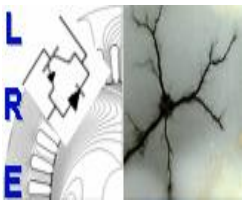
République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique
Département d'Electrotechnique
Laboratoire de Recherche en Electrotechnique



Mémoire de

Master en Electrotechnique

Présenté par
BOUDROUAZ Yacine

Intitulé

Méthode de Newton pour le calcul d'écoulement de puissance optimal

Membres du Jury d'examen

Rapporteur	A.Hellal	Professeur à l'ENP
Président	O. Touhami	Professeur à l'ENP
Examinateurs	R.Ibtiouen	Professeur à l'ENP
	S.Mekhtoub	Professeur à l'ENP

ENP 2015

ملخص

يهدف هذا العمل الى دراسة السريان الأمثل للطاقة الكهربائية بطريقة نيوتن حيث نسعى لتخفيض تكلفة انتاج الكهرباء و أيضا للحد من الخسائر أثناء توزيع الكهرباء على الشبكة. نحسب هذا السريان الأمثل بواسطة الطريقة السابقة الذكر حيث نقارن ونناقش النتائج المتحصل عليها وكذا العوامل المؤثرة سلبا وإيجابا على السير الأمثل للطاقة الكهربائية .

الكلمات المفتاحية : السريان الأمثل للطاقة الكهربائية ، طريقة نيوتن

Résumé

Ce travail traite le problème d'écoulement de puissance optimal en générale et nous discutons le cas de la répartitions économique de puissance. Pour cela, nous résoudrons ce problème avec une méthodes conventionnelle dont la méthode de newton . Enfin , nous représentons les différentes résultats obtenus.

Mots clés : écoulement de puissance optimal, répartitions économique de puissance , méthodes conventionnelles .

Abstract

This work deals with the problem of opf in general and discuss the case of economical dispatch.For this,we solve this problem with one conventional method which Newton method. Finally, we represent the different results obtained.

Key words : opf, economical dispatch, conventional method.

Remerciements

En premier lieu, je remercie le bon Dieu miséricordieux et clément, qui m'a guidés dans la bonne voie des sciences et de la connaissance.

Je tiens à remercier mon promoteur **Pr : A.Hellal** qui m'a fait profiter de son expérience et ses précieux conseils dans ce travail et dans le domaine de la recherche scientifique. Puisse ce travail vous satisfaire et témoigner ma grande reconnaissance et ma profonde estime.

Je suis très sensibles à l'honneur que me fait **Pr : O.Touhami** en présidant ce jury. Je tiens à lui exprimer mes remerciements les plus sincères.

j'exprime ma profonde reconnaissance au **Pr : S.Mekhtoub** et au **Pr : R.Ibtiouen** d'avoir accepté d'être membres du jury et de juger ce modeste travail.

Je tiens aussi à remercier mon frère et ami **Islam Ziouani** pour l'aide qui m'a donner et pour sa patience et sa honnêteté et pour ces qualité humaines.

Je tiens également à remercier tous les enseignants ayant contribué à ma formation et ce depuis le tronc commun jusqu'à la dernière année de graduation.

Table des matières

Abréviations

EPO **E**coulement de **P**uissance **O**ptimal

MPI **M**éthode de **P**oint **I**nterieur

OEP **O**ptimisation par **E**ssaim de **P**articules

REP **R**épartition **E**conomique de **P**uissance.

Dédicaces

Je dédie ce travail à mes amis, mes collègues
Je tiens également à réserver une spéciale dédicace aux êtres les plus tendres à mes yeux et les plus chères à mon cœur, ma mère et mon père.
A ma fiancée, mes sœur et mes frères et à toute ma famille.

Yacine.

Introduction

La gestion d'un réseau de production, distribution ou stockage d'énergie est devenue un enjeu tant économique que technique, devant respecter des contraintes climatiques et économiques. Il apparaît donc la nécessité d'optimiser cette gestion afin de profiter du meilleur fonctionnement du réseau tout en respectant les contraintes imposées.

Le problème de répartition économique, qui est un cas des problème d'écoulement de puissance, dans son traitement à l'état statique et dynamique d'un réseau électrique, occupe dans nos jours une place déterminante dans la stratégie concurrentielle de l'entreprise, face à la libéralisation du secteur d'électricité, donc face à une concurrence acharnée, mais aussi par rapport aux nouvelles restrictions liées à l'environnement qu'il faut respecter.

Dans cette logique, un faible coût de production représente un défi pour les sociétés productrices, vu notamment l e prix des combustibles, et toutes les charges supplémentaires liées à la production , aux quelles on peut ajouter des frais comme celles liées au traitement des déchets nucléaires dans le cas de la production nucléaire.

D'un autre coté, la complexité grandissante des réseaux électriques d'aujourd'hui, vu des tailles avec des centaines de jeux de barres et des milliers de kilomètres de lignes de transmission, ainsi que des structures interconnectées, exige qu'une optimisation de la répartition optimale de puissance active générée constitue une nécessité impérative et un coût minimisé représente un objectif primordial.

Notons qu'une optimisation de cette répartition ne doit pas seulement garantir un faible coût de production mais aussi s'accompagner d'une minimisation des pertes de transport (répartition économique avec pertes), ce qui engendre un problème d'optimisation non linéaire.

Plusieurs méthodes sont proposées pour la résolution du problème d'écoulement de puissance optimal dont des méthodes conventionnelles (méthode de Newton, méthode du gradient, méthode des itérations lambda, programmation linéaire ...etc) et d'autres non conventionnelles (méthodes d'intelligence artificielle) comme les algorithmes génétiques (AG), les algorithmes évolutionnaires, optimisation par essaim de particules (OEP ou en anglais PSO) ..etc

Dans ce mémoire, nous allons appliquer la méthode de newton pour la résolution du problème de la répartition économique de puissance.

Le mémoire est organiser comme suit :

— le premier chapitre intitulé *Formulation et méthodes de résolution du problème de l'EPO* donne des définitions de base sur la notion de l'écoulement de puissance optimale, la formulation du problème d'EPO, le dispatching économique et ces différents aspects.

— le deuxième chapitre intitulé *Application de la méthode de newton à la répartition économique de puissance* explique en détail la méthode de newton appliquée au problème avec simulations et discussions des résultats obtenus.
.

Chapitre 1

Formulation et méthodes de résolution du problème de l'écoulement de puissance optimal

1.1 Formulation du problème d'écoulement de puissance optimal

En général, l'OPF comprend tout problème d'optimisation qui cherche à optimiser le fonctionnement d'un système de réseau électrique (en particulier, la génération et la transmission de l'électricité), sous réserve de contraintes imposées par les lois électriques et les limites techniques sur les variables de contrôle. Ce cadre général englobe divers problèmes d'optimisation pour la planification et le fonctionnement des réseaux électriques. Il agit sur les variables de contrôle disponibles afin d'optimiser un objectif tout en satisfaisant les équations d'écoulement de puissance de réseau, les contraintes physiques et opérationnelles. L'écoulement de puissance optimal a été présenté au début des années 60 [1], et même avant par d'autres auteurs. Depuis ce temps, l'EPO est devenu un outil indispensable pour le développement, et la commande en temps réel des réseaux électriques. L'EPO est en général un problème non linéaire, avec des variables continues et discrètes. La formulation du problème d'EPO peut s'écrire comme suit :

$$minf(x,u) \tag{1.1}$$

Lié à :

$$g(x,u) = 0 \tag{1.2}$$

$$h(x) \leq 0 \tag{1.3}$$

$$U_{min} \leq U \leq U_{max} \tag{1.4}$$

Avec :

- u : variable de contrôle
- x : variable d'état
- L'équation 1.1 représente la fonction objective.
- L'équation 1.2 représente les contraintes d'égalités.
- L'équation 1.3 représente les contraintes d'inégalités.
- L'équation 1.4 représente les limites des variables de contrôle.

1.2 Signification des éléments de problème

1.2.1 Variables d'état

L'état du système est défini par la matrice admittance du réseau et les tensions aux nœuds.

1.2.2 Variables de contrôle

Les variables de contrôle dans l'EPO en générale sont :

- P_g : puissance active générée
- ϕ : transformateur de phase.
- Q_g : puissance réactive générée
- V : amplitude de tension

1.2.3 Contraintes égalités

Les contraintes d'égalité de l'EPO reflètent l'état du réseau électrique ainsi que la tension souhaitée à chaque nœud du système. L'état du réseau électrique est exprimé par les équations de l'écoulement de puissance qu'ils assurent l'équilibre du système.

1.2.4 Contraintes inégalités

On peut rencontrer deux types de contraintes inégalités :

- **Limites supérieures et inférieures**
  Concernent les variables de contrôle car elles sont liées au fonctionnement des générateurs et à la stabilité du réseau.
  $Pg_{min} < Pg < Pg_{max}$
  $Qg_{min} < Qg < Qg_{max}$
- **Contraintes inégalité fonctionnelle**
  On prend par exemple la puissance maximale transitée entre deux nœuds.

1.2.5 Fonction objectif

La fonction objectif de l'EPO peut prendre diverses formes selon l'objectif souhaité. Les objectifs les plus utilisés dans l'EPO sont

- la minimisation du coût de production de la puissance.
- la minimisation des pertes actives.
- la minimisation des pertes réactives
- minimisation de délestage
- maximisation de la limite de capacité de charge
- maximisation du profit
- minimisation des déviations causées par les violations des contraintes ... etc

Parmi les objectifs cités, la minimisation du coût de production de la puissance est l'une des fonctions objectif les plus utilisées pour l'EPO, ce qui a fait l'objet de notre travail dans les sections suivantes.

1.3 Étude de problème de la répartition économique de puissance

Le problème de la répartition économique a eu son début à partir du moment que deux ou plusieurs unités se sont engagés à prendre en charge sur un système de puissance dont la demande total dépassait la génération maximal d'une seule unité. Le problème qui se posait à l'opérateur était exactement comment diviser la génération entre les unités.

La répartition économique alors est fait en temps réel, à cause de la variation du coût et de la puissance demandé pour assigner la génération totale parmi les unités disponibles pour satisfaite la demande. La répartition économique est un processus de calcul selon laquelle la production totale nécessaire est répartie entre les unités de production disponibles de sorte que les contraintes imposées sont remplies et que les besoins en énergie en termes de BTU/h ou $\$/h$ sont minimisés.

1.3.1 Formulation du problème de la REP

Le problème de la répartition économique de puissance peut être formulé de la façon suivante :

$$minf(U = P_g) = min \sum_{i=1}^{N_{gen}} a_i + b_i P_{g_i} + c_i P_{g_i}^2 \tag{1.5}$$

Lié à :

$$\sum_{j \in i}^{N} V_i V_j [G_{ij} cos(\delta_i - \delta j) + B_{ij} sin(\delta_i - \delta j)] - Pg_i + Pd_i = 0 \tag{1.6}$$

$$\sum_{j \in i}^{N} V_i V_j [G_{ij} sin(\delta_i - \delta j) - B_{ij} cos(\delta_i - \delta j)] - Qg_i + Qd_i = 0 \tag{1.7}$$

$$P_{g_{min}} \leq P_g \leq P_{g_{max}} \tag{1.8}$$

tel que :

- N : nombre de nœuds.
- N_{gen} : nombre de nœuds de générations.
- G_{ij} :conductance des lignes.
- B_{ij} : susceptance des lignes.
- $\delta_{i,j}$: déphasage de la tension V_{ij} du nœud i(ou j)
- Pg_i : puissance active générée au nœud i.
- Pd_i : puissance active demandée au nœud i.
- Qg_i : puissance réactive générée au nœud i.
- Qd_i : puissance réactive demandée au nœud i.
- P_g est le variable de contrôle.
- V et θ sont les variables d'état.
- Pg_i est la quantité de la production en MW au générateur i.
- 1.5 représente le coût de production total.
- 1.6 et 1.7 représentent les equations de l'écoulement de puissance.
- 1.8 représente les limites des puissances générées

Note

Généralement, la fonction coût de chaque générateur est de forme quadratique de 2^{eme} ordre.

Le problème de la REP peut être résolu par plusieurs méthodes conventionnelles et d'autres non conventionnelles dites d'intelligence artificielle.

1.3.2 Les méthodes conventionnelles

Les techniques classiques dites aussi mathématiques ou encore conventionnelles appliquées au problème de la répartition économique de puissance, peuvent être classifiées en deux groupes. Le premier représente la famille des méthodes d'optimisation non linéaire (ou programmation non linéaire) [1] qui sont basées sur la théorie du calcul différentiel où le gradient et/ou le Hessien qui sont utilisés pour guider la procédure de recherche afin de localiser la solution optimale. Le deuxième groupe inclut les méthodes de programmation linéaire, qui sont fondées sur les techniques du simplexe et du point intérieur [2]. Le dispatching économique est un problème d'optimisation qui consiste à répartir la production de la puissance active générée entre les différentes centrales du réseau, de sorte à exploiter ce dernier de

la manière la plus économique possible. Cette distribution doit évidemment respecter les limites de production des centrales.

1.3.2.1 Méthode du Gradient

La méthode du Gradient cherche à déterminer la direction de descente qui fait décroitre $f(x + dx)$ le plus vite possible. $\nabla f(X_k)$ indique la direction du taux de décroissement de f au point x_k . Plusieurs façons d'utiliser cette direction de descente existent, dont on peut citer :

- Gradient simple
- Gradient à pas optimal
- Gradient conjugué
- Gradient projeté

1.3.2.2 Méthode de newton

La répartition économique de puissance peut être résolue avec pour but de résoudre le gradient du lagrangien $\nabla L_x = 0$. Puisque c'est une fonction vectorielle, le problème peut être formulé pour trouver le chemin qui conduit le gradient vers zéro. La méthode de Newton est utilisée dans ce sens. La méthode de Newton pour une fonction à plusieurs variables est développée comme suit . Soit une fonction g(x) à minimiser tel que :
Soit une fonction $g(x)$ à minimiser tel que
$g(x + \Delta x) = g(x) + [g'(x)]$ et $\Delta x = 0$.
La construction de la matrice Jacobienne $[G'(x)]$.
l'ajustement en chaque itération se fait suivant :
$\Delta x = -[g'(x)]^{-1} g(x)$. Maintenant si la fonction $g(x)$ est le vecteur de gradient ∇L_x, donc $\nabla x = -\left[\dfrac{\partial \nabla L_x}{\partial x}\right]^{-1} \nabla L$.
Les éléments de la matrice jacobienne sont les dérivées partielles de second ordre, et la matrice $[G(x)]$ est appelée matrice hessienne.

1.3.2.3 La méthode d'itération de Lambda

La méthode d'itération de Lambda est l'une des méthodes utilisées pour trouver la valeur de Lambda du système et trouver le dispatching économique optimal des générateurs. Contrairement aux autres méthodes d'itération, comme : Gauss-Seidel et Newton-Raphson, la méthode d'itération de Lambda n'utilise pas la valeur précédente de l'inconnue pour trouver la

valeur suivante, c'est-à-dire qu'il n'y a pas d'équation qui calcule la valeur suivante en fonction de la valeur précédente. La valeur suivante est prédéfinie par intuition, elle est projetée avec interpolation de la bonne valeur possible jusqu'à ce que le décalage spécifié soit obtenu. On va maintenant discuter comment trouver le dispatching économique optimal utilisant cette dernière.

— La méthode exige qu'il y ait une correspondance entre une valeur *lambda* et l'output (en MW) de chaque générateur.

— — La méthode commence avec des valeurs de lambda en dessous et en-dessus de la valeur optimale (qui est inconnue), puis par itération limite la valeur optimale.

1.3.2.4 Méthode du point intérieur

Les méthodes de point intérieur fonctionnent comme suit : à partir d'une valeur initiale du paramètre de perturbation de la condition de complémentarité du système μ, strictement positif, et d'un point initial x_0 strictement réalisable, on résout de manière approchée le problème barrière. On calcule ensuite un nouveau paramètre de perturbation μ strictement positif et inférieur au précédent. On obtient alors un nouveau problème barrière à résoudre. On le résout de manière approchée à partir de la solution approchée du problème barrière précédent et ainsi de suite. On va donc résoudre, de manière approchée, une suite de problèmes barrières à μ fixé, la suite des paramètres de perturbation tendant vers 0, jusqu'à l'obtention d'une solution du problème initial.

1.3.3 Les méthodes d'intelligence artificielle

1.3.3.1 Algorithmes génétiques

Les algorithmes génétiques (AG) développés par J. Holland présentent des qualités intéressantes pour la résolution de problèmes d'optimisation complexes. Ils tentent de simuler le processus d'évolution des espèces dans leur milieu naturel : soit une transposition artificielle de concepts basiques de la génétique et des lois de survie énoncés par Darwin. Et aussi de la théorie de décision du mathématicien Hadamard du siècle passé.
Rappelons que la génétique représente un individu par un code, c'est-à-dire un ensemble de données (appelées chromosomes), identifiant complètement

l'individu. La reproduction est dans ce domaine un mixage aléatoire de chromosomes de deux individus, donnant naissance à des individus enfants ayant une empreinte génétique nouvelle, héritée des parents. La mutation génétique est caractérisée dans le code génétique de l'enfant par l'apparition d'un chromosome nouveau, inexistant chez les individus parents.
Ce phénomène génétique d'apparition de " mutants" est rare mais permet d'expliquer les changements dans la morphologie des espèces, toujours dans le sens d'une meilleure adaptation au milieu naturel.
La disparition de certaines espèces est expliquée par " les lois de survie " selon lesquelles seuls les individus les mieux adaptés auront une longévité suffisante pour générer une descendance. Les individus peu adaptés auront une tendance à disparaître.
C'est une sélection naturelle qui conduit de génération en génération à une population composée d'individus de plus en plus adaptés. Un algorithme génétique est construit de manière tout à fait analogue.
Dans l'ensemble des solutions d'un problème d'optimisation, une population est constituée de solutions convenablement marquées par un codage qui les identifie complètement. Une procédure d'évaluation est nécessaire à la détermination de la force de chaque individu de la population. Viennent ensuite une phase de sélection (en sélectionnant les individus suivant de leur force) et une phase de recombinaison (opérateurs artificiels de croisement et de mutation) qui génèrent une nouvelle population d'individus, qui ont de bonnes chances d'être plus forts que ceux de la génération précédente. De génération en génération, la force des individus de la population augmente et après un certain nombre d'itérations, la population est entièrement constituée d'individus tous forts, soit de solutions quasi-optimales du problème posé [3].

1.3.3.2 Algorithme à évolution différentielle

Classée parmi les méthodes méta-heuristiques stochastiques d'optimisation, l'algorithme à évolution différentielle (DEA) [4] est une technique relativement récente, conçue pour optimiser des problèmes sur les domaines continus. Cet algorithme est inspiré des algorithmes génétiques et les stratégies évolutionnistes combinées avec une technique géométrique de recherche. Les algorithmes génétiques changent la structure des individus en utilisant la mutation et le croisement, alors que les stratégies évolutionnistes réalisent l'auto adaptation par une manipulation géométrique des individus.

Dans cette approche, chaque variable de décision est représentée dans le chromosome (ou individu) par un nombre réel. Comme dans tout algorithme évolutionnaire, la population initiale du DEA est générée aléatoirement, puis évaluée. Après cela, le processus de sélection prend place. Au cours de la phase de sélection, trois parents sont choisis et ils génèrent un seul enfant (ou descendant) qui est en concurrence avec un parent pour déterminer lequel subsistera à la génération suivante. Le DEA génère un seul enfant (au lieu de deux comme dans les algorithmes génétiques) en ajoutant le vecteur de différence pondérée entre deux parents à un troisième parent. Si le vecteur résultant donne une plus faible valeur de la fonction objectif que celle donnée par un membre prédéterminé de la population, le vecteur nouvellement généré remplace le vecteur auquel il a été comparé.

1.3.3.3 Optimisation par essaim de particules

L'optimisation par essaim de particules est une technique évolutionnaire qui utilise une population de solutions candidates pour développer une solution optimale au problème. Le degré d'optimalité est mesuré par une fonction fitness [5]. Il est inspiré du comportement collectif et de l'intelligence émergente qui existent dans les sociétés à population organisée. Les membres de la population, particules, sont dispersés dans l'espace du problème, ainsi que le comportement de l'essaim peut être décrit en se plaçant du point de vue d'une particule. Au départ de l'algorithme, un essaim est réparti au hasard dans l'espace de recherche, chaque particule ayant également une vitesse aléatoire. Chaque particule est représentée par une position et une vitesse, qui sont mis à jour comme suit :

$$X_i^{k+1} = V_i^{k+1} + X_i^k \tag{1.9}$$

$$V_i^{k+1} = \omega V_i^k + c_1 rand_1(Pbest_i^k - X_i^k) + c_2 rand_2(Gbest^k - X_i^k) \tag{1.10}$$

Remarque

Notons que les générateurs à combustibles distincts possèdent différents coûts pour fournir le même montant d'énergie électrique. C'est important de se rendre compte que le générateur le plus rentable du système ne peut pas produire de l'électricité au coût le plus bas et qu'un générateur économique ne peut pas être le plus rentable, puisqu'un générateur qui se trouve trop loin du lieu de consommation entraine des pertes de transmission énormes.

Cependant, ces pertes varient en fonction de la répartition des puissances entre les centrales et la charge.

Dans ce chapitre, nous avons brièvement discuté de l'écoulement de puissance optimal et la répartition économique de puissance, ensuite, nous avons cité quelques méthodes conventionnelles et intelligentes d'une façon générale. Dans le chapitre suivant, nous présentons en en détails trois méthodes conventionnelles pour la répartition économique de puissance.

Chapitre 2

Application de la méthode de Newton à la répartition économique de puissance

Le but de la répartition économique est de réduire au minimum le coût de production pour satisfaire la demande (charge) d'un système de puissance tout en maintenant sa sécurité. Dans ce chapitre, nous présenterons quelques méthodes conventionnelles d'optimisation pour atteindre le but désiré. D'abord, la fonction objective, $f(x)$, est présentée. Elle représente l'objectif dont nous nous servirons pour réduire au minimum le coût de la production. Puis les contraintes d'égalités et d'inégalités seront présentées et discutées. Ces contraintes modélisent les lois physiques du système de puissance d'énergie électrique, ainsi que les conditions nécessaires pour le maintenir en sécurité. Généralement, les valeurs initiales de l'OPF sont les solutions de l'écoulement de puissance dont nous allons parler ci dessous.

2.1 Calcul de l'écoulement de puissance

Le calcul de l'écoulement de puissance consiste à résoudre les équations(1.6) et (1.7). Chaque nœud dans le réseau est caractérisé par $Pg_i, Qg_i, V_i, \theta_i$. Nous fixons deux paramètres et nous devons déterminer les deux autres. Suivant les valeurs spécifiées, les nœuds sont classés en trois types :

- Nœud de référence (slack bus) : V et θ fixes, P et Q inconnues.
- Nœud de génération : P et V fixes, Q et θ inconnues.
- Nœud de charge : P et Q fixes, V et θ inconnues.

trois méthodes classiques sont généralement utilisées

2.1.1 Méthode de Gauss-Seidel

Les premières méthodes étaient basées sur la méthode de $Gauss - Seidel$ relative à la matrice admittance Y . Elle ne nécessite pas beaucoup d'espace mémoire et sa programmation est relativement simple. Pour un système à plusieurs variables, La méthode de $Gauss - Seidel$ utilise à chaque itération la valeur la plus récente calculée. La méthode de $Gauss - Seidel$ se caractérise par sa faible convergence ; elle peut diverger complètement si la valeur initiale est mal choisie. Mais les petits réseaux ne nécessitent que peu d'itérations pour converger. Les grands réseaux, par contre demandent un grand nombre d'itérations si toutefois ils convergent. Ce désavantage a poussé les chercheurs à développer la méthode de $Newton - Raphson$.

2.1.2 Méthode de Newton-Raphson

Cette méthode nécessite plus de temps par itération que celle de Gauss-Seidel, alors qu'elle ne demande que quelques itérations même pour les grands réseaux. Cependant, elle requiert des capacités de stockage ainsi que des puissances de calcul importantes. A cause de la convergence quadratique de la méthode de$Newton - Raphson$, une solution de haute précision peut être obtenue en quelques itérations seulement. Ces caractéristiques font le succès de la méthode découplée rapide de $Newton - Raphson$.

2.1.3 Méthode Découplée rapide (Fast Decoupled Load Flow)

La variation de la puissance active est moins sensible à la variation de la tension V , par contre, elle est sensible à celle de la phase θ. D'autre part, la variation de la puissance réactive est plus sensible à la variation de la tension V , et est moins sensible à celle de la phase θ. La méthode découplée rapide (FDLF) effectue le même temps d'exécution que celle de $Newton - Raphson$ pour les très petits réseaux. Cependant, elle devient plus rapide pour les réseaux plus importants et pour les tolérances habituelles.

2.2 Méthode de Newton

2.2.1 Définition et généralisation de la méthode

En analyse numérique, la méthode de Newton est un algorithme efficace pour trou- ver numériquement une approximation précise d'un zéro (ou racine) d'une fonction réelle d'une variable réelle. Cette méthode a été développée par $Isaac\ Newton(1643-1727)$) qui a fait les premiers travaux pour la recherche des zéros d'une équation polynomiale, et aussi $Thomas\ Simpson(1710-1761)$ qui a élargi considérablement le domaine d'application de l'algorithme en montrant, grâce à la notion de dérivée, comment on pouvait l'utiliser pour calculer un zéro d'une équation non linéaire et d'un système formé de telles équations.

Nous allons donc chercher à construire une bonne approximation d'un zéro d'une fonction réelle $f(x)$ en considérant son développement de Taylor au premier ordre. Pour cela, partant d'un point x_0 que l'on choisit de préférence proche du zéro, on approche la fonction au premier ordre, autrement dit, on la considère qu'elle est égale à sa tangente en ce point :

$$f(x) = f(x_0) + f'(x_0)(x - x_0) \tag{2.1}$$

Puis, on résout l'équation $f(x) = 0$

On obtient alors un point x_1 qui a en général de bonnes chances d'être plus proche du vrai zéro de f que le point x_0 précédent. Par cette opération, on peut donc améliorer l'approximation par itérations successives. Cette méthode requiert que la fonction possède une tangente en chacun des points de la suite que l'on construit par itération donc, il suffit que f soit dérivable. Formellement, on part d'un point x_0 appartenant à l'ensemble de définition de la fonction et on construit par récurrence la suite

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{2.2}$$

le processus itérative se termine lorsque $(x_{k+1} - x_k)$ inférieur à la tolérance. On peut aussi utiliser la méthode de Newton pour résoudre un système de N équations (non linéaires) à N inconnues $x = [x_1, x_2, ...x_n]$, ce qui revient à trouver un zéro d'une fonction F multivariables , qui devra être différentiable. Dans la formulation donnée ci-dessus, il faut multiplier par l'inverse de la

matrice jacobienne $[F(x)]$ au lieu de diviser par $f`(x)$.

2.2.2 Utilisation de la méthode de Newton dans l'optimisation

L'application de la méthode de *Newton* à l'optimisation consiste à minimiser ou maximiser une fonction objective liée à des contraintes d'égalités et d'inégalités. D'après le problème représenté par les équations 1.1, 1.2, 1.3 nous disposons de M contraintes de type égalités et N contraintes de type inégalités. Le nombre de variables est égal à la dimension du vecteur X, tel que $X = [x1, x2, ..., xk]$.

2.2.3 Développement du Lagrangien, du Gradient, et du Hessien

La solution de problème présenté par 1.1, 1.2, 1.3 avec la méthode de Newton exige la création de lagrangien comme ci-dessous

$$L(z) = f(x) + \lambda^t g(x) + \mu^t h(x) \tag{2.3}$$

avec

- $Z = [x, \lambda, \mu]^t$
- λ et μ sont les vecteurs de multiplicateur de Lagrange.

Le gradient et l'hessien donc peuvent s'exprimer sous la forme suivant :

- $Gradient = \nabla L(z) = \dfrac{\partial L(z)}{\partial z_i}$. Vecteur de la première dérivée partiale du lagrangien
- Hessien $= \nabla^2 L(z) = H = \dfrac{\partial^2 L(z)}{\partial z_i \partial z_j} = \begin{bmatrix} \dfrac{\partial^2 L(z)}{\partial x_i \partial x_j} & \dfrac{\partial^2 L(z)}{\partial x_i \partial \mu_j} & \dfrac{\partial^2 L(z)}{\partial x_i \partial \lambda_j} \\ \dfrac{\partial^2 L(z)}{\partial \mu_i \partial x_j} & 0 & 0 \\ \dfrac{\partial^2 L(z)}{\partial \lambda_i \partial x_j} & 0 & 0 \end{bmatrix}$

H : matrice du deuxième dérivé partial de lagrangien.

Selon la théorie d'optimisation, les conditions nécessaires de Kuhn-Tucker de l'optimalité sont mentionnées comme ci-dessous.

2.2.4 Conditions de Kuhn-Tucker

Soit z^* la solution optimal, alors on aura :

- $\nabla_x L(z^*) = \nabla_x L([x^*, \lambda^*, \mu^*]) = 0$
- $\nabla_\lambda L(z^*) = \nabla_\lambda L([x^*, \lambda^*, \mu^*]) = 0$
- $\nabla_\mu L(z^*) = \nabla_\mu L([x^*, \lambda^*, \mu^*]) = 0$

$\lambda_i^* \geq 0$, si $g(x^*) = 0$ (la contrainte inégalité est active).

$\lambda_i^* = 0$, si $g(x^*) \leq 0$ (la contrainte inégalité est non active).

Le développement du gradient de lagrangien nous donne :

- $\dfrac{\partial L}{\partial U} = \dfrac{\partial F}{\partial U} + \lambda \dfrac{\partial G}{\partial U} + \mu \dfrac{\partial H}{\partial U} = 0$
- $\dfrac{\partial L}{\partial X} = \dfrac{\partial F}{\partial X} + \lambda \dfrac{\partial G}{\partial X} + \mu \dfrac{\partial H}{\partial X} = 0$
- $\dfrac{\partial L}{\partial \lambda} = G(U, X) = 0$
- $\dfrac{\partial L}{\partial \mu} = H(U, X) = 0$

On peut noter que le Lagrangien inclut seulement les inégalités qui sont imposées. Par exemple, si la tension d'un nœud i est dans la plage de fonctionnement désirée, il n'y a aucun besoin d'activer la contrainte inégalité liée à cette tension. Pour la formulation de la méthode de Newton, les contraintes d'inégalités doivent être manipulées en les séparant deux à deux : actif et inactif [6]. Pour des algorithmes efficaces, la détermination de ces contraintes d'inégalités qui sont actives a une grande importance. La boucle externe de l'organigramme exécute cette recherche des contraintes obligatoires ou actives [6].

2.2.5 Algorithme de la méthode de Newton

Une fois le gradient et la hessienne sont déterminer on les introduisant dans algorithme.

1. initialisez le vecteur $[z]$ et les inégalités qui sont imposés.
2. création de lagrangien on prend par considération les contraint d'inégalité active.
3. calculez le hessien et le gradient de lagrangien.
4. résoudre l'équation $\Delta z = -[H]^{-1} \nabla L(z)$.

5. calculez le nouveau $z_{new} = z_{old} + \Delta z$.
6. test si $\Delta z < toll$ aller à 7 si non aller à 3.
7. vérifier si les inégalité sont remplie si oui aller à 9 si non aller à 8.
8. Déterminer le nouveau ensemble d'inégalités et aller à 2.
9. fin.

L'organigramme de la méthode est illustré dans la figure suivante

2.2.6 Application de la méthode de Newton à l'EPO

Dans le domaine des réseaux électriques, la méthode de Newton est bien connue comme l'une des méthodes de solution de l'écoulement de puissance optimal. Elle été l'algorithme standard de la solution du problème de l'EPO pendant une longue période. L'approche de Newton est une formulation souple qui peut être adoptée pour développer des algorithmes différents pour l'EPO adaptés aux exigences des différentes applications comme la répartition économique de puissance.

Soit $Z = [Pg_1, ..., Pg_{ng}, V_{ng+1}, ..., V_n, \theta_2, ..., \theta_n, \lambda_{p_1}, ..., \lambda_{p_n}, \lambda_{q_{ng+1}}, ..., \lambda_{q_n}]$.

La fonction lagrangienne

$$\begin{aligned} L(z) = & \sum_{i=1}^{N_{gen}} a_i + b_i P_{g_i} + c_i P_{g_i}^2 + \\ & \sum_{i=1}^{N} \lambda_{pi} \left[\sum_{j=1}^{n} V_i V_j [G_{ij} cos(\delta_i - \delta j) + B_{ij} sin(\delta_i - \delta j)] - Pg_i + Pd_i \right] + \\ & \sum_{i=ng+1}^{N} \lambda_{qi} \left[\sum_{j=1}^{n} V_i V_j [G_{ij} sin(\delta_i - \delta j) - B_{ij} cos(\delta_i - \delta j)] - Qg_i + Qd_i \right]. \end{aligned} \tag{2.4}$$

La condition d'obtention du minimum de coût de génération avec les équations de l'écoulement de puissance sont vérifier. Aussi, on a :

$$\nabla L(z) = \left[\frac{\partial L}{\partial z_i} \right] = \left[\frac{\partial L}{\partial p_g}, \frac{\partial L}{\partial v}, \frac{\partial L}{\partial \theta}, \frac{\partial L}{\partial \lambda_p}, \frac{\partial L}{\partial \lambda_q} \right]^t = 0. \tag{2.5}$$

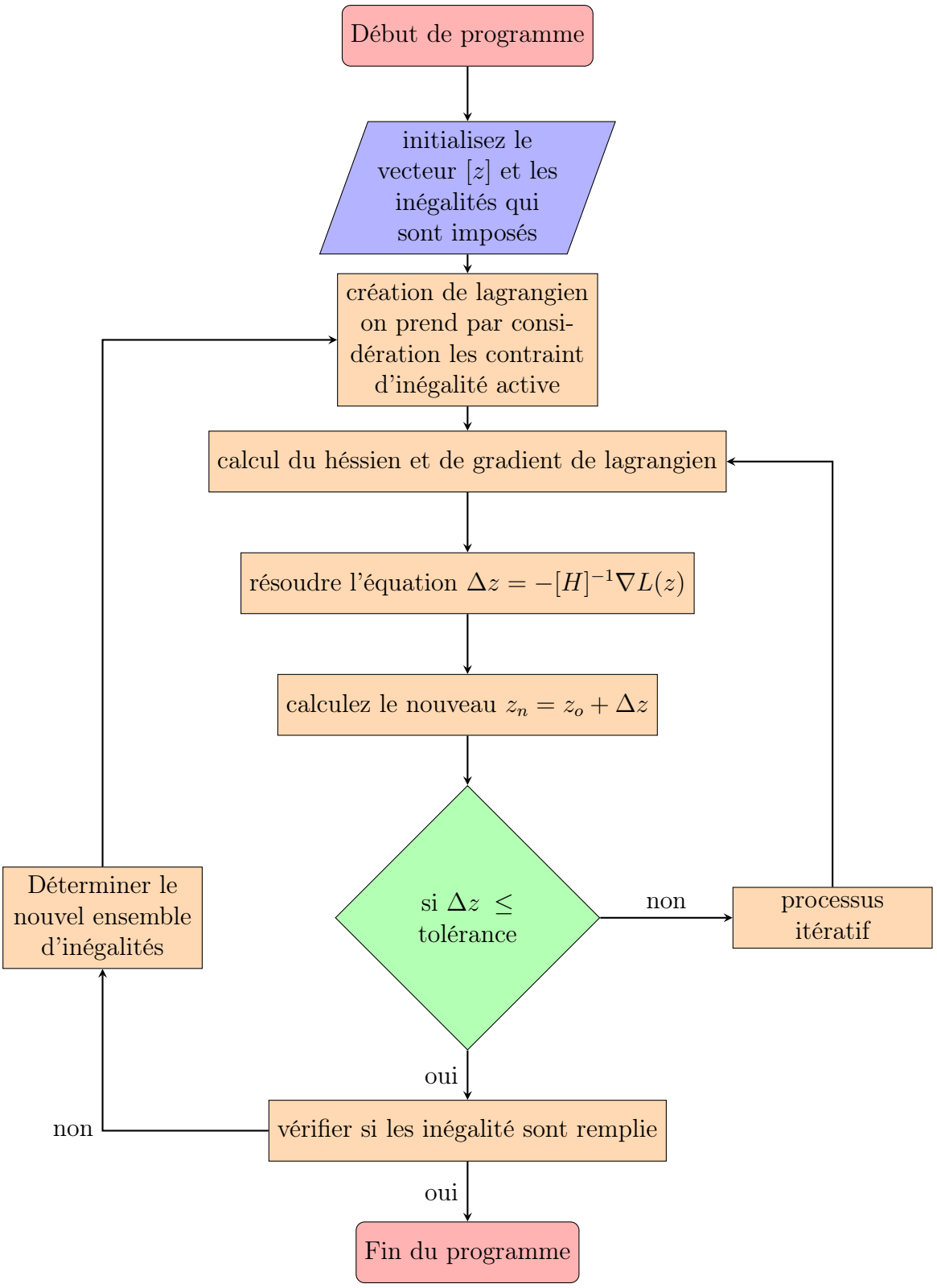


FIGURE 2.1: Organigramme d'algorithme du méthode de Newton

La construction de la matrice hessienne est obtenue à partir du calcul des dérivés mixte :

$$H = \nabla^2 L(z) = \begin{bmatrix} \frac{\partial^2 L(z)}{\partial P_{g_i} \partial P_{g_j}} & \frac{\partial^2 L(z)}{\partial P_{g_i} \partial V_j} & \frac{\partial^2 L(z)}{\partial P_{g_i} \partial \theta_j} & \frac{\partial^2 L(z)}{\partial P_{g_i} \partial \lambda_{P_j}} & \frac{\partial^2 L(z)}{\partial P_{g_i} \partial \lambda_{Q_j}} \\ \frac{\partial^2 L(z)}{\partial V_i \partial P_{g_j}} & \cdots & \cdots & \cdots & \\ \cdots & \cdots & \cdots & \cdots & \\ \frac{\partial^2 L(z)}{\partial \lambda_i \partial P_{g_j}} & \cdots & \cdots & \cdots & \frac{\partial^2 L(z)}{\partial \lambda_i \partial \lambda_j} \end{bmatrix} \tag{2.6}$$

$$\Delta Z = [\Delta P g_1, .., \Delta P g_{ng}, \Delta V_{ng+1}, .., \Delta V_n, \Delta \theta_2, .., \Delta \theta_n, \Delta \lambda_{p_1}, .., \Delta \lambda_{p_n}, \Delta \lambda_{q_{ng+1}}, .., \Delta \lambda_{q_n}] \tag{2.7}$$

Les nouvelles valeurs de $[z]$ sont obtenues comme suit :

$$[z]_{k+1} = [z]_k + [\Delta z] \tag{2.8}$$

Le calcul itératif se poursuivra jusqu'à ce que $[\Delta z]$ soit inférieur à une tolérance fixée.

2.2.7 Analyse des résultats obtenus par la méthode de Newton

Ensuite, nous avons appliqué la méthode de Newton sur le même système 9 nœuds. Les résultats obtenus sont illustrés au tableau 4.3

TABLE 2.1: Valeurs de P_g optimal calculés par la méthode de Newton

Numéro de nœud	Valeurs initiales de P_g(MW)	P_g après OPF(MW)	autres valeurs de P_g(MW)	P_g après OPF (MW)
1	71.95	90.74	100	90.752
2	163	134.69	100	134.691
3	85	94.51	119.95	94.511
P_g total (MW)	le coût initiale ($/h)	le coût optimisé ($/h)	le coût initiale ($/h)	le coût optimisé ($/h)
319.95	5438.32	5316.84	5537.5	5316.90

Le tableau 2.1 représente les valeurs de Pg optimal pour deux valeurs initiales différentes. Nous remarquons d'après le tableau que l'optimisation par

la méthode de newton ne dépend pas des valeurs initiales comme celle du gradient.

Pour mieux comprendre le comportement de la méthode de newton pour l'EPO pour des différents systèmes, nous appliquons cette dernière sur un réseau de trente nœuds (30 bus). On obtient les résultats suivantes :

TABLE 2.2: Résultats obtenus par la méthode de newton dans un système 30-bus

Numéro de nœuds	P_g du 1^{iter} (MW)	P_g du 2^{iter} (MW)	P_g du 18^{iter} (MW)	P_g du 100^{iter} (MW)
1	26.2	30.6	42.85	42.85
2	60.83	58.43	57.24	57.24
13	36.38	28.46	22.56	22.56
22	21.53	21.59	35.43	35.43
23	19.23	19.39	16.46	16.46
27	27.44	33.16	17.08	17.08
Le coût optimisé ($/h)	592.58	582.84	574.69	574.69

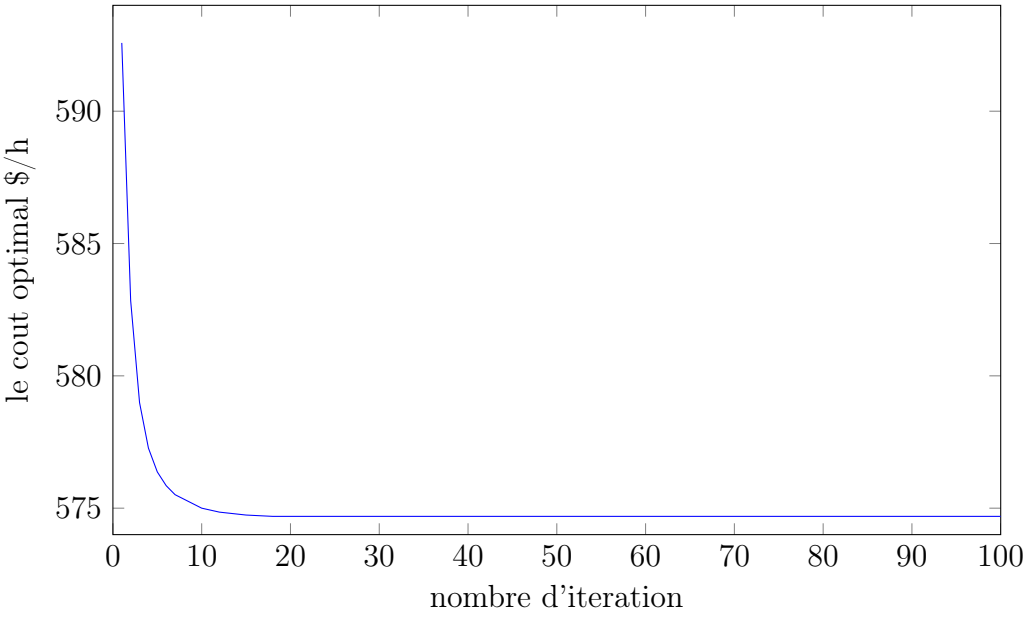


FIGURE 2.2: Méthode de Newton

D'après le tableau, nous voyons que la méthode de newton converge après 18 itérations. Si on compare avec le système (9-bus), on voie bien que le nombre d'itération est augmenté. Donc, le nombre d'itération nécessaire pour que la méthode de newton converge est augmenté suivant la taille de réseau.

Nous remarquons aussi que pour les nœuds 23 et 27 sont deux nœuds de générations qui ont la même fonction coût mais des valeurs de sorties de

P_g différentes. Cela est du au fait que les deux centrales ne sont pas dans la même position. Alors , la centrale la plus proche à la charge donne des valeurs de P_g plus que la centrale la plus éloignée. Après le calcul d'EPO sur ce système (30-bus), on réduit le coût de production d'une valeur de : 18.75 $\$/h$.

Conclusion

Ce mémoire nous a permis de traiter le problème d'écoulement de puissance optimal qui est un des problèmes les plus en vue dans le domaine du fonctionnement des réseaux électriques, surtout que les besoins en énergie électrique augmentent continuellement avec les besoins socio-économiques croissants dans toutes les sociétés du monde. Ce mémoire a essayé d'être un exemple d'application de la méthode de newton pour la résolution de ce problème. Notons que nous nous sommes concentrés sur le cas de la répartition économique, qui est un des problèmes les plus importants de l'écoulement de puissance optimal.

Nous avons ainsi porté notre choix sur une méthode conventionnelle : méthode de Newton , que nous avons détaillé, programmé et simulé. Notre étude nous a permis, de constater que cette méthode converge généralement vite, mais deviennent difficiles à utiliser lorsque le problème physique devient :

- fortement non linéaire (charges, régulation, autres dispositifs).
- la fonction à optimiser n'est pas différentiable.
- la fonction à optimiser comporte plusieurs objectifs simultanés (optimisation multi-objectifs).

Cette difficulté se traduit souvent par :

- leur convergence vers des optimums locaux .
- difficulté majeure liée à leur programmation et leur mise en œuvre.

Références

[1] M. JA, E.-H. ME, and A. R, "A review of selected optimal power flow literature to 1993,part i :nonlinear quadratic programming approach," *IEEE trans Power Sys*, pp. 96–104, 1999.

[2] H. Atun and T.yalcinoz, "Implementing soft computing techniques to solve economic dispatch problem in power systems," *xpert Systems with Applications*, vol. 35, 2008.

[3] O. Guenounou, *Méthodologie de conception de contrôleurs intelligents par l'approche génétique- application à un ioprocédé.* PhD thesis, Université Toulouse III, Paul Sabatier.

[4] M.Basu, "Economic enviremental dispatch using multiobjective differential evolution," *Applied soft Computing*, vol. 11, pp. 2845–2853, 2011.

[5] M. Clerc and J. Kennedy, "The particle swarm : Explosion, stability and convergence in a multi-dimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, 2001.

[6] *Optimal Power Flow Problem & Solution Methodologies.*

Annexe A

Système 9 nœuds

Ci dessous les données du système 30 nœuds

```
function mpc = case9
%CASE9    Power flow data for 9 bus, 3 generator case.
%   Please see CASEFORMAT for details on the case file format.
%
%   Based on data from Joe H. Chow's book, p. 70.

%   MATPOWER
%   $Id: case9.m 2408 2014-10-22 20:41:33Z ray $

%% MATPOWER Case Format : Version 2
mpc.version = '2';

%%-----  Power Flow Data  -----%%
%% system MVA base
mpc.baseMVA = 100;

%% bus data
% bus_i type Pd Qd Gs Bs area Vm Va baseKV zone Vmax Vmin
mpc.bus = [
1 3 0 0 0 0 1 1 0 345 1 1.1 0.9;
2 2 0 0 0 0 1 1 0 345 1 1.1 0.9;
3 2 0 0 0 0 1 1 0 345 1 1.1 0.9;
4 1 0 0 0 0 1 1 0 345 1 1.1 0.9;
5 1 90 30 0 0 1 1 0 345 1 1.1 0.9;
```

```
6 1 0 0 0 0 1 1 0 345 1 1.1 0.9;
7 1 100 35 0 0 1 1 0 345 1 1.1 0.9;
8 1 0 0 0 0 1 1 0 345 1 1.1 0.9;
9 1 125 50 0 0 1 1 0 345 1 1.1 0.9;
];

%% generator data
% bus Pg Qg Qmax Qmin Vg mBase status Pmax Pmin Pc1 Pc2 Qc1min Qc1max
Qc2min Qc2max ramp_agc ramp_10 ramp_30 ramp_q apf
mpc.gen = [
1 0 0 300 -300 1 100 1 250 10 0 0 0 0 0 0 0 0 0 0 0;
2 163 0 300 -300 1 100 1 300 10 0 0 0 0 0 0 0 0 0 0 0;
3 85 0 300 -300 1 100 1 270 10 0 0 0 0 0 0 0 0 0 0 0;
];

%% branch data
% fbus tbus r x b rateA rateB rateC ratio angle status angmin angmax
mpc.branch = [
1 4 0 0.0576 0 250 250 250 0 0 1 -360 360;
4 5 0.017 0.092 0.158 250 250 250 0 0 1 -360 360;
5 6 0.039 0.17 0.358 150 150 150 0 0 1 -360 360;
3 6 0 0.0586 0 300 300 300 0 0 1 -360 360;
6 7 0.0119 0.1008 0.209 150 150 150 0 0 1 -360 360;
7 8 0.0085 0.072 0.149 250 250 250 0 0 1 -360 360;
8 2 0 0.0625 0 250 250 250 0 0 1 -360 360;
8 9 0.032 0.161 0.306 250 250 250 0 0 1 -360 360;
9 4 0.01 0.085 0.176 250 250 250 0 0 1 -360 360;
];
%%-----  OPF Data  -----%%
%% generator cost data
% 1 startup shutdown n x1 y1 ... xn yn
% 2 startup shutdown n c(n-1) ... c0
mpc.gencost = [
2 1500 0 3 0.11 5 150;
2 2000 0 3 0.085 1.2 600;
2 3000 0 3 0.1225 1 335];
```

Annexe B

Système 30 nœuds

```
function mpc = case30
%CASE30    Power flow data for 30 bus, 6 generator case.
%   Please see CASEFORMAT for details on the case file format.
%
%   Based on data from ...
%     Alsac, O. & Stott, B., "Optimal Load Flow with Steady State Security"
%     IEEE Transactions on Power Apparatus and Systems, Vol. PAS 93, No. 3,
%     1974, pp. 745-751.
%   ... with branch parameters rounded to nearest 0.01, shunt values divide
%   by 100 and shunt on bus 10 moved to bus 5, load at bus 5 zeroed out.
%   Generator locations, costs and limits and bus areas were taken from ...
%     Ferrero, R.W., Shahidehpour, S.M., Ramesh, V.C., "Transaction analysi
%     in deregulated power systems using game theory", IEEE Transactions on
%     Power Systems, Vol. 12, No. 3, Aug 1997, pp. 1340-1347.
%   Generator Q limits were derived from Alsac & Stott, using their Pmax
%   capacities. V limits and line |S| limits taken from Alsac & Stott.

%   MATPOWER
%   $Id: case30.m 2408 2014-10-22 20:41:33Z ray $

%% MATPOWER Case Format : Version 2
mpc.version = '2';

%%-----  Power Flow Data  -----%%
%% system MVA base
```

```
mpc.baseMVA = 100;

%% bus data
% bus_i type Pd Qd Gs Bs area Vm Va baseKV zone Vmax Vmin
mpc.bus = [
1 3 0 0 0 0 1 1 0 135 1 1.05 0.95;
2 2 21.7 12.7 0 0 1 1 0 135 1 1.1 0.95;
3 1 2.4 1.2 0 0 1 1 0 135 1 1.05 0.95;
4 1 7.6 1.6 0 0 1 1 0 135 1 1.05 0.95;
5 1 0 0 0 0.19 1 1 0 135 1 1.05 0.95;
6 1 0 0 0 0 1 1 0 135 1 1.05 0.95;
7 1 22.8 10.9 0 0 1 1 0 135 1 1.05 0.95;
8 1 30 30 0 0 1 1 0 135 1 1.05 0.95;
9 1 0 0 0 0 1 1 0 135 1 1.05 0.95;
10 1 5.8 2 0 0 3 1 0 135 1 1.05 0.95;
11 1 0 0 0 0 1 1 0 135 1 1.05 0.95;
12 1 11.2 7.5 0 0 2 1 0 135 1 1.05 0.95;
13 2 0 0 0 0 2 1 0 135 1 1.1 0.95;
14 1 6.2 1.6 0 0 2 1 0 135 1 1.05 0.95;
15 1 8.2 2.5 0 0 2 1 0 135 1 1.05 0.95;
16 1 3.5 1.8 0 0 2 1 0 135 1 1.05 0.95;
17 1 9 5.8 0 0 2 1 0 135 1 1.05 0.95;
18 1 3.2 0.9 0 0 2 1 0 135 1 1.05 0.95;
19 1 9.5 3.4 0 0 2 1 0 135 1 1.05 0.95;
20 1 2.2 0.7 0 0 2 1 0 135 1 1.05 0.95;
21 1 17.5 11.2 0 0 3 1 0 135 1 1.05 0.95;
22 2 0 0 0 0 3 1 0 135 1 1.1 0.95;
23 2 3.2 1.6 0 0 2 1 0 135 1 1.1 0.95;
24 1 8.7 6.7 0 0.04 3 1 0 135 1 1.05 0.95;
25 1 0 0 0 0 3 1 0 135 1 1.05 0.95;
26 1 3.5 2.3 0 0 3 1 0 135 1 1.05 0.95;
27 2 0 0 0 0 3 1 0 135 1 1.1 0.95;
28 1 0 0 0 0 1 1 0 135 1 1.05 0.95;
29 1 2.4 0.9 0 0 3 1 0 135 1 1.05 0.95;
30 1 10.6 1.9 0 0 3 1 0 135 1 1.05 0.95;
];
```

```
%% generator data
% bus Pg Qg Qmax Qmin Vg mBase status Pmax Pmin Pc1 Pc2 Qc1min Qc1max
Qc2min Qc2max ramp_agc ramp_10 ramp_30 ramp_q apf
mpc.gen = [
1 23.54 0 150 -20 1 100 1 80 0 0 0 0 0 0 0 0 0 0 0 0;
2 60.97 0 60 -20 1 100 1 80 0 0 0 0 0 0 0 0 0 0 0 0;
22 21.59 0 62.5 -15 1 100 1 50 0 0 0 0 0 0 0 0 0 0 0 0;
27 26.91 0 48.7 -15 1 100 1 55 0 0 0 0 0 0 0 0 0 0 0 0;
23 19.2 0 40 -10 1 100 1 30 0 0 0 0 0 0 0 0 0 0 0 0;
13 37 0 44.7 -15 1 100 1 40 0 0 0 0 0 0 0 0 0 0 0 0;
];

%% branch data
% fbus tbus r x b rateA rateB rateC ratio angle status angmin angmax
mpc.branch = [
1 2 0.02 0.06 0.03 130 130 130 0 0 1 -360 360;
1 3 0.05 0.19 0.02 130 130 130 0 0 1 -360 360;
2 4 0.06 0.17 0.02 65 65 65 0 0 1 -360 360;
3 4 0.01 0.04 0 130 130 130 0 0 1 -360 360;
2 5 0.05 0.2 0.02 130 130 130 0 0 1 -360 360;
2 6 0.06 0.18 0.02 65 65 65 0 0 1 -360 360;
4 6 0.01 0.04 0 90 90 90 0 0 1 -360 360;
5 7 0.05 0.12 0.01 70 70 70 0 0 1 -360 360;
6 7 0.03 0.08 0.01 130 130 130 0 0 1 -360 360;
6 8 0.01 0.04 0 32 32 32 0 0 1 -360 360;
6 9 0 0.21 0 65 65 65 0 0 1 -360 360;
6 10 0 0.56 0 32 32 32 0 0 1 -360 360;
9 11 0 0.21 0 65 65 65 0 0 1 -360 360;
9 10 0 0.11 0 65 65 65 0 0 1 -360 360;
4 12 0 0.26 0 65 65 65 0 0 1 -360 360;
12 13 0 0.14 0 65 65 65 0 0 1 -360 360;
12 14 0.12 0.26 0 32 32 32 0 0 1 -360 360;
12 15 0.07 0.13 0 32 32 32 0 0 1 -360 360;
12 16 0.09 0.2 0 32 32 32 0 0 1 -360 360;
14 15 0.22 0.2 0 16 16 16 0 0 1 -360 360;
```

```
16 17 0.08 0.19 0 16 16 16 0 0 1 -360 360;
15 18 0.11 0.22 0 16 16 16 0 0 1 -360 360;
18 19 0.06 0.13 0 16 16 16 0 0 1 -360 360;
19 20 0.03 0.07 0 32 32 32 0 0 1 -360 360;
10 20 0.09 0.21 0 32 32 32 0 0 1 -360 360;
10 17 0.03 0.08 0 32 32 32 0 0 1 -360 360;
10 21 0.03 0.07 0 32 32 32 0 0 1 -360 360;
10 22 0.07 0.15 0 32 32 32 0 0 1 -360 360;
21 22 0.01 0.02 0 32 32 32 0 0 1 -360 360;
15 23 0.1 0.2 0 16 16 16 0 0 1 -360 360;
22 24 0.12 0.18 0 16 16 16 0 0 1 -360 360;
23 24 0.13 0.27 0 16 16 16 0 0 1 -360 360;
24 25 0.19 0.33 0 16 16 16 0 0 1 -360 360;
25 26 0.25 0.38 0 16 16 16 0 0 1 -360 360;
25 27 0.11 0.21 0 16 16 16 0 0 1 -360 360;
28 27 0 0.4 0 65 65 65 0 0 1 -360 360;
27 29 0.22 0.42 0 16 16 16 0 0 1 -360 360;
27 30 0.32 0.6 0 16 16 16 0 0 1 -360 360;
29 30 0.24 0.45 0 16 16 16 0 0 1 -360 360;
8 28 0.06 0.2 0.02 32 32 32 0 0 1 -360 360;
6 28 0.02 0.06 0.01 32 32 32 0 0 1 -360 360;
];

%%-----  OPF Data  -----%%
%% generator cost data
% 1 startup shutdown n x1 y1 ... xn yn
% 2 startup shutdown n c(n-1) ... c0
mpc.gencost = [
2 0 0 3 0.02 2 0;
2 0 0 3 0.0175 1.75 0;
2 0 0 3 0.0625 1 0;
2 0 0 3 0.00834 3.25 0;
2 0 0 3 0.025 3 0;
2 0 0 3 0.025 3 0;
];
```