



Méthode de pénalisation et applications : résolution et optimisation de modèles macroéconomiques

Pierre Nepomiastchy

► To cite this version:

Pierre Nepomiastchy. Méthode de pénalisation et applications : résolution et optimisation de modèles macroéconomiques. Modélisation et simulation. Université Joseph-Fourier - Grenoble I; Institut National Polytechnique de Grenoble - INPG, 1979. tel-00290419

HAL Id: tel-00290419

<https://theses.hal.science/tel-00290419>

Submitted on 25 Jun 2008

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.

THÈSE

Présentée à

L'UNIVERSITE
SCIENTIFIQUE & MEDICALE

L'INSTITUT
NATIONAL POLYTECHNIQUE

DE GRENOBLE

pour obtenir

LE GRADE DE DOCTEUR D'ETAT ES-SCIENCES

par

P i e r r e N E P O M I A S T C H Y



METHODE DE PENALISATION ET APPLICATIONS
RESOLUTION ET OPTIMISATION DE MODELES MACROECONOMIQUES



Soutenue le 29 juin 1979 devant la Commission d'Examen

JURY

Monsieur R. PERRET

Président

Messieurs A. BENSOUSSAN

C. FOULARD

P.M. LARNAC

M. SAKAROVITCH

R. TREMOLIERES

Examineurs

METHODE DE PENALISATION ET APPLICATIONS.

RESOLUTION ET OPTIMISATION DE MODELES MACROECONOMIQUES.

Thèse sur travaux de Pierre NEPOMIASTCHY

Ma profonde gratitude va à Jacques-Louis LIONS, à qui je dois ma formation de chercheur. Que le Directeur du LABORIA trouve également ici l'expression de ma reconnaissance pour les conditions dans lesquelles j'ai pu effectuer ce travail.

Je tiens à remercier tout particulièrement Alain BENSOUSSAN qui est à l'origine de ce travail et n'a cessé, par la suite, de me prodiguer conseils et encouragements.

Ma reconnaissance va également à Monsieur FOULARD qui a bien voulu accepter la direction de cette thèse.

Je remercie Monsieur PERRET de m'avoir fait l'honneur d'accepter la présidence du Jury et Messieurs LARNAC, SAKAROVITCH et TREMOLIERES pour y avoir participé.

Je n'oublie pas Bruno OUDET, qui m'a initié à la modélisation macroéconomique et m'a fourni les données nécessaires à mes expériences numériques, ainsi que mes amis Daniel GABAY, M'hamed RACHDI, Alain RAVELLI et François RECHENMANN qui ont participé à mes travaux sur les modèles macroéconomiques.

Que mes camarades du LABORIA et tout particulièrement Claude LEMARECHAL, soient ici remerciés pour leur très précieuse et très amicale collaboration.

Je tiens enfin à remercier Chantal PICHON pour le soin et la diligence apportés à la réalisation de ce document.

PLAN

	Page
1. PUBLICATIONS PRESENTEES POUR LA THESE	1
2. AUTRES PUBLICATIONS	3
3. PRESENTATION GENERALE DES TRAVAUX	4
4. LA METHODE DE PENALISATION	7
4.1. <i>Pénalisation en programmation mathématique</i>	8
4.2. <i>Pénalisation pour les systèmes gouvernés par des équations différentielles.</i>	12
4.3. <i>Pénalisation pour les systèmes gouvernés par des inéquations différentielles.</i>	14
5. L'ORDONNANCEMENT	17
6. LA GESTION DE LA PRODUCTION PAR LES STOCKS	22
7. LE TEMPS OPTIMAL	28
8. RESOLUTION ET OPTIMISATION DE MODELES MACROECONOMIQUES	33
8.1. <i>Introduction.</i>	33
8.2. <i>Structure du modèle.</i>	34
8.3. <i>Résolution du modèle.</i>	37
8.4. <i>Optimisation du modèle.</i>	40
9. CONCLUSION	43
9.1. <i>La méthode de pénalisation.</i>	43
9.2. <i>Résolution et optimisation de modèles macroéconomiques.</i>	44

1. PUBLICATIONS PRESENTEES POUR LA THESE.

- [1] "Application de la théorie du contrôle optimal à l'aide de la méthode de pénalisation à la résolution d'un problème d'ordonnancement", Journal de Calcul Numérique et de Physique Mathématique, t. 10, n°4, pp. 908-921, Moscou 1970 (en russe).
- [2] "Une méthode de résolution du problème du temps optimal", Journal de Calcul Numérique et de Physique Mathématique, t. 11, n°1, pp. 79-95, Moscou 1971, (en russe).
- [3] (avec C. LEMARECHAL et G. NISSEN), "Comparaison de méthodes de gestion de stocks-production", Cahier IRIA n°9, pp. 101-180, IRIA, 1972.
- [4] "Pénalisation et contrôle optimal", présenté à Novosibirsk (URSS) en juin 1973, publié dans "Sur les méthodes numériques en sciences physiques et économiques", série Méth. Math. de l'Ingénieur (Eds., J.L. Lions et G.I. Martchouk), Dunod 1975.
- [5] "Méthode de pénalisation et applications", IERESM (Inst. Europ. Rech. Et. Sup. Management), Doc. Trav. 75-26 et 75-28 (254 pages), Bruxelles, 1975.
- [6] "Etude d'un système gouverné par une inéquation différentielle", présenté à Novosibirsk (URSS) en juin 1976, publié dans "Analyse numérique des grands systèmes", série Méth. Math. de l'Ingénieur (eds., J.L. Lions et G.I. Martchouk), Dunod 1978.
- [7] "Méthodes d'optimisation adaptées aux modèles économiques", présenté à Versailles en déc. 1976, publié dans New Trends in Systems Analysis", pp. 626-635, (eds., A. Bensoussan et J.L. Lions), Lecture Notes in Control and Information Sciences, Springer Verlag, 1977.
- [8] (avec A. RAVELLI), "Adapted Methods for Solving and Optimizing Quasi-Triangular Econometric Models", Annals of Economic and Social Measurement, Vol. 6, 1978.
- [9] "Résolution d'un problème d'ordonnancement à ressources variables", RAIRO - Recherche Opérationnelle, Vol. 12, n°3, 1978.
- [10] (avec B. OUDET et F. RECHENMANN), "MODULECO, aide à la construction et à l'utilisation de modèles économiques", présenté en déc. 1978 au Colloque International sur l'Analyse et l'Optimisation des Systèmes, à paraître dans Lecture Notes in Control and Information Sciences, Springer Verlag.

- [11] "MODULECO, Software for Macroeconomic Modelization", présenté en déc. 1978 au Colloque IRIA- Novosibirsk - Pavie, à paraître dans la série Méth. Math. de l'Ingénieur, Dunod.
- [12] (avec D. GABAY), "Problemi numerici legati ai grandi modelli econometrici : presentazione del progetto MODULECO", in Teoria dei sistemi ed Economia, A. Marzollo ed., CIMS, Franco Angeli, 1979.
- [13] (avec D. GABAY), "Numerical Methods for Simulation and Optimal Control of Large Scale Macroeconomic Models", in Applications of Stochastic Control to Econometrics and Management, A. Bensoussan, P. Kleindorfer, C.S. Tapiero, eds., North Holland, 1979.

2. AUTRES PUBLICATIONS.

- [14] "Propagation de l'erreur dans les schémas à pas fractionnaires pour les équations aux dérivées partielles linéaires et paraboliques", thèse de Docteur-Ingénieur, Toulouse 1969.
- [15] "Application of Penalty Technique to Solve a Scheduling Problem and Comparison with Combinatorial Methods", Rapport LABORIA n°7, 1973.
- [16] "Une méthode de résolution du problème du temps optimal pour les systèmes gouvernés par des équations différentielles", IERESM, Doc. Travail 74-25, Bruxelles, 1974.
- [17] "Compléments sur la pénalisation en programmation convexe", IERESM, Doc. Travail 74-44, Bruxelles, 1974.
- [18] (avec D. GABAY, M. RACHDI et A. RAVELLI), "Etude, résolution et optimisation de modèles macroéconomiques", Rapport LABORIA n°312, 1978.
- [19] (avec A. RAVELLI et F. RECHENMANN), "An Automatic Method to Get an Econometric Model in a Quasi-Triangular Form", Rapport LABORIA n° 313, 1978.
- [20] "MODULECO, projet de système informatique pour la construction et l'utilisation de modèles macroéconomiques", présenté en 1978 au Colloque sur la Décision Parlementaire et l'Informatique, Sénat, proceedings à paraître.

3. PRESENTATION GENERALE DES TRAVAUX.

L'Analyse des Systèmes, à laquelle se rattachent les travaux de cette thèse, est une discipline scientifique dont les contours sont encore mal définis, ce qui est largement dû à l'extrême diversité de ses applications (sciences de l'ingénieur, économétrie, sociologie, urbanisme, écologie, médecine, linguistique ... etc...). On peut cependant donner des caractéristiques générales de ce que l'on appelle "étude systémique". Celle-ci comporte toujours en effet les étapes suivantes :

- 1) la définition précise des contours du phénomène étudié, afin de distinguer sans ambiguïté le phénomène de son environnement, et l'établissement de la liste des variables mesurant l'état du phénomène d'une part, les grandeurs de l'environnement en interaction avec le phénomène d'autre part.
- 2) la modélisation, c'est-à-dire la construction du système d'équations liant les variables définies ci-dessus, avec identification des paramètres inconnus s'il y a lieu. Ce système d'équations est appelé le modèle.
- 3) la simulation ; pour des valeurs données des variables de l'environnement (variables exogènes, de contrôle, de perturbation ...), il s'agit de calculer, à l'aide du modèle, l'évolution dynamique des variables "internes" (appelées d'état ou endogènes) et des variables de sortie.

De plus, l'étude systémique d'un phénomène peut comporter :

- 4) une étude de structure du modèle conduisant entre autres à la construction de maquettes (modèles réduits) et à la décomposition du modèle en sous-modèles.
- 5) une analyse de sensibilité, c'est-à-dire la réponse du modèle à une modification marginale de son environnement.
- 6) l'optimisation ; l'utilisateur du modèle dresse la liste des variables de contrôle (instruments), des contraintes s'il y a lieu et choisit un critère (fonction objectif) qui est en général une fonction des

variables de sortie ; il s'agit alors de trouver les variables de contrôle qui, par l'intermédiaire du modèle, donneront les variables de sortie qui optimisent le critère choisi compte tenu des contraintes.

Ainsi replacées dans le contexte général de l'Analyse des Systèmes, les recherches effectuées dans le cadre de cette thèse ont été limitées à :

- l'optimisation déterministe (i.e. non stochastique) des modèles microéconomiques ;
- l'étude de structure, la simulation et l'optimisation des modèles macroéconomiques déterministes.

Les phénomènes choisis comme domaines d'application de cette thèse ont donc toujours été étudiés par l'intermédiaire d'un modèle préalablement construit et fourni à l'auteur par un spécialiste du domaine d'application considéré. L'objet des travaux que nous présentons est d'aider ce spécialiste à utiliser son modèle.

Il va de soi que les problèmes de mathématiques appliquées et d'automatique posés par l'utilisation d'un modèle dépendent de ce dernier. Cependant, il est possible en général de dégager, pour chaque domaine d'application, des problèmes qui soient suffisamment généraux pour être communs à un grand nombre de modèles de ce domaine d'une part et qui soient suffisamment précis pour que leurs algorithmes de résolution puissent être concrètement utilisables par les spécialistes de ces domaines d'autre part.

Pour chacun des domaines d'application mentionnés ci-dessous, nous avons étudié plusieurs modèles réels fournis par des spécialistes, nous avons dégagé des problèmes de mathématiques appliquées et d'automatique communs à ces modèles, nous avons proposé des outils méthodologiques pour leur résolution, nous avons programmé les algorithmes dérivés de ces outils et nous avons testé nos programmes sur au moins un des modèles proposés.

Entre avril 1969 et septembre 1975, nous avons choisi les applications microéconomiques suivantes :

- l'ordonnancement (cf. [1], [5], [9] et [15]) ;
- la gestion de stocks-production (cf. [3] et [5]) ;
- l'atterrissage d'un avion en temps minimal (cf. [5] et [16]).

L'unité entre ces trois études apparemment totalement disjointes réside dans la méthodologie employée : la méthode de pénalisation.

Depuis septembre 1975, nous nous sommes orientés vers la macroéconomie dans le but d'obtenir des algorithmes efficaces (parce que spécialement adaptés) de résolution et d'optimisation des grands modèles macroéconomiques de l'Administration (cf. [7], [8], [13], [18], [19]). Depuis 1978, ces travaux s'insèrent dans MODULECO, projet de système informatique pour la modélisation économique (cf. [10], [11], [12] et [20]) projet réalisé par un Club (regroupant essentiellement l'IRIA, l'INSEE, la Direction de la Prévision, l'IMAG et le Sénat) dont l'auteur de cette thèse est le Directeur.

4. LA METHODE DE PENALISATION.

Rappelons brièvement l'idée directrice de la méthode de pénalisation. Supposons que le problème à résoudre, appelé problème P, soit la minimisation d'une fonctionnelle $J(x)$ sur un ensemble D. On définit une fonctionnelle $G(x)$, appelée "fonction de pénalité" de l'ensemble D, comme une fonctionnelle qui, par définition, est non négative pour tout x et nulle si et seulement si $x \in D$. On définit alors un "critère pénalisé" $J_\epsilon(x) = J(x) + \frac{1}{\epsilon} G(x)$, où ϵ est un paramètre strictement positif et un "problème pénalisé" P_ϵ qui est la minimisation sans contraintes du critère J_ϵ .

Disons d'abord que la méthode de pénalisation n'a de sens que si le problème P_ϵ est nettement plus simple à résoudre que le problème P. C'est bien entendu le cas pour tous les exemples d'application étudiés dans cette thèse. De plus, pour les systèmes gouvernés par des équations différentielles avec contraintes sur l'état, le principe du maximum de Pontryagin est applicable au problème P_ϵ et n'est pas applicable au problème P, d'où une différence qualitative de difficulté entre ces deux problèmes.

Pour que l'application de la méthode de pénalisation ait un sens, il faut également que toute solution du problème P_ϵ pour de "petites" valeurs du paramètre ϵ soit une solution "approchée" du problème P. En d'autres termes, il faut que la méthode converge. L'étude de la convergence dépend du cadre mathématique dans lequel le problème d'optimisation est posé. En effet, le seul résultat général connu (Fiacco- Mac Cormick en 1966) montre que si la suite des solutions des problèmes P_ϵ converge, alors elle converge nécessairement vers une des solutions du problème P. Ce résultat, bien qu'étant d'un intérêt pratique essentiel, ne constitue qu'une validation a posteriori de l'emploi de la méthode, puisqu'il n'assure pas a priori la convergence.

Nous avons étudié la convergence de la méthode de pénalisation dans deux cadres mathématiques précis, la programmation mathématique et le contrôle optimal des systèmes gouvernés par des équations différentielles, parce que ce sont les cadres mathématiques des problèmes concrets

d'Analyse des Systèmes que nous avons eus à résoudre. Nous rappelons ci-dessous les résultats théoriques que nous avons obtenu. Nous renvoyons le lecteur aux paragraphes suivants de cette thèse pour l'application concrète de ces résultats et à l'étude bibliographique de notre travail [5] pour l'application de la pénalisation dans d'autres cadres mathématiques.

Pour tous les problèmes d'optimisation que nous avons été amenés à résoudre, le critère d'optimisation était économique. Nous avons été naturellement amenés à donner une interprétation économique de la méthode de pénalisation. L'interprétation théorique est bien connue : le terme de pénalité $\frac{1}{\epsilon} G(x)$ est un coût de violation des contraintes et lorsque ϵ tend vers zéro, le coût marginal de violation augmentant indéfiniment, la solution du problème pénalisé tend à ne plus violer les contraintes. Cette interprétation, utile pour "expliquer" la méthode de pénalisation, n'a malheureusement pas d'intérêt pratique et nous en donnerons de plus utilisables pour chacun des cas traités.

4.1. Pénalisation en programmation mathématique.

Soit D le sous-ensemble de l'espace euclidien R^n défini comme étant l'ensemble des x vérifiant :

$$(1a) \quad g_k(x) = 0 \quad , \quad k = 1, \dots, p$$

$$(1b) \quad h_k(x) \leq 0 \quad , \quad k = 1, \dots, q$$

Le cas où toutes les contraintes sont d'égalité (resp. d'inégalité) n'est qu'un cas particulier de notre étude. Soit J une fonctionnelle de R^n dans R et P le problème initial :

Problème P : trouver $x^* \in D$ tel que :

$$(2) \quad J(x^*) \leq J(x) \quad , \quad \forall x \in D$$

On pose successivement :

$$(3) \quad G(x) = \sum_{k=1}^p g_k^2(x) + \sum_{k=1}^q (\max [h_k(x), 0])^2$$

$$(4) \quad J_{\varepsilon}(x) = J(x) + \frac{1}{\varepsilon} G(x)$$

Le problème pénalisé P_{ε} est alors :

Problème P_{ε} : trouver $x_{\varepsilon} \in R^n$ tel que :

$$(5) \quad J_{\varepsilon}(x_{\varepsilon}) \leq J_{\varepsilon}(x) \quad , \quad \forall x \in R^n$$

Le problème (5) est la minimisation sans contraintes d'une fonction de plusieurs variables et l'on peut utiliser pour le résoudre des algorithmes très efficaces (méthodes de quasi-Newton à métrique variable) ou extrêmement simples (gradient simple, une demi-douzaine d'instructions FORTRAN) ce qui n'est pas le cas du problème (2). Notons que contrairement aux méthodes duales, nous n'avons pas augmenté le nombre de variables. Le jeu d'hypothèses classique est :

$$(6) \quad \left\{ \begin{array}{l} - \text{l'ensemble } D \text{ défini par (1) n'est pas vide.} \\ - \text{les fonctions } g_k, h_k \text{ et } J \text{ sont toutes continues.} \\ - \text{la fonction } J \text{ est faiblement coercive (*)} \end{array} \right.$$

Le résultat classique est le suivant :

Théorème 1.

En plus des hypothèses (6), on suppose que les fonction g_k sont linéaires, les fonctions h_k sont convexes et la fonction J est strictement convexe. Alors le problème P a une solution unique x^* , le problème P_{ε} a une solution unique x_{ε} pour tout $\varepsilon > 0$ et x_{ε} converge vers x^* dans R^n lorsque ε tend vers zéro par valeurs positives.

□

Nous avons démontré en [5] le résultat original suivant :

(*) i.e. pour toute suite $\{x_m\}$ telle que $\|x_m\| \rightarrow \infty$, on a $J(x_m) \rightarrow +\infty$.

Théorème 2.

On fait uniquement les hypothèses (6) et l'on a les résultats suivants. Le problème P a au moins une solution et l'on appelle S^* l'ensemble de ses solutions. Pour tout $\varepsilon > 0$, le problème P_ε a au moins une solution. Si l'on appelle $\{x_\varepsilon\}$ la suite telle que, pour tout $\varepsilon > 0$, x_ε est l'une quelconque des solutions du problème P_ε , alors $\{x_\varepsilon\}$ est l'union de sous-suites, chacune d'entre elles convergeant vers un élément de S^* lorsque ε tend vers zéro par valeurs positives.

□

On peut formuler ce résultat autrement : la distance dans R^n de x_ε , solution arbitraire de P_ε , à l'ensemble S^* des solutions de P , tend vers zéro avec ε .

Ce résultat constitue bien une validation a priori de l'emploi de la méthode et nous n'en connaissons pas de meilleur dans le cas non convexe. Ce résultat est original car si Fiacco - McCormick (*) avaient démontré en 1968 la convergence de la pénalisation dans le cas non convexe, ils faisaient alors des hypothèses directement sur l'ensemble S^* des solutions de P , ensemble par définition inconnu de l'utilisateur de la méthode.

Nous avons par ailleurs obtenu [5] des résultats allant plus loin que le théorème 1 dans le cas convexe et différentiable :

Théorème 3.

On suppose :

- (i) les fonctions g_k , h_k et J sont toutes deux fois continûment différentiables par rapport à toutes les variables. On appelle G_k'' , H_k'' , J'' les hessiens de ces fonctions.
- (ii) La matrice $J''(x)$ est coercive uniformément en x , les matrices $G_k''(x)$ sont identiquement nulles (i.e. les g_k sont linéaires) et les matrices $H_k''(x)$ sont positives (les h_k sont convexes).
- (iii) D est non vide.

(*) FIACCO A.V. ; Mc CORMICK G.P. "Non Linear Programming : Sequential

- (iv) pour tout point x de la frontière de D , l'ensemble des gradients des contraintes actives en x forme un système libre de $\mathbb{R}^n$.
- (v) hypothèse technique, détaillée page 33 du tome 1 de [5], dont l'explicitation impliquerait trop de notations nouvelles. Cette hypothèse concerne la régularité du domaine D ; elle fut toujours satisfaite dans nos applications et l'on peut montrer qu'elle est toujours satisfaite si toutes les contraintes sont linéaires.

On a alors :

- a) le théorème 1 s'applique (évident).
- b) la convergence de x_ε vers x^* est d'ordre $O(\varepsilon)$, c'est-à-dire que la suite $(x_\varepsilon - x^*) / \varepsilon$ est bornée.
- c) le vecteur λ_ε , de dimension $p+q$, défini par :

$$(7a) \quad \lambda_{k\varepsilon} = \frac{2}{\varepsilon} g_k(x_\varepsilon) \quad , \quad k = 1, \dots, p$$

$$(7b) \quad \lambda_{k\varepsilon} = \frac{2}{\varepsilon} \max [h_{k-p}(x_\varepsilon), 0] \quad , \quad k = p+1, \dots, p+q$$

converge vers la solution duale unique du problème P lorsque ε tend vers zéro par valeurs positives.

□

Le résultat 3 a un intérêt pratique important, car il permet d'appliquer au vecteur λ_ε , facile à calculer, les interprétations économiques classiques des variables duales (prix des contraintes).

Nous avons appris postérieurement à la publication de [5] que les résultats b et c de notre théorème 3 avaient été antérieurement démontrés par Fiacco - Mac Cormick^(*). Il reste que leur démonstration est très différente de la nôtre et que plusieurs résultats intermédiaires que nous avons obtenus sont, au moins à notre connaissance, originaux.

(*) cf. note en bas de la page 10.

4.2. Pénalisation pour les systèmes gouvernés par des équations différentielles.

Dans de très nombreux domaines d'application (aéronautique, aérospatiale, gestion de stocks-production en temps continu, modèles macroéconomiques théoriques en temps continu ... etc...) le formalisme mathématique utilisé pour décrire le comportement dynamique du phénomène est le système d'équations différentielles ordinaires.

L'optimisation d'un système gouverné par des équations différentielles est parfois désignée par "théorie du contrôle optimal" quoique, à notre avis, ce terme désigne une notion plus générale englobant en particulier l'optimisation des systèmes gouvernés par des équations aux dérivées partielles (systèmes à paramètres distribués).

On pourrait penser que l'optimisation des systèmes gouvernés par des équations différentielles est un problème résolu, au moins en théorie, par le principe du maximum de Pontryagin. En réalité, l'énoncé classique de ce principe n'est valable que lorsque les contraintes du modèle ne portent que sur le contrôle et sont indépendantes du temps. Or, il se trouve que la majorité des problèmes d'application, et en particulier la gestion de stocks-production, font intervenir des contraintes portant explicitement sur l'état du système (stocks positifs par exemple). Certes, il existe des extensions théoriques du principe du maximum au cas de contraintes sur l'état (à commencer par celle de Pontryagin, au chapitre VI de son livre) mais nous n'en connaissons aucune qui soit utilisable sur le plan pratique (i.e. pour calculer numériquement le contrôle optimal).

Pour le traitement des contraintes sur l'état, on dispose essentiellement de deux techniques, la dualité et la pénalisation, dont l'application pratique à un problème de gestion de stocks-production a été comparée dans [3]. Cette comparaison n'a pas montré que l'une des deux méthodes domine totalement l'autre (chacune ayant ses qualités et ses défauts) mais l'on peut dire que la pénalisation est plus simple à mettre en oeuvre et qu'elle s'applique aussi au cas non convexe.

Bien que l'application de la méthode de pénalisation à la résolution de ce genre de problèmes soit ancienne et d'usage courant, la validité de son emploi a été très peu étudiée et semble se réduire à deux résultats de validation a posteriori par Lasdon - Warren - Rice et Okamura. Nous avons démontré [5] des résultats originaux de validation a priori dans l'espace $L^2(0,T;R^n)$ des vecteurs-fonctions de carré sommable.

Les hypothèses générales de notre étude sont les suivantes. Toutes les fonctions définissant le problème sont continues par rapport à x et u et lipschitziennes par rapport à t . Le domaine des contrôles admissibles n'est pas vide. Le critère est faiblement coercif.

Nous obtenons alors dans le cas non convexe un résultat de convergence analogue au théorème 2 (page 10) en faisant une hypothèse de compacité très peu restrictive : une au moins des solutions du problème initial est "assez régulière" (en fait, équi-continue par morceaux).

Dans le cas convexe (c'est-à-dire équation d'état linéaire, contraintes et critère convexe) nous démontrons un théorème de convergence analogue au théorème 1 (avec convergence faible du contrôle et convergence forte de l'état) sans faire d'hypothèse sur la régularité de la solution.

Nous proposons ensuite (cf. [5] pp. 117-128) deux méthodes constructives de résolution du problème pénalisé (en donnant notre avis sur le choix à faire en fonction des caractéristiques du problème) : d'une part, le calcul du gradient réduit par la méthode de l'état-adjoint et d'autre part l'application du principe du maximum avec l'algorithme de Newton - Raphson pour la résolution numérique du problème aux deux bouts.

Nous donnons (cf. [5] , pp. 128-131) une interprétation économique, en termes de prix d'usage, des variables adjointes du problème pénalisé. Cette interprétation, basée sur l'interprétation économique connue du principe du maximum, est d'un intérêt pratique considérable. En effet, non seulement ces prix constituent pour l'utilisateur une information qui

vient s'ajouter à celle donnée par la valeur du contrôle optimal mais, de plus, l'expérience acquise (en particulier à EDF) prouve que l'information fournie par les prix est en général plus fiable car moins sensible aux perturbations.

4.3. Pénalisation pour les systèmes gouvernés par des inéquations différentielles.

Soit le problème de gestion de stocks-production unidimensionnel très simple suivant. Pour tout t , on appelle $x(t)$ le niveau du stock, $u(t)$ l'intensité de production (entre 0 et 1), b le coefficient technologique de production et $d(t)$ la demande instantanée. Lorsque le niveau du stock le permet ($x(t) > 0$), la demande est satisfaite et la loi d'évolution du système est :

$$(8) \quad \frac{dx}{dt} = b u(t) - d(t)$$

La formalisation classique de l'optimisation de ce système est la suivante. On interdit la rupture de stock, ce qui revient à imposer la contrainte $x(t) \geq 0$, et l'on cherche la politique de contrôle $u(.)$ qui minimise un coût de production et de stockage compte tenu du fait que x et u doivent satisfaire :

$$(9) \quad \frac{dx}{dt} = b u(t) - d(t) , x(t) \geq 0, \quad 0 \leq u(t) \leq 1, \quad \text{presque tout } t$$

On est alors dans le cadre mathématique étudié au paragraphe précédent, car l'équation différentielle (8) est toujours vérifiée. D'un point de vue économique, ce formalisme revient à supposer que le coût de rupture de stock est infini. Dans de nombreuses applications, cette approche n'est pas satisfaisante, car la rupture de stock, c'est-à-dire le fait qu'un client ne soit pas servi, est certes dommageable, mais n'est pas exclue. Si l'on admet l'éventualité d'une rupture de stock, alors pour tout intervalle de temps pendant lequel la demande peut être satisfaite (régime "libre") le système évolue selon la loi (8) et pour tout intervalle de temps pendant lequel la demande ne peut pas être

entièrement satisfaite (régime "bloqué") le système suit la trajectoire $x(t) = 0$.

Dans cette approche, on voit que l'équation d'état du système n'est pas la même suivant que le régime est libre ou bloqué. On peut montrer (cf. [6]) que la loi d'évolution du système est alors :

$$(10) \quad \begin{cases} \left[\frac{dx}{dt} - b u(t) + d(t) \right] x(t) = 0 & \text{pour presque tout } t \\ x(t) \geq 0, \quad \frac{dx}{dt} \geq b u(t) - d(t) & , \text{ pour presque tout } t \end{cases}$$

c'est-à-dire que la loi d'évolution est une inéquation différentielle.

Le problème d'optimisation posé est alors le suivant : minimiser un critère comprenant les coûts de production, de stockage et de rupture, compte tenu de (10) et de $u(t) \in [0,1]$.

La motivation de l'étude ayant été montrée sur un exemple choisi volontairement le plus simple possible, disons brièvement que nous avons étudié les systèmes où l'équation (8) est remplacée par une équation générale (dimension n) et la contrainte $x(t) \geq 0$ est remplacée par $x(t) \in K$, où K est un domaine de $\mathbb{R}^n$. Il faut noter que, dans ce contexte, même le problème de la résolution du modèle (trouver x connaissant u) n'est pas évident.

Nous avons étudié dans [6] le cas où l'inéquation différentielle est linéaire, où K est convexe et où le critère est convexe. Nous avons montré comment la méthode de pénalisation permet de construire une équation différentielle dont la solution soit une approximation de l'inéquation différentielle initiale. Cette approche donne à la fois des résultats théoriques sur l'existence et l'unicité de la solution du problème initial et une méthode constructive de résolution de ce dernier en fournissant un problème approché dont la résolution est standard. Tous ces résultats sont originaux, au moins à notre connaissance.

Notons au passage une application inhabituelle de la méthode de pénalisation : l'étude théorique de l'existence de la solution du problème initial (*).

Nous avons appliqué numériquement dans [6] la méthode proposée à l'optimisation d'un problème de gestion de stocks-production avec coût de rupture.

Nos résultats ont été étendus au cas où l'équation (8) dépend non linéairement de l'état dans une thèse que nous avons dirigée (**).

(*) on montre que la solution du problème pénalisé converge vers une solution du problème initial, ce qui prouve que le problème initial a solution.

(**) J.R. TCHABO, "Etude d'un système gouverné par une inéquation différentielle non linéaire", Thèse de 3e Cycle, Université Paris IX, 1977

5. L'ORDONNANCEMENT.

Nous renvoyons à [5] (pp. 19-22) le lecteur intéressé par une étude bibliographique complète des travaux sur l'ordonnancement antérieurs à 1975. Nous renvoyons également à [5] (pp. 49-80) pour l'application de la méthode de pénalisation à un problème d'ordonnancement classique (modèle de Conway).

Nous allons rappeler brièvement ici nos travaux (cf. [5] pp. 81-97 et [9]) sur la résolution d'un problème d'ordonnancement à ressources variables. Ce problème constitue une généralisation des modèles de Conway et de Manne. Mis à part les travaux de Trémolières spécialement adaptés au cas où toutes les opérations élémentaires sont de durée identique, nous ne connaissons pas de méthode de résolution autre que la nôtre du problème formulé ci-dessous, bien que celui-ci corresponde beaucoup mieux à la réalité que le classique modèle de Conway. Notre modèle constitue, par comparaison au modèle de Conway, une généralisation dans le sens où :

- a) nous ne supposons pas que les machines sont toutes distinctes et toutes disponibles en permanence ;
- b) les règles de précédence peuvent être plus complexes ;
- c) il peut y avoir, en plus, des contraintes de dates de début au plus tôt et de fin au plus tard spécifiques à chaque opération.

Décrivons brièvement ce modèle. Il s'agit d'exécuter n tâches notées (i) , chacune d'entre elles étant composée de a_i opérations élémentaires appelées simplement opérations. Nous donnerons plus loin un exemple concret de tâches et d'opérations. On désigne par s le double indice $\{i, j\}$, une opération est notée indifféremment O_s ou O_{ij} (j -ème opération de la i -ème tâche) et, par convention, on note respectivement $s-1$ et $s+1$ les doubles indices $\{i, j-1\}$ et $\{i, j+1\}$.

On dispose pour exécuter ces opérations d'un certain nombre m de groupes de machines. Toutes les machines d'un même groupe sont identiques et interchangeables. Pour chaque opération, on sait sur quel groupe de machines doit être effectué le travail, mais on a le choix de la machine à l'intérieur du groupe. En d'autres termes, et contrairement au modèle classique, l'affectation opérations-machines n'est pas complètement faite à l'avance.

Pour un atelier d'usinage, on pourra avoir :

- tâches : fabriquer les pièces A, B, C ... etc...
- opérations : passer la pièce A sur le tour, la pièce B sur la fraiseuse
- machines : 5 tours, 3 fraiseuses, 8 pointeuses, ... etc...

On suppose donné le tableau $K(k,t)$ qui indique le nombre de machines du groupe k disponibles à la date t . Le fait de rendre le tableau K dépendant du temps permet de tenir à la fois des indisponibilités prévues (jours de maintenance, etc...) et des croissances prévues (livraison d'une machine supplémentaire, etc...) pour chaque ressource. Notons que le modèle classique est simplement le cas particulier où le tableau K est identiquement égal à 1.

Un ordonnancement, pour être admissible, doit satisfaire les 6 règles suivantes :

- règle 1 : la première opération de chaque tâche ne peut commencer avant une date connue, d_i^- , appelée date de début au plus tôt de la tâche (date de livraison des matières premières).
- règle 2 : la dernière opération de chaque tâche ne peut se terminer après une date connue, d_i^+ , appelée date de fin au plus tard de la tâche (carnet de commande).
- règle 3 : les interruptions dans l'exécution d'une opération sont interdites et le temps d'exécution de chaque opération p_s est connu.
- règle 4 : pour toute opération O_s qui n'est pas la dernière de sa tâche, il doit s'écouler au moins une durée connue q_s entre le début des opérations O_s et O_{s+1} . Dans le modèle classique, on a toujours $q_s = p_s$; ici, nous acceptons des contraintes technologiques et organisationnelles plus complexes (voir plus loin).
- règle 5 : chaque opération O_s doit être exécutée sur une machine quelconque du groupe k_s , où le tableau k_s est connu.
- règle 6 : pour tout k et tout t , le nombre d'opérations en cours d'exécution sur le groupe de machines numéro k pendant la période t , ne doit pas dépasser le nombre $K(k,t)$ connu.

Nous appellerons u_s la date de début de l'exécution de l'opération O_s . Nous appellerons $N(k,t)$ le nombre d'opérations réellement en cours sur le groupe de machines k à la date t . On montre (cf. [5] et [9]) qu'un ordonnancement est entièrement défini par le vecteur u de composantes u_s , que le tableau N est entièrement défini par la donnée de u et qu'un ordonnancement u est admissible si et seulement si il vérifie :

$$(11a) \quad u_{i,1} \geq d_i^-, \quad i=1,\dots,n$$

$$(11b) \quad u_s + p_s \leq d_i^+, \quad \text{si } O_s \text{ est la dernière opération de } (i)$$

$$(11c) \quad u_{s+1} \geq u_s + q_s, \quad \text{si } O_s \text{ n'est pas la dernière opération de } (i)$$

$$(11d) \quad N(k,t) \leq K(k,t), \quad \forall k \text{ et } \forall t$$

Le tableau N étant entièrement défini par la donnée du vecteur u , nous définirons l'application $u \rightarrow G(u)$ par :

$$(12) \quad G(u) = \sum_k \sum_t \max [N(k,t) - K(k,t), 0]$$

La fonction G est non négative et elle est nulle si et seulement si la contrainte (11d) est toujours satisfaite. La fonction G est donc une fonction de pénalité de la contrainte (11d), c'est-à-dire de la contrainte cumulative (Règle 6) de notre problème. Les contraintes (11a), (11b), et (11c) étant linéaires, on n'a pas intérêt à les pénaliser et nous appellerons D l'ensemble des u vérifiant ces contraintes. Par conséquent, un ordonnancement est admissible si et seulement si $u \in D$ et $G(u) = 0$.

Si le problème d'ordonnancement posé est la minimisation d'un critère, on peut appliquer les techniques proposées au paragraphe 4.1.

En réalité, dans de nombreux problèmes d'ordonnancement, le problème posé est la recherche d'un ordonnancement admissible. C'est le cas lorsque, le personnel étant payé au mois et les machines appartenant à l'entreprise, il n'y a pas lieu de distinguer entre deux ordonnancements admissibles et la seule difficulté est de respecter le carnet de commande.

Dans ce cas, on cherche u vérifiant $u \in D$ et $G(u) = 0$. Comme la fonction est non-négative, ce problème est équivalent à la minimisation de la fonction G sous les contraintes linéaires $u \in D$. Nous avons mis au point (cf. [5] et [9]) un algorithme de résolution de ce problème spécialement adapté à la forme très particulière de G et de D .

La fonction G s'interprète naturellement comme un coût de violation de la contrainte cumulative (11d), c.a.d. de la règle 6. Nous appellerons problème P la minimisation de G sous les contraintes $u \in D$. Le problème P s'interprète : parmi tous les ordonnancements respectant les règles 1 à 5, trouver celui qui viole le moins la règle 6.

Lorsqu'il existe au moins un ordonnancement admissible, toute fonction G pénalisant (11d) convient, en ce sens que toute solution du problème P est un ordonnancement admissible et réciproquement.

Supposons maintenant, ce qui est en général réaliste, qu'il soit possible de violer la règle cumulative 6, soit en louant des machines supplémentaires, soit en faisant exécuter certaines opérations à l'extérieur. Cette violation entraîne naturellement un coût que nous appellerons globalement "coût de sous-traitance". Le problème P est alors la minimisation de ce coût de sous-traitance, qui est nul pour tout ordonnancement admissible.

On voit donc que notre méthode, élaborée pour la recherche d'un ordonnancement admissible, s'applique de façon très naturelle au cas assez courant où le carnet de commande de l'entreprise est (dans la période considérée) surchargé. Dans ce cas, en effet, il n'y a pas d'ordonnancement admissible et le problème est bien de minimiser un coût de sous-traitance. Le critère (12) est la durée totale de sous-traitance; dans un problème concret, il faut naturellement pondérer le critère (12) par les coûts horaires de sous-traitance pour chaque machine.

Nous avons appliqué (cf. [5] et [9]) notre méthode de recherche d'un ordonnancement admissible à la résolution d'un problème concret pour une usine de fabrication de pièces détachées de machines textiles

qui nous a été fourni par l'Université Technique de Manchester (Angleterre). La règle de précédence (règle 4) était relativement complexe : il y avait des opérations exécutées à l'extérieur, des opérations ne demandant aucune ressource (séchage des pièces), chaque opération se décomposait en réglage de la machine et exécution, il était possible de commencer le réglage pour l'opération suivante avant la fin de l'opération en cours etc... Quoiqu'il en soit, nous avons réussi à formuler ce problème sous la forme (11) et à le résoudre à la satisfaction de l'utilisateur : l'algorithme fourni ordonnance 288 opérations sur 13 machines en 14 secondes de calcul d'un ordinateur GE 235.

Par la suite, notre méthode a été appliquée avec succès à un autre problème réel, l'optimisation de la production d'une usine d'armements à Tarbes (*) .

(*) cf. M. Mondain "Système de calcul des plans de production à long terme avec optimisation assistée", Thèse de 3e Cycle, Bordeaux, mai 1976.

6. LA GESTION DE PRODUCTION PAR LES STOCKS.

On se propose de montrer ici sur un exemple, comment les méthodes théoriques exposées dans le paragraphe 4.2. peuvent s'appliquer à un problème réel. Nous avons choisi un problème précis de gestion de la production par les stocks. Les méthodes restent applicables si le problème est plus complexe que celui que nous avons choisi à titre d'exemple, à condition toutefois que soient satisfaites les hypothèses fondamentales suivantes :

- a) l'intensité de production peut varier continûment entre zéro et la capacité maximale de production ;
- b) la demande en produits finis est connue à l'avance.

Nous étudions ici le cas d'une usine produisant, pendant une période de temps $[0, T]$ fixée, n produits finis non périssables. Chacun de ces produits finis est fabriqué par une machine qui lui est propre.

Pour chaque machine (i) , on introduit une variable de décision u_i dont la valeur $u_i(t)$ à chaque instant indique le taux d'utilisation de la machine (i) , taux compris entre 0 et 1 : si $u_i(t) = 1/2$, cela signifie que la machine (i) fonctionne à l'instant t à la moitié de son rendement maximal qui est noté m_i .

Chaque produit fini (i) , une fois fabriqué, est conservé dans un stock qui lui est propre et dont le niveau à l'instant t est noté $s_i(t)$.

Pour chaque produit fini (i) , il y a une demande dont on suppose qu'elle est une fonction du temps $d_i(t)$ connue à l'avance sur la période d'étude $[0, T]$.

La fabrication de ces n produits finis nécessite l'emploi de r matières premières non périssables. La i -ème matière première arrive dans le système avec un flux $v_i(t)$ connu à l'avance sur la période $[0, T]$ et elle est conservée dans un stock qui lui est propre dont le niveau est noté $S_i(t)$. On appelle q_{ij} la consommation marginale de la machine (j) en matière première (i) , c'est-à-dire que si la machine (j) fonctionne

pendant la période infinitésimale $[t, t+dt]$ avec l'intensité $u_j(t)$, alors elle consommera une quantité égale à $q_{ij} u_j(t)dt$ de la i -ème matière première pendant cette période.

On pourrait aisément ajouter au modèle une consommation de matières premières périssables (électricité,...) en ajoutant le coût de leur emploi au critère (15).

Ce modèle de gestion de la production par les stocks, schématiquement représenté par la figure 1, est la version légèrement simplifiée d'un modèle réel (qui nous a été fourni par la SEMA) d'une entreprise de coloriage de fils de laine (les matières premières sont des fils blancs de différentes épaisseurs et les produits finis sont des fils de différentes épaisseurs et de différentes couleurs).

Pour condenser les notations introduites ci-dessus, nous noterons s, u, d les vecteurs de dimension n de composantes s_i, u_i, d_i et S et v les vecteurs de dimension r de composantes S_i et v_i . Enfin, nous noterons M la matrice diagonale de dimension n et d'éléments m_i et Q la matrice (r,n) d'éléments q_{ij} .

On vérifiera alors aisément (cf. [3] ou [5]) que l'évolution du niveau des stocks est décrite par les deux systèmes d'équations différentielles :

$$(13a) \quad \frac{ds}{dt} = M u(t) - d(t)$$

$$(13b) \quad \frac{dS}{dt} = v(t) - Q u(t)$$

On suppose évidemment que l'état initial des stocks est connu :

$$(13c) \quad s(0) = s_0 \quad ; \quad S(0) = S_0$$

Par définition, les variables de décision u_i sont contraintes par :

$$(13d) \quad 0 \leq u_i(t) \leq 1 \quad , \quad i=1, \dots, n \quad , \quad \text{pour presque tout } t \in [0, T]$$

On ne peut fabriquer de produit fini en l'absence des matières premières nécessaires, d'où :

$$(13e) \quad S_i(t) \geq 0 \quad , \quad i=1, \dots, r \quad , \quad \text{pour presque tout } t \in [0, T]$$

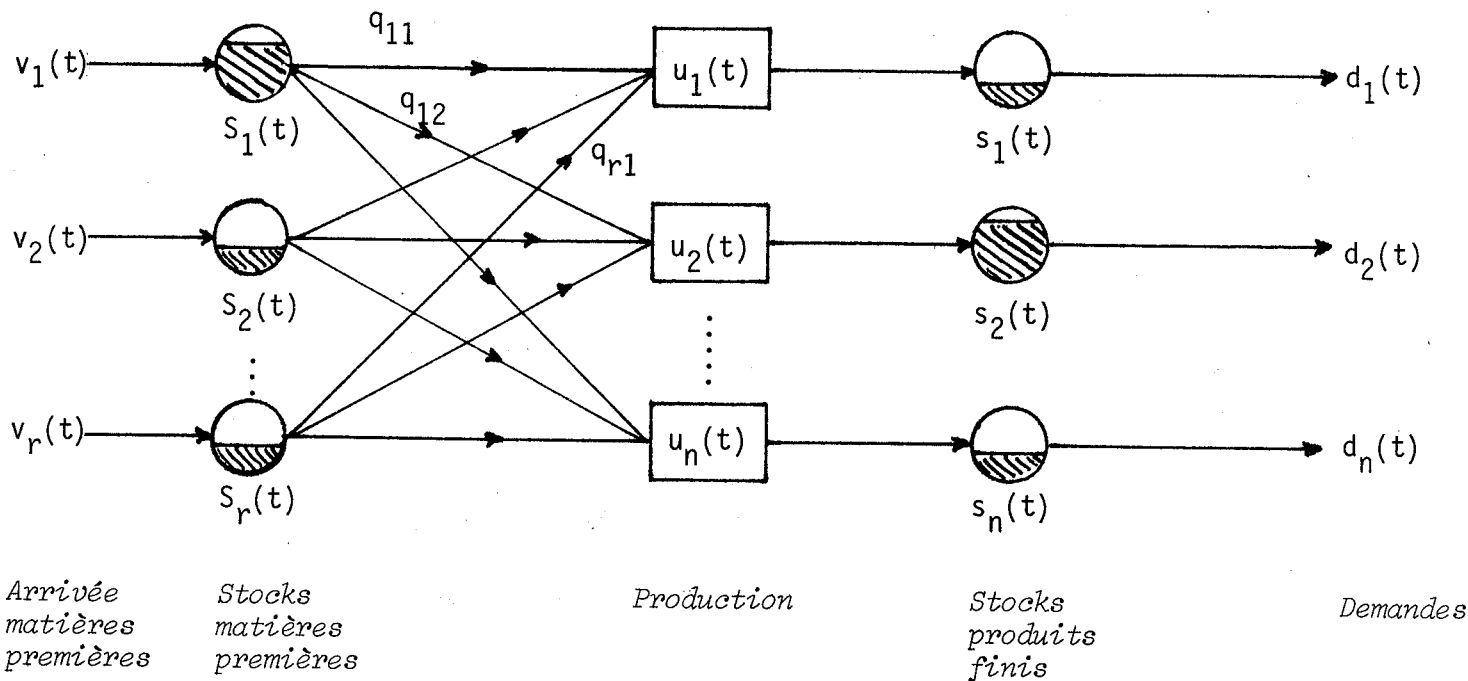
Nous supposons que les stocks de produits finis sont bornés supérieurement et inférieurement :

$$(13f) \quad a_i \leq s_i(t) \leq b_i \quad , \quad i=1, \dots, n \quad , \quad \text{pour presque tout } t \in [0, T]$$

où b_i représente la capacité maximale de stockage du i -ème produit fini et où a_i , qui est négatif, représente la rupture de stock maximale autorisée. En effet, dans notre modèle, la demande est forcément satisfaite. Lorsque le stock d'un produit est physiquement vide, cela signifie seulement que la demande sera satisfaite plus tard. On appelle alors $s_i(t)$ la valeur (avec le signe moins) représentant le total des demandes non satisfaites à l'instant t .

Enfin, pour que la situation à l'instant final T ne soit pas détériorée, nous imposerons aux stocks de matières premières et de produits finis d'avoir des valeurs fixées à l'avance en fin de période :

$$(13g) \quad s(T) = s_f \quad ; \quad S(T) = S_f$$



Utilisant la fonction :

$$(14) \quad \rho(x) = [\max(x, 0)]^2$$

nous définirons le coût total par :

$$(15) \quad J = \sum_{i=1}^n \int_0^T \{CP_i u_i^2(t) + CS_i \rho[s_i(t)] + CR_i \rho[-s_i(t)]\} dt$$

où les coefficients CP_i , CS_i et CR_i , tous positifs, représentent respectivement les coûts marginaux de production, de stockage et de rupture de stock.

D'après les conditions (13c) et (13g), les stocks initiaux et finaux sont fixés, ce qui implique que la production totale sur la période $[0, T]$ est fixée. On en déduit (cf. [3] ou [5]) que minimiser la norme dans L^2 des u_i revient à minimiser l'écart des u_i à leur moyenne, donc à les lisser. Les coefficients CP_i de (15) sont donc des coefficients de lissage de la production.

En ce qui concerne le stockage, nous avons choisi des coûts marginaux proportionnels à la valeur absolue du stock, avec des coefficients de proportionnalité différents suivant que le stock est positif (coût marginal de stockage $CS_i s_i(t)$) ou négatif (coût marginal de rupture de stock $CR_i |s_i(t)|$), les deux étant combinés dans la formule (15) grâce à la fonction (14).

On pourrait bien entendu utiliser les méthodes proposées en 4.2 avec une fonction coût beaucoup plus complexe que celle proposée en (15) et introduire, en particulier, des coûts de stockage des matières premières. Rappelons que nous exposons seulement ici l'exemple du problème de l'usine de coloriage de fils de laine.

Le problème posé est de minimiser le critère (15) compte tenu de (13). Il faut noter qu'à cause des contraintes sur l'état (13e) et (13f), le principe du maximum n'est pas applicable au modèle (13), (15). Nous avons donc utilisé la méthode de pénalisation. Le critère pénalisé J_ϵ est alors défini par :

$$(16) \quad g(s, S, u) = \sum_{i=1}^n [\rho(-u_i) + \rho(1-u_i) + \rho(a_i - s_i) + \rho(s_i - b_i)] \\ + \sum_{i=1}^r \rho(-S_i)$$

$$(17) \quad J_\epsilon = J + \frac{1}{\epsilon} \int_0^T g[s(t), S(t), u(t)] dt$$

et le problème pénalisé P_ϵ est la minimisation du critère (17) sous les équations d'état (13a) (13b) et les conditions aux limites (13c) et (13g) mais sans contraintes. Le principe du maximum est, cette fois, parfaitement applicable et l'on trouvera en [3] ou [5] les détails pratiques de sa mise en oeuvre.

L'étude théorique brièvement exposée au paragraphe 4.2 de cette thèse nous permet d'affirmer que le problème initial (13), (15) a une solution unique u^* , que le problème pénalisé P_ϵ a une solution unique u_ϵ pour tout $\epsilon > 0$ et que la suite $\{u_\epsilon\}$ converge vers u^* dans $L^2(0, T; \mathbb{R}^n)$ fort (*) lorsque ϵ tend vers zéro.

Lorsque l'on cherche à résoudre numériquement ce problème de contrôle optimal il faut bien, pour résoudre les équations différentielles (13a), (13b), discrétiser le temps ; en pratique, on calculera une solution approchée $u_{h\epsilon}$ qui dépend à la fois du paramètre ϵ de pénalisation et du paramètre h de discrétisation du temps. Nous avons constaté empiriquement que pour que $u_{h\epsilon}$ approche u^* , il est nécessaire que l'erreur de discrétisation soit "négligeable" devant l'erreur de pénalisation, d'où une sorte de "condition de stabilité de la forme h/ϵ borné" rappelant l'intégration numérique des équations aux dérivées partielles évolutives.

(*) on a la convergence forte du contrôle parce que le critère est quadratique par morceaux. Dans le cas général du modèle linéaire et du critère convexe, on n'a que la convergence faible du contrôle.

En pratique, cela implique que si l'on désire que les contraintes (13d), (13e), (13f) soient vérifiées avec une grande précision (d'où le choix d'un "petit" ϵ), alors il faut intégrer les équations d'état (13a), (13b) avec une précision au moins aussi grande (d'où le choix d'un "petit" h).

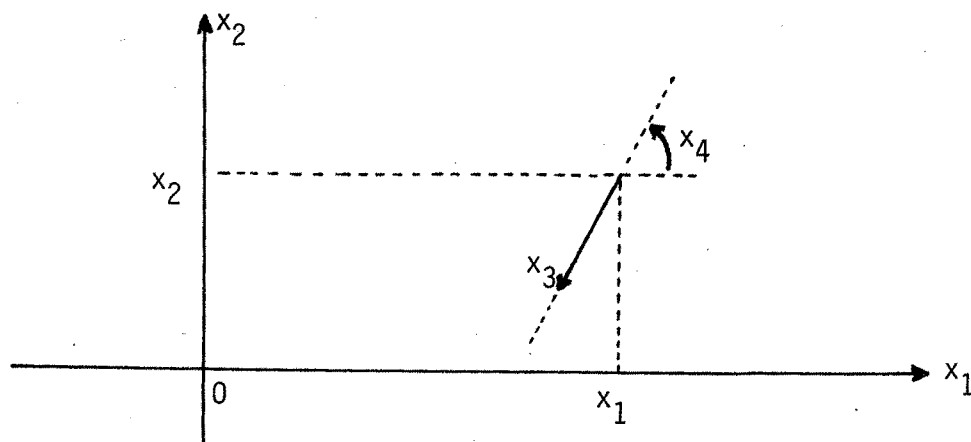
Ces questions concernant les aspects pratiques de l'application de la méthode de pénalisation au contrôle optimal des systèmes gouvernés par des équations différentielles sont discutées en détails dans [3] et [5].

On trouvera également dans [3] une comparaison, en application au modèle ci-dessus, de la pénalisation avec deux autres méthodes : la programmation dynamique et la programmation mathématique (discrétisation directe du modèle (13) - (15) et application de l'algorithme d'Uzawa). Disons seulement ici que la programmation dynamique donne ici des résultats déplorables et que la programmation mathématique, si elle donne ici des résultats comparables à la pénalisation, est moins générale (l'algorithme d'Uzawa suppose la convexité du modèle).

7. LE TEMPS OPTIMAL.

Commençons l'exposé par l'étude de la version simplifiée d'un modèle mis au point par Balakrishnan dans le cadre d'un contrat avec l'US Air Force. Il est clair que si, dans la réalité, les équations du modèle sont plus complexes, cela ne change rien à la nature du problème et aux méthodes de résolution.

Soit donc un avion volant dans un plan vertical et se préparant à atterrir. Le bout de la piste d'atterrissage étant choisi comme axe de coordonnées, l'axe Ox_1 étant la piste d'atterrissage, la position géométrique de l'avion est définie par son abscisse x_1 et par son altitude x_2 . Sa vitesse, qui est dans un plan vertical, est définie par son module x_3 et par l'angle x_4 que fait le vecteur vitesse avec le plan horizontal.



L'état de l'avion est alors défini par le vecteur $x = \{x_1, x_2, x_3, x_4\}$. On peut agir sur cet état d'une part par u_1 , variation instantanée du module du vecteur vitesse (manette des gaz, volets de freinage, etc....) et d'autre part, par u_2 , variation instantanée de l'angle x_4 (manche à balai). Le contrôle $u = \{u_1, u_2\}$ et l'état x étant normalisés, l'équation d'état de l'avion est (de façon très approchée) :

$$(18a) \quad \frac{dx_1}{dt} = -x_3(t) \cos [x_4(t)] \quad ; \quad \frac{dx_2}{dt} = -x_3(t) \sin [x_4(t)]$$

$$(18b) \quad \frac{dx_3}{dt} = u_1(t) \quad ; \quad \frac{dx_4}{dt} = u_2(t)$$

$$(18c) \quad x(0) = x_0$$

Le vol étant soumis aux contraintes :

$$(19) \quad -1 \leq u_i(t) \leq +1, \quad i=1,2, \quad \text{p.p. } t \in [0, N]$$

où N est un majorant du temps d'atterrissage.

Le vol est terminé à la date T si à cet instant la position géométrique de l'avion est le bout de la piste, $x_1(T) = x_2(T) = 0$, et si sa vitesse est nulle, $x_3(T) = 0$; de plus, l'avion doit atterrir tangentiellement à la piste, d'où $x_4(T) = 0$. Finalement, la condition sur l'état final est :

$$(20) \quad x(T) = 0$$

Le problème posé est de faire atterrir l'avion le plus vite possible, c'est-à-dire de minimiser T tel que (20) soit réalisé, compte tenu de (18) et (19).

Ce n'est pas un problème dont la résolution numérique est classique, dans la mesure où la date T , à laquelle la condition sur l'état final (20) doit être vérifiée, est a priori inconnue, ce qui empêche d'utiliser de façon standard la méthode de l'état adjoint.

Nous appellerons $u \rightarrow x(u)$ l'application de $U = L^2(0, N; \mathbb{R}^2)$ dans $X = L^2(0, N; \mathbb{R}^4)$ l'application équivalente à l'équation d'état (18), où N est un majorant du temps d'atterrissage.

Pour tout $T \geq 0$, nous appellerons $U(T)$ l'ensemble des contrôles u vérifiant (19) et tels que l'état associé $x(u)$ vérifie (20), i.e. tels que $x(T; u) = 0$. On voit que $U(T)$ est l'ensemble des contrôles permettant de faire atterrir l'avion en un temps T . L'avion pouvant rester immobile au bout de la piste après l'atterrissage, il est clair que :

$$(21) \quad u \in U(T_1) \implies u \in U(T_2), \quad \forall T_2 \leq T_1$$

ce qui implique en particulier :

$$(22) \quad U(T_1) \subset U(T_2), \quad \forall T_2 \leq T_1,$$

relation fondamentale pour notre algorithme de résolution

On vérifiera que le problème du temps optimal, noté problème P^* , peut se formuler :

Problème P^* :

a) trouver T^* tel que:

$$(23a) \quad U(T) = \emptyset, \quad \forall T < T^*$$

$$(23b) \quad U(T^*) \neq \emptyset$$

b) trouver un élément quelconque de $U(T^*)$.

Soient maintenant V l'ensemble des $u \in U$ vérifiant les contraintes (19) et $u \rightarrow J(T,u)$ la fonctionnelle :

$$(24) \quad J(T,u) = \|x(T;u)\|^2$$

On définit le problème $Q(T)$, où T est un paramètre :

Problème $Q(T)$: trouver $u_T \in V$ tel que :

$$(25) \quad J(T, u_T) \leq J(T, u), \quad \forall u \in V$$

On montre aisément que ce problème a au moins une solution pour $T \geq 0$ et que :

$$(26) \quad J(T, u_T) = 0 \iff u_T \in U(T) \iff U(T) \text{ n'est pas vide}$$

Puisque T est un paramètre, le problème $Q(T)$ est un problème à temps final fixé classique qui peut être résolu par une des méthodes exposées au paragraphe 4.2 de cette thèse.

Nous allons montrer comment on peut résoudre le problème du temps optimal P^* en résolvant une suite de problèmes $Q(T)$.

N étant un majorant du temps d'atterrissage, $U(N)$ n'est pas vide et la résolution de $Q(N)$ donne un élément $u_0 \in U(N)$. Par ailleurs,

il est évident que $U(0)$ est vide (l'avion n'est pas au bout de la piste au départ) sans quoi le problème n'aurait pas de sens. On pose $S_0 = 0$ et $T_0 = N$. Au début on connaît donc un couple S_0, T_0 avec $U(S_0)$ vide, $U(T_0)$ non vide et l'on connaît $u_0 \in U(T_0)$.

Supposons qu'au début de l'étape k de l'algorithme, on connaisse un couple S_{k-1}, T_{k-1} avec $U(S_{k-1})$ vide, $U(T_{k-1})$ non vide et que l'on connaisse $u_{k-1} \in U(T_{k-1})$. On pose :

$$(27) \quad \hat{T} = \frac{1}{2}(S_{k-1} + T_{k-1})$$

On résout le problème $Q(\hat{T})$ et soit $u_{\hat{T}}$ l'une de ses solutions. Deux cas peuvent se présenter :

a) $J(\hat{T}, u_{\hat{T}}) \neq 0$. Alors, d'après (26), $U(\hat{T})$ est vide. On pose $S_k = \hat{T}$, $T_k = T_{k-1}$ et $u_k = u_{k-1}$.

b) $J(\hat{T}, u_{\hat{T}}) = 0$. Alors, d'après (26), $u_{\hat{T}} \in U(\hat{T})$. On pose $S_k = S_{k-1}$, $T_k = \hat{T}$ et $u_k = u_{\hat{T}}$.

A la fin de l'étape k , on a bien un couple $\{S_k, T_k\}$ avec $U(S_k)$ vide, $U(T_k)$ non vide et l'on connaît $u_k \in U(T_k)$. Par construction, on a aussi $T_k - S_k = (T_{k-1} - S_{k-1}) / 2$. D'où l'on déduit $T_k - S_k = N / 2^k$ et $T_k - S_k \rightarrow 0$ quand $k \rightarrow \infty$. On montre alors (cf. ci-dessous) :

Théorème 4.

la suite $\{u_k\}$, construite par l'algorithme ci-dessus est l'union de sous-suites $\{u_k^j\}$, chacune d'entre elles convergeant dans U fort vers l'une des solutions du problème du temps optimal (23) lorsque $k \rightarrow \infty$.

□

On notera qu'à chaque étape de l'algorithme, on connaît un plan de vol admissible permettant d'atterrir en un temps T_k et l'on sait que $T_k \leq T^* + N/2^k$. On peut donc arrêter l'algorithme dès que l'erreur $N/2^k$ est jugée suffisamment faible.

Nous avons démontré le théorème 4 (cf. [5]) dans le cas où :

- a) l'équation d'état (18) est une équation différentielle générale de dimension n ,
- b) les contraintes (19) sont générales, i.e. le contrôle, l'état et le temps peuvent apparaître explicitement dans leur expression, à la condition toutefois que ces contraintes définissent un domaine de contrôles admissibles qui soit borné,
- c) la condition sur l'état final (20) peut avoir la forme générale $x(T) \in A$, où A est un fermé de $\mathbb{R}^n$.

Nous avons démontré le théorème 4 dans le cas général d'une part et dans le cas "équation d'état linéaire - contraintes convexe" d'autre part avec, dans les deux cas, des hypothèses de régularité du modèle tout à fait semblables à celles du paragraphe 4.2 de cette thèse.

On vérifiera (cf. [5], p. 201) que la résolution numérique du problème du temps optimal pour les systèmes gouvernés par des équations différentielles a été très peu étudiée, que la solution élégante du changement de variable (cf. [5], pp. 199-201) pose des problèmes numériques considérables et que la méthode décrite ici (*) ainsi que les résultats théoriques de convergence sont originaux.

(*) exposée sur l'exemple concret du problème d'atterrissage d'avion, la méthode se généralise aisément (cf. [5]).

8. RESOLUTION ET OPTIMISATION DE MODELES MACROECONOMIQUES.

8.1. Introduction.

Nous renvoyons le lecteur intéressé à [13] et [18] pour une étude bibliographique assez complète de cette question. Le travail présenté ici a été volontairement limité à la résolution et à l'optimisation de modèles macroéconomiques non-linéaires, déterministes et traités comme un seul bloc. Des extensions, en particulier au cas de modèles à plusieurs blocs sont actuellement en cours d'étude.

Nous avons étudié une dizaine de modèles macroéconomiques utilisés par les administrations française et américaine et une demi-douzaine de logiciels spécialisés dans la modélisation économique.

Nous avons tout d'abord constaté une augmentation régulière de la taille des modèles; par exemple, DMS, l'un des modèles les plus récents construits par l'INSEE, comporte 1263 équations non linéaires pour chaque période de temps. Dans ces conditions, on conçoit aisément l'importance pratique de méthodes efficaces de résolution et d'optimisation de modèles macroéconomiques.

Nous avons constaté ensuite que les logiciels spécialisés résolvent les modèles à l'aide de packages généraux de recherche opérationnelle et que l'optimisation est quasi- inexistante (*).

Nous avons constaté enfin l'absence quasi-totale, jusqu'à une date très récente, de recherches pluridisciplinaires dans ce domaine, c.a.d. rassemblant des économistes et des mathématiciens.

(*) une méthode d'optimisation démodée (Davidon-Fletcher-Powell) pour le

Le résultat est que ces logiciels spécialisés ne sont pas adaptés à la spécificité des modèles macroéconomiques. Ceci est bien perçu aujourd'hui et au congrès de 1977 du National Bureau of Economic Research une session entière a été consacrée à la résolution et à l'optimisation de modèles macroéconomiques, avec trois papiers (Drud, Lasdon, et Népomiastchy - Ravelli [8]) portant sur l'adaptation de la méthode de Newton à la structure particulière qu'ont les matrices de Jacobi du système (matrices très creuses et quasi-triangulaires).

Les travaux sur la résolution et l'optimisation de modèles macroéconomiques cités en référence ([7], [8], [12], [13], [18] et [19]) ont été effectués par Gabay, Népomiastchy, Rachdi, Ravelli et Rechenmann, chercheurs d'un projet IRIA - Laboria placés sous la responsabilité scientifique de l'auteur de cette thèse. Cependant, les résultats présentés ici sont personnels, sauf mention explicite du contraire.

8.2. Structure du modèle.

Pour clarifier l'exposé, aucune distinction ne sera faite sans la structure du modèle entre les variables prédéterminées (prologue) et post-déterminées (épilogue) d'une part (*) et les variables du coeur d'autre part (**). La présentation complète de nos méthodes est faite dans [18].

Soit un modèle dynamique en temps discret de périodes $t=1, \dots, T$. Soient n le nombre de variables endogènes, x_t^i la valeur à t de la i ème variable endogène (***) et x_t le vecteur $\{x_t^1, \dots, x_t^n\}$. Soient r le nombre de variables de contrôle, u_t^i la valeur à t de la i ème variable de contrôle et u_t le vecteur $\{u_t^1, \dots, u_t^r\}$. Soient p et q les retards maximaux sur les variables endogènes et de contrôle (respectivement).

(*) calculées en résolvant un système triangulaire.

(**) calculées en résolvant un système d'équations simultanées (c.a.d. non triangulaire).

(***) ou variable d'état.

On suppose alors que le modèle est défini par le système :

$$(28a) \quad x_t^i = f_t^i(x_t, x_{t-1}, \dots, x_{t-p}, u_t, u_{t-1}, \dots, u_{t-q}) ,$$

$$i=1, \dots, n \quad \text{et} \quad t=1, \dots, T,$$

$$(28b) \quad x_t^i \text{ donné pour } i=1, \dots, n \quad \text{et} \quad t \leq 0$$

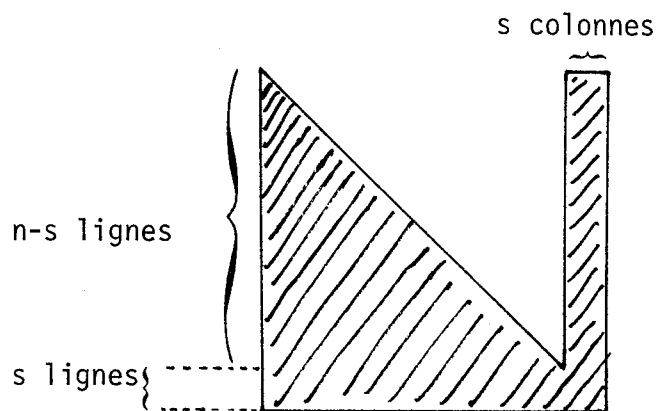
On suppose que les fonctions f_t^i sont continûment différentiables et que, au moins pour toute valeur "raisonnable" du contrôle u^* , le système (28) a une solution x et une seule. Lors de la résolution du modèle pour la période t , le vecteur :

$$(29) \quad d_t = \{x_{t-1}, x_{t-2}, \dots, x_{t-p}, u_t, u_{t-1}, \dots, u_{t-q}\}$$

est connu et le problème est de trouver x_t vérifiant le système :

$$(30) \quad x_t^i = f_t^i(x_t, d_t) \quad , \quad i=1, \dots, n$$

A l'aide d'un algorithme dû à Ravelli et exposé en détails dans [19], on peut renuméroter les variables et les équations de (30) pour mettre (30) sous une forme quasi-triangulaire. Plus précisément, le jacobien de (30) aura la forme :



où seules les parties non identiquement nulles sont hachurées. En d'autres termes, on appellera variables de bouclage le vecteur $y_t = \{x_t^{n-s+1}, \dots, x_t^n\}$

(*) cette hypothèse est précisée et discutée dans [18].

des s dernières variables du modèle et équations de bouclage les s dernières équations du modèle. On voit d'après la figure que le système des $n-s$ premières équations est triangulaire pour un y_t donné, c'est-à-dire que les $n-s$ premières variables se déduisent de y_t par simple application successive des formules (30). On verra ci-dessous que plus le système est triangulaire, c.a.d. plus le nombre s de variables de bouclage est petit, et plus le système (30) est facile à résoudre.

On trouvera ci-dessous le nombre de variables de bouclage, trouvées à l'aide de l'algorithme de Ravelli, pour plusieurs modèles réels :

<i>Modèle</i>	<i>Origine</i>	<i>Nombre d'équations (n)</i>	<i>Nombre de variables de bouclage (s)</i>
PIMPON	Minist. Economie (DP)	14	2
FAIR	Univ. Yale (USA)	83	6
STAR	INSEE / DP	139	3
MPS	Fed. Res. Bank (USA)	400	20

On voit sur ce tableau que les modèles réels peuvent être mis sous une forme très quasi-triangulaire, ce qui justifie les méthodes décrites ci-dessous.

Le temps de calcul de l'algorithme de Ravelli est de 4,14 secondes pour le modèle FAIR et de 47,20 secondes pour le modèle MPS sur un ordinateur IBM 370 - 168. Bien que croissant rapidement avec le nombre d'équations, ces temps de calcul sont négligeables. En effet, cette renumérotation n'est faite qu'une seule fois, au moment de la construction du modèle. Valable pour toutes les simulations ultérieures, la renumérotation n'est même pas remise en cause par une réestimation du modèle, pourvu que la forme analytique des équations ne soit pas modifiée.

Une fois la renumérotation faite, et par définition du vecteur y_t des variables de bouclage, le système (30) s'écrit :

$$(31a) \quad x_t^i = f_t^i(x_t^1, \dots, x_t^{i-1}, y_t, d_t) \quad , \quad i=1, \dots, n-s$$

$$(31b) \quad x_t^i = f_t^i(x_t, d_t) \quad , \quad i=n-s+1, \dots, n$$

où le système (31a) est donc triangulaire inférieur pour un y_t donné.

8.3. Résolution du modèle.

On a vu que (31a) équivaut à une application $y_t \rightarrow x_t(y_t)$ où $x_t(y_t)$ est calculé par simple évaluation successive des fonctions f_t^i . Evidemment, avec une valeur quelconque de y_t , $x_t(y_t)$ ne vérifie pas les équations de bouclage et il y a une erreur, qui ne dépend que de y_t , que nous noterons $\varphi_t(y_t)$. Le problème est donc de trouver y_t de telle sorte que cette erreur soit nulle :

$$(32) \quad \varphi_t^i(y_t) = 0 \quad , \quad i=1, \dots, s$$

Il est théoriquement possible d'éliminer analytiquement les variables x_t^i , $i \leq n-s$ des équations (31) afin d'obtenir les expressions analytiques des fonctions φ_t^i . En pratique cependant, cela est irréalisable pour les modèles dépassant la vingtaine d'équations. Par contre, à l'aide de (31), on peut calculer la valeur numérique de $\varphi_t(y_t)$ pour toute valeur numérique donnée du vecteur y_t . Dans ce cas, les dérivées partielles des fonctions φ_t^i ne peuvent être calculées analytiquement et, par conséquent, la méthode la plus simple de résolution du système (32) est l'algorithme de Gauss-Seidel qui n'utilise pas les dérivées de φ .

Il faut remarquer que l'algorithme de Gauss-Seidel, appliqué au système (32) de dimension s ($s=3$ pour STAR) et non au système initial (30) de dimension n ($n=139$ pour STAR), est adapté à la structure quasi-triangulaire du modèle. Cependant, cette méthode a les défauts habituels des algorithmes de Gauss-Seidel, à savoir :

a) la convergence est lente (d'ordre 1) ;

b) la convergence dépend fortement de l'ordre choisi pour les équations, de la normalisation des variables et des paramètres de relaxation (de frottement) choisis pour le bouclage ; tous ces paramètres (s'ils sont mal choisis, la méthode peut diverger) sont déterminés empiriquement par un grand nombre d'essais au hasard.

Nous proposons de résoudre (32) à l'aide de la méthode de Newton. Celle-ci a une convergence rapide (d'ordre 2), n'a pas de paramètres de relaxation à déterminer et sa convergence est assurée pourvu que l'approximation initiale de x_t soit "bonne", ce qui est le cas ici avec comme approximation initiale la solution du modèle pour $t-1$. Pour $t=1$ (première période de simulation), l'approximation initiale est x_0 , condition initiale du modèle connue d'après (28b).

L'application de la méthode de Newton directement au système (30) implique à chaque itération la résolution d'un système linéaire de dimension n , ce qui donne des temps de calcul désastreux pour la simulation de STAR (18 secondes dans les mêmes conditions que le test de comparaison de Gauss Seidel et Newton présenté ci-dessous).

Appliquant la méthode de Newton au système (32), nous ne devons résoudre à chaque itération qu'un système linéaire de dimension s (avec $s=3$ pour STAR). La seule difficulté est donc le calcul des dérivées partielles des fonctions φ_t^i . Si l'expression analytique de ces dérivées est inaccessible, par contre, partant du code de Fortran des fonctions f_t^i définissant le modèle, Rechenmann a écrit (en Formac) un programme qui génère automatiquement le code Fortran des expressions analytiques des dérivées de f_t^i , ce qui permet le calcul des $\partial \varphi_t^i / \partial y_t^j$. En effet, par définition de φ , on a :

$$(33) \quad \varphi_t^i(y_t) = y_t^i - f_t^{n-s+i}[x_t(y_t), d_t] \quad , \quad i = 1, \dots, s$$

avec $y_t^i = x_t^{n-s+i}$. On déduit de (31a) :

$$(34) \quad \frac{\partial x_t^i}{\partial y_t^j} = \sum_{k=1}^{i-1} \frac{\partial f_t^i}{\partial x_t^k} \frac{\partial x_t^k}{\partial y_t^j} + \frac{\partial f_t^i}{\partial y_t^j} \quad , \quad i = 1, \dots, n-s, j = 1, \dots, s$$

Notons que pour tout j , (34) est un système triangulaire par rapport aux variables $\partial x_t^i / \partial y_t^j$, $i = 1, \dots, n-s$. Dérivant (33), on a :

$$(35) \quad \frac{\partial \varphi_t^i}{\partial y_t^j} = \delta_{ij} - \sum_{k=1}^{n-s} \frac{\partial f_t^{n-s+i}}{\partial x_t^k} \frac{\partial x_t^k}{\partial y_t^j} - \frac{\partial f_t^{n-s+i}}{\partial y_t^j} \quad , \quad i, j = 1, \dots, s$$

Par conséquent, la matrice $((\partial x_t^k / \partial y_t^j))$ ayant été calculée en résolvant les s systèmes triangulaires (34), on déduit directement la jacobienne $((\partial \varphi_t^i / \partial y_t^j))$ du système (32) des formules (35).

Rappelons qu'il faut résoudre $\varphi_t(y_t) = 0$, où φ_t est donné par la formule (33). Nous avons donc choisi comme test d'arrêt :

$$(36) \quad \sum_{i=1}^s \left| \frac{f_t^{n-s+i} - y_t^i}{y_t^i} \right| \leq \epsilon$$

Pour faire les tests de comparaison, nous avons choisi le modèle STAR (139 équations) pour $T = 5$ périodes. Les méthodes retenues sont d'une part la méthode de Newton classique et d'autre part la méthode de Gauss-Seidel après optimisation des paramètres de relaxation, les deux étant appliquées au système "réduit" (32). La table ci-dessous donne le temps de calcul d'une simulation sur l'IBM 370-168 du C.I.R.C.E. :

$(\epsilon \text{ precision})$	<i>Gauss-Seidel</i>	<i>Newton</i>
10^{-2}	0,097 sec.	0,042 sec.
10^{-4}	0,344 sec.	0,067 sec.
10^{-7}	0,762 sec.	0,068 sec.

Nous avons donc prouvé d'une part la supériorité de Newton sur Gauss-Seidel (*) et d'autre part l'intérêt de la technique des variables de bouclage (18 secondes pour Newton appliqué directement à (30)).

(*) Supériorité particulièrement forte lorsque la précision demandée est

On trouvera dans [18] une comparaison beaucoup plus détaillée, le temps moyen de simulation étant estimé en fonction de la qualité de l'approximation initiale pour 5 méthodes du second ordre : 3 variantes de Newton, la méthode des moindres carrés et la méthode des sécantes, cette dernière programmée par Rachdi (*). Parmi ces méthodes, donnant des temps de calcul assez comparables, deux dominent (la méthode de Newton modifiée et la méthode des sécantes) sans qu'il soit possible de choisir entre elles.

8.4. L'optimisation du modèle.

Il s'agit de minimiser le critère :

$$(37) \quad j(x,u) = \sum_{t=1}^T j_t(x_t, \dots, x_{t-p}, u_t, \dots, u_{t-q})$$

où x et u sont liés par le modèle (28) et où j_t est supposé être continûment différentiable pour tout t .

Renvoyant le lecteur à [13] ou [18] pour la description des méthodes d'optimisation, nous allons exposer ici la méthode que nous proposons, la méthode de l'état adjoint, pour calculer le gradient réduit $J'(u)$ du critère $J(u) = j[x(u), u]$, où $x(u)$ est la solution du modèle (28) associé au contrôle u .

La méthode de l'état adjoint est connue pour l'optimisation des systèmes de la forme $dx/dt = f(x,t)$. Elle s'applique directement aux systèmes discrets de la forme $x_t = f_t(x_{t-1})$. Sa généralisation au cas $x_t = f_t(x_{t-1}, \dots, x_{t-p})$, en introduisant des variables d'état supplémentaires, est classique. Par contre, son application au modèle (28) qui n'est pas résolu en x_t et sans introduction de variables d'état supplémentaires est originale.

Introduisons les notations suivantes :

$$(38) \quad \delta_t = \begin{cases} 1 & \text{si } t \leq T \\ 0 & \text{si } t > T \end{cases}$$

(*) cf. M. RACHDI : "Quelques algorithmes de résolution de modèles macro-

$$(39) \quad z_t^i = \sum_{k=0}^p \delta_{t+k} \frac{\partial j_{t+k}^i}{\partial x_t^i} ; \quad w_t^i = \sum_{k=0}^q \delta_{t+k} \frac{\partial j_{t+k}^i}{\partial u_t^i}$$

$$(40) \quad F_t^k = \left(\left(\frac{\partial f_{t+k}^i}{\partial x_t^j} \right) \right) ; \quad G_t^k = \left(\left(\frac{\partial f_{t+k}^i}{\partial u_t^j} \right) \right)$$

Soit $\psi = \{\psi_1, \dots, \psi_T\}$, avec $\psi_t \in R^n$, la solution de :

$$(41) \quad \psi_t = z_t + \sum_{k=0}^p \delta_{t+k} (F_t^k)^* \psi_{t+k} , \quad t=1, \dots, T,$$

où $(F_t^k)^*$ est la matrice transposée de F_t^k . L'équation (41) est appelée système adjoint du problème d'optimisation et sa solution ψ est appelée état-adjoint. Nous avons démontré (cf. [13] ou [18]) que le gradient réduit $J'(u)$ peut être déduit de la connaissance de ψ par la formule :

$$(42) \quad J'_t(u) = w_t + \sum_{k=0}^q \delta_{t+k} (G_t^k)^* \psi_{t+k} , \quad t=1, \dots, T$$

Pour calculer le gradient, la seule difficulté est donc de résoudre le système adjoint (41), que l'on résout de façon rétrograde, c.a.d. pour $t=T, T-1, T-2, \dots, 1$. En effet, lors du calcul de ψ_t , le vecteur :

$$(43) \quad c_t = z_t + \sum_{k=1}^p \delta_{t+k} (F_t^k)^* \psi_{t+k} ,$$

qui ne fait intervenir que des $\psi_{t'}$, avec $t' > t$, est connu (*) et le calcul de ψ_t se ramène à la résolution du système linéaire :

$$(44) \quad \psi_t = c_t + (F_t^0)^* \psi_t$$

De la notation (40), et de la structure (31) du modèle, on déduit que (44) s'écrit :

$$(45a) \quad \psi_t^i = \sum_{j=i+1}^n \frac{\partial f_t^j}{\partial x_t^i} \psi_t^j + c_t^i , \quad i=1, \dots, n-s$$

$$(45b) \quad \psi_t^i = \sum_{j=1}^n \frac{\partial f_t^j}{\partial x_t^i} \psi_t^j + c_t^i , \quad i=n-s+1, \dots, n$$

(*) en particulier, pour $t=T$, on a simplement $c_t = z_t$.

Utilisant la même technique que pour résoudre le modèle, soit $\lambda = \{\psi_t^{n-s+1}, \dots, \psi_t^n\}$ le vecteur de dimension s des variables de bouclage adjointes. Pour tout λ donné, à l'aide de (45a), on peut calculer ψ_t^i , $i \leq n-s$ en résolvant un système linéaire qui est triangulaire supérieur (on calcule ψ_t^{n-s} , puis ψ_t^{n-s-1} et ainsi de suite jusqu'à ψ_t^1). On a alors une erreur sur les équations (45b), qui dépend uniquement de λ , notée $E(\lambda)$. D'après (45), il est clair que l'application $\lambda \rightarrow E(\lambda)$ est affine et peut s'écrire $E(\lambda) = A\lambda - b$. Le calcul de $E(0)$ donne b . Nous avons montré (cf. [13] ou [18]) que la matrice A n'est autre que la transposée de la matrice $((\partial \varphi_t^i / \partial y_t^j))$ calculée pour l'algorithme de résolution. Il suffit donc de résoudre le système linéaire $A\lambda = b$, qui est seulement de dimension s , pour calculer λ et en déduire le vecteur ψ à l'aide du système triangulaire (45a). Enfin, la formule (42) donne le gradient.

La méthode de l'état adjoint demande, pour un calcul du gradient $J'(u)$, $2T$ évaluations des fonctions (45a) et T résolutions d'un système linéaire de dimension s . Par ailleurs, la méthode des différences finies, généralement utilisée pour les modèles économiques, demande $rT^2/2$ résolution du système non linéaire (30) de dimension n avec, pour STAR, $r=10$, $n=139$ et $s=3$. On conçoit donc que la méthode de l'état adjoint est beaucoup plus efficace.

C'est ce qui apparaît dans la table ci-dessous, qui donne le temps de calcul sur IBM 370-168 d'un gradient du modèle STAR pour différentes valeurs du nombre T de périodes :

T	$M1$ (différences finies)	$M2$ (état adjoint)	$M1 / M2$
10	0,65 sec.	0,016 sec.	40,6
20	2,39 sec.	0,031 sec.	77
30	4,90 sec.	0,046 sec.	106,5

L'utilisation de la méthode de l'état-ajoint exige la connaissance des jacobiens F_t^k et G_t^k du modèle que nous avons calculés à l'aide du dérivateur formel écrit par Rechenmann et mentionné au paragraphe précédent.

9. CONCLUSION.

9.1. La méthode de pénalisation.

Le travail que nous présentons sur la méthode de pénalisation représente une étude assez complète de la pénalisation "classique", c'est-à-dire la pénalisation extérieure et quadratique qui associe à une contrainte $g(x) \leq 0$ la fonction de pénalité $G(x) = [\max(g(x), 0)]^2$.

A notre avis, les résultats théoriques de convergence de la méthode que nous avons obtenus sont suffisants, c'est-à-dire que nous n'avons jamais rencontré de problèmes concrets pour lesquels ces résultats ne sont pas applicables. Par ailleurs, nous ne pensons pas que ces résultats peuvent être améliorés. De même, nous pensons avoir fait à peu près le tour des aspects pratiques de l'application de la méthode de pénalisation classique à la résolution de problèmes concrets.

Etant la plus simple à mettre en oeuvre, la pénalisation classique est très fréquemment utilisée, d'autant plus qu'aucune autre pénalisation n'a encore été apportée la preuve concrète de sa supériorité. Nous espérons donc que les résultats et les algorithmes présentés ici seront utiles. Cependant, ce sujet ne se justifie pas, à notre avis, de développement ultérieurs. Il n'en est pas de même pour les autres formes de pénalisation.

En ce qui concerne le lagrangien augmenté, les travaux de Rockafellar et autres ont fait le tour des aspects théoriques, au moins pour la programmation mathématique. Sans doute quelques travaux sont encore nécessaires pour les systèmes gouvernés par des équations et des inéquations différentielles, mais il faut surtout apporter la preuve concrète que cette méthode est plus efficace que la pénalisation classique malgré l'augmentation du nombre de variables indépendantes qu'elle entraîne.

Il nous paraît beaucoup plus prometteur d'étudier la pénalisation linéaire qui associe à une contrainte $g(x) \leq 0$ la fonction de pénalité $G(x) = \max[g(x), 0]$. Comme la pénalisation classique, cette méthode n'introduit pas de variables supplémentaires et comme le lagrangien augmenté, elle présente l'avantage de donner la solution exacte pour tout ϵ inférieur à un certain ϵ_0 . Par exemple, si l'on minimise $J(x) = x^2$

sous la contrainte $x \geq 1$, la pénalisation classique donne la solution approchée $x_\epsilon = 1 / (1+\epsilon)$; alors que la pénalisation linéaire donne $x_\epsilon = \frac{1}{2\epsilon}$ pour tout $\epsilon \geq \frac{1}{2}$ mais donne la solution exacte $x_\epsilon = 1$ pour tout $\epsilon \leq \frac{1}{2}$.

La pénalisation linéaire a toujours été rejetée car la fonction de pénalité $G(x) = \max [g(x), 0]$ n'est pas différentiable, ce qui pose des problèmes pour l'optimisation du critère pénalisé. Cependant, les travaux nombreux récemment publiés sur l'optimisation non différentiable ont permis de mettre au point des algorithmes performants, en particulier, ceux de C. Lemaréchal. Il serait donc à notre avis très intéressant de tester ces algorithmes, après les avoir modifiés pour tenir compte de la non-différentiabilité très particulière de $G(x)$, en application à la pénalisation linéaire.

Notons que la démonstration des théorèmes 1 et 2 utilise seulement la continuité de la fonction de pénalité. La convergence de la méthode de pénalisation linéaire est donc acquise en programmation mathématique. Il n'en va pas de même pour les systèmes gouvernés par des équations différentielles, et dans ce domaine, tout reste à faire.

9.2. Résolution et optimisation de modèles macroéconomiques.

Les algorithmes proposés dans cette thèse ont déjà apporté la preuve de leur efficacité par rapport aux méthodes classiques pour des modèles à un secteur et ne dépassant pas 100 équations.

Leur application au modèle METRIC de l'INSEE (qui comporte 400 équations et 12 secteurs) est en cours et donnera vraisemblablement lieu à des développements de nos recherches sur la résolution et l'optimisation de modèles macroéconomiques.

L'augmentation de la taille des modèles n'exige pas seulement la recherche d'algorithmes mathématiques de plus en plus efficaces. En effet, tous les algorithmes efficaces (en fait, tous les algorithmes sauf Gauss-Seidel) supposent un certain nombre de manipulations analytiques du modèle (renumérotation des variables, calcul analytique des dérivées)

avant toute simulation. Pour des modèles de grande taille, ces manipulations analytiques doivent nécessairement être automatisées. Cela est possible grâce aux programmes de calcul formel écrits par Rechenmann dans le cadre du projet Moduléco dirigé par l'auteur de cette thèse (cf. [10], [11], [12], et [20]).

Il est probable par ailleurs que les idées que nous avons développées pour la résolution et l'optimisation, et en particulier les techniques de variables de bouclage et d'état-adjoint, pourront être utilisées dans deux directions opposées : en amont de la simulation, pour l'estimation simultanée des équations ; en aval de la simulation, pour l'analyse de sensibilité et la linéarisation.



AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 5 de l'arrêté du 16 Avril 1974,

VU les rapports de Messieurs :

- A. BENSOUSSAN, Professeur à l'Université de PARIS IX
- Cl. FOULARD, Professeur à l'Institut National
Polytechnique de GRENOBLE,
Directeur de l'Ecole Française de Papeterie
- R. TREMOLIERES, Maître de Conférences à l'Institut
d'Administration des Entreprises
- AIX EN PROVENCE -

Monsieur Pierre NEPOMIASTCHY

est autorisé à présenter une thèse en soutenance pour l'obtention du
grade de DOCTEUR D'ETAT, discipline SCIENCES.

Grenoble, le 25 Mai 1979

Le Président de l'U.S.M.G.

Le Président de l'I.N.P.G.



Ph. TRAYNARD
Président
de l'Institut National Polytechnique

