HAL
open science

Élaboration automatique de consignes pour la conduite d'un réseau de distribution d'eau

Gérard Joalland

To cite this version:

Gérard Joalland. Élaboration automatique de consignes pour la conduite d'un réseau de distribution d'eau. Automatique / Robotique. Université Paris Dauphine - Paris IX, 1978. Français. NNT : . pastel-00833892

HAL Id: pastel-00833892

https://pastel.hal.science/pastel-00833892v1

Submitted on 13 Jun 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

T H E S E

présentée

A LA FACULTE DES SCIENCES PARIS IX

par

G. JOALLAND

pour obtenir

LE GRADE DE DOCTEUR - INGENIEUR

ELABORATION AUTOMATIQUE DE CONSIGNES POUR

LA CONDUITE D'UN RESEAU DE DISTRIBUTION D'EAU

REMERCIEMENTS

Je tiens à remercier le Professeur BENSOUSSAN pour m'avoir fait l'honneur d'examiner ce travail et d'accepter la présidence de mon jury de soutenance.

Je remercie également M. FACY, ingénieur à la Société Lyonnaise des Eaux de l'Eclairage, qui a bien voulu faire partie de mon jury.

Ce travail a été préparé au Centre d'Automatique-Informatique de l'Ecole des Mines de Paris. Que son directeur, M. BERNHARD, trouve ici ma reconnaissance pour m'avoir accueilli dans son centre où j'ai trouvé un environnement amical et stimulant.

Je remercie vivement M. COHEN, qui a dirigé ce travail et qui, par son aide et ses encouragements amicaux, a permis de le mener à bien.

Enfin je tiens à remercier Mme LE GALLIC pour la réalisation matérielle de ce document.

SOMMAIRE

INTRODUCTION Page : 1 - 4

CHAPITRE I :
Présentation du problème de commande optimale Page : 5 - 16

CHAPITRE II :
Modélisation d'un réseau hydraulique Page : 16 - 26

CHAPITRE III :
Méthode de décomposition - coordination Page : 27 - 56

CHAPITRE IV :
Méthode d'agrégation - désagrégation Page : 57 - 73

CHAPITRE V :
Résultats - Conclusions Page : 74 - 91

ANNEXES Page : 92 - 140

INTRODUCTION

Nous présentons dans cette étude deux méthodes pour la résolution d'un problème de commande optimale d'un système dynamique "complexe,", un réseau de distribution d'eau potable.

Précisons tout de suite cette notion de système "complexe", dans le cas qui nous intéresse ici : il s'agit d'un système dont les caractéristiques (nature des commandes, liaison entrée-sortie non explicite ... rendent difficile voire impossible sa résolution pour les méthodes classiques de l'optimisation dynamique (Programmation dynamique ou méthodes utilisant le principe du maximum de Pontriaguine).

Les méthodes que nous proposons se réfèrent aux techniques de décomposition. Dans la première il s'agira d'une décomposition horizontale et dans la deuxième d'une décomposition verticale, notions que nous précisons plus loin (Chapitre 1).

Le réseau de distribution d'eau qui a servi de base à l'étude est la partie rive-droite de la Seine du réseau de la région parisienne Sud de la SLEE (*). Ce réseau s'étend sur les deux rives de la Seine et alimente en eau potable 94 communes, réparties sur 4 départements (Seine et Marne, Essonne, Val de Marne, Yvelines). En 1975, il desservait, outre de gros consommateurs industriels (Aéroport d'Orly, CEA Saclay, ...), 770.000 habitants.

Il possède 3 usines alimentées en eau de surface (eau de la Seine), 12 usines en eau profonde (eau de nappes phréatiques), 7 réservoirs et comporte près de 3000 Kms de canalisations. Le "sous-réseau" de la rive droite de la Seine et celui de la rive gauche sont reliés par une seule canalisation de gros diamètre (1,20m), le premier possédant 2 réservoirs et 5 usines.

(*) Société Lyonnaise des Eaux et de l'Eclairage.

Il s'agit d'élaborer une aide à la décision pour la conduite en temps réel des installations de refoulement de l'eau dans le réseau. Ceci s'inscrit dans le cadre plus général de l'automatisation de la commande dunréseau. Déjà, l'information concernant celui-ci (niveau de l'eau dans les réservoirs, état de fonctionnement des groupes de pompes, pressions statiques dans les très grosses canalisations (plus de un mètre de diamètre) est centralisée, ainsi que la prise de décision (mise en marche et arrêt des pompes) par l'intermédiaire, de lignes de télétransmission et, dans un proche avenir, un mini-calculateur réalisera un certain nombre de tâches : surveillance, alarme, édition du journal de bord, stockage de données (consommations journalières, entre autres), comme c'est déjà le cas pour certaines installations existantes : Houston (USA) [2] Chambery(France) [1] , St Gall (Suisse), ...

C'est la présence de ce mini-calculateur qui peut inciter à élaborer une politique de refoulement non seulement admissible (c'est a dire satisfaisant la demande des consommateurs) mais optimisant le coût de refoulement (somme de deux termes : l'un lié au prix de revient de l'eau, l'autre à la dépense d'énergie) tout en respectant un certain nombre de contraintes : les réservoirs doivent contenir un volume d'eau de sécurité ; la pression statique dans les canalisations ne doit être ni trop haute (pour éviter la rupture de celles-ci), ni trop basse (sinon, effets de cavitation possibles) ; il existe une limitation de pompage dans les nappes phréatiques, ...

Plusieurs problèmes, à priori similaires, ont été ou sont actuellement traités. Pour ceux-ci, le projet regroupant des Universitaires (Université de Cambridge) et des industriels (E.W.W.C) pour l'alimentation du réseau d'eau de la région de Cambridge (Grande-Bretagne) nous semble le plus ambitieux [3] Le problème d'optimisation est proche de celui que nous venons de définir. Toutefois, le réseau de distribution possède diverses caractéristiques qui n'existent pas dans celui qui sert de support à cette étude :

- présence de pompes à régime variable c'est à dire des pompes où il est possible de choisir directement le débit refoulé

(le réseau de la SLEE ne possède que des pompes en "tout ou rien").
- le coût de traitement de l'eau est identique en tous les points du réseau (dans le réseau SLEE il y a l'eau de surface au coût de traitement élevé et l'eau profonde qui ne nécessite pas ou peu de traitement).
- dans une même zone géographique, on trouve des consommateurs d'un type spécifique (courbe de la consommation au cours de la journée de type industriel, ou rural, entre autres); des points de production ; des possibilité de stockage. Ces zones homogènes constituent les noeuds du graphe, ce qui signifie qu'on ne se préoccupe pas des problèmes de transfert d'eau à l'intérieur d'une zone.
- les transferts d'eau de zone à zone sont maitrisés à l'aide de surpresseurs.

Toutes ces caractéristiques permettent une modèlisation simple, qui se limite à des équations de débits seulement. De plus, dans le projet E.W. W.C.-Cambridge-University, l'objectif semble être plutôt une régulation autour d'un régime moyen, bien connu dans la pratique (ou obtenu par une simulation interactive, en cas de modification du réseau). Dans le réseau rive-droite de la Seine, l'absence de pompes à régime variable, de surpresseurs;la répartition géographique des réservoirs (plutôt au Nord) et des usines (plutôt au Sud) rendent les problèmes de transferts d'eau très important. Ces transferts ne sont connus qu'à travers un modèle débit-pression qui nous sera indispensable pour optimiser le régime de fonctionnement moyen.

Quant aux autres réseaux automatisés, il s'agit le plus souvent de petites installations (un réservoir et une usine) ou des réseaux avec des capacités de stockage (de l'ordre de la journée ou plus) qui rendent le problème d'optimisation trivial ou, pour le moins, considérablement simplifié (*).

(*) Nous avons eu récemment connaissance d'un réseau très comparable au notre dans la région de San José (Californie) et de l'étude faite par Systems Control Inc. (Palo-Alto) [4]
Il semble cependant que le problème de l'optimisation journalière traitée dans cette thèse soit précisément celui qui n'a pas encore été résolu dans ce cas.

PLAN GENERAL

Dans le chapitre I, après une présentation du problème et des difficultés liées à sa résolution, nous décrivons succintement les deux méthodes envisagées pour sa résolution.

Le chapitre II traite de la modèlisation d'un réseau hydraulique et des méthodes de résolution des équations caractérisant un réseau.

Le chapitre III commence par une présentation du support théorique de la méthode de décomposition-coordination. Nous décrivons ensuite cette méthode dans le cas qui nous intéresse, avec toutes les simplifications et hypothèses nécessaires pour traiter le problème.

Le chapitre IV se rapporte à la deuxième méthode choisie pour la résolution du problème (nous l'appellerons méthode d'agrégation-désagrégation). Après un présentation théorique dans un cas particulier, nous présentons l'algorithme pratique pour la résolution du problème du refoulement optimal.

Le chapitre V est un chapitre de présentation des résultats obtenus avec les deux méthodes et où nous tirons quelques conclusions.

CHAPITRE I : PRESENTATION DU PROBLEME DE COMMANDE OPTIMALE

I - 1. Introduction

La détermination des consignes optimales de refoulement peut être formulée comme un problème de commande optimale (déterministe) d'un système dynamique :

- la hauteur d'eau de chaque réservoir ou le volume d'eau dans une nappe phréatique constitue une composante de l'état du système. En effet, si S_i désigne la section du $i^{ème}$ réservoir (supposé avoir une forme cylindrique), et q_i le débit entrant ou sortant de ce réservoir on a, à l'instant t :

$$S_i . \overset{o}{x}_i(t) = q_i(t)$$, x_i désignant la hauteur d'eau dans le réservoir i et $\overset{o}{x}_i$ sa dérivée. (idem pour une nappe qui est aussi, en fait, un "réservoir").

- Les commandes consistent pour chaque groupe de pompes de refoulement installées aux usines, en le choix des configurations de marche-arrêt de chacune des pompes, à tout instant, parmi les configurations autorisées.

- Les consommations aux divers points du réseau (modélisées par un débit total, modulé dans le temps, réparti géographiquement selon des "poids" fixes et connus) constituent les entrées non commandées du système.

- Les contraintes ont déjà été évoquées précédemment : la pression statique dans les canalisations doit être comprise entre un minimum et un maximum ; la hauteur d'eau dans un réservoir ne doit pas descendre en dessous d'un minimum (elle ne dépasse pas non plus un maximum : un clapet ferme le réservoir s'il est plein) ; les nappes phréatiques ne doivent pas se vider ; les pompes ne peuvent être arrêtées puis mises en route dans un délai de moins d'une heure.

- le critère à minimiser est somme de deux termes : le coût de l'énergie électrique nécessaire au refoulement de l'eau dans le réseau ; un coût de traitement de l'eau.

- La période d'optimisation est de 24 heures. L'instant initial correspondra à un changement de tarification de l'électricité (6h ou 22h), afin de faciliter le choix de l'état initial et de la cible finale.

1 - 2. Formalisation du Problème

I - 2 -a. Modélisation du système dynamique

Nous désignons par $\mathcal{U}$ l'ensemble des commandes. Soit m le nombre d'usines de refoulement et l_i $i=\overline{1,m}$, le nombre de groupes de pompes pouvant refouler à l'usine i, numérotés de 1 à l_i.

On a : $\mathcal{U} = \{u/u = (u_1,\ldots,u_i,\ldots,u_m), u_i \in \{0,1,\ldots l_i\}\}$

sachant que $u_i = 0$ signifie qu'aucune pompe ne refoule à l'usine i, pour la configuration u.

Posons $T = \{0,1,\ldots,23,24\}$. T désigne en fait la période d'optimisation, discrétisée en pas de temps de 1 heure (qui correspond au délai séparant l'arrêt et la mise en marche des pompes).

Remarque. Le choix de la décomposition de la période d'optimisation en 24 pas d'une heure peut paraître arbitraire. En effet, le fait de ne pouvoir arrêter et mettre en route une pompe trop souvent peut aussi être pris en compte en ajoutant au critère du problème un coût de mise en marche des pompes, l'intervalle de temps n'étant plus discrétisé, du moins dans la théorie : on résoud alors un problème de contrôle impulsionnel [8], [9]. Notre choix a le mérite de traduire directement le voeu exprimé par l'utilisateur sans chercher à ajuster un coût de démarrage difficle à saisir.

Soient n le nombre de réservoirs, et p le nombre d'usines où les pompes refoulant de l'eau provenant de nappes phréatiques.

Soient $x_t = (x_t^1, \ldots, x_t^n)$ et $y_t = (y_t^1, \ldots, y_t^p)$, x_t^i désignent la hauteur d'eau dans le réservoir i et y_t^j le volume d'eau dans la nappe j, à l'instant t, $t \in T$.

On a les équations d'état suivantes :

(1) $$x_{t+1}^i = x_t^i + \mathfrak{z}_t^i(x_t, u_t)/S_i$$

(2) $$\mathcal{Y}_{t+1}^j = \mathcal{Y}_t^j - \mathfrak{s}_t^j(x_t, u_t) + D_t^j(\mathcal{Y}_t^j)$$

$t \in T - \{24\}$

$u_t \in \mathcal{U}$

$i \in \{1, \ldots, n\}$

$j \in \{1, \ldots, p\}$

$\mathfrak{s}_t^j(x,u)$ désigne le volume d'eau refoulé par les pompes de l'usine j durant l'intervalle de temps $[t,t+1[$, pour la commande u, l'état du $i^{\text{ème}}$ réservoir à l'instant t étant x^j, $j \in \{1, \ldots, n\}$

D_t^j "résumé" les apports d'eau dans la nappe j, dans l'intervalle $[t,t+1[$. Ce terme qui caractérise en fait la nappe est supposé connu. On suppose que ces D_t^j sont indépendants entre eux .

$\mathfrak{z}_t^i(x,u)$ désigne le volume d'eau entrant ou sortant du $i^{\text{ème}}$ réservoir dans l'intervalle $[t,t+1[$ pour la commande u, $u \in \mathcal{U}$, si le niveau dans le $j^{\text{ème}}$ réservoir, à l'instant t, x^j, $j \in \{1, \ldots, n\}$.

S_i désigne la section du $i^{ème}$ réservoir (tous les réservoirs sont supposés cylindriques, pour simplifier).

Connaissant x et u, la détermination de $\mathfrak{s}_t^j(x,u)$ et $\mathfrak{z}_t^i(x,u)$ se fait par la résolution d'un certain nombre d'équations implicites liant les débits et les pressions dans le réseau de distribution d'eau (cf. chapitre II).

Comme la section d'un réservoir est importante, on suppose que le débit d'eau entrant ou sortant d'un réservoir durant un pas de temps est constant, pour une consigne donnée (la hauteur d'eau dans un réservoir varie de moins de 30 cm en 1 heure, ce qui entraîne une variation de 3 % sur la pression statique au "pied" du réservoir).

Si $Q_t^i(x,u)$ désigne le débit au réservoir i dans l'intervalle $[t,t+1[$ pour la consigne u, $u_t \in \mathcal{U}$, et une hauteur d'eau x^j, dans le réservoir j, à l'instant t, le fait que le réservoir i est muni d'un clapet s'écrit :

$$(3) \qquad \mathfrak{z}_t^i(x,u) = \text{Min}\ \{(x_{max}^i - x^i).S_i \quad , \quad Q_t^i(x,u).\tau\}$$

avec : $\begin{cases} x_{max}^i : \text{hauteur maximale dans le réservoir } i. \\ \tau : \text{durée de l'intervalle } [t,t+1[, \text{ en seconde.} \end{cases}$

(si le pas de temps est de 1 heure, $\tau = 3600$ secondes).

I.2.b. <u>Modélisation des contraintes</u>

1) D'un point de vue "géographique", un réseau hydraulique est un sous-ensemble $\mathcal{R}$ d'un plan $\mathcal{P}$.

Soit P_a, $a \in \mathcal{R}$, la pression statique en un point a du réseau.

On a la contrainte :

$$(4) \qquad P_a \in [P_{min}, P_{max}]$$

avec P_{min} : pression minimum tolérée

P_{max} : pression maximum (pour éviter la rutpture des canalisations).

2) Le niveau x^i dans le réservoir i ne doit pas descendre en dessous d'un niveau minimum de sécurité x_{min} :

(5) $$x^i \geq x^i_{min} \quad , \quad i \in \{1,\dots,n\}$$

3) Le volume d'eau γ^j dans la nappe j est supérieure à un minimum γ^j_{min} :

(6) $$\gamma^j \geq \gamma^j_{min} \quad , \quad j \in \{1,\dots,p\}$$

4) La cible finale est décrite par :

(7) $$x^i_{24} \in [\bar{x}^i_1, \bar{x}^i_2]$$

Nous ne pouvons choisir une cible ponctuelle $(\bar{x}^i_1 = \bar{x}^i_2)$ car nous disposons de commandes discrètes : le problème n'aurait presque surement pas de solution dans ce cas et même si nous choisissons une cible trop petite, il est encore possible de ne pas obtenir de solution.

En ce qui concerne le choix de x^i_1 et x^i_2, il dépend de l'instant initial de l'optimisation. Si celui-ci correspond au début d'une période de tarification électrique forte, on prendra $\bar{x}^i_2 = x^i_{max}$ (et $\bar{x}^i_1$ "légèrement" plus petit que $\bar{x}^i_2$) ; s'il correspond au début d'une période de tarification basse (début de la nuit), on prendra $\bar{x}^i_1 = x^i_{min}$. Le choix peut paraître raisonnable surtout pour de faibles capacités de stockage (ce qui est le cas ici) et il permet de tourner la difficulté de l'optimisation sous contrainte de périodicité en préjugeant de certains points de passage de la trajectoire optimale.

1 - 2.c. <u>Critère</u>

Soient : $s^l_t(x_t,u_t)$ le volume d'eau fourni par l'usine l , $l = \overline{1,m}$, dans l'intervalle $[t,t+1[$, pour la configuration u_t , $u_t \in U$ et une hauteur d'eau x^i_t dans le réservoir i;

$H_t^l(x_t,u_t)$ l'élévation de pression apportée par les groupes de pompes qui refoulent en l et $\eta_t^l(x_t,u_t)$ leur rendement.

Soient A_t^l le coût unitaire de l'énergie électrique à l'usine l, à l'instant t, et B^l le coût marginal de traitement d'un m^3 d'eau (*).

Le critère $\mathcal{J}$ s'écrit :

$$(8) \qquad \mathcal{J} = \sum_{t\in T-\{24\}} \sum_{l=1}^{m} A_t^l \cdot \frac{s_t^l(x_t,u_t)\cdot H_t^l(x_t,u_t)}{\eta_t^l(x_t,u_t)} + B^l \cdot s_t^l(x_t,u_t)$$

Le problème de la détermination des consignes optimales de refoulement dans un réseau hydraulique peut donc s'écrire :

$$(Pb) \qquad \min_{u_t\in \mathcal{U}} \{\mathcal{J}\}$$

sous les contraintes : $\{(1),(2),\ldots,(7)\}$

<u>Remarque</u> : Nous n'avons pas considéré un problème de commande stochastique bien que l'évolution de la consommation ne soit pas déterministe. D'une part nous ne disposions pas de modèle stochastique pour celle-ci, au cours de l'étude ; le problème s'en trouverait notablement compliqué d'autre part. Toutefois, afin de prendre en compte les "petites" perturbations de la consommation autour de la trajectoire prévue, nous élaborerons des commandes en boucle fermée.

(*) Le coût de l'énergie électrique peut varier selon les usines car il dépend de la puissance souscrite. Toutefois, les variations de ce coût selon les usines sont faibles et on peut considérér que $A_t^l = A_t$, $\forall l \in \{1,\ldots,m\}$.

I - 3. Difficultés de résolution

Si les capacités de stockage sont supérieures à la consommation durant la plus longue période de tarification électrique haute et si les possibilités de refoulement permettent de remplir suffisament les réservoirs durant la période de tarification basse, le problème devient simple : il suffit de refouler l'eau durant les périodes ou l'énergie électrique est peu chère et de vider les réserves durant les périodes où celle-ci est chère. Ceci ne résoud pas, de toute façon, le problème de la répartition géographique optimale (ou admissible).

Hormis ce cas, le problème de commande optimale présente de nombreuses difficultés :

- commande discrète
- discontinuité de la dynamique
- contraintes d'état
- liaison implicite état-commande.

Aussi est-il peu envisageable de le résoudre par méthodes utilisant le principe du maximum de Pontryaguine : il semble en effet très difficile de déterminer le vecteur adjoint de la dynamique (on envisagera quand même une méthode heuristique pour sa détermination, chapitre VI). De plus cette méthode ne peut être qu'heuristique puisque la théorie ne s'applique que si la dynamique est continue.

La programmation dynamique parait alors être la méthode de résolution. Il se pose alors une difficulté importante liée à l'utilisation de cette méthode : le volume des calculs devient trop lourd pour obtenir sur un calculateur la résolution d'un tel problème en un temps "raisonnable".

En effet, il faut résoudre N fois les équations du réseau définissant la relation implicite état-commande (1/2 seconde d'unité centrale sur un ordinateur moyen, type IRIS 80

Or :

$$N = \prod_{i=1}^{m} (1+l_i) \cdot \prod_{j=1}^{(n+p)} r_j \cdot 24$$

avec : r_j : nombre de discrétisations de la $j^{ème}$ composante du vecteur d'état (niveau d'eau dans un réservoir ou volume d'eau dans une nappe phréatique).

Si on reprend les évaluations pratiques correspondant au réseau rive droite de la Seine de la SLEE [34], on trouve des temps de calcul dépassant le siècle !

I - 4. Présentation sommaire des méthodes de résolution

1) Décomposition horizontale

La décomposition horizontale a été abordée par de nombreux auteurs. Citons, entre autres, Mesarovic et al. [10] Takahara [11], Cohen [12], Titli [13].

Cette méthode de décomposition des calculs tient compte de la structure du système : le réseau est considéré comme la concaténation de plusieurs sous-réseaux interconnectés (ce qui d'ailleurs correspondait à une situation réelle autrefois). Sur chacun des sous-réseaux est défini un problème de commande optimale de système dynamique. Pour un sous-système, l'ensemble des commandes est formé des différentes configurations de pompes qui peuvent refouler dans le sous-réseau correspondant ; l'état comprend les niveaux dans les réservoirs et les volumes dans les nappes phréatiques du sous-réseau ; les contraintes sont celles évoquées plus haut.

Les méthodes de décomposition horizontale distinguent généralement deux niveaux de calcul :

- un niveau global - dit niveau coordonnateur - où sont définis les problèmes locaux et où sont gérés les interactions entre sous-systèmes.
- un niveau local où sont résolus les sous-problèmes.

Après chaque résolution locale, l'information sur les différents sous-réseaux est transférée au niveau coordonnateur qui définit la valeur des interactions et les critères des différents sous-systèmes, pour une nouvelle résolution locale. Bien entendu, les sous-problèmes sont définis pour qu'il soit aisé de déterminer la solution du problème initial à partir des

solutions des différents sous- problèmes (en admettant que la procédure que nous venons de définir devient stationnaire au bout d'un certain nombre de transferts "niveau local - niveau global").

La figure 1 représente la décomposition horizontale, telle que nous venons de la présenter.

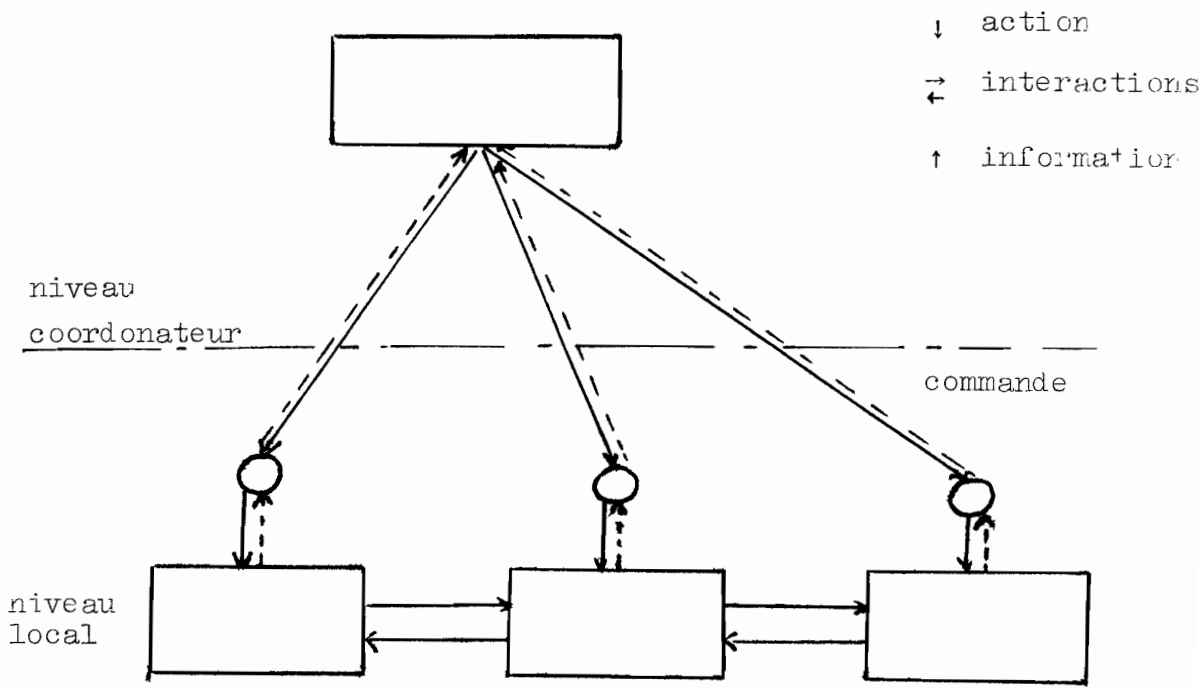


Fig 1.

Il est facile de voir que la taille des problèmes locaux est notablement réduite. Désignons par k le nombre de sous-réseaux, n_i le nombre de réservoirs, p_i le nombre d'usines alimentées par des nappes phréatiques et m_i le nombre total d'usines dans le sous-réseau i $(i=1,\ldots,k)$. Au niveau local, la résolution des k sous-problèmes par la programation dynamique requiert Nd résolutions des équations des sous-réseaux (temps de calcul bien sûr inférieur au temps de calcul correspondant au réseau global), avec :

$$N_d = [\sum_{i=1}^{k} (\prod_{l=1}^{m_i} L_l^i).(\prod_{p=1}^{(n_i+p_i)} N_p^i)24]$$

L_l^i : nombre de commandes à la $l^{ème}$ usine du sous réseau i.

N_l^i : nombre de discrétisations de $l^{ème}$ variable d'état du sous-réseau i.

<u>Application numérique</u>

- <u>Réseau global</u>. 4 réservoirs ; 3 nappes ; 6 usines ; 10 pas de discrétisation pour les réservoirs et 5 pour les nappes ; 10 commandes par usine.

$$N = 10^6.10^4(5)^3.24 = 3(10)^{13}.$$

- 4 <u>sous-réseaux</u>. <u>Sous réseau 1 et 2</u> : 1 réservoir ; 1 nappe, 2 usines.

  <u>Sous-réseau 3</u> : 1 réservoir, 1 usine, 1 nappe.

  <u>Sous-réseau 4</u> : 1 réservoir, 1 usine.

$$N_d = [2.(10^2.10.5)+(10.10.5)+10.10].24$$

soit : $N_d < 3\ 10^5$

Dans notre exemple nous avons $\frac{N}{N_d} > 10^8$, ce qui veut dire que nous pouvons réaliser 10^8 pas de coordination (nombre de transferts niveau coordonnateur-niveau local), au moins, pour obtenir des temps de calcul équivalents avec la résolution globale et celle par décomposition-coordination (en négligeant les temps de calcul de la coordination et, des transferts).

Outre la réduction considérable du volume des calculs, la décomposition horizontale présente d'autres avantages :

- possibilités de calculs parallèles (multiprocesseurs).
- réduction de la mémoire centrale nécessaire.
- structure souple permettant de s'adapter à la croissance du système (modification d'un ou plusieurs sous-systèmes : adjonction de sous-systèmes).

- enfin, dans une mise en oeuvre en ligne, répartition de la décision (informatique répartie-décentralisation).

2) Décomposition verticale

Cette méthode, souvent appelée "méthode hiérarchisée", est liée à l'existence de différents niveaux d'information sur le système considéré, selon le point de vue d'où l'on se place.

A notre connaissance peu de résultats mathématiques significatifs ont été obtenus dans ce domaine. Citons les travaux d'Aoki [14] sur la construction de modèles plus ou moins agrégés d'un système, de Bertand et al. [15], de Davison [16].

La méthode que nous proposons -qui nous semble originale- distingue deux niveaux d'approche d'un réseau de distribution d'eau :

- un niveau "fin" qui est celui que nous avons présenté plus haut. Ce modèle correspond aux équations (1) et (2) dont la résolution passe par la détermination des équations hydrauliques du réseau.
- un niveau "agrégé" que ne s'intéresse qu'à des bilans de débit du type :

$$(\alpha) \qquad X_{t+1} = X_t + Q_t - C_t$$

X_t : réserve globale à l'instant t.

Q_t : volume d'eau refoulé dans l'intervalle de temps $[t,t+1]$.

C_t : Consommation globale dans l'intervalle $[t,t+1]$.

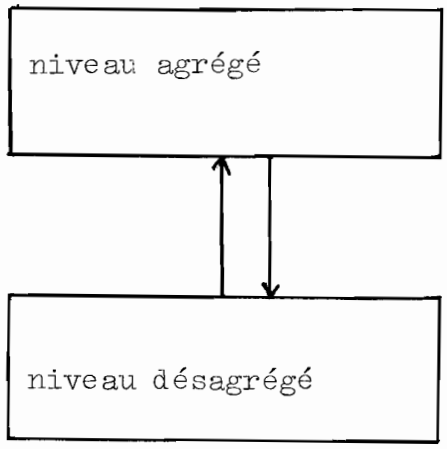
L'équation (α) est la version discrétisée de l'équation continue correspondante. Nous retiendrons le pas de temps d'une heure, comme précédemment.

La méthode distingue deux niveaux de calcul :

- un niveau "haut" où l'on résoud un problème de commande optimale d'un système dynamique agrégé.
- un niveau "bas" ou l'on cherche à répartir le refoulement

agrégé entre les différentes usines.

Il s'agit d'une procédure itérative où les deux niveaux échangent de l'information : le niveau bas élabore le critère agrégé du niveau haut ; celui-ci fournit le critère de désagrégation de la commande au niveau bas.



Avec cette méthode le gain sur les temps de calcul sont également très importants.

En effet, le problème de commande optimale est fait sur un modèle à une seule variable d'état et avec équation d'état très simple. Le problème de désagrégation est une séquence de 24 problèmes statiques (avec résolution des équations hydrauliques du réseau) et demande donc M fois moins de temps que la résolution globale par la programation dynamique avec

$M = \prod_{i=1}^{n+p} r_i$ (avec les mêmes notations que précédemment).

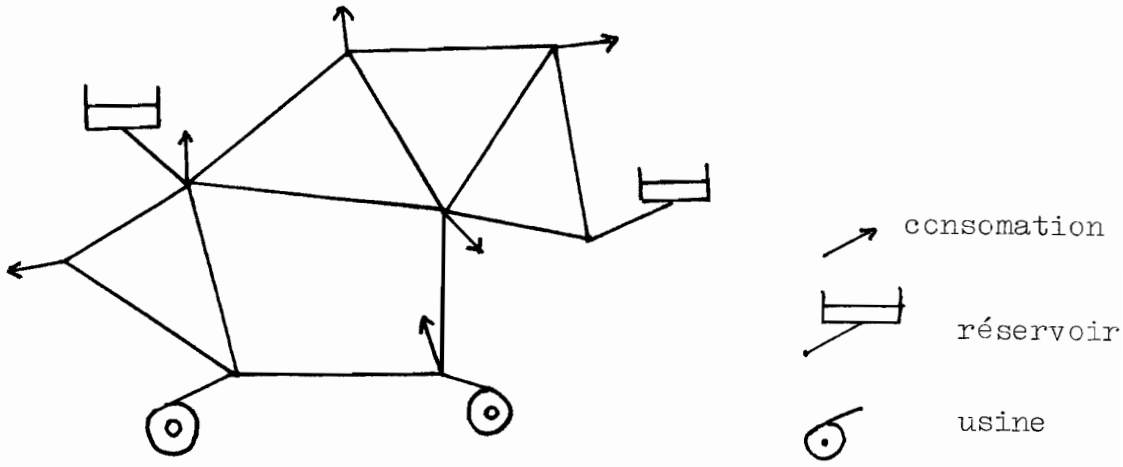
CHAPITRE II : MODELISATION D'UN RESEAU HYDRAULIQUE

II - 1 Généralités

Un réseau hydraulique peut être modèlisé comme un graphe valué :

. les sommets du graphe (ou noeuds) sont les connexions de canalisations, les points de stockage (réservoirs) et de refoulement (usines).

. les arcs (ou "branches") représentent les canalisations.



La consommation d'eau est, dans la pratique, répartie tout au long des canalisations. Pour simplifier, nous la supposons concentrée aux noeuds du réseau, hypothèse cohérente avec la précision recherchée et la finesse avec laquelle nous connaissons la répartion géographique de cette consommation.

II - 2 Equations hydrauliques

Supposons fixés, à un instant donné, la configuration de pompes qui refoulent dans le réseau, le volume d'eau dans chaque réservoir, le débit total de la consommation et sa répartition entre les différents noeuds.

Pour avoir alors une image instantanée du réseau hydraulique, il faut connaître les débits transitant dans les canalisations, refoulés par les pompes, entrant ou sortant des réservoirs, ainsi que la pression statique en chaque point de connexion de canalisations (*).

Ces débits et ses pressions sont liés par un ensemble de relations : les équations hydrauliques du réseau.

1) Equations de bilan de débit aux noeuds.

Ce sont les équations,de conservation de la masse, appliquées à tous les noeuds du réseau, pour un fluide incompressible : la somme algébrique des débits entrant et sortant d'un noeud est nulle :

$$(\alpha) \qquad \sum_{j \in I_i} Q_{ij} + C_i + T_i = 0$$

I_i : Ensemble des noeuds liés au noeud i par une branche

Qi_j : débit transitant dans la branche (i,j); $Q_{ji} = -Qi_j$

T_i : débit, autre que la consommation, au noeud i

La consommation C_i est une donnée. En tout noeud i qui n'est pas un point de stockage ou de refoulement le débit T_i est nul. Par convention, le débit en un noeud est positif s'il est "entrant" (dans le réseau), négatif s'il est "sortant".

(*) Il faudrait théoriquement connaitre la pression statique en tout point d'une canalisation. En fait, sauf cas très rares (point intermédiaire d'altitude élevée ou très basse par rapport à ses extrémités)elle est encadrée par les pressions aux deux extrémités de la canalisation.

2) Equations de perte de charge dans les branches

Elles caractérisent la dissipation d'énergie dans les canalisations, due aux frottements du fluide sur les parois.

La perte de charge est supposée proportionnelle au carré du débit (au signe près) :

$$(\beta) \qquad -(p_j+h_j) + (p_i+h_i) = R_{ij} \cdot Q_{ij} |Q_{ij}|$$

p_j : pression statique au noeud j

h_j : altitude du noeud j (par rapport à un niveau 0 de référence).

R_{ij} : "résistance" de la branche (i,j) ; $R_{ji} = R_{ij}$

Q_{ij} : débit dans la branche (i,j)

$H_i = (p_i+h_i)$ s'appelle la charge au noeud i. Le coefficient R_{ij} dépend de la rugosité, de la longueur et du diamètre intérieur de la canalisation (i,j). Par convention Q_{ij} est positif s'il transite de i vers j, négatif sinon.

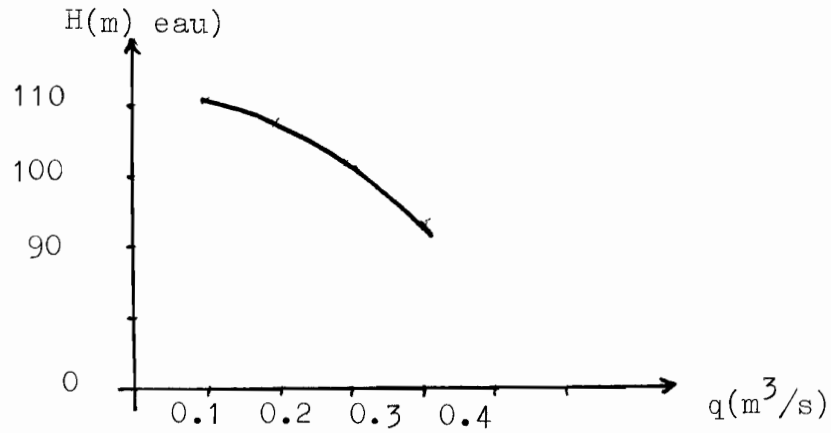
3) Equations caractéristiques des pompes

Si, au noeud i, une ou plusieurs pompes refoulent dans le réseau, le débit refoulé T_i et la charge H_i sont liées par la relation :

$$(\gamma) : \quad H_i = f_i(T_i)$$

Une bonne approximation de f_i est donnée par :

$$f_i(q) = a_i + b_i q + c_i q^2 \text{ , avec } c_i \leq 0.$$



4) Equations de charge au réservoir

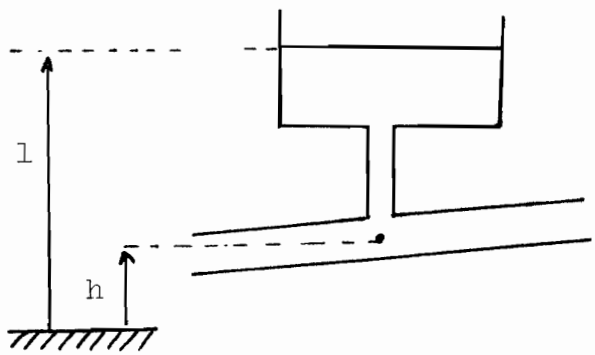
Si le noeud j est un point avec réservoir, la pression statique dans la canalisation est liée à la hauteur d'eau dans le réservoir par la relation :

$$(\delta) \quad p_j + h_j = l_j$$

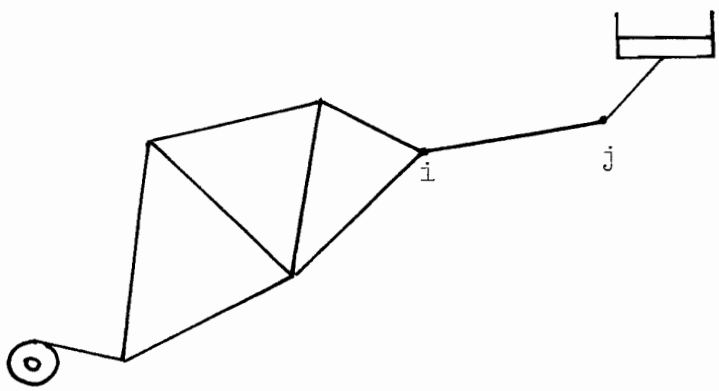
P_j : pression statique dans la canalisation, au noeud j

h_j : altitude au pied du réservoir

l_j : hauteur d'eau dans le réservoir



Remarque. On néglige la perte de charge dans la canalisation liant le réservoir au réseau. Il est possible d'éviter cette approximation : il suffit de considérer le noeud avec réservoir comme lié au réseau par cette canalisation.



Pour la résolution de cet ensemble d'équation $\{\alpha, \beta, \gamma, \delta\}$, nous pouvons distinguer trois sortes de noeuds :

i) les noeuds où le débit T_i est nul. La pression statique est alors la variable à déterminer.

ii) les noeuds équipés de pompes en fonctionnement : la charge et le débit sont tous les deux à déterminer (mais nous disposons d'une relation supplémentaires du type γ).

iii) Les noeuds avec réservoir ou la pression statique est imposée : c'est le débit qui est l'inconnue.

Notons qu'un noeud avec usine peut être du type i) si aucune pompe ne refoule en ce noeud pour la configuration choisie. De même, un noeud avec réservoir sera du type i) si le clapet de sécurité est fermé (ce clapet se ferme si le réservoir est plein pour éviter qu'il ne déborde).

II - 3 Résolution des équations hydrauliques

II -3-1 Méthode de Hardy-Cross

Si les équations (α) sont satisfaites (équations de bilans de débits), $\{(\beta), (\partial), (\delta)\}$ le seront aussi si un certain nombre de chemins du réseau vérifient un relation de perte de charge de la forme :

$$(\Pi) \quad \sum_{i \in E_k} R_i Q_i |Q_i| = H_k^{in} - H_k^{fin}$$

E_k : Ensemble des branches composant le chemin
R_i : résistance de la branche i
Q_i : débit dans la branche i
H_k^{in} : charge au noeud initial du chemin k
H_k^{fin} : charge au noeud final.

Si le réseau possède N mailles indépendantes (*), NRES réservoirs (NRES ≥ 1) et NPØM noeuds avec pompe en fonctionnement, il faut vérifier NMA relations du type (Π) avec

$$NMA = (N + NPØM + NRES - 1) \;;$$

. N chemins sont les mailles indépendantes choisies. Dans ce cas, on a bien sûr : $H^{in} - H^{fin} = 0$.

. NPØM chemins lient les NPØM noeuds avec pompes au "noeud de référence" (qui sera un noeud avec réservoir). Alors on a : $H^{in} = f(q)$, f désigant la caractéristique débit-pression du groupe de pompes refoulant au noeud initial du chemin et H^{fin} est la charge au noeud de référence.

. Si NRES ≥ 2 , (NRES - 1) chemins lient les noeuds avec réservoir au noeud de référence. H^{in} est alors la hauteur d'eau au réservoir du noeud initial.

REMARQUE : La charge est une notion relative : elle dépend du niveau de référence à partir duquel sont évaluées les altitudes. Nous désignons par noeud de référence le noeud dont la charge sert à évaluer les autres charges : celui avec réservoir de plus grosse section (donc qui a une hauteur d'eau évoluant le plus lentement à priori) nous semble le mieux convenir pour servir de référence (de pression).

Dans la méthode de Hardy-Cross on suppose que les équations (α) sont résolues et on résoud alors les équations (Π) par une méthode de Newton relaxée.

(*) Une maille est un chemin fermé, le noeud initial est confondu avec le noeud final. Des mailles sont dites indépendantes si chacune est seulle à posséder une branche du réseau. Si le réseau comporte NO noeuds et NB branches, il a N = (NB-NO+1) mailles indépendantes.

Chemin après chemin, on détermine la modification δq à apporter au débit de chaque branche du chemin pour que, au premier ordre près, la relation (Π) soit satisfaite.

1) <u>Le chemin n° k est une maille</u>

$$\delta q^k = \frac{-\sum_{i \in E_k} R_i Q_i |Q_i|}{2 \sum_{i \in E_k} R_i |Q_i|}$$

2) <u>Le noeud initial du chemin k est équipé d'un groupe de pompes en fonctionnement</u>

Si la caractéristique débit-pression du groupe de pompe est : $f(q) = a + bq + cq^2$

$$\delta q^k = \frac{-f(S) + H^{hin} - \sum_{i \in E_k} R_i Q_i |Q_i|}{2 \sum_{i \in E_k} R_i |Q_i| - (b+2cS)}$$

3) <u>Il y a un réservoir au noeud initial du chemin k</u>

$$\delta q^k = \frac{-H^{in} + H^{fin} - \sum_{i \in E_k} R_i Q_i |Q_i|}{2 \sum_{i \in E_k} R_i |Q_i|}$$

. Cette modification δq est ajouter aux débits Q_i, $i \in E_k$, et au débit du noeud initial ; elle est à retrancher de celui du noeud final.

. La procédure est arrêtée quand la modification de débit apportée est jugée suffisamment faible. Pour connaître alors la charge aux différents noeuds, il suffit de résoudre les équations de perte de charge (β) pour les branches d'un chemin passant par tous les noeuds du réseau et ayant comme noeud initial le noeud de référence.

REMARQUE. 1) Au cours de l'algorithme, les équations de bilan de débit (α) restent satisfaites. En effet, supposons qu'au noeud i on ait :

$$(\alpha_i) \qquad \sum_{l \in I_i} Q_{il} + C_i + T_i = 0$$

Soit Sq^k la modification apportée au chemin k. Si i est un neoud de ce chemin, un nombre pair de débits de l'équation ci-dessus vont être modifiés consécutivement à la correction Sq^k. En effet :

. Si dans le chemin k, i n'est pas un noeud terminal, un noeud m et un autre n sont liés a i par une branche :

Q_{im} devient $(Q_{im} + \delta q^k)$ et Q_{ni} devient $(Q_{ni} + \delta q^k)$

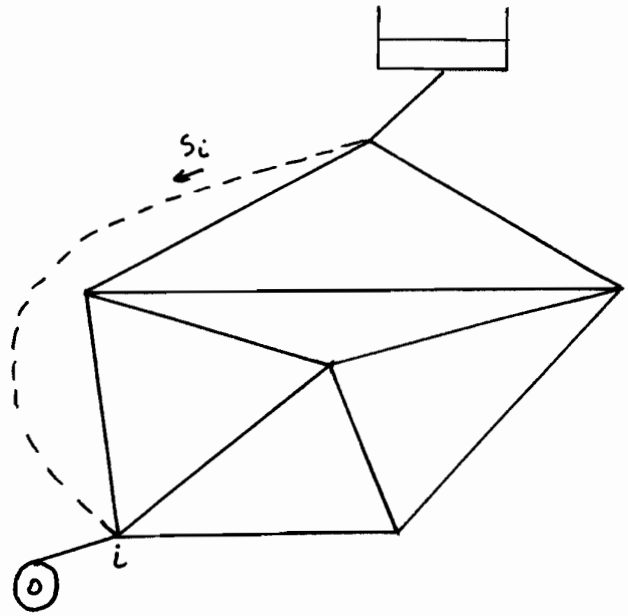
Comme $Q_{ni} = -Q_{in}$ l'équation (α_i) ne sera pas modifiée.

. Si c'est le noeud initial du chemin k, un noeud le suit (dans le sens de parcours du chemin) : Q_{im} devient $(Q_{im} - \delta_q)$ et T_i devient $(T_i + \delta_q)$, donc (α_i) reste vraie (même raisonnement si i est le noeud final du chemin k).

Nous ne connaissons pas de démonstration de convergence de la méthode de Hardy-Cross. Toutefois nous avons eu l'occasion de vérifier sa robustesse et sa rapidité de convergence (par des réseaux comportant une trentaine d'équations du type (Π)).

. Dans la littérature concernant les réseaux hydrauliques [5], [6], nous avons seulement trouvé la description de la méthode de Hardy-Cross dans le cas particulier où le réseau ne possède qu'un seul réservoir et où les caractéristiques de pompe sont des fonctions constantes (c'est à dire que l'on fixe leur débit) : la méthode consiste alors à la résolution des

des équations concernant les mailles indépendantes du réseau, uniquement. Notre extension de la méthode revient à rajouter des branches fictives au réseau liant les noeuds avec pompes au réservoir au noeud de référence. La branche rajoutée à l'un de ces noeuds caractérise la pompe ou le réservoir situé au noeud considéré (fig.5).



----- branche fictive

II) On trouvera en annexes, la description d'un programme DEDEB, utilisant cette méthode. Un autre programme PINIT permet de déterminer un chemin passant par tous les neouds du réseau (pour le calcul rapide des charges aux noeuds, une fois connus les débits) et un troisième DEMA1 choisit les chemins liant les noeuds avec pompe ou réservoir au noeud de référence. (On explore en préordre un arbre formé des noeuds du réseau).

II - 3 - 2. Autres méthodes de résolution

Parmi les autres méthodes de résolution des équations hydrauliques d'un réseau, certaines supposent que les équations de perte de charge (β) sont satisfaites [7] . Les pressions sont alors les variables. Il s'agit de résoudre des équations de la forme :

$$(\mu)\quad \varphi_i(H) = \sum_{j \in I_i} \operatorname{sgn}(H_i - H_j)\sqrt{|H_i - H_j|} + C_i + g_i(H_i) = 0$$

$\operatorname{sgn}(x) = 1$ si $x \geq 0$; -1 si $x < 0$

H_i : charge au noeud i ; $H = (H_1, H_2, \ldots,)$

C_i : consommation au noeud i

$g_i(H_i)$: débit, autre que la consommation, pour la charge H_i. $g_i \equiv 0$ si i est un noeud sans pompe ni réservoir).
Si i désigne un noeud avec réservoir, $\varphi_i(H) = \text{cste}$.

La résolution de telles équations, par minimisation d'une fonctionnelle ψ de la forme $\psi(H) = \sum_i (\varphi_i(H))^2$ n'a pas donné de résultats satisfaisants (convergence très lente ou divergence).

La méthode de minimisation que nous avons utilisée était une méthode gradient conjugué. La raison de la divergence est sans doute la non-differentiabilité de la fonction ψ.

CHAPITRE III : METHODE DE DECOMPOSITION-COORDINATION

Nous présentons dans ce chapitre la première des deux méthodes utilisées pour tenter de résoudre notre problème : la décomposition-coordination.

Après un court paragraphe d'introduction, nous reprenons l'essentiel d'un article de Cohen [17] qui contient une présentation plus récente d'un précédent algorithme par Cohen-Joalland [18] qui sert de support théorique à la méthode utilisée pour la résolution de notre problème. L'objet du troisième paragraphe est d'appliquer la théorie au cas concret qui nous intéresse avec toutes les simplifications et heuristiques que nécessitent cette application. Le dernier paragraphe est consacré aux problèmes que posent la décomposition du réseau en sous-réseau.

III - 1 Introduction

De nombreux auteurs ont utilisé les techniques de décomposition-coordination pour la résolution de "grands" problèmes d'optimisation [12], [13], [19], [20].

La méthode que nous décrivons s'inscrit dans le cadre de la coordination par prédiction des interactions, pour reprendre la terminologie de Mesarovic et al [10].

Le principe de coordination est utilisé ici pour la résolution d'un problème de minimisation sous contraintes. Le critère à minimiser et les contraintes sont tous les deux décomposés : on forme ainsi plusieurs problèmes de minimisation sous contraintes, mais supposés plus faciles à résoudre (car de tailles plus petites que celle du problème initial).

L'application du formalisme développé au problème de la résolution d'un problème de commande optimale d'un système dynamique, permet de retrouver, entre autres, l'algorithme que propose Takahara [11] pour la résolution par décomposition-coordination, du problème linéaire-quadratique (Les critères de convergences proposés permettent de donner des conditions suffisantes de convergence de cet algorithme).

En ce qui concerne l'utilisation de cette méthode à la résolution du problème de la détermination des consignes de refoulement dans un réseau hydraulique - et, rappelons-le, il s'agit l'une utilisation heuristique de la théorie -, les deux questions essentielles qui se posent sont les suivantes : l'algorithme utilisé converge-t-il ? Si oui, les temps de calcul sont-ils "raisonnables" ? A cette dernière question, seule l'expérimentation apportera une réponse. Et si les conditions de convergence proposées pour le problème linéaire-quadratique ne peuvent être utilisées pour répondre à la première, elles lui apportent néanmoins un éclairage intéressant.

III - 2 Le principe du problème auxiliaire et son application aux algorithmes de décomposition-coordination.

III - 2a - Le principe du problème auxiliaire

Soient $\mathcal{U}$ un espace de Banach dans R ; J une fonctionnelle sur $\mathcal{U}$, semi continue inférieurement (s.c.i) convexe et différentiable ; J_1 une seconde fonctionnelle sur $\mathcal{U}$, s.c.i. et convexe ; $\mathcal{U}^f$ un sous ensemble de $\mathcal{U}$, convexe et fermé.

Soit alors le problème principal P.P. :

$$(\text{P.P.}) : \quad \min_{u \in \mathcal{U}^f} \{J(u) + J_1(u)\}$$

Nous considérons maintenant K une autre fonctionnelle sur $\mathcal{U}$, ayant les mêmes propriétés que J, et une famille

$(G_\varepsilon^v)_{\substack{v \in \mathcal{U} \\ \varepsilon \in R^+}}$ de fonctionnelle sur $\mathcal{U}$, définie comme suit :

$$(1) \quad \forall u \in \mathcal{U}, G_\varepsilon^v(u) = K(u) + \langle \varepsilon J'(v) - K'(v), u \rangle + \varepsilon J_1(u)$$

avec :

$\langle \cdot, \cdot \rangle$: produit de dualité entre $\mathcal{U}$ et son dual topologique $\mathcal{U}^*$.

$J'(u)$: dérivée de Gateaux (où à fortiori de Fréchet) $J : \mathcal{U} \to R$ en u (l'accent désigne la dérivation).

On définit alors la famille de problèmes auxiliaires (en abrégé P.A) :

$$(\gamma_\varepsilon^v) : \quad \min_{u \in \mathcal{U}^f} \{G_\varepsilon^v(u)\} \quad , \quad \varepsilon \in R_+^* , \ v \in \mathcal{U}$$

On a le lemme fondamental :

Lemme 1 : Si v est solution de γ_ε^v, v est aussi solution de P.P.

Preuve :

Si $u^* \in \mathcal{U}^f$ est solution de P.P., il vérifie l'inéquation variationnelle :

(2) : $\forall u \in \mathcal{U}^f$, $\langle J'(u), u-u^* \rangle + J_1(u) - J_1(u^*) \geq 0$

Si $v \in \mathcal{U}^f$ est solution de γ_ε^v, on a donc :

(3) $\forall u \in \mathcal{U}^f$: $\langle \varepsilon J'(v), (u-v) + \varepsilon J_1(u) - \varepsilon J_1(v) \geq 0$

Comme $\varepsilon > 0$, v est aussi solution de P.P.

■

Cette propriété est à rapprocher du concept d'applicabilité" dans la terminologie de Mesarovic et al. [10].

Pratiquement, ce lemme est utilisé de la façon suivante : étant donné une valeur initiale u^0, on détermine u^1 solution de $\gamma_{\varepsilon_0}^{u^0}$, u^2 solution de $\gamma_{\varepsilon_1}^{u_1}$, ...

L'algorithme de base proposé consiste donc à se donner une famille $\{K^k, \varepsilon^k, k \in \mathbb{N}\}$, K^k désignant une fonctionnelle sur $\mathcal{U}$ et ε^k un réel positif, $\forall k \in \mathbb{N}$.

1) Algorithme I

0) Choisir u^0 (élément de $\mathcal{U}^f$ de préférence); faire $k=0$.

1) Résoudre PA^k :

$$\gamma_{\varepsilon^k}^{u^k}) : \min_{u \in \mathcal{U}^f} \{K^k(u) + \langle \varepsilon^k . J'(u^k) - (K^k)'(u^k), u \rangle + \varepsilon^k J_1(u)\}$$

Soit u^{k+1} la solution de PA^k.

2) Test d'arrêt (par exemple : $\|u^k - u^{k+1}\| < \rho$)

Si test d'arrêt non satisfait : faire $k \leftarrow k+1$ et retourner en 1).

Si J et les K^k, $k \in \mathbb{N}$, sont différentiables et coercives et si leurs dérivées sont Lipchitziennes, la suite $(u^k)_{k \in \mathbb{N}}$ converge fortement vers la solution unique u^* de P.P. pour ε^k assez petit. Plus précisément, dans [17], Cohen donne la condition :

$$(4) \qquad 0 < \alpha \leq \varepsilon^k \leq \beta \leq \frac{2b}{A}$$

avec :

b : coefficient le coercivité des K^k, c'est à dire que :

$$(5) \quad \forall u,v \in \mathcal{U}^f : \langle (K^k)'(u)-(K^k)'(v), u-v \rangle \geq b\|u-v\|^2$$

A : coefficient de Lipchitz de $J' : \mathcal{U}^f \to R$;

$$(6) \quad \forall u,v \in \mathcal{U}^f, \quad \|J'(u)-J'(v)\| \leq A\|u-v\|$$

<u>Remarque</u>. Le principe du problème auxiliaire n'est qu'une généralisation d'une idée assez fréquement utilisée dans la littérature consacrée à la programmation convexe. Citons, entre autres, Céa [21], Aubin [22], Lions [23], Golstein [24]. Souvent, pour ces auteurs, le problème auxiliaire est choisi suffisamment régulier pour assurer l'unicité de sa solution (le problème P P. étant mal conditionné, par exemple) ; ici, il sera choisi pour rendre aisé la décomposition (en tenant compte de la structure de $\mathcal{U}$ et de $\mathcal{U}^f$, éventuellement de J_1).

2) <u>Variantes à l'algorithme 1</u>

Nous ne changeons pas le PA (γ^v_ε) si nous remplaçons la fonctionnelle G^v_ε par $\frac{1}{\varepsilon} G^v_\varepsilon$, ce qui revient à prendre $\varepsilon = 1$ dans

(1) à condition de remplacer K par $\frac{1}{\varepsilon}K$. La condition de convergence "ε suffisamment petit" peut aussi s'écrire "K^k suffisamment coercif". Cette dernière condition peut aussi s'obtenir en remplaçant K^k par (K^k+M^k), M^k désignant une fonctionnelle quadratique, par exemple :

$$(7) \qquad M^k(u) = \gamma\|u-u^k\|^2 \quad , \quad \gamma > 0$$

Comme $(M^k(u^k))' = 0$, l'addition de M^k ne rajoute pas de termes linéaires supplémentaires dans le critère de $\gamma_{\varepsilon^k}^{u^k}$.

Ce type de modification quadratique a été souvent utilisé pour assurer la convergence d'algorithme (citons, entre autres, Rockafellar [35]).

Une autre façon d'introduire les ε est par l'intermédiaire des techniques de relaxation. Au pas $(k+1)$ de l'algorithme I, on résoud le problème $\gamma_{\varepsilon^{k+1}}^{\hat{u}^{k+1}}$, au lieu de $\gamma_{\varepsilon^{k+1}}^{u^{k+1}}$, avec :

$$(8) \qquad \begin{aligned} \hat{u}^{k+1} &= (1-\varepsilon^k)u^{k+1} + \varepsilon^k\hat{u}^k \\ \hat{u}^o &= u^o \end{aligned}$$

Avec les mêmes conditions sur J et K^k, la condition (4) est aussi suffisante pour que la suite $(\hat{u}^k)_{k\in N}$ converge fortement.

III - 2b - Application du principe du problème auxiliaire pour la résolution d'un problème d'optimisation sous contraintes couplantes.

Nous allons utiliser le principe du problème auxiliaire pour résoudre un problème d'optimisation sous contraintes couplantes. Nous présentons un algorithme "à un niveau" dont l'application a la résolution d'un problème de commande optimale permet de retrouver une généralisation de l'algorithme de Takahara [11].

1) Algorithme "à un niveau"

Soient $\mathcal{U}$ et $\mathcal{C}$ deux espaces hilbertiens dans R. Soit $C \subset \mathcal{C}$ cone fermé, convexe, d'intérieur non vide et positif (c'est à dire que $C \cap (-C) = 0$).

Définissons alors le problème P.P. :

$$(9) \quad \min_{u \in \mathcal{U}^f} \{J(u)\} \quad \text{sous} \quad \theta(u) \in (-C)$$

avec $\mathcal{U}^f \subset \mathcal{U}$, fermé, convexe

$J : \mathcal{U} \to R$ s.c.i, convexe et différentiable.

$\theta : \mathcal{U} \to \mathcal{C}$ convexe au sens suivant :

$\forall u, w \in \mathcal{U}$, $\forall \alpha \in [0,1]$

$$\alpha(\theta(u)) + (1-\alpha)\theta(w) - \theta(\alpha u + (1-\alpha)w) \in C$$

et différentiable.

Formons le Lagrangien d'un tel problème

$$L(u, p) = J(u) + \langle p, \theta(u) \rangle$$

avec $p \in C^*$ ($C^* \subset \mathcal{C}^*$: cone conjugué de C)

Il est bien connu que (9) est équivalent à la recherche du point-selle de L sur $\mathcal{U}^f \times C^*$, sous réserve d'une condition de qualification des contraintes [25].

Avec les hypothèses faites, il suffit de résoudre les deux inéquations variationnelles :

$$(10) \quad u^* \in \mathcal{U}^f \quad \text{et} \quad \forall u \in \mathcal{U}^f \quad \langle L'_u(u^*, p^*), u - u^* \rangle \geq 0$$

$$(11) \quad p^* \in C^* \quad \text{et} \quad \forall p \in C^* \quad \langle L'_p(u^*, p^*), p - p^* \rangle \leq 0$$

$\forall k \in N$, choisissons alors un lagrangien auxiliaire

$$\phi^k : \mathcal{U} \times \mathcal{C} \to R$$

(12) $\forall u \in \mathcal{U}, \quad \forall p \in \mathcal{C} \qquad \phi^k(u,p) = K^k(u) + \langle p, \Omega^k(u)\rangle$

K^k et Ω^k désignant des applications du même type respectivement que J et θ.

Le problème auxiliaire PA^k pour un couple donné $(u^k, p^k) \in \mathcal{U}^f \times C^*$, consiste à trouver le point-selle sur $\mathcal{U}^f \times C^*$ de la fonctionnelle :

(13) $$(u,p) \to K^k(u) + \langle \varepsilon J'(u^k) - K^{k'}(u^k), u\rangle + \langle p^k, (\varepsilon.\theta'(u^k) - \Omega^{k'}(u^k)).u\rangle + \langle p, \Omega^k(u) + \rho\theta^k(u^k) - \Omega^k(u^k)\rangle$$

avec ε et $\rho > 0$.

D'après le théorème du point-selle du Lagrangian [25], la solution de PA^k est aussi celle du problème d'optimisation sous contrainte :

(14) $$\min_{u \in \mathcal{U}^f} \{K^k(u) + \langle \varepsilon J'(u^k) - K^{k'}(u^k), u\rangle + \langle p^k, (\varepsilon\theta'(u^k) - \Omega^{k'}(u^k)).u\rangle\}$$

sous la contrainte :

(15) $$(\Omega^k(u) + \rho.\theta(u^k) - \Omega^k(u^k)) \in (-C)$$

L'algorithme est le suivant :

<u>Algorithme 11</u>

0) Choisir (u^o, p^o) ; faire $k \leftarrow 0$
1) Résoudre (14)-(15) ; soit (u^{k+1}, p^{k+1}) une solution
2) Si le test d'arrêt est positif stop ; sinon faire $k \leftarrow k+1$ et retourner en 1)

Remarques

Cet algorithme est à un seul niveau de calcul pour la résolution de PA^k, d'où son nom (Cohen propose également un algorithme "à deux niveau" dans lequel PA^k est décomposé en deux problèmes PA_u^k et PA_p^k). Hormis le cas où J et K^k sont quadratiques et θ et Ω^k linéaires [18], nous ne connaissons pas de preuve de convergence de cet algorithme.

2) Algorithme de Takahara généralisé

Soit le problème de commande optimale :

$$(16) \quad \min_{(u,x)} \{\varphi(x(T)) + \int_{t_o}^{T} L(x(\alpha),u(\alpha))d\alpha$$

$$(17) \quad \theta(x(T)) = 0 \in R^r \quad , \quad T \text{ donné}$$

$$(18) \quad \dot{x}(t) = F(x(t),u(t)) \quad , \quad t \in [t_o,T] \ , \quad u(t) \in R^m$$

$$x(t_o) = \alpha \in R^n$$

Le problème de commande optimale peut être vu comme un problème de minimisation sous contraintes :

- La fonctionnelle a minimiser est définie par (16).
- Les contraintes sont définies par (17) et (18).

On fait alors les hypothèses suivantes :

- La trajectoire de l'état $x:[t_o,T] \to R^n$ est un élément de l'espace de Sobolev $H^1([t_o,T]; R^n)$
- La commande $u:[t_o,T] \to R^m$ est un élément de $L^2([t_o,T]; R^m)$

Les contraintes sont donc définies dans $\mathcal{C}=L^2([t_o,T];R^n) \times R^r$, et comme l'on identifie canoniquement un espace de Hilbert à son dual, on a $\mathcal{C}^* = \mathcal{C}$.

lemme 2

L'adjoint $p^*=(\lambda^*,\nu^*) \in \mathcal{C}^*$, donné par le principe du minimum de Pontriaguine, vérifie :

$$(19)\quad \begin{cases} \lambda^* = -\left[\frac{\partial F}{\partial x}(x^*,u^*)\right]'\lambda^* - \left[\frac{\partial L}{\partial x}(x^*,u^*)\right]' \\ \lambda^*(T) = \left[\frac{\partial \varphi}{\partial x}(x^*(T)\right]' + \left[\frac{\partial \theta}{\partial x}(x^*(T))\right]'.\nu \end{cases}$$

où (x^*,u^*) sont les trajectoires optimales de l'état et de la commande et λ^* l'état adjoint du système dynamique.

Une preuve de ce lemme est donné dans [26].

Nous allons utiliser l'algorithme 11 en identifiant la paire (x,u) avec la variable u dans le formalisme général.

Afin de définir la séquence de problèmes auxiliaires PA^k, décomposons R^n et R^m en N sous-espaces R^{n_i} et R^{m_i}, $(\sum_{i=1}^{N} n_i=n\ ,\ \sum_{i=1}^{N} m_i=m)$ et supposons que la $i^{ème}$ partie de (18) s'écrit :

$$(20)\quad \begin{cases} \dot{x}_i = F_i(x,u) = f_i(x_i,u_i,h_i(x,u)) \\ x_i(t_o) = \alpha_i \end{cases}$$

avec

$$(21)\quad h_i(x,u) \triangleq (x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_N,u_1,\ldots,u_{i-1},u_{i+1},\ldots,u_N)$$

Les h_i permettent de définir les variables d'interaction :

$$(22)\qquad v_i \triangleq h_i(x,u) \quad , \quad i=1,\ldots,N$$

L'équation (20) peut alors être considérée comme l'équation de dynamique d'un $i^{ème}$ sous-système avec les équations d'interconnexion (22).

En ce qui concerne la contrainte finale (17), nous la décomposons également entre les différents sous-problèmes, mais il est possible qu'un sous-problème (ou plusieurs) n'ait pas de contraintes finales. R^r est donc décomposé en N sous-espaces R^{r_i}, $i \in \{1,\dots,N\}$, avec $\sum_{i=1}^{N} r_i = r$.

Nous définissons alors les K^k sous forme additive :

$$(23) \qquad K^k(x,u) = \sum_{i=1}^{N} K_i^k(x_i,u_i)$$

avec :

$$(24) \qquad K_i^k(x_i,u_i) = \varphi(\mathcal{R}_i^k(x(T)) + \int_{t_o}^{T} L(\mathcal{R}_i^k x, \mathcal{R}_i^k u)dt$$

$$(25) \qquad \mathcal{R}_i^k x = (x_1^k,\dots,x_{i-1}^k,x_i,x_{i+1}^k,\dots,x_N^k) \quad , x \in R^n$$

(même notation pour $\mathcal{R}_i^k u$, $u \in R^n$)

Choisissons les contraintes Ω^k du problème auxiliaire PA^k comme suit :

$$(26) \qquad \Omega^k = (\Omega_1^k, \dots, \Omega_N^k)$$

avec :

$$(27) \quad \Omega_i^k(x_i,u_i)=0 \quad \Leftrightarrow \quad \begin{cases} \overset{\circ}{x}_i = f_i(x_i,u_i,h_i(x^k,u^k)) \in R^{n_i} \\ x_i(t_o)=\alpha_i \in R^{n_i} \\ \theta_i(\mathcal{R}_i^k x(T))=0 \in R^{r_i} \quad \text{si} \quad r_i > 0 \end{cases}$$

(pas de contraintes finales si $r_i=0$)

On dit qu'il y a "prédiction des interactions", puisque pour chaque sous-système i les interactions sont fixées à $v_i^k = h_i(x_k^k,u^k)$, au pas k de l'algorithme.

En tenant compte de (14) et (15), avec $\varepsilon = \rho = 1$, PA^k se décompose en N problèmes de commandes optimales PA_i^k, $i = \overline{1,N}$:

$$(28) \quad \min_{u_i(t) \in R^{m_i}} \Big\{ \varphi(\mathcal{R}_i^k x(T)) + \sum_{j \neq i} \nu_j^{k\prime} \cdot \frac{\partial \theta_j}{\partial \lambda_i} (x^k(T)) x_i(T)$$

$$+ \int_{t_o}^{T} L(\mathcal{R}_i^k x, \mathcal{R}_i^k u) dt$$

$$+ \sum_{j \neq i} \int_{t_o}^{T} \lambda_j^{k\prime} \Big(\frac{\partial F_j}{\partial x_i} (x^k, u^k) x_i + \frac{\partial F_j}{\partial u_i} (x^k, u^k) u_i \Big] dt \Big\}$$

sous la contrainte (27).

$$x_i^k : [t_o, T] \to R^{n_i} \ , \ u_i^k : [t_o, T] \to R^{m_i}, \quad \lambda_i^k : [t_o, T] \to \mathbb{R}^{n_i}$$

désignant respectivement les trajectoires de l'état, de la commande et de l'état adjoint du $i^{ème}$ problème auxiliaire $PA_i^{(k-1)}$ et ν_i^k $(\in R^{r_i})$ le multiplicateur associé à la contrainte finale de ce même sous-problème.

<u>Remarque 1</u>. Le choix des Ω^k et K^k peut sembler arbitraire : on aurait pu définir des problèmes auxiliaires linéaires-quadratiques, par exemple (ce que propose d'ailleurs Cohen dans l'article qui nous sert de référence ici [17]). En fait nous avons préféré définir des problèmes auxiliaires en tenant compte de notre objectif : appliquer l'algorithme à la résolution du problème du refoulement optimal dans un réseau hydraulique. C'est pourquoi nous définissons des "dynamiques locales", des contraintes locales qui ne seront pas définies arbitrairement en pratique (mais en théorie, elles peuvent l'être !).

<u>Remarque 2.</u> Dans (28),

$$[\frac{\partial F_j}{\partial x_i}(x^k,u^k)x_i + \frac{\partial F_j}{\partial u_i}(x^k,u^k)u_i]$$

peut être remplacé par $[\frac{\partial f_j}{\partial v_j}(x_j^k,u_j^k,v_j^k)h_j(\mathcal{R}_i^k x,\mathcal{R}_i^k u)]$

avec :

$$v_j^k = h_j(x^k,u^k)$$

sans changer autre chose au problème auxiliaire PA^k. Ceci correspond à un changement de la fonctionnelle auxiliaire (24).

Avec la définition de h_j donnée ci-dessus, ce changement ne présente pas d'intérêt. Il n'en n'est plus de même si l'on définit, comme ce sera le cas pour le problème du refoulement optimal dans un réseau hydraulique, la dynamique découplée $f_i : R^{n_i} \times R^{m_i} \to R^{n_i}$ et la fonction de découplage

$$h_i : R^n \times R^m \to R^{p_i}$$

d'une autre façon pourvu que l'on ait encore :

$$f_i(x_i,u_i,h_i(x,u)) = F_i(x,u) \ , \qquad \forall (x,u) \in R^{n+m}$$

<u>Rappel des étapes de l'algorithme</u>

i) <u>initialisation</u>

faire $k=0$; choisir $(u^o(t),x^o(t),\lambda^o(t),v^o(t))_{t \in [t_o,T]}$

et $\nu^o \in R^r$

ii) <u>Résolution des problèmes locaux</u>

Résoudre les problèmes auxiliaires (PA_i^k) , $i \in \{1,\dots,N\}$

soient $(u_i^{k+1}(t),x_i^{k+1}(t),\lambda_i^{k+1}(t))_{t \in (t_o,T)}$

et ν_i^{k+1} la solution de (PA_i^k).

iii) Coordination

faire : $v_i^{k+1} = h_i(x^{k+1}, u^{k+1})$, $i \in \{1,\dots,N\}$

iv) Test d'arrêt

Si test négatif, faire $k \leftarrow k+1$ et retourner en ii).

3) A propos de la convergence de l'algorithme de Takahara

L'algorithme de Takahara [10] correspond au cas particulier où le problème initial a une dynamique linéaire et un critère quadratique. (Dans l'article original [10], il n'y avait pas de contraintes finales).

Comme nous l'avons dit, pour ce cas particulier, nous disposons de conditions de convergence. Reprenons l'algorithme 11 dans le cas linéaire-quadratique.

Soient : $J(u) = \frac{1}{2} \cdot \langle u-f, E(u-f)\rangle$

$$K^k(u) = K(u) = \frac{1}{2} \langle u-g, F(u-g)\rangle$$

$$\theta(u) = G(u-c)$$

$$\Omega^k(u) = \Omega(u) = H(u-\alpha)$$

E et F désignant deux opérateurs linéaires sur $\mathcal{U}$, auto-adjoints, coercifs et de norme finie ; G et H étant des opérateurs linéaires continues de $\mathcal{U}$ dans $\mathcal{C}$, surjectifs.

Théorème 1. La séquence (u^{k+1}, p^{k+1}) engendrée par les PA^k converge fortement vers l'unique solution (u^*, p^*) de P.P. (9) si les conditions suivantes sont satisfaites :

(30) 1) $(GE^{-1}H^* + HE^{-1}G^*)$ est un opérateur coercif.

2) $\varepsilon = \rho$ sont choisis pour que

(31) $(F - \varepsilon E/2)$ soit coercif.

On trouvera dans [18] la preuve de ce théorème. Remarquons que l'hypothèse (30) est destinée à assurer que (32) est possible avec un ε suffisamment petit (cette possibilité existe pour (31) puisque F est cœrcif). Il y a donc une condition préalable pour assurer la convergence de l'algorithme par ajustement de ε : cette condition correspond à une "contrainte de compatibilité" entre G et H, ce qui signifie grossièrement que l'on ne peut découper la contrainte n'importe comment (ce qui semble assez naturel) !.

En ce qui concerne l'algorithme de Takahara, comme G et H dépendent de la dynamique du système initial et de celle du système découplé en sous-systèmes, cette condition revient à limiter les possibilités de définition des dynamiques des sous-systèmes. Nous aurons l'occasion de reparler de ce problème lorsque nous évoquerons le problème du découpage d'un réseau hydraulique en sous-réseaux interconnectés.

4) Résolution parallèle ou séquentielle des sous-problèmes d'optimisation.

Tel qu'il a été présenté, l'algorithme de Takahara généralisé consiste à résoudre séparément N sous-problèmes de commande optimale, données par (27)-(28). Une fois ces N problèmes résolus, au pas k de l'algorithme, on détermine les prédictions $v_i^{k+1}=h_i(x^{k+1},u^{k+1})$ à appliquer au sous-système i pour la résolution au pas suivant (k+1). Pour reprendre la terminologie généralement adoptée en analyse numérique pour désigner ce type d'algorithme où il y a une résolution parallèle de plusieurs problèmes, on dit qu'il s'agit de "l'algorithme de Takahara , version Jacobi".

Mais on peut aussi concevoir une résolution séquentielle des N problèmes PA_i^k. Plus précisément, une fois résolution PA_1^k, avec une prédiction v_1^k, on résoud PA_2^k avec une prédiction $v_2^k = h_2(\bar{x}^k,\bar{u}^k)$ avec $\bar{x}^k=(x_1^{k+1},x_2^k,\ldots,x_N^k)$ et $\bar{u}^k = (u_1^{k+1},u_2^k,\ldots,u_N^k)$ (en définissant de même la mise à jour des paramètres λ^k et ν^k),

et ainsi de suite, la résolution de PA_N^k étant suivie par celle de PA_1^{k+1}. Il s'agit alors de "l'algorithme de Takahara, version Gauss-Seidel".

III - 3 Utilisation de l'algorithme de Takahara généralisé pour la résolution du problème du refoulement optimal dans un réseau hydraulique.

Comme nous l'avons déjà dit, cette utilisation de l'algorithme de Takahara généralisé est heuristique. La présentation est faite dans le cas particulier d'un réseau découpé en 3 sous-réseaux avec un seul point d'interaction (fig 1). L'extension au cas général ne présente pas de difficultés.

III - 3 - 1. Décomposition en sous réseaux

a) généralités

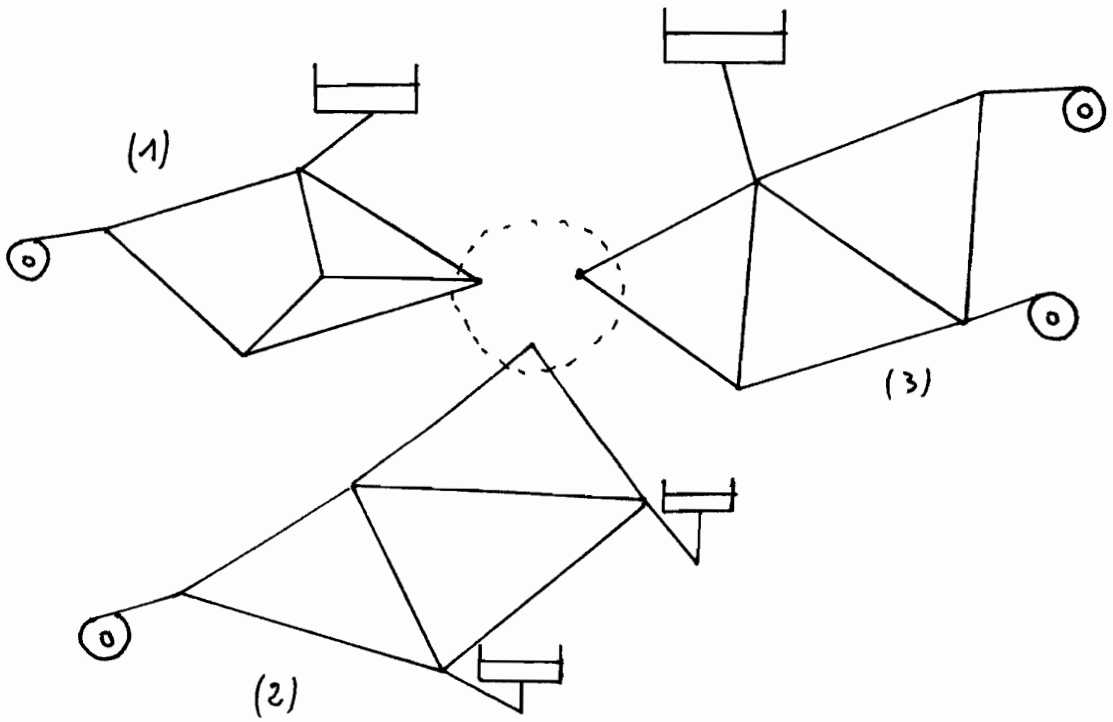


Fig. 1

Une fois fixés la consommation, sa répartition dans le réseau, le niveau d'eau dans les réservoirs et les groupes de pompe refoulant dans le réseau, la caractéristique "débit pression" de chaque sous-réseau (pris séparément), au point d'interaction, est définie de façon unique. Plus précisément on peut définir deux triplets $(P_i)_{i=\overline{1,3}}$ et $(Q_i)_{i=\overline{1,3}}$, $P_i : \mathbb{R} \to \mathbb{R}$ désignant la caractéristique de la pression du point d'interaction pour le sous-réseau i, $i \in \{1,2,3\}$, en fonction du débit et $Q_i : R \to R$ la caractéristique inverse de P_i $(Q_i = P_i^{-1})$.

Pour que la liaison des trois sous-réseau "redonne" le réseau initial, il faut :

$$P_1(q_1) = P_2(q_2) = P_3(q_3) = p \tag{33}$$

ou

$$q_i = Q_1(p) \quad , \; i \in \{1,2,3\} \tag{33 bis}$$

avec : $\begin{cases} p : \text{pression au point d'interaction} \\ q_i : \text{débit entrant } (q_i \geq 0) \text{ au sortant } (q_i < 0) \text{ du sous-réseau } i \end{cases}$

On a :

$$\sum_{i=1}^{3} q_i = 0$$

en supposant qu'il n'y a pas d'autres débits ou consommation au point d'interaction.

Les relations (33) et (33bis) peuvent aussi s'écrire, entre autres :

$$P_1(-Q_2(p)-Q_3(p)) = p \tag{34}$$

L'équation (34) correspond au schéma d'interaction suivant :

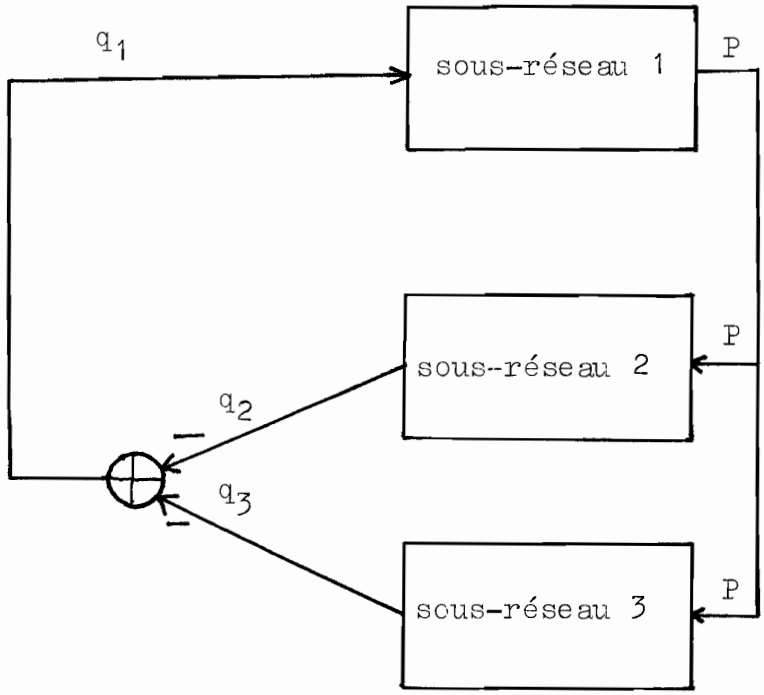


Fig. 2

b) Résolution des équations hydrauliques par décomposition-coordination

Afin de préciser davantage le rôle des variables d'interaction, présentons maintenant succintement la méthode de décomposition-coordination pour la résolution des équations hydrauliques, dans ce cas de figure.

Initialement, on se donne une pression p^o que l'on prédit au sous-réseau 1 (bien entendu, on peut faire jouer à 2 ou 3 le rôle de 1).

i) Version Jacobi

La résolution parallèle des 3 sous-réseaux donne une pression $p^1 = P_1(q^o)$ et les débits $q_2^1 = Q_2(p^o)$ et

$q_3^1 = Q_3(p^o)$, respectivement fournies par le sous-réseau 1, le sous réseau 2 et le sous réseau 3. La pression p^1 sera la prédiction appliquée à 2 et 3 et $q^1 = -(q_2^1 + q_3^1)$ le débit prédit à 1, lors de la résolution du problème auxiliaire PA^2, etc ...

ii) Version Gauss-Seidel

Supposons que les trois sous-problèmes sont résolus dans l'ordre 1,2,3.

La résolution de PA_1^o, avec la prédiction q^o, "donnera" une pression $p_2^{1/3} = P_1(q^o)$ qui sera la prédiction fournie au sous-réseau 2 lors de la résolution de PA_2^o et on en déduit $q_2^{2/3} = Q_2(p_2^{1/3})$. On résoud ensuite PA_3^o avec la prédiction $p_3^{2/3} = P_1(q^o) = p_2^{1/3}$ (on pourrait donc résoudre PA_2^o et PA_3^o en parallèle) et on en déduit le débit $q_3^1 = Q_3(p_3^{2/3})$; la prédiction $q^1 = -(q_3^1+q_2^{2/3})$ sera appliquée au sous-réseau 1 lors de la résolution de PA_1^1, etc ...

III - 3 - 2. Détermination des problèmes locaux

a) dynamique

Le sous-réseau i, $i \in \{1,2,3\}$ possède m_i usines de refoulement, dont p_i refoulant de l'eau de nappe, n_i réservoirs. On a :

$$(35) \quad \sum_{i=1}^{3} n_i = n \quad ; \quad \sum_{i=1}^{3} m_i = m \quad ; \quad \sum_{i=1}^{3} p_i = p$$

Si $n_i = p_i = 0$, le sous-problème i n'est pas dynamique : sa résolution revient à résoudre une séquence de 24 problèmes de minimisation statiques, c'est à dire non couplés dans le temps. Nous n'avons pas pu considérer de tels sous-problèmes, en pratique, aussi ne développerons nous pas davantage ce cas particulier. Dans tout ce qui suit, on suppose que tous les sous problèmes sont dynamiques.

Désignons par $\bar{\mathfrak{F}}^{l}_{i,t}(\delta_i;u_i;v_i)$ le volume d'eau entrant ou sortant du $l^{\text{ème}}$ réservoir du sous réseau i, $l \in \{1, \ldots, n_i\}$, dans l'intervalle de temps $[t,t+1]$, pour la commande u_i, si le niveau du $k^{\text{ème}}$ réservoir, à l'instant t, est δ^k_i $(k \in \{1, \ldots, n\})$ et si la prédiction est v_i.

Désignons par $\bar{\mathfrak{g}}^{j}_{i,t}(\delta_i;u_i;v_i)$ le volume d'eau refoulé par les pompes de l'usine j $(j \in \{1,\ldots,p_i\})$ le même sous-réseau, dans les mêmes conditions.

On a les équations d'état, pour le sous-réseau i :

$$(36)\begin{cases} \delta^{l}_{i,t+1} = \delta^{l}_{i,t} + \bar{\mathfrak{F}}^{l}_{i,t}(\delta_{i,t};u_{i,t};v_{i,t})/S_l \\ \varphi^{j}_{i,t+1} = \varphi^{j}_{i,t} - \bar{\varphi}^{j}_{i,t}(\delta_{i,t};u_{i,t};v_{i,t}) + D^{j}_{t}(\varphi^{j}_{i,t}) \\ l \in \eta_i \quad ; \quad j \in \rho_i \end{cases}$$

$(\xi_{i,t})_{t\in T} = (\delta_{i,t};\varphi_{i,t})_{t\in T}$ correspond à la dynamique découplée du sous-système i. η_i et ρ_i désignent respectivement l'ensemble des réservoirs et des nappes phréatiques du sous-réseau i $(\bigcup_i \eta_i = \{1, \ldots, n\}$; $\bigcup_i \rho_i = \{1, \ldots, p\})$.

L'équation (36) définit une fonction $f_{i,t}$ caractérisant la dynamique découplée du sous-système i un peu différente de la fonction f_i qui caractérisait la dynamique découplée du même sous-système dans la présentation de l'algorithme de Takahara (27). En effet, les interactions correspondaient tout simplement aux variables d'état de commande des autres systèmes, dans cette présentation ; ici, ce sont des variables d'une autre nature (en l'occurence la pression ou le débit au point d'interaction) qui "résument" l'influence des autres sous-systèmes par le sous-système i.

En conservant les conventions faites plus haut (le sous-réseau 1 prédit la pression à 2 et 3 ; les sous-réseaux 2 et 3 prédisent le débit à 1), on peut définir trois fonctions :

$$\overline{P}_{1,t} : \mathcal{M}_1 \times R^{n_1} \times \mathbb{R} \to \mathbb{R}$$

(37) $$\overline{Q}_{2,t} : \mathcal{M}_2 \times R^{n_2} \times \mathbb{R} \to \mathbb{R}$$

$$\overline{Q}_{3,t} : \mathcal{M}_3 \times R^{n_3} \times \mathbb{R} \to \mathbb{R}$$

avec : .$\mathcal{M}_i$ ensemble des groupes de pompes pouvant refouler dans le sous-réseau i $(\prod_i \mathcal{M}_i = \mathcal{U})$.

.$P_{1,t}(u_1;x_1;q))$: pression au noeud d'interaction, pour le sous-réseau 1, à l'instant t, si la commande de ce sous-réseau est u_1, l'état des réservoirs est x_1 et le débit prédit q_1.

$\overline{Q}_{i,t}(u_i;x_i;p)$: débit au noeud d'interaction du sous-réseau i , $i \in \{2,3\}$, à l'instant t, si la commande de ce sous réseau est u_i, l'état x_i et la pression prédite p (avec la convention de signe habituelle : le débit est positif s'il est entrant ; négatif s'il est sortant.

On a :

(38) $$\overline{P}_{1,t}(u_1;x_1; -[\overline{Q}_{2,t}(u_2;x_2,p)+\overline{Q}_{3,t}(u_3;x_3;p)]) = p$$

$$\forall u \in \mathcal{U} \quad , \forall x \in R^n \quad ; \forall p \in R$$

b) <u>Contraintes</u>

Les contraintes autres que la dynamique du système sont locales (cf, chap 1).

. Contraintes de pression (1-4).
. Contraintes sur le niveau d'eau dans les réservoirs et nappes (1.5,1.6).
. Contraintes finales (I.7).

c) Critère

C'est surtout pour la définition des critères locaux que nous sommes contraints de faire un certain nombre de simplifications et d'utiliser la théorie précédemment développée de manière heuristique :

Reprenons le critère du sous-problème i, tel qu'il est défini par (28).

i) comme la contrainte finale est découplée ((7), chap. I), le terme $[\sum_{j\neq i} \nu_j^{k'} \cdot \frac{\partial \theta_j}{\partial x_i} (x^k(T))x_i(T)]$ est nul

ii) comme le critère du problème initial est additif ((8), chap. II), le terme $\int_{t_o}^{T} L(\mathcal{R}_i^k x, \mathcal{R}_i^k u)dt$ correspond dans dans notre cas à

$$(39) \quad \sum_{t\in T-\{24\}} \sum_{l\in \mathcal{M}_i} A_t^l \cdot [\overline{g}_{i,t}^l(\delta_{i,t};u_{i,t};v_{i,t}^k)\overline{H}_{i,t}^l(\delta_{i,t};u_{i,t};v_{i,t}^k)] \times$$

$$[\eta_{i,t}^l(\delta_{i,t};u_{i,t};v_{i,t}^k)]^{-1} + B^l \cdot \overline{g}_{i,t}^l(\delta_{i,t};u_{i,t};v_{i,t}^k)$$

avec :

$\overline{H}_{i,t}^l(x_i,u_i,v_i)$: pression à la sortie de l'usine l, à l'instant t, sachant que la hauteur d'eau dans les réservoirs du sous-réseau i est x_i, la consigne u_i et la prédiction v_i .

$\overline{\eta}_{i,t}^l(x_i,u_i,v_i)$: rendement des pompes en cette usine l, dans les mêmes conditions.

iii) Il reste à expliciter la dernière partie de (28) qui correspond à "l'apport" des autres sous-systèmes dans le critère.

On suppose que les sous-problèmes sont résolus par la programmation dynamique. On connaît donc, au pas k de l'algorithme, le revenu $V_i^k : R^{n_i+p_i} \times T \to R$ du sous-système i

($V_i^k(\xi_i,t)$ est le coût optimal partant de la phase (ξ_i,t) pour atteindre la cible finale). V_i^k va nous permettre d'obtenir une estimation de l'état adjoint $\lambda_{i,t}^k$ du sous-système i, à l'instant t :

$$(40) \quad \phi_{i,t}^k = \frac{\partial V_i^k}{\partial \xi_i}(\xi_{i,t+1}^k, t+1) \quad \text{avec :}$$

$(\xi_{i,t}^k)_{t\in T}$: trajectoire de l'état du $i^{ème}$ sous-système au pas $(k-1)$ de l'algorithme.

Comme les variables de prédictions prennent des valeurs continues (ce sont des pressions ou des débits), on peut également évaluer :

$$(41) \quad \gamma_{j,t}^k = \frac{\partial f_{j,t}}{\partial v_j}(\xi_{j,t}^k; u_{j,t}^k; v_{j,t}^k)$$

avec :

. (f_i,t) : dynamique du sous-système i, à l'instant t, c'est à dire que l'équation :

$$(42) \quad \xi_{i,t+1} = f_{i,t}(\xi_{i,t}; u_{i,t} v_{i,t}), t \in T-\{24\}$$

est équivalente aux équations (36).

. (v_i^k,t) : prédiction du sous-système i, à l'instant t, au pas $k-1$ de l'algorithme.

. $(u_{i,}^k,t)_{t\in T\{24\}}$ Commande optimale du sous-problème i, au pas $(k-1)$ de l'algorithme.

$(\gamma_{j,t}^k)$ caractérise la sensibilité de la dynamique du sous-système j aux variations de la prédiction s'appliquant à ce sous-système.

En tenant compte de la remarque 2 du § III- 2-b, le terme $(\sum_{j\neq i}\int_{t_o}^{T} \lambda_j^{k'}[\frac{\partial f_j}{\partial v_j}(x_j^k,u_j^k,v_j^k)h_j(\mathcal{R}_i^k x,\mathcal{R}^k u)]$ devient ici, avec les mêmes conventions que précédemment :

<u>Pour le sous-réseau 1</u> :

$$(43)\quad \sum_{t\in T-\{24\}} -(\psi_{2,t}^k\cdot\gamma_{2,t}^k+\psi_{3,t}^k\cdot\gamma_{3,t}^k)\overline{P}_{1,t}(u_{1,t};\xi_{1,t};q_t^k)$$

<u>Pour le sous réseau i = 2 et 3</u>

$$(44)\quad \sum_{t\in T-\{24\}} \psi_{1,t}^k\cdot\gamma_{1,t}^k\ \overline{Q}_{i,t}(u_{i,t};\xi_{i,t};p_t^k)$$

avec :

$$(45)\quad q_t^k = -\sum_{i=2,3} \overline{Q}_{i,t}(u_{i,t}^k;\xi_{i,t}^k;p^{k-1})$$

et :

$$(46)\quad p_t^k = \overline{P}_{1,t}(u_{1,t}^k;\xi_{1,t}^k;q^{k-1})$$

dans la version Jacobi de l'algorithme, ou :

$$(46\text{ bis})\qquad p_t^k = \overline{P}_{1,t}(u_{1,t}^k\ ;\ \xi_{1,t}^k,q^k)$$

dans la version Gauss-Seidel.

Le terme $(\psi_{1,t}^k\cdot\gamma_{1,t}^k)$ peut être considéré comme le coût marginal d'un mètre cube d'eau "fourni" au sous-réseau 1.

$(\psi_{2,t}^k\cdot\gamma_{2,t}^k)$ et $(\psi_{3,t}^k\cdot\gamma_{3,t}^k)$ désignant les coûts marginaux d'une unité de pression fournie aux sous-réseaux 2 et 3 (respectivement).

La détermination pratique des coûts marginaux se fera de la façon suivante :

- on simule la commande $(u^k_{i,t})_{t\in T-\{24\}}$, avec la prédiction $(v^{k-1}_{i,t})_{t\in T-\{24\}}$.
- au pas de temps t, pour calculer $(\psi^k_{i,t}\cdot\gamma^k_{i,t}), i \in\{1,2,3\}$ on détermine la variation de l'état $(\Delta\xi^k_{i,t+1})$ à l'instant $(t+1)$ que provoque une perturbation $(\Delta v^{k-1}_{i,t})$ de la prédiction.
- Il suffit d'évaluer alors la variation $(\Delta V^k_i(\xi_{i,t+1},t+1))$ associée à la variation $(\Delta\xi^k_{i,t+1})$, $V^k_i : R \times T \to R$ désignant le revenu du problème PA^{k-1}_i.

On a :

$$(47) \quad \psi^k_{i,t} = \frac{(\Delta V^k_i(\xi^k_{i,t+1},t+1))}{(\Delta\xi^k_{i,t+1})}$$

et :

$$(48) \quad \gamma^k_{i,t} = \frac{(\Delta\xi^k_{i,t+1})}{(\Delta v^{k-1}_{i,t})}$$

En résumé, nous reprenons les diverses étapes de l'algorithme (version Jacobi) :

i) <u>initialisation</u>

Choisir les coûts associés à chaque variable d'interaction $(\psi^o_{i,t}\cdot\gamma^o_{i,t})_{t\in T-\{24\}}$ et une trajectoire pour chaque variable d'interaction $(v^o_{i,t})_{t\in T-\{24\}}$. Faire $k = 0$.

ii) <u>résolution des problèmes locaux</u>

$$(49) \quad \min_{(u_{i,t};\xi_{i,t})} \{\mathcal{J}^k_i\}$$

sous les contraintes

$$\{I4,\ I5,\ I\ 6,\ 17,\ 36\}$$

avec :

$$(50)\quad \mathcal{J}_1^k = \sum_{t\in T-\{24\}} [\sum_{l\in\mathcal{M}_1} A_t^l\ \overline{s}_{1,t}^l(\delta_{1,t};u_{1,t};q_t^k)\cdot\overline{H}_{1,t}^l(\delta_{1,t};u_{1,t};q_t^k] \times$$

$$[\eta_{1,t}^l(\delta_{1,t};u_{1,t};q^k)]^{-1} + B^l\overline{s}_{1,t}^l(\delta_{1,t};u_{\ ,t};q_t^k)$$

$$+ (-\psi_{2,t}^k\cdot\gamma_{2,t}^k - \psi_{3,t}^k\cdot\gamma_{3,t}^k)P_{1,t}(u_{1,t}(u_{1,t};\xi_{1,t};q_t^k)]$$

[pour $\mathcal{J}_2^k$ et $\mathcal{J}_3^k$ remplacer le dernier terme par

$[\psi_{1,t}^k\cdot\gamma_{1,t}^k\cdot\overline{Q}_{2,t}(u_{2,t};\xi_{2,t};p_t^k)]$ et $[\psi_{1,t}^k\cdot\gamma_{1,t}^k\cdot\overline{Q}_{3,t}(u_{3,t};\xi_{3,t};p_t^k)]$,

respectivement, les autres termes sont identiques, aux indices près].

iii) <u>Coordination</u>

. Calculer $(\psi_{i,t}^{k+1}\cdot\gamma_{i,t}^{k+1})_{t\in T-\}24\}}$

. Prédire les variables d'interaction (p_t^{k+1}) et $(q_t^{k+1})_{t\in T-\{24\}}$ (résultats de la simulation de $(u_{i,t}^{k+1})_{t\in T-\{24\}}$, $i \in\{1,2,3\}$, avec utilisation des formules (45) et (46)).

iV) <u>Test d'arrêt</u>

Par exemple $\mathrm{Max}(a^k,b^k,c^k) \leq \varepsilon$

avec :

$$a^k = \underset{i}{\mathrm{Max}} \{\mathcal{J}_i^k - \mathcal{J}_i^{k-1}\}$$

$$b^k = \underset{t}{\mathrm{Max}} \{|p_t^{k+1} - p_t^k|, |q_t^{k+1} - q_t^k|\}$$

$$c^k = \underset{\substack{t \\ i \in \{1,2,3\}}}{\mathrm{Max}} \{-|\psi_{i,t}^{k+1} \cdot \gamma_{i,t}^{k+1} - \psi_{i,t}^k \gamma_{i,t}^k|\}$$

- Si test d'arrêt négatif, faire $(k \leftarrow k+1)$ et retourner en ii).

III - 4. Remarques concernant la convergence

a) Les conditions (30), (31), (32) de l'algorithme de décomposition-coordination, dans le cas linéaire quadratique, montrent qu'il est possible de "forcer" la convergence (par modification des critères locaux ou par utilisation des techniques de relaxation) si la décomposition des contraintes est "correcte". Et nous utiliserons bien sûr ces possibilités pour la résolution de notre problème.

Toutefois, il est bien malaisé de préciser ce qu'il faut entendre par décomposition "correcte" des contraintes, ou plus exactement ici, ce qu'est une "bonne" décomposition d'un réseau en sous-réseaux.

Il y a les cas où cette notion semble intuitivement évidente : prenons, par exemple, le cas d'un réseau formé de deux parties reliées par une seule canalisation de petite dimension (fig.3).

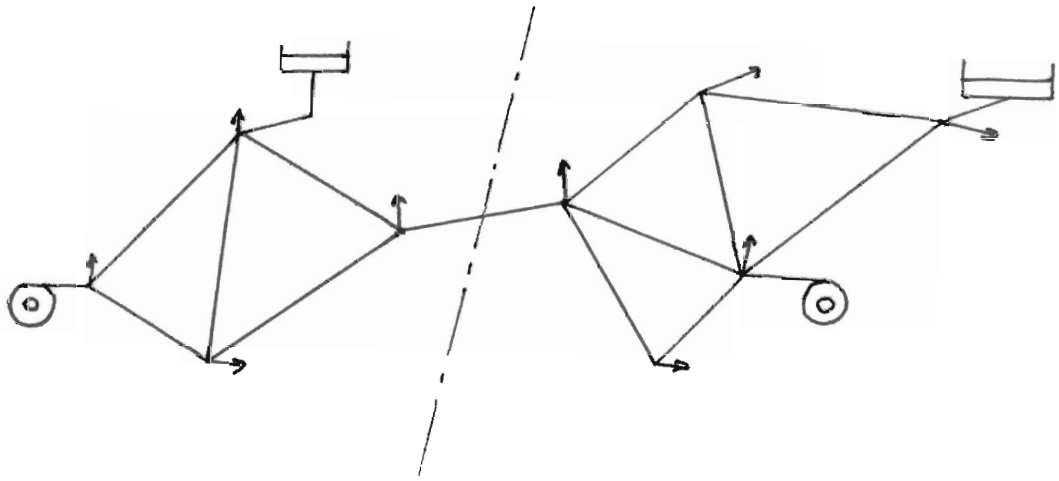


Fig. 3

Mais il n'en n'est plus de même dans le cas général. Quant à la notion "d'intensité de couplage", elle n'est nullement évidente, comme le montre l'exemple suivant :

Reprenons les équations (29)-(32) avec :

$$H = [1,0] \; ; \; G = [1,g] \; ; \; E = F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad , \quad g \in R$$

On a :

$$(GE^{-1}H^{*}+HE^{-1}G^{*}) = 2 \quad , \; \forall g \in R$$

On peut donc choisir un $\varepsilon > 0$, pour que (31) et (32) soient satisfaites :

$$\varepsilon < \mathrm{Min}\,(2, \frac{4}{3(1+g^2)})$$

Or g traduit le couplage au niveau de la contrainte (il n'y a pas de couplage par le critère ici) : même si g est grand ("couplage fort"), l'algorithme peut converger.

b) Mais il semble normal de considérer qu'un découpage en sous-réseaux est inadapté si la résolution des équations hydrauliques par décomposition coordination est impossible, pour une commande et un pas de temps donnés.

Aussi, considérons, à titre d'exemple, un réseau décomposé en deux sous-réseaux connectés en un seul point.

Soit $p = P_i(q)$ (respectivement $Q_i = P_i^{-1}$) la caractéristique débit-pression (respectivement pression-débit) du sous-réseau i , $i \in \{1,2\}$, au point d'interaction. Pour simplifier on suppose que P_i et Q_i sont linéaires.

Pour que la juxtaposition des deux sous-réseaux soit possible il faut :

$$p_1 = p_2$$
$$q_1 = - q_2$$

p_i (respectivement q_i) désignant la pression (respectivement le débit) au noeud d'interaction, coté sous-réseau $i, i \in \{1,2\}$.

Appliquons l'algorithme de décomposition-coordination (version Gauss-Seidel) pour résoudre les équations hydrauliques du réseau. On suppose que le sous-réseau 1 "prédit" la pression et le sous-réseau 2 le débit. Au pas (k+1) de l'algorithme, on :

$$q_2^{k+1} = Q_2(p_1^k)$$

$$p_1^{k+1} = P_1(-q_2^{k+1})$$

On a donc :

$$p_1^{k+1} = P_1 \cdot Q_2 \cdot p_1^k = -\left(\frac{P_1}{P_2}\right) \cdot p_1^k$$

Une condition suffisante de convergence de la suite $(p^k$ $(p_1^k)_{k \in N}$ est le caractère contractant de $\left(\frac{P_1}{P_2}\right)$ soit :

$$\left|\frac{P_1}{P_2}\right| < 1$$

Cette condition signifie que le sous-réseau qui a la caractéristique de sortie (au point d'interaction) la plus stable en pression doit prédire la pression (et l'autre le débit).

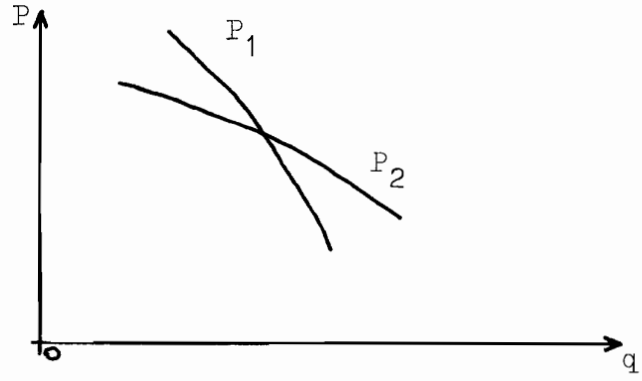


Fig.4 : exemple de mauvais choix des prédictions.

S'il existe plusieurs points d'interaction, le principe est le même : le réseau qui a la caractéristique en débit-pression la plus stable en pression (la pente de la courbe caractéristique est la plus faible) prédit la pression, au point d'interaction considéré ; les autres sous réseau connectés en ce point, prédisent le débit.

Mais la difficulté vient du fait que les caractéristiques débit-pression aux divers points d'interaction ne sont pas figés : elles se modifient au cours du temps en raison de l'évolution des niveaux dans les réservoirs et chaque fois que l'on change la configuration de pompes refoulant dans le réseau ! Ceci veut dire qu'il faudrait, pour respecter le principe que nous venons d'énoncer, modifier les points à débit (ou à pression) imposé à chaque essai d'une commande, une fois le découpage choisi, ce qui est pratiquement hors de question.

Ceci limite aussi l'intérêt des méthodes de décomposition de graphes valués, comme celles développées par Himmeblau [27] Milgram [28], Richetin [29], entre autres. En effet, ces méthodes déterminent le "meilleur" découpage d'un graphe valué, et la valuation qui nous semble la mieux adaptée à notre problème ne peut être déterminée qu'après avoir réalisé le découpage !

Pratiquement, nous choisirons le découpage le plus naturel, c'est à dire celui qui semble entraîner les couplages entre sous-réseaux les plus faibles possibles (forte résistance des canalisations reliant les réseaux, petits débits transitant d'un réseau vers l'autre), tâche qui peut éventuellement être automatiser en utilisant les méthodes que nous venons de citer. Ensuite, on déterminera pour chaque point d'interaction, les pentes des courbes caractéristiques de chaque sous-réseau liés au point d'interaction considéré et nous utiliserons le principe présenté ci-dessus en retenant quelques points de fonctionnement typiques.

CHAPITRE IV : METHODE D'AGREGATION - DESAGREGATION

IV - 1. Présentation heuristique de la méthode

Comme nous l'avons déjà dit, une des difficultés majeures du problème est liée au caractère implicite de la relation entrée-sortie, par l'intermédiaire des équations liant les débits et les fusions dans un réseau hydraulique. La description du système par un tel modèle est indispensable pour représenter les contraintes (il faut non seulement connaître la pression statique dans les canalisations, mais aussi la répartition des débits entre les différents réservoirs).

Toutefois, en ce qui concerne le problème d'optimisation, le gain obtenu par une journée de refoulement provient essentiellement de la "bonne" utilisation des stocks : une partie de l'eau refoulée aux heures où l'énergie électrique est la moins chère est stockée dans les réservoirs pour être restituée aux heures où l'électricité est la plus chère (*).

Reprenons le critère $\mathcal{J}$ ((8), chapitre I) :

$$\mathcal{J} = \sum_{t \in T\{24\}} \sum_{l=1}^{m} A_t^l \frac{s_t^l(x,u) H_t^l(x,u)}{\eta_t^l(x,u)} + B_t^l s_t^l(x,u)$$

Or :

1) A_t^l est pratiquement indépendant de l (voir note de bas de page - § I-2-C).

(*) même en l'absence de modulation du coût de l'électricité au cours de la journée, il serait intéressant de stocker l'eau provenant des usines où le traitement de l'eau est le moins cher (eau de nappe phréatique/pour l'utiliser aux heures de pointe car les contraintes sur les nappes limitent le refoulement instantané en ces usines.

2) $H_t^l(x,u)$ et $\eta_t^l(x,u)$ varient peu en fonction de l'état x et de la consigne u et sont peu différents d'une usine à l'autre.

On peut donc considérer approximativement le terme correspndant à l'énergie électrique homogène "géographiquement" : il devient, à l'instant t, proportionnel à $[\sum_{l=1}^{m} s_t^l(x,u)]$.

L'idée est alors d'utiliser une méthode itérative à deux niveaux :

- au niveau haut, on résoud un problème dynamique agrégé. La dynamique s'écrit :

$$(1) \quad \begin{cases} X_{t+1} = X_t - C_t + P_t & t \in T-\{24\} \\ X_o \text{ donné} \end{cases}$$

avec
- X_t : Volume total dans les réservoirs à l'instant t.
- C_t : Consommation totale dans l'intervalle de temps $[t,t+1]$.
- P_t : Volume total refoulé par les usines dans l'intervalle $[t,t+1]$.

- au niveau local, on cherche a distribuer le refoulement entre les différentes usines, en tenant compte des informations du niveau haut.

IV - 2. Présentation théorique dans un cas particulier

Nous présentons dans ce paragraphe l'algorithme d'agrégation-désagrégation pour la résolution d'un problème de commande optimale particulier, mais qui présente certaines similitudes avec le problème pratique que nous souhaitons résoudre.

a) hypothèses-notations

Soit le système dynamique dont l'état $(x_t)_{t\in T}$ vérifie l'équation :

$$(2) \quad \begin{cases} x_{t+1} = \mathfrak{F}_t(x_t,u_t) & \forall\ t \in T-\{N\} \\ x_o \text{ donné} \end{cases}$$

avec $\mathfrak{F}_t : R^n \times R^m \to R^n$

et : $T = \{0,1,\dots,N\}$

$(\mathfrak{F}_t)_{t\in T-\{N\}}$ est supposée posséder la propriété suivante :

(Π) Il existe un triplet $(G,H,F_t)_{t\in T-\{N\}}$ tel que :

$$(3) \quad G(\mathfrak{F}_t(x,u)) = F_t(G(x),Hu) \quad , \forall (x,u) \in R^n \times R^m$$

$$\forall t \in T-\{N\}$$

avec :

$$\begin{cases} G : R^n \to R^{n_o} \\ H : R^m \to R^{m_o} \quad , \quad H \text{ linéaire } [H \in \mathcal{L}(R^m,R^{m_o})] \\ F_t : R^n \times R^m \to R^{n_o} \\ n_o < n \\ m_o < m \end{cases}$$

G (respectivement H) désigne l'agrégation de l'état (respectivement de la commande) et $(F_t)_{\in T-\{N\}}$ définit la dynamique agrégée.

Considérons le critère $\mathcal{J}$:

$$\mathcal{J} = \sum_{t\in T-\{N\}} \mathcal{L}_t(G(x_t),u_t) + g_N(G(x_N)).$$

et le problème de commande optimale (PP_1) :

(PP_1) : $\min\limits_{(x_t,u_t)} \{\mathcal{J}\}$ sous les contraintes :

(5) $\quad (G(x_t), Hu_t) \in \mathcal{C}_t \subset R^{n_o} \times R^{m_o}$

(6) $\quad G(x_N) \in \mathcal{X}_N$ ($\mathcal{X}_N$ est la cible finale)

$(x_t)_{t\in T}$ et $(ut)_{t\in T-\{N\}}$ étant liés par l'équation d'état (2).

Le problème (PP_1) est particulier en ce sens qu'il ne dépend que de l'agrégé de l'état. Il est donc équivalent au problème (PP_2) :

(PP_2) $\min\limits_{(X_t,u_t)} \{J\}$ sous les contraintes :

(5bis) : $(X_t, Hu_t) \in \mathcal{C}_t$

(6bis) $\quad (X_N) \in \mathcal{X}_N$

$(X_t)_{t\in T}$ et $(u_t)_{t\in T-\{N\}}$ étant liés par l'équation dynamique agrégé :

(2bis) $$\begin{cases} X_{t+1} = F_t(X_t, Hu_t) & , t\in T-\{N\} \\ X_o = G(x_o) \end{cases}$$

Le critère J s'écrit :

(4bis) $$J = \sum_{t\in T-\{N\}} \mathcal{L}_t(X_t, u_t) + \mathcal{S}_N(X_N)$$

Posons :

(7) $\quad \mathcal{X}_t = \{X \mid \exists u \in R^m \ , \ (X, Hu) \in \mathcal{C}_t\}$

$U_t(X) = \{u \mid (X, Hu) \in \mathcal{C}_t\}$

On suppose $\mathcal{C}_t$ non vide, et H surjective d'où l'on déduit que $\mathcal{X}_t$ est non vide, ainsi que $U_t(X)$, $\forall X \in \mathcal{X}_t$.

(PP_2) se décompose en deux problèmes, l'un statique (Pbs), l'autre dynamique (Pbd) :

$$(Pbs) : \begin{cases} \text{Calculer } L_t(X,P) = \min\limits_{Hu=P} \{\mathcal{L}_t(X,u)\} \\ \forall (X,P) \in \mathcal{C}_t \ , \ \forall t \in T-\{N\} \end{cases}$$

(Pbd) : $\min\limits_{(X_t,P_t)} \{\overline{J}\}$ sous la contrainte (6bis) et (9) :

$$(9) : \quad (X_t,P_t) \in \mathcal{C}_t$$

$(X_t)_{t\in T-\{N\}}$ et $(P_t)_{t\in T-\{N\}}$ étant liés par l'équation d'état :

$$(10) \quad \begin{cases} X_{t+1} = F_t(X_t,P_t) \\ X_o = G(x_o) \end{cases}$$

et $\overline{J}$ étant défini par :

$$(11) \quad \overline{J} = \sum_{t\in T-\{N\}} L_t(X_t,P_t) + g_N(X_N)$$

On voit que P_t joue le rôle de la commande (agrégée) dans PPd.

b) <u>La méthode d'agrégation-désagrégation</u>

Supposons que l'on ait calculé $L_t(X,P)$ et que (Pbd) soit résolu par la programation dynamique. On connait donc $V: R^{n_o} \times T \to \overline{R}$, le revenu de ce problème. V permet de résoudre le problème initial (PP_1) par la résolution de (N-1) problèmes statiques de minimisation. En effet, si nous désignons par $(u^*_t)_{t\in T-\{N\}}$ la séquence de commande optimale de (PP_1) et par $(x^*_t)_{t\in T}$ la trajectoire d'état associée à cette séquence, nous avons :

$$(\delta)\begin{cases} u_t^* = \underset{u\in U_t(G(x_t^*))}{\text{Arg min}} \{\mathcal{L}_t(G(x_t^*),u)+V(G(\mathcal{F}_t(x_t^*,u)),t+1)\} \\ x_{t+1}^* = \mathcal{F}_t(x_t^*,u_t^*) \quad , \quad t \in T-\{N\} \\ x_o^* = x_o \end{cases}$$

Le problème (δ) correspond à la tâche de désagrégation. Il peut paraître surperflu puisque la résolution de (Pbd) donne la commande agrégée optimale $(P_t^*)_{t\in T-\{N\}}$ qui vérifie :

$$P_t^* = \underset{(X_t^*,P_t)\in \mathcal{C}_t}{\text{Arg min}} \left\{L_t(X_t^*,P_t) + V(F_t(X_t^*,P_t),t+1)\right\}$$

et la résolution de (Pbs) donne $(u_t^*)_{t\in T-\{N\}}$ c'est à dire

$$(12) \qquad u_t^* = \underset{Hu=P_t^*}{\text{Arg min}} \{\mathcal{L}_t(X_t^*,u_t^*)\}$$

On verra par la suite les raisons qui nous font définir (δ).

Au lieu de résoudre (Pbs), puis (Pbd), puis (δ), nous allons imbriquer ces trois problèmes en utilisant les idées développées au chapitre précédent, concernant le principe du problème auxiliaire. Plus précisément nous allons avoir un algorithme comprenant les étapes suivantes :

i) agrégation : résolution par la programmation dynamique d'un problème auxiliaire PA^k, correspond à un problème de commande optimale du système dynamique agrégé. Soit V^k le revenu de PA^k.

ii) désagrégation : résolution d'un problème de type (δ) avec le revenu V^k.

On calcule le critère intervenant dans (PA^{k+1}) à partir du résultat de la désagrégation, en tenant compte de la définition de L_t.

Reprenons plus en détail ces deux étapes de l'algorithme :

i) <u>agrégation</u>

Soient $A_t^k \in R^{r_o}$ et $B_t^k \in R^{m_o}$, $t \in T-\{N\}$.

Définissons le problème (PA^k) :

$$(PA^k) \quad \min_{(X_t,P_t)} \{\bar{J}^k\} \quad \text{sous (6bis), (19) et (10).}$$

avec :

$$(13) \quad \bar{J}^k = \sum_{t\in T-\{N\}} A_t^{k\prime}.X_t + B_t^{k\prime}.P_t + \alpha \|X_t - X_t^k\|^2 + \beta\|P_t - P_t^k\|^2 + g_N(X_N)$$

$\alpha, \beta > 0$

Le calcul de A_t^k et B_t^k est supposé effectué au niveau bas de l'algorithme (mais doit permettre de satisfaire les "conditions d'applicabilité" définies par le lemme 1, chapitre III).

Soit $V^k : R^{n_o} \times T \to \bar{R}$ le revenu associé à PA^k

ii) <u>désagrégation et calcul du critère de (PA^{k+1})</u>

- On résoud le problème (δ) avec le revenu V^k à la place de V.

Désignons par $(x_t^{k+1})_{t\in T}$ et $(u_t^{k+1})_{t\in T-\{N\}}$ respectivement la trajectoire de l'état et la séquence de commandes solution du problème (δ).

- Le calcul de $(A_t^{k+1}, B_t^{k+1})_{t\in T-\{N\}}$ se fait en tenant compte de (Pbs).

On suppose que la fonction $u \to \mathcal{L}_t(X,u)$ est convexe, différentiable, $\forall t \in T-\{N\}$, $\forall X \in \mathcal{X}_t$, et que la fonction $P \to L_t(X,P)$ est différentiable.

On a alors :

$$(14) \quad \frac{\partial L_t}{\partial P}(X_t^*, P_t^*))' = (HH')^{-1}H.\frac{\partial \mathcal{L}_t}{\partial u}(X_t^*, u_t^*)$$

avec :

$$(15)\qquad \begin{cases} P_t^* = Hu_t^k \\ X_t^* = G(x_t^*) \end{cases}$$

$(u_t^*, x_t^*)_{t\in T-\{N\}}$ étant solution de $(\mathcal{E})$.

En effet, avec les hypothèses faites sur $\mathcal{L}_t$, on a (cf. Luenberger [25]) (théorème de Kuhn et Tucker).

$$(16)\qquad \exists \mu_t(X_t^*, P_t^*) \in R^{m_c} \quad , \quad \frac{\partial \mathcal{L}_t}{\partial u}(X_t^*, u_t^*) + \mu_t'(X_t^*, P_t^*).H = 0$$

En tenant compte de la surjectivité de H, on a donc :

$$(17)\qquad \mu_t'(X_t^*, P_t^*) = -\frac{\partial \mathcal{L}_t}{\partial u}(X_t^*, u_t^*)H'(HH')^{-1}$$

et la différentiabilité de L_t en P entraîne ([25], page 207)

$$(18)\qquad \mu_t'(X_t^*, P_t^*) = -\frac{\partial L_t}{\partial P}(X_t^*, P_t^*)$$

Aussi nous prendrons :

$$(19)\qquad B_t^{k+1} = [H(HH')^{-1}.\frac{\partial \mathcal{L}_t}{\partial u}(G(x_t^{k+1}, u_t^{k+1})]'$$

Nous allons de plus faire l'hypothèse suivante :

Les fonctions $X \to L_t(X,P)$ et $X \to \mathcal{L}_t(X,u)$ sont continuement différentiables.

On peut alors utiliser le théorème de Danskin [33] :

$$(20)\qquad \frac{\partial L}{\partial X}(X_t^*, P_t^*) = \frac{\partial \mathcal{L}_t}{\partial X}(X_t^*, u_t^*).$$

Aussi nous prenons :

$$(21)\qquad A_t^{k+1} = \frac{\partial \mathcal{L}_t}{\partial X}(G(x_t^{k+1}) \ , \ u_t^{k+1})$$

Nous avons l'algorithme d'agrégation-désagrégation :

i) <u>initialisation</u>

Choisir la,séquence $(A_t^o, B_t^o)_{k\in N}$; faire $k = 0$.

i) <u>agrégation</u>

Résoudre (PA^k). Soit V^k le revenu de (PA^k).

iii) <u>désagrégation</u>

Résoudre la séquence de $(N-1)$ problèmes de minimisation donnés par (δ) avec le revenu V^k. A chaque pas de temps t, évaluer (A_t^{k+1}, B_t^{k+1}).

iv) <u>test d'arrêt</u> ; si le test est négatif faire $k \leftarrow k+1$ et retourner en ii).

c) <u>Commentaires</u>

Nous n'avons pas de preuve de convergence de cet algorithme, mais on peut espérer, en tenant compte des remarques du chapitre précédent, que les coefficients α et β qui pondèrent les termes quadratiques dans le critère du problème dynamique agrégé, joueront un rôle dans l'obtention de la convergence.

Un premier point que nous voulons soulever concerne les problèmes de minimisation au niveau bas.

A notre connaissance, la méthode d'agrégation généralement proposées ne se préoccupent pas de désagrégation. Le plus souvent (Acki [14], Bertrand et al [15], Lamda et Rao [30]), il n'y a pas d'agrégation du vecteur de commande : on se contente de déterminer la commande optimale sur un système réduit en définissant un critère agrégé, approchant le critère initial (ce critère agrégé est défini arbitrairement le plus souvent). L'objectif est plutôt la recherche d'un critère minimisant "l'écart" entre la solution optimale du problème initial et celle du problème agrégé. Mais, et celà peut paraître une supériorité sur notre méthode, il n'est pas nécessaire de supposer que la dynamique, le critère et les

contraintes ne dépendent que de l'agrégé de l'état.

En fait, de nombreux problèmes (économie, énergie, hydraulique) peuvent présenter la propriété (Π). Et on peut très bien utiliser notre méthode avec un critère $\mathcal{L}_t$ dépendant de l'état initial (et non de l'état agrégé seulement) : les calculs donneront une commande sous optimale mais elle sera admissible puisque la désagrégation se fait en tenant compte des contraintes réelles.

On peut aussi faire le lien avec les méthodes où l'on résoud séquentiellement l'équation d'Hamilton-Jacobi-Bellman, la commande u étant donnée, pour en déduire le revenu V ; puis on détermine u, à V donné (méthode envisagée notamment par Howard pour les processus de Markov et par Kelly [32]), [31].)

IV-3 Application au problème du refoulement optimal dans un réseau hydraulique.

Afin d'utiliser l'algorithme d'agrégation-désagrégation, nous faisons un certain nombre d'heuristiques et de simplifications.

La dynamique du problème possède la propriété (Π). La fonction d'agrégation de l'état est l'application linéaire :

$$G \quad R^{n+p} \to R^2$$

$$G_1(x,y) = \sum_{i=1}^{n} S_i x_i \; ; \; x = (x_1, \cdots, x_n)$$

$$G_2(x,y) = \sum_{j=1}^{p} y_j \; ; \; y = (y_1, \cdots, yp)$$

$$\begin{cases} x_i : \text{hauteur d'eau dans le } i^{\text{ème}} \text{ réservoir.} \\ y_j : \text{volume d'eau dans la } j^{\text{ème}} \text{ nappe.} \\ S_i : \text{section di } i^{\text{ème}} \text{ réservoir} \end{cases}$$

La nature des commandes du système ne permet pas de définir de fonction d'agrégation de la commande. Toutefois, cela devient possible si l'on considère les volumes d'eau refoulés durant une période de temps (1 heure), en chaque usine, comme une composante du vecteur de commande.

On a alors une fonction d'agrégation :

$$H : R^m \to R^2$$

$$H=(H_1,H_2) \; ; \quad H_1(q) = \sum_{i=1}^{m} q_i(1-\alpha_i) \; ; \; q = (q_1, \cdots, q_m)$$

$$H_2(q) = \sum_{j=1}^{m} \alpha_j q_j$$

avec :

$$\begin{cases} q_i : \text{volume refoulé à l'usine} \\ \alpha_j = 1 \quad \text{si } j \text{ est une usine refoulant de l'eau de nappe.} \\ \alpha_j = 0 \text{ sinon.} \end{cases}$$

La dynamique du système agrégé s'écrit :

$$(22) \qquad \begin{cases} X_{t+1} = X_t - C_t + P_t \quad ; \; t \in \{0, \cdots, N\} \\ Y_{t+1} = Y_t - a_t P_t + D_t \end{cases}$$

avec :

$$\begin{cases} X_t : \text{volume total dans les réservoirs à l'instant } t. \\ C_t : \text{consommation totale dans l'intervalle de temps } [t,t+1[\\ P_t : \text{volume total d'eau refoulé par les usines dans l'intervalle } [t,t+1[\\ a_t P_t : \text{volume d'eau fournie par l'ensemble des nappes dans l'intervalle } [t,t+1[. \end{cases}$$

D_t : apport total d'eau dans les nappes, dans l'intervalle $[t,t+1[$.

Y_t : volume total d'eau dans les nappes, à l'instant t.

A l'instant t, la commande du système est le volume d'eau fournie par les pompes P_t et le coefficient a_t (qui donne la proposition d'eau de nappe par rapport au volume total refoulé).

Considérer que le critère est fonction de la commande et de l'agrégé de l'état n'est pas une approximation grossière : on peut même écrire qu'il ne dépend que de la commande puisque la variation sur le coût de refoulement est de moins 1% lorsque les niveaux des réservoirs passent du plus bas au plus haut.

Les contraintes, elles, ne sont pas fonction de l'agrégé de l'état et de la commande : il y a des contraintes sur le niveau d'eau dans chaque réservoir et dans les nappes, d'une part ; il y a des contraintes sur les pressions (en particulier en chaque usine), d'autre part. Afin d'utiliser l'algorithme d'agrégation-désagrégation, nous allons "relaxer" les contraintes.

1 - au niveau haut la seule contrainte qui sera prise en compte portera sur le volume des stocks :

$$(23)\quad \begin{array}{lll} X_{min} \leq X_t \leq X_{max} & , & t \in T-\{24\} \\ Y_{min} \leq Y_t \leq Y_{max} & , & t \in T-\{24\} \\ X_{24} \in [\overline{X}_1, \overline{X}_2] & ; & Y_{24} \in [\overline{Y}_1, \overline{Y}_2] \end{array}$$

X_{min} : volume d'eau minimale dans les réservoirs.

X_{max} : capacité totale des réservoirs.

Y_{min} : volume minimal d'eau dans les nappes.

Y_{max} : volume maximal d'eau dans les nappes .

$[\overline{X}_1,\overline{X}_2] \times [\overline{Y}_1,\overline{Y}_2]$ modélise la cible finale (cf.I.2.b).

Au niveau bas, on reprend un modèle en débit-pression, avec les commandes réelles (c'est à dire en tout ou rien"). La séquence de 24 problèmes statiques prendra en compte toutes les contraintes :

- les contraintes de pression.
- les contraintes sur les niveaux d'eau dans les réservoirs.
- les contraintes sur les nappes.

Comme le coût du refoulement ne dépend pratiquement pas du volume des stocks, le critère agrégé est de la forme :

$$(24) \quad J = \sum_{t=0}^{23} [\mu_t^k(1-a_t)P_t + \nu_t^k a_t P_t + \alpha_1 (X_t - X_t^k)^2 + \alpha_2 (Y_t - Y_t^k)^2]$$

μ_t^k : coût marginal d'un m3 d'eau surface, à l'instant t, moyenné géographiquement.

ν_t^k : coût marginal d'un m3 d'eau de nappe phréatique, à l'instant t, moyenné géographiquement.

$$X_t^k = \sum_{i=1}^{n} x_{t,i}^k . S_i \quad ; \quad Y_t^k = \sum_{j=1}^{n} Y_{t,j}^k$$

$(x_{t,i}^k)_{t\in T}$: trajectoire du niveau d'eau dans le ième réservoir, obtenue par résolution des 24 problèmes statiques du niveau bas, au pas (k-1) de l'algorithme.

$(Y_{t,j}^k)_{t\in T}$: trajectoire du volume d'eau dans $j^{ème}$ nappe phréatique, au pas (k-1) de l'algorithme.

<u>Remarque</u> : En toute rigueur, μ_t^k et ν_t^k ne peuvent être définis car la commande est discrète. Nous allons à nouveau considérer que la commande du système initial est le débit

refoulé par les usines (seulement pour ce calcul et non pour la séquence de minimisations statiques).

Désignons par $\mathcal{C}_t(x,q)$ le coût du refoulement dans l'intervalle de temps $[t,t+1[$, si le niveau dans le réservoir i est x_i, $(x=(x_1, \ldots, x_n)$ et le volume d'eau refoulé par l'usine j est q_j $(q=(q_1, \ldots, q_m))$.

On a (cf. équation (19)) :

$$\mu_t^{k+1} = \frac{1}{m-p} \sum_{i=1}^{m} (1-\alpha_i) \cdot \frac{\partial \mathcal{C}_t(x^{k+1}, q^{k+1})}{\partial q_i}$$

$$\nu_t^{k+1} = \frac{1}{p} \sum_{i=1}^{p} \alpha_i \cdot \frac{\partial \mathcal{C}_t}{\partial q_i}$$

On donne au chapitre suivant, lors de la présentation des résultats, un moyen pratique d'évaluation de

$$[\frac{\partial \mathcal{C}_t}{\partial q} (x^k, q^k)].$$

IV-4. Conclusions

En guise de conclusion pour ce chapitre, nous présentons les avantages et les inconvénients de la méthode d'agrégation-désagrégation.

L'un des avantages de la méthode est bien sûr la réduction du volume des calculs (cf. le paragraphe 4 du chapitre I).

Afin de réduire encore davantage la taille des calculs, on peut également envisager d'utiliser la méthode de décomposition-coordination par la résolution des problèmes de minimisation statiques lors de la désagrégation. Nous ne l'avons pas fait car la coordination des problèmes s'est avérée difficile voire impossible.

Un autre avantage de la méthode est le caractère admissible de la séquence de commandes élaborée au niveau bas (désagrégation), puisque celle-ci prend en compte toutes les contraintes s'appliquant sur le système.

Il est de plus possible de prévoir une utilisation en ligne d'un tel algorithme : en début de chaque heure, on détermine la commande à appliquer durant l'heure en tenant compte du niveau réel dans les réservoirs.

En début de chaque journée (ou plus souvent éventuellement), on résoud le problème dynamique agrégé pour déterminer le revenu qui sera intégré aux problèmes du niveau bas. Nous développons ces possibilités d'optimisation au chapitre suivant.

Un de ces inconvénients de notre méthode est surtout son caractère de sous-optimalité, encore que pour le problème, cela ne soit pas trop important : la consommation est assez mal connue et les gains à espérer sur la bonne utilisation des stocks restent quand mêmes faibles (5% environ en raison des volumes assez limités de ces derniers).

Enfin, nous voudrions revenir sur le problème de l'utilisation du principe du minimum de Pontryagine pour la résolution de notre problème. Nous avons déjà eu l'occasion de développer les raisons qui rendent difficiles l'utilisation de ce principe. En fait, si rigoureusement il en est ainsi, il est toujours possible de l'utiliser en faisant des hypothèses simplificatrices.

En effet, dans notre méthode d'agrégation-désagrégation, l'information transmise par le niveau haut au niveau bas est une "information duale" : le revenu du problème dynamique agrégé. On aurait pu remplacer le revenu $V(X,t)$ de la phase (X,t) par $\frac{\partial V}{\partial X}(X_t^*,t).X$, $(X_t^*)_{t\in T}$ désignant la trajectoire optimale du problème agrégé. Or $\frac{\partial V}{\partial X}(X_t^*,t)$ désigne le vecteur adjoint du système dynamique agrégé (en l'absence de contraintes). La question que l'on se pose alors est la suivante : pourquoi ne pas déterminer un adjoint agrégé et non l'adjoint réel ? En effet si celui-ci peut être évalué on peut utiliser un

algorithme tel l'algorithme du "Min-H" [31], avec les mêmes notations que précédemment :

i) initialisation ; poser $k = 0$, choisir une séquence de commandes

$$(u_t^o)_{t\in T-\{N\}}$$

ii) déterminer la trajectoire de l'état $(x_t^k)_{t\in}$ associée à la commande $(u_t^k)_{t\in T-\{N\}}$:

$$\begin{cases} x_{t+1}^k = \mathcal{F}_t(x_t^k, u_t^k) \\ x_o^k = x_o \end{cases}$$

iii) déterminer l'adjoint $(\lambda_t^k)_{t\in T}$, connaissant $(x_t^k)_{t\in T}$ et $(u_t^k)_{t\in T-\{N\}}$ par résolution de l'équation adjointe :

$$\begin{cases} \lambda_t^k = + \dfrac{\partial \mathcal{F}_t}{\partial x}(x_t^k, u_t^k)\lambda_{t+1}^k + \dfrac{\partial \mathcal{L}_t}{\partial x}(x_t^k, u_t^k) \\ \lambda_T = \dfrac{\partial g(x_n^k)}{\partial x} \end{cases}$$

iv) Résoudre la séquence de problèmes de minimisation statiques $((Pbs)_t^k)_{t\in T-\{N\}}$:

$$(Pbs)_t^k : \quad \min_u \{\mathcal{L}_t(x_t^k, u) + \lambda_{t+1}^{k'} \mathcal{F}_t(x_t^k, u)\}$$

et appelons (u_t^{k+1}) la solution de $(Pbs)_t^k$

v) Test d'arrêt ; si test non satisfait retourner en ii).

Sous des hypothèses de régularité suffisante, la convergence de cet algorithme a été montrée ainsi que la décroissance du critère au cours des itérations.

En ce qui concerne notre problème, on peut envisager une évaluation de $\frac{\partial \mathcal{F}_t}{\partial x}$ et $\frac{\partial \mathcal{L}_t}{\partial x}$. Une fois résolues les équations hydrauliques pour la commande u_t^k, pour les niveaux x_t^k dans les réservoirs, après avoir légèrement modifié le niveau d'un réservoir, on résoud à nouveau les équations hydrauliques : on en déduit la variation correspondante dans les autres réservoirs et la variation du critère ; on recommence l'opération pour chacun des réservoirs.

On voit que l'évaluation est laborieuse : elle risque de plus de donner des résultats peu précis. Notre méthode d'agrégation-désagrégation nous semble donc mieux adaptée pour résoudre le problème.

Mais la difficulté majeure vient bien sûr de la discontinuité de la dynamique : comme il semble hors de question de déterminer les sauts de l'adjoint lors que les contraintes d'état sont saturées l'utilisation de cet algorithme devient difficile, voire impossible.

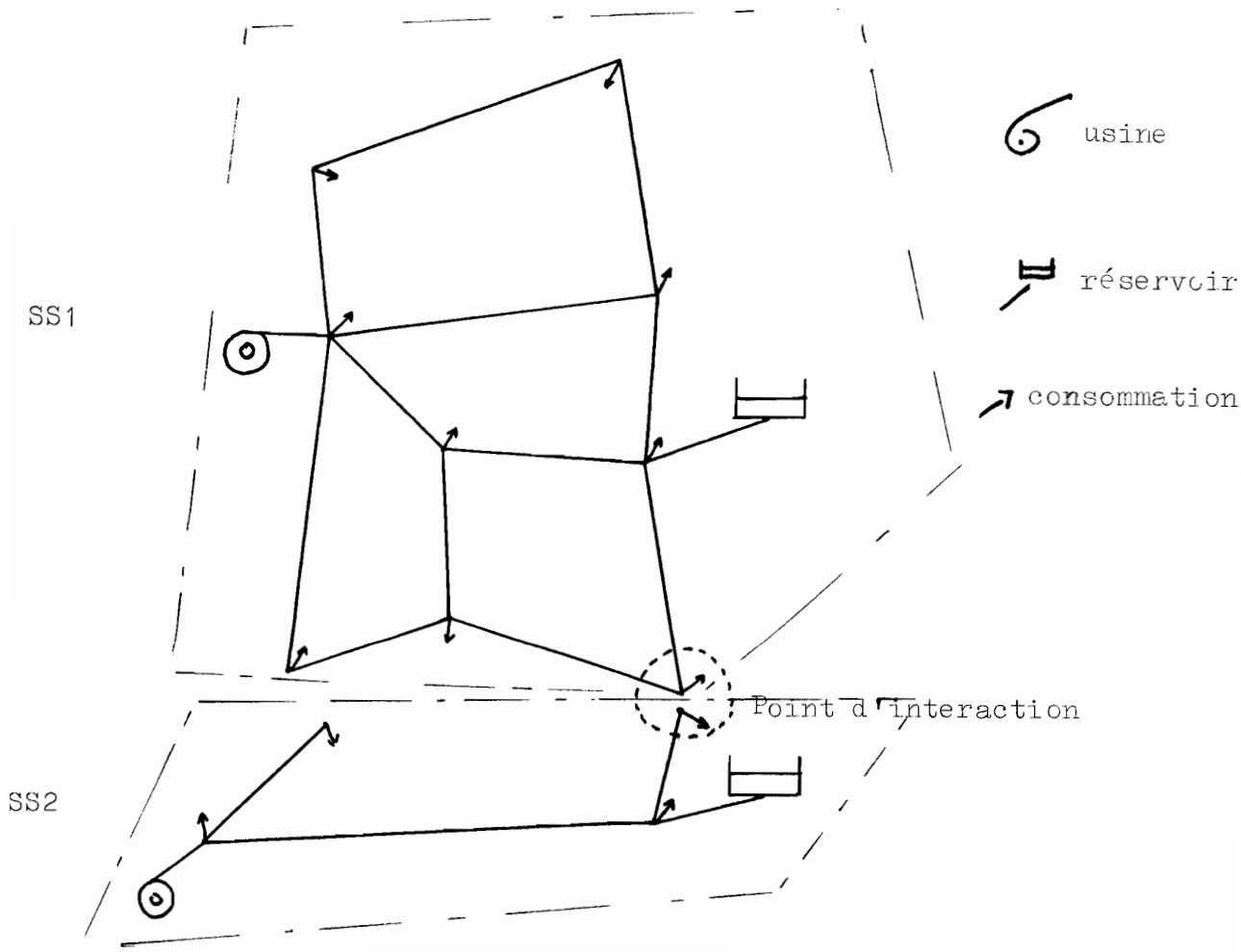
CHAPITRE V - RESULTATS OBTENUS ; CONCLUSIONS

V - 1 - Méthode de décomposition - coordination

V-1.1 - Premier cas.

Pour tester la méthode, nous avons d'abord considéré un petit réseau comportant deux usines et deux réservoirs.

Nous l'avons découpé en deux sous-réseaux reliés par un seul point d'interaction (fig.1).



Nous avons testé la convergence pour différentes caractéristiques de la canalisation liant les deux sous-réseaux.

Chaque sous-réseau possède un réservoir,(*) une usine (sans nappe phréatique).

Nous avons utilisé la technique de relaxation pour améliorer la convergence : au pas de k de l'algorithme, on prend comme prédiction une combinaison convexe des prédictions utilisées au pas $(k-1)$ et de celles déterminées par simulation au pas k ; même chose pour les prix marginaux associés aux variables d'interaction "libres".

a) lissage

Nous avons lissé les trajectoires des différentes variables d'interaction. Plus précisément, si nous désignons par $(q_t)_{t\in T-\{24\}}$ la trajectoire d'une de ces variables, nous remplaçons q_t par

$$\bar{q}_t = \left(\frac{q_{t-1}+2\cdot q_t+q_{t+1}}{4}\right) \quad , \forall t = \overline{1,23} \quad ; \quad \bar{q}_{\bullet} \text{ et } \bar{q}_{24}$$

sont ensuite calculés par extrapolation de $(\bar{q}_t)_{t=\overline{1,23}}$.

Cette technique de lissage a également été utilisée pour la trajectoire $(\xi^{k+1}_{i,t})_{t\in T}$ de l'état du $i^{ème}$ sous système (au pas k de l'algorithme), ainsi que pour les coûts marginaux associés au débit et à la pression du point d'interaction.

Le lissage permet d'atténuer les effets dus à la nature en "tout ou rien" des commandes (on cherche en fait à "convexifier" l'ensemble de commandes, par cette technique) .

(*) nous n'avons pu obtenir la convergence de l'algorithme si l'un des sous-réseaux ne possède pas de réservoir.

b) <u>Détermination des prix marginaux de la pression et du débit au noeud d'interaction</u>.

Les mesures effectuées sur le réseau ont montré que la caractéristique débit-pression au noeud d'interaction pour le sous-réseau SS1 était plus "stable" en pression que celle du sous-réseau 2, à chaque pas de temps.

Aussi, c'est ce sous-réseau qui impose la pression, et l'autre le débit, au noeud d'interaction (cf. chapitre III).

Plaçons-nous au pas k de l'algorithme. Désignons par

$(M_t^k)_{t\in T-\{24\}}$ la séquence de coûts marginaux associée à la pression donnée par SS1, au noeud d'interaction, dans le critère J_1^k ; par $(L_t^k)_{t\in T\{24\}}$ la séquence associée au débit sortant de SS2, dans le critère J_2^k.

Soient $(u_{1,t}^{k+1})_{t\in T\{24\}}$ la séquence de consignes optimales au pas k, pour SS1, et $(\bar{\xi}_{1,t}^{k+1})_{t\in T}$ la trajectoire (lissée) de l'état qui lui est associée.

. La détermination de L_t^{k+1}, $t = 1,\overline{23}$, se fait en partie lors de la simulation pour déterminer $(\xi_{1,t}^{k+1})_{t\in T}$.

Une fois résolues les équations hydrauliques pour la commande u_t^{k+1} et la prédiction (q_t^k), on les résoud à nouveau pour une prédiction $(q_t^k + \Delta q)$, Δq étant "petit".

On en déduit la variation $(\Delta\xi_{1,t+1}^{k+1})$ du niveau dans le réservoir 1, au pas de temps $(t+1)$.

Ensuite, on prend :

$$L_t^{k+1} = \frac{\Delta\xi_{1,t+1}^{k+1}}{\Delta q} \cdot \frac{\Delta V_1^{k+1}\ (\xi_{1,t+1}^{k+1},\ t+1)}{\Delta x}$$

$V_1^{k+1}(x;t)$ désignant le revenu à la phase (x,t) (au pas k de l'algorithme).

Il ne reste plus qu'à lisser la séquence $(L_t^{k+1})_{t\in T-\{24\}}$.

[Il en est de même pour la détermination de $(M_t^{k+1})_{t\in T-\{24\}}$.]

c) Résultats

La convergence a été obtenue en quatre itérations environ (c'est à dire qu'à la 5ème itération, les trajectoires lissées des prédictions et des variables duales étaient confondues avec celles de la 4ème itération).

Nous présentons ces résultats fig.2.

V.1-2- Deuxième cas

a) Le deuxième réseau considéré est toujours décomposé en deux sous-réseaux, possédant chacun une usine et un réservoir, avec deux points d'interaction.

Nous n'avons pu obtenir de convergence de notre algorithme de décomposition-coordination pour un tel réseau. Pourtant nous pensons avoir initialisé correctement l'algorithme : nous avons simulé le réseau global avec une séquence de consignes permettant de vider les réservoirs à la fin de la période où l'électricité est chère et les remplissant en fin de période de tarification basse ; nous avons relevé la trajectoire des différentes variables d'interaction pour cette séquence de consignes.

Une fois lissées, ces différentes trajectoires ont été prises comme séquence de prédictions initiales $(v_t^o)_{t\in T-\{24\}}$.

Précisons également le choix de la séquence initiale de prix marginaux associés aux pressions et aux débits aux noeuds d'interaction.

Considérons pour cela le sous-réseau 1 (la procédure est similaire pour le sous réseau 2).

Avec la prédiction $(v_{1,t}^o)_{t\in T-\{24\}}$, on simule la séquence de commande $(u_{1,t}^o)_{t\in T-\{24\}}$. A chaque pas de temps t, on perturbe légèrement la prédiction (v_1^o,t) : on en déduit la variation sur le coût de la commande $(u^o{}_{1,t})$. Le rapport de cette variation à la variation de chaque prédiction donnera le coût marginal associé à la pression ou au débit (selon la variable prédite), au point d'interaction considéré.

évolution, au cours des itérations, de la trajectoire du débit au noeud d'interaction.

débit ($\times 10^{-2} m^3/s$)

3.0
2.0
1.0

6 10 18 24 2 4 5 6 temps (heure)

évolution, au cours des itérations, de la trajectoire du prix marginal de l'eau au point d'interaction.

prix

300.
200.
100.

6 8 10 14 20 22 2 4 5 temps (heure)

o 11ère itération
x 2ème itération
Δ 3 ème itération
□ 3 ème itération
+ 5ème itération

méthode de décomposition coordination appliquée a deux sous-réseaux n'ayant qu'un point d'interaction

Fig. 2

b) Nous avons ensuite testé la coordination pour une séquence de problèmes de résolution des équations hydrauliques : chaque sous-réseau ne possède à un instant donné qu'une seule commande possible.

Si au premier pas de temps, la convergence est possible, elle devient déjà plus délicate au deuxième pas, et il y a divergence au troisième pas (fig 3).

La raison de cette dégradation vient de l'accroissement au cours du temps de "l'écart" entre les valeurs prises par les pressions et les débits aux points d'interaction, lors de l'initialisation de l'algorithme, et les valeurs solutions, au pas de temps considéré.

Nous avons donc abandonné cette méthode de décomposition-coordination pour la résolution du problème de la détermination des consignes optimales de refoulement pour le réseau de la SLEE (région parisienne sud), qui servait de support à l'étude.

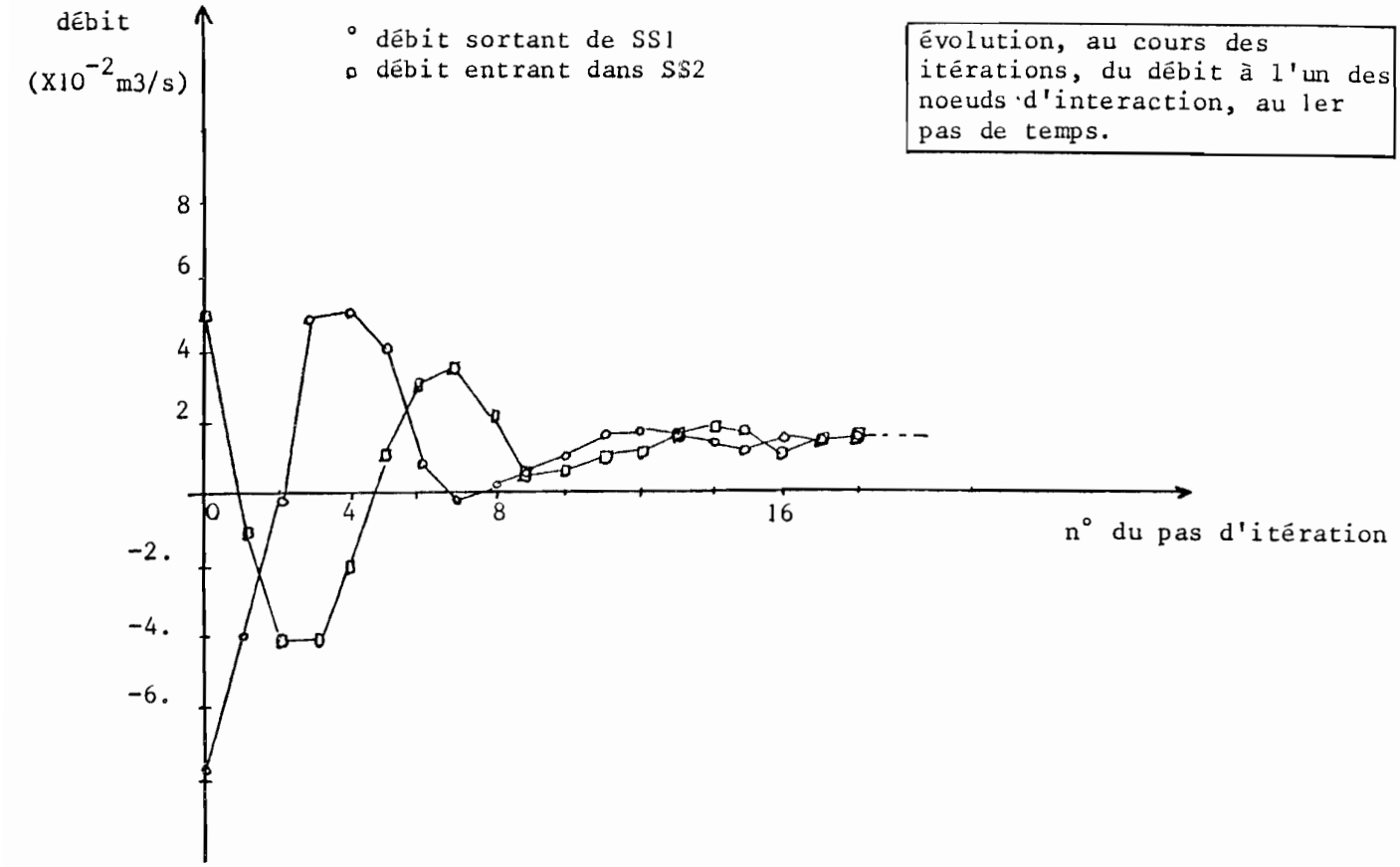
V - 2 - Méthode d'agrégation-désagrégation

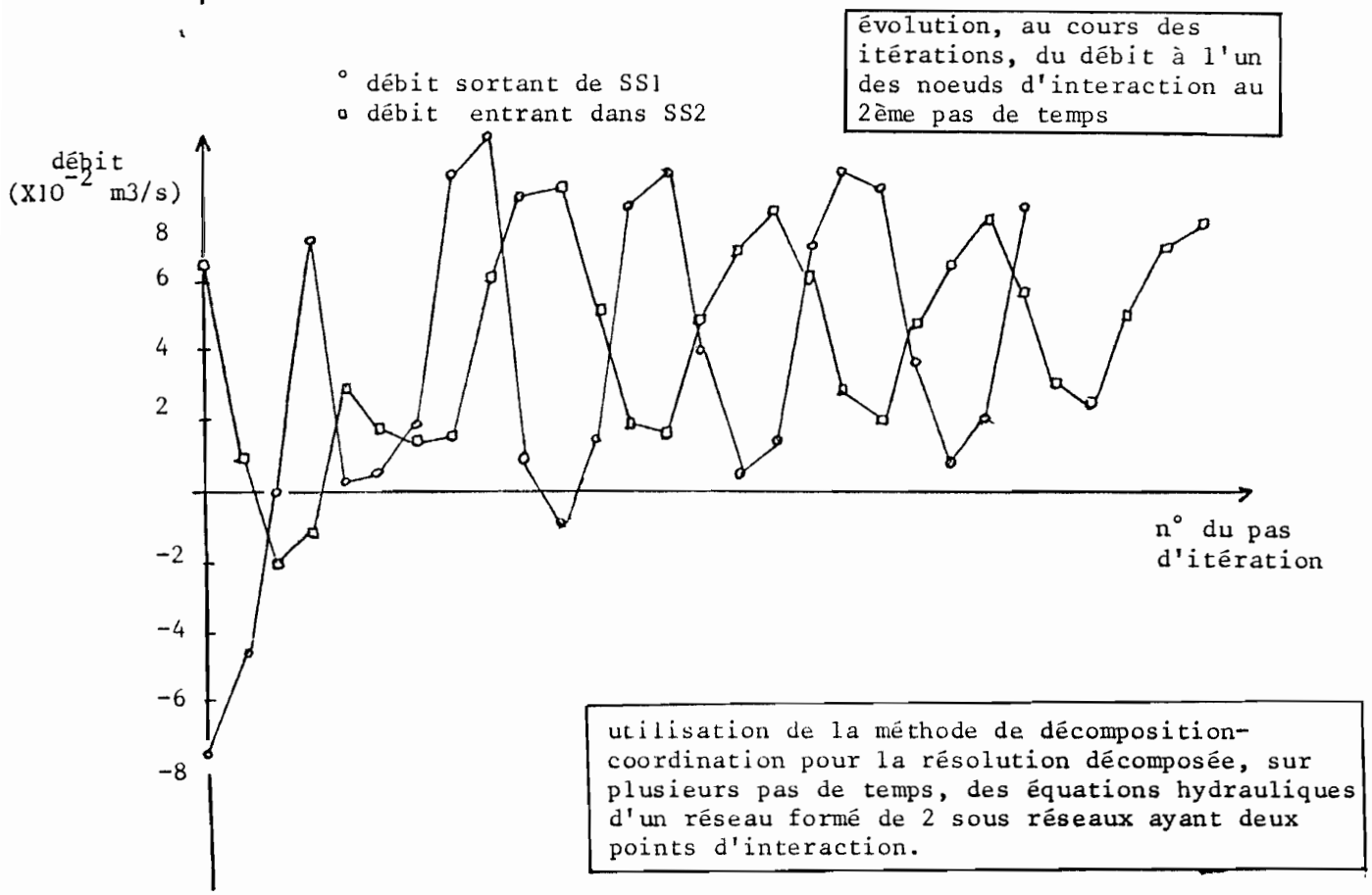
V-2-1 - Premier cas

Afin de mesurer le degré de sous-optimalité de la méthode, nous avons considéré un réseau ne comportant qu'un seul réservoir et deux usines (pas de nappe phréatique). Il est alors possible de comparer la séquence de consignes obtenue avec cette méthode avec la séquence optimale, déterminée par la programmation dynamique, sur le modèle "fin" en débit-pression.

De plus, pour que l'utilisation des stocks permette un gain appréciable sur le coût journalier de pompage, nous avons considéré un critère ne comportant que le coût de l'énergie consommée par les pompes (pas de coût de traitement de l'eau).

La séquence proposée par la méthode d'agrégation-désagrégation a un coût de 711 Frs. et la séquence optimale 696 Frs, soit une différence de 2% environ, ce qui est peu.





utilisation de la méthode de décomposition-coordination pour la résolution décomposée, sur plusieurs pas de temps, des équations hydrauliques d'un réseau formé de 2 sous réseaux ayant deux points d'interaction.

Fig. 3

V - 2.2 - Deuxième cas

a) Nous avons ensuite considéré un modèle simplifié du réseau rive-droite de la Seine de la SLEE-RPS. Le réseau comporte 23 noeuds et 31 branches, 5 usines et deux réservoirs (fig 4).

Comme nous n'avons pas de modèle des nappes phréatiques, nous n'avons pas pu tenir compte de celles-ci dans notre étude.

Les deux réservoirs sont de taille très différente : l'un a une capacité de stockage de 16500 m3 et l'autre de 2000 m3 seulement (y compris les réserves de sécurité).

Nous avons pris une consommation journalière moyenne (92 000 m3) avec une modulation dans le temps correspondant à une journée standard de milieu de semaine d'un mois d'été.

b) Rappelons qu'au pas (k+1) de l'algorithme d'agrégation-désagrégation, le critère du problème de commande optimale, pour le modèle dynamique agrégé, est de la forme :

$$(\Pi) \quad \sum_{t=0}^{23} L_t^k . P_t + \alpha (X_t - X_t^k)^2 + \alpha (X_{24} - X_{24}^k)^2$$

avec :

L_t^k : coût marginal, moyenné géographiquement, de l'eau refoulée par les pompes .

P_t : volume d'eau fourni par les pompes dans l'intervalle de temps $[t, t+1]$

α : coefficient de pondération $(\alpha > 0)$.

X_t^k : volume d'eau dans les stocks, à l'instant t, lors de l'itération k.

Nous avons initialiser l'algorithme avec une séquence de coûts $(L_t^o)_{t \in \{0,\ldots,23\}}$ nulle. et une trajectoire de l'état agrégé $(X_t^o)_{t \in \{0,\ldots,24\}}$ supposée éloignée de l'optimum :

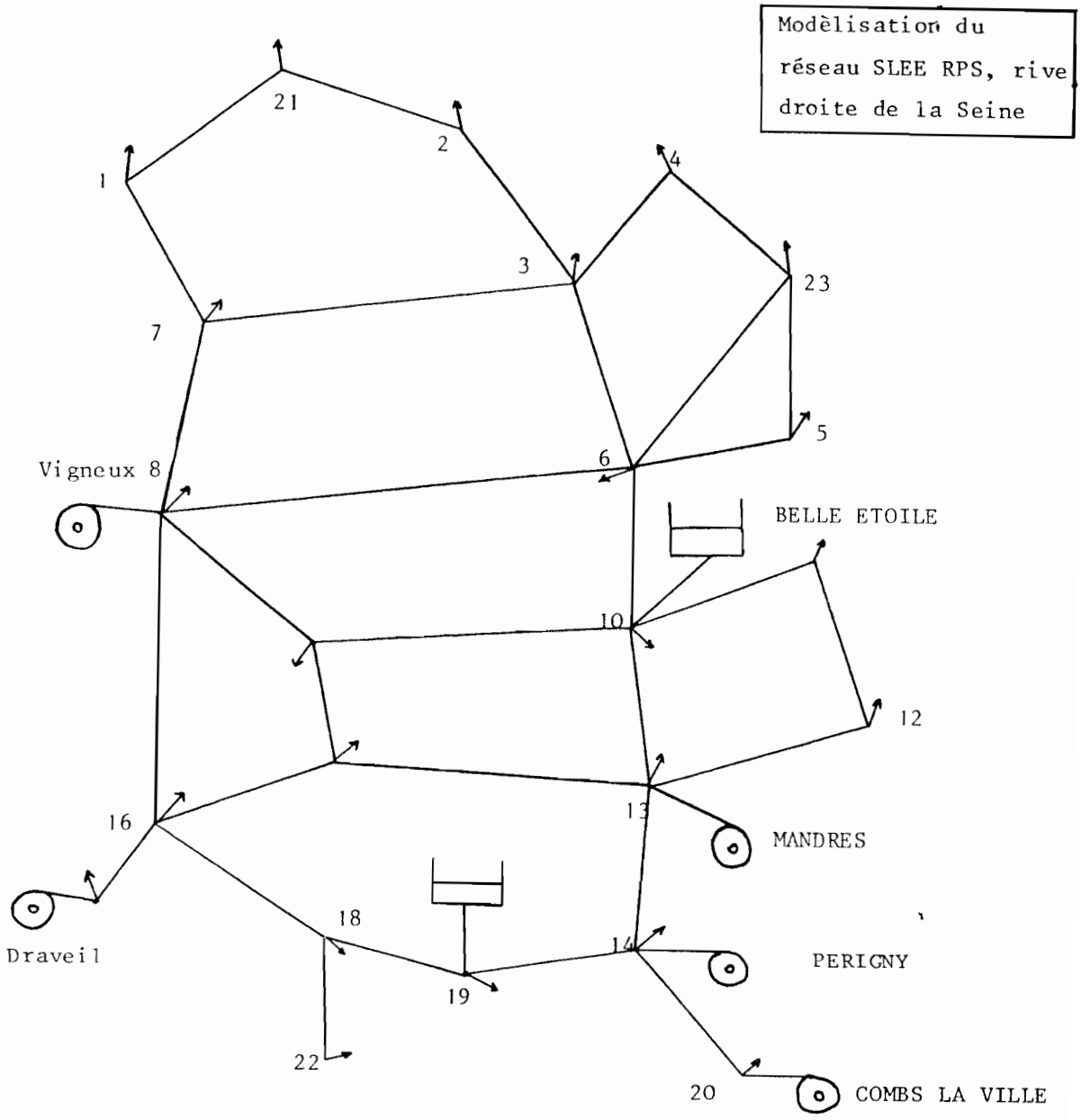

Modèlisation du réseau SLEE RPS, rive droite de la Seine
21
2
1
4
3
23
7
5
6
Vigneux 8
BELLE ETOILE
10
12
13
16
MANDRES
18
Draveil
14
PERIGNY
19
22
20
COMBS LA VILLE

Fig. 4

$$X_t^o = X_{max} \qquad \forall t \in \{0,24\}$$

(X_{max} : capacité maximum des deux réservoirs).

Au cours des itérations, le coût décroit avant d'osciller autour d'une valeur que l'on suppose proche de l'optimum. Mais comme les gains à espérer sont faibles, la décroissance du critère est peu importante (fig.5).

<u>Remarque</u> :

Décrivons la méthode utilisée pour évaluer L_t^k, $t \in \{0,\ldots,23\}$.

Rappelons que L_t^k correspond à la moyenne géographique du coût de l'eau refoulée aux différentes usines de refoulement, à l'instant t (cf. équation 29, chapitre IV).

Si nous simulons le réseau avec la consigne u_t^k, nous connaissons le débit refoulé et le coût associé à ce refoulement en chaque usine, dans l'intervalle de temps $[t,t+1]$.

Soit q_i^k le volume d'eau refoulée par les pompes de l'usine i, pour la commande u_t^k.

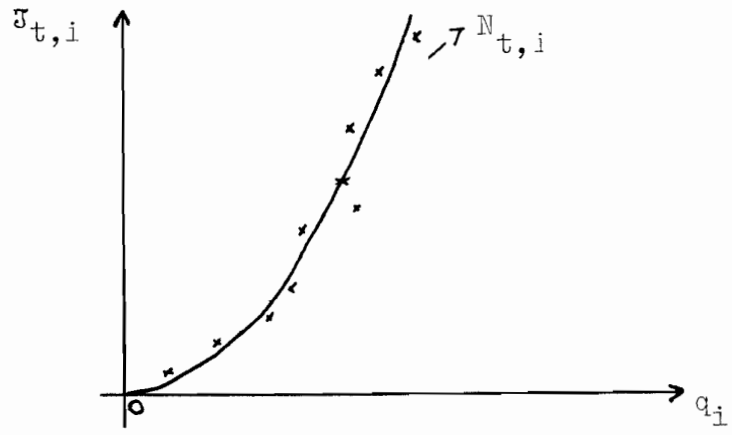
Soit $\mathcal{J}_{t,i}(x_t^k,q^k)$ le coût de refoulement dans l'intervalle de temps $[t,t+1]$, à l'usine i.

On ne peut prendre :

$$\mu_t^k = \frac{1}{m} \sum_{i=1}^{m} \frac{\Delta\mathcal{J}_{t,i}(x_t^k,q_i^k)}{\Delta q_i}$$

($\Delta\mathcal{J}_{t,i}(x_i^k,q_i^k)$ désignant la variation du coût de refoulement à une perturbation Δq_i) car la dérivée de $\mathcal{J}_{t,i}$ en q_i est alors évaluée le long de la caractéristique en débit-pression des pompes qui refoulent à l'usine i. (Si la pente de celle-ci est fortement négative, on peut très bien obtenir un coût marginal de l'eau négatif !). L'évaluation doit se faire en considérant que la perturbation Δq_i oblige à "passer" sur une caractéristique en débit-pression "proche" de celle associée à la consigne $u_{t,i}^k$.

Pratiquement, nous avons évalué une parabole $\eta_{t,i} : \mathbb{R}^+ \to \mathbb{R}$ qui au débit q_i, associe un coût moyen $\eta_{t,i}(q_i)$, à l'usine i. Pour évaluer cette fonction, nous avons simulé le réseau pour toutes les consignes utilisables. Nous avons ensuite déterminé la parabole qui approche au mieux (au sens des moindres carrés) les couples de points $(q_i, \mathcal{J}_{t,i}(x_t, q_i))$. Cette évaluation s'est faite pour un niveau de réservoirs x_t "moyen".



On a pris alors :

$$\mu_t^k = \frac{1}{m} \sum_{i=1}^{m} \frac{d\eta_t(x_t^k, q_i^k)}{d_{q_i}}$$

V-2-3 - <u>Troisième cas</u>

Nous avons utilisé la méthode dans le cas où les stocks sont importants : le réseau précédent est supposé posséder un réservoir de 8000 m3 (noeud n° 10) et un autre de 7500 m3 (noeud n°19), non compris les réserves de sécurité.

Pour que la mauvaise utilisation des stocks coûte "cher", nous avons pris un critère ne comportant que le coût de l'énergie électrique consommée par les pompes de refoulement.

Nous avons, de plus, initialisé l'algorithme avec une trajectoire agrégé éloignée de la trajectoire "idéale" : les réservoirs sont au 1/3 vides à la fin de la période tarifiare basse et pleins à la fin de la période de haut tarif, ceci afin de tester la robustesse de l'algorithme et la décroissance du critère au cours des itérations.

Effectivement, le coût de la séquence de consignes élaborées à chaque pas de l'algorithme diminue puis oscille autour d'une valeur estimée proche du minimum possible (fig.6).

V-2-4 - Quatrième cas

Nous avons enfin test la méthode lorsque le système possède trois variables d'état. On a supposé que le réseau possède un troisième réservoir au noeud n°21, la capacité des réservoirs étant respectivement de 5000, 5500 et 7000 m3 (non compris les réserves de sécurité).

Nous avons de nouveau initialisé l'algorithme avec une séquence de coûts agrégés nulle et une trajectoire de l'état agrégé éloignée de la trajectoire "normale".

Le critère décroit à nouveau, mais la décroissance est plus faible (fig.7).

V-2-5 - A propos du choix des commandes admissibles

La réduction des volumes de calcul apportée par la méthode d'agrégation-désagrégation par rapport à celle qui consiste à résoudre le problème global par la programmation dynamique dépend essentiellement du nombre de réservoirs, non du nombre de consignes possibles.

Si nous désignons par NT le nombre de pas de temps, NC1 le nombre de commandes, NPE le nombre de pas de discrétisation de chaque réservoir et NRES le nombre de réservoirs il faut N résolutions des équations hydrauliques pour résoudre le problème par la programmation dynamique avec :

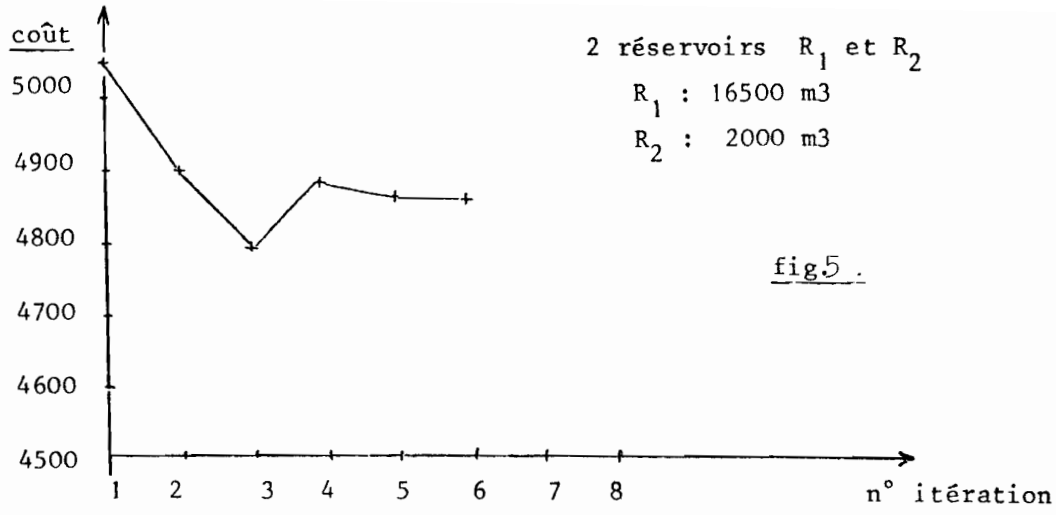


fig.5 .

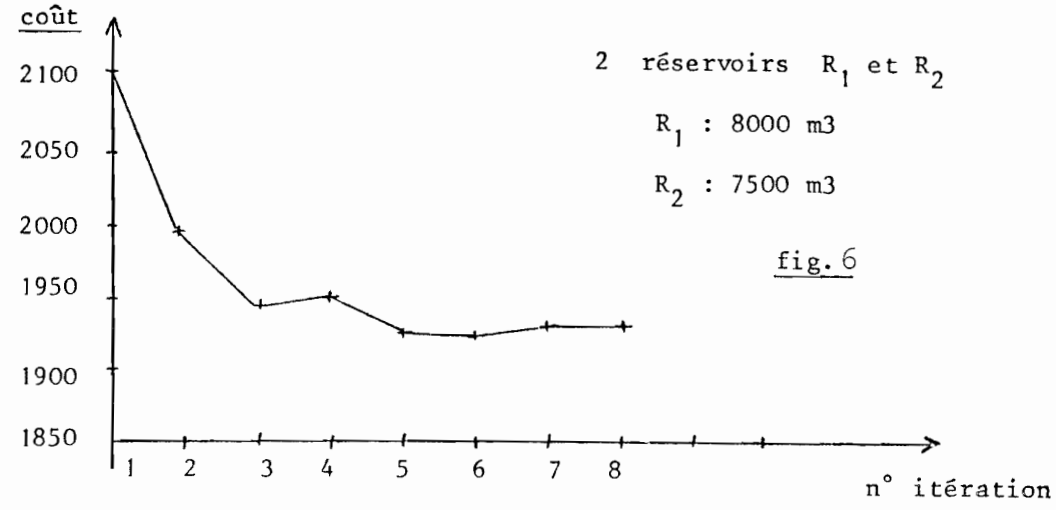


fig. 6

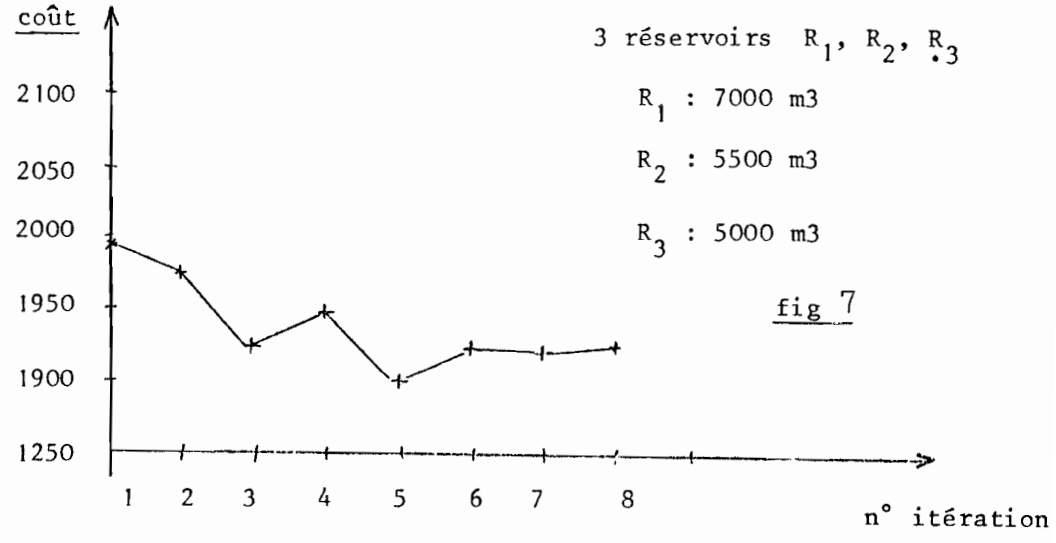


fig 7

les itérations pour diverses capacités de stockages du coût de

$$N = NT \times NC1 \times (NPE)^{NRES}$$

Une désagrégation sur NT pas de temps ne demande que (NT × NC1) résolutions des équations hydrauliques.

Toutefois comme NC1 est grand (de l'ordre de 10^6 pour le réseau SLEE-RPS, rive droite de la Seine), il est indispensable de limiter le nombre de consignes possibles. Aussi avons nous réalisé une première sélection manuelle, portant sur 200 commandes, qui permet d'obtenir un choix suffisant de débits refoulant dans le réseau (répartis entre le débit nul et le débit maximum). A partir de cette première sélection, un programme détermine un certain nombre de consignes (30 à 50 environ) de la façon suivante :

1. L'ensemble des consignes doit permettre de refouler le débit le plus petit et le débit le plus grand.

2. "L'écart" entre le débit, total fourni par deux consignes est supérieur à un "pas" donné.

3. Parmi les consignes qui donnent un débit total "proche" (écart inférieur au pas), on choisit celle qui refoule au coût le plus bas.

Le temps de calcul sur un mini-calculateur T1600 Télémécanique est alors de 20mn environ par itération d'agrégation-désagrégation dans le cas où nous sélectionnons 30 commandes. La programmation dynamique utilisée pour résoudre le problème global demanderait 350 heures de calcul sur le même ordinateur, dans le cas où l'on dispose de 3 réservoirs (dont le volume est discrétisé en 10 pas) et 30 commandes.

V - 3 - Conclusions

a) Les perspectives offertes par la méthode de décomposition-coordination pour la résolution du problème du refoulement optimal semblent limitées puisqu'il n'a pas été possible de l'utiliser avec une décomposition en deux sous-réseaux, avec deux points d'interaction seulement.

Toutefois, dans le cas d'un réseau formé de deux sous-réseaux connectés par une seule canalisation, cette méthode peut être utilisée avec profit. C'est en particulier le cas pour le réseau SLEE-RPS (les sous-réseaux sont de part et d'autre de la Seine).

On peut alors imaginer une procédure associant les deux méthodes : on utilise la technique de décomposition-coordination avec deux sous-systèmes, la résolution de ceux-ci se faisant à l'aide de la méthode d'agrégation-désagrégation.

b) C'est sur cette dernière méthode que nous fondons nos espoirs d'automatisation d'un réseau hydraulique, à l'aide d'un calculateur numérique.

En effet, comme nous l'avons déjà dit, il est possible de concevoir une utilisation en temps réel de l'algorithme :

- au début de chaque heure on mesure la hauteur d'eau dans chaque réservoir. On connaît alors le volume des stocks et il est possible de déterminer la commande à utiliser, à l'instant considéré, par résolution d'un problème du type (δ), chapitre IV.
- En tenant compte de la trajectoire de l'état et des coûts agrégés pour la dernière période, on remet à jour la fonction revenu V, par résolution du problème dynamique agrégé. Cette résolution est très rapide car les calculs sont très simples (moins de 5 secondes sur T1600 avec 10 pas de discrétisation de l'état).

Cette structure en "boucle fermée" permet de corriger les petites fluctuations de l'évolution dans le temps de la consommation, par rapport à l'évolution théorique prise en compte dans le modèle (fig 8).

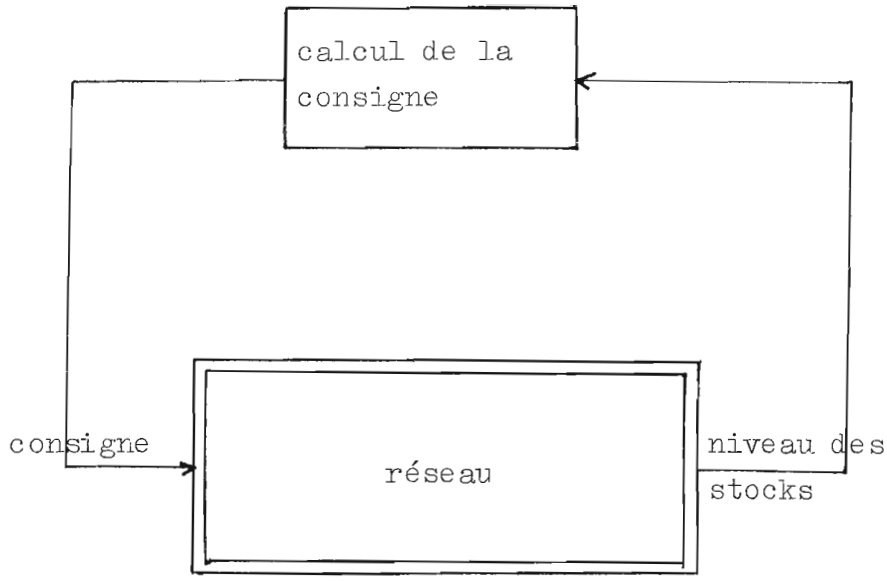


Fig. 8

On peut aussi concevoir une résolution du problème dynamique agrégé sur un horizon de plusieurs jours (1 semaine par exemple), cette résolution était toujours faite une fois par jour (horizon glissant). Cela permettrait de réduire l'arbitraire sur le choix de la cible finale. Ce type de commande du système est à rapprocher de la commande OLFC [36].

Si les fluctuations de la consommation deviennent importantes, il est nécessaire de "recaller" le modèle : il s'agit alors de commande adaptative. Ce recalage peut être fait en tenant compte de l'écart entre la trajectoire théorique et la trajectoire réelle de l'état dynamique du système agrégé (c'est à dire le volume total d'eau dans les réservoirs), par exemple : en fonction de cet écart, on peut redéfinir l'évolution de la consommation. On résoud alors le problème dynamique agrégé avec cette nouvelle hypothèse sur la consommation (sur un horizon donné), puis on désagrège comme précédemment (fig.9).

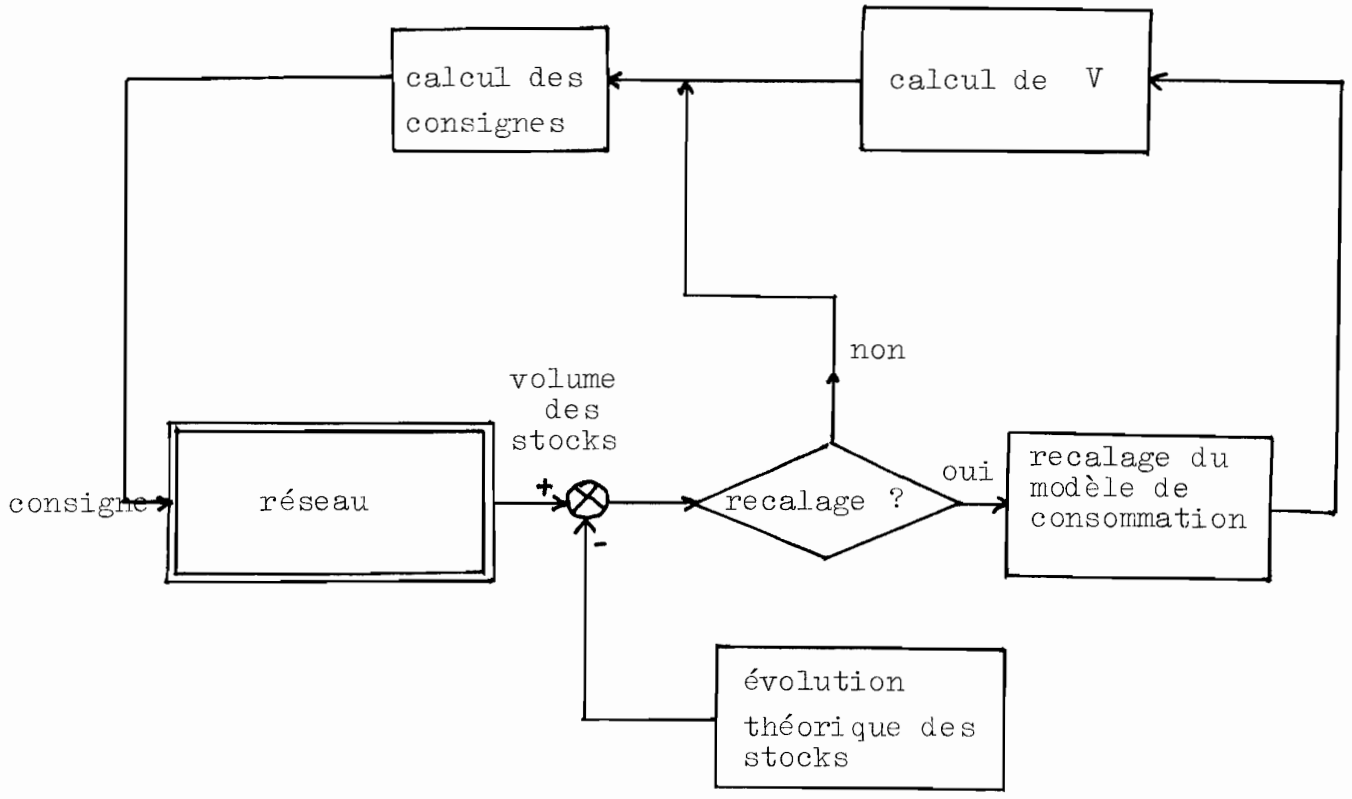


Fig.9

Ce modèle adaptatif demande sans doute de traiter préalablement une masse de données statistiques importante. Mais la mise en oeuvre reste ensuite assez simple. On peut concevoir l'existence d'un certain nombre de tables -sur disque-, donnant la modulation dans le temps de la consommation pour divers situations (temps chaud, temps pluvieux, week-end, etc...). Pour chaque période d'optimisation on choisit une courbe-type et selon l'écart entre les trajectoires théoriques et réelles du volume d'eau dans les réservoirs, on choisira une autre table (à l'aide d'une table de décision, par exemple).

c) Outre les possibilités d'automatisation qu'elle offre, la méthode permet aussi d'avoir une meilleure connaissance du réseau. Elle peut être une aide à la décision dans certaines circonstances, par exemple le choix d'un investissement.

En particulier pour le choix de l'emplacement d'une usine, de la taille d'une canalisation, la méthode donne des indications intéressantes : coûts marginaux de l'eau aux usines, gains théoriques sur les coûts de refoulement, etc...

ANNEXE O

DESCRIPTION DES PROGRAMMES

ANNEXE 0

1- Description de la structure des données

Nous pouvons distinguer trois sortes de données : les données d'entrée que fournit l'utilisateur d'AGREG et qui sont lues par LECTUR ; les données internes (pointeurs, piles, etc...) que peut ignorer l'utilisateur ; les données de sorties fournies par le programme (séquence de consignes élaborées, coût de cette séquence, évolution des hauteurs d'eau dans les réservoirs au cours du temps, etc...).

1-1. données d'entrée

Nous allons décrire, en fait, le fichier de données que lit le sous-programme LECTUR (ce fichier peut-être un paquet de cartes, un fichier stocké sur disque, etc...).

Sauf mention contraire, les entiers sont au FØRMAT I5 et les réels au FØRMAT E10.3.

ligne 1 : titre.

L'utilisateur écrit ce qu 'il veut sur cette ligne.

ligne 2 : colonne 1 à 5 (à droite dans la zone).

NMAX : nombre d'itérations maximum pour l'algorithme d'agrégation-désagrégation. Si NMAX = 0, AGREG n'utilise que les sous-programmes LECTUR et ECRIF.

ligne 3 : titre.

ligne 4 : colonnes 1 à 5, 6 à 10, etc ...

NNØ : nombre de noeuds dans le réseau (les noeuds sont numérotés de 1 à NNØ)

NMA : nombre de mailles indépendantes du réseau (les mailles sont numérotées de 1 à NMA).

NBR : nombre de branches (les branches sont numérotées de 1 à NBR).

NPØM : nombre d'usines de refoulement (les usines sont numérotées, de 1 à NPØM).

NRES : nombre de réservoirs (les réservoirs sont numérotés de 1 à NRES).

NTØT : nombre d'itérations maximum pour la résolution des équations hydrauliques par la méthode de Hardy-Cross.

NKT : nombre de "points" de temps (égal au nombre de pas de temps plus 1) de la période d'optimisation.

NRØT : nombre de permutations circulaires dans les tableaux de données dépendant du temps (voir plus loin).

ligne 5 : colonnes 1 à 5 et 6 à 10.

NPE : nombre de points de discrétisation de la variable dynamique agrégée (volume des stocks).

NEI : nombre de points d'états initiaux pour le problème dynamique agrégé.

colonnes 20 à 30.

CØEMUL : durée, en seconde, du pas de temps.

ligne 6: carte titre.

lignes 7 :

Ces lignes correspondent à la description des NBR branches du réseau (dans l'ordre de leur numérotation) : numéro du noeud-origine, numéro du noeud extrémité, résistance. Pour une branche (I,J), I sera le noeud origine si I<J et le noeud extrémité sinon. La numérotation des branches respecte également une relation d'ordre : si la branche L_1 est (I1,J1) et la branche L_2 est (I2,J2), avec I1<J1 et I2<J2, on aura $L_1 < L_2$ si (I1 <I2) ou si (I1 = I2 et J1< J2.)

colonnes 1 à 5

NØEB(1,L) : numéro du noeud-origine de la branche L(L=1,...,NBR)

colonnes 6 à 10

NØEB(2,L) : numéro du noeud extrémité de la branche L(L=1,...,NBR)

colonnes 20 à 30

R(L) : résistance de la branche L (L = 1, ...,NBR)

ligne 8 titre

Cette ligne est absente si le réseau ne possède pas de mailles indépendantes.

ligne 9

Il s'agit de la description de NMA mailles indépendantes du réseau. Les mailles sont décrites dans l'ordre de leur numérotation, à raison d'une par ligne. Toutefois si le nombre de noeuds est supérieur à 11, une ou plusieurs lignes seront nécessaires pour décrire la maille (12 entiers par ligne, au plus.

1ère ligne, colonnes 1 à 5

NTM(I) : nombre de noeuds de la maille I (I=1,...,NMA)

colonnes 5 à 10, 11 à 15, ... (+ éventuellement, 2ème ligne colonnes 1 à 5, 6 à 10, ...).

NØEM(L,I) : numéro du Lème noeud à la maille I

(I=1, ...,NMA ; L = 1, ..., NTM(I))

remarque : la liste des noeuds d'une maille est ordonnée en suivant l'ordre de rencontre des noeuds lorsque l'on parcourt la maille à partir du noeud initial (choisi arbitrairement) Le noeud initial n'est mentionné qu'une fois.

ligne 10 : titre.

lignes 11 :

Ces lignes permettent la description, dans l'ordre de leur numérotation, des noeuds du réseau : consommation et altitude.

colonnes 11 à 20 et 21 à 30

CØNS(L) : débit, moyenné dans l'intervalle de temps, au noeud L, en m^3/s. (L=1,...,NNØ).

Pour convention, il s'agit d'un débit fourni par le réseau (consommation), il est affecté du signe négatif (-) et s'il s'agit d'une fourniture d'eau au réseau, il est affecté d'un signe positif (+). Ce débit modélise la "consommation" au noeud.

ALT(L) : altitude au noeud L, en m.(L=1,...,NNØ)

ligne 12 : titre

lignes 13:

1ère ligne: colonnes 1 à 5 et 6 à 10

KPI(1) : numéro du noeud correspondant à la 1ère usine.

NPNØ(1) : nombre de groupes de pompes pouvant refouler à la 1ère usine.

Les lignes suivantes permettent de donner la caractéristique en débit-pression et la courbe de rendement des groupes pouvant refouler à la 1ère usine. Pour un groupe, ces données sont lues par LECTUR sur deux lignes.

1ère ligne, Colonnes 6 à 15, 16 à 25, 26 à 30

X1 : débit minimum que peut refouler le groupe, en m3/s.

Y1 : pression statique à la sortie du groupe, pour le débit X1, en mètre d'eau.

ET1: rendement du groupe pour ce débit X1, au FØRMAT F5.3

colonnes 31 à 40, 41 à 50, 51 à 55

X2 : débit "moyen" que peut refouler le groupe, en m3/s.

Y2 : pression statique pour le débit X2, en mètre d'eau.

ET2: rendement correspondant (FØRMAT:F5.3).

2ème ligne , colonnes 6à 15, 16 à 25, 26 à 30

X3 : débit maximum que peut refouler le groupe, en m3/s.

Y3 : pression correspondante, en m/eau.

ET4 : rendement correspondant (FØRMAT:F 5.3).

A la suite de ces lignes de description des groupes de la première usine, il y a les lignes donnant les caractéristiques des groupes de la 2ème, 3ème, ..., NPØM^{ème} usine (≠)

A partir de ces données, sont élaborées des courbes de débit-pression et de débit-rendement, approximations paraboliques (ou linéaires) des courbes réelles.

ligne 14 , colonnes 1 à 5

NC1: nombre de consignes de refoulement (les consignes sont numérotées de 1 à NC1).

lignes 15 ,

Elles correspondent à la liste des consignes de refoulement, à raison d'une consigne par ligne.

colonnes 1 à 5, 6 à 10, ...

NTAC (L,J) : numéro du groupe qui refoule à l'usine L, pour la consigne J (L=1, ..., NPØM ; J=1,NC1). Par convention si, pour la consigne J, aucun groupe ne refoule à l'usine L, alors NTAC(L,J)=0.

lignes 16 : titre.

lignes 17 :

Ces lignes permettent la description des caractéristiques des réservoirs (une ligne par réservoir, dans l'ordre de la numérotation des réservoirs).

colonnes 1 à 5

KL(L) : numéro du noeud correspondant au réservoir L(L=1,...,NRES).

(≠)Par convention, les groupes de l'usine L,L=1,...,NPØM, sont numérotés de 1 à NPNØ(L), dans l'ordre de leur description.

colonnes 11 à 20 ; 21 à 30 ; 31 à 40

XMAX(L) : hauteur maximale du réservoir L, en mètre (L=1,...,NRES).

XMIN(L) : hauteur minimale (tenant compte de la réserve de sécurité), en mètre (L=1,...,NRES).

SS(L) : volume du réservoir L, non compris la réserve de sécurité, en m3 (L=1, ...,NRES).

remarque : la dernière de ces lignes (réservoir NRES) correspond au noeud de référence.

ligne 18 , colonnes 1 à 10

PMAX : pression statique maximum dans les canalisations, en mètre d'eau.

colonnes 11 à 20

PMIN : pression statique minimum dans les canalisations (en mètre d'eau).

colonnes 21 à 30 et 31 à 40

XMAF : valeur maximum de la "fourchette" dans laquelle doit se trouver le volume total des réservoirs à la fin de la période d'optimisation, en m3,

XMIF : valeur minimum de cette "fourchette", en m3.

colonnes 41 à 50

QTØT : tolérance maximum sur les débits dans les branches du réseau lors de la résolution des équations hydrauliques par la méthode de Hardy-Cross.

ligne 19 : titre

ligne 20

colonnes 1 à 5, 6 à 10, ...

LXI(L) : Lème "point" d'état, à l'instant initial, dans la méthode d'agrégation (L=1,...,NEI). Le volume total dans les réservoirs est discrétisé en NPE valeurs. Le "point" 1 correspond à un volume total nul, le point I à un volume (I-1) x PAES, le point NPE au volume total (NPE-1)$\times$ PAES, PAES désignant le pas d'espace (PAES est calculé implicitement dans le sous-programme LECTUR, à partir de NPE et de la somme des volumes des réservoirs).

lignes 21

colonnes 1 à 20, 11 à 20, ...

CØFI(I) : coût, à l'instant final NKT, sur l'état du système dynamique agrégé (I=1,...,NPE). On peut prendre, par exemple, CØFI(I) = 0. si (XMIF ≤ (I-1)xPAES < XMAF) et CØFI(I)=10^{36} sinon.

ligne 22 : titre

lignes 23

Ces lignes décrivent la modulation, au cours de la période d'optimisation, du cout de l'énergie électrique, usine par usine, (la tarification peut être différente selon les usines, mais elle est supposée indépendante de la puissance à fournir).

La description de cette tarification est faite à raison de six pas de temps par ligne .Pour toutes les usines, la donnée du tarif au pas de temps initial se fait en début de ligne.

1ère ligne, 2ème ligne, ... colonnes 1 à 10, 11 à 20, ..., , , 51 à 6

C1(K,L) : coût de l'énergie électrique consommée par une machine de 10kw, fonctionnant dans le pas de temps K(x), avec la tarification de l'usine L(K=1,...,(NKT-1) ; L=1,...,NPØM)

ligne 24 : titre

lignes 25 colonnes 1 à 10, 11 à 20, ..., 51 à 60

C2(L) : coût marginal du traitement de CØEMUL m3 d'eau à l'usine L, pendant un pas de temps (L=1,...,NPØM) (rappelons que CØEMUL est la durée, en sec seconde, d'un pas de temps).

ligne 26 : titre

lignes 27 : colonnes 1 à 10, 11 à 20, ..., 51 à 60

CØMU(K) : écart, exprimé en pourcentage,de la consommation au pas de temps K par rapport à la consommation moyenne CØNTØT, avec CØNTOT

avec $CØNTØT = \sum_{I=1}^{NNØ} CØNS(I)$. (K = 1,...,(NKT-1)

Par exemple, si la consommation au pas de temps 7 est 25% en dessous de la consommation moyenne, on aura :

CØMU(7) =-0.25.

ligne 28 : titre

(x) le pas de temps 1 correspond au pas de temps initial.

lignes 29 : colonnes 1 à 25, 6 à 10, ..., 56 à 60

JT(K) : "point" de discrétisation de l'espace d'état du système dynamique agrégé, correspondant au volume total d'eau dans les réservoirs à l'instant K (K=1,...,NKT) pour la trajectoire agrégée d'initialisation de l'algorithme d'agrégation-désagrégation ($(X^{\circ}_t)_{t=0,T}$ avec les notations de paragraphe 2)

ligne 30 colonnes 1 à 10

ALPHA : coefficient de pondération de la partie quadratique du critère du problème dynamique agrégé (c'est le nombre réel positif α de l'équation 2 du paragraphe 2).

ligne 31 : titre

lignes 32

1ère, 2ème, ..., ligne, colonne 1 à 10, ..., 51 à 60

CØJAG(K) : prix marginal moyen de l'eau à l'instant K (K=1,...,NKT-1) correspondant à la séquence de prix pour l'initialisation de l'algorithme d'agrégation-désagrégation ($(L^{\circ}_t)_{t=0,T-1}$ avec les notations du paragraphe 2).

ligne 32 : titre

ligne 33 : colonnes 1 à 10

EPS : pas de discrétisation de la commande pour le problème dynamique agré en m3. EPS permet la détermination du nombre de commandes agrégées NCØ

ligne 34: titre.

ligne 35 : colonnes 1 à 20, 11 à 20, ..., 51 à 60

PR(L) hauteur d'eau, à l'instant initial, dans le réservoir L (L=1,...,NPE)

1.2) données de sortie

Nous désignons ainsi les données élaborées par AGREG qui ont un lien étroit avec la détermination de la séquence optimale de refoulement dans un réseau hydraulique.

JCØØP(K) : numéro de la consigne à utiliser à l'instant K (K=1,...,NKT-1).

Le tableau JCØØP permet de retrouver les groupes qui refoulent dans le réseau par l'intermédiaire du tableau NTAC. Par exemple, à l'instant K, à l'usine L, il faudra utiliser le groupe LL avec LL=NTAC(L,JCØØP(K)).

CØUTØ : coût de la séquence de consignes (contenue dans le tableau JCØØP.

REV : tableau des revenus pour le problème dynamique agrégé.

REV(K,I) représente le revenu au "point" I de l'espace d'état agrégé, à l'instant K.
(Ce tableau se modifie au cours des itérations d'agrégation-désagrégation).

CØUAG : tableau donnant la séquence de prix du critère pour le système dynamique agrégé.
CØUAG(K) est la valeur de ce prix à l'instant K.

Nous n'avons pas défini le tableau correspondant à l'évolution des hauteurs d'eau des réservoirs.

Toutefois, il est possible de connaître cette évolution par l'intermédiaire d'un sous-programme d'écriture des résultats, appelé par le sous-programme SIMUL à chaque pas de temps : la hauteur d'eau dans le réservoir L(L=1,...,NRES) est PRE(KL(L) (on peut connaître la pression en tout noeud du réseau pour la consigne JCØØP(K), à l'instant K-(K=1,...,NKT-1)- par le tableau PRE).

1.3) Données internes

Nous ne décrivons pas la structure des données internes mais lors de la description des sous-programmes utilisés, certains de ces données seront explicitées.

De plus, nous donnons en annexes, les listings des sous-programmes avec des commentaires qui précisent le rôle de plusieurs d'entre elles.

2) Sous-programmes utilisés

Nous avons déjà précisé que le programme est formé de 5 sous programmes, correspondant aux principales fonctions d'AGREG (lecture

des données, écriture des résultats, etc...) et les sous-programmes appellent eux-mêmes différents sous-programmes. Ce chapitre a pour but de présenter la structure d'AGREG et de décrire succintement les sous-programmes.

2.1) Lecture des données-initialisation

2.1 a) LECTUR

LECTUR lit les données d'entrée (cf §3) et détermine les différentes données d'initialisation. Il fait appel à quatre sous-programmes : PECIR, PINIT, DEMAI et PRINI. Toutes les données sont transmises en CØMMØN. appel : CALL LECTUR

2.1.b) PECIR

Ce sous-programme permet de définir une permutation circulaire sur les éléments d'un tableau

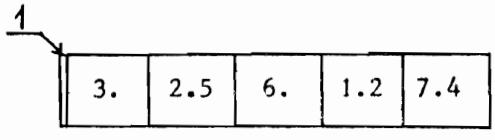
appel : CALL PECIR (A,NA,NR,NS)

A : tableau de NA éléments.

NR : "valeur" de la permutation (Si NR=0, pas de permutation

NS : indicateur du sens de la rotation : +1 si rotation à gauche ; -1 si rotation à droite.

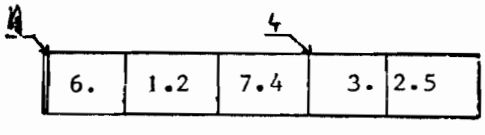
Exemple : Soit un tableau A de 5 réels (fig 1)



Si l'on veut déplacer les éléments de A de 2 adresses vers la gauche, l'appel de PECIR s'écrit :

CALL PECIR (A,5,2,-1)

A devient alors :



Ce sous-programme est utilisé dans LECTUR pour tous les tableaux de données d'entrée dépendant du temps (CØMU, C1, ...). Supposons que l'on ait défini un réseau hydraulique pour lequel on cherche les consignes de refoulement pour une journée, avec un pas de temps d'une heure (NKT=25). Supposons également que nous ayons pris comme instant initial 6h du matin, lo de la création du fichier des données d'entrée.

Si nous désirons alors prendre comme instant initial 7h du matin, sans changer les autres données, il ne sera pas nécessaire de réécrire le fichier en modifiant l'ordre de lecture des tableaux de données dépendant du temps. Il suffit de prendre NRØT=1 (rotation de 1).(*)

2.1.c) PINIT

PINIT détermine un chemin MAIN passant par tous les noeuds du réseau et ayant comme noeud initial le noeud de référence. Ce chemin permet la détermination des chemins liant le noeud de référence aux noeuds où existe une usine ou un réservoir. Il permet également de faire transiter la modification de débit à apporter à un noeud pour que l'équation de bilan de débit en ce noeud (éq. β , §1) soit satisfaite.

appel : PINIT

2.1.d) DEMAI

DEMAI détermine les chemins, autres que les mailles indépendantes du réseau, liant le noeud de référence, aux noeuds avec usine u réservoir.

Après appel de DEMAI, NMA (qui désignait le nombre de mailles indépendantes du réseau) désigne le nombre de chemins devant vérifier une relation de perte de charge dans la méthode de Hardy Cross.

appel : CALL DEMAI

2.1.e) PRINI

PRINI teste les différentes configurations de pompes pouvant refouler dans le réseau et détermine les différentes commandes pour le système dynamique agrégé. L'écart entre deux commandes agrégées est inférieur à EPS (rappelons que ces commandes sont des volumes d'eau "entrant" dans le réseau durant un pas de temps). Ce nombre de commandes agrégées est NCØ.

(*) Il faudra peut-être aussi modifié la cible finale [XMIF,XMAF] du problème dynamique agrégé.

Remarque : Si NMAX = 0, le tableau de consignes NTAC est modifié par PRINI. Celui-ci sélectionne parmi les consignes donnant globalement le même débit (le volume produit durant un pas de temps est le même à EPS près) celle qui coûte le moins cher. Ceci permet de faire une première sélection afin de réduire le volume des calculs lors de l'utilisation de la méthode d'agrégation-désagrégation. Le nombre de consignes sélectionnées est également NCØ.

On conseille donc à l'utilisateur de créer son fichier de données et de faire un premier passage d'AGREG avec NMAX = 0 et un pas de commande agrégé EPS assez petit, (de façon à obtenir une trentaine de commandes agrégées).

Ce premier passage permettra de tester la validité des données d'entrée et de déterminer un ensemble de consignes "raisonnables".

Il suffira alors de modifier le fichier de données en tenant compte des données résultant de ce premier passage (en particulier prendre pour tableau de consignes la sélection effectuée) et de lancer à nouveau AGREG avec NMAX > 0 et un EPS supérieur au précédent (le double par exemple).

PRINI fait appel à trois sous-programmes : RESØL, REMØD et SØMME

appel : CALL PRINI (EPS)

2.1.f) SØMME

appel CALL SØMME(A,B,NA,NB,N,X)

A: tableau de N réels

B: tableau de N réels

SØMME permet d'effectuer les calculs intermédiaires pour la détermination, par une méthode des moindres carrés, d'une fonction parabolique liant le volume total fourni par les pompes durant un pas de temps au coûtde fourniture de ce volume d'eau. Cette fonction servira à la détermination des coûts marginaux, moyennés géographiquement, de l'eau aux points de refoulement.

Nous décrivons plus loin les programmes RESØL et REMØD.

2.2. Révolution du problème dynamique agrégé

Cette résolution se fait par l'intermédiaire de PRØDYN, sous-programme de résolution d'un problème de commande optimale d'un système dynam. dynamique à une variable d'état, dont la description est donnée en annexe.

Nous allons seulement décrire les sous-programmes utilisés dans le cas particulier du problème.

2.2a) CØNTR

CØNTR teste la violation des contraintes du type (3) et (4), §2, pour chaque commande agrégée, chaque point de discrétisation de l'état agrégé et chaque pas de temps.

appel CALL CØNTR (I,J,K,1,KID,PAES,NKT,NPE,1)

I : point de discrétisation de l'état

J : numéro de la commande agrégée

K : pas de temps considéré

KID : indicateur de violation des contraintes (KID=0 si les contraintes sont satisfaites, KID=1 sinon)

PAES: valeur du pas de discrétisation de l'espace d'état

NKT : nombre de "points" de temps dans la période d'optimisation.

NPE : nombre de points de discrétisation de l'espace d'état

2.2b) CØCØ

CØCØ, sous-programme de type FUNCTIØN donne le coût agrégé d'une commande utilisée durant un pas de temps, pour un état agrégé donné.

appel CØCØ(I,J,K)

I : "point" de discrétisation de l'état agrégé

J : numéro de la commande agrégée

K : instant considéré.

2.2c) FSD

FSD permet de caractériser la dynamique du système agrégé (équation (1), §2)

appel CALL FSD(I,J,K,PAES,XKP1)

I : point de discrétisation de l'état.

J : numéro de la commande

K : instant considéré

PAES : valeur du pas de discrétisation de l'état

XKPM : valeur de l'état à l'instant (K+1)

2 .3. Désagrégation

4.3.a) SIMUL

A chaque itération de l'algorithme d'agrégation-désagrégation, SIMUL élabore une séquence de consignes de refoulement pour la période d'optimisation. La détermination de cette séquence se fait par la révolution de (NKT-1) problèmes de minimisations du type PB^k, §2, V^k désignant le tableau de revenu élaboré par PRØDYN à l'itération k de l'algorithme.

appel CALL SIMUL

SIMUL appelle plusieurs sous-programmes : RESØL, REMØD, CØNRE, ETAT, CREA, RESK1, ECRIO.

2.3b) RESØL

RESØL est le sous-programme qui, étant donnée une consigne, un pas de temps et une hauteur d'eau dans chaque réservoir, permet de déterminer les débits dans les canalisations, les pressions et les débits aux différents noeuds.

appel CALL RESØL (K,J)

K : instant considéré (K = 1, ...,(NKT-1))

J : numéro de la consigne (J=1,...,NC1)

RESØL appelle trois sous-programme : CØEPØ, DEDEB, REINR

2.3c) CØEPØ

Rappelons que pour la résolution des équations hydrauliques, nous distinguons trois sortes de noeuds ; ceux ou le débit est connu et indépendant de la pression ; ceux ou la pression est fixe (noeud avec

réservoir) ; ceux enfin où le débit entrant et la pression sont liés par une relation (noeud où refoulent une ou plusieurs pompes). Les deux derniers types cités forment un ensemble qui se modifie avec la consigne utilisée et la hauteur d'eau dans les réservoirs (un noeud avec réservoir devient un noeud à débit imposé si le réservoir est plein). Aussi avons-nous un tableau KØD précisant le type de chaque noeud :

$$\text{KØD}(N) = \begin{cases} 0 & \text{si } N \text{ est un noeud à débit imposé} \\ 1 & \text{si } N \text{ est un noeud à pression imposée} \\ 2 & \text{si } N \text{ est un noeud avec une relation débit-pression} \end{cases}$$
$$(N=1,\ldots,\text{NNØ})$$

CØEPØ recherche dans la liste des caractéristiques des pompes CAPØ , celles qui correspondent à la consigne utilisée et les chargent dans un tableau CØE. Il initialise également le tableau KØD.

<u>appel</u> : CALL CØEPØ(J)

J : numéro de la consigne utilisée (J=1,...,N C 1)

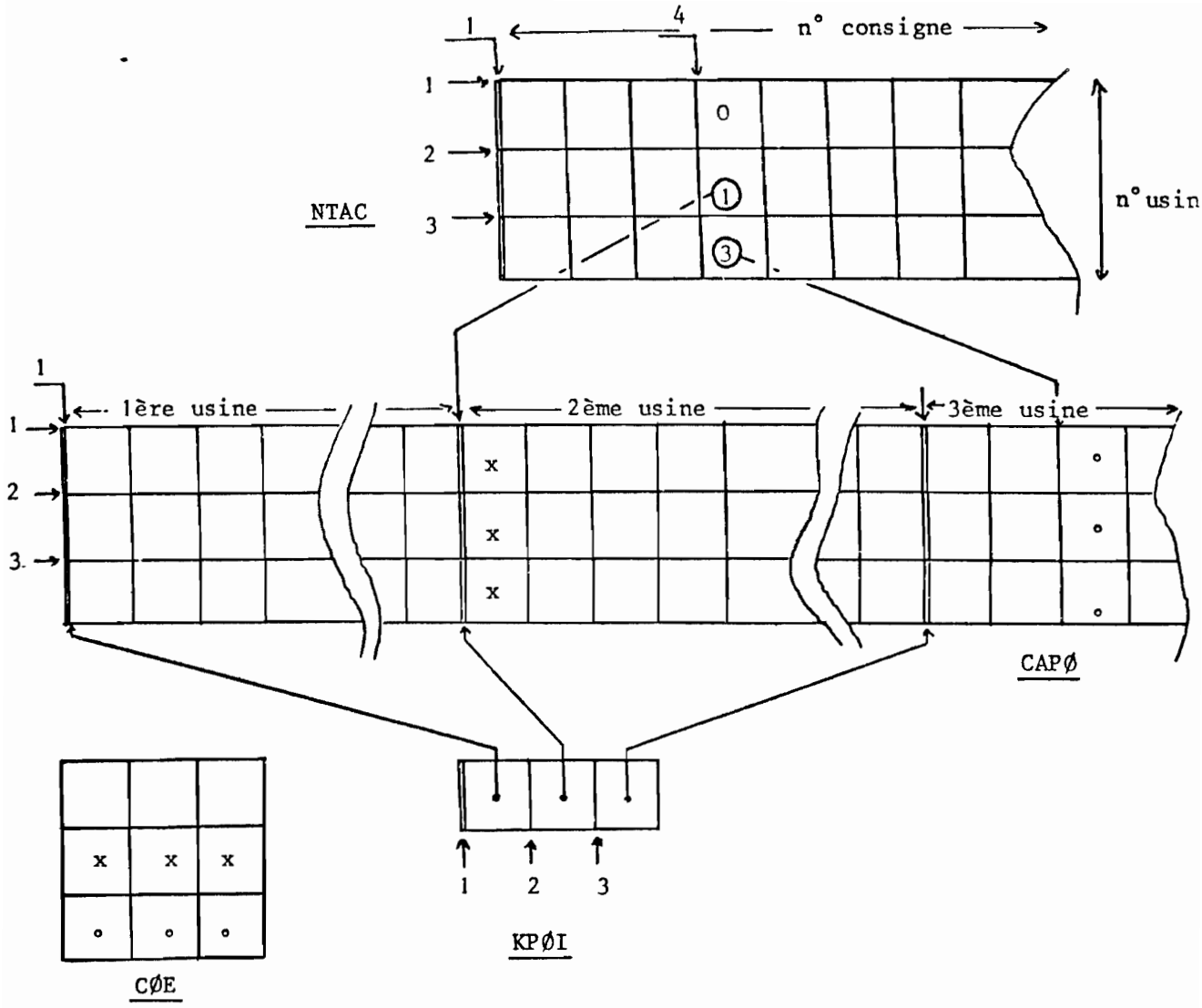
<u>exemple</u> : considérons un réseau avec 3 usines. Supposons qu'à la consigne n°4 corresponde la pompe n°0 à la première usine (c'est à dire qu'aucune pompe ne refoule en cette usine pour la consigne "4"), la pompe n° à la 2ème usine et la pompe n°3 à la 3ème usine.

Après appel de CØEPØ, les caractéristiques des pompes 1 (2ème usine) et 3 (3ème) sont chargées dans CØE. KØD est mis à 2 pour les numéros des noeuds correspondant aux usines 2 et 3 et celui de la 1ère usine est mis à 0 puisqu'il n'y a pas de relation débit pression en ce noeud pour la configuration n°4.

<u>remarque</u> : On a déjà précisé que les caractéristiques des pompes sont supposées paraboliques, c'est à dire que la relation liant le débit Q à la pression P si la sortie de la pompe est de la forme :

$$P = ao + a_1 Q + a_2 Q^2$$

La courbe caractéristique du groupe n°I de l'usine n°L seront ao= CAPØ(1,N) ; a_1 = CAPØ(2,N) ; a_3 = CAPØ(3,N) avec (N=KPØI(L)-1+I) et KPØI(L) désignant l'adresse dans CAPØ de la courbe caractéristique premier groupe de pompes refoulant à l'usine n°L (L=1,...,NPØM).



2.31. DEDEB

DEDEB résoud pour une configuration donnée du réseau (c'est à dire étant connues les consommations en chaque noeud, les caractéristiques des pompes et les hauteurs d'eau dans les différents réservoirs) ses équations hydrauliques.

Ce programme utilise un sous-programme : EQMA2 qui détermine la modification de débit à apporter à un chemin pour que l'équation de perte de charge y soit satisfaite (équation α , §1)

appel CALL DEDEB

2.3e) REINR

Le sous-programme DEDEB suppose que les équations de bilan de débit au noeud (équations (α), §1) sont satisfaites. Supposons que ces équations sont satisfaites. Elles le seront après avoir modifier le débit sortant d'un noeud. -la consommation par exemple - si nous répercutons la modification jusqu'au noeud de référence par l'intermédiaire des branches de chemin d'initialisation MAIN.

C'est le rôle du sous-programme REINR de transmettre la modification de débit.

appel : CALL REINR (N,DEL)

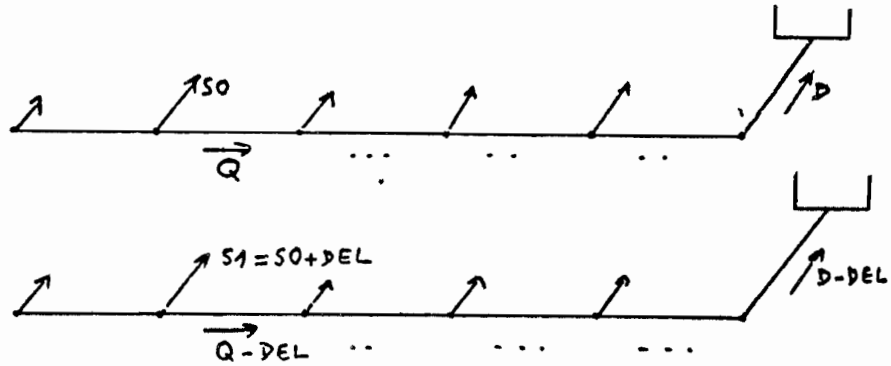
N : numéro du noeud où se fait la modification de débit DEL :

exemple : A l'instant K la consommation au noeud N est SO et à l'instant (K+1) S1. Supposons satisfaite à l'instant K les équations de bilan de débit. Au noeud N, à l'instant K+1, elles le seront au noeud N après appel de REINR :

CALL REINR(N,DEL)

avec DEL = S1-SO

Instant K :



Instant (K+1)

2 .3f) REMØD

Il se peut que la méthode de Hardy-Cross diverge si l'initialisation du réseau est trop "éloigné" du régime d'équilibre. Après la résolution des équations hydrauliques par RESØL, un test sur le nombre d'itérations permet de vérifier la convergence de la procédure de calcul. S'il y a eu divergence, l'appel de REMØD permet d'initialiser différamment le réseau avant de réutiliser DEDEB. S'il y a à nouveau divergence, il y a

il y a une nouvelle initialisation puis résolution par DEDEB, et ainsi de suite jusqu'à ce qu'une initialisation permette d'obtenir la convergence. Toutefois, le nombre de réinitialisations a volontairement été limité à 4, la méthode de réinitialisation utilisée permettant, le plus souvent, d'obtenir la convergence en moins de 2 réinitialisations.

Outre DEDEB, REMØD fait appel au sous-programme REINR

appel : CALL REMØD(K)

2.3g) CØURE

CØURE est le sous-programme qui détermine le coût d'une consigne utilisé durant un pas temps.

appel : CALL CØURE(J,K)

J : numéro de la consigne ($J \in \{1,\ldots,NC1\}$)

K : pas de temps considéré ($J \in \{1, \ldots, NKT-1\}$)

Comme nous avons supposé les courbes de rendements des pompes paraboliques, trois coefficients définissent chaquz courbe de rendement. Ces coefficients sont rangés dans le tableau CØEPØ.

exemple : Si le débit sortant de l'usine n°L est S, pour la consigne J, le rendement du groupe refoulent à l'usine L est
$REN = b_o+b_1S+b_2S^2$ avec b_o=REPØ(1,KK), b_1=REPØ(2,KK), b_2= REPØ(3,KK) ;

KK=KPØI(L)+M-1 et M=NTAC(L,J).

2.3h) CØNRE

CØNRE vérifie la non-violation des contraintes : la pression statique au noeud est supérieur au minimum et inférieur au maximum ; les réservoirs ne se vident pas.

appel CALL CØNRE(KID)

KID : indicateur ; KID=0 si les contraintes sont vérifiées ; KID $\neq$ 0 sinon.

2.3.J) ETAT-

ETAT permet de déterminer le volume total des stocks à l'instant (K+1) (K=1,...,NKT-1) connaissant ce volume à l'instant K et les débits entrant dans les réservoirs durant l'intervalle de temps [K,K+1]

appel CALL ETAT (X,XK)

XK: Volume des stocks à l'instant K

X : Volume des stocks à l'instant (K+1)

2.3.k) RESK1

RESK1 est le sous-programme qui initialise les données pour l'instant (K+1), (K=1,...,NKT-1) lorsque la tache de désagrégation à l'instant K est réalisée.

appel CALL RESK1 (K,XK)

K : pas de temps

XK : volume des stocks à l'instant (K+1), sachant que la consigne utilisée dans l'intervalle [K,K+1] est la consigne qui est solution du problème de désagrégation au pas de temps K.

2.31) ECRIO

ECRIO permet l'écriture des principaux résultats à chaque pas de temps K : hauteurs d'eau dans les réservoirs, coût de la consigne utilisée à l'instant K, coût cumulé des K premières consignes (K = 1,..., NKT-1).

appel CALL ECRIO(K)

K : pas de temps considéré

2.4. Ecriture des résultats d'une itération de l'algorithme d'agrégation-désagrégation

appel CALL ECRIT

Ce sous-programme "donne", à l'utilisateur la trajectoire agrégée (X_t^k) la séquence de coûts agrégés (L_t^k) et la séquence de consignes JCØØP.

2.5) Ecritures des principales données d'entrée

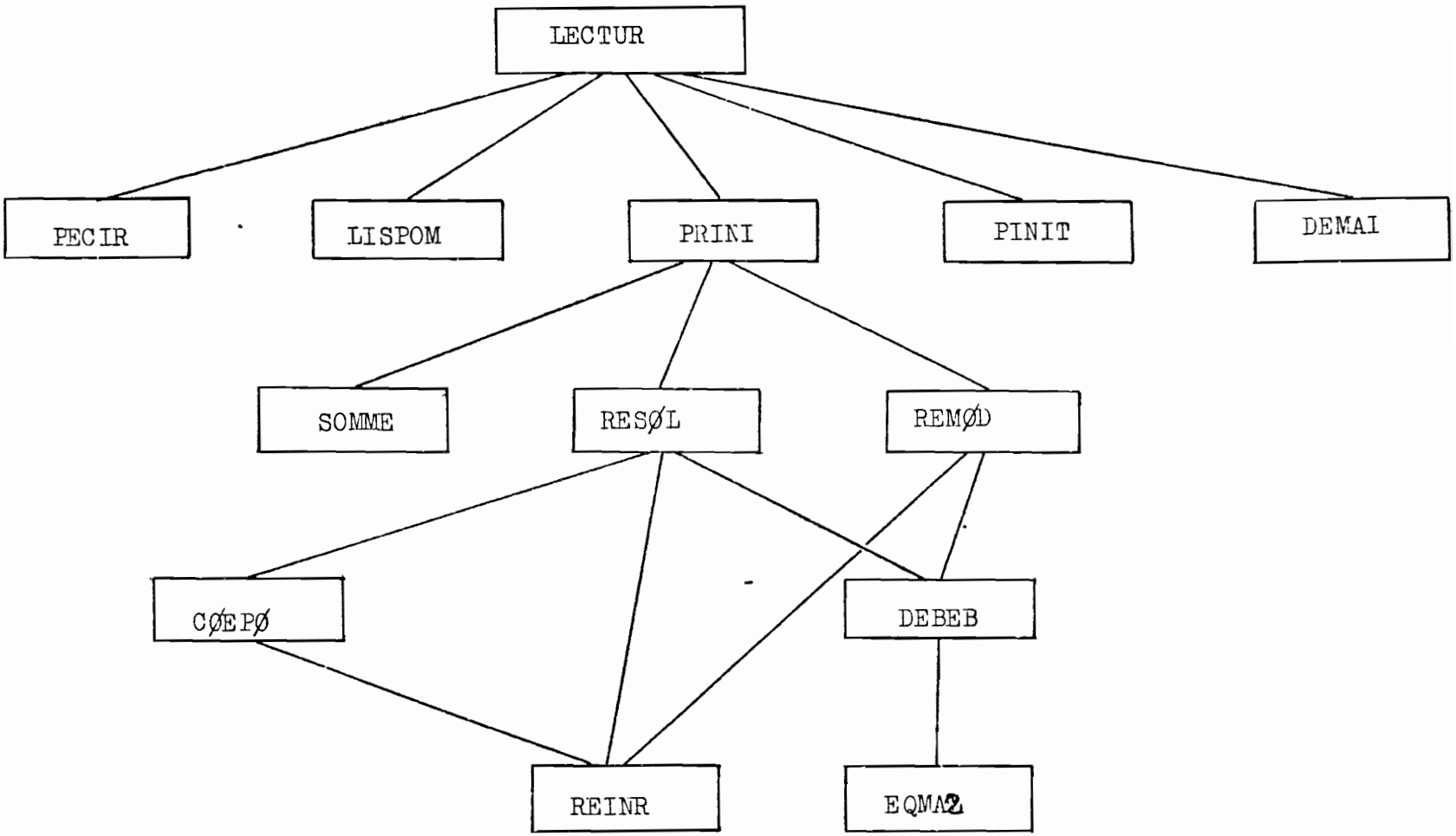
Le sous programme permet de vérifier si les principales données d'entrée résistance des canalisations, coût des différentes consignes, etc... sont correctes.

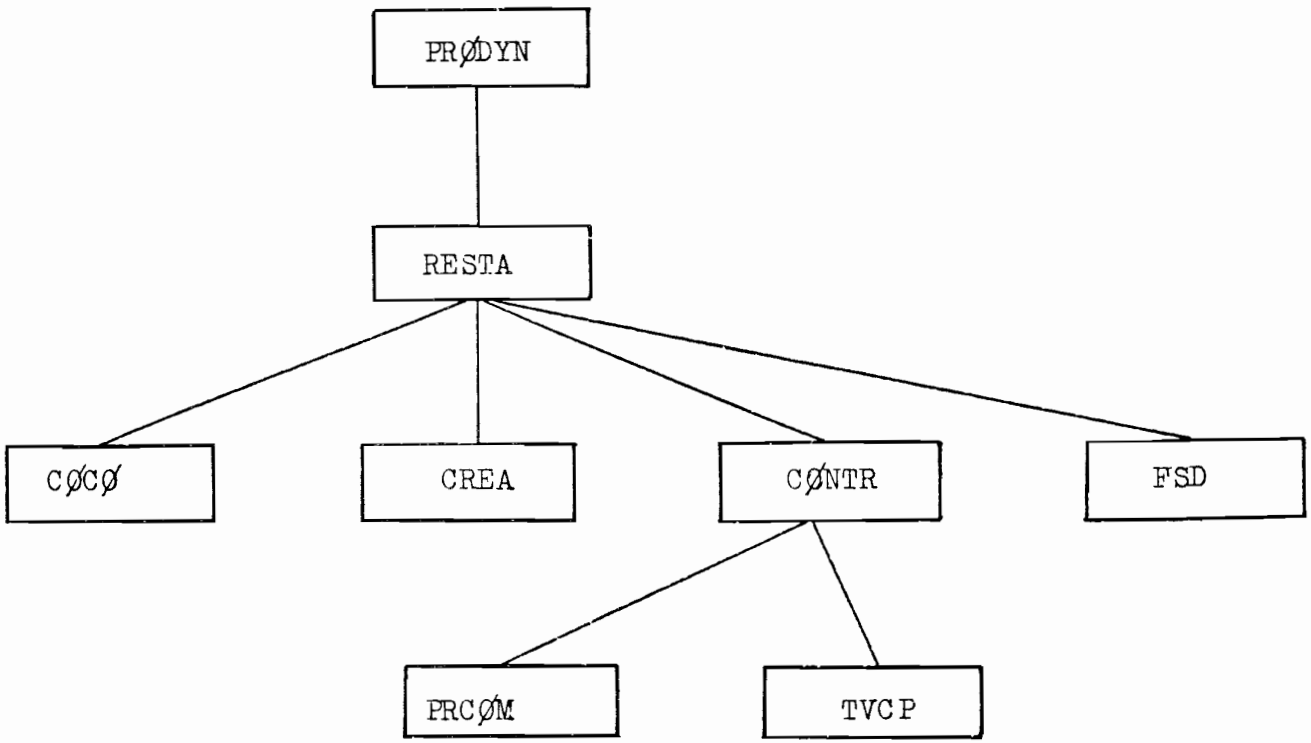
appel CALL ECRIF

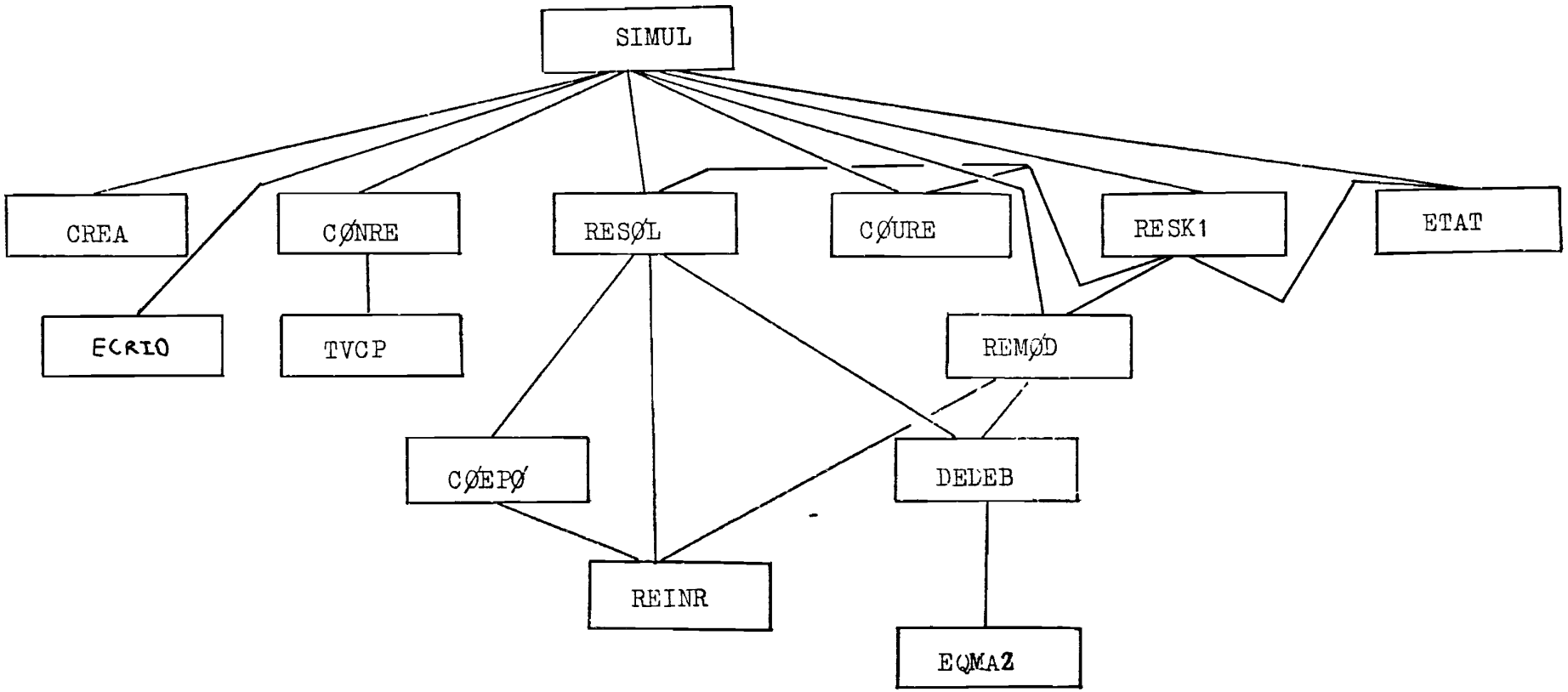
Lorsque le nombre d'itérations d'agrégation-désagrégation est nul (NMAX=0), AGREG est arrêté après exécution de ECRIF.

A N N E X E 1

STRUCTURE DES SOUS-PROGRAMMES


LECTUR
PECIR
LISPOM
PRINI
PINIT
DEMAI
SOMME
RESØL
REMØD
CØEPØ
DEBEB
REINR
EQMA2


PRØDYN
RESTA
CØCØ
CREA
CØNTR
FSD
PRCØM
TVCP


SIMUL
CREA
CØNRE
RESØL
CØURE
RESK1
ETAT
ECRIO
TVCP
REMØD
CØEPØ
DELEB
REINR
EQMAZ

ANNEXE 2

LISTING DES PROGRAMMES FØRTRAN

```
      COMMON /NLOC/NNO,NBR,NFUM,NPES,NMA,K2,NTOT,NIF,NRFF,NCO,NC1,
     1KN,NSF,NPE,NET,KP(30),NTM(20),KOD(30),MAIN(60),MAINB(60),KL(4),
     2KP1(30),KP1(6),KPOI(6),NPNO(6),LXI(5),KAT(30),NUMA(2,20),
     3NOEM(30,20),NTAC(6,200),TCOOP(375),NOEB(2,50),JCOOP(25)
      COMMON /RLOC/FMIN,FMAX,GTOT,PAES,XLOT,COEMUL,XMIF,XMAF,
     1R(50),FRF(30),CONS(30),S(50),R(50),AL1(30),C1(150),COMU(25),
     2C2(6),COFI(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),TACOM(30),CONTOT,COUTO
      COMMON/RLOC1/ALPHA,FRAAG(25),PREI(4),COUPR(30),COCAG(2),NMAX
      CALL LECTUR
      CALL ECRIF
      IF(NMAX.50.50.JO
   10 DO 20 I=1,NMAX
      CALL PROBYN(NKT,NPE,NCO,TIC,NEL,LXI,COFI,REV,TCOOP,PAES)
      CALL SIMU
   20 CALL ECRIT
   50 CALL EXIT
      END
```

```
      SUBROUTINE LECTUR
      COMMON /RLOC/NNO,NBR,NPOM,NRES,NMA,K2,NTOT,NT1,NREF,NCO,NC1,
     1KN,NKT,NPE,NE1,KP(30),NTM(20),KOD(30),MAIN(60),MAINB(60),KL(4),
     2KP1(30),KPT(6),KPOT(6),NPNO(6),LXI(5),KAU(30),NOMA(2,20),
     3NOEM(30,20),NTAC(6,200),ICOOP(375),NOEB(2,50),JCOOP(75)
      COMMON /RLOC/PMIN,PMAX,QTOT,PAES,XTOT,COEMUL,XMIF,XMAF,
     1Q(50),PRE(30),CONS(30),S(30),R(50),ALT(30),C1(150),CUMU(25),
     2C2(6),COFI(15),CAPO(3,150),RFPO(3,150),RFV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),IACOM(30),CONTOT,COUTO
      COMMON/RLOC1/ALPHA,TRAAG(25),PRFT(4),COUPR(30),COCAG(2),NMAX
      LEC=129
      READ(LEC,1000)TITRE
      READ(LEC,3000)NMAX
C           * LECT  DU NOMBRE DE NOEUDS,DE MAILLES,ETC....  *
      READ(LEC,1000)TITRE
      READ(LEC,3000)NNO,NMA,NBR,NPOM,NRES,NTOT,NKT,NROT
      KN=NKT
      READ(LEC,2000)NPE,NE1,COEMUL
C           *  LECT. DES RESISTANCES DES BRANCHES  *
      I1=1
      KAU(1)=1
      READ(LEC,1000)TITRE
      DO 60 I=1,NBR
      READ(LEC,2000)N1,N2,RE
      IF(I1-N1)10,20,20
   10 I2=I1+1
      DO 15 N=I2,N1
   15 KAU(N)=I
      I1=N1
   20 NOEB(1,I)=N1
      NOEB(2,I)=N2
      R(I)=RE
      Q(I)=0.
   30 CONTINUE
      N3=N1+1
      DO 35 I=N3,NNO
   35 KAU(I)=NBR+1
C
      IF(NMA)65,65,37
C           *    LECT  DES MAILLES & REMPLACEMENTS DES NO DE NOEUDS
C                PAR CEUX DES BRANCHES          *
   37 READ(LEC,1000)TITRE
      DO 60 I=1,NMA
      READ(LEC,3000)N,(NOEM(L,I),L=1,N)
      NTM(I)=N
      N=N+1
      NOEM(N,I)=NOEM(1,I)
      NOMA(1,I)=NOEM(1,I)
      NOMA(2,I)=NOEM(N,I)
C
      DO 60 L=2,N
      N1=NOEM(L-1,I)
      N2=NOEM(L,I)
      IF(N1-N2)40,40,50
   40 N3=KAU(N1)
```

```
      N4=KAD(N1+1)-1
      DO 45 K=N3,N4
      IF(NOEB(2,K)-N2)45,47,45
   45 CONTINUE
   47 NOEM(L-1,1)=K
      GO TO 60
   50 N3=KAD(N2)
      N4=KAD(N2+1)-1
      DO 55 K=N3,N4
      IF(NOEB(2,K)-N1)55,57,55
   55 CONTINUE
   57 NOEM(L-1,1)=-K
   60 CONTINUE
C
C            *  LECT. DES CONS. ET ALT.      *
   65 READ(LEC,1000)TITRE
      CONTOT=0.
      DO 70 I=1,NNO
      READ(LEC,4000)CONS(I),ALT(I)
      CONTOT=CONTOT+CONS(I)
      KOD(I)=0
   70 S(I)=-CONS(I)
      CONTOT=CONTOT*COEMUL
C
      IF(NPUM)80,80,75
C            *  CARACTERISTIQUES DES POMPES ; CONSIGNES POSSIBLES    *
   75 READ(LEC,1000)TITRE
      CALL LISPOM
C
   80 IF(NRES)100,100,85
C            *  CARACTERISTIQUES DES RESERVOIRS    *
   85 READ(LEC,1000)TITRE
      XTOT=0.
      DO 90 L=1,NRES
      READ(LEC,5000)J,P1,P2,SS
      PRE(L)=(P1+P2)/2.
      KI(L)=1
      KOD(J)=1
      SEC(L)=SS/((P1-P2)*COEMUL)
      XMAX(L)=P1
      XMIN(L)=P2
      XTOT=XTOT+SS
   90 CONTINUE
      PAES=XTOT/(NPE-1)
      NREF=1
  100 READ(LEC,6000)PMAX,PMIN,XMAF,XMIF,QTOT
C
C            * POINTS INITIAUX ; COUT FINAL     *
      READ(LEC,1000)TITRE
      READ(LEC,3000)(LXI(L),L=1,NET)
      READ(LEC,4000)(CUFI(L),L=1,NPE)
      READ(LEC,1000)TITRE
      K=NKT-1
C            * COUT ELECTRICITE ; COUT MARGINAL DE TRAITEMENT DE L'EAU  *
      DO 110 L=1,NPUM
      CI(L,K)=0.
```

```
      READ(LEC,6000) (C1((LL-1)*NKT+L),L=1,K)
      L=(LL-1)*NKT+1
  110 CALL FECIR(C1(L),K,NROT,1)
      READ(LEC,1000) TITRE
      READ(LEC,6000) (C2(L),L=1,NPOM)
      READ(LEC,1000)TITRE
C           *  MODULATION DE LA CONS.      *
      READ(LEC,6000) (COMU(L),L=1,K)
      CALL FECIR(COMU,K,NROT,1)
      COMU(NKT)=0.
C           *   TRAJECTOIRE AGREGEE INITIALE   *
      READ(LEC,1000)TITRE
      READ(LEC,3000) (JCOOP(L),L=1,NKT)
      DO 180 L=1,NKT
  180 TRAAG(L)=(JCOOP(L)-1)*PAES
      CALL FECIR(TRAAG,K,NROT,1)
      READ(LEC,6000)ALPHA
      DO 190 L=1,NPE
      A=((L-1)*PAES-TRAAG(NKT))/COEMIL
  190 COF1(L)=COF1(L)+ALPHA*A*A
C           *    COUT AGREGE INITIAL   *
      READ(LEC,1000)TITRE
      READ(LEC,6000) (COUAG(L),L=1,K)
      CALL FECIR(COUAG,K,NROT,1)
C           *   'PAS' DE LA COMMANDE AGREGEE    *
      READ(LEC,1000)TITRE
      READ(LEC,6000)EPS
C           * HAUTEURS D'EAU DANS LES RES. A L'INSTANT INITIAL  *
      READ(LEC,1000)TITRE
      READ(LEC,6000) (PRE(KL(L)),L=1,NRES)
C           *    DETERMINATION DES COMMANDES AGREGEES   *
      CALL PINIT
      CALL DEMAI
      CALL PRINI(EPS)
      RETURN
 1000 FORMAT(A4)
 2000 FORMAT(2I5,10X,2E10.3)
 3000 FORMAT(12I5)
 4000 FORMAT(10X,2E10.3)
 5000 FORMAT(I5,5X,3E10.3)
 6000 FORMAT(6E10.3)
      END
```

```
      SUBROUTINE TVCP(P,PMIN,PMAX,IDV)
      IDV=0
      IF(P-PMAX)20,20,10
   10 IDV=2
   15 RETURN
   20 IF(P-PMIN)25,25,15
   25 IDV=1
      RETURN
      END
```

```
      SUBROUTINE LISPOM
      COMMON /NLOC/NNO,NBR,NPOM,NRES,NMA,K2,NTLT,NIT,NREF,NCO,NC1,
     1KN,NCT,NPF,NET,KP(30),NTM(20),KUU(30),MAIN(60),MAINR(60),KL(4),
     2KP1(30),KPI(6),KPOI(6),NPNO(6),LXT(5),KAO(60),NOMA(2,20),
     3NOEM(30,20),NTAC(6,200),TCOOP(375),NOFR(2,50),JCOOP(25)
      COMMON /RLOC/PMIN,PMAX,QTOT,PAFS,XTOT,COFMUL,XMTF,XMAF,
     1Q(50),PRF(30),CONS(30),S(30),R(50),ALT(30),CI(150),COMU(25),
     2C2(6),CUFT(15),CAPO(3,150),REPO(3,150),REV(375),COF(4,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),TACOM(30),CONTOT,COUTO
      KPO=1
      LEC=129
      DO 40 J=1,NPOM
      KPOI(J)=KPO
C
C          *  LECTURE DES DONNEES PROPRES AUX POMPES DE L'USINE  J  *
      READ(LEC,1000)LL,N1
      KPI(J)=LL
      KP(LL)=J
      NPNO(J)=N1
      DO 40 K=1,N1
      READ(LEC,2000)X1,Y1,ET1,X2,Y2,ET2,X3,Y3,ET3
      A=(Y1-Y2)/(X1-X2)
      B=(Y1-Y3)/(X1-X3)
      C=(A-B)/(X2-X3)
      IF(C)30,30,40
   30 D=A-C*(X1+X2)
      IF(D)35,35,40
   35 CAPO(2,KPO)=D
      CAPO(3,KPO)=C
      CAPO(1,KPO)=Y1-CAPO(2,KPO)*X1-C*X1**2
      GO TO 50
   40 CAPO(2,KPO)=0.
      A=X1*X1+X2*X2+X3*X3
      B=X1+X2+X3
      C=X1*Y1+X2*Y2+X3*Y3
      D=Y1+Y2+Y3
      CAPO(2,KPO)=(3.*C-B*D)/(3.*A-B*B)
      CAPO(1,KPO)=(D-CAPO(2,KPO)*B)/3.
C
   50 A=(ET1-ET2)/(X1-X2)
      B=(ET1-ET3)/(X1-X3)
      C=(A-B)/(X2-X3)
      REPO(2,KPO)=A-C*(X1+X2)
      REPO(3,KPO)=C
      REPO(1,KPO)=ET1-REPO(2,KPO)*X1-C*X1**2
C
   60 KPO=KPO+1
C
C          *  LECTURE DE LA LISTE DES CONSIGNES  *
C
      READ(LEC,1000)NT
      DO 70 J=1,NT
   70 READ(LEC,1000)(NTAC(I,J),I=1,NPOM)
      RETURN
 1000 FORMAT(10I5)
 2000 FORMAT(2X,2(E10.3,F5.3)/5X,2E10.3,F5.3)
      END
```

```
      SUBROUTINE FINIT
      COMMON /MOD/NNR,NAR,NFOM,NAFS,NSA,KT,NIOT,NFT,NKEF,NC0,NC1,
     1KN,NS1,NPE,NFI,FF(30),NTM(20),NJU(30),MAIN(60),MAINR(60),KL(4),
     2KF1(30),KF1(6),NFOT(6),NFNT(6),LXT(5),KAD(30),NOMA(2,20),
     3NUEM(50,20),NTAC(6,200),TOOOF(325),NOFR(2,50),JCOFF(25)
      COMMON /RI/PC/PMIN,PMAX,QTIO,PAFS,XTOT,COFMIL,XMTF,XMAF,
     1B(50),PRF(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2T2(6),COFI(15),CAPU(6,150),FFPT(3,150),REV(375),COF(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),TACOM(30),CONTOT,COUTO
C
C             *  DETERMINATION DU CHEMIN D'INITIALISATION MATN  *
C
      N=NREF
      MATN(1)=N
      K2=1
      I=1
      NO=1
      KF1(1)=1
   81 N1=N
      N2=K2
      K2=K2+1
      N3=KAD(N1)-1
      IF(N3)860,860,83
   83 DO 86 K=1,N3
      IF(N1-NOFR(2,K))86,84,86
   84 N=NOFR(1,K)
      DO 85 J=1,N2
      IF(MATN(J)-N)85,84,85
   85 CONTINUE
      MAIN(K2)=N
      I=I+1
      NO=NO+1
      KF1(NO)=K2
      MAINR(K2-1)=-K
      IF(I-NNO)81,100,100
   86 CONTINUE
  860 N3=KAD(N1)
      N4=KAD(N1+1)-1
      IF(N3-N4)861,861,890
  861 DO 89 K=N3,N4
   87 N=NOFR(2,K)
      DO 88 J=1,N2
      IF(MATN(J)-N)88,89,88
   88 CONTINUE
      MAIN(K2)=N
      I=I+1
      NO=NO-1
      KF1(NO)=K2
      MAINR(K2-1)=K
      IF(I-NNO)81,100,100
   89 CONTINUE
  890 NO=NO-1
      N3=KF1(NO)
      N=MAIN(N3)
      MATN(K2)=N
C
      IF(N1-N)891,891,894
  891 N3=KAD(N1)
      N4=KAD(N1+1)-1
      DO 892 J=N3,N4
      IF(NOFR(2,J)-N1)892,89[illegible],892
  892 CONTINUE
  893 MAINR(K2-1)=1
      GO TO 81
C
  894 N3=KAD(N)
      N4=KAD(N+1)-1
      DO 895 K=N3,N4
      IF(NOFR(2,K)-N1)895,896,895
  895 CONTINUE
  896 MAINR(K2-1)=-1
      GO TO 81
C
  100 RETURN
```

```
      SUBROUTINE DEMAI
      COMMON /NLOC/NNO,NAR,NPOM,NRES,NMA,K2,NTOT,NTI,NREF,NCO,NC1,
     1KN,NKT,NPE,NET,KE(30),NIM(20),KOD(30),MAIN(60),MAIND(60),KL(4),
     2KPT(30),KPI(6),KPOI(6),NPNO(6),LXI(5),KAD(30),NOMA(2,20),
     3NOEM(30,20),NTAG(6,200),ICOUP(375),NOEB(2,50),JCOOP(25)
      COMMON /RLOC/PMIN,PMAX,QTOT,PAFS,XTOT,COEMUL,XMIF,XMAF,
     1Q(50),PRE(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COEI(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),IACOM(30),CONTOT,COUTO
C           *OFTER. DES CHEMINS AUTRES QUE LES MAILLES INDEPENDANTES  *
      DO 1 L=1,NPOM
      NN=KPT(L)
    1 KOD(NN)=2
      NO=K2-1
      IF(NPOM+NRES-1)130,130,110
  110 NM=NMA+1
      NOMA(1,NM)=0
      DO 127 K=1,NO
      N2=MAIN(K2-K+1)
      IF(N2-NREF)510,127,510
  510 IF(KOD(N2)-1)127,111,111
  111 DO 112 I=NM,NMA
      IF(N2-NOMA(1,I))112,127,112
  112 CONTINUE
      NMA=NMA+1
      NOMA(1,NMA)=N2
      N=0
      DO 113 J=K,K2
      N=N+1
  113 NOEM(N,NMA)=MAIN(K2+1-J)
      I=N-1
      DO 117 KK=1,I
      IF(NOEM(KK,NMA))114,117,114
  114 N1=KK+1
      DO 117 J=N1,N
      IF(NOEM(KK,NMA)-NOEM(J,NMA))117,115,117
   15 DO 116 N3=N1,J
  116 NOEM(N3,NMA)=0
  117 CONTINUE
      N4=0
      DO 119 KK=1,N
      IF(NOEM(KK,NMA))118,119,118
  118 N4=N4+1
      NOEM(N4,NMA)=NOEM(KK,NMA)
  119 CONTINUE
      NIM(NMA)=N4-1
      NOMA(2,NMA)=NREF
  120 DO 1270 N=2,N4
      N1=NOEM(N-1,NMA)
      N2=NOEM(N,NMA)
      IF(N1-N2)121,121,124
  121 N3=KAD(N1)
      N5=KAD(N1+1)-1
      DO 122 J=N3,N5
      IF(NOEB(2,J)-N2)122,123,122
  122 CONTINUE
  123 NOEM(N-1,NMA)=J
      GO TO 1270
  124 N3=KAD(N2)
      N5=KAD(N2+1)-1
      DO 125 J=N3,N5
      IF(NOEB(2,J)-N1)125,126,125
  125 CONTINUE
  126 NOEM(N-1,NMA)=-J
 1270 CONTINUE
  127 CONTINUE
  130 RETURN
      END
```

```
[illegible]

      DO 150 I=1,NFOM
C          *  ... DE LA J EME CONSIGNE  *
      CALL ...
      IF(NT(I).EQ.NT(I))CALL MOMUD(I)
      STO=...
      DO 111 II=1,NFOM
   11 ...
      DO 120 L=1,NCO
      IF(ABS( ... STO)-COEMUL-EPS)130,130,120
  120 CONTINUE
      NCOM=NCO+1
      IACOM(NCO)=STO
      COUPE(NCO)=COUPE( ... )
      IF(NMAX)125,125,150
  125 DO 127 II=1,NFOM
  127 NTAC(II,NCO)=NTAC(II, ...)
      GO TO 150
C
  130 A=COUPE( ... )
      IF(COUPE(L)/(TACOM(L)+1.E-10)-A/STO)150,150,135
  135 IF(NMAX)137,137,145
  137 DO 140 II=1,NFOM
  140 NTAC(II,L)=NTAC(II, ...)
  145 COUPE(L)=A
      IACOM(L)=STO
C
  150 CONTINUE
C
C          *CALCUL DE ... PAR METH. DES MOINDRES CARRES  *
C
      CALL SOMME(TACOM,TACOM,1,1,NCO,A0)
      CALL SOMME(TACOM,TACOM,2,1,NCO,A1)
      CALL SOMME(TACOM,TACOM,2,2,NCO,A2)
      CALL SOMME(TACOM,COUPE,1,1,NCO,B0)
      CALL SOMME(TACOM,COUPE,2,1,NCO,B1)
      ...
C
C          ... PERMET ... DETERMINATION DES
C              ... MAXIMAUX
C
      ...
      ...
      DO 500 ...
  500 ...
      RETURN
      END
```

```
      SUBROUTINE SOMME(A,B,N,M,...)
      DIMENSION A(1),B(1)
      X=0.
      DO 10 I=1,N
   10 X=X+(A(I)**N)*(B(I)**M)
      RETURN
      END
```

```
      SUBROUTINE CONRE(KID)
      COMMON /NLOC/NNO,NBR,NPOM,NRES,NMA,K2,NTOT,NIT,NREF,NCO,NC1,
     1KN,NKT,NPE,NFT,KP(30),NTM(20),KOD(30),MAIN(60),MAINB(60),KL(4),
     2KP1(30),KP1(6),KPOI(6),NPNO(6),LXI(5),KAU(30),NOMA(2,20),
     3NOEM(30,20),NTAC(6,200),ICOOP(375),NGEB(2,50),JCOOP(25)
      COMMON /RLOC/PMIN,PMAX,QTOT,PAES,XTOT,COEMUL,XMIF,XMAF,
     1Q(50),PRE(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COFI(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),TACOM(30),CONTOT,COUTO
C           *  VERIFICATION DE LA NON-VIOLATION DES CONTRAINTES *
      KID=0
C            * PRESSIONS AUX NOEUDS  *
      DO 10 L=1,NNO
      P=PRE(L)-ALT(L)
      CALL TVCP(P,PMIN,PMAX,KID)
      IF(KID)50,10,50
   10 CONTINUE
C           * HAUT  D'EAU DANS LE RES. DE REF.  *
      P=PRE(NREF)-S(NREF)/SEC(NRES)
      CALL TVCP(P,XMIN(NRES),XMAX(NRES),KID)
      IF(KID)50,12,50
   12 LL=NRES-1
      IF(LL)50,50,15
C           * LES AUTRES RES  SE VIDENT-ILS ?  *
   15 DO 40 L=1,LL
      N=KL(L)
      P=PRE(N)-S(N)/SEC(L)
      CALL TVCP(P,XMIN(L),XMAX(L),KID)
      IF(KID-1)40,50,30
   30 KID=0
   40 CONTINUE
C
   50 RETURN
      END
```

```
      SUBROUTINE ETAT(X,XK)
      COMMON /NLOC/NNO,NBR,NPOM,NRES,NMA,K2,NTOT,NIT,NREF,NCO,NC1,
     1KN,NKT,NPE,NFT,KP(30),NTM(20),KOD(30),MAIN(60),MAINB(60),KL(4),
     2KP1(30),KP1(6),KPOI(6),NPNO(6),LXI(5),KAU(30),NOMA(2,20),
     3NOEM(30,20),NTAC(6,200),ICOOP(375),NGEB(2,50),JCOOP(25)
      COMMON /RLOC/PMIN,PMAX,QTOT,PAES,XTOT,COEMUL,XMIF,XMAF,
     1Q(50),PRE(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COFI(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),TACOM(30),CONTOT,COUTO
      X=XK-COEMUL*S(NREF)
C
      LL=NRES-1
      IF(LL)30,30,2
    2 DO 20 L=1,LL
      N=KL(L)
      P=PRE(N)-S(N)/SEC(L)
      IF(P-XMAX(L))10,5,5
    5 IF(KOD(N))20,20,8
    8 X=X+SEC(L)*(XMAX(L)-PRE(N))*COEMUL
      GO TO 20
   10 IF(KOD(N))15,15,17
   15 X=X+COEMUL*(P-XMAX(L))*SEC(L)
      GO TO 20
   17 X=X-COEMUL*S(N)
   20 CONTINUE
   30 RETURN
      END
```

```
      FUNCTION COURT(J,K)
      COMMON /RLOC/NNO,NRR,NPOM,NRFS,NMA,K2,NTOT,NTT,NREF,NCO,NC1,
     1KN,NKT,NFE,NFT,KP(30),NTM(20),KOD(30),MAIN(60),MAINR(60),KL(4),
     2KPT(30),KPT(6),KPOT(6),NENO(6),IXT(5),KAD(30),NOMA(2,20),
     3NOEM(30,20),NIAC(6,200),ICOUP(375),NOER(2,50),JCONP(25)
      COMMON /RLOC/PMIN,PMAX,DTOT,PAFS,XTOT,COEMUL,XMIF,XMAF,
     1Q(50),PRE(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COFT(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),IACOM(30),CONTOT,COUTO
      IF(KN-K)5,50,5
    5 KN=K
C            * INITIALISATION DU RESEAU POUR UN NOUVEAU PAS DE TEMPS *
C
      DO 6 L=1,NRR
    6 Q(L)=0
      DO 7 L=1,NNO
    7 S(L)=0
   35 DF=(COMU(K)+1.)/(1.+COMU(NKT))
      COMU(NKT)=COMU(K)
   37 DO 40 L=1,NNO
      CONS(L)=DF*CONS(L)
      S(L)=S(L)-CONS(L)
   40 CALL REINR(L,CONS(L))
C            * CHANGE DES CARACT DES POMPES POUR LA CONSIGNE J *
   50 CALL COEFO(J)
C            * RES PAR HARDY-CROSS DES EQUA HYD *
      CALL HFHFR
      DO 60 L=1,NPOM
      N=KPT(L)
      IF(KOD(N))60,60,55
   55 IF(S(N))70,60,60
   60 CONTINUE
      RETURN
   70 NTT=JTOT
      RETURN
      END
```

```
      SUBROUTINE RESOL(K,J)
      COMMON /RLOC/NNO,NRR,NPOM,NRFS,NMA,K2,NTOT,NTT,NREF,NCO,NC1,
     1KN,NKT,NFE,NFT,KP(30),NTM(20),KOD(30),MAIN(60),MAINR(60),KL(4),
     2KPT(30),KPT(6),KPOT(6),NENO(6),IXT(5),KAD(30),NOMA(2,20),
     3NOEM(30,20),NIAC(6,200),ICOUP(375),NOER(2,50),JCONP(25)
      COMMON /RLOC/PMIN,PMAX,DTOT,PAFS,XTOT,COEMUL,XMIF,XMAF,
     1Q(50),PRE(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COFT(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),IACOM(30),CONTOT,COUTO
      IF(KN-K)5,50,5
    5 KN=K
C            * INITIALISATION DU RESEAU POUR UN NOUVEAU PAS DE TEMPS *
C
      DO 6 L=1,NRR
    6 Q(L)=0
      DO 7 L=1,NNO
    7 S(L)=0
   35 DF=(COMU(K)+1.)/(1.+COMU(NKT))
      COMU(NKT)=COMU(K)
   37 DO 40 L=1,NNO
      CONS(L)=DF*CONS(L)
      S(L)=S(L)-CONS(L)
   40 CALL REINR(L,CONS(L))
C            * CHANGE DES CARACT DES POMPES POUR LA CONSIGNE J *
   50 CALL COEFO(J)
C            * RES PAR HARDY-CROSS DES EQUA HYD *
      CALL HFHFR
      DO 60 L=1,NPOM
      N=KPT(L)
      IF(KOD(N))60,60,55
   55 IF(S(N))70,60,60
   60 CONTINUE
      RETURN
   70 NTT=JTOT
      RETURN
      END
```

```
      SUBROUTINE COEPO(J)
      COMMON /NLOC/NNO,NBR,NPOM,NRES,NMA,K2,NTOT,NIT,NREF,NCO,NC1,
     1KN,NKT,NEF,NEI,KP(30),NTM(20),KOD(30),MAIN(60),MAINB(60),KL(4),
     2KP1(30),KPT(6),KPOT(6),NPNO(6),EXT(5),KAD(30),NOMA(2,20),
     3NOEM(30,20),NTAC(6,200),ICOOP(375),NOEB(2,50),JCOOP(25)
      COMMON /RLOC/PMIN,PMAX,QTOT,PAFS,XTOT,COEMUL,XMIF,XMAF,
     1Q(50),PRE(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COEI(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),TACOM(30),CONTOT,COUTO
      DEL=0.1
      DO 30 L=1,NPOM
      INK=KPT(L)
      KK=NTAC(L,J)
      IF(KK)20,20,10
   10 KKP1=KK+KPOT(L)-1
      IF(S(INK)-0.01)12,12,15
   12 CALL REINR(INK,DEL)
   15 KOD(INK)=2
C           *UN GROUPE REFOULE A L'USINE L POUR LA CONSIGNE J *
      COE(L,1)=CAPO(1,KKP1)+ALT(INK)
      COE(L,2)=CAPO(2,KKP1)
      COE(L,3)=CAPO(3,KKP1)
      GO TO 30
   20 KOD(INK)=0
C     *  AUCUN GROUPE NE REFOULE A L'USINE L POUR LA CONSIGNE J  *
      DEF=-S(INK)
      CALL REINR(INK,DEF)
   30 CONTINUE
      RETURN
      END

      SUBROUTINE REINR(NIN,DELQ)
      COMMON /NLOC/NNO,NBR,NPOM,NRES,NMA,K2,NTOT,NIT,NREF,NCO,NC1,
     1KN,NPT,NPF,NEI,KP(30),NTM(20),KOD(30),MAIN(60),MAINB(60),KL(4),
     2KP1(30),KPT(6),KPOT(6),NPNO(6),EXT(5),KAD(30),NOMA(2,20),
     3NOEM(30,20),NTAC(6,200),ICOOP(375),NOEB(2,50),JCOOP(25)
      COMMON /RLOC/PMIN,PMAX,QTOT,PAFS,XTOT,COEMUL,XMIF,XMAF,
     1Q(50),PRE(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COEI(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),TACOM(30),CONTOT,COUTO
C
C       *  DELQ: MODIF  APPORTEE AU BILAN DE DEBIT DU NOEUD NIN  *
      IF(NREF-NIN)5,30,5
    5 DO 10 I=2,K2
      N=MAIN(I)
      N3=MAINB(I-1)
      N4=IABS(N3)
      Q(N4)=Q(N4)-DELQ*(N3/N4)
      IF(N-NIN)10,20,10
   10 CONTINUE
   20 S(NREF)=S(NREF)-DELQ
      S(NIN)=S(NIN)+DELQ
   30 RETURN
      END
```

```
      SUBROUTINE PECIR(A,NN,NR,ID)
      DIMENSION A(1)
      IF(NR)30,30,2
    2 N=NN-1
      DO 20 I=1,NR
      IF(ID)5,30,7
    5 B=A(NN)
      J=NN
      GO TO 8
    7 B=A(1)
      J=1
    8 DO 10 L=1,N
      A(J)=A(J+ID)
   10 J=J+ID
   20 A(J)=B
   30 RETURN
            END
```

```
      SUBROUTINE SIMUL
      COMMON /NLOC/NNO,NBR,NPOM,NRES,NMA,K2,NTOT,NIT,NREF,NCO,NC1,
     1KN,NKT,NPE,NFT,KP(30),NTM(20),KOD(30),MAIN(60),MAINB(60),KL(4),
     2KP1(30),KP1(6),KPOT(6),NPNO(6),LXI(5),KAU(30),NOMA(2,20),
     3NOEM(30,20),NTAC(6,200),ICOOP(375),NOEB(2,50),JCOUP(25)
      COMMON /RLOC/PMIN,PMAX,QTOT,PAES,XTOT,COEMUL,XMIF,XMAF,
     1Q(50),PRE(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COFT(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),TACUM(30),CONTOT,COUTO
      XK=0.
C           *   INITIALISATION *
      DO 10 L=1,NRES
      N=KL(L)
      IF(PRE(N)-XMAX(L))5,5,3
    3 XK=XK+SEC(L)*(XMAX(L)-XMIN(L))
      GO TO 10
    5 XK=XK+SEC(L)*(PRE(N)-XMIN(L))
   10 KOD(N)=1
      XK=XK*COEMUL
      NK=NKT-1
      COUTO=0.
      DO 30 K=1,NK
      JCOOP(K)=0
      COUT=1.E33
C           *  ESSAI DES NC1 CONSIGNES  *
      DO 20 J=1,NC1
      CALL RESOL(K,J)
      IF(NIT.EQ.NTOT)CALL REMOD(K)
      CALL CONRE(KID)
      IF(KID)20,12,20
   12 CALL ETAT(X,XK)
      CALL CREA(REV,X,PAES,NPE,NKT,K,REVMO,ICOOP)
      COUT1=COURE(J,K)+REVMO
      IF(COUT-COUT1)20,20,15
   15 JCOOP(K)=J
      COUT=COUT1
   20 CONTINUE
      CALL ECRIO(K)
      IF(JCOOP(K))40,40,30
C           *  MISE A JOUR DES DONNEES POUR LE PAS (K+1)  *
   30 CALL RESK1(K,XK)
   40 RETURN
      END
```

```
      SUBROUTINE REMOD(K)
      COMMON /NLOC/NNO,NBR,NPOM,NRES,NMA,K2,NTOT,NIT,NREF,NCO,NC1,
     1KN,NKT,NPE,NET,KP(30),NIM(20),KOD(30),MAIN(60),MAINR(60),KL(4),
     2KPT(30),KPT(6),KPOT(6),NPNO(6),LXT(5),KAD(30),NOMA(2,20),
     3NOEM(30,20),NTAC(6,200),ICOOP(375),NOFB(2,50),JCOUF(25)
      COMMON /RLOC/PMIN,PMAX,QTOT,PAES,XTOT,COEMUL,XMIF,XMAF,
     1Q(50),PRE(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COEJ(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUNG(75),TACOM(30),CONTOT,COUTO

      EPS=0.2
      NIM=0
    4 DO 6 L=1,NBR
    6 Q(L)=0.
      DO 7 L=1,NNO
    7 S(L)=0.
      DO 40 L=1,NNO
      S(L)=S(L)-CONS(L)
   40 CALL REINR(L,CONS(L))
      DO 50 L=1,NPOM
      N=KPT(L)
      IF(KOD(N))50,50,45
   45 DEL=NIM*EPS
      CALL REINR(N,DEL)
   50 CONTINUE
      CALL DEDEB
   62 DO 65 L=1,NPOM
      N=KPT(L)
      IF(KOD(N))65,65,61
   61 IF(S(N))70,63,63
   63 P=COE(L,1)+(COE(L,2)+COE(L,3)*S(N))*S(N)
      IF(ABS(P-PRE(N))-0.75)65,65,70
   65 CONTINUE
      RETURN
   70 NIM=NIM+1
      IF(NIM-4)4,4,80

   80 CALL EXIT
      END
```

```
      SUBROUTINE RESK1(K,XK)
      COMMON /NLIC/NND,NBR,NPOM,NRFS,NMA,K2,NTOT,NTT,NREF,NCO,NC1,
     1KN,NKT,NPT,NFT,NF(30),NTM(20),KOU(30),MAIN(60),MAINR(60),KL(4),
     2KP1(30),KPT(6),KPOI(6),NPND(6),LXI(5),KAU(30),NOMA(2,20),
     3NUFD(30,20),NTAC(6,200),ICOOP(375),NOSR(2,50),JCOUP(25)
      COMMON /RLOC/PMIN,PMAX,UTOT,PAFS,XTOT,COEMUL,XMIF,XMAF,
     1Q(50),PRF(30),COMS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COFT(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(25),TACOM(30),CONTOT,COUTO
      COMMON/RLOC1/ALPHA,TRAAG(25),PRET(4),COUPR(30),COCAG(2),NMAX
C
C           * DETER. DE LA REPARTITION DES DEBITS DANS LE RESEAU
C             POUR LA CONSIGNE  ICOOP(K)    *
      I=JCOOP(K)
      CALL  RESOL(K,I)
      IF(NTT.EQ.NTOT)CALL REMOD(K)
      CALL  ETAT(XK,XK)
      PRE(NREF)=PRE(NREF)-S(NREF)/SEC(NRFS)
      LL=NRFS-1
      IF(LL)12,12,7
    7 DO 10 L=1,LL
      N=KP(L)
      PRF(N)=PRE(N)-S(N)/SEC(L)
      CALL  TVCP(PRF(N),XMIN(L),XMAX(L),KTO)
      IF(KTO-1)8,8,5
    5 KOU(N)=0
      DEL=-S(N)
      CALL  REINR(N,DEL)
      GO TO 10
    8 KOU(N)=1
   10 CONTINUE
C           *VOLUME  TOTAL DANS LES RES. A L'INSTANT  (K+1)   *
   12 TRAAG(K+1)=XK
      COU=COURE(J,K)
C           *  DTER.' DE COUAG(K)    *
      COUTO=COUTO+COU
      TT=0
      STO=0.
      DO 40 L=1,NPOM
      N=KPT(L)
      IF(KOU(N))40,40,35
   35 STO=S(N)+STO
      TT=TT+1
   40 CONTINUE
      IF(TT)50,50,45
   45 COUAG(K)=COCAG(2)*STO+COCAG(1)+(COU-COURF(J,10))/(STO*COEMUL)
      RETURN
   50 A=0.
      B=0
      DO 60 L=1,NPOM
      A=A+C1(L-1)*NKT+K)
      B=B+C1(L-1)*NKT+10)
   60 COUAG(K)=COCAG(1)*A/B
      RETURN
      END
```

```
      SUBROUTINE DEDER
      COMMON /NLOC/NNO,NBR,NPOM,NPES,NMA,K2,NTOT,NTT,NREF,NCO,NC1,
     1KN,NPT,NPE,NFI,KP(30),NTM(20),KOD(30),MAIN(60),MAINB(60),KL(4),
     2KP1(30),KPT(6),KPOT(6),NPNO(6),LXT(5),KAD(30),NOMA(2,20),
     3NOEM(30,20),NTAC(6,200),ICOOP(375),NOEB(2,50),JCOOP(25)
      COMMON /RLOC/PMIN,PMAX,QTOT,PAES,XTOT,COEMUL,XMTF,XMAF,
     1Q(50),PRE(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COFT(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),TACUM(30),CONTOT,COUTO
      COMMON/RLOC1/ALPHA,TRAAG(25),PRET(4),COUPR(30),COCAG(2),NMAX
      DO 1 L=1,NPOM
      N=KP1(L)
    1 PRE(N)=COE(L,1)+COE(L,2)*S(N)+COE(L,3)*S(N)**2
      LL=NRES-1
      IF(LL)9,9,5
    5 DO 7 L=1,LL
      N=KL(L)
      IF(KOD(N))7,7,6
    6 PRET(L)=PRE(N)

    7 CONTINUE
    9 NTT=0
      TES=0.
      IF(NMA)200,200,10
   10 IF(NTOT-NTT)200,30,30
C
   30 TES=0.
      DO 100 T=1,NMA
      N1=NOMA(1,T)
      N2=NOMA(2,T)
      IF(N1-N2)32,35,32
   32 IF(KOD(N1)*KOD(N2))100,100,35
   35 NO=NTM(T)
      CALL EQMA2(X,T,NO)
      IF(TES-ABS(X))40,40,50
   40 TES=ABS(X)
   50 IF(N1-N2)55,90,55
   55 S(N1)=S(N1)+X
      S(N2)=S(N2)-X
      IF(KOD(N1)-1)70,70,60
   60 N=KP(N1)
      PRE(N1)=COE(N,1)+COE(N,2)*S(N1)+COE(N,3)*S(N1)**2
   70 IF(KOD(N2)-1)90,90,80
   80 N=KP(N2)
      PRE(N2)=COE(N,1)+COE(N,2)*S(N2)+COE(N,3)*S(N2)**2
   90 DO 100 K=1,NO
      N3=NOEM(K,T)
      N4=IABS(N3)
      Q(N4)=Q(N4)+X*(N3/N4)
  100 CONTINUE
      NTT=NTT+1
C
      IF(TES-QTOT)200,200,10
C
  200 DO 210 I=2,K2
      N1=MAIN(I-1)
      N2=MAIN(I)
      N3=MAINB(I-1)
      N4=IABS(N3)
      CO=Q(N4)
  210 PRE(N2)=PRE(N1)-R(N4)*CO*ABS(CO)*(N3/N4)
      IF(LL)230,230,212
  212 DO 220 L=1,LL
      N=KL(L)
      IF(KOD(N))220,220,215
  215 PRE(N)=PRET(L)
  220 CONTINUE
  230 RETURN
      END
```

```
      SUBROUTINE FDMA2(X,I,NO)
      COMMON /NLOC/NNO,NPR,NPOM,NRFS,NMA,K2,NTUT,NIT,NREF,NCO,NC1,
     1KN,NP1,NPE,NFI,KP(30),NTM(30),KOD(30),MAIN(60),MAIN8(60),KL(4),
     2KP1(30),KPT(6),KPO1(6),NPNO(6),LXT(5),KAD(30),NUMA(2,20),
     3NOFM(30,20),NTAC(6,200),ICOOP(375),NOE8(2,50),JCOOP(25)
      COMMON /RLOC/PMIN,PMAX,QTOT,PAES,XTOT,COEMUI,XMIF,XMAF,
     1Q(50),PRE(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COFT(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEL(4),COUAG(75),TACOM(30),CONTOT,COUTO

      N1=NOMA(1,I)
      N2=NOMA(2,I)
      X=0.
      Y=0.
      IF(KOD(N1)-1)2,2,1
    1 N=KP(N1)
      Y=-(S(N1)*COE(N,3)+COE(N,2)/2.)
    2 IF(KOD(N2)-1)4,4,3
    3 N=KP(N2)
      Y=Y+S(N2)*COE(N,3)+COE(N,2)/2.
    4 X=PRE(N1)-PRE(N2)
C
      DO 10 K=1,NO
      N3=NOFM(K,I)
      N4=IABS(N3)
      S0=Q(N4)
      S1=R(N4)*ABS(S0)
      Y=Y+S1
   10 X=X-S1*S0*(N3/N4)
      X=X/Y/2.
      RETURN
      END
```

```
      SUBROUTINE CREA(REV,XKP1,PAES,NPE,NKT,K,REVMO,ICOOP)
      DIMENSION REV(NKT,1),ICOOP(NKT,1)
      IF(XKP1) 10,10,20
   10 IF(ICOOP(K+1,2))12,12,15
   12 REVMO=REV(K+1,1)
      RETURN
   15 REVMO=((REV(K+1,2)-REV(K+1,1))/PAES)*XKP1+REV(K+1,1)
      RETURN
   20 IF(XKP1-(NPE-1)*PAES)40,40,30
   30 IF(ICOOP(K+1,NPE-1))32,32,35
   32 REVMO=REV(K+1,NPE)
      RETURN
   35 REVMO=((REV(K+1,NPE)-REV(K+1,NPE-1))/PAES)
     1*(XKP1-(NPE-1)*PAES)+REV(K+1,NPE)
      RETURN
   40 I=IFIX(XKP1/PAES)+1
   60 REVMO=((REV(K+1,I+1)-REV(K+1,I))/PAES)*(XKP1-(I-1)*PAES)
     1+REV(K+1,I)
      RETURN
      END
```

```
      SUBROUTINE PRODYN(NKT,NPE,NCO,IDC,NEI,LXI,COFI,REV,ICOOP,PAES)
      DIMENSION LXI(1),COFI(1),REV(NKT,1),ICOOP(NKT,1)
      NPAT=-1
      DO 7 I=1,NPE
      ICOOP(NKT,I)=0
      ICOOP(1,I)=0
    7 REV(NKT,I)=COFI(I)
      IF(NKT-2) 60,40,10
   10 K=NKT
      NPAR=-1
      I=1
      DO 30 IK=3,NKT
      I=I+NPAR
      K=K-1
      NPAR=-NPAR
      DO 30 II=1,NPE
      I=I+NPAR
      NPAT=-NPAT
      CALL RESTA(NCO,IDC,NPE,NKT,PAES,REV,ICOOP,I,K,1,NPAT)
   30 CONTINUE
   40 K=1
      DO 50 L=1,NEI
      I=LXI(L)
      NPAT=-NPAT
   50 CALL RESTA(NCO,IDC,NPE,NKT,PAES,REV,ICOOP,I,K,1,NPAT)
   60 RETURN
      END
```

```
      SUBROUTINE RESTA(NCO,IDC,NPE,NKT,PAES,REV,ICOOP,I,K,NS,NA)
      DIMENSION REV(NKT,1),ICOOP(NKT,1)
      REV(K,I)=1.E+35
      ICOOP(K,I)=0
      IF(NA)02,02,04
   02 J=NCO+1
      GO TO 07
   04 J=0
C
   07 DO 30 L=1,NCO
      J=J+NA
      CALL CONTR(L,J,K,IDC,KD,PAES,NKT,NPE,NS)
      IF(KD) 30,10,30
   10  IF(NS)15,15,12
   12 CALL FST(I,J,K,PAES,XKP1)
      CALL CRFA(REV,XKP1,PAES,NPE,NKT,K,REVMO,ICOOP)
      GO TO 18
   15 REVMO=0.
   18 REVP=REVMO+CDCO(I,J,K)
      IF(REVP-REV(K,I)) 20,30,30
   20 REV(K,I)=REVP
      ICOOP(K,I)=J
   30 CONTINUE


      RETURN
      END
```

```
      SUBROUTINE CONTR(I,J,K,TDC,TDV,PAEF,NKT,NPP,NSNT)
      COMMON /NLOC/NNO,NBR,NFOM,NRES,NMA,K2,NTOT,NTT,NREF,NCO,NC1,
     1KN,NKT,NPE,NEI,KP(30),NTM(20),KOD(30),MATN(60),MATNR(60),KL(4),
     2KP1(30),KPI(6),KPOT(6),NPNO(6),LXI(5),KAO(30),NOMA(2,20),
     3NOEM(30,20),NTAC(4,200),TCOOP(375),NOEB(2,50),JLOOP(25)
      COMMON /RLOC/PMIN,PMAX,RTOT,PAES,XTOT,COEMUL,XMIF,XMAF,
     1Q(50),PRE(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COFI(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),IACOM(30),CONTOT,COUTO
      TDV=0
      X=(T-1)*PAES+IACOM(J)+CONTOT*(1.+COMU(K))
      IF(K+1-NKT)10,20,20
   10 CALL TVCP(X,0.,XTOT,TDV)
      IF(TDV)12,12,15
   12 CALL PRCOM(ICOOP,NPE,PAES,K,TDV,X,NKT)
   15 RETURN
   20 CALL TVCP(X,XMIF,XMAF,TDV)
      RETURN
      END
```

```
      FUNCTION COCO(I,J,K)
      COMMON /NLOC/NNO,NBR,NFOM,NRES,NMA,K2,NTOT,NTT,NREF,NCO,NC1,
     1KN,NKT,NPE,NEI,KP(30),NTM(20),KOD(30),MATN(60),MATNR(60),KL(4),
     2KP1(30),KPI(6),KPOT(6),NPNO(6),LXI(5),KAO(30),NOMA(2,20),
     3NOEM(30,20),NTAC(4,200),TCOOP(375),NOEB(2,50),JLOOP(25)
      COMMON /RLOC/PMIN,PMAX,RTOT,PAES,XTOT,COEMUL,XMIF,XMAF,
     1Q(50),PRE(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COFI(15),CAPO(3,150),REPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SEC(4),COUAG(75),IACOM(30),CONTOT,COUTO
      COMMON/RLOC1/ALPHA,TRAAG(25),PRET(4),COUPR(30),COCAG(2),NMAX
      A=(TRAAG(K)-(I-1)*PAES)/COEMUL
      COCO=ALPHA*A*A+IACOM(J)*COUAG(K)
      RETURN
      END
```

```
      SUBROUTINE PRCOM(ICOOP,NPE,PAES,K,TDV,XKP1,NKT)
      DIMENSION ICOOP(NKT,1)
      I=(FIX(XKP1/PAES)+1
      TDV=0
      IF(XKP1)5,5,20
    5 IF(ICOOP(K+1,1))7,7,10
    7 TDV=3
   10 RETURN
   20 XMAX=(NPE-1)*PAES
      IF(XKP1-XMAX)30,40,60
   30 IF(ICOOP(K+1,I))7,7,10
   40 IF(ICOOP(K+1,I+1))7,7,50
   50 IF(ICOOP(K+1,I+1))7,7,10
      END
```

```
      SUBROUTINE ECRIT
      COMMON /NLOC/NNO,NBR,NPOM,NRES,NMA,K2,NTOT,NTT,NREF,NCO,NC1,
     1LN,NKT,NPE,NEI,NP(30),NTM(20),KOD(30),MAIN(60),MAINB(60),KL(4),
     2NPI(30),KPI(6),KPOI(6),NPNO(6),LXI(5),KAD(30),NOMA(2,20),
     3NOEM(30,20),NTAC(6,200),ICOOP(375),NOEB(2,50),JCOOP(25)
      COMMON /RLOC/PMIN,PMAX,QTOT,PAFS,XTOT,COFMUL,XMTF,XMAF,
     1Q(50),PRF(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COFT(15),CAPO(3,150),RFPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SFC(4),COUAG(75),TACOM(30),CONTOT,COUTO
      COMMON/RLOC1/ALPHA,TRAAG(25),PRET(4),COUPR(30),COCAG(2),NMAX
      LWR=2
      WRITE(LWR,1000)
 1000 FORMAT(10X,"COUT AGREGE ET TRA  AGREGEE")

      NN=NKT-1
      L=NPE
  210 DO 5 L=1,NN
    5 WRITE(LWR,2000)COUAG(L),TRAAG(L)
 2000 FORMAT(5(2X,F10.4))
      WRITE(LWR,2500)
 2500 FORMAT(10X,"COM  OPT  AGRE.")
      DO 30 LL=1,NPE
      L=NPE-LL+1
   30 WRITE(LWR,3000)(ICOOP((L-1)*NKT+K),K=1,NKT)
 3000 FORMAT(2X,25I2)
      WRITE(LWR,3500)
 3500 FORMAT(10X,"COM. OPT.")
      WRITE(LWR,3000)(JCOOP(L),L=1,NN)
      WRITE(LWR,4000)COUTO
 4000 FORMAT(5X,"COUT REEL",5X,F10.4)
      RETURN
      END
```

```
      SUBROUTINE ECRTO(K)
      COMMON /NLOC/NNO,NBR,NPOM,NRES,NMA,K2,NTOT,NTT,NREF,NCO,NC1,
     1LN,NKT,NPE,NEI,NP(30),NTM(20),KOD(30),MAIN(60),MAINB(60),KL(4),
     2KPI(30),KPI(6),KPOI(6),NPNO(6),LXI(5),KAD(30),NOMA(2,20),
     3NOEM(30,20),NTAC(6,200),ICOOP(375),NOEB(2,50),JCOOP(25)
      COMMON /RLOC/PMIN,PMAX,QTOT,PAFS,XTOT,COFMUL,XMTF,XMAF,
     1Q(50),PRF(30),CONS(30),S(30),R(50),ALT(30),C1(150),COMU(25),
     2C2(6),COFT(15),CAPO(3,150),RFPO(3,150),REV(375),COE(6,3)
     3,XMIN(4),XMAX(4),SFC(4),COUAG(75),TACOM(30),CONTOT,COUTO
      LWR=2
      WRITE(LWR,1000)K,ICOOP(K)
 1000 FORMAT(5X,"K=",I3,5X,"ICOOP #",I3)
      WRITE(LWR,2000)(PRE(KI(L)),L=1,NRES)
 2000 FORMAT(5X,4(2X,F10.4))
      RETURN
      END
```

```
 1  NPMA
 2      7
 3  NNO,NMA,..
 4     23     9    31     5     2    99    25    16
 5     11    .1              +3.600E+03
 6  BRANCHES
 7      1     7              +0.700E+02
 8      1    21              +0.580E+02
 9      2     3              +0.540E+02
10      2    21              +0.116E+03
11      3     4              +0.340E+03
12      3     6              +0.345E+02
13      3     7              +0.350E+03
14      4    23              +6.375E+03
15      5     6              +0.480E+03
16      5    23              +0.226E+03
17      6     8              +0.900E+01
18      6    10              +0.630E+01
19      6    23              +9.000E+03
20      7     8              +0.185E+02
21      8    11              +0.134E+03
22      8    16              +0.632E+02
23      9    10              +0.340E+03
24      9    12              +0.875E+02
25     10    11              +0.134E+03
26     10    13              +0.120E+03
27     11    15              +0.665E+04
28     12    13              +0.124E+03
29     13    14              +0.238E+03
30     13    15              +0.400E+03
31     14    19              +0.280E+03
32     14    20              +0.142E+03
33     15    16              +0.260E+03
34     16    17              +0.170E+03
35     16    18              +0.138E+03
36     18    19              +0.100E+03
37     18    22              +0.276E+04
38  MAILLES
39      5     1    21     2     3     7
40      4     3     4    23     6
41      3     6    23     5
42      4     7     3     6     8
43      4     8     6    10    11
44      4     6    11    15    16
45      4    11    10    13    15
46      6    10     9    12    13    15    11
47      6    15    13    14    19    18    16
48  CONS. ET ALI
49    1          -8.980E-02+0.370E+02
50    2          -8.170E-02+0.560E+02
51    3           1.230E-02+0.580E+02
52    4          -2.700E-02+0.650E+02
53    5           4.220E-02+0.470E+02
54    6           3.400E-02+0.480E+02
55    7           9.600E-02+0.680E+02
```

```
    8          -1.220E-01+0.320E+02
    9           1.790E-03+0.830E+02
   10          -3.190E-02+0.900E+02
   11           4.280E-02+0.910E+02
   12           7.030E-03+0.900E+02
   13           5.060E-02+0.300E+02
   14          -2.210E-02+0.470E+02
   15          -4.210E-02+0.810E+02
   16          -7.470E-02+0.750E+02
   17           7.840E-02+0.280E+02
   18           2.960E-02+0.840E+02
   19           7.160E-03+0.820E+02
   20           1.550E-01+0.250E+02
   21           0.000E-00+0.720E+02
   22          -0.000E+00+0.550E+02
   23          -4.220E-02+0.900E+02
POMPES
     8     5
    1   +1.944E-01+0.107E+030.740+2.500E-01+0.990E+020.800
        +3.333E-01+0.850E+020.850
    2   +1.944E-01+0.110E+030.740+2.500E-01+0.106E+030.800
        +3.333E-01+0.910E+020.850
    3   +4.167E-01+0.107E+030.750+5.278E-01+0.990E+020.810
        +6.944E-01+0.850E+020.845
    4   +3.888E-01+0.107E+030.740+5.000E-01+0.990E+020.800
        +6.666E-01+0.850E+020.850
    5   +6.111E-01+0.107E+030.745+7.778E-01+0.990E+020.800
        +1.028E-00+0.850E+020.850
   20     3
    1   +0.500E-01+1.440E+020.800+0.600E-01+1.220E+020.770
        +0.700E-01+0.950E+020.660
    2   +0.100E-00+1.440E+020.800+1.200E-01+1.220E+020.770
        +1.400E-01+0.950E+020.660
    3   +0.044E-00+1.200E+020.760+5.000E-02+1.100E+020.770
        +5.556E-02+1.000E+020.770
   14     5
    1   +0.100E+00+0.970E+020.770+0.120E-00+0.910E+020.820
        +0.130E+00+0.870E+020.820
    2   +0.900E-01+0.900E+020.810+0.100E+00+0.860E+020.830
        +0.110E-00+0.780E+020.820
    3   +1.600E-01+0.970E+020.780+0.177E-00+0.910E+020.810
        +2.190E-01+0.870E+020.800
    4   +0.210E-00+0.900E+020.810+0.230E-00+0.860E+020.820
        +0.245E-00+0.800E+020.800
    5   +0.273E-00+0.900E+020.810+0.295E-00+0.850E+020.820
        +0.310E-00+0.800E+020.810
   13     3
    1   +0.500E-01+1.160E+020.570+0.900E-01+1.120E+020.780
        +1.200E-01+1.010E+020.820
    2   +0.100E-00+1.030E+020.640+0.150E-00+1.010E+020.780
        +0.200E-00+0.950E+020.780
    3   +0.180E-00+1.040E+020.720+0.245E-00+1.010E+020.790
        +0.310E-00+0.950E+020.790
   17     3
    1   +8.333E-02+1.080E+020.780+1.111E-01+1.000E+020.780
        +1.399E-01+0.830E+020.550
```

```
112     2   +1.444E-01+1.080E+020.780+2.222E-01+1.000E+020.780
113         +2.778E-01+0.830E+020.550
114     3   +2.440E-01+1.080E+020.780+3.333E-01+1.000E+020.780
115         +4.167E-01+0.830E+020.550
116      30
117       0     0     0     0     0
118       0     0     0     0     1
119       0     0     0     0     3
120       0     0     2     0     0
121       0     1     0     0     0
122       0     0     1     0     0
123       0     0     0     2     0
124       0     0     0     0     2
125       0     1     1     1     0
126       0     1     1     1     2
127       0     1     1     1     3
128       0     2     4     2     0
129       0     2     4     2     2
130       0     2     5     2     0
131       0     2     5     2     2
132       1     1     2     2     2
133       3     0     0     1     0
134       3     0     0     1     2
135       1     2     4     2     0
136       1     2     4     2     2
137       1     2     5     2     0
138       1     2     5     2     2
139       5     0     0     2     0
140       5     0     0     2     2
141       4     2     5     2     0
142       4     2     5     2     2
143       5     2     2     1     0
144       5     2     2     1     2
145       5     2     5     2     0
146       5     2     5     2     3
147  RESERVOIRS
148      19       +1.279E+02+1.210E+02+1.800E+03
149      10       +1.267E+02+1.230E+02+9.100E+03
150  +1.100E+02+0.250E+02+1.090E+04+8.500E+03+0.700E-03
151  POINTS INITIALX
152      11
153  +1.000E+36+1.000E+36+1.000E+36+1.000E+36+1.000E+36
154  +1.000E+03+1.000E+02+1.000E+01+0.000E+00+0.000E+00
155  TARIFS E.O.F.
156  +7.370E-01+0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00
157  +0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00
158  +0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00
159  +0.447E-00+0.447E-00+0.447E-00+0.447E-00+0.447E-00+0.447E-00
160  +0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00
161  +0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00
162  +0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.447E-00+0.447E-00
163  +0.447E-00+0.447E-00+0.447E-00+0.447E-00+0.447E-00+0.447E-00
164  +0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00
165  +0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00
166  +0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.447E-00+0.447E-00
167  +0.447E-00+0.447E-00+0.447E-00+0.447E-00+0.447E-00+0.447E-00
```

```
168  +0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00
169  +0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00
170  +0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.447E-00+0.447E-00
171  +0.447E-00+0.447E-00+0.447E-00+0.447E-00+0.447E-00+0.447E-00
172  +0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00
173  +0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.737E-00
174  +0.737E-00+0.737E-00+0.737E-00+0.737E-00+0.447E-00+0.447E-00
175  +0.447E-00+0.447E-00+0.447E-00+0.447E-00+0.447E-00+0.447E-00
```

```
176  COUT DE L'EAU PAR DISTNE
177  +0.198E+03+0.360E+02+0.360E+02+0.360E+02+1.958E+02
178  MODULATION DE LA CONS.
179   4.723E-01+1.205E-01+1.962E-01+3.252E-01+5.011E-01+4.424E-01
180  +5.245E-01+5.480E-01+5.480E-01+5.480E-01+5.245E-01+5.832E-01
181  +5.480E-01+8.999E-01+9.936E-01+7.825E-01+3.838E-01+2.026E-01
182   4.371E-01-5.544E-01-5.544E-01-5.578E-01-5.075E-01-6.482E-01
183      TRA  AGR. PREVUE PT COEF. DE POND.
184      11   11   11   11   11   11   10   10   10   10   10   10
185      10    9    9    9    9    9    9    9    9    9   10   10
186      11
187  +1.000E+02
188  COUT AGR  INT
189  +3.000E-02+3.000E-02+3.000E-02+3.000E-02+3.000E-02+3.000E-02
190  +3.000E-02+3.000E-02+3.000E-02+3.000E-02+3.000E-02+3.000E-02
191  +3.000E-02+3.000E-02+3.000E-02+3.000E-02+2.000E-02+2.000E-02
192  +2.000E-02+2.000E-02+2.000E-02+2.000E-02+2.000E-02+2.000E-02
193        PAS DE LA CON. AGR.
194  +5.000E+02
195  HAUTEURS INITIALES
196  +1.260E+02+1.265E+02
```

ANNEXE 3

Sous programme PRODYN

PRODYN

Programme de résolution d'un problème de commande optimale d'un système dynamique, à une seule variable d'état, par la programmation dynamique.

1) FORMULATION DU PROBLEME DE COMMANDE OPTIMALE

On cherche à minimiser en (x_k, u_k) le critère :

$$J = \sum_{k=1}^{NKT-1} L_k(x_k, u_k) + S(x_{NKT}, NKT)$$

sous les contraintes :

$$\begin{cases} x_{k+1} = F_k(x_k, u_k) & k = \overline{1, (NKT-1)} \\ x_1 \in X_{in} & (X_{in} \text{ donné}) \end{cases} \qquad (1)$$

$$(x_k, u_k) \in C_k \quad , \; k = \overline{1, (NKT-1)} \qquad (2)$$

L'équation (1) correspond à la dynamique du système. On suppose que l'état x_k est scalaire.

L'équation (2) désigne les contraintes liant l'état et la commande à chaque instant.

2) HYPOTHESES ET NOTATIONS UTILISEES POUR PRODYN

L'état prend ses valeurs dans un ensemble discrétisé X, $X = \{ 0., PAES, \ldots, (NPE-1) \times PAES \}$ PAES désigant le pas d'espace d'état et NPE le nombre de points de discrétisation.

$\neq$ La commande (u_k), $k=\overline{1,NKT-1}$, prend ses valeurs dans un ensemble

$U = \{ a_1, \ldots, a_{NCØ} \} \subset R^{IDC}$, NCØ étant le nombre de commandes et IDC leur "dimension".

$\neq$ L'ensemble d'états initiaux X_{in} est un sous-ensemble de X. Il est formé de NEI éléments, correspondant aux points de discrétisation LXI(1), ..., LXI(NEI):

$$X_{in} = \{ (LXI(1)-1) \times PAES, \ldots, (LXI(NEI)-1) \times PAES \}$$

$\neq$ Le coût final S est donné par un tableau CØFI :

CØFI(L)=S((L-1)$\times$ PAES, NKT) , $L = \overline{1,NPE}$

$\neq$ La programmation dynamique détermine le tableau des Revenus et celui des commandes optimales.

Pour la phase (I,K), $I \in \{1,\ldots,NPE\}$, $K \in \{1,\ldots,NKT\}$, le revenu est REV(K,I) et la commande optimale ICØØP(K,I).

Posons :

$XK=(I-1) \times$ PAES ; $XKP1(u)=F_K(XK,u)$ $K \in \{1,\ldots,NKT\}$, $u \in U$

On a :

$$REV(K,I) = \min_{\substack{UK \in U \\ (UK,XK) \in CK}} \{ L_K(XK,UK) + vm(XKP1(UK),K+1) \} \qquad (3)$$

vm(x,K+1) désignant le coût optimal entre (K+1) et NKT, partant de la phase (x,K+1) (vm sera calculé par approximation à partir de REV(L,K+1),$L=\overline{1,NPE}$)

Si $UX(K,I)=a_J$, $J \in \{1,\ldots,NCØ\}$, est l'argument du minimum du membre de droite de l'équation (3), on a :

ICØØP(K,I) = J

Si ce miminum n'existe pas pour une phase (I,K) donnée, alors REV(K,I) = 10^{35} et ICØØP(K,I) = 0.

3) APPEL DU PROGRAMME (≠)

CALL PRØDYN(NKT, NPE, NCØ, IDC, NEI, LXI, CØFI, REV, ICØØP, PAES)

NKT : nombre de points de discrétisation en temps.

NPE : nombre de points de discrétisation d'espace.

NCØ : nombre de commandes.

IDC : "dimension" de la commande.

NEI : nombre d'états initiaux.

LXI : liste des points d'états initiaux; dim(LXI)=NEI.

CØFI: cout final; dim(CØFI)=NPE

REV : tableau des revenus; dim(REV)=NKT x NPE.

ICØØP : tableau des commandes optimales; dim(ICØØP)=NKT x NPE.

PAES : pas de la variable d'état.

4) SOUS-PROGRAMMES UTILISES

4.1) Sous-programme RESTA

PRODYN fait appel à un seul sous-programme : RESTA. Ce dernier permet de résoudre un problème d'optimisation statique : dans PRØDYN, il détermine pour chaque phase (I,K), $I \in \{1, \ldots, NPE\}$, $K \in \{1, \ldots, NKT\}$, la commande optimale, admissible, ICØØP(K,I).

≠ Appel de RESTA

CALL RESTA (NCØ, IDC, NPE, NKT, PAES, REV, ICØØP, I, NS, NA)

I : indice d'état.

K : indice de temps

NS :indicateur de la nature du problème à résoudre (dans PRODYN, NS=1).

NA : 1 si on "balaie" les commandes de a_1 à $a_{NCØ}$;-1 si on les balaie" de $a_{NCØ}$ à a_1.

≠ Remarque : RESTA peut bien sûr être utilisé seul pour la résolution d'un ou plusieurs problèmes statiques Il suffit de prendre I=1, NPE=1 et NS=0 ; si, alors, on appelle NKT fois le sous-programme RESTA, avec le paramètre K variant de 1 à NKT, on aura résolu NKT problèmes statiques (ICØØP(K,1) - ou ICØØP(K) - sera la solution du $K^{ème}$ problème.

4.2) Sous-programmes appelés par RESTA.

RESTA utilise 4 sous-programmes : CONTR, FSD, CREA, CØCØ, qui doivent être définis par l'utilisateur. Les trois premiers cités sont du ty subroutine et le dernier du type FUNCTION.

4.2.1) Sous-programme CONTR

Il permet de vérifier si la contrainte (2) est satisfaite.

+ appel de CØNTR

CALL CØNTR (I, J, K, KID, PAES, NKT, NPE, NS)

I : point d'espace d'état.

J : point de commande (la commande utilisée est alors a_j).

K : point de temps.

$$KID = \begin{cases} 0 & \text{si } ((I-1)\ PAES,\ a_j) \in C_K \\ 1 & \text{sinon} \end{cases}$$

4.2.2) Sous programme FSD

Il détermine la dynamique du système.

+ appel de FSD

CALL FSD (I, J, K, PAES, XKP1)

On aura : $XKP1 = F_K(XK,\ a_j)$ avec $XK = (I-1) \times PAES$.

+ Remarques : Notre définition de $U = \{a_1, \ldots, a_{NCØ}\}$, est purement formelle puisque pratiquement on définit la commande a_J par J. On peut bien sûr définir un tableau $(a_1, \ldots, a_{NCØ})$, qui sera transmis à FSD et à CØNTR en "common", mais dans de nombreux cas celà ne sera pas nécessaire.

4.2.3.) Sous-Programme CREA

Pour une phase (I,K) et une commande J, l'appel du sous-programme F permet de déterminer l'état XKP1 à l'instant (K+1). Comme on ne connait pas le revenu en (XKP1 , K+1), mais seulement aux points de discrétisation de l'espace d'état, CREA permet de déterminer ce revenu à partir des valeurs connues.

✗ Appel de CREA

CALL CREA (REV, XKP1, PAES, NPE, NKT, K, REVMØ, ICØØP)

REVMØ : revenu à la phase (XKP1, K+1) connaissant les revenus REV(L,I), $I \in \{1, \ldots, NPE\}$, $L \in \{K+1, \ldots, NKT\}$.

✗ Un sous-programme CREA a été écrit qui détermine REVMØ par interpolation linéaire sur REV(K+1,I) et REV(K+1, I+1), $(I-1) \times PAES \leq XKP1 < I \times PAES$; il suppose donc qu'aux phases (I,K+1) et (I+1, K+1) il existe une commande optimale (la vérification de cette hypothèse est faite, en principe, dans CØNTR).

✗ Remarque: si le sous-programme RESTA est utilisé seul pour résoudre un problème statique (NS=0), les instructions d'appel de FSD et CREA ne sont pas prises en compte.

4.2.4) Sous-programme CØCØ

Ce sous-programme, du type "FUNCTION", permet la détermination des coûts de la commande pour chaque point d'état et de temps.

Si $ = CØCØ(I,J,K), $ sera le coût de la commande J à la phase (I,K)

BIBLIOGRAPHIE

[1] J. BACHELERY : "Un calculateur industriel surveille et commande l'adduction d'eau potable dans la région de Chambéry" TSM L'eau, Mai 1972.

[2] J. SMITH : "Central Surveillance and Control of the City of Houston's Water Systems". Journal of AWWA, Oct. 1974.

[3] F. FALLSIDE, P.F. PERRY : "Optimal Control of a Water Supply Network by a Decomposition Technique". Report CNED/B-ELEC/TR 20. University of Cambridge, Dpt. of Engineering, 1974.

[4] H.S. RAO, R.A. SEITLE : "Computer Applications in Urban Water Distribution System Control". Journal of Dynamic Systems, Measurement, and Control. Aug. 75.

[5] A. DUPONT : "Hydraulique Urbaine", Eyrolles, 1971.

[6] G. MONTEIL : "Cours d'Hydraulique Théorique", Eyrolles.

[7] R.P. DONACHIE : "A WRA Network Analysis Program", TP 95, Water Research Association, England, March 1973.

[8] A. BENSOUSSAN, J.L. LIONS : "Nouvelles Méthodes en Controle Impulsionnel" Applied Mathematics and Optimization Springer-Verlag, 1974.

[9] C. LEGUAY : "Application du contrôle stochastique à un problème de gestion optimale d'énergie", Thèse de Docteur-Ingénieur, Paris IX, 1975.

[10] M.D. MESAROVIC, D. MACKO, Y. TAKAHARA : "Two Coordination Principles and their Application in Large Scale Systems Control" Automatica, Vol. 6, 1970.

[11] Y. TAKAHARA : "Multilevel Approach to dynamic Optimization" Report SRC, Systems Research Center, Case Western Reserve University, Cleveland (Ohio), 1964.

[12] G. COHEN : "Contribution à la théorie de la commande décentralisée et à la coordination en ligne des systèmes dynamiques", Thèse de docteur-ingénieur, Université d'Orsay, 1975.

[13] A. TITLI : "Contribution à l'étude des structures de commande hiérarchisée en vue de l'optimisation des processus complexes, Thèse d'état Toulouse, 1972.

[14] M. AOKI : "Control of Large-Scale Dynamic Systems by aggregation", IEEE Automatic Control, Volume AC 13, June 1968.

[15] P. BERTRAND, C. MICHAILESCO, J.M. SIRET : "Réduction par agrégation et applications", Journées de Formation AFCET, Toulouse, Octobre 1976.

[16] E.J. DAVISON : "The Robust Decentralized Control of a General Servonne chanism Problem", Control System Report, n° 7423, University of Torents, 1975.

[17] G. COHEN : "Optimization by decomposition and coordination", IEEE Transactions on Automatic control, vol AC 23, n° 2, April 1978.

[18] G. COHEN, G. JOALLAND : "Coordination Methods by the Prédiction Principle in Large Dynamic Constrained Optimization Problems", Proc. IFAC Symposium on Large Scale Systems, Udine, Italy, June 1976.

[19] M.D. MESAROVIC, D. MACKO, Y. TAKAHARA : "Theory of Multilevel Hierarchical Systems", New York, Academic, 1970.

[20] L.S. LASDON : "Optimization Theory for Large Scale Systems", MAC MILLAN, 1970.

[21] J. CEA : "Optimisation : Théorie et Algorithmes", Paris, Dunod, 1971.

[22] J.P. AUBIN : "Méthode de Décentralisation par les Subventions", European Institute for Advanced Studies in Management" (Bruxelles, Belgique), note Interne, 1973 (Mai).

[23] J.L. LIONS : "Controle Optimal de Systemes Gouverné par des équations aux dérivées partielles", Paris, Dunod-Gauthier-Villars, 1968.

[24] A.A. GOLDSTEIN : "Constructive Real Analysis" New-York, HARPER and ROW, 1967.

[25] D.G. LUENBERGER : "Optimization by Vector Space Methods", New York, Wiley, 1969.

[26] J. BENSOUSSAN : "Teoria Moderna de Control Optimo" Cuadermos del Instituto de Mathematica Beppo Levi, 1974.

[27] G. HIMMELBLEAU : "Decomposition of Large Scale Systems", North Holland Publisming Company, 1973.

[28] M. MILGRAM, B. DUBUISSON : "Présentation de trois méthodes de décomposition des systèmes -comparaison", Journées de Formation AFCET "Commmande et Gestion des systèmes complexes", Octobre 1976.

[29] M. RICHETIN : "Analyse structurale des systèmes complexes en vue d'une commande hiérarchisée", Thèse d'état, Toulouse, 1976.

[30] S.V. RAO, S.S. LAMDA : "Suboptimal Control of Industrial Regulators Via Agregation", Proc IFAC Symposium on Large Scale Systems, Udine, 1976.

[31] KELLEY, H.J. : "Method of Gradients, Optimization Techniques", Ed. G. Leitmann, Academic Press, 1962.

[32] R.A. HOWARD :"Dynamic Programming and Markov Processes", MIT Press, Cambridge, 1960.

[33] V.M. DANSKIN : "The theory of Max-Min", Berlin, Springer Verlag, 1967.

[34] G. COHEN, G. JOALLAND : "Application de la commande décentralisée et de la coordination au problème du refoulement optimal dans un réseau hydraulique", Congrès AFCET, Toulouse, Nov. 1975.

[35] R.T. ROCKAFELLAR : "Monotone operators and the proximal point algorithm", SIAM J Control Optimization, vol 14, pp 877-898, Aug. 1976.