο δυσκαμψίας του μέλους με τους κατάλληλους

	μετασχηματισμούς, αν υπάρχει στερεός κόμβος σε κάποιο άκρο του.
Beam.beamLength	Υπολογίζει και επιστρέφει το μήκος του μέλους από τις συντεταγμένες των άκρων του.
Beam.releaseEdge	Διαχειρίζεται την τυχούσα εσωτερική ελευθέρωση και καλεί τη λειτουργία που αποθηκεύει τα μητρώα δυσκαμψίας.
Beam.localAxes	Υπολογίζει και επιστρέφει τα μοναδιαία διανύσματα των τοπικών αξόνων του κάθε μέλους.
Beam.rotationMatrix	Επιστρέφει τα μητρώα περιστροφής (3x3 και 12x12) του μέλους.
Beam.rot_theta	Επιστρέφει το μητρώο μετασχηματισμού των τοπικών αξόνων στην περίπτωση που υπάρχει γωνία στροφής περί τον διαμήκη άξονα x.
addElement	Δημιουργεί ένα νέο μέλος και το αποθηκεύει στην δομή δεδομένων Beams. Χρησιμοποιείται από το αρχείο του γραφικού περιβάλλοντος και εισάγονται τα δεδομένα που εισάγει ο χρήστης.
transformReleases	Επιστρέφει το μητρώο μετασχηματισμού που αφορά τις εσωτερικές ελευθερώσεις.
nodeNextName	Υπολογίζει και επιστρέφει το επόμενο ελεύθερο όνομα ώστε να δοθεί σε δημιουργούμενο κόμβο.
beamNextName	Υπολογίζει και επιστρέφει το επόμενο ελεύθερο όνομα ώστε να δοθεί σε δημιουργούμενο μέλος.
edgeRotationMatrix	Υπολογίζει και επιστρέφει το μητρώο περιστροφής για άκρο μέλους που έχει ελευθέρωση με στροφή.
rotateSupport	Επιστρέφει το μητρώο περιστροφής κεκλιμένης στήριξης.
elasticNode	Επιστρέφει το μητρώο δυσκαμψιών στην περίπτωση ελαστικών στηρίξεων.
nodalLoad	Αποθηκεύει στην δομή δεδομένων των φορτίων τα επικόμβια φορτία.
getLoadVals	Υπολογίζει και επιστρέφει το σύνολο των δράσεων

	παγίωσης και τις επιστρέφει στην κατάλληλη μορφή τους για αποθήκευση και επεξεργασία από το πρόγραμμα.
beamLoad	Αποθηκεύει στην δομή δεδομένων των φορτίων τα φορτία επί των μελών.
getComponents	Επιστρέφει τις συνιστώσες ενός φορτίου στο τοπικό σύστημα συντεταγμένων του κάθε μέλους (άρα επιστρέφει μόνο τιμές των τριών συνιστωσών αφού στη φάση αυτή δεν ενδιαφέρουν τα διανύσματα) προκειμένου να υπολογιστούν οι δράσεις παγίωσης.
getPointReactions	Επιστρέφει τις δράσεις παγίωσης που αφορούν σε σημειακό φορτίο
getUniformReactions	Επιστρέφει τις δράσεις παγίωσης που αφορούν σε ομοιόμορφα κατανεμημένο φορτίο
getTriangReactions	Επιστρέφει τις δράσεις παγίωσης που αφορούν σε τριγωνικά κατανεμημένο φορτίο
getTrapezoidReactions	Επιστρέφει τις δράσεις παγίωσης που αφορούν σε τραπεζοειδώς κατανεμημένο φορτίο
nodalSupport	Διαχειρίζεται μια στήριξη κόμβου και την αποθηκεύει στη δομή που ανήκει.
deleteBeam	Διαχειρίζεται τη διαγραφή των μελών και ό,τι άλλο προκύπτει από την διαγραφή τους (κόμβοι, φορτία κτλ)
deleteNode	Διαχειρίζεται τη διαγραφή των κόμβων και ό,τι άλλο προκύπτει από την διαγραφή τους (μέλη, φορτία, στηρίξεις κτλ)
displacement	Διαχειρίζεται μετατοπίσεις ή στροφές στις οποίες υπόκειται ένας κόμβος και τις αποθηκεύει στη δομή που ανήκουν.

### 3.2.4 Το αρχείο ireact3d.py

Η λειτουργία του είναι αποκλειστικά ο υπολογισμός των δράσεων παγίωσης των μελών. Αρχικά είχε χρησιμοποιηθεί η βιβλιοθήκη sympy, η οποία δίνει δυνατότητα χρήσης συμβολικών παραστάσεων και επίλυσης ορισμένων ολοκληρωμάτων. Έτσι, τα φορτία που εφαρμόζονταν μπορούσαν να πάρουν οποιαδήποτε μορφή επιθυμούσε ο χρήστης, με την εισαγωγή απλώς της συμβολικής παράστασης, π.χ. ευθείες κεκλιμένες ή μη και καμπύλες. Καθώς όμως η χρήση της βιβλιοθήκης παρουσίαζε προβλήματα στην ενσωμάτωση με το CxFreeze, η συγκεκριμένη προσέγγιση εγκαταλείφθηκε. Αντ' αυτού, οι δράσεις παγίωσης δίνονται με τη χρήση έτοιμων σχέσεων, ανάλογα με το φορτίο που εισάγει ο χρήστης (π.χ. η ροπή πάκτωσης στην αρχή ενός μέλους μήκους  $L$  λόγω ενός σημειακού φορτίου  $P$  σε απόσταση  $a$  από την αρχή και  $b$  από το τέλος είναι  $P*a*b^2/L^2$ ).

Τα φορτία που εισάγονται συνοδεύονται από έναν κωδικό όπως παρακάτω:

FE	Ομοιόμορφα κατανεμημένο φορτίο σε ολόκληρο το μήκος του μέλους
PE	Ομοιόμορφα κατανεμημένο φορτίο σε μέρος του μήκους του μέλους
FT	Τριγωνικά κατανεμημένο φορτίο σε ολόκληρο το μήκος του μέλους
PT	Τριγωνικά κατανεμημένο φορτίο σε μέρος του μήκους του μέλους
Po	Σημειακό φορτίο σε κάποιο σημείο του μέλους
M	Σημειακή ροπή
Επιπλέον, υπάρχει δυνατότητα εισαγωγής και τραπεζοειδούς φορτίου, το οποίο δεν υπολογίζεται από ξεχωριστή λειτουργία αλλά από τον συνδυασμό ομοιόμορφα κατανεμημένου και τριγωνικού.	

Οι παραπάνω κωδικοί αφορούν μόνο στη μορφή του φορτίου, με την έννοια ότι στον κώδικα εμφανίζονται τόσο με το πρόθεμα 'c' όσο και με το πρόθεμα

‘a’, π.χ. cFE, cPE, aFE, aPE κτλ. Η χρήση του ‘c’ είναι για τον υπολογισμό της τέμνουσας στη διεύθυνση της κάθε συνιστώσας του φορτίου και της ροπής στον κάθετο σε αυτό άξονα.

Έτσι, για παράδειγμα ένα τριγωνικό φορτίο θα υπολογίζεται ως:

```
36 def cFT(eL, q, a, b):
37     #για αυξανόμενο τριγωνικό. για φορτίο που μειώνεται αναποδα τα Q1,Q2,M1,M2
38     q1, q2 = q
39     l = eL.L
40     Q1 = 3*q2*l/20
41     Q2 = 7*q2*l/20
42     M1 = q2*l*l/30
43     M2 = q2*l*l/20
44
45     if q1 == 0:
46         return M1, -M2, -Q1, -Q2
47
48     else:
49         Q1, Q2 = Q2, Q1
50         M1, M2 = M2, M1
51         return M1, -M2, -Q1, -Q2
52
```

Αρχικά γίνεται ο υπολογισμός για ένα αυξανόμενο τριγωνικά φορτίο, το οποίο εφαρμόζεται σε όλο το μήκος του μέλους, και στη συνέχεια, αν βρεθεί ότι το φορτίο είναι μειούμενο, αντιστρέφονται οι τιμές των ακραίων δράσεων.

Το πρόθεμα ‘a’ χρησιμοποιείται για τον υπολογισμό της αξονικής δράσης που οφείλεται στη συγκεκριμένη συνιστώσα. Το πρόγραμμα, κατά τον υπολογισμό των δράσεων, ελέγχει όλες τις συνιστώσες του κάθε φορτίου και τις μετατρέπει σε διανύσματα στο τοπικό σύστημα συντεταγμένων του κάθε μέλους. Στη συνέχεια, υπολογίζει για την κάθε μία τις δράσεις και στο τέλος τις συνθέτει σε ένα διάνυσμα στο καθολικό σύστημα. Επισημαίνεται ότι τα δεδομένα που καλείται να εισάγει ο χρήστης είναι επιλογή μεταξύ σημειακού, ομοιόμορφα κατανεμημένου, τριγωνικού και τραπεζοειδούς φορτίου, και την ακριβή τοποθέτηση του φορτίου επί του μέλους. Η διαδικασία χρήσης των παραπάνω κωδικών δεν τον αφορά.

Τέλος, προκειμένου να μειωθούν οι γραμμές του προγράμματος και να γίνει μια πιο τεχνική προγραμματιστικά προσέγγιση, χρησιμοποιείται η λειτουργία:

```

9
10 def comInLoads(function, el, q, a=0, b=0):
11
12     if function[0]=='c':
13         m1,m2,q1,q2 = getattr(sys.modules[__name__], function)(el,q, a, b)
14         return m1,m2,q1,q2
15     elif function[0]=='a':
16         n1, n2 = getattr(sys.modules[__name__], function)(el,q, a, b)
17         return n1, n2
18

```

Όλοι οι υπολογισμοί των ακραίων δράσεων φιλτράρονται από την συγκεκριμένη λειτουργία, επιλέγεται κάθε φορά ο ζητούμενος και ουσιαστικά είναι αυτή που τελικά επιστρέφει τις υπολογισθείσες τιμές των δράσεων.

### 3.2.5 Το αρχείο GUI.py

#### 3.2.5.1 Γενικά

Περιέχει όλες τις λειτουργίες που αφορούν στο κομμάτι του περιβάλλοντος εργασίας. Είναι αυτό που δημιουργεί το αρχικό παράθυρο κατά την έναρξη της εφαρμογής και που ελέγχει όλα τα υπόλοιπα τμήματα. Κατά τη εκτέλεσή του διαβάζονται από τον υπολογιστή όλα τα αρχεία που χρησιμοποιούνται από το πρόγραμμα. Έτσι, φορτώνονται στη μνήμη τόσο τα αρχεία που θα χρησιμοποιηθούν αργότερα για την ανάλυση των φορέων, όσο και τα δεδομένα των διατομών που είναι διαθέσιμες στον χρήστη. Εδώ, επίσης, λαμβάνει χώρα η επεξεργασία των δεδομένων για την τρισδιάστατη απεικόνιση, η αποθήκευση και επαναφόρτωση εργασιών του χρήστη, αποθήκευση των αποτελεσμάτων αναλύσεων, προεπισκόπηση των αναλύσεων που έχουν περατωθεί για κάθε φορέα – εργασία ξεχωριστά και άνοιγμα αρχείων αποτελεσμάτων. Τέλος, από εδώ δίνεται η εντολή της ανάλυσης με εισαγωγή κάποιων παραμέτρων.

#### 3.2.5.2 Σύντομη Περιγραφή

Στον παρακάτω πίνακα φαίνονται οι κλάσεις που χρησιμοποιούνται με μια σύντομη περιγραφή τους.

MainWindow	Το κυρίως παράθυρο του προγράμματος όπου εισάγονται τα δεδομένα από τον χρήστη και γίνεται η εποπτική
------------	---

	παρουσίαση της γεωμετρίας του φορέα και των επιμέρους χαρακτηριστικών του.
QPosFloatValidator	Η ουσιαστική της λειτουργία είναι να περιορίζει τις τιμές που εισάγονται ώστε να έχουν τη μορφή θετικού δεκαδικού αριθμού.
QFloatValidator	Η ουσιαστική της λειτουργία είναι να περιορίζει τις τιμές που εισάγονται ώστε να έχουν τη μορφή δεκαδικού αριθμού.
StructureViewer	Η ενσωματωμένη στο κυρίως παράθυρο οθόνη προβολής του φορέα.
AnalysisWindow	Το παράθυρο που ανοίγει όταν ο χρήστης εκτελεί την ανάλυση.
ViewResultsWindow	Το παράθυρο που ανοίγει όταν ο χρήστης επιθυμεί να εντοπίσει αποθηκευμένες αναλύσεις του συγκεκριμένου φορέα στον οποίο εργάζεται.

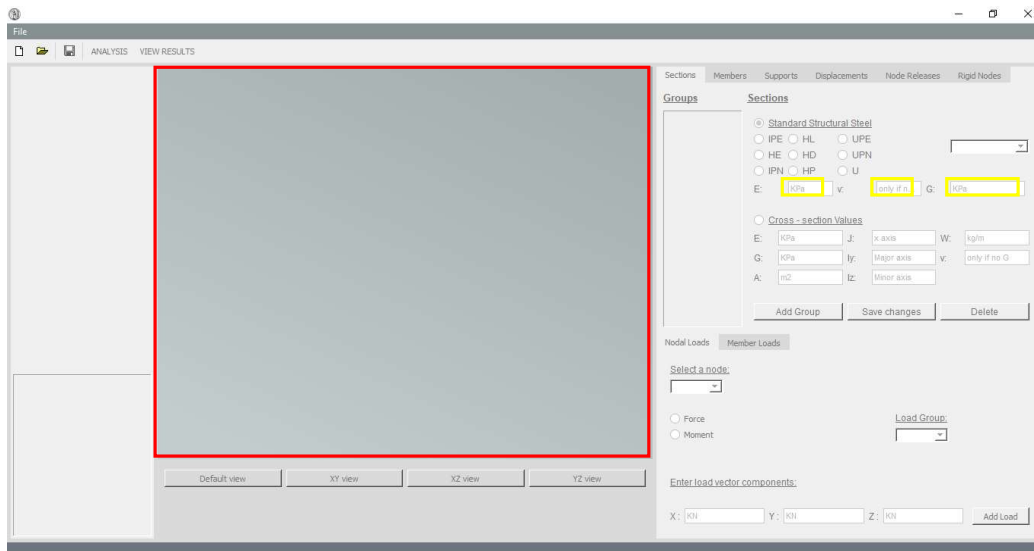
Η κάθε κλάση έχει αποκλειστικά τις δικές τις λειτουργίες, με κάποιες από αυτές να είναι όμοιες, λόγω αδυναμίας της κάθε κλάσης να έχει πρόσβαση σε αυτές των υπολοίπων. Σε γενικές γραμμές η εκτέλεση του αρχείου του περιβάλλοντος εργασίας έχει ως εξής:

- Κατά την εκτέλεση του αρχείου και αφού διαβαστούν όλες οι γραμμές, στο τέλος εκτελείται η λειτουργία main. Αυτή είναι που καλεί την κλάση MainWindow.
- Η κλάση MainWindow δημιουργεί το κεντρικό παράθυρο. Όταν κλείσει αυτό το παράθυρο τερματίζεται και το πρόγραμμα. Καμία λειτουργία του συγκεκριμένου παραθύρου δεν είναι δυνατόν να κληθεί από άλλη κλάση ή άλλο αρχείο, διότι αυτό αυτομάτως θα σήμαινε δημιουργία νέου παραθύρου, όμοιου με το αρχικό (με τις αρχικές συνθήκες – άρα επανέναρξη του προγράμματος σε νέο παράθυρο). Αυτός είναι και ο λόγος για τον

οποίο οι λειτουργίες του κεντρικού μπορούν να κληθούν για εκτέλεση μόνο από το ίδιο.

- Οι κλάσεις `QPosFloatValidator` και `QFloatValidator` καλούνται κατά τη δημιουργία εκείνων των αντικειμένων του κεντρικού παραθύρου που τις χρησιμοποιούν. Ο χρήστης δεν το αντιλαμβάνεται παρά μόνο ως περιορισμό όταν θα δοκιμάσει να εισάγει κάποια τιμή μη αποδεκτή
- Η κλάση `StructureViewer` καλείται επίσης κατά την έναρξη του κεντρικού παραθύρου και δημιουργεί την οθόνη απεικόνισης του φορέα.
- Τέλος, οι κλάσεις `AnalysisWindow` και `ViewResultsWindow` καλούνται να δημιουργήσουν τα παράθυρα της ανάλυσης και της εύρεσης αποθηκευμένων αποτελεσμάτων αντίστοιχα, μόνο όταν ο χρήστης το ζητήσει. Αυτά τα παράθυρα, μπορούν να ανοίξουν και να κλείσουν όσες φορές θέλει ο τελευταίος, ενώ υπάρχει περιορισμός των ενεργειών επί του κεντρικού παραθύρου όταν ένα από αυτά είναι ενεργοποιημένο. Συνεπώς, ποτέ δεν μπορούν να είναι ανοιχτά, συγχρόνως, πάνω από δύο παράθυρα (το κεντρικό και ένα ακόμα) με δυνατότητα εκτέλεσης εργασίας μόνο σε ένα.

Παρακάτω φαίνεται το κεντρικό παράθυρο όταν ανοίξει, χωρίς κανένα δεδομένο. Στο κόκκινο πλαίσιο είναι η οθόνη που δημιουργεί η κλάση `StructureViewer`, ενώ στα κίτρινα κάποια από τα αντικείμενα που χρησιμοποιούν τις `QPosFloatValidator` και `QFloatValidator`.



### 3.2.5.3 Η κλάση MainWindow

Βασίζεται στην κλάση `PyQt5.QtWidgets.QMainWindow`, που αποτελεί δομικό στοιχείο του `PyQt5`. Ουσιαστικά, η `QMainWindow` αποτελεί μια δεξαμενή λειτουργιών, τις οποίες μπορεί ο προγραμματιστής να χρησιμοποιήσει προκειμένου να διαμορφώσει το κεντρικό παράθυρο του προγράμματος όπως επιθυμεί. Ενώ υπάρχουν και άλλοι τρόποι δημιουργίας παραθύρου, όπως η χρήση μιας άλλης κλάσης, της `QDialog`, η πληρότητα της `QMainWindow` την καθιστά την πλέον κατάλληλη για το κυρίως παράθυρο.

Η πρώτη λειτουργία της κλάσης (όπως και κάθε κλάσης στην `Python`) είναι η `init()` η οποία περιέχει όλα τα αντικείμενα του γραφικού περιβάλλοντος τα οποία είναι ορατά στον χρήστη. Δημιουργεί δηλαδή όλα τα πλαίσια, κουμπιά, καρτέλες, κείμενα, ανοιγόμενα μενού κτλ. Τα αντικείμενα, κατά τη συγγραφή του κώδικα, γράφονται με τη σειρά που επιθυμεί ο προγραμματιστής. Επειδή ο όγκος είναι μεγάλος, θα πρέπει να τηρείται μια λογική σειρά (π.χ. από αριστερά προς τα δεξιά, από πάνω προς τα κάτω, πρώτα τα αντικείμενα μετά οι βασικές λειτουργίες τους κτλ), ενώ θα πρέπει απαραίτητως να γίνεται χρήση σχολίων. Εναλλακτικά, το κάθε κομμάτι του παραθύρου μπορεί να αναπτύσσεται σαν εντελώς ξεχωριστή λειτουργία (π.χ. ανάλογα με τη θέση του στο παράθυρο, ή ανάλογα με τη χρήση του στο πρόγραμμα) με όποιο τρόπο διάκρισης και επίπεδο ανάλυσης θέλουμε.

Για να γίνει το τελευταίο πιο σαφές, ο προγραμματιστής μπορεί, αντί να δημιουργήσει όλα τα αντικείμενα απευθείας μέσα στην `init()`, να δημιουργήσει τρεις ή τέσσερις ξεχωριστές λειτουργίες, μέσα στην κάθε μία να δημιουργεί τα αντικείμενα που τον ενδιαφέρουν και όλες αυτές να τις καλεί από την `init()`. Με αυτόν τον τρόπο υπάρχει σαφής διάκριση των αντικειμένων και των λειτουργιών τους.

Τα κυριότερα αντικείμενα που χρησιμοποιήθηκαν στο κεντρικό παράθυρο είναι τα παρακάτω:

<b>Ονομασία αντικειμένου</b>	<b>Περιγραφή</b>	<b>Χρήση</b>
QtWidgets.QMenuBar	Είναι η γραμμή των μενού στο πάνω μέρος του παραθύρου.	Στο παρόν τα τρία αντικείμενα είναι αλληλένδετα (χωρίς να αποκλείεται η χρήση τους με διαφορετικό τρόπο)
QtWidgets.QMenu	Δημιουργεί ένα μενού για την προαναφερθείσα γραμμή (π.χ. File)	με την δημιουργία της γραμμής να ακολουθείται από δημιουργία διαφόρων μενού και στη συνέχεια πρόσθεση επιθυμητών επιλογών στο κάθε ένα από αυτά.
QtWidgets.QAction	Δημιουργεί μία επιλογή για το προηγούμενο μενού	
QtWidgets.QWidget	Είναι μια γενική μορφή αντικειμένων η οποία σε κάποιες περιπτώσεις είναι απαραίτητη χωρίς ιδιαίτερες λειτουργίες.	Όταν δημιουργηθεί το παράθυρο (MainWindow) απαιτείται η χρήση του συγκεκριμένου αντικειμένου και τοποθέτησή του στο

		<p>παράθυρο ως centralwidget.</p> <p>Αποτελεί δηλαδή ένα είδος πρόσθετου καμβά, πάνω στον οποίο τοποθετούνται όλα τα επόμενα αντικείμενα. Δεν έχει καμιά άλλη λειτουργία.</p>
QtWidgets.QFrame	<p>Είναι ένα πλαίσιο το οποίο επίσης δέχεται τοποθέτηση αντικειμένων.</p>	<p>Η χρήση του αφορά σχεδόν αποκλειστικά σε ομαδοποιήσεις αντικειμένων, διαχωρισμό των περιοχών του παραθύρου, μορφοποιήσεις που αφορούν περιγράμματα και χρώματα κτλ.</p>
QtWidgets.QTreeWidget	<p>Πλαίσιο μορφής συμπτυσσόμενου δεντροειδούς διαγράμματος, που απεικονίζει δηλαδή τα εισαγόμενα στοιχεία ιεραρχιμένα, με το κάθε ένα να μπορεί να δεχτεί νέα λίστα από στοιχεία και ούτω καθεξής. Το συγκεκριμένο πλαίσιο μπορεί να δεχτεί ποικιλία</p>	<p>Χρησιμοποιήθηκε εκτενώς σε όλες σχεδόν τις καρτέλες εισαγωγής δεδομένων με εισαγόμενα στοιχεία μόνο γραμμές κειμένου και όχι πιο σύνθετα αντικείμενα. Υπάρχει αυτό του κεντρικού παραθύρου (maintree) καθώς και ένα σε</p>

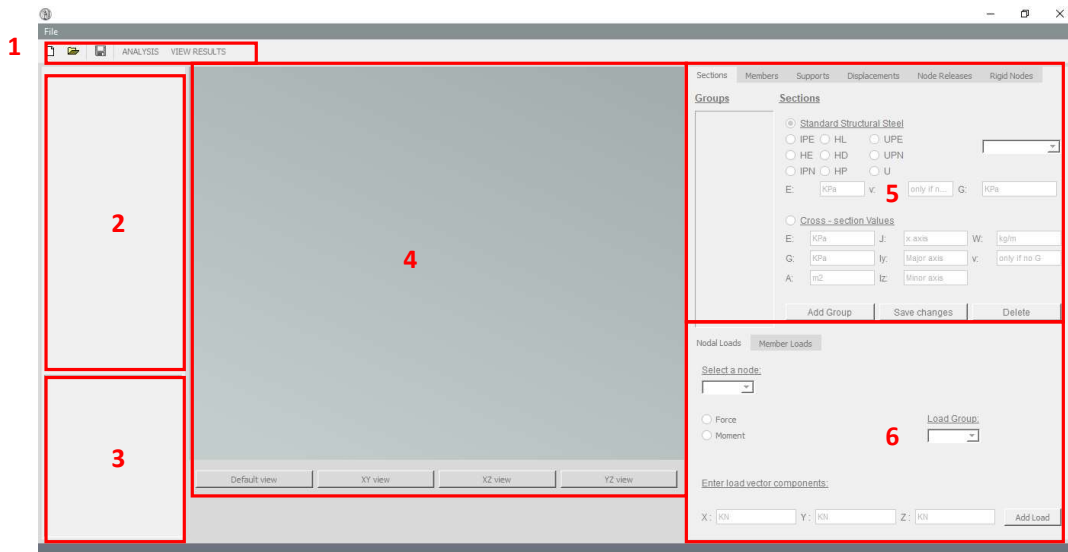
	αντικειμένων.	κάθε καρτέλα (group tree, member tree, support tree κτλ)
QtWidgets.QListWidget	Αντικείμενο παρουσίασης (εδώ μόνο κειμένου) σε μορφή λίστας, με κάθε νέο στοιχείο που τοποθετείται να μπαίνει στην επόμενη γραμμή	Είναι το πλαίσιο κάτω ακριβώς από το δέντροειδές διάγραμμα του κεντρικού παραθύρου, που εμφανίζει σύντομη περιγραφή της γεωμετρίας του φορέα.
QtWidgets.QPushButton	Είναι τα κουμπιά που πιέζει ο χρήστης και στα οποία αργότερα αντιστοιχίζονται λειτουργίες και εντολές.	Σε όλα σχεδόν τα πλαίσια και καρτέλες
QtWidgets.QTabWidget	Πλαίσιο που διαθέτει καρτέλες	Αποτελεί το πλαίσιο της εισαγωγής δεδομένων για τη μορφοποίηση του φορέα καθώς και εκείνο για την εισαγωγή φορτίων
QtWidgets.QLabel	Πλαίσιο κειμένου	Σε όλα σχεδόν τα πλαίσια και καρτέλες
QtWidgets.QRadioButton	Αντικείμενο ενεργοποίησης ή απενεργοποίησης κάποιας συγκεκριμένης επιλογής.	Είναι για παράδειγμα οι επιλογές των πρότυπων διατομών
QtWidgets.QComboBox	Αντικείμενο επιλογής με μορφή ανοιγόμενου μενού που δέχεται μια λίστα από	Είναι για παράδειγμα τα πλαίσια επιλογής κόμβων κατά την

	αντικείμενα και μπορεί να αλλάζει δυναμικά (κατά τη διάρκεια λειτουργίας του προγράμματος).	πρόσθεση νέων μελών. Αρχικά δεν διαθέτουν στοιχεία αλλά με κάθε νέο μέλος προστίθενται οι αντίστοιχοι κόμβοι τους οποίους ο χρήστης μπορεί να επιλέξει αντί να εισάγει ξανά τις ίδιες συντετεγμένες.
QtWidgets.QLineEdit	Αντικείμενο εισαγωγής κειμένου. Το εισηγμένο κείμενο λαμβάνεται και χρησιμοποιείται αναλόγως.	Όλα τα πλαίσια εισαγωγής τιμών που υπάρχουν στο παράθυρο.
QtGui.QFont	Αντικείμενο περιγραφής μορφής κειμένου.	Όλα τα αντικείμενα που μπορούν να περιέχουν κείμενο διαθέτουν ένα QFont. Στα περισσότερα αφέθηκε η προεπιλογή, ενώ σε κάποια δημιουργήθηκαν νέα, με σκοπό την ομοιογένεια και βέλτιστη αναγνώριση των διαφορετικών λειτουργιών.
QtWidgets.QGridLayout	Ρυθμίζει την τοποθέτηση των εισαγομένων αντικειμένων στο αντικείμενο – δέκτη (π.χ.	Η χρήση αποσκοπεί τόσο στην ομοιόμορφη κατανομή των

	κουμπιά σε ένα πλαίσιο) σε μορφή γραμμών και στηλών. Η ακριβής θέση τους πρέπει υποχρεωτικά να καθοριστεί από τον χρήστη.	αντικειμένων και στην ομοιογένεια των μεγεθών τους, όσο και στη διατήρηση των παραπάνω σε
QtWidgets.QVBoxLayout	Όπως το προηγούμενο αλλά σε μορφή μιας στήλης (δηλαδή υποχρεωτικά το ένα κάτω από το άλλο). Και εδώ η ακριβής θέση πρέπει να καθοριστεί.	περίπτωση αλλαγής των διαστάσεων και αναλογιών του παραθύρου.
QtWidgets.QHBoxLayout	Όπως το προηγούμενο αλλά σε μορφή μιας γραμμής (δηλαδή υποχρεωτικά το ένα δίπλα στο άλλο). Και εδώ η ακριβής θέση πρέπει να καθοριστεί.	

Για να γίνουν πιο κατανοητά τα παραπάνω δίνεται ένα απλουστευμένο παράδειγμα: Στο κεντρικό παράθυρο (Mainwindow) τοποθετείται ένα γενικής χρήσης αντικείμενο (QWidget - εδώ με ονομασία centralwidget). Πάνω σε αυτό μπορούν να τοποθετηθούν π.χ. μια γραμμή εργαλείων (QMenuBar), ένα πλαίσιο (QFrame), ένα πλαίσιο με καρτέλες (QTabWidget) και ένα πλαίσιο μορφής συμπτυσσόμενου δέντρου (QTreeWidget). Μέσα σε κάθε ένα από αυτά, τέλος, μπορούν να τοποθετηθούν κουμπιά (QPushButton), ανοιγόμενα κουτιά επιλογών (QComboBox), πλαίσια κειμένων (QLabel) και ότι άλλο θέλουμε. Επίσης, αν επιθυμούμε την κατανομή των αντικειμένων στο οποιοδήποτε πλαίσιο με συγκεκριμένες αναλογίες μεγέθους και ακριβούς θέσης μπορούμε να εφαρμόσουμε ένα layout (QGridLayout, QVBoxLayout ή QHBoxLayout).

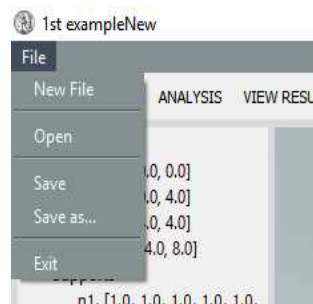
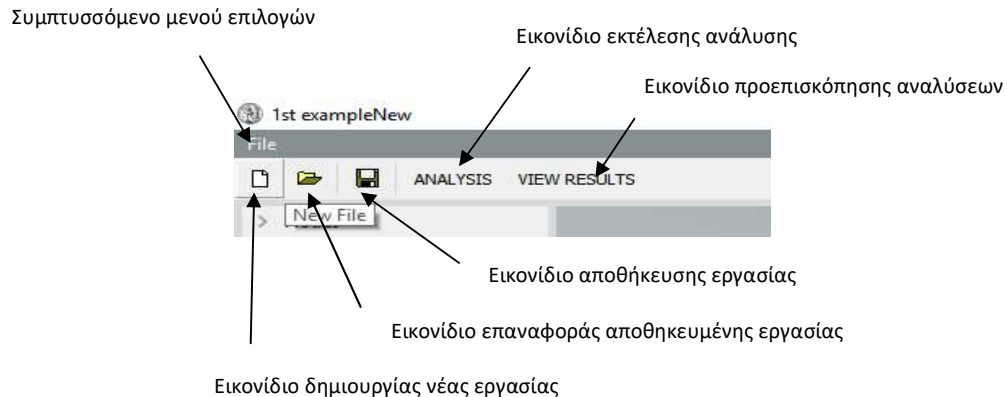
Στο πρώτο επίπεδο, τα αντικείμενα του παραθύρου που δημιουργεί η κλάση, είναι τα παρακάτω:



1. Γραμμή εργαλείων (QMenuBar). Εδώ έχουν προστεθεί δύο toolbars (γραμμές δηλαδή με εργαλεία - εικονίδια) τα οποία δεν αποτελούν αντικείμενα του PyQt5 αλλά λειτουργίες.
2. Πλαίσιο παρουσίασης στοιχείων του φορέα σε μορφή συμπυκνόμενου δέντρου (QTreeWidget). Διαθέτει λειτουργία μενού με πάτημα του δεξιού πλήκτρου του ποντικιού.
3. Υποπαραθύρο που αποτελείται από μία λίστα (QListWidget) στην οποία εμφανίζονται πληροφοριακά στοιχεία για τα στοιχεία του φορέα, κάθε φορά που ο χρήστης επιλέγει ένα από αυτά στο υποπαραθύρο 2.
4. Η οθόνη προβολής του φορέα σε τρισδιάστατη μορφή.
5. Πλαίσιο καρτελών (QTabWidget) όπου ο χρήστης εισάγει τα στοιχεία του φορέα.
6. Καρτέλες (QTabWidget) όπου εισάγονται τα φορτία. Τα δύο πλαίσια καρτελών 5 και 6 είναι τοποθετημένα μέσα σε ένα άλλο πλαίσιο (QFrame) το οποίο δεν είναι ορατό στον χρήστη και αποσκοπεί στην ρύθμιση των αναλογιών τους και την εξασφάλιση μη αλληλοεπικάλυψης.

Παρακάτω αναλύονται τα επιμέρους τμήματα του παραθύρου, χωρίς να δίνονται αναλυτικές πληροφορίες για τη λειτουργία τους (αυτές περιλαμβάνονται στο Εγχειρίδιο Χρήσης) αλλά περισσότερες λεπτομέρειες για τα αντικείμενα και τον κώδικα.

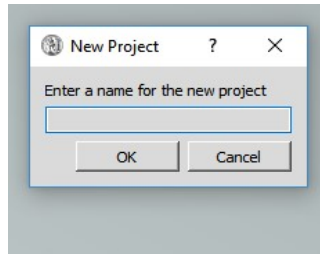
### 3.2.5.3.1 Γραμμή εργαλείων και μενού



#### Λεπτομέρειες της γραμμής εργαλείων και των λειτουργιών της.

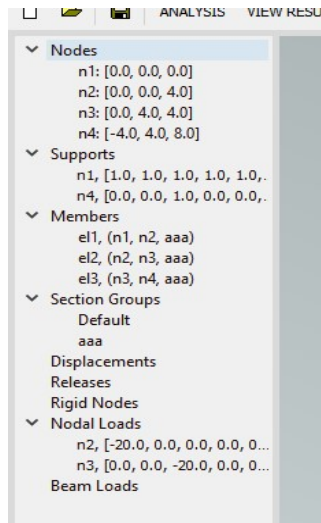
Το ανοιγόμενο μενού της γραμμής εργαλείων αποτελείται από τα αντικείμενα όπως περιγράφηκαν στον προηγούμενο πίνακα. Το 'File' είναι ένα `QtWidgets.QMenu` που τοποθετείται στην `QtWidgets.QMenuBar` και τα 'New File', 'Open', 'Save', 'Save as...' και 'Exit' είναι αντικείμενα `QtWidgets.QAction` που προστίθενται στο μενού. Με τον ίδιο ακριβώς τρόπο θα δημιουργούσαμε ένα δεύτερο ή τρίτο μενού δίπλα στο πρώτο. Τα εικονίδια της γραμμής εργαλείων δεν είναι ξεχωριστά αντικείμενα αλλά προστίθενται στην γραμμή με τη χρήση του 'addToolBar' που αποτελεί αναπόσπαστη εσωτερική λειτουργία της κλάσης `QtWidgets.QMainWindow`.

Όλα τα παραπάνω συνδέονται στη συνέχεια με μια συγκεκριμένη λειτουργία ώστε με το πάτημά τους να την εκτελούν. Έτσι, το 'New File' εκτελεί την έναρξη μιας νέας εργασίας αφού πρώτα ανοίξει το παράθυρο:



που θα ζητήσει από τον χρήστη ένα νέο όνομα για την εργασία του. Όταν ονομαστεί θα ξεκινήσει ουσιαστικά το πρόγραμμα. Γενικά, όλες οι υπόλοιπες επιλογές και εικονίδια της γραμμής ακολουθούν την επικρατούσα λογική των περισσότερων δημοφιλών προγραμμάτων.

### 3.2.5.3.2 Πλαίσιο Σύνοψης Δεδομένων



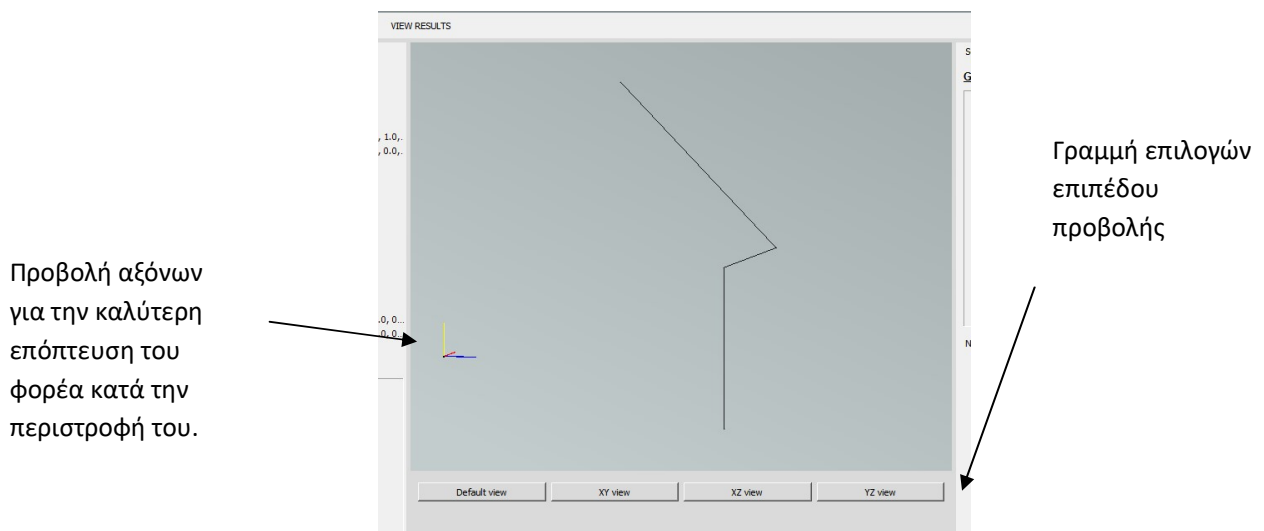
Λεπτομέρεια του αντικειμένου τύπου δέντρου (QTreeWidget), από εκτελεσμένη εφαρμογή, με τα στοιχεία που αποτελούν τον φορέα.

Η εισαγωγή και ιεράρχηση των δεδομένων στο συγκεκριμένο πλαίσιο γίνεται με τον εξής τρόπο (όλα τα παρακάτω φαίνονται σαν υλοποίηση κώδικα συγκεντρωτικά στη λειτουργία generateProjectTree της παρούσας κλάσης):

Αρχικά πρέπει να δημιουργηθούν τα κύρια αντικείμενα. Εδώ ονομάζονται όλα με τη μεταβλητή item\_0 για να μας είναι πιο εύκολο να τα ξεχωρίσουμε ως αντικείμενα πρώτου επιπέδου. Έπειτα, τοποθετούνται στην ακριβή θέση που επιθυμούμε με τη χρήση της topLevelItem που αποτελεί αναπόσπαστη λειτουργία του QTreeWidgetItem και, τέλος, προστίθενται στο κάθε ένα από αυτά τα δευτερεύοντα αντικείμενα (εδώ τα ονομάζουμε item\_1). Τονίζεται ότι μπορούν να προστεθούν όσα επίπεδα θέλουμε. Έτσι, δημιουργείται μια σαφής ιεράρχηση των δεδομένων και μπορεί ο χρήστης να αναπτύξει όποια και όσα από αυτά θέλει.

Επιπλέον, το συγκεκριμένο αντικείμενο μπορεί να έχει και λειτουργία μενού για περαιτέρω διαθέσιμες επιλογές, με πάτημα του δεξιού πλήκτρου του ποντικιού πάνω σε κάποιο από τα δεδομένα. Το μενού είναι διαφορετικό για κάθε δεδομένο και συνδέεται με διαφορετικές εντολές. Η κατασκευή και αυτού του μενού ακολουθεί τη λογική του μενού της γραμμής εργαλείων.

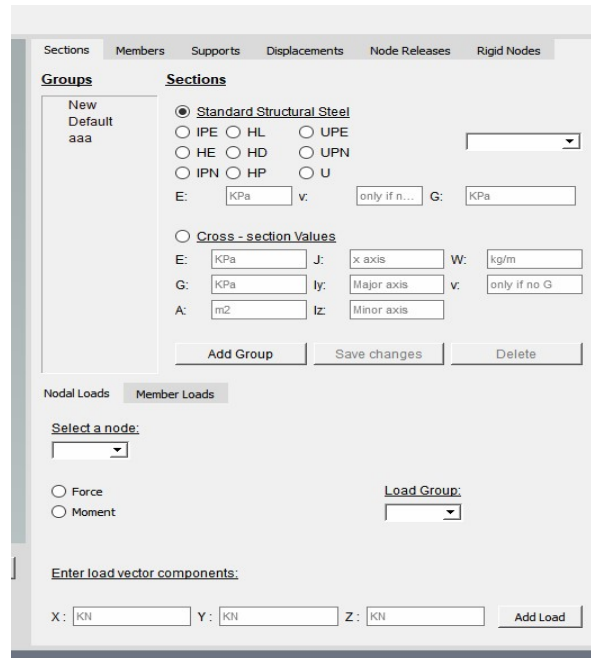
### 3.2.5.3.3 Παράθυρο τρισδιάστατης προβολής



Λεπτομέρεια της οθόνης τρισδιάστατης προβολής, από εκτελεσμένη εφαρμογή, με τα στοιχεία που αποτελούν τον φορέα.

Το συγκεκριμένο παράθυρο δημιουργείται από την κλάση StructureViewer και θα παρουσιαστεί με περισσότερες λεπτομέρειες στην αντίστοιχη παράγραφο.

### 3.2.5.3.4 Καρτέλες Εισαγωγής Δεδομένων



#### Λεπτομέρεια των πλαισίων καρτελών με ενεργοποιημένη στο πάνω μέρος την καρτέλα διατομών

Αποτελείται από ένα πλαίσιο QFrame, το οποίο περιέχει δύο πλαίσια QTabWidget, για την μόρφωση του φορέα και την εισαγωγή των φορτίων, αντίστοιχα.

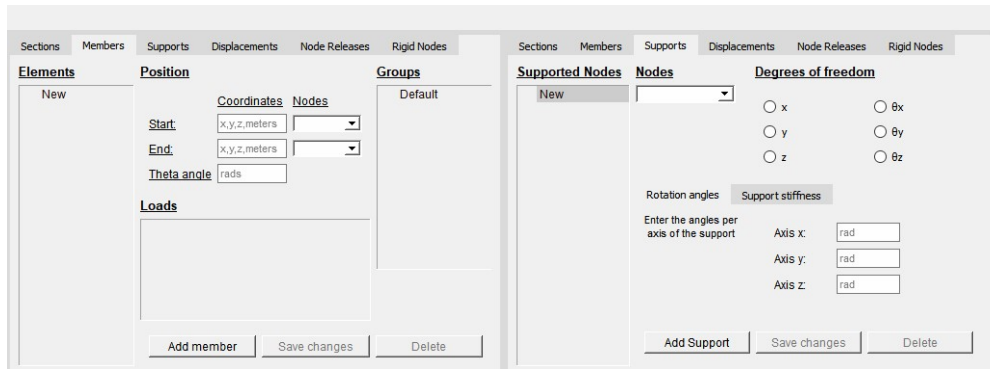
Η πρώτη καρτέλα (Sections) δέχεται δεδομένα που αφορούν τις διατομές των μελών. Στο αριστερό υπάρχει μια λίστα (QTreeWidget) με τις διατομές που εισάγει ο χρήστης, ενώ στο δεξιό η επιλογή μεταξύ προτύπων διατομών ή απλής εισαγωγής τιμών χαρακτηριστικών της διατομής. Στις πρότυπες διατομές ο χρήστης επιλέγει αρχικά έναν τύπο διατομής (αποτελούνται από αντικείμενα QRadioButton που είναι ενεργοποιημένα ή απενεργοποιημένα), ενώ στη συνέχεια και ανάλογα με τη διατομή που επέλεξε, εμφανίζονται στο ανοιγόμενο QComboBox όλα τα διαθέσιμα μεγέθη του συγκεκριμένου τύπου. Οι θέσεις εισαγωγής κειμένου (π.χ. για το μέτρο

ελαστικότητας ή λόγο Poisson) είναι QLineEdit. Στο κάτω μέρος υπάρχουν κουμπιά (QPushButton) για τη δημιουργία νέας διατομής, αποθήκευση αλλαγών σε επιλεγμένη διατομή, από τις ήδη υπάρχουσες, καθώς και διαγραφή κάποιας επιλογής.

Όταν επιλεγεί το κουμπί της δημιουργίας νέας διατομής, το πρόγραμμα συλλέγει ότι δεδομένο έχει περαστεί στην καρτέλα (είτε αυτό αποτελεί κείμενο είτε κάποια ενεργοποίηση επιλογής) και τα μεταφράζει στα δεδομένα που απαιτούνται από τα υπόλοιπα αρχεία προκειμένου να αποθηκευτούν στις δομές δεδομένων. Αν επιλεγεί αντίστοιχα το κουμπί της αλλαγής στοιχείων μιας διατομής, που προϋποθέτει την επιλογή μιας υπάρχουσας και αλλαγή των δεδομένων της, το πρόγραμμα εκτελεί ακριβώς την ίδια διαδικασία.

Η λήψη των τιμών από το αντικείμενο QLineEdit, καθώς αυτό επιστρέφει μόνο κείμενο (string), κάνει αναγκαία τη μετατροπή τους σε αριθμούς (integers ή floats). Το τελευταίο, επιβάλλει τη χρήση περιορισμών, ώστε σε περίπτωση που ο χρήστης εισάγει μη αποδεκτές τιμές, αυτές να απορριφθούν, πράγμα που εδώ επιτυγχάνεται με τη χρήση των κλάσεων QPosFloatValidator και QFloatValidator. Έτσι, αν ο χρήστης εισάγει σε κάποιο πεδίο, που περιμένει να λάβει μια αριθμητική τιμή, έναν οποιοδήποτε χαρακτήρα εκτός από αριθμό ή '.' δεν θα τον κάνει δεκτό και δεν θα τον εμφανίσει καθόλου. Ουσιαστικά, θα είναι σαν να μη λειτουργεί το πληκτρολόγιο για μη αποδεκτούς χαρακτήρες. Εναλλακτικά, μπορεί να χρησιμοποιηθεί ένα κατάλληλο πλαίσιο ειδοποιήσεων, που αρχικά να δέχεται οποιοδήποτε χαρακτήρα, αλλά όταν ο χρήστης επιλέξει την τελική δημιουργία, να εκτελείται συγκεντρωτικός έλεγχος σε όλες τις τιμές και να επισημαίνονται οι εσφαλμένες.

Επίσης, όταν ο χρήστης επιλέξει μια συγκεκριμένη διατομή από όσες έχει δημιουργήσει, το πρόγραμμα εντοπίζει ποια είναι αυτή, μεταβαίνει στη δομή δεδομένων όπου έχει αποθηκευτεί και συμπληρώνει όλα τα πεδία με τις τιμές αυτής. Το τελευταίο είναι ιδιαίτερα χρήσιμο για την εποπτεία των εισηγμένων και χρησιμοποιείται σε όλες τις καρτέλες.



Όπως φαίνεται στην εικόνα, οι υπόλοιπες καρτέλες ακολουθούν

παρόμοια λογική και περιέχουν τα ίδια αντικείμενα.

### 3.2.5.4 Η κλάση StructureViewer

Στα προηγούμενα, φάνηκε ο τρόπος με τον οποίο υλοποιείται η δημιουργία των διαφόρων αντικειμένων του κεντρικού παραθύρου. Αναφέρθηκε, ότι εκτός από το να γράφεται ο κώδικας εξ ολοκλήρου μέσα στην λειτουργία `init()` (όπως στο παρόν), μπορεί να δημιουργούνται ομαδοποιημένα όπως επιθυμεί ο χρήστης μέσα σε ξεχωριστές λειτουργίες (functions) και στη συνέχεια αυτές να καλούνται από την `init()`. Κάτι ανάλογο συμβαίνει με το τμήμα του παραθύρου που αποτελεί την οθόνη προβολής του μοντέλου του φορέα, μόνο που δεν χρησιμοποιείται απλώς μια ξεχωριστή λειτουργία, αλλά ολόκληρη κλάση. Ο λόγος για το τελευταίο είναι ότι το συγκεκριμένο αντικείμενο διαφοροποιείται σημαντικά από τα υπόλοιπα, λόγω των γραφικών που υποστηρίζει. Έτσι, ουσιαστικά το βασίζουμε σε μια διαφορετική

κλάση του PyQt5, την QtWidgets.QGraphicsView και πλέον μπορούμε να χρησιμοποιήσουμε όλες εκείνες τις ιδιαίτερες λειτουργίες που παρέχει η συγκεκριμένη κλάση.

Η διαδικασία απλουστευτικά είναι η εξής: αρχικά δημιουργείται το αντικείμενο της κλάσης StructureViewer, έπειτα τοποθετούμε ένα άλλο αντικείμενο, το QGraphicsScene, το οποίο είναι αυτό που φέρει όλα τα γραφικά στοιχεία (γραμμές, κουκίδες κτλ) και τέλος αρχίζουμε να τοποθετούμε τα τελευταία ένα προς ένα.

Αρχικά δημιουργούνται οι απαραίτητες δομές αποθήκευσης δεδομένων, που θα υποστηρίξουν τα γραφικά. Αυτές είναι οι παρακάτω:

GraphNodes	Λεξικό, με κλειδιά τα ονόματα των κόμβων του φορέα και τιμές τις συντεταγμένες τους (σε τρεις διαστάσεις). Κάθε φορά που προστίθεται ή διαγράφεται ένας κόμβος από το κεντρικό παράθυρο, οι ίδιες αλλαγές συμβαίνουν και εδώ.
GraphBeams	Λεξικό με κλειδιά τα ονόματα των μελών του φορέα και τιμές τους κόμβους αρχή – πέρατος. Κάθε φορά που προστίθεται ή διαγράφεται ένα μέλος από το κεντρικό παράθυρο, οι ίδιες αλλαγές συμβαίνουν και εδώ.
GRAPHBEAMS	Λεξικό με κλειδιά τα ονόματα των μελών του φορέα και τιμές τα γραφικά αντικείμενα που δημιουργεί η κλάση QGraphicsView. Αυτά δεν είναι απλώς γραμμές (ο χρήστης βέβαια τις αντιλαμβάνεται ως τέτοιες), αλλά έχουν πλήθος ιδιοτήτων και λειτουργιών, κάποιες από τις οποίες χρησιμοποιούνται εδώ.
GRAPHAXES	Λεξικό με κλειδιά τα ονόματα των αξόνων (x, y, z) και τιμές τους κόμβους αρχής – πέρατος του κάθε άξονα. Η συγκεκριμένη δομή δεν μεταβάλλεται, ενώ παίρνει τις συντεταγμένες των κόμβων των αξόνων από μια άλλη δομή, την Axenodes, η οποία επίσης μένει αμετάβλητη.

Πριν εξηγηθεί εν συντομία ο τρόπος δημιουργίας των γραφικών, αναφέρονται οι κυριότερες λειτουργίες της κλάσης:

drawStructure	Ο κορμός πάνω στον οποίο βασίζονται όλα τα υπόλοιπα. Κάθε φορά που συμβαίνει οποιαδήποτε αλλαγή στα δεδομένα του φορέα (κόμβοι - μέλη), η εικόνα που βλέπει ο χρήστης δημιουργείται από την αρχή με την συγκεκριμένη λειτουργία.
rotateStructure	Καλείται όταν ζητείται περιστροφή της κατασκευής γύρω από κάποιον άξονα.
wheelEvent	Αποτελεί μια λειτουργία η οποία αναζητά γεγονότα που συμβαίνουν εντός του παραθύρου των γραφικών (π.χ κινήσεις του ποντικιού, πάτημα πλήκτρων, κίνηση του τροχού του ποντικιού κτλ) και αντιδρά αναλόγως. Στο παρόν το γεγονός που ενδιαφέρει είναι η κίνηση του τροχού στο ποντίκι και η αντίδραση (η λειτουργία που εκτελείται) είναι η μεγέθυνση ή σμίκρυνση της εικόνας.
rotateZ	Λειτουργίες που έχουν σκοπό την περιστροφή περί των αξόνων z και x, αντίστοιχα.
rotateX	
convert3Dto2D	Η λειτουργία που μεταφράζει τα σημεία (κόμβους) τριών διαστάσεων που έχουμε διαθέσιμα, σε δύο διαστάσεις ανάλογα με την οπτική γωνία που επιθυμούμε να χρησιμοποιήσουμε. Υπάρχει μια γωνία που αποτελεί την προεπιλογή και είναι αυτή με την οποία ξεκινάει η προβολή πριν οποιαδήποτε περιστροφή εκτελέσει ο χρήστης.

Κάθε φορά που καλείται η drawStructure, πρώτα απ' όλα αδειάζει το παράθυρο από τυχόν γραφικά προηγούμενης εικόνας. Στη συνέχεια εντοπίζονται όλα τα μέλη του φορέα (από την GraphBeams) και με τη χρήση της convert3Dto2D εισάγονται στο παράθυρο. Συγχρόνως, το κάθε μέλος, εισάγεται ως γραφικό αντικείμενο στην GRAPHBEAMS, ώστε να χρησιμοποιηθεί αργότερα για οποιοδήποτε μετασχηματισμό. Όταν ο χρήστης γυρίζει τον τροχό του ποντικιού,

ανιχνεύεται το αντίστοιχο γεγονός, το λαμβάνει η `wheelEvent` και εκτελεί μια εσωτερική λειτουργία της γραφικής παρουσίασης, την `setScale`, μεγεθύνει, δηλαδή την εικόνα ή την σμικρύνει. Παράλληλα, επιβάλλονται κατάλληλες αλλαγές σε όλα τα αντικείμενα που έχουν αποθηκευτεί στην `GRAPHBEAMS`, ώστε τα πάχη των γραμμών να μην μεγαλώνουν ή μικραίνουν, ενώ στους άξονες της `GRAPHAXES`, να μην αλλάζουν καθόλου μέγεθος.

Η `rotateStructure` δεν είναι παρά μια υλοποίηση της `drawStructure` με λιγότερους υπολογισμούς, οι οποίοι δεν πρέπει να εκτελούνται κατά την περιστροφή. Η περιστροφή εκτελείται είτε με τη χρήση των πλήκτρων στη βάση του παραθύρου, που δίνουν συγκεκριμένες οπτικές γωνίες, είτε με τα πλήκτρα `a`, `s`, για περιστροφή περί τον άξονα `x` και `z`, `x` περί τον άξονα `z`. Για τον σκοπό αυτό χρησιμοποιείται η `KeyPressEvent`, που αποτελεί λειτουργία εντοπισμού συμβάντων στο γραφικό παράθυρο. Εδώ τα συμβάντα που μας ενδιαφέρουν είναι η πίεση των παραπάνω πλήκτρων στο πληκτρολόγιο, με αποτέλεσμα την εκτέλεση της προαναφερθείσας λειτουργίας.

#### 3.2.5.5 Η κλάση `AnalysisWindow`

Η δομή της `AnalysisWindow`, παραπέμπει στο κεντρικό παράθυρο, αποτελεί δηλαδή επίσης ένα ξεχωριστό παράθυρο με διάφορα αντικείμενα, τα οποία δημιουργούνται μέσα σε μια λειτουργία που ονομάζεται `init()`, και ένα σύνολο λειτουργιών. Η διαφορά είναι ότι βασίζεται σε μια κλάση αντικειμένων του `PyQt5` που ονομάζεται `QDialog` και έχει παρόμοια χρήση με την `QMainWindow`, αλλά με περιορισμένες δυνατότητες.

Εδώ, οι βασικές λειτουργίες είναι η `analysis`, που καλεί το αντίστοιχο αρχείο να εκτελέσει τους υπολογισμούς, η `saveResults`, η οποία αποθηκεύει τα αποτελέσματα που λαμβάνει από το προηγούμενο αρχείο και η `saveProject`, που ουσιαστικά δεν διαφέρει από την ομώνυμη του κεντρικού παραθύρου και εκτελεί την αποθήκευση της εργασίας προτού ξεκινήσει η ανάλυση.

### 3.2.5.6 Η κλάση ViewResultsWindow

Σε ότι αφορά τη δομή και κατασκευή, ισχύει ό,τι και στην προηγούμενη κλάση.

Οι βασικότερες λειτουργίες είναι η `readFile`, που διαβάζει τα αποθηκευμένα αρχεία αποτελεσμάτων, και η `viewResults`, που δημιουργεί τα γραφήματα των εσωτερικών εντάσεων.

### 3.2.5.7 Η λειτουργία ‘main’

```
4874
4875 def main():
4876     import sys
4877     global MAINWINDOW
4878     MAINWINDOW = QtCore.QCoreApplication.instance()
4879     if MAINWINDOW is None:
4880         MAINWINDOW = QtWidgets.QApplication(sys.argv)
4881     MAINWINDOW.aboutToQuit.connect(MAINWINDOW.deleteLater)
4882     splash_pix = QtGui.QPixmap('pyrforos_bw.png')
4883     splash = QtWidgets.QSplashScreen(splash_pix, QtCore.Qt.WindowStaysOnTopHint)
4884     splash.setMask(splash_pix.mask())
4885     splash.show()
4886     MAINWINDOW.processEvents()
4887     m=MainWindow()
4888     m.show()
4889     splash.hide()
4890     splash.finish(m)
4891
4892     sys.exit(MAINWINDOW.exec_())
4893
4894 main()
4895
4896
```

Στη λειτουργία ‘main’ που είναι η τελευταία του αρχείου και η οποία είναι εκείνη που ουσιαστικά ενεργοποιεί το ‘Mainwindow’, προκειμένου να ανοίξει το κεντρικό παράθυρο, φαίνονται τα εξής:

α. Στις γραμμές 4877 – 4881 και 4892, μια διαδικασία, που εξασφαλίζει ότι το πρόγραμμα θα τερματίζεται σωστά χωρίς να δημιουργούνται σφάλματα έως την επόμενη εκκίνησή του.

β. Στις γραμμές 4882 – 4885, μια σειρά εντολών για την εμφάνιση της αρχικής εικόνας (εδώ ο ‘Πυρφόρος’), είναι προαιρετικές και περισσότερο βελτίωσης της εμφάνισης.

γ. Στις γραμμές 4887 – 4888, ο ορισμός της μεταβλητής του ‘MainWindow’, ώστε να δημιουργηθεί ένα αντικείμενο από την ομώνυμη κλάση

(δηλαδή το κεντρικό παράθυρο) και στη συνέχεια να εμφανιστεί στην οθόνη του υπολογιστή.

δ. Στην γραμμή 4894 καλείται η 'main'.

Συνοψίζοντας, αφού γίνει ανάγνωση ολόκληρου του αρχείου, μόνο στην τελευταία γραμμή ('main') καλείται η εκτέλεση του προγράμματος.

### 3.2.6 Το αρχείο setup.py

Η λειτουργία του περιορίζεται στο να δημιουργήσει το εκτελέσιμο αρχείο για το περιβάλλον των Windows. Όπως αναφέρεται στη βιβλιογραφία της συγκεκριμένης βιβλιοθήκης, τα απαιτούμενα modules (αυτόνομα τμήματα της κάθε χρησιμοποιούμενης βιβλιοθήκης – π.χ numpy, pandas, κτλ) αναγνωρίζονται και ενσωματώνονται αυτόματα στο δημιουργούμενο αρχείο. Παρ'όλ'αυτά, εντοπίστηκε μια αδυναμία του cxFreeze στο να συμπεριλαμβάνει κάποια αρχεία (κυρίως της μορφής .dll). Έτσι, στο 'build\_exe\_options' του setup.py περιλαμβάνονται αναλυτικά τόσο τα αρχεία που αδυνατεί να εντοπίσει από μόνο του το cxFreeze, όσο και αυτά που ο χρήστης θεωρεί απαραίτητα ώστε ένα λειτουργεί σωστά το πρόγραμμα. Επιπλέον, όπως έγινε και στην παρούσα εργασία, ορισμένα αρχεία που δεν ήταν δυνατόν να συμπεριληφθούν ακόμα και με την παραπάνω διαδικασία, τοποθετήθηκαν εκ των υστέρων στον δημιουργηθέντα φάκελο. Επίσης, όπως φαίνεται παρακάτω, στο 'setup' μπορούν να δοθούν το επιθυμητό όνομα, η έκδοση, μια περιγραφή του αρχείου, καθώς και οι επιλογές που έχουν καθοριστεί προηγουμένως (στο 'build\_exe\_options')

```
setup(  
    name = "GUInew",  
    version = "1.01",  
    description = "GUInew",  
    options = {"build_exe": build_exe_options},  
    executables = [Executable("GUInew.py", base = base)]) # "Win32GUI"
```

## Κεφάλαιο 4: Χρήση του προγράμματος

### 4.1 Εγκατάσταση

Δεν απαιτείται εγκατάσταση, παρά μόνο τοποθέτηση του φακέλου του προγράμματος σε οποιαδήποτε θέση στον υπολογιστή. Αυτό που είναι σημαντικό για τη σωστή λειτουργία είναι πως ό,τι διαβάζει το πρόγραμμα, το διαβάζει μέσα στον φάκελο στον οποίο βρίσκεται το αρχείο exe. Επομένως δεν είναι δυνατόν να λειτουργήσει αν αφαιρεθεί κάποιο αρχείο ή αλλάξει θέση εσωτερικά στον φάκελο (π.χ. λόγω ομαδοποίησης αρχείων και τοποθέτησής τους σε νέο φάκελο ακόμη και μέσα στον αρχικό).

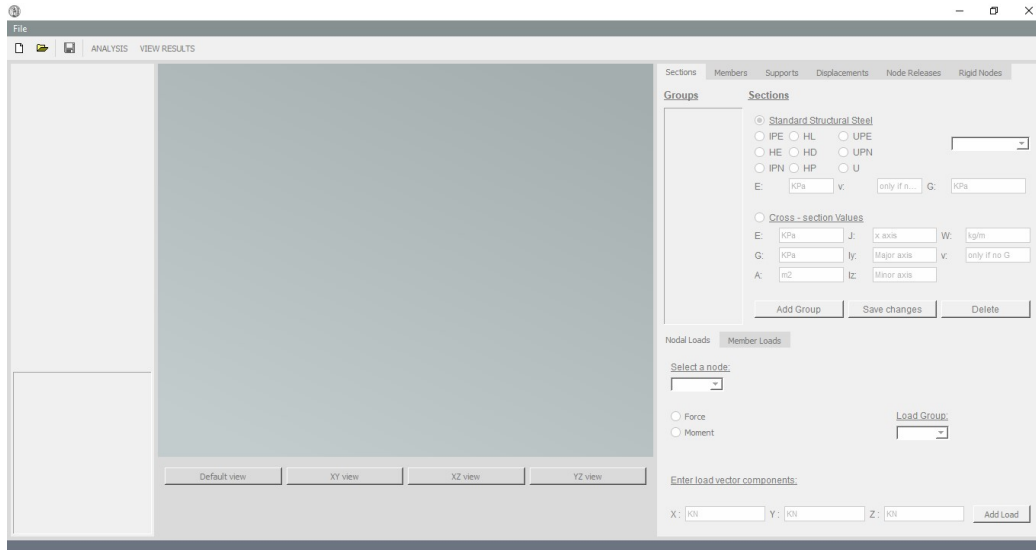
Σε ότι αφορά τις αποθηκευμένες εργασίες, αυτές μπορούν να αποθηκευτούν όπου επιθυμεί ο χρήστης. Αντιθέτως, τα αρχεία των αναλύσεων το πρόγραμμα μπορεί να τα εντοπίσει μόνο αν είναι στον αρχικό φάκελο, όπου και αποθηκεύονται αυτόματα, και μόνο αν δεν έχει αλλαχθεί η ονομασία τους. Ο λόγος είναι ότι τα συγκεκριμένα ονομάζονται αυτόματα από το πρόγραμμα με το όνομα του αρχείου εργασίας (π.χ. *iliasKeskinisBeam.txt*), με την πρόσθεση του *'\_results'* και ενός αριθμού (*iliasKeskinisBeam\_results1.txt*). Βέβαια, το γεγονός ότι αποτελούν αρχεία *txt* δίνει τη δυνατότητα στον χρήστη να τα ανοίγει σε όποιο σημείο του υπολογιστή βρίσκονται, με τον περιορισμό ότι δεν μπορεί να δει τα διαγράμματα των δράσεων.



Αν το πρόγραμμα τρέχει σε περιβάλλον Python, οι απαιτήσεις (π.χ. βιβλιοθήκες) είναι αυτές που έχουν περιγραφεί σε προηγούμενη παράγραφο, με όλους τους παραπάνω περιορισμούς σε ισχύ.

## 4.2 Έναρξη του προγράμματος

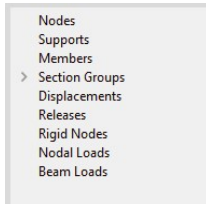
### 4.2.1 Γενικά

Η έναρξη του προγράμματος γίνεται με διπλό κλικ πάνω στο εικονίδιο της εφαρμογής που βρίσκεται στον φάκελο. Κατά την έναρξη του προγράμματος ο χρήστης βλέπει την παρακάτω οθόνη,

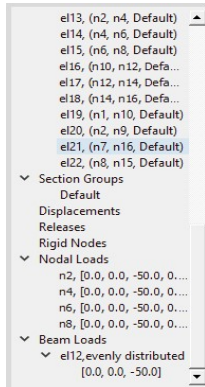


η οποία έχει όλες τις λειτουργίες απενεργοποιημένες. Οι μοναδικές επιλογές στην παρούσα κατάσταση είναι η δημιουργία νέου έργου με το εικονίδιο  ή άνοιγμα κάποιας αποθηκευμένης εργασίας από το εικονίδιο . Οι ίδιες, αντίστοιχα, λειτουργίες μπορούν να εκτελεστούν και από το μενού File με τις Open και New.

Στο αριστερό του παραθύρου βρίσκεται η λίστα με όλα τα στοιχεία του μοντέλου του φορέα. Αρχικά είναι κενή, αλλά με την επιλογή νέας ή αποθηκευμένης εργασίας δημιουργείται όπως στην εικόνα



Φαίνονται οι κόμβοι, οι στηρίξεις, τα μέλη, οι διατομές, οι μετατοπίσεις – στροφές των κόμβων, οι εσωτερικές ελευθερώσεις των μελών, οι στερεοί κόμβοι και τα φορτία (επικόμβια και μελών)

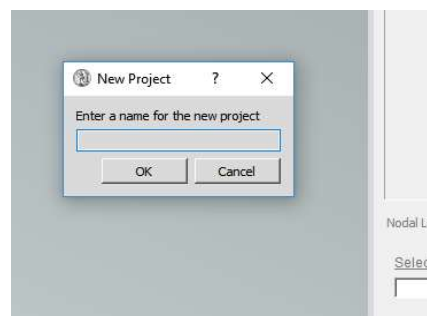


Όταν υπάρχουν δεδομένα για τον φορέα, τα παραπάνω μπορούν να αναπτυχθούν για να μπορεί να τα δει ο χρήστης πιο αναλυτικά.

Στο μέσο βρίσκεται το παράθυρο προβολής του φορέα σε τρεις διαστάσεις, στο δεξιό άνω οι καρτέλες εισαγωγής των δεδομένων και κάτω οι καρτέλες εισαγωγής φορτίων. Επίσης, κάτω από την κεντρική λίστα υπάρχει μια δεύτερη που δίνει μια σύνοψη των λεπτομερειών του κάθε δεδομένου που θα επιλέξει ο χρήστης από την πρώτη.

#### 4.2.2 Δημιουργία νέας εργασίας

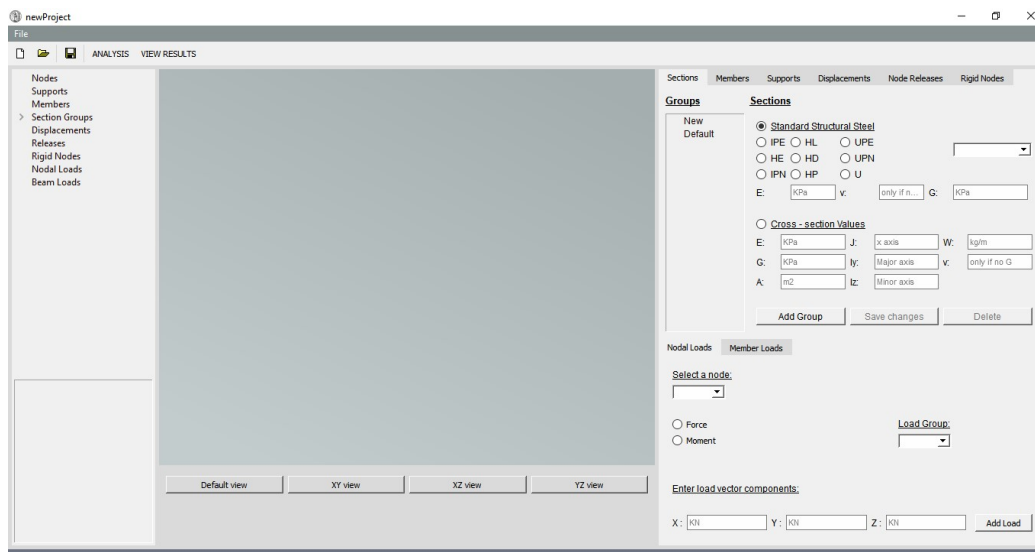
Με το πάτημα του αντίστοιχου πλήκτρου εμφανίζεται το παράθυρο:



το οποίο απαιτεί την εισαγωγή ονόματος για το νέο έργο. Αν ο χρήστης δεν εισάγει όνομα, το όνομα περιέχει μη αποδεκτούς χαρακτήρες ή αλληλουχίες χαρακτήρων, το πρόγραμμα δίνει ανάλογη ειδοποίηση στον χρήστη, όπως φαίνονται στον πίνακα:

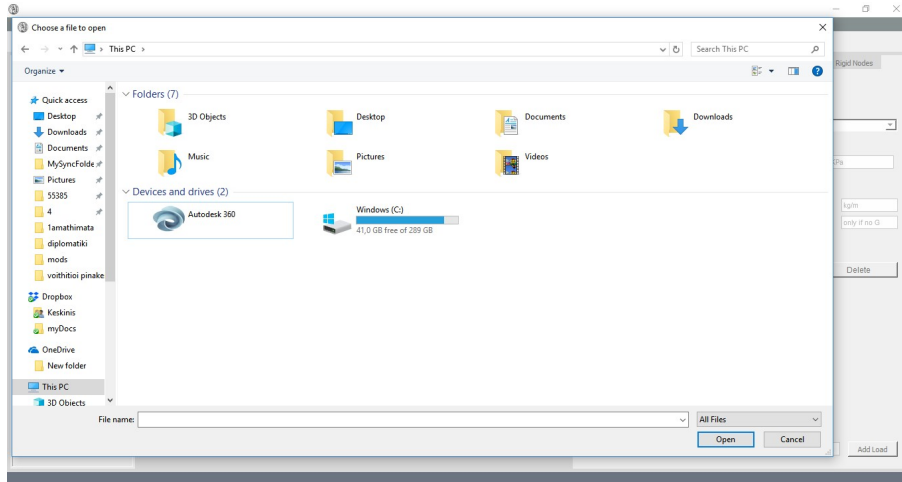
Όνομα εργασίας	Αιτία απόρριψης	Μήνυμα ειδοποίησης
Περιέχει ένα ή περισσότερα από τα <, >, :, /, \, ?, *,	Μη αποδεκτός χαρακτήρας	The name contains one or more invalid characters
Είναι ένα από τα CON, PRN, AUX, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9	Μη αποδεκτό όνομα	This is an invalid name
Ο τελευταίος χαρακτήρας είναι το κενό ή το ‘.’	Μη αποδεκτός τελευταίος χαρακτήρας	The last character is invalid

Αν το όνομα γίνει δεκτό (π.χ. newProject) ενεργοποιούνται όλες οι λειτουργίες και καρτέλες, το παράθυρο παίρνει την ονομασία της εργασίας και εμφανίζεται στο αριστερό η λίστα με όλα τα στοιχεία του φορέα (προς το παρόν είναι κενή και υπάρχει μόνο το όνομα μιας ομάδας διατομών ‘Default’ που χρησιμοποιείται για την έγκυρη εισαγωγή μελών από τον χρήστη, αν αυτός προηγουμένως δεν έχει ορίσει κάποια δικιά του).



### 4.2.3 Άνοιγμα αποθηκευμένης εργασίας

Με το πάτημα του αντίστοιχου εικονιδίου, εμφανίζεται το παράθυρο:



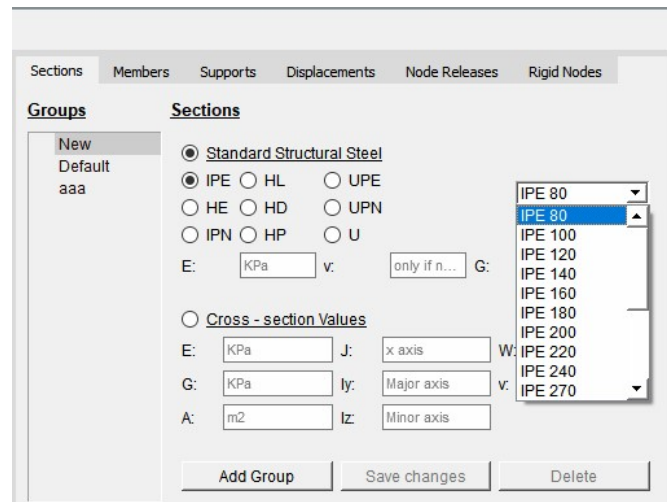
μέσω του οποίου ο χρήστης επιλέγει την εργασία που θέλει. Όταν την επιλέξει για άνοιγμα, το πρόγραμμα θα εμφανίσει όλα τα δεδομένα ακριβώς όπως αποθηκεύτηκαν.

## 4.3 Εισαγωγή δεδομένων

### 4.3.1 Εισαγωγή διατομών

Group	Section Type	E	J	W	G	Iy	v	A	Iz
aaa	Cross - section Values	30000000.0	0.00179	0.0	12500000.0	0.0026	0.0	0.125	0.000651

Στην καρτέλα των διατομών ο χρήστης μπορεί να επιλέξει μεταξύ διαθέσιμων πρότυπων διατομών, καθώς και να δημιουργήσει δικές του. Η επιλογή από τις πρότυπες προϋποθέτει επιλογή του στοιχείου 'New' στη λίστα (διαφορετικά θα εμφανίζεται η εκάστοτε επιλεγμένη διατομή), στη συνέχεια του 'Standard Structural Steel', έπειτα τον τύπο της διατομής και τέλος, την διάσταση όπως στην εικόνα:



Επιπλέον, χρειάζεται να καθορίσει τις τιμές του μέτρου ελαστικότητας, του λόγου Poisson και/ή του μέτρου διάτμησης.

Στην περίπτωση των διαμορφωμένων διατομών χρειάζεται ομοίως η επιλογή του στοιχείου 'New', μετά του 'Cross-section Values' και τέλος όλες οι τιμές που θα εισαχθούν από τον χρήστη.

Όταν συμπληρωθούν τα απαραίτητα δεδομένα, που για τον σκοπό αυτό έχει δημιουργηθεί μια σειρά ειδοποιήσεων που αποτρέπουν, κατά το δυνατόν, τον χρήστη να ορίσει εσφαλμένες τιμές, το πρόγραμμα ζητάει τον καθορισμό ονόματος για την διατομή (εδώ για παράδειγμα έχει οριστεί μια διατομή διαμορφωμένη με ονομασία 'aaa'). Με την επιτυχή εισαγωγή ονόματος, δημιουργείται η διατομή και εμφανίζεται άμεσα στη λίστα.

Είτε μια διατομή έχει εισαχθεί ως πρότυπη, είτε ως διαμορφωμένη, όταν την επιλέξει ο χρήστης από τη λίστα, εμφανίζονται όλες οι τιμές της (στην περίπτωση των προτύπων οι τιμές λαμβάνονται από τους πίνακες του προγράμματος).

Με μια διατομή επιλεγμένη, στη συνέχεια αλλαγή των χαρακτηριστικών της και επιλογή του 'Modify', αποθηκεύονται οι αλλαγές. Εδώ χρειάζεται προσοχή, διότι η αλλαγή των χαρακτηριστικών της διατομής με την προαναφερθείσα ενέργεια δεν συνεπάγεται και αποθήκευση της αλλαγής στο αρχείο της εργασίας.

Η διαγραφή διατομών μπορεί να γίνει είτε μεμονωμένα με δεξί κλικ στη λίστα, είτε επιλέγοντας μία ή περισσότερες διατομές, με χρήση του Ctrl, και πάτημα του κουμπιού 'Delete'.

#### 4.3.2 Εισαγωγή μελών

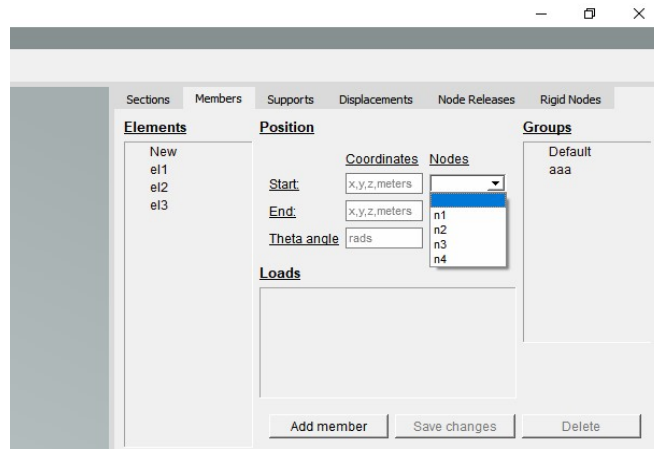
The screenshot shows a software interface with several tabs: Sections, Members, Supports, Displacements, Node Releases, and Rigid Nodes. The 'Members' tab is selected. On the left, under 'Elements', there is a list with 'New'. The main area is divided into 'Position' and 'Groups'. Under 'Position', there are two sub-sections: 'Coordinates' and 'Nodes'. The 'Coordinates' section has 'Start' and 'End' fields, both containing 'x,y,z,meters', and a 'Theta angle' field containing 'rads'. The 'Nodes' section has two dropdown menus. The 'Groups' section has a 'Default' dropdown. At the bottom, there are three buttons: 'Add member', 'Save changes', and 'Delete'.

Στην καρτέλα εισαγωγής μελών ο χρήστης καλείται να εισάγει τα μέλη του φορέα με συντεταγμένες αρχής – πέρατος, στις θέσεις της στήλης 'coordinates'. Οι συντεταγμένες πρέπει να είναι οριοθετημένες με κόμματα (50,10,30) και δεν πρέπει να περιέχουν άλλους χαρακτήρες. Γενικά, έχει καλυφθεί ένα μεγάλο μέρος, αν όχι όλων, των σφαλμάτων που ενδέχεται να κάνει ο χρήστης και τα οποία επισημαίνονται με κατάλληλες ειδοποιήσεις.

Τα μέλη ονομάζονται e11, e12, e13 ενώ οι κόμβοι n1, n2, n3 κτλ.

Όταν έχει εισάγει τουλάχιστον ένα μέλος, θα μπορεί να δει στα κουτιά της στήλης 'nodes' τους υπάρχοντες κόμβους και αν επιθυμεί κάποιο νέο μέλος να

περιέχει έναν από αυτούς, να τον επιλέξει χωρίς να χρειάζεται να εισάγει εκ νέου συντεταγμένες.

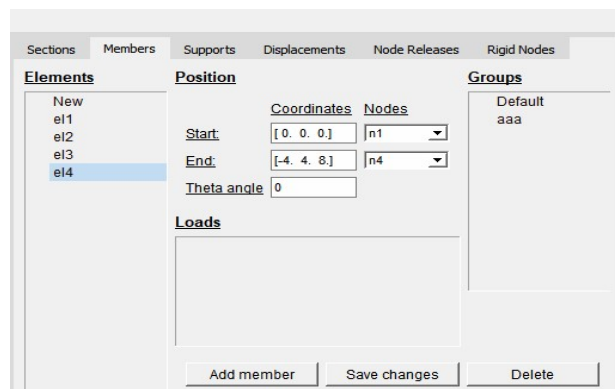


Επίσης, θα πρέπει να εισαχθεί και γωνία στροφής του μέλους περί τον άξονα x (αν υπάρχει). Η θετική γωνία θεωρείται η δεξιόστροφη.

Τέλος, ο χρήστης έχει επιλογή μεταξύ των ομάδων διατομών που έχει αποθηκεύσει από την προηγούμενη καρτέλα. Όποια διατομή είναι επιλεγμένη στη λίστα στα δεξιά, είναι αυτή που αποκτά το μέλος. Σημειώνεται, ότι κατά την έναρξη η προεπιλογή είναι η 'Default', η οποία έχει μηδενικές τιμές.

Όταν συμπληρωθούν όλα τα δεδομένα, η πίεση του 'Add member' θα δημιουργήσει ένα νέο μέλος, που θα εμφανιστεί στη λίστα στα δεξιά.

Αν ο χρήστης επιλέξει ένα από τα μέλη της λίστας, θα εμφανιστούν οι συντεταγμένες, οι κόμβοι αρχής-τέλους, η γωνία και τα φορτία του στις αντίστοιχες θέσεις:



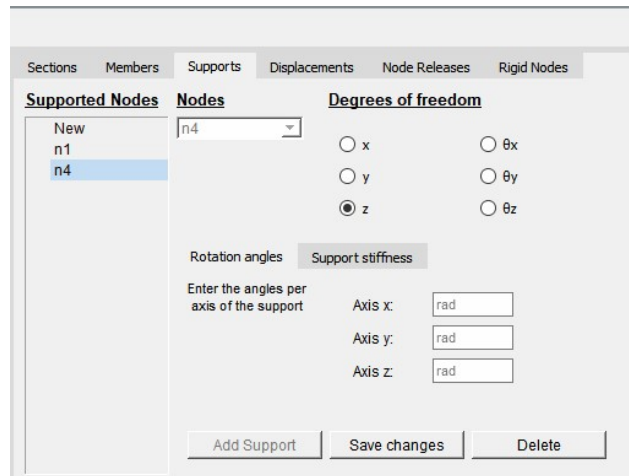
και αν χρειαστεί αλλαγές στα δεδομένα του, συμπληρώνει τις νέες τιμές και πιέζει το 'Save changes'. Το τελευταίο ισχύει μόνο για την γωνία και τη διατομή. Δεν είναι δυνατόν να αλλαχθούν οι κόμβοι αρχής ή πέρατος και πρέπει να διαγραφεί το μέλος με το 'Delete' και να δημιουργηθεί εκ νέου. Αυτή η ενέργεια θα διαγράψει οτιδήποτε έχει προσθέσει στο μέλος (φορτία, ελευθερώσεις κτλ) οπότε είναι σημαντικό να ελέγξει την ορθότητα των μελών προκειμένου να αποφύγει επανάληψη ενεργειών. Επίσης, η διαγραφή ενός μέλους ή η αλλαγή διατομής του, μπορεί να γίνει με το δεξί πλήκτρο του ποντικιού (αφού έχει επιλεγεί η επιθυμητή νέα διατομή). Για ενέργειες σε περισσότερα μέλη μπορεί, αφού τα επιλέξει όλα με ταυτόχρονο πάτημα του Ctrl, να επιλέξει διαγραφή ή αλλαγή (αλλαγή σε περισσότερα του ενός μελών μπορεί να γίνει μόνο για διατομές).

#### 4.3.3 Εισαγωγή στηρίξεων

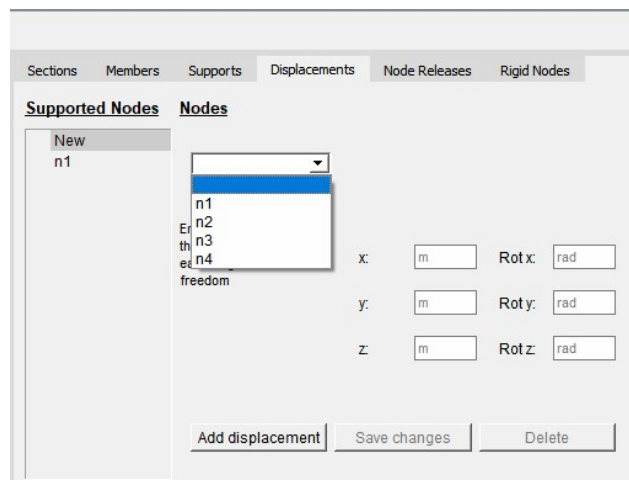
Στην καρτέλα των στηρίξεων, ο χρήστης αρχικά πρέπει να επιλέξει τον κόμβο στον οποίο θα επιβάλλει περιορισμούς. Οι κόμβοι εμφανίζονται στην ανοιγόμενη λίστα. Στην συνέχεια επιλέγει τους αντίστοιχους βαθμούς ελευθερίας που επιθυμεί να αποτελούν μέρη της στήριξης. Έτσι μια πλήρης πάκτωση θα έχει επιλεγμένους όλους τους βαθμούς ελευθερίας.

Επίσης, από την καρτέλα 'Rotation angles' μπορεί να εισάγει γωνίες για περιπτώσεις κεκλιμένων στηρίξεων (θετικές γωνίες θεωρούνται οι δεξιόστροφες), σε rad. Τέλος, στο 'Support stiffness' μπορεί να ορίσει σταθερές ελατηρίων για ελαστικές στηρίξεις σε KN/m.

Όταν εισάγει όλα τα δεδομένα, επιλέγει το 'Add Support'. Η χρήση των κουμπιών και της λίστας στηρίξεων είναι όμοια με τις προηγούμενες καρτέλες, σε ότι αφορά την εμφάνιση των δεδομένων μιας επιλεγμένης στήριξης, αλλαγές δεδομένων, διαγραφή κτλ.



#### 4.3.4 Εισαγωγή μετατοπίσεων και στροφών κόμβων



Προκειμένου να οριστούν μετατοπίσεις και στροφές θα πρέπει να επιλεγεί το στοιχείο 'New' από τη λίστα, έπειτα κάποιος κόμβος και στη συνέχεια να συμπληρωθούν οι τιμές στις αντίστοιχες θέσεις.

Σε ότι αφορά τη χρήση και τις λειτουργίες της λίστας των κόμβων και των κουμπιών, ισχύει ότι και στα προηγούμενα.

Οι μονάδες που χρησιμοποιούνται είναι μέτρα και ακτίνια, όπως και οι αντίστοιχες ενδείξεις. Και εδώ, όπως και σε όλες τις καρτέλες, έχουν ενσωματωθεί λειτουργίες προκειμένου να αποφευχθούν, κατά το δυνατόν, σφάλματα στην εισαγωγή δεδομένων.

#### 4.3.5 Εισαγωγή εσωτερικών ελευθερώσεων μελών

The screenshot shows the 'Node Releases' dialog box. The 'Elements' dropdown is open, showing 'el1', 'el2', 'el3', and 'el4'. The 'Start Node' and 'End Node' sections each have three rows of input fields for 'x', 'y', and 'z' coordinates, and 'Rot x', 'Rot y', and 'Rot z' rotation values, all currently set to 'rad'. The 'Add Release', 'Save changes', and 'Delete' buttons are visible at the bottom.

Στην παρούσα καρτέλα ο χρήστης μπορεί να εφαρμόσει ελευθερώσεις στα άκρα των μελών του φορέα. Αφού επιλέξει ένα μέλος, εισάγει τιμές γωνιών στην αρχή ή στο πέρας του. Οι γωνίες αφορούν στο τοπικό σύστημα συντεταγμένων του κάθε μέλους και είναι δεξιόστροφες. Επισημαίνεται ότι αν ένα μέλος έχει μια ελευθέρωση που στο τοπικό σύστημα δεν σχηματίζει γωνία με τη διεύθυνση του μέλους, εισάγεται η τιμή 0 ενώ οι υπόλοιπες θέσει μένουν κενές. Εισάγεται δηλαδή κάποια τιμή γωνίας (μηδενική ή μη) μόνο κατά για τους βαθμούς ελευθερίας εκείνους, όπου υπάρχει ελευθέρωση.

Σε ότι αφορά τη χρήση και τις λειτουργίες της λίστας των μελών και των κουμπιών, ισχύει ότι και στα προηγούμενα.

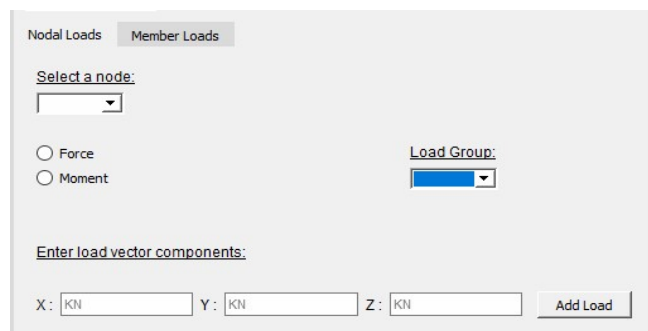
#### 4.3.6 Ορισμός στερεών κόμβων

The screenshot shows the 'Node Releases' dialog box with the 'Rigid Nodes' tab selected. The 'Elements' dropdown is open, showing 'el1', 'el2', 'el3', and 'el4'. Below the dropdown, there are 'Start' and 'End' labels. Underneath, there are two columns of input fields for 'Dx', 'Dy', and 'Dz' values. The 'Add', 'Save changes', and 'Delete' buttons are visible at the bottom.

Ο χρήστης είναι δυνατόν να ορίσει στερεούς κόμβους από την τελευταία καρτέλα. Η χρήση της λίστας των μελών και των κουμπιών είναι όμοια με τις προηγούμενες καρτέλες, ενώ για την εισαγωγή δεδομένων πρέπει αρχικά να επιλεγεί κάποιο στοιχείο και στη συνέχεια να οριστούν τιμές για την αρχή ή το πέρας του. Όλες οι τιμές αφορούν στο τοπικό σύστημα του μέλους.

Λόγω μη πλήρους ανάπτυξης του προγράμματος, η συγκεκριμένη λειτουργία θα δώσει ορθά αποτελέσματα μόνο στις περιπτώσεις στερεών κόμβων όπου καταλήγει μόνο ένα μέλος του φορέα (π.χ. σε πέδιλο). Στις υπόλοιπες περιπτώσεις προτείνεται μια πιθανώς πιο προσεγγιστική μοντελοποίηση. Το τμήμα του μέλους (ή των μελών) που συνδέονται στον στερεό κόμβο να αντιμετωπίζονται ως ξεχωριστά μέλη, τα οποία να διαθέτουν, σε αναλογία προς τη διατομή του κάθε μέλους, μεγάλες τιμές μέτρου ελαστικότητας, ροπών αδρανείας και επιφάνειας διατομής (το ίδιο βάρος τους θα παραμένει το ίδιο με του μέλους). Τα τμήματα αυτά εκ των πραγμάτων θα έχουν επικουρικό χαρακτήρα και δε θα χρειάζονται περαιτέρω αξιολόγηση, όσον αφορά τις εσωτερικές εντάσεις τους.

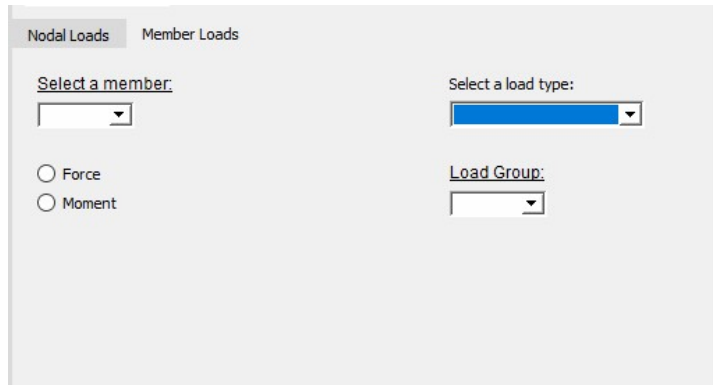
#### 4.3.7 Εισαγωγή επικόμβιων φορτίων



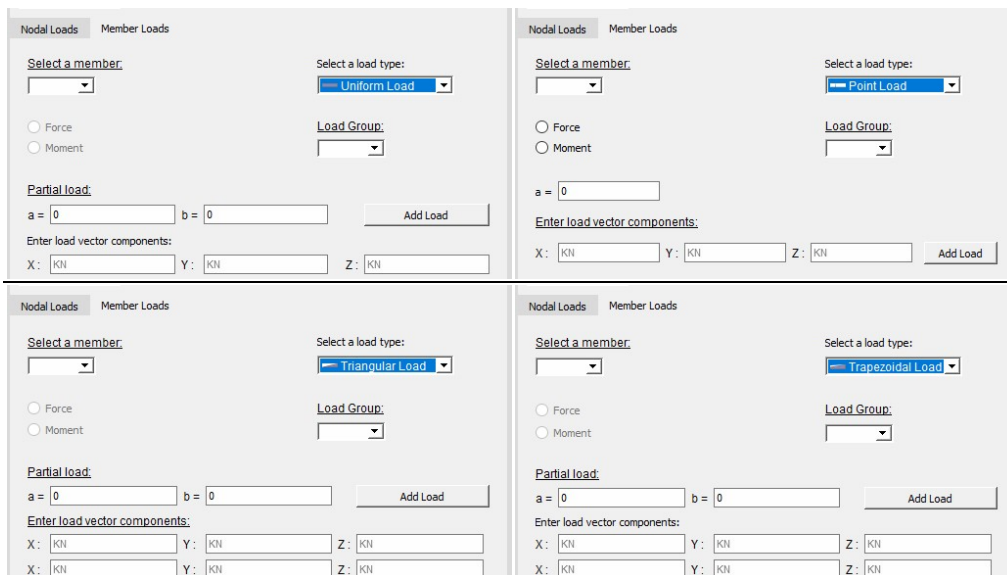
Η εισαγωγή φορτίων εκτελείται από το πλαίσιο καρτελών στο κάτω δεξιό μέρος του παραθύρου με πρώτη καρτέλα, αυτήν των επικόμβιων. Ο χρήστης καλείται να επιλέξει κόμβο, αν το φορτίο είναι δύναμη ή ροπή, μεταξύ μόνιμων και κινητών φορτίων και τελικά να καθορίσει το διάνυσμα του φορτίου. Το διάνυσμα θα δίνεται στο καθολικό σύστημα. Με το πάτημα του 'Add Load' το φορτίο εισάγεται και εμφανίζεται στην κεντρική λίστα στο αριστερό του παραθύρου όπου βρίσκονται όλα τα στοιχεία του μοντέλου του φορέα. Εδώ δεν μπορεί να γίνει αλλαγή των

δεδομένων του κάθε φορτίου, παρά μόνο διαγραφή και επαναπροσδιορισμός. Η διαγραφή γίνεται από την συγκεντρωτική λίστα με το δεξί πλήκτρο του ποντικιού και επιλογή 'Delete'.

#### 4.3.8 Εισαγωγή φορτίων μελών



#### Η αρχική καρτέλα



Η καρτέλα των φορτίων των μελών, όπως μεταβάλλεται, ανάλογα με τον τύπο φορτίου που επιλέγεται

Η καρτέλα φορτίων των μελών διαθέτει μια ανοιγόμενη λίστα όπου επιλέγεται κάποιο μέλος, μια λίστα με τους διαθέσιμους τύπους φορτίων (σημειακό και ομοιόμορφο, τριγωνικά ή τραπεζοειδή κατανεμημένα) και, όπως και στην προηγούμενη, επιλογή μεταξύ μονίμων ή κινητών. Με την επιλογή κάποιου τύπου φορτίου, μεταβάλλονται οι υπόλοιπες λειτουργίες εισαγωγής δεδομένων όπως παρακάτω:

α. Σημειακό φορτίο

Ο χρήστης επιλέγει μεταξύ δύναμης και ροπής, καθορίζει την απόσταση του φορτίου από την αρχή του μέλους καθώς και το διάνυσμα του φορτίου, ακριβώς όπως στα επικόμβια. Είναι σημαντικό να δίνεται προσοχή στο ποια είναι η αρχή και το πέρας του κάθε μέλους (φαίνεται στην κεντρική λίστα του παραθύρου) ώστε να αποφεύγονται σφάλματα.

β. Ομοιόμορφο κατανεμημένο φορτίο

Ορίζεται η απόσταση από τον κόμβο αρχής του μέλους, από την οποία ξεκινάει να δρα το φορτίο (τιμή  $a$ ) και η απόσταση από το σημείο που σταματάει το φορτίο μέχρι τον κόμβο πέρατος (τιμή  $b$ ). Επίσης, το μέγεθος του φορτίου εισάγεται ως διάνυσμα στο καθολικό σύστημα συντεταγμένων.

γ. Τριγωνικό κατανεμημένο φορτίο

Ομοίως ορίζονται οι αποστάσεις από τον κόμβο αρχής και έως τον κόμβο πέρατος όπως προηγουμένως, ενώ το μέγεθος του φορτίου καθορίζεται ως εξής: Αν το φορτίο έχει αύξουσα μορφή, το πρώτο διάνυσμα είναι μηδενικό, ενώ το δεύτερο είναι η τελική τιμή του. Αν έχει φθίνουσα μορφή, ισχύει το αντίθετο, ενώ και εδώ τα διανύσματα περιγράφονται στο καθολικό σύστημα.

δ. Τραπεζοειδές κατανεμημένο φορτίο

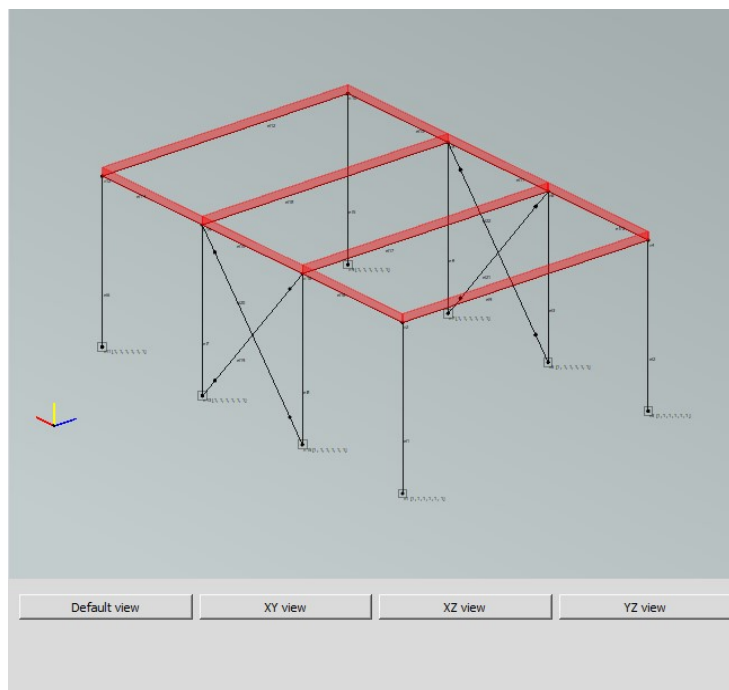
Ομοίως ορίζονται οι αποστάσεις από τον κόμβο αρχής και έως τον κόμβο πέρατος όπως προηγουμένως, ενώ το μέγεθος του φορτίου καθορίζεται από την αρχική και τελική τιμή του τραπεζίου, εκφρασμένες ως διανύσματα στο καθολικό σύστημα.

Τα φορτία, όταν εισαχθούν με το 'Add Load', θα εμφανιστούν άμεσα στην κεντρική λίστα του παραθύρου. Η αλλαγή των στοιχείων τους δεν είναι δυνατή, όπως και στα επικόμβια, ενώ η διαγραφή τους γίνεται όμοια, με το δεξί πλήκτρο του ποντικιού.

Υπενθυμίζεται ότι σε περίπτωση που διαγραφεί κάποιος κόμβος ή μέλος από το μοντέλο, αυτομάτως θα διαγραφούν και τα φορτία του.

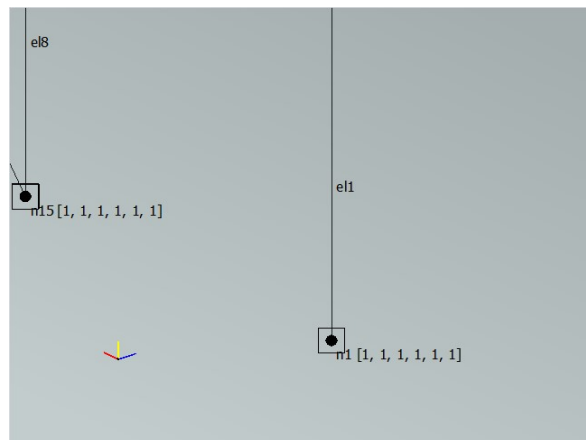
Σε ότι αφορά τα ίδια βάρη των μελών του φορέα, αυτά εφαρμόζονται αυτόματα με την εισαγωγή του κάθε μέλους και σημαίνονται ως τέτοια. Παρόλα αυτά, αν για οποιονδήποτε λόγο κατά τη φάση της ανάλυσης, ο χρήστης δεν επιθυμεί να τα συμπεριλάβει, μπορεί να το επιλέξει από το αντίστοιχο παράθυρο (η συγκεκριμένη επιλογή θα περιγραφεί στην αντίστοιχη παράγραφο).

#### 4.4. Προβολή του μοντέλου του φορέα

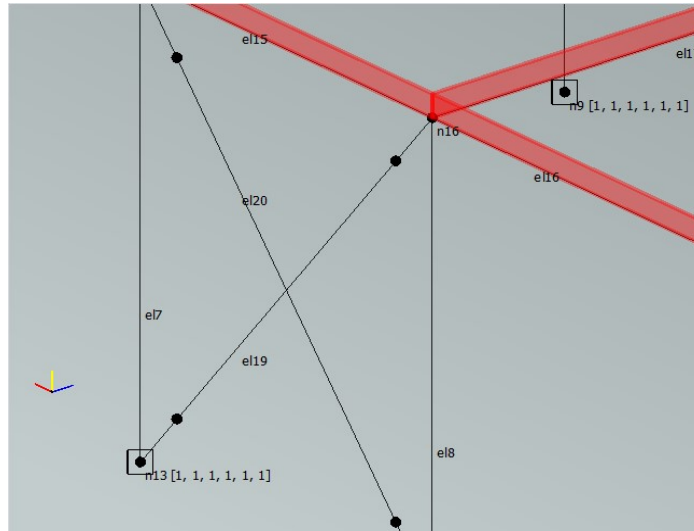


Το τμήμα του παραθύρου, στο οποίο προβάλλεται ο φορέας,, διαθέτει λειτουργίες περιστροφής και αλλαγής οπτικής γωνίας. Έτσι, με τη χρήση των πλήκτρων ‘z’ και ‘x’ από το πληκτρολόγιο, μπορεί να περιστραφεί περί τον άξονα z, ενώ με τη χρήση των ‘a’ και ‘s’, περί τον άξονα x. Ακόμη, καθώς η τοποθέτηση του φορέα σε συγκεκριμένη κλίση με τη χρήση των πλήκτρων δεν παρέχει ακρίβεια, υπάρχουν οι έτοιμες επιλογές ‘Default view’, για την τρισδιάστατη προβολή, όπως στην εικόνα, και ‘XY view’, ‘XZ view’ και ‘YZ view’ για την προβολή αποκλειστικά των αντίστοιχων επιπέδων.

Επίσης, στο κάτω αριστερό φαίνονται οι άξονες, μπλε ο x, κόκκινος ο y και κίτρινος ο z, οι οποίοι περιστρέφονται μαζί με το μοντέλο για καλύτερη κατανόηση της γεωμετρίας.



Οι πακτώσεις συμβολίζονται με ένα τετράγωνο, οι αρθρώσεις με τρίγωνο, οι κλίσεις, σε τουλάχιστον μία κατεύθυνση, με ένα ορθογώνιο με κύλιση και οι υπόλοιπες στηρίξεις που δεν εμπίπτουν στα παραπάνω με έναν κύκλο. Επίσης, σε όλους τους κόμβους και μέλη υπάρχουν οι ονομασίες τους, ενώ ειδικά για τις στηρίξεις υπάρχει και η κωδικοποίηση που χρησιμοποιείται στο πρόγραμμα (π.χ. [1,1,1,0,0,0] για άρθρωση, [1,1,1,1,1,1] για πάκτωση κτλ.).

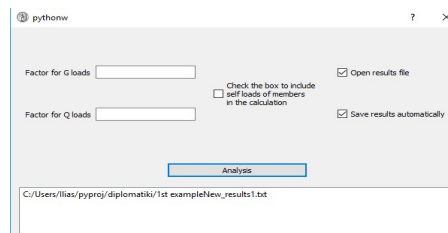


Ακόμη, σε περίπτωση που υπάρχει κάποια εσωτερική ελευθέρωση σε μέλη, όπως συμβαίνει εδώ με τους κατακόρυφους συνδέσμους δυσκαμψίας, αυτές σημειώνονται με έναν κύκλο που θυμίζει κόμβο και βρίσκεται λίγο μετά την αρχή του μέλους ή λίγο πριν το πέρας, ανάλογα με το που υπάρχει η αντίστοιχη ελευθέρωση. Τέλος, τα φορτία σημαίνονται με κόκκινο χρώμα.

Κατά την περιστροφή του φορέα, ο χρήστης μπορεί να δει τα πάντα πλην των φορτίων, τα οποία φαίνονται μόνο στην προεπιλεγμένη γωνία, που είναι αυτή της εικόνας.

#### 4.5. Ανάλυση μοντέλου φορέα

Όταν έχει ολοκληρωθεί η μόρφωση του μοντέλου του φορέα, ο χρήστης μπορεί να επιλέξει να εκτελεστεί ανάλυσή του. Με το πάτημα του πλήκτρου **ANALYSIS** ανοίγει το αντίστοιχο παράθυρο, όπως στην εικόνα:



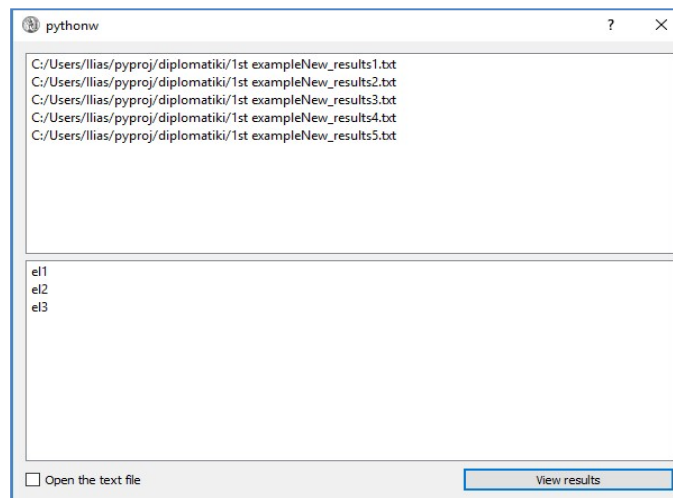
Εδώ, σε ότι αφορά την ανάλυση αυτή καθαυτή, υπάρχουν οι επιλογές των συντελεστών φορτίσεων για μόνιμα και κινητά φορτία και η συμπερίληψη ή όχι των



Όπως προαναφέρθηκε, προκειμένου τα αρχεία των αποτελεσμάτων να είναι ορατά τόσο στο παράθυρο της ανάλυσης, ως προηγούμενες αναλύσεις, όσο και στο παράθυρο επισκόπησης των αποτελεσμάτων (θα περιγραφεί στη συνέχεια) δεν θα πρέπει να μεταφέρονται σε άλλη θέση στον υπολογιστή, πέρα από αυτήν όπου δημιουργούνται από το πρόγραμμα, και δεν θα πρέπει να μετονομάζονται, ώστε να μπορούν να συσχετιστούν με τον εκάστοτε φορέα.

#### 4.6. Επισκόπηση αποθηκευμένων αναλύσεων

Με το πάτημα του πλήκτρου **VIEW RESULTS** ανοίγει το αντίστοιχο παράθυρο, όπως στην εικόνα:




Στην λίστα στο πάνω μέρος του παραθύρου φαίνονται 5 αρχεία προηγούμενων αναλύσεων, ενώ στο κάτω όλα τα μέλη του φορέα. Αν ο χρήστης επιλέξει ένα αρχείο και κάποιο από τα μέλη και πατήσει το 'View results', θα εμφανιστούν τα διαγράμματα των εσωτερικών εντάσεων. Επίσης, αν ενεργοποιήσει το 'Open the text file' χωρίς να επιλέξει κάποιο μέλος, θα ανοίξει και το αρχείο των τιμών. Οι δύο ενέργειες μπορούν να εκτελεστούν και ταυτόχρονα.

#### 4.7. Μενού κύριας λίστας

Το μενού της κύριας λίστας ανοίγει με το δεξί πλήκτρο του ποντικιού πάνω στο στοιχείο που ενδιαφέρει. Έως τώρα υπάρχει μόνο η επιλογή 'Delete'.

#### 4.8. Αποθήκευση εργασίας

Το εικονίδιο  χρησιμοποιείται για την αποθήκευση της εργασίας ως έχει. Εκτελεί την ίδια λειτουργία με την επιλογή 'Save' από το μενού της γραμμής εργαλείων, ενώ στο ίδιο μενού υπάρχει η επιλογή 'Save as...' που ανοίγει το παράθυρο επιλογής θέσης και ονόματος για το νέο αρχείο που πρόκειται να δημιουργηθεί. Σε ότι αφορά στους περιορισμούς της ονοματοδοσίας (αλληλουχίες και μεμονωμένοι χαρακτήρες) ισχύουν τα ίδια με ότι περιγράφηκαν σχετικά με τον αρχικό ορισμό ονόματος εργασίας.

#### 4.9. Κλείσιμο του προγράμματος

Η έξοδος από το πρόγραμμα γίνεται από το μενού της γραμμής εργαλείων, ή από το εικονίδιο του παραθύρου ή με τον συνδυασμό Alt+F4. Πριν εκτελεστεί ο τερματισμός, ο χρήστης θα ερωτηθεί αν επιθυμεί το κλείσιμο, την αποθήκευση της εργασίας του ή την ακύρωση της ενέργειας.

#### 4.10. Ανεπιθύμητη λειτουργία

Έχει εντοπιστεί και δεν έχει έως τώρα επιλυθεί μία ενέργεια η οποία προκαλεί πρόβλημα.

Κατά το άνοιγμα των αποτελεσμάτων της ανάλυσης αμέσως μετά το πέρας των υπολογισμών, ο χρήστης πρέπει πρώτα να κλείσει το αρχείο .txt και έπειτα το παράθυρο της ανάλυσης, διαφορετικά το πρόγραμμα σταματάει τη λειτουργία του. Το ίδιο πρέπει να γίνεται και στο παράθυρο της παρουσίασης των αποτελεσμάτων.



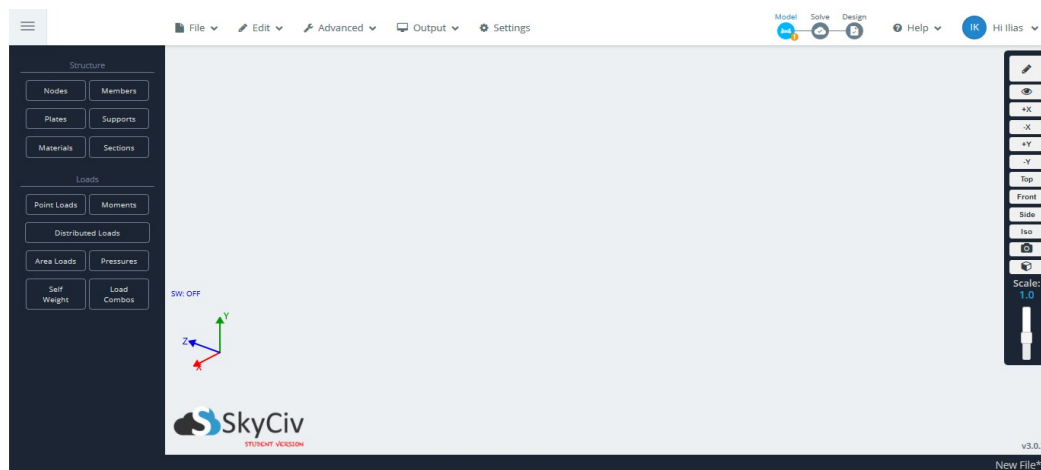
## Κεφάλαιο 5: Εφαρμογή

### 5.1 Γενικά

Εκτός από επιλυμένα παραδείγματα που χρησιμοποιήθηκαν προκειμένου να ελεγχθούν τα εξαγόμενα αποτελέσματα και να επιτευχθεί η αποσφαλμάτωση, πραγματοποιήθηκε και σύγκριση με λογισμικό του εμπορίου. Το λογισμικό που χρησιμοποιήθηκε για τον σκοπό αυτό είναι το Structural3D student edition (δεν έχει διαφορές από την εμπορική έκδοσή του), της εταιρίας SkyCiv, το οποίο λειτουργεί σε cloud (δηλαδή μόνο online).

### 5.2 Περιγραφή του SkyCiv

Η αρχική οθόνη που βλέπει ο χρήστης είναι η παρακάτω:

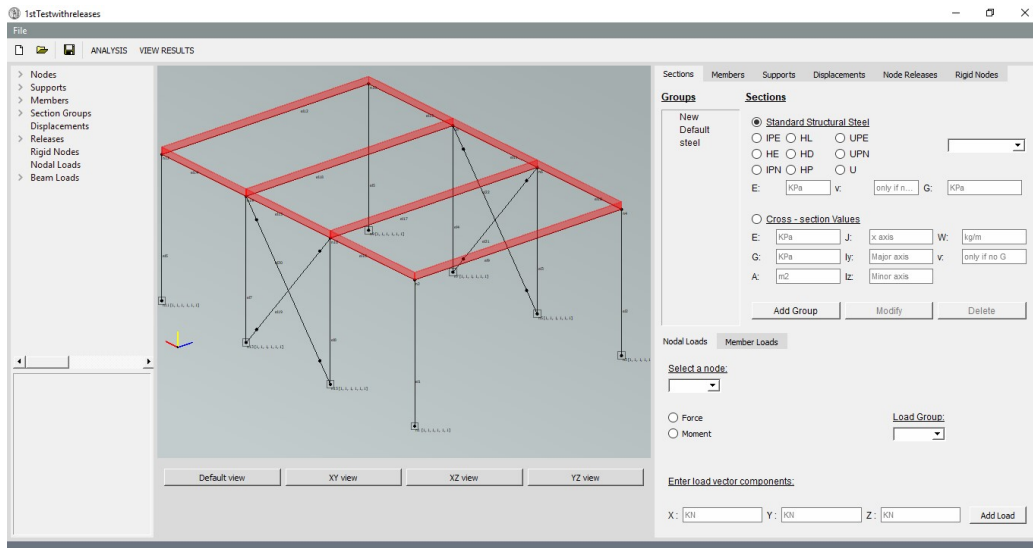
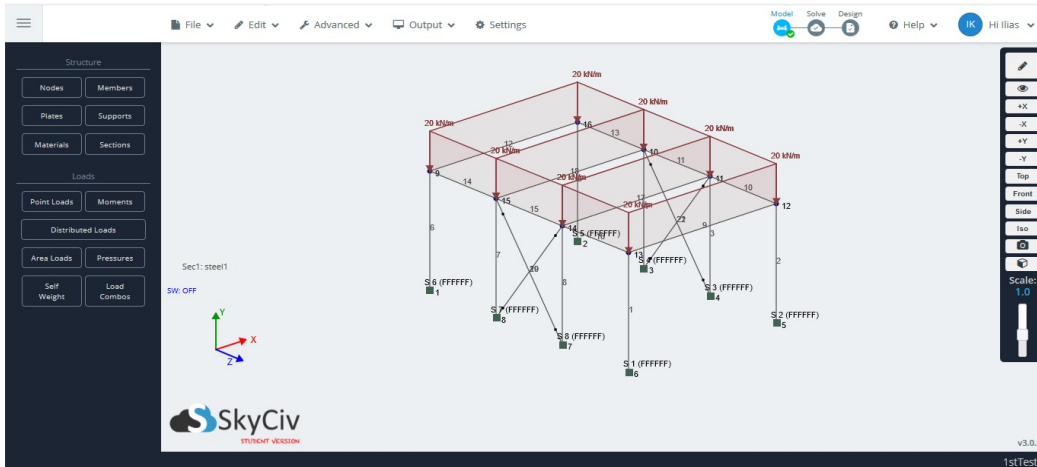


Αριστερά βρίσκονται όλα τα εργαλεία που χρειάζεται προκειμένου να υλοποιήσει το μοντέλο του φορέα, ενώ δεξιά υπάρχουν επιλογές θέασης. Στο πάνω μέρος μπορεί να βρει επιλογές φακέλων και ρυθμίσεις.

### 5.3 Περιγραφή του μοντέλου του φορέα

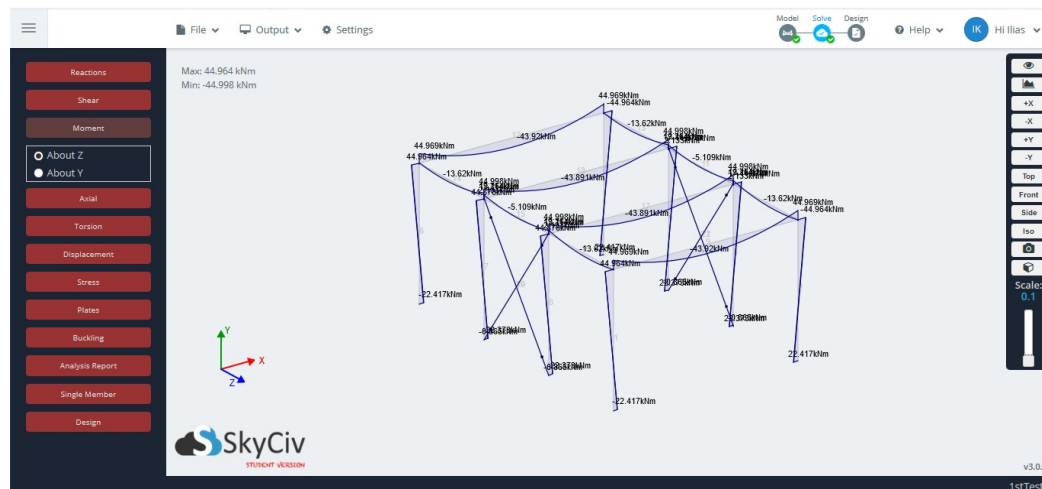
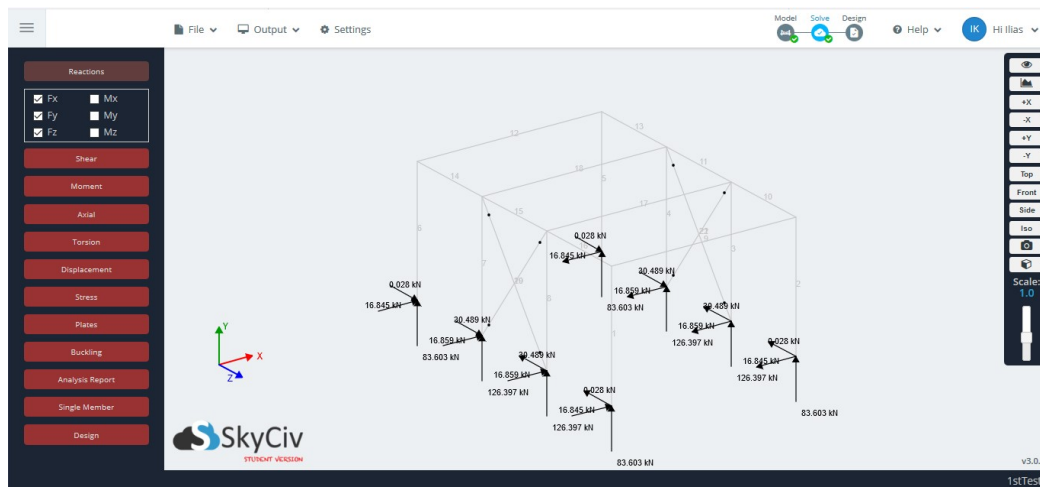
Χρησιμοποιήθηκε ένας απλός φορέας με 18 μέλη, εκ των οποίων τα 8 είναι υποστυλώματα, με κατακόρυφους χιαστί συνδέσμους δυσκαμψίας στις πλευρές με το μεγαλύτερο μήκος και ομοιόμορφα καταναμημένα φορτία σε όλα τα οριζόντια μέλη με μέτρο 20 KN/m.

Παρακάτω φαίνονται οι οθόνες του SkyCiv και του προγράμματος της παρούσας εργασίας, όπως απεικονίζουν το παραπάνω μοντέλο.



## 5.4 Ανάλυση του φορέα και εξαγωγή αποτελεσμάτων

Το SkyCiv επιτρέπει την αποθήκευση των διαφόρων εργασιών του χρήστη σε cloud. Η ανάλυση έγινε με ενεργοποίηση της επιλογής γραμμικής ανάλυσης. Η παρουσίαση των αποτελεσμάτων γίνεται όπως στις παρακάτω εικόνες, με επιλογές των μεγεθών που επιθυμεί να δει ο χρήστης από το αριστερό της οθόνης.



Επειδή η αναλυτική παρουσίαση όλων των αποτελεσμάτων μέσω του περιβάλλοντος του SkyCiv θα αποβεί ιδιαίτερα μακροσκελής, συγκεντρώνονται τα αποτελέσματα σε πίνακες, όπως παρακάτω. Έχουν εφαρμοστεί οι απαραίτητες αλλαγές που προκύπτουν από την διαφορετική τοποθέτηση των αξόνων, προκειμένου να είναι ευκολότερα αντιληπτή η σύγκριση, ενώ παρατηρούνται μικροδιαφορές στις τιμές. Οι διαφορές αυτές οφείλονται πιθανόν σε στρογγυλοποιήσεις που έγιναν κατά τη διάρκεια των υπολογισμών και δεν επηρεάζουν το τελικό αποτέλεσμα, με την έννοια ότι σε ό,τι αφορά τις κρίσιμες τιμές που θα χρησιμοποιηθούν κατά τον σχεδιασμό, η μέγιστη ποσοστιαία διαφορά είναι μικρότερη του 2%.

	<b>Αποτελέσματα με το SkyCiv</b>	<b>Αποτελέσματα με το παρόν πρόγραμμα</b>
<b>Αντιδράσεις</b>		
Κόμβος 1	16,845	16,85
	-0,028	1,37
	83,603	85,26
	0,037	-1,83
	22,417	22,42
	0,006	0
Κόμβος 3	-16,845	-16,85
	-0,028	1,37
	83,603	85,26
	0,037	-1,83
	-22,417	-22,42
	0,006	0
Κόμβος 5	-16,859	-17,45
	30,489	29,27
	126,397	124,74
	0,857	0,47
	-22,436	-23,2
	0,022	2,44

Κόμβος 7	-16,859	-17,45
	-30,489	-29,27
	126,397	124,74
	-0,857	-0,47
	-22,417	-23,2
	-0,022	-2,44
Κόμβος 9	-16,859	-16,85
	0,028	-1,37
	83,603	85,26
	-0,037	1,83
	-22,417	-22,42
	0,006	0
Κόμβος 11	16,845	16,85
	0,028	-1,37
	83,603	85,26
	-0,037	1,83
	22,436	22,42
	-0,006	0
Κόμβος 13	16,845	17,45
	-30,489	-29,27
	126,397	124,74
	-0,857	-0,47
	22,436	23,2
	0,022	2,44
Κόμβος 15	16,859	17,45
	30,489	29,27
	126,397	124,74
	0,857	0,47
	22,436	23,2
	-0,022	-2,44

	<b>Αποτελέσματα με το SkyCiv</b>	<b>Αποτελέσματα με το παρόν πρόγραμμα</b>
<b>Μετατοπίσεις - στροφές</b>		
Κόμβος 2	0.047	0,044
	-0.042	-0,036
	-0.311	-0,317
		-1,346
		6,106
		0,35
Κόμβος 4	-0.047	-0,044
	-0.042	-0,036
	-0.311	-0,317
		-1,346
		-6,106
		-0,35
Κόμβος 6	-0.047	-0.051
	-0.042	-0.04
	-0.322	-0.317
		0.382
		-5.446
		0,884
Κόμβος 8	-0.047	-0.051
	0.042	0.04
	-0.322	-0.317
		-0.382
		-5.446
		-0.884
Κόμβος 10	-0.047	-0.044
	0.042	0.036
	-0.311	-0.317
		1.346
		-6.106

		0.35
Κόμβος 12	0.047	0.044
	0.042	0.036
	-0.311	-0.317
		1.346
		6.106
		-0.35
Κόμβος 14	0.047	0.051
	0.042	0.04
	-0.322	-0.317
		-0.382
		5.446
		0.884
Κόμβος 16	0.047	0.051
	-0.042	-0.04
	-0.322	-0.317
		0.382
		5.446
		-0.884

**Ακραίες δράσεις μελών με το πρόγραμμα SkyCiv**

Μέλη	Αξονική	Y	Z	Ροπή στρέψης	Y	Z
1	83.603	0.028	-16.845	-0.006	-22.417	0.037
	83.603	0.028	-16.845	-0.006	44.964	-0.074
2	83.603	0.028	16.845	0.006	22.417	0.037
	83.603	0.028	16.845	0.006	-44.964	-0.074
3	86.736	-0.006	16.812	-0.069	22.373	-0.008
	86.736	-0.006	16.812	-0.069	-44.876	0.017
4	86.736	0.006	16.812	0.069	22.373	0.008

	86.736	0.006	16.812	0.069	-44.876	-0.017
5	83.603	-0.028	16.845	-0.006	22.417	-0.037
	83.603	-0.028	16.845	-0.006	-44.964	0.074
6	83.603	-0.028	-16.845	0.006	-22.417	-0.037
	83.603	-0.028	-16.845	0.006	44.964	0.074
7	86.736	0.006	-16.812	-0.069	-22.373	0.008
	86.736	0.006	-16.812	-0.069	44.876	-0.017
8	86.736	-0.006	-16.812	0.069	-22.373	-0.008
	86.736	-0.006	-16.812	0.069	44.876	0.017
9	16.839	0.000	60.000	0.000	44.969	0.000
	16.839	0.000	-60.000	0.000	44.969	0.000
10	0.028	0.007	23.603	-0.004	0.074	0.006
	0.028	0.007	-36.397	-0.004	19.264	-0.014
11	-30.474	0.000	30.000	0.000	17.114	0.007
	-30.474	0.000	-30.000	0.000	17.114	0.007
12	16.839	0.000	60.000	0.000	44.969	0.000
	16.839	0.000	-60.000	0.000	44.969	0.000
13	0.028	-0.007	36.397	0.004	19.264	-0.014
	0.028	-0.007	-23.603	0.004	0.074	0.006
14	0.028	0.007	23.603	-0.004	0.074	0.006
	0.028	0.007	-36.397	-0.004	19.264	-0.014
15	-30.474	0.000	30.000	0.000	17.114	0.007
	-30.474	0.000	-30.000	0.000	17.114	0.007
16	0.028	-0.007	36.397	0.004	19.264	-0.014
	0.028	-0.007	-23.603	0.004	0.074	0.006
17	16.866	0.000	60.000	0.000	44.998	-0.004
	16.866	0.000	-60.000	0.000	44.998	-0.004
18	16.866	0.000	60.000	0.000	44.998	0.004
	16.866	0.000	-60.000	0.000	44.998	0.004
19 έως 22	50.026	0.047	-0.600	0.000	-0.865	0.079
	50.026	0.047	-0.600	0.000	2.133	-0.158

**Ακραίες δράσεις μελών με το πρόγραμμα της παρούσας εργασίας**

Σημειώνεται ότι οι διαφορές πρόσημων σε κάποιες από τις τιμές οφείλονται στους παρακάτω λόγους:

α. Σε ότι αφορά τις αξονικές, το SkyCiv ορίζει τις θλίβουσες ως θετικές και τις εφελκούμενες ως αρνητικές, ενώ στο πρόγραμμα που αναπτύχθηκε στο παρόν, ορίζονται σύμφωνα με το τοπικό σύστημα συντεταγμένων του κάθε μέλους, δηλαδή οι θλίβουσες ως θετικές στην αρχή του μέλους και αρνητικές στο πέρας.

β. Για τις υπόλοιπες δράσεις, τόσο στη διαφορά τοποθέτησης των αξόνων, όσο και στο γεγονός ότι το SkyCiv εμφανίζει την τιμή η οποία συμπίπτει με την τιμή του αντίστοιχου διαγράμματος στην αρχή ή στο πέρας του μέλους, ενώ στο παρόν εμφανίζονται οι τιμές που προκύπτουν από τις εξισώσεις (άρα αφορούν την ισορροπία του μέλους ως μέλους συνολικά του φορέα). Κατά τον υπολογισμό των διαγραμμάτων η τελευταία διαφορά εξαλείφεται και ισχύει μόνο εκείνη της αναντιστοιχίας των αξόνων.

Μέλη	Αξονική	Y	Z	Ροπή στρέψης	Y	Z
1	85.27	1.37	-16.85	0.0	22.42	1.83
	-85.27	-1.37	16.85	0.0	44.96	3.63
2	85.27	1.37	16.85	0.0	-22.42	1.83
	-85.27	-1.37	-16.85,	0.0	-44.96	3.63
3	85.27	0.0	15.01	0.0	-19.97	0.0
	-85.27	0.0	-15.01	0.0	-40.07	0.0
4	85.27	0.0	15.01	0.0	-19.97	0.0
	-85.27	0.0	-15.01	0.0	-40.07	0.0
5	85.27	-1.37	16.85	0.0	-22.42	-1.83
	-85.27	1.37	-16.85	0.0	-44.96	-3.63
6	85.27	-1.37	-16.85	0.0	22.42	-1.83
	-85.27	1.37	16.85	0.0	44.96	-3.63
7	85.27	0.0	-15.01	0.0	19.97	0.0
	-85.27	0.0	15.01	0.0	40.07	0.0
8	85.27	0.0	-15.01	0.0	19.97	0.0
	-85.27	0.0	15.01	0.0	40.07	0.0

9	15.78	0.0	60.0	0.0	-44.97	0.0
	-15.78	0.0	60.0	0.0	44.97	0.0
10	1.43	0.0	25.25	0.0	-3.63	0.0
	-1.43	0.0	34.75	0.0	17.86	2.51
11	-28.69	0.0	30.0	0.0	-16.88	1.57
	28.69	0.0	30.0	0.0	16.88	-1.57
12	15.78	0.0	60.0	0.0	-44.97	0.0
	-15.78	0.0	60.0	0.0	44.97	0.0
13	1.43	0.0	34.75	0.0	-17.86	-2.51
	-1.43	0.0	25.25	0.0	3.63	0.0
14	1.43	0.0	25.25	0.0	-3.63	0.0
	-1.43	0.0	34.75	0.0,	17.86	2.51
15	-28.69	0.0	30.0	0.0	-16.88	1.57
	28.69	0.0	30.0	0.0	16.88	-1.57
16	1.43	0.0	34.75	0.0	, -17.86	-2.51
	-1.43	0.0	25.25	0.0	3.63	0.0
17	18.29	0.0	60.0	0.0	-46.6	0.0
	-18.29	0.0	60.0	0.0	46.6	0.0
18	18.29	0.0	60.0	0.0	-46.6	0.0
	-18.29	0.0	60.0	0.0	46.6	0.0
19 έως 22	49.41	~0	0.0	0.0	0.0	~0
	-49.41	~0	0.0	0.0	0.0	~0

## Κεφάλαιο 6: Επίλογος

### 6.1 Γενικά

Στον επίλογο της παρούσας εργασίας θεωρείται σκόπιμο να αναφερθούν πρωτίστως δυσχέρειες που αντιμετωπίστηκαν κατά την υλοποίηση του προγράμματος, οι οποίες θα αποτελέσουν οδηγό, σε περίπτωση μελλοντικών επεκτάσεων ή βελτιώσεων. Θα ακολουθήσει σχολιασμός εξαγόμενων συμπερασμάτων και τέλος θα αναφερθούν σκέψεις για μελλοντικές χρήσεις.

### 6.2 Δυσχέρειες και Αντιμέτωπιση

α. Σε ότι αφορά το PyQt5, η ελλιπής βιβλιογραφία και απουσία κάποιου ολοκληρωμένου εγχειριδίου χρήσης, που πιθανότατα οφείλεται στο τεράστιο σε έκταση Qt5, που αποτελεί τη βάση του, δυσχέραναν αρχικά την πρόοδο της εργασίας. Το γεγονός ότι υπάρχει ένας πλήρης οδηγός, που όμως δεν είναι αρκετά αναλυτικός, γραμμένος για το Qt5, άρα για γλώσσα C++, οδήγησε αναγκαστικά στην κατανόηση κατ'ελάχιστο των βασικών αρχών της σύνταξης της νέας γλώσσας. Πολύτιμο βοήθημα αποτέλεσαν ιστοσελίδες όπως η [sourceforge.net](http://sourceforge.net) και [stackoverflow.com](http://stackoverflow.com), όπου απαντάται κάθε είδους απορία και ερώτηση από εμπειρότερους χρήστες (παρ'όλ'αυτά και εδώ χρειάζεται μια βασική γνώση της C++).

Τέλος, στα αρχικά στάδια ανάπτυξης του προγράμματος, χρησιμοποιήθηκε εκτενώς το Qt Designer, μια εφαρμογή που εγκαθίσταται μαζί με το PyQt5 και στην οποία ο σχεδιασμός του περιβάλλοντος γίνεται με οπτικοποιημένο τρόπο, δηλαδή τα αντικείμενα αποτελούν blocks που ο χρήστης τα τοποθετεί όπου θέλει με τη χρήση του ποντικιού. Στη συνέχεια, εξάγεται ως αρχείο XML και τέλος με τη χρήση του command prompt μετατρέπεται σε αρχείο python. Η παραπάνω διαδικασία παρά το γεγονός ότι ουσιαστικά δημιουργεί ένα έτοιμο παράθυρο όπως ακριβώς σχεδιάστηκε στο οπτικοποιημένο περιβάλλον του Qt Designer, δεν αποτελεί ολοκληρωμένη λύση, αφού ο αυτοματοποιημένος τρόπος με τον οποίο εξάγεται ο κώδικας σε Python, ποτέ δεν ήταν ο ζητούμενος. Η χρήση του, όμως, αποκλειστικά

ως ενδεικτικού τρόπου γραφής μεμονωμένων τμημάτων κώδικα, κατά τα πρώτα στάδια εκμάθησης του PyQt5, κρίνεται απαραίτητη.

β. Η εξαγωγή εκτελέσιμων αρχείων με τη χρήση του CxFreeze παρουσίασε επίσης αρκετά προβλήματα. Από την αρχή της συγγραφής του κώδικα, αναζητήθηκε κάποιο αξιόπιστο εργαλείο, όμως λόγω απειρίας η επιλογή έγινε με βάση απόψεις άλλων χρηστών. Δεν θα γίνει αναφορά σε όλα τα πιθανά προβλήματα που μπορεί να προκύψουν, αλλά θα περιγραφούν μόνο οι ενέργειες που χρειάστηκαν για το παρόν πρόγραμμα, προκειμένου να λειτουργήσει σωστά μετά την εξαγωγή του εκτελέσιμου αρχείου. Αυτές είναι οι παρακάτω:

(1) Αντιγραφή όλων των αρχείων ```mkl_*.dll``` καθώς και του ```libiomp5md.dll``` από τον φάκελο ```Anaconda3\Library\bin``` στον φάκελο ```Anaconda3\Lib\site-packages\numpy\core```, του εξαγόμενου φακέλου.

(2) Αντιγραφή του αρχείου ```qt.conf``` από τον φάκελο ```Anaconda3``` απευθείας στον εξαγόμενο φάκελο.

(3) Αντιγραφή του φακέλου ```platforms``` από τον φάκελο ```Anaconda3``` απευθείας στον εξαγόμενο φάκελο.

Οι αναφερθείσες ενέργειες δεν ήταν δυνατόν να εκτελεστούν με την ενσωμάτωση των αρχείων στο αρχείο `setup.py`, το οποίο δέχεται όλες τις παραμέτρους που θέτει ο χρήστης, συνεπώς έγινε εκ των υστέρων. Η επίλυση των παραπάνω προήλθε έπειτα από εκτενή έρευνα και πολλαπλές δοκιμές.

### 6.3 Συμπεράσματα

Η σύγκριση του λογισμικού που αναπτύχθηκε στη παρούσα εργασία, με το εμπορικό λογισμικό SkyCin δείχνει ότι το πρώτο αποτελεί ένα πρόσθετο εργαλείο για τον σπουδαστή, σε επίπεδο εξαγωγής αξιόπιστων αποτελεσμάτων. Η διαδικασία κατανόησης των περιορισμών του παρόντος, σε σχέση με κάποιο εξελιγμένο λογισμικό, κυρίως σε ότι αφορά την προσομοίωση του κάθε φορέα, καθώς η καθοδήγηση και η πρόληψη σφαλμάτων είναι πιο περιορισμένη, θα τον οδηγήσει σε περαιτέρω διερεύνηση της μηχανικής συμπεριφοράς των κατασκευών και της σχέσης μοντέλου - πραγματικού φορέα.

#### 6.4 Προτάσεις

Ο μέχρι τώρα έλεγχος του προγράμματος δίνει αξιόπιστα αποτελέσματα. Παρ'όλ'αυτά, το γεγονός ότι συνεχώς προκύπτουν μικροσφάλματα που προέρχονται από περιπτώσεις φορέων που δεν έχουν επαναληφθεί, αποτελεί ένδειξη ότι θα προκύψει ανάγκη για επιπλέον διορθώσεις. Προτείνεται μια οργανωμένη και ενδεδειγμένη εξέταση διαφορετικών μοντέλων, προκειμένου να εντοπιστούν σφάλματα υπολογιστικής φύσης. Εναλλακτικά, μπορεί να προβλεφθεί μια διαδικασία αναφορών προβλημάτων από χρήστες. Τα παραπάνω μπορούν να εφαρμοστούν και για προβλήματα που αφορούν την γενικότερη λειτουργία του προγράμματος.

Σε επίπεδο γραφικού περιβάλλοντος, πολλά από τα εργαλεία που αναπτύχθηκαν (π.χ. μενού, γραμμή εργαλείων κ.ά.), έχουν περισσότερο ενδεικτικό χαρακτήρα. Έτσι, παρέχουν μεν περιορισμένες επιλογές, αλλά μπορούν πολύ εύκολα να επεκταθούν, αφού ο υπάρχων κώδικας αποτελεί τον οδηγό.

Τέλος, προτείνεται η βελτίωση της τρισδιάστατης απεικόνισης του μοντέλου, η οποία λόγω χρονικών περιορισμών δεν ήταν δυνατόν να αναπτυχθεί περαιτέρω και που μπορεί να μην αποτελεί το βασικό σκέλος ενός λογισμικού στατικής ανάλυσης, όμως είναι ένα απαραίτητο μέσο κατανόησης της γενικότερης δομής του φορέα.

## Βιβλιογραφία - Παραπομπές

Ανάλυση ραβδωτων φορέων με μητρωικες μεθόδους – μέθοδος αμεσης στιβαρότητας  
(Μ. Παπαδρακακης – Ε.Ι. Σαπουντζάκης)

<https://skyciv.com>

<http://doc.qt.io/qt-5>

<https://anaconda.org>

<https://www.riverbankcomputing.com>

<https://stackoverflow.com>

<https://sourceforge.net>

<https://zetcode.com/gui/pyqt5>

<http://python.org.gr>

<http://www.numpy.org>

<https://pandas.pydata.org>

<https://cx-freeze.readthedocs.io/en/latest/index.html>

<http://www.eng.ucy.ac.cy/petros>

<http://users.teiath.gr/fkokkinos>

<http://engprog.com/tag/direct-stiffness-method>

<https://pythonforengineers.com>