

Simulateur d'écoulement de puissance en régime triphasé

par

Antoine BEAUDRY

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN GÉNIE ÉLECTRIQUE
M. Sc. A.

MONTRÉAL, LE 11 OCTOBRE 2016

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

©Tous droits réservés, Antoine Beaudry, 2016

©Tous droits réservés

Cette licence signifie qu'il est interdit de reproduire, d'enregistrer ou de diffuser en tout ou en partie, le présent document. Le lecteur qui désire imprimer ou conserver sur un autre media une partie importante de ce document, doit obligatoirement en demander l'autorisation à l'auteur.

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Pierre-Jean Lagacé, directeur de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Ambrish Chandra, président du jury
Département de génie électrique à l'École de technologie supérieure

M. Maarouf Saad, examinateur
Département de génie électrique à l'École de technologie supérieure

M. Laurent Lenoir, examinateur externe
Institut de recherche d'Hydro-Québec (IREQ)

M. Serge Lefebvre, examinateur externe
Institut de recherche d'Hydro-Québec (IREQ)

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

«DATE»

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

J'aimerais premièrement remercier ma femme de m'avoir appuyé tout au long de mes études, et de m'avoir donné le courage de me rendre jusqu'au bout. Merci d'être toujours là pour moi, dans les bons comme dans les moins bons moments, et de m'avoir encouragé à persévérer au cours des quatre ans que j'ai passés à l'École de Technologie Supérieure.

De plus, j'aimerais remercier Thierry Lavoie et Bruno Bigras de m'avoir aidé avec la programmation C++ pour l'intégration de la librairie KLU, ainsi que de m'avoir expliqué la base de la programmation lorsque j'ai fait la transition entre Python et C++. Ce geste m'a permis de sauver énormément de temps et d'énergie, et j'en serai toujours reconnaissant.

Une autre personne que j'aimerais remercier est le professeur Pierre-Jean Lagacé. Vous avez été disponible tout au long de mon projet, et vous m'avez beaucoup aidé avec le développement de mon programme.

Finalement, j'aimerais remercier mes parents qui ont toujours cru en moi, et qui m'ont toujours encouragé malgré les différentes embûches que j'ai rencontrées au cours de ma vie. Je suis convaincu que je n'aurais pas pu réussir sans vous.

Enfin je veux dédier ce mémoire à l'ensemble des personnes énumérées ci-haut. À votre façon, vous m'avez tous et chacun aidé à réussir dans ce long processus scolaire. Merci pour tout !

SIMULATEUR D'ÉCOULEMENT DE PUISSANCE EN RÉGIME TRIPHASÉ

Antoine BEAUDRY

RÉSUMÉ

Ce mémoire présente la conceptualisation d'un simulateur d'écoulement de puissance d'un réseau électrique triphasé tout en abordant l'intégration des dérivations monophasées, des transformateurs Yy comprenant les impédances de mise à la terre, des transformateurs Yd , des transformateurs monophasés et des transformateurs à changement de prises. Cela a pour but d'aider l'équipe de chercheurs de l'institut de recherche d'Hydro-Québec (IREQ) à calculer l'écoulement de puissance d'un réseau de distribution associé au réseau de transport électrique.

Pour ce faire, il faut premièrement expliquer comment construire un fichier de données permettant de lancer l'application. Par la suite, il faut démontrer comment installer les différentes librairies au cas où l'IREQ désirerait faire des modifications subséquentes dans les fichiers sources. Troisièmement, il faut faire la modélisation des différentes composantes afin de calculer de façon précise la matrice d'admittance Y du réseau triphasé complet. Une fois celle-ci calculée, et sachant que la puissance active des barres de génération, ainsi que la puissance active et réactive des barres de charge sont connues, il faut créer une matrice calculant les erreurs résiduelles des puissances active et réactive de ces barres (matrice F). Une fois cela fait, il faut calculer la matrice Jacobienne. Celle-ci fait la dérivation des puissances active et réactive de chaque barre par rapport au déphasage et à la tension. La prochaine étape consiste à expliquer deux méthodes itératives qui permettent de calculer les indices de rapprochement. Les valeurs finales des tensions du réseau seront trouvées à l'aide de ces indices. Ceci est fait tout et aussi longtemps que les erreurs résiduelles sont plus grandes que la valeur maximale exigée par le programme. Les deux méthodes utilisées sont les méthodes Newton-Raphson et Levenberg-Marquardt. Maintenant que le cœur du simulateur est expliqué, il faut intégrer la régulation de tension. Celle-ci permet de maintenir une ou plusieurs barres dans le réseau entre des valeurs spécifiques indiquées dans le fichier de données. L'avant-dernière étape de ce mémoire est d'expliquer comment lire le fichier de rapport qui est généré après chaque simulation. C'est ici que le simulateur intègre toutes les informations concernant le réseau, ainsi que la puissance distribuée dans l'ensemble de celui-ci. Finalement, la dernière étape est la validation du simulateur par rapport au prototype (Pierre-Jean Lagacé, 2011).

Il y a plusieurs recommandations à faire afin d'aider à la convergence des résultats. Une d'entre elles consiste à faire l'implémentation d'une pondération avec la méthode Levenberg-Marquardt. Cela permettrait de résoudre les cas ayant un réseau de distribution connecté au réseau de transport, car sans pondération, la majorité des systèmes divergent, et ne sont pas capables de converger aux bons résultats.

Mots-clés : Newton-Raphson, Levenberg-Marquardt, réseau triphasé, simulation

POWER FLOW SIMULATOR FOR A THREE-PHASE ELECTRICAL GRID

Antoine BEAUDRY

ABSTRACT

This thesis presents the conceptualization of a power flow simulator for a three-phase electrical grid while addressing the integration of single-phase bypasses, Yy transformers with ground impedances, Yd transformers, single-phase transformers and tap changing transformers. Its goal is to help the IREQ team to calculate the power flow of a distribution grid associated with a transmission grid.

We must first explain how to build a data file which will launch the application. Then, we must demonstrate how to install the various data libraries in case the IREQ wishes to make subsequent changes in the source files. Thirdly, we must conduct the modeling of the various components in order to accurately calculate the admittance matrix Y of the complete three-phase grid. Once calculated, and knowing the active power of the generation bars, as well as the active and reactive power of the load bars, the simulator will create a matrix to solve the residual errors of the active and reactive power of these bars (matrix F). Once created, the Jacobian matrix is calculated. It calculates the derivation of the active and reactive powers for each bar with respect to the phase and voltage. The next step is to explain the two iterative methods that allow us to calculate the reconciliation indicators. The final values of the grid's voltage will be found using these indicators. This will happen as long as the residual errors are larger than the maximum value required by the program. The two methods are the Newton-Raphson method and Levenberg-Marquardt method. Now that the central structure of the simulator is explained, we must integrate the voltage regulation. It allows us to maintain one or more bars in the grid between specific values indicated in the data file. The penultimate step of this thesis is to explain how to read the report file that is generated after each simulation. This is where the simulator incorporates all the information about the grid, as well as the power distributed throughout it. Lastly, we must do a validation of the simulator in comparison to the prototype (Pierre-Jean Lagacé, 2011).

There are several recommendations to be made in order to help with the results convergence. One of these recommendations is the implementation of a weighting with the Levenberg-Marquardt method. This would solve the situation in which there is a distribution grid connected to the transmission grid, because without weighting, most systems will diverge, and will not be able to converge in order to provide suitable results.

Keywords: Newton-Raphson, Levenberg-Marquardt, simulation, three-phase electrical grid

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 REVUE DE LITTÉRATURE	3
1.1 Problématique	3
1.2 Objectifs	3
1.3 Revue de littérature	4
1.3.1 Différence entre C++, Java, Go et Scala pour un même programme	4
1.3.2 Écoulement de puissance	5
1.3.2.1 Écoulement de puissance optimale (OPF)	5
1.3.2.2 Écoulement de puissance avec Gauss-Seidel.....	6
1.3.2.3 Écoulement de puissance NR modifié pour l'intégration éolienne	8
1.3.2.4 Écoulement de puissance à l'aide d'une combinaison de la méthode Newton-Raphson et de la méthode d'optimisation de Newton.....	9
1.3.2.5 Améliorer la convergence d'un écoulement de puissance par la méthode de Levenberg-Marquardt.....	10
1.3.2.6 Solution d'écoulement de puissance multi phase pour les réseaux de distribution à grand échelle utilisant l'algorithme MANA	13
CHAPITRE 2 MÉTHODOLOGIE	17
2.1 Introduction.....	17
2.2 Construction des fichiers de données.....	17
2.2.1 Explications de base.....	17
2.2.2 Barres monophasées et triphasées.....	18
2.2.3 Lignes monophasées et triphasées	19
2.2.4 Transformateurs	20
2.2.5 Très important.....	21
2.3 Installation des librairies	21
2.3.1 Étapes pour l'installation des librairies.....	21
2.3.2 Versions installées	22
2.4 Modélisation des composantes	22
2.4.1 Explication	22
2.4.2 Lignes.....	22
2.4.2.1 Lignes moyennes (80 km <= l <= 400km)	22
2.4.2.2 Lignes longues	24
2.4.3 Transformateurs	27
2.4.4 Transformateur monophasé	27
2.4.5 Transformateur triphasé	29
2.4.6 Transformateur triphasé Yd0	30

2.4.6.1	Création de la matrice d'admittance quand $Z_1 = 0$ et $Y = 0$	31
2.4.6.2	Création de la matrice d'admittance où seulement $Z_1 = 0$	33
2.4.6.3	Création de la matrice d'admittance avec toutes les variables (Z_1 , Z_2 et Y).....	34
2.4.7	Transformateur triphasé $Yy0$	37
2.4.7.1	Création de la matrice d'admittance où $Z_1 = 0$ et $Y = 0$	37
2.4.7.2	Création de la matrice d'admittance incluant toutes les variables du schéma	39
2.5	Création de la matrice d'admittance totale	41
2.5.1	Modélisation de la matrice avant l'optimisation.....	41
2.5.2	Optimisation de la matrice d'admittance	43
2.5.2.1	Barre de génération (PV) ou d'équilibre (SW) vers barre de charge (PQ)	44
2.5.2.2	Barre de charge (PQ) vers barre de génération (PV) ou d'équilibre (SW)	46
2.5.2.3	Barre de génération (PV) vers barre de génération (PV) ou d'équilibre (SW)	48
2.6	Comparaison entre la matrice d'admittance complète et la matrice d'admittance simplifiée.....	49
2.7	Implémentation des méthodes dans le simulateur	50
2.7.1	Méthode Newton-Raphson	50
2.7.1.1	Matrice d'erreur résiduelle.....	50
2.7.1.2	Création de la matrice Jacobienne	52
2.7.1.3	Calcul de l'espace mémoire requis pour créer la matrice Jacobienne.....	52
2.7.1.4	Formules utilisées pour calculer la matrice Jacobienne.....	53
2.7.1.5	Création de la boucle itérative	54
2.7.2	Méthode Levenberg-Marquardt	54
2.7.2.1	Ajouter l'opérateur Lagrangien (Lambda).....	54
2.7.2.2	Création de la boucle itérative	55
2.7.3	Paramètres à déterminer.....	55
2.7.4	Conditions d'arrêt de la boucle itérative	56
2.8	Implémentation de la régulation	56
2.8.1	Algorithme utilisé pour l'implémentation de la régulation.....	56
2.8.2	Algorithme utilisé pour l'implémentation de la régulation de tension	56
2.8.3	Explication de l'algorithme de régulation.....	58
2.8.4	Calcul de la variable a_1 pendant la régulation.....	60
2.8.5	Calcul de la matrice d'admittance avec la régulation	61
2.9	Lecture du rapport	62
2.9.1	Structure du rapport	62
2.10	Conclusion	63
CHAPITRE 3 ÉTUDE DE CAS		65
3.1	Introduction.....	65
3.2	Cas #1 : Réseau simple avec deux barres triphasées PQ à impédance constante	66

3.2.1	Éléments à insérer dans le fichier de données	67
3.2.2	Résultats obtenus	67
3.2.3	Validation des résultats	68
3.2.4	Discussion et interprétation des résultats	69
3.3	Cas #2 : Réseau triphasé avec une barre PV et une PQ à puissance constante.....	70
3.3.1	Éléments à insérer dans le fichier de données	70
3.3.2	Résultats obtenus	71
3.3.3	Validation des résultats	72
3.3.4	Discussion et interprétation des résultats	73
3.4	Cas #3 : Réseau triphasé avec implémentation d'une ligne monophasée (une barre monophasée)	73
3.4.1	Éléments à insérer dans le fichier de données	74
3.4.2	Résultats obtenus	74
3.4.3	Cas #3.2 : Réseau triphasé avec implémentation d'une ligne monophasée (80 barres monophasées).....	76
3.4.4	Éléments à insérer dans le fichier de données	77
3.4.5	Résultats obtenus	78
3.4.6	Validation des résultats	78
3.4.7	Discussion et interprétation des résultats	79
3.5	Cas #4 : Deux barres triphasées avec une barre monophasée munie d'un transformateur changeur de prise (augmentation de prise).....	80
3.5.1	Éléments à insérer dans le fichier de rapport	80
3.5.2	Résultats obtenus	81
3.5.3	Validation des résultats	83
3.5.4	Discussion et interprétation des résultats	84
3.6	Cas #5 : Réseau triphasé composé d'une barre de charge débalancée	84
3.6.1	Éléments à insérer dans le fichier de données	85
3.6.2	Résultats obtenus	85
3.6.3	Validation des résultats	87
3.6.4	Discussion et interprétation des résultats	87
3.7	Cas #6 : Plusieurs barres triphasées avec un transformateur Yy0 qui contrôle une des barres de charge dans le réseau (diminution de prise).....	88
3.7.1	Éléments à insérer dans le fichier de données	88
3.7.2	Résultats obtenus	89
3.7.3	Validation des résultats	91
3.7.4	Discussion et interprétation des résultats	92
3.8	Cas #7 : Réseau triphasé complet mal conditionné (3 000 barres triphasées).....	93
3.8.1	Éléments à insérer dans le fichier de données avec V_I proche de V_F	93
3.8.2	Résultats obtenus	94
3.8.3	Éléments à insérer dans le fichier de données avec V_I loin de V_F	96
3.8.4	Résultats obtenus	97
3.8.5	Validation des résultats	98
3.8.6	Discussion et interprétation des résultats	98
3.9	Discussion	99

CONCLUSION.....	101
RECOMMANDATIONS	103
ANNEXE I GAUSS-SEIDEL POUR 1 CAS MONOPHASÉ.....	105
ANNEXE II ÉTUDE DE CAS 1	107
ANNEXE III ÉTUDE DE CAS 2	109
ANNEXE IV ÉTUDE DE CAS 3	111
ANNEXE V ÉTUDE DE CAS 4	115
ANNEXE VI ÉTUDE DE CAS 5	117
ANNEXE VII ÉTUDE DE CAS 6	119
ANNEXE VIII ÉTUDE DE CAS 7	123
BIBLIOGRAPHIE.....	125

LISTE DES TABLEAUX

	Page	
Tableau 2.1	Paramètres connus et à déterminer du simulateur.....	55
Tableau 3.1	Itérations du prototype (Cas #1)	67
Tableau 3.2	Itérations du simulateur (Cas #1).....	68
Tableau 3.3	Valeurs des tensions (Cas #1).....	68
Tableau 3.4	Itérations du prototype (Cas #2)	71
Tableau 3.5	Itérations du simulateur (Cas #2).....	71
Tableau 3.6	Valeurs des tensions (Cas #2)	72
Tableau 3.7	Itérations du prototype (Cas #3)	75
Tableau 3.8	Itérations du simulateur (Cas #3)	75
Tableau 3.9	Valeurs des tensions (une barre monophasée) (Cas #3)	76
Tableau 3.10	Valeurs des tensions (80 barres monophasées) (Cas #3)	78
Tableau 3.11	Itérations du prototype (Cas #4)	81
Tableau 3.12	Itérations du simulateur (Cas #4)	82
Tableau 3.13	Changement de prises du transformateur (Cas #4)	82
Tableau 3.14	Valeur des tensions (Cas #4).....	83
Tableau 3.15	Itérations du prototype (Cas #5)	86
Tableau 3.16	Itérations du simulateur (Cas #5).....	86
Tableau 3.17	Résultats des tensions (Cas #5).....	86
Tableau 3.18	Itérations du prototype (Cas #6)	89
Tableau 3.19	Itérations du simulateur (Cas #6)	89
Tableau 3.20	Changement de prises du transformateur (Cas #5)	90
Tableau 3.21	Résultats des tensions (Cas #6).....	91

Tableau 3.22	Itérations du prototype quand V_I est proche de V_F (Cas #7)	94
Tableau 3.23	Itérations du simulateur quand V_I est proche de V_F (Cas #7).....	94
Tableau 3.24	Résultats des tensions quand V_I est proche de V_F (Cas #7).....	95
Tableau 3.25	Itérations du prototype quand V_I est loin de V_F (Cas #7)	97
Tableau 3.26	Itérations du simulateur quand V_I est loin de V_F (Cas #7).....	97
Tableau 3.27	Résultats des tensions quand V_I est loin de V_F (Cas #7)	98

LISTE DES FIGURES

	Page	
Figure 1.1	Schéma représentant un réseau avec une barre SW, une barre PQ et une barre PV.....	6
Figure 2.1	Schéma simplifié d'une ligne.....	23
Figure 2.2	Circuit équivalent d'un transformateur monophasé.....	27
Figure 2.3	Prise d'un transformateur.....	28
Figure 2.4	Circuit équivalent d'un transformateur Yd0	30
Figure 2.5	Pertes dans une branche quand $Z_1 = 0$	33
Figure 2.6	Pertes dans une branche d'un transformateur	34
Figure 2.7	Circuit équivalent d'un transformateur Yy0	37
Figure 2.8	Pertes dans une branche d'un transformateur	39
Figure 2.9	Schéma d'un réseau monophasé simple	42
Figure 2.10	Circuit équivalent d'une barre PV vers une barre PQ.....	44
Figure 2.11	Circuit simplifié d'une barre PV vers une barre PQ	45
Figure 2.12	Circuit simplifié d'une barre PQ vers une barre PV	47
Figure 2.13	Circuit simplifié d'une barre PV vers une barre PV	48
Figure 3.1	Réseau avec deux barres PQ en série (Cas #1)	66
Figure 3.2	Réseau triphasé avec une barre PV et une PQ à puissance constante (Cas #2).....	70
Figure 3.3	Réseau triphasé avec une barre PQ monophasé (Cas #3)	74
Figure 3.4	Réseau avec x barres monophasées (Cas #3).....	77
Figure 3.5	Deux barres triphasées avec une barre monophasée munie d'un transformateur changeur de prise (Cas #4)	80
Figure 3.6	Réseau ayant une barre de charge débalancée (Cas #5)	85

Figure 3.7	Réseau triphasé avec un transformateur Yy0 (Cas #6)	88
Figure 3.8	Schéma tronqué d'un réseau triphasé composé de 3 000 barres (Cas #7).....	93
Figure 3.9	Diagramme radar pour l'ensemble des résultats pour le cas #7.....	96

LISTE DES SYMBOLES ET UNITÉS DE MESURE

pu	pas d'unité
SW	barre d'équilibre
PV	barre de génération
PQ	barre de charge
I	intensité de courant
A	ampère
V	volt
W	watt
P	puissance active
Q	puissance réactive
S	puissance apparente
Hz	hertz
Yy0	transformateur y-y avec 0 degré de déphasage
Yd0	transformateur y-d avec 0 degré de déphasage
a ₁	prise au primaire d'un transformateur
a ₂	prise au secondaire d'un transformateur
BCNO	numéro de barre contrôlée
Vcont	tension contrôlée

INTRODUCTION

L'électricité est partout autour de nous. Que ce soit à la maison, au travail, ou même dans un véhicule, cette ressource est indispensable au fonctionnement de notre société. Cela dit, personne ne peut déterminer comment l'ensemble de la population gère son électricité. Par exemple, même si la plupart du temps la majorité de la population prend une douche le matin, ce n'est pas tout le monde qui la prend à la même heure. Cela cause une fluctuation sur les lignes de tension, et oblige les chercheurs à utiliser des outils afin de calculer comment transite la puissance dans l'ensemble du réseau. L'analyse des écoulements de puissances ayant des conditions de départs différentes est donc essentielle pour faire une évaluation réelle de la fluctuation des tensions et des puissances par rapport au temps. En d'autres mots, l'écoulement de puissance est nécessaire pour faire l'évaluation de la réaction d'un réseau vis-à-vis certaines contraintes ou conditions afin de voir comment les puissances et les tensions changent dans un réseau. C'est pour cette raison que l'objectif de ce mémoire est d'ajouter un outil de travail pouvant calculer l'écoulement de puissance d'un réseau électrique triphasé. Cela permettra aux chercheurs de mieux déterminer comment cette puissance est distribuée dans un réseau électrique à un moment donné.

Ce mémoire est séparé en trois grandes parties. Il débute par une revue de littérature au sujet des différentes méthodes connues étant capables de résoudre un écoulement de puissance. La deuxième partie porte sur la méthodologie complète à ce qui a trait à la compréhension du simulateur. Les sous-sections qui s'y retrouvent sont les suivantes :

- la méthode employée pour construire les fichiers de données ;
- l'explication de l'installation des librairies permettant de changer le code source ;
- la modélisation des composantes introduites dans le simulateur ;
- la construction de la matrice d'admittance d'un réseau monophasé ou triphasé, ainsi que sa réduction afin d'augmenter la vitesse du simulateur ;
- l'explication des méthodes implémentées ;
- l'algorithme utilisé afin de faire de la régulation de tension ;

- les explications nécessaires afin de bien comprendre les rapports qui sont obtenues après chaque simulation.

La dernière partie de ce mémoire présente l'étude de plusieurs cas afin de vérifier la fiabilité du simulateur. Les différents éléments à insérer dans le fichier de rapport sont présentés pour chacun des cas étudiés. Aussi présentés pour chaque cas sont les résultats obtenus, leurs validations, ainsi qu'une discussion visant à observer les résultats et en tirer des conclusions. À la fin des cas se trouve une discussion plus approfondie permettant de tirer des conclusions sur les résultats obtenus par rapport à la problématique, ainsi que par rapport à la revue de littérature.

CHAPITRE 1

REVUE DE LITTÉRATURE

1.1 Problématique

Un réseau électrique est composé de trois sous-sections. Il y a la production, le transport et la distribution vers les charges. Plusieurs simulateurs pouvant calculer l'écoulement de puissance de réseaux électriques de grande envergure (8 500 nœuds ou 3 000 barres, par exemple) existent actuellement, mais aucun d'entre eux n'est capable de résoudre un système (que ce soit avec la méthode à points fixe, la méthode Newton-Raphson, etc.) lorsqu'il y a de grands réseaux de transport jumelés avec des réseaux de distribution quand l'indice de conditionnement se rapproche de l'infini. Cela est dû au fait que la différence entre les impédances d'un réseau de distribution vis-à-vis celles d'un réseau de transport est très grande. Donc, plus l'indice est élevé, plus il est difficile pour les simulateurs de résoudre l'écoulement de puissance. Cela vient donc poser la problématique du projet : créer un simulateur pouvant calculer l'écoulement de puissance d'un réseau électrique triphasé complet où il y a un indice de conditionnement élevé.

1.2 Objectifs

Cinq objectifs à accomplir ont été identifiés. Premièrement, il faut être capable de faire le lien entre le réseau de transport triphasé et le réseau de distribution, qu'il soit triphasé ou monophasé. Ensuite, il faut tenir compte des déséquilibres de la charge. Il faut aussi effectuer les ajustements et les contrôles nécessaires que l'on retrouve normalement dans un réseau électrique, par exemple, les transformateurs changeurs de prises, les transformateurs étoile-étoile modélisés avec les impédances de mise à la terre et les barres de génération. Finalement, une fois tous ces éléments inclus dans le simulateur, il faut être capable de faire l'étude des différents types de charges, que ce soit les charges à impédances constantes, les charges à puissances constantes, les charges à courants constants, ou les charges ayant

comme facteur NP ou NQ, et de créer un rapport avec le logiciel afin de voir comment le programme réagit vis-à-vis différents réseaux et comment il gère la régulation.

1.3 Revue de littérature

Créer un simulateur fait partie du travail à accomplir pour ce mémoire, il faut donc être capable de le créer dans un langage qui est optimal pour l'IREQ, afin que l'équipe de chercheurs ait un nouvel outil lui permettant de pousser un peu plus loin ses recherches. Pour ce qui a trait aux outils disponibles permettant de faire l'écoulement de puissance, plusieurs méthodes mathématiques ont déjà été développées, et de nombreux articles ont été écrits à leurs sujets. C'est pourquoi la revue de littérature est séparée en deux sections. La première section est la différence de vitesse d'exécution entre différents langages de programmation et la seconde est tout ce qui a trait aux méthodes étudiées permettant de faire l'écoulement de puissance d'un réseau électrique.

1.3.1 Différence entre C++, Java, Go et Scala pour un même programme

Selon un article écrit par Robert Hundt en 2011, la différence entre les langages de programmation se fait en se basant sur plusieurs critères. Il faut donc regarder non seulement la vitesse d'exécution, mais aussi le nombre de lignes de codes, l'espace mémoire utilisé, et la vitesse de compilation. Cela dit, tous les langages de programmation ont leurs forces et leurs faiblesses. Dans cet article, le chercheur a fait la comparaison entre quatre langages soit le C++, le Java, le Go et le Scala sur tous les critères énumérés précédemment. En regardant premièrement le nombre de lignes de code, le langage qui se démarque des autres est le Scala. Puisqu'il est beaucoup plus compact que les autres langages pour l'écriture du code, l'écriture de celui-ci prend beaucoup moins de temps. Deuxièmement, pour ce qui a trait à l'espace mémoire utilisé (RAM), le langage qui se démarque est le C++ optimisé. Ce langage est d'autant plus intéressant pour un écoulement de puissance, car les opérations mathématiques sont effectuées sur de grandes matrices, et lorsqu'un programme exécute de nombreuses opérations, il a besoin d'avoir beaucoup d'espace mémoire. Troisièmement, il y a la vitesse de compilation. Le langage qui est le plus rapide dans ce secteur est Go. Même si

les autres langages sont un peu plus lents, ce n'est pas le critère le plus important pour ce qui est du choix du langage. Par contre, le dernier critère est la vitesse d'exécution. Le langage qui se distingue des autres est le C++ compilé. Selon cet article, lorsqu'un programme nécessite d'avoir un temps d'exécution extrêmement rapide, le C++ compilé est sans aucun doute le meilleur choix à prendre. Cependant, d'autres choix de langages compilés comme le Fortran ou le Julia n'ont pas été abordés dans cet article. Ils auraient pu être évalués afin de voir quel est le langage qui se démarque le plus en terme de rapidité d'exécution et de temps de développement.

1.3.2 Écoulement de puissance

Les écoulements de puissances sont utilisés afin d'observer les tensions et puissances à chacune des barres et ceci à l'aide de simulations permettant d'en comparer les résultats avec les tensions et puissances obtenues en temps réel lorsqu'il y a des changements ou des ajustements dans le réseau (perte de lignes, changement brusque de charges, etc.). Ils sont donc utilisés pour voir la réaction d'un réseau électrique lorsque celui-ci est en régime stable et permanent. Ceci permettra alors de prévoir comment gérer des situations où les conditions dans celui-ci changent. Cette étude de réseau aide les ingénieurs à prévoir comment celui-ci réagira vis-à-vis différentes situations qu'ils pourraient affronter. Pour ce faire, il faut utiliser différents outils mathématiques permettant de calculer l'écoulement de puissance, et ce, dans toutes les situations possibles.

1.3.2.1 Écoulement de puissance optimale (OPF)

Une des façons d'étudier le comportement des réseaux électriques non linéaires est de calculer l'écoulement de puissance optimale ou OPF. Selon un article écrit par Stephen Frank et coll. en 2012, cette méthode pour calculer l'écoulement de puissance est de plus en plus étudiée et permet d'avoir des taux de convergence excessivement élevés. Cette façon de faire consiste à trouver les points d'opérations de l'ensemble des barres d'un réseau, quelles que soient les contraintes du système, ainsi que ses limitations en contrôle. De plus, les points d'opération doivent être continuellement modifiés afin d'intégrer les barres de générations

variables (éoliennes, panneaux solaires, etc.). Dans cet article, les auteurs décrivent différentes méthodes en parlant des avantages et des inconvénients des différentes solutions analysées.

Cela dit, le plus grand problème d'utilisation de cette méthode est sa vitesse de convergence. Cette vitesse est très importante, car il faut, par exemple, faire plusieurs écoulements de puissance pour avoir une idée de la fluctuation du réseau dans une journée complète. Si le processus prend trop de temps pour un instant donné, il sera donc inconcevable de l'utiliser pour observer la fluctuation du réseau pendant une certaine période, car cela prendrait trop de temps. Il est cependant quasi inimaginable qu'un simulateur utilisant cette méthode ait une convergence qui se fait dans un temps raisonnable pour un réseau de très grande dimension. C'est pourquoi la majorité des simulateurs utilisent les méthodes itératives.

1.3.2.2 Écoulement de puissance avec Gauss-Seidel

Plusieurs méthodes ayant des convergences différentes ont été conçues pour faciliter les calculs de l'écoulement de puissance d'un réseau électrique. Une méthode itérative efficace pour des cas simples est la méthode Gauss-Seidel (Pierre-Jean Lagacé, 2011).

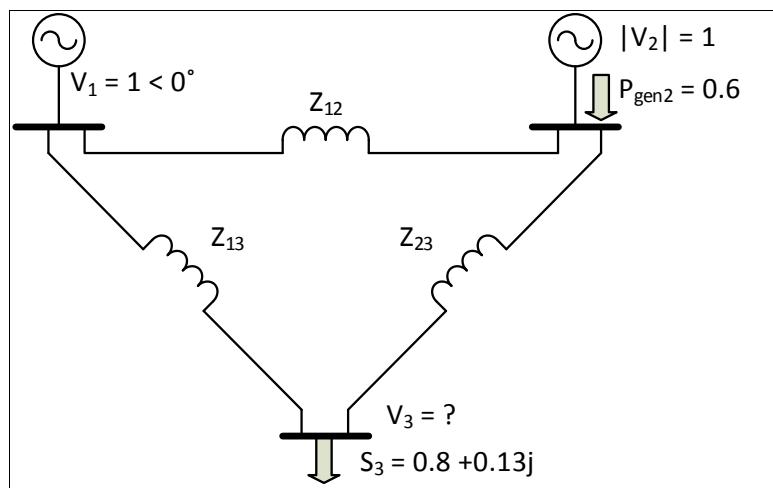


Figure 1.1 Schéma représentant un réseau avec une barre SW, une barre PQ et une barre PV

Avec le schéma précédent, la procédure qu'il faut suivre afin de calculer la matrice d'admittance est la suivante. Premièrement, étant donné que c'est un réseau monophasé, la matrice totale sera une matrice 3x3. Deuxièmement, pour la première case de la matrice (Y_{11}), il faut regarder les impédances qui sont connectées à la barre numéro un. Sachant qu'il y a simplement deux impédances de lignes, et que l'admittance est le contraire de l'impédance, il faut additionner l'inverse de ces deux composantes. Ensuite, pour la case Y_{12} de la matrice, qui représente les éléments se trouvant sur la ligne entre la barre un et la barre deux, par convention, la valeur est égale à la négation de l'inverse de l'impédance ($-1/Z_{12}$). Donc, avec le schéma précédent, si les réactances de lignes sont toutes égales, il est possible de dire que $Y_{11} = Y_{22} = Y_{33}$, et $Y_{12} = Y_{13} = Y_{23}$. De plus, Y_{11} est égal à $1/Z_{12}+1/Z_{13}$ et $Y_{12} = -1/Z_{12}$. Les contraintes de la figure 1.1 sont les puissances de la barre de charge et la puissance active générée par la barre de génération.

À partir des admittances trouvées, la méthode consiste à faire l'écoulement de puissance en calculant les tensions à chacune des barres de façon itérative. De plus, cette méthode est tirée de la méthode de Jacobi qui utilise les tensions obtenues lors de la dernière itération « k » afin d'estimer les tensions à la prochaine itération « $k+1$ ». Voici les équations qui régissent les tensions et puissances pour chaque barre en utilisant la méthode Gauss-Seidel (Pierre-Jean Lagacé, 2011).

$$V_1^{k+1} = \frac{1}{Y_{11}} * \left(\left(\frac{S_1^k}{V_1^k} \right)^* - Y_{12}V_2^k - Y_{13}V_3^k \right) \quad (1.1)$$

$$V_2^{k+1} = \frac{1}{Y_{22}} * \left(\left(\frac{S_2^k}{V_2^k} \right)^* - Y_{21}V_1^k - Y_{23}V_3^k \right) \quad (1.2)$$

$$V_3^{k+1} = \frac{1}{Y_{33}} * \left(\left(\frac{S_3^k}{V_3^k} \right)^* - Y_{32}V_2^k - Y_{31}V_1^k \right) \quad (1.3)$$

Les équations permettant de trouver la puissance à la barre #1 en utilisant la méthode Gauss-Seidel (Pierre-Jean Lagacé, 2011) sont :

$$S_1 = S_{gen1} - S_{Ybus1} - S_{chg1} \quad (1.4)$$

Et,

$$S_{Ybus1} = V_1 * (Y_{11}V_1 + Y_{12}V_2 + Y_{13}V_3)^* \quad (1.5)$$

La même procédure peut être appliquée pour les barres 2 et 3. Une fois cela fait, il est possible de trouver les différentes tensions à partir d'une simple itération, mais seulement si les valeurs de départs pour la barre de génération et pour la barre de charge sont proches des valeurs finales. Pour une barre de génération, la valeur du module de tension et la puissance active générée sont connues, et les deux inconnues sont le déphasage et la puissance réactive fournie par le générateur. Pour la barre de charge, les puissances sont normalement constantes et connues, et c'est pourquoi les variables inconnues sont le module et la phase de la tension. Connaissant les variables inconnues du réseau, il est possible de faire plusieurs itérations en modifiant ces variables tout en prenant soin de ne pas modifier les valeurs connues.

Un exemple démontrant cette méthode programmée en Matlab, ainsi que le fichier de données nécessaire avec une comparaison des résultats obtenus avec le simulateur utilisant la méthode Levenberg-Marquardt (*Voir ANNEXE I, p.105*).

1.3.2.3 Écoulement de puissance NR modifié pour l'intégration éolienne

Les éoliennes sont de plus en plus répandues dans le monde. C'est une solution très viable d'un point de vue économique. Par contre, l'introduction de ce type d'équipement entraîne des changements dans le réseau, que ce soit par rapport à la stabilité des tensions du réseau complet ou par rapport à ses conditions d'opérations. Selon l'article écrit par Yan Li et coll. en 2011, en utilisant la méthode itérative Newton-Raphson, il faut modifier son algorithme afin d'inclure les générateurs asynchrones, ainsi que les génératrices à doubles alimentations

(DFIG) afin d'être capable de converger lorsqu'un réseau est composé de parcs éoliens. Cela permet d'avoir des simulations qui reflètent beaucoup mieux un réseau en temps réel, car ceux-ci sont composés de plus en plus de ce type de générateurs. L'article révise donc la méthode Newton-Raphson tout en validant les résultats obtenus avec la plateforme PSASP en démontrant qu'en modifiant cette méthode, il est possible de converger même si le réseau est composé de génératrices éoliennes. Les chercheurs ont donc premièrement décrit le fonctionnement des générateurs asynchrones, et des générateurs à double alimentation (DFIG) en régime permanent. Ensuite, ils ont modifié les équations calculant les puissances du réseau en révisant la matrice Jacobienne. Finalement, ils ont démontré que le changement de la méthode Newton-Raphson donne de meilleures convergences permettant de calculer l'écoulement de puissances de grands réseaux composés de parcs éoliens.

1.3.2.4 Écoulement de puissance à l'aide d'une combinaison de la méthode Newton-Raphson et de la méthode d'optimisation de Newton

Dans l'article d'Andrey Pazderin et Sergey Yuferev en 2009, les chercheurs montrent à l'aide de la non-linéarité du système et de la grosseur des matrices pourquoi la complexité du réseau électrique vient empêcher la convergence, quelle que soit la méthode itérative utilisée. C'est pourquoi ils ont étudié la méthode d'optimisation de Newton avec la méthode itérative de Newton-Raphson permettant d'obtenir de meilleurs résultats, c'est-à-dire au niveau de la convergence des valeurs finales. Selon les chercheurs, l'utilisation de l'optimisation de Newton avec la méthode Newton-Raphson permet de converger même si les valeurs initiales sont trop loin des valeurs finales pour que la méthode Newton-Raphson puisse converger par elle-même. Cela vient évidemment changer l'algorithme de la méthode itérative, mais, en plus de la création de la matrice Jacobienne, il y a aussi la création d'une matrice Hessienne. Grâce à la combinaison de ces matrices lors des itérations, le fait que le déterminant de la matrice Hessienne soit plus grand que zéro, mais plus petit que celui de la matrice Jacobienne permet de converger naturellement vers un point d'opération fonctionnel, et cela même si les valeurs initiales des tensions sont trop loin pour que la méthode NR puisse converger par elle-même. De plus, les chercheurs ont émis la possibilité d'utiliser une matrice avec des coefficients de pondération permettant de faire un ajustement aux points

d'opérations lorsque ces points se trouvent à l'extérieur d'un domaine de faisabilité. Cela permet de résoudre des écoulements de puissance de réseaux qui ne pourraient être résolus en utilisant la méthode NR seule.

1.3.2.5 Améliorer la convergence d'un écoulement de puissance par la méthode de Levenberg-Marquardt

Dans l'article écrit par P.J. Lagacé et coll. en 2008, de nombreuses méthodes itératives sont expliquées et chacune d'entre elles a des avantages et des inconvénients. Premièrement, ils expliquent que le processus itératif est indispensable lorsqu'il s'agit de résoudre un écoulement de puissance, car la majorité des cas sont composés d'équations non linéaires. De plus, ils expliquent que la majorité des méthodes itératives sont inspirées de la méthode Gauss-Seidel, qui a été créée pour trouver les tensions de chacune des barres dans un réseau distinct. Lorsqu'il s'agit d'écoulement de puissance, les méthodes itératives servent à minimiser l'écart entre les erreurs résiduelles. L'article explique la différence entre les niveaux de convergence de ces méthodes et montre quels niveaux sont les plus susceptibles de converger dans un réseau mal conditionné.

Tout d'abord, ils démontrent les modèles mathématiques employés dans un système d'équations. La première méthode étudiée est aussi la plus connue, soit la méthode Newton-Raphson. Celle-ci est utilisée afin de résoudre un ensemble d'équations non linéaires dans la mesure où les valeurs initiales sont proches des valeurs finales. Une façon d'expliquer la logique est de commencer par donner des valeurs initiales à un vecteur X . Il faut par la suite créer une boucle, et calculer, à l'intérieur de celle-ci, une série d'erreurs résiduelles pour tenter de les réduire afin de converger les variables inconnues vers les valeurs finales. L'opérateur calculant ces erreurs doit faire de l'ordonnancement, de la factorisation, ainsi que de la substitution directe et inverse afin de résoudre l'équation. Une fois le pas de calcul calculé, il suffit d'ajouter le vecteur X au pas de calcul. Ensuite, il faut continuer à itérer jusqu'à ce que l'erreur résiduelle soit plus petite que la tolérance spécifiée. Si ce n'est pas le cas, deux situations peuvent avoir eu lieu, soit que l'erreur résiduelle est plus grande que la limite supérieure ou que le nombre d'itérations excède le nombre maximum spécifié.

L'équation principale permettant de calculer le pas de calcul est le suivant pour la méthode Newton-Raphson :

$$\Delta X = -\nabla F(X) \backslash F(X)$$

ΔX : Accroissement pour l'ensemble des variables inconnues

$\nabla F(X)$: Gradient de la fonction F , ou matrice Jacobienne

$F(X)$: Fonction F , ou matrice des variables connues

(1.6)

Le prochain modèle expliqué dans l'article utilise une boucle itérative avec un ajustement Hessian (un peu comme l'article précédent). Cet algorithme permet d'avoir une meilleure convergence que la méthode Newton-Raphson parce qu'il utilise la série de Taylor de deuxième ordre pour calculer le pas de calcul. L'équation le calculant permet de minimiser les erreurs résiduelles utilisant l'estimation d'un algorithme d'un point fixe ayant « i » itérations. Cela veut dire que pendant une itération de la boucle principale, l'équation du pas de calcul peut être répétée à l'intérieur d'une autre boucle permettant d'avoir un meilleur taux de convergence. Par contre, seul le pas de calcul est calculé pendant ces itérations. Les matrices Jacobienne et Hessienne changent seulement au début de chaque itération de la boucle principale. Suivant cette méthode, trois situations peuvent être conclues, soit les erreurs résiduelles dépassant la limite supérieure imposée et le processus diverge, soit le nombre maximal d'itérations est dépassé et le processus diverge ou soit l'amplitude des erreurs résiduelles à l'itération « i » est inférieure à la limite désirée et le processus converge. Voici l'équation du pas de calcul pour cette méthode :

$$\Delta X = -(\nabla F(X) + \frac{X^T \nabla^2 F(X)}{2}) \backslash F(X)$$
(1.7)

La troisième méthode expliquée dans l'article est la méthode Newton-Raphson utilisant des ajustements sur la matrice Jacobienne. L'équation permettant d'estimer la matrice Hessienne peut être modifiée pour faire une estimation sur la matrice Jacobienne. Cela dit, semblablement à la méthode avec ajustement de la matrice Hessienne, il faut calculer à « i »

reprises (où i est plus grand ou égal à 0) la matrice Jacobienne dans une boucle externe de la boucle principale. Il faut donc recalculer la matrice Jacobienne à « i » reprises pour chacune des itérations. Par contre, si « i » est égal à zéro, cette méthode sera exactement la même que la méthode Newton-Raphson. Voici l'équation (1.6) modifiée afin d'utiliser cette méthode.

$$\Delta X = -\nabla F(X + \frac{\Delta X}{2}) \backslash F(X) \quad (1.8)$$

La quatrième méthode expliquée par le chercheur est la méthode Levenberg-Marquardt. Cette méthode est très utilisée dans le monde de l'ingénierie, car elle utilise une approche différente en intégrant une variable permettant d'avoir un accroissement moins grand lors des itérations. Cela permet d'avoir un meilleur taux de réussite en terme de convergence par rapport aux autres méthodes étudiées précédemment. Voici l'équation représentant le pas de calcul pour minimiser les erreurs résiduelles en intégrant la série de Taylor du premier ordre :

$$\Delta X = -(\nabla F(X_0)^T \nabla F(X_0)) \backslash \nabla F(X_0)^T \varepsilon_0 \quad (1.9)$$

Cependant, cette équation est identique à la méthode Newton-Raphson et elle n'obtient pas un meilleur taux de convergence si les valeurs initiales sont éloignées de la solution. Par contre, en utilisant cette équation, il est possible d'introduire le Lagrangien lambda qui sert de contrainte supplémentaire à l'équation du pas de calcul. Cette contrainte est tout simplement la distance qu'il y a entre les valeurs finales et leurs valeurs initiales (X). Dans le cas d'un écoulement de puissance, le Lagrangien est calculé par rapport à la somme au carré des erreurs résiduelles. L'équation du nouveau pas de calcul est donc la suivante :

$$\Delta X = -(\nabla F(X_0)^T \nabla F(X_0) + \lambda I) \backslash \nabla F(X_0)^T \varepsilon_0 \quad (1.10)$$

À partir de cette équation, il est néanmoins possible de changer la valeur de lambda à chaque itération, de telle manière à avoir une valeur qui est directement reliée à la somme des erreurs résiduelles au carré. Cette méthode permet donc d'avoir un lambda très grand en début de problème afin de s'approcher rapidement et un beaucoup plus petit lambda tout au long des

itérations afin d'obtenir un meilleur taux de convergence qu'avec la méthode Newton-Raphson. Donc, pour chacune des itérations, il faut calculer lambda, et ensuite calculer le pas de calcul.

Finalement, la dernière méthode expliquée dans l'article est celle de Levenberg-Marquardt avec ajustement. Cette méthode permet d'augmenter la performance quant à la convergence du système. La méthode est pratiquement la même que la méthode de Levenberg-Marquardt sauf qu'il faut recalculer la matrice Jacobienne pour chaque itération.

Une fois toutes ces méthodes expliquées, il faut les appliquer sur l'écoulement de puissance afin de déterminer leurs convergences. Le vecteur X est composé des puissances actives et réactives du réseau sachant que ce sont les contraintes du système. Selon l'article, en appliquant des ajustements sur les matrices Hessiennes ou Jacobiennes, le taux de convergence d'un écoulement de puissance avec ajustement va jusqu'à tripler, comparativement à un écoulement sans aucun ajustement. Cependant, cela ne veut pas dire que ces méthodes trouvent des solutions pour n'importe quelle topologie de réseau électrique. Par contre, pour améliorer encore plus la convergence, il faut utiliser la méthode Levenberg-Marquardt qui minimise la somme au carré des puissances actives et réactives partout dans le réseau et qui multiplie ce facteur par une matrice identité. Sans dire que cela fonctionne pour tous les cas, les chercheurs disent qu'en ajoutant les ajustements à cette méthode, cela permet d'avoir un taux de convergence qui est raisonnable et qui peut fonctionner pour la majorité des cas.

1.3.2.6 Solution d'écoulement de puissance multi phase pour les réseaux de distribution à grand échelle utilisant l'algorithme MANA

Dans cet article (I. Kocar et coll.), les chercheurs implémentent l'analyse des nœuds augmentés et modifiés (MANA) à divers algorithmes déjà existants afin de comparer la convergence ainsi que la rapidité des simulateurs sur un réseau de distribution à grande échelle connu (8 500 nœuds). La première section de l'article introduit la formulation de l'algorithme MANA lorsqu'il est appliqué à un système d'équations non-linéaire.

$$Ax = b \quad (1.11)$$

Où « A » est le système d'équation, « b » est le vecteur des variables connus et « x » est le vecteur des variables qui sont inconnus. À partir de ce système d'équations les chercheurs développent une vision élargie de la même équation en introduisant les matrices des différents coefficients augmentés qui correspondent avec la matrice d'admittance au système correspondant.

$$\begin{bmatrix} Y_n & A_c \\ A_r & A_d \end{bmatrix} \begin{bmatrix} V_n \\ I_x \end{bmatrix} = \begin{bmatrix} I_n \\ V_x \end{bmatrix} \quad (1.12)$$

Par la suite, ils montrent comment ils calculent chacun des éléments présents dans le système, c'est-à-dire A_c (coefficients des colonnes augmentés), A_r (coefficients des rangées augmentées) et A_d (coefficients de la diagonale augmentés). Une fois cela fait, ils introduisent cet approche à plusieurs algorithmes permettant de résoudre l'écoulement de puissance connu. La première méthode avec laquelle ils ajoutent l'algorithme de MANA est la méthode faisant la résolution d'un écoulement de puissance à points fixe. Avec cette méthode, les chercheurs estiment qu'il y a convergence pour la majorité des réseaux de distributions, mais que le principal avantage est la vitesse de résolution du problème. Pour la deuxième méthode, lorsqu'ils introduisent l'approche MANA à la méthode Newton-Raphson, cette manipulation change drastiquement l'équation de la matrice Jacobienne. Les chercheurs ont indiqué que la méthode Newton avec l'approche MANA converge dans des temps assez rapides lorsque les valeurs initiales sont proches des valeurs finales. Cependant, le temps de résolution est beaucoup plus lent qu'avec la méthode précédente. De plus, le fait de factoriser à toute les itérations ralenti de beaucoup la vitesse de résolution de problème. Une autre méthode qu'ils ont vérifiée est la méthode de Newton sans faire la mise-à-jour de la matrice Jacobienne à chaque itération. Cette méthode permet d'avoir une vitesse d'exécution très rapide sans être capable d'avoir un taux de convergence aussi bon que pour la méthode Newton-Raphson. En conclusion, cet article montre la comparaison entre trois solutions capables de résoudre les écoulements de puissance avec l'algorithme de MANA.

Cet algorithme permet de résoudre des réseaux de distribution de grande taille ayant des topologies incluant les charges débalancées et des phases multiples sans avoir aucune limitation connue. De plus, l'article montre qu'il est possible d'utiliser les méthodes ensemble, car si la méthode à point fixe est incapable de converger, il est possible d'utiliser la méthode Newton pour quelques itérations, et par la suite de changer à la méthode à points fixes, car elle est beaucoup plus rapide.

CHAPITRE 2

MÉTHODOLOGIE

2.1 Introduction

Cette section sert à bien comprendre l'interprétation des fichiers de données par le simulateur, la possibilité de faire des modifications au code source, les composantes qui ont été modélisées dans le simulateur, les méthodes employées par celui-ci pour calculer l'écoulement de puissance, le fonctionnement de la régulation dans le programme, et l'interprétation du fichier de rapport. Cette section est donc subdivisée en plusieurs sous-sections, et permet au lecteur de mieux comprendre le fonctionnement du simulateur.

2.2 Construction des fichiers de données

2.2.1 Explications de base

La construction des fichiers de données est l'une des étapes les plus importantes pour visualiser l'écoulement de puissance d'un réseau électrique. Voici en quelques lignes comment il faut construire un fichier de données afin que le simulateur puisse le comprendre et faire les calculs nécessaires pour réaliser l'écoulement de puissance.

Premièrement, toutes les lignes commençant par le symbole « % » sont ignorées par le simulateur. Cet indice sert à insérer des commentaires, et ainsi faciliter la lecture des fichiers de données. La première ligne non ignorée du simulateur est donc celle commençant par « Titre1 ». Cette ligne sert seulement à indiquer le titre du fichier de données. La seconde ligne est celle débutant par « Sbase= ». Cette ligne désigne la base de la puissance du réseau étudiée. Par la suite, il faut séparer les différentes composantes afin de bien construire leurs tableaux respectifs dans le simulateur.

2.2.2 Barres monophasées et triphasées

Pour les barres d'un réseau distinct, le fichier de données doit respecter une certaine nomenclature. Il doit en effet séparer les barres monophasées des barres triphasées. Pour ce qui est des barres monophasées, le simulateur commence à enregistrer les informations seulement lorsqu'il lit la ligne suivante : « Bar1={ ». Par contre, s'il s'agit de barres triphasées, le simulateur commence à enregistrer les informations lorsqu'il lit : « Bar3={ ». Il faut bien comprendre que les données des barres monophasées et triphasées ne sont pas enregistrées dans le même tableau dans le simulateur afin de faciliter les calculs. À partir de ce moment, il enregistre chacune des données correspondantes à la liste de composantes indiquée ici même (si aucune valeur n'est associée à une variable, il faut mettre 0 dans le fichier de données) :

- Bar_no : Numéro de barre ;
- Name : Nom de la barre ;
- Type : Type de la barre (SW est 1, PV est 2 ou PQ est 3) ;
- Ebase : Tension de base de la barre (Ex. : 735 pour 735 kV) ;
- Module : Module de tension de la barre (pu) ;
- Angle : Angle en degré de déphasage de la barre. Il faut bien le spécifier s'il s'agit d'une barre monophasée (phase a = 0 ; phase b = -120 ; phase c = 120) ;
- Gsh : Conductance de la barre (pu) ;
- Bsh : Susceptance de la barre (pu) ;
- Pgen : Puissance active générée par la barre (pu) ;
- Qgen : Puissance réactive générée par la barre (pu) ;
- Pchg : Puissance active constante à la barre (pu) ;
- Qchg : Puissance réactive constante à la barre (pu) ;
- Pchg_I : Puissance active à courant constant à la barre (pu) ;
- Qchg_I : Puissance réactive à courant constant à la barre (pu) ;
- Pchg_NP : Puissance active à indice NP à la barre (pu) ;
- Qchg_NQ : Puissance réactive à indice NQ à la barre (pu) ;
- NP : Facteur constant pour la puissance active (pu) ;

- NQ : Facteur constant pour la puissance réactive (pu).

Une fois que toutes ces données ont été enregistrées pour l'ensemble des barres du réseau, le simulateur change de lignes jusqu'à ce qu'il rencontre « }; ».

2.2.3 Lignes monophasées et triphasées

Tout comme avec les barres, le simulateur enregistre les données des lignes monophasées dans un tableau différent que les données des lignes triphasées. Donc, il commence à enregistrer les données seulement lorsqu'il lit la ligne suivante : « Ligne1={ », ou « Ligne3={ ». Une fois fait, il enregistre les informations dans l'ordre suivant (si aucune valeur n'est associée à une variable, il faut mettre 0 dans le fichier de données) :

- Bran_I : Barre de départ de la ligne. Pour les lignes monophasées, il faut spécifier correctement quelle est sa phase (phase a = x ; phase b = x-1i ; phase c = x+1i) ;
- Bran_J : Barre d'arrivée de la ligne. Pour les lignes monophasées, il faut spécifier correctement quelle est sa phase (phase a = x ; phase b = x-1i ; phase c = x+1i) ;
- Bran_ckt : Numéro de circuit où se trouve la branche ;
- Bran_name : Nom de la branche ;
- Resi : Résistance de la branche (pu) ;
- Réac : Réactance de la branche (pu) ;
- Cond : Conductance de la branche (pu) ;
- Susc : Susceptance de la branche (pu).

Comme mentionné précédemment, deux tableaux distincts sont créés afin de faciliter les calculs par la suite. Lorsqu'une ligne est enregistrée, le simulateur saute à la prochaine et enregistre dans l'ordre précédent toutes les informations jusqu'à ce qu'il rencontre « }; ».

2.2.4 Transformateurs

Finalement, le fichier de données doit être fait de manière à ce que le simulateur comprenne s'il s'agit d'un transformateur YY, YD, ou monophasé. Pour ce faire, le simulateur commence à enregistrer les données des transformateurs seulement lorsqu'il lit une ligne commençant par : « TransfoYY={ », ou « TransfoYD={ » ou « Transfo1={ ». Une fois fait, il fait la lecture des composantes dans l'ordre suivant (si aucune valeur n'est associée à une variable, il faut mettre 0 dans le fichier de donnée) :

- Bran_I : Barre de départ du transformateur. Pour les lignes monophasées, il faut spécifier correctement quelle est sa phase (phase a = x ; phase b = x-1i ; phase c = x+1i) ;
- Bran_J : Barre d'arrivée du transformateur. Pour les lignes monophasées, il faut spécifier correctement quelle est sa phase (phase a = x ; phase b = x-1i ; phase c = x+1i) ;
- Bran_ckt : Numéro de circuit où se trouve le transformateur ;
- Bran_name : Nom du transformateur ;
- Z_1 : Impédance Z_1 du transformateur (pu) ;
- Z_2 : Impédance Z_2 du transformateur (pu) ;
- Y = Susceptance du transformateur (pu) ;
- Z_N : Impédance Z_N connectée à la terre d'un transformateur (pu) ;
- $a_1\text{min}$: Valeur minimale de a_1 lorsque la prise est égale à 0 ou au maximum ;
- a_1 : Valeur de transformation au primaire du transformateur ;
- $a_1\text{max}$: Valeur maximale de a_1 lorsque la prise est égale à 0 ou au maximum ;
- a_2 : Valeur de transformation au secondaire du transformateur ;
- Nprise : Nombre de prises que comporte le transformateur changeur de prise ;
- BCNO : Barre contrôlée par le transformateur changeur de prise ;
- $V\text{min}$: Valeur de tension minimale de la barre contrôlée (pu) ;
- $V\text{max}$: Valeur de tension maximale de la barre contrôlée (pu).

Pour les transformateurs triphasés, il y a une variable de plus à enregistrer que pour les transformateurs monophasés (Z_N), car il n'y a pas d'impédance de mise à la terre pour les transformateurs triphasés. Il est donc primordial que le simulateur soit composé de deux

tableaux différents pour ces deux types. Une fois toutes les données enregistrées, le simulateur saute à la ligne suivante jusqu'à ce que celle-ci soit composée de : « }; ».

2.2.5 Très important

Lors de la construction du fichier de données, chacune des valeurs doit être immédiatement suivie d'une virgule, car le simulateur regarde tout de suite avant celle-ci afin de voir s'il s'agit d'un nombre imaginaire. S'il s'agit de barres ou branches monophasées, c'est encore plus important de mettre la virgule (Ex. : 5-1i,) (*Voir ANNEXE III, p.109*).

2.3 Installation des librairies

2.3.1 Étapes pour l'installation des librairies

Voici les étapes à suivre pour installer toutes les librairies, et ainsi pouvoir recréer un fichier exécutable à partir du fichier « Makefile » fourni. Il faut premièrement installer le logiciel Cygwin et ajouter dans l'onglet « Devel », l'ensemble des librairies commençant par « make », « cmake », et « gcc ». Il faut ensuite installer la librairie Eigen (facilement repérable en ligne), et changer le chemin indiqué dans les fichiers « Fonction.h » et « Makefile ». Par la suite, il faut installer la librairie SuiteSparse et indiquer les dossiers correspondant dans le fichier de construction. Dans le répertoire de configuration de SuiteSparse, il faut mettre en commentaire la ligne 159 du fichier « point h » afin de pouvoir créer la librairie. Voici l'ensemble des répertoires où il faut créer les librairies avec la commande « make » :

- \SuiteSparse\SuiteSparse_config\
- \SuiteSparse\AMD\
- \SuiteSparse\KLU\
- \SuiteSparse\CXSparse\

2.3.2 Versions installées

Lors de la conception du simulateur, la version utilisée de SuiteSparse est la version 4.4.4. Les versions antérieures n'ont pas été testées avec les étapes expliquées dans la section 2.3.1.

2.4 Modélisation des composantes

2.4.1 Explication

Calculer la matrice d'admittance est une étape qui est nécessaire lorsqu'il faut calculer l'écoulement de puissance. Que ce soit avec la méthode de Newton-Raphson ou Levenberg-Marquardt, il est nécessaire de calculer cette matrice afin de trouver les courants dans tous les nœuds du réseau. Une fois cette matrice calculée, il est possible de calculer les courants dans l'ensemble du réseau à partir de l'équation suivante :

$$I = YV \quad (2.1)$$

Cependant, avant de trouver la matrice d'admittance du réseau complet, il faut trouver les matrices d'admittances de l'ensemble des composantes qui le constitue. C'est pourquoi ce chapitre explique comment modéliser les composantes afin de calculer la matrice d'admittance complète.

2.4.2 Lignes

2.4.2.1 Lignes moyennes (80 km <= l <= 400km)

Une des caractéristiques principales des lignes moyennes est que leurs phases sont complètement indépendantes l'une par rapport à l'autre. C'est donc beaucoup plus facile de modéliser cette composante que ce soit pour un réseau monophasé ou un réseau triphasé. Cela dit, l'impédance mutuelle entre les conducteurs n'a pas été prise en considération, ce qui fait qu'il n'y a pas de couplage entre les phases lorsqu'il s'agit d'un réseau triphasé. Voici

maintenant un schéma représentant une ligne moyenne, c'est-à-dire une ligne dont la distance d'une barre à l'autre est entre 80 et 400 kilomètres. Il faut donc trouver la matrice d'admittance pour qu'elle réponde à l'équation (2.1).

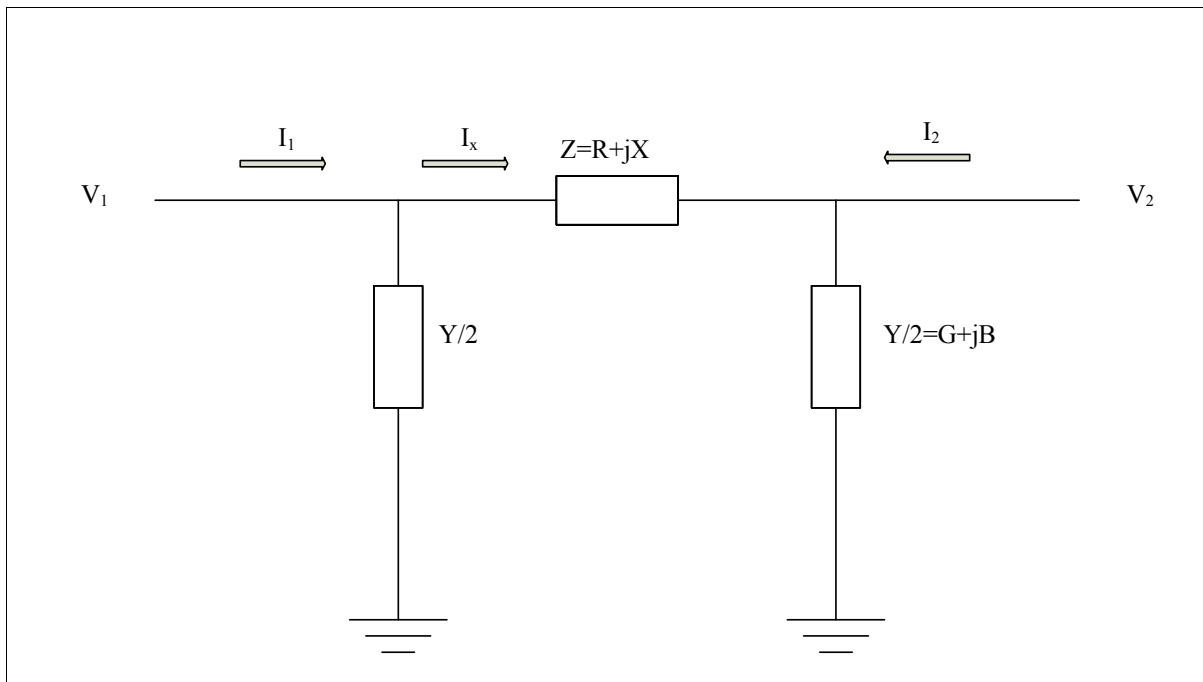


Figure 2.1 Schéma simplifié d'une ligne

Premièrement, il faut appliquer les lois de Kirchhoff des nœuds et des mailles sur la Figure 2.1.

$$V_1 = \left(I_1 - \frac{YV_1}{2} \right) Z + V_2 \quad (2.2)$$

Par la suite, il faut simplifier l'équation, ce qui donne :

$$V_1 = ZI_1 - Z \frac{YV_1}{2} + V_2 \quad (2.3)$$

$$V_1 + Z \frac{YV_1}{2} - V_2 = ZI_1 \quad (2.4)$$

Avec l'équation (2.4), il faut faire une substitution pour avoir chaque variable qu'une seule fois dans l'équation.

$$ZI_1 = \left(1 + \frac{ZY}{2}\right)V_1 - V_2 \quad (2.5)$$

Finalement, il faut simplement diviser la partie de droite par Z pour trouver I_1 , ce qui donne :

$$I_1 = \left(\frac{1}{Z} + \frac{Y}{2}\right)V_1 - \frac{V_2}{Z} \quad (2.6)$$

À partir de cette équation, il est possible de déduire Y_{11} et Y_{12} pour la matrice d'admittance de cette composante. De plus, Y_{11} est égal à Y_{22} , et Y_{12} est égal à Y_{21} . Cela veut donc dire que la matrice d'admittance est la suivante ($a = (1/Z+Y/2)$, $b = -1/Z$) :

$$Y_{bus}(2x2) = \begin{bmatrix} a & b \\ b & a \end{bmatrix} \quad (2.7)$$

Sachant que les phases n'ont aucun impact l'une par rapport à l'autre pour les lignes triphasées, voici la matrice d'admittance pour ce type de ligne :

$$Y_{bus}(6x6) = \begin{bmatrix} a & 0 & 0 & b & 0 & 0 \\ 0 & a & 0 & 0 & b & 0 \\ 0 & 0 & a & 0 & 0 & b \\ b & 0 & 0 & a & 0 & 0 \\ 0 & b & 0 & 0 & a & 0 \\ 0 & 0 & b & 0 & 0 & a \end{bmatrix} \quad (2.8)$$

2.4.2.2 Lignes longues

Il y a plusieurs différences entre les lignes moyennes et les lignes longues. Premièrement, il faut prendre en considération la longueur de la ligne, ainsi que son impédance

caractéristique, car il y a des pertes dans celle-ci. Ensuite, pour les lignes triphasées (bien que la modélisation des lignes triphasées n'ait pas été développée dans ce mémoire), les phases ont une interaction les unes par rapport aux autres. L'équation (2.8) sera donc modifiée de beaucoup, car il n'y aura aucune case nulle dans la matrice. Voici comment procéder à ce qui a trait à la modélisation des lignes longues monophasées.

Premièrement, il faut calculer les deux variables Z_c et γ qui sont nécessaires pour la modélisation.

$$Z_c = \sqrt{\frac{Z}{Y}} \quad (2.9)$$

$$\gamma = \sqrt{ZY} \quad (2.10)$$

À partir des équations (2.9) et (2.10), il est possible de trouver les variables A, B, C et D. Ces variables sont utilisées afin de trouver la matrice de transmission en monophasé d'un réseau ayant des lignes longues (l est la longueur de la ligne). Voici les équations pour trouver V_1 et I_1 de la Figure 2.1 à partir de la matrice de transmission :

$$V_1 = AV_2 + BI_2 \quad (2.11)$$

$$I_1 = CV_2 + DI_2 \quad (2.12)$$

Dans ce sens, voici les équations pour trouver les variables A, B, C, et D.

$$A = \cosh(\gamma l) \quad (2.13)$$

$$B = Z_c \sinh(\gamma l) \quad (2.14)$$

$$C = \frac{\sinh(\gamma l)}{Z_c} \quad (2.15)$$

$$D = \cosh(\gamma l) \quad (2.16)$$

À partir de ces quatre équations, il est possible de déterminer la matrice d'admittance pour une ligne longue monophasée en faisant quelques opérations mathématiques. À noter que si la ligne longue est une ligne sans perte, le résultat de la matrice d'admittance sera le même que pour une ligne moyenne. Voici les équations permettant de déterminer les quatre composantes de la matrice d'admittance à partir des composantes de la matrice de transmission :

$$Y_{11} = D/B \quad (2.17)$$

$$Y_{12} = (BC - AD)/B \quad (2.18)$$

$$Y_{21} = -1/B \quad (2.19)$$

$$Y_{22} = A/B \quad (2.20)$$

La matrice d'admittance d'une ligne longue monophasée est donc :

$$Y_{bus}(2x2) = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \quad (2.21)$$

Par contre, ce type de ligne n'a pas été inclus dans le simulateur que ce soit pour le monophasé ou le triphasé.

2.4.3 Transformateurs

Il y a plusieurs types de transformateurs modélisés dans le simulateur. Il y a le transformateur monophasé, le transformateur YY à 0 degré de déphasage et le transformateur YD à 0 degré de déphasage. Évidemment, il est possible de rajouter des transformateurs dans le simulateur, mais pour l'instant, celui-ci calcule l'écoulement de puissance d'un réseau qu'avec ces trois types. De plus, l'ajout de la régulation de tension avec ces trois types de transformateurs est expliqué un peu plus bas.

2.4.4 Transformateur monophasé

Voici comment le transformateur monophasé est modélisé :

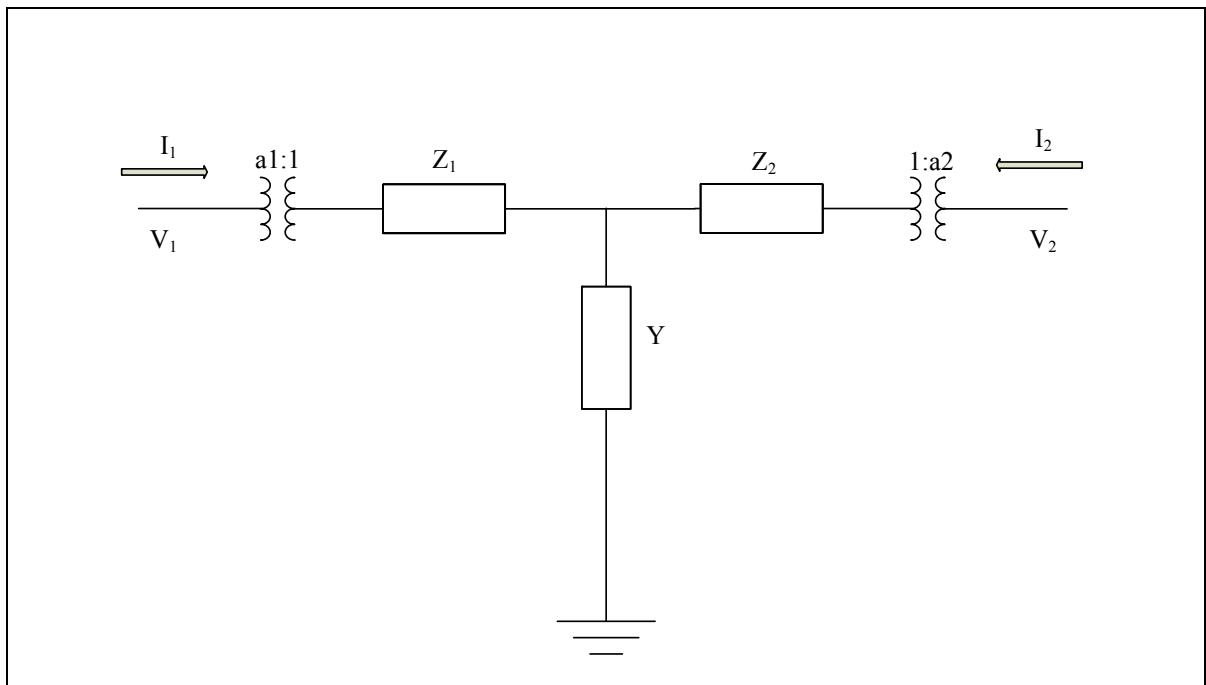


Figure 2.2 Circuit équivalent d'un transformateur monophasé

La première étape pour modéliser ce transformateur est de trouver les matrices de transmission de chacune des composantes, et par la suite, les multiplier puisqu'ils sont tous en série.

Tout d'abord, voici le premier item à modéliser :

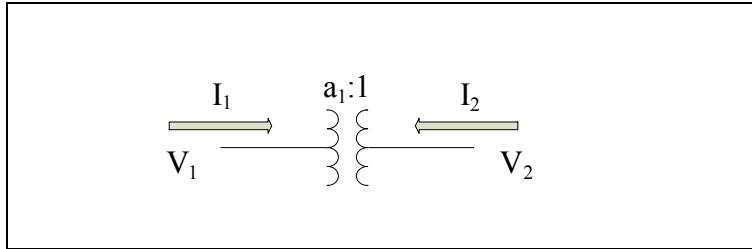


Figure 2.3 Prise d'un transformateur

En utilisant les équations suivantes, il est facile de trouver la matrice de transmission, car :

$$V_1 = a_1 V_2 \quad (2.22)$$

$$I_1 = (1/a_1) I_2 \quad (2.23)$$

Donc,

$$T_1 \text{ et } T_5 = \begin{bmatrix} a_1 & 0 \\ 0 & \frac{1}{a_1^*} \end{bmatrix} \quad (2.24)$$

Pour les deux impédances séries Z_1 et Z_2 , on retrouve :

$$T_2 \text{ et } T_4 = \begin{bmatrix} 1 & Z \\ 0 & 1 \end{bmatrix} \quad (2.25)$$

Pour l'élément shunt Y , on retrouve :

$$T_3 = \begin{bmatrix} 1 & 0 \\ Y & 1 \end{bmatrix} \quad (2.26)$$

Donc, la matrice de transmission totale est :

$$T(2x2) = \begin{bmatrix} \frac{a_1}{a_2}(1 + Z_1Y) & a_1a_2^*(Z_1 + Z_2 + Z_1Z_2Y) \\ \frac{Y}{a_1^*a_2} & \frac{a_2^*}{a_1^*}(1 + Z_2Y) \end{bmatrix} \quad (2.27)$$

En utilisant les équations (2.17) à (2.20) et en calculant la matrice d'admittance à partir de la matrice de transmission, il est possible de déterminer la matrice d'admittance pour un transformateur monophasé.

$$Y(2x2) = \begin{bmatrix} \frac{1 + Z_2Y}{a_1a_1^*(Z_1 + Z_2 + Z_1Z_2Y)} & \frac{-1}{a_1^*a_2(Z_1 + Z_2 + Z_1Z_2Y)} \\ \frac{-1}{a_2^*a_1(Z_1 + Z_2 + Z_1Z_2Y)} & \frac{1 + Z_1Y}{a_2a_2^*(Z_1 + Z_2 + Z_1Z_2Y)} \end{bmatrix} \quad (2.28)$$

2.4.5 Transformateur triphasé

Premièrement, voici l'explication des lettres et du numéro pour les transformateurs modélisés. La lettre majuscule indique le côté primaire du transformateur, la lettre minuscule indique le côté secondaire, et le numéro indique le déphasage entre le primaire et le secondaire. De plus, si la lettre est un « y », il s'agit d'un transformateur ayant une topologie en étoile, tandis que s'il s'agit de la lettre « d », il s'agit d'un transformateur ayant une topologie en triangle.

Sachant qu'il faut la matrice d'admittance pour résoudre un écoulement de puissance avec la méthode Newton-Raphson, il faut être capable de déterminer cette matrice pour chacun des transformateurs. De plus, sachant que les variables connues sont les tensions des phases, il faut chercher à trouver les équations permettant de calculer les courants. Les figures ainsi que les équations pour les transformateurs triphasés sont tirées du manuel concernant le programme d'écoulement de puissance NR (Pierre-Jean Lagacé, 2011).

2.4.6 Transformateur triphasé Yd0

Le premier transformateur modélisé est l'étoile-delta sans aucun déphasage. En premier lieu, voici le schéma du circuit équivalent pour ce type de transformateur :

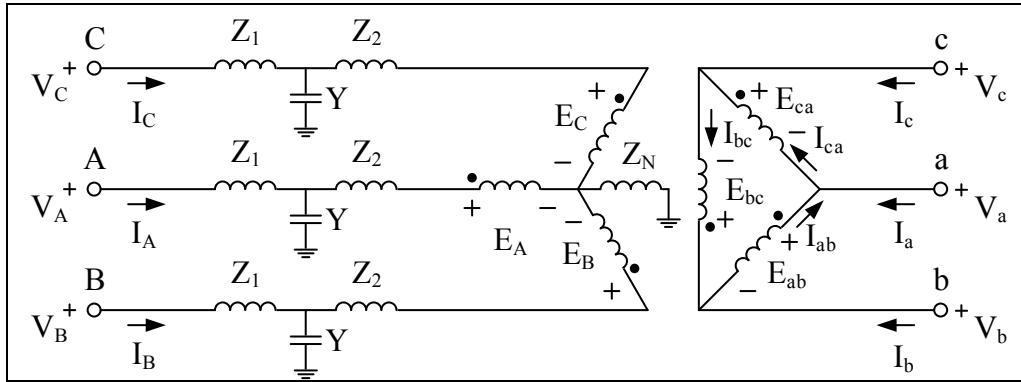


Figure 2.4 Circuit équivalent d'un transformateur Yd0

En considérant que le transformateur est idéal, voici les équations des tensions monophasées du circuit :

$$\frac{E_A}{E_{ab}} = \frac{N_Y}{N_\Delta} = \frac{1}{\sqrt{3}}, \quad \frac{E_B}{E_{bc}} = \frac{N_Y}{N_\Delta} = \frac{1}{\sqrt{3}}, \quad \frac{E_C}{E_{ca}} = \frac{N_Y}{N_\Delta} = \frac{1}{\sqrt{3}} \quad (2.29)$$

En procédant de la même façon, voici les équations du courant :

$$N_Y I_A = -N_\Delta I_{ab}, \text{ donc } I_A = \sqrt{3} I_{ab}, I_B = \sqrt{3} I_{bc}, I_C = \sqrt{3} I_{ca} \quad (2.30)$$

Il est maintenant possible de trouver le rapport entre E_A et E_{ab} :

$$E_A * \sqrt{3} = E_{ab} \quad (2.31)$$

2.4.6.1 Cr  ation de la matrice d'admittance quand $Z_1 = 0$ et $Y = 0$

En utilisant la loi de Kirchhoff pour les mailles dans la phase A du primaire du transformateur, les  quations suivantes sont obtenues :

$$V_A = Z_2 I_A + E_A + Z_N (I_A + I_B + I_C) \quad (2.32)$$

$$V_A = (Z_2 + Z_N) I_A + Z_N I_B + Z_N I_C + E_A \quad (2.33)$$

$$V_A = (Z_2 + Z_N) I_A + Z_N I_B + Z_N I_C + \frac{E_{ab}}{\sqrt{3}} \quad (2.34)$$

$$V_A - \frac{V_a}{\sqrt{3}} + \frac{V_b}{\sqrt{3}} = (Z_2 + Z_N) I_A + Z_N I_B + Z_N I_C \quad (2.35)$$

$$Donc, A = Z_2 + Z_N \text{ et } B = Z_N \quad (2.36)$$

En faisant la m me op ration pour les deux autres phases, il est possible de cr er la matrice suivante :

$$\begin{bmatrix} 1 & 0 & 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 \\ 0 & 1 & 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ 0 & 0 & 1 & 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} A & B & B \\ B & A & B \\ B & B & A \end{bmatrix} \begin{bmatrix} I_A \\ I_B \\ I_C \end{bmatrix} \quad (2.37)$$

Maintenant, il est possible de calculer le courant en utilisant la loi des n uds de Kirchhoff, ce qui donne :

$$I_a = I_{ca} - I_{ab} = \frac{I_c}{\sqrt{3}} - \frac{I_A}{\sqrt{3}} \quad (2.38)$$

Ce qui donne pour l'ensemble des phases la matrice suivante :

$$\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 \\ 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} I_A \\ I_B \\ I_C \end{bmatrix} \quad (2.39)$$

Avec toutes les matrices précédentes, il est maintenant possible de leur donner un nom de variable distinct pour chacune d'entre elles :

$$T(3 \times 3) = \begin{bmatrix} -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 \\ 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}} \end{bmatrix} \quad (2.40)$$

$$Z(3 \times 3) = \begin{bmatrix} A & B & B \\ B & A & B \\ B & B & A \end{bmatrix} \quad (2.41)$$

$$P(3 \times 6) = \begin{bmatrix} 1 & 0 & 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 \\ 0 & 1 & 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ 0 & 0 & 1 & 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \quad (2.42)$$

À partir de ces matrices, il est possible de calculer la matrice d'admittance totale d'un transformateur Yd0 lorsque $Z_1 = 0$ et $Y=0$:

$$\begin{bmatrix} I_A \\ I_B \\ I_C \\ I_a \\ I_b \\ I_c \end{bmatrix} = Y_1(6 \times 6) \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} Z^{-1}P \\ TZ^{-1}P \end{bmatrix} \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} \quad (2.43)$$

2.4.6.2 Création de la matrice d'admittance où seulement $Z_1 = 0$

Voici un schéma correspondant aux pertes dans une ligne d'un transformateur pour ce cas :

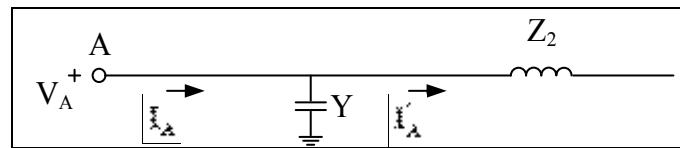


Figure 2.5 Pertes dans une branche quand $Z_1 = 0$

Le courant au primaire du transformateur pour cette figure correspondant à l'équation suivante :

$$I_A = I_A' + YV_A \quad (2.44)$$

En utilisant cette équation et en intégrant l'équation de la matrice d'admittance telle que vue dans l'équation précédente, cela donne :

$$\begin{bmatrix} I_A - YV_A \\ I_B - YV_B \\ I_C - YV_C \\ I_a \\ I_b \\ I_c \end{bmatrix} = Y_1 \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} \quad (2.45)$$

En ajoutant Y dans l'équation, la matrice d'admittance totale pour ce cas est :

$$\begin{bmatrix} I_A \\ I_B \\ I_C \\ I_a \\ I_b \\ I_c \end{bmatrix} = Y_2 \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} = \left[Y_1 + \begin{bmatrix} Y & 0 & 0 & 0 & 0 & 0 \\ 0 & Y & 0 & 0 & 0 & 0 \\ 0 & 0 & Y & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \right] \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} \quad (2.46)$$

2.4.6.3 Création de la matrice d'admittance avec toutes les variables (Z_1 , Z_2 et Y)

Voici un schéma correspondant aux pertes dans une ligne d'un transformateur pour ce cas :

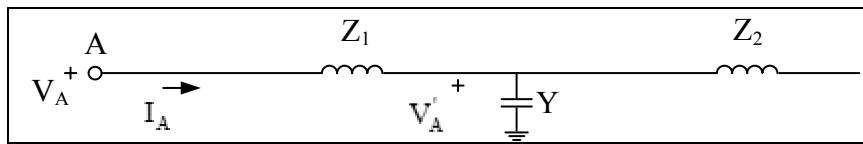


Figure 2.6 Pertes dans une branche d'un transformateur

En ajoutant l'impédance Z_1 au primaire, la tension augmente. Voici l'équation correspondante à la tension au primaire du transformateur pour ce cas :

$$V_A = V'_A + Z_1 I_A \quad (2.47)$$

En utilisant cette équation et en intégrant l'équation de la matrice d'admittance telle que vue dans le dernier cas, cela donne :

$$\begin{bmatrix} I_A \\ I_B \\ I_C \\ I_a \\ I_b \\ I_c \end{bmatrix} = Y_2(6 \times 6) \begin{bmatrix} V_A - Z_1 I_A \\ V_B - Z_1 I_B \\ V_C - Z_1 I_C \\ V_a \\ V_b \\ V_c \end{bmatrix} \quad (2.48)$$

(2.49)

$$\begin{bmatrix} I_A \\ I_B \\ I_C \\ I_a \\ I_b \\ I_c \end{bmatrix} = Y_2 \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} - Y_2 \begin{bmatrix} Z_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & Z_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_A \\ I_B \\ I_C \\ I_a \\ I_b \\ I_c \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} + Y_2 \begin{bmatrix} Z_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & Z_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_A \\ I_B \\ I_C \\ I_a \\ I_b \\ I_c \end{bmatrix} = Y_2 \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} \quad (2.50)$$

$$\begin{bmatrix} I_A \\ I_B \\ I_C \\ I_a \\ I_b \\ I_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.51)$$

$$+ Y_2 \begin{bmatrix} Z_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & Z_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{-1} Y_2 \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix}$$

$$\begin{bmatrix} I_A \\ I_B \\ I_C \\ I_a \\ I_b \\ I_c \end{bmatrix} = Y_3(6 \times 6) \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} \quad (2.52)$$

$$Y_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} + Y_2 \begin{bmatrix} Z_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & Z_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{-1} Y_2 \quad (2.53)$$

En intégrant dans cette équation les formules de Y_1 et Y_2 obtenus précédemment, la matrice d'admittance totale d'un transformateur Y_{d0} est :

$$Y_3(6 \times 6) = \begin{bmatrix} Y_{11} & Y_{12} & Y_{12} & Y_{14} & -Y_{14} & 0 \\ Y_{12} & Y_{11} & Y_{12} & 0 & Y_{14} & -Y_{14} \\ Y_{12} & Y_{12} & Y_{11} & -Y_{14} & 0 & Y_{14} \\ Y_{14} & 0 & -Y_{14} & Y_{44} & Y_{45} & Y_{45} \\ -Y_{14} & Y_{14} & 0 & Y_{45} & Y_{44} & Y_{45} \\ 0 & -Y_{14} & Y_{14} & Y_{45} & Y_{45} & Y_{44} \end{bmatrix} \quad (2.54)$$

Voici les équations résultantes pour l'ensemble des variables :

$$Y_{11} = \frac{2(1 + Z_2 Y)}{3(Z_2 + Z_1(1 + Z_2 Y))} + \frac{1 + (Z_2 + 3Z_N)Y}{3((Z_2 + Z_N) + Z_1(1 + (Z_2 + 3Z_N)Y))} \quad (2.55)$$

$$Y_{12} = \frac{-Z_N}{(Z_1 + Z_2 + Z_1 Z_2 Y)(Z_1 + Z_2 + 3Z_N + Z_1 Y(Z_2 + 3Z_N))} \quad (2.56)$$

$$Y_{14} = \frac{-1}{\sqrt{3}(Z_1 + Z_2 + Z_1 Z_2 Y)} \quad (2.57)$$

$$Y_{44} = \frac{2(1 + Z_1 Y)}{3(Z_1 + Z_2 + Z_1 Z_2 Y)} \quad (2.58)$$

$$Y_{45} = \frac{-(1 + Z_1 Y)}{3(Z_1 + Z_2 + Z_1 Z_2 Y)} \quad (2.59)$$

2.4.7 Transformateur triphasé Yy0

Le deuxième transformateur triphasé modélisé est l'étoile/étoile sans aucun déphasage. Il est possible de trouver les équations calculant le courant dans chaque branche par rapport à la tension. Une fois fait, il est possible de trouver la matrice d'admittance du transformateur. En premier lieu, voici un schéma représentant son circuit équivalent :

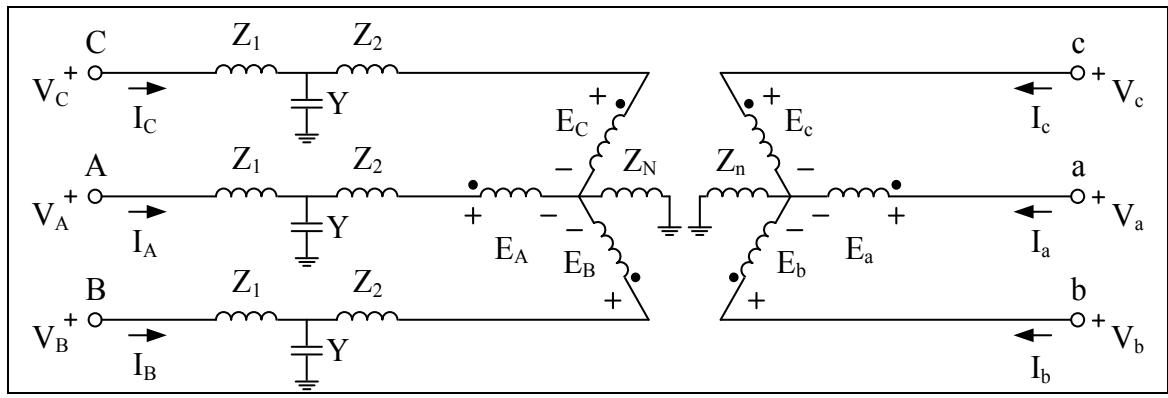


Figure 2.7 Circuit équivalent d'un transformateur Yy0

En considérant que le transformateur est idéal, les tensions du primaire, ainsi que celle du secondaire suivent les formules suivantes :

$$\frac{E_A}{E_a} = \frac{N_H}{N_X} = 1, \quad \frac{E_B}{E_b} = \frac{N_H}{N_X} = 1, \quad \frac{E_C}{E_c} = \frac{N_H}{N_X} = 1 \quad (2.60)$$

La même procédure peut être appliquée pour les courants :

$$N_H I_A = -N_X I_a, \text{ donc } I_A = -I_a, I_B = -I_b, I_C = -I_c \quad (2.61)$$

2.4.7.1 Création de la matrice d'admittance où $Z_1 = 0$ et $Y = 0$

En utilisant la loi de Kirchhoff pour les mailles dans la phase A du primaire du transformateur, les équations suivantes sont obtenues :

$$V_A = Z_2 I_A + E_A + Z_N (I_A + I_B + I_C) \quad (2.62)$$

$$V_A = (Z_2 + Z_N) I_A + Z_N I_B + Z_N I_C + E_A \quad (2.63)$$

$$V_A = (Z_2 + Z_N) I_A + Z_N I_B + Z_N I_C + E_a \quad (2.64)$$

$$E_A = V_a - Z_n I_a - Z_n I_b - Z_n I_c \quad (2.65)$$

$$V_A - V_a = (Z_2 + Z_N) I_A + Z_N I_B + Z_N I_C - Z_n I_a - Z_n I_b - Z_n I_c \quad (2.66)$$

$$V_A - V_a = (Z_2 + Z_N + Z_n) I_A + (Z_N + Z_n) I_B + (Z_N + Z_n) I_C \quad (2.67)$$

$$\text{où } (Z_2 + Z_N + Z_n) = A \text{ et } (Z_N + Z_n) = B \quad (2.68)$$

En faisant la même opération pour les deux autres phases, il est possible de créer la matrice suivante :

$$\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} A & B & B & 0 & 0 & 0 \\ B & A & B & 0 & 0 & 0 \\ B & B & A & 0 & 0 & 0 \\ 0 & 0 & 0 & A & B & B \\ 0 & 0 & 0 & B & A & B \\ 0 & 0 & 0 & B & B & A \end{bmatrix} \begin{bmatrix} I_A \\ I_B \\ I_C \\ I_a \\ I_b \\ I_c \end{bmatrix} \quad (2.69)$$

De cette équation, il est possible de remplacer la matrice du côté gauche de l'égalité par la variable P et la matrice du côté droit par Z . En utilisant ces variables, voici l'équation nécessaire pour trouver la matrice d'admittance du transformateur sans les variables Z_1 et Y :

$$\begin{bmatrix} I_A \\ I_B \\ I_C \\ I_a \\ I_b \\ I_c \end{bmatrix} = Y_1(6 \times 6) \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} = Z^{-1}P \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} \quad (2.70)$$

2.4.7.2 Crédation de la matrice d'admittance incluant toutes les variables du schéma

Voici un schéma correspondant aux pertes dans une ligne d'un transformateur :

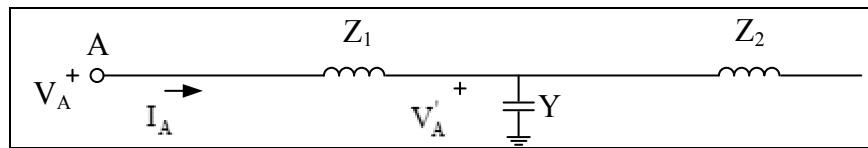


Figure 2.8 Pertes dans une branche d'un transformateur

Pour ce cas-ci, il faut utiliser la même démarche qu'il y a plus haut pour le transformateur Yd0. Donc pour l'effet d'Y dans la matrice d'admittance, l'équation est la suivante :

$$Y_2 = Y_1 + \begin{bmatrix} Y & 0 & 0 & 0 & 0 & 0 \\ 0 & Y & 0 & 0 & 0 & 0 \\ 0 & 0 & Y & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.71)$$

Cependant, l'équation totale en intégrant la variable Z_1 est totalement identique au transformateur précédent. En effet, voici l'équation de la matrice d'admittance totale d'un transformateur Yy0 :

$$Y_3(6 \times 6) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.72)$$

$$+ Y_2 \begin{bmatrix} Z_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & Z_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{-1} Y_2$$

$$Y_3(6 \times 6) = \begin{bmatrix} Y_{11} & Y_{12} & Y_{12} & Y_{14} & Y_{15} & Y_{15} \\ Y_{12} & Y_{11} & Y_{12} & Y_{15} & Y_{14} & Y_{15} \\ Y_{12} & Y_{12} & Y_{11} & Y_{15} & Y_{15} & Y_{14} \\ Y_{14} & Y_{15} & Y_{15} & Y_{44} & Y_{45} & Y_{45} \\ Y_{15} & Y_{14} & Y_{15} & Y_{45} & Y_{44} & Y_{45} \\ Y_{15} & Y_{15} & Y_{14} & Y_{45} & Y_{45} & Y_{44} \end{bmatrix} \quad (2.73)$$

Voici les équations de l'ensemble des variables de la matrice d'admittance :

$$Y_{11} = \frac{2(1 + Z_2 Y)}{3(Z_1 + Z_2 + Y Z_1 Z_2)} \quad (2.74)$$

$$+ \frac{1 + (Z_2 + 3Z_N)Y}{3(Z_1 + Z_2 + 3(Z_N + Z_n) + Y Z_1 (Z_2 + 3(Z_N + Z_n)))}$$

$$Y_{12} = \frac{-(Z_N + Z_n)}{(Z_1 + Z_2 + Y Z_1 Z_2)(Z_1 + Z_2 + 3(Z_N + Z_n) + Y Z_1 (Z_2 + 3(Z_N + Z_n)))} \quad (2.75)$$

$$Y_{14} = \frac{-(Z_1 + Z_2 + 2(Z_N + Z_n) + Z_1 Y (Z_2 + 2(Z_N + Z_n)))}{(Z_1 + Z_2 + Y Z_1 Z_2)(Z_1 + Z_2 + 3(Z_N + Z_n) + Y Z_1 (Z_2 + 3(Z_N + Z_n)))} \quad (2.76)$$

$$Y_{15} = \frac{1}{3(Z_1 + Z_2 + YZ_1Z_2)} - \frac{1}{3(Z_1 + Z_2 + 3(Z_N + Z_n) + YZ_1(Z_2 + 3(Z_N + Z_n)))} \quad (2.77)$$

$$Y_{44} = \frac{2(1 + Z_1Y)}{3(Z_1 + Z_2 + YZ_1Z_2)} + \frac{(1 + Z_1Y)}{3(Z_1 + Z_2 + 3(Z_N + Z_n) + YZ_1(Z_2 + 3(Z_N + Z_n)))} \quad (2.78)$$

$$Y_{45} = \frac{-(1 + Z_1Y)^2(Z_N + Z_n)}{(Z_1 + Z_2 + YZ_1Z_2)(Z_1 + Z_2 + 3(Z_N + Z_n) + YZ_1(Z_2 + 3(Z_N + Z_n)))} \quad (2.79)$$

2.5 Création de la matrice d'admittance totale

2.5.1 Modélisation de la matrice avant l'optimisation

Une fois toutes les composantes modélisées, il est possible de créer la matrice d'admittance totale du système. Pour ce faire, voici une figure montrant un réseau monophasé composé de trois barres.

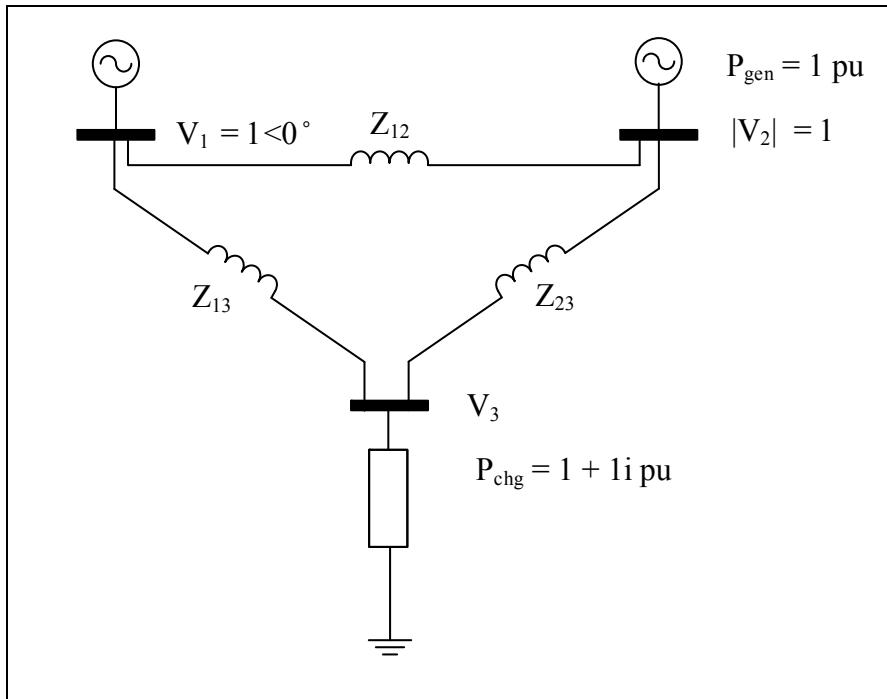


Figure 2.9 Schéma d'un réseau monophasé simple

Ce réseau est composé d'une barre d'équilibre, d'une barre de génération, ainsi que d'une barre de charge. Pour calculer la matrice d'admittance de ce réseau, il faut tout simplement additionner les admittances connectées aux barres pour tous les Y_{ii} et soustraire l'admittance entre les barres pour les Y_{ij} .

$$Y_{\text{bus}}(3 \times 3) = \begin{bmatrix} \left(\frac{1}{Z_{12}} + \frac{1}{Z_{13}}\right) & -\frac{1}{Z_{12}} & -\frac{1}{Z_{13}} \\ -\frac{1}{Z_{12}} & \left(\frac{1}{Z_{12}} + \frac{1}{Z_{23}}\right) & -\frac{1}{Z_{23}} \\ -\frac{1}{Z_{13}} & -\frac{1}{Z_{23}} & \left(\frac{1}{Z_{13}} + \frac{1}{Z_{23}}\right) \end{bmatrix} \quad (2.80)$$

Par contre, s'il faut calculer la matrice d'admittance du même réseau, mais cette fois en triphasé, il faut prendre en considération chacune des phases pour l'ensemble des barres du réseau électrique. Cela dit, chaque ligne de la matrice d'admittance correspond à une seule phase pour une barre donnée. En utilisant l'équation (2.8) pour toutes les lignes, il est facile

de calculer la matrice d'admittance de l'ensemble du réseau. Étant donné que le réseau monophasé est composé de trois barres, cela correspond à une matrice trois par trois (n par n). Donc, si le réseau est triphasé, il faut tout simplement multiplier « n » par trois afin d'insérer l'ensemble des phases. Donc, pour la Figure 2.9, la matrice d'admittance en triphasé est une matrice de dimensions neuf par neuf. En prenant les résultats de l'équation (2.80), voici le résultat de la matrice en triphasé :

$$Y(9 \times 9) = \begin{bmatrix} Y_{11} & 0 & 0 & Y_{12} & 0 & 0 & Y_{13} & 0 & 0 \\ 0 & Y_{11} & 0 & 0 & Y_{12} & 0 & 0 & Y_{13} & 0 \\ 0 & 0 & Y_{11} & 0 & 0 & Y_{12} & 0 & 0 & Y_{13} \\ Y_{21} & 0 & 0 & Y_{22} & 0 & 0 & Y_{23} & 0 & 0 \\ 0 & Y_{21} & 0 & 0 & Y_{22} & 0 & 0 & Y_{23} & 0 \\ 0 & 0 & Y_{21} & 0 & 0 & Y_{22} & 0 & 0 & Y_{23} \\ Y_{31} & 0 & 0 & Y_{32} & 0 & 0 & Y_{33} & 0 & 0 \\ 0 & Y_{31} & 0 & 0 & Y_{32} & 0 & 0 & Y_{33} & 0 \\ 0 & 0 & Y_{31} & 0 & 0 & Y_{32} & 0 & 0 & Y_{33} \end{bmatrix} \quad (2.81)$$

La matrice précédente montre un réseau triphasé composé de lignes non couplées (lignes moyennes). Cette matrice peut donc devenir très encombrante dans la mémoire du simulateur, et c'est pourquoi il est préférable d'utiliser une matrice creuse dans le simulateur afin de faciliter le temps de calcul et réduire l'espace mémoire requis. Cependant, créer cette matrice est un processus qui prend beaucoup de temps. Il est par contre possible de réduire la taille de la matrice d'admittance totale tout en gardant la même précision que lorsqu'elle n'est pas réduite. C'est pourquoi le prochain point aborde la façon d'optimiser la matrice d'admittance afin d'améliorer la vitesse tout en gardant la même précision qu'avant.

2.5.2 Optimisation de la matrice d'admittance

Voici comment il est possible de minimiser la matrice d'admittance en utilisant quelques manipulations mathématiques. Cette section est séparée en quatre sous-sections. Premièrement, il y a une minimisation de la matrice si le côté primaire d'une ligne ou d'un transformateur est une barre de génération et le côté secondaire est une barre de charge. Deuxièmement, il y a l'opposé. Troisièmement, il y a le cas où le côté primaire est une barre

de génération (barre PV) et le côté secondaire est aussi une barre de génération (barre PV). Finalement, il y a le cas où le côté primaire et le côté secondaire sont des barres de charges (barres PQ). Donc, les barres de générations et d'équilibre sont comptées comme étant une seule barre (nombre de barres = $n+1$), tandis que les barres de charges sont comptées comme trois barres pour inclure l'ensemble des phases (nombre de barres = $n+3$). Les figures et les équations suivantes qui se trouvent dans cette section sont toutes tirées du manuel pour le simulateur d'écoulement de puissance NR (Pierre-Jean Lagacé, 2011) pour l'optimisation de la matrice d'admittance.

2.5.2.1 Barre de génération (PV) ou d'équilibre (SW) vers barre de charge (PQ)

Premièrement, voici le circuit représentant une ligne triphasée partant d'une barre de génération vers une barre de charge :

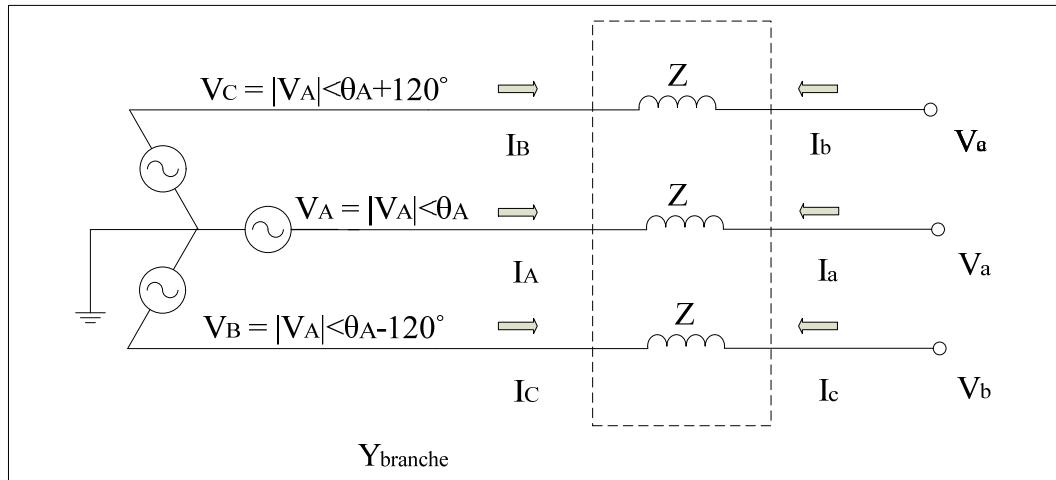


Figure 2.10 Circuit équivalent d'une barre PV vers une barre PQ

Il est possible de minimiser ce schéma en introduisant des transformateurs monophasés pour les lignes de la phase B et C. Il suffit de mettre un rapport de transformation de -120 degrés et +120 degrés respectivement, et la matrice d'admittance passera d'une matrice six par six à une matrice quatre par quatre. Voici le circuit avec l'intégration des trois transformateurs :

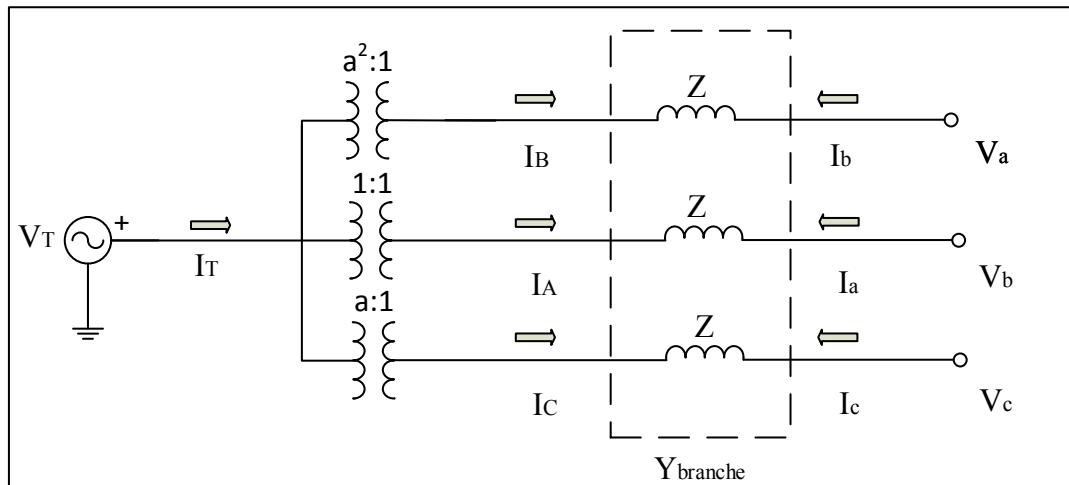


Figure 2.11 Circuit simplifié d'une barre PV vers une barre PQ

Il faut maintenant utiliser la loi de Kirchhoff des nœuds et des mailles pour obtenir les matrices suivantes :

$$a = 1 \angle 120^\circ \quad (2.82)$$

$$\begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a^2 & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} V_T \\ V_a \\ V_b \\ V_c \end{bmatrix} \quad (2.83)$$

$$\begin{bmatrix} I_T \\ I_a \\ I_b \\ I_c \end{bmatrix} = \begin{bmatrix} 1 & a & a^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} I_A \\ I_B \\ I_C \\ I_a \\ I_b \\ I_c \end{bmatrix} \quad (2.84)$$

De plus,

$$\begin{bmatrix} I_A \\ I_B \\ I_C \\ I_a \\ I_b \\ I_c \end{bmatrix} = Y_{branche} * \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} \quad (2.85)$$

Donc, voici l'équation qu'il faut utiliser en utilisant les trois équations précédentes :

$$\begin{bmatrix} I_T \\ I_a \\ I_b \\ I_c \end{bmatrix} = \begin{bmatrix} 1 & a & a^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} * Y_{branche} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ a^2 & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} V_T \\ V_a \\ V_b \\ V_c \end{bmatrix} \quad (2.86)$$

Cela dit, s'il faut ajouter des éléments dans la matrice d'admittance, il faut tout simplement additionner la matrice calculée avec l'équation précédente ce qui donne :

$$Y_{réseau}^{4x4} = Y_{réseau}^{4x4} + \begin{bmatrix} 1 & a & a^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} * Y_{branche} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ a^2 & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.87)$$

En utilisant cette équation, la matrice d'admittance totale diminuera par rapport à l'autre approche expliquée. Cela est dû au fait que même si la matrice d'admittance de la branche contient 36 éléments, la taille de la matrice d'admittance pour cette branche diminue et ne peut contenir que 16 éléments maximum au total. Cette méthode est donc implémentée dans le simulateur afin de diminuer la taille des matrices.

2.5.2.2 Barre de charge (PQ) vers barre de génération (PV) ou d'équilibre (SW)

En utilisant la même méthode que celle expliquée précédemment, c'est-à-dire l'implémentation des transformateurs monophasés pour chaque phase du côté des barres de

générations, il est possible de réduire la taille de la matrice d'admittance du réseau lorsqu'il s'agit d'une ligne triphasée partant d'une barre de charge, et allant vers une barre de génération. Premièrement, voici le circuit équivalent de ce cas-ci :

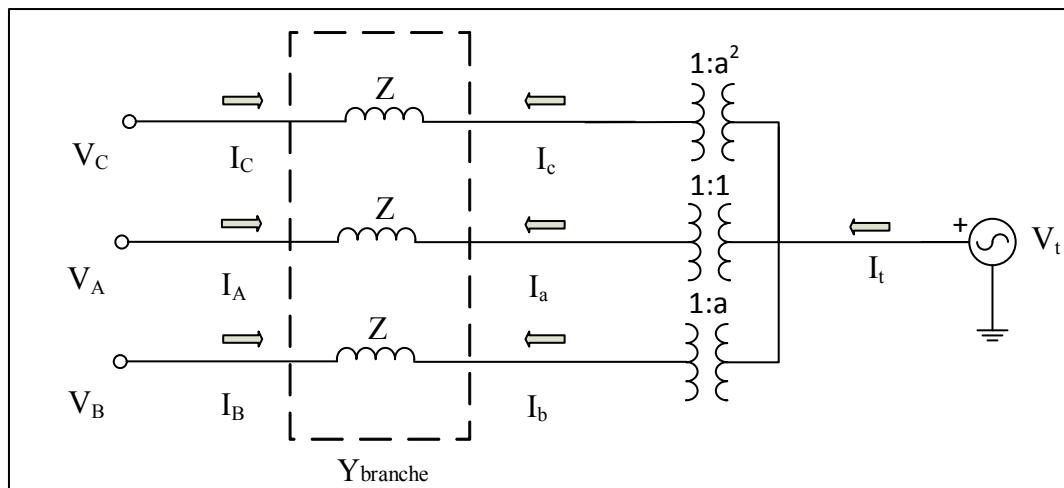


Figure 2.12 Circuit simplifié d'une barre PQ vers une barre PV

De la même façon que pour une branche avec extrémités de type PQ-PV, il faut appliquer les lois de Kirchhoff sur la Figure 2.12. Voici les équations résultantes :

$$\begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a^2 \\ 0 & 0 & 0 & a \end{bmatrix} * \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_t \end{bmatrix} \quad (2.88)$$

$$\begin{bmatrix} I_A \\ I_B \\ I_C \\ I_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & a & a^2 \end{bmatrix} * \begin{bmatrix} I_A \\ I_B \\ I_C \\ I_a \\ I_b \\ I_c \end{bmatrix} \quad (2.89)$$

En utilisant les deux équations précédentes, il est maintenant possible d'additionner la matrice d'admittance d'une ligne à la matrice d'admittance du réseau avec l'équation suivante :

$$Y_{réseau}^{4x4} = Y_{réseau}^{4x4} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * Y_{branche} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a^2 \\ 0 & 0 & 0 & a \end{bmatrix} \quad (2.90)$$

2.5.2.3 Barre de génération (PV) vers barre de génération (PV) ou d'équilibre (SW)

Pour ce qui est de ce type de branche, il est possible de diminuer encore plus la matrice d'admittance de la branche. Voici le circuit équivalent représentant ce cas avec l'implémentation de transformateurs idéaux des deux côtés de la ligne :

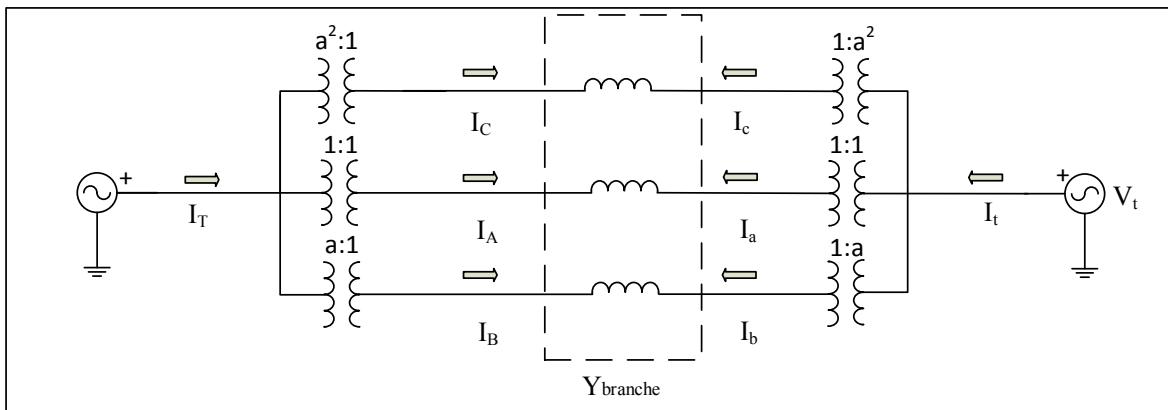


Figure 2.13 Circuit simplifié d'une barre PV vers une barre PV

En appliquant les lois de Kirchhoff au circuit de la Figure 2.13, les équations obtenues sont :

$$\begin{bmatrix} V_A \\ V_B \\ V_C \\ V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ a^2 & 0 \\ a & 0 \\ 0 & 1 \\ 0 & a^2 \\ 0 & a \end{bmatrix} * \begin{bmatrix} V_T \\ V_t \end{bmatrix} \quad (2.91)$$

$$\begin{bmatrix} I_A \\ I_B \\ I_C \\ I_t \end{bmatrix} = \begin{bmatrix} 1 & a & a^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & a & a^2 \end{bmatrix} * \begin{bmatrix} I_A \\ I_B \\ I_C \\ I_a \\ I_b \\ I_c \end{bmatrix} \quad (2.92)$$

De la même manière que pour les deux autres cas, la matrice d'admittance pour ce type de branche est réduite de beaucoup. Elle passe donc d'une matrice six par six à une matrice deux par deux. Voici l'équation implémentée dans le simulateur qui fait ces manipulations :

$$Y_{réseau}^{2x2} = Y_{réseau}^{2x2} + \begin{bmatrix} 1 & a & a^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & a & a^2 \end{bmatrix} * Y_{branche} * \begin{bmatrix} 1 & 0 \\ a^2 & 0 \\ a & 0 \\ 0 & 1 \\ 0 & a^2 \\ 0 & a \end{bmatrix} \quad (2.93)$$

2.6 Comparaison entre la matrice d'admittance complète et la matrice d'admittance simplifiée

La principale différence entre les méthodes est la grosseur des matrices manipulées dans le programme. Pour la première méthode, chacune des phases est considérée comme étant une barre en elle-même. C'est-à-dire que pour la matrice d'admittance totale du système, le nombre de lignes et de colonnes est donné par l'équation suivante :

$$Nb_{lignes} \text{ ou } Nb_{colonnes} = n * 3 \quad (2.94)$$

n étant le nombre de barres du système

Tandis que pour la méthode optimisée, le nombre de lignes et de colonnes de la matrice d'admittance sera réduit, car seulement une phase est comptée par barre d'équilibre ou de génération, et trois phases par barres de charges. Cela permet de simplifier la taille des matrices d'erreurs résiduelles, Jacobienne et bien sûr d'admittance. Le temps de calcul du simulateur sera donc moins élevé, et étant donné que le temps est une des variables les plus importantes du programme, c'est cette approche qui a été programmée dans le simulateur.

2.7 Implémentation des méthodes dans le simulateur

2.7.1 Méthode Newton-Raphson

La première méthode implémentée dans le simulateur est celle de Newton-Raphson. Cette méthode itérative lorsqu'utilisée pour un écoulement de puissance sert à minimiser l'écart entre les puissances actives et réactives connues et leurs résultats à partir d'équations matricielles. Les différentes équations de cette section proviennent du manuel NR (Pierre-Jean Lagacé, 2011).

2.7.1.1 Matrice d'erreur résiduelle

Que ce soit pour la méthode Newton-Raphson ou Levenberg-Marquardt, cette étape est utilisée lors de la résolution du problème. Lorsqu'il faut calculer un écoulement de puissance, les puissances actives des barres de génération et les puissances actives et réactives des barres de charges sont connues. De plus, la sommation des puissances entrantes et sortantes de chacune des barres doit être égale à zéro. Par exemple, si une barre de génération fournit un pu, il faut qu'en additionnant la charge de cette barre, ainsi que la puissance absorbée par les admittances du réseau, le résultat soit égal à zéro. Cela dit, voici les équations permettant de déterminer les puissances pour toutes les barres d'un réseau électrique sachant que le courant est le produit de la matrice d'admittance du réseau avec les tensions.

$$I_{réseau} = Y_{réseau} * V_{réseau} \quad (2.95)$$

$$S_{réseau} = V_{réseau} * (Y_{réseau} * V_{réseau})^* \quad (2.96)$$

L'équation précédente donne les puissances totales de l'ensemble du réseau. Il faut cependant séparer les puissances actives et réactives pour déterminer chacune des contraintes du système. Voici les équations des contraintes des puissances actives et réactives pour une barre i donnée :

$$\begin{aligned} P_{barre,i} &= P_{barre,i}^{chg} - P_{barre,i}^{gen} + P_{Ybus,i} \\ Q_{barre,i} &= Q_{barre,i}^{chg} - Q_{barre,i}^{gen} + Q_{Ybus,i} \end{aligned} \quad (2.97)$$

Afin de bien calculer la puissance correspondante aux charges, les équations suivantes ont été intégrées dans le simulateur, car les charges ne sont pas calculées de la même manière s'il s'agit d'une charge à courant constant, à impédance constante ou à facteur NP ou NQ. Voici l'équation des puissances aux charges sur une barre i donnée :

$$\begin{aligned} P_{barre,i}^{chg} &= P_{chg,i} + (P_{chg\,P,i} + P_{chg\,I,i} * |V_i| + P_{chg\,Z,i} * |V_i^2| + \\ &\quad P_{chg\,NP,i} * |V_i^{NP_i}|) \\ Q_{barre,i}^{chg} &= Q_{chg,i} + (Q_{chg\,P,i} + Q_{chg\,I,i} * |V_i| + Q_{chg\,Z,i} * |V_i^2| + \\ &\quad Q_{chg\,NQ,i} * |V_i^{NQ_i}|) \end{aligned} \quad (2.98)$$

Finalement, la puissance du réseau est directement reliée aux impédances de branches, ainsi qu'aux différentes tensions des barres (pour une barre i donnée) :

$$\begin{aligned} P_{Ybus,i} &= |V_i| * \sum_{x=1}^N |V_x| * (G_{ix} \cos \theta_{ix} + B_{ix} \sin \theta_{ix}) \\ Q_{Ybus,i} &= |V_i| * \sum_{x=1}^N |V_x| * (G_{ix} \sin \theta_{ix} - B_{ix} \cos \theta_{ix}) \end{aligned} \quad (2.99)$$

L'ensemble de ces équations permet de calculer les variations des tensions, ce qui permet de déterminer l'erreur résiduelle. Celle-ci doit être le plus proche de zéro lors de l'écoulement de puissance, car les valeurs de ces puissances sont connues au départ du problème.

2.7.1.2 Creation de la matrice Jacobienne

Tout comme la matrice d'erreur résiduelle, la matrice Jacobienne est nécessaire afin de calculer l'ajustement des tensions. Ce qui permet d'avoir une erreur résiduelle nulle. Celle-ci consiste à déterminer les dérivées des puissances de l'ensemble des barres de génération et de charges du réseau par rapport aux angles et aux tensions avoisinantes.

2.7.1.3 Calcul de l'espace mémoire requis pour créer la matrice Jacobienne

La première étape est de déterminer où se trouvent les valeurs de la matrice d'admittance non nulles. Normalement, si la matrice n'avait pas été réduite, la solution pour le calcul de l'espace mémoire aurait été relativement simple. Cependant, étant donné que celle-ci est réduite, il faut non seulement regarder quels sont les types des barres de chacune des lignes de la matrice, mais aussi il faut prendre en considération quelles sont les composantes qui relient les barres entre elles. Car, un transformateur $Yd0$ venant d'une barre de charge allant vers une autre barre de charge, a 36 éléments dans la matrice d'admittance non nuls.

Cela dit, voici la procédure utilisée dans le programme. Premièrement, le simulateur fait une boucle de l'ensemble des barres PV et PQ trouvés dans le réseau, et pour chacune des lignes (éléments ii), il additionne 2 au compteur. Ensuite, pour chacun des éléments non nuls de la matrice d'admittance qui ne sont pas dans la ligne diagonale (éléments ij), mais qui doivent être dérivées, il ajoute 2 aussi. Cela fait donc une estimation de l'espace mémoire que le simulateur doit accorder à la matrice Jacobienne. Il fait cette opération à chaque fois qu'il y a un changement dans la matrice d'admittance, car la matrice Jacobienne change en nombre d'éléments.

2.7.1.4 Formules utilisées pour calculer la matrice Jacobienne

Voici les principales formules utilisées pour calculer l'ensemble de la matrice Jacobienne :

$$\begin{aligned}
 J_{P\theta,ii} &= \frac{\delta P_i}{\delta \theta_i} = -B_{ii}|V_i|^2 + |V_i| \sum_{x=1}^N |V_x| (B_{ix} \cos \theta_{ix} - G_{ix} \sin \theta_{ix}) \\
 J_{P\theta,ij} &= \frac{\delta P_i}{\delta \theta_j} = |V_i| |V_j| (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \\
 J_{PV,ii} &= \frac{\delta P_i}{\delta V_i} = P_{chg\ p,i} + P_{chg\ l,i} * |V_i| + P_{chg\ z,i} * |V_i^2| \\
 &\quad + P_{chg\ NP,i} * |V_i^{NPi-1}| + G_{ii}|V_i| \\
 &\quad + \sum_{x=1}^N |V_x| (G_{ix} \cos \theta_{ix} + B_{ix} \sin \theta_{ix}) \\
 J_{PV,ij} &= \frac{\delta P_i}{\delta V_j} = |V_i| (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \\
 J_{Q\theta,ii} &= \frac{\delta Q_i}{\delta \theta_i} = -G_{ii}|V_i|^2 + |V_i| \sum_{x=1}^N |V_x| (G_{ix} \cos \theta_{ix} + B_{ix} \sin \theta_{ix}) \\
 J_{Q\theta,ij} &= \frac{\delta Q_i}{\delta \theta_j} = |V_i| |V_j| (-G_{ij} \cos \theta_{ij} - B_{ij} \sin \theta_{ij}) \\
 J_{QV,ii} &= \frac{\delta Q_i}{\delta V_i} = Q_{chg\ p,i} + Q_{chg\ l,i} * |V_i| + Q_{chg\ z,i} * |V_i^2| \\
 &\quad + Q_{chg\ NQ,i} * |V_i^{NQi-1}| - B_{ii}|V_i| \\
 &\quad + \sum_{x=1}^N |V_x| (G_{ix} \sin \theta_{ix} - B_{ix} \cos \theta_{ix}) \\
 J_{QV,ij} &= \frac{\delta Q_i}{\delta V_j} = |V_i| (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij})
 \end{aligned} \tag{2.100}$$

Où l'indice « ii » représente la diagonale de la matrice Jacobienne et « ij » tous les autres éléments. Dans un écoulement de puissance, la matrice Jacobienne sert à faire moduler les

angles et les modules des barres de l'ensemble du réseau afin d'obtenir une erreur résiduelle qui soit nulle.

2.7.1.5 Création de la boucle itérative

Une fois les matrices F et J calculées, il faut créer la boucle itérative en suivant ces étapes :

- faire l'initialisation des variables ;
- faire la boucle itérative. À l'intérieur de celle-ci, calculer la matrice F et J , ainsi que l'accroissement dX pour chaque itération et additionner les tensions par leurs accroissements respectifs ;

$$dX = -J|F \quad (2.101)$$

- créer un rapport et faire la validation pour plusieurs cas.

2.7.2 Méthode Levenberg-Marquardt

Cette méthode est implémentée dans le simulateur, car sa convergence est supérieure à celle de la méthode Newton-Raphson. Elle utilise les mêmes matrices, soit F et J , mais il faut y ajouter le calcul de la variable λ . Avec cette méthode, cette variable permet d'avoir un accroissement qui est plus petit qu'avec la méthode Newton-Raphson. Un meilleur taux de réussite pour la convergence est alors obtenu. Cependant, avoir un accroissement plus petit peut aussi dire que le simulateur doit faire plus d'itérations pour converger, ce qui nuit au temps de calcul de l'écoulement de puissance.

2.7.2.1 Ajouter l'opérateur Lagrangien (Lambda)

Tel que spécifié précédemment, le Lagrangien est indispensable à cette approche. Dans le simulateur, cet opérateur est égal à la somme des puissances des barres au carré divisées par le nombre de barres connectées au réseau. Par exemple, si le réseau contient 3 000 barres, et

qu'après une itération, la somme des carrés est égale à 30 000, l'opérateur lambda est égal à 10. Ce paramètre est arbitraire et peut donc être modifié au besoin.

2.7.2.2 Création de la boucle itérative

La boucle itérative utilise la même approche qu'avec la méthode Newton-Raphson. La seule différence avec cette méthode est qu'il faut ajouter le calcul de lambda au calcul de la variable d'accroissement dX. Voici son équation :

$$dX = -(J^T J + \lambda I) \backslash (J^T F) \quad (2.102)$$

Où J^T est la matrice transposée de la matrice Jacobienne.

2.7.3 Paramètres à déterminer

Que ce soit avec la méthode Newton-Raphson ou avec la méthode Levenberg-Marquardt, l'écoulement de puissance a les mêmes paramètres à déterminer. Sachant que pour chaque type de barres, il y a deux paramètres qui sont connus et deux qui sont à déterminer, voici un tableau récapitulatif de ces paramètres pour le simulateur :

Tableau 2.1 Paramètres connus et à déterminer du simulateur

Type de barre	Paramètres connus	Paramètres à déterminer
SW	Module de tension Angle de tension	Puissance active Puissance réactive
PV	Module de tension Puissance active	Angle de tension Puissance réactive
PQ	Puissance active Puissance réactive	Module de tension Angle de tension

2.7.4 Conditions d'arrêt de la boucle itérative

Afin d'arrêter la boucle itérative, il y a quelques opérations qui doivent survenir. En voici la liste :

- les erreurs résiduelles sont inférieures à la limite indiquée (1×10^{-10}) et le processus converge ;
- le nombre d'itérations maximal est atteint ;
- les valeurs des tensions sont trop grandes et le processus diverge.

2.8 Implémentation de la régulation

2.8.1 Algorithme utilisé pour l'implémentation de la régulation

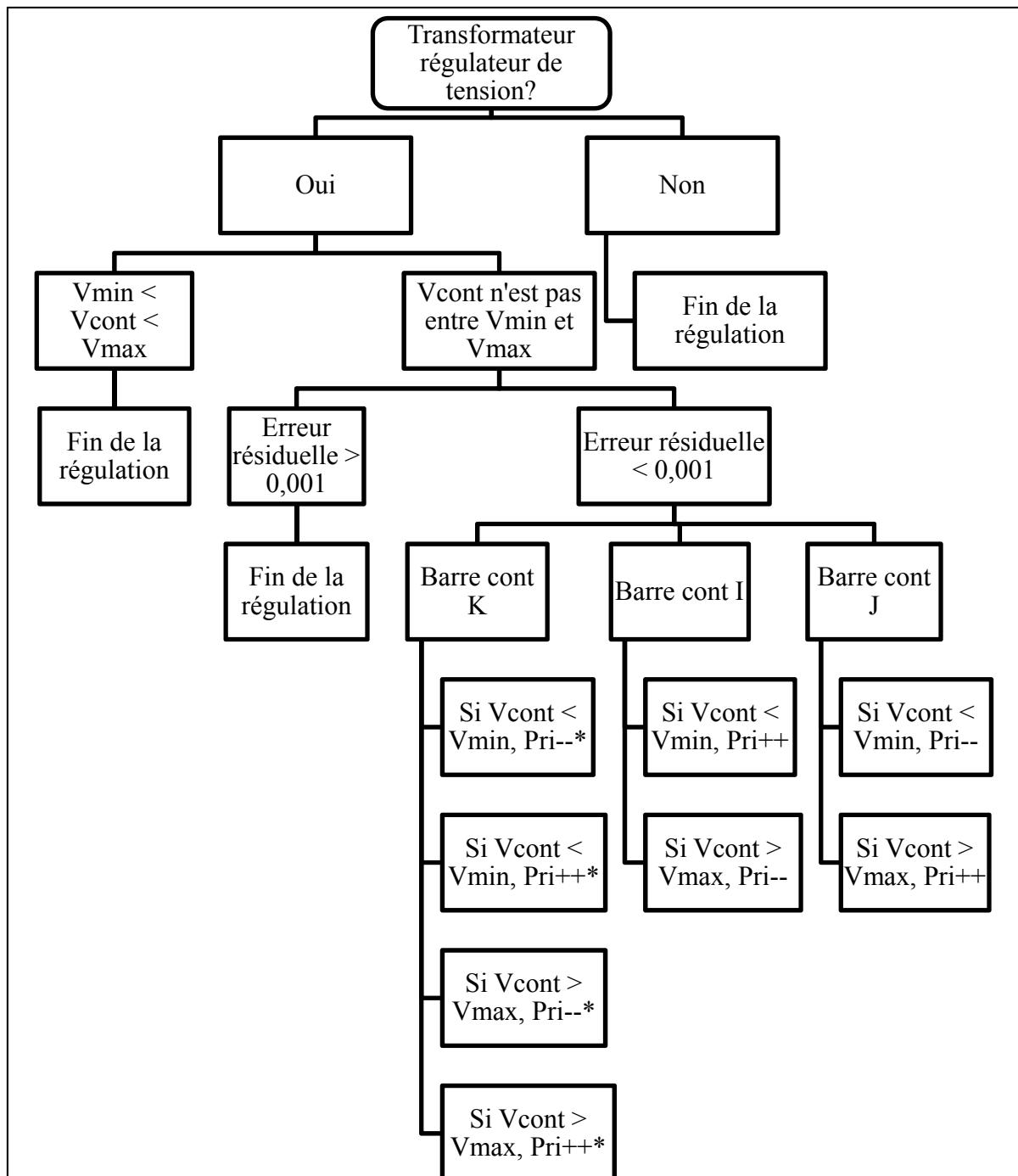
La régulation est une étape qui est essentielle à un réseau électrique. Sans régulation, le réseau ne peut être contrôlé, et la tension fournie par la distribution peut fluctuer. Cela peut causer des bris matériels chez les clients, et peut même être dangereux pour les équipements des postes de distribution si la tension augmente trop. C'est pourquoi le simulateur est muni de transformateurs régulateurs de tension.

2.8.2 Algorithme utilisé pour l'implémentation de la régulation de tension

La régulation est implémentée de façon à ce que le programme fasse les ajustements nécessaires lorsqu'il y a des barres où leurs tensions sont régulées.

L'algorithme utilisé dans le programme sert de modèle pour modifier ou non les prises des transformateurs régulateurs de tension. Cet algorithme permet de comprendre plus simplement comment le programme réagit lorsqu'il s'agit de ce type de transformateur et comment il gère la régulation d'une barre qui est contrôlée n'importe où dans le réseau. Étant donné la taille des conditions à vérifier, l'algorithme est simplifié à la page suivante, mais il y a toute l'information nécessaire à sa fonctionnalité dans le paragraphe qui le suit.

Algorithme 2.1 Schéma représentant l'algorithme des transformateurs régulateurs de tension



2.8.3 Explication de l'algorithme de régulation

Lorsqu'il y a un transformateur régulateur de tension dans le réseau, le simulateur suit différentes conditions à chaque itération. Il faut premièrement vérifier qu'il s'agit bel et bien de ce type de transformateur. Pour ce faire, il faut vérifier que la variable BCNO est assignée à une barre spécifique dans le réseau. Ensuite, le simulateur vérifie si la valeur de la tension se situe entre la valeur minimale et la valeur maximale spécifiée. Si c'est le cas, il n'y a aucune régulation à faire et le simulateur peut passer à la prochaine itération. Si ce n'est pas le cas, il faut entreprendre le processus conditionnel de la boucle de régulation.

La première condition de cette boucle est de vérifier si l'erreur résiduelle est plus petite que 0,001 pu. Si ce n'est pas le cas, le simulateur quitte la boucle de régulation et continue vers la prochaine itération. Si c'est le cas, il faut cependant vérifier quelques conditions avant de continuer. La première condition est de vérifier s'il faut arrondir le nombre de prises, car ce nombre doit obligatoirement être un entier. Donc, si par erreur un nombre réel est entré dans le fichier de lecture, il faut l'arrondir à l'entier le plus près. Ensuite, il faut s'assurer que la prise de départ est bel et bien un nombre plausible pour le nombre de prises donné avec les valeurs de a1min et a1max. Si ce n'est pas le cas, le simulateur arrondi a1 à la valeur la plus proche. Une fois ces deux conditions vérifiées, le simulateur continu sa boucle en vérifiant où se situe la barre contrôlée dans le réseau par rapport au transformateur. Cette barre peut être située au primaire du transformateur (I), au secondaire (J) ou ailleurs dans le réseau (K).

Voici comment l'algorithme fonctionne lorsque la barre contrôlée se situe au primaire du transformateur (I) régulateur de tension. Premièrement, si la tension de la barre contrôlée est plus petite que la tension minimale de régulation, le simulateur change la prise de manière à ce que celle-ci augmente (Ex. : Prise 6 vers 7). Cependant, si la prise est déjà désignée au nombre maximal, celle-ci ne peut être augmentée, et un message mentionnant cette information est affiché dans le rapport. Deuxièmement, si la tension de la barre contrôlée est plus grande que la tension maximale de régulation, le changement de prise ira dans la direction opposée (Ex. : Prise 6 vers 5). Par contre, si la prise est déjà désignée au nombre

minimal, il n'y a aucun ajustement à faire. De plus, si la dernière condition est vraie, il y aura un message dans le rapport mentionnant qu'il y a une tentative de changer la prise en dessous de la valeur minimale.

Par contre, si la barre contrôlée se situe au secondaire du transformateur (J), le changement de prise se fait exactement à l'inverse que lorsque la barre contrôlée se situe au primaire. Donc, si la tension de cette barre est plus petite que la tension minimale de régulation (V_{min}), le simulateur change de prise en direction de sa valeur minimum. Il y a encore une fois vérification de la prise minimale, et un message correspondant, si tel est le cas. Pour le cas opposé, c'est exactement le contraire qu'au paragraphe précédent.

Enfin, si la barre contrôlée se situe ailleurs dans le réseau (K), plusieurs conditions sont vérifiées. Premièrement, étant donné le fait que cette barre peut être n'importe où dans le réseau, si la tension de la barre contrôlée est plus petite que la tension minimale, le simulateur augmentera la prise du transformateur régulateur de tension. De plus, une fois les tensions calculées à la fin d'une itération du programme, le simulateur garde en mémoire les valeurs des transformateurs régulateurs afin de pouvoir les vérifier à la boucle suivante. En effet, lors de l'itération suivante, une comparaison est effectuée afin de savoir si la tension de la barre contrôlée s'est approchée ou non de la valeur de régulation minimale requise. Si le contraire s'est produit (que la tension de la barre contrôlée s'éloigne de la valeur minimale), le changement de prise se fait dans l'autre sens. Deuxièmement, si la tension de la barre contrôlée est plus grande que la valeur maximale, le simulateur diminue la prise du transformateur. Si comme dans le cas précédent, la valeur de la tension de la barre s'éloigne, le changement de prise se fait dans l'autre sens. Cependant, s'il y a un effet oscillatoire (augmentation de la prise, ensuite diminution, et ensuite augmentation de la prise), le simulateur arrête de faire cette oscillation après quelques tentatives. Cela veut dire qu'il est incapable de satisfaire aux valeurs minimales et maximales de régulation. Cela peut être dû à des valeurs incohérentes ou à des valeurs identiques qui sont lues dans le rapport.

2.8.4 Calcul de la variable a_1 pendant la régulation

Une fois les conditions vérifiées, le simulateur doit changer la matrice d'admittance s'il y a eu un changement de prise, car la valeur de la variable a_1 a changé.

Tout d'abord, il faut déterminer la valeur initiale des prises par rapport aux valeurs entrées dans le fichier de lecture. En sachant quelles sont les valeurs de $a_1\text{min}$, $a_1\text{max}$, a_1 , ainsi que du nombre de prises du transformateur. Le simulateur calcule la position initiale de la prise à partir de l'équation suivante :

$$\text{Prise initiale} = \frac{(a_1 - a_1\text{min}) * (Nprise - 1)}{(a_1\text{max} - a_1\text{min})} + 1 \quad (2.103)$$

Une fois le nombre de prises initiales calculé, et tout dépendant de l'acheminement de l'algorithme, le simulateur fait différentes opérations afin de modifier a_1 pour les transformateurs changeurs de prises. Voici l'équation utilisée afin de déterminer comment changer de prises en utilisant l'équation précédente :

$$a_1 = \frac{(\text{Prise initiale} - 1) * (a_1\text{max} - a_1\text{min})}{(Nprise - 1)} + a_1\text{min} \quad (2.104)$$

Cependant, cette équation fonctionne seulement pour avoir la valeur selon la prise initiale. Donc, si le simulateur doit augmenter la prise du transformateur, il doit augmenter de 1 la valeur de la prise initiale. Donc l'équation précédente devient l'équation suivante :

$$a_1 = \frac{(\text{Prise initiale}) * (a_1\text{max} - a_1\text{min})}{(Nprise - 1)} + a_1\text{min} \quad (2.105)$$

Toutefois, si le simulateur doit diminuer la prise du transformateur, il doit diminuer de 1 la valeur de la prise initiale, c'est-à-dire l'équation suivante :

$$a_1 = \frac{(Prise \text{ } initiale - 2) * (a_1 \text{ } max - a_1 \text{ } min)}{(Nprise - 1)} + a_1 \text{ } min \quad (2.106)$$

Ces deux équations fonctionnent pour une barre qui est contrôlée au primaire ou au secondaire d'un transformateur régulateur de tension. Par contre, ce n'est pas le cas lorsque la barre est ailleurs dans le réseau. De ce fait, le simulateur recalcule pour savoir s'il faut augmenter ou diminuer la prise du transformateur, et finalement utilise la première équation automatiquement pour les autres cas une fois la nouvelle prise calculée. Si après une itération, la valeur des tensions s'éloigne des valeurs de régulation, le simulateur fait tout simplement l'inverse afin de s'en rapprocher.

2.8.5 Calcul de la matrice d'admittance avec la régulation

Une fois la variable a_1 calculée, le simulateur doit changer la matrice d'admittance en utilisant celle-ci stockée en mémoire et en faisant les opérations nécessaires pour a_1 et a_2 avec les barres correspondantes des transformateurs régulateurs de tension. Donc que ce soit pour des barres monophasées ou triphasées, il faut recalculer la matrice d'admittance du réseau lorsqu'il y a un changement de prise. Cela fait varier la tension des barres correspondantes à la régulation.

Voici les opérations mathématiques (Pierre-Jean Lagacé, 2011) lorsqu'il s'agit d'un transformateur monophasé avec a_1 et a_2 :

$$Y_{bus, mono}(2x2) = \begin{bmatrix} \frac{Y_{11}}{a_1^2} & \frac{Y_{12}}{(a_2^*) * a_1} \\ \frac{Y_{22}}{(a_1^*) a_2} & \frac{Y_{11}}{a_2^2} \end{bmatrix} \quad (2.107)$$

Pour une barre triphasée, la matrice est beaucoup plus grosse et dense, mais il faut quand même simplement insérer a_1 et a_2 aux bons endroits dans la matrice lorsqu'il s'agit de transformateur régulateur de tension.

Voici la matrice d'admittance d'un transformateur (Pierre-Jean Lagacé, 2011) en y ajoutant a_1 et a_2 lorsqu'il y a ce type de transformateur (Yd0 et Yy0) :

$$Y_{transfo}(6 \times 6) = \begin{bmatrix} \frac{Y_{11}}{a_1^* a_1} & \frac{Y_{12}}{a_1^* a_1} & \frac{Y_{13}}{a_1^* a_1} & \frac{Y_{14}}{a_1 a_2^*} & \frac{Y_{15}}{a_1 a_2^*} & \frac{Y_{16}}{a_1 a_2^*} \\ \frac{Y_{21}}{a_1^* a_1} & \frac{Y_{22}}{a_1^* a_1} & \frac{Y_{23}}{a_1^* a_1} & \frac{Y_{24}}{a_1 a_2^*} & \frac{Y_{25}}{a_1 a_2^*} & \frac{Y_{26}}{a_1 a_2^*} \\ \frac{Y_{31}}{a_1^* a_1} & \frac{Y_{32}}{a_1^* a_1} & \frac{Y_{33}}{a_1^* a_1} & \frac{Y_{34}}{a_1 a_2^*} & \frac{Y_{35}}{a_1 a_2^*} & \frac{Y_{36}}{a_1 a_2^*} \\ \frac{Y_{41}}{a_1^* a_1} & \frac{Y_{42}}{a_1^* a_1} & \frac{Y_{43}}{a_1^* a_1} & \frac{Y_{44}}{a_1 a_2^*} & \frac{Y_{45}}{a_1 a_2^*} & \frac{Y_{44}}{a_1 a_2^*} \\ \frac{Y_{51}}{a_2 a_1^*} & \frac{Y_{52}}{a_2 a_1^*} & \frac{Y_{53}}{a_2 a_1^*} & \frac{Y_{54}}{a_2^* a_2} & \frac{Y_{55}}{a_2^* a_2} & \frac{Y_{56}}{a_2^* a_2} \\ \frac{Y_{61}}{a_2 a_1^*} & \frac{Y_{62}}{a_2 a_1^*} & \frac{Y_{63}}{a_2 a_1^*} & \frac{Y_{64}}{a_2^* a_2} & \frac{Y_{65}}{a_2^* a_2} & \frac{Y_{66}}{a_2^* a_2} \end{bmatrix} \quad (2.108)$$

Il est facile de remarquer que la matrice d'admittance est semblable à celle du transformateur monophasé, sauf qu'il faut y ajouter a_1 et a_2 pour les trois phases. Lorsque la matrice est recalculée, il faut tout simplement remplacer les valeurs correspondantes dans la matrice d'admittance totale du réseau.

2.9 Lecture du rapport

2.9.1 Structure du rapport

Le rapport est séparé en plusieurs parties. Premièrement, il est créé seulement à la fin de l'écoulement de puissance qu'il y ait convergence ou non. Cependant, pour un rapport complet, il doit y avoir eu une convergence. Au début du rapport se trouve la méthode utilisée pour l'écoulement de puissance, soit Newton-Raphson ou Levenberg-Marquardt, et par la suite, il y a l'ensemble des données du réseau avec le nombre pour chacun des items qui sont lus dans le fichier de données.

L'étape suivante est la boucle itérative. Dans cette étape, il y a pour chacune des itérations les erreurs résiduelles des puissances actives, réactives et apparentes, ainsi que les barres où elles ont été calculées. De plus, il y a la valeur de la variable lambda qui est calculée lorsque le simulateur utilise la méthode Levenberg-Marquardt. (Elle est indiquée aussi pour la méthode Newton-Raphson.) Ensuite, lorsqu'il y a de la régulation, les informations de celle-ci sont rajoutées pour chacune des itérations.

Par la suite, il y a les résultats pour l'ensemble des barres s'il y a convergence une fois la simulation terminée. Les informations suivantes sont indiquées pour chacune de ses phases de chaque barre. Il y a la valeur initiale et finale des tensions et angles de celles-ci par rapport à la barre d'équilibre. Ensuite, s'il y a lieu, il y a les différents types de puissances correspondantes aux charges appropriées, soit la puissance pour une charge à impédance constante (Sshunt), la puissance pour une charge à courant constant (Schg_I), la puissance pour une charge à puissance constante (Schg), et la puissance pour une charge ayant un facteur NP constant (Schg_NP).

Finalement, à la fin du rapport, il y a les puissances pour les branches du réseau qu'il soit monophasé ou triphasé. Ces branches sont indiquées de manière à savoir quelles sont les barres connectées à leurs débuts et à leurs fins. Il est donc possible de valider les résultats, car en additionnant l'ensemble des puissances passant par une barre, la résultante doit être zéro.

2.10 Conclusion

Avec tous les outils abordés au cours de ce chapitre, il est maintenant facile de créer des fichiers de données, de modifier le code source au besoin, de faire différentes simulations d'un grand nombre de réseaux, et de bien comprendre le rapport final.

Une étude de cas exhaustive permettra de valider le fonctionnement du simulateur qui deviendra un outil additionnel pour l'équipe de recherche de l'IREQ.

CHAPITRE 3

ÉTUDE DE CAS

3.1 Introduction

Ce chapitre présente les résultats obtenus avec le simulateur. De plus, afin de faciliter le travail, les résultats sont comparés avec l'utilisation d'un prototype fait en MatLab. Celui-ci a été validé à partir du logiciel libre PSAT (F. Milano, 2005), c'est-à-dire que les résultats obtenus par le prototype sont les mêmes que ceux obtenus avec le logiciel libre. La validation des résultats est très importante afin de tirer de bonnes conclusions sur la fiabilité du simulateur. Pour chacun des cas étudiés, il y a plusieurs étapes à suivre. Tout d'abord, ils sont tous représentés par un schéma démontrant comment les barres sont reliées les unes par rapport aux autres. Par la suite, il y a une brève explication du fichier de lecture. Ensuite sont présentés les résultats obtenus des itérations avec la méthode Newton-Raphson (pour Levenberg-Marquardt, les résultats sont présentés en ANNEXE pour chaque cas) pour le simulateur, ainsi que pour le prototype. Il y a aussi un tableau comparant les valeurs absolues et les angles des tensions pour l'ensemble des barres, ainsi qu'une vérification de la puissance générée vis-à-vis celle qui se trouve dans le réseau. Finalement, il y a la discussion des résultats où différentes conclusions peuvent être tirées.

Pour chaque figure associée à un cas, l'ensemble des composants est indiqué comme s'ils sont des composants monophasés. C'est seulement dans le fichier de données qu'il faut bien indiquer si les données sont monophasées ou triphasées.

Pour le troisième cas, ainsi que pour le dernier cas, l'inverse de la norme de Frobenius de la matrice Jacobienne multiplié par la norme de Frobenius de cette même matrice est calculé afin d'en voir son conditionnement. Plus l'indice est élevé, plus la matrice est mal conditionnée. Voici l'équation permettant de calculer cette norme (V. Chellaboina et coll., 1995) :

$$||A||_F = \sqrt{\text{tr}(AA^*)} \quad (3.1)$$

Où tr est la somme des valeurs de la diagonale d'une matrice.

3.2 Cas #1 : Réseau simple avec deux barres triphasées PQ à impédance constante

Ce premier réseau sert à valider la capacité du simulateur à faire le calcul de barres avec des charges à impédances constantes. Voici le schéma correspondant :

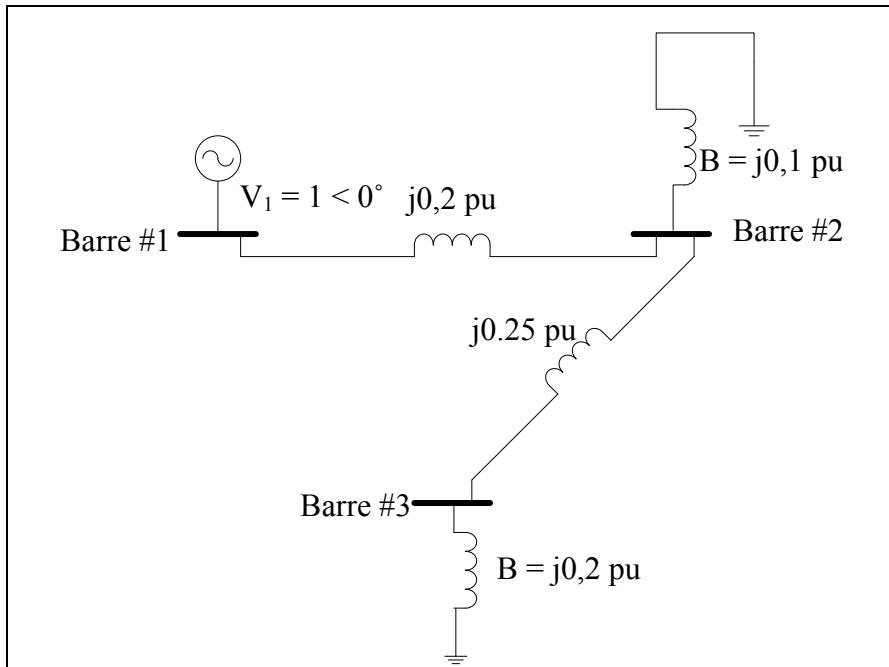


Figure 3.1 Réseau avec deux barres PQ en série (Cas #1)

À partir de ce schéma, il est facile de faire un fichier de lecture avec les différentes nomenclatures expliquées au chapitre 2 de ce mémoire.

3.2.1 Éléments à insérer dans le fichier de données

Voici en quelques lignes les paramètres importants pour ce cas. Premièrement, il faut mettre les valeurs des modules de tensions des trois barres aux valeurs initiales, c'est-à-dire 1 pu. Ensuite, il faut indiquer qu'il y a des susceptances aux barres de charges de 0,1 pu et de 0,2 pu pour les barres 2 et 3 respectivement. De plus, il faut en indiquer le type. La première est une barre d'équilibre. Il faut donc indiquer que c'est une barre de type 1 dans le fichier de lecture. Pour ce qui est des deux autres, ce sont des barres de charges. Il faut donc indiquer que ces barres sont de type 3. Une fois cela fait, il faut indiquer quels sont les paramètres des branches du réseau. En premier lieu, il faut indiquer le départ, ainsi que l'arrivée de la ligne correspondante. Ensuite, il faut indiquer sa composition, soit de résistances, d'inductances, de condensateurs, ou de susceptances. Dans notre cas, il faut indiquer que les lignes sont composées de réactances de 0,2 pu et de 0,25 pu pour les branches allant de la barre 1 à 2 et 2 à 3 respectivement (*Voir ANNEXE II, p.107, Tableau-A II-1 et Tableau-A II-2*). Une fois fait, il est maintenant possible de démarrer la simulation et d'en faire la comparaison avec le prototype de la méthode Newton-Raphson.

3.2.2 Résultats obtenus

Voici les résultats des itérations entre les deux programmes :

Tableau 3.1 Itérations du prototype (Cas #1)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	$3,0809 \times 10^{-15}$ (2b)	$-2,0000 \times 10^{-01}$ (3a)	$3,8729 \times 10^{-01}$
1	$-4,2284 \times 10^{-15}$ (2b)	$3,1346 \times 10^{-02}$ (3c)	$5,6010 \times 10^{-02}$
2	$5,0998 \times 10^{-15}$ (2b)	$4,1384 \times 10^{-04}$ (3a)	$7,2340 \times 10^{-04}$
3	$4,0031 \times 10^{-15}$ (2c)	$7,2228 \times 10^{-08}$ (3a)	$1,2539 \times 10^{-07}$
4	$-2,9508 \times 10^{-15}$ (2b)	$-3,2170 \times 10^{-15}$ (2b)	$7,2429 \times 10^{-15}$

Tableau 3.2 Itérations du simulateur (Cas #1)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	$1,526 \times 10^{-16}$ (3-B)	-0,2 (3-A)	0,387298
1	$1,284 \times 10^{-16}$ (3-B)	0,0313463 (3-B)	0,0560103
2	$4,456 \times 10^{-16}$ (2-B)	0,000413845 (3-A)	0,000723408
3	$6,327 \times 10^{-17}$ (2-B)	$7,2278 \times 10^{-08}$ (3-A)	$1,25478 \times 10^{-07}$
4	$4,905 \times 10^{-16}$ (2-B)	$5,682 \times 10^{-15}$ (2-A)	$1,13771 \times 10^{-14}$

Il faut maintenant examiner les résultats obtenus pour les tensions des barres (modules et angles). Voici un tableau démontrant les résultats, quel que soit la méthode utilisée (ils sont identiques) et ceux du prototype.

Tableau 3.3 Valeurs des tensions (Cas #1)

Barre#	Simulateur		Prototype	
	Module de tension	Angle	Module de tension	Angle
1a	1	0	1	0
1b	1	-120	1	-120
1c	1	120	1	120
2a	1,06622	0	1,06622	0
2b	1,06622	-120	1,06622	-120
2c	1,06622	120	1,06622	120
3a	1,12233	0	1,12233	0
3b	1,12233	-120	1,12233	-120
3c	1,12233	120	1,12233	120

3.2.3 Validation des résultats

Il est maintenant important de vérifier les puissances aux barres respectives pour être certain que les résultats sont les mêmes (*Voir ANNEXE II, p.108, Tableau-A II-4 et Tableau-A II-5*). Pour la barre d'équilibre, la puissance générée est de -0,993266 pu, et c'est exactement la même réponse pour le prototype. Pour la première barre de charge, le simulateur calcule la puissance pour chaque phase de -0,113682 pu. Le prototype calcule encore une fois

exactement la même réponse. Finalement, pour la dernière barre de charge, le simulateur, ainsi que le prototype arrivent à une puissance réactive de -0,251927 pu.

Pour valider les résultats, il faut multiplier par trois les puissances des barres de charges et les additionner (ce qui donne -1,096827 pu). Ensuite, il faut enlever les pertes des branches. Pour ce qui en est de la branche partant de la barre d'équilibre à la première barre de charge, les pertes totales des trois phases sont de 0,065772 pu. Pour la deuxième branche, et en procédant de la même façon, le total des pertes est de 0,037788 pu. En faisant l'addition de ces trois résultats, la réponse donne exactement la puissance générée par la barre d'équilibre.

3.2.4 Discussion et interprétation des résultats

Les résultats des tensions des barres obtenus sont exactement les mêmes pour le simulateur et pour le prototype. Cela démontre que le simulateur est capable d'intégrer des barres de charges à impédances constantes. De plus, il est possible de conclure que le simulateur donne les mêmes résultats que le prototype pour ce qui a trait aux itérations. En utilisant la méthode Levenberg-Marquardt, les résultats sont encore une fois concluants (*Voir ANNEXE II, p.107 pour l'ensemble des résultats*). La différence qu'il y a lors des itérations est que les erreurs résiduelles ne sont pas exactement les mêmes pour les deux simulateurs, car les calculs employés ne sont pas tout à fait exécutés en même temps. Néanmoins, le simulateur converge vers les bons résultats en utilisant les deux méthodes. De plus, la validation permet de s'assurer que toutes les puissances générées sont bien présentes dans l'ensemble du réseau. Il est possible de voir qu'il n'y a aucun déphasage entre les phases a, b, et c des trois barres. Cela est dû au fait qu'il n'y a aucune puissance active dans le réseau.

3.3 Cas #2 : Réseau triphasé avec une barre PV et une PQ à puissance constante

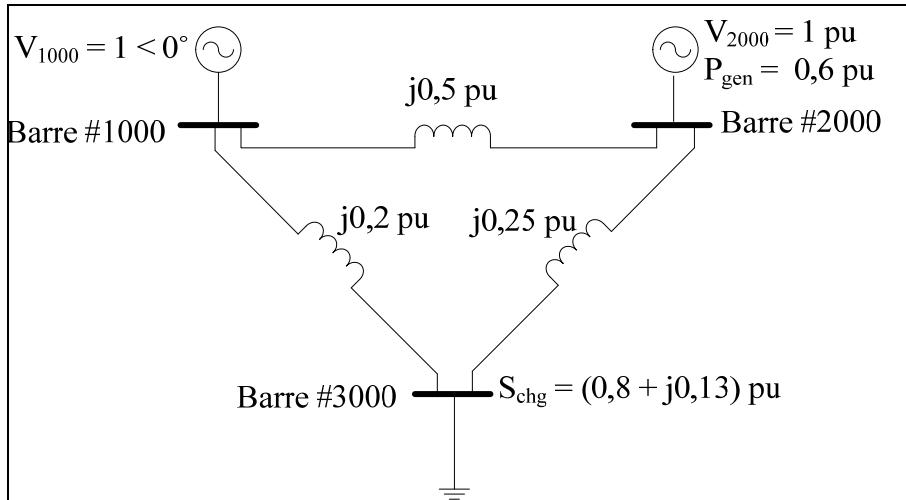


Figure 3.2 Réseau triphasé avec une barre PV et une PQ à puissance constante (Cas #2)

Dans ce cas, il est possible de vérifier si les barres de génération sont bien implémentées. De plus, les numéros des barres sont différents que pour le cas précédent. Il est donc possible de vérifier si le simulateur comprend bien même si les numéros des barres ne sont pas une suite logique de chiffres commençant par 1. Finalement, ce cas permet aussi de vérifier si les barres de charges à puissance constante sont bien implémentées.

3.3.1 Éléments à insérer dans le fichier de données

Pour la construction du fichier de lecture, il faut simplement entrer les paramètres aux bons endroits, et bien indiquer les différentes valeurs. Pour la barre d'équilibre, c'est exactement le même processus que pour le cas précédent, mais, pour ce cas, son numéro est 1 000. Pour la barre de génération (Type 2), il faut indiquer 2 000 comme numéro de barre, et 0,6 pu comme étant la puissance qui est générée dans le fichier de lecture. Pour la barre de charge, c'est pratiquement la même chose que pour le cas ayant une impédance constante, mais il faut changer d'endroit où il faut insérer les valeurs. Il faut donc indiquer 0,8 pu à l'onglet Pchg, et 0,13 pu à l'onglet Qchg du fichier de données. De plus, il faut indiquer que le numéro de la

barre est 3 000. Pour ce qui est des branches, c'est la même chose que pour le premier cas, donc aucune explication n'est nécessaire (*Voir ANNEXE III, p.109, Tableau-A III-1 et Tableau-A III-2*).

3.3.2 Résultats obtenus

Voici les erreurs résiduelles des puissances pour chaque itération :

Tableau 3.4 Itérations du prototype (Cas #2)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	-1,80000 (2000a)	1,30 x 10 ⁻⁰¹ (3000a)	2,28269
1	-2,47 x 10 ⁻⁰² (2000a)	4,06 x 10 ⁻⁰² (3000c)	7,7738 x 10 ⁻⁰²
2	-1,36 x 10 ⁻⁰⁴ (2000a)	2,16 x 10 ⁻⁰⁴ (3000a)	4,2355 x 10 ⁻⁰⁴
3	-4,30 x 10 ⁻⁰⁹ (2000a)	6,45 x 10 ⁻⁰⁹ (3000b)	1,2930 x 10 ⁻⁰⁸
4	4,22 x 10 ⁻¹⁵ (3000c)	-1,67 x 10 ⁻¹⁵ (3000b)	5,3204 x 10 ⁻¹⁵

Tableau 3.5 Itérations du simulateur (Cas #2)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	-1,8000 (2000-A)	0,1300 (3000-A)	2,2827
1	-0,02467 (2000-A)	0,0405517 (3000-C)	0,0777385
2	1,36 x 10 ⁻⁰⁴ (2000-A)	0,0002160 (3000-A)	0,000423545
3	-4,307 x 10 ⁻⁰⁹ (2000-A)	6,4500 x 10 ⁻⁰⁹ (3000-B)	1,29384 x 10 ⁻⁰⁸
4	1,1102 x 10 ⁻¹⁵ (2000-A)	-5551 x 10 ⁻¹⁶ (3000-B)	1,82428 x 10 ⁻¹⁵

Maintenant, il est temps de comparer les résultats des tensions finales du simulateur avec ceux obtenus à l'aide du prototype.

Tableau 3.6 Valeurs des tensions (Cas #2)

Barre#	Simulateur		Prototype	
	Module de tension	Angle	Module de tension	Angle
1000a	1	0	1	0
1000b	1	-120	1	-120
1000c	1	120	1	120
2000a	1	3,36	0,99999	3,36
2000b	1	-116,64	0,99999	-116,64
2000c	1	123,36	0,99999	123,36
3000a	0,98073	-3,71	0,98073	-3,71
3000b	0,98073	-123,71	0,98073	-123,71
3000c	0,98073	116,29	0,98073	116,29

3.3.3 Validation des résultats

Il faut maintenant regarder les résultats des puissances afin de s'assurer que la barre de charge a une puissance constante et aussi afin de valider les résultats (*Voir ANNEXE III, p.110, Tableau-A IV-4 et Tableau-A IV-5*). Le simulateur obtient une puissance générée par la barre d'équilibre de $(0,6 + j 0,330185)$ pu. De plus, la barre de génération génère une puissance de $(1,8 + j 0,331021)$ pu. Sachant qu'à la barre de charge il y a une puissance constante de $(0,8 + j 0,13)$ pu par phase, il est facile de remarquer que la puissance active totale de la charge correspond à la puissance totale générée du réseau. De plus, étant donné qu'il n'y a aucune perte résistive dans le réseau, cela valide la puissance active. Pour la puissance réactive totale générée, il faut additionner la puissance réactive de la barre de génération avec celle de la barre d'équilibre. Cela donne une puissance réactive totale générée de 0,661206 pu. Sachant que la puissance réactive totale des charges est de 0,39, le reste de la puissance est censée se trouver dans les branches du circuit. Donc, en vérifiant la branche reliant la barre d'équilibre à la barre de génération, il y a 0,02062848 pu pour l'ensemble des lignes. Ensuite, pour la branche reliant la barre d'équilibre à la barre de charge, il y a un total de 0,0671967 pu dans les lignes. Enfin, pour la branche reliant la barre de génération à la barre de charge, il y a un total de 0,1833783 pu. Cela fait donc un total de

puissance réactive dans les lignes de $0,27120348$ pu. En additionnant cette valeur à $0,39$ pu, le résultat est bel et bien le même que la puissance générée par les barres de génération.

3.3.4 Discussion et interprétation des résultats

Pour ce cas, il est possible de voir que la tension de chacune des phases est constante, car la charge est équilibrée ($0,8 + 0,13j$) pu par phase. Cela permet aussi de conclure que les angles entre les phases ont un déphasage constant de 120 degrés. De plus, la tension de la barre de génération est constante (1 pu), car c'est une des variables qui est connue dans le problème, et elle doit être constante pour l'ensemble de ses phases. En regardant les tableaux des itérations des deux simulateurs, il est possible de constater que lorsque les valeurs sont plus petites que 1×10^{-10} , les deux valeurs ne sont plus égales. Cela est probablement dû au fait que le simulateur ne fait pas exactement tous les calculs en même temps que le prototype. En sachant cela, il est donc possible de conclure que le simulateur est capable d'intégrer des barres de génération ainsi que des barres de charges à puissances constantes (*Voir ANNEXE III, p.109 pour l'ensemble des résultats avec la méthode Levenberg-Marquardt*).

3.4 Cas #3 : Réseau triphasé avec implémentation d'une ligne monophasée (une barre monophasée)

Ce cas démontre la possibilité d'intégrer un réseau de distribution avec un réseau de transport. Le plus gros problème avec ce type de réseau est que les impédances du réseau de transport sont beaucoup plus grandes que celles utilisées pour le réseau de distribution. Pour ce cas, il faut changer en pu l'ensemble des composantes, et faire des simulations avec une seule barre monophasée qui est par la suite divisée en plusieurs barres. Il faut alors comparer les résultats avec le réseau composé d'une seule barre monophasée. Cela permet de vérifier si le simulateur est capable de faire l'écoulement de puissance pur des réseaux mal conditionnés. Voici le réseau avec une seule barre de charge monophasée connectée à un réseau triphasé :

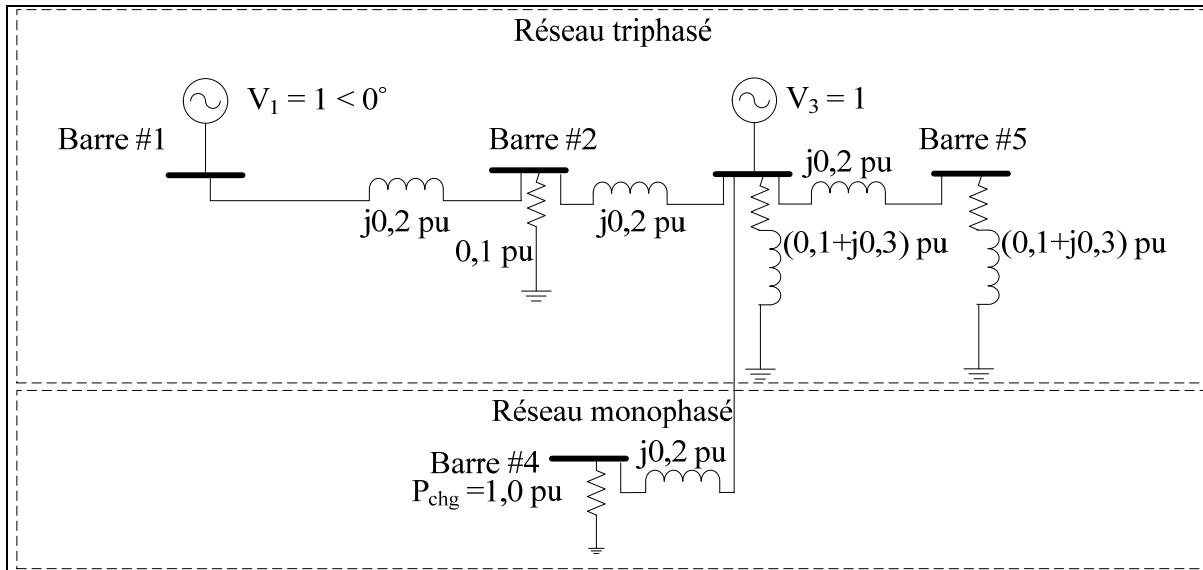


Figure 3.3 Réseau triphasé avec une barre PQ monophasé (Cas #3)

3.4.1 Éléments à insérer dans le fichier de données

Comme mentionné au chapitre concernant la construction des fichiers de données, il faut bien indiquer les types de composantes qui se trouvent dans le réseau. Dans la figure ci-haut, il faut s'assurer de séparer les composantes triphasées des composantes monophasées. Pour ce qui en est de la construction du fichier de données, c'est la même procédure que dans les deux cas précédents, c'est-à-dire qu'il faut indiquer les composantes connectées sur les lignes, ainsi que sur les barres (*Voir ANNEXE IV, p.111, Tableau-A IV-1 à 4*).

3.4.2 Résultats obtenus

Il faut maintenant vérifier si les itérations du prototype sont les mêmes que les itérations du simulateur en utilisant la méthode Newton-Raphson. Voici les erreurs résiduelles des puissances pour chaque itération :

Tableau 3.7 Itérations du prototype (Cas #3)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	1,0000 (5a)	-3,000 x 10 ⁻⁰¹ (4b)	1,19163
1	1,9829 x 10 ⁻⁰² (3)	9,96671 x 10 ⁻⁰² (5a)	1,6603 x 10 ⁻⁰¹
2	5,74726 x 10 ⁻⁰⁴ (5a)	2,08658 x 10 ⁻⁰³ (5a)	2,3807 x 10 ⁻⁰³
3	5,31474 x 10 ⁻⁰⁷ (5a)	1,11869 x 10 ⁻⁰⁶ (5a)	1,34405 x 10 ⁻⁰⁶
4	2,28817 x 10 ⁻¹³ (5a)	3,67483 x 10 ⁻¹³ (5a)	4,88990 x 10 ⁻¹³

Tableau 3.8 Itérations du simulateur (Cas #3)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	1,0000 (5-A)	-0,30000 (4-A)	1,19164
1	0,01983 (3-A)	0,0996671 (5-A)	0,166032
2	0,000574727 (5-A)	0,00208658 (5-A)	0,00238075
3	5,314 x 10 ⁻⁰⁷ (5-A)	1,1187 x 10 ⁻⁰⁶ (5-A)	1,34401 x 10 ⁻⁰⁶
4	-2,2643 x 10 ⁻¹³ (3-A)	3,6010 x 10 ⁻¹³ (5-A)	4,8182 x 10 ⁻¹³

Il faut aussi vérifier les résultats des tensions des barres obtenus par le simulateur vis-à-vis ceux obtenus par le prototype. De plus, ce sera possible de valider avec ces résultats une fois la barre monophasée divisée afin de voir si le système est capable de converger ou non lorsque le réseau est mal conditionné.

Tableau 3.9 Valeurs des tensions (une barre monophasée) (Cas #3)

Barre#	Simulateur		Prototype	
	Module de tension	Angle	Module de tension	Angle
1a	1	0	1	0
1b	1	-120	1	-120
1c	1	120	1	120
2a	0,992721	-7,47	0,992721	-7,47
2b	0,992721	-127,5	0,992721	-127,5
2c	0,992721	112,53	0,992721	112,53
3a	1	-13,79	1	-13,79
3b	1	-133,8	1	-133,8
3c	1	106,21	1	106,21
4a	0.978906	-25,58	0.978906	-25,58
5a	1,06359	-15,01	1,06359	-15,01
5b	1,06359	-135,0	1,06359	-135,0
5c	1,06359	104,99	1,06359	104,99

3.4.3 Cas #3.2 : Réseau triphasé avec implémentation d'une ligne monophasée (80 barres monophasées)

Évidemment, un réseau de distribution a beaucoup plus qu'une seule barre, et c'est pourquoi il faut faire une simulation en divisant la barre monophasée en multiples barres monophasées, et observer si les résultats des barres triphasées concordent avec celles indiquées dans le tableau précédent.

En se fiant aux paramètres de départ avec le réseau monophasé à une seule barre, il faut subdiviser les valeurs de la réactance de ligne, ainsi que la valeur de la charge à puissance constante de la barre. En reprenant l'idée de la figure du cas avec une seule barre monophasée, voici la figure correspondant à un réseau de distribution de plusieurs barres monophasées :

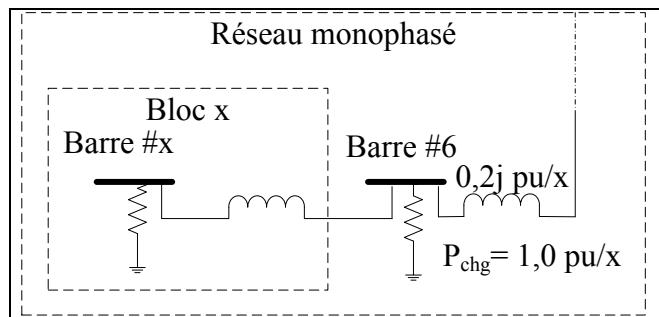


Figure 3.4 Réseau avec x barres monophasées (Cas #3)

Cette figure montre un réseau avec plusieurs barres ayant comme caractéristiques une impédance de ligne, ainsi qu'une charge à puissance constante correspondant aux valeurs de départ de la barre monophasée divisée par un chiffre aléatoire (x). Voici les résultats obtenus avec le simulateur lorsqu'il y a 80 barres monophasées ayant une charge à puissance constante de 0,0125 pu. De plus, les valeurs des impédances de lignes ont aussi été divisées par 80, ce qui donne une impédance de 0,0025j pu par ligne.

3.4.4 Éléments à insérer dans le fichier de données

Il faut donc créer 79 nouvelles barres monophasées et insérer la puissance constante calculée ci-haut sous l'onglet « Pchg ». Ensuite, il faut créer 79 nouvelles lignes monophasées et insérer l'impédance calculée de 0,0025 pu sous l'onglet « reac ». Finalement, il faut insérer les numéros d'entrant et de sortant pour chacune d'entre elles. Cela vient donc de transformer le réseau monophasé d'une barre à un réseau monophasé de 80 barres ayant exactement la même puissance (*Voir ANNEXE IV, p.113, Tableau-A IV-7 et Tableau-A IV-8*).

3.4.5 Résultats obtenus

Tableau 3.10 Valeurs des tensions (80 barres monophasées) (Cas #3)

Barre#	Simulateur		Prototype	
	Module de tension	Angle	Module de tension	Angle
1a	1	0	1	0
1b	1	-120	1	-120
1c	1	120	1	120
2a	0,992721	-7,47	0,992721	-7,47
2b	0,992721	-127,5	0,992721	-127,5
2c	0,992721	112,53	0,992721	112,53
3a	1	-13,79	1	-13,79
3b	1	-133,8	1	-133,8
3c	1	106,21	1	106,21
4a	0,999832	-13,93	0,999832	-13,93
5a	0,999671	-14,07	0,999671	-14,07
6a	0,999515	-14,21	0,999515	-14,21
7a	0,999366	-14,35	0,999366	-14,35
8a	0,999222	-14,49	0,999222	-14,49
...
84a	1,06359	-15,01	1,06359	-15,01
84b	1,06359	-135,0	1,06359	-135,0
84c	1,06359	104,99	1,06359	104,99

3.4.6 Validation des résultats

Pour simplifier le travail, la validation des résultats est faite avec une seule barre de charge monophasée. Premièrement, la barre d'équilibre donne la même génération en pu que ce soit pour une barre de charge ou 80 barres de charges monophasées (*Voir ANNEXE IV, p.113 Tableau-A IV-6 et Tableau-A IV-9*). Celle-ci génère une puissance de $(1,93502 + j 0,235443)$ pu. La barre de génération, elle, génère une puissance de $(0 - j 1,44182)$ pu. Cependant, celle-ci est connectée à une charge à impédance constante qui consomme un total de $(0,3 - j 0,9)$ pu. Finalement, la barre de charge triphasée (Barre #2) consomme un total de $(0,0985495 + j 0)$ par phase ou $(0,2956485 + j 0)$ pu total. La deuxième barre de charge

triphasée (Barre #5) consomme quant à elle une puissance de $(0,113122 - j 0,339367)$ pu par phase ou $(0,339366 - j 1,018101)$ pu total. Le total générée est donc de $(1,93502 - j 1,206377)$ pu. Le total consommé par l'ensemble des charges du réseau est de $(1,93502 - j 1,918101)$ pu. Étant donné qu'il n'y a aucune perte résistive dans les lignes, le total des charges égale le total de la génération pour la partie réelle (puissance active).

Pour ce qui est des lignes, il y a des pertes réactives. Premièrement, pour la ligne triphasée partant de la barre #2 à la barre #3, le total des pertes est de $j 0,1818267$ pu. Entre la barre #1 et la barre #2, les pertes sont de $j 0,2533143$ pu. Ensuite, pour la ligne entre la barre #3 et la barre #4, le total est de $j 0,067875$ pu. Finalement pour la branche monophasée, il y a un total de $j 0,208712$ pu. Donc, le total dans les lignes est de $j 0,711728$ pu. Pour la partie réactive, la somme de la partie imaginaire des charges soustraite par la somme de la partie imaginaire de la puissance générée donne donc effectivement zéro.

3.4.7 Discussion et interprétation des résultats

Pour ce qui en est des itérations, que ce soit pour une barre monophasée ou 80 barres monophasées, les résultats des erreurs résiduelles sont exactement les mêmes pour le simulateur ou le prototype. Évidemment, il y a encore l'exception de la quatrième itération où les erreurs sont inférieures à 1×10^{-10} , mais, tel que mentionné précédemment, ce n'est pas un problème. Il faut maintenant comparer les modules et angles des tensions pour le cas d'une seule barre monophasée, ainsi que pour le cas avec 80 barres monophasées. Dans les deux cas, les trois barres triphasées ont convergé vers les mêmes résultats. Cela montre donc que le simulateur est capable de calculer l'écoulement de puissance d'un réseau de transport avec un réseau de distribution, et de converger en quatre itérations pour les deux cas. Cependant, il faut faire attention, car sachant que les impédances des réseaux de transport sont beaucoup plus grandes que celles des réseaux de distribution, une simple division par 80 est probablement loin d'être la réalité. Il faudrait donc, refaire cette expérience en divisant par 1 000 ou même 10 000 pour se rapprocher de la réalité. Rien ne dit que le simulateur convergerait aux bons résultats en divisant par des chiffres d'aussi grande échelle. En

regardant la multiplication des normes pour les deux cas, et sachant que plus la norme s'éloigne de 1, plus le réseau est mal conditionné, la norme de la première situation donne un résultat de 54,0188 tandis que pour la deuxième situation, la norme est de $2,2489 \times 10^5$. Il est donc possible de conclure que le simulateur est capable de résoudre des réseaux mal conditionnés lorsque la norme est de cette échelle. Cela dit, afin de voir l'avantage de l'intégration de la méthode Levenberg-Marquardt dans le simulateur (cas #7), il faut regarder sur un grand réseau triphasé (3 000 barres) où la multiplication des normes est plus grande.

3.5 Cas #4 : Deux barres triphasées avec une barre monophasée munie d'un transformateur changeur de prise (augmentation de prise)

Voici le réseau triphasé avec une seule barre de charge monophasée reliée par un transformateur changeur de prises :

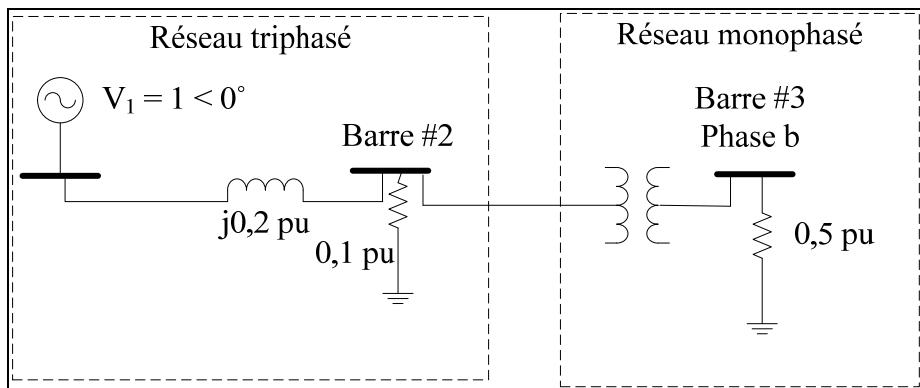


Figure 3.5 Deux barres triphasées avec une barre monophasée munie d'un transformateur changeur de prise (Cas #4)

3.5.1 Éléments à insérer dans le fichier de rapport

Dans ce cas-ci, il faut bien comprendre comment construire le fichier de données afin de faire l'écoulement de puissance correctement. Pour le réseau triphasé, c'est exactement la même procédure que pour les deux premiers cas. Cependant pour le réseau monophasé, il n'y a pas de ligne dans ce réseau. Une fois fait, il faut écrire les valeurs correspondantes au réseau comme vu au chapitre 2. Il est très important de bien indiquer les phases voulues pour faire

l'écoulement de puissance. Dans ce cas précis, le transformateur est connecté entre la phase de la barre 2 et sur la phase b de la barre 3. Une fois fait, il faut indiquer les différentes valeurs des impédances du transformateur Z_1 , Z_2 , et Y . Dans ce cas, les valeurs entrées sont respectivement $0.5i$, $0.5i$, et $0i$. Étant donné que c'est un transformateur, il n'y a pas de variable ZN dans le fichier de données. Finalement, c'est un transformateur régulateur, donc le simulateur a besoin des valeurs des variables associées à la régulation. En premier lieu, la valeur de $a1min$ est 0.9, celle de $a1$ (valeur de départ) est 1, $a1max$ est égal à 1.1, et la valeur de $a2$ est égale à 1. Il faut indiquer le nombre de prises du transformateur, et dans ce cas c'est 11. Ensuite, il faut indiquer quelle est la barre qui est contrôlée par le transformateur. Dans ce cas c'est la phase b de la barre 3. Enfin, il faut indiquer les valeurs minimales et maximales de la tension que cette barre peut atteindre. Pour ce cas, la tension minimale est placée à 0.5 et la tension maximale est placée à 0.82 (*Voir ANNEXE V, p.115, Tableau-A V-1 à 4*).

3.5.2 Résultats obtenus

Maintenant que le fichier de données est créé, il est possible de lancer la simulation et de faire la vérification à l'aide du prototype afin de s'assurer que les deux réagissent de la même façon lorsqu'il y a un transformateur monophasé à changeur de prises. Premièrement, voici les différentes itérations obtenues avec les deux programmes utilisant la méthode Newton-Raphson. En premier, voici les résultats des erreurs résiduelles pour chaque itération ainsi que les changements nécessaires du transformateur pour la régulation :

Tableau 3.11 Itérations du prototype (Cas #4)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	$4,9999 \times 10^{-01}$ (3b)	$-3,1225 \times 10^{-16}$ (2b)	$5,2915 \times 10^{-01}$
1	$2,0574 \times 10^{-02}$ (3b)	$1,5837 \times 10^{-01}$ (2b)	$2,0214 \times 10^{-01}$
2	$2,4711 \times 10^{-03}$ (2b)	$1,4396 \times 10^{-02}$ (3b)	$1,7360 \times 10^{-02}$
3	$7,2098 \times 10^{-03}$ (3b)	$-2,1102 \times 10^{-02}$ (2b)	$2,7620 \times 10^{-02}$
4	$6,9269 \times 10^{-03}$ (3b)	$-2,0005 \times 10^{-02}$ (2b)	$2,6196 \times 10^{-02}$
5	$6,5590 \times 10^{-03}$ (3b)	$-1,8952 \times 10^{-02}$ (2b)	$2,4811 \times 10^{-02}$
6	$1,11839 \times 10^{-04}$ (3b)	$2,2368 \times 10^{-04}$ (3b)	$2,54334 \times 10^{-04}$
7	$4,3922 \times 10^{-08}$ (3b)	$8,7844 \times 10^{-08}$ (3b)	$9,82960 \times 10^{-08}$
8	$6,8173 \times 10^{-15}$ (3b)	$1,3726 \times 10^{-14}$ (3b)	$1,5565 \times 10^{-14}$

Tableau 3.12 Itérations du simulateur (Cas #4)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	0,5 (3-B)	4,78784e-016 (2-B)	0,52915
1	0,0205745 (3-B)	0,158374 (2-B)	0,202138
2	0,00247108 (2-B)	0,0143955 (3-B)	0,0173603
3	0,00720979 (3-B)	-0,0211025 (2-B)	0,0276201
4	0,00692695 (3-B)	-0,0200048 (2-B)	0,0261961
5	0,00655902 (3-B)	-0,0189522 (2-B)	0,0248111
6	0,00011184 (3-B)	0,00022368 (3-B)	0,000254334
7	4,3915 x 10 ⁻⁰⁸ (3-B)	8,7831 x 10 ⁻⁰⁸ (3-B)	9,82823 x 10 ⁻⁰⁸
8	5,3395 x 10 ⁻¹⁵ (3-B)	1,0775 x 10 ⁻¹⁴ (3-B)	1,21449 x 10 ⁻¹⁴

Tableau 3.13 Changement de prises du transformateur (Cas #4)

Itérations	Explication
0	Aucun changement de prise à la première itération
1	(0,202138 > 0,001) Erreur résiduelle trop grande pour faire un ajustement
2	(0,0173603 > 0,001) Erreur résiduelle trop grande pour faire un ajustement
3	Prise ++ quand (Vcontrôlé > Vmaximum) : 0,856409 > 0,82. De la prise 6 à 7
4	Prise ++ quand (Vcontrôlé > Vmaximum) : 0,841129 > 0,82. De la prise 7 à 8
5	Prise ++ quand (Vcontrôlé > Vmaximum) : 0,826309 > 0,82. De la prise 8 à 9
6	Aucun changement de prise nécessaire
7	Aucun changement de prise nécessaire
8	Aucun changement de prise nécessaire

Tel que démontré dans la section au sujet de la régulation, la valeur de l'erreur résiduelle doit être plus petite que 0,001 avant de pouvoir permettre au transformateur de changer de prise. Une fois atteint, le simulateur entre dans la boucle de régulation pour chaque itération tant et aussi longtemps que la tension de la barre contrôlée ne respecte pas les conditions entrées dans le fichier de données.

Il faut maintenant s'assurer que les tensions (modules et angles) sont identiques entre le prototype et le simulateur. Voici le tableau montrant la valeur de chacune des barres :

Tableau 3.14 Valeur des tensions (Cas #4)

Barre#	Simulateur		Prototype	
	Module de tension	Angle	Module de tension	Angle
1a	1	0	1	0
1b	1	-120	1	-120
1c	1	120	1	120
2a	0,9998	-1,15	0,9998	-1,15
2b	0,9619	-125	0,9619	-125
2c	0,9998	118,9	0,9998	118,9
3b	0,8116	-151,6	0,8116	-151,6

3.5.3 Validation des résultats

Encore une fois, il faut être certain des résultats obtenus, et la seule façon d'y parvenir est de valider les résultats en vérifiant si toute l'énergie générée est bel et bien consommée dans le réseau (*Voir ANNEXE V, p.116, Tableau-A V-6 et Tableau-A V-7*). Premièrement, il faut regarder la génération d'énergie. La seule barre faisant cela est la barre d'équilibre. Celle-ci génère $(0,621834 + j 0,213035)$ pu. Par la suite, la barre de charge triphasée consomme $(0,09996 + j 0)$ pu pour les phases a et c, mais consomme $(0,0925254 + j 0)$ pu pour la phase b, car c'est la seule qui est reliée à la barre de charge monophasée. Celle-ci consomme donc un total de $(0,329389 + j 0)$ pu. Un total de $(0,621834 + j 0)$ pu est obtenu en additionnant le total des charges du réseau. C'est la totalité de la puissance active générée dans le réseau. Pour la puissance réactive, il faut calculer les pertes de chacune des branches, et les additionner afin de voir si la puissance générée est la même. Pour la branche triphasée commençant à la barre #1 et terminant à la barre #2, le total des pertes dans cette ligne est de $(0 + j 0,0039984)$ pu pour la somme des phases a et c, et de $(0 + j 0,044341)$ pu pour la phase b. Finalement, la perte de la ligne monophasée entre la barre #2 et la barre #3 est de $(0 + j 0,164695)$ pu. En additionnant le tout, il y a exactement le total de la puissance réactive fourni par la barre d'équilibre.

3.5.4 Discussion et interprétation des résultats

Dans ce cas-ci, il faut vérifier que le simulateur réajuste correctement la prise du transformateur afin d'atteindre le point de régulation. Comme ils se doivent, les résultats des itérations du prototype et du simulateur sont encore identiques. En regardant les résultats des modules et angles de tension de la barre #2, ceux-ci ne sont pas pareils pour l'ensemble des phases, car il y a un débancement qui se crée en ajoutant la ligne monophasée. Cette facette se voit aussi dans le cas précédent. Pour ce qui a trait aux changements de prises, le simulateur ne fait rien même si la tension est supérieure à la tension maximale pour les deux premières itérations, car les erreurs résiduelles sont encore trop grandes. À la suite de cela, il entre dans la boucle de régulation et fait les ajustements nécessaires pour que la tension de la barre contrôlée soit dans les limites de la régulation. Les mêmes résultats sont obtenus avec la méthode Levenberg-Marquardt pour la régulation (*Voir ANNEXE V, p.115*). Il est donc permis de croire que le simulateur peut intégrer ce type de transformateur sans aucun problème.

3.6 Cas #5 : Réseau triphasé composé d'une barre de charge débalancée

Le prochain cas vérifie la fonctionnalité des charges débalancées dans les réseaux électriques. Pour ce faire, il faut insérer une barre triphasée et la séparer en trois barres monophasées, en insérant différentes valeurs pour les charges correspondant aux phases.

Voici le schéma équivalent correspondant à ce cas :

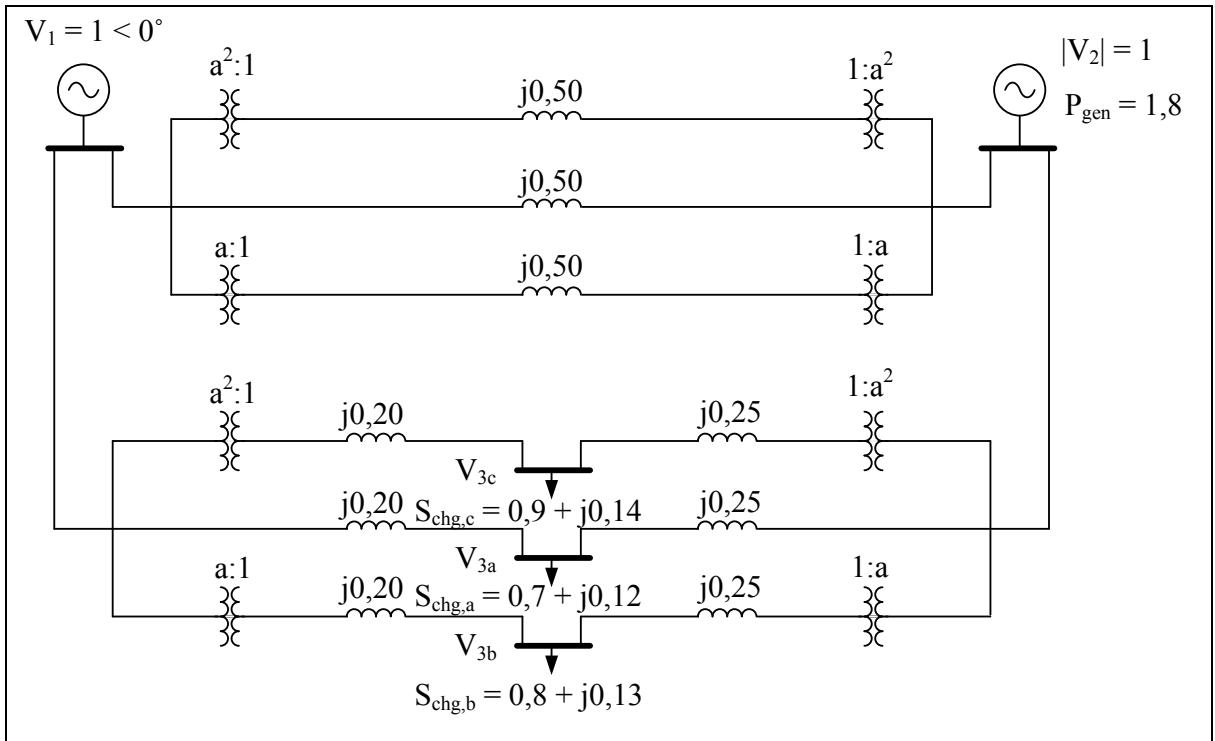


Figure 3.6 Réseau ayant une barre de charge débalancée (Cas #5)

3.6.1 Éléments à insérer dans le fichier de données

Pour la construction du rapport, il faut créer un onglet pour les trois barres monophasées, et insérer les valeurs des puissances constantes respectives pour chacune d'entre elles (*Voir ANNEXE VI, p.117, Tableau-A VI-1 à 4*).

3.6.2 Résultats obtenus

Voici les résultats obtenus pour le simulateur et le prototype :

Tableau 3.15 Itérations du prototype (Cas #5)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	-1,8000 (2)	$1,4000 \times 10^{-01}$ (3c)	2,2871
1	$-2,4827 \times 10^{-02}$ (2)	$5,0088 \times 10^{-02}$ (3c)	$7,9533 \times 10^{-02}$
2	$-1,4340 \times 10^{-04}$ (2)	$3,3280 \times 10^{-04}$ (3c)	$4,7201 \times 10^{-04}$
3	$7,1454 \times 10^{-09}$ (3c)	$1,5631 \times 10^{-08}$ (3c)	$1,9544 \times 10^{-08}$
4	$1,2212 \times 10^{-15}$ (3c)	$1,3878 \times 10^{-15}$ (3b)	$2,1582 \times 10^{-15}$

Tableau 3.16 Itérations du simulateur (Cas #5)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	-1,8 (2-A)	0,14 (3-C)	2,28712
1	-0,0248272 (2-A)	0,0500878 (3-C)	0,0795327
2	-0,000143396 (2-A)	0,000332798 (3-C)	0,000472007
3	$7,1473 \times 10^{-09}$ (3-C)	$1,564 \times 10^{-08}$ (3-C)	$1,95498 \times 10^{-08}$
4	$-1,2213 \times 10^{-15}$ (3-C)	$-1,5821 \times 10^{-15}$ (3-A)	$2,72189 \times 10^{-15}$

Encore une fois pour le cas présent, les résultats des erreurs résiduelles lors des itérations du simulateur, ainsi que celles du prototype sont pratiquement identiques. Voici maintenant les résultats des tensions, quelle que soit la méthode utilisée comparativement aux résultats obtenus avec le prototype :

Tableau 3.17 Résultats des tensions (Cas #5)

Barre#	Simulateur		Prototype	
	Tension	Angle	Tension	Angle
1a	1	0	1	0
1b	1	-120	1	-120
1c	1	120	1	120
2a	1	3,36	0,99999	3,36
2b	1	-116,64	0,99999	-116,64
2c	1	123,36	0,99999	123,36
3a	0,9829	-3,05	0,9829	-3,05
3b	0,9807	-123,7	0,9807	-123,7
3c	0,978438	115,6	0,9784	115,62

3.6.3 Validation des résultats

Il faut maintenant valider les résultats en calculant la différence entre la puissance qui entre dans le réseau, et celle qui est consommée par les composantes (*Voir ANNEXE VI, p.118, Tableau-A VI-6 et Tableau-A VI-7*). Premièrement, la barre de génération produite ($0,6 + j 0,33159$) pu de puissance totale. La barre de génération (barre #2) produit quant à elle un total de ($1,8 + j 0,332148$) pu. Le total généré dans le réseau est donc, un grand total de ($2,4 + j 0,663738$) pu. Pour la puissance consommée, il faut faire comme dans les cas précédents, c'est-à-dire additionner la puissance des charges avec celle des lignes du réseau. Premièrement, pour l'ensemble des charges, il y a un total de ($2,4 + j 0,39$) pu de puissance qui est consommée. Tandis qu'il faut calculer les pertes pour chacune des lignes. Premièrement, pour la ligne partant de la barre d'équilibre vers la barre de génération, il y a un total de ($0 + j 0,02063058$) pu de puissance consommée. Ensuite, pour la ligne partant de la barre d'équilibre vers la barre de charge, il y a un total de ($0 + j 0,0686007$) pu. Finalement, pour la ligne reliant la barre de génération avec la barre de charge, il y a un total de ($0 + j 0,184507$) pu. Le total consommé par les lignes est donc de ($0 + j 0,27373828$) pu. En faisant l'addition de cette valeur avec le total consommé par la charge, il y a bel et bien le total égal à la puissance générée dans le réseau. Le simulateur est donc capable de calculer l'écoulement de puissance d'un réseau débalancé sans aucun problème.

3.6.4 Discussion et interprétation des résultats

Tout d'abord, les résultats des itérations pour les deux simulateurs sont encore exactement les mêmes. Pour ce qui en est des résultats des modules des barres d'équilibre et de génération, ils doivent être effectivement égaux à 1. En regardant de plus près les résultats, les valeurs des modules des phases de la barre de charge sont toutes différentes, dû au fait que leurs puissances sont différentes. Cela cause un débalancement dans le réseau. Il est encore une fois ici possible d'affirmer que grâce à ce cas-ci, le simulateur est capable de faire des écoulements de puissance dans des réseaux qui sont débalancés. Cela dit, la grande majorité des réseaux réels le sont, et il devient important qu'un programme puisse être capable de faire ces calculs. Les deux méthodes itératives donnent exactement les mêmes résultats, mais

si par exemple, le réseau était mal conditionné (avec une norme très élevé) et débalancé, la méthode Newton-Raphson ne serait plus suffisante pour résoudre l'écoulement de puissance.

3.7 Cas #6 : Plusieurs barres triphasées avec un transformateur Yy0 qui contrôle une des barres de charge dans le réseau (diminution de prise)

Voici le schéma représentant ce réseau :

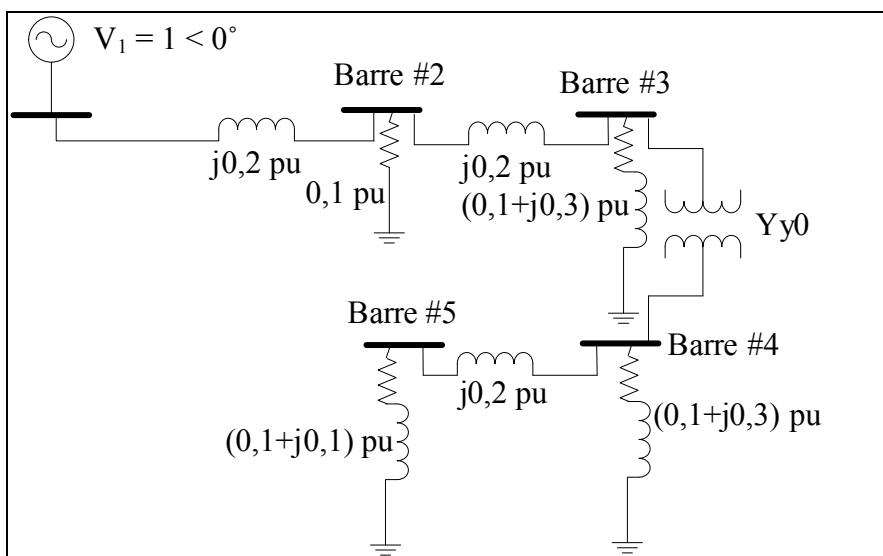


Figure 3.7 Réseau triphasé avec un transformateur Yy0 (Cas #6)

3.7.1 Éléments à insérer dans le fichier de données

Dans ce cas, la barre contrôlée est la barre #2. Il faut donc en premier lieu insérer cette barre dans la case correspondante du fichier de lecture. Par la suite, il faut indiquer les valeurs minimale et maximale de a_1 . Pour ce cas, elles sont respectivement 0.9 et 1.1. Ensuite, il faut indiquer la valeur minimale de tension pour la barre contrôlée ainsi que sa valeur maximale qu'elle peut atteindre, etc. (Voir ANNEXE VII, p.119, Tableau-A VII-1 à 3).

3.7.2 Résultats obtenus

Voici maintenant les valeurs des erreurs résiduelles maximales pendant l'écoulement de puissance :

Tableau 3.18 Itérations du prototype (Cas #6)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	1,0000 x 10 ⁻⁰¹ (2b)	1,0000 x 10 ⁻⁰¹ (4a)	4,5826 x 10 ⁻⁰¹
1	1,0232 x 10 ⁻⁰² (4c)	3,2451 x 10 ⁻⁰² (3b)	8,4085 x 10 ⁻⁰²
2	5,5698 x 10 ⁻⁰⁴ (4b)	1,4111 x 10 ⁻⁰³ (4a)	3,2042 x 10 ⁻⁰³
3	-3,3005 x 10 ⁻⁰³ (3b)	-2,7979 x 10 ⁻⁰² (3a)	5,3997 x 10 ⁻⁰²
4	3,3225 x 10 ⁻⁰³ (3c)	2,8828 x 10 ⁻⁰² (3c)	5,5149 x 10 ⁻⁰²
5	3,4761 x 10 ⁻⁰³ (3c)	3,0186 x 10 ⁻⁰² (3c)	5,7746 x 10 ⁻⁰²
6	3,6442 x 10 ⁻⁰³ (3c)	3,1647 x 10 ⁻⁰² (3c)	6,0534 x 10 ⁻⁰²
7	5,0698 x 10 ⁻⁰⁵ (3b)	2,4092 x 10 ⁻⁰⁴ (3b)	5,4579 x 10 ⁻⁰⁴
8	2,1171 x 10 ⁻⁰⁸ (4c)	8,2998 x 10 ⁻⁰⁸ (4c)	1,6754 x 10 ⁻⁰⁷
9	4,0711 x 10 ⁻¹⁵ (4c)	1,4784 x 10 ⁻¹⁴ (4c)	2,5317 x 10 ⁻¹⁴

Tableau 3.19 Itérations du simulateur (Cas #6)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	0,1 (2-B)	0,1 (4-C)	0,458258
1	0,0102322 (4-A)	0,0324507 (3-A)	0,0840845
2	0,000556976 (4-C)	0,00141112 (4-C)	0,00320423
3	-0,00330046 (3-A)	-0,0279792 (3-A)	0,053997
4	0,00332254 (3-A)	0,0288278 (3-C)	0,0551485
5	0,00347613 (3-A)	0,0301856 (3-A)	0,0577462
6	0,00364417 (3-C)	0,0316465 (3-B)	0,060534
7	5,0698 x 10 ⁻⁰⁵ (3-C)	0,000240915 (3-A)	0,000545786
8	2,1173 x 10 ⁻⁰⁸ (4-A)	8,3011 x 10 ⁻⁰⁸ (4-B)	1,67562 x 10 ⁻⁰⁷
9	3,7313 x 10 ⁻¹⁵ (4-B)	3,8243 x 10 ⁻¹⁴ (5-B)	9,3327 x 10 ⁻¹⁴

Les valeurs observées dans les tableaux précédents sont les mêmes. Maintenant, il faut voir comment la régulation a eu lieu afin de s'assurer que la tension de la barre contrôlée respecte les limites imposées dans le fichier de lecture.

Tableau 3.20 Changement de prises du transformateur (Cas #5)

Itérations	Explication
0	Aucun changement de prise à la première itération
1	$(0,0840845 > 0,001)$ Erreur résiduelle trop grande pour faire un ajustement
2	$(0,00320423 > 0,001)$ Erreur résiduelle trop grande pour faire un ajustement
3	Prise + delta quand $(V_{\text{contrôlée}} > V_{\text{max}})$: $0,966428 > 0,96$. De la prise 6 à 7 Direction = 1
4	Prise -- quand $(V_{\text{contrôlée}} > V_{\text{max}})$: $0,969768 > 0,96$. De la prise 7 à 6
5	Prise + delta quand $(V_{\text{contrôlée}} > V_{\text{max}})$: $0,966498 > 0,96$. De la prise 6 à 5 Direction = -1
6	Prise + delta quand $(V_{\text{contrôlée}} > V_{\text{max}})$: $0,963077 > 0,96$. De la prise 5 à 4 Direction = -1
7	Aucun changement de prise nécessaire
8	Aucun changement de prise nécessaire
9	Aucun changement de prise nécessaire

Il faut maintenant étudier les résultats des tensions obtenus afin de voir si la barre est bel et bien contrôlée. Voici le tableau de résultats du simulateur ainsi que ceux du prototype :

Tableau 3.21 Résultats des tensions (Cas #6)

Barre#	Simulateur		Prototype	
	Module de tension	Angle	Module de tension	Angle
1a	1	0	1	0
1b	1	-120	1	-120
1c	1	120	1	120
2a	0,959407	-4,24	0,959407	-4,24
2b	0,959407	-124,24	0,959407	-124,24
2c	0,959407	115,76	0,959407	115,76
3a	0,923169	-7,63	0,923169	-7,63
3b	0,923169	-127,63	0,923169	-127,63
3c	0,923169	112,37	0,923169	112,37
4a	0,931703	-21,94	0,931703	-21,94
4b	0,931703	-141,94	0,931703	-141,94
4c	0,931703	98,06	0,931703	98,06
5a	0,950519	-23,11	0,950519	-23,11
5b	0,950519	-143,11	0,950519	-143,11
5c	0,950519	96,89	0,950519	96,89

La tension de la barre contrôlée est bel et bien sous de la valeur maximale indiquée dans le fichier de données. Le simulateur a donc bien fait le travail de régulation.

3.7.3 Validation des résultats

Encore une fois, il faut faire la vérification de la puissance fournie par rapport à la puissance qui est consommée dans le réseau (*Voir ANNEXE VII, p.120, Tableau-A VII-5 et Tableau-A VII-6*). Premièrement, il n'y a que la barre d'équilibre qui génère de la puissance. Cela dit, celle-ci génère un total de $(1,06328 + j 0,648225)$ pu en puissance totale. Pour ce qui en est de la puissance consommée, il faut faire comme les cinq cas précédents, c'est-à-dire additionner la puissance des barres avec la puissance des lignes, et vérifier s'il y a bien la même puissance générée.

Pour simplifier les explications, la puissance indiquée est la puissance de l'ensemble des phases et des lignes. En premier lieu, la puissance consommée à la barre #2 est de $(0,2761386 + j 0)$ pu totale. Ensuite pour la barre #3, la puissance consommée est de $(0,2556726 - j 0,767019)$ pu. Pour ce qui est de la barre #4, la puissance totale consommée est de $(0,260421 - j 0,781263)$ pu. Finalement, pour la barre #5, la puissance totale consommée est de $(0,2710461 - j 0,2710461)$ pu. Le total consommé par les barres est donc de $(1,06328 - j 1,8193281)$ pu. Pour la ligne qui débute à la barre #1 et qui se rend à la barre #2, la puissance totale est de $(0 + j 0,103383)$ pu. Pour la ligne qui va de la barre #2 à la barre #3, la puissance est de $(0 + j 0,066375)$ pu. La ligne partant de la barre #4 et qui se rend à la barre #5, la puissance totale est de $(0 + j 0,010842)$ pu. Et finalement, pour le transformateur, la puissance consommée est de $(0 + j 2,256951)$ pu. En additionnant les puissances des lignes avec celle des barres, la puissance totale générée est bel et bien la puissance générée.

3.7.4 Discussion et interprétation des résultats

Ce cas permet de valider que le simulateur est capable d'intégrer les transformateurs triphasés régulateurs de tension. En regardant les résultats des modules et angles des barres, toutes les barres sont équilibrées. Pour ce qui en est du changement de prise, celui-ci fait comme le transformateur monophasé et attend que les erreurs résiduelles se situent sous la consigne indiquée dans la boucle de régulation. Une fois fait, il regarde l'emplacement de la barre contrôlée (quelque part dans le réseau). Tel que vu dans la section au sujet de la régulation, le simulateur commence par faire un essai par rapport à la direction qu'il doit prendre, car il ne sait pas où la barre contrôlée se situe par rapport à lui. Étant donné qu'il s'éloigne de la solution avec un premier essai, il change de direction afin de s'approcher de la solution, et ce, jusqu'à ce que la valeur de la tension soit dans les limites imposées pour la régulation. Encore une fois, ceci démontre que le simulateur est capable d'intégrer les transformateurs triphasés dans le programme, et pourrait reproduire un cas de grande envergure avec plusieurs transformateurs monophasés et triphasés sans aucun problème.

3.8 Cas #7 : Réseau triphasé complet mal conditionné (3 000 barres triphasées)

Ce dernier cas permet de comparer les deux méthodes intégrées dans le simulateur lorsqu'il faut faire l'écoulement de puissance d'un réseau triphasé mal conditionné. Ce réseau a été conçu à partir d'une multitude de réseaux connectés les uns aux autres. L'ensemble des valeurs a été changé afin de concorder sur la même échelle de mesure (1 pu). En faisant l'ensemble de ces opérations, l'indice de conditionnement des matrices est devenu très grand. La multiplication des normes pour ce cas est donc de $4,8873 \times 10^6$. Elle est beaucoup plus grande que celle calculée pour le troisième cas. Il s'agit donc d'un réseau encore moins bien conditionné que ceux des cas précédents. Voici premièrement un aperçu du schéma complet représentant le réseau composé de 3 000 barres triphasées et de 4 783 lignes triphasées :

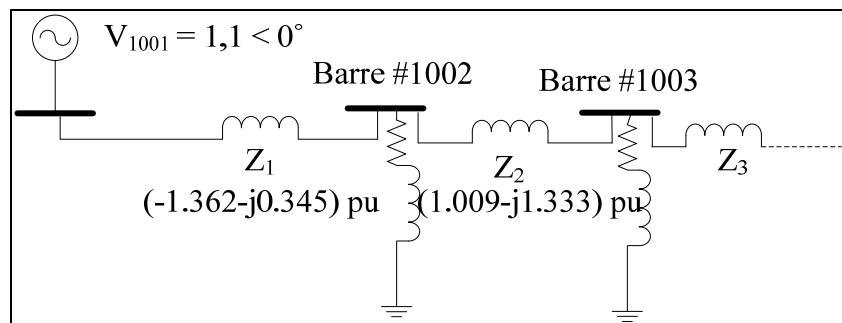


Figure 3.8 Schéma tronqué d'un réseau triphasé composé de 3 000 barres (Cas #7)

3.8.1 Éléments à insérer dans le fichier de données avec V_I proche de V_F

Dans le fichier de données, il faut simplement indiquer les valeurs des puissances obtenues à l'aide des barres de génération, celles des puissances consommées par les barres de charges, ainsi que les impédances de lignes pour l'ensemble du réseau (*Voir ANNEXE VIII, p.123, Tableau-A VIII-1 et Tableau-A VIII-2 pour quelques lignes*). Pour ce qui en est des valeurs de départ des modules et angles des tensions, il faut s'assurer qu'elles soient très proches de la solution afin que le prototype et le simulateur puissent converger en utilisant la méthode Newton-Raphson.

3.8.2 Résultats obtenus

Voici un tableau illustrant les résultats des itérations par le prototype, ainsi qu'un tableau illustrant les itérations du simulateur lorsque les valeurs initiales sont extrêmement proches des valeurs finales.

Tableau 3.22 Itérations du prototype quand V_I est proche de V_F (Cas #7)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	$-5,92 \times 10^{-3}$ (3610a)	$-2,62 \times 10^{-2}$ (3610a)	$9,6608 \times 10^{-2}$
1	$6,82 \times 10^{-7}$ (3610a)	$2,63 \times 10^{-6}$ (3610a)	$5,9594 \times 10^{-6}$
2	$5,12 \times 10^{-12}$ (3007a)	$2,14 \times 10^{-12}$ (1225a)	$2,0032 \times 10^{-11}$

Tableau 3.23 Itérations du simulateur quand V_I est proche de V_F (Cas #7)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	-0,005918 (3610-A)	-0,026185 (3610-A)	0,0966081
1	$6,83 \times 10^{-7}$ (3610-A)	$2,63 \times 10^{-6}$ (3610-A)	$5,96146 \times 10^{-6}$
2	$5,06 \times 10^{-12}$ (3007-A)	$7,01 \times 10^{-12}$ (3644-A)	$4,54316 \times 10^{-11}$

Il faut maintenant examiner les résultats des modules et angles de quelques-unes des barres afin de s'assurer que le simulateur trouve des résultats semblables au prototype. Voici le tableau représentant quelques résultats :

Tableau 3.24 Résultats des tensions quand V_1 est proche de V_F (Cas #7)

Barre#	Simulateur		Prototype	
	Tension	Angle	Tension	Angle
1001a	1,1	0	1,1	0
1001b	1,1	-120	1,1	-120
1001c	1,1	120	1,1	120
1002a	1,01862	-19,141	1,01862	-19,141
1002b	1,01862	-139,141	1,01862	-139,141
1002c	1,01862	100,859	1,01862	100,859
1003a	0,93904	-61,0661	0,93904	-61,0661
1003b	0,93904	-181,0661	0,93904	-181,0661
1003c	0,93904	58,9339	0,93904	58,9339
1004a	0,93951	-60,1147	0,93951	-60,1147
1004b	0,93951	-180,1147	0,93951	-180,1147
1004c	0,93951	59,8853	0,93951	59,8853
...
4000a	1,02146	-22,3026	1,02146	-22,3026
4000b	1,02146	-142,3026	1,02146	-142,3026
4000c	1,02146	97,6974	1,02146	97,6974

Voici le diagramme radar permettant de voir comment les tensions finales sont distribuées dans l'ensemble du réseau triphasé :

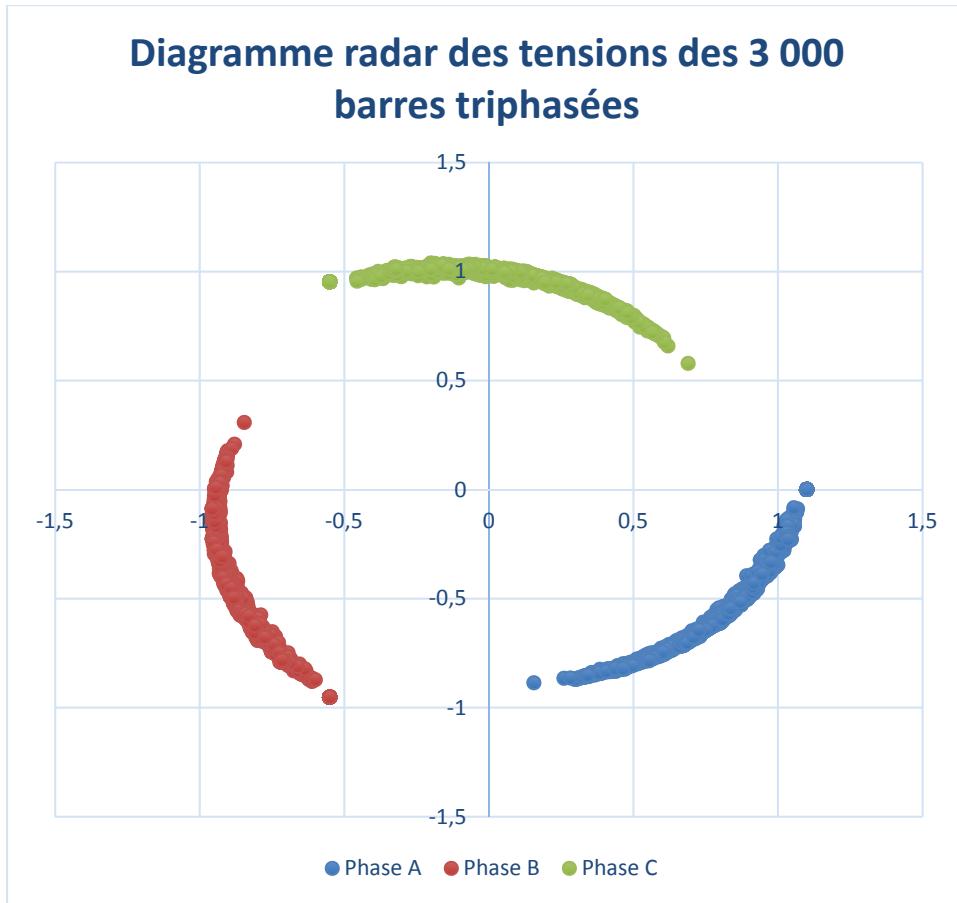


Figure 3.9 Diagramme radar pour l'ensemble des résultats pour le cas #7

3.8.3 Éléments à insérer dans le fichier de données avec V_I loin de V_F

Les valeurs initiales doivent maintenant être changées afin de s'éloigner des valeurs finales afin que la méthode Newton-Raphson soit incapable de converger. Cela permet de montrer l'efficacité de la méthode Levenberg-Marquardt lorsque les réseaux sont mal conditionnés et que les valeurs initiales sont un peu plus éloignées des valeurs finales. Il faut donc changer dans le fichier de données les angles des valeurs initiales des barres à 0,5 fois la valeur du fichier précédent. Cela vient éloigner les valeurs initiales et fait diverger la méthode Newton-Raphson. Voici les résultats itératifs du prototype, ainsi que ceux du simulateur quand celui-ci utilise la méthode Levenberg-Marquardt :

Tableau 3.25 Itérations du prototype quand V_I est loin de V_F (Cas #7)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	-25,574 (3033)	-3,0095 (3049c)	1,2050 x 10 ²
1	-38,195 (1002c)	74,861 (1002c)	1,5463 x 10 ²
2	-16,352 (1002c)	8,4643 (1002b)	35,083
3	1,1999 x 10 ² (1002c)	53,154 (1002c)	2,5539 x 10 ²
4	24,999 (1002c)	10,622 (1002c)	54,334
5	-7,7795 (1002b)	2,4715 (4000a)	17,771
6	-3,1834 (3322c)	1,6075 (3322b)	12,770
7	9,27 x 10 ⁻⁰¹ (4000c)	1,0680 (3991c)	6,47137
8	25,126 (2883b)	17,275 (2830b)	73,919
9	3,6507 (4000b)	24,463 (3322b)	72,409
10	9,2766 (2834b)	24,523 (2830b)	59,806
11	-41,21 (4000b)	87,708 (3999b)	2,7890 x 10 ²
12	28,005 (3992c)	52,635 (3992c)	1,4862 x 10 ²
13	5,5724 x 10 ² (2318c)	2,0004 x 10 ³ (2563c)	4,8889 x 10 ³

Tableau 3.26 Itérations du simulateur quand V_I est loin de V_F (Cas #7)

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	-25,5735 (3033-A)	-3,0095 (3049-A)	120,508
1	-0,538708 (2955-A)	0,983054 (3049-C)	18,2105
2	0,178657 (1589-B)	0,218306 (1589-A)	2,06958
3	-0,009661 (1506-A)	0,0109297 (3954-B)	0,268714
4	-0,025713 (1002-B)	0,0567725 (1002-B)	0,213558
5	0,0045222 (3322-C)	0,019372 (1002-A)	0,0832638
6	0,013717 (3322-A)	0,0692853 (1002-C)	0,132011
7	0,0010159 (3322-B)	0,0032855 (1002-B)	0,0150821
8	0,0011381 (3322-A)	0,0046109 (1002-A)	0,00989575
9	-1,33 x 10 ⁻⁰⁵ (2199C)	1,382 x 10 ⁻⁰⁵ (1002-B)	0,000112433
10	-1,46 x 10 ⁻⁰⁸ (2199-B)	1,23 x 10 ⁻⁰⁸ (1002-A)	1,03269 x 10 ⁻⁰⁷
11	1,73 x 10 ⁻¹³ (2125-B)	4,55 x 10 ⁻¹³ (1997-A)	3,39354 x 10 ⁻¹²

3.8.4 Résultats obtenus

En regardant seulement les itérations du prototype, il est possible de constater que le processus diverge. Il est donc impossible d'avoir des résultats concluants en utilisant le prototype lorsque les valeurs initiales sont éloignées et que la multiplication des normes de

Frobenius est plus élevée. Cela dit, voici un aperçu des modules et angles des tensions obtenus avec le simulateur utilisant la méthode Levenberg-Marquardt :

Tableau 3.27 Résultats des tensions quand V_I est loin de V_F (Cas #7)

Barre#	Simulateur	
	Tension	Angle
1001a	1,1	0
1001b	1,1	-120
1001c	1,1	120
1002a	1,01862	-19,141
1002b	1,01862	-139,141
1002c	1,01862	100,859
1003a	0,93904	-61,0661
1003b	0,93904	-181,0661
1003c	0,93904	58,9339
1004a	0,93951	-60,1147
1004b	0,93951	-180,1147
1004c	0,93951	59,8853
...
4000a	1,02146	-22,3026
4000b	1,02146	-142,3026
4000c	1,02146	97,6974

3.8.5 Validation des résultats

La validation des résultats n'a pas lieu, car le processus est identique à celui des cas précédents, et le réseau triphasé complet est formé avec le même type de composantes. Il va donc dire que la somme des puissances générées est égale à la somme des puissances consommées dans le réseau.

3.8.6 Discussion et interprétation des résultats

Ce cas représente un réseau réel de 3 000 barres et son indice de conditionnement est de $1,2109 \times 10^4$ ce qui est très élevé en comparaison aux autres cas. Il est donc moins bien

conditionné que ceux-ci. Cela dit, si les valeurs initiales des barres de charges et des barres de générations sont proches des valeurs finales dans le fichier de données, le prototype, ainsi que le simulateur sont capables de converger en utilisant la méthode Newton-Raphson. Par contre, aussitôt que ces valeurs changent un peu, il est impossible de converger utilisant cette méthode. Ce cas démontre ainsi l'avantage de l'intégration de la méthode Levenberg-Marquardt dans le simulateur, car si les valeurs finales sont inconnues et que le réseau est mal conditionné, cette méthode est capable de les trouver. En regardant les tableaux des résultats des tensions obtenues, il est possible de constater que même si les valeurs initiales sont loin des valeurs finales, la méthode Levenberg-Marquardt permet de converger vers les mêmes résultats que pour l'autre situation. De plus, il faut que le temps d'exécution soit en bas de cinq secondes pour qu'il soit considéré comme raisonnable. Donc, la vitesse d'exécution du prototype pour un instant donné lorsque les valeurs sont très proches est d'environ 40 secondes, tandis que pour le simulateur, le temps d'exécution du programme est de seulement 1,5 à 2,5 secondes (tout dépendant de la méthode utilisée). L'étude d'une multitude de cas simultanément est donc possible à cause de cette rapidité d'exécution. En conclusion, ce cas permet de faire l'étude de réseaux triphasés complets qui ont un indice de conditionnement jusqu'à $4,8873 \times 10^6$. Cela veut dire que la matrice d'admittance est mal conditionnée. De plus, il est possible de conclure avec ce cas que la méthode Levenberg-Marquardt est capable de converger là où la méthode Newton-Raphson n'est pas capable. Cela permet donc de faire l'étude de réseaux triphasés complets qui sont impossibles à résoudre en utilisant la méthode NR.

3.9 Discussion

Tel que mentionné dans la revue de littérature, plusieurs méthodes peuvent être utilisées pour faire un écoulement de puissance. Que ce soit la méthode Gauss-Seidel, la méthode Newton-Raphson, la méthode Levenberg-Marquardt ou les OPF, chaque méthode a ses avantages et ses inconvénients. Deux de ces méthodes ont été implémentées dans le simulateur présenté dans ce mémoire afin de calculer les écoulements de puissance : la méthode Newton-Raphson et la méthode Levenberg-Marquardt. En introduisant la deuxième méthode, et grâce

à son avantage d'être plus précise, il est possible d'aller plus loin lorsqu'il faut résoudre des réseaux mal conditionnés. En faisant l'étude des cas, le simulateur a été testé de façon à voir comment celui-ci réagit lorsqu'il y a un réseau mal conditionné (cas #3 et cas #7). Les temps d'exécution du cas #3 sont pratiquement identiques, que ce soit lorsque l'indice de conditionnement est de 54,0188 ou $2,2489 \times 10^5$. Il est donc possible d'affirmer que l'indice de conditionnement n'a pas un grand impact sur les temps de calculs. Ensuite, plus l'indice de conditionnement est élevé ($4,8873 \times 10^6$), moins la méthode Newton-Raphson est capable de converger si les valeurs initiales sont loin des valeurs finales. C'est à ce moment qu'en implémentant la méthode Levenberg-Marquardt, le simulateur devient beaucoup plus robuste, et est capable de résoudre des réseaux électriques qui ne peuvent être résolus avec la méthode Newton-Raphson. Pour ce qui en est des autres cas, ils ont été réalisés de façon à démontrer que le simulateur est capable de faire l'écoulement de puissance pour un ensemble de composantes communes. C'est pourquoi il était nécessaire de faire une étude sur plusieurs cas, afin de s'assurer de la fiabilité du programme. Enfin, le temps d'exécution du simulateur est quand même très rapide, c'est-à-dire qu'il prend environ 2,5 secondes pour résoudre un écoulement de puissance d'un réseau triphasé complet de 3 000 barres. De plus, le simulateur atteint cette vitesse d'exécution sans même avoir été optimisé. Enfin, le temps de calculs pour les réseaux étudiés est directement relié à la grosseur des matrices. Donc, plus le réseau est gros (3 000 barres), plus la résolution du système d'équation est lente. Il est donc fort à parier qu'en optimisant le code, la vitesse pourrait être encore plus élevée, ce qui permettrait de faire l'étude de profil de tension dans des temps compétitifs comparativement aux autres simulateurs disponibles sur le marché.

CONCLUSION

En conclusion, les réseaux électriques sont extrêmement complexes et faire le calcul de la propagation de la puissance à l'intérieur de ceux-ci devient très difficile. Heureusement, plusieurs méthodes mathématiques ont été développées permettant de calculer le tout en quelques fractions de seconde. Cependant, la problématique est qu'une grande partie de ces méthodes sont incapables d'intégrer un réseau de distribution avec un réseau de transport parce que la différence entre les impédances pour ces deux types de réseaux est extrêmement grande. Ceci rend la matrice Jacobienne mal conditionnée, et empêche ainsi l'un ou l'autre des réseaux à converger vers les bonnes valeurs finales. De plus, les changements brusques (perte de lignes, coup de foudre, débalancement des charges, etc.) rendent les réseaux imprévisibles et il est nécessaire d'avoir des outils permettant d'évaluer la manière dont ceux-ci réagissent en simulation afin de bien gérer ces changements en temps réel. La méthodologie (chapitre 2) consiste donc à comprendre comment le simulateur créé dans le cadre de ce mémoire calcule l'écoulement de puissance pour un réseau triphasé complet afin de faciliter l'étude des cas présentés au troisième chapitre. En effectuant cette étude avec le simulateur, il est possible de résoudre une multitude de réseaux électriques qui utilisent plusieurs types de transformateurs et qui font de la régulation de tension. De plus, le cas #3 présente un très petit exemple d'un réseau de transport triphasé qui est connecté à un réseau de distribution monophasé. Même si les impédances sont 80 fois plus petites, ce qui rend le réseau moins bien conditionné, le simulateur converge encore après seulement quatre itérations, et démontre qu'il est possible d'implémenter ce genre de réseau. Même si ce cas n'est pas représentatif d'un réseau réel, car la taille du réseau est trop petite, il permet de démontrer que le simulateur est capable d'intégrer les déviations monophasées avec un réseau triphasé. De plus, ce cas démontre finalement que l'indice de conditionnement n'a pas un gros impact sur le temps d'exécution du simulateur. Dernièrement, le dernier cas présente un réseau triphasé complet ayant 3 000 barres et un indice de conditionnement élevé comparativement aux autres cas. Cela veut dire que la matrice d'admittance est encore moins bien conditionnée. Le simulateur réussit quand même à résoudre l'écoulement de puissance en utilisant la méthode Newton-Raphson, mais seulement si les valeurs initiales sont

pratiquement les mêmes que les valeurs finales. Par contre, aussitôt que les valeurs initiales sont changées, il est impossible d'utiliser cette méthode, car les matrices sont mal conditionnées et le processus diverge. En utilisant la méthode Levenberg-Marquardt, le processus réussit quand même à converger ce qui démontre l'avantage de l'implémentation de cette méthode dans le simulateur. De plus, en regardant le temps d'exécution du processus itératif, il est important de noter que pour un réseau triphasé complet de 3 000 barres, il ne prend que quelques secondes pour trouver les valeurs finales.

L'implémentation de la méthode Levenberg-Marquardt dans le simulateur permet de résoudre des cas où la méthode Newton-Raphson est incapable de le faire. Cela donne un outil supérieur en termes de convergence à l'équipe de recherche de l'IREQ. Cependant, pour une étude plus approfondie, il faudrait prendre ce simulateur et faire beaucoup plus de simulations avec des réseaux connus afin de déterminer s'il peut converger en tout temps.

RECOMMANDATIONS

Une première recommandation serait d'optimiser le code C++, car il a été impossible de le faire en raison d'un manque de temps. Cela permettrait d'accélérer de beaucoup la vitesse d'exécution du programme et ainsi lui permettrait d'être compétitif par rapport aux simulateurs connus.

Il faudrait aussi intégrer une autre boucle afin de faire l'écoulement de puissance d'un réseau pour une période donnée (profil de tensions) et non seulement pour un instant donné. Ce serait beaucoup plus utile pour les chercheurs de l'IREQ afin qu'ils puissent valider certains résultats pour des périodes déterminées.

Une autre recommandation serait de créer un algorithme permettant de changer la pondération (qui est intégré dans le code) de la méthode Levenberg-Marquardt, afin que le simulateur puisse converger presque en tout temps (à moins que l'indice de conditionnement soit égal à l'infini), et ce, quel que soit le réseau étudié. Cet outil pourrait donc devenir un outil indispensable aux chercheurs.

Une dernière recommandation serait de faire des écoulements de puissances de réseaux de transport connus branchés sur des réseaux de distribution avec des dimensions réelles. Cela permettrait de voir si le simulateur est capable d'avoir les mêmes résultats que le prototype ou que d'autres simulateurs disponibles sur le marché.

ANNEXE I

GAUSS-SEIDEL POUR 1 CAS MONOPHASÉ

Tableau-A I-1 Données des barres monophasées pour le simulateur

Numéro de barre	Nom	Type	Ebase	Module	Angle	Pgen	Qgen	Pchg	Qchg
1000	SW1	1	1	1	0	0	0	0	0
2000	PV2	2	1	1	0	0,6	0	0	0
3000	PQ3	3	1	1	0	0	0	0,8	0,13

Tableau-A I-2 Données des branches monophasées pour le simulateur

Barre départ	Barre arrivée	CKT	Nom	Rés.	Réac.	Cond.	Susc.
1000	2000	1	'L 1 2'	0	0,5	0	0
1000	3000	1	'L 1 3'	0	0,2	0	0
2000	3000	1	'L 2 3'	0	0,25	0	0

Tableau-A I-3 Résultats des itérations avec la méthode Levenberg-Marquardt

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	0,8 (3-A)	0,13 (3-A)	1,00841
1	0,0129308 (3-A)	0,0405527 (3-A)	0,0433529
2	8,3311 x 10 ⁻⁵ (3-A)	0,000216072 (3-A)	0,000235989
3	2,8318 x 10 ⁻⁹ (3-A)	6,45035 x 10 ⁻⁹ (3-A)	7,18947 x 10 ⁻⁹
4	-2,2205 x 10 ⁻¹⁶ (3-A)	2,22045 x 10 ⁻¹⁶ (3-A)	3,14018 x 10 ⁻¹⁶

Tableau-A I-4 Résultats des tensions avec la méthode Levenberg-Marquardt

Barre#	Simulateur		Matlab (Gauss-Seidel)	
	Module de tension	Angle	Module de tension	Angle
1	1	0	1	0
2	1	3,36	1	3,36
3	1	-3,71	1	-3,71

Tableau-A I-5 Résultats des puissances des barres avec la méthode Levenberg-Marquardt

Barre#	Type	Valeur (pu)
1	Sgen	$0,2 + j 0,110062$
2	Sgen	$0,6 + j 0,11034$
3	Sshunt	$0,8 + j 0,13$

Tableau-A I-6 Résultats des puissances des lignes avec la méthode Levenberg-Marquardt pour le premier cas

Type	Barre#	Valeur (pu)
Ligne triphasée Barre 1 à 2	1	$-0,11722 + j 0,00343808$
	2	$0,11722 + j 0,00343808$
Ligne triphasée Barre 1 à 3	1	$0,31722 + j 0,106623$
	3	$-0,31722 - j 0,0842241$
Ligne triphasée Barre 2 à 3	2	$0,48278 + j 0,106902$
	3	$-0,48278 - j 0,0457759$

ANNEXE II

ÉTUDE DE CAS 1

Tableau-A II-1 Données des barres triphasées pour le simulateur

Numéro de barre	Nom	Type	Ebase	Module	Angle	Pgen	Qgen	Gsh	Bsh
1	SW1	1	1	1	0	0	0	0	0
2	PQ2	3	1	1	0	0	0	0	0,1
3	PQ3	3	1	1	0	0	0	0	0,2

Tableau-A II-2 Données des branches triphasées pour le simulateur

Barre départ	Barre arrivée	CKT	Nom	Rés.	Réac.	Cond.	Susc.
1	2	1	'L_1_2'	0	0,2	0	0
2	3	1	'L_1_3'	0	0,25	0	0

Tableau-A II-3 Résultats des itérations avec la méthode Levenberg-Marquardt

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	$1,5266 \times 10^{-16}$ (3-B)	-0,2 (3-A)	0,387298
1	$5,7333 \times 10^{-16}$ (2-B)	0,0313273 (3-B)	0,0559746
2	$6,3988 \times 10^{-16}$ (2-B)	0,000413356 (3-A)	0,000722548
3	$-5,1870 \times 10^{-16}$ (2-B)	$7,21054 \times 10^{-08}$ (3-A)	$1,25179 \times 10^{-07}$
4	$-5,0294 \times 10^{-16}$ (2-B)	$5,75783 \times 10^{-15}$ (2-B)	$1,1444 \times 10^{-14}$

Tableau-A II-4 Résultats des puissances des barres avec la méthode Levenberg-Marquardt

Barre#	Type	Valeur (pu)
1a	Sgen	- j 0,993266
2a	Sshunt	- j 0,113682
2b	Sshunt	- j 0,113682
2c	Sshunt	- j 0,113682
3a	Sshunt	- j 0,251927
3b	Sshunt	- j 0,251927
3c	Sshunt	- j 0,251927

Sgen : Puissance générée par la barre de génération.

Sshunt : Puissance consommée par la barre.

Tableau-A II-5 Résultats des puissances des lignes avec la méthode Levenberg-Marquardt pour le premier cas

Type	Barre#	Valeur (pu)
Ligne triphasée Barre 1 à 2	1a	0 - j 0,331089
	1b	-2,23562e-016 - j 0,331089
	1c	2,23562e-016 - j 0,331089
	2a	0 + j 0,353013
	2b	-5,55112e-017 + j 0,353013
	2c	5,55112e-017 + j 0,353013
Ligne triphasée Barre 2 à 3	2a	0 - j 0,239331
	2b	2,77556e-017 - j 0,239331
	2c	-2,77556e-017 - j 0,239331
	3a	0 + j 0,251927
	3b	-4,16334e-017 + j 0,251927
	3c	4,16334e-017 + j 0,251927

ANNEXE III

ÉTUDE DE CAS 2

Tableau-A III-1 Données des barres triphasées pour le simulateur

Numéro de barre	Nom	Type	Ebase	Module	Angle	Pgen	Qgen	Pchg	Qchg
1000	SW1	1	1	1	0	0	0	0	0
2000	PV2	2	1	1	0	0,6	0	0	0
3000	PQ3	3	1	1	0	0	0	0,8	0,13

Tableau-A III-2 Données des branches triphasées pour le simulateur

Barre départ	Barre arrivée	CKT	Nom	Rés.	Réac.	Cond.	Susc.
1000	2000	1	'L_1_2'	0	0,50	0	0
1000	3000	1	'L_1_3'	0	0,25	0	0
2000	3000	1	'L_2_3'	0	0,25	0	0

Tableau-A III-3 Résultats des itérations avec la méthode Levenberg-Marquardt

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	-1,8 (2000-A)	0,13 (3000-A)	2,2827
1	-0,024658 (2000-A)	0,0405569 (3000-B)	0,0777626
2	-0,000136 (2000-A)	0,000216186 (3000-B)	0,000423924
3	$-4,31 \times 10^{-9}$ (2000-A)	$6,45 \times 10^{-9}$ (3000-A)	$1,29481 \times 10^{-8}$
4	$-6,66 \times 10^{-16}$ (3000-B)	$1,53 \times 10^{-15}$ (3000-C)	$1,90707 \times 10^{-15}$

Tableau-A III-4 Résultats des puissances des barres avec la méthode Levenberg-Marquardt

Barre#	Type	Valeur (pu)
1	Sgen	$0,6 + j 0,330185$
2	Sgen	$1,8 + j 0,331021$
3a	Sshunt	$0,8 + j 0,13$
3b	Sshunt	$0,8 + j 0,13$
3c	Sshunt	$0,8 + j 0,13$

Sgen : Puissance générée par la barre de génération.

Sshunt : Puissance consommée par la barre.

Tableau-A III-5 Résultats des puissances des lignes avec la méthode Levenberg-Marquardt pour le premier cas

Type	Barre#	Valeur (pu)
Ligne triphasée Barre 1 à 2	1a	-0,11722 + j 0,00343808
	1b	-0,11722 + j 0,00343808
	1c	-0,11722 + j 0,00343808
	2a	0,11722 + j 0,00343808
	2b	0,11722 + j 0,00343808
	2c	0,11722 + j 0,00343808
Ligne triphasée Barre 1 à 3	1a	0,31722 + j 0,106623
	1b	0,31722 + j 0,106623
	1c	0,31722 + j 0,106623
	3a	-0,31722 - j 0,0842241
	3b	-0,31722 - j 0,0842241
	3c	-0,31722 - j 0,0842241
Ligne triphasée Barre 2 à 3	2a	0,48278 + j 0,106902
	2b	0,48278 + j 0,106902
	2c	0,48278 + j 0,106902
	3a	-0,48278 - j 0,0457759
	3b	-0,48278 - j 0,0457759
	3c	-0,48278 - j 0,0457759

ANNEXE IV

ÉTUDE DE CAS 3

Tableau-A IV-1 Données des barres triphasées pour le simulateur

Numéro de barre	Nom	Type	Ebase	Module	Angle	Pgen	Qgen	Gsh	Bsh
1	SW1	1	1	1	0	0	0	0	0
2	PQ2	3	1	1	0	0	0	0,1	0
3	PV3	2	1	1	0	0	0	0,1	0,3
4	PQ5	3	1	1	0	0	0	0,1	0,3

Tableau-A IV-2 Données des barres monophasées pour le premier cas pour le simulateur

Numéro de barre	Nom	Type	Ebase	Module	Angle	Pgen	Qgen	Pchg	Qchg
4	PQ4	3	1	1	0	0	0	1	0

Tableau-A IV-3 Données des branches triphasées pour le simulateur

Barre départ	Barre arrivée	CKT	Nom	Rés.	Réac.	Cond.	Susc.
1	2	1	'L_1_2'	0	0,2	0	0
2	3	1	'L_2_3'	0	0,2	0	0
3	5	1	'L_3_5'	0	0,2	0	0

Tableau-A IV-4 Données des branches monophasées pour le premier cas pour le simulateur

Barre départ	Barre arrivée	CKT	Nom	Rés.	Réac.	Cond.	Susc.
3a	4a	1	'L_3_4'	0	0,2	0	0

Tableau-A IV-5 Résultats des itérations avec la méthode Levenberg-Marquardt

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	1 (4-A)	-0,3 (4-A)	1,19164
1	0,0200586 (3-A)	0,0995802 (4-A)	0,165721
2	0,000583951 (4-A)	0,00208718 (4-A)	0,0023835
3	$5,3720 \times 10^{-7}$ (4-A)	$1,12223 \times 10^{-6}$ (4-A)	$1,35122 \times 10^{-6}$
4	$2,2893 \times 10^{-13}$ (4-A)	$3,60822 \times 10^{-13}$ (4-A)	$4,83272 \times 10^{-13}$

Tableau-A IV-6 Résultats des puissances des barres avec la méthode Levenberg-Marquardt

Barre#	Type	Valeur (pu)
1	Sgen	$0,6 + j 0,330185$
2a	Sshunt	$0,0985495 + j 0$
2b	Sshunt	$0,0985495 + j 0$
2c	Sshunt	$0,0985495 + j 0$
3	Sgen	$0 - j 1,44182$
	Sshunt	$0,3 - j 0,9$
4	Schg	$1 - j 0$
5a	Sshunt	$0,113122 - j 0,339367$
5b	Sshunt	$0,113122 - j 0,339367$
5c	Sshunt	$0,113122 - j 0,339367$

Sgen : Puissance générée par la barre de génération.

Sshunt : Puissance consommée par la barre.

Schg : Puissance consommée par la charge.

Tableau-A IV-7 Données des barres monophasées pour le deuxième cas pour le simulateur

Numéro de barre	Nom	Type	Ebase	Module	Angle	Pgen	Qgen	Pchg	Qchg
4	PQ4	3	1	1	0	0	0	0,0125	0
5	PQ5	3	1	1	0	0	0	0,0125	0
6	PQ6	3	1	1	0	0	0	0,0125	0
7	PQ7	3	1	1	0	0	0	0,0125	0
...
83	PQ83	3	1	1	0	0	0	0,0125	0

Tableau-A IV-8 Données des branches monophasées
pour le deuxième cas pour le simulateur

Barre départ	Barre arrivée	CKT	Nom	Rés.	Réac.	Cond.	Susc.
3a	4a	1	'L_3_4'	0	0,0025	0	0
4a	5a	1	'L_4_5'	0	0,0025	0	0
...
82a	83a	1	'L_4_5'	0	0,0025	0	0

Tableau-A IV-9 Résultats des puissances des lignes avec la méthode Levenberg-Marquardt
pour le premier cas

Type	Barre#	Valeur (pu)
Ligne triphasée Barre 1 à 2	1a	0,645005 + j 0,0784808
	1b	0,645005 + j 0,0784808
	1c	0,645005 + j 0,0784808
	2a	-0,645005 + j 0,0059573
	2b	-0,645005 + j 0,0059573
	2c	-0,645005 + j 0,0059573
Ligne triphasée Barre 2 à 3	2a	0,546456 - j 0,0059573
	2b	0,546456 - j 0,0059573
	2c	0,546456 - j 0,0059573
	3a	-0,546456 + j 0,0665662
	3b	-0,546456 + j 0,0665662
	3c	-0,546456 + j 0,0665662
Ligne triphasée Barre 3 à 5	3a	0,113122 - j 0,316742
	3b	0,113122 - j 0,316742
	3c	0,113122 - j 0,316742
	5a	-0,113122 + j 0,339367
	5b	-0,113122 + j 0,339367
	5c	-0,113122 + j 0,339367
Ligne monophasée Barre 3 à 4	3a	1 + j 0,208712
	4a	-1 + j 3,60156e-013

ANNEXE V

ÉTUDE DE CAS 4

Tableau-A V-1 Données des barres triphasées pour le simulateur

Numéro de barre	Nom	Type	Ebase	Module	Angle	Pgen	Qgen	Gsh	Bsh
1	SW1	1	1	1	0	0	0	0	0
2	PQ2	3	1	1	0	0	0	0,1	0

Tableau-A V-2 Données des branches triphasées pour le simulateur

Barre départ	Barre arrivée	CKT	Nom	Rés.	Réac.	Cond.	Susc.
1	2	1	'L_1_2'	0	0,2	0	0

Tableau-A V-3 Données des barres monophasées pour le simulateur

Numéro de barre	Nom	Type	Ebase	Module	Angle	Pgen	Qgen	Gsh	Bsh
3b	PQ3	3	1	1	-120	0	0	0,5	0

Tableau-A V-4 Données des transformateurs monophasés pour le simulateur

Barre départ	Barre arrivée	Z ₁	Z ₂	a _{1min}	a ₁	a _{1max}	N _p	BCNO	V _{min}	V _{max}
2b	3b	0,5i	0,5i	0,9	1	1,1	11	3b	0,50	0,82

Tableau-A V-5 Résultats des itérations avec la méthode Levenberg-Marquardt

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	0,5 (3-B)	$4,78784 \times 10^{-16}$ (2-B)	0,52915
1	0,0209063 (3-B)	0,157954 (2-B)	0,201507
2	0,00242152 (2-B)	0,0143017 (3-B)	0,0172408
3	0,00721051 (3-B)	-0,0211043 (2-B)	0,0276202
4	0,00692693 (3-B)	-0,0200048 (2-B)	0,0261961
5	0,00655902 (3-B)	-0,0189522 (2-B)	0,0248111
6	0,000111842 (3-B)	0,00022369 (3-B)	0,000254344
7	$4,3915 \times 10^{-08}$ (3-B)	$8,78314 \times 10^{-08}$ (3-B)	$9,82823 \times 10^{-08}$
8	$5,3823 \times 10^{-15}$ (3-B)	$1,06961 \times 10^{-14}$ (3-B)	$1,20439 \times 10^{-14}$

Tableau-A V-6 Résultats des puissances des barres avec la méthode Levenberg-Marquardt

Barre#	Type	Valeur (pu)
1	Sgen	$0,621834 + j 0,213035$
2a	Sshunt	$0,09996 + j 0$
2b	Sshunt	$0,0925254 + j 0$
2c	Sshunt	$0,09996 + j 0$
3a	Sshunt	$0,329389 + j 0$

Sgen : Puissance générée par la barre de génération.

Sshunt : Puissance consommée par la barre.

Tableau-A V-7 Résultats des puissances des lignes avec la méthode Levenberg-Marquardt

Type	Barre#	Valeur (pu)
Ligne triphasée Barre 1 à 2	1a	$0,09996 + j 0,0019992$
	1b	$0,421914 + j 0,209036$
	1c	$0,09996 + j 0,0019992$
	2a	$-0,09996 - j 1,88261e-015$
	2b	$-0,421914 - j 0,164695$
	2c	$-0,09996 + j 0$
Transformateur mono. Barre 2 à 3	2b	$0,329389 + j 0,164695$
	3b	$-0,329389 + j 1,06026e-014$

ANNEXE VI

ÉTUDE DE CAS 5

Tableau-A VI-1 Données des barres triphasées pour le simulateur

Numéro de barre	Nom	Type	Ebase	Module	Angle	Pgen	Qgen	Gsh	Bsh
1	SW1	1	1	1	0	0	0	0	0
2	PV2	2	1	1	0	0,6	0	0	0

Tableau-A VI-2 Données des barres monophasées pour le simulateur

Numéro de barre	Nom	Type	Ebase	Module	Angle	Pgen	Qgen	Pchg	Qchg
3a	PQ3a	3	1	1	0	0	0	0,7	0,12
3b	PQ3b	3	1	1	-120	0	0	0,8	0,13
3c	PQ3c	3	1	1	120	0	0	0,9	0,14

Tableau-A VI-3 Données des branches triphasées pour le simulateur

Barre départ	Barre arrivée	CKT	Nom	Rés.	Réac.	Cond.	Susc.
1	2	1	'L_1_2'	0	0,50	0	0
1	3	1	'L_1_3'	0	0,20	0	0
2	3	1	'L_2_3'	0	0,25	0	0

Tableau-A VI-4 Résultats des itérations avec la méthode Levenberg-Marquardt

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	-1,8 (2-A)	0,14 (3-C)	2,28712
1	-0,0248146 (2-A)	0,0500925 (3-C)	0,0795567
2	-0,00014353 (2-A)	0,000332986 (3-C)	0,000472412
3	$7,162 \times 10^{-9}$ (3-C)	1.56441×10^{-8} (3-C)	$1,95634 \times 10^{-8}$
4	-9.9920×10^{-16} (3-C)	1.97065×10^{-15} (3-C)	3.33922×10^{-15}

Tableau-A VI-5 Résultats des puissances des barres avec la méthode Levenberg-Marquardt

Barre#	Type	Valeur (pu)
1	Sgen	0,6 + j 0,33159
2	Sgen	1,8 + j 0,332148
3a	Schg	0,7 + j 0,12
3b	Schg	0,8 + j 0,13
3c	Schg	0,9 + j 0,14

Sgen : Puissance générée par la barre de génération.

Schg : Puissance consommée par la charge.

Tableau-A VI-6 Résultats des puissances des lignes avec la méthode Levenberg-Marquardt

Type	Barre#	Valeur (pu)
Ligne triphasée Barre 1 à 2	1a	-0,117226 + j 0,00343843
	1b	-0,117226 + j 0,00343843
	1c	-0,117226 + j 0,00343843
	2a	0,117226 + j 0,00343843
	2b	0,117226 + j 0,00343843
	2c	0,117226 + j 0,00343843
Ligne triphasée Barre 1 à 3	1a	0,26125 + j 0,0925867
	1b	0,317214 + j 0,106623
	1c	0,373214 + j 0,122065
	3a	-0,26125 - j 0,077222
	3b	-0,317214 - j 0,0842247
	3c	-0,373214 - j 0,0912273
Ligne triphasée Barre 2 à 3	2a	0,43875 + j 0,093069
	2b	0,482786 + j 0,106903
	2c	0,526786 + j 0,121861
	3a	-0,43875 - j 0,042778
	3b	-0,482786 - j 0,0457753
	3c	-0,526786 - j 0,0487727

ANNEXE VII

ÉTUDE DE CAS 6

Tableau-A VII-1 Données des barres triphasées pour le simulateur

Numéro de barre	Nom	Type	Ebase	Module	Angle	Pgen	Qgen	Gsh	Bsh
1	SW1	1	1	1	0	0	0	0	0
2	PQ2	3	1	1	0	0	0	0,1	0
3	PQ3	3	1	1	0	0	0	0,1	0,3
4	PQ4	3	1	1	0	0	0	0,1	0,3
5	PQ5	3	1	1	0	0	0	0,1	0,1

Tableau-A VII-2 Données des branches triphasées pour le simulateur

Barre départ	Barre arrivée	CKT	Nom	Rés.	Réac.	Cond.	Susc.
1	2	1	'L 1 2'	0	0,20	0	0
2	3	1	'L 1 3'	0	0,20	0	0
4	5	1	'L 2 3'	0	0,20	0	0

Tableau-A VII-3 Données des transformateurs triphasés pour le simulateur

Barre départ	Barre arrivée	Z ₁	Z ₂	Y	a _{1min}	a ₁	a _{1max}	N _p	BCNO	V _{min}	V _{max}
3	4	0,5i	0,5i	-1i	0,9	1	1,1	11	2	0,92	0,96

Tableau-A VII-4 Résultats des itérations avec la méthode Levenberg-Marquardt

Itération	Pmax (Barre)	Qmax (Barre)	Sqrt (S)
0	0.1 (2-B)	0.1 (4-C)	0,458258
1	0,0104734 (4-C)	0,0321631 (3-B)	0,0834827
2	0,000565734 (4-A)	0,00138292 (4-B)	0,00318661
3	-0,00330063 (3-C)	-0,0279792 (3-A)	0,053997
4	0,00332254 (3-C)	0,0288278 (3-B)	0,0551485
5	0,00347614 (3-B)	0,0301856 (3-B)	0,0577462
6	0,00364417 (3-C)	0,0316464 (3-A)	0,060534
7	5,0706 x 10 ⁻⁰⁵ (3-C)	0,000240894 (3-A)	0,0005456
8	2,1153 x 10 ⁻⁰⁸ (4-C)	8,28596 x 10 ⁻⁰⁸ (4-C)	1,67297 x 10 ⁻⁰⁷
9	2,6092 x 10 ⁻¹⁵ (4-A)	3,88347 x 10 ⁻¹⁴ (5-B)	9,2455 x 10 ⁻¹⁴

Tableau-A VII-5 Résultats des puissances des barres avec la méthode Levenberg-Marquardt

Barre#	Type	Valeur (pu)
1	Sgen	1,06328 + j 0,648225
2a	Sshunt	0,0920462 + j 0
2b	Sshunt	0,0920462 + j 0
2c	Sshunt	0,0920462 + j 0
3a	Sshunt	0,0852242 - j 0,255673
3b	Sshunt	0,0852242 - j 0,255673
3c	Sshunt	0,0852242 - j 0,255673
4a	Sshunt	0,086807 - j 0,260421
4b	Sshunt	0,086807 - j 0,260421
4c	Sshunt	0,086807 - j 0,260421
5a	Sshunt	0,0903487 - j 0,0903487
5b	Sshunt	0,0903487 - j 0,0903487
5c	Sshunt	0,0903487 - j 0,0903487

Sgen : Puissance générée par la barre de génération.

Sshunt : Puissance consommée par la barre.

Tableau-A VII-6 Résultats des puissances des lignes avec la méthode Levenberg-Marquardt

Type	Barre#	Valeur (pu)
Ligne triphasée Barre 1 à 2	1a	$0,354426 + j 0,216075$
	1b	$0,354426 + j 0,216075$
	1c	$0,354426 + j 0,216075$
	2a	$-0,354426 - j 0,181614$
	2b	$-0,354426 - j 0,181614$
	2c	$-0,354426 - j 0,181614$
Ligne triphasée Barre 2 à 3	2a	$0,26238 + j 0,181614$
	2b	$0,26238 + j 0,181614$
	2c	$0,26238 + j 0,181614$
	3a	$-0,26238 - j 0,159489$
	3b	$-0,26238 - j 0,159489$
	3c	$-0,26238 - j 0,159489$
Ligne triphasée Barre 4 à 5	4a	$0,0903487 - j 0,0867347$
	4b	$0,0903487 - j 0,0867347$
	4c	$0,0903487 - j 0,0867347$
	5a	$-0,0903487 + j 0,0903487$
	5b	$-0,0903487 + j 0,0903487$
	5c	$-0,0903487 + j 0,0903487$
Transformateur triphasé Barre 3 à 4	3a	$0,177156 + j 0,415161$
	3b	$0,177156 + j 0,415161$
	3c	$0,177156 + j 0,415161$
	4a	$-0,177156 + j 0,347156$
	4b	$-0,177156 + j 0,347156$
	4c	$-0,177156 + j 0,347156$

ANNEXE VIII

ÉTUDE DE CAS 7

Tableau-A VIII-1 Quelque données des barres triphasées pour le simulateur

Numéro de barre	Type	Module	Angle	Pgen	Qgen	Pchg	Qchg
1001	1	1.1	0	7.7061	2.5910	0	0
1002	3	1.0186	-19.14	0	0	-1.362	-0.348
1003	3	0.93904	-61.065	0	0	1.008	-0.133
1004	3	0.93951	-60.114	0	0	0.909	-0.133
1005	3	0.98722	-43.66	0	0	0.154	-0.092
1006	3	0.98606	-43.854	0	0	0.218	-0.083
1007	3	0.98630	-43.932	0	0	0.207	-0.072
...
4000	3	1.0214	-22.302	0	0	-1.251	-0.290

Tableau-A VIII-2 Quelque données des branches triphasées pour le simulateur

Barre départ	Barre arrivée	CKT	Rés.	Réac.	Cond.	Susc.
1005	1219	1	0.00585038	0.02677424	0	0.00450412
1006	1220	1	0.00369341	0.03654067	0	0.00554043
1007	1221	1	0.00482052	0.03622532	0	0.00525757
...
4000	3999	1	0.00339594	0.03425244	0	0.00456863

BIBLIOGRAPHIE

- Aravindhababu, P. et R. Ashokkumar. 2008. « A Fast Decoupled Power Flow for Distribution Systems », Electric power components and systems, volume 36: p.932-940.
- Chellaboina, V. et W. M. Haddad. 1995. « Is the Frobenius matrix norm induced? », IEEE Transactions on Automatic Control, volume 40: p.2137-2139.
- De Oliveira-De Jesus, P. M., M. A. Alvarez et J. M. Yusta. 2013. « Distribution power flow method based on a real quasi-symmetric matrix », Electric power systems research, volume 95: p.148-159.
- Durga Prasad, G., A. K. Jana et S. C. Tripathy. 1990. « Modifications to Newton-Raphson load flow for ill-conditioned power systems », International journal of electrical power & energy systems, volume 12: p.192-196.
- Esmaeli, Abdolreza, Mohammad Abedini et Mohammad H. Moradi. 2016. « A novel power flow analysis in an islanded renewable microgrid », Renewable energy, volume 96: p.914-927.
- Frank, Stephen, Ingrida Steponavice et Steffen Rebennack. 2012. « Optimal power flow: a bibliographic survey I: Formulations and deterministic methods », p221-258. Published online: 24 April 2012 © Springer-Verlag 2012.
- Hundt, Robert. 2011. « Loop Recognition in C++/Java/Go/Scala ». In Proceedings of Scala Days 2011. (California, 2-3 juin 2011), p.1-10. Google: Scala Days 2011.
- Kamel, S., M. Abdel Akher et F. Jurado. 2013. « Improved NR current injection load flow using power mismatch representation of PV bus », International journal of electrical power & energy systems, volume 53: p.64-68.
- Kılıç, Ulaş et Kürşat Ayan. 2014. « Optimal power flow solution of two-terminal HVDC systems using genetic algorithm », Electrical Engineering, volume 96: p.65-77.
- Kim, Chan-Yong, Jeong-Min Kang, Ji-Hoon Kim, Bong-Yeol Choi et Kyung-Youn Kim. 2012. « Modified Newton-Raphson method using a region of interest in electrical impedance tomography », Journal of the Korean Physical Society, volume 61: p.1199-1205.
- Kocar, Ilhan, Jean Mahseredjian, Ulas Karaagac, Gurkan Soykan, et Omar Saad. 2014. "Multiphase Load-Flow Solution for Large-Scale Distribution Systems Using MANA." In IEEE Transactions on Power Delivery, p.908-915.

- Kulworawanichpong, Thanatchai. 2010. « Simplified Newton–Raphson power-flow solution method », International journal of electrical power & energy systems, volume 32: p.551-558.
- Lagacé, Pierre-Jean. 2011. « Programme d'écoulement de puissance NR ». 84 pages. En ligne. < http://prof.etsmtl.ca/pjlagace/Recherche/nr3/Latex_2015_06_11.zip >. Consulté le 6 janvier 2015.
- Lagacé, Pierre-Jean, M.H. Vuong et I. Kamwa. 2008. «Improving Power Flow Convergence by Newton Raphson with a Levenberg-Marquardt Method », IEEE PES GM, paper 08GM0391.
- Li, Yan, Yulei Luo, Buhan Zhang et Chengxiong Mao. 2011. « A Modified Newton-Raphson Power Flow Method Considering Wind Power ». In 2011 Asia-Pacific Power and Energy Engineering Conference. (China, 25 - 28 mars 2011), p.1-5. Huazhong University of Science and Technology: IEEE conference.
- Milano, F. 2005. 'An Open Source Power System Analysis Toolbox', IEEE Transactions on Power Systems, volume 20: p.1199-1206.
- Nguyen, Cuong, Cuong P. Nguyen et Alexander Flueck. 2015. « A Novel Agent-Based Distributed Power Flow Solver for Smart Grids. » In IEEE Transactions on Smart Grid, p.1261-1270.
- Pazderin, Andrey et Sergey Yuferev. 2009. « Power Flow Calculation by Combination of Newton-Raphson Method and Newton's Method in Optimization ». In Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE. (Portugal, 3-5 November 2009), p.1693-1696. Ural State Technical University: IEEE conference.
- Sundaresh, Lakshmi et P. S. Nagendra Rao. 2014. « A modified Newton–Raphson load flow scheme for directly including generator reactive power limits using complementarity framework », Electric power systems research, volume 109 : p.45-53.
- Teng, Jen-Hao, Teng Jen-Hao et Chang Chuo-Yean. 2002. « A novel and fast three-phase load flow for unbalanced radial distribution systems. » In IEEE Transactions on Power Systems, p.1238-1244.
- Theocharis, Andreas, Vasilis Charalampakos, Anastasios Drosopoulos et John MiliasArgitis. 2012. « Equivalent circuit of photovoltaic generator using Newton-Raphson algorithm », Compel, volume 31: p.1224-1245.
- Torelli, Francesco, Alfredo Vaccaro et Ning Xie. 2013. « A Novel Optimal Power Flow Formulation Based on the Lyapunov Theory. » In IEEE Transactions on Power Systems, p.4405-4415.