



# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΗΣ ΙΣΧΥΟΣ

## Ανάπτυξη διαδραστικού εκπαιδευτικού υλικού στα Συστήματα Ηλεκτρικής Ενέργειας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΑΝΔΡΙΑΝΑΣ ΔΗΜΗΤΡΙΟΥ

**Επιβλέπων :** Χατζηαργυρίου Νικόλαος  
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2020





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΗΣ ΙΣΧΥΟΣ

## Ανάπτυξη διαδραστικού εκπαιδευτικού υλικού στα Συστήματα Ηλεκτρικής Ενέργειας

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

**ΑΝΔΡΙΑΝΑΣ ΔΗΜΗΤΡΙΟΥ**

**Επιβλέπων :** Χατζηαργυρίου Νικόλαος  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22<sup>η</sup> Οκτωβρίου 2020.

(Υπογραφή)

.....  
Χατζηαργυρίου Νικόλαος  
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....  
Παπαθανασίου Σταύρος  
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....  
Γεωργιάκης Πάυλος  
Αναπλ. Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2020

(Υπογραφή)

.....

**ΑΝΔΡΙΑΝΑ ΔΗΜΗΤΡΙΟΥ**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © ΑΝΔΡΙΑΝΑ ΔΗΜΗΤΡΙΟΥ, 2020

Με επιφύλαξη παντός δικαιώματος. All rights reserved

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

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τους συγγραφείς και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.



## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Νικόλαο Χατζηαργυρίου για τη δυνατότητα που μου έδωσε να ασχοληθώ με την παρούσα διπλωματική εργασία.

Επίσης θα ήθελα να ευχαριστήσω τον Πάνο Κοτσαμπόπουλο που με την υπομονετική του καθοδήγηση με βοήθησε να την ολοκληρώσω.

Ευχαριστώ θερμά όλους τους φίλους και συμφοιτητές που πορευτήκαμε μαζί αυτά τα χρόνια και ιδιαίτερα την Έφη, τη Μαρία, την Αλεξάνδρα και τον Γιώργο.

Τέλος ευχαριστώ πολύ τους γονείς μου, Φάνη και Χρυσάνθη και τα αδέρφια μου, Κρυσταλία και Αργύρη για τη συνεχή τους υποστήριξη και ενθάρρυνση όλα τα χρόνια των σπουδών μου.

Η εργασία αυτή είναι αφιερωμένη στην οικογένειά μου.



## Περίληψη

Η αυξανόμενη πολυπλοκότητα και ο εξελισσόμενος χαρακτήρας των Συστημάτων Ηλεκτρικής Ενέργειας, με την ενσωμάτωση ανανεώσιμης και διεσπαρμένης παραγωγής, καθώς και καινοτόμων τεχνολογιών ελέγχου, πληροφορικής και επικοινωνιών, δημιουργεί νέες εκπαιδευτικές ανάγκες. Σε αυτό το πλαίσιο παρουσιάζονται σύγχρονες τάσεις και εργαλεία για τη βελτίωση της εκπαίδευσης των Ηλεκτρολόγων Μηχανικών στα Συστήματα Ηλεκτρικής Ενέργειας.

Ιδιαίτερη έμφαση δίνεται στο Jupyter Notebook, μια διαδραστική πλατφόρμα που παρέχει ένα περιβάλλον ανάπτυξης εκπαιδευτικού υλικού, συνδυάζοντας κείμενο, εικόνες και κώδικα, σε ένα ενιαίο έγγραφο, προσβάσιμο μέσω του διαδικτύου. Αναλύονται τα πλεονεκτήματα της χρήσης του ως εκπαιδευτικό εργαλείο και παρουσιάζονται αντιπροσωπευτικά παραδείγματα αξιοποίησης του σε μαθήματα Ηλεκτρολόγων Μηχανικών από τη διεθνή βιβλιογραφία.

Στη συνέχεια περιγράφεται το διαδραστικό εκπαιδευτικό υλικό που αναπτύχθηκε σε Jupyter Notebook και αφορά θέματα της Ανάλυσης ΣΗΕ, όπως η συμβολή της Διεσπαρμένης Παραγωγής στη ρύθμιση τάσης και η Ανάλυση Ροής Φορτίου. Οι εφαρμογές αποσκοπούν στη δημιουργία μιας πιο διαισθητικής αντίληψης των φαινομένων που εξετάζουν, στην καλύτερη εμπέδωση της μεθοδολογίας και στην εξοικείωση των φοιτητών με προγραμματιστικά εργαλεία χρήσιμα για τη μετέπειτα πορεία τους.

Στην παρούσα εργασία αντανakλάται η εμπειρία που αποκτήθηκε από τη βιβλιογραφική ανασκόπηση και κυρίως την ανάπτυξη του εκπαιδευτικού υλικού σε Jupyter Notebook, προτείνονται ιδέες για μελλοντική αξιοποίησή του και παρατίθενται συμπεράσματα για τη συνεισφορά του εργαλείου αυτού στην ανάπτυξη καινοτόμων μεθόδων διδασκαλίας στο πεδίο των Συστημάτων Ηλεκτρικής Ενέργειας.

**Λέξεις Κλειδιά:** Συστήματα Ηλεκτρικής Ενέργειας (ΣΗΕ), ευφυή δίκτυα, Τεχνολογίες Πληροφορικής και Επικοινωνιών (ΤΠΕ), Jupyter Notebook, εργαστηριακή εκπαίδευση, ρύθμιση τάσης, ανάλυση ροής φορτίου





## **Abstract**

The rising complexity and evolving nature of power systems, with the integration of renewable and distributed generation as well as innovative control, information and communication technologies, creates new educational needs. In this context, modern trends and tools for the improvement of power systems education are presented.

Focus is put on Jupyter Notebook, an interactive platform that provides an educational environment, combining text, images and code into a single document, accessible through a web browser. The advantages of its use as an educational tool are analyzed and examples of its implementation in Electrical Engineering courses from the international literature are presented.

The development of interactive educational material using Jupyter Notebook in the field of Power Systems Analysis is described, focusing on the contribution of Distributed Generation to voltage regulation and the Load Flow Analysis. The applications aim to create a more intuitive perception of the phenomena they examine, to improve the understanding of the methodology and to familiarize students with programming tools useful for their future career.

The present work reflects the experience gained from the literature review and mainly from the development of educational material in Jupyter Notebook. It proposes ideas for its future implementation and presents conclusions for the contribution of this tool to the development of innovative teaching methods in the field of power systems.

**Keywords:** Power Systems, Smart Grids, Information and Communication Technology (ICT), Jupyter Notebook, laboratory education, voltage regulation, load flow analysis

# Περιεχόμενα

Περιεχόμενα.....	11
Ευρετήριο Σχημάτων.....	13
<b>Κεφάλαιο 1: Εισαγωγή .....</b>	<b>15</b>
Σύνοψη .....	15
1.1 Σύγχρονες τάσεις στα Συστήματα Ηλεκτρικής Ενέργειας .....	15
1.1.1 Ενσωμάτωση των ΑΠΕ στο δίκτυο .....	19
1.1.2 Αποθήκευση ενέργειας.....	19
1.1.3 Ενσωμάτωση ηλεκτρικών οχημάτων .....	20
1.1.4 Διεσπαρμένη παραγωγή μικρής κλίμακας και μικροδίκτυα .....	20
1.1.5 Διαμόρφωση της αγοράς και έξυπνος καταναλωτής .....	20
1.1.6 Ενσωμάτωση των ΤΠΕ και άλλων προηγμένων τεχνολογιών.....	21
1.2 Σύγχρονες τάσεις στην εκπαίδευση των ΣΗΕ .....	22
1.2.1 Massive Open Online Courses (MOOCs).....	23
1.2.2 Διαδραστικά Notebooks (Jupyter Notebooks) .....	24
1.2.3 Επαυξημένη πραγματικότητα (Augmented reality) .....	26
1.3 Αντικείμενο της εργασίας.....	28
<b>Κεφάλαιο 2: Αξιοποίηση των Jupyter Notebooks στην εκπαιδευτική διαδικασία</b>	<b>30</b>
Σύνοψη .....	30
2.1 Τι είναι το Jupyter Notebook.....	30
2.2 Πλεονεκτήματα του Jupyter Notebook στην Εκπαίδευση .....	34
2.4 Χρήση του Jupyter Notebook στην Εκπαίδευση Μηχανικών: μελέτες περίπτωσης .....	36
2.4.1 Jupyter Notebooks για εργαστήρια στα «Συστήματα Ηλεκτρικής Ενέργειας» με χρήση υπολογιστή. ....	36
2.4.2 Jupyter Notebooks για εργαστήρια στον «Έλεγχο και Δυναμική Συμπεριφορά ΣΗΕ» με χρήση υπολογιστή. ....	38
2.4.3 Διαδικτυακές Εργαστηριακές Ασκήσεις για την Εκπαίδευση στις Ηλεκτρικές Μηχανές.....	40
2.5 Πρόσβαση στο Jupyter Notebook .....	44
2.5.1 Τοπική εγκατάσταση.....	44
2.5.2 Χρήση μέσω απομακρυσμένου server .....	44
2.5.3 Υποστήριξη λειτουργίας του Jupyter Notebook .....	46
<b>Κεφάλαιο 3: Ανάπτυξη διαδραστικού υλικού για τη ρύθμιση τάσης.....</b>	<b>48</b>
Σύνοψη .....	48

3.1 Ρύθμιση τάσης στα Δίκτυα Διανομής .....	48
3.2 Εφαρμογή για τη ρύθμιση τάσης σε Jupyter Notebook .....	51
3.1.1 Περιγραφή του προγράμματος.....	51
3.1.2 Παράδειγμα εκτέλεσης του προγράμματος.....	56
<b>Κεφάλαιο 4: Ανάπτυξη διαδραστικού υλικού στη ροή φορτίου.....</b>	<b>58</b>
Σύνοψη .....	58
4.1 Ροή Φορτίου .....	58
4.1.1 Εξισώσεις Ροών Φορτίου .....	58
4.1.2 Θεμελίωση του προβλήματος ροών φορτίου .....	60
4.1.3 Τεχνικές επίλυσης προβλήματος ροών φορτίου .....	62
4.2 Εφαρμογή στη ροή φορτίου .....	65
4.2.1 Υπάρχον εκπαιδευτικό υλικό στο πρόγραμμα MATLAB .....	66
4.2.2 Παράδειγμα εφαρμογής του προβλήματος ροής φορτίου σε γλώσσα MATLAB στο Jupyter Notebook.....	69
4.2.3 Εργαστηριακή άσκηση σε γλώσσα MATLAB στο cloud (CoCalc).....	84
4.2.4 Εργαστηριακή άσκηση σε γλώσσα Python στο cloud (Colab).....	96
<b>Κεφάλαιο 5 : Συμπεράσματα .....</b>	<b>106</b>
Σύνοψη .....	106
5.1 Εμπειρία χρήσης .....	106
5.2 Σχετικές εργασίες – Μελλοντική έρευνα .....	109
5.3 Συμπεράσματα.....	111
<b>Βιβλιογραφία .....</b>	<b>113</b>
<b>Δημοσιεύσεις .....</b>	<b>119</b>
<b>Παράρτημα .....</b>	<b>120</b>
Π1 .....	120
Π2 .....	123
Π3 .....	128

## Ευρετήριο Σχημάτων

Σχήμα 1. 1 .....	16
Σχήμα 1. 2 .....	18
Σχήμα 1. 3 .....	24
Σχήμα 1. 4 .....	25
Σχήμα 1. 5 .....	27
Σχήμα 2. 1 .....	30
Σχήμα 2. 2 .....	31
Σχήμα 2. 3 .....	31
Σχήμα 2. 4 .....	32
Σχήμα 2. 5 .....	35
Σχήμα 2. 6 .....	37
Σχήμα 2. 7 .....	39
Σχήμα 2. 8 .....	42
Σχήμα 2. 9 .....	47
Σχήμα 3. 1 .....	49
Σχήμα 3. 2 .....	54
Σχήμα 3. 3 .....	55
Σχήμα 3. 4 .....	57
Σχήμα 4. 1 .....	59
Σχήμα 4. 2 .....	68
Σχήμα 4. 3 .....	71
Σχήμα 4. 4 .....	73
Σχήμα 4. 5 .....	74
Σχήμα 4. 6 .....	77
Σχήμα 4. 7 .....	78
Σχήμα 4. 8 .....	79
Σχήμα 4. 9 .....	80
Σχήμα 4. 10 .....	82
Σχήμα 4. 11 .....	83
Σχήμα 4. 12 .....	85
Σχήμα 4. 13 .....	86
Σχήμα 4. 14 .....	87
Σχήμα 4. 15 .....	88
Σχήμα 4. 16 .....	89
Σχήμα 4. 17 .....	90
Σχήμα 4. 18 .....	91
Σχήμα 4. 19 .....	92
Σχήμα 4. 20 .....	93
Σχήμα 4. 21 .....	94
Σχήμα 4. 22 .....	95
Σχήμα 4. 23 .....	97
Σχήμα 4. 24 .....	98
Σχήμα 4. 25 .....	99

Σχήμα 4. 26.....	100
Σχήμα 4. 27.....	101
Σχήμα 4. 29.....	102
Σχήμα 4. 30.....	103
Σχήμα 4. 31.....	104
Σχήμα 4. 32.....	105

# Κεφάλαιο 1: Εισαγωγή

## Σύνοψη

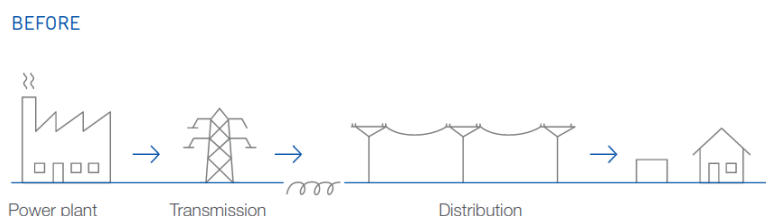
Το κεφάλαιο αυτό αποτελεί μια περιληπτική εισαγωγή στα σύγχρονα Συστήματα Ηλεκτρικής Ενέργειας (ΣΗΕ). Περιγράφει τις κυριότερες προκλήσεις που αντιμετωπίζουν τα παραδοσιακά ΣΗΕ, οι οποίες οδηγούν σταδιακά στη μετάβαση στα Ευφυή Δίκτυα (Smart Grids). Η αυξανόμενη χρήση των Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ), η ηλεκτροκίνηση, οι βελτιώσεις της τεχνολογίας για την απόκτηση ενός πιο αξιόπιστου και ανθεκτικού δικτύου και οι προκλήσεις που αφορούν την άνευ εκπομπών και βιώσιμη παροχή ενέργειας αποτελούν τους κύριους μοχλούς της τρέχουσας μετάβασης προς ένα έξυπνο ενεργειακό σύστημα, το οποίο υποστηρίζεται από τις συνεχιζόμενες εξελίξεις στον τομέα των Τεχνολογιών της Πληροφορικής και των Επικοινωνιών (ΤΠΕ). Ως αποτέλεσμα αυτής της μετάβασης, πρέπει να δοθεί ιδιαίτερη προσοχή στις εκπαιδευτικές πτυχές που καλύπτουν την ολοένα αυξανόμενη πολυπλοκότητα των ενεργειακών συστημάτων ώστε οι σημερινοί και μελλοντικοί μηχανικοί και ερευνητές να αποκτήσουν ευρεία κατανόηση των θεμάτων που αφορούν τα Ευφυή Δίκτυα. Σε αυτό το πλαίσιο, παρουσιάζονται σύγχρονες εκπαιδευτικές τάσεις και εργαλεία που μπορούν να ενισχύσουν την εκπαίδευση των ΣΗΕ.

## 1.1 Σύγχρονες τάσεις στα Συστήματα Ηλεκτρικής Ενέργειας

Το ηλεκτρικό δίκτυο είναι αναμφίβολα μία από τις σπουδαιότερες καινοτομίες της μηχανικής και αποτελεί τον μεγαλύτερο εξοπλισμό που έχει κατασκευαστεί ως σήμερα. Έχει καταστεί θεμελιώδες για τον σύγχρονο πολιτισμό, παρέχοντας ηλεκτρική ενέργεια άμεσα, ευέλικτα και σχετικά οικονομικά σε σπίτια, γραφεία, εργοστάσια, βιομηχανίες σε όλο τον κόσμο, ώστε αυτή να μετατραπεί σε κάθε είδους χρήσιμη μορφή ενέργειας όπως θερμότητα, φως, κίνηση και να καλύψει τις ανάγκες των καταναλωτών. Λόγω των πολυάριθμων πλεονεκτημάτων της, η ηλεκτρική ενέργεια αναγνωρίζεται ολοένα και περισσότερο ως κλειδί για την κοινωνική πρόοδο και την οικονομική ευημερία, βελτιώνοντας την ποιότητα ζωής [1], [2].

Τα παραδοσιακά ΣΗΕ είναι μεγάλα και πολύπλοκα διασυνδεδεμένα δίκτυα τα οποία επιτρέπουν την παροχή ηλεκτρικής ενέργειας από τους παραγωγούς προς τους καταναλωτές. Η ανάπτυξη και εφαρμογή των πρώτων ΣΗΕ ξεκίνησε στις ΗΠΑ στα τέλη του 19<sup>ου</sup> αιώνα. Έκτοτε η εξέλιξή τους ήταν ταχύτερη και η παγκόσμια εξάπλωσή τους ραγδαία. Οι αρχές λειτουργίας τους στηρίζονται στις επαναστατικές ανακαλύψεις και την θεωρητική εργασία των Thomas Edison και Nikola Tesla, οι οποίες έθεσαν τις βάσεις για την κατασκευή των σύγχρονων ΣΗΕ και διαμόρφωσαν το κύριο μοντέλο οργάνωσής τους, που ισχύει ακόμη και σήμερα. Με βάση αυτό, η ηλεκτρική ενέργεια παράγεται από την καύση ορυκτών καυσίμων (όπως ο άνθρακας, ο λιγνίτης, το πετρέλαιο) στους σταθμούς παραγωγής και μεταφέρεται μέσω των γραμμών μεταφοράς υψηλής τάσης σε μεγάλες αποστάσεις προς κεντρικά σημεία του δικτύου, τους υποσταθμούς. Από εκεί η τάση υποβιβάζεται και η ηλεκτρική ενέργεια διανέμεται

στους καταναλωτές ώστε να χρησιμοποιηθεί με διάφορους τρόπους, όπως βιομηχανική, εμπορική και οικιακή χρήση [3], [4]. Ορισμένα κύρια χαρακτηριστικά του μοντέλου αυτού είναι η συγκεντρωμένη παραγωγή ενέργειας, η ροή της προς μία μόνο κατεύθυνση και η άμεση παραγωγή και κατανάλωσή της, εξαιτίας της απουσίας εμπορικής λύσης για την αποθήκευση της ηλεκτρικής ενέργειας σε μεγάλη κλίμακα [5], [6]. Στο Σχήμα 1.1 φαίνεται η δομή του παραδοσιακού ΣΗΕ.



Σχήμα 1. 1: Το παραδοσιακό ΣΗΕ [7].

Είναι γεγονός ότι αυτή η παραδοσιακή μορφή οργάνωσης υπηρέτησε άριστα τις ενεργειακές ανάγκες της κοινωνίας για πολλά χρόνια. Ωστόσο τις τελευταίες δεκαετίες τα δεδομένα και οι απαιτήσεις έχουν σταδιακά αλλάξει λόγω σοβαρών κοινωνικών, οικονομικών και τεχνικών ζητημάτων, με αποτέλεσμα τα ΣΗΕ ολοένα και περισσότερο να καλούνται να αντιμετωπίσουν προκλήσεις για τις οποίες δεν έχουν σχεδιαστεί. Παρουσιάζεται λοιπόν η ανάγκη ενίσχυσης ή και αντικατάστασης των «γηρασμένων» υποδομών παραγωγής, μεταφοράς και διανομής του ΣΗΕ, με την παράλληλη ενσωμάτωση νέων τεχνολογιών που θα επιτρέψουν ένα καθαρότερο, πιο έξυπνο και πιο αξιόπιστο ενεργειακό σύστημα, εξασφαλίζοντας ταυτόχρονα την κοινωνική ευημερία, την οικονομική ανάπτυξη και την προστασία του περιβάλλοντος. Την ίδια στιγμή αμφισβητείται η βασική φιλοσοφία της ανάπτυξης του συστήματος με βασικά χαρακτηριστικά την συγκεντρωμένη παραγωγή και τις μεγάλες γραμμές μεταφοράς, καθώς ένα πλήθος ετερογενών στοιχείων εμπλέκονται σταδιακά στο ΣΗΕ όπως η αποκέντρωση της παραγωγής, η ενσωμάτωση αποθήκευσης (στάσιμη ή κινούμενη), η απελευθέρωση των αγορών και άλλα, με αποτέλεσμα να αυξάνεται η πολυπλοκότητα [1], [5], [8]-[10].

Οι βασικές προκλήσεις που υφίσταται το δίκτυο συνοψίζονται παρακάτω:

### Περιβαλλοντικές προκλήσεις:

Υπάρχει έντονη ανάγκη μετάβασης από συμβατικές σε ανανεώσιμες πηγές ενέργειας, όπως ηλιακή και αιολική, ώστε να ελαχιστοποιηθούν οι επιπτώσεις της κλιματικής αλλαγής λόγω των εκπομπών διοξειδίου του άνθρακα που οφείλονται στην θερμική παραγωγή ενέργειας, αλλά και για να εξασφαλιστεί ο μακροπρόθεσμος ενεργειακός εφοδιασμός. Επιπλέον μέτρο για τη μείωση των εκπομπών των αερίων του θερμοκηπίου αποτελεί ο εξηλεκτρισμός των μεταφορών, ο οποίος σηματοδοτεί την εμφάνιση μιας νέας κατηγορίας φορτίων, τα κινητά φορτία που μπορούν να χρησιμοποιηθούν και ως πηγές ενέργειας. Αυτή η σταδιακή αύξηση της διεσπαρμένης παραγωγής δημιουργεί προκλήσεις στην υπάρχουσα υποδομή λόγω της μεγάλης



γεωγραφικής διασποράς, αλλά και λόγω των διακυμάνσεων στη λειτουργία τους. Επίσης, το ΣΗΕ δεν είναι σχεδιασμένο να διαχειριστεί την αμφίδρομη ροή ενέργειας ούτε φορτία των οποίων η κατανάλωση δεν περιορίζεται σε μία τοποθεσία. Η εγκατάσταση συσκευών αποθήκευσης ενέργειας και άλλες προηγμένες τεχνολογίες χρειάζονται ώστε να ενσωματωθούν ομαλά τα παραπάνω στοιχεία και να αυξηθεί η αξιοπιστία και ευστάθεια του συστήματος [4], [5], [11].

#### **Ανάγκες αγοράς και καταναλωτών:**

Η βιομηχανία του ηλεκτρισμού ενθαρρύνει τον ανταγωνισμό στην αγορά ενέργειας. Έτσι παρέχεται η δυνατότητα στους καταναλωτές να επιλέγουν πιο «καθαρές» και ευέλικτες ενεργειακές λύσεις καθώς και να συμμετέχουν ενεργά στην παραγωγή ενέργειας. Ωστόσο η απελευθέρωση της αγοράς ενέργειας οδηγεί σε σενάρια που το σύστημα δεν ήταν σχεδιασμένο να διαχειριστεί [5], [12].

#### **Καινοτόμες τεχνολογίες:**

Καθώς ο προγραμματισμός της παραγωγής γίνεται πιο πολύπλοκος, λόγω της στοχαστικής λειτουργίας των ΑΠΕ και του μεταβαλλόμενου χαρακτήρα της ζήτησης, προηγμένες τεχνολογίες χρειάζονται ώστε να κάνουν τη χρήση των ΑΠΕ πιο αξιόπιστη, ασφαλή και αποδοτική. Επιπλέον, για την καλύτερη λειτουργία του ΣΗΕ υπάρχει ανάγκη αξιοποίησης των πληροφοριών που διατίθενται σε όλη την κατακευχημένη δομή του δικτύου, από την παραγωγή και μεταφορά μέχρι τη διανομή και τη χρήση της ενέργειας. Οι ΤΠΕ μπορούν να διαδραματίσουν σημαντικό ρόλο στην αντιμετώπιση αυτών των προκλήσεων, συνεισφέροντας στον δραστικό ψηφιακό μετασχηματισμό των ενεργειακών συστημάτων. Με τη χρήση έξυπνων μετρητών και άλλων αναπτυσσόμενων τεχνολογιών, δίνεται η δυνατότητα συλλογής και επεξεργασίας δεδομένων σε πραγματικό χρόνο (π.χ. για την κατανάλωση ενέργειας, για την κατάσταση του εξοπλισμού) και με την αμφίπλευρη επικοινωνία προς κάθε σημείο του δικτύου μπορούν να αναπτυχθούν στρατηγικές βέλτιστου ελέγχου με στόχο τη βελτίωση της λειτουργίας, της αξιοπιστίας και της ευστάθειας του δικτύου [4], [13]-[15]. Βέβαια, η εκτεταμένη συλλογή δεδομένων στο ηλεκτρικό δίκτυο θα επεκτείνει την πρόσβαση σε δεδομένα για το ίδιο το δίκτυο και τους χρήστες του, γεγονός που εγείρει βασικές κοινωνικές ανησυχίες για την ασφάλεια στον κυβερνοχώρο (cyber security) [16], [17].

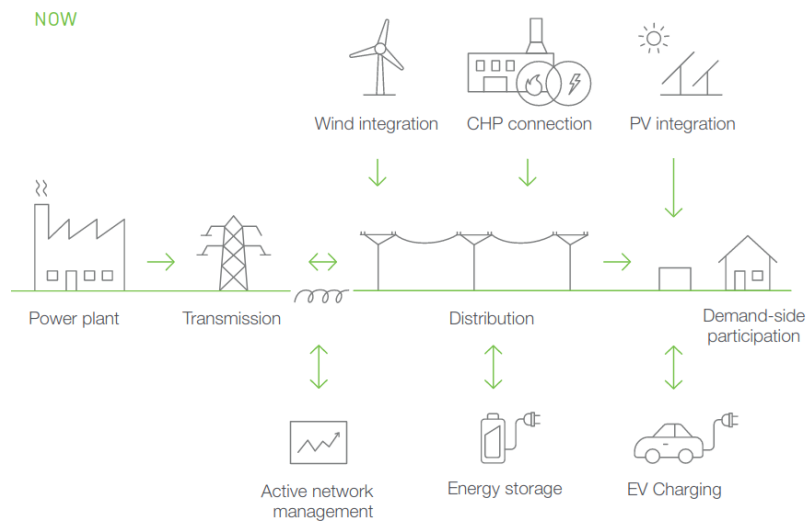
Όλα τα παραπάνω καταδεικνύουν ότι είναι απαραίτητος ο επαναπροσδιορισμός του τρόπου παραγωγής, μεταφοράς, διανομής και χρήσης της ηλεκτρικής ενέργειας με έναν πιο αποτελεσματικό, αποδοτικό, οικονομικό και περιβαλλοντικά βιώσιμο τρόπο. Έτσι λοιπόν, τα τελευταία χρόνια, υφίσταται μια μαζική μετάβαση προς τα λεγόμενα Ευφυή Δίκτυα (Smart Grids). Ο Tumballa et al. (2016) σημειώνει ότι το έξυπνο δίκτυο δεν έχει έναν παγκοσμίως αποδεκτό ορισμό, ωστόσο, με απλά λόγια, είναι ένα ευφυές σύστημα. Αποτελεί ένα επόμενης γενιάς δίκτυο, το οποίο παύει να είναι παθητικό και μετατρέπεται σε ένα ενεργό, δυναμικό, πολύπλοκο, ευφυές σύστημα, που περιλαμβάνει, εκτός από τα παραδοσιακά δίκτυα, ένα υψηλό ποσοστό ανανεώσιμης διεσπαρμένης παραγωγής και ευέλικτων φορτίων με αμφίδρομη μεταφορά ενέργειας, ΤΠΕ που επιτρέπουν ανίχνευση και επεξεργασία πληροφοριών, τεχνητή νοημοσύνη, αυτοματισμούς, έλεγχο και ένα νέο πολυσύνθετο μοντέλο αγοράς, με ενεργή συμμετοχή των καταναλωτών, με αμφίδρομη επικοινωνία σε πραγματικό χρόνο και

ροή μεγάλου όγκου πληροφοριών. Αυτό το μοντέρνο δίκτυο μπορεί να αποθηκεύει, να επικοινωνεί και να παίρνει αποφάσεις. Λειτουργεί πιο συνεργατικά, υπεύθυνα και οργανικά ώστε να βελτιστοποιήσει την απόδοση του συστήματος. Η διαφοροποίηση στον σχεδιασμό του ευφυούς δικτύου από το παραδοσιακό είναι ότι δίνει έμφαση στην αξιοπιστία, στην αποδοτικότητα και στην ασφάλεια αυξάνοντας τη γνώση. Καλύτερη πληροφόρηση οδηγεί σε πιο αποδοτική λειτουργία, ενώ παράλληλα η πιο σταθερή παροχή μειώνει το κόστος των καταναλωτών [6], [12], [15], [16], [18].

### Προκλήσεις υποδομής:

Το παραδοσιακό ΣΗΕ δυσκολεύεται να αντιμετωπίσει την απρόβλεπτη ανάπτυξη της υπολογιστικής ισχύος η οποία προέκυψε από τα σημαντικά βήματα προόδου που πραγματοποιήθηκαν τις τελευταίες δεκαετίες στους τομείς των ΤΠΕ. Καθώς η «ψηφιακή» κοινωνία μας εξελίσσεται ταχέως, η αυξανόμενη ανάγκη των βιομηχανικών και οικιακών πελατών εξαρτάται και απαιτεί παροχή ηλεκτρικής ενέργειας υψηλής ποιότητας και αξιοπιστίας μέσω της απόκτησης μιας πιο ανθεκτικής υποδομής δικτύου [5], [12].

Στο Σχήμα 1.2 παρουσιάζεται η τοπολογία του Ευφυούς Δικτύου. Το Ευφύες Δίκτυο μπορεί να γίνει καλύτερα κατανοητό όταν αντιπαραβάλλεται με το παραδοσιακό ΣΗΕ όπως φαίνεται στον Πίνακα 1.1.



Σχήμα 1. 2 : Το Ευφύες Δίκτυο [7].

<b>Παραδοσιακό ΣΗΕ</b>	<b>Ευφυές Δίκτυο</b>
Συγκεντρωμένη παραγωγή ενέργειας	Διεσπαρμένη παραγωγή ενέργειας
Λιγότερα δεδομένα	Μεγάλος όγκος δεδομένων
Μικρός αριθμός αισθητήρων	Πολλοί αισθητήρες
Λίγη ή καθόλου παρακολούθηση	Αυτόματη παρακολούθηση
Χειροκίνητος έλεγχος και επαναφορά	Αυτόματος έλεγχος και επαναφορά
Λιγότερα προβλήματα ασφάλειας και ιδιωτικότητας	Επιρρεπές σε προβλήματα ασφάλειας και ιδιωτικότητας
Χρειάζεται ανθρώπινους χειρισμούς σε περιπτώσεις διακοπών	Αυτοματοποιημένα συστήματα προστασίας
Άμεση παραγωγή και κατανάλωση ενέργειας	Χρήση συστημάτων αποθήκευσης
Περιορισμένος έλεγχος	Αυξημένα συστήματα ελέγχου
Αργή ανταπόκριση σε περίπτωση ανάγκης	Γρήγορη ανταπόκριση σε περίπτωση ανάγκης
Λιγότερες επιλογές των χρηστών	Πολλές επιλογές των χρηστών

Πίνακας 1.1 : Το παραδοσιακό ΣΗΕ σε σύγκριση με το Ευφυές Δίκτυο [6]  
 Παρακάτω παρουσιάζονται μερικές από τις πιο σημαντικές πτυχές του Ευφυούς Δικτύου:

### 1.1.1 Ενσωμάτωση των ΑΠΕ στο δίκτυο

Στο Ευφυές Δίκτυο, ενέργεια από διάφορες πηγές συνδυάζεται για να εξυπηρετήσει τις ανάγκες των καταναλωτών, ελαχιστοποιώντας τις επιπτώσεις της κλιματικής αλλαγής και μεγιστοποιώντας τη βιωσιμότητα. Επιπρόσθετα με τους συμβατικούς σταθμούς παραγωγής, ενέργεια παράγεται από τον ήλιο, τον άνεμο, τη βιομάζα, τη γεωθερμία, τα κύματα και άλλες ανανεώσιμες πηγές. Το έξυπνο δίκτυο υποστηρίζει όχι μόνο συγκεντρωμένη παραγωγή μεγάλης κλίμακας αλλά και μικρής κλίμακας κατανεμημένη παραγωγή, η οποία πρέπει να διανεμηθεί στο δίκτυο από διάφορες τοποθεσίες προς τους καταναλωτές. Γι' αυτό απαιτείται η ανάπτυξη νέων τεχνολογιών αυτοματισμού, ελέγχου και αμφίδρομης ροής ενέργειας στο ΣΗΕ. Επιπλέον, λόγω της μεταβλητότητας και της αβεβαιότητας της παραγωγής αυτών των μονάδων, χρειάζονται προηγμένες τεχνολογίες ώστε να ενσωματωθούν ομαλά στο κεντρικό δίκτυο, δηλαδή να ανταποκρίνεται η παραγωγή στη ζήτηση κάθε στιγμή με οικονομικά βιώσιμο και αποδοτικό τρόπο, με μικρές απώλειες και κίνδυνο αστάθειας καθώς και με υψηλό επίπεδο ποιότητας [5], [6], [18].

### 1.1.2 Αποθήκευση ενέργειας

Το Έξυπνο Δίκτυο διαθέτει πολυάριθμα κέντρα αποθήκευσης ενέργειας, στάσιμα και κινούμενα, τα οποία έχουν τη δυνατότητα να αποθηκεύουν ενέργεια και αντίστοιχα να παρέχουν την αποθηκευμένη ενέργεια. Αυτές οι μονάδες μπορούν να υποστηρίξουν το σύστημα βοηθώντας στη ρύθμιση τάσης και συχνότητας, εξασφαλίζοντας εφεδρεία ενέργειας, ανακουφίζοντας το σύστημα σε περίπτωση συμφόρησης, αντιμετωπίζοντας

τις επιπτώσεις των ξαφνικών εναλλαγών φορτίου και των διακυμάνσεων της ηλιακής και της αιολικής παραγωγής, μετατοπίζοντας την κατανάλωση ενέργειας από τις ώρες αιχμής με αντιστάθμιση ισχύος[4], [5], [18].

### **1.1.3 Ενσωμάτωση ηλεκτρικών οχημάτων**

Πολλά φορτία και πηγές που συνδέονται στο Ευφυές Δίκτυο δεν είναι στάσιμα. Εξελίξεις στην τεχνολογία των μπαταριών καθιστούν τα ηλεκτρικά οχήματα εμπορικά βιώσιμα. Αυτά τα οχήματα αντιπροσωπεύουν κινητά φορτία αλλά και πιθανές πηγές ενέργειας καθώς οι μπαταρίες στο εσωτερικό τους μπορούν να συνδεθούν οπουδήποτε στο ΣΗΕ και να φορτιστούν ή να χρησιμοποιηθούν για παροχή ενέργειας. Πρόκειται για την τεχνολογία vehicle-to-grid (V2G), η οποία μπορεί να βελτιώσει την απόδοση, την ευστάθεια και την αξιοπιστία του δικτύου, ωστόσο προκαλεί μεγαλύτερη πολυπλοκότητα στην κατανομή του φορτίου και στις ροές ισχύος [5], [13], [19].

### **1.1.4 Διεσπαρμένη παραγωγή μικρής κλίμακας και μικροδίκτυα**

Η διεσπαρμένη παραγωγή και τα μικροδίκτυα (microgrids) αντιπροσωπεύουν μια εναλλακτική αρχιτεκτονική για την οργάνωση του δικτύου. Πρόκειται για μονάδες σχετικά μικρής κλίμακας όπως ανεμογεννήτριες, φωτοβολταϊκά, αποθηκευτικά συστήματα, που δημιουργούν νησίδες με δυνατότητα αυτόνομης λειτουργίας. Δεν αποτελούν περιθωριακά στοιχεία του δικτύου, αντίθετα διαδραματίζουν σημαντικό ρόλο παρέχοντας ενεργειακή ποικιλότητα, μειώνοντας τη ζήτηση για κεντρικούς σταθμούς θερμικής παραγωγής και αυξάνοντας την εφεδρεία ενέργειας, την αξιοπιστία του συστήματος και την ανθεκτικότητά του σε περιπτώσεις εκτεταμένων διαταραχών. Επιπλέον, περιορίζουν την απώλεια ενέργειας, η οποία αποτελεί ένα σημαντικό ποσοστό της παραγόμενης ενέργειας, καθώς η παραγωγή βρίσκεται κοντά στο σημείο της κατανάλωσης και αποφεύγεται η μεταφορά της στις γραμμές μεγάλου μήκους [5], [11]

### **1.1.5 Διαμόρφωση της αγοράς και έξυπνος καταναλωτής**

Οι εξελίξεις στους τομείς των Ευφύων Δικτύων που αναφέρθηκαν μεταμορφώνουν και εκσυγχρονίζουν την αγορά ενέργειας καθώς και τα προϊόντα και τις υπηρεσίες που παρέχονται στους καταναλωτές. Για παράδειγμα με την ανάπτυξη των ΑΠΕ και την υποστήριξη της αμφίδρομης ροής ενέργειας στο δίκτυο, εάν ο καταναλωτής παράγει ηλεκτρική ενέργεια στο σπίτι του με τη χρήση φωτοβολταϊκών, μπορεί να παρέχει στο δίκτυο την ενέργεια που δεν καταναλώνει. Έτσι μια νέα αγορά εμφανίζεται στα Ευφυή Δίκτυα, όπου καταναλωτές και ομάδες καταναλωτών παίζουν ενεργό ρόλο στο ΣΗΕ: μπορούν από απλοί καταναλωτές να γίνουν ταυτόχρονα και παραγωγοί (prosumers είναι ο νέος όρος που έχει καθιερωθεί για να περιγράψει τον διττό αυτό ρόλο), να συνδιαμορφώνουν και να δημιουργούν ενεργειακές κοινότητες (συνεταιρισμούς) [12]. Επιπρόσθετα ο εξοπλισμός των καταναλωτών στο Ευφυές Δίκτυο αποτελείται από έξυπνες συσκευές που επικοινωνούν με άλλα σημεία του δικτύου. Ο καταναλωτής θα μπορεί να γνωρίζει κάθε στιγμή το μέγεθος της κατανάλωσής του και τις επιμέρους καταναλώσεις σημαντικών συσκευών. Με τέτοιου είδους λεπτομερή πληροφόρηση, ο καταναλωτής είναι πλέον σε θέση να προσαρμόσει την ενεργειακή συμπεριφορά του, μεγιστοποιώντας το οικονομικό του όφελος και υποστηρίζοντας το δίκτυο [5].

### 1.1.6 Ενσωμάτωση των ΤΠΕ και άλλων προηγμένων τεχνολογιών

Κλειδί στην επιτυχία των Ευφυών Δικτύων αποτελεί η συνεργασία και αλληλεπίδραση των ΣΗΕ με τις ΤΠΕ και άλλες προηγμένες τεχνολογίες ώστε να καλυφθούν οι μελλοντικές ανάγκες για μεγαλύτερη ευελιξία και αξιοπιστία στην παροχή και χρήση της ενέργειας. Τα δίκτυα επικοινωνιών και τα υπολογιστικά συστήματα καλούνται να διαδραματίσουν σημαντικό ρόλο στην αντιμετώπιση διάφορων προκλήσεων που υφίσταται το ΣΗΕ:

1. Η ανάγκη εφαρμογής διανεμημένων μηχανισμών ελέγχου ώστε διάφορα στοιχεία του δικτύου όπως πηγές, φορτία και αποθηκευτικές μονάδες να μπορούν να ελεγχθούν τοπικά ή να παίρνουν αποφάσεις.
2. Η ανάγκη για ακριβή πρόβλεψη της ζήτησης σε οποιοδήποτε σημείο του δικτύου λίγες ώρες ή μέρες πριν.
3. Η ανάγκη για ακριβή εκτίμηση της παραγωγής από ΑΠΕ όπως φωτοβολταϊκά και ανεμογεννήτριες, η οποία απαιτεί συσχέτιση με την πρόβλεψη του καιρού.
4. Η ανάγκη μείωσης της μέγιστης ζήτησης για την επίτευξη ενός πιο αποδοτικού δικτύου με μεθόδους όπως διακοπή φορτίου, ευφυή διαχείριση φορτίου, δυναμική τιμολόγηση.
5. Η ανάγκη για προηγμένες τεχνολογίες αποθήκευσης ενέργειας για την εξομάλυνση της ζήτησης κατά τις ώρες αιχμής.

Επιπλέον ο σχεδιασμός νέων αρχιτεκτονικών για τα Ευφυή Δίκτυα είναι απαραίτητος για την ομαλή ενσωμάτωση των ΣΗΕ με τις ΤΠΕ και άλλες καινοτόμες τεχνολογίες. Αυτό θα επιτρέψει τον συνδυασμό και την αλληλεπίδραση ενός πλήθους διασυνδεδεμένων ετερογενών στοιχείων που συνεργάζονται σε πραγματικό χρόνο ώστε να πετύχουν ένα νέο επίπεδο καταναμημένου ελέγχου, ενημέρωσης, επικοινωνίας, ευφυίας και ασφάλειας. Έξυπνοι μετρητές, αισθητήρες και προηγμένες εφαρμογές παρακολούθησης του δικτύου επιτρέπουν τη συγκέντρωση πολύ μεγάλου όγκου δεδομένων, σε πραγματικό χρόνο, σε όλη τη διεσπαρμένη δομή του δικτύου. Αυτές οι πληροφορίες διαδίδονται μέσω αμφίδρομων υποδομών επικοινωνίας σε οποιοδήποτε σημείο του δικτύου και πρέπει να επεξεργαστούν γρήγορα και αποδοτικά από τους αλγορίθμους που τρέχουν στα κέντρα ελέγχου, ώστε να μπορούν να χρησιμοποιηθούν για την πρόβλεψη του τι θα συμβεί στη συνέχεια, την λήψη αποφάσεων και την ανάπτυξη στρατηγικών βέλτιστου ελέγχου. Οι ΤΠΕ ασχολούνται επίσης με την επικοινωνία των σχετικών φορέων (παραγωγοί, καταναλωτές, φορείς εκμετάλλευσης δικτύων κλπ), για τη βελτιστοποίηση και την παρακολούθηση των διασυνδεδεμένων τμημάτων τους [15]. Χρειάζεται επιπλέον ενσωμάτωση μηχανισμών ασφαλείας για την προστασία των δεδομένων του δικτύου και όλων των χρηστών καθώς και για τον εντοπισμό και την πρόληψη πιθανών επιθέσεων [16], [17]. Συνοπτικά, οι ΤΠΕ βρίσκονται στην καρδιά όλων των λειτουργιών του Ευφυούς Δικτύου. Μπορούν να συνδέσουν όλα τα στοιχεία του επιτρέποντας την ανταλλαγή πληροφοριών, προηγμένη διαχείριση και έλεγχο του δικτύου.

## 1.2 Σύγχρονες τάσεις στην εκπαίδευση των ΣΗΕ

Ο μετασχηματισμός των παραδοσιακών ενεργειακών συστημάτων σε Ευφυή Δίκτυα ανοίγει νέους δρόμους και δημιουργεί νέες προκλήσεις στην εκπαίδευση και κατάρτιση των νέων μηχανικών. Η αυξανόμενη πολυπλοκότητα των ενεργειακών συστημάτων, με την ενσωμάτωση ανανεώσιμων και διεσπαρμένων μονάδων παραγωγής, επικοινωνίας, ελέγχου, πληροφορικής και άλλων προηγμένων τεχνολογιών, συνεπάγεται ένα περίπλοκο διεπιστημονικό πανόραμα που περιλαμβάνει εξελισσόμενα θέματα για τα οποία οι παραδοσιακοί Ηλεκτρολόγοι Μηχανικοί και Μηχανικοί Υπολογιστών δεν είναι εξ ολοκλήρου έτοιμοι. Η κλασική εκπαίδευση ηλεκτρικής ενέργειας συνήθως δεν καλύπτει επαρκώς τα θέματα Ευφυών Δικτύων, θέτοντας προκλήσεις σε νέους ερευνητές και φοιτητές αλλά και στη βιομηχανία [20],[21].

Δεδομένου ότι η υλοποίηση των Ευφυών Δικτύων είναι ακόμα ένα εξελισσόμενο πεδίο, απαιτούνται επαγγελματίες υψηλής ειδίκευσης για να απορροφηθούν στους τομείς της έρευνας, της εργασίας, της καινοτομίας σήμερα και μελλοντικά. Οι επόμενες γενιές μηχανικών θα καλεστούν να καθορίσουν πλήρως την έννοια του Ευφυούς Δικτύου, να το εφαρμόσουν και να το αναπτύξουν περαιτέρω. Χρειάζονται λοιπόν υψηλού επιπέδου μηχανικοί με ευρεία κατανόηση των προαναφερθέντων θεμάτων, εμπειρία και διαφορετικό προφίλ. Αυτό οδηγεί στην ανάγκη μετασχηματισμού του εκπαιδευτικού ρόλου των προπτυχιακών προγραμμάτων, που καλούνται τώρα να ενισχύσουν τις ερευνητικές ικανότητες των φοιτητών. Η εισαγωγή νέων μεθοδολογιών διδασκαλίας και αξιολόγησης που αποσκοπούν στη δημιουργία κριτικής σκέψης και ευρηματικότητας των σημερινών και μελλοντικών ερευνητών και μηχανικών καθίσταται ολοένα και περισσότερο απαραίτητη ώστε αυτοί να είναι σε θέση να κατανοήσουν και να αντιμετωπίσουν τις προκλήσεις καθώς και να προτείνουν και να εφαρμόσουν νέες μεθόδους. Αυτό δίνει περισσότερη έμφαση στη σημασία ανάπτυξης δεξιοτήτων, παρά στην παραδοσιακή μετάδοση γνώσης. Τα τρέχοντα θέματα δεν μπορούν να αντιμετωπιστούν μόνο μέσω «τυποποιημένων» γνωστών λύσεων. Η διδασκαλία τους θα πρέπει να βασίζεται και στην ενίσχυση της κριτικής σκέψης και των ερευνητικών ικανοτήτων, αντί να προσφέρει μια συμβατική προσέγγιση στην επίλυση προβλημάτων, διότι, στην περίπτωση αυτή δεν διαθέτουν ήδη μια «τυποποιημένη» λύση [20], [21].

Σε αυτό το πλαίσιο, δεδομένου του διεπιστημονικού χαρακτήρα και της ολοένα αυξανόμενης πολυπλοκότητας των ενεργειακών συστημάτων, όπως περιγράφεται παραπάνω, πρέπει να αναπτυχθούν νέα εκπαιδευτικά προγράμματα, μέθοδοι και εργαλεία, ικανά να συνδυάσουν τους διαφορετικούς τομείς γνώσης και να επιτρέψουν στους φοιτητές να κατανοήσουν τη σύνδεση και την αλληλεπίδραση των διάφορων ετερογενών στοιχείων μέσα στο Ευφύες Δίκτυο. Πολλά πανεπιστήμια έχουν ενσωματώσει νέα μαθήματα στο προπτυχιακό πρόγραμμα σπουδών ή εμπλούτισαν τα υπάρχοντα μαθήματά τους με πιο σύγχρονο υλικό. Ορισμένα πανεπιστήμια έχουν επίσης δημιουργήσει μεταπτυχιακά προγράμματα σπουδών σε σχετικά θέματα. Η διδασκαλία εκτελείται με παραδοσιακές μεθόδους, όπως διαλέξεις στο αμφιθέατρο, αλλά συνοδεύεται και από προγραμματιστικά εργαλεία, προηγμένες προσομοιώσεις (π.χ. Co-simulation) και εργαστηριακές ασκήσεις (π.χ. hardware in the loop simulation), όπου εκπαιδευτικές μέθοδοι όπως η ενεργός μάθηση, η μάθηση με βάση

το πρόβλημα και η βιωματική μάθηση εφαρμόζονται ολοένα και περισσότερο [20], [22], [84].

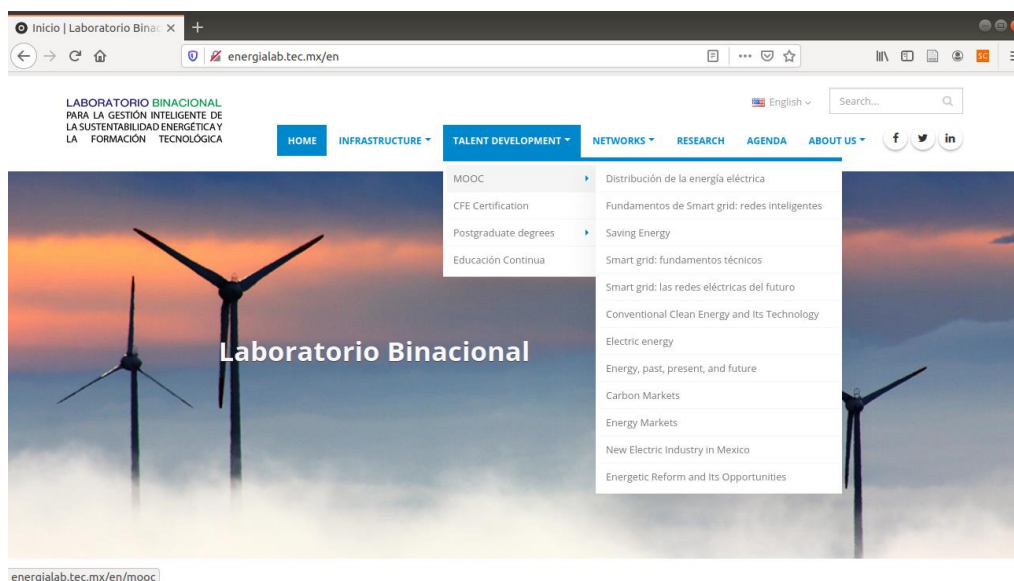
Παράλληλα, η σύγχρονη τεχνολογική πρόοδος παρέχει ένα πλήθος νέων εργαλείων και εκπαιδευτικών δυνατοτήτων, περιβαλλόντων, πλατφορμών και χώρων για προσωπική, συνεργατική, τυπική και άτυπη μάθηση. Μερικά από τα μαθησιακά περιβάλλοντα που μπορούν να επιφέρουν καθοριστικές αλλαγές στην εκπαίδευση των μηχανικών αφορούν την ηλεκτρονική μάθηση (e-learning) με την ανάπτυξη Μαζικών Ανοικτών Ηλεκτρονικών Μαθημάτων (MOOCs), η οποία μεταμορφώνει την παραδοσιακή διδασκαλία, καθώς τα όρια μεταξύ αίθουσας και εργαστηριακής εκπαίδευσης γίνονται πιο ευέλικτα. Στο ίδιο πλαίσιο οι ανοιχτοί εκπαιδευτικοί πόροι (open educational resources) και η χρήση συνεργατικών (collaborative) εργαλείων μπορούν να συνεισφέρουν στην υποστήριξη της διδασκαλίας και στη βελτίωση της εκπαιδευτικής διαδικασίας. Ένα τέτοιο παράδειγμα αποτελούν τα διαδραστικά (interactive) Notebooks, εργαλεία που παρέχουν ένα προγραμματιστικό περιβάλλον για την ανάπτυξη επιστημονικού περιεχομένου και προωθούν την ελεύθερη πρόσβαση σε αυτό. Επιπλέον, νέοι τρόποι εργαστηριακής εκπαίδευσης προκύπτουν, οι οποίοι διευκολύνουν τους φοιτητές να κατανοήσουν τη σύνδεση της θεωρίας και της εφαρμογής όπως η βιωματική εκμάθηση μέσω της μεθόδου Επαυξημένης Πραγματικότητας (Augmented Reality), που δημιουργεί δυναμικές μαθησιακές εμπειρίες και ενισχύει την αφομοίωση του εκπαιδευτικού περιεχομένου. Οι παραπάνω εκπαιδευτικές μέθοδοι μπορούν να συνδυαστούν μεταξύ τους, για παράδειγμα η διδασκαλία ενός MOOC μαθήματος μπορεί να υποστηριχθεί από διαδραστικά Notebooks, ενώ απομακρυσμένα εργαστηριακά πειράματα (remote-lab experiments) μπορούν να αξιοποιήσουν διαδραστικά Notebooks ή επαυξημένη πραγματικότητα για την πραγματοποίηση προσομοιώσεων, δημιουργώντας έτσι καινοτόμες εκπαιδευτικές εμπειρίες για τους νέους μηχανικούς δίνοντας τους την ευκαιρία να επικοινωνούν και να συνεργάζονται για την εφαρμογή και δοκιμή της θεωρητικής γνώσης στην πράξη, σε παγκοσμίως διανεμημένες ομάδες, όπως συμβαίνει συχνά στη βιομηχανία. Όπως γίνεται αντιληπτό λοιπόν, οι διαδικασίες της διδασκαλίας και της μάθησης σε διάφορα μαθήματα της προπτυχιακής εκπαίδευσης των μηχανικών, μπορούν να επωφεληθούν δραστικά από αυτού του είδους τους πόρους [20], [23]-[27]. Παρακάτω παρουσιάζονται συνοπτικά αυτές οι κατηγορίες μαθησιακών περιβαλλόντων.

### 1.2.1 Massive Open Online Courses (MOOCs)

Μια από τις σύγχρονες εκπαιδευτικές τάσεις που κερδίζει έδαφος τα τελευταία χρόνια είναι τα Μαζικά Ανοικτά Ηλεκτρονικά Μαθήματα (MOOCs) που διατίθενται από πανεπιστήμια σε όλο τον κόσμο. Αυτή η μορφή e-learning μπορεί να συμβάλει στην εκπαίδευση και κατάρτιση φοιτητών, νέων μηχανικών, ερευνητών και επαγγελματιών που απασχολούνται σε τομείς των ευφυών ενεργειακών συστημάτων δίνοντας ελεύθερη πρόσβαση σε εκπαιδευτικό υλικό χωρίς περιορισμό προαπαιτούμενων γνώσεων, ηλικίας, οικονομικής κατάστασης ή γεωγραφικής τοποθεσίας. Είναι γεγονός ότι το υπάρχον υλικό, όπως βιβλία και εγχειρίδια, τείνουν να απαιτούν πολύ χρόνο για την ανάγνωση και κατανόηση του περιεχομένου τους και συχνά είναι πολύ θεωρητικά για τις ανάγκες των ενδιαφερόμενων. Επιπλέον, πολλές διαθέσιμες πληροφορίες που υπάρχουν στο διαδίκτυο μπορεί να μην έχουν αξιοπιστία και συνήθως δεν είναι αρκετά δομημένες για να παρέχουν μια ολοκληρωμένη και συνεπή εικόνα ενός πολύπλευρου θέματος, όπως τα Ευφυή Δίκτυα [28]. Τα MOOCs λοιπόν, μπορούν να καλύψουν με

ευέλικτο τρόπο εκπαιδευτικά κενά που προκύπτουν λόγω της συνεχούς εξέλιξης στον τομέα των ευφυών ενεργειακών συστημάτων.

Η οργάνωση του περιεχομένου αυτών των μαθημάτων εξαρτάται από τα κίνητρα και τους στόχους των εκπαιδευτικών που τα παραδίδουν, και συχνά έχουν την πρόθεση να μεταφέρουν στο κοινό τους μια πιο γενική εποπτεία του αντικειμένου που εξετάζουν χρησιμοποιώντας διάφορες καινοτόμες παιδαγωγικές μεθόδους, όπως η ανεστραμμένη τάξη (flipped classroom), η μάθηση με βάση την πρόκληση (challenge-based learning) και η «παιχνιδοποίηση» (gamification). Το υλικό διανέμεται μέσω κάποιας ηλεκτρονικής πλατφόρμας (π.χ. Coursera, edX) και μπορεί να περιλαμβάνει οπτικοακουστικό υλικό, σημειώσεις διαλέξεων, ερωτηματολόγια και εργασίες για τους φοιτητές, οι οποίοι μπορούν να επικοινωνούν μεταξύ τους και με τους εκπαιδευτικούς μέσω φόρουμ κ.ά. [28], [29]. Αυτό το εκπαιδευτικό περιβάλλον μπορεί να επιτρέψει στους μηχανικούς των ΣΗΕ και των ΤΠΕ, ανεξάρτητα με τον τομέα εξειδίκευσής τους, να μοιραστούν τις γνώσεις και να αποκτήσουν μια πιο συνολική αντίληψη για τα ευφυή ενεργειακά συστήματα. Το Σχήμα 1.3 παρουσιάζει ενδεικτικά την ιστοσελίδα του Binational Laboratory για την Ευφυή Διαχείριση της Ενεργειακής Βιωσιμότητας και της Τεχνολογικής Εκπαίδευσης, όπου είναι διαθέσιμα διάφορα MOOCs που αφορούν ποικίλα θέματα Έξυπνων Δικτύων [28].



Σχήμα 1. 3 : Ιστοσελίδα του Binational Laboratory για την Ευφυή Διαχείριση της Ενεργειακής Βιωσιμότητας και της Τεχνολογικής Εκπαίδευσης (<http://energiablab.tec.mx/>) [28].

## 1.2.2 Διαδραστικά Notebooks (Jupyter Notebooks)

Στο ίδιο πλαίσιο, εργαστηριακά μαθήματα που χρησιμοποιούν προγραμματιστικά πειραματικά περιβάλλοντα, όπως τα διαδραστικά Notebooks, παίζουν σημαντικό ρόλο, όχι μόνο γιατί βοηθούν στην εξοικείωση των νέων μηχανικών με έννοιες των ΤΠΕ μέσω της απόκτησης προγραμματιστικών και υπολογιστικών γνώσεων και δεξιοτήτων με φιλικό προς τον χρήστη τρόπο, αλλά και γιατί προσφέρουν τη δυνατότητα



μετάδοσης δεξιοτήτων διαδραστικής εξερεύνησης και πειραματισμού βάσει δεδομένων και αποτελεσμάτων, οι οποίες είναι πολύτιμες για τους μελλοντικούς ερευνητές και επαγγελματίες μηχανικούς.

Παρόλο που υπάρχουν διάφορες πλατφόρμες που παρέχουν τέτοιου είδους διαδικτυακούς πόρους, το Jupyter Notebook είναι ένα από τα εργαλεία που αποκτά ευρύτερη εφαρμογή στη διδασκαλία τα τελευταία χρόνια [30]. Πρόκειται για ένα εργαλείο ανοιχτού κώδικα που συνδυάζει αφηγηματικό κείμενο με εικόνες, μαθηματικές εξισώσεις, εκτελέσιμο κώδικα και τα αποτελέσματα του κώδικα αυτού. Προσφέρει τη δυνατότητα δημιουργίας ενός υπολογιστικού εγγράφου, για διαδραστική εξερεύνηση δεδομένων και επιστημονικούς υπολογισμούς σε διάφορες γλώσσες προγραμματισμού, που μπορεί εύκολα να διαβαστεί και να αναπαραχθεί [31].

The screenshot shows a Jupyter Notebook page with the following content:

**Enunciado**  
 Ante un fallo fase-fase-tierra (diftásico a tierra) en a barra 6 de la figura calcúe la tensión en la barra 5 y la corriente por el neutro del transformador T1.  
 Tome como datos:

- Generadores G1, G2:  $X_0 = 0.09 \text{ pu}$ ,  $X_1 = X_2 = 0.35 \text{ pu}$ .
- Transformadores T1 y T2:  $X_0 = X_1 = X_2 = X_n = 0.05 \text{ pu}$ .
- Transformador T3:  $X_0 = X_1 = X_2 = X_n = 0.03 \text{ pu}$ .
- Línea L42:  $X_0 = 1.80 \text{ pu}$ ,  $X_1 = X_2 = 0.45 \text{ pu}$ .
- Líneas L25 y L45:  $X_0 = 1.15 \text{ pu}$ ,  $X_1 = X_2 = 0.35 \text{ pu}$ .

**Sistema eléctrico:**

The diagram shows a power system with six buses (1-6). Bus 1 is connected to generator G1 and transformer T1. Bus 2 is connected to transformer T1 and line L25. Bus 3 is connected to generator G2 and transformer T2. Bus 4 is connected to transformer T2 and line L42. Bus 5 is connected to line L25, line L45, and transformer T3. Bus 6 is connected to transformer T3 and has a fault symbol. A fault impedance of  $j0.03$  is shown between bus 5 and bus 6. The fault is a phase-to-phase-to-ground fault.

**Resolución**

```
In [1]: # Importar librerías genéricas:
import numpy as np
import casadi as cs
import matplotlib.pyplot as plt

# Formato general de salida de resultados numéricos:
np.set_printoptions(precision=3, suppress=True)
```

**Hipótesis**

1. Dado que no se especifican los índices horarios de los transformadores, estos se despreciarán.
2. Dado que no se especifica el valor de la impedancia de falta a tierra, ni de la impedancia de contacto entre fases, se tomará como defecto franco o rígido tanto entre fases como con tierra.
3. Todas las impedancias de puesta a tierra de neutro que no se especificquen se considerarán nulas (puesta a tierra rígida).

Σχήμα 1. 4 : Jupyter Notebooks για την επίλυση ασκήσεων στα ΣΗΕ (<https://github.com/miguel-simon/SEP>).

Αυτός ο συνδυασμός περιεχομένου και κώδικα δημιουργεί μια ισχυρή νέα μορφή επικοινωνίας που βασίζεται στην αλληλεπίδραση με τα δεδομένα και καθιστά το Jupyter Notebook ένα ελκυστικό μαθησιακό περιβάλλον [32]. Η χρήση αυτού του είδους πόρων, όπου το κείμενο, οι εικόνες, ο κώδικας και τα αποτελέσματά του μπορούν να συνδυαστούν με αρμονικό και κατανοητό τρόπο, γεφυρώνει το χάσμα μεταξύ θεωρίας και εφαρμογής και μπορεί να συμβάλει στη διερεύνηση καινοτόμων προσεγγίσεων διδασκαλίας και μάθησης, σε διάφορα μαθήματα της ανώτερης εκπαίδευσης, ιδιαίτερα σε μαθήματα μηχανικών.

Σε ότι αφορά τους νέους ηλεκτρολόγους μηχανικούς, που χρειάζεται να εκπαιδευτούν περισσότερο σε τομείς των ΤΠΕ ώστε να μπορούν να ανταποκριθούν στα νέα δεδομένα, το Jupyter Notebook αποτελεί ένα εργαλείο ιδανικό για την εξοικείωση χρηστών, με μικρή ή καθόλου εμπειρία στον προγραμματισμό, όχι μόνο σε υπολογιστικές προσομοιώσεις που μπορούν να βοηθήσουν στην καλύτερη κατανόηση της λειτουργίας των ΣΗΕ, αλλά και σε θέματα ανάλυσης δεδομένων, τεχνητής νοημοσύνης, μηχανικής μάθησης, cyber security, τα οποία διέπουν πολλές πτυχές της λειτουργίας του έξυπνου δικτύου. Στο Σχήμα 1.4 φαίνεται ένα παράδειγμα της δομής ενός διαδραστικού Jupyter Notebook σχετικό με εκπαιδευτικό υλικό μαθήματος των ΣΗΕ, που μπορεί να περιλαμβάνει εκφώνηση, επεξηγηματικό κείμενο, εικόνες, κώδικα για τους υπολογισμούς και τα αποτελέσματα του κώδικα αυτού, σε ένα έγγραφο.

### 1.2.3 Επαυξημένη πραγματικότητα (Augmented reality)

Ένα ακόμη εργαλείο που μπορεί να βοηθήσει τους νέους μηχανικούς στην πιο σφαιρική κατανόηση των διαφορετικών πτυχών των ευφυών ενεργειακών συστημάτων και να φέρει πιο κοντά τις διαφορετικές ειδικότητες επαγγελματιών που εμπλέκονται σε αυτό, είναι η επαυξημένη πραγματικότητα. Αυτή η τεχνική μειώνει το χάσμα μεταξύ του εικονικού και του πραγματικού κόσμου, καθώς το πραγματικό τρισδιάστατο περιβάλλον αναμειγνύεται σε πραγματικό χρόνο με εικονικά αντικείμενα που δημιουργούνται μέσω υπολογιστικών προγραμμάτων, για τη δημιουργία μιας νέας διαδραστικής εμπειρίας για τον χρήστη.

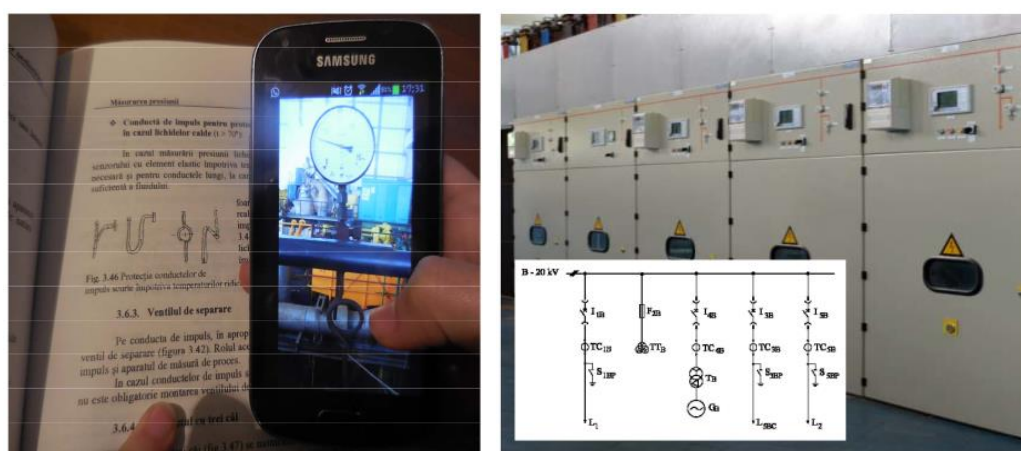
Η εισαγωγή σε ένα επαυξημένο περιβάλλον πραγματοποιείται μέσω κάποιας συσκευής όπως υπολογιστή, tablet, έξυπνο τηλέφωνο ή ειδικά γυαλιά. Ένα βασικό χαρακτηριστικό της επαυξημένης πραγματικότητας είναι η ικανότητά της να ανταποκρίνεται στην είσοδο του χρήστη και να παρέχει πολύ περισσότερες λεπτομέρειες και πληροφορίες σχετικά με στοιχεία του πραγματικού κόσμου σε σύγκριση με την παραδοσιακή προσέγγιση μάθησης στο αμφιθέατρο. Δυναμικές διεργασίες, μεγάλος όγκος δεδομένων, πολύ μεγάλα ή πολύ μικρά αντικείμενα μπορούν να μεταφερθούν στον προσωπικό χώρο του φοιτητή σε μια κλίμακα και σε μια μορφή, την οποία μπορεί εύκολα να κατανοήσει και να επεξεργαστεί. Έτσι αυξάνεται το ενδιαφέρον του για το διδασκόμενο αντικείμενο, δημιουργούνται δυναμικές μαθησιακές εμπειρίες και ενισχύεται η αφομοίωση του εκπαιδευτικού περιεχομένου [23], [24], [27], [33].

Στον τομέα της εκπαίδευσης των ΣΗΕ, η επαυξημένη πραγματικότητα μπορεί να συνεισφέρει σημαντικά παρουσιάζοντας τις πληροφορίες που είναι διαθέσιμες σε κάποιο σημείο του εξοπλισμού ή εμπλουτίζοντας το μέρος ή το αντικείμενο προς μελέτη με πρόσθετες πληροφορίες, εικόνες, οδηγίες με φιλικό προς τον χρήστη τρόπο. Παρακάτω αναφέρονται μερικά παραδείγματα αξιοποίησης της επαυξημένης πραγματικότητας στην εκπαίδευση των ΣΗΕ, που χρησιμοποιούνται μεταξύ άλλων για την εκπαίδευση νέων τεχνικών στη βιομηχανία:

- Παροχή των διαθέσιμων δεδομένων από μετρήσεις που πραγματοποιούνται σε πραγματικό χρόνο.
- Τρισδιάστατη επί τόπου οπτικοποίηση του εξοπλισμού και των συστημάτων.
- Βήμα προς βήμα καθοδήγηση για ενέργειες που πρέπει να πραγματοποιηθούν.

- Βοήθεια για τη λήψη αποφάσεων.
- Παροχή των αποτελεσμάτων από εφαρμογές Web SCADA.
- Πρόγνωση της μελλοντικής εξέλιξης των συστημάτων.
- Βοήθεια για την ανίχνευση σφαλμάτων και τη διαχείριση διακοπής λειτουργίας
- Δυνατότητα αποτελεσματικής επικοινωνίας [33].

Στο Σχήμα 1.5 παρουσιάζονται δύο παραδείγματα αξιοποίησης της επαυξημένης πραγματικότητας στην εκπαίδευση των ΣΗΕ. Στην πρώτη εικόνα φαίνεται πως μπορεί να χρησιμοποιηθεί σε συνδυασμό με κάποιο βιβλίο ή εγχειρίδιο και στη δεύτερη πως μπορεί να παρέχει επί τόπου πληροφορίες για το ηλεκτρολογικό σχέδιο ενός υποσταθμού [24], [33].



Σχήμα 1. 5 :Εφαρμογές της επαυξημένης πραγματικότητας στην εκπαίδευση των ΣΗΕ [33].

### 1.3 Αντικείμενο της εργασίας

Όπως αναλύθηκε στο παρόν κεφάλαιο, η ψηφιακή εποχή θέτει νέες προκλήσεις και η βιομηχανία απαιτεί δεξιότητες όπως ευελιξία, κριτική σκέψη, ικανότητα επίλυσης προβλημάτων, συνεργασία και επικοινωνία. Γνωρίζοντας το ρόλο και την σημασία των ΤΠΕ, η εκπαίδευση των νέων μηχανικών πρέπει να προσαρμοστεί σε αυτή τη νέα εποχή καθιστώντας τους ικανούς να εργαστούν σε ένα δυναμικό περιβάλλον και να εξελίσσονται σύμφωνα με τις συνεχείς αλλαγές, συμπεριλαμβανομένης της χρήσης των σύγχρονων ΤΠΕ.

Όσον αφορά τα σύγχρονα ΣΗΕ, η εισαγωγή των ΤΠΕ και άλλων προηγμένων τεχνολογιών, έχει οδηγήσει σε μεγάλη αύξηση του εξοπλισμού που είναι «έξυπνος». Αυτό έχει σημαντικές επιπτώσεις στην κατανόηση των λειτουργιών του συστήματος από τους ερευνητές και τους επαγγελματίες των ΣΗΕ, που δεν έχουν πλήρη γνώση αυτών των θεμάτων. Από την άλλη πλευρά οι επαγγελματίες των ΤΠΕ συχνά δυσκολεύονται να κατανοήσουν τη λειτουργία του ΣΗΕ. Καθώς αυτοί οι τομείς, μεταξύ άλλων, συνδέονται στενά λόγω της εμφάνισης των ευφύων συστημάτων ενέργειας, είναι σημαντικό να δημιουργηθούν συνδέσεις μεταξύ τους. Φυσικά είναι δύσκολο να επιτευχθεί μια διεξοδική κατανόηση όλων των τομέων, ωστόσο η κατανόηση των θεμελιωδών στοιχείων κάθε περιοχής, χωρίς να θυσιάζεται η εξειδίκευση σε κάθε συγκεκριμένο τομέα θα γίνει όλο και πιο αναγκαία [20].

Ως εκ τούτου, σε μια προσπάθεια να εκτεθούν περισσότερο οι προπτυχιακοί φοιτητές της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών (ΣΗΜΜΥ) του ΕΜΠ, σε μεθόδους που προωθούν τους παραπάνω στόχους, επιλέξαμε να δημιουργήσουμε εκπαιδευτικό υλικό που αφορά θέματα Ανάλυσης Συστημάτων Ηλεκτρικής Ενέργειας, στο Jupyter Notebook. Η επιλογή αυτής της πλατφόρμας κινητοποιείται από το γεγονός ότι το Jupyter Notebook είναι ένα ανοικτό, διαδραστικό, υπολογιστικό περιβάλλον, το οποίο συνδυάζει ζωντανό κώδικα και τα αποτελέσματά του με επεξηγηματικό κείμενο και οπτικοποιήσεις σε ένα μοναδικό αρχείο [32]. Αποτελεί ένα πολύ χρήσιμο εργαλείο στην πανεπιστημιακή εκπαίδευση και ιδιαίτερα στους κλάδους των μηχανικών, καθώς επιτρέπει τη σύνδεση των φυσικών και μαθηματικών εννοιών με πρακτικές εφαρμογές και τη συσχέτισή τους με τον πραγματικό κόσμο. Είναι επίσης ιδανικό για τον σχεδιασμό εκπαιδευτικών σεναρίων, καθώς μπορεί να συνεισφέρει στην εξερεύνηση καινοτόμων προσεγγίσεων και να βελτιώσει τις εκπαιδευτικές δραστηριότητες [32], [34]-[36]. Με αυτόν τον τρόπο μεταφέρει τους φοιτητές από ένα παθητικό και στατικό μαθησιακό περιβάλλον, σε μια ενεργή μαθησιακή εμπειρία επιτρέποντάς τους να αλληλεπιδράσουν με το περιεχόμενο, να σκεφτούν κριτικά, να κατανοήσουν καλύτερα τη μεθοδολογία και να είναι σε θέση να την εφαρμόσουν.

Παράλληλα το Jupyter επιτρέπει στους ενεργειακούς φοιτητές να πειραματιστούν με ένα εργαλείο που βοηθά στην ανάπτυξη θεμελιωδών υπολογιστικών δεξιοτήτων καθώς και πρακτικών Επιστήμης των Δεδομένων, μοντελοποίησης, προσομοίωσης, συστημάτων και επίλυσης προβλημάτων. Με αυτόν τον τρόπο τους δίνει την ευκαιρία να εξοικειωθούν περισσότερο με μεθόδους και διαδικασίες που αφορούν το πεδίο των ΤΠΕ, προετοιμάζοντάς τους για μελλοντικές προκλήσεις που θα καλεστούν να

αντιμετωπίσουν ως φοιτητές, ερευνητές και επαγγελματίες του τομέα των ευφών ενεργειακών συστημάτων.

## Κεφάλαιο 2: Αξιοποίηση των Jupyter Notebooks στην εκπαιδευτική διαδικασία

### Σύνοψη

Οι τεχνολογικές εξελίξεις στο πεδίο των ευφύων ενεργειακών συστημάτων επιτάσσουν την υποστήριξη της εκπαιδευτικής διαδικασίας των νέων μηχανικών με νέες πρακτικές και εργαλεία, ώστε οι φοιτητές να αποκτήσουν μια πολύπλευρη εκπαίδευση που ενισχύει μεταξύ άλλων την κριτική σκέψη, τις ερευνητικές ικανότητες και στοχεύει στην καλλιέργεια των δεξιοτήτων της ευελιξίας και της προσαρμοστικότητας στα συνεχώς μεταβαλλόμενα δεδομένα της εποχής μας, λαμβάνοντας υπόψη τον εξελισσόμενο ρόλο των ΤΠΕ. Ένα υποσχόμενο εργαλείο είναι το Jupyter Notebook, μία διαδραστική πλατφόρμα που παρέχει ένα περιβάλλον ανάπτυξης εκπαιδευτικού υλικού, συνδυάζοντας κείμενο, εικόνες και κώδικα, σε ένα ενιαίο έγγραφο, προσβάσιμο μέσω του διαδικτύου. Στο παρόν κεφάλαιο παρουσιάζονται τα χαρακτηριστικά του εργαλείου αυτού καθώς και τα πλεονεκτήματα της χρήσης του ως εκπαιδευτικό περιβάλλον. Επίσης υποδεικνύονται αντιπροσωπευτικά παραδείγματα αξιοποίησης του στην εκπαιδευτική διαδικασία και συγκεκριμένα σε μαθήματα Ηλεκτρολόγων Μηχανικών. Τέλος, περιγράφονται οι εναλλακτικοί τρόποι με τους οποίους οι φοιτητές μπορούν να αποκτήσουν πρόσβαση σε αυτό και αναφέρονται χρήσιμες προεκτάσεις που διευκολύνουν τη χρήση του.

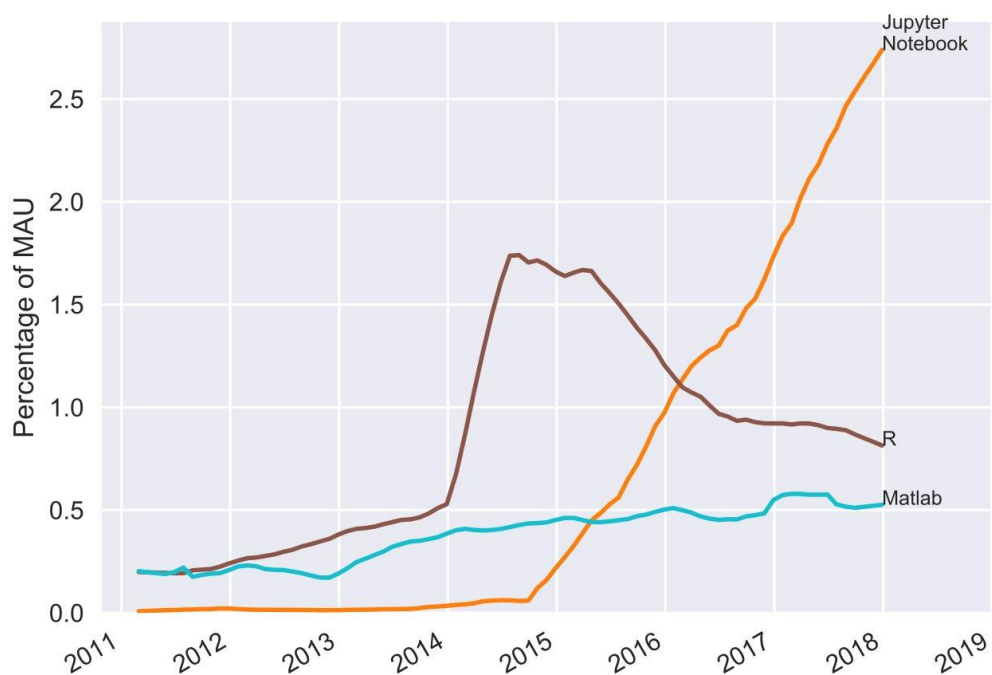
### 2.1 Τι είναι το Jupyter Notebook

Οι δημιουργοί του Jupyter Notebook [38] το περιγράφουν ως ένα σύνολο εργαλείων ανοικτού κώδικα (open-source web application) που επιτρέπει τη διαδραστική και πειραματική επεξεργασία δεδομένων και ως μια πλατφόρμα για τη δημιουργία υπολογιστικών αφηγήσεων. Αποτελεί την εξέλιξη της IPython [39], ενός εργαλείου για διαδραστική χρήση της γλώσσας Python, η ιδέα του οποίου βασίζεται σε προϋπάρχοντα επιστημονικά υπολογιστικά λογισμικά, όπως το Mathematica. Υποστηρίζει περισσότερες από 100 γλώσσες προγραμματισμού, με έμφαση στην Python, και περιλαμβάνει ένα οικοσύστημα προεκτάσεων για το οποίο έχει συμβάλλει μια μεγάλη κοινότητα.



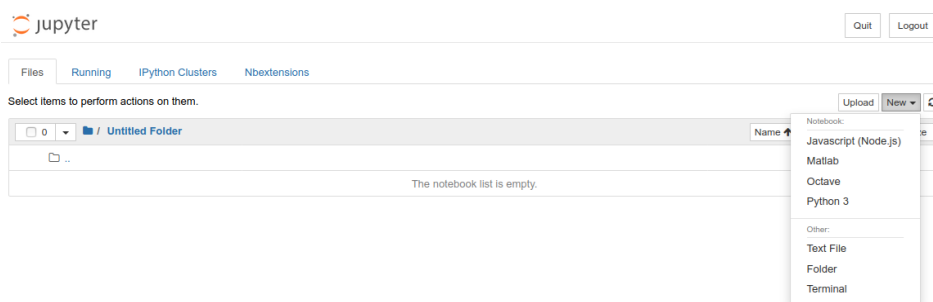
Σχήμα 2. 1 : Το όνομα του Jupyter είναι μια αναφορά στις τρεις βασικές γλώσσες προγραμματισμού που υποστηρίζονται από αυτό, οι οποίες είναι οι Julia, Python και R [32].

Η χρήση της πλατφόρμας έχει αυξηθεί σημαντικά τα τελευταία χρόνια σε σύγκριση με άλλα δημοφιλή υπολογιστικά πακέτα (π.χ. Matlab και R, βλ. Σχήμα 2.2), γεγονός που συνδέεται στενά με την ανάπτυξη της Python.



Σχήμα 2. 2 : Οι μηνιαίοι ενεργοί χρήστες (monthly active users) του Jupyter Notebook και δύο άλλων δημοφιλών υπολογιστικών εργαλείων [82].

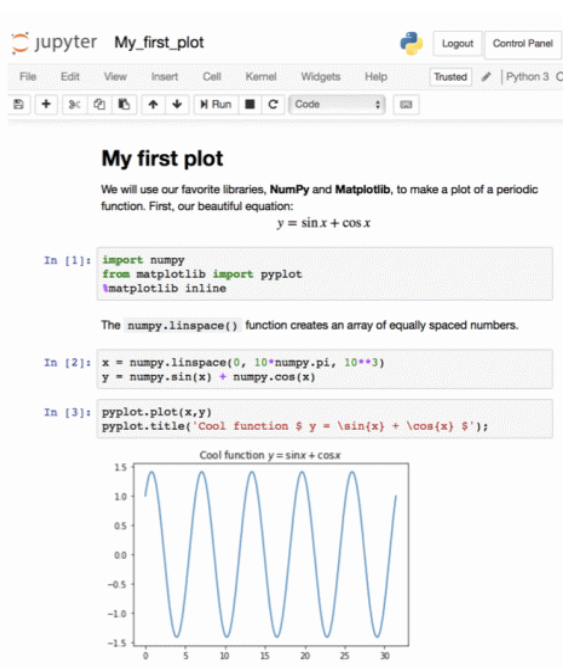
Το Jupyter Notebook είναι ένα έγγραφο, που υποστηρίζει την ανάμειξη δεδομένων, ζωντανού κώδικα (live code), εξισώσεων, οπτικοποιημένου υλικού (εικόνες, βίντεο) και αφηγηματικού κειμένου. Ο κώδικας είναι οργανωμένος σε κελιά (code cells), τα οποία μπορούν να επεξεργαστούν και να εκτελεστούν μεμονωμένα. Η έξοδος από κάθε κελί εμφανίζεται ακριβώς κάτω από αυτό και αποθηκεύεται ως μέρος του εγγράφου. Τα Jupyter Notebooks μπορούν να εμφανίσουν γραφικές παραστάσεις, μαθηματικές εξισώσεις (χρησιμοποιώντας την σύνταξη LaTeX) και διαδραστικά στοιχεία παραμέτρων εισόδου χρήστη (widgets). Το κείμενο μπορεί να παρεμβληθεί με τον κώδικα και τα αποτελέσματά του ώστε να εξηγήσει και να τονίσει συγκεκριμένα μέρη, μέσω των κελιών κειμένου (markdown cells), σχηματίζοντας μια πλούσια υπολογιστική αφήγηση [31]. Τα αρχεία Jupyter αποθηκεύονται με την επέκταση .ipynb.



Σχήμα 2. 3 : Η αρχική σελίδα του Jupyter και οι διαθέσιμες επιλογές πυρήνων για την δημιουργία ενός αρχείου.

Στο Σχήμα 2.3 φαίνεται η αρχική σελίδα του Jupyter. Η δημιουργία ενός νέου αρχείου πραγματοποιείται πατώντας New στη δεξιά πλευρά του παραθύρου, όπου εμφανίζονται επιλογές για τους διαθέσιμους πυρήνες (kernels). Στο συγκεκριμένο λογισμικό που δείχνει το στιγμιότυπο οθόνης υπάρχουν εγκατεστημένοι οι πυρήνες της Python, του Octave, της Matlab και της Javascript.

Αντίστοιχα στο Σχήμα 2.4 παρουσιάζεται ένα παράδειγμα αρχείου Jupyter Notebook σε γλώσσα Python. Όπως φαίνεται, περιλαμβάνει τίτλο, κείμενο, μαθηματική εξίσωση σε μορφή LaTeX και τον απαραίτητο κώδικα της Python για τη δημιουργία μιας απλής γραφικής παράστασης. Ο κώδικας εκτελείται πατώντας την επιλογή Run στη γραμμή εργαλείων και η γραφική παράσταση εμφανίζεται ακριβώς κάτω από τον κώδικα.



Γραμμή εργαλείων

Κελί κειμένου με τίτλο, επεξήγηση και εξίσωση.

Κελί κώδικα που προσδιορίζει τις απαραίτητες βιβλιοθήκες της Python.

Σύντομη επεξήγηση

Κελιά κώδικα για τον ορισμό δύο μεταβλητών και τη δημιουργία μιας απλής γραφικής παράστασης.

Σχήμα 2. 4 : Αρχείο notebook σε γλώσσα Python για τη δημιουργία μιας γραφικής παράστασης [62].

Οι δυνατότητες που περιγράφηκαν παραπάνω, επιτρέπουν την ανάπτυξη του Jupyter Notebook ως χώρο εργασίας για την προβολή και εξερεύνηση διδασκόμενων εννοιών. Λόγω αυτής της χρησιμότητας καθώς και της εύκολης διάδοσής του, συνιστάται ως εκπαιδευτικό περιβάλλον. Είναι ιδανικό για επιστημονική έρευνα, ακαδημαϊκές ή επαγγελματικές δραστηριότητες σε τομείς των πεδίων STEM (Science, Technology, Engineering and Mathematics) όπως Εκπαίδευση μηχανικών, Προγραμματισμό και Επιστήμη των Υπολογιστών, Στατιστική και Πιθανότητες, Μηχανική Μάθηση, Επιστήμη των Δεδομένων, Φυσική, Μαθηματικά, Χημεία, Βιολογία κ.ά.

Από τους εκπαιδευτικούς μπορεί να αξιοποιηθεί για ένα ευρύ φάσμα μαθησιακών σκοπών. Για την ανάπτυξη και οργάνωση διδακτικού υλικού του μαθήματος σε διάφορες μορφές (όπως εγχειρίδια και σημειωματάρια), την παρουσίαση διαλέξεων, την εκτέλεση ζωντανού κώδικα, υπολογιστικών και αριθμητικών προσομοιώσεων, την ανάθεση εργασιών καθώς και την επίλυσή τους στο αμφιθέατρο, τον σχεδιασμό



εργαστηρίων, τη δημιουργία ανοικτών ψηφιακών μαθημάτων κ.ά. Πολλές φορές μπορεί να προσαρμοστεί σε ήδη υπάρχουσες δραστηριότητες μαθημάτων και άλλες φορές να τις επεκτείνει, για να αυξηθεί το ενδιαφέρον, η συμμετοχή, η κατανόηση και η απόδοση των φοιτητών.

Οι νέοι χρήστες μπορούν να ξεκινήσουν με μικρά βήματα, ενσωματώνοντάς το σε μεμονωμένες ενότητες, εργασίες ή δραστηριότητες. Μια τέτοια προσέγγιση είναι χρήσιμη για να παρατηρηθεί η αλληλεπίδραση του μαθησιακού κοινού με το περιβάλλον αυτό και να διερευνηθεί η αξιοποίησή του με ελάχιστο κόστος και κίνδυνο. Κάποιοι μπορεί να επιλέξουν να δημιουργήσουν εξολοκλήρου νέο εκπαιδευτικό υλικό στο Jupyter ή εναλλακτικά να επωφεληθούν από τους διαδικτυακούς πόρους που υπάρχουν ήδη για παράδειγμα στο GitHub, έναν δημοφιλή ιστότοπο για φιλοξενία, διαχείριση και συνεργατική επεξεργασία κώδικα [40]. Το βασικό χαρακτηριστικό που το καθιστά ελκυστικό σε φοιτητές και εκπαιδευτικούς, είναι ότι συνδυάζει τις εξηγήσεις που παραδοσιακά υπάρχουν στα βιβλία με την αλληλεπίδραση μιας εφαρμογής.

## 2.2 Πλεονεκτήματα του Jupyter Notebook στην Εκπαίδευση

Η χρήση των Jupyter Notebooks και των προγόνων τους, IPython Notebooks, έχει απασχολήσει ήδη από την εμφάνισή τους την ακαδημαϊκή κοινότητα και έτσι υπάρχει μία σχετικά εκτενής βιβλιογραφία αναφορικά με εφαρμογές στη διδασκαλία μαθημάτων για φοιτητές σε διάφορα επιστημονικά αντικείμενα, όπως η Επιστήμη των Δεδομένων [41], η Τεχνητή Νοημοσύνη [42], η Επιστήμη των Υπολογιστών [43], η Στατιστική [44], [45], οι Πιθανότητες [46], η Αριθμητική Ανάλυση [47], οι Διαφορικές εξισώσεις [48], η Αεροδυναμική [49], η Φυσική [50] καθώς και στη διεξαγωγή εργαστηριακών ασκήσεων σε αντικείμενα Ηλεκτρολόγων [34], [35], [51], [52] Μηχανολόγων [37], Πολιτικών [53] και Χημικών Μηχανικών [54]. Το πλήθος των τεκμηριωμένων εφαρμογών που έχουν πραγματοποιηθεί ως σήμερα συνέβαλε στο να εξαχθούν χρήσιμα συμπεράσματα για τα πλεονεκτήματα που παρουσιάζει η χρήση των Jupyter Notebooks στη διδασκαλία, τα οποία συνοψίζονται παρακάτω.

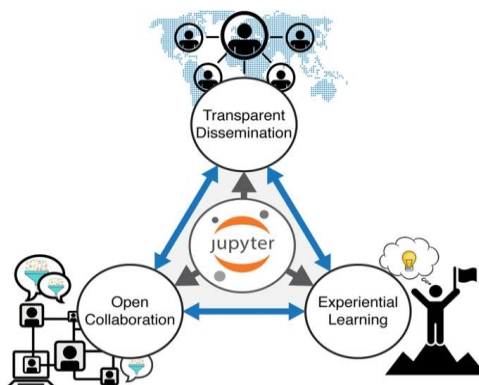
Ένα καθοριστικό χαρακτηριστικό του Jupyter Notebook αποτελεί ο συνδυασμός θεωρίας, παραδειγμάτων και πρακτικών εφαρμογών, που βελτιώνει τη διαδικασία της μάθησης καθώς επιτρέπει την καλύτερη απορρόφηση των εννοιών του μαθήματος και συνεπώς την ενίσχυση της γνώσης. Προκαλείται αξιοσημείωτη διαφορά στην κατανόηση των μαθητών όταν η κάθε έννοια ακολουθείται από ένα παράδειγμα ή μια δραστηριότητα, ειδικά μάλιστα όταν η δυσκολία είναι σταδιακά αυξανόμενη [48].

Άλλο ένα ιδιαίτερα σημαντικό πλεονέκτημα είναι το διαδραστικό περιεχόμενο. Η δυνατότητα αλληλεπίδρασης αυξάνει το ενδιαφέρον και τα κίνητρα των μαθητών [32], [34], [35] καθώς έχουν την ευκαιρία να ξεφύγουν από την παθητική προβολή του περιεχομένου του μαθήματος και να ασχοληθούν με την εξερεύνηση, ανάλυση, σύνθεση και αξιολόγησή του με ενεργούς τρόπους. Αυτή η δυνατότητα πειραματισμού επηρεάζει ουσιαστικά τη νοοτροπία των φοιτητών, ενθαρρύνοντάς τους να μάθουν «να μαθαίνουν» και συμβάλλοντας στη θεμελίωση μιας διαισθητικής αντίληψης του αντικειμένου το οποίο εξετάζουν. Παράλληλα τους βοηθά να αναπτύξουν δεξιότητες όπως η κριτική σκέψη και η δημιουργικότητα.

Χρειάζεται επίσης να σημειωθεί ότι η εισαγωγή της συστηματικής ενεργού μάθησης ως τρόπου απόκτησης γνώσεων, μέσω της χρήσης των Jupyter Notebooks, όχι μόνο ευνοεί την αύξηση της συμμετοχής των φοιτητών στο μάθημα και τη βελτίωση των επιδόσεών τους [34], [35], [37] αλλά προσφέρει και οφέλη πέρα από τα όρια του εκάστοτε μαθήματος. Η διαδραστικότητα οδηγεί στην εμπλοκή τους με το υλικό καλλιεργώντας το ενδιαφέρον για την εξερεύνηση των εννοιών, γεγονός που είναι πιθανό να τροφοδοτήσει το ενδιαφέρον και τη ενεργή συμμετοχή τους και σε άλλα μαθήματα.

Ένα ακόμη βασικό χαρακτηριστικό των Jupyter Notebooks είναι ότι δίνουν τη δυνατότητα στους χρήστες να οργανώνουν κώδικα, απεικονίσεις και κείμενο σε μια υπολογιστική αφήγηση. Έτσι, αφενός ενισχύονται οι επικοινωνιακές δεξιότητες των φοιτητών. Αφετέρου, η έκθεση σε μαθησιακές εμπειρίες που περιέχουν μεθοδολογίες και κώδικα τους εξοικειώνει με τον προγραμματισμό και συμβάλλει στη συγκρότηση των βασικών θεμελίων της υπολογιστικής σκέψης, που θεωρείται μια πολύ βασική δεξιότητα στη σημερινή εποχή [31], [32].

Τέλος, ένα σημαντικό πλεονέκτημα των Jupyter Notebooks είναι η κοινότητα στην οποία έχει αναπτυχθεί, η οποία προωθεί την ελεύθερη πρόσβαση και τη συνεργασία [47]. Η ενσωμάτωση του εργαλείου σε μαθήματα διασφαλίζει ότι το περιεχόμενό τους παραμένει προσβάσιμο από όλους τους φοιτητές και εκτός του εκπαιδευτικού ιδρύματος, γεγονός που ευνοεί ισότητα και την αυτόνομη μάθηση. Αυτό βεβαίως εξυπηρετεί και τους εκπαιδευτικούς αφού το υλικό ενός μαθήματος μπορεί να διαδοθεί και να επαναχρησιμοποιηθεί και από άλλους [37].



Σχήμα 2. 5 : Τα Jupyter Notebooks παρέχουν ένα αποτελεσματικό μέσο για διάδοση γνώσεων, μια κεντρική και διαδραστική πλατφόρμα για ανοιχτή συνεργασία και ευκαιρίες για βιωματική μάθηση [55].

## 2.4 Χρήση του Jupyter Notebook στην Εκπαίδευση Μηχανικών: μελέτες περίπτωσης

Με αφορμή το αντικείμενο αυτής της διπλωματικής, παρουσιάσαμε στο 6<sup>ο</sup> Πανελλήνιο Επιστημονικό Συνέδριο: Ένταξη και Χρήση των ΤΠΕ στην Εκπαιδευτική Διαδικασία [68], την εργασία μας με τίτλο «Αξιοποίηση των Jupyter Notebooks στην εκπαιδευτική διαδικασία», με σκοπό τη διερεύνηση της χρήσης του εργαλείου αυτού στις διαδικασίες της διδασκαλίας και της μάθησης. Συγκεκριμένα αναφερθήκαμε σε ποιες ανάγκες της σύγχρονης σχολικής εκπαίδευσης θα μπορούσε να ανταποκριθεί, ποια εφόδια μπορεί να προσφέρει στους μαθητές και κατά πόσο οι τελευταίοι είναι σε θέση να αλληλεπιδράσουν με αυτό. Επίσης αναδείξαμε αντιπροσωπευτικές μελέτες περίπτωσης στη σχολική εκπαίδευση, όπου η ένταξή του εργαλείου σε διεθνές επίπεδο φαίνεται να βρίσκεται ακόμα σε πρώιμο στάδιο [71]-[74].

Συνοψίζοντας τα συμπεράσματα της εργασίας μας, η ανάπτυξη καινοτόμων εκπαιδευτικών εμπειριών μέσω του Jupyter Notebook, όπως αναφέρθηκε παραπάνω, παρουσιάζει αρκετά πλεονεκτήματα. Από τις εκπαιδευτικές περιπτώσεις που μελετήθηκαν σε μαθητές σχολείου φαίνεται ότι, με την κατάλληλη καθοδήγηση, είναι απόλυτα ικανοί να αλληλεπιδράσουν με ένα εκπαιδευτικό περιβάλλον που θεωρείται πιο δύσκολο, και μάλιστα να ασχοληθούν με απαιτητικές έννοιες. Γίνεται φανερό, λοιπόν ότι αποτελεί ένα εργαλείο που μπορεί να συνεισφέρει στη βελτίωση της εκπαιδευτικής διαδικασίας στο σχολείο, αλλά και στο πανεπιστήμιο, όπως δηλώνει και το πλήθος των εφαρμογών που είναι διαθέσιμες και ανοικτές στο διαδίκτυο (π.χ. στο GitHub).

Λαμβάνοντας υπόψη ότι το Jupyter Notebook είναι κατά βάση ένα προγραμματιστικό εργαλείο, θα προσπαθήσουμε να δώσουμε μια καλύτερη εικόνα για τις δυνατότητες της αξιοποίησής του σε παραδοσιακά μαθήματα Ηλεκτρολόγων Μηχανικών, όπως τα Συστήματα Ηλεκτρικής Ενέργειας, μέσω της παράθεσης ορισμένων αντιπροσωπευτικών παραδειγμάτων, όπου η χρήση του συνδυάζεται με διαλέξεις και άλλα εργαλεία για να ενισχύσει την εργαστηριακή εκπαίδευση των φοιτητών.

### 2.4.1 Jupyter Notebooks για εργαστήρια στα «Συστήματα Ηλεκτρικής Ενέργειας» με χρήση υπολογιστή.

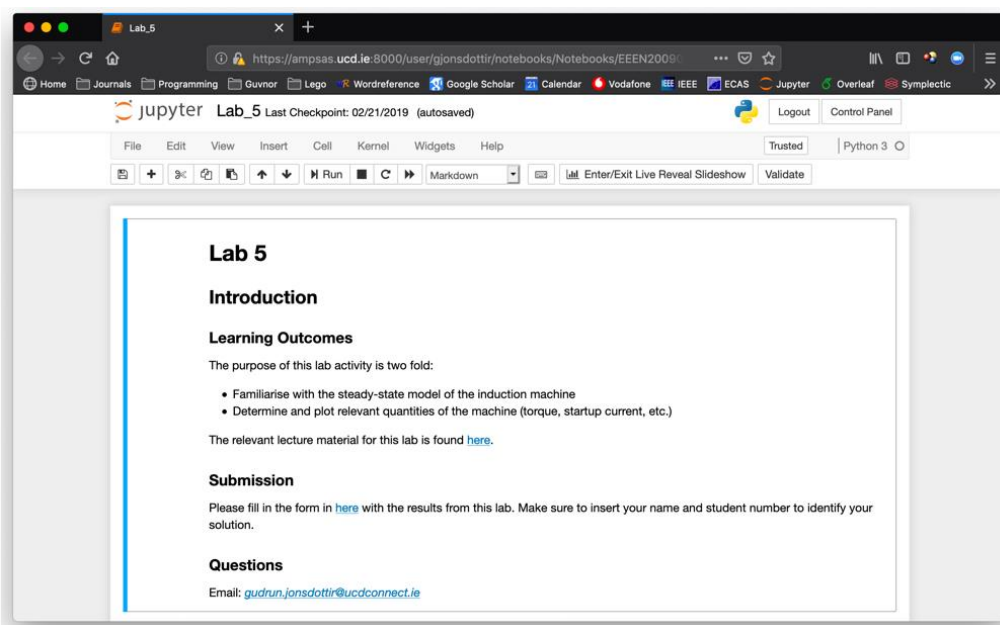
Το ακαδημαϊκό έτος 2018/2019 υλοποιήθηκαν στο Πανεπιστημιακό Κολέγιο του Δουβλίνου, εργαστήρια για το μάθημα «Συστήματα Ηλεκτρικής Ενέργειας», τα οποία αναπτύχθηκαν σε Jupyter Notebooks [52]. Το μάθημα αυτό απευθύνεται σε (160) φοιτητές Ηλεκτρολόγους, Ηλεκτρονικούς και Μηχανολόγους Μηχανικούς. Σκοπός της ενότητας αυτής είναι η εισαγωγή των φοιτητών στις κύριες έννοιες κυκλωμάτων εναλλασσόμενου ρεύματος, στις βασικές γνώσεις Ηλεκτρικών Μηχανών (μετασχηματιστές, κινητήρες επαγωγής και σύγχρονες γεννήτριες) και στην Ανάλυση Συστημάτων Ηλεκτρικής Ενέργειας.

Τα εργαλεία που χρησιμοποιήθηκαν ήταν το Jupyter Notebook και το Google forms. Αρχικά σε συνδυασμό με το Jupyter χρησιμοποιήθηκε η γλώσσα MATLAB γιατί οι δευτεροετής φοιτητές είναι πιο εξοικειωμένοι με αυτήν παρά με την Python. Όμως εν

τέλει επικράτησε η επιλογή της Python για δύο λόγους: 1. Η γνώση της Python είναι σημαντικό κριτήριο στη βιομηχανία. 2. Η επέκταση του MATLAB για το Jupyter δεν λειτουργούσε σωστά με την άδεια χρήσης που παρείχε το Πανεπιστήμιο, ενώ η Python δεν χρειάζεται καμία άδεια χρήσης και είναι εύκολη στη εγκατάσταση.

Από τους φοιτητές ζητήθηκε να λύσουν μια ποικιλία ασκήσεων εισάγοντας τους στην έννοια της «μοντελοποίησης» μέσω εξισώσεων. Κάθε Jupyter Notebook σχεδιάστηκε έτσι ώστε οι φοιτητές να επιλύουν «αντίστροφα προβλήματα» (inverse problems), δηλαδή να βρίσκουν μέσω της τεχνικής δοκιμής και σφάλματος τις παραμέτρους ενός δεδομένου μηχανήματος ή ενός απλού συστήματος ισχύος, για να ικανοποιούν συγκεκριμένες απαιτήσεις σχεδίασης. Επιπλέον, είχαν την ευκαιρία να οπτικοποιήσουν μεταβλητές που σχετίζονται με κάθε ηλεκτρική συσκευή ή σύστημα που μελετούσαν, μέσω γραφικών παραστάσεων στο Jupyter Notebook.

Συγκεκριμένα, οι εργαστηριακές δραστηριότητες χωρίστηκαν σε δύο μέρη: (i) δοκιμή ανοιχτού κυκλώματος και βραχυκυκλώματος μετασχηματιστή 3 kVA. και (ii) επτά Jupyter Notebooks που αντιστοιχούν στα επτά μέρη του μαθήματος. Οι φοιτητές είχαν μια εβδομάδα για να ολοκληρώσουν κάθε Jupyter Notebook, το οποίο ανατίθεται μόλις το αντίστοιχο θέμα έχει καλυφθεί από τον καθηγητή. Οι Google forms χρησιμοποιήθηκαν ώστε να απλοποιηθεί η διαδικασία της συλλογής και βαθμολόγησης των απαντήσεων των φοιτητών.



Σχήμα 2. 6: Jupyter Notebook για το εργαστήριο Συστημάτων Ηλεκτρικής Ενέργειας στο Πανεπιστημιακό Κολέγιο Δουβλίνου [52].

Από ερωτηματολόγια που τέθηκαν στους φοιτητές σχετικά με τις εργαστηριακές δραστηριότητες φαίνεται ότι αυτή η προσέγγιση επίλυσης των προβλημάτων που συζητούνται στο μάθημα, βελτιώνει την καμπύλη μάθησης των φοιτητών. Ταυτόχρονα, ενισχύονται οι δεξιότητές τους στον προγραμματισμό, κάτι που είναι σημαντικό για την επαγγελματική τους πορεία στο μέλλον. Οι απαντήσεις έδειξαν ότι σχεδόν το 50% των φοιτητών συνάντησε δυσκολίες στην χρήση του Jupyter, γεγονός

που οφείλεται κυρίως στην έλλειψη εξοικείωσης με την Python. Για το λόγο αυτό, οι διοργανωτές του μαθήματος σκοπεύουν να συμπεριλάβουν τις επόμενες χρονιές μια σύντομη εισαγωγή στην Python στην αρχή του μαθήματος. Επιπλέον παρόλο που λιγότερο του 10% των φοιτητών δυσκολεύτηκε στη χρήση των Google forms, κάποιιοι δήλωσαν δυσαρέσκεια για την ανάγκη χρήσης και των δύο πλατφορμών, πρόβλημα που οι διδάσκοντες θα επιχειρήσουν να λύσουν αξιοποιώντας τις προεκτάσεις του Jupyter Notebook (όπως το nbgrader) ώστε να ενσωματωθούν οι δυνατότητες της συλλογής και διόρθωσης των απαντήσεων στο ίδιο αρχείο της άσκησης.

Ακόμη, πολλοί φοιτητές δήλωσαν ότι βρήκαν βοηθητική τη χρήση κώδικα για την καλύτερη κατανόηση της θεωρίας του μαθήματος, κυρίως μέσω των οπτικοποιήσεων και των γραφικών παραστάσεων. Επίσης θεωρούν ότι τα Jupyter Notebooks είναι ένας καλός τρόπος για την ταυτόχρονη εκμάθηση της Python και του υλικού του μαθήματος. Τα τελικά αποτελέσματα της εξέτασης του μαθήματος είναι ενδεικτικά. Συγκεκριμένα το έτος 2017/2018 (πριν την χρήση των Jupyter Notebooks) πέρασε το μάθημα το 58% των φοιτητών με μέσο όρο D-, ενώ το έτος 2018/2019 (με τη χρήση των Jupyter Notebooks) πέρασε το 80% των φοιτητών με μέσο όρο βαθμολογίας C. Συνολικά οι διοργανωτές του μαθήματος σημειώνουν ότι το Jupyter Notebook προσφέρει μια ευέλικτη λύση που μπορεί να προσαρμοστεί στις δεξιότητες και στα ενδιαφέροντα των μαθητών.

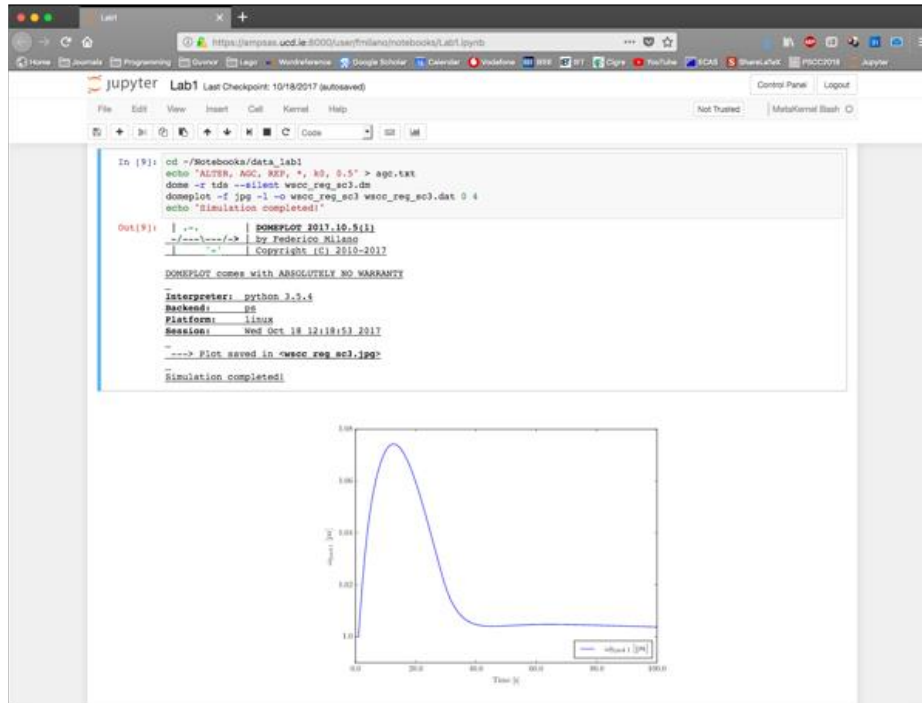
#### **2.4.2 Jupyter Notebooks για εργαστήρια στον «Έλεγχο και Δυναμική Συμπεριφορά ΣΗΕ» με χρήση υπολογιστή.**

Το ακαδημαϊκό έτος 2017/2018 οι διδάσκοντες του μαθήματος «Έλεγχος και Δυναμική Συμπεριφορά ΣΗΕ» στο Πανεπιστημιακό Κολέγιο του Δουβλίνου, οργάνωσαν ένα εργαστήριο στο περιβάλλον του Jupyter Notebook που αφορά 15 φοιτητές, αποκλειστικά από τη σχολή Ηλεκτρολόγων Μηχανικών [51]. Για τους σκοπούς αυτού του εργαστηρίου χρησιμοποιήθηκαν οι πλατφόρμες Dome και Jupyter Notebook.

Το Dome είναι ένα, παλαιού τύπου, Unix πρόγραμμα, που βασίζεται στη γλώσσα Python. Δεν παρέχει γραφικά και εκτελείται από τη γραμμή εντολών, μέσω ενός, όχι και τόσο φιλικού προς το χρήστη, τερματικού Unix. Το πρόγραμμα αυτό αναπτύχθηκε για τις ανάγκες εργαστηριακών δραστηριοτήτων σε θέματα μοντελοποίησης, ελέγχου, ευστάθειας και ανάλυσης συστημάτων ισχύος και η χρήση του στην εκπαίδευση παρουσιάζει πολλά πλεονεκτήματα. Ωστόσο η εκμάθηση ενός τέτοιου εργαλείου καταλαμβάνει σημαντικό χρόνο του μαθήματος. Το Jupyter Notebook επιλέχθηκε γιατί μπορεί να βοηθήσει στην ελαχιστοποίηση του χρόνου που χρειάζεται να αφιερώσουν οι φοιτητές στην εξοικείωση με ένα προγραμματιστικό περιβάλλον.

Ακολουθώντας κάποια βήματα οι διδάσκοντες του μαθήματος ενσωμάτωσαν το Dome στο Jupyter Notebook. Το πλεονέκτημα της εκτέλεσης του Dome μέσω του Jupyter είναι ότι ανοίγοντας μια σελίδα του browser, οι φοιτητές μπορούν εύκολα να δοκιμάσουν διάφορες τιμές παραμέτρων και να οπτικοποιήσουν το αποτέλεσμα αυτών των αλλαγών. Για την ίδια διαδικασία, με τη χρήση μόνο του Dome, θα χρειαζόταν να ανοίξουν τέσσερα διαφορετικά παράθυρα για την εκτέλεση της προσομοίωσης. Άλλο ένα πλεονέκτημα αποτελεί το γεγονός ότι το κείμενο μπορεί να συνοδεύει με οδηγίες τον κώδικα, τις οποίες οι φοιτητές μπορούν να ακολουθούν για να εκτελέσουν τις προσομοιώσεις, να τρέξουν τον δικό τους κώδικα και τελικά να προσθέσουν τα δικά

τους σχόλια και συμπεράσματα. Στο τέλος του εργαστηρίου οι φοιτητές δεν χρειάζεται να γράψουν από την αρχή αναφορά καθώς το αρχείο του Jupyter που αποθηκεύουν περιέχει όλη την ροή της εργασίας τους και έτσι οι διδάσκοντες αρκεί να ελέγξουν μόνο τα σημεία του αρχείου που οι φοιτητές κλήθηκαν να συμπληρώσουν.



Σχήμα 2. 7: Χρήση του Dome ενσωματωμένο στο Jupyter Notebook για το μάθημα «Έλεγχος και Δυναμική Συμπεριφορά ΣΗΕ» στο Πανεπιστημιακό Κολέγιο Δουβλίνου [51].

Οι στόχοι του μαθήματος είναι η βασική κατανόηση των εννοιών του ελέγχου συχνότητας και τάσης του ΣΗΕ, η γνώση ελέγχου όλων των βασικών συσκευών που χρησιμοποιούνται στα συστήματα μεταφοράς Υψηλής Τάσης και η υλοποίηση πρακτικών παραδειγμάτων που βασίζονται σε αριθμητικές προσομοιώσεις.

Το μάθημα περιλαμβάνει μια εισαγωγική διάλεξη και τέσσερις εργαστηριακές ασκήσεις. Οι ασκήσεις έχουν σκοπό να αναπτύξουν στους φοιτητές τις ακόλουθες δεξιότητες: κριτική ανάλυση των αποτελεσμάτων, ικανότητα επίλυσης προβλημάτων, ικανότητα αναγνώρισης των βασικών σημείων ενός φαινομένου και η ικανότητα κατανόησης της αντίδρασης ενός συστήματος ισχύος, με βάση τα αποτελέσματα της προσομοίωσης. Το τελευταίο είναι πολύ σημαντικό ώστε οι φοιτητές να είναι σε θέση να καταλάβουν αν τα αποτελέσματα των προσομοιώσεων είναι λογικά ή όχι και γιατί.

Οι εργαστηριακές δραστηριότητες αφορούν τα παρακάτω θέματα: i) αδρανειακή απόκριση και ρύθμιση συχνότητας σύγχρονων μηχανών, ii) αυτόματος έλεγχος τάσης σύγχρονων μηχανών και σταθεροποιητών συστήματος, iii) ρύθμιση τάσης υπό φορτίο (ΣΑΤΥΦ – Σύστημα Αλλαγής Τάσης Υπό Φορτίο), iv) έλεγχος συχνότητας μέσω ασύγχρονων συσκευών.

Ο διδακτορικός φοιτητής που ασχολείται με την υλοποίηση των εργαστηριακών ασκήσεων, έχοντας και την εμπειρία των παλαιότερων εργαστηρίων υπογραμμίζει ότι

το Jupyter Notebook επέτρεψε στους φοιτητές να επικεντρωθούν στο αντικείμενο των ασκήσεων, καθώς δεν είχαν τόσες απορίες για τη χρήση του λογισμικού σε σύγκριση με τις προηγούμενες χρονιές. Ωστόσο αυτό δεν μεταφράστηκε απολύτως σε καλύτερες επιδόσεις στις αναφορές και αυτό ίσως οφείλεται στο γεγονός ότι η φαινομενική απλότητα του Jupyter Notebook, οδήγησε κάποιους λιγότερο κινητοποιημένους φοιτητές να κάνουν μικρότερη προσπάθεια για να ολοκληρώσουν τις εργασίες.

Η γενική άποψη των φοιτητών για την χρήση του Jupyter Notebook με το Dome ήταν θετική. Μπορούσαν να δουλέψουν ανεξάρτητα και εξοικειώθηκαν γρήγορα με το εργαλείο, ωστόσο κάποιοι τόνισαν ότι τους περιόρισε η χρήση του Jupyter για τη δημιουργία της αναφοράς και αυτό ίσως οφείλεται στο γεγονός ότι, τουλάχιστον στην αρχή, δεν γνώριζαν τόσο καλά τις δυνατότητές του. Άλλο ένα θέμα ήταν ότι τα Jupyter Notebooks είχαν προσαρμοστεί για την εκτέλεση βασικών, σχετικά εύκολων προσομοιώσεων και όσοι φοιτητές ήθελαν να δοκιμάσουν πιο δύσκολες προσομοιώσεις έπρεπε να ετοιμάσουν τα απαραίτητα αρχεία από την αρχή στο Dome και έπειτα να τα προσαρμόσουν στο Jupyter, κάνοντας επιπλέον κόπο.

Λαμβάνοντας τα παραπάνω υπόψη οι διδάσκοντες του μαθήματος σκοπεύουν να λύσουν τα προβλήματα επαναπροσδιορίζοντας το περιεχόμενο της εισαγωγικής διάλεξης στοχεύοντας να δείξουν στους φοιτητές κάποιες πρακτικές λειτουργίες του Dome και επιπλέον πως να εκμεταλλεύονται όσο γίνεται καλύτερα τις δυνατότητες του περιβάλλοντος του Jupyter Notebook σε ότι αφορά την παρουσίαση κειμένου, εξισώσεων και αποτελεσμάτων. Συνολικά οι διοργανωτές του μαθήματος σημειώνουν ότι το Jupyter Notebook προσφέρει μια ευέλικτη λύση που μπορεί να προσαρμοστεί στις δεξιότητες και στα ενδιαφέροντα των μαθητών.

### **2.4.3 Διαδικτυακές Εργαστηριακές Ασκήσεις για την Εκπαίδευση στις Ηλεκτρικές Μηχανές.**

Από το Τμήμα Ηλεκτρικής Βιώσιμης Ενέργειας, του Τεχνολογικού Πανεπιστημίου του Delft (TU Delft), προτείνεται μια μεθοδολογία για την ενσωμάτωση διαδικτυακών εργαστηριακών ασκήσεων στην ενεργή διαδικασία μάθησης [83]. Η εργασία παρουσιάζει πώς μπορούν να χρησιμοποιηθούν διαδικτυακές εργαστηριακές ασκήσεις για την εκπαίδευση στις Ηλεκτρικές Μηχανές και περιγράφεται ένα πρακτικό παράδειγμα για να δείξει την αποτελεσματικότητά τους.

Όπως επισημαίνουν οι συγγραφείς, τα τελευταία χρόνια η Πανεπιστημιακή εκπαίδευση γενικά αλλά και τα παραδοσιακά μαθήματα Ηλεκτρολόγων Μηχανικών, όπως οι Ηλεκτρικές Μηχανές ειδικότερα, υφίστανται μια μετάβαση, κυρίως λόγω των αναδυόμενων εφαρμογών, της ανάπτυξης της τεχνολογίας και της ψηφιοποίησης.

Συγκεκριμένα αναφέρουν ότι σήμερα οι φοιτητές προτιμούν να συγκεντρώνουν πληροφορίες από ηλεκτρονικά μέσα. Γι' αυτό γίνονται προσπάθειες στα πιο σύγχρονα συγγράμματα να συμπεριλαμβάνεται ψηφιακό εκπαιδευτικό υλικό, όπως οπτικοποιήσεις και υπολογιστικές προσομοιώσεις.

Επιπλέον, επειδή οι ηλεκτρικές μηχανές πλέον αποτελούν μέρος εφαρμογών των ΑΠΕ και της ηλεκτροκίνησης, διδάσκονται όλο και συχνότερα ως στοιχεία μεγαλύτερων

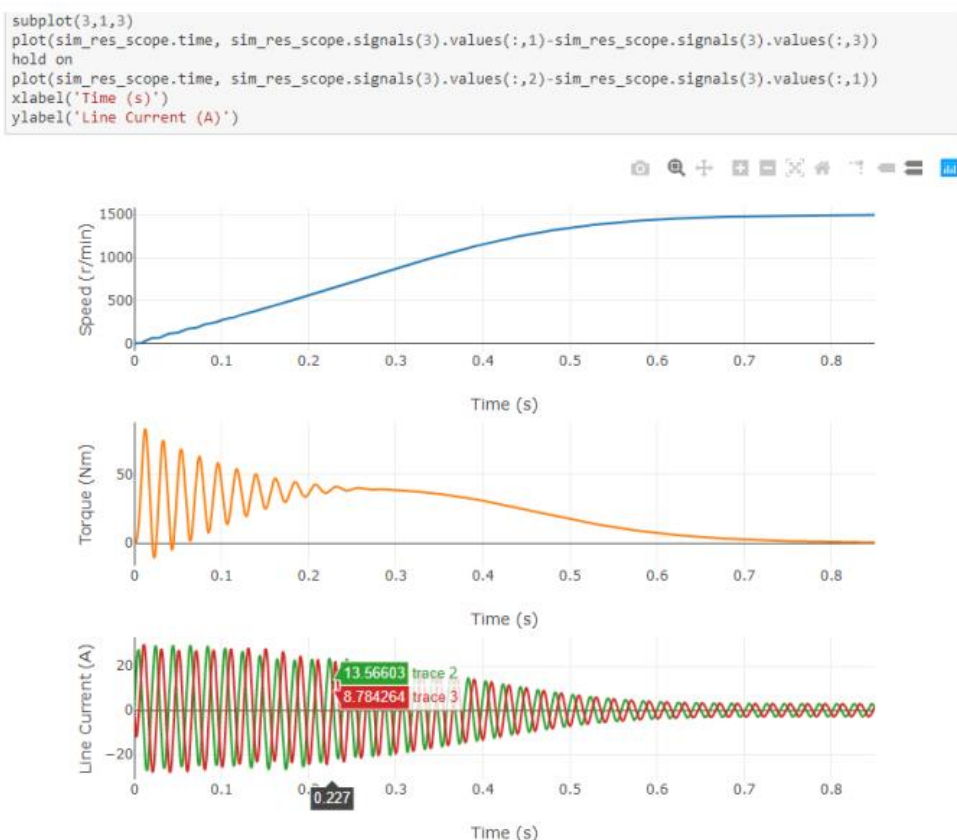


συστημάτων, όπως ανεμογεννήτριες και ηλεκτρικά αυτοκίνητα, παρά ως ένα ανεξάρτητο θέμα, γι' αυτό υπάρχει ανάγκη να διατίθεται εύκολα προσβάσιμο διαδικτυακό υλικό (π.χ. με τα MOOCs).

Ταυτόχρονα, όπως συμβαίνει με τα περισσότερα παραδοσιακά μαθήματα Ηλεκτρολόγων Μηχανικών, για το μάθημα των Ηλεκτρικών Μηχανών, οι εργαστηριακές ασκήσεις είναι πολύ σημαντικό μέρος της εκπαιδευτικής διαδικασίας και μάλιστα, με τη βοήθεια των ΤΠΕ έχει γίνει σημαντική πρόοδος στην εφαρμογή εικονικών και απομακρυσμένων εργαστηρίων (virtual and remote labs). Ωστόσο η άμεση αλληλεπίδραση της πρακτικής εφαρμογής και του θεωρητικού μέρους είναι ακόμη περιορισμένη.

Για να μειωθεί αυτό το χάσμα, οι συγγραφείς, σχεδίασαν μια πιλοτική διαδικτυακή εργαστηριακή άσκηση την οποία προσάρμοσαν για το μεταπτυχιακό μάθημα «Έλεγχος Ηλεκτρικών Μηχανών» και εφάρμοσαν στο εξάμηνο 2018-2019, για μια ομάδα 12 φοιτητών. Οι αναφορές των εργαστηρίων σχεδιάστηκαν έτσι ώστε να συνδέσουν τη θεωρία με το διαδικτυακό πρακτικό μέρος. Για τους σκοπούς του εργαστηρίου χρησιμοποιήθηκαν: (i). η φυσική διάταξη του εργαστηρίου Ηλεκτρικών Μηχανών, (ii). σχετικό λογισμικό GUI και (iii). μια διαδραστική πλατφόρμα ΤΠΕ για την εφαρμογή των αναφορών.

Τα διαδραστικά εργαστήρια χρησιμοποιήθηκαν για την επεξήγηση της θεωρίας στους φοιτητές. Στη συγκεκριμένη περίπτωση προτείνεται από τους συγγραφείς το Jupyter Notebook ως ένα εργαλείο που μπορεί να κάνει τις επεξηγήσεις πιο διαδραστικές, να κρατήσει το ενδιαφέρον των φοιτητών, να παρουσιάσει τις μαθηματικές εξισώσεις, τη διαδικασία μοντελοποίησης και τα αποτελέσματα της προσομοίωσης. Οι φοιτητές μπορούν να αλληλεπιδράσουν με τον διδάσκοντα εκτελώντας τον κώδικα MATLAB στο Jupyter Notebook, να υποβάλλουν τις αναφορές τους καθώς και να πειραματιστούν με τον κώδικα μετά τη λήξη του εργαστηρίου. Η εμπειρία αυτής της πρακτικής στο TU Delft έχει δείξει ότι οι διαδραστικές υπολογιστικές πλατφόρμες, όπως το Jupyter Notebook, καθιστούν πιο αποδοτική τη διδασκαλία, διευκολύνοντας τη διόρθωση των αναφορών, την προσωπική μελέτη των φοιτητών και τη μετάδοση της γνώσης.



Σχήμα 2. 8: Παράδειγμα προσομοίωσης της εκκίνησης μιας επαγωγικής μηχανής στο Jupyter Notebook, σε γλώσσα MATLAB [83].

Σύμφωνα με τους συγγραφείς, οι αναφορές θα πρέπει να σχεδιάζονται με έναν σταδιακό τρόπο ώστε να χτίζουν τη γνώση και την αυτοπεποίθηση των φοιτητών. Προτείνεται μια ροή που ξεκινά από το θεωρητικό υπόβαθρο, συνεχίζει στην μοντελοποίηση και προσομοίωση στον υπολογιστή και καταλήγει στα εργαστηριακά πειράματα και την ανάλυση των αποτελεσμάτων.

Το θέμα της συγκεκριμένης διαδικτυακής εργαστηριακής άσκησης είναι «Βαθμωτός έλεγχος μηχανής επαγωγής με ανάδραση ταχύτητας και αντιστάθμιση ολίσθησης». Πριν από την εργαστηριακή άσκηση, οι αναφορές σχετικά με το σχεδιασμό και την εφαρμογή του ελεγκτή, τη μοντελοποίηση και την προσομοίωση της επαγωγικής μηχανής και την ανάπτυξη υλικού έχουν ολοκληρωθεί, ελεγχθεί και συζητηθεί στη διαδραστική συνεδρία.

Για το πρακτικό μέρος, μια κάμερα χρησιμοποιείται για την παρακολούθηση της διάταξης του εργαστηρίου. Επίσης, χρησιμοποιούνται δύο μετρητές για την παρακολούθηση της ρυθμιζόμενης και της μετρούμενης ταχύτητας, δύο χειριστήρια για τη ρύθμιση της ταχύτητας αναφοράς και τη σύνδεση του πεδίου αναφοράς και παλμογράφος για τη μέτρηση της τάσης φάσης της επαγωγικής μηχανής και της διαμόρφωσης της κυματομορφής. Το αντίστοιχο λογισμικό τρέχει δίπλα στον πίνακα λειτουργίας για τον έλεγχο του παλμογράφου και την παρακολούθηση των κύριων κυματομορφών. Ο πίνακας λειτουργίας είναι ένα GUI που δημιουργήθηκε στο MATLAB App Designer.

Για συγκεκριμένο χρονικό διάστημα, ο κάθε φοιτητής που συνδέεται μέσω της ιστοσελίδας μεταφέρεται στην απομακρυσμένη επιφάνεια εργασίας του υπολογιστή και μπορεί να αλληλεπιδράσει με τον ελεγκτή και τον παλμογράφο χρησιμοποιώντας το δικό του ποντίκι και πληκτρολόγιο. Στην ηλεκτρονική εργαστηριακή συνεδρία, οι φοιτητές καλούνται να ορίσουν τρεις διαφορετικές ταχύτητες αναφοράς και να τις μετρήσουν. Για κάθε ταχύτητα αναφοράς, πρέπει να μετρούν τη διαμόρφωση και τις κυματομορφές του ρεύματος κάτω από τρεις διαφορετικές ρυθμίσεις σύνδεσης του πεδίου. Μετά τη διαδικτυακή εργαστηριακή άσκηση, καλούνται να συγκρίνουν τα αποτελέσματα που λαμβάνονται με τις μετρήσεις, τις προσομοιώσεις και την αναλυτική μοντελοποίηση. Τα αποτελέσματα και η ανάλυση πρέπει να υποβάλλονται στην αναφορά.

Στο τέλος όλοι οι φοιτητές κατάφεραν να ολοκληρώσουν την εργαστηριακή άσκηση και συγκεντρώθηκαν 12 αναφορές. Όλοι ήταν σε θέση να καταγράψουν τα απαιτούμενα δεδομένα και να πραγματοποιήσουν τις συγκρίσεις μεταξύ των αποτελεσμάτων του πειράματος και των αποτελεσμάτων προσομοίωσης και να εξηγήσουν ποιοτικά τις μη ιδανικές συμπεριφορές. Οι περισσότεροι από τους φοιτητές έδωσαν συνολικά θετικά σχόλια και δύο από αυτούς περίμεναν μια πιο δύσκολη εργαστηριακή εργασία.

Συνολικά μπορούν να εξαχθούν τα ακόλουθα συμπεράσματα:

- 1) Το διαδικτυακό εργαστήριο μπορεί να εφαρμοστεί για την εκπαίδευση Ηλεκτρικών Μηχανών είτε στον χώρο του Πανεπιστημίου είτε διαδικτυακά.
- 2) Μπορεί να συνδυαστεί με προσομοιώσεις και διαδραστικές αναφορές ώστε να καταστήσει τη διαδικασία μάθησης πιο ενεργή και αποτελεσματική.
- 3) Μειώνει τον χρόνο που απαιτείται για κάθε εργαστήριο, γεγονός που μπορεί να ωφελήσει τη μαζική εκπαίδευση.
- 4) Εκτός από τη διάταξη του εργαστηρίου, το λογισμικό και οι πλατφόρμες ΤΠΕ (Jupyter Notebook) είναι απαραίτητα για την εφαρμογή του διαδικτυακού εργαστηρίου.

Η εφαρμογή που επιλέχθηκε στο paper αφορά τις Ηλεκτρικές Μηχανές, ωστόσο, αυτή η μεθοδολογία μπορεί να εφαρμοστεί και σε άλλους τομείς Ηλεκτρολόγων Μηχανικών, π.χ. Ηλεκτρονικά Ισχύος και Μικροδίκτυα. Οι διδάσκοντες του TU Delft σημειώνουν επίσης ότι η διαδικτυακή πλατφόρμα παρέχει μια λύση για την πρακτική εκπαίδευση και την έρευνα κατά τη διάρκεια της πανδημίας COVID-19.

## 2.5 Πρόσβαση στο Jupyter Notebook

Αφού παρουσιάσαμε τα χαρακτηριστικά του Jupyter Notebook, τα πλεονεκτήματα της χρήσης του ως εκπαιδευτικό περιβάλλον καθώς και αντιπροσωπευτικά παραδείγματα αξιοποίησης του στην εκπαιδευτική διαδικασία, μπορούμε να προχωρήσουμε στο επόμενο βήμα που αφορά τον τρόπο με τον οποίο θα το χρησιμοποιήσουν οι φοιτητές. Η επιλογή αυτή εξαρτάται από τη μορφή του μαθήματος (διαλέξεις, εργαστήριο, με φυσική παρουσία, διαδικτυακά κ.λπ.), τους πόρους που παρέχονται από το Πανεπιστήμιο, τις δυνατότητες των φοιτητών και άλλους παράγοντες που θα αναλυθούν στη συνέχεια.

### 2.5.1 Τοπική εγκατάσταση

Στην πράξη, η πρόσβαση των φοιτητών στα Jupyter Notebooks, μπορεί να πραγματοποιηθεί με διάφορους τρόπους. Αρχικά, μπορεί να γίνει τοπική εγκατάσταση, που σημαίνει ότι κάθε φοιτητής καλείται να εγκαταστήσει και να τρέξει το Jupyter στον υπολογιστή του. Μια δημοφιλής διανομή λογισμικού που περιλαμβάνει το Jupyter αποτελεί το Anaconda, το οποίο είναι εύκολο να εγκατασταθεί σε Windows, Mac και Linux.

Η λύση αυτή έχει πλεονεκτήματα, όπως ότι από τη στιγμή που οι φοιτητές θα εγκαταστήσουν το λογισμικό στους υπολογιστές τους, θα έχουν πάντα πρόσβαση σε αυτό και θα μπορούν να δουλέψουν οπουδήποτε, να το χρησιμοποιήσουν για οποιοδήποτε μάθημα, εργασία ή δραστηριότητα εκτός του προγράμματος σπουδών τους, και έτσι να εξοικειωθούν με ένα εργαλείο χρήσιμο στην ακαδημαϊκή και επαγγελματική τους πορεία.

Ωστόσο υπάρχουν και μειονεκτήματα όπως ότι κάθε φοιτητής θα πρέπει να έχει υπολογιστή με αρκετή χωρητικότητα και υπολογιστική ισχύ. Επίσης παρόλο που η εγκατάσταση είναι σχετικά εύκολη, απαιτεί χρόνο και η διαδικασία της διαφέρει ανάλογα με τον τύπο του λειτουργικού συστήματος του κάθε υπολογιστή (Windows, Mac, Linux), γεγονός που μπορεί να ταλαιπωρήσει φοιτητές και εκπαιδευτικούς.

Διαφορετικά το Jupyter μπορεί να εγκατασταθεί τοπικά στους υπολογιστές του εργαστηρίου, ώστε όλοι οι φοιτητές να έχουν εγγυημένα πρόσβαση σε αρκετή υπολογιστική ισχύ και να μειωθεί ο χρόνος τεχνικής υποστήριξης που θα χρειαζόταν για την εγκατάσταση του προγράμματος σε ατομικούς υπολογιστές. Ωστόσο αυτό συνοδεύεται από περιορισμούς στο χρονικό διάστημα στο οποίο μπορούν να εργαστούν οι φοιτητές και ίσως κάποιες δυσκολίες στη διατήρηση της σωστής λειτουργίας του λογισμικού σε όλα τα μηχανήματα, καθώς μπορεί να χρειαστούν επιπλέον ενημερώσεις και εγκαταστάσεις πακέτων.

### 2.5.2 Χρήση μέσω απομακρυσμένου server

Μια εναλλακτική προσέγγιση είναι η χρήση του Jupyter μέσω απομακρυσμένου server. Σε αυτή την περίπτωση, οι φοιτητές δεν χρειάζεται να εγκαταστήσουν τίποτα. Αρκεί να ανοίξουν ένα πρόγραμμα περιήγησης (browser) και να φορτώσουν μια σελίδα (URL). Υπάρχουν διάφοροι τρόποι να τρέξει το Jupyter σε απομακρυσμένο server:

1. Να τρέξει σε έναν server του εκπαιδευτικού ιδρύματος.
2. Να τρέξει σε ένα προσωρινό περιβάλλον στο cloud.
3. Να τρέξει σε ένα μόνιμο περιβάλλον στο cloud.

Με αυτό τον τρόπο όλοι οι φοιτητές έχουν πρόσβαση στους ίδιους πόρους, με τη διαφορά ότι μπορούν να τους χρησιμοποιήσουν οπουδήποτε και η δουλειά τους αποθηκεύεται αυτόματα στο cloud, οπότε δεν υπάρχει περίπτωση να χαθεί.

Η παροχή πρόσβασης στους φοιτητές μέσω κάποιου server του εκπαιδευτικού ιδρύματος μπορεί να πραγματοποιηθεί μέσω μιας υπηρεσίας που λέγεται JupyterHub [56] και αποτελεί ένα λογισμικό ανοιχτού κώδικα που προσφέρει μια cloud-based εφαρμογή του Jupyter για κάθε χρήστη σε μια ομάδα. Οι χρήστες μπορούν να τρέχουν και να αποθηκεύουν notebooks και άλλα αρχεία σε έναν απομακρυσμένο υπολογιστή, καθώς και να τα κατεβάζουν στον υπολογιστή τους και να δουλεύουν σε κάποια τοπική εγκατάσταση του Jupyter, αν το επιθυμούν. Με αυτόν τον τρόπο οι φοιτητές δεν χάνουν χρόνο στην εγκατάσταση λογισμικού και μπορούν να ασχοληθούν κατευθείαν με το περιεχόμενο του μαθήματός τους. Ωστόσο η εγκατάσταση και διαμόρφωση του JupyterHub, μπορεί να είναι πιο επίπονη για τους εκπαιδευτικούς γιατί απαιτεί πιο εξειδικευμένες γνώσεις σε θέματα που αφορούν τα λειτουργικά συστήματα, τη διαχείριση και τη δικτύωσή τους.

Η πιο εύκολη επιλογή για την εκτέλεση του Jupyter στο cloud είναι η χρήση κάποιας δωρεάν υπηρεσίας που παρέχει ένα προσωρινό περιβάλλον το οποίο μπορεί να χρησιμοποιηθεί χωρίς καμία εγκατάσταση. Ένα τέτοιο παράδειγμα αποτελεί το Binder [57], το οποίο επιτρέπει στον κάτοχο ενός συνόλου αρχείων Jupyter Notebooks που είναι αποθηκευμένα σε κάποιο αποθετήριο όπως το GitHub, να τα δημοσιεύει με ένα link σε οποιοδήποτε χρήστη, δίνοντας του την ευκαιρία να εκτελέσει ζωντανά τον κώδικα. Το περιβάλλον αυτό είναι ιδανικό για μικρά παραδείγματα σε μαθήματα που δεν πρόκειται να γίνει εκτενής χρήση του Jupyter. Ωστόσο παρουσιάζει ορισμένους περιορισμούς, όπως ότι αν τα notebooks βασίζονται σε πιο εξειδικευμένα πακέτα, μπορεί να μην πληρούνται οι προϋποθέσεις για την εκτέλεσή τους καθώς επίσης ότι δεν είναι ιδιαίτερα αξιόπιστο, αφού πρόκειται για προσωρινό περιβάλλον και οποιαδήποτε αλλαγή στο αρχείο θα χαθεί, μετά το κλείσιμο της καρτέλας.

Τέλος υπάρχουν διάφορες εμπορικές λύσεις για χρήση του Jupyter στο cloud που παρέχουν δωρεάν υπηρεσίες. Ενδεικτικά αναφέρουμε τις παρακάτω υπηρεσίες:

- CoCalc [58] είναι ένα online υπολογιστικό περιβάλλον ανοιχτού κώδικα που υποστηρίζει το Jupyter Notebook. Επιτρέπει σε πολλαπλούς χρήστες να επεξεργάζονται ένα notebook ταυτόχρονα, και παρέχει αυτόματα μια ποικιλία δημοφιλών πυρήνων (kernels), όπως το Octave που είναι σε μεγάλο βαθμό συμβατό με το MATLAB. Είναι δωρεάν στη χρήση και υπάρχει δυνατότητα παροχής επιπλέον υπολογιστικών πόρων καταβάλλοντας μηνιαία, ετήσια ή με βάση το εκάστοτε μάθημα, τέλη συνδρομής.
- Google Colaboratory [59]
- Kaggle Kernels [60]

- Microsoft Azure notebooks [61]

Το μεγαλύτερο πλεονέκτημα αυτών των υπηρεσιών είναι ότι δεν απαιτείται από τον χρήστη καμία εγκατάσταση, παρά μόνο η δημιουργία ενός λογαριασμού. Ωστόσο παρουσιάζουν και ορισμένα μειονεκτήματα όπως ότι μπορεί να είναι δύσκολη ή αδύνατη η εγκατάσταση επιπλέον πακέτων που μπορεί να χρειαστούν και ότι η εξασφάλιση επιπλέον χωρητικότητας και υπολογιστικών πόρων μπορεί να είναι ακριβή [62].

Για την ανάπτυξη του υλικού αυτής της εργασίας έγινε τοπική εγκατάσταση του Jupyter Notebook σε προσωπικό υπολογιστή. Επίσης χρησιμοποιήθηκαν οι δωρεάν cloud πλατφόρμες CoCalc και Google Colaboratory (Colab), ώστε να παρουσιαστεί και η χρήση μιας πιο εύκολης εναλλακτικής.

### 2.5.3 Υποστήριξη λειτουργίας του Jupyter Notebook

Επιπλέον, για τη διευκόλυνση των χρηστών, ένα οικοσύστημα προσθηκών και επεκτάσεων ανοιχτού κώδικα, που συνεχώς ανανεώνονται, αναπτύσσεται γύρω από το Jupyter Notebook συμπληρώνοντάς το, όπως το Nbconvert, το Nbviewer [63], το Nbgrader [64] και τα Nbextensions.

- Το Nbconvert μετατρέπει τα αρχεία notebook σε άλλες μορφές όπως HTML, LaTeX, PDF και Slide Shows καθιστώντας τα προσβάσιμα χωρίς να χρειάζεται εγκατάσταση του λογισμικού του Jupyter.
- Το Nbviewer είναι μια διαδικτυακή υπηρεσία που παρέχει προβολή αρχείων notebook δημοσιευμένα οπουδήποτε στο διαδίκτυο, σε μορφή HTML, με δυνατότητα προβολής του αρχικού συνδέσμου του αρχείου από όπου, αν ο αναγνώστης ενδιαφέρεται, μπορεί να το κατεβάσει και να το τροποποιήσει.
- Το Nbgrader είναι ένα εργαλείο σχεδιασμένο να διευκολύνει την διαδικασία βαθμολόγησης και ταυτόχρονα απλοποιεί τις διαδικασίες της δημιουργίας, διανομής, συλλογής και αξιολόγησης εργασιών.
- Τα Nbextensions είναι απλά πρόσθετα που ενσωματώνονται στο Jupyter Notebook και με μια απλή ενεργοποίηση επεκτείνουν τη βασική λειτουργικότητά του. Για παράδειγμα υπάρχουν επεκτάσεις για τη δημιουργία πίνακα περιεχομένων (Table of Contents), για την αυτόματη μορφοποίηση του κώδικα (Autoper8), την αυτόματη αρίθμηση εξισώσεων (Equation Auto Numbering) και πολλές ακόμη λειτουργίες όπως φαίνεται στο Σχήμα 2.9.

The screenshot shows the Jupyter 'Configurable nbextensions' interface. At the top, there is a 'Quit' and 'Logout' button. Below the navigation tabs (Files, Running, Clusters, Nbextensions), the title 'Configurable nbextensions' is displayed. A checkbox at the top left allows disabling configuration for nbextensions without explicit compatibility. A search filter is provided below. The main area contains a grid of nbextension names, each with a checkbox. The 'Snippets Menu' extension is highlighted in blue. Other visible extensions include 'Autopep8', 'CodeMirror mode extensions', 'Gist-it', 'Keyboard shortcut editor', and 'Table of Contents (2)'. The 'jupyter-js-widgets/extension' and 'plotlywidget/extension' are also visible and highlighted in yellow.

Σχήμα 2. 9 : Πίνακας των Nbextentions.

Αξιοποιώντας, μεταξύ άλλων, αυτές τις υπηρεσίες, το υπολογιστικό περιβάλλον, τα δεδομένα, ο επιστημονικός κώδικας, και οι συνθήκες εκτέλεσης ενός Jupyter Notebook μπορούν να αποθηκευτούν σε ένα αρχείο στο GitHub, επιτρέποντας την επαναχρησιμοποίησή του. Στα παραπάνω ως προστεθεί το γεγονός ότι πρόσφατα κυκλοφόρησε το επόμενης γενιάς Project Jupyter, το JupyterLab, το οποίο φέρει όλα τα δομικά στοιχεία του κλασικού Jupyter, καθώς και μερικά νέα στοιχεία (όπως καινούριο περιβάλλον εργασίας χρήστη, ευέλικτη διάταξη, νέες λειτουργίες, υποστήριξη επιπλέον μορφών αρχείων, επιπλέον επεκτάσεις κ.α.), ενσωματωμένα όλα κάτω από μια στέγη [32].

## Κεφάλαιο 3: Ανάπτυξη διαδραστικού υλικού για τη ρύθμιση τάσης

### Σύνοψη

*Τα Δίκτυα Διανομής έχουν σχεδιαστεί για να διανέμουν την ηλεκτρική ενέργεια στους καταναλωτές, με τη ροή της ισχύος να είναι προς μία μόνο κατεύθυνση. Ωστόσο, με τη σημαντική διεύρυνση της Διεσπαρμένης Παραγωγής (ΔΠ), η ροή της ισχύος μπορεί να αντιστραφεί με αποτέλεσμα την ανύψωση τάσης. Στο κεφάλαιο αυτό μελετάται το θεωρητικό υπόβαθρο της συμβολής της ΔΠ στη ρύθμιση τάσης και με βάση αυτό, παρουσιάζεται μια διαδραστική εφαρμογή που αναπτύχθηκε στο περιβάλλον του *Jupyter Notebook*, σε γλώσσα *Python*, με στόχο την αύξηση της κατανόησης του θέματος αυτού από τους φοιτητές μέσω ενός πρακτικού παραδείγματος.*

### 3.1 Ρύθμιση τάσης στα Δίκτυα Διανομής

Τα τελευταία χρόνια, τα δίκτυα διανομής και μεταφοράς έρχονται αντιμέτωπα με την αυξανόμενη ενσωμάτωση μονάδων ΔΠ, συχνά από ΑΠΕ, που λόγω της μεταβλητότητας και αβεβαιότητας στην παραγωγή τους, θέτουν νέες προκλήσεις στη λειτουργία του συστήματος. Καθώς προβλέπεται η μελλοντική παραγωγή ενέργειας να βασίζεται στις μονάδες αυτές θα πρέπει να είναι σε θέση να υποστηρίξουν το σύστημα παρέχοντας επικουρικές υπηρεσίες, δηλαδή όλες τις υπηρεσίες που απαιτούνται από τον διαχειριστή του δικτύου μεταφοράς ή διανομής για να διασφαλίσουν την ακεραιότητα και ευστάθεια του συστήματος μεταφοράς ή διανομής καθώς και την ποιότητα ισχύος. Οι επικουρικές υπηρεσίες παρέχονται από μονάδες παραγωγής, ελεγχόμενα φορτία και άλλες συσκευές [65].

Αναφέρονται οι παρακάτω επικουρικές υπηρεσίες:

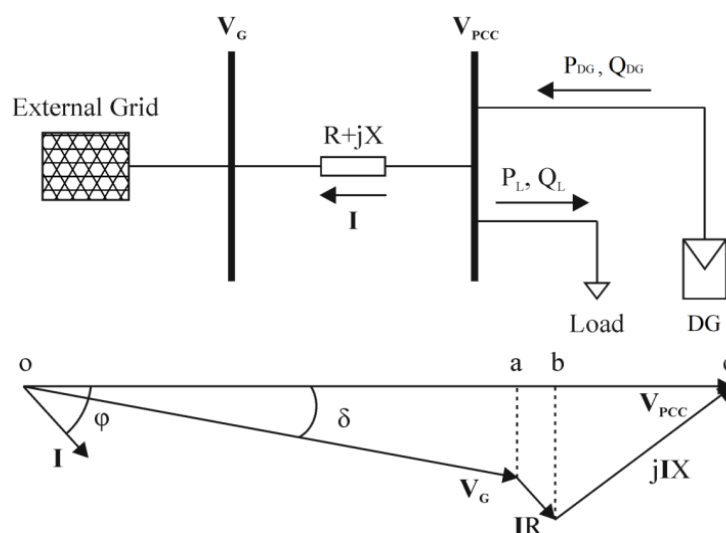
- Ρύθμιση συχνότητας
- Ρύθμιση τάσης
- Στρεφόμενη εφεδρεία
- Στατή εφεδρεία
- Δυνατότητα αυτόνομης εκκίνησης (black start)
- Απομακρυσμένος αυτόματος έλεγχος των γεννητριών
- Αντιστάθμιση απωλειών δικτύου
- Ενέργειες έκτακτης ανάγκης

Στο κεφάλαιο αυτό εστιάζουμε στη συμβολή της ΔΠ στη ρύθμιση τάσης, την οποία μελετάμε στην εφαρμογή που σχεδιάσαμε.

Παραδοσιακά η ρύθμιση τάσης στα δίκτυα διανομής γίνεται με τη χρήση συσκευών όπως μετασχηματιστές με ΣΑΤΥΦ, πυκνωτές κ.α., με βασικό σκοπό την αντιμετώπιση της πτώσης τάσης που προκαλείται από τη ροή ισχύος στις γραμμές για την κάλυψη του φορτίου. Με την σημαντική διεύρυνση πλήθους μονάδων ΔΠ στη μέση και στη



χαμηλή τάση, η παραγωγή από τις μονάδες ΔΠ μπορεί πλέον να καλύπτει σημαντικό μέρος του φορτίου ή ακόμα και να το ξεπερνάει προκαλώντας αναστροφή ροής ισχύος και άρα ανύψωση τάσης. Έτσι, σε ορισμένες περιπτώσεις (όταν το φορτίο είναι χαμηλό και η παραγωγή από ΔΠ υψηλή) η τάση μπορεί να ξεπεράσει τα επίπεδα που ορίζουν τα πρότυπα. Σε δίκτυο διανομής με μία κεντρική γραμμή, αν σε κάθε ζυγό η παραγωγή της ΔΠ είναι υψηλότερη από το φορτίο, η ανύψωση της τάσης γίνεται μεγαλύτερη όσο μεγαλώνει η απόσταση από τον μετασχηματιστή και το μεγαλύτερο πρόβλημα παρουσιάζεται στο άκρο της γραμμής [65].



Σχήμα 3. 1: Φορτίο με ενεργό και άεργο ισχύ ( $P_L, Q_L$ ) και μονάδα ΔΠ με ενεργό και άεργο ισχύ ( $P_{DG}, Q_{DG}$ ) που συνδέονται στο δίκτυο μέσω γραμμής διανομής (άνω), και το αντίστοιχο διανυσματικό διάγραμμα (κάτω) [65].

Η πτώση τάσης σε περίπτωση που φορτίο συνδέεται στο ανάντη δίκτυο μέσω γραμμής ωμικής αντίστασης  $R$  και επαγωγικής αντίδρασης  $X$ , δίνεται κατά προσέγγιση από την παρακάτω σχέση:

$$\Delta V \approx I \cdot R \cdot \cos\varphi + I \cdot X \cdot \sin\varphi \quad (3.1)$$

Μία απόδειξη της σχέσης με βάση το διανυσματικό διάγραμμα (Σχήμα 3.1) για περίπτωση ανύψωσης τάσης εξαιτίας της διείσδυσης ΔΠ παρατίθεται παρακάτω.

$$\widetilde{V}_{PCC} = \widetilde{V}_G + \widetilde{I} \cdot (R + jX) \quad (3.2)$$

Όπου  $V_G$  είναι η τάση του δικτύου και  $V_{PCC}$  η τάση του ζυγού στον οποίο συνδέεται το φορτίο ( $P_L, Q_L$ ) και η μονάδα ΔΠ ( $P_{DG}, Q_{DG}$ ). Για μικρή γωνία  $\delta$  θεωρείται η προβολή της  $V_G$  στον οριζόντιο άξονα και η μεταβολή της τάσης υπολογίζεται προσεγγιστικά ως:

$$\Delta V = V_{PCC} - V_G = oc - oa = ac = ab + bc \Rightarrow$$

$$\Delta V \approx I \cdot R \cdot \cos\varphi + I \cdot X \cdot \sin\varphi = \frac{V_{PCC} \cdot I \cdot R \cdot \cos\varphi + V_{PCC} \cdot I \cdot X \cdot \sin\varphi}{V_{PCC}} \Rightarrow$$

$$\Delta V \approx \frac{P_{PCC} \cdot R + Q_{PCC} \cdot X}{V_{PCC}} \approx \frac{P_{PCC} \cdot R + Q_{PCC} \cdot X}{V_G} \quad (3.3)$$

Όπου  $P_{PCC}$  και  $Q_{PCC}$  είναι η συνολική ενεργός και άεργος ισχύς που ρέει από τον ζυγό.

$$P_{PCC} = P_{DG} - P_L \quad (3.4)$$

$$Q_{PCC} = Q_{DG} - Q_L \quad (3.5)$$

Οπότε η σχέση (3.3) τροποποιείται ως:

$$\begin{aligned} \Delta V &\approx \frac{(P_{DG} - P_L) \cdot R + (Q_{DG} - Q_L) \cdot X}{V_{PCC}} \\ &\approx \frac{(P_{DG} - P_L) \cdot R + (Q_{DG} - Q_L) \cdot X}{V_G} \end{aligned} \quad (3.6)$$

Η ποσοστιαία μεταβολή της τάσης προκύπτει ως:

$$\varepsilon\% = \frac{\Delta V}{V_{PCC}} \cdot 100 \approx \frac{\Delta V}{V_G} \cdot 100 \approx \frac{P_{PCC} \cdot R + Q_{PCC} \cdot X}{V_G^2} \cdot 100 \quad (3.7)$$

Συνήθως η τάση του δικτύου ( $V_G$ ) θεωρείται πως έχει την ονομαστική της τιμή ( $V_n$ ). Σημειώνεται πως με την παραπάνω θεώρηση η  $\Delta V$  είναι θετική όταν παρουσιάζεται ανύψωση τάσης και αρνητική όταν παρουσιάζεται πτώση τάσης. Θεωρώντας πως η μονάδα ΔΠ λειτουργεί με μοναδιαίο συντελεστή ισχύος, παρουσιάζεται ανύψωση τάσης στο ζυγό της ΔΠ και του φορτίου όταν:

$$P_{DG} \cdot R > P_L \cdot R + Q_L \cdot X \quad (3.8)$$

Με βάση τη σχέση (3.6), οι βασικοί μέθοδοι αντιμετώπισης της ανύψωσης τάσης εξαιτίας της διείσδυσης ΔΠ αναφέρονται παρακάτω:

- Αναβάθμιση του δικτύου
- Απορρόφηση αέργου ισχύος από τις μονάδες ΔΠ
- Περιορισμός της ενεργού ισχύος των μονάδων ΔΠ
- Χρήση συστημάτων αποθήκευσης (π.χ. που απορροφούν ενεργό ισχύ όταν η παραγωγή από ΔΠ είναι υψηλή)
- Επίσης, η χρήση μετασχηματιστή με ΣΑΤΥΦ [65].

## 3.2 Εφαρμογή για τη ρύθμιση τάσης σε Jupyter Notebook

Με βάση το παραπάνω θεωρητικό υπόβαθρο αναπτύξαμε μια εφαρμογή που αφορά τη συμβολή της ΔΠ στη ρύθμιση τάσης, στο περιβάλλον του Jupyter Notebook. Στόχος της εφαρμογής είναι να αυξήσει την κατανόηση των φοιτητών για το θέμα αυτό μέσω της έκθεσης τους σε ένα διαδραστικό υπολογιστικό περιβάλλον το οποίο περιλαμβάνει συνοπτικά την θεωρία, τις απαραίτητες μαθηματικές εξισώσεις και τον κώδικα που εφαρμόζει τις εξισώσεις αυτές, και απεικονίζει τα αντίστοιχα αποτελέσματα.

Συγκεκριμένα, η εφαρμογή αυτή πραγματοποιήθηκε σε τοπική εγκατάσταση του Jupyter Notebook, σε υπολογιστή με λειτουργικό σύστημα Linux. Για την εγκατάσταση ακολουθήθηκαν οι οδηγίες που δίνονται στην επίσημη σελίδα του λογισμικού. Για το παράδειγμα αυτό επιλέχθηκε η γλώσσα προγραμματισμού Python καθώς υποστηρίζει πολλές εξειδικευμένες λειτουργίες του Jupyter Notebook, που χρειάζονται για την υλοποίηση του προγράμματος αυτού, είναι σχετικά εύκολη και υπάρχει πληθώρα υποστηρικτικού υλικού στο διαδίκτυο. Η εκτέλεση του συγκεκριμένου προγράμματος δεν θα ήταν δυνατή σε κάποια cloud πλατφόρμα (όπως το CoCalc και το Colab), καθώς δεν μπορούν να υποστηρίξουν ορισμένα από τα πακέτα που χρησιμοποιούνται.

### 3.1.1 Περιγραφή του προγράμματος

Στην παρακάτω προσομοίωση, μελετούμε τη συμβολή της ΔΠ στη ρύθμιση του φαινομένου της ανύψωσης τάσης. Αυτό γίνεται μέσω του προγράμματος που υλοποιήσαμε, το οποίο ουσιαστικά υπολογίζει και εμφανίζει κάθε στιγμή, σε μορφή γραφικών παραστάσεων τις τιμές της τάσης ( $V_{PCC}$ ) και της ενεργού και της αέργου ισχύος ( $P_{PCC}$ ,  $Q_{PCC}$ ) του ζυγού στον οποίο είναι συνδεδεμένα η μονάδα ΔΠ και το φορτίο, του Σχήματος 3.1. Ταυτόχρονα δίνεται η δυνατότητα αλληλεπίδρασης του χρήστη με το πρόγραμμα, μέσω τριών διαδραστικών στοιχείων εισόδου που δημιουργεί. Αυτά έχουν τη μορφή «κυλιόμενης μπάρας» και χρησιμοποιούνται για την ρύθμιση τριών παραμέτρων: i) της ενεργού ισχύος της μονάδας ΔΠ ( $P_{DG}$ ), του χωρητικού συντελεστή ισχύος ( $\cos\phi$ ) ή του επαγωγικού συντελεστή ισχύος ( $\cos\phi$ ) της μονάδας ΔΠ.

Κατά το πείραμα αυτό, παρατηρούμε πως η ενεργός ισχύς που παράγει η ΔΠ, καθώς επίσης και η δυνατότητά της να χειριστεί την άεργο ισχύ είναι καθοριστικής σημασίας για την τάση που αναπτύσσεται στον ζυγό. Η ανάστροφη ροή ενεργού ισχύος πάνω στη γραμμή διανομής τείνει να αυξήσει την τάση στη θέση της μονάδας ΔΠ. Η έγχυση της αέργου ισχύος από τη μονάδα ΔΠ (μέσω χωρητικού συντελεστή ισχύος), αυξάνει περεταίρω την τάση, ενώ αντίθετα η απορρόφηση αέργου ισχύος από τη μονάδα ΔΠ (μέσω επαγωγικού συντελεστή ισχύος), βοηθάει στον περιορισμό της ανύψωσης τάσης.

Τα Σχήματα 3.2 και 3.3 παρουσιάζουν το αρχείο Jupyter Notebook της εφαρμογής στη ρύθμιση τάσης. Όπως φαίνεται περιέχει τίτλο, συνοπτικό κείμενο που αφορά το θεωρητικό υπόβαθρο της προσομοίωσης, σχήμα που απεικονίζει το μονογραμμικό διάγραμμα του δικτύου με φορτίο και μονάδα ΔΠ συνδεδεμένα σε ζυγό και το αντίστοιχο διανυσματικό διάγραμμα, κελιά κώδικα, τις μαθηματικές εξισώσεις που

αφορούν τη ρύθμιση τάσης και χρησιμοποιούνται στο πρόγραμμά μας με αρίθμηση ώστε να γίνει η αντιστοιχία στον κώδικα, οδηγίες για τα βήματα που μπορεί να ακολουθήσει ο χρήστης και τέλος τα αποτελέσματα της προσομοίωσης.

Το αρχείο μας περιλαμβάνει συνολικά τέσσερα κελιά κώδικα (γκρι πλαίσια) και για να τρέξει το πρόγραμμά μας χρειάζεται να εκτελεστούν και τα τέσσερα. Αυτό γίνεται με την επιλογή Run All ή εκτελώντας με τη σειρά το καθένα ξεχωριστά, επιλέγοντας τα πατώντας τα πλήκτρα Shift + Enter.

Παρακάτω δίνουμε μια σύντομη επεξήγηση του κώδικα αυτού:

1. Το πρώτο κελί (Σχήμα 3.2) φαίνεται να μην περιέχει κώδικα αλλά στην πραγματικότητα είναι ένα αναδιπλωμένο κελί. Εκεί βρίσκονται όλα τα πακέτα και οι βιβλιοθήκες της Python που είναι απαραίτητα για να τρέξει το συγκεκριμένο πρόγραμμα. Σε αυτή την περίπτωση έχουμε «κρύψει» το περιεχόμενο του κελιού για εξοικονόμηση χώρου αλλά και επειδή δεν περιέχει κάποια πληροφορία που να συνδέεται άμεσα με την προσομοίωση. Ωστόσο, αν ο χρήστης επιθυμεί να εξοικειωθεί με τις διαδικασίες της Python μπορεί ανοίξει το κελί και να δει το περιεχόμενό του, πατώντας το βελάκι στα αριστερά.
2. Στο δεύτερο κελί (Σχήμα 3.2) κώδικα δηλώνονται όλες οι μεταβλητές που χρησιμοποιούνται στο πρόγραμμα και ανατίθενται τιμές στις σταθερές μεταβλητές. Συγκεκριμένα η ωμική αντίσταση και η επαγωγική αντίδραση της γραμμής διανομής ορίζονται ως  $R = 2.35 \Omega$  και  $X = 1.71 \Omega$ , η ενεργός και η άεργος ισχύς του φορτίου παίρνουν τις τιμές  $P_L = 300 \text{ kW}$  και  $Q_L = 227 \text{ kVAr}$  αντίστοιχα, και η φασική τάση του δικτύου θεωρείται ότι έχει την ονομαστική της τιμή  $V_G = 230 \text{ kV}$ .
3. Το τρίτο κελί (Σχήμα 3.3), το οποίο είναι επίσης αναδιπλωμένο, περιέχει τον κώδικα για τη δημιουργία των διαδραστικών στοιχείων εισόδου χρήστη για τη ρύθμιση των παραμέτρων της ενεργού ισχύος της μονάδας ΔΠ ( $P_{DG}$ ), του χωρητικού και του επαγωγικού συντελεστή ισχύος ( $\cos\phi$ ).
4. Το τέταρτο κελί (Σχήμα 3.3), περιέχει αρχικά τον κώδικα που εμφανίζει τα διαδραστικά στοιχεία. Έπειτα (αναδιπλωμένο για συντομία) περιέχει τον ορισμό της συνάρτησης plot, η οποία στη συνέχεια του προγράμματος χρησιμοποιείται ώστε να λαμβάνει τα αποτελέσματα των υπολογισμών ( $V_{PCC}$ ,  $P_{PCC}$ ,  $Q_{PCC}$ ) και να τα εμφανίζει συναρτήσει του χρόνου, δημιουργώντας τρεις γραφικές παραστάσεις οι οποίες μεταβάλλονται «ζωντανά». Επίσης περιλαμβάνει το βασικό πρόγραμμα που εκτελεί τους υπολογισμούς των εξισώσεων της ρύθμισης τάσης. Το πρόγραμμα, παρακολουθεί την τιμή των διαδραστικών στοιχείων κάθε στιγμή και μέσω των εξισώσεων που εκτελούνται διαδοχικά σε ένα while loop, υπολογίζονται συνεχώς οι τιμές της τάσης ( $V_{PCC}$ ) και της ενεργού και της άεργου ισχύος ( $P_{PCC}$ ,  $Q_{PCC}$ ) του ζυγού στον οποίο είναι συνδεδεμένα η μονάδα ΔΠ και το φορτίο, και εμφανίζονται μέσω της plot στις αντίστοιχες γραφικές παραστάσεις. Η διαδικασία που ακολουθείται μέσα στο while loop είναι η εξής:

- Αρχικά λαμβάνονται τα δεδομένα για την ενεργό ισχύ  $P_{DG}$  και τον συντελεστή ισχύος  $\cos\varphi$  από τα διαδραστικά στοιχεία.
- Μέσα στο loop, δύο συνθήκες ελέγχουν την τιμή του  $\cos\varphi$  και αν πρόκειται για επαγωγικό ή χωρητικό συντελεστή ισχύος. Πρώτα υπολογίζεται η γωνία  $\varphi$  μέσω της τιμής του συντελεστή ισχύος από τη σχέση (3.9).

$$\varphi = \cos^{-1}\varphi \quad (3.9)$$

- Στη συνέχεια υπολογίζεται η τιμή της αέργου ισχύος της μονάδας ΔΠ,  $Q_{DG}$  από τη σχέση (3.10), η οποία προκύπτει από τις σχέσεις (3.11) και (3.12).

$$Q_{DG} = P_{DG} \cdot \tan\varphi \quad (3.10)$$

$$P_{DG} = V_{PCC} \cdot I \cdot \cos\varphi \quad (3.11)$$

$$Q_{DG} = V_{PCC} \cdot I \cdot \sin\varphi \quad (3.12)$$

- Εάν πρόκειται για επαγωγικό συντελεστή ισχύος τότε η μονάδα ΔΠ απορροφά άεργο ισχύ οπότε η τιμή της αέργου ισχύος  $Q_{PCC}$  του ζυγού στον οποίο είναι συνδεδεμένα η μονάδα ΔΠ και το φορτίο δίνεται από τη σχέση (3.13), ενώ αν πρόκειται για χωρητικό συντελεστή ισχύος τότε εγχέεται άεργος ισχύς από τη μονάδα ΔΠ και η τιμή της αέργου ισχύος  $Q_{PCC}$  του ζυγού δίνεται από τη σχέση (3.5).

$$Q_{PCC} = -Q_{DG} - Q_L \quad (3.13)$$

- Στη συνέχεια υπολογίζεται η τιμή της ενεργού ισχύος  $P_{PCC}$  του ζυγού, από τη σχέση (3.4) και η τιμή της τάσης  $V_{PCC}$  από τη σχέση (3.14), όπου  $\Delta V$  δίνεται από τη σχέση (3.6).

$$\Delta V = V_{PCC} - V_G \Rightarrow V_{PCC} = \Delta V + V_G \quad (3.14)$$

- Οι τιμές των υπολογιζόμενων μεγεθών εμφανίζονται ακριβώς από κάτω σε τρεις γραφικές παραστάσεις, συναρτήσει του χρόνου.

**Η συμβολή της Διεσπαρμένης Παραγωγής στη ρύθμιση τάσης**

Τα σύγχρονα συστήματα δυνάμεις έχουν σχεδιαστεί για να διακρίνουν τα μέρη στους καταναλωτές, με τη ροή της πραγματικής και της άεργου ισχύος να είναι πάντα από το υψηλότερο προς στο χαμηλότερο επίπεδο τάσης.

Ωστόσο, με τη σημαντική διεύθυνση της Διεσπαρμένης Παραγωγής (ΔΠ), η ροή της ισχύος μπορεί να αντιστραφεί με αποτέλεσμα την ανάσχεση τάσης. Έτσι, σε ορισμένες περιπτώσεις, η τάση μπορεί να ξεπεράσει το ανώτατο όριο.

Σχήμα 1: Δίκτυο με διεσπομένη ΔΠ

Στο Σχήμα 1 φαίνεται το φορτίο με ενεργό και άεργο ισχύ ( $P_L, Q_L$ ) και η μονάδα ΔΠ με ενεργό και άεργο ισχύ ( $P_{DG}, Q_{DG}$ ) που συνδέονται στο δίκτυο μέσω γραμμής αμφοτέρωθεν αντίστασης  $R$  και επαγωγικής αντίδρασης  $X$ .

$V_G$  είναι η τάση του δικτύου και  $V_{RCC}$  η τάση του ζυγού φορτίου.

$P_{RCC}$  και  $Q_{RCC}$  είναι η συνολική ενεργός και άεργος ισχύς που ρέει από τον ζυγό φορτίου.

Στην παρακάτω προσομοίωση θα μελετήσουμε τη συμβολή της ΔΠ στη ρύθμιση του φαινομένου της ανάσχεσης τάσης.

- Θα ορίσουμε ότι η άεργος ενεργός ισχύος από μια μονάδα ΔΠ ( $P_{DG}$ ) στη γραμμή δυνάμεις τείνει να αυξήσει την τάση στο σημείο συνόδου της μονάδας ΔΠ ( $V_{RCC}$ ).
- Η άεργος άεργος ισχύος από τη μονάδα ΔΠ ( $Q_{DG}$ ), μέσω της ρύθμισης του χωρητικού συντελεστή ισχύος (cos φ), αυξάνει περαιτέρω την τάση.
- Για αντίστροφη απορρόφηση άεργου ισχύος από τη μονάδα ΔΠ ( $Q_{DG}$ ), μέσω της ρύθμισης του επαγωγικού συντελεστή ισχύος (cos φ) ρυθμίζει στη μείωση της τάσης.

Στα παρακάτω τμήμα κώδικα δίνονται οι απαραίτητες βιβλιοθήκες της Python για την εκτέλεση του προγράμματος. Χάρη συντομίες παρουσιάζουμε αυτό το κελί αναδιπλωμένο, αλλά το περιεχόμενό του μπορεί να εμφανιστεί πατώντας το βλ. Ακόμα δείξαμε η γραμμή από το σχήμα.

```
In [1]: # Imports
```

Στη συνέχεια ορίζονται οι μεταβλητές που θα χρησιμοποιηθούν.

- $R = 2.35\Omega, X = 1.71\Omega, P_L = 300kW, Q_L = 227kVAr$
- Η τάση του δικτύου θεωρείται ότι έχει την ονομαστική της τιμή  $V_G = 230kV$

```
In [2]: global Pdg
global Qdg
global P1
global Q1
global Prcc
global Qrcc
global pasc_pf_in
global pasc_pf_out
R = 2.35
X = 1.71
P1 = 300
Q1 = 227
Vg = 230
```

Το παρακάτω τμήμα κώδικα δημιουργεί και εμφανίζει 3 διαδραστικά στοιχεία κατόπιν χρήστη για την χειροκίνητη ρύθμιση των τιμών της ενεργού ισχύος της μονάδας ΔΠ ( $P_{DG}$ ), του επαγωγικού και του χωρητικού συντελεστή ισχύος (cos φ).

Μετά με ένα while loop υπολογίζονται κλίμα κλιμαί οι τιμές των  $V_{RCC}$ ,  $P_{RCC}$  και  $Q_{RCC}$  από τις παρακάτω σχέσεις και οι αλλαγές τους παρουσιάζονται σε τρεις γραφικές παραστάσεις σε συνάρτηση με τον χρόνο.

Η γωνία φ υπολογίζεται από τη σχέση

$$\varphi = \cos^{-1} \text{cos} \varphi \quad (1)$$

Από τις σχέσεις:

$$P_{DG} = V_{RCC} \cdot I \cdot \cos \varphi \quad (2a)$$

$$Q_{DG} = V_{RCC} \cdot I \cdot \sin \varphi \quad (2b)$$

προκύπτει η σχέση:

$$Q_{DG} = P_{DG} \cdot \tan \varphi \quad (2)$$

Η ενεργός και άεργος ισχύς της μονάδας ΔΠ δίνεται από τις σχέσεις:

$$P_{RCC} = P_{DG} - P_L \quad (3)$$

$$Q_{RCC} = -Q_{DG} - Q_L \quad (4a)$$

$$Q_{RCC} = -Q_{DG} - Q_L \quad (4b)$$

Η σχέση (4a) ισχύει όταν η μονάδα ΔΠ λειτουργεί με επαγωγικό ΣΙ, οπότε απορροφά άεργο ισχύ. Αντίστροφη η σχέση (4b) ισχύει όταν η μονάδα λειτουργεί με χωρητικό ΣΙ, έπεται παράγει άεργο ισχύ!

Σχήμα 3. 2 : Στιγμιότυπο από το αρχείο Jupyter Notebook για την εφαρμογή στη ρύθμιση τάσης που περιλαμβάνει τίτλο, κείμενο, σχήμα, εξισώσεις και τον κώδικα για την δήλωση μεταβλητών(1/3).

Επόμενη:

Η μεταβολή της τάσης υπολογίζεται παραγωγικά ως:

$$\Delta V = \frac{P_{PCC} \cdot R + Q_{PCC} \cdot X}{V_{PCC}} \approx \frac{P_{PCC} \cdot R + Q_{PCC} \cdot X}{V_G} \quad (5)$$

Από τη σχέση (5) υπολογίζεται η τάση του όγκου φορτίου:

$$\Delta V = V_{PCC} - V_G \Rightarrow V_{PCC} = \Delta V + V_G \quad (6)$$

Ακολουθώντας τα παρακάτω βήματα μπορούμε να παρατηρήσουμε την επίδραση της μόνιας ΔΤ στη ρύθμιση τάσης:

1. Αρχικό η ενεργός ισχύς που παράγει η ΔΤ είναι  $P_{DG} = 300 \text{ kW}$  και  $\cos\phi = 1$  ορα  $Q_{DG} = 0$ .
2. Αυξάνουμε την ενεργή ισχύ στην τιμή  $P_{DG} = 350 \text{ kW}$  διατηρώντας το  $\cos\phi$  σταθερό και παρατηρούμε την αύξηση της τάσης  $V_{PCC}$ . Λόγω της απώλειας της ενεργού ισχύος πάνω στη γραμμή διασποράς.
3. Αλλάζουμε σφαιρικά την ενεργή ισχύ και ρυθμίζοντας το  $\cos\phi = 0.9$  χωρικό παρατηρούμε ότι η τάση  $V_{PCC}$  αυξάνεται περαιτέρω λόγω της γρήγοτης πηγών ισχύος από τη μόνια ΔΤ.
4. Διατηρώντας σταθερή την ενεργή ισχύ και ρυθμίζοντας το  $\cos\phi = 0.9$  επαγωγικό παρατηρούμε μείωση της τάσης  $V_{PCC}$ , καθώς απορροφάται άεργος ισχύς από τη μόνια ΔΤ.

```

In [3]: # Slider variables

In [4]: out = widgets.interactive_output(f, {'w': w, 'pf_in': pf_in, 'pf_ca': pf_ca})
display(out)

def train():
    sTime = np.array(1)
    sTemp = np.array(1)
    stemp1 = np.array(1)
    stemp2 = np.array(1)

    while True:
        try:
            # Αν cosφ επαγωγικός ή cosφ = 1
            if (pf_in.value == 1 and pf_ca.value == 1) or (pf_in.value != past_pf_in):
                phi = math.acos(pf_in.value) # (1)
                Pdg = w.value # (2)
                Qdg = Pdg * math.tan(phi) # (2a)
                Qpcc = -Qdg - Ql # (4a)

                # Αν cosφ χωρτικός
            elif pf_ca.value != past_pf_ca:
                phi = math.acos(pf_ca.value) # (1)
                Pdg = w.value # (2)
                Qdg = Pdg * math.tan(phi) # (2)
                Qpcc = Qdg - Ql # (4b)

            Ppcc = Pdg - Pl # (3)
            DV = (Ppcc * R + Qpcc * X) / Vg # (5)
            Vpcc = Vg + DV # (6)

            print("Vpcc =", Vpcc, "kV", ", Ppcc =", Ppcc, "kW", ", Qpcc =", Qpcc, "kVAR")

            newTime = datetime.now()
            temp = Vpcc
            sTime = np.append(sTime, newTime) # Store the values
            stemp = np.append(stemp, temp)
            stemp1 = np.append(stemp1, Ppcc)
            stemp2 = np.append(stemp2, Qpcc)
            plot(sTime, stemp, stemp1, stemp2)

        except KeyboardInterrupt:
            break

    try:
        thread = threading.Thread(target=train, args=())
        thread.start()
    except:
        print("error")

```

300  1.00  1.00

[300, 1.0, 1.0]

Vpcc = 228.8452688695652 kV , Ppcc = 0 kW , Qpcc = -227.0 kVAR

Vpcc = 228.8452688695652 kV , Ppcc = 0 kW , Qpcc = -227.0 kVAR

In [ ]:

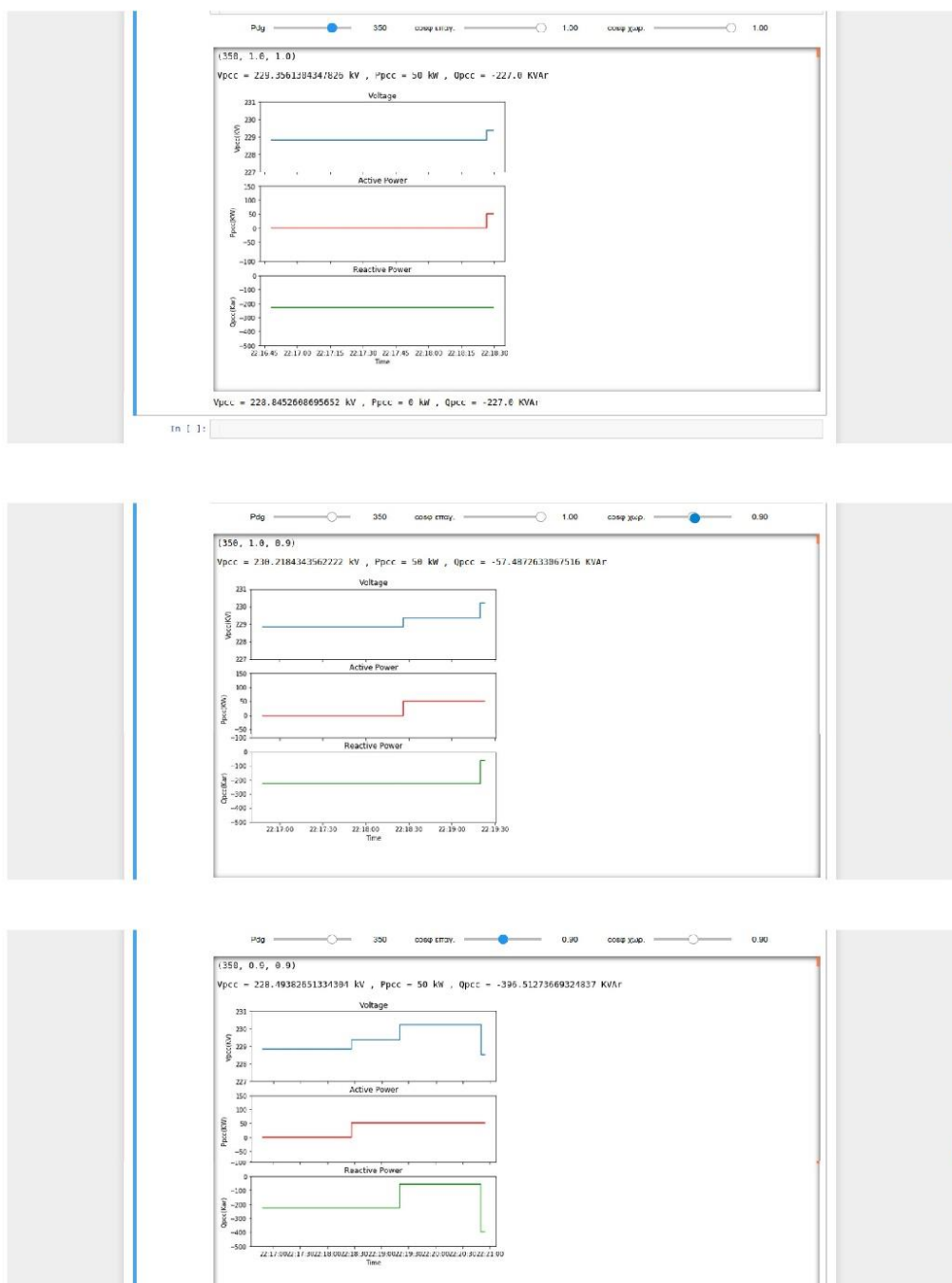
Σχήμα 3. 3 : Στιγμιότυπο από το αρχείο Jupyter Notebook για την εφαρμογή στη ρύθμιση τάσης που περιλαμβάνει το βασικό πρόγραμμα για την εκτέλεση των υπολογισμών και την εμφάνιση των γραφικών παραστάσεων των  $V_{PCC}$ ,  $P_{PCC}$  και  $Q_{PCC}$  (2/3).

### 3.1.2 Παράδειγμα εκτέλεσης του προγράμματος

Μπορούμε να παρατηρήσουμε την επίδραση της μονάδας ΔΠ στη ρύθμιση τάσης ακολουθώντας τα παρακάτω βήματα:

1. Τρέχουμε το πρόγραμμα διατηρώντας τις αρχικές τιμές που έχουμε ορίσει, δηλαδή  $P_{DG} = 300 \text{ kW}$  και συντελεστή ισχύος της μονάδας ΔΠ  $\cos\phi = 1$ . Όπως φαίνεται στο Σχήμα 3.3 τα αποτελέσματα που προκύπτουν είναι  $V_{PCC} = 228.845 \text{ kV}$ ,  $P_{PCC} = 0 \text{ kW}$  και  $Q_{PCC} = -227 \text{ kVAr}$ .
2. Στη συνέχεια αυξάνοντας την ενεργό ισχύ που παράγει η ΔΠ στην τιμή  $P_{DG} = 350 \text{ kW}$  και διατηρώντας τον συντελεστή ισχύος της μονάδας ΔΠ  $\cos\phi = 1$ , παρατηρούμε στο πρώτο στιγμιότυπο του Σχήματος 3.4 ότι η τάση αυξάνεται και η ενεργός ισχύ στον ζυγό αυξάνονται. Συγκεκριμένα  $V_{PCC} = 229.356 \text{ kV}$ ,  $P_{PCC} = 50 \text{ kW}$  και  $Q_{PCC} = -227 \text{ kVAr}$ . Εφόσον ο συντελεστής ισχύος παραμένει μοναδιαίος,  $Q_{DG} = 0 \text{ kVAr}$ , άρα δεν επηρεάζεται η τιμή της αέργου ισχύος του ζυγού.
3. Στο επόμενο βήμα διατηρούμε την ενεργό ισχύ της ΔΠ,  $P_{DG} = 350 \text{ kW}$  και ρυθμίζουμε τον συντελεστή ισχύος  $\cos\phi = 0.9$  χωρητικό. Όπως παρατηρούμε στο δεύτερο στιγμιότυπο του Σχήματος 3.4, η τάση του ζυγού αυξάνεται περεταίρω και επίσης μειώνεται η συνολική απορρόφηση αέργου ισχύος καθώς πλέον η ΔΠ παράγει άεργο ισχύ. Οι νέες τιμές είναι  $V_{PCC} = 230.218 \text{ kV}$ ,  $P_{PCC} = 50 \text{ kW}$  και  $Q_{PCC} = -57.487 \text{ kVAr}$ .
4. Τέλος, διατηρώντας σταθερή την ενεργό ισχύ που παράγει η ΔΠ,  $P_{DG} = 350 \text{ kW}$ , ρυθμίζουμε τον συντελεστή ισχύος  $\cos\phi = 0.9$  επαγωγικό. Στο τρίτο στιγμιότυπο του Σχήματος 3.4 φαίνεται ότι ο επαγωγικός χαρακτήρας συμβάλει στη μείωση της τάσης του ζυγού και άρα στον περιορισμό της ανύψωσης τάσης. Συγκεκριμένα  $V_{PCC} = 228.494 \text{ kV}$ ,  $P_{PCC} = 50 \text{ kW}$  και  $Q_{PCC} = -396.513 \text{ kVAr}$ .





Σχήμα 3. 4 : Στιγμιότυπα από το αρχείο Jupyter Notebook για την εφαρμογή στη ρύθμιση τάσης που δείχνουν τις γραφικές παραστάσεις των  $V_{PCC}$ ,  $P_{PCC}$  και  $Q_{PCC}$  για i) αύξηση της  $P_{DG}$ , ii) ρύθμιση χωρητικού ΣΙ και iii) ρύθμιση επαγωγικού ΣΙ (3/3).

## Κεφάλαιο 4: Ανάπτυξη διαδραστικού υλικού στη ροή φορτίου

### Σύνοψη

Η ανάλυση ροών φορτίου είναι θεμελιώδους σημασίας για τη μελέτη των ΣΗΕ, ως εκ τούτου αποτελεί αναπόσπαστο κομμάτι της εκπαίδευσης των φοιτητών ηλεκτρολόγων μηχανικών. Στο παρόν κεφάλαιο παρουσιάζουμε το θεωρητικό υπόβαθρο της θεμελίωσης του προβλήματος ροής φορτίου και τις τεχνικές επίλυσής του, με έμφαση στη μέθοδο Newton – Raphson. Στη συνέχεια εξετάζουμε το εκπαιδευτικό υλικό που χρησιμοποιείται στο εργαστήριο του μαθήματος Ανάλυση ΣΗΕ Ι της ΣΗΜΜΥ-ΕΜΠ στο λογισμικό MATLAB, το οποίο αξιοποιούμε για την δημιουργία αντίστοιχου εκπαιδευτικού υλικού στο Jupyter Notebook. Έπειτα περιγράφουμε τρεις εφαρμογές που υλοποιήσαμε σε γλώσσες MATLAB και Python οι οποίες μπορούν να αξιοποιηθούν για τις ανάγκες του μαθήματος στο εργαστήριο και πιθανώς στο αμφιθέατρο

### 4.1 Ροή Φορτίου

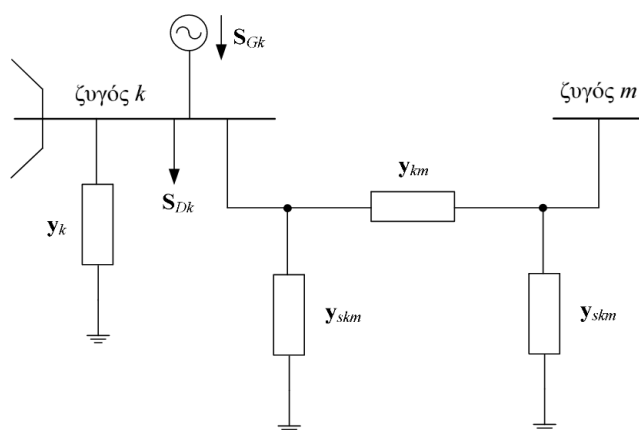
Η ανάλυση ροών φορτίου (ή ροής ισχύος) συνίσταται στον υπολογισμό των τάσεων κατά μέτρο και γωνία σε όλους τους ζυγούς καθώς και των ροών ενεργού και αέργου ισχύος σε όλες τις γραμμές ενός ΣΗΕ για ένα δεδομένο σημείο λειτουργίας, το οποίο ορίζεται από τις καταναλώσεις των φορτίων, την παραγωγή ενεργού ισχύος και τις τάσεις των γεννητριών. Οι μελέτες ροών φορτίου είναι πολύ χρήσιμες για διάφορους λόγους. Μερικοί από αυτούς αναφέρονται παρακάτω:

1. Είναι απαραίτητες για την επιλογή της πλέον οικονομικής λειτουργίας των γεννητριών του συστήματος. Επειδή στην περίοδο της ημέρας τα φορτία μεταβάλλονται συνεχώς, απαιτείται συχνά ο υπολογισμός εκ νέου της παραγόμενης ισχύος κάθε γεννήτριας.
2. Είναι απαραίτητες για τη διατήρηση τάσεων και ροών εντός προκαθορισμένων ορίων λειτουργίας.
3. Είναι απαραίτητες στη μελέτη ενδεχόμενων διαταραχών
4. Είναι απαραίτητες σε μελέτες επέκτασης του συστήματος παραγωγής και μεταφοράς ηλεκτρικής ενέργειας [75].

Η διαδικασία εύρεσης λύσης στο πρόβλημα της ροής φορτίου επιτυγχάνεται με τη βοήθεια επαναληπτικών αριθμητικών μεθόδων.

#### 4.1.1 Εξισώσεις Ροών Φορτίου

Οι εξισώσεις ροών φορτίου μπορούν να καταστρωθούν με τη βοήθεια του Σχήματος 4.1 που απεικονίζει ένα γενικευμένο ζυγό ενός ΣΗΕ.



Σχήμα 4. 1 : Γενικευμένος ζυγός ενός ΣΗΕ [76].

Στο σχήμα αυτό φαίνεται η γενική περίπτωση ενός ζυγού  $k$  του συστήματος, με γεννήτρια και φορτίο που εκφράζονται ως ισχύεις (μιαδική παραγόμενη ισχύς  $S_{Gk}$  και μιαδική ισχύς φορτίου  $S_{Dk}$ ). Η γραμμή μεταφοράς μεταξύ των ζυγών  $k$  και  $m$  παριστάνεται με το ονομαστικό κύκλωμα  $\Pi$  (αγωγιμότητες  $y_{km}$  και  $y_{skm}$ ). Στον ζυγό  $k$  είναι επίσης συνδεδεμένη εγκάρσια σύνθετη αγωγιμότητα  $y_k$  που παριστάνει οποιοδήποτε συνδυασμό πυκνωτών, αυτεπαγωγών ή φορτίων που παριστάνονται με σταθερή αγωγιμότητα. Η τάση στον ζυγό  $k$  συμβολίζεται με  $V_k$  ενώ η τάση στον ζυγό  $m$  με  $V_m$ .

Σε κάθε ζυγό ενός ΣΗΕ, υπάρχουν δύο εξισώσεις ροών φορτίου: το ισοζύγιο ενεργού ισχύος και το ισοζύγιο αέργου ισχύος. Οι εξισώσεις ροών φορτίου μπορούν να γραφούν σε Καρτεσιανή, Πολική ή Υβριδική μορφή, ωστόσο στην παρούσα εργασία εξετάζεται η Υβριδική μορφή καθώς αυτή χρησιμοποιείται για την επίλυση των ασκήσεων του μαθήματος.

Το ισοζύγιο ενεργού ισχύος στο ζυγό  $k$  δίνεται από τη σχέση:

$$\begin{aligned} P_k &= P_{Gk} - P_{Dk} \\ &= V_k^2 G_{kk} + V_k \sum_{\substack{m \in A(k) \\ k \neq m}} V_m [G_{km} \cos(\delta_k - \delta_m) + B_{km} \sin(\delta_k - \delta_m)] \end{aligned} \quad (4.1)$$

και το ισοζύγιο αέργου ισχύος στο ζυγό  $k$  δίνεται από τη σχέση:

$$\begin{aligned} Q_k &= Q_{Gk} - Q_{Dk} \\ &= -V_k^2 B_{kk} + V_k \sum_{\substack{m \in A(k) \\ k \neq m}} V_m [G_{km} \sin(\delta_k - \delta_m) - B_{km} \cos(\delta_k - \delta_m)] \end{aligned} \quad (4.2)$$

Όπου:

$$\widetilde{V}_k = V_k e^{j\delta_k} = V_k \angle \delta_k \quad (4.3)$$

$$\widetilde{V}_m = V_m e^{j\delta_m} = V_m \angle \delta_m \quad (4.4)$$

$$S_{Gk} = P_{Gk} + jQ_{Gk} \quad (4.5)$$

$$S_{Dk} = P_{Dk} + jQ_{Dk} \quad (4.6)$$

$$Y_{kk} = G_{kk} + jB_{kk} \quad (4.7)$$

$$Y_{km} = G_{km} + jB_{km} \quad (4.8)$$

όπου  $Y_{kk}$  είναι τα στοιχεία της κύριας διαγωνίου και  $Y_{km}$  είναι τα στοιχεία της μη κύριας διαγωνίου του πίνακα αγωγιμοτήτων  $Y_Z$  του ΣΗΕ με:

$$Y_{kk} = y_k + \sum_{m \in A(k)} (y_{skm} + y_{km}) \quad (4.9)$$

$$Y_{km} = -y_{km} \quad (4.10)$$

$$Y_Z = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1k} & Y_{1m} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & \dots & Y_{2k} & Y_{2m} & \dots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ Y_{k1} & Y_{k2} & \dots & Y_{kk} & Y_{km} & \dots & Y_{kn} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \dots & Y_{nk} & Y_{nm} & \dots & Y_{nn} \end{bmatrix} \quad (4.11)$$

$$= \begin{bmatrix} G_{11} + jB_{11} & \dots & \dots & \dots & \dots & \dots & G_{1n} + jB_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \dots & G_{kk} + jB_{kk} & G_{km} + jB_{km} & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ G_{n1} + jB_{n1} & \dots & \dots & \dots & \dots & \dots & G_{nn} + jB_{nn} \end{bmatrix}$$

Το σύμβολο  $m \in A(k)$  σημαίνει το σύνολο των ζυγών που συνδέονται με το ζυγό  $k$  [76].

Για ένα σύστημα  $n$  ζυγών υπάρχουν  $n$  εξισώσεις της μορφής (4.1) και  $n$  εξισώσεις της μορφής (4.2). Οι  $2n$  εξισώσεις αυτές αποτελούν τις εξισώσεις ροής φορτίου σε υβριδική μορφή.

#### 4.1.2 Θεμελίωση του προβλήματος ροών φορτίου

Γενικά, σε κάθε ζυγό  $k$  του συστήματος αντιστοιχούν τέσσερις μεταβλητές. Αυτές είναι:

- |                              |                              |
|------------------------------|------------------------------|
| 1. Έγχυση πραγματικής ισχύος | $P_{k,sp} = P_{Gk} - P_{Dk}$ |
| 2. Έγχυση αέργου ισχύος      | $Q_{k,sp} = Q_{Gk} - Q_{Dk}$ |
| 3. Μέτρο τάσεως              | $V_k$                        |
| 4. Φασική γωνία τάσεως       | $\delta_k$                   |

Οι ζυγοί του συστήματος ταξινομούνται σε τρεις κατηγορίες:

1. Ζυγός ταλάντωσης ή ζυγός αναφοράς: Για τον ζυγό αυτό η πραγματική και άεργος εγγεόμενη ισχύς αφήνονται «ελεύθερες» ενώ το μέτρο τάσεως και η φασική γωνία θεωρούνται γνωστά. Συνήθως  $\delta_k = 0$ .
2. Ζυγός φορτίου ή ζυγός PQ: Οποιοσδήποτε ζυγός για τον οποίο η έγχυση πραγματικής ισχύος ( $P_{Dk}$ ) και αέργου ( $Q_{Dk}$ ) θεωρούνται γνωστές.
3. Ζυγός παραγωγής ή ζυγός PV: Οποιοσδήποτε ζυγός για τον οποίο η έγχυση πραγματικής ισχύος ( $P_{Gk}$ ) και το μέτρο τάσεως ( $V_k$ ) θεωρούνται γνωστές.

Για τη διευκόλυνση των συμβολισμών ακολουθούμε την παρακάτω αρίθμηση ζυγών. Ζυγός 1 είναι ο ζυγός ταλάντωσης,  $2, \dots, n-m$  είναι οι ζυγοί παραγωγής (ζυγοί PV) και  $n-m+1, \dots, n$  οι ζυγοί φορτίου (ζυγοί PQ), όπου  $n$  είναι ο αριθμός των ζυγών του συστήματος και  $m$  ο αριθμός των ζυγών φορτίου (PQ).

Για τη θεμελίωση του προβλήματος ροών φορτίου σε υβριδική μορφή, ο ελάχιστος αριθμός μεταβλητών που περιγράφουν την κατάσταση του συστήματος είναι:

1. Οι φασικές γωνίες τάσεων όλων των ζυγών εκτός από τον ζυγό ταλαντώσεως:  $\delta_2, \dots, \delta_n$
2. Τα μέτρα τάσεων όλων των ζυγών φορτίου:  $V_{n-m+1}, V_{n-m+2}, \dots, V_n$

Οι μεταβλητές αυτές ονομάζονται μεταβλητές κατάστασης και συνιστούν το διάνυσμα κατάστασης  $x$ .

$$x = \begin{bmatrix} \delta_i \\ V_i \end{bmatrix} = \begin{bmatrix} \delta_2 \\ \vdots \\ \delta_n \\ V_{n-m+1} \\ \vdots \\ V_n \end{bmatrix} \quad (4.12)$$

$$\text{Με } \delta = [\delta_2, \dots, \delta_n]^T, V = [V_{n-m+1}, \dots, V_n]^T$$

Το διάνυσμα κατάστασης είναι διάστασης  $n-1+m$  και για να υπολογιστεί απαιτούνται  $n-1+m$  ανεξάρτητες εξισώσεις. Οι εξισώσεις αυτές είναι:

1.  $n-1$  εξισώσεις πραγματικής ισχύος (4.1): μία για κάθε ζυγό εκτός από τον ζυγό ταλάντωσης.
2.  $m$  εξισώσεις αέργου ισχύος (4.2): μία για κάθε ζυγό φορτίου.

Οι ανωτέρω εξισώσεις συνιστούν τις εξισώσεις του προβλήματος ροών φορτίου σε υβριδική μορφή [75].

### 4.1.3 Τεχνικές επίλυσης προβλήματος ροών φορτίου

Το πρόβλημα ροών φορτίου συνίσταται στη λύση ενός συνόλου μη γραμμικών εξισώσεων. Οι αριθμητικές μέθοδοι που χρησιμοποιούνται για τους υπολογισμούς των ροών φορτίου μπορούν να λύσουν τις μη γραμμικές αλγεβρικές εξισώσεις των ροών φορτίου για μεγάλα συστήματα με εκατοντάδες ζυγούς, με απώλειες, με ικανοποιητική ακρίβεια. Μεταξύ άλλων, δύο βασικές μέθοδοι που εφαρμόζονται στις μελέτες ροών φορτίου είναι η μέθοδος Gauss – Seidel και η μέθοδος Newton – Raphson. Και οι δύο αυτές μέθοδοι ξεκινούν από μια αρχική λύση και με διαδοχικές ανακυκλώσεις προσεγγίζουν την τελική ως εξής: Η αρχική εκτίμηση χρησιμοποιείται στις εξισώσεις του προβλήματος για να βρεθεί μια καινούρια βελτιωμένη εκτίμηση, η οποία χρησιμοποιείται στη συνέχεια για να βρεθεί μια τρίτη ακόμη καλύτερη κ.ο.κ. Οι επαναλήψεις συνεχίζονται μέχρι να ικανοποιηθούν τα κριτήρια σύγκλισης [77].

$$|\Delta P_k^{(n)}| \leq \varepsilon, |\Delta Q_k^{(n)}| \leq \varepsilon \quad (4.13)$$

Στην παρούσα ενότητα παρουσιάζουμε τη μέθοδο Newton – Raphson καθώς αυτή χρησιμοποιήθηκε για τις ανάγκες των εφαρμογών που αναπτύχθηκαν.

Η εφαρμογή της μεθόδου Newton – Raphson συνίσταται στον υπολογισμό των άγνωστων μέτρων και των γωνιών των τάσεων του συστήματος. Αυτό πραγματοποιείται μέσω της επίλυσης της γενικής εξίσωσης Newton – Raphson:

$$[\Delta x] = -[J_{ac}]^{-1}[\Delta f] \Rightarrow \begin{bmatrix} \Delta \delta \\ \Delta V \end{bmatrix} = - \begin{bmatrix} H & N \\ J & L \end{bmatrix}^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \quad (4.14)$$

Στην εξίσωση αυτή το διάνυσμα  $[\Delta f]$  δημιουργείται από τη διαφορά του διανύσματος των προσδιορισμένων τιμών των ισχύων  $P_{k,sp}, Q_{k,sp}$  που δίνονται ως δεδομένα του προβλήματος ροής φορτίου, και του διανύσματος των κάθε φορά προσεγγίσεων  $P_k$  και  $Q_k$  που υπολογίζονται από τις εξισώσεις (4.1), (4.12).

$$[\Delta f] = \begin{bmatrix} P_{k,sp} - P_k \\ Q_{k,sp} - Q_k \end{bmatrix} \quad (4.15)$$

Η Ιακωβιανή μήτρα έχει τη μορφή [78]:

$$J_{ac} = \begin{bmatrix} H & N \\ J & L \end{bmatrix} = \begin{bmatrix} \frac{\partial \Delta P}{\partial \delta} & \frac{\partial \Delta P}{\partial V} \\ \frac{\partial \Delta Q}{\partial \delta} & \frac{\partial \Delta Q}{\partial V} \end{bmatrix} \quad (4.16)$$

Ο υποπίνακας  $H$  έχει διαστάσεις  $(n-1) \times (n-1)$ . Τα διαγώνια και μη διαγώνια στοιχεία του δίνονται από τις σχέσεις:

$$H_{kk} = \frac{\partial \Delta P_k}{\partial \delta_k} \quad (4.17)$$

$$\begin{aligned}
&= V_k \sum_{\substack{m \in A(k) \\ k \neq m}} V_m [G_{km} \sin(\delta_k - \delta_m) - B_{km} \cos(\delta_k - \delta_m)] \\
&= Q_k + V_k^2 B_{kk}
\end{aligned}$$

$$H_{km} = \frac{\partial \Delta P_k}{\partial \delta_m} = -V_k V_m [G_{km} \sin(\delta_k - \delta_m) - B_{km} \cos(\delta_k - \delta_m)] \quad (4.18)$$

Ο υποπίνακας N έχει διαστάσεις  $(n-1) \times n$ . Τα διαγώνια και μη διαγώνια στοιχεία του δίνονται από τις σχέσεις:

$$N_{kk} = \frac{\partial \Delta P_k}{\partial V_k} = -\frac{P_k}{V_k} - V_k G_{kk} \quad (4.19)$$

$$N_{km} = \frac{\partial \Delta P_k}{\partial V_m} = -V_k [G_{km} \cos(\delta_k - \delta_m) + B_{km} \sin(\delta_k - \delta_m)] \quad (4.20)$$

Ο υποπίνακας J έχει διαστάσεις  $m \times (n-1)$ . Τα διαγώνια και μη διαγώνια στοιχεία του δίνονται από τις σχέσεις:

$$J_{kk} = \frac{\partial \Delta Q_k}{\partial \delta_k} = -P_k + V_k^2 G_{kk} \quad (4.21)$$

$$J_{km} = \frac{\partial \Delta Q_k}{\partial \delta_m} = V_k V_m [G_{km} \cos(\delta_k - \delta_m) + B_{km} \sin(\delta_k - \delta_m)] \quad (4.22)$$

Ο υποπίνακας L έχει διαστάσεις  $m \times m$ . Τα διαγώνια και μη διαγώνια στοιχεία του δίνονται από τις σχέσεις:

$$L_{kk} = \frac{\partial \Delta Q_k}{\partial V_k} = -\frac{Q_k}{V_k} + V_k B_{kk} \quad (4.23)$$

$$L_{km} = \frac{\partial \Delta Q_k}{\partial V_m} = -V_k [G_{km} \sin(\delta_k - \delta_m) - B_{km} \cos(\delta_k - \delta_m)] \quad (4.24)$$

Οι νέες τιμές του διανύσματος κατάστασης υπολογίζονται χρησιμοποιώντας τις τιμές της προηγούμενης επανάληψης, δηλαδή του διανύσματος αρχικών τιμών (και των επόμενων προσεγγίσεων) :

$$[\delta]^{(n+1)} = [\Delta \delta] + [\delta]^{(n)} \quad (4.25)$$

$$[V]^{(n+1)} = [\Delta V] + [V]^{(n)} \quad (4.26)$$

Στη γραμμή μεταφοράς k-m του Σχήματος [4.1], η ροές ενεργού ισχύος  $P_{km}$  και  $P_{mk}$  υπολογίζονται από τις σχέσεις [76]:

$$P_{km} = (g_{km} + g_{skm}) V_k^2 - V_k V_m [g_{km} \cos(\delta_k - \delta_m) + b_{km} \sin(\delta_k - \delta_m)] \quad (4.27)$$

$$P_{mk} = (g_{km} + g_{skm})V_m^2 - V_m V_k [g_{km} \cos(\delta_m - \delta_k) + b_{km} \sin(\delta_k - \delta_m)] \quad (4.27)$$

Αντίστοιχα οι ροές αέργου ισχύος  $Q_{km}$  και  $Q_{mk}$  υπολογίζονται από τις σχέσεις:

$$Q_{km} = -(b_{km} + b_{skm})V_k^2 - V_k V_m [g_{km} \sin(\delta_k - \delta_m) - b_{km} \cos(\delta_k - \delta_m)] \quad (4.29)$$

$$Q_{mk} = -(b_{km} + b_{skm})V_m^2 - V_m V_k [g_{km} \sin(\delta_m - \delta_k) - b_{km} \cos(\delta_m - \delta_k)] \quad (4.30)$$

όπου:

$$y_{km} = g_{km} + jb_{km} \quad (4.31)$$

$$y_{km} = g_{km} + jb_{km} \quad (4.32)$$

Ο αλγόριθμος Newton – Raphson μπορεί να δοθεί συνοπτικά:

1. Δίνονται αρχικές τιμές στα  $V$  και  $\delta$ , όπου αυτά δεν είναι καθορισμένα (αρχικό διάλυμα κατάστασης).
2. Υπολογίζονται οι ισχύεις των ζυγών από τις εξισώσεις (4.1), (4.1). Είναι γνωστά τα  $P$  για όλους τους ζυγούς (πλην του ζυγού αναφοράς) και τα  $Q$  για τους ζυγούς φορτίου. Συνεπώς χρειάζεται ο υπολογισμός των  $P_2 \dots P_n$  και των  $Q_{m+1} \dots Q_n$  που θα δώσουν τις αποκλίσεις  $[\Delta P]$  και  $[\Delta Q]$  που χρειάζεται η εξίσωση.
3. Υπολογίζεται η Ιακωβιανή μήτρα.
4. Επιλύεται το σύστημα και υπολογίζονται οι αποκλίσεις  $[\Delta V]$ ,  $[\Delta \delta]$ , δηλαδή οι διορθώσεις  $[\Delta x]$ .
5. Αν οι αποκλίσεις είναι μεγαλύτερες από τα όρια ανοχής των σχέσεων (4.13), επανερχόμαστε στο βήμα (2).
6. Μετά τη σύγκλιση υπολογίζονται τα  $P_1$ ,  $Q_1$  του ζυγού αναφοράς και κατόπιν οι ροές φορτίου στις γραμμές [77].



## 4.2 Εφαρμογή στη ροή φορτίου

Η ανάλυση των ροών φορτίου είναι το σημείο εκκίνησης για όλες τις επιμέρους μελέτες των ΣΗΕ και ως εκ τούτου το πρόγραμμα υπολογισμού της ροής ισχύος είναι το πιο συνηθισμένο καθημερινό εργαλείο των αναλυτών των ΣΗΕ [3]. Για το λόγο αυτό αποτελεί αναπόσπαστο και βασικό κομμάτι της κατάρτισης των φοιτητών Ηλεκτρολόγων Μηχανικών. Στο μάθημα Ανάλυση ΣΗΕ (Μόνιμη Κατάσταση Λειτουργίας) του τομέα Ηλεκτρικής Ισχύος της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών (ΣΗΜΜΥ) του ΕΜΠ, πέρα από την ανάλυση του θεωρητικού υποβάθρου του αντικειμένου αυτού, την επεξήγηση της μεθοδολογίας και την επίλυση ασκήσεων στο αμφιθέατρο, πραγματοποιούνται δύο εργαστηριακές ασκήσεις που αποσκοπούν στην εξοικείωση των φοιτητών με την μελέτη και επίλυση προβλημάτων που σχετίζονται με τη ροή φορτίου. Συγκεκριμένα περιλαμβάνουν θέματα αντιστάθμισης αέργου ισχύος, αυξομειώσεων παραγωγής – ζήτησης και των ορίων μεταβολής τους, αντικατάστασης γραμμών μεταφοράς ή ενίσχυσής τους, μεταβολών των ορίων των τάσεων στους ζυγούς εκτός επιτρεπτών ορίων, μεταβολών των ρευμάτων των γραμμών, επίδρασης της μεταβολής ενδιάμεσων λήψεων μετασχηματιστών ισχύος, μελέτης απωλειών, επίδρασης από τη διείσδυση μονάδων ΑΠΕ και μονάδων ΔΠ, ποσοτικοποίησης οικονομικών πλεονεκτημάτων κ.ά.

Οι δύο εργαστηριακές ασκήσεις οι οποίες έχουν αναπτυχθεί με χρήση του λογισμικού MATLAB, στοχεύουν:

- Στην κατανόηση από τους φοιτητές των φαινομένων που αναφέρθηκαν παραπάνω.
- Στην εισαγωγή τους στο προγραμματιστικό περιβάλλον MATLAB.
- Στην απόκτηση εμπειρίας στην ανάλυση ΣΗΕ χρησιμοποιώντας υπολογιστικά εργαλεία που εφαρμόζουν την γλώσσα προγραμματισμού MATLAB για τον υπολογισμό της ροής φορτίου.

Λαμβάνοντας υπόψη τους παραπάνω στόχους επιδιώξαμε να υλοποιήσουμε και να παρουσιάσουμε τον κώδικα της ροής φορτίου χρησιμοποιώντας το περιβάλλον του Jupyter Notebook, με σκοπό την πιθανή αξιοποίησή του για τις ανάγκες των διαλέξεων ή/και του εργαστηρίου του μαθήματος Ανάλυσης ΣΗΕ. Σκοπός αυτής της προσπάθειας δεν είναι η απαξίωση του ρόλου του εργαλείου MATLAB για την εκπαίδευση των μηχανικών, αλλά η διερεύνηση μιας εκπαιδευτικής προσέγγισης που τονίζει τις παρακάτω πτυχές:

1. Μειώνει το χάσμα μεταξύ θεωρίας και εφαρμογής παραθέτοντας τις μαθηματικές εξισώσεις παράλληλα με τον κώδικα και τα αποτελέσματά του ώστε να γίνονται ξεκάθαρα τα βήματα της μεθοδολογίας και συνεπώς να αυξηθεί η κατανόηση των φοιτητών σχετικά με τα θέματα της ροής φορτίου.
2. Δίνεται παραπάνω έμφαση στον προγραμματιστικό κώδικα, με τα σχόλια που τονίζουν και εξηγούν ορισμένα σημεία, ώστε οι φοιτητές να έχουν την ευκαιρία να τον μελετήσουν, να αλληλεπιδράσουν και να εξοικειωθούν με αυτόν.

Σε αυτό το πλαίσιο δημιουργήσαμε τρεις εφαρμογές στο περιβάλλον του Jupyter Notebook:

1. Μια πρώτη εφαρμογή σε γλώσσα MATLAB, για την επίλυση ενός παραδείγματος ροής φορτίου από τις ασκήσεις που παρουσιάζονται στο μάθημα, στο οποίο δίνεται έμφαση στη θεωρία και σε όλα τα απαιτούμενα βήματα της μεθοδολογίας.
2. Μια εφαρμογή σε γλώσσα MATLAB που αφορά την 1<sup>η</sup> εργαστηριακή άσκηση και περιλαμβάνει τη μεθοδολογία των κώδικα τα αριθμητικά αποτελέσματα της προσομοίωσης καθώς και γραφική αναπαράσταση των αποτελεσμάτων.
3. Την αντίστοιχη εφαρμογή της 1<sup>ης</sup> εργαστηριακής άσκησης σε γλώσσα Python.

Οι εφαρμογές αυτές αναπτύχθηκαν στην τοπική εγκατάσταση του Jupyter Notebook καθώς και στις cloud εναλλακτικές του Jupyter Notebook, CoCalc και Colab, οι οποίες είναι πιο εύκολα προσβάσιμες.

#### 4.2.1 Υπάρχον εκπαιδευτικό υλικό στο πρόγραμμα MATLAB

Παρακάτω ακολουθεί μια σύντομη περιγραφή του εκπαιδευτικού υλικού του εργαστηρίου Ανάλυσης ΣΗΕ ώστε να δοθεί μια εικόνα της διάρθρωσης και του περιεχομένου του, μέρος του οποίου χρησιμοποιείται και στις εφαρμογές που υλοποιήσαμε. Επίσης παρατίθεται ένα στιγμιότυπο οθόνης του εκτελέσιμου αρχείου της ροής φορτίου και των αριθμητικών αποτελεσμάτων του, τα οποία χρησιμοποιήθηκαν για τον έλεγχο της ορθότητας των αποτελεσμάτων της εφαρμογής μας σε Jupyter Notebook.

Για κάθε εργαστηριακή άσκηση παρέχεται ένα αρχείο εισαγωγής δεδομένων (data4buses.m και dataLFA.m αντίστοιχα), στα οποία επεμβαίνουν οι φοιτητές. Εκτελώντας τα αρχεία αυτά αυτομάτως λύνεται το πρόβλημα της ροής φορτίου και εμφανίζονται όλα τα αποτελέσματα στο παράθυρο του προγράμματος MATLAB. Στο Σχήμα 4.2 φαίνεται η δομή του αρχείου data4buses.m, ωστόσο η δομή του αρχείου dataLFA.m είναι όμοια.

Το αρχείο περιλαμβάνει δύο πίνακες για την εισαγωγή των δεδομένων του συστήματος.

Ο πρώτος πίνακας φέρει την ονομασία busdata και περιλαμβάνει έντεκα στήλες που αφορούν τα δεδομένα των ζυγών: 1) Αρίθμηση ζυγών, 2) χαρακτηρισμός ζυγών (0 για τους PQ, 2 για τους PV, 1 για τον ζυγό αναφοράς), 3) τιμές των τάσεων (α.μ.) κάθε ζυγού (εάν δεν δίνονται χρησιμοποιείται η τιμή  $V=1$ ), 4) τιμές των γωνιών (σε μοίρες-degree) κάθε ζυγού (εάν δε δίνονται χρησιμοποιείται η τιμή  $\delta=0$ ), 5) ζήτηση ενεργού ισχύος για ένα ζυγό (MW), 6) ζήτηση αέργου ισχύος (MVar), 7) παραγόμενη ενεργός ισχύς στον ζυγό που εξετάζεται (MW), 8) παραγόμενη αέργο ισχύς (MVar), 9) ελάχιστα όρια αέργου ισχύος (Mvar) για τις γεννήτριες, 10) μέγιστα όρια αέργου

ισχύος αντίστοιχα (MVar), 11) δεδομένα αντιστάθμισης αέργου ισχύος με πυκνωτές, δηλώνοντας την έγχυση αέργου ισχύος στον αντίστοιχο ζυγό (MVar).

Ο δεύτερος πίνακας έξι στηλών, φέρει την ονομασία linedata και περιέχει δεδομένα των γραμμών μεταφοράς και των μετασχηματιστών:

1) και 2) συνδέσεις που υπάρχουν μεταξύ των ζυγών, 3) αντίσταση (α.μ.) κάθε γραμμής, 4) αντίδραση (α.μ.) κάθε γραμμής, 5) το ήμισυ της συνολικής εγκάρσιας χωρητικότητας της γραμμής (α.μ.), 6) για γραμμή με μετασχηματιστές ισχύος που έχουν μεταβολή της σχέσης μετασχηματισμού (συνήθως θεωρούνται ονομαστικές λήψεις με  $t=1$ ).

Στη συνέχεια ακολουθούν τέσσερα αρχεία:

1. Το αρχείο Lfbust.m για τον σχηματισμό του πίνακα αγωγιμοτήτων.
2. Το αρχείο Lfnewton για την επίλυση της ροής φορτίου με τη μέθοδο Newton – Raphson. Εναλλακτικά υπάρχουν και δύο άλλα αρχεία Lfgausst.m και Decouplet.m , για την επίλυση της ροής φορτίου βάσει της μεθόδου Gauss – Seidel και Ταχείας Αποζευγμένης μεθόδου αντίστοιχα.
3. Το αρχείο Busoutt.m που υπολογίζει και εμφανίζει τα τελικά αποτελέσματα της ροής φορτίου.
4. Το αρχείο Lineflowt.m που υπολογίζει και εμφανίζει τις ροές ισχύος στις γραμμές καθώς και τις αντίστοιχες απώλειες ισχύος.

```

1 clear
2
3 baseMVA = 100; accuracy = 0.001; maxIter = 100;
4
5
6 %      Bus Bus Voltage Angle ---Load---- Generator---- Injected
7 %      No. code V(M) Degree MW Mvar MW Mvar Q(m) Qmax Mvar
8 busdata=[1 1 1.00 0.0 0 0 0 0 0 0 0
9          2 2 1.05 0.0 0 0 30 0 0 500 0
10         3 0 1.00 0.0 30 5 0 0 0 500 0
11         4 0 1.00 0.0 40 10 0 0 0 0 0];
12
13
14 %      Line code
15 %      Bus bus R X 1/2 B = 1 for lines
16 %      nl nr p.u. p.u. p.u. > 1 or < 1 tr. tap at bus nl
17 linedata=[1 2 0.0 0.1 0.0 1
18           1 4 0.0 0.2 0.0 1
19           2 3 0.0 0.2 0.0 1
20           3 4 0.0 0.1 0.0 1];
21
22
23
24 lfbust % form the bus admittance matrix
25
26 lfnewton % Power flow solution by Newton-Raphson method
27
28 lfgauss % Power flow solution by Gauss-Seidel method
29
30 %decouple % Power flow solution by fast decoupled method
31
32 Runout % Prints the power flow solution on the screen
33
34 Lineflow % Computes and displays the line flow and losses
35

```

```

>> data4Buses

Power flow Solution by Newton-Raphson Method
Maximum Power Mismatch = 4.02589e-05
No. of Iterations = 3

Bus Voltage Angle ---Load---- Generator--- Injected
No. Mag. Degree MW Mvar MW Mvar Q(m) Qmax Mvar
1 1.000 0.00 0.00 0.00 40.00 -49.53 0.00
2 1.050 -0.22 0.00 0.00 30.00 72.62 0.00
3 1.014 -3.89 30.00 5.00 0.00 0.00 0.00
4 1.002 -4.12 40.00 10.00 0.00 0.00 0.00
Total 70.00 15.00 70.00 23.09 0.00

Line Flow and Losses
--Line-- Power at bus & line flow --Line loss-- Transformer
from to MW Mvar MVA MW Mvar tap
1 2 40.000 -49.529 63.654 0.000 2.516
  4 35.954 0.467 35.957 0.000 2.506
2 3 30.000 72.622 36.574 0.000 2.516
  4 34.045 20.118 35.546 0.000 2.837
3 2 -30.000 -5.000 36.414 0.000 2.837
  4 4.016 12.281 12.930 0.000 0.163
4 1 -40.000 10.000 41.231 0.000 2.506
  3 -4.045 -12.118 12.776 0.000 0.163
Total loss 0.000 8.101

```

Σχήμα 4. 2 : Πρόγραμμα για την επίλυση της ροής φορτίου με τη μέθοδο Newton – Raphson στο MATLAB και τα αποτελέσματα μετά την εκτέλεσή του.

## 4.2.2 Παράδειγμα εφαρμογής του προβλήματος ροής φορτίου σε γλώσσα MATLAB στο Jupyter Notebook

Το Jupyter Notebook υποστηρίζει τον πυρήνα του MATLAB, ωστόσο για να λειτουργήσει αυτός ο συνδυασμός απαιτείται να προϋπάρχει εγκατεστημένο στον υπολογιστή το πρόγραμμα MATLAB. Οι οδηγίες που ακολουθήσαμε έχουν δοκιμαστεί με επιτυχία για την έκδοση MATLAB R2017a σε περιβάλλον Linux, Mac και Windows. Επιπλέον, απαιτείται εγκατάσταση του λογισμικού του Jupyter Notebook και της Python, αν δεν είναι ήδη εγκατεστημένα, καθώς και η εκτέλεση ορισμένων εντολών για τη διαμόρφωση της Python και του πυρήνα της Matlab ώστε να μπορούν να αλληλεπιδράσουν μεταξύ τους. Οι οδηγίες παρέχονται αναλυτικά online από το μάθημα Εφαρμοσμένα Μαθηματικά, Εισαγωγή στον Επιστημονικό Υπολογισμό του Πανεπιστημίου του Harvard [79], καθώς και στο επίσημο site του MATLAB ([www.mathworks.com](http://www.mathworks.com)). Όταν ολοκληρωθεί αυτή η διαδικασία, ο κώδικας MATLAB μπορεί να τρέξει στο περιβάλλον του Jupyter Notebook.

Από τα παραπάνω αρχεία του εργαστηρίου, τα δεδομένα εισόδου busdata και linedata, ο κώδικας για τον υπολογισμό της μήτρας αγωγιμοτήτων του συστήματος στο αρχείο Lfbybust.m, καθώς και ο κώδικας για την εμφάνιση των αποτελεσμάτων της ροής φορτίου στα αρχεία Busoutt.m και Lineflowt.m χρησιμοποιήθηκαν σχεδόν αυτούσια. Ωστόσο, ο κώδικας για τον υπολογισμό της ροής φορτίου που περιέχεται στο αρχείο Lfnewtont.m δεν μπορούσε να αξιοποιηθεί για δύο λόγους:

1. Στον κώδικα αυτό, η επίλυση της ροής φορτίου προσεγγίζεται με τους συμβολισμούς και τις εξισώσεις της Πολικής μορφής, η οποία αν και δίνει τα σωστά αποτελέσματα, δε συνάδει με το τυπολόγιο που παρέχεται στους φοιτητές στο μάθημα και το οποίο καλούνται να ακολουθήσουν κατά την επίλυση των ασκήσεων.
2. Η δομή του συγκεκριμένου κώδικα είναι πολύπλοκη, γεγονός που καθιστά επίπονη τη διαμόρφωσή του στα κελιά κώδικα του Jupyter Notebook με λειτουργικό και κατανοητό τρόπο. Ειδικά για τον εκπαιδευτικό σκοπό της υλοποίησης αυτών των ασκήσεων χρειαζόμαστε μικρά τμήματα κώδικα, που έχουν τη δυνατότητα να εκτελεστούν μεμονωμένα ώστε να γίνεται εύκολα η αντιστοίχιση των εξισώσεων με τον κώδικα που περιέχεται στο κελί ακριβώς από κάτω, και με τα αποτελέσματά του.

Για τους λόγους αυτούς ήταν σημαντικό να εστιάσουμε στην προσαρμογή του κώδικα ώστε να ικανοποιεί τις ανάγκες του μαθήματος. Καθώς η ανάλυση ροής φορτίου είναι ένα ιδιαίτερα συνηθισμένο αντικείμενο και η γλώσσα MATLAB ευρέως διαδεδομένη στους κλάδους των μηχανικών, επιχειρήσαμε να χρησιμοποιήσουμε μια γνωστή υλοποίηση του κώδικα της ροής φορτίου που κυκλοφορεί «ελεύθερα» στο διαδίκτυο (συγκεκριμένα ανακτήθηκε από το αποθετήριο GitHub [80]) και διαθέτει τα επιθυμητά γνωρίσματα που αναφέρθηκαν. Ο κώδικας αυτός, αφού διαμορφώθηκε κατάλληλα ώστε να ταιριάζει με τα δεδομένα και με τα υπόλοιπα τμήματα του κώδικα που είχαμε στη διάθεσή μας, δοκιμάστηκε και έγιναν οι απαραίτητες διορθώσεις ώστε να δίνει τα σωστά αποτελέσματα, δηλαδή αποτελέσματα συμβατά με αυτά του ήδη υπάρχοντος κώδικα του εργαστηρίου.

Στο σημείο αυτό θα παρουσιάσουμε ένα πρώτο παράδειγμα που αντικατοπτρίζει τη δομή και τη λειτουργία ενός Jupyter Notebook το οποίο μπορεί να χρησιμοποιηθεί ως ένα «ζωντανό έγγραφο» για την επεξήγηση και την επίλυση μιας άσκησης ροής φορτίου. Η συγκεκριμένη άσκηση έχει λυθεί κατά τη διάρκεια του μαθήματος Ανάλυση ΣΗΕ, στο αμφιθέατρο, το ακαδημαϊκό έτος 2016-2017.

Όπως παρατηρούμε στο Σχήμα 4.3, στην αρχή του εγγράφου αναγράφεται ο τίτλος και στη συνέχεια δίνονται τα στοιχεία της άσκησης. Συγκεκριμένα η τοπολογία του συστήματος τριών ζυγών και τα αντίστοιχα δεδομένα, δηλαδή η ονομαστική τάση  $V_{base}$ , η κοινή βάση ισχύος ( $baseMVA$ ), οι επαγωγικές αντιδράσεις και των γραμμών μεταφοράς ( $Z_{12(\alpha)}$ ,  $Z_{21(\beta)}$ ,  $Z_{13}$ ,  $Z_{23}$ ), τα δεδομένα που είναι διαθέσιμα για κάθε ζυγό δηλαδή τάσεις ( $V_1$ ,  $V_2$ ), παραγόμενη ενεργός ισχύς ( $P_{G2}$ ) και ισχείς φορτίου ( $P_{D2}$ ,  $Q_{D2}$ ,  $P_{D3}$ ,  $Q_{D3}$ ).

#### Βήμα 1ο: Ανά μονάδα τιμές, ισοζύγιο ισχύος

1. Γίνεται υπολογισμός των ανά μονάδα τιμών των ισοδύναμων αντιστάσεων των γραμμών μεταφοράς του συστήματος, μέσω των τύπων (1), (2) και (3), οι οποίοι χρησιμοποιούνται στο 1<sup>ο</sup> κελί κώδικα και τα αποτελέσματα εμφανίζονται ακριβώς από κάτω (Σχήμα 4.3).
2. Στο 2<sup>ο</sup> και 3<sup>ο</sup> κελί κώδικα παρέχονται ο πίνακας `linedata` για τα δεδομένα των γραμμών μεταφοράς που υπολογίσαμε και ο πίνακας `busdata` για τα δεδομένα των ζυγών αντίστοιχα (Σχήμα 4.3).
3. Γίνεται το ισοζύγιο ισχύος για κάθε ζυγό ξεχωριστά και υπολογίζονται οι ανά μονάδα τιμές της έγχυσης ενεργού ( $P_{sp}$ ) και αέργου ισχύος ( $Q_{sp}$ ) από τις σχέσεις (4) και (5), οι οποίες εκφράζονται ως μιγαδικές ισχύς ( $S$ ) από τη σχέση (6). Στο 4<sup>ο</sup> κελί κώδικα πραγματοποιούνται αυτές οι πράξεις: Αρχικά ορίζονται οι απαραίτητες μεταβλητές ως στήλες μηδενικών με μέγεθος όσο ο αριθμός των ζυγών του προβλήματος ( $n_{bus}=3$ ). Μέσα σε ένα `for loop` που επαναλαμβάνεται για κάθε ζυγό οι μεταβλητές αυτές αντιστοιχίζονται στα δεδομένα των ζυγών του πίνακα `busdata` και εκτελούνται οι εξισώσεις (4), (5) και (6), τα αποτελέσματα των οποίων εμφανίζονται από κάτω (Σχήμα 4.4).
4. Με τη σχέση (7), οι γωνίες ( $\delta$ ) μετατρέπονται από μοίρες σε ακτίνια όπως φαίνεται στο 5<sup>ο</sup> κελί κώδικα (Σχήμα 4.4).

**Ανάλυση Συστημάτων Ηλεκτρικής Ενέργειας - Μόνιμη Κατάσταση Λειτουργίας**

**ΡΟΗ ΦΟΡΤΙΟΥ**

Σχήμα 1: Σύστημα τριών ζυγών

Για το σύστημα ηλεκτρικής ενέργειας με την τοπολογία του παραπάνω σχήματος δίνονται τα ακόλουθα δεδομένα:

- Ονομαστική τάση 150kV
- Κινητή βάση ισχύος 100MVA
- Επαγωγικές αντιδράσεις και αγωγιμότητες των γραμμών μεταφοράς:
  - $Z_{12} = Z_{21} = j0.1 \text{ α.μ.}$
  - $Z_{13} = j15 \Omega$
  - $Z_{23} = j0.25 \text{ α.μ.}$  (βάση ισχύος 250 MVA)
- Δεδομένα ζυγών:  $P_1 = P_2 = 1 \text{ α.μ.}$ ,  $P_{D2} = 200 \text{ MW}$ ,  $P_{D3} = 50 \text{ MW}$ ,  $Q_{D2} = 50 \text{ MVar}$ ,  $P_{D3} = 300 \text{ MW}$ ,  $Q_{D3} = 100 \text{ MVar}$

**Βήμα 1α: Εύρεση των ανά μονάδα τιμών του συστήματος**

**Ισοδύναμες αντιστάσεις γραμμών**

Υπολογίζουμε τη βασική τιμή της αντίστασης:

$$Z_0 = \frac{V_B^2}{S_B} \quad (1)$$

και στη συνέχεια, αν δεν δίνονται ήδη, υπολογίζουμε τις ανά μονάδα τιμές των αντιδράσεων:

$$Z_{(a,i)} = \frac{Z(\Omega)}{Z_B} \quad (2)$$

Προσοχή: αν η τιμή της αντίστασης είναι σε άλλη βασική τιμή ισχύος από αυτή που έχει ορισθεί πρέπει να κάνουμε αναγωγή στη νέα, με τη σχέση:

$$Z_{N_{\text{new}}} = Z_{N_{\text{old}}} \cdot \left( \frac{V_{B, \text{old}}}{V_{B, \text{new}}} \right)^2 \cdot \frac{S_{B, \text{old}}}{S_{B, \text{new}}} \quad (3)$$

Στο παραδύναμα έχουμε:

```
In [1]:
Vbase = 150;
baseMVA = 100;
Sbold = 250;
j = sqrt(-1);
Z13 = j*15;
Z23 = j*0.25;

Zbase = Vbase^2/baseMVA % (1)
Z13 = Z13/Zbase % (2)
Z23 = Z23*(baseMVA/Sbold) % (3)

Zbase =
    225
Z13 =
    0.0000 + 0.06671i
Z23 =
    0.0000 + 0.10000i
```

**Δεδομένα εισόδου**

Ανδράγματα των γραμμών μεταφοράς και μεταποιημάτων:

```
In [2]:
% Line code
% Bus bus R X I/2 B = 1 for Lines
% nl nr p.u. p.u. p.u. > 1 or < 1 tr. tap at bus n1
linedata=[1 2 0.0 0.05 0.0 1
           1 3 0.0 0.0667 0.0 1
           2 3 0.0 0.1 0.0 1];
```

Δεδομένα των ζυγών:

```
In [3]:
% Bus Bus Voltage Angle Load Generator Injected
% No code: Mvar Dcprcc MW Mvar MW Mvar
busdata=[1 1 1.00 0.0 0 0 0 0 0
          2 2 1.00 0.0 50 50 200 0 0
          3 0 1.00 0.0 300 100 0 0 0];
```

Σχήμα 4. 3 : Στιγμιότυπο από το παράδειγμα επίλυσης ροής φορτίου σε Jupyter Notebook με τίτλο, μονογραμμικό διάγραμμα, κείμενο, εξισώσεις και κώδικα για τον υπολογισμό των α.μ. τιμών και την εισαγωγή των δεδομένων των γραμμών μεταφοράς και των ζυγών (1/9).

## Βήμα 2ο: Υπολογισμός του πίνακα αγωγιμοτήτων του συστήματος

1. Παρατίθενται οι σχέσεις (8) και (9) για τον υπολογισμό των διαγώνιων ( $Y_{kk}$ ) και μη διαγώνιων ( $Y_{km}$ ) στοιχείων του πίνακα αγωγιμοτήτων ( $Y_{bus}$ ) θεωρώντας τη γενική περίπτωση ενός ΣΗΕ. Στο συγκεκριμένο παράδειγμα ο πίνακας αγωγιμοτήτων έχει μέγεθος  $3 \times 3$  και για να υπολογιστούν τα στοιχεία του πρέπει πρώτα όλες οι σύνθετες αντιστάσεις ( $Z_{km}$ ) να μετατραπούν σε αγωγιμότητες ( $y_{km}$ ) από τις σχέσεις (10) και (11) (Σχήμα 4.4). Η διαδικασία αυτή φαίνεται στο 6<sup>ο</sup> κελί κώδικα, όπου αρχικά στις μεταβλητές των γραμμών μεταφοράς αναθέτονται οι τιμές του πίνακα `linedata`. Στη συνέχεια εκτελούνται οι σχέσεις (10) και (11). Έπειτα ένα διπλό `for loop` που επαναλαμβάνεται για όλους τους συνδυασμούς γραμμών υπολογίζει τα μη διαγώνια στοιχεία του πίνακα αγωγιμοτήτων από τη σχέση (9) και ένα διπλό `for loop` που επαναλαμβάνεται για όλους τους ζυγούς υπολογίζει τα διαγώνια στοιχεία από τη σχέση (8). Έτσι προκύπτει ο πίνακας αγωγιμοτήτων όπως φαίνεται και στα αποτελέσματα του κώδικα (Σχήμα 4.5).
2. Σε καρτεσιανή μορφή λαμβάνουμε το πραγματικό ( $G_{km}$ ) και το φανταστικό μέρος ( $B_{km}$ ) του πίνακα αγωγιμοτήτων από την εξίσωση (12) (Σχήμα 4.5).



**Έγχιση πραγματικής και αέργου ισχύος**

Γίνεται το καθήκον ισχύος για κάθε ζυγό  $k$ . Εξαρτάται και υπολογίζονται οι ανά μονάδα πρές της έγχυσης ενεργού και αέργου ισχύος από τις σχέσεις:

$$P_{p,k} = \frac{P_{G,k} - P_{D,k}}{S_{base}} \quad (4)$$

$$Q_{p,k} = \frac{Q_{G,k} - Q_{D,k}}{S_{base}} \quad (5)$$

Οι μιγαδικές ισχύεις εκφράζονται ως:

$$S = P_{p,k} + jQ_{p,k} \quad (6)$$

```
In [4]: Vn=0; delta=0; yload=0; deltad=0;
nbus = length(busdata(:,1));
kb=1:nbus-1; delta=1; V=1;
Pg = zeros(nbus,1); Pq = zeros(nbus,1); Qg = zeros(nbus,1); Qd = zeros(nbus,1);
Psp = zeros(nbus,1); Qsp = zeros(nbus,1); Qsh = zeros(nbus,1);
S = zeros(nbus,1);

for k=1:nbus
n=busdata(k,1); kb(n)=busdata(k,2); Vn(n)=busdata(k,3); delta(n)=busdata(k,4);
Pn(n)=busdata(k,5); Qn(n)=busdata(k,6); Pq(n)=busdata(k,7); Qq(n) = busdata(k,8);
Psp(n)=(Qg(n)-Qd(n))/baseva; % (4)
Qsp(n)=(Qg(n)-Qd(n))/baseva; % (5)
S(n) = Psp(n) + j*Qsp(n); % (6)
end

Psp
Qsp
S

Psp =
     0
  1.5000
 -3.0000
     0
 -0.5000
 -1.0000
S =
 0.0000 + 0.0000i
 1.5000 - 0.5000i
-3.0000 - 1.0000i
```

Μετατροπή γωνιών από μοίρες σε ακτίνια:

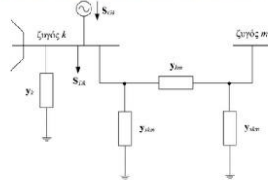
$$(\text{rad}) = \frac{\pi}{180} \cdot \delta^\circ \quad (7)$$

```
In [5]: for n=1:nbus
delta(n) = pi/180*delta(n); % (7)
end
delta

delta =
     0     0     0
```

**Βήμα 2ο: Υπολογισμός του πίνακα αγωγιμοτήτων του συστήματος**

Στο παρακάτω σχήμα φαίνεται η γενική περίπτωση ενός ζυγού  $k$  του συστήματος, με γεννήτρια και φορτίο που εκφράζονται ως ισχύεις (μιγαδική παραγωγή ισχύος  $S_{Gk}$  και μιγαδική ισχύος φορτίου  $S_{Dk}$ ). Η γραμμική μεταφοράς μεταξύ των ζυγών  $k$  και  $m$  παριστάνεται με το ονομαστικό κύκλωμα  $\Pi$  (αγωγιμότητες  $Y_{km}$  και  $Y_{mk}$ ). Στον ζυγό  $k$  είναι επίσης συνδεδεμένη ενεργός συνθήκη αγωγιμότητα  $Y_k$  που παριστάνει οποιαδήποτε συνδυασμό πυκνωτών, αυτεπαγωγών ή φορτίων που παριστάνονται με σταθερή αγωγιμότητα.



**Σχήμα 2: Γενικευμένος ζυγός συστήματος ηλεκτρικής ενέργειας**

Τα διαγώνια στοιχεία  $Y_{kk}$  και τα μη διαγώνια στοιχεία  $Y_{km}$  του πίνακα αγωγιμοτήτων του ορίζονται ε.π.ε. από τις παρακάτω σχέσεις:

$$Y_{kk} = Y_k + \sum_{m \neq k} (Y_{km} + Y_{km}) \quad (8)$$

$$Y_{km} = -Y_{km} \quad (9)$$

Ο πίνακας αγωγιμοτήτων έχει μέγεθος  $(n \times n)$  όπου  $n$  ο αριθμός των ζυγών του συστήματος που εξετάζουμε.

Στην περίπτωση μας ο πίνακας αγωγιμοτήτων έχει μέγεθος  $(3 \times 3)$ .

$$Y_{bus} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & Y_{23} \\ Y_{31} & Y_{32} & Y_{33} \end{bmatrix}$$

- Αρχικά όλες οι σύνθετες αντιστάσεις μετατρέπονται σε σύνθετες αγωγιμότητες από τη σχέση (10):

$$Z_{km} = R_{km} + jX_{km} \quad (10)$$

$$Y_{km} = \frac{1}{Z_{km}} \quad (11)$$

- και στη συνέχεια υπολογίζονται τα διαγώνια και τα μη διαγώνια στοιχεία του πίνακα αγωγιμοτήτων από τις σχέσεις (8) και (9) αντίστοιχα.

$$Y_{11} = Y_{12} + Y_{13}, \quad Y_{22} = Y_{21} - Y_{21}, \quad Y_{33} = Y_{31} + Y_{32}$$

$$Y_{12} = Y_{21} = -Y_{12}, \quad Y_{13} = Y_{31} = -Y_{13}, \quad Y_{23} = Y_{32} = -Y_{23}$$

Σχήμα 4. 4 : Στιγμιότυπο από το παράδειγμα επίλυσης ροής φορτίου σε Jupyter Notebook με εξισώσεις και κώδικα για των υπολογισμό των εγχύσεων ενεργού και αέργου ισχύος σε κάθε ζυγό καθώς και τις εξισώσεις για τον υπολογισμό του πίνακα αγωγιμοτήτων (2/9).

```
In [6]: n1 = linedata(:,1); %line from bus
nr = linedata(:,2); %line to bus
R = linedata(:,3); %line resistance
X = linedata(:,4); %line reactance
Bc = j*linedata(:,5);
g = linedata(:,6);
nbr=length(linedata(:,1)); %number of branch
G=[];B=[];

%the impedance of line
Z = R + j*X % (10)
%the admittance of line
y = ones(nbr,1)/Z % (11)

for n = 1:nbr
    Ybus=zeros(nbus,nbus); % initialize Ybus to zeros
    % formation of the off diagonal elements
    for kb=1:nbr
        Ybus(n1(k),nr(k))-Ybus(n1(k),nr(k))-y(k); % (9)
        Ybus(nr(k),n1(k))-Ybus(n1(k),nr(k));
    end
end
% formation of the diagonal elements
for nm=1:nbus
    for k=1:nbr
        if n1(k)==nm
            Ybus(n,n) = Ybus(n,n)+y(k); % (8)
        elseif nr(k)==nm
            Ybus(n,n) = Ybus(n,n)+y(k); % (8)
        else, end
    end
end
Ybus

Z =
    0.0000 + 0.0560i
    0.0000 + 0.0667i
    0.0000 + 0.1060i
y =
    0.0000 -20.0000i
    0.0000 -14.9925i
    0.0000 -10.0000i
Ybus =
    0.0000 -34.9925i  0.0000 +20.0000i  0.0000 -14.9925i
    0.0000 +20.0000i  0.0000 -30.0000i  0.0000 -10.0000i
    0.0000 -14.9925i  0.0000 +10.0000i  0.0000 -24.9925i
```

Ο πίνακας αγωγιμότητας σε καρτεσιανή μορφή γράφεται:

$$Y_{km} = G_{km} + jB_{km} \quad (12)$$

```
In [7]: G = real(Ybus) % Conductance matrix...
B = imag(Ybus) % Susceptance matrix...

G =
    0 0 0
    0 0 0
    0 0 0
B =
   -34.9925  20.0000  14.9925
    20.0000  -30.0000  10.0000
    14.9925  10.0000  -24.9925
```

### Θήμα 3ο: Το πρόβλημα ροών φορτίου

Το πρόβλημα των ροών φορτίου (ή ισχύος) συνίσταται στον προσδιορισμό των μεταβλητών του συστήματος:

- των τάσεων κατά μέτρο και γωνία σε όλους τους ζυγούς
- των ροών ενεργού και άεργου ισχύος σε όλες τις γραμμές
- των απωλειών, κ.α.

σε μια δεδομένη μόνιμη κατάσταση λειτουργίας, η οποία αντιστοιχεί σε μια ορισμένη εκάνα κατανομή των φορτίων, παραγόμενης ενεργού ισχύος, και τάσεων των γεννητών.

#### 1. Χαρακτηρισμός ζυγών του συστήματος

Τύποι ζυγών ροής φορτίου

Τύπος ζυγού	Γνωστές μεταβλητές	Άγνωστες μεταβλητές
Τελάνωση ή απορρόση	$V =  v_m, \delta = 0^\circ$	$P_D, Q_D$
Ζυγοί παραγωγής (PV)	$P_G, V$	$Q_G, \delta$
Ζυγοί φορτίου (PQ)	$P_D, Q_D$	$V, \delta$

Στο παραδείγμα είναι:

- Ζυγός 1 Ζυγός τελάνωσης (slack bus). Θεωρούμε ότι η τάση του είναι σταθερή κατά μέτρο και γωνία  $V_1 = 1 \angle 0^\circ$  α.μ.
- Ζυγός 2 Ζυγός παραγωγής (PV), αφού είναι γνωστό η παραγόμενη ενεργός ισχύς  $P_{G2} = 200\text{MW}$  και το μέτρο της τάσης  $V_2 = 1$  α.μ.
- Ζυγός 3 Ζυγός φορτίου (PQ), αφού είναι γνωστές οι ισχύεις φορτίου  $P_{D3} = 300\text{MW}, Q_{D3} = 100\text{MVAR}$

Στον πίνακα των δεδομένων των ζυγών ο ζυγός τελάνωσης συμβολίζεται με 1, οι ζυγοί PV με 2 και οι ζυγοί PQ με 0.

Στο παρακάτω κώδικα οι μεταβλητή  $nPV$  εμφανίζει το πλήθος των ζυγών PV του προβλήματος, ενώ η μεταβλητή  $nPQ$  εμφανίζει το πλήθος των ζυγών PQ. Οι συναρτήσεις  $pn$  και  $mq$  βρίσκουν τους ζυγούς PV και PQ αντίστοιχα.

```
In [8]: nPV = 0;
nPO = 0;

for k = 1:nbus
    if kb(k)==2 % PV bus
        nPV = nPV+1; end
    if kb(k)==3 % PQ bus
        nPO = nPO+1; end
end
nPV
nPO

pn = find(kb == 2 | kb == 1); % PV buses...
mq = find(kb == 0); % PQ buses...

nPV =
    1
nPO =
    1
```

Σχήμα 4. 5 : Στιγμιότυπο από το παράδειγμα επίλυσης ροής φορτίου σε Jupyter Notebook με τον υπολογισμό του πίνακα αγωγιμοτήτων, τη θεωρία και τον κώδικα για τον χαρακτηρισμό των ζυγών του συστήματος (3/9).

### Βήμα 3ο: Το πρόβλημα ροών φορτίου

1. Χαρακτηρισμός των ζυγών του συστήματος. Επεξηγούνται οι τύποι των ζυγών του συστήματος και προσδιορίζονται για το συγκεκριμένο σύστημα. Στο 8<sup>ο</sup> κελί κώδικα ένα for loop που επαναλαμβάνεται για κάθε ζυγό υπολογίζει τον αριθμό των PV ( $n_{PV}$ ) και PQ ( $n_{PQ}$ ) ζυγών. Έπειτα η συνάρτηση find βρίσκει και αποθηκεύει στα διανύσματα  $p_n$  και  $p_q$ , τον αύξοντα αριθμό των ζυγών PV και PQ αντίστοιχα (Σχήμα 4.5).
2. Σκιαγραφούνται τα δεδομένα και τα ζητούμενα του προβλήματος. Προσδιορίζεται το διάνυσμα κατάστασης ( $X$ ) και οι εξισώσεις ενεργού ( $\Delta P_k$ ) και αέργου ισχύος ( $\Delta Q_k$ ) από τις σχέσεις (13) και (14). Περιγράφεται η επαναληπτική διαδικασία που θα ακολουθηθεί μέσω ενός while loop που ελέγχει κάθε φορά αν ικανοποιούνται οι συνθήκες των σχέσεων (15). Για να εκτελεστεί απρόσκοπτα η διαδικασία της επίλυσης του προβλήματος ροής φορτίου, απαιτείται όλος ο κώδικας που περιέχεται στο while loop να γραφτεί σε ένα κελί κώδικα του Jupyter Notebook. Ωστόσο για να εξετάσουμε αναλυτικά τα βήματα του προγράμματος, θα παρουσιάσουμε τμηματικά την 1<sup>η</sup> επανάληψη και τα αποτελέσματά της και στη συνέχεια θα τρέξουμε συνολικά τον κώδικα. Στο 9<sup>ο</sup> κελί κώδικα φαίνεται ότι ορίζονται οι μεταβλητές P, Q ως στήλες μηδενικών με μέγεθος  $n_{bus}$ . Μέσα σε ένα διπλό for loop που επαναλαμβάνεται για όλους τους συνδυασμούς ζυγών υπολογίζονται τα P, Q και ακριβώς από κάτω σχηματίζονται οι επαναληπτικές εξισώσεις της ροής φορτίου των σχέσεων (13), (14). Στη συνέχεια του κώδικα φαίνεται πως λαμβάνουμε από τα αποτελέσματα που υπολογίστηκαν μόνο αυτά που μας ενδιαφέρουν, δηλαδή τα  $\Delta Q$  των PQ ζυγών και τα  $\Delta P$  όλων των ζυγών εκτός του ζυγού αναφοράς. Τα τελικά διανύσματα ενώνονται και δημιουργείται το διάνυσμα διαφορών M (Σχήματα 4.6 – 4.7).
3. Για να υπολογιστεί η Ιακωβιανή μήτρα ( $J_{ac}$ ) που δίνεται από τη σχέση (18), αρχικά ορίζονται στο 10<sup>ο</sup> κελί κώδικα οι υποπίνακες H, N, J, L που την αποτελούν και οι διαστάσεις τους (Σχήμα 4.7).
4. Στα επόμενα τέσσερα κελιά κώδικα υπολογίζονται οι υποπίνακες αυτοί, με βάση τις εξισώσεις που αντιστοιχούν στον καθένα, οι οποίες παραθέτονται ακριβώς πριν από κάθε κελί κώδικα, ενώ από κάτω εμφανίζεται το αποτέλεσμα που προκύπτει (Σχήματα 4.7 – 4.8).
5. Στο 15<sup>ο</sup> κελί κώδικα, ενώνονται τα παραπάνω διανύσματα για να σχηματιστεί η Ιακωβιανή μήτρα. Εδώ λαμβάνουμε το  $-J_{ac}$  που θα μας χρειαστεί στη συνέχεια (Σχήμα 4.8).

Βήμα 4ο: Υπολογισμός των νέων τάσεων και γωνιών του συστήματος

1. Από τη γενική εξίσωση Newton – Raphson (31), υπολογίζεται το διάνυσμα των αγνώστων  $X$  όπως φαίνεται στο 16<sup>ο</sup> κελί κώδικα (Σχήμα 4.8).
2. Οι νέες τιμές υπολογίζονται από τις σχέσεις (32), (33) χρησιμοποιώντας τις τιμές της προηγούμενης επανάληψης. Για την 1<sup>η</sup> επανάληψη χρησιμοποιούνται οι αρχικές τιμές. Όπως βλέπουμε στο 17<sup>ο</sup> κελί κώδικα οι εξισώσεις (32), (33) χρησιμοποιούνται για την εύρεση όλων των γωνιών εκτός της γωνίας του ζυγού αναφοράς και όλων των τάσεων των PQ ζυγών (Σχήμα 4.9).
3. Οι μοίρες μετατρέπονται σε ακτίνια από τη σχέση (34), όπως φαίνεται στο 18<sup>ο</sup> κελί κώδικα (Σχήμα 4.9).

### 2. Δεδομένα και ζητούμενα

**Το διάνυσμα κατάσταση**

Θεωρούμε ότι  $m$  είναι ο συνολικός αριθμός των ζυγίων του συστήματος και  $m$  ο αριθμός των ζυγίων φορτίου PQ. Γενικά για  $n$  ζυγιά έχουμε  $2n$  αγνώστους. Δεδομένου ότι η τάση στο ζυγό τολάντωσης είναι δεδομένη κατά μέτρο και γωνία, ενώ τα μέτρα των τάσεων είναι γνωστά στους ζυγούς παραγωγής, οι μεταβλητές που πρέπει να προσδιοριστούν για να περιγραφεί πλήρως η κατάσταση λειτουργίας του συστήματος είναι οι ακόλουθες:

- οι φασικές γωνίες τάσεων όλων των ζυγίων εκτός από το ζυγό τολάντωσης
- τα μέτρα τάσεων όλων των ζυγίων φορτίου.

Οι μεταβλητές αυτές αποικελούν το διάνυσμα κατάσταση  $X = \begin{bmatrix} \delta_2 \\ \dots \\ \delta_n \\ V_{n-m+1} \\ \dots \\ V_n \end{bmatrix}$  με διάσταση  $n - 1 + m$ . Άρα για την εύρεση των αγνώστων χρειαζόμαστε  $n - 1 + m$  ανεξάρτητες εξισώσεις.

- $n - 1$  εξισώσεις πραγματικής ισχύος (για ζυγούς PV και PQ):

$$\Delta P_k^{(n)} = P_{k,p,k} - V_k^{(n)} \cdot \sum_{m \in A(k)} V_m^{(n)} \cdot [G_{km} \cdot \cos(\delta_k^{(n)} - \delta_m^{(n)}) + B_{km} \cdot \sin(\delta_k^{(n)} - \delta_m^{(n)})] \quad (13)$$

- $m$  εξισώσεις αέργου ισχύος (για ζυγούς PQ):

$$\Delta Q_k^{(n)} = Q_{k,p,k} - V_k^{(n)} \cdot \sum_{m \in A(k)} V_m^{(n)} \cdot [G_{km} \cdot \sin(\delta_k^{(n)} - \delta_m^{(n)}) - B_{km} \cdot \cos(\delta_k^{(n)} - \delta_m^{(n)})] \quad (14)$$

Στο παράδειγμα ενός γραμμήματος την τάση του ζυγού 1 κατά μέτρο και γωνία  $V_1 = 1 \angle 0^\circ$  α.μ. καθώς και ο μέτρο της τάσης του ζυγού 2,  $V_2 = 1$  α.μ. Επίσης, ένας αγνώστος του προβλήματος είναι 3, οι γωνίες  $\delta_2, \delta_3$  και η τάση  $V_3$ . Το διάνυσμα κατάσταση λοιπόν έχει τη μορφή  $X = \begin{bmatrix} \delta_2 \\ \delta_3 \\ V_3 \end{bmatrix}$  και για να υπολογιστεί θα χρειαστούν δύο εξισώσεις πραγματικής ( $\Delta P_2, \Delta P_3$ ) και μία εξίσωση αέργου ισχύος ( $\Delta Q_3$ ).

**Δηλαδή:**

- **ζυγοί PV, PQ:**

$$\Delta P_2 = P_{p,2} - V_2 \cdot V_1 \cdot [G_{21} \cdot \cos(\delta_2 - \delta_1) + B_{21} \cdot \sin(\delta_2 - \delta_1)] - V_2 \cdot V_3 \cdot [G_{23} \cdot \cos(\delta_2 - \delta_3) + B_{23} \cdot \sin(\delta_2 - \delta_3)]$$

$$\Delta P_3 = P_{p,3} - V_3 \cdot V_1 \cdot [G_{31} \cdot \cos(\delta_3 - \delta_1) + B_{31} \cdot \sin(\delta_3 - \delta_1)] - V_3 \cdot V_2 \cdot [G_{32} \cdot \cos(\delta_3 - \delta_2) + B_{32} \cdot \sin(\delta_3 - \delta_2)]$$

- **ζυγοί PQ:**

$$\Delta Q_3 = Q_{p,3} - V_3 \cdot V_1 \cdot [G_{31} \cdot \sin(\delta_3 - \delta_1) - B_{31} \cdot \cos(\delta_3 - \delta_1)] - V_3 \cdot V_2 \cdot [G_{32} \cdot \sin(\delta_3 - \delta_2) - B_{32} \cdot \cos(\delta_3 - \delta_2)]$$

**Επαναληπτική διαδικασία**

- Πριν ξεκινήσουν οι επαναληπτικές της μεθόδου, Newton-Raphson θεωρείται μια ενιαία τιμή εκκίνησης για τις τάσεις όλων των ζυγίων του δικτύου, που δεν έχουν καθορισμένη τιμή, στα  $V = 1 \angle 0^\circ$  α.μ.
- Χρησιμοποιώντας αυτή την αρχική εκτίμηση υπολογίζεται, με τη βοήθεια των εξισώσεων ροής φορτίου μια βελτιωμένη δεύτερη εκτίμηση η οποία χρησιμοποιείται για να βρεθεί μία τρίτη κ.ο.κ.
- Οι επαναλήψεις συνεχίζονται μέχρι οι νίκες τιμές των μεταβλητών του συστήματος να διαφέρουν από τις αρχικές πολύ λίγο, δηλαδή όταν ικανοποιούνται τα κριτήρια σύγκλισης:

$$\begin{cases} |\Delta P_k^{(n)}| \leq \epsilon \\ |\Delta Q_k^{(n)}| \leq \epsilon \end{cases} \quad (15)$$

Οι δοσμέ αναλυτικά την 1η επανάληψη:

Στο πρόγραμμα αυτό η επαναληπτική διαδικασία εκκαστεί μέσα ενός while loop, το οποίο στην είσοδο του ελέγχει το κριτήριο σύγκλισης των σχέσεων (15). Στο παρακάτω κομμάτι κώδικα υπολογίζονται οι επαναληπτικές εξισώσεις ροής φορτίου σε όλους τους ζυγούς του ΣΠΕ. Από αυτές χρειαζόμαστε τις δύο εξισώσεις πραγματικής ισχύος ( $\Delta P_2, \Delta P_3$ ) που αποθηκεύονται στο διάνυσμα dP και τη μία εξίσωση αέργου ισχύος ( $\Delta Q_3$ ), που αποθηκεύεται στο διάνυσμα dQ. Το διάνυσμα αυτό ανιωννεται για να δημιουργηθεί το διάνυσμα διαφορών M που έχει ίδιες διαστάσεις με το X.

```
In [9]: Tol = 1;
Iter = 0;
%while (Tol) >= 0.001 % Iteration starting.. (16)
Iter = Iter + 1;
P = zeros(nbus,1); Q = zeros(nbus,1); % Calculate P and Q

for k = 1:nbus
    for m = 1:nbus
        P(k) = P(k) + Vr(k)*Vr(m)*(G(k,m)*cos(delta(k)-delta(m)) + B(k,m)*sin(delta(k)-delta(m))); % (13)
        Q(k) = Q(k) + Vr(k)*Vr(m)*(G(k,m)*sin(delta(k)-delta(m)) - B(k,m)*cos(delta(k)-delta(m))); % (14)
    end
end

dPa = Psp-P; % (13)
dQa = Qsp-Q; % (14)

k = 1;
dQ = zeros(nPQ,1);
for l = 1:nbus
    if kb(1) == 0
        dQ(k,1) = dQa(1);
        k = k+1;
    end
end

dP = dPa(2:nbus)
dQ

% Mismatch Vector
M = [dP; dQ] % (17)

dP =
    1.5000
   -3.0000
dQ =
    *
```

Σχήμα 4. 6 : Στιγμιότυπο από το παράδειγμα επίλυσης ροής φορτίου σε Jupyter Notebook όπου εξηγείται η διαδικασία της δημιουργίας του διανύσματος διαφορών μέσω των επαναληπτικών εξισώσεων ροής φορτίου και δίνεται ο αντίστοιχος κώδικας (4/9).

$\delta P = \begin{bmatrix} 1.5000 \\ -3.0000 \\ -1.0000 \end{bmatrix}$   
 $Q = \begin{bmatrix} 1.5000 \\ -3.0000 \\ -1.0000 \end{bmatrix}$

### 3. Υπολογισμός Ιακωβιανής μήτρας

Η Ιακωβιανή μήτρα έχει τη μορφή

$$Jac = \begin{bmatrix} H & N \\ J & L \end{bmatrix} \quad (18)$$

- Ο υποπίνακας H έχει διαστάσεις  $(n-1) \times (n-1)$
- Ο υποπίνακας N έχει διαστάσεις  $(n-1) \times m$
- Ο υποπίνακας J έχει διαστάσεις  $m \times (n-1)$
- Ο υποπίνακας L έχει διαστάσεις  $m \times m$

Στο παράδειγμα αυτό η Ιακωβιανή μήτρα έχει τη μορφή

$$Jac = \begin{bmatrix} H_{22} & H_{23} & N_{23} \\ H_{32} & H_{33} & N_{33} \\ J_{32} & J_{33} & L_{33} \end{bmatrix}$$

```

In [10]: clear H N J L
H = zeros(nbus-1,nbus-1);
N = zeros(nbus-1,npQ);
J = zeros(npQ,nbus-1);
L = zeros(npQ,npQ);

```

$$H_{ik} = \frac{\partial \Delta P_k}{\partial \delta_k} = V_k^2 \cdot B_{ik} + Q_k \quad (19)$$

όπου  $Q_k = V_k \cdot \sum_{l \neq m} V_m \cdot [G_{km} \cdot \sin(\delta_k - \delta_m) - B_{km} \cdot \cos(\delta_k - \delta_m)]$  (20)

$$H_{lm} = \frac{\partial \Delta P_k}{\partial \delta_m} = -V_k \cdot V_m \cdot [G_{km} \cdot \sin(\delta_k - \delta_m) - B_{km} \cdot \cos(\delta_k - \delta_m)], k \neq m \quad (21)$$

Θα είναι:

$$H_{22} = V_2^2 \cdot B_{22} + V_2 \cdot V_1 \cdot [G_{21} \cdot \sin(\delta_2 - \delta_1) - B_{21} \cdot \cos(\delta_2 - \delta_1)] + V_2 \cdot V_3 \cdot [G_{23} \cdot \sin(\delta_2 - \delta_3) - B_{23} \cdot \cos(\delta_2 - \delta_3)]$$

$$H_{23} = -V_2^2 \cdot V_3 \cdot [G_{23} \cdot \sin(\delta_2 - \delta_3) - B_{23} \cdot \cos(\delta_2 - \delta_3)]$$

$$H_{32} = -V_3^2 \cdot V_2 \cdot [G_{32} \cdot \sin(\delta_3 - \delta_2) - B_{32} \cdot \cos(\delta_3 - \delta_2)]$$

$$H_{33} = V_3^2 \cdot B_{33} + V_3 \cdot V_1 \cdot [G_{31} \cdot \sin(\delta_3 - \delta_1) - B_{31} \cdot \cos(\delta_3 - \delta_1)] + V_3 \cdot V_2 \cdot [G_{32} \cdot \sin(\delta_3 - \delta_2) - B_{32} \cdot \cos(\delta_3 - \delta_2)]$$

```

In [11]: for i = 1:(nbus-1)
k = i+1;
for j = 1:(nbus-1)
m = j+1;
if k == m
for m = 1:nbus
H(i,j) = H(i,j) + Vm(k)*Vm(m)*(G(k,m)*sin(delta(k)-delta(m)) - B(k,m)*cos(delta(k)-delta(m))); % (20)
end
H(i,j) = (Vm(k)^2)*B(k,k) + H(i,j); % (19)
else
H(i,j) = -Vm(k)*Vm(m)*(G(k,m)*sin(delta(k)-delta(m)) - B(k,m)*cos(delta(k)-delta(m))); % (21)
end
end
end
H

```

$$N_{ik} = \frac{\partial \Delta P_k}{\partial V_k} = -V_k \cdot G_{ik} - \frac{P_k}{V_k} \quad (22)$$

όπου  $\frac{P_k}{V_k} = \sum_{l \neq m} V_m \cdot [G_{km} \cdot \cos(\delta_k - \delta_m) + B_{km} \cdot \sin(\delta_k - \delta_m)]$  (23)

$$N_{km} = \frac{\partial \Delta P_k}{\partial V_m} = -V_k \cdot [G_{km} \cdot \cos(\delta_k - \delta_m) + B_{km} \cdot \sin(\delta_k - \delta_m)], k \neq m \quad (24)$$

$N_{23} = -V_2 \cdot [G_{23} \cdot \cos(\delta_2 - \delta_3) + B_{23} \cdot \sin(\delta_2 - \delta_3)]$   
 $N_{33} = -V_3^2 \cdot G_{33} - V_3 \cdot [G_{31} \cdot \cos(\delta_3 - \delta_1) + B_{31} \cdot \sin(\delta_3 - \delta_1)] - V_2 \cdot [G_{32} \cdot \cos(\delta_3 - \delta_2) + B_{32} \cdot \sin(\delta_3 - \delta_2)]$

```

In [12]: for i = 1:(nbus-1)
k = i+1;
for j = 1:npQ
m = pc(j);
if k == m
for m = 1:nbus
N(i,j) = N(i,j) + Vm(m)*(G(k,m)*cos(delta(k)-delta(m)) + B(k,m)*sin(delta(k)-delta(m))); % (23)
end
N(i,j) = -Vm(k)*G(k,k) - N(i,j); % (22)
else
N(i,j) = -Vm(k)*(G(k,m)*cos(delta(k)-delta(m)) + B(k,m)*sin(delta(k)-delta(m))); % (24)
end
end
end
N

```

N =  
0  
0

Σχήμα 4. 7 : Στιγμιότυπο από το παράδειγμα επίλυσης ροής φορτίου σε Jupyter Notebook με τις εξισώσεις και τον κώδικα για τον υπολογισμό των υποπινάκων H και N της Ιακωβιανής μήτρας (5/9).

$$J_{kk} = \frac{\partial \Delta Q_k}{\partial \delta_k} = V_k^2 \cdot G_{kk} - P_k \quad (25)$$

$$\text{όπου } P_k = V_k \cdot \sum_{m \neq k} V_m \cdot [G_{km} \cdot \cos(\delta_k - \delta_m) + B_{km} \cdot \sin(\delta_k - \delta_m)] \quad (26)$$

$$J_{km} = \frac{\partial \Delta Q_k}{\partial \delta_m} = V_k \cdot V_m \cdot [G_{km} \cdot \cos(\delta_k - \delta_m) + B_{km} \cdot \sin(\delta_k - \delta_m)], k \neq m \quad (27)$$

$$J_{32} = V_1 \cdot V_2 \cdot [G_{32} \cdot \cos(\delta_3 - \delta_2) + B_{32} \cdot \sin(\delta_3 - \delta_2)]$$

$$J_{33} = V_3^2 \cdot G_{33} - V_3 \cdot V_1 \cdot [G_{31} \cdot \cos(\delta_3 - \delta_1) - B_{31} \cdot \sin(\delta_3 - \delta_1)] - V_3 \cdot V_2 \cdot [G_{32} \cdot \cos(\delta_3 - \delta_2) - B_{32} \cdot \sin(\delta_3 - \delta_2)]$$

```
In [13]: for i = 1:nPO
k = pq(i);
for j = 1:(nbus-1)
n = j+1;
if k == n
for m = 1:nbus
J(i,j) = J(i,j) + Vn(k)*Vm(m)*(G(k,m)*cos(delta(k)-delta(m)) + B(k,m)*sin(delta(k)-delta(m))); % (26)
end
else
J(i,j) = (Vn(k)^2)*G(k,k)-J(i,j); % (25)
else
J(i,j) = Vn(k)*Vm(m)*(G(k,m)*cos(delta(k)-delta(m)) + B(k,m)*sin(delta(k)-delta(m))); % (27)
end
end
end
end
J =
0      0
```

$$L_{kk} = \frac{\partial \Delta Q_k}{\partial V_m} = V_k \cdot B_{kk} - \frac{Q_k}{V_k} \quad (28)$$

$$\text{όπου } \frac{Q_k}{V_k} = \sum_{m \neq k} V_m \cdot [G_{km} \cdot \sin(\delta_k - \delta_m) - B_{km} \cdot \cos(\delta_k - \delta_m)] \quad (29)$$

$$L_{km} = \frac{\partial \Delta Q_k}{\partial V_m} = -V_k \cdot [G_{km} \cdot \sin(\delta_k - \delta_m) - B_{km} \cdot \cos(\delta_k - \delta_m)], k \neq m \quad (30)$$

$$L_{33} = V_3 \cdot B_{33} - V_3 \cdot [G_{31} \cdot \sin(\delta_3 - \delta_1) - B_{31} \cdot \cos(\delta_3 - \delta_1)] - V_3 \cdot [G_{32} \cdot \sin(\delta_3 - \delta_2) - B_{32} \cdot \cos(\delta_3 - \delta_2)]$$

```
In [14]: for i = 1:nPO
k = pq(i);
for j = 1:nPO
n = pq(j);
if k == n
for k = 1:nbus
L(i,j) = L(i,j) + Vn(m)*(G(k,m)*sin(delta(k)-delta(m)) - B(k,m)*cos(delta(k)-delta(m))); % (29)
end
else
L(i,j) = Vn(k)*B(k,k)-L(i,j); % (28)
else
L(i,j) = -Vn(k)*(G(k,m)*sin(delta(k)-delta(m)) - B(k,m)*cos(delta(k)-delta(m))); % (30)
end
end
end
end
L =
-24.9925
```

Ετοιμάζω η Ιακωβιανή μήτρα για την 1η επανάληψη θα είναι:

```
In [15]: Jac = -[H N; J L] % Jacobian matrix
Jac =
30.0000 -10.0000 0
10.0000 24.9925 0
0 0 24.9925
```

#### 4. Υπολογισμός των νέων τάσεων και γωνιών του συστήματος

Από τη γενική εξίσωση Newton-Raphson υπολογίζεται το διάνυσμα των αγνώστων:

$$X = -J^{-1}M^{(0)} \Rightarrow \begin{bmatrix} \Delta \delta \\ \Delta V \end{bmatrix} = \begin{bmatrix} H & N \\ J & L \end{bmatrix}^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \quad (31)$$

```
In [16]: % Correction Vector
X = inv(Jac)*M % (31)
dTh = X(1:nbus-1) % Change in Voltage Angle...
dV = X(nbus:end) % Change in Voltage Magnitude...
X =
0.0115
-0.1154
-0.0400
dTh =
0.0115
-0.1154
dV =
-0.0400
```

Σχήμα 4. 8 : Στιγμιότυπο από το παράδειγμα επίλυσης ροής φορτίου σε Jupyter Notebook όπου υπολογίζονται οι υποπίνακες J και L της Ιακωβιανής μήτρας, δίνεται η τελική μορφή της και σχηματίζεται το διάνυσμα αγνώστων X (6/9).

Οι νέες τιμές υπολογίζονται χρησιμοποιώντας τις τιμές της προηγούμενης επανάληψης:

$$\begin{bmatrix} \delta \\ \mathbf{V} \end{bmatrix}^{(n+1)} = \begin{bmatrix} \Delta\delta \\ \Delta\mathbf{V} \end{bmatrix} + \begin{bmatrix} \delta \\ \mathbf{V} \end{bmatrix}^{(n)} \quad (32)$$

$$\begin{bmatrix} \delta \\ \mathbf{V} \end{bmatrix}^{(n+1)} = \begin{bmatrix} \Delta\delta \\ \Delta\mathbf{V} \end{bmatrix} + \begin{bmatrix} \delta \\ \mathbf{V} \end{bmatrix}^{(n)} \quad (33)$$

Απλάτη:

$$\begin{bmatrix} 0 \\ \delta_2 \\ \delta_3 \end{bmatrix}^{(1)} = \begin{bmatrix} \Delta\delta \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

και

$$\begin{bmatrix} 1 \\ 1 \\ \mathbf{V}_3 \end{bmatrix}^{(1)} = \begin{bmatrix} \Delta\mathbf{V} \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

```
In [17]: % Updating State Vectors..
% Voltage Angle..
delta(2:nbus) = dTh + delta(2:nbus)' % (32)
% Voltage Magnitude..
k = 1;
for i = 2:nbus
    if kb(i) == 0
        Vn(i) = dV(k) + Vn(i); % (33)
        k = k+1;
    end
end
Vn
..
delta =
    0    0.8115   -0.1154
Vn =
    1.0000    1.0000    0.9600
```

Μετατροπή γωνιών από ακέια σε μοίρες:

$$\delta' = \frac{180'}{\pi} \cdot (\text{rad}) \quad (34)$$

```
In [18]: delrad=180/pi*delta % (34)
deltad =
    0    0.6603   -6.6133
```

### 5. Υπολογισμός ισχύων στους ζυγούς

Οι ισχύς που χρειάζεται να προσδιοριστούν αν πρόκειται:

- Για ζυγό καλάντισης, είναι η ενεργός και η άεργος ισχύς του,  $P_G, Q_G$
- Για ζυγό PV, η άεργος ισχύς του,  $Q_G$

Οι ισχύς αυτές δίνονται από τις σχέσεις:

$$P_G = \bar{P}_k \cdot S_{base} + P_D \quad (35)$$

$$Q_G = Q_c \cdot S_{base} + Q_D \quad (36)$$

Και οι μηχανικές ισχύς εκφράζονται ως:

$$S_k = P_k + jQ_k \quad (37)$$

```
In [19]: k=0;
for n = 1:nbus
    if kb(n) == 1
        k=k+1;
        S(n) = P(n)+j*Q(n); % (37)
        Pg(n) = P(n)*baseva + Pd(n); % (35)
        Qg(n) = Q(n)*baseva + Qd(n); % (36)
        elseif kb(n) == 2
            k=k+1;
            S(n)=P(n)+j*Q(n); % (37)
            Qg(n) = Q(n)*baseva + Qd(n); % (36)
        end
        yload(n) = (Pd(n) - j*Qd(n))/(baseva*Vn(n)^2);
    end
end
Pg
Pq
Qg
```

Το αρχείο Emissi υπολογίζει και εμφανίζει τα τελικά αποτελέσματα της ροής φορτίου και το αρχείο Lineflow υπολογίζει και εμφανίζει τις ροές ισχύος στις γραμμές καθώς και τις αντίστοιχες απώλειες ισχύος (όλα τα μεγέθη σε MW και MVar).

### 6. Υπολογισμός Ροών Ισχύος Γραμμής

Σχήμα 3: Ροή ισχύος στη γραμμή k – m

Στη γραμμή μεταφοράς k – m του Σχήματος 3:

- Η ανα μονάδα ροή ενεργού ισχύος  $P_{km}$  στον ζυγό k, με κατεύθυνση από τον ζυγό k, προς τον ζυγό m.

Σχήμα 4. 9 : Στιγμιότυπο από το παράδειγμα επίλυσης ροής φορτίου σε Jupyter Notebook με τις εξισώσεις και τον κώδικα για τον υπολογισμό των νέων τάσεων, γωνιών και ισχύων του συστήματος (7/9).



### Βήμα 5ο: Αποτελέσματα ροής φορτίου

1. Προσδιορίζονται η ενεργός ( $P_G$ ) και άεργος ισχύς ( $Q_G$ ) του ζυγού αναφοράς, η άεργος ισχύς ( $Q_G$ ) για τον ζυγό PV και τελικά όλες οι μιγαδικές ισχύς ( $S$ ) από τις εξισώσεις (35), (36) και (37) αντίστοιχα, όπως φαίνεται στο 19<sup>ο</sup> κελί κώδικα (Σχήμα 4.9).
2. Στη συνέχεια παρουσιάζονται οι γενικές εξισώσεις των ροών ισχύος των γραμμών και στο 20<sup>ο</sup> κελί κώδικα εκτελούνται τα αρχεία Busoutt.m και Lineflow.m που εμφανίζουν τα τελικά αποτελέσματα της ροής φορτίου για την 1<sup>η</sup> επανάληψη: τις τάσεις, τις γωνίες και τις ισχύεις στους ζυγούς, τις ροές ισχύος και τις απώλειες στις γραμμές. Ας σημειωθεί ότι όλες οι εντολές σχετικές με την επανάληψη του κώδικα έως αυτό το σημείο έχουν μπει ως σχόλια ώστε να δείξουμε τμηματικά κάθε βήμα της εκτέλεσης του προγράμματος και τα αντίστοιχα αποτελέσματα (Σχήμα 4.10).
3. Στο 21<sup>ο</sup> κελί κώδικα (το οποίο είναι αναδιπλωμένο) έχουμε συμπεριλάβει όλο τον κώδικα που εξηγήσαμε παραπάνω, ενεργοποιώντας αυτή τη φορά το κεντρικό while loop και τις συνθήκες του ώστε να μπορέσει ο κώδικας να τρέξει επαναληπτικά. Η συγκεκριμένη άσκηση, καθώς είναι σχεδιασμένη για επίλυση «με το χέρι» και όχι για να τρέξει σε προγραμματιστικό περιβάλλον, συγκλίνει μετά από πολλές επαναλήψεις και γι' αυτό ορίσαμε να εμφανιστούν τα αποτελέσματα των τριών πρώτων επαναλήψεων, όπως φαίνεται παρακάτω (Σχήματα 4.10 – 4.11).

• Η ανα μονάδα ροή ενεργού ισχύος  $P_{km}$  στον ζυγό  $k$ , με κατεύθυνση από τον ζυγό  $k$  προς τον ζυγό  $m$ :

$$P_{km} = (g_{km} + g_{skm}) \cdot V_m^2 - V_k \cdot V_m \cdot [g_{km} \cdot \cos(\delta_k - \delta_m) + b_{km} \cdot \sin(\delta_k - \delta_m)] \quad (38)$$

• Η ανα μονάδα ροή ενεργού ισχύος  $P_{mk}$  στον ζυγό  $m$ , με κατεύθυνση από τον ζυγό  $m$  προς τον ζυγό  $k$ :

$$P_{mk} = (g_{km} + g_{skm}) \cdot V_m^2 - V_m \cdot V_k \cdot [g_{km} \cdot \cos(\delta_m - \delta_k) + b_{km} \cdot \sin(\delta_m - \delta_k)] \quad (39)$$

• Οι ανα μονάδα απώλειες ενεργού ισχύος  $P_{Loss_{km}}$ :

$$P_{Loss_{km}} = P_{km} + P_{mk} \quad (40)$$

Στη γραμμή μεταφοράς  $k - m$  του Σχήματος 3:

• Η ανα μονάδα ροή αεργού ισχύος  $Q_{km}$  στον ζυγό  $k$ , με κατεύθυνση από τον ζυγό  $k$  προς τον ζυγό  $m$ :

$$Q_{km} = -(b_{km} + b_{skm}) \cdot V_k^2 - V_k \cdot V_m \cdot [g_{km} \cdot \sin(\delta_k - \delta_m) + b_{km} \cdot \cos(\delta_k - \delta_m)] \quad (41)$$

• Η ανα μονάδα ροή αεργού ισχύος  $Q_{mk}$  στον ζυγό  $m$ , με κατεύθυνση από τον ζυγό  $m$  προς τον ζυγό  $k$ :

$$Q_{mk} = (b_{km} + b_{skm}) \cdot V_m^2 - V_m \cdot V_k \cdot [g_{km} \cdot \sin(\delta_m - \delta_k) + b_{km} \cdot \cos(\delta_m - \delta_k)] \quad (42)$$

• Οι ανα μονάδα απώλειες ενεργού ισχύος  $Q_{Loss_{km}}$ :

$$Q_{Loss_{km}} = Q_{km} + Q_{mk} \quad (43)$$

Στο σημείο αυτό ολοκληρώνεται η 1η επανάληψη της μεθόδου. Η μεταβλητή `Tol` παίρνει την τιμή της μέγιστης απόλυτης τιμής του διακόσματος διαφορικού `M` ώστε να ελεγχθεί το κριτήριο σύγκλισης στην αρχή του `while` loop.

```
In [14]: Resoult
LineFlow
%Tolerance...
%Tol = max(abs(M));
end
```

Power Flow Solution by Newton-Raphson Method  
Maximum Power Mismatch = 1  
No. of Iterations = 1

Bus No.	Voltage Mag.	Angle Degree	-----Load-----		---Generation---		Injected Pvar
			MW	Mvar	MW	Mvar	
1	1.000	0.00	0.00	0.00	0.00	0.00	0.00
2	1.000	0.66	50.00	50.00	200.00	0.00	0.00
3	0.960	-6.61	300.00	100.00	0.00	0.00	0.00
Total			350.00	150.00	200.00	0.00	0.00

Line Flow and Losses

--Line-- from to	Power at bus & line flow			--Line loss--		Transformer tap
	MW	Mvar	MVA	MW	Mvar	
1	0.000	0.000	0.000			
	-23.050	0.133	23.050	0.000	0.266	
	165.758	69.565	179.763	0.000	21.554	
2	0.000	0.000	158.114			
	23.050	0.133	23.050	0.000	0.266	
	121.543	47.737	130.582	0.000	17.052	
3	0.000	0.000	316.228			
	-165.758	-48.011	172.571	0.000	21.554	
	-121.543	-30.606	125.357	0.000	17.052	
Total loss				0.000	38.871	

Το παρακάτω κελί περιέχει όλο τον κώδικα της ροής φορτίου με τη μέθοδο Newton-Raphson που περιγράφει αναλυτικά παραπάνω, το οποίο τρέχει και εμφανίζει τα αποτελέσματα για τις τρεις πρώτες επαναλήψεις.

```
In [26]: % Newton_Raphson Iterations=
```

Power Flow Solution by Newton-Raphson Method  
Maximum Power Mismatch = 1  
No. of Iterations = 1

Bus No.	Voltage Mag.	Angle Degree	-----Load-----		---Generation---		Injected Pvar
			MW	Mvar	MW	Mvar	
1	1.000	0.00	0.00	0.00	0.00	0.00	0.00
2	1.000	0.66	50.00	50.00	200.00	0.00	0.00
3	0.960	-6.61	300.00	100.00	0.00	0.00	0.00
Total			350.00	150.00	200.00	0.00	0.00

Line Flow and Losses

--Line-- from to	Power at bus & line flow			--Line loss--		Transformer tap
	MW	Mvar	MVA	MW	Mvar	
1	0.000	0.000	0.000			
	-23.050	0.133	23.050	0.000	0.266	
	165.758	69.565	179.763	0.000	21.554	
2	0.000	0.000	0.000			
	23.050	0.133	23.050	0.000	0.266	
	121.543	47.737	130.582	0.000	17.052	
3	0.000	0.000	316.228			
	-165.758	-48.011	172.571	0.000	21.554	
	-121.543	-30.606	125.357	0.000	17.052	

Σχήμα 4. 10 : Στιγμιότυπο από το παράδειγμα επίλυσης ροής φορτίου σε Jupyter Notebook όπου δίνονται οι γενικές εξισώσεις ροών ισχύος και τα αποτελέσματα ροής φορτίου για την πρώτη επανάληψη (8/9).

```

Total loss          0.000  30.871
Power Flow Solution by Newton-Raphson Method
Maximum Power Mismatch = 3
No. of Iterations = 2

Bus Voltage Angle Load Generation Injected
No. Mag. Degree MW Mvar MW Mvar Mvar
1 1.000 0.00 0.00 0.00 142.71 69.70 0.00
2 1.000 0.67 50.00 50.00 200.00 97.87 0.00
3 0.950 -6.98 300.00 100.00 0.00 0.00 0.00
Total 350.00 150.00 342.71 167.57 0.00

Line Flow and Losses
--Line-- Power at bus & line flow --Line loss-- Transformer
from to MW Mvar MVA MW Mvar tap
1 142.708 69.698 212.408
2 -23.358 0.136 23.359 0.000 0.273
3 173.262 84.944 192.965 0.000 24.836
2 144.593 47.870 192.463
1 23.358 0.136 23.359 0.000 0.273
3 126.576 58.072 139.261 0.000 19.394
3 -207.301 -78.697 316.228
1 -173.262 -60.108 183.393 0.000 24.836
2 -126.576 -38.678 132.353 0.000 19.394

Total loss          0.000  44.543
Power Flow Solution by Newton-Raphson Method
Maximum Power Mismatch = 0.213034
No. of Iterations = 3

Bus Voltage Angle Load Generation Injected
No. Mag. Degree MW Mvar MW Mvar Mvar
1 1.000 0.00 0.00 0.00 149.00 85.00 0.00
2 1.000 0.67 50.00 50.00 200.00 108.21 0.00
3 0.950 -6.99 300.00 100.00 0.00 0.00 0.00
Total 350.00 150.00 349.00 103.29 0.00

Line Flow and Losses
--Line-- Power at bus & line flow --Line loss-- Transformer
from to MW Mvar MVA MW Mvar tap
1 149.904 85.000 234.904
2 23.362 0.136 23.363 0.000 0.273
3 173.362 85.748 193.409 0.000 24.951
2 149.934 58.700 208.142
1 23.362 0.136 23.363 0.000 0.273
3 126.638 58.669 139.513 0.000 19.472
3 209.838 98.786 316.228
1 173.362 60.798 183.714 0.000 24.951
2 -126.630 -39.137 132.547 0.000 19.472

Total loss          0.000  44.696

```

#### Βιβλιογραφία

- Παπαδό, Β. Ανάλυση Συστήματος Ηλεκτρικής Ενέργειας: Μόνη Κατάσταση Λειτουργίας (Τόμος 1), Αθήνα 1995, Εκδόσεις ΕΜΠ
- Βουρνά, Κ., Κονιάς, Γ. Εισαγωγή στα Συστήματα Ηλεκτρικής Ενέργειας, Αθήνα 2001
- Γεωργιάδης, Π. Σύγχρονα Συστήματα Μεταφοράς και Διανομής Ηλεκτρικής Ενέργειας, Ηλεκτρονικό βιβλίο, Συντάκτορας ΕΛΠΗΓΑΚ/Ακαδημαϊκών Βιβλιοθηκών (ΖΕΑΒ), Αθήνα, 2015. ISBN: 978-960-603-038-3

In | | :

Σχήμα 4. 11 : Στιγμιότυπο από το παράδειγμα επίλυσης ροής φορτίου σε Jupyter Notebook όπου εμφανίζονται τα αποτελέσματα ροής φορτίου των τριών πρώτων επαναλήψεων (9/9).

### 4.2.3 Εργαστηριακή άσκηση σε γλώσσα MATLAB στο cloud (CoCalc)

Η βασική εφαρμογή στην οποία εργαστήκαμε ώστε να υλοποιήσουμε τον κώδικα της ροής φορτίου αναφέρεται σε ένα λίγο πιο σύνθετο σύστημα τεσσάρων ζυγών, το οποίο εξετάζεται στην 1<sup>η</sup> Εργαστηριακή Άσκηση του μαθήματος Ανάλυση ΣΗΕ, όπως αναφέρθηκε παραπάνω.

Η εφαρμογή αυτή παρουσιάζεται μέσω του CoCalc, μιας cloud εναλλακτικής του Jupyter Notebook, η οποία διαθέτει τον πυρήνα Octave, ο οποίος είναι σε μεγάλο βαθμό συμβατός με τον πυρήνα MATLAB. Η επιλογή αυτή διευκολύνει τον χρήστη, καθώς του επιτρέπει να ξεπεράσει το εμπόδιο της εγκατάστασης λογισμικού και να τρέξει το πρόγραμμά του, δημιουργώντας απλά έναν λογαριασμό στο site του CoCalc μέσω mail. Ωστόσο η εφαρμογή μας έχει εκτελεστεί και στην τοπική εγκατάσταση του Jupyter και τα αποτελέσματα είναι όμοια.

Εν προκειμένω δεν θα σταθούμε στην περιγραφή του αρχείου και στην ανάλυση του κώδικα και της λειτουργίας του καθώς είναι σχεδόν ίδια με την προηγούμενη εφαρμογή. Ωστόσο υπογραμμίζεται ότι σε αυτό το παράδειγμα γίνεται σύγκριση των αποτελεσμάτων της εφαρμογής στο CoCalc (σύγκλιση, αριθμητικά αποτελέσματα) (Σχήματα 4.19 – 4.20) με αυτά του λογισμικού MATLAB (Σχήμα 4.2) ώστε να διαπιστωθεί η ομοιότητά τους, γεγονός που υποδεικνύει την ορθότητα του κώδικα μας. Αυτό βέβαια είναι αναμενόμενο καθώς εκεί βασιστήκαμε για να εξασφαλίσουμε ότι το πρόγραμμά μας λειτουργεί σωστά.

Επίσης αξίζει να σημειωθεί ότι στην εφαρμογή αυτή προσθέσαμε τα απαραίτητα τμήματα κώδικα για την κατασκευή οχτώ γραφικών παραστάσεων για την καλύτερη απεικόνιση των αποτελεσμάτων. Συγκεκριμένα δείχνουν την μεταβολή των τάσεων σε κάθε έναν από τους ζυγούς (A, B, C, D), της ενεργού και αέργου παραγωγής του ζυγού αναφοράς (A) και των ζυγών PV (B), σε κάθε επανάληψη (Σχήματα 4.20 – 4.21).

Εργαστήριο ΑΣΗΕ1 - CoCalc

https://cocalc.com/projects/f64da667-7905-4b56-927c-76561b2f2d74/files/erkeia/Newton\_Rap...  
 Projects Εργαστήριο ΑΣΗΕ1 CoCalc Help Account

Files New Log Find Settings Newton\_Raphson\_method\_matlabX Newton\_Raphson\_method\_matlabX  
 B I U X X' s ss I E C - P @ o y E E E I x A T1 H -

File Save TimeTravel Reload Q Q Print HTML

### Ανάλυση Συστημάτων Ηλεκτρικής Ενέργειας - Μόνιμη Κατάσταση Λειτουργίας

#### ΡΟΗ ΦΟΡΤΙΟΥ

**Σχήμα 1: Σύστημα τεσσάρων ζυγών**

Για το σύστημα ηλεκτρικής ενέργειας με την τοπολογία του παραπάνω σχήματος δίνονται τα ακόλουθα δεδομένα:

- Επαγωγικές αντιδράσεις των γραμμών μεταφοράς:  $Z_{AB(1)} = Z_{AB(2)} = j0,2$  α.μ.,  $Z_{CD} = j0,1$  α.μ.,  $Z_{DA} = Z_{BC} = j0,2$  α.μ.
- Στοιχεία ζυγών:
  - A: Τάση  $V_A = 1,0$  α.μ.
  - B: Παραγωγή  $P_B = 0,3$  α.μ., Τάση  $V_B = 1,0$  α.μ.
  - C: Φορτίο  $P_C = 0,3$  α.μ.,  $Q_C = 0,05$  α.μ.
  - D: Φορτίο  $P_D = 0,4$  α.μ.,  $Q_D = 0,1$  α.μ.

**Δεδομένα εισόδου**

Παρακάτω δίνονται δύο πίνακες για την εισαγωγή των δεδομένων του συστήματος.  
 Αλφάβητα των ζυγών:

- H 1η στήλη είναι για την αριθμηση ζυγών.
- H 2η για τον χαρακτηρισμό των ζυγών (0 για τους PQ, 2 για τους PV, 1 για τον ζυγό ταλάντωσης).
- H 3η για τις τιμές των τάσεων (α.μ.) κάθε ζυγού (εάν δεν δίνονται χρησιμοποιείται η τιμή  $V = 1$ ).
- H 4η για τις τιμές των γωνιών (σε μοίρες-degree) κάθε ζυγού (εάν δε δίνονται χρησιμοποιείται η τιμή  $\theta = 0$ ).
- H 5η τη ζήτηση αέργου ισχύος (MVar).
- H 6η την ζήτηση αέργου ισχύος (MW).
- H 7η την παραγόμενη άεργου ισχύ (MVar).
- H 8η την παραγόμενη άεργου ισχύ (MW).
- H 9η στήλη δηλώνει για τις γεννήτριες (ζυγούς) τα κλάσματα άεργου ισχύος (MVar).
- H 10η τα μέγιστα όρια αέργου ισχύος αντίστοιχα (MVar).
- H 11η στήλη χρησιμοποιείται για τα δεδομένα αντιστάθμισης αέργου ισχύος με πυκνωτές, δηλώνοντας την έγχυση αέργου ισχύος στον αντίστοιχο ζυγό (MVar).

```
In [1]: % Bus Bus Voltage Angle ---Load--- Generator----- Injected
% No code Mag. Degree MW Mvar MW Mvar Qin Qmax Mvar
busdata=[1 1 1.00 0.0 0 0 0 0 0 0 0 0
2 2 1.05 0.0 0 0 30 0 0 500 0
3 0 1.00 0.0 30 5 0 0 0 500 0
4 0 1.00 0.0 40 10 0 0 0 0 0];
```

Δεδομένα των γραμμών μεταφοράς και μετασχηματιστών:

- H 1η και 2η στήλη δηλώνουν τις συνδέσεις που υπάρχουν μεταξύ των ζυγών.
- H 3η στήλη αναφέρεται στην αντίσταση (α.μ.) κάθε γραμμής.
- H 4η στην αντίδραση (α.μ.) κάθε γραμμής.
- H 5η στο ήμισυ της συνολικής εγκάρσιας χωρητικότητας της (α.μ.), δηλαδή χρησιμοποιείται το μονοδύναμο μοντέλο Π.
- H 6η στήλη αναφέρεται σε γραμμή με μετασχηματιστές ισχύος που έχουν μεταβολή της σχέσης μετασχηματισμού (συνήθως θεωρούνται ονομαστικές ή γαμμα με  $l = 1$ ).

```
In [2]: % Bus bus R X Line code
% nl nr p.u. p.u. p.u. = 1 for lines
% > 1 or < 1 tr. tap at bus nl
linedata=[1 2 0.0 0.1 0.0 1
1 4 0.0 0.2 0.0 1
2 3 0.0 0.2 0.0 1
3 4 0.0 0.1 0.0 1];
```

**Έγχυση πραγματικής και αέργου ισχύος**

Καθορισμός της βάσης ισχύος σε MVA,  $S_{base} = 100$

(basemva = 100)

Σχήμα 4. 12 : Στιγμιότυπο από την επίλυση της ροής φορτίου της 1ης εργαστηριακής άσκησης στο CoCalc που δείχνει τον τίτλο, το σχήμα και τα δεδομένα των ζυγών και των γραμμών του συστήματος (1/11).

Γίνεται το ισοζύγιο ισχύος για κάθε ζυγό  $k$  ξεχωριστά και υπολογίζονται οι ανά μονάδα τιμές της έγχυσης ενεργού και αέργου ισχύος από τις σχέσεις:

$$P_{sp,k} = \frac{P_{G,k} - P_{D,k}}{S_{base}} \quad (4)$$

$$Q_{sp,k} = \frac{Q_{G,k} - Q_{D,k}}{S_{base}} \quad (5)$$

Οι μιγαδικές ισχύες εκφράζονται ως:

$$S = P_{sp,k} + jQ_{sp,k} \quad (6)$$

```
In [3]: basevna = 100;
Vm=0; delta=0; yload=0; deldat=0;
nbus = length(busdata(:,1));
kb=[]; Vm=[]; delta=[]; Va=[];
Pg = zeros(nbus,1); Pd = zeros(nbus,1); Qg = zeros(nbus,1); Qd = zeros(nbus,1);
Psp = zeros(nbus,1); Qsp = zeros(nbus,1); Qsh = zeros(nbus,1);
S = zeros(nbus,1);

for k=1:nbus
n=busdata(k,1); kb(n)=busdata(k,2); Vm(n)=busdata(k,3); delta(n)=busdata(k,4);
Pd(n)=busdata(k,5); Qd(n)=busdata(k,6); Pg(n)=busdata(k,7); Qg(n)=busdata(k,8);
Psp(n)=(Pg(n)-Pd(n))/basevna; % (4)
Qsp(n)=(Qg(n)-Qd(n))/basevna; % (5)
S(n) = Psp(n) + j*Qsp(n); % (6)
end

Psp
Qsp
S
```

Psp =

```
0.00000
0.30000
-0.30000
-0.40000
```

Qsp =

```
0.00000
0.00000
-0.05000
-0.10000
```

S =

```
0.00000 + 0.00000i
0.30000 + 0.00000i
-0.30000 - 0.05000i
-0.40000 - 0.10000i
```

Μετατροπή γωνιών από μοίρες σε ακτίνια:

$$(\text{rad}) = \frac{\pi}{180} \cdot \delta^\circ \quad (7)$$

```
In [4]: for n=1:nbus
delta(n) = pi/180*delta(n); % (7)
end
delta

delta =

0 0 0 0
```

#### Υπολογισμός του πίνακα αγωγιμοτήτων του συστήματος

- Αρχικά όλες οι συνέτες αντιστάσεις μετατρέπονται σε συνθετες αγωγιμοτητες από τη σχέση (9):

$$Z_{km} = R_{km} + jX_{km} \quad (8)$$

$$y_{km} = \frac{1}{Z_{km}} \quad (9)$$

- και στη συνέχεια υπολογίζονται τα διαγώνια και τα μη διαγώνια στοιχεία του πίνακα αγωγιμοτήτων από τις παρακάτω σχέσεις.

$$Y_{kk} = y_k + \sum_{n \in A(k)} (y_{skm} + y_{km}) \quad (10)$$

$$Y_{km} = -y_{km} \quad (11)$$

Ο πίνακας αγωγιμοτήτων έχει μέγεθος  $(n \times n)$  όπου  $n$  ο αριθμός των ζυγών του συστήματος που εξετάζουμε.

Στο παράδειγμα αυτό ο πίνακας αγωγιμοτήτων έχει μέγεθος  $(4 \times 4)$ .

$$Y_{bus} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} \end{bmatrix}$$

Σχήμα 4. 13 : Στιγμιότυπο από την επίλυση της ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης στο CoCalc με εξισώσεις και κώδικα για των υπολογισμό των εγχύσεων ενεργού και αέργου ισχύος σε κάθε ζυγό καθώς και τις εξισώσεις για τον υπολογισμό του πίνακα αγωγιμοτήτων (2/11).

```
In [15]: j=sqrt(-1);
nl = linedata(:,1); %line from bus
nr = linedata(:,2); %line to bus
R = linedata(:,3); %line resistance
X = linedata(:,4); %line reactance
Bc = j*linedata(:,5); % the admittance of branch to ground
a = linedata(:, 6); % the tap ratio of transformer
nbr=length(linedata(:,1)); %number of branch
nbus = max(max(nl), max(nr)); %number of bus
G=[];B=[];

%the impedance of line
Z = R + j*X
%the admittance of line
y= ones(nbr,1)./Z %(\delta)

for n = 1:nbr
Ybus=zeros(nbus,nbus); % initialize Ybus to zeros
% formation of the off diagonal elements
for k=1:nbr
Ybus(nl(k),nr(k))=Ybus(nl(k),nr(k))-y(k); %(\delta)
Ybus(nr(k),nl(k))=Ybus(nl(k),nr(k));
end
end
% formation of the diagonal elements
for n=1:nbus
for k=1:nbr
if nl(k)==n
Ybus(n,n) = Ybus(n,n)+y(k); %(\delta)
elseif nr(k)==n
Ybus(n,n) = Ybus(n,n)+y(k); %(\delta)
else, end
end
end
end
Ybus =
Z =
0.069000 + 0.166000i
0.069000 + 0.209000i
0.069000 + 0.209000i
0.069000 + 0.109000i

y =
0 - 10i
0 - 5i
0 - 5i
0 - 10i

Ybus =
0 - 15i 0 + 10i 0 + 0i 0 + 5i
0 + 10i 0 - 15i 0 + 5i 0 + 0i
0 + 0i 0 + 5i 0 - 15i 0 + 10i
0 + 5i 0 + 0i 0 + 10i 0 - 15i
```

#### Τύποι ζυγών ροής φορτίου

Τύπος ζυγού	Γνωστές μεταβλητές	Άγνωστες μεταβλητές
Ταλάντωση ή αναφοράς	$V = 1 \text{ a.u.}, \delta = 0^\circ$	$P_C, Q_C$
Ζυγός παραγωγής (PV)	$P_G, V$	$Q_G, \delta$
Ζυγός φορτίου (PQ)	$P_D, Q_D$	$V, \delta$

Στο παράδειγμα είναι:

- **Ζυγός A:** Ζυγός ταλάντωσης (slack bus). Θεωρούμε ότι η τάση του είναι σταθερή κατά μέτρο και γωνία  $V_1 = 1 \angle 0^\circ$  a.u..
- **Ζυγός B:** Ζυγός παραγωγής (PV), αφού είναι γνωστά η παραγόμενη ενεργός ισχύς  $P_{G_2} = 0$ , 3a.u. και το μέτρο της τάσης  $V_2 = 1$  a.u.
- **Ζυγός C:** Ζυγός φορτίου (PQ), αφού είναι γνωστές οι ισχύεις φορτίου  $P_{C_2} = 0$ , 3a.u.,  $Q_{C_2} = 0$ , 05a.u.
- **Ζυγός D:** Ζυγός φορτίου (PQ), αφού είναι γνωστές οι ισχύεις φορτίου  $P_{D_2} = 0$ , 4a.u.,  $Q_{D_2} = 0$ , 1a.u.

Στο παρακάτω κείμενο κώδικα η μεταβλητή  $n_{PV}$  εμφανίζει το πλήθος των ζυγών PV (σημειώθηκαν, ενώ η μεταβλητή  $n_{PQ}$  εμφανίζει το πλήθος των ζυγών PQ. Οι συναρτήσεις  $s_1$ ,  $p_v$  και  $p_q$  βρίσκουν το ζυγό ταλάντωσης και τους ζυγούς PV και PQ αντίστοιχα.

```
In [17]: nPV = 0;
nPQ = 0;

for k = 1:nbus
if kb(k)==2 % PV bus
nPV = nPV+1;end
if kb(k)==0
nPQ = nPQ+1;end % PQ bus
end
nPV
nPQ

s1 = find(kb == 1) % slack bus
pv = find(kb == 2) % PV Buses..
pq = find(kb == 0) % PQ Buses..

nPV = 1
nPQ = 2
s1 = 1
pv = 2
pq =

3 4
```

Σχήμα 4. 14 : Στιγμιότυπο από την επίλυση της ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης στο CoCalc με τον υπολογισμό του πίνακα αγωγιμοτήτων, τη θεωρία και τον κώδικα για τον χαρακτηρισμό των ζυγών του συστήματος (3/11).

**Δεδομένα και ζητούμενα**

**Το διάνυσμα κατάστασης**

Θεωρούμε ότι  $n$  είναι ο συνολικός αριθμός των ζυγίων του συστήματος και  $m$  ο αριθμός των ζυγίων φορτίου PQ. Γενικά για  $n$  ζυγούς έχουμε  $2n$  αγνώστους. Δεδομένου ότι η τάση στο ζυγό ταλάντωσης είναι γνωστή κατά μέτρο και γωνία, ενώ τα μέτρα των τάσεων είναι γνωστά στους ζυγούς παραγωγής, οι μεταβλητές που πρέπει να προσδιοριστούν για να περιγραφεί πλήρως η κατάσταση λειτουργίας του συστήματος είναι οι ακόλουθες:

- οι φυσικές γωνίες τάσεων όλων των ζυγών εκτός από το ζυγό ταλάντωσης
- τα μέτρα τάσεων όλων των ζυγών φορτίου.

Οι μεταβλητές αυτές αποτελούν το διάνυσμα κατάστασης  $X = \begin{bmatrix} \delta_2 \\ \dots \\ \delta_n \\ V_{n-m+1} \\ \dots \\ V_n \end{bmatrix}$  με διάσταση  $n - 1 + m$ .

Άρα για την εύρεση των αγνώστων χρειάζονται  $n - 1 + m$  ανεξάρτητες εξισώσεις.

- $n - 1$  εξισώσεις πραγματικής ισχύος (για ζυγούς PV και PQ):

$$\Delta P_k^{(n)} = P_{sp,k} - V_k^{(n)} \cdot \sum_{m \in A(k)} V_m^{(n)} \cdot [G_{km} \cdot \cos(\delta_k^{(n)} - \delta_m^{(n)}) + B_{km} \cdot \sin(\delta_k^{(n)} - \delta_m^{(n)})] \quad (13)$$

- $m$  εξισώσεις αέργου ισχύος (για ζυγούς PQ):

$$\Delta Q_k^{(n)} = Q_{sp,k} - V_k^{(n)} \cdot \sum_{m \in A(k)} V_m^{(n)} \cdot [G_{km} \cdot \sin(\delta_k^{(n)} - \delta_m^{(n)}) - B_{km} \cdot \cos(\delta_k^{(n)} - \delta_m^{(n)})] \quad (14)$$

Στο παράδειγμα αυτό γνωρίζουμε την τάση του ζυγού A κατά μέτρο και γωνία  $V_A = 1 \angle 0^\circ$  ο.μ. καθώς και το μέτρο της τάσης του ζυγού B,  $V_B = 1$  ο.μ. Επομένως οι άγνωστοι του προβλήματος είναι  $\delta$ , οι γωνίες  $\delta_B, \delta_C, \delta_D$  και η τάσεις  $V_C, V_D$ .

Το διάνυσμα κατάστασης λοιπόν έχει τη μορφή  $X = \begin{bmatrix} \delta_B \\ \delta_C \\ \delta_D \\ V_C \\ V_D \end{bmatrix}$  και για να υπολογιστεί θα χρειαστούν τρεις εξισώσεις πραγματικής ( $\Delta P_A, \Delta P_B, \Delta P_C$ ) και δύο εξισώσεις αέργου ισχύος ( $\Delta Q_C, \Delta Q_D$ ).

**Επαναληπτική διαδικασία**

- Πριν ξεκινήσουν οι επαναλήψεις της μεθόδου Newton-Raphson θεωρείται μια ενιαία τιμή εκκίνησης για τις τάσεις όλων των ζυγίων του δικτύου, που δεν έχουν καθορισμένη τιμή, στα  $V = 1 \angle 0^\circ$  ο.μ.
- Χρησιμοποιώντας αυτή την αρχική εκτίμηση υπολογίζεται, με τη βοήθεια των εξισώσεων ροής φορτίου μια βελτιωμένη δεύτερη εκτίμηση η οποία χρησιμοποιείται για να βρεθεί μία τρίτη κ.ο.κ.
- Οι επαναλήψεις συνεχίζονται μέχρι οι νέες τιμές των μεταβλητών του συστήματος να διαφέρουν από τις αρχικές πολύ λίγο, δηλαδή όταν ικανοποιούνται τα κριτήρια σύγκλισης.

$$\begin{cases} |\Delta P_k^{(n)}| \leq \varepsilon \\ |\Delta Q_k^{(n)}| \leq \varepsilon \end{cases} \quad (15)$$

Στο πρόγραμμα αυτό το κριτήριο είναι:

$$|M^{(n)}| < \varepsilon \quad (16)$$

όπου  $M$  το διάνυσμα διαφορών:

$$M = \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \quad (17)$$

και  $\varepsilon$  η ακρίβεια των τελικών μεγεθών

$$\varepsilon = 0.001$$

Η επαναληπτική διαδικασία στο πρόγραμμα γίνεται με ένα while loop, το οποίο στην εισοδό του ελέγχει το κριτήριο σύγκλισης της σχέσης (16). Ο στόχος, για να ελεγχίσουμε αναλυτικά την 1η επανάληψη, έχουμε χωρίσει τον κώδικα σε τμήματα και έχουμε βάλει τον έλεγχο σε σχόλιο, καθώς για να τρέξει επαναληπτικά πρέπει όλος ο κώδικας που περιέχεται στο while loop να βρίσκεται μέσα στο ίδιο cell (cell). Στο παρακάτω κομμάτι κώδικα υπολογίζονται οι επαναληπτικές εξισώσεις ροής φορτίου σε όλους τους ζυγούς του ΣΗΕ. Από αυτές χρειάζμαστε τις τρεις εξισώσεις πραγματικής ισχύος ( $\Delta P_A, \Delta P_B, \Delta P_C$ ), που αποθηκεύονται στο διάνυσμα dP και τις δύο εξισώσεις αέργου ισχύος ( $\Delta Q_C, \Delta Q_D$ ), που αποθηκεύονται στο διάνυσμα dQ. Τα διανύσματα αυτά ενώνονται για να δημιουργηθεί το διάνυσμα διαφορών M που έχει ίδιες διαστάσεις με το X.

```
In [18]: Tol = 1;
Iter = 0;
%while (Tol >= 0.001) % Iteration starting.. (16)
Iter = Iter + 1;
P = zeros(nbus,1); Q = zeros(nbus,1); % Calculate P and Q

for k = 1:nbus
    for m = 1:nbus
        P(k) = P(k) + Vm(k)*Vm(m)*(G(k,m)*cos(delta(k)-delta(m)) + B(k,m)*sin(delta(k)-delta(m))); % (13)
        Q(k) = Q(k) + Vm(k)*Vm(m)*(G(k,m)*sin(delta(k)-delta(m)) - B(k,m)*cos(delta(k)-delta(m))); % (14)
    end
end

dPa = Psp-P; % (13)
dQa = Qsp-Q; % (14)

k = 1;
dQ = zeros(nPQ,1);
for i = 1:nbus
    if kb(i) == 0
        dQ(k,1) = dQa(i);
        k = k + 1;
    end
end
```

Σχήμα 4. 15: Στιγμιότυπο από την επίλυση της ροής φορτίου της 1ης εργαστηριακής άσκησης στο CoCalc όπου εξηγείται η διαδικασία της δημιουργίας του διανύσματος διαφορών μέσω των επαναληπτικών εξισώσεων ροής φορτίου και δίνεται ο αντίστοιχος κώδικας (4/11).



```

k = 1;
dQ = zeros(nPO,1);
for i = 1:nbus
    if kb(i) == 0
        dQ(k,1) = dQa(i);
        k = k+1;
    end
end
dP = dPa(2:nbus)
dQ
% Mismatch Vector
M = [dP; dQ] % (17)
dP =
    0.36000
   -0.36000
   -0.46000
dQ =
    0.26000
   -0.16000
M =
    0.30000
   -0.30000
   -0.40000
    0.20000
   -0.10000

```

### 3. Υπολογισμός Ιακωβιανής μήτρας

Η Ιακωβιανή μήτρα έχει τη μορφή

$$Jac = \begin{bmatrix} H & N \\ J & L \end{bmatrix} \quad (18)$$

- Ο υποπίνακας  $H$  έχει διαστάσεις  $(n-1) \times (n-1)$
- Ο υποπίνακας  $N$  έχει διαστάσεις  $(n-1) \times m$
- Ο υποπίνακας  $J$  έχει διαστάσεις  $m \times (n-1)$
- Ο υποπίνακας  $L$  έχει διαστάσεις  $m \times m$

Στο παράδειγμα αυτό ο υποπίνακας  $H$  έχει διαστάσεις  $(3 \times 3)$ , ο  $N \rightarrow (3 \times 2)$ , ο  $J \rightarrow (2 \times 3)$  και ο  $L \rightarrow (2 \times 2)$ .  
Επομένως η Ιακωβιανή μήτρα έχει τη μορφή:

$$Jac = \begin{bmatrix} H_{22} & H_{23} & H_{24} & N_{23} & N_{24} \\ H_{32} & H_{33} & H_{34} & N_{33} & N_{34} \\ H_{42} & H_{43} & H_{44} & N_{43} & N_{44} \\ J_{32} & J_{33} & J_{34} & L_{33} & L_{34} \\ J_{42} & J_{43} & J_{44} & L_{43} & L_{44} \end{bmatrix}$$

```

In [11]: clear H N J L
H = zeros(nbus-1, nbus-1);
N = zeros(nbus-1, nPO);
J = zeros(nPO, nbus-1);
L = zeros(nPO, nPO);

```

Το διαγώνιο και μη διαγώνιο στοιχεία του υποπίνακα  $H$  δίνονται από τις σχέσεις:

$$H_{kk} = \frac{\partial \Delta P_k}{\partial \delta_k} = V_k^2 \cdot B_{kk} + Q_k \quad (19)$$

$$\text{άσσυ} \quad Q_k = V_k \cdot \sum_{k \neq m} V_m \cdot [G_{km} \cdot \sin(\delta_k - \delta_m) - B_{km} \cdot \cos(\delta_k - \delta_m)] \quad (20)$$

$$H_{km} = \frac{\partial \Delta P_k}{\partial \delta_m} = -V_k \cdot V_m \cdot [G_{km} \cdot \sin(\delta_k - \delta_m) - B_{km} \cdot \cos(\delta_k - \delta_m)], k \neq m \quad (21)$$

```

In [12]: for i = 1:(nbus-1)
k = i+1;
for j = 1:(nbus-1)
m = j+1;
if k == m
for m = 1:nbus
H(i,j) = H(i,j) + Vm(k)*Vm(m)*(G(k,m)*sin(delta(k)-delta(m)) - B(k,m)*cos(delta(k)-delta(m))); % (2)
end
end
H(i,j) = (Vm(k)^2)*B(k,k) + H(i,j); % (19)
else
H(i,j) = -Vm(k)*Vm(m)*(G(k,m)*sin(delta(k)-delta(m)) - B(k,m)*cos(delta(k)-delta(m))); % (21)
end
end
end
H
H =
   -15.75000    5.25000    -0.00000
    5.25000   -15.25000    10.00000
    0.00000    10.00000   -15.00000

```

Σχήμα 4. 16 Στιγμιότυπο από την επίλυση της ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης στο CoCalc που δείχνει το διάνυσμα διαφορών, τις σχέσεις και τον κώδικα για τον υπολογισμό του υποπίνακα  $H$  της Ιακωβιανής μήτρας (5/11).

Τα διαγώνια και μη διαγώνια στοιχεία του υποπίνακα  $N$  δίνονται από τις σχέσεις:

$$N_{kk} = \frac{\partial \Delta P_k}{\partial V_k} = -V_k \cdot G_{kk} - \frac{P_k}{V_k} \quad (22)$$

$$\text{όπου } \frac{P_k}{V_k} = \sum_{k \neq m} V_m \cdot [G_{km} \cdot \cos(\delta_k - \delta_m) + B_{km} \cdot \sin(\delta_k - \delta_m)] \quad (23)$$

$$N_{km} = \frac{\partial \Delta P_k}{\partial V_m} = -V_k \cdot [G_{km} \cdot \cos(\delta_k - \delta_m) + B_{km} \cdot \sin(\delta_k - \delta_m)], k \neq m \quad (24)$$

```
In [13]: for i = 1:(nbus-1)
          k = i+1;
          for j = 1:nPQ
              m = pq(j);
              if k == m
                  for m = 1:nbus
                      N(i,j) = N(i,j) + Vm(m)*(G(k,m)*cos(delta(k)-delta(m)) + B(k,m)*sin(delta(k)-delta(m))); % (23)
                  end
                  N(i,j) = Vm(k)*G(k,k) - N(i,j); % (22)
              else
                  N(i,j) = -Vm(k)*(G(k,m)*cos(delta(k)-delta(m)) + B(k,m)*sin(delta(k)-delta(m))); % (24)
              end
          end
      end
      N
      N =
      -0 -0
      -0 -0
      -0 -0
```

Τα διαγώνια και μη διαγώνια στοιχεία του υποπίνακα  $J$  δίνονται από τις σχέσεις:

$$J_{kk} = \frac{\partial \Delta Q_k}{\partial \delta_k} = V_k^2 \cdot G_{kk} - T_k \quad (25)$$

$$\text{όπου } T_k = V_k \cdot \sum_{k \neq m} V_m \cdot [G_{km} \cdot \cos(\delta_k - \delta_m) + B_{km} \cdot \sin(\delta_k - \delta_m)] \quad (26)$$

$$J_{km} = \frac{\partial \Delta Q_k}{\partial \delta_m} = V_k \cdot V_m \cdot [G_{km} \cdot \cos(\delta_k - \delta_m) + B_{km} \cdot \sin(\delta_k - \delta_m)], k \neq m \quad (27)$$

```
In [14]: for i = 1:nPQ
          k = pq(i);
          for j = 1:(nbus-1)
              m = i+1;
              if k == m
                  for m = 1:nbus
                      J(i,j) = J(i,j) + Vm(k)*Vm(m)*(G(k,m)*cos(delta(k)-delta(m)) + B(k,m)*sin(delta(k)-delta(m))); %26)
                  end
                  J(i,j) = (Vm(k)^2)*G(k,k) - J(i,j); % (25)
              else
                  J(i,j) = Vm(k)* Vm(m)*(G(k,m)*cos(delta(k)-delta(m)) + B(k,m)*sin(delta(k)-delta(m))); % (27)
              end
          end
      end
      J
      J =
      0 0 0
      0 0 0
```

Τα διαγώνια και μη διαγώνια στοιχεία του υποπίνακα  $L$  δίνονται από τις σχέσεις:

$$L_{kk} = \frac{\partial \Delta Q_k}{\partial V_m} = V_k \cdot B_{kk} - \frac{Q_k}{V_k} \quad (28)$$

$$\text{όπου } \frac{Q_k}{V_k} = \sum_{k \neq m} V_m \cdot [G_{km} \cdot \sin(\delta_k - \delta_m) - B_{km} \cdot \cos(\delta_k - \delta_m)] \quad (29)$$

$$L_{km} = \frac{\partial \Delta Q_k}{\partial V_m} = -V_k \cdot [G_{km} \cdot \sin(\delta_k - \delta_m) - B_{km} \cdot \cos(\delta_k - \delta_m)], k \neq m \quad (30)$$

```
In [15]: for i = 1:nPQ
          k = pq(i);
          for j = 1:nPQ
              m = pq(j);
              if k == m
                  for m = 1:nbus
                      L(i,j) = L(i,j) + Vm(m)*(G(k,m)*sin(delta(k)-delta(m)) - B(k,m)*cos(delta(k)-delta(m))); % (29)
                  end
                  L(i,j) = Vm(k)*B(k,k) - L(i,j); % (28)
              else
                  L(i,j) = -Vm(k)*(G(k,m)*sin(delta(k)-delta(m)) - B(k,m)*cos(delta(k)-delta(m)));
              end
          end
      end
      L
      L =
      -14.750 16.000
      10.000 -15.000
```

Σχήμα 4. 17 : Στιγμιότυπο από την επίλυση της ροής φορτίου της 1ης εργαστηριακής άσκησης στο CoCalc που δείχνει τις σχέσεις και τον κώδικα για τον υπολογισμό των υποπινάκων N, J, L της Ιακωβιανής μήτρας (6/11).

Επομένως η Ιακωβιανή μήτρα για την 1η επανάληψη θα είναι:

```
In [17]: % Jacobian matrix
Jac = [H N; J L] % (18)
Jac =
    15.75000   -5.25000    0.00000    0.00000    0.00000
    -5.25000   15.25000   -10.00000    0.00000    0.00000
     0.00000   -10.00000    15.00000    0.00000    0.00000
    -0.00000    0.00000    0.00000   14.75000   -10.00000
    -0.00000    0.00000    0.00000   -10.00000    15.00000
```

#### 4. Υπολογισμός των νέων τάσεων και γωνιών του συστήματος

Από τη γενική εξίσωση Newton-Raphson υπολογίζεται το διάνυσμα των αγνώστων:

$$X = J^{-1}\Delta I^{(n)} \Rightarrow \begin{bmatrix} \Delta\delta \\ \Delta V \end{bmatrix} = \begin{bmatrix} H & N \\ J & L \end{bmatrix}^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \quad (31)$$

```
In [18]: % Correction Vector
X = inv(Jac)*M % (31)
dTh = X(1:nbus-1) % Change in Voltage Angle..
dV = X(nbus:end) % Change in Voltage Magnitude..
X =
    -0.0037166
    -0.0602927
    -0.0721951
     0.0164948
     0.0043299
dTh =
    -0.0037166
    -0.0602927
    -0.0721951
dV =
     0.0164948
     0.0043299
```

Οι νέες τιμές υπολογίζονται χρησιμοποιώντας τις τιμές της προηγούμενης επανάληψης:

$$\begin{bmatrix} \delta \\ V \end{bmatrix}^{(n+1)} = \begin{bmatrix} \Delta\delta \\ \Delta V \end{bmatrix} + \begin{bmatrix} \delta \\ V \end{bmatrix}^{(n)} \quad (24)$$

$$\begin{bmatrix} \delta \\ V \end{bmatrix}^{(n+1)} = \begin{bmatrix} \Delta\delta \\ \Delta V \end{bmatrix} + \begin{bmatrix} \delta \\ V \end{bmatrix}^{(n)} \quad (25)$$

Δηλαδή:

$$\begin{bmatrix} 0 \\ \delta_B \\ \delta_C \\ \delta_D \end{bmatrix}^{(1)} = \begin{bmatrix} \Delta\delta \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

και

$$\begin{bmatrix} 1 \\ 1.05 \\ V_C \\ V_D \end{bmatrix}^{(1)} = \begin{bmatrix} \Delta V \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

```
In [19]: % Updating State Vectors..
% Voltage Angle..
delta(2:nbus) = dth + delta(2:nbus) % (32)
% Voltage Magnitude..
k = 1;
for i = 2:nbus
    if kb(i) == 0
        Vn(i) = dV(k) + Vm(i); % (33)
        k = k+1;
    end
end
Vn =
delta =
    0.000000   -0.063717   -0.069293   -0.072195
Vn =
    1.0000    1.0500    1.0165    1.0043
```

Μετατροπή γωνιών από ακτίνια σε μοίρες:

$$\delta^\circ = \frac{180^\circ}{\pi} \cdot (\text{rad}) \quad (34)$$

```
In [20]: deltad=180/pi*delta % (34)
deltad =
    0.00000   -0.21295   -3.91288   -4.13648
```

Σχήμα 4. 18 : Στιγμιότυπο από την επίλυση της ροής φορτίου της 1ης εργαστηριακής άσκησης στο CoCalc που δείχνει την τελική μορφή της Ιακωβιανής μήτρας, τον υπολογισμό του διανύσματος αγνώστων X, τις εξισώσεις και τον κώδικα για τον υπολογισμό των νέων τάσεων και γωνιών του συστήματος (7/11).

**5. Υπολογισμός ισχύων στους ζυγούς**

Οι ισχύς που χρειάζεται να προσδιοριστούν αν πρόκειται:

- Για ζυγό ταλάντωσης, είναι η ενεργός και η άεργος ισχύς του,  $P_G, Q_G$
  - Για ζυγό PV, είναι η άεργος ισχύς του,  $Q_G$
- Οι ισχύς αυτές δίνονται από τις σχέσεις:

$$P_G = P_k \cdot S_{base} + P_D \quad (35)$$

$$Q_G = Q_k \cdot S_{base} + Q_D \quad (36)$$

Και οι μιγαδικές ισχύς εκφράζονται ως:

$$S_k = P_k + jQ_k \quad (37)$$

```
In [21]: k=0;
for n = 1:nbus
    if kb(n) == 1
        k=k+1;
        S(n) = P(n)+j*Q(n); % (37)
        Pg(n) = P(n)*baseva + Pd(n); % (35)
        Qg(n) = Q(n)*baseva + Qd(n); % (36)
    elseif kb(n) == 2
        k=k+1;
        S(n)=P(n)+j*Q(n); % (37)
        Qg(n) = Q(n)*baseva + Qd(n); % (36)
    end
    yload(n) = (Pd(n) - j*Qd(n))/(baseva*Vm(n)^2);
end
Pg
Qg
```

Pg =

```
0
30
0
0
```

Qg =

```
-50.0000
78.7500
0.0000
0.0000
```

Στο σημείο αυτό ολοκληρώνεται η 1η επανάληψη της μεθόδου.

Η μεταβλητή Tol παίρνει την τιμή της μέγιστης απόλυτης τιμής του διανύσματος διαφορών M ώστε να ελεγχθεί το κριτήριο σύγκλισης στην αρχή του while loop.

Στο παρακάτω κελί, το αρχείο Busout υπολογίζει και εμφανίζει τα τελικά αποτελέσματα της ροής φορτίου και το αρχείο Lineflow υπολογίζει και εμφανίζει τις ροές ισχύος στις γραμμές καθώς και τις αντίστοιχες απώλειες ισχύος (όλα τα μεγέθη σε MW και MVar).

```
In [22]: Busout
Lineflow
% Tolerance...
% Tol = max(abs(M));
%end
```

Power Flow Solution by Newton-Raphson Method  
Maximum Power Mismatch = 1  
No. of Iterations = 1

Bus No.	Voltage Mag.	Angle Degree	----Load----		---Generation---		Injected Mvar
			MW	Mvar	MW	Mvar	
1	1.000	0.00	0.00	0.00	0.00	-50.00	0.00
2	1.050	-0.21	0.00	0.00	30.00	78.75	0.00
3	1.016	-3.91	30.00	5.00	0.00	0.00	0.00
4	1.004	-4.14	40.00	10.00	0.00	0.00	0.00
Total			70.00	15.00	30.00	28.75	0.00

Line Flow and Losses

Line from to	Power at bus & line flow			--Line loss--		Transformer tap
	MW	Mvar	MVA	MW	Mvar	
1	0.000	-50.000	100.000			
	3.902	-49.993	50.145	0.000	2.515	
	36.222	-0.857	36.233	0.000	2.626	
2	0.000	78.750	157.500			
	-3.902	52.507	52.652	0.000	2.515	
	34.438	18.793	39.189	0.000	2.786	
3	0.000	-25.000	39.414			
	-34.438	-15.917	37.930	0.000	2.786	
	3.984	12.373	12.959	0.000	0.164	
4	0.000	0.000	41.231			
	-36.222	3.482	36.389	0.000	2.626	
	-3.984	-12.210	12.843	0.000	0.164	
Total loss				0.000	8.090	

Το παρακάτω κελί περιέχει όλο τον κώδικα της ροής φορτίου με τη μέθοδο Newton-Raphson που περιγράψαμε αναλυτικά παραπάνω, το οποίο τρέχει και εμφανίζει τα αποτελέσματα για τις τρεις πρώτες επαναλήψεις όπου και συγκλίνει.

Σχήμα 4. 19 : Στιγμιότυπο από την επίλυση της ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης στο CoCalc που υπολογίζει τις νέες ισχύς του συστήματος και εμφανίζει τα αποτελέσματα ροής φορτίου για την πρώτη επανάληψη (8/11).

```

end

```

Power Flow Solution by Newton-Raphson Method  
Maximum Power Mismatch = 1  
No. of Iterations = 1

Bus No.	Voltage Mag.	Angle Degree	-----Load----- MW Mvar		---Generation--- MW Mvar		Injected Mvar
1	1.000	0.00	0.00	0.00	0.00	-50.00	0.00
2	1.050	-0.21	0.00	0.00	30.00	78.75	0.00
3	1.014	-3.91	30.00	5.00	0.00	0.00	0.00
4	1.004	-4.14	40.00	10.00	0.00	0.00	0.00
Total			70.00	15.00	30.00	28.75	0.00

Line Flow and Losses

--Line-- from to	Power at bus & line flow MW Mvar MVA			--Line loss-- MW Mvar	Transformer tap
1	0.000	-50.000	100.000		
2	3.902	-49.993	50.145	0.000	2.515
4	36.222	-0.857	36.233	0.000	2.626
2	0.000	78.750	157.500		
1	-3.902	52.567	52.652	0.000	2.515
3	34.438	18.703	39.189	0.000	2.786
3	0.000	-25.000	30.414		
2	-34.438	-15.917	37.938	0.000	2.786
4	3.984	12.373	12.999	0.000	0.164
4	0.000	0.000	41.231		
1	-36.222	3.482	36.389	0.000	2.626
3	-3.984	-12.219	12.843	0.000	0.164
Total loss				0.000	8.090

Power Flow Solution by Newton-Raphson Method  
Maximum Power Mismatch = 0.4  
No. of Iterations = 2

Bus No.	Voltage Mag.	Angle Degree	-----Load----- MW Mvar		---Generation--- MW Mvar		Injected Mvar
1	1.000	0.00	0.00	0.00	40.12	-50.85	0.00
2	1.050	-0.22	0.00	0.00	30.00	71.21	0.00
3	1.014	-3.89	30.00	5.00	0.00	0.00	0.00
4	1.002	-4.12	40.00	10.00	0.00	0.00	0.00
Total			70.00	15.00	70.12	20.36	0.00

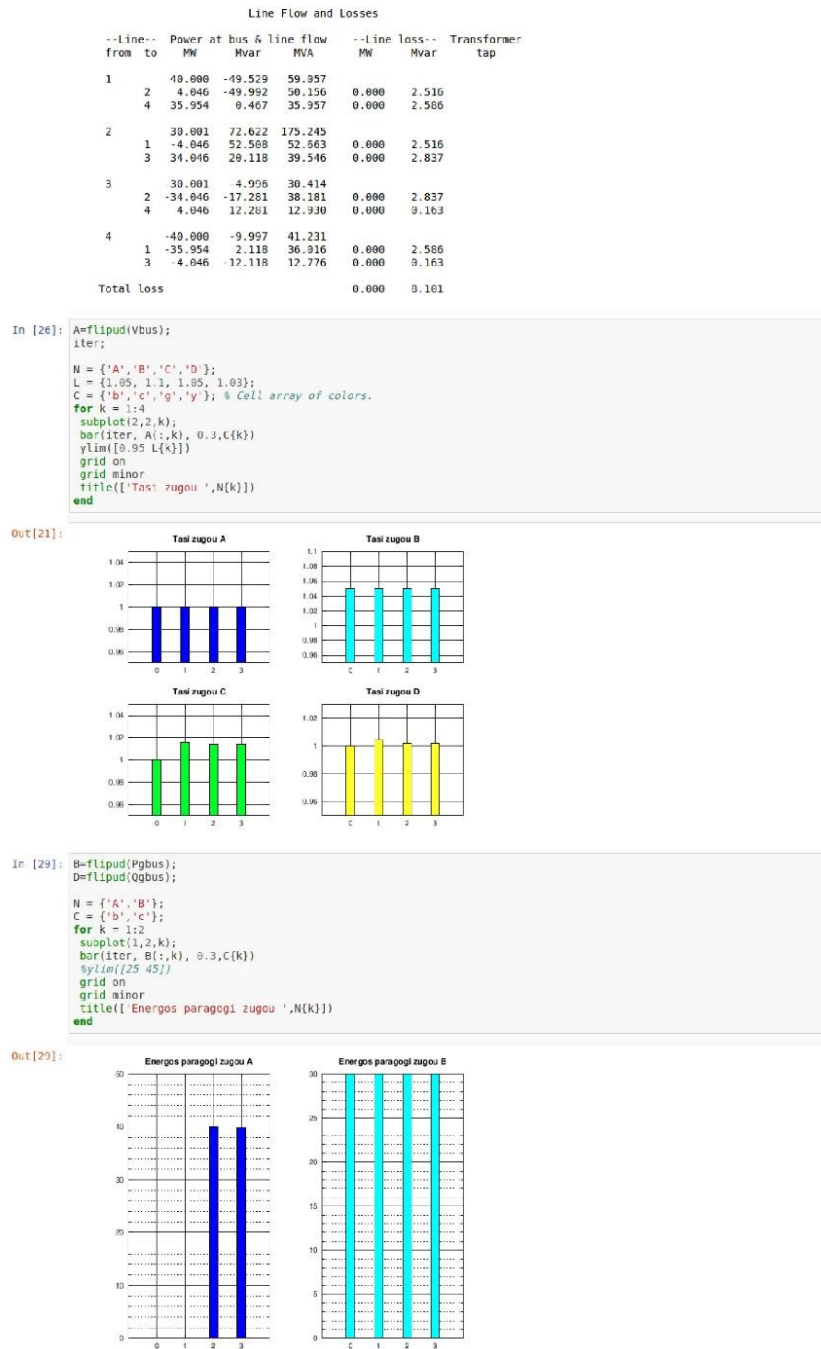
Line Flow and Losses

--Line-- from to	Power at bus & line flow MW Mvar MVA			--Line loss-- MW Mvar	Transformer tap
1	40.125	-50.850	61.574		
2	4.046	-49.992	50.156	0.000	2.516
4	35.954	0.404	35.957	0.000	2.586
2	30.535	71.210	172.955		
1	-4.046	52.568	52.663	0.000	2.516
3	34.047	20.114	39.544	-0.000	2.837
3	-30.454	-3.543	30.414		
2	-34.047	-17.277	38.180	-0.000	2.837
4	4.046	12.261	12.931	0.000	0.163
4	-40.206	-0.727	41.231		
1	-35.954	2.122	36.017	0.000	2.586
3	-4.046	-12.119	12.776	0.000	0.163
Total loss				-0.000	8.161

Power Flow Solution by Newton-Raphson Method  
Maximum Power Mismatch = 0.0145679  
No. of Iterations = 3

Bus No.	Voltage Mag.	Angle Degree	-----Load----- MW Mvar		---Generation--- MW Mvar		Injected Mvar
1	1.000	0.00	0.00	0.00	40.00	-49.53	0.00
2	1.050	0.22	0.00	0.00	30.00	72.62	0.00
3	1.014	-3.89	30.00	5.00	0.00	0.00	0.00
4	1.002	-4.12	40.00	10.00	0.00	0.00	0.00
Total			70.00	15.00	70.00	23.09	0.00

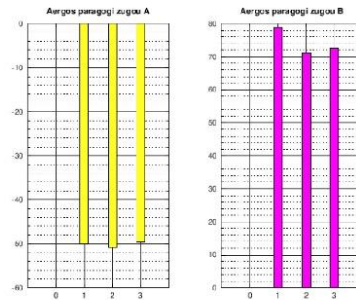
Σχήμα 4. 20 : Στιγμιότυπο από την επίλυση της ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης στο CoCalc όπου εμφανίζονται τα αποτελέσματα της ροής φορτίου για τρεις επαναλήψεις όπου ο κώδικας συγκλίνει (9/11).



Σχήμα 4. 21 : Στιγμιότυπο από την επίλυση της ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης στο CoCalc όπου εμφανίζεται η γραφική αναπαράσταση των τιμών των τάσεων των ζυγών για κάθε επανάληψη καθώς και η ενεργός παραγωγή των ζυγών αναφοράς και PV (10/11).

```
In [27]: N = {'A', 'B'};
C = {'y', 'm'};
for k = 1:2
subplot(1,2,k);
bar(iter, D(:,k), 0.3, C(k))
grid on
grid minor
title(['Αεργος παραγογι ζυγου ', N(k)])
end
```

Out[27]:



Σχήμα 4. 22 : Στιγμιότυπο από την επίλυση της ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης στο CoCalc όπου εμφανίζεται η γραφική αναπαράσταση των τιμών της αέργου παραγωγής των PQ ζυγών για κάθε επανάληψη (11/11).

#### 4.2.4 Εργαστηριακή άσκηση σε γλώσσα Python στο cloud (Colab)

Η τρίτη υλοποίηση που επιχειρήσαμε, είναι η κατασκευή του κώδικα ροής φορτίου σε γλώσσα Python. Η επιλογή αυτή έγινε καθώς η Python είναι μια σχετικά εύκολη γλώσσα προγραμματισμού, ιδιαίτερα δημοφιλής, αφού υποστηρίζεται από πολυάριθμα εργαλεία ελεύθερου λογισμικού και ανοικτούς πόρους στο διαδίκτυο. Αναμφίβολα από τεχνικής άποψης το MATLAB υπερέχει έναντι της Python, σε κάποιους συγκεκριμένους τομείς, ωστόσο η Python είναι μια συνεχώς εξελισσόμενη γλώσσα προγραμματισμού που αναμένεται να παρέχει ολοένα και περισσότερες λύσεις και γι' αυτό καθίσταται ιδιαίτερα χρήσιμη για τους μελλοντικούς μηχανικούς.

Για την παρουσίαση του υλικού χρησιμοποιούμε την cloud πλατφόρμα Colab, η οποία δεν χρειάζεται καμία εγκατάσταση για τη χρήση της, παρά μόνο τη δημιουργία ενός λογαριασμού. Ωστόσο θα μπορούσε να χρησιμοποιηθεί η τοπική εγκατάσταση του Jupyter ή κάποια άλλη cloud εναλλακτική, αφού η Python είναι συμβατή με τις περισσότερες πλατφόρμες.

Ο κώδικάς μας για την εφαρμογή αυτή ακολουθεί παρόμοια δομή με τα προηγούμενα παραδείγματα. Για την κατασκευή του βασιστήκαμε σε αντίστοιχο υλικό που διατίθεται online στο GitHub [81], εντούτοις χρειάστηκαν πολλές διορθώσεις ώστε να προκύψουν τα σωστά αποτελέσματα για τις τάσεις, τις γωνίες και τις ισχύεις των ζυγών, τα οποία συμβαδίζουν με αυτά του κώδικα MATLAB. Στην περίπτωση αυτή δεν συμπεριλήφθηκαν οι υπολογισμοί των ροών ισχύος και των απωλειών των γραμμών.

Όπως και στην προηγούμενη εφαρμογή απεικονίσαμε γραφικά τα αποτελέσματα (Σχήμα 4.30).



Newton-Raphson method in python lab.ipynb

https://colab.research.google.com/drive/1UV2D9vxcUjjiG5-3D35rroVTVOKFCF

Newton-Raphson method in python lab.ipynb

File Edit View Insert Runtime Tools Help All changes saved

RAM 1.5 GB Disk 1.5 GB Editing

Ανάλυση Συστημάτων Ηλεκτρικής Ενέργειας - Μόνιμη Κατάσταση Λειτουργίας

ΡΟΗ ΦΟΡΤΙΟΥ

Σχήμα 1: Σύστημα τεσσάρων ζυγίων

Για το σύστημα ηλεκτρικής ενέργειας με την τοπολογία του παραπάνω σχήματος δίνονται τα ακόλουθα δεδομένα:

- Επαγωγικές αντιδράσεις των γραμμών μεταφοράς:  $Z_{AB(1)} = Z_{AB(2)} = j0,2$  α.μ.,  $Z_{CD} = j0,1$  α.μ.,  $Z_{DA} = Z_{BC} = j0,2$  α.μ.
- Στοιχεία ζυγίων:
  - α. Τύση  $V_A = 1,0$  α.μ.
  - β. Παραγωγή  $P_B = 0,3$  α.μ., Τύση  $V_B = 1,0$  α.μ.
  - γ. Φορτίο  $P_C = 0,3$  α.μ.,  $Q_B = 0,05$  α.μ.
  - δ. Φορτίο  $P_D = 0,4$  α.μ.,  $Q_D = 0,1$  α.μ.

Εισαγωγή των βιβλιοθηκών της Python που θα χρησιμοποιηθούν:

```
[26] import numpy as np
import math
from math import cos, sin
import matplotlib.pyplot as plt
%matplotlib inline
```

Δεδομένα εισόδου

Δεδομένα των ζυγίων:

```
[27] def define_bus_data():
    bdataname = ['MUM', 'TYPE', 'VOLT', 'DEGREE', 'PLOAD', 'QLOAD', 'PGER', 'QGER', 'QMIN', 'QMAX', 'IN3VAR']
    busdata = np.array([[1, 1, 1.00, 0.0, 0, 0, 0, 0, 0, 0, 0, 0],
                        [2, 2, 1.05, 0.0, 0, 0, 30, 0, 0, 0, 200, 0],
                        [3, 0, 1.00, 0.0, 30, 5, 0, 0, 0, 0, 300, 0],
                        [4, 0, 1.00, 0.0, 40, 10, 0, 0, 0, 0, 0, 0]])
    num_bus = busdata.shape[0]
    return busdata, num_bus

busdata, num_bus = define_bus_data()
```

Δεδομένα των γραμμών μεταφοράς και μετασχηματιστών:

```
[28] def define_line_data():
    ldataname = ['NL', 'NR', 'R', 'X', '1/2Bc', 'TAP']
    linedata = np.array([[1, 2, 0.0, 0.1, 0.0, 11],
                        [1, 4, 0.0, 0.2, 0.0, 11],
                        [2, 3, 0.0, 0.2, 0.0, 11],
                        [3, 4, 0.0, 0.1, 0.0, 11]])
    num_lin = linedata.shape[0]
    return linedata, num_lin

linedata, num_lin = define_line_data()
```

Στο παρακάτω τμήμα κώδικα γίνεται ανάθεση των δεδομένων των ζυγίων στις αντίστοιχες μεταβλητές. Οι γωνίες μετατρέπονται από μοίρες σε ακτίνια από τη σχέση:

$$(\text{rad}) = \frac{\pi}{180} \cdot \delta^\circ \quad (1)$$

Γίνεται το ισοζύγιο ισχύος για κάθε ζυγό  $k$  ξεχωριστά και υπολογίζονται οι ανά μονάδα τιμές της έγχυσης ενεργού και αεργού ισχύος από τις σχέσεις:

$$P_{s,k} = \frac{P_{G,k} - P_{D,k}}{S_{\text{base}}} \quad (2)$$

$$Q_{s,k} = \frac{Q_{G,k} - Q_{D,k}}{S_{\text{base}}} \quad (3)$$

Σχήμα 4. 23 : Κώδικας επίλυσης ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης σε Python στο Colab που δείχνει τον τίτλο, το σχήμα, τα δεδομένα των ζυγίων και των γραμμών του συστήματος και εξισώσεις για των υπολογισμό των εγχύσεων ενεργού και αέργου ισχύος σε κάθε ζυγό (1/9).

```

[20] bus_type = [bus1] for bus in busdata]
Vn = [bus[2] for bus in busdata]
Pd = [bus[4] for bus in busdata]
Qd = [bus[5] for bus in busdata]
Pg = [bus[6] for bus in busdata]
Qg = [bus[7] for bus in busdata]
Osh = [bus[10] for bus in busdata]

delta = fnmath.pi/180*bus[3] for bus in busdata] # (1)
print ('delta', delta)

basevna = 100
# Initialize Psp, Qsp
Psp = [(bus[6] - bus[4])/basevna for bus in busdata] # (2)
Qsp = [(bus[7] - bus[5])/basevna for bus in busdata] # (3)
print ('Psp=', Psp)
print ('Qsp=', Qsp)

D: delta= [0.0, 0.0, 0.0, 0.0, 0.0]
Psp= [0.0, 0.2, -0.3, -0.4]
Qsp= [0.0, 0.0, -0.05, -0.1]

```

Υπολογισμός του πίνακα αγωγιμοτήτων του συστήματος

Η συνάρτηση calculation\_ybus υπολογίζει τον πίνακα αγωγιμοτήτων του συστήματος.

- Αρχικά όλες οι σύνθετες αντιστάσεις μετατρέπονται σε σύνθετες αγωγιμότητες από τη σχέση (5):

$$Z_{km} = R_{km} + jX_{km} \quad (4)$$

$$y_{km} = \frac{1}{Z_{km}} \quad (5)$$

- και στη συνέχεια υπολογίζονται τα διαγώνια και τα μη διαγώνια στοιχεία του πίνακα αγωγιμοτήτων από τις παρακάτω σχέσεις:

$$Y_{kk} = y_k + \sum_{m \in A(k)} (y_{km} + y_{km}) \quad (6)$$

$$Y_{km} = -y_{km} \quad (7)$$

Ο πίνακας αγωγιμοτήτων έχει μέγεθος (nbus x nbus) όπου n ο αριθμός των ζυγών του συστήματος που εξετάζουμε.

Στο παράδειγμα αυτό ο πίνακας αγωγιμοτήτων έχει μέγεθος (4x4):

$$Y_{bus} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} \end{bmatrix}$$

```

[30] # Function for calculation of admittance matrix
def calculation_ybus(nbus, linedata):
#initialize a matrix, nbus x nbus, complex type
y_bus = np.zeros((nbus, nbus), dtype=complex)

# Calculation of elements of the lines
for linedata in linedata:
p = int(linedata[0]) - 1
q = int(linedata[1]) - 1

zkm = complex(linedata[2], linedata[3]) # (4)
bsh = complex(0, linedata[4])
ykm = 1/zkm # (5)
y_bus[p,p] += ykm + bsh # (6)
if p != q:
y_bus[q,q] += ykm + bsh # (6)
y_bus[p,q] -= ykm # (7)
y_bus[q,p] = y_bus[p,q]

# Return admittance matrix
return y_bus

# Admittance matrix
y_bus = calculation_ybus(nbus, linedata)
print ('Ybus=')

```

D: ybus=

```

[[0. -15.j 0. +10.j 0. +0.j 0. +5.j]
 [0. +10.j 0. -15.j 0. +5.j 0. +0.j]
 [0. +0.j 0. +5.j 0. -15.j 0. +10.j]
 [0. +5.j 0. +0.j 0. +10.j 0. -15.j]]

```

Ο πίνακας αγωγιμοτήτων σε καρτεσιανή μορφή γράφεται:

$$Y_{km} = G_{km} + jB_{km} \quad (8)$$

```

[31] # Real part of admittance matrix
G = y_bus.real
print ('G=')
# Imaginary part of admittance matrix
B = y_bus.imag
print ('B=')

```

D: G=

```

[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
B=
[[-15. 10. 0. 5.]
 [10. -15. 5. 0.]
 [0. 5. -15. 10.]
 [5. 0. 10. -15.]]

```

Σχήμα 4. 24 : Κώδικας επίλυσης ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης σε Python στο Colab με τον κώδικα για των υπολογισμό των εγχύσεων ενεργού και αέργου ισχύος σε κάθε ζυγό, τις εξισώσεις και τον κώδικα για τον υπολογισμό του πίνακα αγωγιμοτήτων (2/9).

▼ **Χαρακτηρισμός ζυγών του συστήματος**

Τύποι ζυγών ροής φορτίου

Τύπος ζυγού	Γνωστές μεταβλητές	Άγνωστες μεταβλητές
Ταλάντωση ή αναφορά	$V = 1 \text{ a.u.}, \delta = 0^\circ$	$P_G, Q_G$
Ζυγός παραγωγής (PV)	$P_G, V$	$Q_G, \delta$
Ζυγός φορτίου (PQ)	$P_D, Q_D$	$V, \delta$

Στο παράδειγμα είναι:

- **Ζυγός Α:** Ζυγός ταλάντωσης (slack bus). Θεωρούμε ότι η τάση του είναι σταθερή κατά μέτρο και γωνία  $V_1 = 1 \angle 0^\circ$  α.μ.
- **Ζυγός Β:** Ζυγός παραγωγής (PV), αφού είναι γνωστά η παραγόμενη ενεργός ισχύς  $P_{BG} = 0, 3 \text{ a.u.}$  και το μέτρο της τάσης  $V_B = 1$  α.μ.
- **Ζυγός Γ:** Ζυγός φορτίου (PQ), αφού είναι γνωστές οι ισχύεις φορτίου  $P_{CG} = 0, 3 \text{ a.u.}, Q_{CG} = 0, 05 \text{ a.u.}$
- **Ζυγός Δ:** Ζυγός φορτίου (PQ), αφού είναι γνωστές οι ισχύεις φορτίου  $P_{DG} = 0, 4 \text{ a.u.}, Q_{DG} = 0, 1 \text{ a.u.}$

Στον πίνακα των δεδομένων των ζυγών ο ζυγός ταλάντωσης συμβολίζεται με 1, οι ζυγοί PV με 2 και οι ζυγοί PQ με 0.

Στο παρακάτω τμήμα κώδικα η μεταβλητή  $nPV$  εμφανίζει το πλήθος των ζυγών PV του προβλήματος, ενώ η μεταβλητή  $nPQ$  εμφανίζει το πλήθος των ζυγών PQ. Οι συντακτικές  $sl, pv$  και  $pq$  βρίσκουν το ζυγό ταλάντωσης και τους ζυγούς PV και PQ αντίστοιχα.

```

[] nPV=0
nPQ=0

for bus in busdata:
    if int(bus[1])==2:
        nPV+=1
    elif int(bus[1])==0:
        nPQ+=1

print ('nPV=', nPV)
print ('nPQ=', nPQ)

# Flag for slack bus
sl = [True if int(bus[1]) == 1 else False for bus in busdata]
print ('sl-', sl)
# Flag for pv bus
pv = [True if int(bus[1]) == 2 else False for bus in busdata]
print ('pv-', pv)
# Flag for pq bus
pq = [True if int(bus[1]) == 0 else False for bus in busdata]
print ('pq-', pv)

D: nPV= 1
nPQ= 2
sl= [True, False, False, False]
pv= [False, True, False, False]
pq= [False, True, False, False]
    
```

▼ **Δεδομένα και ζητούμενα**

**Το διάνυσμα κατάστασης**

Θεωρούμε ότι  $n$  είναι ο συνολικός αριθμός των ζυγών του συστήματος και  $m$  ο αριθμός των ζυγών φορτίου PQ. Γενικά για  $n$  ζυγούς έχουμε  $2n$  αγνώστους. Δεδομένου ότι η τάση στο ζυγό ταλάντωσης είναι δεδομένη κατά μέτρο και γωνία, ενώ τα μέτρα των τάσεων είναι γνωστά στους ζυγούς παραγωγής, οι μεταβλητές που πρέπει να προσδιοριστούν για να περιγραφεί πλήρως η κατάσταση λειτουργίας του συστήματος είναι οι ακόλουθες:

- οι φασικές γωνίες τάσεων όλων των ζυγών εκτός από το ζυγό ταλάντωσης
- τα μέτρα τάσεων όλων των ζυγών φορτίου.

Οι μεταβλητές αυτές αποτελούν το διάνυσμα κατάστασης  $X = \begin{bmatrix} \delta_2 \\ \vdots \\ \delta_n \\ V_{n-m+1} \\ \vdots \\ V_n \end{bmatrix}$  με διάσταση  $n - 1 \mid m$ . Άρα για την εύρεση των αγνώστων χρειάζονται  $n - 1 \mid m$  ανεξάρτητες εξισώσεις.

- $n - 1$  εξισώσεις πραγματικής ισχύος (για ζυγούς PV και PQ):
 
$$\Delta P_k^{(n)} = P_{p,k} - V_k^{(n)} \cdot \sum_{m \in A(k)} V_m^{(n)} \cdot [G_{km} \cdot \cos(\delta_k^{(n)} - \delta_m^{(n)}) + B_{km} \cdot \sin(\delta_k^{(n)} - \delta_m^{(n)})] \quad (9)$$
- $m$  εξισώσεις αέργου ισχύος (για ζυγούς PQ):
 
$$\Delta Q_k^{(n)} = Q_{q,k} - V_k^{(n)} \cdot \sum_{m \in A(k)} V_m^{(n)} \cdot [G_{km} \cdot \sin(\delta_k^{(n)} - \delta_m^{(n)}) - B_{km} \cdot \cos(\delta_k^{(n)} - \delta_m^{(n)})] \quad (10)$$

Στο παράδειγμα αυτό γνωρίζουμε την τάση του ζυγού Α κατά μέτρο και γωνία  $V_A = 1 \angle 0^\circ$  α.μ. καθώς και το μέτρο της τάσης του ζυγού Β,  $V_B = 1$  α.μ. Επομένως οι άγνωστοι του προβλήματος είναι 5, οι γωνίες  $\delta_B, \delta_C, \delta_D$  και η ύψους  $V_C, V_D$ . Το διάνυσμα κατάστασης λοιπόν έχει τη μορφή  $X = \begin{bmatrix} \delta_B \\ \delta_C \\ \delta_D \\ V_C \\ V_D \end{bmatrix}$  και για να υπολογιστεί θα χρειαστούν τρεις εξισώσεις πραγματικής ( $\Delta P_A, \Delta P_B, \Delta P_C$ ) και δύο εξισώσεις αέργου ισχύος ( $\Delta Q_C, \Delta Q_D$ ).

Σχήμα 4. 25 : Κώδικας επίλυσης ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης σε Python στο Colab με τον υπολογισμό του πίνακα αγωγιμοτήτων, τη θεωρία και τον κώδικα για τον χαρακτηρισμό των ζυγών του συστήματος καθώς και τη διαδικασία της δημιουργίας του διανύσματος διαφορών μέσω των επαναληπτικών εξισώσεων ροής φορτίου (3/9).

Επαναληπτική διαδικασία

- Πριν ξεκινήσουν οι επαναλήψεις της μεθόδου Newton-Raphson θεωρείται μια ενδιάμεση εκκίνηση για τις τάσεις όλων των ζυγίων του δικτύου, που δεν έχουν καθορισμένη τιμή, στα  $V = 1 \angle 0^\circ$  α.μ.
- Χρησιμοποιώντας αυτή την αρχική εκτίμηση υπολογίζεται, με τη βοήθεια των εξισώσεων ροής φορτίου με βελτιωμένη δεύτερη εκτίμηση η οποία χρησιμοποιείται για να βρεθεί μία τρίτη κ.ο.κ.
- Οι επαναλήψεις συνεχίζονται μέχρι οι νέες τιμές των μεταβλητών του συστήματος να διαφέρουν από τις αρχικές πολύ λίγο, δηλαδή όταν ικανοποιούνται τα κριτήρια σύγκλισης.

$$\begin{cases} |\Delta P_k^{(n)}| \leq \varepsilon \\ |\Delta Q_k^{(n)}| \leq \varepsilon \end{cases} \quad (11)$$

Στο πρόγραμμα αυτό το κριτήριο είναι:

$$|M^{(n)}| \leq \varepsilon \quad (12)$$

όπου  $M$  το διάνυσμα διαφορών:

$$M = \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \quad (13)$$

και

$$\varepsilon = 0.001$$

Η επαναληπτική διαδικασία γίνεται με ένα while loop, το οποίο στην εισαδό του ελέγχει το κριτήριο σύγκλισης της σχέσης (12). Ωστόσο, για να εξηγήσουμε αναλυτικά την 1η επανάληψη έχουμε χωρίσει τον κώδικα σε τμήματα και έχουμε βάλει τον έλεγχο σε σχόλιο, καθώς για να τρέξει επαναληπτικά πρέπει όλος ο κώδικας που περιέχεται στο while loop να βρίσκεται μέσα στο ίδιο κελί (cell). Στο παρακάτω κομμάτι κώδικα υπολογίζονται οι επαναληπτικές εξισώσεις ροής φορτίου σε όλους τους ζυγούς του ΣΗΕ. Από αυτές χρειάζομαστε τις τρεις εξισώσεις πραγματικής ισχύος ( $\Delta P_A, \Delta P_B, \Delta P_C$ ), που αποθηκεύονται στο διάνυσμα dP και τις δύο εξισώσεις αέργου ισχύος ( $\Delta Q_C, \Delta Q_D$ ), που αποθηκεύονται στο διάνυσμα dQ. Τα διανύσματα αυτά ενώνονται για να δημιουργηθεί το διάνυσμα διαφορών M που έχει ίδιες διαστάσεις με το X.

```
[38] # Initialize dP and dQ
dP = np.zeros((num_bus, 1))
dQ = np.zeros((num_bus, 1))

Tol=1
iter = 0

# Iteration starting...
while (Tol >= 0.001 and _iter_ <= 10): # (12)
    _iter_ += 1

# Initialize Active and Reactive Power
Pcal = np.zeros((num_bus, 1))
Qcal = np.zeros((num_bus, 1))

# Active and reactive power
for k in range(num_bus):
    for m in range(num_bus):
        Pcal[k,0] += Vm[k]*Vm[m]*(G[k,m]*cos(delta[k]-delta[m]) - B[k,m]*sin(delta[k]-delta[m])) # (9)
        Qcal[k,0] += Vm[k]*Vm[m]*(G[k,m]*sin(delta[k]-delta[m]) - B[k,m]*cos(delta[k]-delta[m])) # (10)

# Calculation of dP and dQ
for n in range(num_bus):
    if s1[n]:
        dP[n] = 0.0
        dQ[n] = 0.0
    else:
        dP[n] = Psp[n] - Pcal[n, 0] # (9)
        dQ[n] = Qsp[n] - Qcal[n, 0] # (10)

k = 0
d0 = np.zeros((nP0, 1))
for i in range(num_bus):
    if nq[i] == True:
        dQ[k,0] = dQo[i]
        k += 1

dP = dPa[1:]
print ('dP=\n', dP)
print ('dQ=\n', dQ)

M = np.vstack([dP, dQ])
print ('M=\n', M)
```

```
dP=
[[ 0.3]
 [-0.3]
 [-0.4]]
dQ=
[[ 0.2]
 [-0.1]]
M=
[[ 0.3]
 [-0.3]
 [-0.4]
 [ 0.2]
 [-0.1]]
```

Σχήμα 4. 26 : Κώδικας επίλυσης ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης σε Python στο Colab όπου δίνεται ο κώδικας για τη δημιουργία του διανύσματος διαφορών (4/9).

3. Υπολογισμός Ιακωβιανής μήτρας

Η Ιακωβιανή μήτρα έχει τη μορφή

$$Jac = \begin{bmatrix} H & N \\ J & L \end{bmatrix} \quad (14)$$

- Ο υποπίνακας  $H$  έχει διαστάσεις  $(n-1) \times (n-1)$
- Ο υποπίνακας  $N$  έχει διαστάσεις  $(n-1) \times m$
- Ο υποπίνακας  $J$  έχει διαστάσεις  $m \times (n-1)$
- Ο υποπίνακας  $L$  έχει διαστάσεις  $m \times m$

Στο παράδειγμα αυτό ο υποπίνακας  $H$  έχει διαστάσεις  $(3 \times 3)$ , ο  $N \rightarrow (3 \times 2)$ , ο  $J \rightarrow (2 \times 3)$  και ο  $L \rightarrow (2 \times 2)$   
 Επομένως η Ιακωβιανή μήτρα έχει τη μορφή:

$$Jac = \begin{bmatrix} H_{22} & H_{23} & H_{24} & N_{23} & N_{24} \\ H_{32} & H_{33} & H_{34} & N_{33} & N_{34} \\ H_{42} & H_{43} & H_{44} & N_{43} & N_{44} \\ J_{32} & J_{33} & J_{34} & L_{33} & L_{34} \\ J_{42} & J_{43} & J_{44} & L_{43} & L_{44} \end{bmatrix}$$

```
[14] # Calculation of Jacobian matrix
# Initialization of submatrices of Jacobian matrix
H = np.zeros((num_bus-1, num_bus-1))
N = np.zeros((num_bus-1, nPQ))
J = np.zeros((nPQ, num_bus-1))
L = np.zeros((nPQ, nPQ))
```

Τα διαγώνια και μη διαγώνια στοιχεία του υποπίνακα  $H$  δίνονται από τις σχέσεις:

$$H_{kk} = \frac{\partial \Delta P_k}{\partial \delta_k} = V_k^2 \cdot B_{kk} + Q_k \quad (15)$$

$$H_{km} = \frac{\partial \Delta P_k}{\partial \delta_m} = -V_k \cdot V_m \cdot [G_{km} \cdot \sin(\delta_k - \delta_m) - B_{km} \cdot \cos(\delta_k - \delta_m)], k \neq m \quad (16)$$

```
[15] # Calculation of H
for i in range(num_bus-1):
    k=i+1
    for j in range(num_bus-1):
        m=j+1
        for m in range (num_bus):
            if k==m:
                H[k-1, m-1] = Vm[k]**2*B[k,k]+Qcal[k] # (15)
            elif k!=m:
                H[k-1, m-1] = - Vm[k]*Vm[m]*(G[k,m]*sin(delta[k]-delta[m]) - B[k,m]*cos(delta[k]-delta[m])) # (16)
print ('H=\n', H)
```

Ο  $H =$

$$\begin{bmatrix} 15.75 & 5.25 & -0.1 \\ 5.25 & -15.25 & 10 \\ -0.1 & 10 & -15 \end{bmatrix}$$

Τα διαγώνια και μη διαγώνια στοιχεία του υποπίνακα  $N$  δίνονται από τις σχέσεις:

$$N_{kk} = \frac{\partial \Delta P_k}{\partial V_k} = -V_k \cdot G_{kk} - \frac{P_k}{V_k} \quad (17)$$

$$N_{km} = \frac{\partial \Delta P_k}{\partial V_m} = -V_k \cdot [G_{km} \cdot \cos(\delta_k - \delta_m) + B_{km} \cdot \sin(\delta_k - \delta_m)], k \neq m \quad (18)$$

```
[16] # Calculation of N
for i in range(num_bus-1):
    k=i+1
    for j in range (num_bus):
        if pq[j] == True:
            m=j
            for m in range (num_bus):
                if k==m:
                    N[k-1, m-nPV-1] = - Vm[k]*G[k, k] - Pcal[k]/Vm[k] # (17)
                elif k!=m:
                    N[k-1, m-nPV-1] = -Vm[m]*(G[k,m]*cos(delta[k]-delta[m]) + B[k,m]*sin(delta[k]-delta[m])) # (18)
print ('N=\n', N)
```

Ο  $N =$

$$\begin{bmatrix} -0. & -0. \\ 0. & -0. \\ 0. & -0. \end{bmatrix}$$

Τα διαγώνια και μη διαγώνια στοιχεία του υποπίνακα  $J$  δίνονται από τις σχέσεις:

$$J_{kk} = \frac{\partial \Delta Q_k}{\partial \delta_k} = V_k^2 \cdot G_{kk} - P_k \quad (19)$$

$$J_{km} = \frac{\partial \Delta Q_k}{\partial \delta_m} = V_k \cdot V_m \cdot [G_{km} \cdot \cos(\delta_k - \delta_m) + B_{km} \cdot \sin(\delta_k - \delta_m)], k \neq m \quad (20)$$

```
[17] # Calculation of J
for i in range(num_bus):
    if pq[i] == True:
        k=i
        for j in range (num_bus-1):
            m=j+1
            if k==m:
                for m in range (num_bus):
                    if k==m:
                        J[k-nPV-1, m-1] = Vm[k]**2*G[k, k] - Pcal[k] # (19)
                    elif k!=m:
                        J[k-nPV-1, m-1] = Vm[k]*Vm[m]*(G[k,m]*cos(delta[k]-delta[m]) + B[k,m]*sin(delta[k]-delta[m])) # (20)
print ('J=\n', J)
```

Σχήμα 4. 27: Κώδικας επίλυσης ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης σε Python στο Colab όπου υπολογίζονται οι υποπίνακες  $H$ ,  $N$  και  $J$  της Ιακωβιανής μήτρας. (5/9).

```

print ('J=\n', J)
L =
[[0. 0. 0.]
 [0. 0. 0.]]

```

Τα διαγώνια και μη διαγώνια στοιχεία του υποπίνακα  $L$ , δίνονται από τις σχέσεις:

$$L_{kk} = \frac{\partial \Delta Q_k}{\partial V_m} = V_k \cdot B_{kk} - \frac{Q_k}{V_k} \quad (21)$$

$$L_{km} = \frac{\partial \Delta Q_k}{\partial V_m} = -V_k \cdot [G_{km} \cdot \sin(\delta_k - \delta_m) - B_{km} \cdot \cos(\delta_k - \delta_m)], k \neq m \quad (22)$$

```

[38] # Calculation of L
for i in range(num_bus):
    i!=pq[i] == True:
        k=i
        for j in range(num_bus):
            if pq[j] == True:
                m=j
                if k==m:
                    for m in range(num_bus):
                        if k==m:
                            L[k-nPV-1, m-nPV-1] = Vm[k] * B[k, k] - Qcall[k]/Vm[m] # (21)
                        elif k!=m:
                            L[k-nPV-1, m-nPV-1] = -Vm[k]*(G[k, m]*sin(delta[k]-delta[m]) - B[k, m]*cos(delta[k]-delta[m])) # (22)
print ('L=\n', L)
L =
[[ -14.75  10.  ]
 [ 10.  -15.  ]]

```

Επομένως η Ιακωβιανή μήτρα για την 1η επανάληψη θα είναι:

```

[39] # Form the Jacobian matrix
JAC = -np.hstack((np.hstack((H, N)), np.hstack((J, L)))) # (16)
print ('JAC=\n', JAC)
JAC =
[[ 15.75 -5.25  0.  0.  0.  0. ]
 [ -5.25  15.25 -10.  0.  0.  0. ]
 [  0.  -10.  15.  0.  0.  0. ]
 [ -0.  -0.  0.  14.75 -10.  0. ]
 [ -0.  -0.  0.  0.  -10.  15.  ]]

```

#### 4. Υπολογισμός των νέων τάσεων και γωνιών του συστήματος

Από τη γενική εξίσωση Newton-Raphson υπολογίζεται το διάνυσμα των ανώσεων:

$$X = J^{-1}M^{(n)} \Rightarrow \begin{bmatrix} \Delta \delta \\ \Delta V \end{bmatrix} = \begin{bmatrix} H & N \\ J & L \end{bmatrix}^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \quad (23)$$

```

[40] # Calculation of the inverse Jacobian matrix
INV_JAC = np.linalg.pinv(JAC)

# Find the X vector
X = np.dot(INV_JAC, M) # (23)
print ('X=\n', X)

dTh = X[num_bus-1, 0]
dV = X[num_bus-1, 1]
print ('dTh=\n', dTh)
print ('dV=\n', dV)

X =
[[ -0.00371661]
 [  0.06829268]
 [ -0.07219512]
 [  0.01649485]
 [  0.0043299  ]]
dTh =
[ 0.00371661  0.06829268  0.07219512]
dV =
[ 0.01649485  0.0043299  ]

```

Οι νέες τιμές υπολογίζονται χρησιμοποιώντας τις τιμές της προηγούμενης επανάληψης:

$$\begin{bmatrix} \delta \\ V \end{bmatrix}^{(n+1)} = \begin{bmatrix} \Delta \delta \\ \Delta V \end{bmatrix} + \begin{bmatrix} \delta \\ V \end{bmatrix}^{(n)} \quad (24)$$

$$|V|^{(n+1)} = |\Delta V| + |V|^{(n)} \quad (25)$$

Δηλαδή:

$$\begin{bmatrix} 0 \\ \delta_B \\ \delta_C \\ \delta_D \end{bmatrix}^{(1)} = |\Delta \delta| + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

και

$$\begin{bmatrix} 1 \\ 1.05 \\ V_C \\ V_D \end{bmatrix}^{(1)} = |\Delta V| + \begin{bmatrix} 1 \\ 1.05 \\ 1 \\ 1 \end{bmatrix}$$

Σχήμα 4. 28 : Κώδικας επίλυσης ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης σε Python στο Colab όπου υπολογίζεται ο υποπίνακας L της Ιακωβιανής μήτρας, εμφανίζεται η τελική μορφή της και ξεκινά η διαδικασία υπολογισμού των νέων τάσεων και γωνιών του συστήματος (6/9).

```
[41] #Voltage Angle
delta[1:num_bus] = dTh # (24)
print ('delta-\n', delta)

#Voltage magnitude
k = 0
for i in range (1,num_bus):
    if p[1]==True:
        Vm[i] = dV[k] # (25)
        k += 1

print ('Vm-\n', Vm)

delta=
[0.0, -0.063716885948573665, -0.06829268292682926, -0.07219512195121949]
Vm=
[1.0, 1.05, 1.0164948453608247, 1.0943298969072165]
```

Μετατροπή γωνιών από ακτίνια σε μοίρες:

$$\delta^{\circ} = \frac{180^{\circ}}{\pi} \cdot (rad) \quad (26)$$

```
[42] deltad = np.zeros(num_bus, 1)
for n in range(num_bus):
    deltad[n] = (180/math.pi)*delta[n] # (26)

print ('deltad-\n', deltad)

deltad=
[[0. 1. 1. 1.]
 [-0.21294599]
 [-3.9128825 ]
 [-4.13647579]]
```

### 5. Υπολογισμός ισχύων στους ζυγούς

Οι ισχύς που χρειάζεται να προσδιοριστούν αν πρόκειται:

- Για ζυγό ταλάντωσης, είναι η ενεργός και η άεργος ισχύς του,  $P_G, Q_G$
- Για ζυγό PV, είναι η άεργος ισχύς του,  $Q_G$

Οι ισχύς αυτές δίνονται από τις σχέσεις:

$$P_G = P_e \cdot S_{base} + P_D \quad (27)$$

$$Q_G = Q_e \cdot S_{base} + Q_D \quad (28)$$

Και οι μιγαδικές ισχύεις εκφράζονται ως:

$$S_k = P_k + jQ_k \quad (29)$$

```
[43] S = np.zeros([num_bus, 1], dtype=np.complex )
V = np.zeros([num_bus, 1], dtype=np.complex )

for n in range (num_bus):
    if p[n]==True:
        Qg[n] = Qcal[n]*baseva + Pd[n] # (28)
        S[n] = complex(Pcal[n], Qcal[n]) # (29)
    elif sl[n]==True:
        Pg[n] = Pcal[n]*baseva + Pd[n] # (27)
        Qg[n] = Qcal[n]*baseva + Qd[n] # (28)
        S[n] = complex(Pcal[n], Qcal[n]) # (29)

for n in range (num_bus):
    V[n] = Vm[n]*complex(cos(delta[n]), sin(delta[n]))

print ('Pg=', Pg)
print ('Qg=', Qg)

Pg= [array([0.1], 30.0, 0.0, 0.0]
Qg= [array([ 50.1], array([78.751]), 0.0, 0.0]
```

Στο σημείο αυτό ολοκληρώνεται η 1η επανάληψη της μεθόδου. Η μεταβλητή Tol παίρνει την τιμή της μέγιστης απόλυτης τιμής του διανυσματος διαφορών M ώστε να ελεγχθεί το κριτήριο σύγκλισης στην αρχή του while loop. Το παρακάτω τμήμα κώδικα εμφανίζει τα τελικά αποτελέσματα της ροής φορτίου.

```
[44] Pgt = np.zeros(num_bus, 1)
Qgt = np.zeros(num_bus, 1)
Pdt = np.zeros(num_bus, 1)
Qdt = np.zeros(num_bus, 1)
Osh = np.zeros(num_bus, 1)

Pgt= sum(Pg)
Qgt= sum(Qg)
Pdt= sum(Pd)
Qdt= sum(Qd)
Osh= sum(Osh)

# This program prints the power flow solution in a tabulated form on the screen.
print(' Power Flow Solution by Newton-Raphson Method')
print(' Maximum Power Mismatch = ', Tol)
print(' No. of Iterations = ', _iter_, '\n')

print ('Bus Voltage Angle -----Load----- --Generation-- Injected')
print ('No. Mag. Degree MW Nvar MW Nvar Nvar Nvar ')
print(' ')

for n in range (num_bus):
    print(n+1, '\t', '%.2f'%Vm[n], '\t', '%.2f'%deltad[n], '\t', '%.2f'%Pd[n], '\t', '%.2f'%Qd[n],
          '\t', '%.2f'%Pg[n], '\t', '%.2f'%Qg[n], '\t', '%.2f'%Osh[n])

print(' \n')
print(' total ', '%.2f'%Pdt, '\t', '%.2f'%Qdt, '\t', '%.2f'%Pgt, '\t', '%.2f'%Qgt, '\t', '%.2f'%Osh)

# Tolerance..
```

Σχήμα 4. 29 : Κώδικας επίλυσης ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης σε Python στο Colab όπου υπολογίζονται οι νέες τάσεις, γωνίες και ισχύεις των ζυγών (7/9).

```

Power Flow Solution by Newton-Raphson Method
Maximum Power Mismatch = 1
No. of Iterations = 1

```

Bus No.	Voltage Mag.	Angle Degree	Load MW	Mvar	Generation MW	Mvar	Injected Mvar
1	1.000	0.00	0.00	0.00	0.00	50.00	0.00
2	1.050	0.21	0.00	0.00	30.00	78.75	0.00
3	1.016	-3.91	30.00	5.00	0.00	0.00	0.00
4	1.004	-4.14	10.00	10.00	0.00	0.00	0.00
Total			70.00	15.00	30.00	20.75	0.00

Το παρακάτω κελί περιέχει όλο τον κώδικα της ροής φορτίου με τη μέθοδο Newton-Raphson που περιγράψαμε αναλυτικά παραπάνω, το οποίο τρέχει και εμφανίζει τα αποτελέσματα για τις τρεις πρώτες επαναλήψεις όπου και συγκλίνει.

---

```

[45]
Power Flow Solution by Newton-Raphson Method
Maximum Power Mismatch = 1
No. of Iterations = 1

```

Bus No.	Voltage Mag.	Angle Degree	Load MW	Mvar	Generation MW	Mvar	Injected Mvar
1	1.000	0.00	0.00	0.00	0.00	-56.60	0.00
2	1.050	0.21	0.00	0.00	30.00	78.75	0.00
3	1.016	-3.91	30.00	5.00	0.00	0.00	0.00
4	1.004	-4.14	10.00	10.00	0.00	0.00	0.00
Total			70.00	15.00	30.00	20.75	0.00

```

Power Flow Solution by Newton-Raphson Method
Maximum Power Mismatch = [0.4]
No. of Iterations = 2

```

Bus No.	Voltage Mag.	Angle Degree	Load MW	Mvar	Generation MW	Mvar	Injected Mvar
1	1.000	0.00	0.00	0.00	40.12	-56.85	0.00
2	1.050	-0.22	0.00	0.00	30.00	71.21	0.00
3	1.014	-3.89	30.00	5.00	0.00	0.00	0.00
4	1.002	-4.12	10.00	10.00	0.00	0.00	0.00
Total			70.00	15.00	70.12	20.36	0.00

```

Power Flow Solution by Newton-Raphson Method
Maximum Power Mismatch = [0.01456785]
No. of Iterations = 3

```

Bus No.	Voltage Mag.	Angle Degree	Load MW	Mvar	Generation MW	Mvar	Injected Mvar
1	1.000	0.00	0.00	0.00	40.00	-49.53	0.00
2	1.050	-0.22	0.00	0.00	30.00	72.62	0.00
3	1.014	-3.89	30.00	5.00	0.00	0.00	0.00
4	1.002	-4.12	10.00	10.00	0.00	0.00	0.00
Total			70.00	15.00	70.00	23.09	0.00

```

[46] iter = np.arange(iter+1)
ubus=np. fliplr(Vbus)
pgbus=np. fliplr(Pgbus)
ogbus = np. fliplr(Ogbus)

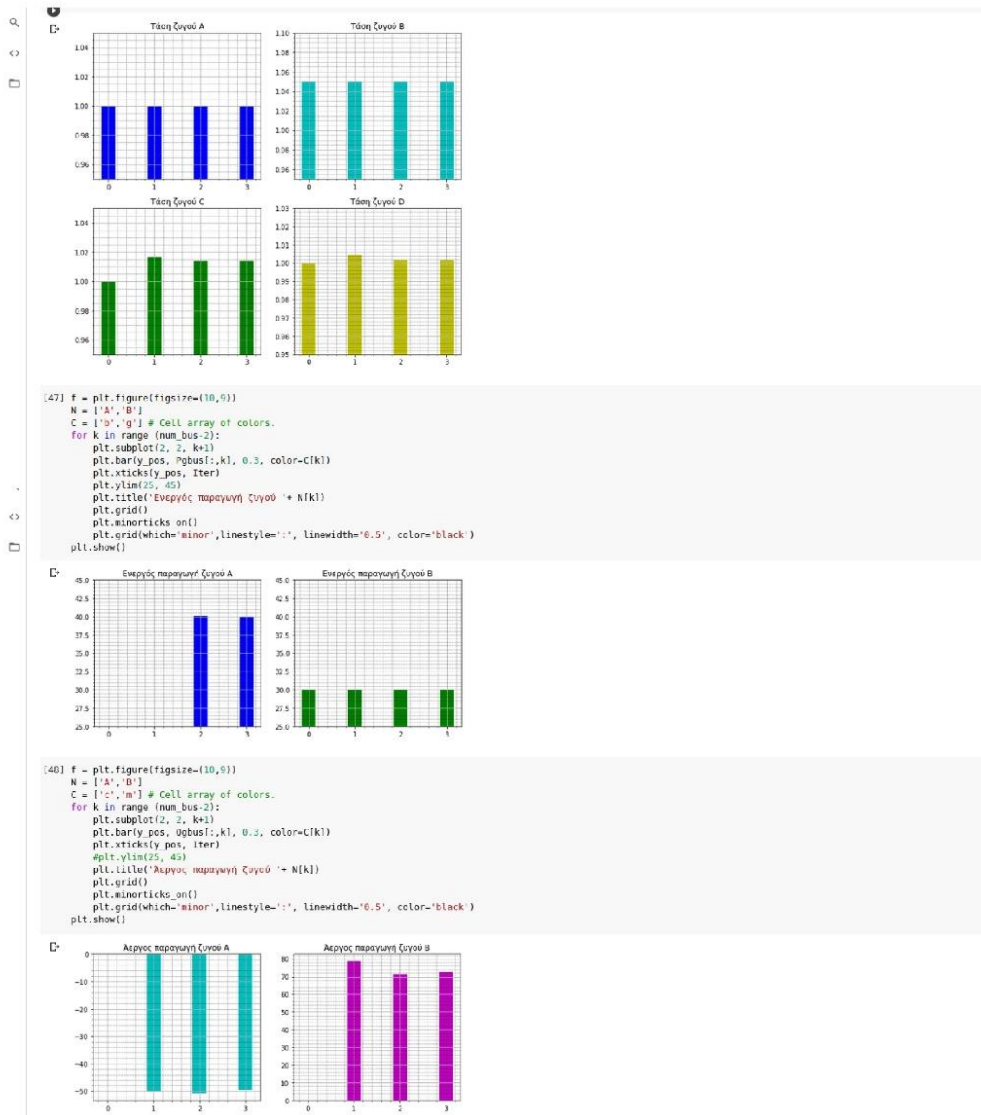
[48] y_pos = np.arange(len(Iter))
f = plt.figure(figsize=(10,9))

N = ['A', 'B', 'C', 'D']
L = [1.05, 1.1, 1.05, 1.03]
C = ['b', 'c', 'g', 'y'] # cell array of colors.
for k in range (num_buses):
    plt.subplot(2, 2, k+1)
    plt.bar(y_pos, Vbus[:,k], 0.3, color=C[k])
    plt.xticks(y_pos, Iter)
    plt.ylim(0.95, L[k])
    plt.title('Τόση ζυγού '+ N[k])
    plt.grid()
    plt.minorticks on()
    plt.grid(which='minor',linestyle=':', linewidth='0.5', color='black')
plt.show()

```

Σχήμα 4. 30 : Κώδικας επίλυσης ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης σε Python στο Colab που δείχνει τα αποτελέσματα της ροής φορτίου (8/9).





Σχήμα 4. 31 : Κώδικας επίλυσης ροής φορτίου της 1<sup>ης</sup> εργαστηριακής άσκησης σε Python στο Colab που δείχνει τις γραφικές αναπαραστάσεις των τάσεων όλων των ζυγών, της παραγωγής ενεργού ισχύος για τους ζυγούς αναφοράς και PV καθώς και της παραγωγής αεργού ισχύος για τους ζυγους PQ, για κάθε επανάληψη (9/9).

## Κεφάλαιο 5 : Συμπεράσματα

### Σύνοψη

*Στην ενότητα αυτή γίνεται μια ανακεφαλαίωση των θεμάτων με τα οποία ασχοληθήκαμε στην εργασία μας. Με βάση την εμπειρία που αποκομίσαμε από τη μελέτη σχετικά με την πλατφόρμα Jupyter Notebook και την χρήση της για την ανάπτυξη εκπαιδευτικού υλικού, παραθέτουμε κάποιες παρατηρήσεις καθώς και προτάσεις για μελλοντική αξιοποίηση του υλικού αυτού. Τέλος συνοψίζουμε τα συμπεράσματά μας.*

### 5.1 Εμπειρία χρήσης

Στην εργασία αυτή παρουσιάσαμε το εργαλείο Jupyter Notebook όπου το κείμενο, οι εικόνες και ο κώδικας μπορούν να συνδυαστούν με αρμονικό τρόπο, με σκοπό τη διερεύνηση καινοτόμων προσεγγίσεων και τη βελτίωση των δραστηριοτήτων διδασκαλίας και μάθησης, ιδιαίτερα σε θέματα που αφορούν την Ανάλυση ΣΗΕ.

Οι εφαρμογές που αναπτύχθηκαν απευθύνονται σε φοιτητές ηλεκτρολόγους μηχανικούς και αποσκοπούν στο να δημιουργήσουν μια πιο διαισθητική αντίληψη των φαινομένων πίσω από τις μαθηματικές εξισώσεις μέσω της αλληλεπίδρασης με διαδραστικά στοιχεία. Επιπλέον επιδιώκουν την καλύτερη κατανόηση της μεθοδολογίας επίλυσης των προβλημάτων μέσω της αναλυτικής παρουσίασης των βημάτων με τη βοήθεια ζωντανού κώδικα που συνοδεύεται από εξισώσεις και επεξηγήσεις. Τέλος στοχεύουν στη εξοικείωση των φοιτητών με προγραμματιστικά εργαλεία χρήσιμα για τη μετέπειτα πορεία τους.

Λόγω της απουσίας αξιολόγησης των προτεινόμενων σεναρίων, δεν μπορούμε να αποδείξουμε ότι ικανοποιούνται οι στόχοι της χρήσης του Jupyter Notebook σε μαθήματα ηλεκτρολόγων μηχανικών, όπως η Ανάλυση ΣΗΕ, αλλά μπορούμε να παρέχουμε κάποιες εμπειρίες σε πρακτικό επίπεδο, που αποκτήσαμε από πρώτο χέρι κατά τη δημιουργία αυτού του εκπαιδευτικού υλικού.

1. Η εγκατάσταση του λογισμικού του Jupyter Notebook σε προσωπικό υπολογιστή είναι εύκολη για τον χρήστη καθώς υπάρχουν αναλυτικές οδηγίες στο διαδίκτυο, επεξηγηματικά video στο Youtube και πολλές διαθέσιμες πληροφορίες σχετικά με την επίλυση τυχόν προβλημάτων.
2. Η κατανόηση των βασικών λειτουργιών του Jupyter Notebook είναι εύκολη ακόμα και για χρήστες με ελάχιστη εμπειρία στον προγραμματισμό. Έτσι ένας αρχάριος χρήστης μπορεί να αρχίσει απευθείας να πειραματίζεται με τις δυνατότητες του εργαλείου αυτού.
3. Η συγγραφή προγραμματιστικού κώδικα, ο εντοπισμός και η διόρθωση σφαλμάτων καθίστανται πιο εύκολα στο Jupyter Notebook σε σχέση με άλλα προγραμματιστικά περιβάλλοντα, καθώς είναι πολύ φιλικό προς τον χρήστη.

4. Στο Jupyter Notebook μπορούν να εκτελεστούν μικρά ή μεγαλύτερα τμήματα κώδικα, τα οποία μπορούμε να χωρίσουμε σε μικρότερα κομμάτια, να τα συνοδεύσουμε με επεξηγήσεις, εξισώσεις και να δούμε απευθείας τα αποτελέσματα που προκύπτουν από το συγκεκριμένο μέρος του κώδικα. Ωστόσο για να τρέξει μια διαδικασία, όπως ένα for loop, ένα while loop, μια if-else συνθήκη κ.λπ. θα πρέπει να περιλαμβάνεται ολόκληρη σε ένα κελί κώδικα.
5. Η εκτέλεση τμημάτων κώδικα που αποτελούν μέρη ενός μεγαλύτερου προγράμματος, θα πρέπει να γίνεται με τη σωστή σειρά για να προκύψουν τα σωστά αποτελέσματα. Κάποιες φορές μπορεί να χρειαστεί επανεκκίνηση ολόκληρου του προγράμματος για να διαγραφούν οι τιμές των μεταβλητών που έχουν προκύψει από προηγούμενη εκτέλεση, καθώς μπορεί να οδηγήσουν στην εμφάνιση λάθος αποτελεσμάτων.
6. Οι δυνατότητες που μπορεί να προσφέρει το Jupyter Notebook συνδέονται άρρηκτα με τον πυρήνα της γλώσσας προγραμματισμού που χρησιμοποιούμε. Συγκεκριμένα, η γλώσσα Python υποστηρίζεται άψογα από το Jupyter Notebook και παρέχει ένα πλήθος εξειδικευμένων επιλογών για τη δημιουργία διαδραστικών στοιχείων, οι οποίες δεν είναι διαθέσιμες όταν χρησιμοποιούμε για παράδειγμα τον πυρήνα MATLAB. Ωστόσο η βασική δομή και πολλές λειτουργίες του Jupyter Notebook είναι ίδιες όπως προγραμματιστική γλώσσα και να χρησιμοποιήσουμε.
7. Η σύνδεση του Jupyter Notebook με τον πυρήνα του MATLAB έχει ένα μικρό βαθμό δυσκολίας ωστόσο υπάρχουν διαθέσιμες οδηγίες στο διαδίκτυο και ορισμένες εναλλακτικές ανάμεσα στις οποίες μπορεί να επιλέξει ο χρήστης.
8. Τα cloud εργαλεία διευκολύνουν αρκετά σε περιπτώσεις που θέλουμε να αποφύγουμε τη διαδικασία της εγκατάστασης λογισμικού, μπορεί όμως να μην είναι κατάλληλα για την εκτέλεση προγραμμάτων που περιλαμβάνουν πιο εξειδικευμένα πακέτα και επίσης μπορεί να έχουν πιο αργή απόκριση, ειδικά κατά την εκτέλεση μεγαλύτερων προγραμμάτων.
9. Η κάθε μία από τις εφαρμογές που αναπτύχθηκαν περιέχει μια ολοκληρωμένη υπολογιστική διαδικασία και περιλαμβάνει τη θεωρία, τον κώδικα και τα αποτελέσματα του κώδικα αυτού σε ένα μοναδικό αρχείο. Το υλικό έχει διαμορφωθεί έτσι ώστε μια σύνθετη διαδικασία, όπως η επίλυση ενός προβλήματος ροής φορτίου να μπορεί να αποτυπωθεί στους φοιτητές καθώς χωρίζεται σε μικρότερα τμήματα, λόγω των κελιών. Με αυτόν τον τρόπο καθίσταται πιο εύκολη η κατανόηση των βημάτων της μεθοδολογίας αλλά και η ανάγνωση του κώδικα.
10. Διάφορες βοηθητικές προεκτάσεις του Jupyter Notebook δίνουν επιπλέον δυνατότητες, όπως να «κρύψουμε» ένα τμήμα κώδικα που μπορεί να μην ενδιαφέρει άμεσα τις ανάγκες του μαθήματος. Με την αναδίπλωση ενός τμήματος κώδικα μπορούμε να καταστήσουμε το αρχείο πιο ευανάγνωστο και να δώσουμε έμφαση στα σημεία που αφορούν την εφαρμογή της

θεωρίας. Σε κάθε περίπτωση, ο αναδιπλωμένος κώδικας είναι διαθέσιμος και οι φοιτητές μπορούν να ανοίξουν το κελί και να τον μελετήσουν, εφόσον το επιθυμούν.

## 5.2 Σχετικές εργασίες – Μελλοντική έρευνα

Στην εργασία αυτή δημιουργήσαμε εκπαιδευτικό υλικό το οποίο μπορεί μελλοντικά να διανθιστεί ή και να χρησιμοποιηθεί αυτούσιο ως μέρος της διδασκαλίας του μαθήματος Ανάλυσης ΣΗΕ. Ο στόχος κάποιων μελλοντικών εργασιών θα μπορούσε να περιλαμβάνει την αναβάθμιση του παρόντος εκπαιδευτικού υλικού ή και την επέκτασή του σε άλλα θέματα, πέρα από τη ρύθμιση τάσης και την ανάλυση του προβλήματος ροής φορτίου. Επίσης το Jupyter Notebook μπορεί να χρησιμοποιηθεί ως ένα μέσο για να συνδέσει εφαρμογές των ΤΠΕ που αξιοποιούνται ολοένα και περισσότερο στα Συστήματα Ηλεκτρικής Ενέργειας, όπως τα Νευρωνικά Δίκτυα και τα Συστήματα Μηχανικής Μάθησης. Είναι ένα εργαλείο που θα μπορούσε να βοηθήσει τους ενεργειακούς φοιτητές και ερευνητές Ηλεκτρολόγους Μηχανικούς να εξοικειωθούν με αυτά τα θέματα.

Σε πρακτικό επίπεδο, οι διδάσκοντες θα μπορούσαν αρχικά να συστήσουν το εκπαιδευτικό υλικό σε Jupyter Notebook στους φοιτητές χρησιμοποιώντας το ως έναν πειραματικό τρόπο επίλυσης ορισμένων αριθμητικών παραδειγμάτων στο αμφιθέατρο ή αναρτώντας το στο mycourses ως συμπληρωματικό εκπαιδευτικό υλικό το οποίο θα έχουν τη δυνατότητα να μελετήσουν, εφόσον το επιθυμούν. Με αυτόν τον τρόπο θα αποκτηθεί μια πρώτη εικόνα για το πως ανταποκρίνεται το σύνολο των φοιτητών, για παράδειγμα αν αυξάνεται το ενδιαφέρον τους για θέματα που παρουσιάζονται μέσω του Jupyter Notebook, η κατανόησή τους και η πρόθεσή τους να εξερευνήσουν οι ίδιοι το περιεχόμενο ενός τέτοιου διαδραστικού υπολογιστικού αρχείου.

Ένα επόμενο στάδιο θα μπορούσε να είναι η δημιουργία ακόμη πιο ολοκληρωμένων δραστηριοτήτων με μορφή εργαστηριακών εφαρμογών, όπου θα δοθεί περισσότερη έμφαση στην αλληλεπίδραση των φοιτητών με τον κώδικα, για παράδειγμα τροποποιώντας παραμέτρους, μέσω διαδραστικών στοιχείων ή χειροκίνητα, συμπληρώνοντας τμήματα κώδικα όπως εξισώσεις στις οποίες βασίζεται η προσομοίωση, με βάση τη θεωρία και τα παραδείγματα που δίνονται, απαντώντας ερωτήσεις, καταγράφοντας αποτελέσματά και παρατηρήσεις στο ίδιο αρχείο. Αυτό θα μπορούσε αρχικά να δοκιμαστεί πιλοτικά σε μια μικρή ομάδα φοιτητών και στη συνέχεια να γίνει αξιολόγηση με βάση την ικανότητα των φοιτητών να ολοκληρώσουν τις ασκήσεις, την επίδοσή τους, το ενδιαφέρον τους και άλλες παραμέτρους. Επίσης θα ήταν χρήσιμο να συγκεντρωθούν τα σχόλια των φοιτητών από την εμπειρία τους κατά τη χρήση του εκπαιδευτικού υλικού σε Jupyter Notebook ή σε κάποια άλλη από τις cloud εναλλακτικές. Με αυτόν τον τρόπο οι διδάσκοντες μπορούν να εξετάσουν αν ικανοποιούνται οι στόχοι της αξιοποίησης αυτών των εργαλείων σε θέματα Ανάλυσης ΣΗΕ, δηλαδή αν επιτυγχάνεται η καλύτερη κατανόηση της θεωρίας και των μεθοδολογιών από τους φοιτητές παράλληλα με την απόκτηση μεγαλύτερης άνεσης με τον προγραμματιστικό κώδικα. Μπορεί ακόμα να γίνει μια σύγκριση με άλλες εκπαιδευτικές μεθόδους όπως για παράδειγμα με τα σημερινά μέσα που χρησιμοποιεί το εργαστήριο, ώστε να εντοπιστούν οι ελλείψεις και τα πλεονεκτήματα αυτής της προσέγγισης.

Τέλος, η χρήση του Jupyter Notebook για την πραγματοποίηση Εργαστηριακών Ασκήσεων σε θέματα Ανάλυσης ΣΗΕ, θα μπορούσε να διευρυνθεί σε έναν μεγαλύτερο αριθμό φοιτητών. Σε μια τέτοια περίπτωση θα χρειαζόταν να δοθεί περισσότερη

προσοχή στον τρόπο με τον οποίο οι διδάσκοντες θα παρείχαν πρόσβαση στους φοιτητές στο εργαλείο αυτό (εγκατάσταση σε υπολογιστές του εργαστηρίου ή σε προσωπικούς υπολογιστές, χρήση server του εκπαιδευτικού ιδρύματος, χρήση cloud εναλλακτικών) ώστε να εξασφαλιστεί ευκολία αλλά ταυτόχρονα αξιοπιστία. Ενδεικτικά αναφέρεται πως σε άλλα μαθήματα της Σχολής ΗΜΜΥ όπου χρησιμοποιείται το Jupyter Notebook στη διδασκαλία, όπως τα Νευρωνικά Δίκτυα και Ευφυή Υπολογιστικά Συστήματα, οι διδάσκοντες προτιμούν τις cloud επιλογές που είναι εύκολα προσβάσιμες σε όλους και ενθαρρύνουν τους φοιτητές να εγκαταστήσουν το λογισμικό στον υπολογιστή τους, εφόσον το επιθυμούν.

### 5.3 Συμπεράσματα

Αναμφίβολα οι τεχνολογικές εξελίξεις στο πεδίο των Έξυπνων Συστημάτων Ηλεκτρικής Ενέργειας επιτάσσουν την αναβάθμιση της εκπαιδευτικής διαδικασίας με νέες πρακτικές και εργαλεία, ώστε οι φοιτητές να αποκτήσουν μια πολύπλευρη εκπαίδευση που προσφέρει κίνητρα για μάθηση και στοχεύει στην καλλιέργεια των δεξιοτήτων της ευελιξίας και της προσαρμοστικότητας στα συνεχώς μεταβαλλόμενα δεδομένα της εποχής μας, λαμβάνοντας υπόψη τον εξελισσόμενο ρόλο των ΤΠΕ.

Στο πλαίσιο αυτό παρουσιάστηκαν σύγχρονες εκπαιδευτικές τάσεις και εργαλεία που μπορούν να ενισχύσουν την εκπαίδευση των ΣΗΕ προς αυτή την κατεύθυνση, όπως τα Μαζικά Ανοικτά Ηλεκτρονικά Μαθήματα (MOOCs), τα διαδραστικά notebooks (Jupyter Notebooks) και η επαυξημένη πραγματικότητα (augmented reality). Ιδιαίτερη έμφαση δόθηκε στο υπολογιστικό εργαλείο Jupyter Notebook, η χρήση του οποίου στην εκπαίδευση παρουσιάζει πολυάριθμα πλεονεκτήματα, σύμφωνα με τη διεθνή βιβλιογραφία.

Μέσα από την παρουσίαση αντιπροσωπευτικών παραδειγμάτων αναδείχθηκαν τρόποι με τους οποίους το Jupyter Notebook μπορεί να χρησιμοποιηθεί σε παραδοσιακά μαθήματα Ηλεκτρολόγων Μηχανικών, όπως τα Συστήματα Ηλεκτρικής Ενέργειας και οι Ηλεκτρικές Μηχανές. Θετικά αξιολογήθηκε από διδάσκοντες και φοιτητές το γεγονός ότι όλες οι δραστηριότητες μπορούν να εκτελεστούν σε ένα μοναδικό αρχείο, που καθοδηγεί τους φοιτητές βήμα βήμα, χωρίς να είναι απαραίτητη η εναλλαγή σε διαφορετικές πλατφόρμες ή αρχεία. Επίσης σε όλες τις περιπτώσεις τονίστηκε η σημασία των οπτικοποιήσεων στην αύξηση του ενδιαφέροντος και της κατανόησης των φοιτητών.

Στη συνέχεια παρουσιάστηκε η πρώτη εφαρμογή που αναπτύχθηκε σε Jupyter Notebook, η οποία διαπραγματεύεται τη συμβολή της Διεσπαρμένης Παραγωγής στη ρύθμιση τάσης και είναι γραμμένη σε γλώσσα προγραμματισμού Python. Το αρχείο αυτό ενσωματώνει θεωρητικές έννοιες, εξισώσεις, τον κώδικα που εφαρμόζει τις εξισώσεις αυτές και τα αποτελέσματα της προσομοίωσης με τη μορφή γραφικών παραστάσεων. Μέσω διαδραστικών εισόδων, δίνεται η δυνατότητα στους φοιτητές να τροποποιούν τις τιμές των παραμέτρων και να βλέπουν απευθείας τις αλλαγές στις γραφικές παραστάσεις. Στόχος αυτής της άσκησης είναι βελτίωση της κατανόησης του θεωρητικού υποβάθρου από τους φοιτητές, μέσω της δημιουργίας μιας πιο διαισθητικής αντίληψης της συμπεριφοράς του συστήματος.

Οι υπόλοιπες εφαρμογές που αναπτύχθηκαν εστιάζουν στην ανάλυση ροής φορτίου. Το πρώτο παράδειγμα παρουσιάζει την επίλυση μιας άσκησης στη ροή φορτίου που διδάσκεται παραδοσιακά στο αμφιθέατρο, στις διαλέξεις του μαθήματος και αναπτύχθηκε στο Jupyter Notebook, σε γλώσσα MATLAB. Σε αυτό επεξηγούνται αναλυτικά όλα τα βήματα της μεθοδολογίας που ακολουθούνται για την επίλυση τέτοιου είδους ασκήσεων με παράλληλη εκτέλεση του αντίστοιχου κώδικα και εμφάνιση των αποτελεσμάτων.

Στο ίδιο πλαίσιο διαμορφώθηκε μια εφαρμογή για την ανάλυση ροής φορτίου στην cloud πλατφόρμα CoCalc σε γλώσσα MATLAB. Η εφαρμογή αυτή, βασίζεται στην 1<sup>η</sup> Εργαστηριακή Άσκηση του μαθήματος Ανάλυση Ροής Φορτίου του προγράμματος

σπουδών της σχολής HMMY, η οποία είναι πιο σύνθετη. Η εφαρμογή στοχεύει κυρίως στην εκτέλεση της προσομοίωσης και στην γραφική αναπαράσταση των αριθμητικών αποτελεσμάτων για την πιο ουσιαστική κατανόησή τους.

Στην τελευταία εφαρμογή παρουσιάστηκε μία εναλλακτική για την 1<sup>η</sup> Εργαστηριακή Άσκηση Ανάλυσης Ροής Φορτίου, σε γλώσσα Python στην cloud πλατφόρμα Colab, ώστε οι φοιτητές να αποκτήσουν πρόσβαση και να μπορούν να μελετήσουν ένα αντίστοιχο παράδειγμα γραμμένο σε αυτή την ολοένα και περισσότερο χρησιμοποιούμενη γλώσσα προγραμματισμού.

Τέλος παρουσιάστηκαν συμπεράσματα από την εμπειρία χρήσης του Jupyter Notebook κατά την ανάπτυξη αυτού του εκπαιδευτικού υλικού και αναλύθηκαν προτάσεις για πιθανή αξιοποίησή του στη συνέχεια.

Λαμβάνοντας υπόψη όλα τα παραπάνω αλλά και συγκεκριμένα τις δυνατότητες του Jupyter Notebook να συνδυάζει κείμενο και κώδικα προγραμματισμού σε ένα μόνο μέσο, να αποδομεί σύνθετα προβλήματα σε πιο απλά και να συνοδεύει τα αριθμητικά αποτελέσματα των προσομοιώσεων με ενδιαφέρουσες γραφικές παραστάσεις και οπτικοποιήσεις, γίνεται φανερό ότι αποτελεί ένα εργαλείο που μπορεί να συνεισφέρει στη βελτίωση της εκπαιδευτικής διαδικασίας στα πεδία Ηλεκτρολόγων Μηχανικών. Η προσαρμοστικότητά του και η εξελισσόμενη φύση του ως περιβάλλον ανοικτού κώδικα μπορεί να συμβάλει στην διερεύνηση καινοτόμων στρατηγικών για την ανάπτυξη νέων μεθόδων διδασκαλίας και την ενίσχυση του προγράμματος σπουδών στα Συστήματα Ηλεκτρικής Ενέργειας, το οποίο χρειάζεται να συμβαδίσει με τις εξελίξεις στα Ευφυή Δίκτυα. Σήμερα, ένα βασικό εφόδιο για την ανάπτυξη νέων πρακτικών και μεθοδολογιών, που προσφέρουν τα εργαλεία ανοιχτού κώδικα όπως το Jupyter Notebook, αποτελεί η διάθεση για συνεργασία και συλλογικότητα κατά τη δημιουργία διδακτικών πόρων, γεγονός που αποδεικνύεται από το πλήθος των εφαρμογών που είναι διαθέσιμες και ανοικτές στο διαδίκτυο (π.χ. στο GitHub). Ακόμα ένα πλεονέκτημα είναι η ευελιξία του Jupyter Notebook καθώς μπορεί να συνδυαστεί με εικονικά (virtual) εργαστήρια, Μαζικά Ανοικτά Ηλεκτρονικά Μαθήματα (MOOCs), διαλέξεις, απομακρυσμένα εργαστήρια και άλλα για να καλύψει τις ανάγκες του εκάστοτε μαθήματος. Η αυξανόμενη ωριμότητα τέτοιων εργαλείων, μπορεί να παρέχει νέους ενδιαφέροντες τρόπους διδασκαλίας και μάθησης. Το πλήθος των τεκμηριωμένων εφαρμογών που έχουν πραγματοποιηθεί ως σήμερα και οι θετικές αντιδράσεις φοιτητών και εκπαιδευτικών είναι ενθαρρυντικά στοιχεία για τη συνέχεια. Τέλος, χρήση τους μπορεί να αποτελέσει λύση για την πρακτική εκπαίδευση κατά τη διάρκεια της πανδημίας.



## Βιβλιογραφία

- [1] S. Srinivasan, U. Kotta and S. Ramaswamy, "A layered architecture for control functionality implementation in smart grids," 2013 10th IEEE INTERNATIONAL CONFERENCE ON NETWORKING, SENSING AND CONTROL (ICNSC), Evry, 2013, pp. 100-105.
- [2] Amin, M., & Stringer, J. (2008). The Electric Power Grid: Today and Tomorrow. *MRS Bulletin*, 33(4), 399-407. doi:10.1557/mrs2008.80
- [3] Βουρνάς Κ., Κονταξής Γ. (2010). ΕΙΣΑΓΩΓΗ ΣΤΑ ΣΥΣΤΗΜΑΤΑ ΗΛΕΚΤΡΙΚΗΣ ΕΝΕΡΓΕΙΑΣ., Εκδόσεις Συμμετρία.
- [4] X. Yu, C. Cecati, T. Dillon and M. G. Simões, "The New Frontier of Smart Grids," in *IEEE Industrial Electronics Magazine*, vol. 5, no. 3, pp. 49-63, Sept. 2011.
- [5] E. Santacana, G. Rackliffe, L. Tang and X. Feng, "Getting Smart," in *IEEE Power and Energy Magazine*, vol. 8, no. 2, pp. 41-48, March-April 2010, doi: 10.1109/MPE.2009.935557.
- [6] Tuballa, Maria Lorena and Abundo, Michael Lochinvar, (2016), A review of the development of Smart Grid technologies, *Renewable and Sustainable Energy Reviews*, 59, issue C, p. 710-725.
- [7] Οι εικόνες προέρχονται από έντυπο στο site των Διαχειριστών Ευρωπαϊκών Δικτύων Διανομής: European Distribution System Operators (E.DSO) <https://www.edsoforsmartgrids.eu/wp-content/uploads/brochure.pdf>
- [8] J. Romero Aguero, A. Khodaei and R. Masiello, "The Utility and Grid of the Future: Challenges, Needs, and Trends," in *IEEE Power and Energy Magazine*, vol. 14, no. 5, pp. 29-37, Sept.-Oct. 2016, doi: 10.1109/MPE.2016.2577899.
- [9] Strasser, T., Pröbstl Andrén, F., Lauss, G. *et al.* Towards holistic power distribution system validation and testing—an overview and discussion of different possibilities. *Elektrotech. Inftech.* 134, 71–77 (2017). <https://doi.org/10.1007/s00502-016-0453-3>
- [10] A. Malozemoff, "Second-generation high-temperature superconductor wires for the electric power grid", *AMSCAnnu. Rev. Mater. Res.*, vol. 42, pp. 373-397, Aug. 2012
- [11] Blumsack, Seth & Fernandez, Alisha, 2012."Ready or not, here comes the smart grid!" *Energy*, Elsevier, vol. 37(1), pages 61-68.
- [12] M. Kolhe, "Smart Grid: Charting a New Energy Future: Research Development and Demonstration", *The Electricity Journal*, vol. 25, issue 2, pp. 88-93, 2012
- [13] Nardelli, P., Rubido, N., Wang, C. *et al.* Models for the modern power grid. *Eur. Phys. J. Spec. Top.* 223, 2423–2437 (2014). <https://doi.org/10.1140/epjst/e2014-02219-6>

- [14] K. Mets, T. Verschueren, C. Develder, T. L. Vandoorn and L. Vandeveldel, "Integrated simulation of power and communication networks for smart grid applications," *2011 IEEE 16th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Kyoto, 2011, pp. 61-65, doi: 10.1109/CAMAD.2011.5941119.
- [15] X. Yu and Y. Xue, "Smart grids: A cyber–physical systems perspective", *Proc. IEEE*, vol. 104, no. 5, pp. 1058-1070, May 2016.
- [16] Y. Mo *et al.*, "Cyber–Physical Security of a Smart Grid Infrastructure," in *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195-209, Jan. 2012.
- [17] J. Hull, H. Khurana, T. Markham and K. Staggs, "Staying in control: Cybersecurity and the modern electric grid," in *IEEE Power and Energy Magazine*, vol. 10, no. 1, pp. 41-48, Jan.-Feb. 2012, doi: 10.1109/MPE.2011.943251.
- [18] K. K. Zame, C. A. Brehm, A. T. Nitica, C. L. Richard and D. S. Schweitzer, "Smart grid and energy storage: Policy recommendations", *Renew. Sustain. Energy Rev.*, vol. 82, pp. 1646-1654, Jul. 2018.
- [19] Yilmaz M, Krein PT. Review of benefits and challenges of vehicle-to-grid technology. *IEEE energy convers. Congr. Expo. IEEE*; 2012. p. 3082–9. doi: 10.1109/ECCE.2012.6342356.
- [20] Kotsampopoulos, P., Hatziargyriou, N., Strasser, T. I., Moyo, C., Rohjans, S., Steinbrink, C., ... Burt, G. M. (2017). Validating Intelligent Power and Energy Systems – A Discussion of Educational Needs. *Industrial Applications of Holonic and Multi-Agent Systems*, 200–212. doi:10.1007/978-3-319-64635-0\_15 doi: https://doi.org/10.1007/978-3-319-64635-0\_15
- [21] Ponce, P., Ibarra, L., Mata, O., & Molina, A. (2019). How to develop research skills among undergraduate engineering students to tackle current ongoing topics? A Smart-Grid case study. *The International Journal of Electrical Engineering & Education*. https://doi.org/10.1177/0020720918816004
- [22] Molina, A., Ponce, P., Baltazar Reyes, G.E. *et al.* Learning perceptions of Smart Grid class with laboratory for undergraduate students. *Int J Interact Des Manuf* 13, 1423–1439 (2019). https://doi.org/10.1007/s12008-019-00603-5
- [23] Noor, A. (2013). Envisioning engineering education and practice in the coming intelligence convergence era — a complex adaptive systems approach, *Open Engineering*, 3(4), 606-619. doi: https://doi.org/10.2478/s13531-013-0122-9
- [24] Opriş, I, Costinaş, S, Ionescu, CS, Gogoaş Nistoran, DE. Step-by-step augmented reality in power engineering education. *Comput Appl Eng Educ*. 2018; 26: 1590– 1602. https://doi.org/10.1002/cae.21969

- [25] May, D. (2020). Cross Reality Spaces in Engineering Education—Online Laboratories for Supporting International Student Collaboration in Merging Realities. *International Journal of Online and Biomedical Engineering (iJOE)*, 16(03), 4-26.
- [26] Barba, Lorena. (2020). Engineers Code: reusable open learning modules for engineering computations. *Computing in Science & Engineering*. PP. 1-1. 10.1109/MCSE.2020.2976002.
- [27] Chaczko Z., Alenazy W., Tran A. (2015) Augmented Reality and the Adapted of Smart Grid Monitoring for Educational Enhancement. In: Borowik G., Chaczko Z., Jacak W., Łuba T. (eds) *Computational Intelligence and Efficiency in Engineering Systems*. Studies in Computational Intelligence, vol 595. Springer, Cham
- [28] F. Cadoux, "MOOC "Smart Grids: the electric grids at the heart of the energy transition",  
2017 27th EAEEIE Annual Conference (EAEEIE), Grenoble, France, 2017, pp. 1-5, doi: 10.1109/EAEEIE.2017.8768649.
- [29] Carrera, J.; Ramírez-Hernández, D. Innovative Education in MOOC for Sustainability: Learnings and Motivations. *Sustainability* 2018, 10, 2990.
- [30] Pérez, F. & Granger, B. E. (2015) Project jupyter: Computational narratives as the engine of collaborative data science. <http://archive.ipython.org/JupyterGrantNarrative-2015.pdf>
- [31] Kluyver, Thomas, Ragan-Kelley, Benjamin, Pérez, Fernando, Granger, Brian, Bussonnier, Matthias, Frederic, Jonathan, Kelley, Kyle, Hamrick, Jessica, Grout, Jason, Corlay, Sylvain, Ivanov, Paul, Avila, Damián, Abdalla, Safia, Willing, Carol and Jupyter development team, (2016) *Jupyter Notebooks – a publishing format for reproducible computational workflows*. Loizides, Fernando and Schmidt, Birgit (eds.) In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. IOS Press. pp. 87-90. (doi:10.3233/978-1-61499-649-1-87).
- [32] Perkel, J.M. (2018). Why Jupyter is data scientists' computational notebook of choice. *Nature* 563 (2018), 145. <https://doi.org/10.1038/d41586-018-07196-1>
- [33] I. Opreș, S. Costinaș, C. S. Ionescu and D. E. G. Nistoran, "Towards augmented reality in power engineering," 2017 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, 2017, pp. 39-44, doi: 10.1109/ATEE.2017.7905160.
- [34] Cardoso A., Leitão J., Teixeira C. (2019) Using the Jupyter Notebook as a Tool to Support the Teaching and Learning Processes in Engineering Courses. In: Auer M., Tsiatsos T. (eds) *The Challenges of the Digital Transformation in Education*. ICL 2018. *Advances in Intelligent Systems and Computing*, vol 917. Springer, Cham doi: [https://doi.org/10.1007/978-3-030-11935-5\\_22](https://doi.org/10.1007/978-3-030-11935-5_22)
- [35] Cardoso, A., Leitão, J., Gil, P., Marques, A. S., & Simões, N. E. (2019b). Demonstration: Using IPython to Demonstrate the Usage of Remote Labs in Engineering Courses – A Case Study Using a Remote Rain Gauge. In M. E. Auer, &

- R. Langmann (Eds.), *Lecture Notes in Networks and Systems* (pp. 714-720). (Lecture Notes in Networks and Systems; Vol. 47). Springer. [https://doi.org/10.1007/978-3-319-95678-7\\_79](https://doi.org/10.1007/978-3-319-95678-7_79)
- [36] Lovejoy, M., Wickert, M. (2015): Using the IPython notebook as the computing platform for signals and systems courses. In: IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE 2015), pp. 289–294 (2015). doi: 10.1109/DSP-SPE.2015.7369568
- [37] Suárez, A., Alvarez-Feijoo, M.A., Fernández González, R., Arce, E. (2018) Teaching optimization of manufacturing problems via code components of a Jupyter Notebook. *Comput Appl Eng Educ.* 2018; 26: 1102– 1110. <https://doi.org/10.1002/cae.21941>
- [38] Jupyter Project, διαθέσιμο στο <http://jupyter.org>
- [39] Pérez, F. & Granger, B. E. (2007) IPython: A System for Interactive Scientific Computing, *Computing in Science and Engineering*, <http://ipython.org/>
- [40] GitHub, διαθέσιμο στο <https://github.com/>
- [41] Brunner RJ, Kim EJ. (2016) Teaching Data Science. International Conference on Computational Science 2016, ICCS 2016, 6-8 June 2016, San Diego, California, USA. DOI: <https://doi.org/10.1016/j.procs.2016.05.513>
- [42] O'Hara, K.J., Blank, D.S., Marshall, J.B. (2015) "Computational Notebooks for AI Education", *Computational Notebooks for AI Education*, pp. 263-268, 2015, April. 15, April.
- [43] Michael Zastre. 2019. Jupyter Notebook in CS1: An Experience Report. In *Proceedings of the Western Canadian Conference on Computing Education (WCCCE '19)*. Association for Computing Machinery, New York, NY, USA, Article 5, 1–6. DOI:<https://doi.org/10.1145/3314994.3325072>
- [44] Toews, C. (2017) Computational Inquiry in Introductory Statistics, *PRIMUS*, 27:7, 707-724, DOI: 10.1080/10511970.2016.1211204
- [45] Çetinkaya-Rundel, M. & Rundel, C. (2018) Infrastructure and Tools for Teaching Computing Throughout the Statistical Curriculum, *The American Statistician*, 72:1, 58-65, DOI: 10.1080/00031305.2017.1397549
- [46] Kany F., Louédoc B. (2018) A SPOC Produced by Sophomores for Their Junior Counterparts. In: Uskov V., Howlett R., Jain L. (eds) *Smart Education and e-Learning 2017*. SEEL 2017. Smart Innovation, Systems and Technologies, vol 75. Springer, Cham
- [47] Ketcheson, D. I. (2014). Teaching numerical methods with IPython notebooks and inquiry-based learning. In Stéfan van der Walt & James Bergstra (Eds.), *Proceedings of the 13th Python in Science Conference*(pp. 19–25). Retrieved from <http://hdl.handle.net/10754/346689>

- [48] Barba, L. & Forsyth, G. (2018). CFD Python: The 12 steps to Navier–Stokes equations. *Journal of Open Source Education*, 1(9), 21. <https://doi.org/10.21105/jose.00021>
- [49] Barba, L. & Mesnard, O. (2019). Aero Python: classical aerodynamics of potential flow using Python. *Journal of Open Source Education*, 2(15), 45, <https://doi.org/10.21105/jose.00045>
- [50] Tor Ole B. Odden and John Burk (2020). Computational Essays in the Physics Classroom. *The Physics Teacher* 58, 252 (2020); <https://doi.org/10.1119/1.5145471>
- [51] F. Milano, G. M. Jónsdóttir, Jupyter notebooks for Computer-based Laboratories on Power System Dynamics and Control, EDULEARN 2018, Palma de Mallorca, Spain, 2-4 July 2018.
- [52] F. Milano, G. M. Jónsdóttir, Jupyter Notebooks for Computer-based Laboratories on Electrical Energy Systems, EDULEARN 2019, Palma de Mallorca, Spain, 1-3 July 2019.
- [53] A. Ribeiro, A. Cardoso, J. A. Sá Marques and N. Eduardo Simões, "Web Interface for River Hydrodynamics Simulation," 2019 5th Experiment International Conference (exp.at'19), Funchal (Madeira Island), Portugal, 2019, pp. 278-279, doi: 10.1109/EXPAT.2019.8876550.
- [54] Srnec, M.N., Upadhyay, S. and Madura, J.D. (2016) Teaching Reciprocal Space to Undergraduates via Theory and Code Components of an IPython Notebook. *Journal of Chemical Education* 2016 93 (12), 2106-2109. doi: 10.1021/acs.jchemed.6b00392
- [55] Mendez, K.M., Pritchard, L., Reinke, S.N. *et al.* Toward collaborative open data science in metabolomics using Jupyter Notebooks and cloud computing. *Metabolomics* 15, 125 (2019). <https://doi.org/10.1007/s11306-019-1588-0>
- [56] JupyterHub Project, διαθέσιμο στο <http://jupyter.org/hub>
- [57] Binder, διαθέσιμο στο <https://mybinder.org/>
- [58] CoCalc, διαθέσιμο στο <https://cocalc.com>
- [59] Google Colaboratory, διαθέσιμο στο <https://colab.research.google.com/>
- [60] Kaggle Kernels, διαθέσιμο στο <https://kaggle.com/kernels>
- [61] Microsoft Azure notebooks, διαθέσιμο στο <https://notebooks.azure.com/>
- [62] Teaching and Learning with Jupyter. <https://jupyter4edu.github.io/jupyter-edu-book/>
- [63] nbviewer, διαθέσιμο στο <https://nbviewer.jupyter.org/>

- [64] NBgrader Project, διαθέσιμο στο <https://github.com/jupyter/nbgrader>
- [67] Π. Χ. Κοτσαμπόπουλος, Προσομοίωση σε πραγματικό χρόνο επιδράσεων διεσπαρμένης παραγωγής σε δίκτυα διανομής με χρήση ψηφιακών μοντέλων και πραγματικού εξοπλισμού, Διδακτορική Διατριβή, ΣΗΜΜΥ, ΕΜΠ, Αθήνα, Μάιος 2017.
- [68] Δημητρίου, Α., Κοτσαμπόπουλος, Π., Χατζηαργυρίου, Ν., Αξιοποίηση των Jupyter Notebooks στην εκπαιδευτική διαδικασία, 6ο Πανελλήνιο Συνέδριο «Ενταξη και Χρήση των ΤΠΕ στην Εκπαιδευτική Διαδικασία», Αθήνα, 18/10/2019-20/10/2019 <https://www.etpe2019.primedu.uoa.gr/>
- [69] Heinemann, B., Opel, S., Budde, L., Schulte, C., Frischemeier, D., Biehler, R., Podworny, S., & Wassong, T. (2018): Drafting a Data Science Curriculum for Secondary Schools. In *Proceedings of the 18th Koli Calling International Conference on Computing Education Re-search* Koli Calling '18). ACM
- [70] Yu, Y. F., Chen, Y. (2018). *Design and Development of High School Artificial Intelligence Textbook Based on Computational Thinking*. *Open Access Library Journal*, Volume 5(09),1-14. DOI: <https://doi.org/10.4236/oalib.1104898>
- [71] Chakravorty, D., McMullen, D., Liu, H., Ghaffari N., Rodriguez, D., Le, S., Gober, N., Wei, Z., Jordan, L., Pennings, M., Rodriguez, D., & Buchanan, C. (2019). Evaluating Active Learning Approaches for Teaching Intermediate Programming at an Early Undergraduate Level. (2019) *Journal of Computational Science Education*, Volume 10, Issue 1, pp. 61 - 66 DOI: <https://doi.org/10.22369/issn.2153-4136/10/1/10>
- [72] Vachovsky, M. E., Wu, G., Chaturapruek, S., Russakovsky, O., Sommer, R., & Fei-Fei, L. (2016). Toward More Gender Diversity in CS through an Artificial Intelligence Summer Program for High School Girls. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*. ACM, New York, NY, USA, (pp. 303-308). DOI: <https://doi.org/10.1145/2839509.2844620>
- [73] Bryant, C., Chen, Y., Chen, Z., Gilmour, J., Gumidyala, S., Herce-Hagiwara, B., Koures, A., Seoyeon, L., Msekela, J., Pham, AT., Remash, H., Remash, M., Schoenle, N., Zimmerman, J., Albright, S. D., & Rebelsky, S. A. (2019). A Middle-School Code Camp Emphasizing Data Science for Social Good. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. ACM, New York, NY, USA, 7. <https://doi.org/10.1145/3159450.3162313>
- [74] Mobasher, B., Dettori, L., Raicu, D., Settimi, R., Sonboli, N., & Stettler, M. (2019). Data Science Summer Academy for Chicago Public School Students. *SIGKDD Explor. Newsl.*, 21, 1 (May 2019), 49–52. DOI: <https://doi.org/10.1145/3331651.3331661>
- [75] Γ. Κονταξής, Ν. Χατζηαργυρίου, "Κέντρα Ελέγχου Ενέργειας", Αθήνα 2003, ΕΜΠ.
- [76] Π. Γεωργιλάκης, Σύγχρονα συστήματα μεταφοράς και διανομής ηλεκτρικής ενέργειας. Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών, 2015.

- [77] Παπαδιάς, Β. (1985). *Ανάλυση Συστήματος Ηλεκτρικής Ενέργειας: Μόνιμη Κατάσταση Λειτουργίας* (Τόμος 1), Αθήνα, Εκδόσεις Ε.Μ.Π.
- [78] Ν. Χατζηαργυρίου, Συμπληρωματικές Σημειώσεις για το Μάθημα Ανάλυση Συστήματος Ηλεκτρικής Ενέργειας Μόνιμη Κατάσταση Λειτουργίας, Αθήνα 1998
- [79] Μάθημα Εφαρμοσμένα Μαθηματικά, Εισαγωγή στον Επιστημονικό Υπολογισμό του Πανεπιστημίου του Harvard. Applied Math 111, Introduction to Scientific Computing, διαθέσιμο στο <https://am111.readthedocs.io/en/latest/>
- [80] Αρχείο στο αποθετήριο GitHub που περιλαμβάνει τον κώδικα της ροής φορτίου με τη μέθοδο Newton – Raphson σε γλώσσα MATLAB: <https://github.com/raghavlite/Newton-Raphson4/tree/master/Code>
- [81] ] Αρχείο στο αποθετήριο GitHub που περιλαμβάνει τον κώδικα της ροής φορτίου με τη μέθοδο Newton – Raphson σε γλώσσα Python: [https://github.com/aleslimacastro/fluxo\\_potencia\\_varredura](https://github.com/aleslimacastro/fluxo_potencia_varredura)
- [82] B. Frederickson, Ranking programming languages by GitHub users, 2018, διαθέσιμο στο <https://www.benfrederickson.com/ranking-programming-languages-by-github-users/>
- [83] J. Dong, T. B. Soeiro, J. Koeners and P. Bauer, "Online Laboratory Sessions for the Education of Electrical Machines and Drives," 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, Netherlands, 2020, pp. 430-435, doi: 10.1109/ISIE45063.2020.9152517.
- [84] P. Kotsampopoulos, V. Kleftakis, N. Hatziaargyriou, "Laboratory Education of Modern Power Systems using PHIL Simulation", IEEE Transactions on Power Systems, Vol. 32, Issue: 5, September 2017

## Δημοσιεύσεις

Α. Δημητρίου, Π. Κοτσαμπόπουλος, Ν. Χατζηαργυρίου, «Αξιοποίηση των Jupyter Notebooks στην εκπαιδευτική διαδικασία», 6ο Πανελλήνιο Επιστημονικό Συνέδριο «Ένταξη και Χρήση των ΤΠΕ στην Εκπαιδευτική Διαδικασία (ΕΤΠΕ), Αθήνα, Οκτώβριος 2019

# Παράρτημα

## Π1

Κώδικας σε Python για τη συμβολή της Διεσπαρμένης Παραγωγής στη ρύθμιση τάσης διαθέσιμος στο GitHub:

[https://nbviewer.jupyter.org/github/AndriDim/Jupyter\\_Notebook\\_files/blob/master/Voltage%20regulation.ipynb](https://nbviewer.jupyter.org/github/AndriDim/Jupyter_Notebook_files/blob/master/Voltage%20regulation.ipynb)

```
# Imports
import matplotlib.pyplot as plt
import pandas
import math

import numpy as np
import time
import pylab as plt
from IPython import display

import matplotlib.dates as md
from datetime import datetime
import datetime as dt

import threading

import ipywidgets as widgets
from ipywidgets import IntSlider
from ipywidgets import FloatSlider

from IPython.display import clear_output
from IPython.display import display

from ipywidgets import interactive, interactive_output
from ipywidgets import HBox

# Enable inline plotting
%matplotlib inline

global Pdg
global Qdg
global P1
global Q1
global Ppcc
global Qpcc
global past_pf_in
global past_pf_ca

R = 2.35
X = 1.17
P1 = 300
Q1 = 227
Vg = 230

# Slider variables
w = widgets.IntSlider(value=300, description='Pdg', min=200,max=400,step=50)
pf_in = FloatSlider(value=1, description='cosφ επαγ.', min=0.8,max=1,step=0.1)
pf_ca = FloatSlider(value=1, description='cosφ χwp.', min=0.8,max=1,step=0.1)

ui = widgets.HBox([w, pf_in, pf_ca])

def f(w, pf_in, pf_ca):
```



```

display((w, pf_in, pf_ca))

def on_value_change_Pdg(change):
    Pdg = change['new']

def on_value_change_pf_in(change):
    pf_in = change['new']

def on_value_change_pf_ca(change):
    pf_ca = change['new']

w.observe(on_value_change_Pdg, names='value')
pf_in.observe(on_value_change_pf_in, names='value')
pf_ca.observe(on_value_change_pf_ca, names='value')

past_pf_in = pf_in.value
past_pf_ca = pf_ca.value

# Plot
def plot(stime, stemp, stemp1, stemp2):
    f, axarr = plt.subplots(3, sharex=True, figsize=(7,7))
    axarr[0].step(stime, stemp)
    axarr[0].set_ylim(227, 231)

    axarr[1].step(stime, stemp1, 'r')
    axarr[1].set_ylim(-100, 150)

    axarr[2].step(stime, stemp2, 'g')
    axarr[2].set_ylim(-500, 0)

    axarr[0].set_ylabel("Vpcc(KV)")
    axarr[1].set_ylabel("Ppcc(KW)")
    axarr[2].set_ylabel("Qpcc(KVar)")

    axarr[2].set_xlabel("Time")

    axarr[0].set_title("Voltage")
    axarr[1].set_title("Active Power")
    axarr[2].set_title("Reactive Power")

    clear_output(wait=True)
    display(plt.gcf())
    plt.gcf().clear()
    time.sleep(2)

out = widgets.interactive_output(f, {'w': w, 'pf_in': pf_in, 'pf_ca': pf_ca})

display(ui, out)

def train():
    stime = np.array([])
    stemp = np.array([])
    stemp1 = np.array([])
    stemp2 = np.array([])

    while True:
        try:
            # Αν cosφ επαγωγικός ή cosφ = 1
            if (pf_in.value == 1 and pf_ca.value == 1) or (pf_in.value !=
past_pf_in):

```

```

    φ = math.acos(pf_in.value) # (1)
    Pdg = w.value
    Qdg =Pdg*math.tan(φ)      # (2)
    Qpcc = - Qdg - Ql        # (4α)

# Av cosφ χωρητικός
elif pf_ca.value != past_pf_ca:
    φ = math.acos(pf_ca.value) # (1)
    Pdg = w.value
    Qdg =Pdg*math.tan(φ)      # (2)
    Qpcc = Qdg - Ql          # (4β)

Ppcc = Pdg - P1             # (3)
ΔV = (Ppcc*R + Qpcc*X)/Vg   # (5)
Vpcc = Vg + ΔV             # (6)

print("Vpcc =", Vpcc, "kV", ", Ppcc =", Ppcc, "kW", ", Qpcc =", Qpcc,
"KVAr")

newTime = datetime.now()
temp = Vpcc
stime = np.append(stime,newTime) # Store the values
stemp = np.append(stemp,temp)
stemp1 = np.append(stemp1,Ppcc)
stemp2 = np.append(stemp2,Qpcc)
plot(stime,stemp,stemp1,stemp2)

except KeyboardInterrupt:
    break

try:
    thread = threading.Thread(target=train, args=())
    thread.start()
except:
    print("error")

```

## Π2

Ο κώδικας σε MATLAB που με μικρές διαφορές χρησιμοποιείται για τις εφαρμογές της Ανάλυσης Ροής Φορτίου, διαθέσιμος στο GitHub μαζί με τα αρχεία Busout.m και Lineflow.m:

Παράδειγμα άσκησης :

[https://nbviewer.jupyter.org/github/AndriDim/Jupyter\\_Notebook\\_files/blob/master/Roh\\_fortiou\\_paradeigma.ipynb](https://nbviewer.jupyter.org/github/AndriDim/Jupyter_Notebook_files/blob/master/Roh_fortiou_paradeigma.ipynb)

Εργαστηριακή Άσκηση:

[https://nbviewer.jupyter.org/github/AndriDim/Jupyter\\_Notebook\\_files/blob/master/Newton\\_Raphson\\_method\\_matlab.ipynb](https://nbviewer.jupyter.org/github/AndriDim/Jupyter_Notebook_files/blob/master/Newton_Raphson_method_matlab.ipynb)

```
% Newton_Raphson iterations
basemva = 100;

%          Bus Bus  Voltage Angle  ---Load---  -----Generator-----  Injected
%          No  code Mag.   Degree  MW   Mvar   MW   Mvar  Qmin  Qmax  Mvar
busdata=[1  1  1.00  0.0   0   0   0   0   0   0   0
          2  2  1.05  0.0   0   0  30   0   0  500  0
          3  0  1.00  0.0  30   5   0   0   0  500  0
          4  0  1.00  0.0  40  10   0   0   0   0   0];

%          Bus bus   R      X      1/2 B      = 1 for lines
%          nl  nr   p.u.  p.u.  p.u.      > 1 or < 1 tr. tap at bus nl
linedata=[1  2  0.0  0.1  0.0      1
           1  4  0.0  0.2  0.0      1
           2  3  0.0  0.2  0.0      1
           3  4  0.0  0.1  0.0      1];

Vm=0; delta=0; yload=0; deltad=0;
nbus = length(busdata(:,1));
kb=[];Vm=[]; delta=[]; V=[];
Pg = zeros(nbus,1); Pd = zeros(nbus,1); Qg = zeros(nbus,1); Qd = zeros(nbus,1);
Psp = zeros(nbus,1); Qsp = zeros(nbus,1); Qsh = zeros(nbus,1);
S = zeros(nbus,1);

for k=1:nbus
n=busdata(k,1); kb(n)=busdata(k,2); Vm(n)=busdata(k,3); delta(n)=busdata(k, 4);
Pd(n)=busdata(k,5); Qd(n)=busdata(k,6); Pg(n)=busdata(k,7); Qg(n) = busdata(k,8);
    delta(n) = pi/180*delta(n);
    Psp(n)=(Pg(n)-Pd(n))/basemva; % (4)
    Qsp(n)=(Qg(n)-Qd(n))/basemva; % (5)
    S(n) = Psp(n) + j*Qsp(n); % (6)
end

nl = linedata(:,1); %line from bus
nr = linedata(:,2); %line to bus
R = linedata(:,3); %line resistance
X = linedata(:,4); %line reactance
Bc = j*linedata(:,5);
a = linedata(:, 6);
nbr=length(linedata(:,1)); %number of branch
G=[];B=[];

%the impedance of line
Z = R + j*X; %(10)
%the admittance of line
y= ones(nbr,1)./Z; %(11)
```

```

for n = 1:nbr
if a(n) <= 0 a(n) = 1; else end
Ybus=zeros(nbus,nbus); % initialize Ybus to zeros
% formation of the off diagonal elements
for k=1:nbr
Ybus(nl(k),nr(k))=Ybus(nl(k),nr(k))-y(k)/(a(k)); %(9)
Ybus(nr(k),nl(k))=Ybus(nl(k),nr(k));
end
end
% formation of the diagonal elements
for n=1:nbus
for k=1:nbr
if nl(k)==n
Ybus(n,n) = Ybus(n,n)+y(k)/(a(k)^2)+ Bc(k); %(8)
elseif nr(k)==n
Ybus(n,n) = Ybus(n,n)+y(k)+Bc(k); %(8)
else, end
end
end
end

G = real(Ybus); % Conductance matrix..
B = imag(Ybus); % Susceptance matrix..

nPV = 0;
nPQ = 0;

for k = 1:nbus
if kb(k)==2 % PV bus
nPV = nPV+1;end
if kb(k)==0
nPQ = nPQ+1;end % PQ bus
end

pv = find(kb == 2 | kb == 1); % PV Buses..
pq = find(kb == 0); % PQ Buses..

Pgbus=[]; Pgit=[];
Qgbus=[]; Qgit=[];

Pgbus= Pg';
Pgit=Pgbus;
Qgbus= Qg';
Qgit=Qgbus;

Vbus = []; Vit=[];
Vbus = Vm;
Vit = Vbus;

Tol = 1;
Iter = 0;
while (Tol >=0.001 && Iter<3) % Iteration starting.. (16)
Iter = Iter +1;
P = zeros(nbus,1);Q = zeros(nbus,1); % Calculate P and Q

for k = 1:nbus
for m = 1:nbus
P(k) = P(k) + Vm(k)*Vm(m)*(G(k,m)*cos(delta(k)-delta(m)) +
B(k,m)*sin(delta(k)-delta(m))); % (13)
Q(k) = Q(k) + Vm(k)*Vm(m)*(G(k,m)*sin(delta(k)-delta(m)) -
B(k,m)*cos(delta(k)-delta(m))); % (14)
end
end

dPa = Psp-P; %(13)

```

```

dQa = Qsp-Q; %(14)

k = 1;
dQ = zeros(nPQ,1);
for i = 1:nbus
    if kb(i) == 0
        dQ(k,1) = dQa(i);
        k = k+1;
    end
end

dP = dPa(2:nbus);
dQ;

% Mismatch Vector
M = [dP; dQ]; % (17)

clear H N J L
H = zeros(nbus-1,nbus-1);
N = zeros(nbus-1,nPQ);
J = zeros(nPQ,nbus-1);
L = zeros(nPQ,nPQ);

for i = 1:(nbus-1)
    k = i+1;
    for j = 1:(nbus-1)
        m = j+1;
        if k == m
            for m = 1:nbus
                H(i,j) = H(i,j) + Vm(k)*Vm(m)*(G(k,m)*sin(delta(k)-delta(m)) -
B(k,m)*cos(delta(k)-delta(m))); % (20)
            end
            H(i,j) =(Vm(k)^2)*B(k,k)+ H(i,j); % (19)
        else
            H(i,j) = -Vm(k)*Vm(m)*(G(k,m)*sin(delta(k)-delta(m)) -
B(k,m)*cos(delta(k)-delta(m))); % (21)
        end
    end
end

for i = 1:(nbus-1)
    k = i+1;
    for j = 1:nPQ
        m = pq(j);
        if k == m
            for m = 1:nbus
                N(i,j) = N(i,j) + Vm(m)*(G(k,m)*cos(delta(k)-delta(m)) +
B(k,m)*sin(delta(k)-delta(m))); % (23)
            end
            N(i,j) =- Vm(k)*G(k,k)- N(i,j); % (22)
        else
            N(i,j) = -Vm(k)*(G(k,m)*cos(delta(k)-delta(m)) +
B(k,m)*sin(delta(k)-delta(m))); % (24)
        end
    end
end

for i = 1:nPQ
    k = pq(i);
    for j = 1:(nbus-1)
        m = j+1;
        if k == m
            for m = 1:nbus

```

```

                J(i,j) = J(i,j) + Vm(k)*Vm(m)*(G(k,m)*cos(delta(k)-delta(m)) +
B(k,m)*sin(delta(k)-delta(m))); %(26)
            end
                J(i,j) = (Vm(k)^2)*G(k,k)-J(i,j); %(25)
            else
                J(i,j) = Vm(k)* Vm(m)*(G(k,m)*cos(delta(k)-delta(m)) +
B(k,m)*sin(delta(k)-delta(m))); %(27)
            end
        end
    end
end

for i = 1:nPQ
    k = pq(i);
    for j = 1:nPQ
        m = pq(j);
        if k == m
            for m = 1:nbus
                L(i,j) = L(i,j) + Vm(m)*(G(k,m)*sin(delta(k)-delta(m)) -
B(k,m)*cos(delta(k)-delta(m))); % (29)
            end
                L(i,j) = Vm(k)*B(k,k)-L(i,j); % (28)
            else
                L(i,j) = -Vm(k)*(G(k,m)*sin(delta(k)-delta(m)) -
B(k,m)*cos(delta(k)-delta(m))); % (30)
            end
        end
    end
end

Jac = -[H   N; J   L]; % Jacobian matrix

% Correction Vector
X = inv(Jac)*M; % (31)

dTh = X(1:nbus-1); % Change in Voltage Angle..
dV = X(nbus:end); % Change in Voltage Magnitude..

% Updating State Vectors..
% Voltage Angle..
delta(2:nbus) = dTh + delta(2:nbus)'; % (32)

% Voltage Magnitude..
k = 1;
for i = 2:nbus
    if kb(i) == 0
        Vm(i) = dV(k) + Vm(i); % (33)
        k = k+1;
    end
end

deltad=180/pi*delta; % (34)

iter = [0:Iter];
Vbus = [Vm; Vit];
Vit = Vbus;

k=0;
for n = 1:nbus
    if kb(n) == 1
        k=k+1;
        S(n)= P(n)+j*Q(n); % (37)
        Pg(n) = P(n)*basemva + Pd(n); % (35)
        Qg(n) = Q(n)*basemva + Qd(n); % (36)
    elseif kb(n) ==2

```

```

        k=k+1;
        S(n)=P(n)+j*Q(n); % (37)
        Qg(n) = Q(n)*basemva + Qd(n); % (36)
    end
yload(n) = (Pd(n)- j*Qd(n))/(basemva*Vm(n)^2);
end

Pgbus = [Pg'; Pgit];
Pgit = Pgbus;

Qgbus = [Qg'; Qgit];
Qgit = Qgbus;

Busout
Lineflow

% Tolerance..
Tol = max(abs(M));
end

A=flipud(Vbus);
iter;

N = {'A','B','C','D'};
L = {1.05, 1.1, 1.05, 1.03};
C = {'b','c','g','y'}; % Cell array of colors.
for k = 1:4
    subplot(2,2,k);
    bar(iter, A(:,k), 0.3,C{k})
    ylim([0.95 L{k}])
    grid on
    grid minor
    title(['Tasi zugou ',N{k}])
end

B=flipud(Pgbus);
D=flipud(Qgbus);

N = {'A','B'};
C = {'b','c'};
for k = 1:2
    subplot(1,2,k);
    bar(iter, B(:,k), 0.3,C{k})
    grid on
    grid minor
    title(['Energos paragogi zugou ',N{k}])
end

N = {'A','B'};
C = {'y','m'};
for k = 1:2
    subplot(1,2,k);
    bar(iter, D(:,k), 0.3,C{k})
    grid on
    grid minor
    title(['Aergos paragogi zugou ',N{k}])
end

```

### Π3

Κώδικας σε Python για την Ανάλυση Ροής Φορτίου διαθέσιμος στο GitHub:

[https://github.com/AndriDim/Jupyter\\_Notebook\\_files/blob/master/Newton\\_Raphson\\_method\\_in\\_Python.ipynb](https://github.com/AndriDim/Jupyter_Notebook_files/blob/master/Newton_Raphson_method_in_Python.ipynb)

```
import numpy as np
import math
from math import cos, sin
import matplotlib.pyplot as plt
%matplotlib inline

def define_bus_data():
    bdataname = ['NUM', 'TYPE', 'VOLT', 'DEGREE', 'PLOAD', 'QLOAD', 'PGER', 'QGER',
                'QMIN', 'QMAX', 'INJMVAR']
    busdata= np.array([[1, 1, 1.00, 0.0, 0, 0, 0, 0, 0, 0,
                        0],
                       [2, 2, 1.05, 0.0, 0, 0, 30, 0, 0, 500,
                        0],
                       [3, 0, 1.00, 0.0, 30, 5, 0, 0, 0, 500,
                        0],
                       [4, 0, 1.00, 0.0, 40, 10, 0, 0, 0, 0,
                        0]])
    num_bus = busdata.shape[0]
    return busdata, num_bus

busdata, num_bus = define_bus_data()

def define_line_data():
    ldataname= ['NL', 'NR', 'R', 'X', '1/2Bc', 'TAP']
    linedata= np.array([[1, 2, 0.0, 0.1, 0.0, 1],
                        [1, 4, 0.0, 0.2, 0.0, 1],
                        [2, 3, 0.0, 0.2, 0.0, 1],
                        [3, 4, 0.0, 0.1, 0.0, 1]])
    num_lin = linedata.shape[0]
    return linedata, num_lin

linedata, num_lin = define_line_data()

bus_type = [bus[1] for bus in busdata]
Vm = [bus[2] for bus in busdata]
Pd = [bus[4] for bus in busdata]
Qd = [bus[5] for bus in busdata]
Pg = [bus[6] for bus in busdata]
Qg = [bus[7] for bus in busdata]
Qsh = [bus[10] for bus in busdata]

delta = [math.pi/180*bus[3] for bus in busdata] # (1)

basemva = 100
# Initialize Psp, Qsp
Psp = [(bus[6] - bus[4])/basemva for bus in busdata] # (2)
Qsp = [(bus[7] - bus[5])/basemva for bus in busdata] # (3)

# Admittance matrix
y_bus = calculation_ybus(num_bus, linedata)

# Real part of admittance matrix
G = y_bus.real
# Imaginary part of admittance matrix
B = y_bus.imag

nPv=0
```



```

nPQ=0

for bus in busdata:
    if int(bus[1])==2:
        nPV+=1
    elif int(bus[1])==0:
        nPQ+=1

# Flag for sluck bus
sl = [True if int(bus[1]) == 1 else False for bus in busdata]
# Flag for pv bus
pv = [True if int(bus[1]) == 2 else False for bus in busdata]
# Flag for pq bus
pq = [True if int(bus[1]) == 0 else False for bus in busdata]

# Initialize dp and dq
dPa = np.zeros([num_bus, 1])
dQa = np.zeros([num_bus, 1])

Vbus = [bus[2] for bus in busdata]
Pgbus = [bus[6] for bus in busdata]
Qgbus = [bus[7] for bus in busdata]

Tol=1
_iter_=0

# Iteration starting..
while (Tol >= 0.001 and _iter_<= 10): # (12)
    _iter_+=1
    # Initialize Active and Reactive Power
    Pcal = np.zeros([num_bus, 1])
    Qcal = np.zeros([num_bus, 1])

    # Active and reactive power
    for k in range(num_bus):
        for m in range(num_bus):
            Pcal[k,0] += Vm[k]*Vm[m]*(G[k,m]*cos(delta[k]-delta[m])+ B[k,m]*sin(delta[k]-
delta[m])) # (9)
            Qcal[k,0] += Vm[k]*Vm[m]*(G[k,m]*sin(delta[k]-delta[m]) -
B[k,m]*cos(delta[k]-delta[m])) # (10)

    # Calculation of dP and dQ
    for nbus in range(num_bus):
        if sl[nbus]:
            dPa[nbus] = 0.0
            dQa[nbus] = 0.0
        else:
            dPa[nbus] = Psp[nbus] - Pcal[nbus, 0]
            dQa[nbus] = Qsp[nbus] - Qcal[nbus, 0]

    k = 0
    dQ = np.zeros([nPQ, 1])
    for i in range (num_bus):
        if pq[i]==True:
            dQ[k,0] = dQa[i]
            k += 1

    dP = dPa[1:]

    M = np.vstack([dP, dQ])

# Calculation of Jacobian matrix

```

```

# Initialization of submatrixes of Jacobian matrix
H = np.zeros([num_bus-1, num_bus-1])
N = np.zeros([num_bus-1, nPQ])
J = np.zeros([nPQ, num_bus-1])
L = np.zeros([nPQ, nPQ])

# Calculation of H
for i in range(num_bus-1):
    k=i+1
    for j in range(num_bus-1):
        m=j+1
        for m in range (num_bus):
            if k==m:
                H[k-1, m-1] = Vm[k]**2*B[k,k]+Qcal[k] # (15)
            elif k!=m:
                H[k-1,m-1] = - Vm[k]*Vm[m]*(G[k,m] * sin(delta[k]-delta[m]) - B[k,m] *
cos(delta[k]-delta[m])) # (16)

# Calculation of N
for i in range(num_bus-1):
    k=i+1
    for j in range (num_bus):
        if pq[j] == True:
            m=j
            for m in range (num_bus):
                if k==m:
                    N[k-1, m-nPV-1] = - Vm[k]*G[k, k]-Pcal[k]/Vm[k] # (17)
                elif k!=m:
                    N[k-1,m-nPV-1] = -Vm[m]*(G[k,m]*cos(delta[k]-delta[m]) +
B[k,m]*sin(delta[k]-delta[m])) # (18)

# Calculation of J
for i in range(num_bus):
    if pq[i] == True:
        k=i
        for j in range (num_bus-1):
            m=j+1
            if k==m:
                for m in range (num_bus):
                    if k==m:
                        J[k-nPV-1, m-1] = Vm[k]**2*G[k, k] - Pcal[k] # (19)
                    elif k!=m:
                        J[k-nPV-1,m-1] = Vm[k]*Vm[m]*(G[k,m]*cos(delta[k]-delta[m]) + B[k,m]
* sin(delta[k]-delta[m])) # (20)

# Calculation of L
for i in range(num_bus):
    if pq[i] == True:
        k=i
        for j in range (num_bus):
            if pq[j] == True:
                m=j
                if k==m:
                    for m in range (num_bus):
                        if k==m:
                            L[k-nPV-1, m-nPV-1] = Vm[k] * B[k, k] - Qcal[k]/Vm[m] # (21)
                        elif k!=m:
                            L[k-nPV-1,m-nPV-1] = -Vm[k]*(G[k,m]*sin(delta[k]-delta[m]) -
B[k,m]*cos(delta[k]-delta[m])) # (22)

# Form the Jacobian matrix
JAC = -np.vstack([np.hstack([H, N]), np.hstack([J, L])]) # (14)

```

```

# Calculation of the inverse Jacobian matrix
INV_JAC = np.linalg.inv(JAC)

# Find the X vector
X = np.dot(INV_JAC, M) # (23)

dTh = X[:num_bus-1,0]
dV = X[-nPQ:,0]

#Voltage Angle..
delta[1:num_bus] += dTh # (24)

#Voltage magnitude
k = 0
for i in range(1,num_bus):
    if pq[i]==True:
        Vm[i] += dV[k] # (25)
        k += 1

deltad = np.zeros([num_bus, 1])
for nbus in range(num_bus):
    deltad[nbus] = (180/math.pi)*delta[nbus] # (26)

S = np.zeros([num_bus, 1], dtype=np.complex )

for n in range (num_bus):
    if pv[n]==True:
        Qg[n] = Qcal[n]*basemva + Pd[n] # (28)
        S[n] = complex(Pcal[n], Qcal[n]) # (29)
    elif sl[n]==True:
        Pg[n] = Pcal[n]*basemva + Pd[n] # (27)
        Qg[n] = Qcal[n]*basemva + Qd[n] # (28)
        S[n] = complex(Pcal[n], Qcal[n]) # (29)

for n in range (num_bus):
    V[n] = Vm[n]*complex(cos(delta[n]),sin(delta[n]))

Vbus = np.vstack([Vm,Vbus])
Pgbus = np.vstack([Pg,Pgbus])
Qgbus = np.vstack([Qg,Qgbus])

busdata[:,2] = Vm
busdata[:,3] = np.transpose(deltad)

Pgt = np.zeros([num_bus, 1])
Qgt = np.zeros([num_bus, 1])
Pdt = np.zeros([num_bus, 1])
Qdt = np.zeros([num_bus, 1])
Qsht = np.zeros([num_bus, 1])

Pgt= sum(Pg)
Qgt= sum(Qg)
Pdt= sum(Pd)
Qdt= sum(Qd)
Qsht= sum(Qsh)

# This program prints the power flow solution in a tabulated form on the screen.
print('          Power Flow Solution by Newton-Raphson Method')
print('          Maximum Power Mismatch = ', Tol)
print('          No. of Iterations = ', _iter_, '\n')

print ('Bus      Voltage Angle -----Load----- ---Generation--- Injected')
print ('No.      Mag.      Degree MW          Mvar      MW          Mvar      Mvar      ')
print( '          ')

```

```

    for n in range (num_bus):

print(n+1, '\t', "%.3f"%Vm[n], '\t', "%.2f"%deltad[n], '\t', "%.2f"%Pd[n], '\t', "%.2f"%Qd[
n],
        '\t', "%.2f"%Pg[n], '\t', "%.2f"%Qg[n], '\t', "%.2f"%Qsh[n])

    print('      \n')
    print('      Total      ',
          "%.2f"%Pdt, '\t', "%.2f"%Qdt, '\t', "%.2f"%Pgt, '\t', "%.2f"%Qgt, '\t', "%.2f"%Qsht, '\n')

    Tol= abs(max(M, key=abs))

Iter = np.arange(_iter_+1)
Vbus=np.flipud(Vbus)
Pgbus=np.flipud(Pgbus)
Qgbus = np.flipud(Qgbus)

y_pos = np.arange(len(Iter))
f = plt.figure(figsize=(10,9))

N = ['A', 'B', 'C', 'D']
L = [1.05, 1.1, 1.05, 1.03]
C = ['b', 'c', 'g', 'y'] # Cell array of colors.
for k in range (num_bus):
    plt.subplot(2, 2, k+1)
    plt.bar(y_pos,Vbus[:,k], 0.3, color=C[k])
    plt.xticks(y_pos, Iter)
    plt.ylim(0.95, L[k])
    plt.title('Τάση ζυγού '+ N[k])
    plt.grid()
    plt.minorticks_on()
    plt.grid(which='minor',linestyle=':', linewidth='0.5', color='black')
plt.show()

f = plt.figure(figsize=(10,9))
N = ['A', 'B']
C = ['b', 'g'] # Cell array of colors.
for k in range (num_bus-2):
    plt.subplot(2, 2, k+1)
    plt.bar(y_pos, Pgbus[:,k], 0.3, color=C[k])
    plt.xticks(y_pos, Iter)
    plt.ylim(25, 45)
    plt.title('Ενεργός παραγωγή ζυγού '+ N[k])
    plt.grid()
    plt.minorticks_on()
    plt.grid(which='minor',linestyle=':', linewidth='0.5', color='black')
plt.show()

f = plt.figure(figsize=(10,9))
N = ['A', 'B']
C = ['c', 'm'] # Cell array of colors.
for k in range (num_bus-2):
    plt.subplot(2, 2, k+1)
    plt.bar(y_pos, Qgbus[:,k], 0.3, color=C[k])
    plt.xticks(y_pos, Iter)
    #plt.ylim(25, 45)
    plt.title('Αεργός παραγωγή ζυγού '+ N[k])
    plt.grid()
    plt.minorticks_on()
    plt.grid(which='minor',linestyle=':', linewidth='0.5', color='black')
plt.show()

```