



Universidad de Oviedo

Trabajo de Fin de Grado

ANO 2018-2019:

TITULO :

Desarrollo del algoritmo de flujo de cargas para redes de distribución eléctricas malladas desequilibradas

Nombre y Apellidos del Tutor de la Escuela :

Joaquin Gonzalez Normiella

Nombre y Apellidos del Estudiante : Pablo Dominguez Chans

Contenido

Agradecimientos.....	6
Glosario	7
Resumen	8
Abstracto.....	9
1. Introduccion.....	10
2. Presentacion de Atos y Atos Worldgrid.....	12
3. Presentacion del tema de las practicas	13
3.1. Cuaderno de especificaciones y descripcion del proyecto	13
4. Investigacion sobre el entorno tecnico del proyecto	14
4.1. Metodos de flujo de cargas	14
4.2. Decision del lenguaje de programacion.....	17
4.3. Programacion	18
4.3.1. Entorno tecnico	18
4.3.2. Funciones del algoritmo.....	20
4.4. Datos de entrada.....	26
5. Implementacion sobre el SCADA DMS Lynx	28
5.1. Presentacion de SCADA Lynx.....	28
5.2. Presentacion del servidor « Loadflow » de LYNX	29
5.3. Implementacion del Loadflow trifasico desequilibrado en LYNX	29
5.3.1. Maquina de ensayos y aplicación dentro de LYNX	30
5.3.2. Implementacion y simulacion con datos externos a LYNX.....	30
5.3.3. Modificacion de la base de datos existente	31
5.3.4. Conexión con los datos en LYNX.....	32
5.4. Exposicion del proyecto a los clientes	33
6. Conclusiones generales sobre el proyecto	34
6.1. Gestion del proyecto : retorno de experience y dificultades encontradas.....	35
6.2. Conclusion general.....	35
7. Bibliografia.....	36
8. Anexo sobre la red en LYNX	37

Resumen

Newton Raphson trifásico desequilibrado es un algoritmo de flujo de cargas que propone una solución para la supervisión de redes de distribución eléctricas desequilibradas radiales y malladas en presencia de fuentes distribuidas. Simplemente, permite determinar el estado completo de la red eléctrica, sabiendo las tensiones (tanto la magnitud como el ángulo de desfase) en todos sus nudos, las potencias inyectadas y consumidas en cada nudo, el tránsito de corriente y de potencias (tanto activa como reactiva) en cada rama, así como las pérdidas en cada ramal. La estabilidad de las redes eléctricas es uno de los componentes más importantes que debemos tener en cuenta. Por ello, en este algoritmo tendremos en cuenta: el número de iteraciones, el error residual, así como el tiempo que el algoritmo tarda en converger y de esta forma podremos conseguir una precisión muy alta.

El método de Newton Raphson trifásico desequilibrado es un algoritmo utilizado para el flujo de cargas dentro de las redes y mediante este es posible realizar la inyección de corriente a través de varios nudos gracias a las fuentes distribuidas. Anteriormente, la corriente se inyectaba mediante un solo nudo, pero ahora ha sido propuesto un nuevo modelo para disminuir las pérdidas en el cable o para inyectar electricidad donde la demanda lo requiera.

Con este algoritmo, nosotros podemos restablecer los valores de las magnitudes más importantes mediante compensadores estáticos o VSC gracias a los parámetros proporcionados en cada rama y cada nudo.



RAPPORT

ANNEE 2017 - 2018

TITRE :

Développement de l'algorithme de répartition de charges pour les réseaux de distribution électrique maillés déséquilibrés

Nom de l'entreprise : ATOS WORLDGRID

Adresse : 28, Rue Gustave Eiffel, 38000 Grenoble

Nom et prénom du Maître de Stage (dans l'entreprise / le laboratoire) :

LEMIUS Lucile

Nom et prénom du Tuteur Ecole

CAIRE Raphael

Stage de :

3A Projet de Fin d'Etudes Filière : ASI HOE GEN IEE ME SEM MASTER
ETUDIANTS A L'ETRANGER AUTRES

Nom et prénom de l'étudiant : DOMINGUEZ CHANS Pablo

Sujet confidentiel : Non

Un rapport confidentiel ne pourra être communiqué à quiconque (étudiant, enseignant, personne extérieure) sauf autorisation spéciale accordée par le Maître de Stage dans l'entreprise/le laboratoire

Contenu

Remerciements.....	6
Glossaire.....	7
Résumé.....	8
Abstract.....	9
1. Introduction	10
2. Présentation de Atos et Atos Worldgrid.....	12
3. Présentation du sujet de stage	13
3.1. Cahier de charges et description du projet	13
4. Recherches sur l’environnement technique du projet.....	14
4.1. Méthodes de répartition de charges	14
4.2. Décision du langage	17
4.3. Programmation	18
4.3.1. Environnement technique	18
4.3.2. Fonctions.....	20
4.4. Données d’entrées	26
5. Implémentation sur le SCADA DMS Lynx.....	28
5.1. Présentation du SCADA Lynx	28
5.2. Présentation du serveur « Loadflow » du LYNX	29
5.3. Implémentation du Loadflow triphasé déséquilibré dans LYNX.....	29
5.3.1. Machine de test et application LYNX disponible.....	30
5.3.2. Implémentation et tests avec des données externes au LYNX	30
5.3.3. Modification de la base de données existante	31
5.3.4. Interfaçage avec les données LYNX	32
5.4. Présentation aux clients.....	33
6. Conclusions générales sur le projet	34
6.1. Gestion du projet : retour d’expérience et difficultés rencontrées	34
6.2. Conclusion générale.....	35
7. Bibliographie	36
8. Annexe du réseau sur LYNX.....	37

Remerciements

Tout d'abord, j'aimerais remercier ma maîtresse de stage Lucile Lemius pour toute l'aide concernant l'environnement de travail et aussi pour m'avoir appris des qualités personnelles très importantes, (pour sa disponibilité, sa patience pour m'avoir expliqué le fonctionnement de LYNX et les éco-conversations). Nous avons passé de bons moments pendant les afterworks sur Place Grenette et elle est un exemple de travail et d'altruisme. Je remercie également mon tuteur de stage Raphael Caire qui m'a donné le support technique et les connaissances sur mon sujet de stage. Je suis très heureux d'avoir eu des professeurs qui m'ont aidé d'une façon directe ou indirecte sur mon projet. Sur l'environnement de travail, j'aimerais remercier Patrick Chitdara pour l'aide qu'il m'a apporté qui est plus qu'un collègue et aussi Choab Bousba qui m'a beaucoup aidé sur la partie technique et m'a montré beaucoup de choses.

Après, la personne qui j'aimerais remercier le plus est Arthur Jourdin qui m'a beaucoup aidé dans les moments de bugs et les moments des réussites. En plus, il m'a donné des très belles et interminables conversations ainsi que expériences.

Pour conclure, j'aimerais remercier ma famille qui est la pierre angulaire de mon développement en tant que personne et en tant qu'étudiant.

« Donnant, donnant »

-P.C.

« Celui qui est pauvre n'est pas celui qui possède peu, mais celui qui a besoin de beaucoup et qui désire toujours en avoir plus. »

-José Mujica

Glossaire

BFS: Backward Forward Sweep

NR: Newton Raphson

DMS : Distribution Management System

GED : Générateur d'Énergie Distribuée

SCADA : Supervision Control And Data Acquisition

SSII : Société de Services en Ingénierie Informatique

PV : Panneaux Photovoltaïques

NR : Newton Raphson

DS : Distributed Sourcing

EnR : Energies renouvelables

Résumé

Le Newton Raphson triphasé déséquilibré est un algorithme de répartition de charges qui propose une solution pour la surveillance des réseaux de distribution électrique déséquilibrés radiaux et maillés en présence de sources distribués. En termes simples, il permet de déterminer l'état électrique complet du réseau, à savoir les tensions (phase, amplitude) à tous les nœuds, les injections et consommation de puissance en chaque nœud, les transits de courant et de puissances (active, réactive) dans toutes les branches, ainsi que les pertes dans chaque branche. La stabilité des réseaux électriques est un problème majeur à résoudre. Cela est pour cette raison qu'avec cet algorithme sur la méthode numérique nous prendrons en compte : le nombre d'itérations, l'erreur résiduelle ainsi que le temps que l'algorithme prend pour converger.

La méthode Newton Raphson triphasé déséquilibré est un algorithme utilisé pour la répartition de charges sur le réseau pour injecter courant avec plusieurs nœuds grâce aux GED. Avant, la courant était injecté par un seul nœud, mais maintenant a été proposé un nouveau modèle pour sauver en pertes en câble.

Avec l'algorithme, nous pouvons rétablir les valeurs des grandeurs dans les zones avec perturbations avec le raccordement à un STATCOM ou VSC grâce aux paramètres données au Newton Raphson triphasé dans chaque nœud et ligne du réseau.

Mots clef: Smart grid, SCADA, DMS, Newton Raphson, Power Flow, triphasé déséquilibré, GED

Abstract

The unbalanced three-phase Newton Raphson method is algorithm is a known load balancing algorithm which proposed a solution in order to monitor radial and meshed unbalanced distribution networks. Basically, it allows us to determine the state of the electrical grid through the voltage (voltage magnitude and phase angle) on each of the nodes, the complex consumption and injections at any node, the current and complex power transit as well as the losses on each of the branches. The stability of electrical network is mayor problem to solve since disturbances produce alteration on the quality of the electricity. This is the reason why this algorithm should be taken into account certain parameters such us: the number of iterations, the residual error and also the time taken so that it converges.

The unbalanced three phase NR is being used as a load balancing method due to the connection of DS to the electrical distribution grid so that it can be done current injection through many nodes. In previous models, the current was injected through a single node so the numerical method would be applied in a single phase then, it would be considered as "balanced". With this proposed model, our main aim is to do loss minimization through the positioning of distributed sourcing which are localized in the distribution network instead of just sending electricity from the production grid.

With the shown algorithm, we will be able to restore the values of voltage, both magnitude and angle, in local areas of the distribution grid due to the presence of disturbances. This process will be done thanks to the connection to a VSC or a STATCOM thanks to the parameters given through NR.

Keywords: Smart grid, SCADA, DMS, Newton Raphson, Power Flow, Three phase unbalanced, DS

1. Introduction

Un **réseau triphasé** est un réseau qui possède trois phases ou trois phases et un neutre. Les réseaux triphasés se trouvent dans la partie production, transmission et distribution du réseau jusqu'aux bâtiments car il y en a certains en réseau monophasé.

Les **réseaux maillés** ont une topologie de réseau qualifiant les réseaux (filaires ou non) dont tous les hôtes sont connectés pair à pair sans hiérarchie centrale, formant ainsi une structure en forme de filet. Par conséquent, chaque nœud doit recevoir, envoyer et relayer les données

Les **réseaux déséquilibrés** sont des réseaux dans lesquels la tension pour chaque phase ou câble sont différentes. Ce type de réseaux possède des charges dont les impédances sont différentes. Cela peut être à cause du vieillissement des câbles ou des impédances. Tous les types de réseaux peuvent être déséquilibrés mais les seules dont la différence est significative sont les réseaux de distribution.

Les **GED** sont les groupes d'énergie distribués, il y en a plusieurs types comme les panneaux photovoltaïques, les éoliennes, les groupes électrogènes, groupes de stockage etc. Ils vont donner des perturbations sur le réseau dues à leur intermittence.

Dans le monde actuelle chaque fois la demande de électricité est de plus en plus importante, c'est pour ça qu'il faut développer les systèmes présentes afin d'avoir un meilleur rendement. Ainsi, les coûts d'ouvrage vont diminuer ainsi que les prix des ressources renouvelables, et grâce à ça, la dépendance en combustibles fossiles baissera.

Tout d'abords, nous avons regardé les contraintes présentes sur les réseaux électriques. Les différentes contraintes sur les réseaux sont : le ferromagnétisme, le flicker, la stabilité, le vieillissement des câbles, l'effet Joule etc. Aussi, avec l'implémentation des GED, nous regarderons cette problématique dans un second temps.

Les GED apportent beaucoup de bénéfices grâce à la diminution des pertes dans les lignes et à l'intégration des EnR proches des zones urbaines, mais l'intermittence de ces sources d'énergie comme les PV ou l'éolien rendent les calculs des charges sur les réseaux assez compliqués. De plus, certains réseaux sont dits « triphasés déséquilibrés », ce qui implique qu'il faut reprendre les calculs pour le réseau équilibré utilisant une seule phase, et implémenter les trois phases. Ce déséquilibre peut être causé par le raccordement monophasé des charges ou des productions sur le réseau, ou tout simplement par le vieillissement non homogène des équipements électriques.

Pour ces raisons, nous mettrons en place une méthode de calcul numérique pour les réseaux maillés de type triphasés comportant des charges déséquilibrées. Il faut utiliser une méthode de ce type parce qu'il faut calculer les différentes inconnues du réseau quand on regarde un

système non linéaire. De plus, avec cette méthode il est plus facile à calculer de calculer les paramètres c'est donc plus rapide.

Les plus grands différences qu'il y a sont qu'il faut changer les équations principales et les adapter en triphasée et linéariser la méthode de calcul. L'optimisation de la méthode de calcul de répartition des charges produira une épargne par rapport aux pertes *. De plus, c'est un investissement du futur dû à l'utilisation des énergies renouvelables.

Actuellement, le constant développement des énergies renouvelables, la minimisation de coûts et les politiques sociales pour intégrer au fur et à mesure les sources distribuées a produit sur le réseau électrique de distribution un énorme changement. Avant, les réseaux de distribution étaient envisagés avec un flux unidirectionnel de courant mais, maintenant la société de chercheurs a proposé une concertation plus efficace, économique et durable, les SMART GRIDS.

Les SMART GRIDS proposent un concept dans le domaine de l'énergie électrique pour minimiser les pertes sur les câbles en utilisant des sources distribuées pour injecter l'électricité sur le réseau d'une façon durable avec des panneaux photovoltaïques, éoliennes etc. Aussi, avec les compteurs intelligents (Linky) nous aurons dans une futur proche une analyse précise de la consommation pour bien prédire l'offre et la demande. Avec le sujet de stage proposé, nous regardons des paramètres sur le réseau pour bien gérer la stabilité en présence de plusieurs nœuds d'injection. Avec les SMART GRIDS, la production décentralisée peut être modélisée comme générateurs ou comme charges négatifs dépendant de nos besoins.

*Formule pour les pertes en câble

$$P = I^2R$$

2. Présentation de Atos et Atos Worldgrid

Atos est une SSII internationale avec un chiffre d'affaires de 13 milliards d'euros (année 2018), qui compte 100 000 collaborateurs dans 73 pays [ATO]. L'activité du groupe s'organise autour de 4 principaux métiers : le conseil et les services technologiques, l'intégration des systèmes, l'infogérance et les services transactionnels de haute technologie. Atos SE(Europaea) est cotée chez Euronext Paris.

Atos regroupe : Atos Consulting, Atos Worldgrid, Bull, Canopy, Unify and Worldline Societas. La filiale Worldgrid regroupe les activités autour de la supervision de la production, du transport et de la distribution d'énergie. AWG compte 1600 collaborateurs dans 15 pays. Elle a développé les SCADA de plus de 70 centrales nucléaires en France, Angleterre, Russie et Chine.

Atos Worldgrid, filiale internationale d'Atos, est un leader mondial des systèmes informatiques dans le domaine de l'énergie. Avec plus de 35 ans d'expertise du marché Energie & Utilities, Atos Worldgrid intègre et fournit des solutions innovantes pour la production, le transport, la distribution et la commercialisation de l'énergie. Nos solutions technologiques apportent une forte valeur ajoutée aux différents acteurs du marché de l'énergie, en leur permettant d'optimiser leurs processus et leurs systèmes d'information techniques ou de gestion. Atos Worldgrid développe mondialement son expertise grâce à plus de 1000 ingénieurs et consultants, et génère un chiffre d'affaires de €150 millions.

Durant les 6 mois de stage, j'ai travaillé pour l'équipe LYNX dans la section B&PS Smart Grid & Distribution sur AWG, sous la supervision de Lucile LEMIUS, analyste développeuse.

3. Présentation du sujet de stage

Le calcul de répartition de charges, aussi connu couramment comme Power Flow, est considéré comme un calcul très couramment utilisé dans le domaine électrique. Simplement, il nous permet de visualiser l'état électrique partout sur le réseau, visualiser les tensions à tous les nœuds, les transits de puissances dans toutes les branches, les pertes, etc.

Le SCADA Lynx permet aux clients énergéticiens de surveiller et de piloter leurs infrastructures et leurs équipements à la fois localement et à distance. Lynx est un système SCADA (Supervisory Control And Data Acquisition) multifonctions installé dans plus de 150 lieux dans le monde et dont le noyau couvre le besoin de différents procédés industriels : les réseaux de distribution et de transport d'électricité, les réseaux de distribution et de transport de gaz, les usines de production d'eau potable et de traitement des eaux usées, les usines de production hydroélectrique. Au-dessus du noyau générique, Lynx propose des fonctions avancées permettant de conduire chaque type de procédé et d'énergie en toute sécurité, en apportant l'aide la plus appropriée à l'opérateur.

Le lynx aussi permet de faire des calculs de répartition de charges sur les réseaux équilibrés monophasés mais ce n'est pas approprié pour les réseaux triphasés avec des charges déséquilibrées, comme c'est le cas sur la majorité des réseaux Basse Tension ou sur certains réseaux Moyenne Tension.

Objectif de ce stage

Ce stage consiste à étudier les différents algorithmes de calculs de répartition de charges existants pour les réseaux triphasés déséquilibrés, et à spécifier, concevoir, effectuer et tester l'intégration dans le produit Lynx de la solution jugée la plus convenable.

3.1. Cahier de charges et description du projet

Déroulement du stage

Recherche bibliographique sur les différentes méthodes de calcul de répartition de charges présentes en triphasé équilibré et déséquilibré

Codage de l'algorithme optimal pour le calcul de répartition de charges

Rédaction des spécifications techniques et fonctionnelles

Intégration de l'algorithme dans le SCADA Lynx : connexion avec le modèle de données existant, développement d l'interface graphique

4. Recherches sur l'environnement technique du projet

4.1. Méthodes de répartition de charges

Pour faire la recherche, je me suis appuyé sur différentes sources. Dès mon arrivé à Atos, j'ai reçu la documentation liée au fonctionnement du Lynx, les différents commandes, fichiers et fonctions sur Linux pour bien connaître la structure du programme.

Pour la partie technique, j'ai utilisé la source IEEE et mes connaissances apprises lors de ma formation à l'ENSE3.

Premièrement, j'ai étudié beaucoup de documentation dans les articles de recherche afin de mieux comprendre l'utilisation de plusieurs méthodes itératives pour choisir celle qui correspond mieux. J'ai pris contacte avec quelques chercheurs (Kees Vuik et Baljinnnyam Sereeter) pour avoir une approche plus profonde sur le Newton Raphson triphasé déséquilibré dont les plusieurs types de « mismatch » (soit puissance, courant) et aussi avec l'utilisation des différentes type de coordonnées.

J'ai également étudié le BFS (Backward Forward Sweep). Je pourrais ainsi comparer les méthodes pour faire un rapport sur le nombre d'itérations, puissance et erreur résiduel. Avec plusieurs rapports, livrés et bouquins, j'ai apprivoisé le fonctionnement mathématique et algorithmique de chacun.

Méthodes de calcul pour le réseau triphasé déséquilibré

Ce sont des méthodes pour étudier les algorithmes qui utilisent des approximations numériques dans des problèmes d'analyse mathématique. Avec ceux-ci, les résultats auront une précision plus importante. De plus, il faut étudier les contraintes pour effectuer le calcul de la meilleure façon possible. Les contraintes les plus importantes par rapport au manque de convergence sont les ceux cités ci-dessous :

Contraintes

Structure radiale ou faiblement maillé

R/X ratio important

Flux de puissance multi-phase

Charges déséquilibrées

Sources distribués (raccordement triphasé out monophasé)

Défis

- Haute vitesse de computation
- Simplicité du programme
- Flexibilité du programme
- Peu de stockage sur l'ordinateur
- Fiabilité de la solution

Méthodes de Calcul existantes

	Backward-Forward sweep	Newton Raphson	Newton-Raphson modifié	Gauss-Seidel	Fast Decoupled load flow
Sensibilité	+	+	++	-	+
Implémentation	++	+	-	++	+
Précision	+	+	++	-	+
Convergence	+	++	++	-	-
Caractéristiques	Ne convient pas pour les réseaux maillés	Convergence quadratique. Problème de convergence sur les réseaux radiaux	La matrice Jacobienne change chaque itération	Plus lente chaque itération, matrice Jacobienne ne change pas. Convergence linéale. Simple	Très lente mais intéressant pour le control optimal de tension. Convergence linéale.

Premièrement, la méthode de **Newton Raphson** semblait offrir peu de convergence, mais au fur et à mesure du temps, divers chercheurs ont implémenté des algorithmes supplémentaires pour résoudre les problèmes de convergence liée à cette méthode.

L'avantage du NR est qu'il possède une grande sensibilité et une précision importante. Dès que la matrice n'est pas dominante diagonalement, il faut linéariser le système matriciel. Aussi, les impédances mutuelles sont très importantes parce qu'elles peuvent conduire à des instabilités. Les différentes façons et types de raccordement du transformateur vont influencer le modèle de charges, qu'il faudra résoudre dans la première partie du calcul de la matrice d'impédance.

Deuxièmement, la méthode de **Backward Forward sweep**, est une méthode qui permet de calculer la répartition de charges appliqué aux réseaux de distribution déséquilibrés. Avec la méthode BFS, nous pouvons calculer la probabilité de Z_k par rapport à X_k . Cet algorithme de programmation dynamique a été utilisé la première fois par Richard Bellman. Si nous avons un ensemble de données x_1, \dots, X_n , nous assumerons une distribution. Dans le « Forward » algorithme, nous calculerons la probabilité de Z_k par rapport à $X_{1:k}$ ainsi que pour calculer le « Backward » algorithme nous regarderons la probabilité de $X_{k+1:n}$ par rapport à Z_k . Nous vous montrerons maintenant la comparaison dans le domaine de l'électricité.

Pour commencer, la partie « Forward » va être calculé et après la partie « Backward ». Pour calculer la partie « Backward » il faut calculer le KVL et le KCL pour savoir la tension pour chaque bus amont de la ligne ou la branche du transformateur. Ensuite, pour le « Forward sweep » s'utilise le principe proportionnel linéaire qui est employé pour mettre à jour la tension pour chaque amont bus.

Une fois qu'est calculé chaque « Backward sweep », l'incompatibilité de la tension calculée et spécifié dans la sous-station sera regardée.

Ensuite, la méthode de **Gauss Seidel**, qui ce n'est pas convenable pour les systèmes de taille importante. Mais la méthode offre une complexité minimale et ne requière pas beaucoup de mémoire due à que la matrice Jacobienne ne change pas à chaque itération.

Dernièrement, la méthode de **Fast Decoupled Load Flow**, cette méthode possède diminué l'effort de la computation significativement.

Une comparaison a été faite entre les différentes méthodes de calcul en regardant la convergence des résultats pour le réseau monophasé avec les données d'un réseau de 33 bus et 66 bus :

Table 3. Balanced distribution networks: DCCase33 and DCCase69.

Methods	Test Cases					
	DCCase33			DCCase69		
	Iter	Time	$\ F(\vec{x})\ _\infty$	Iter	Time	$\ F(\vec{x})\ _\infty$
NR-p-pol [8]	3	0.0123	7.4675×10^{-6}	4	0.0131	5.5875×10^{-9}
NR-p-car	3	0.0067	1.0433×10^{-6}	3	0.0069	8.1777×10^{-6}
NR-p-com	6	0.0058	6.4610×10^{-6}	7	0.0060	4.0138×10^{-6}
NR-c-pol	3	0.0087	1.4291×10^{-9}	3	0.0090	8.5226×10^{-9}
NR-c-car	3	0.0073	1.3954×10^{-9}	3	0.0077	1.9503×10^{-8}
NR-c-com	7	0.0068	5.3792×10^{-6}	10	0.0084	2.7697×10^{-6}
BFS [43]	7	0.0102	1.0454×10^{-6}	7	0.0104	7.7770×10^{-6}

Source : <http://ta.twi.tudelft.nl/nw/users/vuik/papers/Ser17aVW.pdf>

Finalement, nous avons opté pour la méthode de **Newton Raphson** modifié triphasé déséquilibré due à que c'est convenable pour les réseaux maillés et faiblement radiaux sur l'approche choisi, même si la implémentation est plus difficile et il faut prendre en compte les le modèle de charges choisi, le couplage des lignes et d'autres contraintes à prendre en compte.

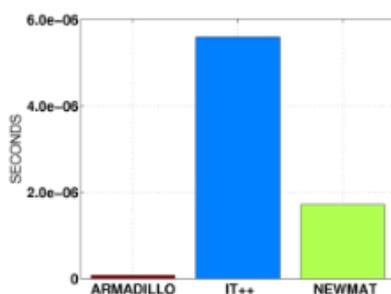
4.2. Décision du langage

Tout d'abord, l'algorithme était avec un développé sur le logiciel MATLAB grâce à son facilité d'implémentation, fonctions faciles à prendre en main, rapidité à coder et à tester. Ce langage induit une vitesse de calcul importante, et nous nous sommes rendu compte que l'analyse en temps réel allait occuper plus d'espace sur la mémoire. De ce fait, nous avons décidé d'utiliser un langage plus puissant avec une haute puissance de calcul et aussi le code source du LYNX est en C et C++, c'est aussi donc plus logique, et en tant qu'industriels, le Matlab n'est pas optimal, lourd, lent, et couteux.

La décision a été prise de choisir C++ avec la librairie Armadillo, qui fournit les fonctions nécessaires à l'algorithme. La solution prend les paramètres de la base de données en temps-réel et en SQL. Dans cet algorithme, la méthode numérique itérative va tourner jusqu'à obtenir la tolérance prédéfinie.

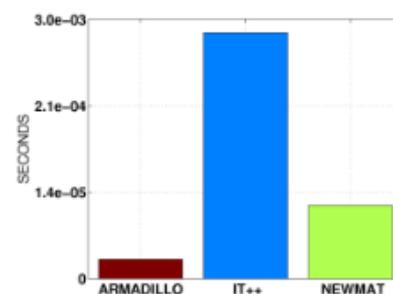
Ici, cela est un rapport de puissance et vitesse d'Armadillo comparé aux autres librairies de calcul matriciel sur C++ comme IT++ ou Newmat.

Rendement pour une expression multi-opération



Matrix size: 4x4

Approximate speed-up relative to:
IT++: 77 times
Newmat: 23 times



Matrix size: 100x100
(y-axis is non-linear)

Approximate speed-up relative to:
IT++: 1086 times (!)
Newmat: 5 times

Source : <http://arma.sourceforge.net/speed.html>

4.3. Programmation

Après avoir commencé à développer, le programme a été divisé en plusieurs parties. Chaque partie correspond à une fonction de l'algorithme.

Une structure organisée et claire du code permet d'optimiser l'exécution de l'algorithme, et de faciliter la compréhension et la maintenance de ce code par l'utilisateur. Chaque variable et fonction doit être définie correctement. Commenter le code est également indispensable

4.3.1. Environnement technique

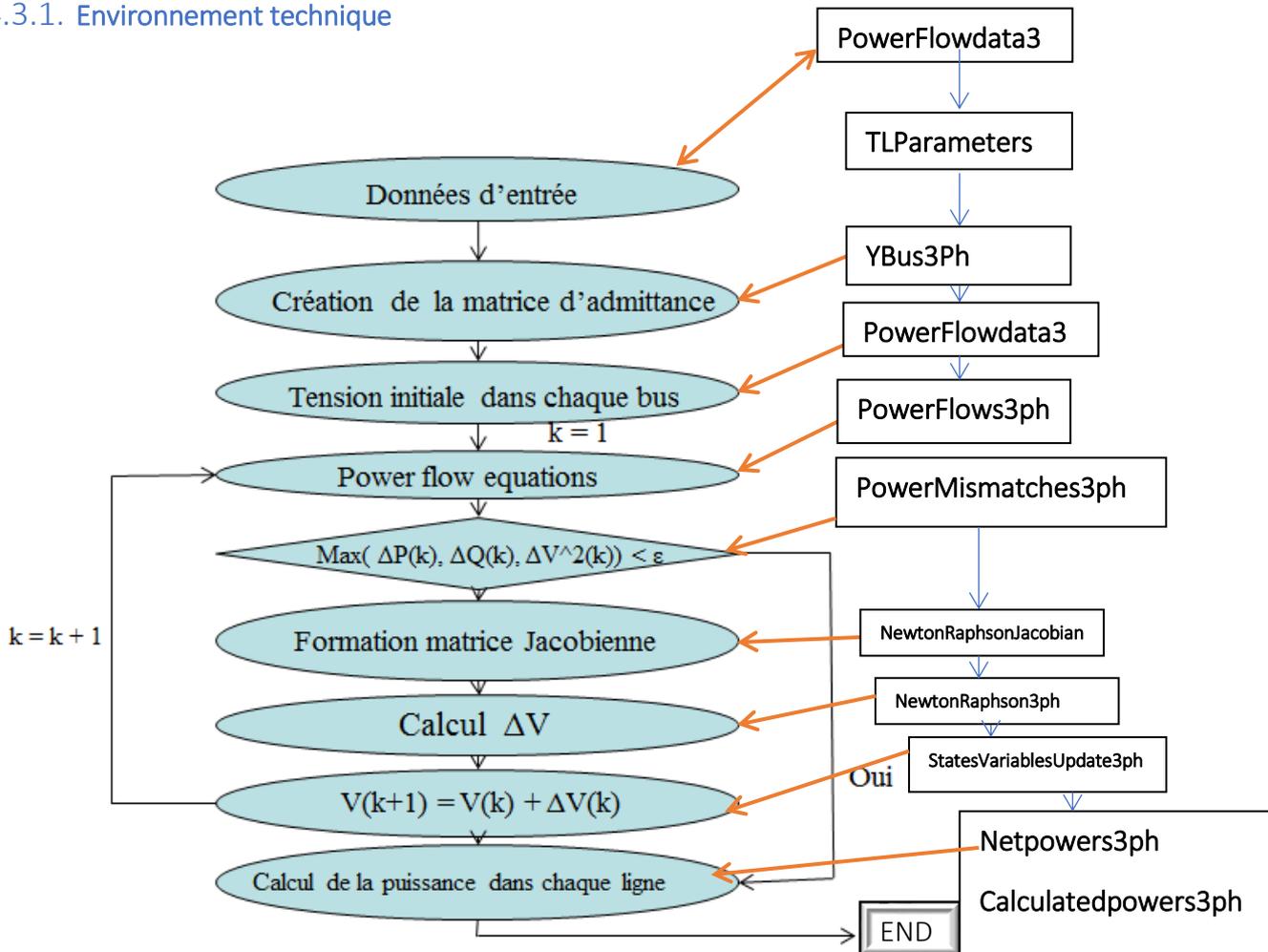


Fig. 1 : Schéma bloc du Newton-Raphson avec Power « Mismatch »

Principes du NR

Il faut avant déterminer le type de bus :

Bus Type	Number of Buses	Known	Unknown
slack node or swing bus	1	$ V_i , \delta_i$	P_i, Q_i
generator node or PV bus	N_g	$P_i, V_i $	Q_i, δ_i
load node or PQ bus	$N - N_g - 1$	P_i, Q_i	$ V_i , \delta_i$

Source : <http://ta.twi.tudelft.nl/nw/users/vuik/papers/Ser17aVW.pdf>

Avant, il faudra connaître le principe :

$$I = YV \Leftrightarrow \begin{bmatrix} I_1^{abc} \\ \vdots \\ I_N^{abc} \end{bmatrix} = \begin{bmatrix} Y_{11}^{abc} & \dots & Y_{1N}^{abc} \\ \vdots & \ddots & \vdots \\ Y_{N1}^{abc} & \dots & Y_{NN}^{abc} \end{bmatrix} \begin{bmatrix} V_1^{abc} \\ \vdots \\ V_N^{abc} \end{bmatrix}$$

Source : <http://ta.twi.tudelft.nl/nw/users/vuik/papers/Ser17aVW.pdf>

Les équations de flux de puissance :

$$\begin{bmatrix} S_k^{abc} \\ S_m^{abc} \end{bmatrix} = \begin{bmatrix} P_k^{abc} + jQ_k^{abc} \\ P_m^{abc} + jQ_m^{abc} \end{bmatrix} = \begin{bmatrix} E_k^{abc} I_k^{abc*} \\ E_m^{abc} I_m^{abc*} \end{bmatrix}$$

Source: Chap. 6.2 FACTS Modelling and simulation in power networks – E. Acha -2004

Pour construire l'algorithme, la définition de stages a été :

- Récupération des données d'entrées :
 - Appel aux paramètres triphasés (**PowerFlowdata3ph**)
 - Prendre en compte la séquence zéro et homopolaire
- Création de la matrice :
 - Positionnement des paramètres sur la matrice d'impédance et admittance
 - Formation de la matrice d'impédance et d'admittance sans la contribution de lignes et shunt (**TLParameters**)
- Formation de la matrice d'impédance avec la contribution de lignes et shunt, formation de la matrice G (réel) et B (imag.) (**YBus3Ph**).
- Calcul de puissances active et réactive net pour les bus de générateurs (slack) et de charge(PQ)(**NetPowers3ph**)

- « Mismatch » ou inadéquation (ΔS) de puissance et enlèvement de puissance active et réactive sur les bus de générateurs et de puissance réactive sur les bus PV (puissance et tension connues)(**PowerMismatches3ph**).
- Formation du Jacobienne et enlèvement de certaines composants sur la matrice pour les bus PV et PQ (**NewtonRaphsonJacobian**)
- Calcul de puissance complexe envoyé, reçoit et perdu (**PowerFlows3ph**).
- Mis à jour de tension et angle de déphasage sur la méthode itérative (**StatesVariablesUpdate3ph**)
- Puissance complexes nettes(**NetPowers3ph**)
- Méthode itérative pour actualiser les valeurs de la tension (amplitude et déphasage) (**NewtonRaphson3ph**)

4.3.2. Fonctions

TLParameters

Inputs

Paramètres des lignes

* Les paramètres de séquence zéro ne sont pas explicitement requis donc ils sont identique à ceux-là de séquence positive

** Les paramètres de séquence zéro ont été pris comme si la valeur était trois fois les valeurs de la séquence positive.

Outputs

TLImpedInv : matrice inverse d'impédance des lignes transmission

TLAdmit : matrice d'admittance des lignes transmission

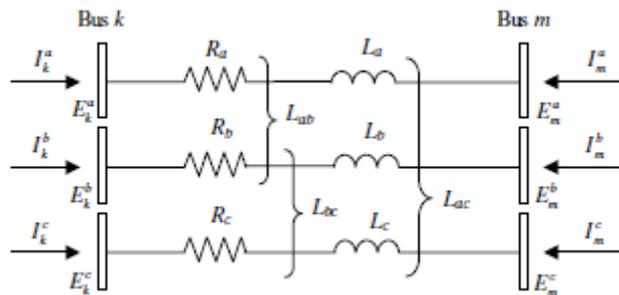
Explication technique et algorithmique

-Formation du cube de la matrice d'impédance et le cube de la matrice d'admittance et dedans chaque matrice va avoir une dimension de 3 x 3 (couplage de lignes de transmission sur chaque bus).

-calcul de l'impédance-self et l'impédance mutuelle. La quatrième phase (mis à la terre) ne va pas être pris en compte donc le calcul de la réduction de Kron a été déjà fait car c'est déjà implémenté sur le calcul.

-Avec les paramètres donnés des lignes, il faut prendre en compte les valeurs de la séquence positive et de la séquence zéro calculé avec la matrice de Fortescue pour équilibrer les phases.

-Quand les valeurs sont positionnés sur la diagonal, nous mettrons Y_{self} et Z_{self} et quand les valeurs sont dehors la diagonale nous mettrons Z_{mutual} et Y_{mutual} . Une fois sortie des boucles, nous invertirons chaque matrice de 3×3 dans $TL_{ImpedInv}$.



Source: Chap. 6.2 FACTS Modelling and simulation in power networks – E. Acha -2004

YBus3ph

Outputs

YR : B

YI : G

Explication technique et algorithmique

Tout d'abord, nous allons entrer la matrice vide avec une dimension de $3n_{bb} \times 3n_{bb}$ qui sera pleine de zéros.

- Accès aux matrices binaires tl_{send} et tl_{rec} dans lequel ils indiquent si le bus envoi ou reçoit de la puissance.

- Selon si c'est en ce qui concerne le nœud i ou nœud j . il va changer la formule de sorte qu'il aura une incidence sur le calcul, alors nous devons garder à l'esprit la matrice d'admittance ou la matrice d'impédance, tous seront entrés dans la matrice YY sur son emplacement approprié.

-Prise en compte la contribution des éléments shunt qui peut être positionné dans le bus ou entre les lignes.

-Ajoutée plus tard à la matrice YY ou à la matrice Y_{bus} la contribution des shunts.

-Modélisation de YR (YY en coordonnées réelles) et YI (YY en coordonnées imaginaires) aussi appelés G et B

$$\begin{aligned}
 Y_{ij} &= \sum_k \frac{-1}{r_{ijk} + jx_{ijk}} \\
 Y_{ji} &= Y_{ij} \\
 Y_{ii} &= -\sum_{j \neq i} Y_{ij} \\
 Y_{jj} &= -\sum_{i \neq j} Y_{ji}
 \end{aligned}$$

Source: Chap. 6.2 FACTS Modelling and simulation in power networks – E. Acha -2004

NewtonRaphson3ph

Inputs

D : vecteur avec les valeurs de $\Delta\theta$ et $\frac{\Delta V}{V}$

Flag : alerte pour faire l'algorithme s'arrêter

Outputs

VM : nouvelle amplitude de la tension

It : nombre d'itération

VA : nouvel angle de la tension

Explication technique et algorithmique

- Obtention des puissances nettes
- Initialiser la variable *it* pour stocker le nombre d'itérations qu'il fait
- Obtention des puissances calculées à travers de la fonction(CalculatedPewer3ph) et il effectuera le « power mismatch » jusqu'à ce que l'indicateur(flag) est égal à 1.
- Cela signifie que l'ensemble du processus cessera d'itérer et aurait atteint un point dans lequel toutes les puissances et les « mismatches » ont été stockées.
- Calcul de la matrice Jacobienne
- Calcul du vecteur D à partir de l'inverse du Jacobienne et le vecteur DPQ (ΔS)
- Mis à jour les valeurs des tensions (amplitude et angle) jusqu'à ce que nous obtenons la tolérance désirée en même temps, que la méthode effectuée les itérations.

$$\begin{bmatrix} \Delta P_\ell^\rho \\ \Delta Q_\ell^\rho \end{bmatrix}^{(i)} = \begin{bmatrix} \frac{\partial P_\ell^\rho}{\partial \theta_j^\rho} & \frac{\partial P_\ell^\rho}{\partial V_j^\rho} V_j^\rho \\ \frac{\partial Q_\ell^\rho}{\partial \theta_j^\rho} & \frac{\partial Q_\ell^\rho}{\partial V_j^\rho} V_j^\rho \end{bmatrix}^{(i)} \begin{bmatrix} \Delta \theta_j^\rho \\ \frac{\Delta V_j^\rho}{V_j^\rho} \end{bmatrix}^{(i)}$$

Source: Chap. 6.2 FACTS Modelling and simulation in power networks – E. Acha -2004

NetPowers3ph

Outputs

PNET : Puissance active nette

QNET : Puissance réactive nette

Explication technique et algorithmique

Premièrement, deux vecteurs remplis de zéros avec une dimension de $1 \times 3 \times \text{nbb}$ vont être faire donc dedans va être stockée la puissance active et réactive nette. Après, nous allons définir une boucle avec une dimension ngn (nombre de bus de générateurs) et une autre boucle avec une dimension égale a trois donc nous faisons une analyse pour chaque ligne. Grace à ça, nous calculerons la puissance active et réactive nette dans les bus de générateurs. Puis, nous ferons le même procès pour les bus de charges.

$$\begin{aligned} \Delta P_\ell^\rho &= [\Delta P_k^a \ \Delta P_k^b \ \Delta P_k^c \ \Delta P_m^a \ \Delta P_m^b \ \Delta P_m^c]^t, \\ \Delta Q_\ell^\rho &= [\Delta Q_k^a \ \Delta Q_k^b \ \Delta Q_k^c \ \Delta Q_m^a \ \Delta Q_m^b \ \Delta Q_m^c]^t, \end{aligned}$$

Source: Chap. 6.2 FACTS Modelling and simulation in power networks – E. Acha -2004

Calculatedpowers3ph & PowerFlows3ph

Outputs

PCAL: puissance active calculé

QCAL: puissance réactive calculé

Explication technique et algorithmique

-Calcul de la tension amont et aval pour les utiliser sur les équations de flux de puissance grâce à la puissance active et réactive nette, au même temps nous voulons signaler si la ligne est en train d'envoyer ou recevoir.

-Fixation des conditions calculer PQsend (colonne fixée) et PQrec (colonne fixée)

-Avec cette méthode, nous positionnons les valeurs de la puissance active et réactive pour tous les lignes quand ils sont soit en train d'envoyer ou recevoir.

PQsend et PQrec est basé sur la somme de:

$$\begin{aligned}
 P_k^\rho &= V_k^\rho \left\{ \sum_{i=k,m} \sum_{j=a,b,c} V_i^j [G_{ki}^{\rho j} \cos(\theta_k^\rho - \theta_i^j) + B_{ki}^{\rho j} \sin(\theta_k^\rho - \theta_i^j)] \right\}, \\
 Q_k^\rho &= V_k^\rho \left\{ \sum_{i=k,m} \sum_{j=a,b,c} V_i^j [G_{ki}^{\rho j} \sin(\theta_k^\rho - \theta_i^j) - B_{ki}^{\rho j} \cos(\theta_k^\rho - \theta_i^j)] \right\}, \\
 P_m^\rho &= V_m^\rho \left\{ \sum_{i=m,k} \sum_{j=a,b,c} V_i^j [G_{mi}^{\rho j} \cos(\theta_m^\rho - \theta_i^j) + B_{mi}^{\rho j} \sin(\theta_m^\rho - \theta_i^j)] \right\}, \\
 Q_m^\rho &= V_m^\rho \left\{ \sum_{i=m,k} \sum_{j=a,b,c} V_i^j [G_{mi}^{\rho j} \sin(\theta_m^\rho - \theta_i^j) - B_{mi}^{\rho j} \cos(\theta_m^\rho - \theta_i^j)] \right\}.
 \end{aligned}$$

Source: Chap. 6.2 FACTS Modelling and simulation in power networks – E. Acha -2004

PowerMismatches3ph

Outputs

DPQ: Vecteur du « mismatch » de puissances active et réactive vecteur qu'après va être actualisé pour regarder si la tolérance est atteinte avec la soustraction de ΔP et ΔQ .

Flag: indicateur pour arrêter d'itérer

Explication technique et algorithmique

Premièrement, le vecteur DPQ rempli de zéros avec une dimension $1 \times n_{max}$ va être formé. Une fois que c'était fait, la computation de DP (vecteur ΔP avec la valeur de puissance active de toutes les lignes) et idem pour DQ (ΔQ).

Notre propos ici est faire le « mismatch » de puissance, enlever la valeur de puissance active et réactive dans les bus de générateurs (slack bus) et enlever la puissance réactive sur le PV bus (puissance active et voltage connues). Après, les valeurs vont être introduit pour la prochaine itération (réarranger les entrées du « mismatch »).

Pour conclure, la vérification va être faite pour savoir si le vecteur DPQ est plus petit que la tolérance pour arrêter le procès itérative avec le flag.

$$\begin{aligned}
 F_i(\vec{x}) &= \Delta S_i(\vec{x}) = S_i^{sp} - S_i(\vec{x}) \\
 &= S_i^{sp} - V_i \sum_{k=1}^N Y_{ik}^* V_k^* \quad \forall i \in 1 \dots N
 \end{aligned}$$

Source: Fig 9: On a comparison of Newton Raphson solvers for power flow problems, B. Sereeter, C. Vuik, and C. Witteveen

NewtonRaphsonJacobian

Outputs

Jac: matrice Jacobienne pour résoudre la méthode numérique

Explication technique et algorithmique

La fonction va former la matrice Jacobienne rempli de zéros avec des dimensions $n_{max} \times n_{max}$, cette dimension est à cause du couplage entre lignes. Pour remplir la matrice, nous voulons implémenter deux boucles avec une dimension n_{bb} . Après nous voulons regarder la position de la valeur (si c'est sur la diagonale ou pas), aussi pour les uns qui sont sur la diagonale, si cette matrice est positionnée sur la diagonale ou pas.

C'est façon à coder est due à que il faut que la méthode soit dominant diagonalement. C'est avec ça que nous pourrions regarder si la méthode converge ou pas. Donc, la dimension totale du Jacobienne est $6 \times n_{bb} \times 6 \times n_{bb}$.

Matrice Jacobienne

$$\begin{bmatrix} \frac{\partial P_\ell^p}{\partial \theta_j^p} & \frac{\partial P_\ell^p}{\partial V_j^p} V_j^p \\ \frac{\partial Q_\ell^p}{\partial \theta_j^p} & \frac{\partial Q_\ell^p}{\partial V_j^p} V_j^p \end{bmatrix}^{(i)}$$

Source: Chap. 6.2 FACTS Modelling and simulation in power networks – E. Acha -2004

Chaque composant matriciel est défini par :

$$\frac{\partial \mathbf{P}_\ell^p}{\partial \theta_j^p} = \begin{bmatrix} \frac{\partial P_\ell^a}{\partial \theta_j^a} & \frac{\partial P_\ell^a}{\partial \theta_j^b} & \frac{\partial P_\ell^a}{\partial \theta_j^c} \\ \frac{\partial P_\ell^b}{\partial \theta_j^a} & \frac{\partial P_\ell^b}{\partial \theta_j^b} & \frac{\partial P_\ell^b}{\partial \theta_j^c} \\ \frac{\partial P_\ell^c}{\partial \theta_j^a} & \frac{\partial P_\ell^c}{\partial \theta_j^b} & \frac{\partial P_\ell^c}{\partial \theta_j^c} \end{bmatrix}, \quad \frac{\partial \mathbf{Q}_\ell^p}{\partial \theta_j^p} = \begin{bmatrix} \frac{\partial Q_\ell^a}{\partial \theta_j^a} & \frac{\partial Q_\ell^a}{\partial \theta_j^b} & \frac{\partial Q_\ell^a}{\partial \theta_j^c} \\ \frac{\partial Q_\ell^b}{\partial \theta_j^a} & \frac{\partial Q_\ell^b}{\partial \theta_j^b} & \frac{\partial Q_\ell^b}{\partial \theta_j^c} \\ \frac{\partial Q_\ell^c}{\partial \theta_j^a} & \frac{\partial Q_\ell^c}{\partial \theta_j^b} & \frac{\partial Q_\ell^c}{\partial \theta_j^c} \end{bmatrix}, \quad \frac{\partial \mathbf{P}_\ell^p}{\partial \mathbf{V}_j^p} = \begin{bmatrix} \frac{\partial P_\ell^a}{\partial V_j^a} & \frac{\partial P_\ell^a}{\partial V_j^b} & \frac{\partial P_\ell^a}{\partial V_j^c} \\ \frac{\partial P_\ell^b}{\partial V_j^a} & \frac{\partial P_\ell^b}{\partial V_j^b} & \frac{\partial P_\ell^b}{\partial V_j^c} \\ \frac{\partial P_\ell^c}{\partial V_j^a} & \frac{\partial P_\ell^c}{\partial V_j^b} & \frac{\partial P_\ell^c}{\partial V_j^c} \end{bmatrix}, \quad \frac{\partial \mathbf{Q}_\ell^p}{\partial \mathbf{V}_j^p} = \begin{bmatrix} \frac{\partial Q_\ell^a}{\partial V_j^a} & \frac{\partial Q_\ell^a}{\partial V_j^b} & \frac{\partial Q_\ell^a}{\partial V_j^c} \\ \frac{\partial Q_\ell^b}{\partial V_j^a} & \frac{\partial Q_\ell^b}{\partial V_j^b} & \frac{\partial Q_\ell^b}{\partial V_j^c} \\ \frac{\partial Q_\ell^c}{\partial V_j^a} & \frac{\partial Q_\ell^c}{\partial V_j^b} & \frac{\partial Q_\ell^c}{\partial V_j^c} \end{bmatrix}$$

Source: Chap. 6.2 FACTS Modelling and simulation in power networks – E. Acha -2004

4.4. Données d'entrées

Nous pouvons distinguer entre deux types de données :

- TR : temps réel ou dynamique
- SQL : données statiques

(1) Caractéristiques électriques statiques et données en temps-réel :

Object	Data name	Unit	Type	Edition	Priority
Busbars					
Nbb	Number of busbars	-	Static	Configuration mode	Mandatory
bustype	Busbar type		Static	Configuration mode	Mandatory
VM	Voltage magnitude				
VA	Voltage angle				
	P				Mandatory
	Q				Mandatory
	Load type	-	Static	Configuration mode	Mandatory
Lines					
Ntl	Number of transmission lines				
Tlsend	Transmission line sending				
Tlrec	Transmission line receiving				
Tlresisp	Direct resistance	ohm/km	Static	Configuration mode	Mandatory
Tlreacp	Direct reactance	ohm/km	Static	Configuration mode	Mandatory
Tlsuscepp	Direct susceptance	µS/km	Static	Configuration mode	Mandatory
Tlcondp	Direct conductance				
Tlresisz	Homopolar resistance	ohm/km	Static	Configuration mode	Mandatory
Tlreacz	Homopolar reactance	ohm/km	Static	Configuration mode	Mandatory

Tlcondz	Homopolar conductance	nF/km	Static	Configuration mode	Mandatory
Tlsuscepz	Homopolar susceptance				
Nsh	Number of shunts		Static		
Shresis	Shunt resistance		Static		
Shreac	Shunt reactance		Static		
Transformers					
	Voltage instruction	kV	Static	Configuration mode	Mandatory
	Regulation level for fixed tap	-	Static	Configuration mode	Mandatory
	Real-time tap position	Int	Dynamic	Real-time acquisition Manual input/command	Mandatory
Generators					
Ngn	Number of generators		Static		
Genbus	Bus of generator		Static		
Pgen	Generated active power		Dynamic		
Qgen	Generated reactive power		Dynamic		
Qmax	Maximum reactive power		Static		
Qmin	Minimum reactive power		Static		
Groups					
	Active power injected	MW	Dynamic	Real-time acquisition Manual input/command	Mandatory
	Reactive power injected or consumed	MAVR	Dynamic	Real-time acquisition Manual input/command	Mandatory
Loads					
Nld	Number of loads		Static		
Loadbus	Load bus		Static		
Pload	Absorbed active power		Dynamic		
Qload	Absorbed reactive power		Dynamic		
Method					
Nmax	Maximum number of iterations		Static		
Tol	Tolerance		Static		
Ntl	Number of line combinations		Static		

Configuration du réseau / topologie :

- Topologie statique : lien entre les objets (branches/nœuds)
- Dynamique : Etat des commutateurs : → flux de puissance/courant, l'alimentation de ton réseau : configuration TR du réseau.

5. Implémentation sur le SCADA DMS Lynx

5.1. Présentation du SCADA Lynx

SCADA DMS LYNX est un produit incluant des avancées spécifiques des fonctionnalités dédiées à l'énergie et secteurs des services publics. Il est déjà utilisé de 150 sites à travers le monde et est reconnue comme la solution parfaitement adaptée permettant accès à des services en temps réel et informations pour améliorer la surveillance et contrôle. Son noyau générique couvre les besoins d'un certain nombre de processus industriels secteur, ente autres distribution d'électricité et réseaux de transport, distribution de gaz et réseaux de transport, ainsi que d'eau potable, usines de produits, traitement des eaux usées, etc.

Le SCADA Lynx dédié à la supervision et au contrôle des réseaux de distribution. Ici, ci-dessous, nous vont vous faire une brève présentation avec la figure ci-joint :

Le premier menu accessible à l'opérateur est le suivant :

Le module « **Dynspv** » de LYNX assure l'interface principale aux opérateurs pour la conduite de dispatching et ce pour des techniques multiples. il permet de ce fait la gestion technique centralisée (GTC)

Modes Etude et Simulation :
 Simulent l'environnement habituel des opérateurs de conduite:
 · interface homme machine identique.
 · configuration du réseau identique.
 · fonctionnalités du système identiques.

L'éditeur graphique « **Dyncad** » est un logiciel complet de dessin assisté par ordinateur. Il permet de construire et adapter les images visualisées par Dynspv. et de définir leurs paramètres d'animation.

Menuelec : pour la mise à jour/ création des paramètres électriques
Menusysteme : pour la mise à jour /création des paramètres système
 Outils de gestion de la Data Base et du paramétrage.

Ici, ci-dessous voici un bref état des lieux de l'architecture de la base de données :

Description sommaire du modèle de la structure des objets dans Lynx

Il existe deux types d'objets dans Lynx : les **objets métier** et les **objets de base**. Les objets métiers sont spécifiques au domaine concerne : ici ce sont les ouvrages (postes), les disjoncteurs, les postes HTA/BT, les segments de lignes, les transformateurs, etc. Les objets de base sont commun à toutes les applications, ils peuvent être digital, analogique, ou complexes (objets calculés).

- Chaque objet est associé à un **comportement**. Les comportements permettent de traiter de manière homogène les objets de même type. On peut créer autant de comportement qu'il y a d'objet.
- Chaque objet est également associé à une **classe** (télésignalisation, télémesure, télécommande, objet calculé digitale ou analogique, ouvrage, etc.) : qui définit ses propriétés.
- Les objets peuvent appartenir à un **champ** (arrivée transfo, barre, groupe, départ transfo, sectionneur, poste, dérivation, segment, etc.) : c'est un groupement d'objets ayant une fonction ou un lien commun. Chaque objet métier est généralement compris dans un champ du même nom.
- Les objets appartiennent à un **domaine** (électrique, hydraulique, etc.) et à un **niveau** (niveau de tension, de pression, etc.).
- Un objet peut être associé à une **image**, créée sur Dyncad, afin d'être visualisée sur Lynx par l'opérateur.

Dans le Lynx, le serveur « Loadflow » existe et permet de réaliser des calculs de répartition de charge sur les réseaux de transport ou de distribution électriques, jusqu'à la basse tension.

5.2. Présentation du serveur « Loadflow » du LYNX

Le loadflow monophasé propose un calcul de flux de puissance qui calcule l'amplitude et le déphasage de la tension dans chaque nœud, et les différents transits (courant, puissance active et réactive) dans chaque branche des réseaux, y compris les lignes et le transformateur. Ce calcul utilise la topologie en temps réel du réseau et les productions ainsi que les données de consommation.

En mode simulation, le loadflow permet aux opérateurs de simuler le comportement du réseau, à partir de diverses situations passées ou futures du réseau, par exemple avec un « empreinte » du réseau (cas enregistré de la supervision à partir de temps réel ou de l'état archivé).

5.3. Implémentation du Loadflow triphasé déséquilibré dans LYNX

Les différentes étapes pour l'implémentation sur SCADA Lynx sont :

- **Découverte et prise en main du LYNX, de la structuration des bases de données, des interfaces. Lecture des documentations, manipulation du logiciel.**
- **Prise en main des langages TCL, C++, et SQL.**

- Rédaction des spécifications techniques et fonctionnelles sur le modèle des spécifications existantes du SCADA Lynx
- Mise à jour de la base de données : modèle triphasé des lignes, transformateurs, etc. Et ajout des paramètres nécessaires
- L'intégration de l'algorithme, connexion avec la base de données TR (temps-réel) et statique (SQL) du LYNX.
- Tests et simulations
- Implémentation de l'interface graphique pour l'utilisateur (paramétrage, visualisation des résultats, etc.)

Le déroulement des spécifications techniques et fonctionnelles est fragmenté en deux phases :

Premièrement, l'étude de la solution en elle-même, comment l'intégrer en analysant les besoins du marché et faire un produit « commercialisable ».

Deuxièmement, l'adaptabilité de l'algorithme au monde industriel d'une façon optimale prenant en compte contraintes comme le temps, les licences et le langage de la machine.

5.3.1 Machine de test et application LYNX disponible

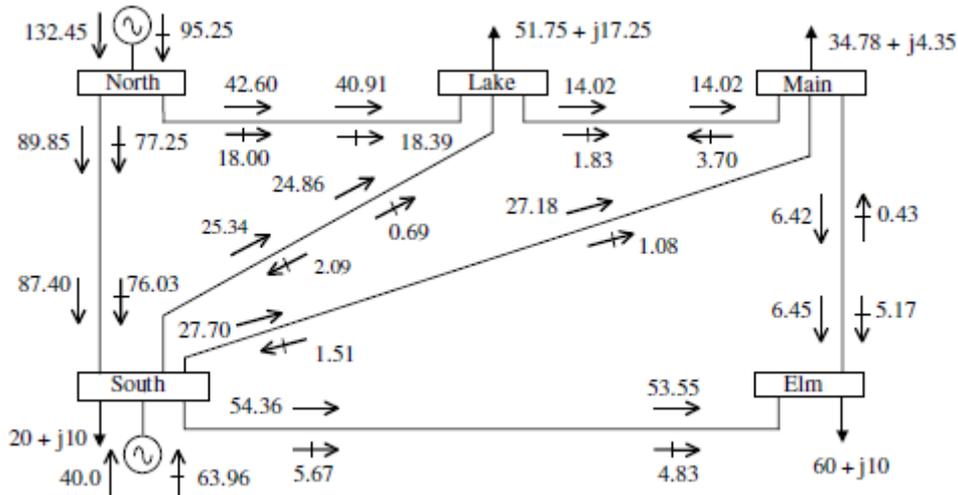
Pour faire les test et simulations de l'algorithme, j'ai utilisé un logiciel que s'appelle MobaXterm. MobaXterm est une boîte à outils ultime pour l'informatique à distance. Cela m'a permis d'utiliser sur mon PC Windows les commandes habituellement disponibles sous Unix/Linux. Il inclut un serveur X et un terminal qui m'a permis d'avoir sous la main un moyen simple et efficace de me connecter et travailler sur des serveurs Unix/Linux à distance. Avec MobaXterm, je peux me connecter à la plateforme LYNX V.7 qui supporte l'intégralité des fonctionnalités LYNX, sur lequel je suis en train d'effectuer mes développements et tests d'intégration.

Avec les données existantes nous pouvons faire la simulation en triphasé équilibré et déséquilibré. Aussi, dans un premier moment, j'ai utilisé Linux Red Hat pour regarder la base de données et la machine. Ici, ci-dessous nous pouvons regarder l'architecture de la base de données :

5.3.2. Implémentation et tests avec des données externes au LYNX

Dans un premier temps, pour faire la simulation nous j'ai utilisé les données trouvées sur un bouquin (FACTS Modelling and simulation of Power Networks - E.Acha) avec un réseau de

distribution maillé avec cinq jeux de barres, sept lignes de transmission, deux bus de générateurs, quatre bus de charges.



Source: Chap. 6.2 FACTS Modelling and simulation in power networks – E. Acha -2004

J’ai réalisé la simulation avec l’algorithme implémenté dans le LYNX mais utilisant ces données fictives externes au LYNX pour le réseau de distribution déséquilibré et prochainement sera également réalisé la simulation pour de ce réseau déséquilibré avec une altération dans les puissances de $\pm 15\%$ par rapport à la puissance complexe de base.

5.3.3. Modification de la base de données existante

Avec les données existantes j’ai pu réaliser des simulations en triphasé équilibré et déséquilibré.

Aussi, dans un premier temps, j’ai utilisé Linux Red Hat pour analyser la base de données et l’application LYNX disponible.

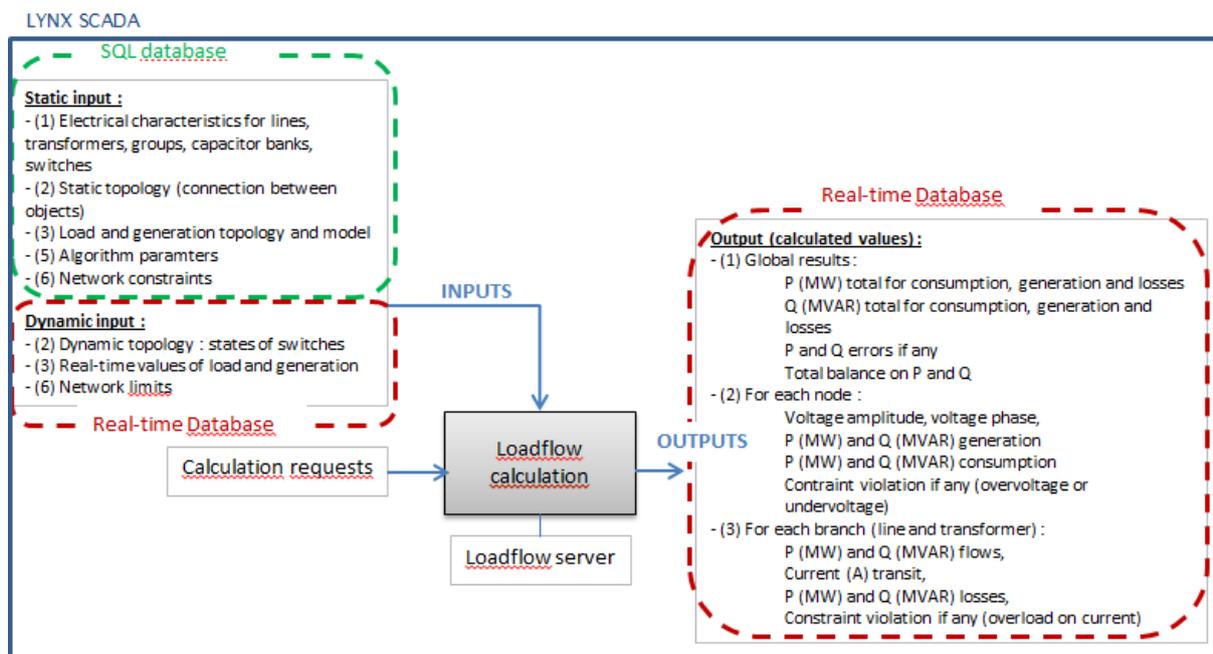
Le Loadflow du LYNX permet uniquement de réaliser le calcul pour les réseaux triphasés équilibrés (ne nécessitant qu’une seule phase). Le modèle de données est également consacré aux réseaux équilibrés et ne supporte les données que d’une seule phase.

Il faut donc aussi mettre à jour les paramètres sur la base de données pour chacune de phases et dimensionner les composants électriques sur le réseau de distribution.

5.3.4. Interfaçage avec les données LYNX

Dans un second temps, je vais connecter l’algorithme avec la base de données LYNX (temps-réel et statique) me permettant de tester les résultats avec des données réel existant sur LYNX. Après, je réaliserai une fonction pour connecter les transformateurs (HV/MV) et faire le « mismatch » de courant pour améliorer la puissance de calcul et la convergence.

Diagramme de calcul global pour la intégration de la fonction sur différentes machines



Ici, sur la base de données nous pouvons distinguer entre deux types de données :

Input statique : collectées à partir de la base de données SQL de Lynx'lynxdb', puis collectées à partir du programme loadflow lors de l'initialisation de la fonction, et stockées dans la base de données en temps réel pour assurer les plus grandes performances du calcul lors de l'utilisation ces données.

Entrée dynamique: collectées à partir de la base de données en temps réel du SCADA Lynx. Lorsqu'un calcul est exécuté, ces données sont collectées instantanément.

5.4. Présentation aux clients

Des clients de l'entreprise gérant un réseau de distribution électrique important en Thaïlande sont venus visiter le site d'ATOS à Grenoble afin de découvrir le produit LYNX.

Leur a été présenté toute la solution LYNX pour surveiller le réseau électrique et rétablir les niveaux de tension donc il peut se mettre une solution avec une production décentralisé.

Le réseau de distribution Thaïlandais étant triphasé déséquilibré, ils étaient très intéressés par mes travaux. J'ai réalisé une présentation PowerPoint pour leur montrer la solution du Loadflow triphasé déséquilibré que j'ai implémenté : comment l'algorithme fonctionne et quels paramètres sont pris en compte. Pendant la présentation, j'ai expliqué la puissance de l'algorithme, les input-output, l'application, le fonctionnement (schéma-bloc), comment obtenir les données, l'intégration sur LYNX et les tests faits.

Cela a permis notamment d'engager des discussions sur le sujet, de parler des difficultés et des opportunités.

6. Conclusions générales sur le projet

6.1. Gestion du projet : retour d'expérience et difficultés rencontrées

Durant mon stage de projet de fin d'études sur Atos Worldgrid, j'ai amélioré mes compétences d'autonomie, patience, passion sur l'étude que je réalise et persévérance.

Ce stage m'a permis de me renseigner sur les différentes méthodes de calculs numériques pour la surveillance des réseaux dont je vérifiais la convergence pour chaque algorithme. Quand j'ai commencé, je ne connaissais pas autant de paramètres par rapport au dimensionnement des réseaux de distribution. Les premiers jours de mon stage sur Atos, ils m'ont montré la machine du LYNX et tous les réseaux de distribution en France aussi j'ai regardé les réseaux de haute et basse tension même si je ne les ai pas utilisés. J'ai appris les différentes fonctions sur LYNX par rapport à la topologie, les alarmes en cas où il y a un défaut, la localisation de défauts et comment les réparer avec une isolation de la zone du défaut. J'ai gagné des notions sur la méthode Agile et comment gérer un projet de cette façon. J'ai appris à coder en C++ avec la librairie Armadillo pour après le connecter sur la base de données en temps-réel qui était en SQL avec un IHM en TCL-TK. J'ai gagné de connaissances par rapport au calcul matriciel comme les matrices sparse et en plus j'apprends à appliquer le calcul matriciel sur le réseau de distribution pour coder d'une façon qui prends moins de espace sur la mémoire avec une nominalisation correctes des variables.

Par rapport au composant interpersonnel, j'ai beaucoup appris, j'ai compris comment est-ce que fonctionne la hiérarchie en entreprise, comment demander de l'aide dans une situation de divergence d'opinion et savoir parler à l'équipe de managers. Une des plus intéressantes choses, qui m'a beaucoup appris, a été la présentation aux clients de mon travail et comment rédiger la présentation pour la rendre plus attractive. Comment être très professionnel et savoir gérer une situation avec des personnes qui ont une différente personnalité et métier. En plus, les spécifications que j'ai rédigé m'ai fait apprendre à me mettre dans le rôle de professeur et savoir exprimer d'une façon très simple concepts si complexes pour après être compris par une personne qui même pas connaît le sujet. J'ai gagné énormément en expérience professionnelle et en compétences techniques.

Unes des raisons pour lesquelles j'ai appris à être plus patiente est à cause d'innombrables problèmes avec la librairie Armadillo. Personnellement, ça était la première fois que j'ai utilisé cette librairie et aussi au même temps la première fois que je utilisai C++ donc dans un premier moment, j'avais beaucoup des erreurs à chaque fois que j'essayai de coder. Les bugs étaient une de mes pires cauchemars, je même rêvais sur ceux-là. En plus, être en totale autonomie fait le procès d'apprentissage plus lent. J'étais habitué à utiliser Matlab que c'est

un langage beaucoup plus intuitif. Mais, je sais que cette opportunité m'ai fait beaucoup grandir en plusieurs aspects et c'est pour ça que je suis très remercié à avoir eu cette opportunité. Donc maintenant, je suis une personne plus adaptif car même si je ne connais pas un logiciel ou quelque connaissance, je vais l'apprendre.

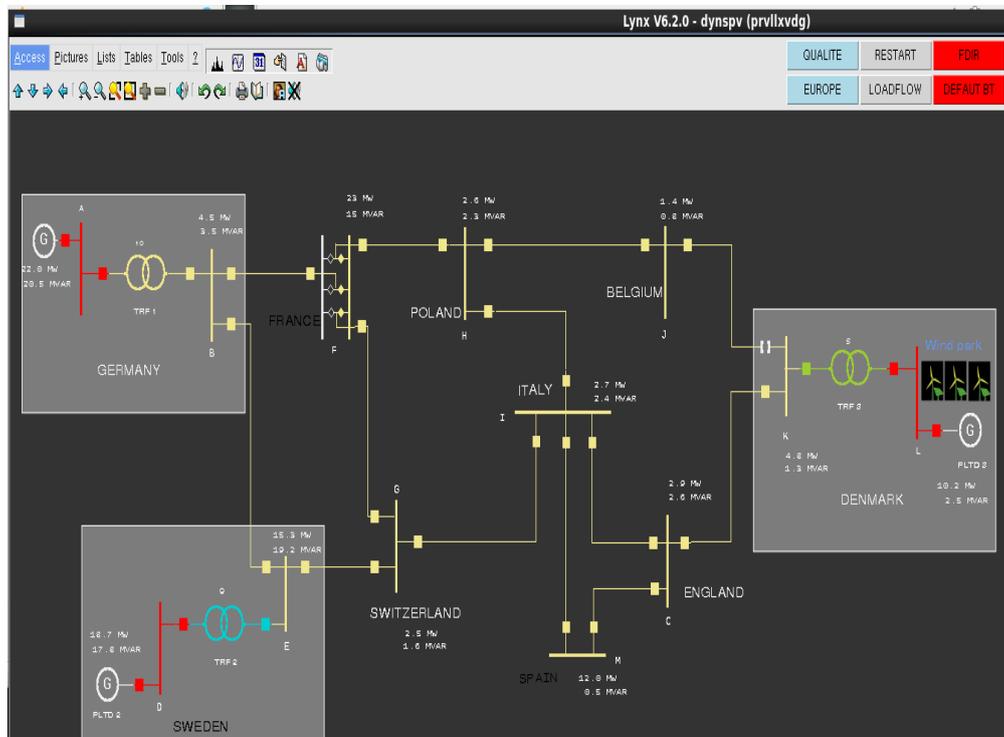
6.2. Conclusion générale

Une fois l'étude bibliographique faite sur les méthodes numériques pour faire le calcul de répartition de charges et l'apprentissage de l'utilisation du SCADA LYNX, l'algorithme de loadflow triphasé déséquilibré a été développé grâce à la bibliothèque Armadillo en langage C++. Dans un premier temps ce fut un défi pour moi de surpasser la barrière de la langue et de la complexité du sujet proposé. Les résultats obtenus sur NR triphasé avec les données du réseau équilibré sont proches des résultats obtenu avec un calcul de répartition de charges monophasé. Même si j'ai rencontré beaucoup de difficultés par rapport au langage, aujourd'hui je peux dire que je suis une personne plus adaptative et plus autonome.

7. Bibliographie

- Acha, E. (2004). *FACTS Modelling and Simulation in Power Networks*. Wiley.
- Ali Sameni, A. B. (2012). *A Modified Newton-Raphson Method for Unbalanced Distribution Systems*.
- B. Sereeter, C. V. (2017). *On a comparison of Newton-Raphson solvers for power flow*.
- B. Stott, O. A. (1974). *Fast Decoupled Load Flow*.
- Baljinnyam Sereeter *, K. V. (2017). *Newton Power Flow Methods for Unbalanced Three Phase Distribution Networks*.
- Chen, Y. (2015). *YBUS Admittance Matrix Formulation*.
- Debora Rosana Ribeiro Penido, S. M. (2008). *Three-Phase Power Flow Based on Four-Conductor*.
- Erhan Demirok, S. B. (2012). *Three-Phase Unbalanced Load Flow Tool for Distribution Networks*.
- <https://atos.net/fr/a-propos-d-atos>. (2018).
- Robert H. Miller, J. H. (1993). *Power System Operation*. McGrawHill.
- Seyed Abbas Taher *, M. H. (2014). *Optimal reconfiguration and DG allocation*.
- Vijayvargia, A., Jain, S., Meena, S., Gupta, V., & Lalwani, M. (2016). *Comparison between Different Load Flow Methodologies by Analyzing Various Bus Systems*. IJEE.
- Weedy, B. (19739). *Electric Power Systems (3rd Edition)*. Wiley.

8. Annexe du réseau sur LYNX



Impression d'écran du réseau sur LYNX

L'algorithme de Newton Raphson triphasé déséquilibré est appliqué sur le réseau de démonstration de LYNX. L'équipe LYNX a dessiné un réseau basique utilisé pour effectuer les simulations de plusieurs fonctions. L'étude a été effectuée sur les niveaux moyenne tension, afin d'avoir la présence de tous les équipements réglables (transformateurs et groupes de production réglables). Les clients déterminent la personnalisation du logiciel pour leurs propres applications, ceux sont les gestionnaires du réseau de distribution.

Le réseau est formé par ces composants :

- 163 bus
- 98 lignes
- 24 transformateurs
- 10 générateurs
- 12 transformateurs équipés de régulateurs en charge
- 1 banc de capacités et d'inductances
- 5 générateurs contrôlables (puissances réactives)