

L. PIBOULEAU

P. FLOQUET

S. DOMENECH

**Optimisation de procédés chimiques par
une méthode de gradient réduit partie I.
Présentation de l'algorithme**

RAIRO. Recherche opérationnelle, tome 19, n° 3 (1985),
p. 247-274

http://www.numdam.org/item?id=RO_1985__19_3_247_0

© AFCET, 1985, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

OPTIMISATION DE PROCÉDÉS CHIMIQUES PAR UNE MÉTHODE DE GRADIENT RÉDUIT PARTIE I. PRÉSENTATION DE L'ALGORITHME (*)

par L. PIBOULEAU, P. FLOQUET et S. DOMENECH ⁽¹⁾

Résumé. — *Un algorithme de type gradient réduit permettant de traiter les deux principales classes de problèmes continus rencontrés en Génie Chimique — l'optimisation des opérations unitaires et la conception optimale de procédés complexes — est présenté dans cet article. Un nombre élevé de variables bornées, un critère généralement implicite, ainsi qu'un ensemble creux de contraintes linéaires ou non, constituent les principales caractéristiques des problèmes répertoriés dans ces deux classes.*

La définition d'une partition des variables et l'utilisation de procédures numériquement stables d'algèbre linéaire, ont permis de développer un algorithme d'optimisation adapté au traitement de problèmes à grande échelle, pouvant comporter plusieurs centaines de variables, soumis à des contraintes linéaires.

Des techniques numériques de linéarisation, associées à cet algorithme, rendent aussi possible la résolution de problèmes à contraintes non linéaires.

Des exemples (numériques et de Génie Chimique) d'illustration de la méthode présentée, seront développés dans un second article.

Mots clés : Problèmes à grande échelle; contraintes linéaires et non linéaires; optimisation; gradient réduit.

Abstract. — *A reduced gradient algorithm, designed for solving the two basic classes of continuous optimization problems in the field of chemical engineering—unit operations optimization and CAD of large-scale processes—is presented in this paper. A great number of bounded variables, an implicit criterion, and a set of linear or nonlinear constraints giving rise to sparse jacobian matrix, form the main features of these two classes of problems.*

In the case of large-scale linearly constrained problems, involving several hundreds of variables, the algorithm implementation is based both on a partition of the variables and on the use of numerically stable matrix factorizations.

Nonlinear constrained problems can also be the means of numerical linearization procedures.

In a second article, the algorithm is illustrated by some test problems, involving numerical and chemical engineering examples.

Keywords: Large-scale problems; Linear and nonlinear constraints; Optimization; Reduced Gradient.

(*) Reçu mars 1984.

(¹) Institut du Génie Chimique, U.A. C.N.R.S. n° 192, Chemin de la Loge, 31078 Toulouse Cedex, France.

NOTATIONS

Lettres majuscules

B ,	matrice de base ($m \times m$);
C ,	matrice des coefficients des contraintes linéaires ($m \times n$);
C_1 ,	matrice des coefficients des n_1 variables intervenant uniquement dans les contraintes linéaires ($m_1 \times n_1$);
C_2 ,	matrice des coefficients des contraintes linéaires liant les n_2 variables qui interviennent aussi dans les contraintes non linéaires ($m_1 \times n_2$);
G ,	hessien de la fonction objectif ($n \times n$);
H ,	matrice orthogonale de Hessenberg supérieure ($s \times s$);
I ,	matrice identité;
J ,	jacobien des contraintes non linéaires ($m_2 \times n_2$);
L ,	facteur triangulaire inférieur de la base ($m \times m$);
N ,	matrice des coefficients des contraintes linéaires pour les variables hors base ($m \times n - s - m$);
P ,	produit de matrices de Givens ($s \times s$);
R ,	facteur triangulaire supérieur du hessien réduit ($s \times s$);
S ,	matrice des coefficients des contraintes linéaires pour les variables superbase ($m \times s$);
U ,	facteur triangulaire supérieur de la base ($m \times m$);
W ,	matrice égale au produit $B^{-1} S$ ($m \times s$);

$$Z = \begin{bmatrix} -W \\ I \\ 0 \end{bmatrix}, \text{ matrice de projection } (n \times s) \text{ pour le hessien.}$$

Lettres Minuscules

b ,	second membre des contraintes linéaires (vecteur de R^m);
b_1 ,	second membre des m_1 contraintes linéaires (vecteur de R^{m_1});
$c(x)$,	expression générale des contraintes (vecteur de R^m);
f ,	fonction objectif à minimiser (fonction de R^n dans R);
g ,	gradient de f (vecteur de R^n);
h ,	gradient réduit projeté (vecteur de R^s);
l ,	borne inférieure pour les variables (vecteur de R^n);

m ,	nombre de contraintes;
m_1 ,	nombre de contraintes linéaires;
m_2 ,	nombre de contraintes non linéaires;
n ,	nombre de variables de f ($n \geq m$);
n_c ,	nombre de variables non contraintes ($n_c \leq n - m$);
n_1 ,	nombre de variables intervenant uniquement dans les contraintes linéaires;
n_2 ,	nombre de variables intervenant dans les contraintes linéaires et non linéaires ($n_1 + n_2 = n$);
p ,	direction de descente (vecteur de R^n);
p_B ,	direction de descente pour les variables de base (vecteur de R^m);
p_N ,	direction de descente pour les variables hors base (vecteur nul de R^{n-m-s});
p_S ,	direction de descente pour les variables superbase (vecteur de R^s);
q ,	contraintes non linéaires (vecteur de R^{m_2});
\bar{q} ,	linéarisation des contraintes non linéaires (vecteur de R^{m_2});
s ,	nombre de variables superbase;
u ,	borne supérieure pour les variables (vecteur de R^n);
x ,	variables de f (vecteur de R^n);
x_B ,	variables de base (vecteur de R^m);
x_N ,	variables hors base (vecteur de R^{n-m-s});
x_S ,	variables superbase (vecteur de R^s).

Lettres grecques

α_{\max} ,	pas maximal dans la direction de descente;
α^* ,	pas optimal dans la direction de descente;
β_1 ,	approximation d'un majorant de la norme de $(R^T R)^{-1}$;
β_2 ,	approximation d'un majorant de la norme de $-W^T I 0$;
β_3 ,	coefficient positif strictement inférieur à 1;
δ ,	incréments sur les variables (vecteur de R^n);
ε_c ,	tolérance sur les contraintes non linéaires;
ε_f ,	tolérance sur la variation de la fonction objectif f ;
ε_g ,	tolérance sur le gradient réduit projeté;
ε_p ,	tolérance sur les pivots de la matrice de base;
ε_x ,	tolérance sur la variation des variables superbase;

ε_α ,	tolérance sur la précision de la recherche monodimensionnelle;
ε_λ ,	tolérance sur les paramètres de Lagrange;
λ ,	paramètres de Lagrange associés aux contraintes linéaires et/ou non linéaires (vecteur de R^m);
λ_1 ,	paramètres de Lagrange associés aux variables hors base (vecteur de R^{n-m-s});
ρ ,	coefficient de pénalisation.

INTRODUCTION

Dans le cadre de la chimie industrielle, l'optimisation des procédés, la commande optimale d'installations existantes et la conception optimale d'unités sont les bases nécessaires à la mise en œuvre d'une politique efficace dans la recherche d'économies d'énergie et/ou de matières premières. C'est pourquoi il nous a paru important de développer une procédure d'optimisation spécialement conçue pour des applications dans le domaine du Génie Chimique, où l'on peut distinguer deux grandes classes de problèmes :

- l'optimisation des opérations unitaires (distillation, réaction chimique, extraction...) à structure figée, fonctionnant en régime permanent ou transitoire;
- l'optimisation et/ou la conception optimale assistée par ordinateur (CAO) de procédés complexes, comportant plusieurs opérations unitaires, fonctionnant en régime permanent.

La première catégorie de problèmes a déjà fait l'objet de nombreux travaux [3-4, 7-9, 24, 46], dans lesquels des méthodes d'optimisation très différentes sont utilisées.

Parallèlement, divers efforts ont été entrepris dans le domaine de la CAO de procédés chimiques, en particulier pour la génération de séquences de réactions chimiques [21], et de distillation [18, 38, 40-41], ainsi que pour la détermination de réseaux d'échangeurs de chaleur [26-28]. Bien qu'en général des hypothèses assez restrictives soient utilisées dans les travaux précités leur nombre sans cesse croissant montre l'importance de ce type de recherche. Il est à noter que ces hypothèses ont souvent pour but de transformer le problème d'optimisation en un problème à structure discrète dont l'espace de recherche est fini.

Nous avons restreint la présente étude au cas de problèmes d'optimisation monocritère, à variables continues. Dans les deux classes de problèmes, le critère considéré, qui peut être de nature économique (coûts des investisse-

ments), énergétique (coûts d'exploitation), technico-économique, ou technique (capacité de production, durée des opérations, sécurité de fonctionnement), n'est pas donné sous une forme explicite. De plus, il comporte en général un nombre élevé de variables indépendantes, dont certaines sont physiquement bornées. Les équations d'équilibre et de bilans (massiques et/ou thermiques) conduisent à des problèmes dont le jacobien de la matrice des contraintes est creux.

La première partie de ce travail a trait au choix d'une méthode d'optimisation permettant de traiter le problème à grande échelle suivant :

$$(P) \quad \begin{cases} \text{Min } f(x), & x \in R^n, & (1) \\ c(x) = 0, & c \text{ est une application de } R^n \text{ dans } R^m \quad (m \leq n) & (2) \\ l \leq x \leq u, & l, u \in R^n, & (3) \end{cases}$$

où le jacobien des contraintes (2) est généralement creux.

La méthode retenue doit conduire à la définition d'un algorithme suffisamment souple pour pouvoir être appliqué à des problèmes de Génie Chimique de nature très différente.

Dans la deuxième partie, un nombre relativement important d'exemples numériques sera présenté afin de valider et de comparer cet algorithme avec d'autres méthodes d'optimisation. Enfin des exemples d'optimisation d'opérations unitaires à structure figée (équilibre chimique, réacteur, saponification, compression) et de CAO d'une unité de production d'acétate de cellosolve, seront détaillés.

CHOIX D'UNE MÉTHODE D'OPTIMISATION

Le choix d'une méthode d'optimisation permettant de résoudre le problème (P) doit s'effectuer en tenant essentiellement compte de la structure particulière de ce dernier.

La fonction objectif du problème comporte des modèles de simulation pour chaque opération unitaire du procédé, ainsi que des modèles d'évaluation des performances techniques et/ou énergétiques de celle-ci. De façon générale, cette fonction est non linéaire avec un grand nombre de variables, et n'est pas donnée sous une forme explicite.

De plus, le jacobien des contraintes définies par la relation (2) est en général creux, car chaque contrainte, qui intervient sur une opération unitaire, ou sur une partie d'opération unitaire, lie un faible nombre de variables indépendantes par rapport au nombre total. Le caractère creux du jacobien des contraintes est d'autant plus prononcé que le procédé est séquentiel.

Il est donc impératif de choisir une procédure d'optimisation qui puisse prendre en compte le nombre important de variables indépendantes et de contraintes ainsi que la nature creuse du jacobien des contraintes, tout en minimisant le nombre d'évaluations du critère et de son gradient.

Le choix d'une méthode d'optimisation adaptée à la structure particulière du problème (P), nécessite au préalable une importante analyse bibliographique, dont on présente ci-après l'essentiel.

Les méthodes qui paraissent être les plus appropriées au traitement du problème sont celles qui utilisent la notion de gradient réduit due à Wolfe [51]. Le principe fondamental des méthodes de gradient réduit consiste, par analogie avec la programmation linéaire, à exprimer à l'aide des contraintes (2) m variables (variables de base), en fonction des $n-m$ variables restantes (variables hors base). La résolution du problème (P) se ramène donc à celle du problème réduit (P1) suivant :

$$(P1) \quad \begin{cases} \text{Min}_{x_N} F(x_N) = f(x_B(x_N), x_N), \\ l_N \leq x_N \leq u_N, \end{cases} \quad (4)$$

$$(5)$$

où x_N (respectivement x_B) est la partition du vecteur x correspondant aux variables hors base (respectivement de base), l_N et u_N sont les composantes de l et u associées aux variables hors base. Le nombre $n-m$ de variables indépendantes du problème réduit (P1) est inférieur au nombre n de variables indépendantes du problème (P). Cette réduction du nombre de variables indépendantes est un point fondamental pour les applications de type Génie Chimique, compte tenu des temps de calcul nécessaires à l'évaluation de la fonction objectif et de son gradient.

La méthode du gradient réduit généralisé proposée par Abadie [1] et mise en œuvre par Abadie et Guigou [2], permet d'étendre au cas des contraintes non linéaires, la méthode de gradient réduit développée initialement dans le cas de contraintes linéaires. Toutefois, la nécessité de conserver en mémoire le jacobien des contraintes, son inverse, ainsi qu'un nombre assez important de vecteurs, rend délicat le traitement des problèmes de grande taille.

En associant les concepts développés par Abadie [1] et les techniques de partitionnement et de mise à jour de matrices, Lasdon *et al.* [25] proposent un algorithme de type gradient réduit permettant de traiter des problèmes de taille plus élevée que la méthode précédente.

Reprenant les méthodes décrites dans les références [2] et [25], Murtagh et Saunders [33-34] présentent un algorithme permettant de traiter des problèmes de grande taille, à contraintes linéaires et dont la matrice des contraintes est

creuse. Le problème général traité par ces auteurs est le problème (P) dans lequel les contraintes (2) s'écrivent :

$$C x = b, \quad (6)$$

où C est une matrice creuse ($m \times n$). Le point fondamental de cette méthode est la définition d'une partition du vecteur x des variables indépendantes en trois sous-vecteurs, correspondant respectivement aux variables de base (x_B), aux variables superbase (x_S) (ce sont les variables qui orientent principalement la recherche) et aux variables hors base (x_N); la matrice des contraintes est également partitionnée en trois sous-matrices B , S et N , de telle sorte que les contraintes (6) s'écrivent :

$$C x = \begin{array}{ccc} m & s & n-m-s \\ \begin{bmatrix} B & S & N \end{bmatrix} \times \begin{bmatrix} x_B \\ x_S \\ x_N \end{bmatrix} = b, \end{array} \quad (7)$$

où la matrice carrée non singulière B , est appelée matrice de base. Les composantes x_B et x_S sont libres de varier entre leurs bornes, tandis que celles de x_N sont maintenues égales à une de leurs bornes. Dans l'algorithme de Murtagh et Saunders, le nombre de variables intervenant au cours de la recherche, est réduit au nombre s (qui est toujours relativement faible) de variables superbase. Cet algorithme, plus connu sous le nom de MINOS (Modular In-core Nonlinear Optimization System), utilise des méthodes numériquement stables, pour factoriser la matrice de base (creuse) B [22-23] pour l'actualiser [5], et pour déterminer des approximations du hessien réduit [13, 29]. Les manuels utilisateurs du programme MINOS, qui est à notre connaissance un des seuls outils permettant de traiter des problèmes à grande échelle, soumis à des contraintes linéaires creuses, correspondent aux références [32] et [43].

Murtagh et Saunders [35] ont étendu les possibilités de MINOS au cas de contraintes non linéaires; le programme résultant porte le nom MINOS/AUGMENTED. Les manuels utilisateurs sont décrits dans les références [36-37, 44]. Le problème à contraintes non linéaires (1), (2), (3) est traité en résolvant une séquence à contraintes linéaires, chaque problème de la séquence étant issu du problème général (P) après linéarisation des contraintes (2) au voisinage d'un point courant, solution du problème précédent dans la séquence.

Gill *et al.* [16] proposent une méthodologie générale pour la mise en œuvre d'une bibliothèque de programmes d'optimisation, et indiquent en particulier que le choix d'algorithmes efficaces de recherche monodimensionnelle le long d'une direction de descente, constitue un point important de toute méthode d'optimisation. Cependant, afin de réduire le nombre d'évaluations de la fonction économique, on peut lors des premières itérations, diminuer le volume des calculs en ne recherchant qu'une approximation de l'optimum du critère dans la direction de recherche; ensuite lorsque l'on est proche de l'optimum, une interpolation quadratique ou cubique en donne une excellente approximation.

Une méthode de type gradient réduit, inspirée de MINOS semble adaptée au traitement du problème d'optimisation des procédés de la chimie industrielle. En effet, ce programme peut traiter un problème d'optimisation non linéaire de grande taille, dont la matrice du jacobien des contraintes est creuse, sans évaluer explicitement le hessien du critère. Toutefois, compte tenu de la structure particulière du problème d'optimisation en Génie Chimique, il est souhaitable d'apporter un nombre important de modifications afin de minimiser le nombre d'évaluations du critère et de son gradient.

Outre le choix d'une procédure d'optimisation de problèmes à grande échelle, se pose aussi celui des techniques d'algèbre linéaire permettant de traiter des matrices creuses de grandes tailles. De telles techniques, en particulier les formes produits d'inverses, ont tout d'abord été utilisées en programmation linéaire à grande échelle. Ainsi Bartels et Golub [5] ont défini un algorithme de programmation linéaire, dans lequel la matrice de base est factorisée sous forme LU . Ensuite, Forrest et Tomlin [12], puis Tomlin [48-49] ont proposé des méthodes pour actualiser les facteurs L et U , lors des changements de base d'un programme linéaire. Très récemment Tolla [47] a présenté une méthode d'actualisation de la décomposition triangulaire de la matrice de base permettant de pallier les instabilités numériques des algorithmes précités. Les algorithmes p^3 et p^4 de Hellerman et Rarick [22-23] permettent de préorganiser une matrice creuse afin de définir aisément ses facteurs L et U . L'utilisation de la méthode de factorisation de Cholesky a été développée par Saunders [42], pour des programmes linéaires à grande échelle.

Dans le cas de l'optimisation de fonctions non linéaires en présence de contraintes linéaires, Goldfarb [19-20] a proposé une méthode numériquement stable pour actualiser les facteurs L et U de la matrice de base creuse.

Les travaux de Gill *et al.* [13, 17] présentent l'intérêt de fournir une série de méthodes permettant de déterminer et d'actualiser des formes produits d'inverses, afin de préserver le caractère creux des matrices.

En ce qui concerne les méthodes de calcul d'approximations du hessien, il ressort de l'article de synthèse de Dennis et More [6] que celles de type quasi-Newton sont très utilisées en optimisation, afin d'actualiser les matrices hessiennes ou jacobienues. Ce type de méthode a été en particulier utilisé par Murtagh et Sargent [31] et par Schubert [45] pour estimer un jacobien creux. Fletcher et Powell [11], puis Powell [39], proposent des méthodes d'approximation de matrices hessiennes creuses, qui conservent la positivité.

Les deux ouvrages de Murray [29] et de Gill et Murray [14] font la synthèse des problèmes soulevés par l'optimisation non linéaire en présence ou non de contraintes, tant en ce qui concerne l'algorithmique que les techniques de calcul matriciel. Par ailleurs, des travaux récents de Vignes [50] et de Tolla [47] proposent des méthodes efficaces de contrôle de la stabilité numérique et de la précision des algorithmes de programmation mathématique.

ALGORITHME DANS LE CAS DE CONTRAINTES LINÉAIRES

Dans le cas où les contraintes du problème (P) sont linéaires et données sous la forme (6), les deux propriétés fondamentales des méthodes de projection [14] s'écrivent :

$$C(x_k + \Delta x_k) = b, \quad (8)$$

$$g(x_k) + G(x_k) \Delta x_k - C^T \lambda = 0, \quad (9)$$

où : x_k est un point acceptable de l'itération k ; Δx_k est un pas tel que le nouveau point $x_{k+1} = x_k + \Delta x_k$ soit aussi acceptable, compte tenu des contraintes linéaires (6) et des contraintes de bornes (3); g est le vecteur gradient de la fonction f ; G est la matrice hessienne de la fonction f ; λ est le vecteur des paramètres de Lagrange.

x_k étant un point acceptable, les relations précédentes deviennent :

$$C \Delta x_k = 0, \quad (10)$$

$$g(x_k) + G(x_k) \Delta x_k = C^T \lambda. \quad (11)$$

En utilisant la partition définie par Murtagh et Saunders [33-34] et donnée par la relation (7), ces deux propriétés s'écrivent :

$$\begin{bmatrix} B & S & N \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \Delta x_{B_k} \\ \Delta x_{S_k} \\ \Delta x_{N_{k+1}} \end{bmatrix} = 0, \quad (12)$$

$$\begin{bmatrix} g_{B_k} \\ g_{S_k} \\ g_{N_k} \end{bmatrix} + G(x_k) \begin{bmatrix} \Delta x_{B_k} \\ \Delta x_{S_k} \\ \Delta x_{N_k} \end{bmatrix} = \begin{bmatrix} B^T & 0 \\ S^T & 0 \\ N^T & I \end{bmatrix} \begin{bmatrix} \lambda \\ \lambda_1 \end{bmatrix}. \quad (13)$$

Dans les relations (12) et (13), la matrice C a été augmentée de la matrice identité d'ordre $(n-m-s)$ pour tenir compte du fait que les variables hors base sont maintenues égales à une de leurs bornes. Par suite de cette contrainte supplémentaire, le vecteur des paramètres de Lagrange est lui-même augmenté du vecteur $\lambda_1 \in R^{n-m-s}$.

Afin de simplifier les écritures, la variable x_k et l'indice d'itération ne seront plus mentionnés.

En développant la relation (12), il est possible d'exprimer l'accroissement Δx_B sur les variables de base, en fonction du pas Δx_S relatif aux variables superbase; en effet on a :

$$\Delta x_N = 0, \quad (14)$$

$$B \Delta x_B + S \Delta x_S = 0. \quad (15)$$

En posant $W = B^{-1}S$, les relations ci-dessus s'écrivent :

$$\Delta x = \begin{bmatrix} -W \\ I \\ 0 \end{bmatrix} \Delta x_S. \quad (16)$$

La relation (13) se simplifie lorsqu'elle est multipliée à gauche par la matrice :

$$\begin{bmatrix} I & 0 & 0 \\ -W^T & I & 0 \\ 0 & 0 & I \end{bmatrix} \quad (17)$$

La première ligne de cette multiplication donne l'expression des paramètres de Lagrange associés aux contraintes linéaires :

$$B^T \lambda = g_B + \begin{bmatrix} I & 0 & 0 \end{bmatrix} G \begin{bmatrix} -W \\ I \\ 0 \end{bmatrix} \Delta x_S. \quad (18)$$

Si $\|\Delta x_S\| = 0$ (x est alors un point stationnaire), en notant par π le vecteur solution de (18), dans ce cas particulier, on a :

$$B^T \pi = g_B. \quad (19)$$

La deuxième ligne de la multiplication donne l'expression du pas Δx_s :

$$\begin{bmatrix} -W^T I 0 \end{bmatrix} G \begin{bmatrix} -W \\ I \\ 0 \end{bmatrix} \Delta x_s = -h, \quad (20)$$

où h est le gradient réduit projeté, défini par :

$$h = g_s - S^T \pi. \quad (21)$$

La troisième ligne permet d'obtenir une expression des paramètres de Lagrange λ_1 relatifs aux variables hors base :

$$\lambda_1 = g_N - N^T \pi. \quad (22)$$

Les relations (19), (20), (21) et (22) définissent donc un algorithme de type gradient réduit, dans lequel la matrice de projection Z^T est $\begin{bmatrix} -W^T I 0 \end{bmatrix}$, le nombre de variables de recherche étant réduit au nombre de variables superbase. Murtagh et Saunders [34] ont montré que ce nombre s de variables superbase est borné par le nombre de variables non linéaires du critère.

Les différences principales de l'algorithme qui est présenté ci-après, par rapport à MINOS, sont les suivantes :

- 1° le vecteur gradient du critère est évalué numériquement;
- 2° il est possible de déterminer automatiquement une solution acceptable de base, et ce sans alourdir l'algorithme;
- 3° on peut calculer le nombre total d'évaluations de la fonction objectif en fonction du nombre d'itérations et de la précision demandée pour la recherche unidimensionnelle le long de la direction de descente;
- 4° les stratégies de factorisation de la base et d'actualisation de l'approximation du hessien réduit sont différentes, dans la mesure où les divers facteurs sont souvent recalculés plutôt qu'actualisés.

A cause du temps de calcul nécessaire à l'évaluation du critère, le troisième point est très important car il permet d'obtenir une borne supérieure sur le temps de calcul nécessaire à la résolution du problème, en fonction du nombre maximal d'itérations demandé et de la précision requise pour la recherche unidimensionnelle. On admet les hypothèses suivantes pour établir l'algorithme :

(H 1) Toutes les variables de la fonction objectif interviennent dans les contraintes linéaires et sont bornées.

(H 2) La matrice de base est factorisée sous la forme LU .

(H 3) Le hessien réduit $\begin{bmatrix} -W^T I 0 \end{bmatrix} G \begin{bmatrix} -W \\ I \\ 0 \end{bmatrix}$ est factorisé en un produit $R^T R$,

où R est une matrice d'ordre s , triangulaire supérieure.

(H 4) On dispose d'un point initial, acceptable ou non, x^0 .

Étape 1 : Détermination d'une solution acceptable de base.

(a) Le vecteur x_N^0 des variables hors base est défini en recherchant les variables de x^0 égales à une de leurs bornes.

(b) Parmi les composantes restantes, on définit m variables de base (vecteur x_B^0) et s variables superbase (vecteur x_S^0). La condition pour que m variables forment une base est que la matrice B correspondante soit non singulière.

(c) On détermine un accroissement Δx_B^0 des variables de base ($\Delta x_S^0 = \Delta x_N^0 = 0$), de façon à ce que le nouveau point x^1 vérifie les contraintes, en résolvant le système :

$$B \Delta x_B^0 = LU \Delta x_B^0 = b - C x^0$$

(si le second membre est nul, ceci est inutile, car le vecteur x^0 vérifie les contraintes).

(d) Si le vecteur $x_B^0 + \Delta x_B^0$ est situé entre ses bornes l_B et u_B , la solution acceptable de base est $x^1 = \begin{bmatrix} x_B^{0T} + \Delta x_B^{0T}, x_S^{0T}, x_N^{0T} \end{bmatrix}^T$. Sinon, on échange une variable de base et une variable superbase, on redéfinit la base B , et on revient en (c).

(e) On initialise l'indice d'itération k avec la valeur 1.

Étape 2 : On évalue la fonction f_k et son gradient g_k en x^k .

Étape 3 : On calcule le gradient réduit projeté h en x^k , en résolvant les deux systèmes :

$$(LU)^T \pi = g_B$$

$$h = g_S - S^T \pi.$$

Étape 4 : On examine la convergence dans l'espace superbase courant.

Si $\|x_S^k - x_S^{k-1}\| \leq \varepsilon_x$ et $|f_k - f_{k-1}| \leq \varepsilon_f$

ou

$\|h\| \leq \varepsilon_g$, on va à l'étape 5.

Sinon on saute à l'étape 6.

Étape 5 : (a) On calcule une estimation des paramètres de Lagrange λ_1 pour les variables hors base :

$$\lambda_1 = g_N - N^T \pi.$$

(b) Pour que les conditions de Kuhn-Tucker soient vérifiées, il faut que pour toutes les variables à leur borne supérieure (respectivement inférieure), les paramètres de Lagrange associés soient négatifs (respectivement positifs). On détermine donc les quantités :

$$\lambda_{1\ C_1} = \underset{\lambda_1 > 0}{\text{Max}} \{ \lambda_1 \} \text{ pour les variables à leur borne supérieure,}$$

$$\lambda_{1\ C_2} = \underset{\lambda_1 < 0}{\text{Min}} \{ \lambda_1 \} \text{ pour les variables à leur borne inférieure.}$$

Si $\lambda_{1\ C_1} \leq \varepsilon_\lambda$ et $|\lambda_{1\ C_2}| \leq \varepsilon_\lambda$ la recherche est terminée. Sinon, on choisit la variable hors base C_i , correspondant à :

$$|\lambda_{1\ C_i}| = \text{Max} \{ \lambda_{1\ C_1}, |\lambda_{1\ C_2}| \},$$

comme nouvelle variable superbase.

(c) La colonne C_i est ajoutée à la matrice S et ôtée de la matrice N . On ajoute la variable correspondante à x_S et on l'ôte de x_N . On définit une nouvelle composante de h , $\lambda_{1\ C_i}$. On ajoute une nouvelle colonne correspondant à la nouvelle variable superbase, à la matrice R ; la procédure est explicitée au paragraphe suivant (relations 28 à 30). Enfin, on incrémente le nombre s de variables superbase de 1.

Étape 6 : La direction de descente p_S pour les variables superbase est donnée par :

$$R^T R p_S = -h.$$

Pour les variables de base, la direction de descente p_B est définie par $LU p_B = -S p_S$.

Étape 7 : (a) On calcule α_{\max} , la valeur limite positive de α , telle que l'une des composantes du vecteur $\overline{x_B^T + \alpha p_B^T, x_S^T + \alpha p_S^T}^T$ atteigne l'une de ses bornes.

(b) Si $\alpha_{\max} \cdot \underset{j=1 \text{ à } s}{\text{Max}} \{ |p_{B_i}|, |p_{S_j}| \} \leq \varepsilon_x$, on va à l'étape 12.

Sinon, on continue séquentiellement.

Étape 8 : On effectue une recherche monodimensionnelle le long de la direction $p = \overline{p_B^T, p_S^T, 0}^T$, afin de déterminer une approximation de la valeur α^* telle que $f(x^k + \alpha^* p) = \underset{0 \leq \alpha \leq \alpha_{\max}}{\text{Min}} \{ f(x^k + \alpha p) \}$.

Étape 9 : On calcule le nouveau point $x^{k+1} = x^k + \alpha^* p$ et on incrémente l'indice d'itération k de 1.

Étape 10 : On évalue la fonction f_k et son gradient g_k en x^k .

Étape 11 : (a) On calcule le nouveau gradient réduit projeté h_1 au point x^k en résolvant :

$$(LU)^T \pi = g_B,$$

$$h_1 = g_S - S^T \pi.$$

(b) On actualise la matrice R pour tenir compte des variations du gradient réduit ($h_1 - h$); la démarche est décrite dans le paragraphe suivant.

(c) On modifie le gradient réduit projeté en faisant $h = h_1$.

Étape 12 : Si une composante de x_B ou de x_S atteint une de ses bornes, on va à l'étape 13. Sinon on revient à l'Étape 4.

Étape 13 : A ce niveau de l'algorithme, une variable de base ou superbase, dont l'indice sera noté i , atteint une de ses bornes.

(a) Si la variable i est une variable de base, alors :

(a.1) On échange la i -ième variable de x_B (respectivement la i -ième colonne de B) avec la j -ième variable de x_S (respectivement la j -ième colonne de S). L'indice j est choisi de telle sorte que la nouvelle matrice de base soit non singulière.

(a.2) On modifie les facteurs L et U , ainsi que la matrice R , comme indiqué ci-après.

(a.3) On calcule le nouveau gradient réduit projeté h en résolvant :

$$(LU)^T \pi = g_B,$$

$$h = g_S - S^T \pi$$

et l'on passe au point (b).

(b) Ici la variable i atteignant sa borne est obligatoirement une variable superbase; on procède alors comme suit.

(b.1) On transforme cette variable en variable hors base en :

- otant la colonne correspondante de S et en l'ajoutant à N ;
- otant la colonne associée de R ;
- otant la i -ème composante des vecteurs x_S et h .

(b.2) On met à nouveau R sous forme triangulaire supérieure (voir ci-dessous).

(b.3) Le nombre s est diminué de 1 et on revient à l'étape 4.

MISE EN OEUVRE DE L'ALGORITHME DANS LE CAS DE CONTRAINTES LINÉAIRES

1° Choix et factorisation de la base

A la première étape de l'algorithme, les variables de base correspondent aux composantes de x^0 qui sont le plus éloignées de leurs bornes. Ce choix a été fait afin d'éviter au maximum les changements de base, et donc la redéfinition des facteurs L et U . Les variables superbase correspondent aux composantes restantes de x . Il est impératif de vérifier que la matrice B associée à x_B^0 est non singulière, afin qu'elle puisse constituer une base. Pour ce faire, une méthode de pivot maximum est utilisée pour factoriser B sous la forme LU . Si tous les pivots sont, en valeur absolue, supérieurs à ε_p , B est non singulière [10], elle constitue alors une base. Dans le cas contraire, on définit une nouvelle matrice de base B , en échangeant une composante de x_B^0 et de x_S^0 et on essaye de la factoriser. En cas d'échec, on réitère le processus jusqu'à l'obtention d'une matrice non singulière. Le facteur U est déterminé en réduisant progressivement à zéro les éléments sous diagonaux de B , ce qui permet aussi la construction des colonnes de L . Des produits successifs de matrices de permutation et de matrices élémentaires (qui sont des matrices identité d'ordre m , mis à part une colonne qui possède des éléments sous-diagonaux non nuls) permettent d'effectuer cette construction progressive [14]. Cette factorisation permet alors de résoudre les systèmes comportant B ou B^T en deux phases d'élimination gaussienne.

2° Évaluation du gradient

L'évaluation du gradient (étapes 2 et 10) en un point x^k est effectuée selon un schéma aux différences finies. Les variables du problème pouvant avoir des significations physiques différentes, il est peu judicieux d'utiliser le même incrément pour chacune d'elles car la sensibilité de la fonction objectif aux variations des variables peut être très différente selon leur nature. Quatre formules d'approximation données ci-dessous, ont été prévues dans l'algorithme :

$$g_i^k = \frac{f(x^k + \delta_i e_i) - f(x^k)}{\delta_i}, \quad (23.a)$$

$$g_i^k = \frac{f(x^k + \delta_i x_i^k e_i) - f(x^k)}{\delta_i x_i^k}, \quad (23.b)$$

$$g_i^k = \frac{f(x^k + \delta_i e_i) - f(x^k - \delta_i e_i)}{2 \delta_i}, \quad (23.c)$$

$$g_i^k = \frac{f(x^k + \delta_i x_i^k e_i) - f(x^k - \delta_i x_i^k e_i)}{2 \delta_i x_i^k}, \quad (23.d)$$

où e_i est le i -ème vecteur canonique et g_i^k la dérivée partielle de f par rapport à x_i au point x^k .

Dans les relations (23), l'indice i est relatif à la i -ème composante des vecteurs. L'incrément est proportionnel à la valeur de la composante dans (23. b) et (23. d), ce qui dans certains cas peut être un avantage. L'approximation du gradient avec les relations (23. a) et (23. b) [respectivement (23. c) et (23. d)], nécessite n (respectivement $2n$) évaluations du critère, et l'erreur d'approximation est d'ordre 1 (respectivement 2).

3° Choix des tolérances

Afin d'effectuer le test de convergence dans l'espace superbase courant (étape 4), il est nécessaire de définir des valeurs adéquates pour les tolérances ε_x , ε_f et ε_g . Une première possibilité consiste à estimer ε_x en fonction de ε_g , car on a :

$$x_S^k - x_S^{k-1} = \alpha^* p_S$$

et

$$R^T R p_S = -h.$$

Il vient :

$$\|x_S^k - x_S^{k-1}\| = \alpha^* \|(R^T R)^{-1} h\|. \quad (24)$$

En notant par β_1 une approximation d'un majorant de la norme de la matrice $(R^T R)^{-1}$, on peut choisir :

$$\varepsilon_x = \alpha^* \beta_1 \varepsilon_g. \quad (25)$$

Il est également possible de définir ε_f en fonction de ε_x , donc de ε_g ; en effet, en développant la fonction f au premier ordre il vient :

$$\begin{aligned} f_k - f_{k-1} &= \Delta x^T g_{k-1} + O(\|\Delta x\|^2), \\ f_k - f_{k-1} &= p_S^T \overline{-(B^{-1} S)^T, I, 0} g_{k-1} + O(\|\Delta x\|^2). \end{aligned} \quad (26)$$

En notant par β_2 une approximation d'un majorant de la norme de $\overline{-(B^{-1} S)^T, I, 0}$ on peut choisir :

$$\varepsilon_f = \beta_1 \beta_2 \varepsilon_g \|g_{k-1}\|. \quad (27)$$

Le test sur les variations de x et de f peut être toutefois conservé car, d'une part le choix des valeurs de β_1 et β_2 (on choisit souvent $\beta_1 = \beta_2 = 1$)

peut être erroné, car les majorants d'erreur d'arrondi calculés à partir de normes de matrices sont la plupart du temps, beaucoup trop importants, ou d'autre part la valeur de ε_g peut être incompatible avec les valeurs de h .

Une seconde démarche a été récemment publiée par Vignes [50] qui s'appuie sur une représentation statistique des erreurs machine et qui ajuste les valeurs des tolérances en fonction de cette évaluation. Cette méthode de permutation-perturbation nécessite trois exécutions du programme d'optimisation ce qui peut entraîner, pour des problèmes de grande taille, une augmentation considérable du volume des calculs.

A l'étape 5 de l'algorithme, lorsque les paramètres de Lagrange λ_1 n'ont pas les signes données par les conditions d'optimalité de Kuhn-Tucker, étant donné que la convergence a été obtenue dans l'espace superbase courant, il est nécessaire de modifier cet espace afin que la recherche puisse évoluer.

4° Addition d'une variable superbase et modification de la matrice R

La solution adoptée consiste à choisir une nouvelle variable superbase, qui est la variable hors base correspondant au « plus mauvais » paramètre de Lagrange $\lambda_{1\ c_i}$. En pratique, l'addition de cette nouvelle variable superbase, se traduit par les modifications suivantes, d'après Murtagh et Saunders [33-34] :

$$\begin{aligned}\bar{S} &= \overline{S, c_{i_p}} \\ \bar{g}_S &= \overline{g_S^T, g_{i_p}^T}^T, \\ \bar{h} &= \overline{h^T, \lambda_{1\ c_i}}^T, \\ \bar{R}^T \bar{R} &= \begin{bmatrix} R^T & 0 \\ r^T & \mu \end{bmatrix} \begin{bmatrix} R & r \\ 0 & \mu \end{bmatrix}. \end{aligned} \quad (28)$$

Les termes r et μ de la relation (28) sont définis comme suit :

$$R^T r = Z^T v \quad \text{où la matrice de projection } Z = \begin{bmatrix} -B^{-1} S \\ I \\ 0 \end{bmatrix}, \quad (29. a)$$

$$\mu^2 = z^T v - r^T r. \quad (29. b)$$

Les vecteurs z et v sont respectivement donnés par :

$$z = \begin{bmatrix} -B^{-1} c_i \\ e_{s+1} \\ 0 \end{bmatrix} \quad (e_{s+1} \text{ est le } (s+1)\text{-ième vecteur unitaire}) \quad (30. a)$$

$$v = G z. \quad (30. b)$$

On peut définir une approximation du vecteur v en utilisant le schéma aux différences finies suivant :

$$v = G z \simeq \frac{g(z + \delta z) - g(z)}{\delta z}.$$

Mais ce procédé nécessite la connaissance analytique du gradient, car il n'est pas envisageable de calculer les dérivées secondes du critère par différences finies, à cause du nombre d'évaluations de ce dernier et des erreurs numériques que cela entraînerait. Pour la classe de problèmes considérés, qui sont caractérisés par un critère généralement implicite nécessitant un volume de calculs important, une solution possible consiste à définir la matrice \bar{R} comme étant la matrice identité d'ordre $(s + 1)$.

Il est à remarquer que la matrice R est la matrice identité au début de la recherche.

5° Recherche monodimensionnelle

Après avoir déterminé la direction de descente p pour les variables de base et superbase, une recherche monodimensionnelle est effectuée le long de cette direction (étape 8). Dans le programme MINOS, cette recherche est effectuée à l'aide de la méthode d'interpolation cubique, proposée par Gill et Murray [15]. Malgré sa souplesse et sa précision, cet algorithme présente l'inconvénient majeur de nécessiter l'évaluation du gradient de la fonction au cours de la recherche. Ces auteurs [15] donnent aussi un algorithme d'interpolation quadratique, moins précis que l'algorithme cubique, mais ne nécessitant pas le calcul du gradient. Toutefois, la formulation de cette méthode ne permet pas de déterminer *a priori* le nombre d'appels au critère en fonction de la précision demandée sur le résultat α^* . L'avantage de la méthode classique du nombre d'or réside dans la connaissance d'une borne supérieure sur le nombre d'évaluations du critère en fonction de la précision souhaitée sur le résultat, compte tenu des temps de calcul nécessaires. Son inconvénient majeur est que la variation de la fonction sur l'intervalle de recherche n'est pas prise en compte, comme dans les méthodes de Gill et Murray [15]. Il est à remarquer que dans les deux classes de méthodes, l'unimodalité de la fonction est toujours l'hypothèse de base. Compte tenu de la complexité des critères considérés en Génie Chimique, l'algorithme du nombre d'or a été retenu afin d'adopter un compromis entre la précision de la recherche monodimensionnelle et le temps de calcul.

Dans cet algorithme, la recherche est arrêtée lorsque l'amplitude de l'intervalle courant est inférieure à la précision ε_α fixée *a priori*. Le résultat α^* peut être obtenu, soit par interpolation sur l'intervalle, soit en prenant son milieu. La valeur de la tolérance ε_α doit être, dans tous les cas, inférieure à la valeur de la tolérance ε_x requise sur les variables :

$$\varepsilon_\alpha = \beta_3 \varepsilon_x \quad \text{avec} \quad \beta_3 < 1. \quad (31)$$

Pour une valeur donnée de β_3 (par exemple $\beta_3 = 0,1$ ou $0,5$), il est possible à l'aide de la relation (25), de définir ε_α en fonction de ε_g .

6° Passage à l'itération suivante. Modification de la matrice R

Lors du passage à l'itération suivante, on définit le nouveau point x^k et le nouveau gradient réduit projeté h_1 . Il est alors nécessaire de modifier la matrice R (étape 11), pour tenir compte de la variation ($h_1 - h$) du gradient réduit projeté. L'approximation $R^T R$ du hessien réduit est modifiée à l'aide de la formule de rang 2 de Davidon-Fletcher-Powell complémentaire, donnée dans la référence [14]. Murtagh et Saunders [33-34] définissent la nouvelle approximation du hessien réduit comme suit :

$$\bar{R}^T \bar{R} = R^T R + \frac{1}{\alpha^* y^T p_s} y y^T + \frac{1}{h^T p_s} h h^T, \quad (32)$$

où $y = h_1 - h$.

La modification (32) est déterminée à l'aide de deux corrections successives de rang 1 du type :

$$\bar{A} = A + \delta z z^T, \quad (33)$$

où z est un vecteur de R^s représentant y (respectivement h) et $\delta = 1/\alpha^* y^T p_s$ (respectivement $1/h^T p_s$) et $A = R^T R$.

Un produit de matrices de rotations planes (ou matrices de Givens), égal à une matrice orthogonale H de Hessenberg supérieure [13] (définie à l'aide de 3 vecteurs), permet d'écrire :

$$\bar{A} = R^T H^T H R.$$

Ensuite un second produit P de matrice de Givens permet de réduire H à une matrice triangulaire supérieure \hat{R} (définie à l'aide de 3 vecteurs); la

matrice \bar{A} est alors donnée par :

$$\bar{A} = R^T H^T P^T P H R,$$

c'est-à-dire par :

$$\bar{A} = \bar{R}^T \bar{R} = R^T \hat{R}^T \hat{R} R \quad \text{avec} \quad \hat{R} = P H, \quad (34)$$

d'où :

$$\bar{R} = \hat{R} R. \quad (35)$$

Toutes les relations permettant de calculer les matrices ci-dessus sont données par Gill *et al.* [13]. Il est à noter que dans la procédure d'optimisation, la matrice R n'est modifiée, que dans le cas où $\alpha^* y^T p_s$ et $h^T p_s$, sont en valeur absolue, supérieurs à la tolérance ε_p , afin d'éviter des instabilités numériques. Dans le cas contraire, la matrice \bar{R} est définie comme étant la matrice identité d'ordre s .

7° Cas de variables atteignant leurs bornes

Lorsque la variable de base x_i atteint une de ses bornes (étape 13. a), la variable superbase x_j qui la remplace dans la base est déterminée selon la procédure donnée par Murtagh et Saunders [33-34]. Les facteurs L et U sont recalculés, plutôt qu'actualisés comme dans le programme MINOS, afin de compenser les erreurs dues à l'évaluation numérique du gradient et à l'initialisation de la matrice R à la matrice identité. La nouvelle matrice \bar{R} , résultant du changement de la i -ème variable superbase, est définie d'après les relations données par Murtagh et Saunders [33-34]. Elle est ensuite remise sous forme triangulaire supérieure par un produit de matrices de rotations planes [13].

Lorsque (étape 13. b), une colonne numéro k est supprimée de la matrice R , les colonnes $k+1, k+2, \dots, s$, possèdent toutes un élément sous-diagonal non nul; un produit de matrices de rotations planes permet de remettre R sous forme triangulaire supérieure.

8° Remarques

A ce niveau, un certain nombre de remarques s'imposent sur l'algorithme qui a été défini sous l'hypothèse restrictive (H 1).

Dans le cas où une variable n'est pas bornée inférieurement (respectivement supérieurement), elle est bornée artificiellement avec un réel négatif (respectivement positif) arbitrairement grand. Ce cas correspond, par exemple, aux variables d'écart, qui sont naturellement bornées d'un seul côté.

Lorsque le problème comporte n_c ($n_c \leq n - m$) variables n'intervenant pas dans les contraintes linéaires (6), et appelées variables non contraintes, ces variables sont rangées dans les n_c dernières composantes de x .

La recherche des variables de base initiales (étape 1) est effectuée sur les $(n - n_c)$ premières composantes de x . Les variables superbase non contraintes sont rangées à la fin du vecteur x_S^0 et correspondent donc aux dernières colonnes de R . Pour ces variables superbase, le gradient réduit projeté est égal à g_S et la direction de descente p_S est $-g_S$; c'est-à-dire que les dernières colonnes de R , associées à ces variables, ne sont jamais modifiées. D'autre part, le vecteur λ_1 des paramètres de Lagrange correspondant aux variables hors base non contraintes est égal à g_N . Enfin lorsqu'une variable de base est échangée avec une variable superbase, cette dernière ne doit pas être une variable non contrainte.

Dans l'algorithme, un vecteur permet de définir la nature de chaque composante de x (hors base à la borne inférieure, hors base à la borne supérieure, superbase et base), la composante numéro i de ce vecteur est relative à la i -ème composante de x . Ce vecteur sert aussi à identifier les colonnes (respectivement les lignes) des matrices B , S , N (respectivement des vecteurs g_B , g_S , g_N et p_B , p_S , p_N) qui sont rangées en réalité dans la matrice C (respectivement dans les vecteurs g et p).

La matrice creuse C , ainsi que les facteurs L et U de la base, sont ainsi mémorisés sous une forme compacte, selon le schéma décrit dans les références [32, 43].

Algorithme dans le cas de contraintes linéaires et non linéaires

La majorité des problèmes d'optimisation rencontrés en pratique, sont soumis à la fois à des contraintes linéaires et non linéaires; nous considérerons que cette classe de problèmes peut s'écrire sous la forme standard suivante :

$$(P2) \quad \begin{cases} \text{Min}_x f(x), & (36) \\ C_1 x_1 + C_2 x_2 = b_1 & (m_1 \text{ lignes}), & (37) \\ q(x_2) = 0 & (m_2 \text{ lignes}), & (38) \\ l \leq x \leq u & (m = m_1 + m_2 \text{ et } n = n_1 + n_2) & (39) \end{cases}$$

où :

$$-x^T = \begin{bmatrix} x_1^T & x_2^T \end{bmatrix};$$

- x_1 (respectivement x_2) est le vecteur contenant les n_1 ($0 \leq n_1 < n$) (respectivement n_2 , $0 < n_2 \leq n$) composantes de x qui interviennent dans les m_1

contraintes linéaires (respectivement les m_1 contraintes linéaires et les m_2 contraintes non linéaires);

— la matrice C_1 et/ou C_2 peut éventuellement être vide;

— $q^T = \overline{q_1, q_2, \dots, q_{m_2}}$ est le vecteur des contraintes non linéaires, supposé être deux fois continuellement dérivable et à dérivées secondes bornées.

Le problème (P2) est traité en résolvant une séquence de problèmes à contraintes linéaires, méthode apparentée à celle du lagrangien augmenté; Murtagh et Saunders [35] ont défini le k -ième problème de la séquence de la façon suivante :

$$(P_{2k}) \left\{ \begin{array}{l} \underset{x}{\text{Min}} L(x, x_{2k}, \lambda_k, \rho) = f(x) - \lambda_k^T (q(x_2) - \bar{q}(x_2)) \\ \quad \quad \quad + \frac{1}{2} \rho (q(x_2) - \bar{q}(x_2))^T (q(x_2) - \bar{q}(x_2)), \quad (40) \\ C_1 x_1 + C_2 x_2 = b_1, \quad (41) \\ \bar{q}(x_2) = 0, \quad (42) \\ 1 \leq x \leq u, \quad (43) \end{array} \right.$$

où :

— k est le numéro du problème (appelé aussi itération majeure), dans la séquence;

— x_{2k} est le point où la linéarisation des contraintes (38) est effectuée;

— $\bar{q}(x_2)$ représente l'approximation linéaire des contraintes (38) au voisinage de x_{2k} ; $\bar{q}(x_2)$ est évalué comme suit :

$$\bar{q}(x_2) = q(x_{2k}) + J(x_{2k})(x_2 - x_{2k}), \quad (44)$$

$J(x_{2k})$ étant la matrice jacobienne de q (dimension $m_2 \times n_2$) évaluée en x_{2k} ;

— λ_k est le vecteur des m_2 paramètres de Lagrange;

— ρ est un facteur positif de pénalisation.

Par suite de la présence du terme de pénalisation, la fonction L de la relation (40) est appelée un lagrangien augmenté.

Le vecteur λ_k doit être aussi proche que possible du vecteur λ^* des paramètres de Lagrange associés à la solution optimale du problème (P_{2k}) . Le choix le plus simple consiste à définir $\lambda_k = \hat{\lambda}$, où $\hat{\lambda}$ est le vecteur des paramètres de Lagrange associés aux contraintes linéarisées à la solution du problème précédent (P_{2k-1}) . Ce choix est dicté par la méthode de Robinson décrite par Gill et Murray [14], et assure une convergence superlinéaire des

solutions x_k^* des problèmes (P_{2k}) vers une solution optimale x^* du problème (P_2) , à condition que l'estimation initiale (x^0, λ_0) soit suffisamment proche de cette solution (x^*, λ^*) .

Pour qu'un point stationnaire du problème (P_{2k}) soit une solution optimale, il faut et il suffit que son hessien réduit soit défini positif en ce point. Or même si le problème initial (P_2) est convexe, la linéarisation des contraintes autour d'un point x_{2k} peut engendrer un problème (P_{2k}) non convexe, lorsque x_{2k} est trop éloigné de cette solution optimale. L'addition du terme $(1/2) \rho (q - \bar{q})^T (q - \bar{q})$ peut rétablir la convexité du problème (P_{2k}) , pour une valeur positive, suffisamment grande de ρ . On choisira donc une valeur positive pour le paramètre ρ si le problème (P_2) est non convexe et/ou fortement non linéairement contraint, cependant, il faut éviter de choisir une valeur trop élevée pour ne pas figer la recherche; dans le cas contraire, une valeur nulle pour ρ pourra être prise. D'autre part, Murtagh [30], Murtagh et Saunders [35], indiquent qu'une pénalisation avec le terme $(1/2) \rho (q - \bar{q})^T (q - \bar{q})$ assure une convergence plus rapide vers une solution du problème (P_2) , qu'une pénalisation avec le terme $(1/2) \rho q^T q$.

De façon à résoudre le problème (P_{2k}) avec l'algorithme défini dans le cas de contraintes linéaires, on le reformule sous la forme équivalente suivant :

$$(P_{3k}) \quad \begin{cases} \min_x L(x, x_{2k}, \lambda_k, \rho) \\ = f(x) - \lambda_k^T (q(x_2) - \bar{q}(x_2)) + \frac{1}{2} \rho (q(x_2) - \bar{q}(x_2))^T (q(x_2) - \bar{q}(x_2)) \quad (45) \\ Cx = b, \quad (46) \\ l \leq x \leq u \quad (47) \end{cases}$$

où la matrice C et le vecteur b sont définis comme suit :

$$C = \begin{bmatrix} n_1 & n_2 \\ C_1 & C_2 \\ 0 & J(x_{2k}) \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}$$

$$b = \begin{bmatrix} b_1 \\ J(x_{2k}) x_{2k} - q(x_{2k}) \end{bmatrix}$$

En supposant que l'on dispose d'un vecteur initial x^0 (ne vérifiant pas nécessairement les contraintes), d'une estimation initiale λ_0 du vecteur des

paramètres de Lagrange (souvent $\lambda_0=0$), ainsi que d'une valeur ρ_0 pour le paramètre ρ , les étapes logiques de l'algorithme sont les suivantes :

Étape 1 : On initialise le numéro k de l'itération majeure.

Étape 2 : On construit la matrice C et le vecteur b .

Étape 3 : On utilise l'algorithme défini dans le cas de contraintes linéaires pour résoudre le problème (P_{3k}) . On note \hat{x} sa solution optimale et $\hat{\lambda}$ le vecteur des paramètres de Lagrange correspondant (qui est en réalité le vecteur π).

Étape 4 : Les m_2 dernières composantes de $\hat{\lambda}$ sont rangées dans λ_{k+1} , et on définit $x_{k+1} = \hat{x}$.

Étape 5 : La convergence des itérations majeures est analysée de la façon suivante.

Si $\|\lambda_{k+1} - \lambda_k\| \leq \varepsilon_\lambda$

et il n'y a pas eu de changement de base au cours de l'itération majeure k

et $\|x_{k+1} - x_k\| \leq \varepsilon_x$.

et $\|q(x_{k+1})\| \leq \varepsilon_c$

la recherche est terminée. Sinon on passe à l'étape 6.

Étape 6 : On définit le paramètre ρ_{k+1} . La procédure adoptée consiste à faire varier ρ en fonction de l'écart des paramètres de Lagrange :

$$\rho_{k+1} = \rho_k \max_{j=1 \text{ à } m_2} \{|\lambda_{k+1, j} - \lambda_{k, j}|\}.$$

Si $\rho_{k+1} > \rho_{\max}$, alors $\rho_{k+1} = \rho_{\max}$, où ρ_{\max} est une valeur maximale arbitraire donnée.

Étape 7 : On fait $k = k + 1$.

Si $k \leq k_{\max}$, on revient à l'étape 2. Sinon, la recherche s'arrête sur le nombre maximal d'itérations majeures.

MISE EN ŒUVRE DE L'ALGORITHME DANS LE CAS DE CONTRAINTES LINÉAIRES ET NON LINÉAIRES

Comme dans le cas de l'algorithme correspondant à des contraintes purement linéaires, l'évaluation du terme d'indice (i, j) du jacobien $J(x_{2k})$, c'est-à-dire $\partial q_i / \partial x_{2k, j}$, est effectuée suivant un des quatre schémas aux différences finies données précédemment.

La construction de la matrice C (étape 2) est délicate car contrairement au cas linéaire, le nombre n_c de variables non contraintes peut évoluer au cours

des itérations majeures. En effet, pour une valeur donnée de x_{2k} , une ou plusieurs colonnes de $J(x_{2k})$ peuvent être nulles, et si les variables correspondantes ne sont pas contraintes linéairement, elles sont alors non contraintes.

Les variables non contraintes linéairement, en nombre n_{cl} , étant rangées à la fin du vecteur x , la matrice C s'écrit :

$$C = \begin{array}{c|c|c|c} n'_1 & n'_2 & n_{cl2} & n_{cl1} \\ \hline C'_1 & C'_2 & 0 & 0 \\ \hline 0 & J(x_{2k}) & & 0 \end{array} (n_{cl1} + n_{cl2} = n_{cl}),$$

où :

- n_{cl1} (respectivement n_{cl2}) ≥ 0 , représente le nombre de composantes de x_1 (respectivement x_2) qui ne sont pas contraintes linéairement;
- C'_1 (respectivement C'_2) représente les $n'_1 = n_1 - n_{cl1}$ (respectivement $n'_2 = n_2 - n_{cl2}$) premières colonnes de C_1 (respectivement C_2).

Les vecteurs x et b sont construits selon le même schéma que la matrice C . Le jacobien $J(x_{2k})$ est évalué colonne par colonne. Si une colonne du jacobien, située dans la matrice C en position $n'_1 + n'_2 + 1$, $n'_1 + n'_2 + 2$, \dots , $n'_1 + n'_2 + n_{cl2}$, n'est pas identiquement nulle, la variable correspondante est contrainte non linéairement, le nombre total n_{cl} de variables non contraintes diminue donc de 1. Dans le cas où cette colonne est identiquement nulle, la variable associée n'est pas contrainte, le nombre n_{cl} garde sa valeur.

Lorsque le nombre total de variables non contraintes évolue au cours des itérations majeures, des permutations peuvent se produire sur les composantes du vecteur x . Il est alors nécessaire de mémoriser ces permutations afin de pouvoir évaluer le critère, les contraintes, ainsi que les dérivées, avec les composantes de x correctement ordonnées (c'est-à-dire dans l'ordre où elles sont données au début de la recherche).

A l'étape 5 de l'algorithme, il est nécessaire de vérifier si la base a été modifiée au cours de l'itération majeure, car les paramètres de Lagrange sont donnés par :

$$\begin{aligned} B_k \lambda_k &= g_B(x_k), \\ B_{k+1} \lambda_{k+1} &= g_B(x_{k+1}) \end{aligned}$$

et l'on ne peut donc les comparer que si $B_k = B_{k+1}$.

CONCLUSION

Les problèmes d'optimisation rencontrés en Génie Chimique sont trop souvent traités par des méthodes spécifiques à chacun d'eux. Il nous a donc paru important de définir une procédure d'optimisation suffisamment générale, permettant de traiter les deux principales classes de problèmes considérés en Génie Chimique, c'est-à-dire d'une part l'optimisation d'opérations unitaires à structure figée, et d'autre part l'optimisation et/ou la CAO de systèmes industriels complexes.

Bien que ces problèmes soient classés dans deux catégories différentes, ils présentent néanmoins, au niveau de leur structure, les principaux points communs suivants :

- la fonction objectif, qui représente une évaluation des performances techniques et/ou économiques du système, comporte un nombre élevé de variables, et n'est pas en général donnée explicitement;
- l'ensemble des contraintes linéaires et/ou non linéaires est généralement creux;
- de part leur nature physique, les variables du système sont bornées.

Compte tenu de cette structure particulière, une procédure de type gradient réduit, couplée avec des algorithmes d'algèbre linéaire numériquement stables, a été mise en œuvre. Des techniques numériques de linéarisation ont permis le traitement de problèmes à contraintes non linéaires. Toutefois, pour ce type de problèmes, le choix d'un facteur de pénalisation et d'une estimation des paramètres de Lagrange peut entraîner des difficultés de convergence. Des exemples numériques d'illustration de la méthode présentée, seront développés dans une seconde partie.

BIBLIOGRAPHIE

1. J. ABADIE, *The GRG Method for Nonlinear Programming*, Design and Implementation of Optimization Software, Harvey Greenberg Sijthoff and Noordhof Ed., 1978.
2. J. ABADIE et J. GUIGOU, *Gradient Réduit Généralisé*, note HI 069/02, EDF, Clamart, France, 1969.
3. M. M. ABBOTT et M. C. VAN NESS, *Théorie et applications de la thermodynamique*, série Schaum, 1978.
4. R. ARIS, R. BELLMAN et A. KALABA, *Some Optimization Problems in Chemical Engineering*, C.E.P. Symposium Series, 56, 1960, p. 95.
5. R. H. BARTELS et G. H. GOLUB, *The Simplex Method of Linear Programming Using LU Decomposition*, Communications of the A.C.M., 12, 1969, p. 266.
6. J. E. DENNIS et J. J. MORE, *Quasi Newton Methods, Motivation and Theory*, S.I.A.M. Rev., 19, 1977, p. 46.

7. S. DOMENECH, *Optimisation d'une opération de rectification discontinue — Commande en temps minimal*, Thèse de Doctorat d'État, I.N.P., Toulouse, 1976.
8. S. DOMENECH et M. ENJALBERT, *Optimisation d'une opération de rectification discontinue par programmation dynamique*, An. Quimi. Real Soc. Esp. Fis. Quim., 74, 1978, p. 319.
9. S. DOMENECH, G. MURATET et N. THERIEN, *Recherche de commandes optimales liées à l'alimentation d'une usine de traitement d'eaux usées*, Tribune du Cebedea, 31, 1978, p. 241.
10. E. DURAND, *Solutions numériques des équations algébriques*, Tome II, Masson et Cie, Paris, 1972.
11. R. FLETCHER et M. J. D. POWELL, *On the Modification of LDL^T Factorizations*, Math. Comp., 28, 1974, p. 1067.
12. J. J. FORREST et J. A. TOMLIN, *Updated Triangular Factors of the Basis to Maintain Sparsity in the Product Form for Simplex Method*, Math. Prog., 2, 1972, p. 263.
13. P. E. GILL, G. H. GOLUB, W. MURRAY et M. A. SAUNDERS, *Methods for Modifying Matrix Factorizations*, Math. Comp., 28, 1974, p. 505.
14. P. E. GILL et W. MURRAY, *Numerical Methods for Constrained Optimization*, Academic Press, 1974.
15. P. E. GILL et W. MURRAY, *Safeguarded Steplength Algorithms for Optimization Using Descent Methods*, N.P.L. Report NAC 37, 1974.
16. P. E. GILL, W. MURRAY, S. M. PICKEN et M. H. WRIGHT, *The Design and Structure of a Fortran Program Library for Optimization*, Report SOL 77-7, Stanford Univ., CA, 1977.
17. P. E. GILL, W. MURRAY et M. A. SAUNDERS, *Methods for Computing and Modifying the LDV Factors of a Matrix*, Math. Comp., 29, 1975, p. 1051.
18. A. GOMEZ et J. D. SEADER, *Separation Sequences Synthesis by a Predictor Based Ordered Search*, A.I.Ch.E. J., 22, 1976, p. 970.
19. D. GOLDFARB, *Matrix Factorization in Optimization of Nonlinear Functions Subject to Linear Constraints*, Math. Prog., 10, 1975, p. 1.
20. D. GOLDFARB, *Matrix Factorization in Optimization of Nonlinear Functions Subject to Linear Constraints — An Addendum*, Math. Prog., 12, 1977, p. 279.
21. J. B. HENDRICKSON, *A General Protocol for Synthesis Design*, Topics in Current Chemistry, 62, 1976, p. 49.
22. E. HELLERMAN et D. RARICK, *Reinversion with the Preassigned Pivot Procedure*, Math. Prog., 1, 1971, p. 195.
23. E. HELLERMAN et D. RARICK, *The Partitioned Preassigned Pivot Procedure (P^4)*, in *Sparse Matrices and their Applications*, Plenum Press, 1972, p. 67.
24. HO THI DIEU, *Optimisation continue d'un procédé chimique au moyen d'un calculateur analogique*, Thèse M.S., Université de Laval, 1968.
25. L. S. LASDON, R. L. FOX et M. W. RATNER, *Nonlinear Optimization Using the Generalized Reduced Gradient Method*, R.A.I.R.O., 3, 1974, p. 73.
26. B. LINNHOFF, D. R. MASON et I. WARDLES, *Understanding Heat Exchanger Networks*, Comp. and Chem. Eng., 3, 1979, p. 295.
27. B. LINNHOFF et J. A. TURNER, *Heat-Recovery Networks: New Insights Yield Big Savings*, Chem. Eng., 11, 1981, p. 56.
28. M. A. MENZIES et A. I. JOHNSON, *Synthesis of Optimal Energy Recovery Networks Using Discrete Methods*, Can. J. Chem. Engineer., 50, 1972, p. 290.

29. W. MURRAY, *Numerical Methods for Unconstrained Optimization*, Academic Press, 1972.
30. B. A. MURTAGH, *On the Simultaneous Solution and Optimization of Large-Scale Engineering Systems*, Comp. and Chem. Eng., 6, 1982, p. 1.
31. B. A. MURTAGH et R. W. H. SARGENT, *Computational Experience with Quadratically Convergent Minimization Methods*, Comp. J., 13, 1970, p. 185.
32. B. A. MURTAGH et M. A. SAUNDERS, *MINOS User's Guide*, Report SOL 77-9, Stanford Univ., CA, 1977.
33. B. A. MURTAGH et M. A. SAUNDERS, *Nonlinear Programming for Large, Sparse Systems*, Report SOL 76-15, Stanford Univ., CA, 1976.
34. B. A. MURTAGH et M. A. SAUNDERS, *Large Scale Linearly Constrained Optimization*, Math. Prog., 14, 1978, p. 41.
35. B. A. MURTAGH et M. A. SAUNDERS, *The Implementation of a Lagrangian-Based Algorithm for Sparse Nonlinear Constraints*, Report SOL 80-1, Stanford Univ., CA, 1980.
36. B. A. MURTAGH et M. A. SAUNDERS, *MINOS/AUGMENTED User's Manual*, Report SOL 80-14, Stanford Univ., CA, 1980.
37. P. V. PRECKEL, *Modules for Use with MINOS/AUGMENTED*, Report SOL 80-15, Stanford Univ., CA, 1980.
38. L. PIBOULEAU et S. DOMENECH, *Une procédure arborescente pour la séparation de mélanges complexes dans l'industrie chimique*, R.A.I.R.O., Rech. Op., 19, 1985, p. 35.
39. M. J. D. POWELL, *A Note of Quasi-Newton Formulae for Sparse Second Derivative Matrices*, Math. Prog., 20, 1981, p. 144.
40. R. N. S. RATHORE, K. A. VAN WORMER et G. J. POWELS, *Synthesis Strategies for Multicomponent Separation Systems with Energy Integration*, A.I.Ch.E.J., 20, 1974, p. 491.
41. F. R. RODRIGO et J. D. SEADER, *Synthesis of Separation Sequences by Ordered Branch Search*, A.I.Ch.E. J., 21, 1975, p. 885.
42. M. A. SAUNDERS, *Large-Scale Linear Programming Using the Cholesky Factorization*, Report STAN-CS-72-252, 1972.
43. M. A. SAUNDERS, *MINOS System Manual*, Report SOL 77-31, Stanford Univ., CA, 1977.
44. M. A. SAUNDERS, *MINOS Distribution Documentation*, Report SOL 80-100, Stanford Univ., CA, 1980.
45. L. K. SCHUBERT, *Modification of a Quasi-Newton Method for Nonlinear Equations with a Sparse Jacobian*, Math. Comp., 25, 1970, p. 27.
46. M. A. STADTHEER, G. D. GARY et R. C. ALKIRE, *Optimization of an Electrolytic Cell with the Use of a GRG Algorithm*, Comp. and Chem. Eng., 7, 1983, p. 27.
47. P. TOLLA, *Amélioration de la stabilité numérique d'algorithmes de résolution de programmes linéaires à matrices de contraintes clairsemées*, R.A.I.R.O., Rech. Op., 18, 1984, p. 19.
48. J. A. TOMLIN, *Modifying Triangular Factors of the Basis in the Simplex method, in Sparse Matrices and their Applications*, Plenum Press, 1972, p. 77.
49. J. A. TOMLIN, *On the Pricing and Backward Transformation in Linear Programming*, Math. Prog., 6, 1974, p. 42.
50. J. VIGNES, *Implémentation des méthodes d'optimisation : Test d'arrêt optimal, contrôle et précision de la solution*, R.A.I.R.O., Rech. Op., 18, 1984, p. 1.
51. P. WOLFE, *Methods of Nonlinear Programming—the Reduced Gradient method*, Recent Advances in Math. Progr., 67, 1963.