

---

# Application d'une méthode de programmation quadratique successive à l'optimisation des puissances dans les réseaux électriques de grande taille

*Application of a recursive quadratic programming method to optimal power flows in large-scale electric power systems*

G. Blanchon (\*), J.F. Bonnans (\*\*), J.C. Dodu (\*\*\*)

---

## Sommaire

1.	Introduction . . . . .	68
2.	Formulation d'une classe de problèmes d'optimisation des puissances dans un réseau électrique . . . . .	70
2.1.	Equations décrivant l'état d'un réseau électrique en régime permanent . . . . .	70
2.2.	Une classe de problèmes d'optimisation des puissances dans un réseau électrique . . . . .	71
2.3.	Formulation synthétique des problèmes d'optimisation . . . . .	75
3.	Un algorithme de type Newton pour les problèmes non linéaires de grande taille . . . . .	78
3.1.	Le problème modèle. Equivalence avec un problème standard de programmation non linéaire . . . . .	78
3.2.	Les méthodes de type Newton (analyse locale) . . . . .	79
3.3.	Analyse de la vitesse de convergence de la méthode de Newton . . . . .	80
3.4.	Convergence globale . . . . .	81

---

(\*) Ingénieur-Chercheur au Département Méthodes d'Optimisation et de Simulation, Service Etudes de Réseaux (Direction des Etudes et Recherches).

(\*\*) Directeur de Recherche à l'Institut National de Recherche en Informatique et en Automatique (INRIA), Domaine de Voluceau, B P 105, 78153 Le Chesnay Cedex.

(\*\*\*) Attaché au Département Méthodes d'Optimisation et de Simulation, Service Etudes de Réseaux (Direction des Etudes et Recherches)

3.5.	Résolution du problème quadratique par une méthode de type gradient réduit conjugué	84
4.	Résultats et applications numériques	88
4.1.	Description du réseau	88
4.2.	Description du problème	89
4.3.	Résultats	90
5.	Annexe : Expression de l'intensité du courant qui circule dans une branche du réseau	94
	Bibliographie	95
1.	Optimisation des réseaux électriques	95
2.	Algorithmes de programmation non linéaire	98

## 1. Introduction

Le problème de la répartition optimale des puissances ("optimal power flow") est l'un des principaux problèmes rencontrés par les ingénieurs d'études de réseaux électriques qui utilisent les outils de la recherche opérationnelle. Depuis le début des années 60, la plupart des méthodes d'optimisation non linéaire ont été expérimentées pour le résoudre. Il a sans doute été formulé pour la première fois sous la forme d'un programme mathématique par Carpentier [C (1962)]. Parmi les méthodes utilisées en premier, on peut citer :

- 1) des méthodes de pénalité [S (1969)<sub>a</sub>], [S (1969)<sub>b</sub>], [S-V-A (1973)],
- 2) des méthodes de gradient réduit, principalement la méthode du gradient réduit généralisé (GRG) [D-T (1968)], [P et al (1968)], [A-C (1969)], [P-B-H (1971)], [P-B-H (1972)], [C (1973)], [C et al (1975)], [A-S (1975)].

A partir des années 80, ont ensuite été appliquées :

- 1) des méthodes de programmation linéaire successive [B-D-M (1983)], [A-S (1983)],
- 2) des méthodes de lagrangiens augmentés [B-H-W (1982)], [F et al (1982)],
- 3) des méthodes de programmation quadratique successive.

Parmi ces dernières, on distingue deux classes :

- 1) Les méthodes de Newton, dans lesquelles la matrice des dérivées secondes du critère quadratique est le hessien du lagrangien du problème [S et al (1984)], [B-H-V (1984)], [A-K (1985)], [M-F (1987)], [VA (1988)].

- 2) Les méthodes de quasi-Newton, dans lesquelles la matrice des dérivées secondes du critère quadratique est une approximation du hessien du lagrangien. Cette approximation est mise à jour à chaque itération en utilisant une formule de quasi-Newton telle que la formule DFP ou la formule BFGS. La méthode principalement utilisée est la méthode de Han-Powell, dans laquelle les mises à jour s'effectuent à l'aide d'une formule BFGS modifiée qui permet de conserver la définie-positivité de la matrice des dérivées secondes du critère quadratique [G-T (1981)], [L-C (1981)], [T-G (1982)], [T-G-K (1983)], [N (1983)], [F et al. (1983)], [F-I-M (1983)], [C et al. (1987)].

Les méthodes de programmation quadratique successive sont considérées aujourd'hui comme les plus efficaces par de nombreux auteurs [S et al. (1984)], [B-H-V (1984)], [A-K (1985)], [M-F (1987)], [C et al. (1987)]. On sait qu'elles possèdent une vitesse de convergence superlinéaire (et même quadratique dans le cas des méthodes de Newton), ce qui, de ce point de vue, les rend supérieures aux autres méthodes. On trouvera des éléments de comparaison avec les méthodes de gradient réduit dans [B-H-W (1982)] et avec les méthodes de lagrangiens augmentés dans [A-K (1985)] et [B-H-V (1984)].

Dans une méthode de quasi-Newton, le hessien du critère quadratique est une matrice pleine, ce qui limite la taille des problèmes étudiés. La dimension du hessien du critère quadratique ne semble pas pouvoir dépasser quelques centaines<sup>(§)</sup>.

Dans une méthode de Newton, le hessien du critère quadratique est le hessien du lagrangien du problème non linéaire. C'est une matrice creuse qui possède le même creux que le jacobien des équations qui modélisent un réseau électrique en régime permanent. On peut envisager des tailles de hessien pouvant atteindre  $4\,000 \times 4\,000$ , ce qui correspond à l'étude de réseaux pouvant comporter jusqu'à 2 000 sommets [B-H-V (1984)].

Par contre, le hessien du lagrangien n'est pas défini positif et il peut être nécessaire de "convexifier" le problème quadratique [A-K (1985)]. La manière la plus simple est d'ajouter au hessien une matrice de la forme  $\alpha I$ , où  $\alpha$  est un coefficient positif ou nul et  $I$  est une matrice identité [M-F (1987)].

De plus, le calcul des dérivées secondes est simple (au moins dans le principe!) car les fonctions non linéaires sont analytiques. Mais, à la différence des autres études consacrées à ce problème qui opèrent sans recherche linéaire, nous allons munir l'algorithme d'outils issus d'une réflexion théorique qui permettent d'assurer la convergence globale tout en gardant le pas égal à 1 (donc la convergence quadratique) près de l'optimum. Ces outils sont l'idée de convexification du problème quadratique s'il ne donne pas une direction de descente d'un critère de recherche linéaire, ainsi qu'un nouveau critère de recherche linéaire. De plus, l'implémentation du problème quadratique est adaptée à la grande taille : résolution par gradient réduit avec factorisation LU creuse et

<sup>(§)</sup>Il ne semble pas, à ce jour, que les formules de quasi-Newton qui permettent de préserver le creux du hessien du lagrangien du problème non-linéaire aient été utilisées.

conjugaison des directions réduites lorsque cela a un sens. Il est intéressant de noter que si, à la suite des travaux de Han [H (1977)], les méthodes de programmation quadratique successive ont été largement utilisées, les codes publiés dans la littérature ont été implantés essentiellement pour des problèmes de petite taille. Au contraire, les implémentations concernant les problèmes de grande taille ont porté sur des méthodes de type Robinson (cf. [M-S (1982)]) ou des méthodes de gradient réduit conjugué pour des systèmes à contraintes linéaires (cf. [S-M (1982)]). Notre implémentation, à laquelle nous avons donné le nom OPSYC (optimisation de systèmes creux), est donc une des premières implémentations de la méthode de Newton pour les problèmes de grande taille, qui reprenne les outils de globalisation de l'algorithme.

## 2. Formulation d'une classe de problèmes d'optimisation des puissances dans un réseau électrique

### 2.1. Equations décrivant l'état d'un réseau électrique en régime permanent

Le régime permanent sinusoïdal qui s'établit dans un réseau électrique parcouru par du courant alternatif, peut être modélisé par un système d'équations linéaires de la forme :

$$\bar{I} = \bar{Y}\bar{V}$$

où  $\bar{Y}$  est la matrice d'admittance nodale du réseau,  $\bar{I}$  est le vecteur des courants injectés en chaque nœud et  $\bar{V}$  est le vecteur des tensions en chaque nœud.  $\bar{Y}$ ,  $\bar{I}$  et  $\bar{V}$  sont des grandeurs complexes. Si le réseau possède  $N$  nœuds,  $\bar{I}$  et  $\bar{V}$  sont des vecteurs ayant  $N$  composantes complexes.  $\bar{Y}$  est une matrice complexe, de dimension  $N \times N$ , dont les éléments  $\bar{Y}_{ik}$  sont calculés à partir des caractéristiques physiques des composants du réseau (par exemple, les résistances et les réactances des lignes). C'est une matrice creuse telle que, pour  $i \neq k$  :

$$\bar{Y}_{ik} = -\frac{1}{r_{ik} + jx_{ik}}$$

si les nœuds  $i$  et  $k$  sont reliés par un circuit électrique de résistance  $r_{ik}$  et de réactance  $x_{ik}$ , et  $\bar{Y}_{ik} = 0$  sinon. On aura ainsi :

$$(2.1) \quad \bar{I}_i = \bar{Y}_{ii}\bar{V}_i + \sum_{k \in A_i} \bar{Y}_{ik}\bar{V}_k, \quad i = 1, \dots, N$$

où  $A_i$  désigne l'ensemble des nœuds connectés au nœud  $i$ .

Au système (2.1), on préfère généralement un modèle faisant intervenir les puissances actives et réactives injectées en chaque nœud telles que :

$$(2.2) \quad P_i - jQ_i = \bar{V}_i^* \bar{I}_i$$

où  $P_i$  est la puissance active injectée au nœud  $i$ ,

$Q_i$  est la puissance réactive injectée au nœud  $i$ ,

$\bar{V}_i^*$  est le nombre complexe conjugué de  $\bar{V}_i$ .

De (2.1) et (2.2), on déduit :

$$\begin{aligned} P_i &= \operatorname{Re} \left[ \bar{V}_i^* \left( \bar{Y}_{ii} \bar{V}_i + \sum_{k \in A_i} \bar{Y}_{ik} \bar{V}_k \right) \right], \quad i = 1, \dots, N \\ Q_i &= -\operatorname{Im} \left[ \bar{V}_i^* \left( \bar{Y}_{ii} \bar{V}_i + \sum_{k \in A_i} \bar{Y}_{ik} \bar{V}_k \right) \right], \quad i = 1, \dots, N \end{aligned}$$

où  $\operatorname{Re}$  et  $\operatorname{Im}$  désignent respectivement la partie réelle et la partie imaginaire d'un nombre complexe  
En posant :

$$\bar{V}_i = V_i \exp(j\theta_i), \quad \bar{Y}_{ii} = y_i \exp(-j\xi_i), \quad \bar{Y}_{ik} = -y_{ik} \exp(-j\xi_{ik})$$

on obtient le système de  $2N$  équations algébriques :

$$(2.3) \quad P_i = y_i V_i^2 \cos \xi_i - \sum_{k \in A_i} y_{ik} V_i V_k \cos(\theta_i - \theta_k + \xi_{ik})$$

$$(2.4) \quad Q_i = y_i V_i^2 \sin \xi_i - \sum_{k \in A_i} y_{ik} V_i V_k \sin(\theta_i - \theta_k + \xi_{ik})$$

On note qu'à chaque sommet  $i$ , sont affectées quatre variables  $P_i$ ,  $Q_i$ ,  $V_i$  et  $\theta_i$ , et deux équations. Dans ce qui suit, les équations (2.3) (2.4) seront écrites sous la forme condensée :

$$(2.5) \quad P = \varphi(\theta, V)$$

$$(2.6) \quad Q = \psi(\theta, V)$$

où  $P$ ,  $Q$ ,  $\theta$  et  $V$  sont des vecteurs de dimension  $N$  ayant respectivement pour composantes  $P_i$ ,  $Q_i$ ,  $\theta_i$ , et  $V_i$ .

**Remarque :** A une grandeur complexe  $\bar{X} = X \exp(j\varphi)$  (par exemple la tension  $\bar{V}_i$ ) correspond le régime permanent sinusoïdal représenté par la fonction du temps :

$$x(t) = X \cos(\omega t + \varphi)$$

où  $\omega/2\pi$  est la fréquence du courant alternatif

## 2.2. Une classe de problèmes d'optimisation des puissances dans un réseau électrique

Dans les études de planification, de gestion prévisionnelle ou d'exploitation des réseaux électriques, on est amené à résoudre des problèmes d'optimisation consistant à minimiser une fonction des variables  $P$ ,  $Q$ ,  $\theta$  et  $V$ , tout en respectant les équations algébriques du réseau (2.5) (2.6) et des contraintes d'inégalité qui traduisent les limites de fonctionnement des composants du réseau (centrales thermiques et hydrauliques, lignes, transformateurs, ...). Un tel problème se pose sous la forme d'un programme mathématique non-linéaire qui s'écrit :

Minimiser  $f(P, Q, \theta, V)$  sous les conditions :

$$(2.5) \quad P = \varphi(\theta, V)$$

$$(2.6) \quad Q = \psi(\theta, V)$$

$$(2.7) \quad g(P, Q, \theta, V) \leq 0$$

$$(2.8) \quad P^m \leq P \leq P^M$$

$$(2.9) \quad Q^m \leq Q \leq Q^M$$

$$(2.10) \quad V^m \leq V \leq V^M$$

### Exemple 1 : Le dispatching économique :

La puissance active injectée au nœud  $i$  est la somme algébrique des puissances fournies par les centrales thermiques et hydrauliques, et de la puissance consommée par les usagers. Nous supposons que le parc de production ne comporte que des groupes thermiques et nous désignerons par :  $G_i$  l'ensemble des groupes thermiques situés au nœud  $i$  ( $G_i$  peut être vide),

$P_{ig}$  ( $g \in G_i$ ) la puissance active fournie par le groupe  $g$ ,

$P_{ic}$  la puissance active consommée au nœud  $i$  ( $P_{ic} \geq 0$  est une donnée).

On aura :

$$(2.11) \quad P_i = \sum_{g \in G_i} P_{ig} - P_{ic}, \quad i = 1, \dots, N$$

où  $P_{ig}$  satisfait des inégalités du type :

$$(2.12) \quad 0 < P_{ig}^m \leq P_{ig} \leq P_{ig}^M$$

Le coût de production d'un groupe thermique en fonction de la puissance active fournie peut être représenté par une courbe convexe, linéaire par morceaux, comme celle de la figure 1. Le problème du dispatching économique consiste à minimiser le coût total de production :

$$(2.13) \quad C_T = \sum_{i=1}^N \sum_{g \in G_i} C_{ig}(P_{ig})$$

en respectant :

- (i) les équations (2.5), (2.6), (2.11),
- (ii) les inégalités (2.9), (2.10), (2.12),
- (iii) les limites d'intensité maximale dans les lignes et les transformateurs.

Les contraintes (iii) sont du type (2.7). Elles sont de la forme :

$$\tau_{ik}(\theta, V) \leq \tau_{ik}^M, \quad \tau_{ki}(\theta, V) \leq \tau_{ki}^M$$

où  $i$  et  $k$  sont deux nœuds du réseau reliés par une branche (une ligne ou un transformateur).  $\tau_{ik}(\theta, V)$  est l'intensité (mesurée côté  $i$ ) du courant qui circule dans la branche reliant les nœuds  $i$  et  $k$ , et  $\tau_{ki}(\theta, V)$  est l'intensité (mesurée côté  $k$ ) de ce même courant (cf. annexe (paragraphe 5)).

### Exemple 2 : L'optimisation du plan de tension

Dans ce problème, on souhaite que la puissance active injectée au nœud  $i$  soit fixée et égale à une valeur donnée  $P_i^0$ . Ceci n'est pas possible a priori car on doit réaliser l'équilibre :

$$(2.14) \quad \sum_{i=1}^N P_i = \pi$$

où  $\pi$  représente les pertes actives du réseau (dissipées par effet Joule dans les résistances des lignes et des transformateurs), et ces pertes ne sont pas connues à l'avance. En pratique, l'équation (2.14) est satisfaite grâce à l'action des régulateurs automatiques de vitesse des alternateurs. Nous supposons que la loi de réglage automatique conduit à des relations de la forme :

$$(2.15) \quad P_i = P_i^0 + \lambda_i p, \quad \forall i \in G$$

où  $G \subset \{1, \dots, N\}$  représente le sous-ensemble des nœuds qui participent au réglage, et où les coefficients  $\lambda_i$  sont donnés et tels que :

$$\lambda_i > 0 \quad (i \in G), \quad \sum_{i \in G} \lambda_i = 1$$

On aura :

$$(2.16) \quad \begin{aligned} P_i &= P_i^0, \quad \forall i \notin G \\ p &= \pi - \sum_{i=1}^N P_i^0 \end{aligned}$$

La puissance réactive injectée au nœud  $i$  sera de la forme :

$$(2.17) \quad Q_i = Q_{ig} + C_i - S_i - Q_{ic}$$

Dans (2.17),  $Q_{ig}$  est la somme algébrique de trois termes :

- la puissance réactive fournie ou absorbée par les groupes,
- la puissance réactive fournie par les condensateurs,
- la puissance réactive absorbée par les selfs.

$Q_{ig}$  satisfait des inégalités du type :

$$(2.18) \quad Q_{ig}^m \leq Q_{ig} \leq Q_{ig}^M, \quad i \in R$$

où  $R \subset \{1, \dots, N\}$  est le sous-ensemble des nœuds où il y a au moins une source de puissance réactive (groupe, condensateur ou self). Dans (2.18),  $Q_{ig}^m$  peut être négatif.

$C_i$  et  $S_i$  sont des variables supplémentaires, que nous désignerons par variables artificielles, et qui sont telles que :

$$(2.19) \quad C_i \geq 0, \quad S_i \geq 0, \quad i = 1, \dots, N$$

$Q_{ig}$  est la puissance réactive consommée au nœud  $i$ . C'est une donnée qui peut être positive ou nulle, ou négative.

Le problème de l'optimisation du plan de tension consiste à minimiser une fonction de la forme :

$$(2.20) \quad F = \alpha p + \beta \sum_{i=1}^N C_i + \gamma \sum_{i=1}^N S_i$$

ou bien de la forme :

$$(2.21) \quad F = \alpha \sum_{i=1}^N (V_i^M - V_i) + \beta \sum_{i=1}^N C_i + \gamma \sum_{i=1}^N S_i$$

où  $\alpha$  est un coefficient positif et où  $\beta > 0$  et  $\gamma > 0$  sont choisis très grands par rapport à  $\alpha$ . La forme (2.20) correspond à la minimisation des pertes actives, alors que la forme (2.21) correspond à la recherche du plan de tension le plus élevé possible.

Les contraintes à respecter sont :

(i) les équations (2.5), (2.6), (2.15), (2.16), (2.17),

(ii) les inégalités (2.10), (2.18), (2.19).

Les variables artificielles  $C_i$  et  $S_i$  sont introduites dans la formulation du problème, pour que celui-ci ait toujours une solution réalisable. Si, à l'optimum, ces variables sont positives, elles peuvent être interprétées comme le volume global de condensateurs et de selfs supplémentaires qu'il faudrait installer au nœud  $i$  pour avoir un fonctionnement du réseau dans les limites de tension admissible.

**Remarque :** Il n'y a pas de contraintes de type (2.7).



### 2.3. Formulation synthétique des problèmes d'optimisation

Les exemples 1 et 2 du paragraphe précédent entrent dans le cadre de la formulation

$$\begin{array}{l} \text{Minimiser } f(x) + \sum_{s=1}^S \phi_s(e_s) \\ e_s = g_s(x), \quad s = 1, \dots, S \\ x^m \leq x \leq x^M \end{array}$$

où  $f$  et  $g_s$  sont des fonctions de  $R^n$  dans  $R$ , et où  $\phi_s : R \rightarrow \bar{R} = ]-\infty, +\infty]$  est une fonction convexe "étendue" (c'est-à-dire, pouvant prendre la valeur  $+\infty$ ). Rappelons que le "domaine effectif" d'une fonction convexe (étendue)  $h : R^n \rightarrow \bar{R}$  est, par définition, l'ensemble :

$$\text{dom } h = \{x \in R^n | h(x) < +\infty\}$$

Une fonction convexe (étendue)  $h : R^n \rightarrow \bar{R}$  est dite "propre" si  $\text{dom } h$  n'est pas vide. Dans les exemples 1 et 2, les fonctions  $\phi_s(e_s)$  considérées sont des fonctions convexes propres telle que :

$$\phi_s(e_s) \geq 0, \quad \forall e_s \in \text{dom } \phi_s$$

Elles sont de plus linéaires par morceaux dans leurs domaines effectifs.

**Remarque :** Par définition de  $\text{dom } \phi_s$ , on a :

$$\phi_s(e_s) = +\infty, \quad \forall e_s \notin \text{dom } \phi_s$$

#### Justification pour le problème du dispatching économique.

Hormis les contraintes de bornes inférieure et supérieure sur les variables, les contraintes du problème sont de quatre types :

$$(2.22) \quad \sum_{g \in G_i} P_{ig} = \varphi_i(\theta, V) + P_{ic}, \quad \text{si } |G_i| \geq 1$$

$$(2.23) \quad 0 = \varphi_i(\theta, V) + P_{ic}, \quad \text{si } |G_i| = 0$$

$$(2.24) \quad Q_i = \psi_i(\theta, V)$$

$$(2.25) \quad \delta_{ik} = \tau_{ik}^M - \tau_{ik}(\theta, V), \quad \delta_{ik} \geq 0$$

Posons  $x = (\theta, V) \in R^{2N}$ . A la contrainte (2.22), on peut faire correspondre la contrainte  $e_s = g_s(x)$  avec :

$$g_s(x) = \varphi_i(\theta, V) + P_{ic}$$

et la fonction  $\phi_s$  définie par :

$$\text{dom } \phi_s = \left[ \sum_{g \in G_i} P_{ig}^m, \sum_{g \in G_i} P_{ig}^M \right]$$

$$(2.26) \quad \phi_s(e_s) = \min \left\{ \sum_{g \in G_i} C_{ig}(P_{ig}) \left| \sum_{g \in G_i} P_{ig} = e_s, P_{ig}^m \leq P_{ig} \leq P_{ig}^M \quad (\forall g \in G_i) \right. \right\}$$

si  $e_s \in \text{dom } \phi_s$ .  $C_{ig}$  étant une fonction convexe linéaire par morceaux,  $\phi_s(e_s)$  (avec  $e_s \in \text{dom } \phi_s$ ) est la valeur optimale d'un programme linéaire équivalent au problème de minimisation (2.26) et ayant pour second membre  $e_s$ . D'après un résultat classique sur la paramétrisation du second membre d'un programme linéaire,  $\phi_s$  est une fonction convexe de  $e_s$ , linéaire par morceaux dans son domaine effectif.

A la contrainte (2.23), on peut faire correspondre la contrainte  $e_s = g_s(x)$  avec :

$$g_s(x) = \varphi_i(\theta, V) + P_{ic}$$

et la fonction  $\phi_s$  définie par :

$$(2.27) \quad \text{dom } \phi_s = \{0\}, \quad \phi_s(e_s) = 0 \text{ si } e_s \in \text{dom } \phi_s$$

A la contrainte (2.24), on peut faire correspondre la contrainte  $e_s = g_s(x)$  avec :

$$g_s(x) = \psi_i(\theta, V)$$

et la fonction  $\phi_s$  définie par :

$$\text{dom } \phi_s = [Q_i^m, Q_i^M], \quad \phi_s(e_s) = 0 \text{ si } e_s \in \text{dom } \phi_s$$

Enfin, à la contrainte (2.25), on peut faire correspondre la contrainte  $e_s = g_s(x)$  avec :

$$g_s(x) = \tau_{ik}^M - \tau_{ik}(\theta, V)$$

et la fonction  $\phi_s$  définie par :

$$\text{dom } \phi_s = [0, +\infty[, \quad \phi_s(e_s) = 0 \text{ si } e_s \in \text{dom } \phi_s.$$

On prendra par ailleurs  $f = 0$ .

### Justification pour le problème de l'optimisation du plan de tension

Hormis les contraintes de bornes inférieure et supérieure sur les variables, les contraintes du problème sont de quatre types :

$$(2.28) \quad 0 = \varphi_i(\theta, V) - \lambda_i p - P_i^0, \text{ si } i \in G$$

$$(2.29) \quad 0 = \varphi_i(\theta, V) - P_i^0, \text{ si } i \notin G$$

$$(2.30) \quad Q_{ig} + C_i - S_i = \psi_i(\theta, V) + Q_{ic}, \text{ si } i \in R$$

$$(2.31) \quad C_i - S_i = \psi_i(\theta, V) + Q_{ic}, \text{ si } i \notin R$$

Posons  $x = (\theta, V, p) \in R^{2N+1}$ . A la contrainte (2.28), on peut faire correspondre la contrainte  $e_s = g_s(x)$  avec :

$$g_s(x) = \varphi_i(\theta, V) - \lambda_i p - P_i^0$$

et la fonction  $\phi_s$  définie par (2.27).

A la contrainte (2.29), on peut faire correspondre la contrainte  $e_s = g_s(x)$  avec :

$$g_s(x) = \varphi_i(\theta, V) - P_i^0$$

et la fonction  $\phi_s$  définie par (2.27).

A la contrainte (2.30), on peut faire correspondre la contrainte  $e_s = g_s(x)$  avec :

$$g_s(x) = \psi_i(\theta, V) + Q_{ic}$$

et la fonction  $\phi_s$  définie par :

$$\text{dom } \phi_s = ] - \infty, +\infty[$$

$$\phi_s(e_s) = \min \{ \beta C_i + \gamma S_i \mid Q_{ig} + C_i - S_i = e_s, Q_{ig}^m \leq Q_{ig} \leq Q_{ig}^M, C_i \geq 0, S_i \geq 0 \}$$

$\phi_s$  est une fonction convexe linéaire par morceaux pour la même raison que celle donnée pour la fonction  $\phi_s$  définie par (2.26). Elle est représentée sur la figure 2.

Enfin, à la contrainte (2.31), on peut faire correspondre la contrainte  $e_s = g_s(x)$  avec :

$$g_s(x) = \psi_i(\theta, V) + Q_{ic}$$

et la fonction  $\phi_s$  définie par :

$$\text{dom } \phi_s = ] - \infty, +\infty[$$

$$\phi_s(e_s) = \min \{ \beta C_i + \gamma S_i \mid C_i - S_i = e_s, C_i \geq 0, S_i \geq 0 \}$$

Cette fonction est représentée sur la figure 3.

On prendra par ailleurs :

$$\begin{aligned} f(x) &= \alpha p \text{ si on minimise les pertes} \\ &= \alpha \sum_{i=1}^N (V_i^M - V_i) \text{ si on maximise les tensions} \end{aligned}$$

### 3. Un algorithme de type Newton pour les problèmes non linéaires de grande taille

#### 3.1. Le problème modèle. Equivalence avec un problème standard de programmation non linéaire

À la suite de la modélisation établie dans la deuxième partie de l'article, nous sommes amenés à considérer des problèmes du type :

$$(3.1) \quad \min_{x \in \mathbb{R}^n, e \in \mathbb{R}^p} f(x) + \sum_{i=1}^p \phi_i(e_i),$$

$$g(x) = e ; a \leq x \leq b,$$

où  $f$  et  $g$  sont des fonctions de  $\mathbb{R}^n$  vers  $\mathbb{R}$  et  $\mathbb{R}^p$ ,  $a$  et  $b$  sont deux vecteurs de  $(\mathbb{R} \cup \{-\infty\})^n$  et  $(\mathbb{R} \cup \{+\infty\})^n$  et  $\phi_i$ ,  $i = 1$  à  $p$  est une application de  $\mathbb{R} \rightarrow \mathbb{R} \cup \{+\infty\}$ , convexe, linéaire par morceaux et propre.

Vérifions l'équivalence de cette formulation avec la formulation standard de la programmation non linéaire :

$$(3.2) \quad \min_{x \in \mathbb{R}^n} F(x) ; h_i(x) = 0, i = 1 \text{ à } q, h_i(x) \leq 0, i = q + 1 \text{ à } p,$$

où  $F$  et  $h_i$  ( $i = 1$  à  $p$ ) sont des applications de  $\mathbb{R}^n$  vers  $\mathbb{R}$  (nous ne discutons pas le cas où il n'y a pas de contraintes d'égalité et/ou d'inégalité). La réduction de (3.2) à (3.1) est facile. Il suffit de prendre  $F = f$ ,  $g = h$ ,  $a = (-\infty)^n$ ,  $b = (+\infty)^n$  et de définir  $\phi_i$ , ( $i = 1$  à  $p$ ) par :

$$\text{dom}(\phi_i) = \{0\} \text{ et } \phi_i(t) = 0 \text{ si } t = 0, i = 1 \text{ à } q,$$

$$\text{dom}(\phi_i) = ]-\infty, 0] \text{ et } \phi_i(t) = 0 \text{ si } t \leq 0, i = q + 1 \text{ à } p.$$

La réduction de (3.1) à (3.2) s'effectue en notant que  $\phi_i$  est le maximum d'un nombre fini de fonctions linéaires :

$$\phi_i(t) = \max\{\alpha_{ij} + \beta_{ij}t ; j = 1 \text{ à } s_i\} \text{ pour } t \in \text{dom}(\phi_i),$$

$$= \min\{\gamma \in \mathbb{R} ; \alpha_{ij} + \beta_{ij}t - \gamma \leq 0, j = 1 \text{ à } s_i\}.$$

Introduisant les variables d'écart  $v_1, \dots, v_p$ , il apparaît que (3.1) équivaut à :

$$(3.3) \quad \min_{x, e, v} f(x) + \sum_{i=1}^p v_i ; g(x) = e ; e_i \in \text{dom}(\phi_i)$$

$$a \leq x \leq b ; \alpha_{ij} + \beta_{ij}e_i - v_i \leq 0, j = 1 \text{ à } s_i, i = 1 \text{ à } p$$

La contrainte  $e_i \in \text{dom}(\phi_i)$  revenant à imposer des bornes sur certaines composantes de  $e$ , nous sommes ainsi ramenés à un cas particulier de (3.2).

**Remarque :** Les deux réductions préservent la régularité des données (ordre de dérivabilité des applications non linéaires...).

### 3.2. Les méthodes de type Newton (analyse locale)

On définit classiquement (et nous renvoyons à [F (1987)], [Be (1982)], [Bo (1989)<sub>a</sub>] pour une exposition plus détaillée) les méthodes de type Newton pour le problème (3.2) de la façon suivante : à  $x \in \mathbb{R}^n$  et  $H$ , matrice  $n \times n$  symétrique, nous associons le problème quadratique :

$$Q(x, H) : \min_{d \in \mathbb{R}^n} \nabla F(x)^t d + \frac{1}{2} d^t H d ; h(x) + h'(x)d \ll 0 ,$$

où la notation  $\ll 0$  est définie par :

Si  $y \in \mathbb{R}^p$ ,  $y \ll 0$  si  $y_i = 0$ ,  $i = 1$  à  $q$  et  $y_i \leq 0$ ,  $i > q$ .

Une méthode de type Newton sans recherche linéaire est du type suivant :

#### Alg. 1 (méthode de type Newton pour le problème (3.2))

- 0) Choisir  $x^0 \in \mathbb{R}^n$  ;  $k \leftarrow 0$ .
- 1) Choisir  $H^k$ , matrice  $n \times n$  symétrique
- 2) Calculer  $d^k$ , solution locale de  $Q(x^k, H^k)$ , et le multiplicateur de Lagrange associé  $\lambda^{k+1}$ .
- 3) Si  $d^k = 0$  ou  $Q(x^k, H^k)$  ne peut être résolu, arrêt.
- 4)  $x^{k+1} \leftarrow x^k + d^k$ ,  $k \leftarrow k + 1$ , aller en 1. ■

Soit  $L(x, \lambda) := F(x) + \lambda^t h(x)$  le lagrangien associé au problème (3.2) et  $\nabla^2 L(x, \lambda)$  son hessien par rapport à  $x$

Lorsque  $H^k$  est pris égal à  $\nabla^2 L(x^k, \lambda^k)$  (ce qui suppose un choix initial soit de  $\lambda^0$ , soit de  $H^0$ ), on parle d'une méthode de Newton (sans recherche linéaire). Dans le cas où aucune contrainte d'inégalité n'est présente, on vérifie que l'algorithme équivaut à la méthode de Newton appliquée au système d'optimalité du premier ordre

Formulons une méthode de type Newton pour le problème (3.1) en utilisant le problème équivalent (3.3). Les variables  $(e, v)$  apparaissant linéairement dans (3.3), on peut donc supposer que l'approximation du terme quadratique est du type  $d^t H d$  avec  $d$  variation de  $x$  et  $H$  matrice  $n \times n$  symétrique.

Il est alors facile de vérifier que le problème quadratique associé à (3.3) se ramène à :

$$\begin{aligned} \widehat{Q}(x, H) : \min_{d \in \mathbb{R}^n, e \in \mathbb{R}^p} \nabla f(x)^t d + \frac{1}{2} d^t H d + \sum_{i=1}^p \phi_i(e_i) \\ g(x) + g'(x)d = e ; a \leq x + d \leq b. \end{aligned}$$

Les données du problème quadratique ne dépendent donc pas de l'estimation de la valeur initiale de  $e$ . Une méthode de Newton (encore définie par référence à (3.3)) revient à prendre  $H = \nabla^2 L(x, \lambda)$  où  $L(x, \lambda) := f(x) + \lambda^t g(x)$  et  $\lambda$  est le multiplicateur de Lagrange associé à la contrainte  $g(x) + g'(x)d - e = 0$  obtenue précédemment. En conclusion nous obtenons :

**Alg. 2 (méthode de type Newton pour le problème (3.1))**

Opérer comme dans Alg. 1 en remplaçant  $Q(x^k, H^k)$  par  $\widehat{Q}(x^k, H^k)$  et en désignant par  $(d^k, \widehat{e}^{k+1})$  une solution locale de  $\widehat{Q}(x^k, H^k)$ .

**Alg. 3 (méthode de Newton pour le problème (3.1))**

Opérer comme dans Alg. 2 avec  $H^k = \nabla^2 L(x^k, \lambda^k)$  où  $\lambda^k$  est le multiplicateur de Lagrange associé à  $(d^{k-1}, \widehat{e}^k)$ , correspondant à la contrainte  $g(x^{k-1}) + g'(x^{k-1})d - e = 0$ .

**3.3. Analyse de la vitesse de convergence de la méthode de Newton**

Soit  $\bar{x}$  une solution locale de (3.2) et  $\bar{\lambda}$  un multiplicateur de Lagrange associé. Considérons les hypothèses suivantes :

**LI** : Les gradients des contraintes actives en  $\bar{x}$  sont linéairement indépendants.

L'hypothèse LI assure l'existence et l'unicité du multiplicateur de Lagrange

**CS2** : Pour toute direction critique non nulle  $d$ , on a :

$$d^t \nabla^2 L(\bar{x}, \bar{\lambda}) d > 0$$

(Condition suffisante du second ordre).

Rappelons qu'une direction  $d \in \mathbb{R}^n$  est critique (pour le point  $\bar{x}$ ) si  $d$  vérifie :

$$\nabla f(\bar{x})^t d \leq 0 \text{ et } \begin{cases} h'_i(\bar{x})d = 0, & \text{si } 1 \leq i \leq q, \\ h'_i(\bar{x})d \leq 0, & \text{si } h_i(\bar{x}) = 0, i > q \end{cases}$$

**CST** : Si  $h_i(\bar{x}) = 0, i > q$ , alors  $\bar{\lambda}_i > 0$  (Complémentarité stricte).

On sait (cf. [F (1987)]) que, pour le problème standard (3.2) sous les hypothèses LI, CS2 et CST, la méthode de Newton converge de façon quadratique, la suite  $\{x^k\}$  convergeant de façon superlinéaire; on a pu établir récemment que ces propriétés restent vérifiées même en l'absence de complémentarité stricte [Bo (1989)<sub>c</sub>]. Passant encore par le problème (3.3), on peut étendre les résultats au problème (3.1). Utilisant le fait que  $\widehat{Q}(x^k, H^k)$  ne dépend pas de la valeur de  $e^k$ , on peut obtenir le théorème 1 de la page suivante.

**Théorème 1** : Soit  $\bar{x}$  solution locale de (3.1) vérifiant LI et  $\bar{\lambda}$  le multiplicateur associé, tel que CS2 soit vérifiée. Alors si  $(x^0, \lambda^0)$  est assez proche de  $(\bar{x}, \bar{\lambda})$ , la méthode de Newton est bien définie et  $(x^k, \lambda^k) \rightarrow (\bar{x}, \bar{\lambda})$  de façon quadratique, la suite  $\{x^k\}$  convergeant de manière superlinéaire. ■

### 3.4. Convergence globale

Nous rappelons le principe de la globalisation de la méthode de Newton (pour le problème (3.2)) à l'aide de la fonction de pénalisation exacte de Han-Pschenichny (cf. [H (1977)]) et nous montrons comment étendre ces principes au problème (3.1). Nous ne rentrons pas dans le détail de mise à jour des paramètres (assez fastidieux) pour lequel nous renvoyons aux articles originaux.

Soit  $h^\#(x)$  définie par :

$$h_i^\#(x) = \begin{cases} h_i(x), & i = 1 \text{ à } q, \\ h_i^+(x) = \max\{0, h_i(x)\}, & i > q \end{cases}$$

et soit  $\|\cdot\|_1$  la norme  $\ell_1$ , c'est-à-dire  $\|z\|_1 = \sum_i |z_i|$ . Soit la fonction dite de pénalisation exacte :

$$\theta_r(x) := F(x) + r \|h^\#(x)\|_1,$$

où  $r > 0$  est le coefficient de pénalisation.

**Théorème 2** (Han [H (1977)] si  $H > 0$ , Bonnans [Bo (1989)<sub>c</sub>]) : Soit  $(\bar{x}, \bar{\lambda})$  vérifiant les hypothèses du théorème 1. Alors, si  $r > |\bar{\lambda}|_\infty = \max\{|\bar{\lambda}_i|, i = 1 \text{ à } p\}$  :

(i)  $\theta_r(\bar{x})$  a un minimum local strict en  $\bar{x}$ ,

(ii) Si  $H > 0$  ou si  $H = \nabla^2 L(x, \lambda)$  et  $(x, \lambda)$  proche de  $(\bar{x}, \bar{\lambda})$  et, si  $d$  est la solution locale de  $Q(x, H)$  de norme minimale, alors la dérivée directionnelle  $\theta'_r(x, d)$  est strictement négative. ■

Pour le problème (3.1) la fonction correspondante à  $\theta_r$  est :

$$P_r(x, e) = f(x) + \phi(e) + r \|g(x) - e\|_1.$$

Nous ne pénalisons pas les contraintes de borne dans la mesure où on prend  $x$  tel que  $a \leq x \leq b$ . De même on suppose que  $e \in \text{dom}(\phi)$ . On obtient alors l'analogue du théorème 2.

On définit la "linéarisation" de  $P_r$  en  $(x^k, e^k)$  pour  $r = r^k$ , soit :

$$P^k(x, e) = f(x^k) + \nabla f(x^k)^t (x - x^k) + \phi(e) + r \|g(x^k) + g'(x^k)(x - x^k) - e\|_1.$$

## Alg. 4

- 0) Choisir  $x^0 \in \mathbb{R}^n$ ,  $k \leftarrow 0$ ,  $\beta \in (0, 1)$ ,  $\sigma \in (0, 1/2)$
- 1) Choisir  $H^k$ , matrice  $n \times n$  symétrique
- 2) Calculer  $(d^k, \hat{e}^{k+1})$ , solution locale de  $\hat{Q}(x^k, H^k)$ .
- 3) Si  $d^k = 0$ , ou si  $\hat{Q}(x^k, H^k)$  ne peut être résolu, arrêter
- 4) Recherche linéaire de type Armijo : choisir  $r = r^k > |\lambda^{k+1}|_\infty$ . Prendre le plus grand pas de la forme  $\rho = \beta^\ell$ ,  $\ell \in \mathbb{N}$ , tel que :

$$P_r(x^k, e^k) - P_r(x^k + \rho d^k, e^k + \rho(\hat{e}^{k+1} - e^k)) \geq \sigma [P_r(x^k, e^k) - P^k(x^k + \rho d^k, e^k + \rho(\hat{e}^{k+1} - e^k))]$$

$$x^{k+1} \leftarrow x^k + \rho d^k, e^{k+1} \leftarrow e^k + \rho(\hat{e}^{k+1} - e^k),$$

$$k \leftarrow k + 1, \text{ aller en 1) } \quad \blacksquare$$

Dans la recherche linéaire, on essaie d'abord de prendre un pas égal à 1. La règle revient à imposer une décroissance de  $P_r$  substantielle relativement à la décroissance du modèle linéarisé  $P^k$ .

Cette méthode souffre d'une difficulté dite de l'effet Maratos (cf. [F (1987)]) : même si  $d$  est la direction de Newton (pour (3.2)), on n'a pas nécessairement  $\theta_r(x + d) > \theta_r(x)$  même si  $(x, \lambda)$  est proche de  $(\bar{x}, \bar{\lambda})$  et si  $r$  est proche de  $|\lambda|_\infty$ .

Divers remèdes ont été proposés afin d'éviter cette difficulté (cf. [F (1987)]). La méthode de [Bo (1990)<sub>a</sub>] consiste en l'emploi de variables d'écart. Nous l'exposons brièvement. Supposant, pour simplifier les notations, que (3.2) ne comporte que des inégalités, il est clair que, pour tout  $\mu \in \mathbb{R}^p$ , (3.2) équivaut à :

$$\min_{x, e} F(x) + \mu^t (h(x) + e) ; h(x) + e = 0 ; e \geq 0,$$

problème auquel est associé la fonction de pénalisation exacte :

$$\hat{\theta}_{r, \mu}(x, e) = F(x) + \mu^t (h(x) + e) + r |h(x) + e|_1.$$

La condition de pénalisation exacte est  $r > |\mu - \bar{\lambda}|_\infty$ . L'idée de [Bo (1990)] est d'éliminer les variables d'écart en minimisant  $\hat{\theta}_{r, \mu}(x, e)$  par rapport à  $e$ . Le problème se décompose et le minimum est :

$$e_i = \begin{cases} 0 & \text{si } \mu_i > r, \\ (-h_i(x))^+ & \text{sinon.} \end{cases}$$

Soit  $I$  l'ensemble des contraintes actives en  $\bar{x}$ . Dans le cas où on a la complémentarité stricte, si  $\mu$  est "proche" de  $\bar{\lambda}$  et si  $i \in I$ , alors  $\mu_i > r$  dès que  $r$  est assez petit. Le critère de recherche linéaire se réduit alors à (si  $x$  est proche de  $\bar{x}$ ) :

$$\tilde{\theta}_{r, \mu}(x) := \min_{e \geq 0} \hat{\theta}_{r, \mu}(x, e) = F(x) + \sum_{i \in I} (\mu_i h_i(x) + r |h_i(x)|),$$



qui s'interprète comme une fonction de pénalisation exacte associée au problème :

$$\min F(x) ; h_I(x) = 0 .$$

Ainsi l'algorithme se ramène localement à un algorithme de résolution de problèmes d'optimisation avec contraintes d'égalité. On montre alors que, pour  $\mu$  proche de  $\bar{\lambda}$  et  $r$  assez petit, l'effet Maratos disparaît (voir [Bo (1989)<sub>b</sub>]).

L'extension des idées précédentes au problème (3.1) amène à considérer le critère :

$$\widehat{P}_{r,\mu}(x, e) := f(x) + \phi(e) + \mu^t (g(x) - e) + r |g(x) - e|_1 ,$$

et

$$\tilde{P}_{r,\mu}(x) := \min \left\{ \widehat{P}_{r,\mu}(x, e) ; e \in \text{dom}(\phi) \right\}$$

Nous considérons une condition (extension de la condition de complémentarité stricte) d'intériorité suivante :

$$(3.4) \quad \text{Si } \bar{e}_i \text{ est un point de discontinuité de } \phi_i, \text{ alors } \bar{\lambda}_i \in \text{Intérieur } \partial\phi_i(\bar{e}_i).$$

La minimisation de  $\widehat{P}_{r,\mu}(x, e)$  par rapport à  $e$  revient à minimiser pour tout  $i$  la fonction convexe :

$$e_i \rightarrow \phi_i(e_i) - \mu_i e_i + r |g_i(x) - e_i| ,$$

et le minimum  $\hat{e}_i$  est caractérisé par :

$$\partial\phi_i(\hat{e}_i) - \mu_i - r \partial|\cdot|(g_i(x) - \hat{e}_i) \ni 0$$

Distinguons deux cas :

- a)  $\phi_i$  dérivable en  $\bar{e}_i$ . Alors  $\bar{\lambda}_i = \phi'_i(\bar{e}_i)$  localement;  $\hat{e}_i$  satisfait  $\bar{\lambda}_i - \mu_i \in r \partial|\cdot|(g_i(x) - \hat{e}_i)$ . Si  $r > |\bar{\lambda}_i - \mu_i|$ , la solution est  $\hat{e}_i = g_i(x)$ .
- b)  $\phi_i$  non dérivable en  $\bar{e}_i$ . Si (3.4) est vérifiée, soit  $\alpha_i := \text{dist}(\bar{\lambda}_i, \text{Frontière } \partial\phi_i(\bar{e}_i))$ . Si  $|\bar{\lambda}_i - \mu_i| \leq \alpha_i/2$  et  $r \leq \alpha_i/2$ , alors  $\partial\phi_i(\bar{e}_i) - \mu_i + [-r, +r] \ni 0$ , donc la solution est  $\hat{e}_i = \bar{e}_i$ .

Finalement, la contribution du  $i^{\text{ème}}$  terme au critère est, sous les hypothèses ci-dessus :

$\bar{\lambda}_i g_i(x)$  dans le cas a)

$\mu_i (g_i(x) - \bar{e}_i) + r |g_i(x) - \bar{e}_i|$  dans le cas b).

**Théorème 3** : Soit  $\bar{x}$ ,  $\bar{e} = g(\bar{x})$  solution locale de (3.1). On suppose les gradients  $\{\nabla g_i(\bar{x}) ; i \text{ tel que } \phi_i \text{ non dérivable en } g_i(\bar{x})\}$  linéairement indépendants (d'où l'unicité du multiplicateur  $\bar{\lambda}$ ). Si (3.4) est vérifiée et si  $(x, \mu)$  est assez proche de  $(\bar{x}, \bar{\lambda})$ , pour  $r > |\bar{\lambda} - \mu|_\infty$  assez petit, on a, à une constante additive près :

$$\tilde{P}_{r,\mu}(x) = f(x) + \sum_{i \in I_1} \bar{\lambda}_i g_i(x) + \sum_{i \in I_2} \left( \mu_i (g_i(x) - \bar{e}_i) + r |g_i(x) - \bar{e}_i| \right)$$

avec :

$$I_1 \text{ (resp. } I_2) = \left\{ i ; \phi_i \text{ dérivable (resp. non dérivable) en } g_i(\bar{x}) \right\}. \quad \blacksquare$$

Ainsi (localement) tout se passe comme si on avait des contraintes d'égalité seulement. La convergence globale ainsi que l'ajustement des paramètres afin d'éviter l'effet Maratos sont alors obtenus en procédant comme dans [Bo (1989)<sub>b</sub>]. L'algorithme s'écrit de la manière suivante :

**Alg. 5** Procéder comme dans **Alg. 4** en remplaçant l'étape 4 par :

4 bis) Choisir  $r = r^k > |\lambda^{k+1}|_\infty$  et une approximation du multiplicateur  $\mu = \mu^k$ .

Poser :

$$\tilde{P}^k(x) := \min \left\{ P^k(x, e) + \mu^t \left( g(x^k) + g'(x^k)(x - x^k) - e \right) ; e \in \text{dom}(\phi) \right\}.$$

Effectuer la recherche linéaire de type Armijo comme dans **Alg. 4** en remplaçant  $P_r$  par  $\tilde{P}_{r,\mu}$  et  $P^k$  par  $\tilde{P}^k$ .

$$x^{k+1} \leftarrow x^k + \rho d^k,$$

$$e^{k+1} = \arg \min \left\{ \hat{P}_{r,\mu}(x^{k+1}, e) ; e \in \text{dom}(\phi) \right\},$$

$k \leftarrow k + 1$ , aller en 1).  $\blacksquare$

**Remarque** : L'algorithme ci-dessus peut s'arrêter du fait que, si  $H^k$  n'est pas définie positive, la direction de déplacement n'est pas nécessairement une direction de descente de la fonction de pénalisation exacte envisagée précédemment. L'algorithme relance alors le programme quadratique en ajoutant à  $H^k$  un multiple convenable de l'identité. Procédant ainsi plusieurs fois si nécessaire, on finit par obtenir une matrice définie positive, qui donne donc nécessairement une direction de descente.

### 3.5. Résolution du problème quadratique par une méthode de type gradient réduit conjugué

Cette section présente l'algorithme employé pour la résolution du problème quadratique  $\hat{Q}(x, H)$ . Le principe en est de traiter les contraintes linéaires par une méthode de réduction de variables, avec pivotage lorsqu'une variable de base se bloque. On effectue alors une descente par une méthode de type gradient réduit conjugué (avec recherche linéaire exacte) afin d'obtenir une terminaison finie. La présence des fonctions  $\phi_i$  nécessite toutefois certains aménagements de l'algorithme.

### 3.5.1. Le gradient réduit

A chaque étape de l'algorithme, nous choisissons  $p$  composantes de  $(d, e)$  de manière à satisfaire les contraintes linéaires :

$$g(x) + g'(x)d - e = 0.$$

Plus précisément, posant  $A = (g'(x), -I)$ , on choisit un sous-ensemble de  $p$  colonnes de  $A$  de rang  $p$  (ceci est toujours possible : il suffit de prendre les  $p$  dernières colonnes). Les  $p$  variables associées (la base) sont considérées comme fonctions affines des  $n$  autres de sorte que les équations soient toujours vérifiées. On supposera que (au début de chaque itération de l'algorithme) les variables de base sont hors de leurs bornes (pour  $d$ ) ou des points de discontinuité de  $\phi'_i$  (pour  $e_i$ ) (hypothèse de non dégénérescence). Posons :

$$z = (d, e),$$

$$B = \text{ensemble des indices des variables de base (à } e_i \text{ est associé l'indice } n + i),$$

$$N = (1, n + p) - B \text{ (indices des variables hors base),}$$

$$A_B, A_N \text{ colonnes de } A \text{ correspondant à } B, N.$$

On note :

$$z = (z_B, z_N), \quad z_B = (d_B, e_B), \text{ etc...}$$

Localement les variables de base sont associées à des "coûts" :

$$c_i = \begin{cases} \partial f(x) / \partial x_i & i \leq n, \\ \phi'_i(e_i) & i > n, \end{cases} \quad \forall i \in B$$

Soit  $c_B$  le vecteur ainsi défini et  $\lambda \in \mathbb{R}^p$  l'estimation du multiplicateur définie par :

$$-A_B^t \lambda = c_B.$$

Pour les composantes hors base de  $d$ , on peut définir le gradient réduit :

$$\nabla_N f(x) + A_N^t \lambda,$$

qui est bien le gradient par rapport à  $x_N$  du coût réduit :

$$z_N \rightarrow \hat{f}(z_N) := f(d_B(z_N), d_N) + \sum_{i \in I_1} \phi_i(e_i(z_N)) + \sum_{i \in I_2} \phi_i(e_i),$$

avec :

$$I_1 = \{i \in (1, p) ; n+i \in B\},$$

$$I_2 = (1, \dots, p) - I_1$$

Pour les composantes hors base de  $e$ , il faut distinguer suivant que le déplacement de  $e$  est positif ou négatif (en raison de la non différentiabilité de  $\phi_i$ ). On a les dérivées directionnelles :

$$\widehat{f}'(z_n, \delta z_N) = \left( \nabla_N f(x) + A_N^t \lambda \right)^t \delta d_N + \sum_{i \in I_2} \left( \phi'_i(e_i, \delta e_i) - \lambda_i \delta e_i \right)$$

où le terme  $-\lambda_i \delta e_i$  représente la contribution de la base à la dérivée directionnelle; le terme  $\phi'_i(e_i, \delta e_i)$  est facile à déterminer. Pour chaque  $i \in I_2$ , si  $\phi_i$  n'est pas différentiable en  $e_i$ , trois situations peuvent se produire :

- a)  $\phi'_i(e_i, \delta e_i) - \lambda_i \delta e_i \geq 0$ ,  $\forall \delta e_i \in \mathbb{R}$ , c'est-à-dire,  $\phi_i^d(e_i) \geq \lambda_i$  et  $\phi_i^g(e_i) \geq \lambda_i$ , où  $\phi_i^d, \phi_i^g$  sont les dérivées directionnelles à droite et à gauche. Dans ce cas on a intérêt à laisser  $e_i$  bloqué.
- b)  $\phi'_i(e_i, \delta e_i) - \lambda_i \delta e_i$  est du signe de  $\delta e_i$ . Alors on choisit de prendre un déplacement de  $e_i \leq 0$
- c)  $\phi'_i(e_i, \delta e_i) - \lambda_i \delta e_i$  est du signe opposé à  $\delta e_i$ . On prendra un déplacement de  $e_i$  positif

Notons que la convexité de  $\phi_i$  empêche d'avoir une dérivée strictement négative à la fois à droite et à gauche. On obtient alors l'existence de  $g$  tel que :

$$\widehat{f}(z_N + \delta z_N) = \widehat{g}^t \delta z_N + o(\delta z_N),$$

étant entendu que  $\widehat{g}_i \delta z_i \leq 0$ ,  $\forall i \in I_2$  tel que  $\phi_i$  n'est pas différentiable en  $e_i$

### 3.5.2. Choix de la direction de descente et recherche linéaire

Pour la direction de descente, il y a le choix évident (direction opposée au gradient projeté) :

$$\delta z_i = \begin{cases} -\widehat{g}_i & \text{si } z - \varepsilon \widehat{g}_i \text{ est admissible pour } \varepsilon > 0 \text{ petit,} \\ 0 & \text{sinon.} \end{cases}$$

A partir d'une direction de descente, l'algorithme effectue une recherche linéaire de la façon suivante.

Le coût en  $z_N + \rho \delta z_N$  est pour  $\rho > 0$  une fonction quadratique par morceaux et de courbure constante (convexe si  $\delta x_N^t H \delta x_N \geq 0$ , ce qui n'est pas vrai en général). Partant d'un pas nul, l'algorithme analyse les intervalles successifs entre deux points de discontinuité de la dérivée et calcule ainsi la première valeur stationnaire de  $\rho$ , qui est adoptée comme pas de l'itération courante.

Si ce pas est interne au premier intervalle analysé, aucune discontinuité de  $\phi'_i$  n'est intervenue lors du déplacement. Dans ce cas, si l'ensemble des variables superbasiques (non bloquées) est inchangé, et si la direction de descente est celle opposée au gradient réduit, on peut *conjuguer* les directions de façon à accélérer la descente. Le processus doit alors, soit converger en un nombre fini d'itérations de gradient conjugué, soit aboutir à un changement de l'ensemble des superbasiques.

En supposant donné un point de départ réalisable, l'algorithme s'écrit alors de la façon suivante (l'indicateur conj, égal à 0 ou 1, indique si on peut conjuguer) :

- 0) Choisir  $x^0, e^0$ , les ensembles  $B^0$  et  $N^0$  ;  $k \leftarrow 0$ ;
- 1) Déterminer le gradient réduit  $\hat{g}$  et l'ensemble  $S^k \subset N$  des superbasiques. Si  $\hat{g} = 0$ , arrêt.
- 2) Calcul de la direction de descente.

Prendre  $\delta z^k$  défini par :

$$\delta z^k = \begin{cases} -\hat{g}_i & \text{si } i \in S^k, \\ 0 & \text{sinon,} \end{cases} \quad \forall i \in N.$$

( $S^k$  est tel que  $\delta z^k$  n'est pas nul).

Si  $k \geq 1$ ,  $S^k = S^{k-1}$  et conj = 1, prendre :

$$\delta z_N^k \leftarrow \delta z_N^k + \beta^k \delta z_N^{k-1},$$

où le scalaire  $\beta^k$  assure la conjugaison des directions.

- 3)  $z^{k+1} \leftarrow z^k + \rho^k \delta z^k$ , où  $\rho^k$  est le premier pas stationnaire.

Si le critère est quadratique pour  $\rho \in [0, \rho^k + \varepsilon]$  ( $\varepsilon > 0$ ), alors conj  $\leftarrow 1$ ; sinon conj  $\leftarrow 0$ .

- 4) Pivotage : Si une variable de base a atteint une borne ou un point de discontinuité de la fonction  $\phi_i$  correspondante, échanger cette variable avec une superbasique de sorte que la matrice de base reste inversible (donc mettre à jour  $B, N$  et les matrices correspondantes)
- 5)  $k \leftarrow k + 1$ , aller en 1.

Le choix de l'ensemble  $S^k$  est critique pour la performance numérique. Le choix de l'ensemble maximal (variables hors base non bloquées par le gradient réduit) s'est avéré très peu performant. Il semble plus efficace de libérer parcimonieusement les variables de manière à obtenir plus rapidement une bonne approximation de l'ensemble des contraintes actives. A cette fin nous réalisons une succession de "cycles", ensembles d'itérations à l'intérieur desquels aucune variable

n'est libérée (c'est-à-dire  $S^k$  décroît de façon monotone). Si la fin d'un cycle est décidée quand  $\delta z^k = 0$ , il est facile d'obtenir la convergence globale. En effet, chaque fin de cycle correspond à la résolution exacte d'un sous-problème dans lequel un sous-ensemble des variables est bloqué sur un nombre fini de points; il existe donc un nombre fini de sous-problèmes. La décroissance monotone du critère assure donc que l'algorithme se termine après un nombre fini de cycles. Il suffit donc de libérer au moins une variable susceptible de permettre la descente à chaque fin de cycle pour obtenir la convergence en un nombre fini d'itérations. Dans notre implémentation, la fin de cycle est décidée si le gradient réduit des superbasiques est (relativement) petit par rapport au gradient réduit projeté.

De plus une variable non basique n'est libérée que si son gradient réduit projeté n'est pas trop petit. Enfin le nombre de variables libérées est contrôlé. Jouant sur les paramètres, l'utilisateur peut ainsi imposer de libérer au plus une variable. Il est intéressant de noter qu'alors l'algorithme se ramène (si le hessien est nul ou négatif) à celui du simplexe.

### 3.5.3. Traitement des non-admissibilités

En général, il n'est pas possible de fournir un point d'entrée admissible. On peut en calculer un par des techniques analogues à la "phase 1" de la programmation linéaire. Nous avons choisi de minimiser directement la fonction de pénalisation exacte (cf. [Ba (1980)]); cette fonction est encore quadratique par morceaux et se minimise avec les techniques exposées précédemment.

## 4. Résultats et applications numériques

L'algorithme a été mis au point et testé tout d'abord sur des réseaux d'étude de 22 sommets, puis 193 sommets, enfin sur un réseau de grande taille (838 sommets). Nous allons décrire et examiner les résultats obtenus sur ce dernier réseau.

### 4.1. Description du réseau

C'est un réseau à THT (400-225 kV) comprenant :

838 sommets,

1 171 lignes,

159 transformateurs.

La consommation est de 65 000 MW.

La production est assurée par 168 groupes thermiques et des groupes hydrauliques raccordés en 67 sommets du réseau.

On dispose, en outre, de 443 sources de réactif se décomposant en :

- 168 groupes thermiques,
- 67 sommets de production hydraulique
- 42 compensateurs synchrones,
- 166 sommets de raccordement de condensateurs ou selfs.

#### 4.2. Description du problème

Le problème traité sera l'optimisation du plan de tension dans deux configurations particulières.

Le réseau considéré est, en fait, un réseau de planification étudié à l'année  $(N+5)$ , ce qui signifie que les charges réactives sont très insuffisamment compensées et un programme de load-flow classique (calcul des transits connaissant les puissances) ne peut fournir de solution et diverge, parce que, en certains sommets, la limite de puissance transmissible est dépassée.

Nous allons donc d'abord effectuer une étude, classique en planification, qui consiste à déterminer les moyens de compensation supplémentaires à installer sur le réseau pour assurer un fonctionnement dans les limites de tension admissibles.

Ce sera notre première étude sur le réseau décrit ci-dessus que nous appellerons "réseau non compensé".

Dans un deuxième temps, nous supposerons que les charges réactives de ce réseau ont été préalablement compensées et nous effectuerons sur ce réseau, que nous appellerons "réseau compensé" une étude de type exploitation : à partir d'un plan de tension "plat" (même tension initiale en tous les nœuds) nous déterminerons la meilleure répartition des puissances réactives.

A chaque fois, nous testerons deux critères d'optimisation :

- la minimisation des pertes actives, d'une part,
- la maximisation du plan de tension, d'autre part.

Dans les deux problèmes, les puissances actives sont supposées fixées. Les variables sont :

- les tensions en phase et module :  $\theta, V$  en chaque sommet, soit 1 676 variables,
- les pertes actives sur le réseau : 1 variable,
- les productions réactives des sources de réactif, soit 443 variables,
- les variables artificielles  $C$  et  $S$  en chaque sommet, soit 1 676 variables.

Donc un total de 3 796 variables. Le problème comporte :

- 1 676 contraintes d'égalité,
- des contraintes de bornes inférieure et supérieure sur les modules de tension et les productions réactives des sources de réactif,
- des contraintes de positivité sur les variables artificielles.

### 4.3. Résultats

Les principaux résultats figurent dans les tableaux 1 à 4 dans lesquels :

- PQD : est le numéro du programme quadratique.
- NIT : le nombre d'itérations de chaque programme quadratique.
- NICG : le nombre de calculs de gradients réduits.
- NIRL : le nombre d'étapes dans la recherche linéaire.
- FACT : le nombre de changements de base effectués dans chaque programme quadratique.
- CONJ : le nombre de conjugaisons des directions dans l'algorithme du gradient conjugué.
- CYCL : le nombre de cycles. Nous avons adopté une stratégie de libération des variables hors base qui consiste à ne rendre superbasique qu'une partie des variables dont le gradient projeté est non nul, les autres variables restant bloquées. Un cycle est la résolution du programme quadratique dans lequel on ne remet pas en cause cette partition.
- PAS : le pas de la recherche linéaire d'ARMIJO.
- CRITERE : la valeur, après chaque programme quadratique, de la fonctionnelle  $\hat{P}_{r,\mu}(x, e)$ .
- RESIDU : la mesure en norme  $\ell_1$  des conditions d'optimalité. Cette mesure donne l'ordre de grandeur de la distance à l'optimum.
- RESIDU se compose de quatre parties :
- RESI 1 : norme  $\ell_1$  de la condition d'optimalité concernant les gradients du critère et des contraintes actives
- RESI 2 : somme des écarts des variables hors base aux bornes prédites comme étant actives.
- RESI 3 : norme  $\ell_1$  de  $[g(x) - e]$ .
- RESI 4 : distance en norme  $\ell_1$  des multiplicateurs de Lagrange au sous-différentiel de  $\phi$

Les principaux points à noter sont :



- la très rapide décroissance du critère,
- la qualité de la solution obtenue (valeur de RESISU),
- l'efficacité de la stratégie de libération des variables : en effet, le programme quadratique tend à débloquer trop de variables, ce qui gêne l'identification des contraintes actives,
- la mise en œuvre effective de la conjugaison des directions, de façon plus nette évidemment pour le réseau non compensé qui comporte 42 variables superbasiques à l'optimum, alors que le réseau compensé n'en comporte que 4

Tous ces résultats ont pu être comparés avantageusement à ceux obtenus par un algorithme de programmation linéaire successive disponible à EDF [B-D-M (1983)]; en particulier, les temps de calcul sont divisés par 2.

Les temps de calcul indiqués dans les tableaux sont sur IBM 3090.

TABLEAU 1 : Réseau 838 sommets non compensé avec minimisation des pertes

PQD	NIT	NICG	NIRL	FACT	CONJ	CYCL	PAS	CRITERE	RESI1	RESI2	RESI3	RESI4	RESIDU
								117 617.5	10.0	0.5	3.1	0.0	13.6
1	88	88	172	68	0	18	0.50	97 489.8	284.5	57.5	39.5	3.3	384.8
2	69	69	72	4	49	6	1.00	78 310.7	882.6	0.0	26.3	1.7	910.6
3	79	79	78	3	67	3	1.00	77 795.5	177.8	0.0	0.7	0.3	178.7
4	6	6	5	0	3	0	1.00	77 766.6	4.0	0.0	0.0	0.3	4.2
5	1	1	0	0	0	0	1.00	77 766.6	0.1	0.0	0.0	0.3	0.4

Temps de calcul : 19.392 secondes.

TABLEAU 2 : Réseau 838 sommets compensé avec minimisation des pertes

PQD	NIT	NICG	NIRL	FACT	CONJ	CYCL	PAS	CRITERE	RESI1	RESI2	RESI3	RESI4	RESIDU
								25 095.3	10.0	5.1	0.7	0.0	15.8
1	90	90	393	56	0	10	0.50	19 480.9	47.4	3.7	7.2	13.9	72.2
2	65	65	87	27	0	13	1.00	13 656.7	136.7	0.0	5.2	1.5	143.4
3	24	24	23	3	9	3	1.00	13 555.3	9.9	0.0	0.2	0.1	10.2
4	1	1	0	0	0	0	1.00	13 549.8	0.1	0.0	0.0	0.1	0.2
5	1	1	0	0	0	0	1.00	13 549.8	0.0	0.0	0.0	0.1	0.2

Temps de calcul : 16.308 secondes

TABLEAU 3 : Réseau 838 sommets non compensé avec maximisation des tensions

PQD	NIT	NICG	NIRL	FACT	CONJ	CYCL	PAS	CRITERE	RESI1	RESI2	RESI3	RESI4	RESIDU
								125 875.9	1 568.0	0.5	3.1	0.0	1 571.6
1	84	84	152	70	0	17	0.50	105 890.7	10.9	103.2	55.6	9.9	179.6
2	145	145	151	11	101	10	1.00	86 298.0	860.5	0.0	29.6	1.4	891.5
3	125	125	128	5	101	7	1.00	85 664.6	77.6	0.0	0.3	0.3	78.2
4	4	4	3	0	1	1	1.00	85 577.9	0.3	0.0	0.0	0.3	0.6
5	1	1	0	0	0	0	1.00	85 577.9	0.0	0.0	0.0	0.3	0.3

Temps de calcul : 26.417 secondes.

TABLEAU 4 : Réseau 838 sommets compensé avec maximisation des tensions

PQD	NIT	NICG	NIRL	FACT	CONJ	CYCL	PAS	CRITERE	RESI1	RESI2	RESI3	RESI4	RESIDU
								31 402.3	1 572.0	5.1	0.7	0.0	1 577.8
1	102	102	375	61	0	17	0.50	26 496.5	0.0	1.6	5.9	14.9	22.3
2	47	47	77	20	0	7	1.00	21 305.9	119.0	0.0	5.4	0.0	124.4
3	1	1	0	0	0	0	1.00	21 204.7	6.3	0.0	0.0	0.0	6.3
4	1	1	0	0	0	0	1.00	21 204.2	0.0	0.0	0.0	0.0	0.0

Temps de calcul : 13.729 secondes

### 5. Annexe : Expression de l'intensité du courant qui circule dans une branche du réseau

Une branche du réseau (une ligne ou un transformateur) peut être représentée par son schéma en  $\pi$  (cf. figure 4). Sur ce schéma,  $\bar{y}'_{ik}$  est l'admittance en dérivation côté  $i$ , reliant le sommet  $i$  au point neutre ("la terre"), et  $\bar{y}'_{ki}$  est l'admittance en dérivation côté  $k$ , reliant le sommet  $k$  au point neutre.

On pose :

$$\bar{y}'_{ik} = y'_{ik} \exp(-j \xi'_{ik}), \quad \bar{y}'_{ki} = y'_{ki} \exp(-j \xi'_{ki}).$$

La puissance active  $P_{ik}$  et la puissance réactive  $Q_{ik}$ , qui circulent dans la branche reliant les sommets  $i$  et  $k$ , ont pour expression, côté  $i$  :

$$P_{ik} = (y_{ik} \cos \xi_{ik} + y'_{ik} \cos \xi'_{ik}) V_i^2 - y_{ik} V_i V_k \cos(\theta_i - \theta_k + \xi_{ik})$$

$$Q_{ik} = (y_{ik} \sin \xi_{ik} + y'_{ik} \sin \xi'_{ik}) V_i^2 - y_{ik} V_i V_k \sin(\theta_i - \theta_k + \xi_{ik})$$

Côté  $k$ , elles s'écrivent :

$$P_{ki} = (y_{ik} \cos \xi_{ik} + y'_{ki} \cos \xi'_{ki}) V_k^2 - y_{ik} V_i V_k \cos(\theta_k - \theta_i + \xi_{ik})$$

$$Q_{ki} = (y_{ik} \sin \xi_{ik} + y'_{ki} \sin \xi'_{ki}) V_k^2 - y_{ik} V_i V_k \sin(\theta_k - \theta_i + \xi_{ik})$$

Désignons par  $T_{ik}$  l'intensité (mesurée côté  $i$ ) du courant qui circule dans la branche reliant les nœuds  $i$  et  $k$ , et par  $T_{ki}$  l'intensité (mesurée côté  $k$ ) de ce même courant. On a :

$$T_{ik} = \left( (P_{ik}^2 + Q_{ik}^2) / V_i^2 \right)^{1/2}, \quad T_{ki} = \left( (P_{ki}^2 + Q_{ki}^2) / V_k^2 \right)^{1/2}$$

On posera :

$$\tau_{ik}(\theta, V) = T_{ik}, \quad \tau_{ki}(\theta, V) = T_{ki}$$

## BIBLIOGRAPHIE

## 1. Optimisation des réseaux électriques

- [A-C (1969)]     **ABADIE (J.), CARPENTIER (J.)**  
 "Generalization of the Wolfe reduced gradient method to the case of non linear constraints", dans : *Optimization*, R. Fletcher (ed ), Academic Press, London, p. 37-49.
- [A-K (1985)]     **AOKI (K.), KANEZASHI (M.)**  
 "A modified Newton method for optimal power flow using quadratic approximated power flow", *IEEE Trans. on PAS*, Vol PAS 104, n° 8, p 2 119-2 125.
- [A-S (1975)]     **ALSAC (O.), STOTT (B.)**  
 "Decoupled algorithms on optimal load flow calculations", *IEEE PES Summer Meeting, Paper A75 545-4, San Francisco.*
- [A-S (1983)]     **ALSAC (O.), STOTT (B.)**  
 "Experience with successive linear programming for optimal rescheduling of active and reactive power", *International Symposium on Control Applications for Power Systems, Florence.*
- [B-D-M (1983)]   **BLANCHON (G.), DODU (J.C.), MERLIN (A.)**  
 "Developing a new tool for real-time control in order to coordinate the regulation of reactive power and the voltage schedule in large-scale EHV power systems", *International Symposium on Control Applications for Power Systems, Florence*
- [B-H-V (1984)]   **BURCHETT (R.C.), HAPP (H.H.), VIERATH (D.R.)**  
 "Quadratically convergent optimal power flow", *IEEE PES Winter Meeting, Paper 84 WM 045-1, Dallas, Texas.*
- [B-H-W (1982)]   **BURCHETT (R.C.), HAPP (H.H.), WIRGAU (K.A.)**  
 "Large scale optimal power flow", *IEEE PES Winter Meeting, Paper 82 WM 065-1, New-York.*
- [C (1962)]       **CARPENTIER (J.)**  
 "Contribution à l'étude du dispatching économique", *Bulletin de la Société Française des Electriciens, Série 8, Vol 3.*

- [C (1973)]            **CARPENTIER (J.)**  
 “Differential injections method : A general method for secure and optimal load flow”, *Proceedings of the IEEE PICA Conference, Minneapolis, p. 255-262.*
- [C et al. (1975)]    **CARPENTIER (J.), SAMINADEN (V.), BOULLE (D.), GIRARD (R.), NGUYEN (V.T.)**  
 “Real and reactive decoupling possibilities in optimal load flows : A compact version of the differential injections method”, *Proceedings of the fifth Power System Computation Conference, Cambridge.*
- [C et al (1987)]    **COVA (B.), GRANELLI (G.P.), MONTAGNA (M.), SILVESTRI (A.), INNORTA (M.), MARANNINO (P.)**  
 “Large-scale application of the Han-Powell algorithm to compact models of static and dynamic dispatch of real power”, *Int. J. of Electrical Power and Energy Systems, Vol. 9, n° 3, p. 130-141.*
- [D-T (1968)]        **DOMMEL (H.), TINNEY (W.F.)**  
 “Optimal power flow solutions”, *IEEE Trans. on PAS, Vol PAS-87, n° 10, p. 1 866-1 876.*
- [D et al. (1967)]    **DOPAZZO (J.F.), KLITIN (O.A.), SIAGG (G.W.), WAISON (M.)**  
 “An optimization technique for real and reactive power allocation”, *Proceedings of the IEEE, Vol. 55, n° 11, p. 1 877-1 885.*
- [F-I-M (1983)]     **FRANCHI (L.), INNORTA (M.), MARANNINO (P.)**  
 “The Han-Powell algorithm applied to the optimization of the reactive generation in a large-scale electric power system”, *IFAC Symp. Large Scale Systems : Theory and Applications, Warsaw, p. 178-184.*
- [F et al. (1983)]    **FRANCHI (L.), INNORTA (M.), MARANNINO (P.), SABELLI (C.)**  
 “Evaluation of economy and/or security oriented objective functions for reactive power scheduling in large scale systems”, *IEEE Trans. on PAS, Vol. PAS-102, n° 10, p. 3 481 -3 487.*
- [F et al. (1982)]    **FERNANDES (R.A.), LANGE (F.), BURCHEIT (R.C.), HAPP (H.H.), WIRGAU (K.A.)**  
 “Large scale reactive power planning”, *IEEE PES Summer Meeting, Paper 82 SM 376-2, San Francisco.*

- [G-T (1981)] **GIRAS (T.C.), TALUKDAR (S.N.)**  
"Quasi-Newton method for optimal power flows", *Int. J. of Electrical Power and Energy Systems*, Vol. 3, n° 2, p. 59-64.
- [L-C (1981)] **LIPOWSKY (J.S.), CHARALEMBOUS (C.)**  
"Solution of optimal load flow problem by modified recursive quadratic programming method", *IEE Proc.*, Vol. 28 prt C, n° 5, p. 288-294.
- [M-F (1987)] **MARIA (G.A.), FINDLAY (J.A.)**  
"A Newton optimal power flow program for Ontario Hydro EMS", *IEEE Trans. on Power Systems*, Vol. PWRS-2, n° 3, p. 576-584.
- [N (1983)] **NADIRA (R.)**  
"Optimal reactive power flows for power system operation", *M.S. Thesis, Case Western Reserve University, Cleveland, Ohio.*
- [P-B-H (1971)] **PESCHON (J.), BREE (D.W.), HADJU (L.P.)**  
"Optimal solutions involving system security", *Proceedings of the IEEE PICA Conference, Boston*, p. 210-218.
- [P-B-H (1972)] **PESCHON (J.), BREE (D.W.), HADJU (L.P.)**  
"Optimal power flow solutions for power system planning", *Proceedings of the IEEE*, Vol. 60, n° 1, p. 64-70.
- [P et al. (1968)] **PESCHON (J.), PIERCY (D.S.), TINNEY (W.F.), TVEII (O.J.), CUENOD (M.)**  
"Optimal control reactive power flow", *IEEE Trans. on PAS*, Vol. PAS-87, n° 1, p. 40-48.
- [S (1969)<sub>a</sub>] **SASSON (A.)**  
"Nonlinear programming solutions for load-flow minimum-loss and economic dispatch problem", *IEEE Trans. on PAS*, Vol. PAS-88, n° 4, p. 399-409.
- [S (1969)<sub>b</sub>] **SASSON (A.)**  
"Combined use of the Powell and Fletcher-Powell nonlinear programming methods for optimal load flows", *IEEE Trans. on PAS*, Vol. PAS-88, n° 10, p. 1 530-1 537.

- [S-V-A. (1973)] **SASSON (A.)**, **VILORIA (F.)**, **ABOYTES (F.)**  
“Optimal load-flow solution using the hessian matrix”, *IEEE Trans. on PAS*,  
*Vol. PAS-92*, n° 1, p. 31-41.
- [S et al. (1984)] **SUN (D.I.)**, **ASHLEY (B.)**, **BREWER (B.)**, **HUGHES (A.)**, **TINNEY (W.F.)**  
“Optimal power flow by Newton approach”, *IEEE PES Winter Meeting, Paper*  
*84 WM 044-4*, Dallas, Texas.
- [T-G (1982)] **TALUKDAR (S.N.)**, **GIRAS (T.C.)**  
“A fast and robust variable metric method for optimum power flows”, *IEEE*  
*Trans. on PAS*, *Vol. PAS-101*, n° 2, p. 415-420
- [T-G-K (1983)] **TALUKDAR (S.N.)**, **GIRAS (T.C.)**, **KALYAN (V.K.)**  
“Decompositions for optimal power flows”, *Proc. of the IEEE PICA Confe-*  
*rence, Houston*, p. 20-26.
- [VA (1988)] **VAN AMERONGEN (R.A.M.)**  
“Optimal power flow solved with sequential reduced quadratic programming”,  
*Archiv für Elektrotechnik* 71, p. 213-219.

## 2. Algorithmes de programmation non linéaire

- [Ba (1980)] **BARIELS (R.H.)**  
“A penalty linear programming method using reduced gradient basis exchange  
techniques”, *Lin. Alg. Appl.* 29, p. 17-32.
- [Be (1982)] **BERTSEKAS (D.)**  
“Constrained optimization and Lagrange multiplier methods”, *Academic*  
*Press, New-York*.
- [Bo (1989)<sub>a</sub>] **BONNANS (J.F.)**  
“An introduction to Newton type algorithms for nonlinearly constrained op-  
timization problems”, in : *New methods of optimization and their industrial*  
*use*, J.P. Penot ed., Birkhäuser Verlag, p. 1-17.
- [Bo (1989)<sub>b</sub>] **BONNANS (J.F.)**  
“Asymptotic admissibility of the unit stepsize in exact penalty methods”,  
*SIAM J. Control Optim.* 27, p. 631-641.



- [Bo (1989)<sub>c</sub>] **BONNANS (J.F.)**  
“Local study of Newton type algorithms for constrained problems”, *Lecture Notes in Mathematics 1405*, S. Dolecki ed., Springer-Verlag, p. 13-24.
- [Bo (1990)<sub>a</sub>] **BONNANS (J.F.)**  
*A paraître (extension de [Bo (1989)<sub>b</sub>] aux inégalités).*
- [Bo (1990)<sub>b</sub>] **BONNANS (J.F.)**  
“Rates of convergence of Newton type methods for variational inequalities and nonlinear programming”, *Rapport INRIA*, à paraître.
- [Bo (1990)<sub>c</sub>] **BONNANS (J.F.)**  
“Théorie de la pénalisation exacte”, *M<sup>2</sup>AN (Modélisation Mathématique et Analyse Numérique) 24*, à paraître.
- [F (1983)] **FLEICHER (R.)**  
“Penalty functions”, in : *Mathematical Programming : the state of the art*, A. Bachem et al. ed., Springer-Verlag, Berlin, p. 87-114.
- [F (1987)] **FLETCHER (R.)**  
“Practical methods of optimization” (second edition), *Wiley, Chichester*.
- [H (1977)] **HAN (S.P.)**  
“A globally convergent method for nonlinear programming”, *J. Optimiz. Theory Appl. 22*, p. 297-309.
- [M-S (1982)] **MURIAGH (B.A.), SAUNDERS (M.A.)**  
“A projected lagrangian algorithm and its implementation for sparse nonlinear constraints”, *Math. Programming Study 16*, p. 84-117.
- [R (1982)] **REID (J.K.)**  
“A sparsity exploiting variant of the Bartels-Golub decomposition for linear programming bases”, *Math. Prog. 24*, p. 55-69.
- [S-M (1982)] **SHANNO (D.F.), MARSTEN (R.E.)**  
“Conjugate gradient methods for linearly constrained nonlinear programming”, *Math. Programming Study 16*, p. 149-161.

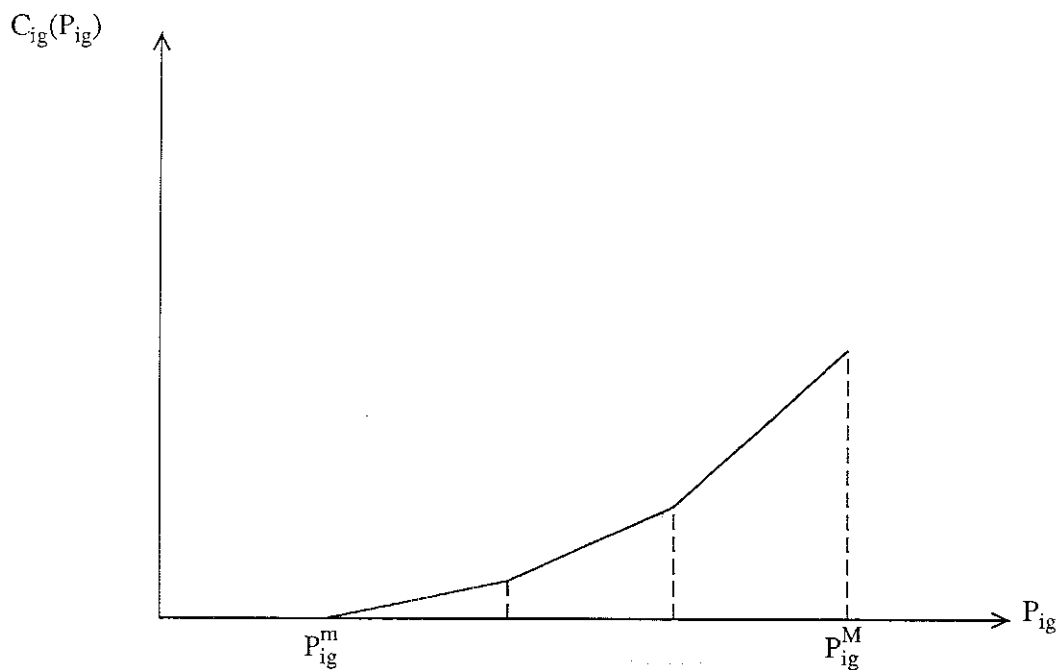


Figure 1

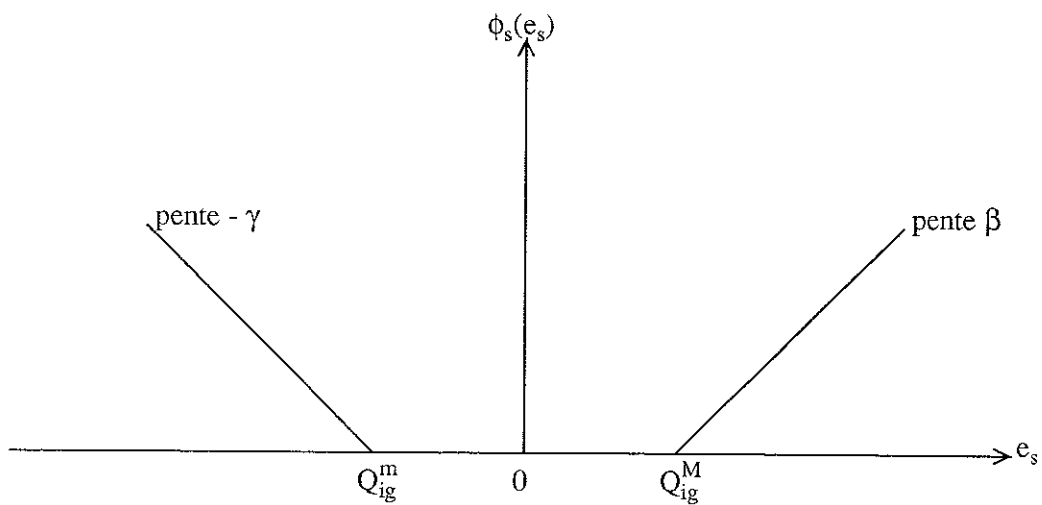


Figure 2

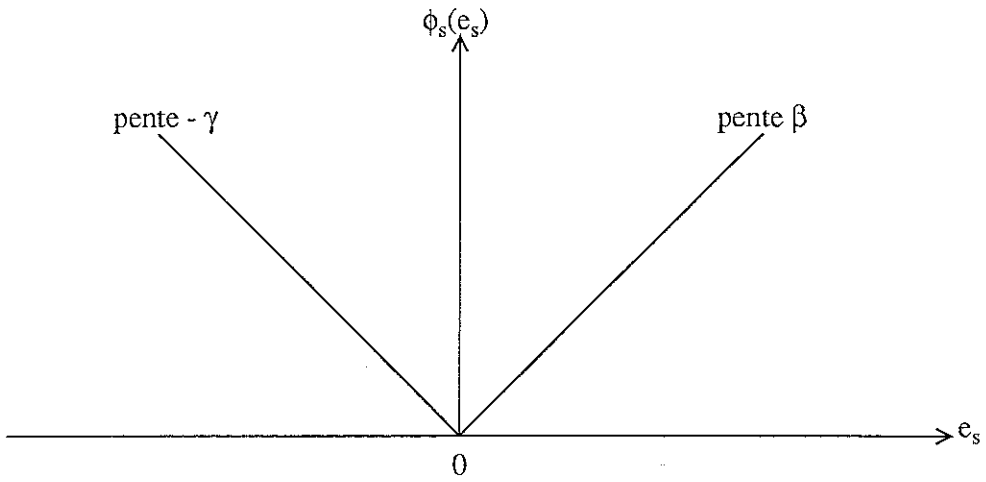


Figure 3

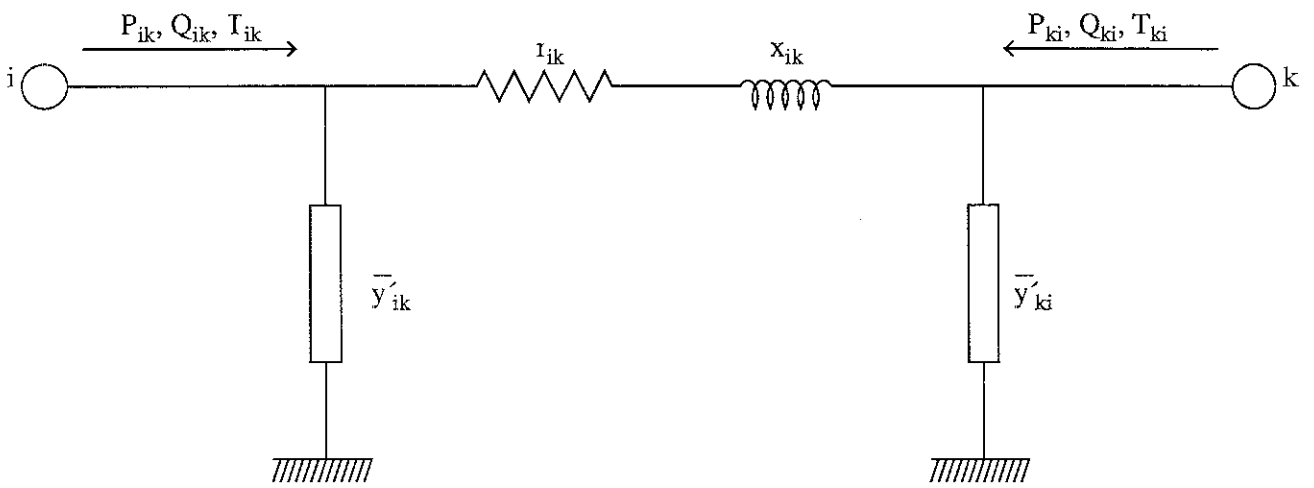


Figure 4