

UNIVERSITÉ SAÂD DAHLAB DE BLIDA

Faculté des Sciences

Département de Mathématiques

MÉMOIRE DE MAGISTÈRE

en Mathématiques

Spécialité : Recherche Opérationnelle

PROBLÈME DE LA COUVERTURE COMPTABLE EN FINANCE : MODÉLISATION ET TECHNIQUES DE RÉOLUTION

Par

BENKACI Azzeddine

devant le jury composé de :

F. Hannane	Professeur, USDB, Blida	Président
M. Moulai	Professeur, USTHB, Alger	Examineur
A. Derbala	Maître de conférence -A-, USDB, Blida	Examineur
A. Malki	Maître assistant -A-, USDB, Blida	Examineur
A. Mataoui	Analyste quantitative, REUTERS, France	Invité
S. Manseur	Maître de conférence -B-, USDB, Blida	Encadreur

Blida, mai 2010

RÉSUMÉ

Dans ce mémoire, nous nous intéressons au problème de la couverture comptable dont l'objectif est de réduire au minimum la sensibilité de prix du marché d'un portefeuille financier, aux variations futures des taux d'intérêt.

Le problème de la couverture comptable est formulé comme un problème d'optimisation combinatoire. Pour sa résolution, nous appliquons des méthodes duales d'optimisation de type lagrangien augmenté. L'algorithme d'*Uzawa* à pas constant est étudié, implémenté et testé.

Afin de réduire l'effort du calcul sans altérer énormément la qualité de la couverture, nous appliquons une méthode approchée, fondée sur une heuristique de type gloutonne.

Un programme informatique a été développé, et des techniques d'amélioration de performance ont été adaptées pour accélérer l'exécution du programme.

Nous reportons des résultats numériques, obtenus en testant ce programme sur un exemple type. Ces résultats confirment l'efficacité de cette approche.

Mots-clés :

optimisation non linéaire, programme quadratique, méthodes duales, gestion des risques, couverture comptable, sensibilité.

ABSTRACT

In this report, we are interested in the problem of hedge accounting in which the objective is to reduce the market value sensitivity of a financial portfolio to the future variations of interest rates.

The problem of hedge accounting is formulated as a combinatorial optimization problem. For the resolution, we apply the dual optimization methods of augmented Lagrangian. Uzawa's algorithm with constant step is studied, implemented and tested.

To reduce the effort of the calculation without altering enormously the quality of the hedge, we apply an approached method based on a greed heuristics.

A computer program dedicated to the resolution of the problem of hedge accounting is developed, and some techniques were adapted to accelerate the program execution.

We report numerical results obtained by testing this program on a sample problem of hedge. These results confirm the efficiency of this approach.

Keywords:

nonlinear optimization, quadratic program, dual methods, risk management, hedge accounting, sensitivity.

الإرتفاع المفاجئ لمنحنى معدل الفائدة في السوق المالي يشكل خطرا على المحافظ المالية للمستثمرين وذلك لانخفاض ثمنها السوقي، وهذا الخطر يسمى : "خطر معدل الفائدة"، وللتقليل من درجة هذا الخطر يقوم بعض المستثمرين بعملية التغطية على الثمن السوقي للمحافظ المالية.

التغطية لمعدل الفائدة هي عملية يقوم بها المستثمر المالي، وذلك للتخفيف من الخسارة المحتملة والناجمة عن تذبذب معدل الفائدة، غير أن استعمالها يتطلب دراسة شبة كاملة للسوق المالي وكذلك تطبيق بعض الطرق والتقنيات الرياضية المتطورة.

التمثيل الرياضي لمسألة التغطية، الذي نتناوله في هذه المذكرة، هو عبارة عن مسألة تقليل تديريي. والمسألة التحنانية عبارة عن مشكل التقليل التربياعي تحت قيود خطية. التركيبية الخاصة لهذه المسألة، تستدعي استعمال طرق ذات كفاءة عالية ومتينة وذلك لحل هذا المشكل.

الهدف من هذا البحث هو تطبيق الطرق الثنائية للتقليل باستعمال تقنيات دالة لاجرانج الموسعة، وذلك لمعالجة مسألة التغطية لمعدل الفائدة.

في هذا البحث تم تطوير و ضبط خوارزمية " إزاوا ذات خطوة ثابتة"، لحل مشكل التقليل التربياعي تحت قيود خطية. و نظرا لصعوبة معالجة مسألة التغطية، تم تطبيق طريقة تقريبية، وذلك للحصول على حلول ذات جودة عالية في مدة زمنية معقولة.

قمنا أيضا بإنشاء وتطوير برنامج حسابي يساعد على اتخاذ القرارات في هذا المجال، كما تم تطبيق بعض التقنيات التي تساعد على تسريع هذا البرنامج.

النتائج العددية المتحصل عليها بواسطة تطبيق الطرق المذكورة أعلاه، تؤكد كفاءة و فعالية الطريقة المتبعة لمعالجة المسألة.

REMERCIEMENTS

Tout d'abord, je tiens à remercier, Dieu Le Tout Puissant, pour m'avoir préservé et guidé vers la connaissance et le savoir.

Je tiens à exprimer toute ma reconnaissance à Monsieur *Salah Manseur*, Maître de conférence à l'université de Blida, pour m'avoir encadré et guidé tout au long de ce mémoire. Il m'a permis de réaliser ce travail en me manifestant sa confiance et en m'encourageant dans mon travail, et surtout pour avoir toujours été à l'écoute de mes problèmes.

Le travail avec Monsieur *Abdelaziz Mataoui*, Analyste Quantitatif - RFS, (France), a été un véritable plaisir, ses qualités scientifiques remarquables m'ont ouvert de nouvelles perspectives. Il m'a permis d'élargir le champ de mes connaissances, et je tiens à le remercier particulièrement pour son aide inestimable dans ce travail.

J'adresse mes remerciements à mon enseignant, Monsieur *Ali Derbala*, Maître de conférence à l'université de Blida, d'avoir fait parti du jury, et pour m'avoir fait partager ses connaissances, ainsi que pour ses nombreux conseils et ses idées précieuses.

Je suis très honoré par la présence au jury du Professeur, *Farouk Hannane*, de l'université Saâd Dahlab de Blida. Je le remercie très sincèrement d'avoir accepté d'être président du jury.

J'adresse mes remerciements à Monsieur le professeur, *Mustapha Moulai*, Chef de Département de la Recherche Opérationnelle à l'université des sciences de la technologie Houari Boumediene, qui m'a fait l'honneur d'accepter de juger mon travail.

Mes vifs remerciements à mon enseignant, Monsieur *Abderrahmane Malki*, Chargé de cours à l'université de Blida, d'avoir fait parti du jury. Avec lui j'ai beaucoup appris.

Je ne saurais terminer sans mentionner tout ce que je dois à mes parents et à mes frères, qui m'ont toujours soutenu et supporté. Sans eux ce travail n'aurait pu voir le jour.

Enfin, je remercie toutes les personnes qui m'ont aidé par leur soutien et leur confiance à accomplir ce modeste travail.

TABLE DES MATIÈRES

RÉSUMÉ	01
REMERCIEMENT	04
TABLE DE MATIÈRES	05
LISTE D'ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX	08
INTRODUCTION.....	10
CHAPITRE 1 : QUELQUES NOTIONS SUR LES INSTRUMENTS FINANCIERS	13
1.1 Introduction	13
1.2 Obligation classique	13
1.2.1 Flux monétaires et modalités de remboursement du capital	14
1.2.2 Actualisation des flux monétaires	17
1.2.3 Taux de rendement interne	19
1.3 Risque de taux d'intérêt	21
1.4 Portefeuille obligataire	25
1.5 La courbe des taux d'intérêt.....	26
1.6 Valorisation des obligations et risque de taux d'intérêt	31
1.6.1 Obligation à taux fixe	32
1.6.2 Obligation à taux variable	32
1.7 Swap de taux d'intérêt	33
1.7.1 Valorisation	35
1.7.2 Calcul de sensibilité	36
1.8 Scénarios de déformation de la courbe des taux.....	37
1.8.1 Reconstitution de la courbe des taux (zéro-coupons)	38
1.9 Conclusion	39
CHAPITRE 2 : MÉTHODES DUALES D'OPTIMISATION NON LINÉAIRE	40
2.1 Introduction	40
2.2 Problème de l'optimisation non linéaire	40
2.3 Notions fondamentales	41
2.4 Optimisation sans contraintes	43
2.4.1 Conditions d'optimalité	43
2.4.2 Méthodes numériques de gradient	44
2.5 Optimisation avec contraintes	47
2.5.1 Les conditions nécessaires d'optimalité de <i>Kuhn</i> et <i>Tucker</i>	47
2.5.2 Condition suffisante d'optimalité	48
2.5.3 Dualité lagrangienne classique	50

2.6	Méthodes de dualité.....	51
2.6.1	La méthode d' <i>Uzawa</i>	51
2.6.2	La méthode d' <i>Arrow Hurwicz</i>	51
2.6.3	Conditions de convergence des méthodes de dualité	52
2.7	Méthodes de pénalité.	52
2.7.1	Méthode de pénalité extérieure.....	53
2.7.2	Conditions de convergence des méthodes de pénalité.	54
2.7.3	Approximation des multiplicateurs de <i>Kuhn</i> et <i>Tucker</i> à l'optimum	55
2.8	Le lagrangien augmenté	55
2.8.1	Motivation	55
2.8.2	Méthodes des multiplicateurs	56
2.8.3	Méthodes de lagrangien augmenté	57
2.9	Application au problème quadratique à contraintes linéaires d'inégalité	59
2.9.1	Algorithme d' <i>Uzawa</i> à pas constant	61
2.9.2	Complexité de l'algorithme d' <i>Uzawa</i> à pas constant	62
2.9.3	Exemple et résultats numériques	65
2.9.4	Influence de choix du pas ρ sur le nombre d'itérations	66
2.10	Conclusion	69
CHAPITRE 3 : PROBLÈME DE LA COUVERTURE COMPTABLE		70
3.1	Introduction	70
3.2	Position et modélisation du problème	72
3.3	Extension du problème	78
3.3.1	Motivation	78
3.3.2	Procédure de sélection	79
3.3.3	Généralisation de la procédure de sélection.	79
3.4	Formulation du problème	80
3.5	Méthodes de résolution	81
3.5.1	Méthode exacte	81
3.5.2	Méthode approchée	84
3.6	Conclusion	93
CHAPITRE 4 : IMPLÉMENTATION ET TESTS		94
4.1	Introduction	94
4.2	Techniques d'amélioration	94
4.2.1	Réduction du nombre de contraintes du sous problème quadratique (<i>SQ</i>)	94
4.2.2	Construction des pools de prêts	96
4.2.3	Classement de swaps	97
4.3	Méthodologie de choix des paramètres	97
4.4	Programme de la couverture comptable	98

4.5	Mise en œuvre informatique	100
1.5.1	Exemple test	100
4.5.2	Initialisation des paramètres du programme	101
4.5.3	Simulation de la courbe des taux d'intérêt	102
4.6	Résultats expérimentaux	105
4.6.1	Les sensibilités des meilleurs instruments synthétiques	105
4.6.2	Effet du paramètre P sur la qualité de la couverture	108
4.6.3	Effet des paramètres k_{\max}, NC_{\max} sur le temps d'exécution	109
4.7	Conclusion	111
	CONCLUSION	112
	LISTE DES SYMBOLES ET DES ABRÉVIATIONS.	115
	RÉFÉRENCES	117
	ANNEXES	119

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

Figure 1.1 : Un prêt pour l'investisseur est une dette pour l'émetteur.....	13
Figure 1.2 : Représentation des flux monétaires d'une obligation ordinaire sur la flèche du temps.	14
Figure 1.3 : Différentes modalités de remboursement d'une obligation.....	16
Figure 1.4.a : Actualisation d'un seul flux monétaire.....	17
Figure 1.4.b : Actualisation d'une série de flux monétaires.....	17
Figure 1.5 : Évolution de prix d'une obligation ordinaire en fonction du taux d'actualisation ...	20
Figure 1.6 : Position de la <i>Duration</i> sur la flèche du temps.....	23
Figure 1.7 : Comparaison de deux obligations ayant la même <i>Sensibilité</i> et de <i>Convexité</i> différente.....	24
Figure 1.8 : Exemple de courbe des taux d'intérêt.....	27
Figure 1.9 : Relation entre le taux zéro-coupon et <i>forward</i>	30
Figure 1.10.a : Échange des flux monétaires d'un swap prêteur.....	34
Figure 1.10.b : Échange des flux monétaires d'un swap emprunteur.....	34
Figure 1.11 : Échange d'un revenu à taux fixe en revenu à taux variable.	37
Figure 1.12 : Les principales déformations futures de la courbe des taux.....	39
Figure 2.1.a : Ensemble convexe.....	41
Figure 2.1.b : Ensemble non-convexe.....	41
Figure 2.2.a : Fonction convexe.....	41
Figure 2.2.b : Fonction non-convexe.....	41
Figure 2.3 : Caractérisation de la convexité en terme du gradient.....	42
Figure 2.4 : Différentes formes de courbes de niveau d'une fonction f à deux variables.	46
Figure 2.5 : Illustration des conditions de <i>Kuhn</i> et <i>Tucker</i> sur un exemple à deux dimensions	48
Figure 2.6 : Illustration de la notion de point-selle.....	49
Figure 2.7 : Variation du nombre d'itérations (N_{iter}) en fonction de ρ	66
Figure 2.8 : ρ optimal théorique et expérimental en fonction de r	68
Figure 3.1 : Réduction de l'amplitude des variations de prix du portefeuille.....	72
Figure 3.2 : Schéma conceptuel de la couverture comptable.....	73
Figure 3.3 : Les différentes déformations de la courbe des taux d'intérêt.....	76
Figure 3.4 : Sélection optimale du sous-portefeuille \hat{H} , pour un swap S	79
Figure 3.5 : Exemple de couverture d'un portefeuille de prêts avec trois swaps.	80
Figure 3.6 : Schéma illustrant le principe de la méthode exacte pour un seul swap.	82
Figure 3.7 : Sélection du meilleur sous portefeuille \hat{H} pour le swap S	87

Figure 3.8 : Organigramme de la couverture comptable.....	91
Figure 4.1 : Construction des pools de prêts en terme de la <i>Duration</i>	97
Figure 4.2 : La courbe des taux d'intérêt de référence.	103
Figure 4.3 : Les différents scénarios de la courbe des taux.....	104
Figure 4.4 : Les sensibilités des instruments synthétiques en chaque période t (en années).	106
Figure 4.5 : Évolution de l'écart de la sensibilité, pour chaque swap, en fonction du paramètre P	108
Figure 4.6 : Évolution du temps d'exécution en fonction du paramètre k_{\max}	110
Figure 4.7 : Évolution du temps d'exécution en fonction du paramètre NC_{\max}	110
Tableau 1.1 : Répartition des flux monétaires d'obligation.....	20
Tableau 1.2 : Évaluation de la <i>VAN</i> pour différents taux de rendement.....	21
Tableau 1.3 : Les caractéristiques du portefeuille.....	25
Tableau 1.4 : Le taux de rendement d'un portefeuille composé de deux obligations.....	26
Tableau 1.5 : L'échange d'intérêts d'un swap prêteur.....	35
Tableau 1.6 : Évaluation d'un swap prêteur par la méthode obligataire.....	36
Tableau 2.1 : Illustration du calcul de la complexité temporelle de l'algorithme d' <i>Uzawa</i> à pas fixe (2.9).	64
Tableau 2.2 : Le nombre d'itérations (N_{iter}) en fonction du paramètre ρ , (cas : $r = 30$).....	66
Tableau 2.3 : Le nombre d'itérations (N_{iter}) en fonction du paramètre ρ , (cas : $r = 50$).....	66
Tableau 2.4 : ρ optimal (expérimental) en fonction de r	68
Tableau 4.1 : Caractéristiques des prêts.....	100
Tableau 4.2 : Caractéristiques des swaps	100
Tableau 4.3 : Initialisation des paramètres du programme.	101
Tableau 4.4 : Initialisation des paramètres de l'algorithme d' <i>Uzawa</i> à pas fixe (2.9).	101
Tableau 4.5 : Valeurs estimées des paramètres de la courbe des taux de référence.....	103
Tableau 4.6 : Valeurs estimées des taux d'intérêt de référence.	103
Tableau 4.7 : Valeurs estimées des paramètres pour les différents scénarios de la courbe des taux.	104
Tableau 4.8 : Les sensibilités des instruments synthétiques en chaque période t (en années).....	105
Tableau 4.9 : La valeur de l'écart de la sensibilité, pour chaque swap, et du ratio d'efficacité pour chaque scénario.	107
Tableau 4.10 : Les portions des prêts couverts en (%) pour chaque swap.	107
Tableau 4.11 : Valeurs des meilleurs écarts de la sensibilité, pour chaque swap en fonction du paramètre P	108
Tableau 4.12 : La valeur du ratio d'efficacité, pour chaque scénario, en fonction du paramètre P .	
Tableau 4.13 : Effet de k_{\max} sur le temps d'exécution effectif, avec ($NC_{\max} = 350$).....	110
Tableau 4.14 : Effet de NC_{\max} sur le temps d'exécution effectif, avec ($k_{\max} = 6$).....	110
Tableau 4.15 : Caractéristiques de la machine.	110

INTRODUCTION

La Recherche Opérationnelle (en abrégé *RO*), a pour but d'apporter une démarche scientifique à la résolution de problèmes de décision complexes issus du monde réel. Sa vocation scientifique est de construire des modèles formels d'aide à la décision, en particulier les modèles liés à des problèmes d'optimisation, et de proposer des méthodes de résolution efficace de ces modèles. La recherche opérationnelle intervient de manière cruciale dans des secteurs de plus en plus nombreux et diversifiés comme les transports, la production industrielle, la planification, l'informatique, l'administration, ..., etc.

Bien que le champ d'application traditionnel de la recherche opérationnelle demeure autour des métiers de l'industrie, de nombreux problèmes émergent dans les secteurs de la finance, de la banque et de l'assurance, faisant apparaître un besoin d'aide à la décision.

Dans ce travail, nous nous intéressons au problème de la couverture comptable d'un portefeuille de prêts bancaires par des swaps de taux d'intérêt. Ce problème est issu du domaine de gestion du risque bancaire.

La gestion du risque a toujours préoccupé les banques et les établissements financiers. Son importance remonte aux années quatre-vingt, lors de la faillite des caisses d'épargne américaines. Depuis cette période des avancées théoriques, technologiques et en termes de régulation prudentielle ont permis le développement de l'analyse et la gestion du risque. Ce développement a pu se réaliser grâce aussi aux outils d'aide à la décision qui sont mis à la disposition des décideurs.

Parmi les problèmes de gestion de risque, le problème de la réduction de l'écart entre la sensibilité de l'actif et du passif, due aux variations des taux d'intérêt. Ce problème offre plusieurs approches de réduction du risque.

La couverture comptable ([29], [32], [35]) est une approche qui consiste à réduire au minimum et pour un horizon de temps donné, le risque engendré par les fluctuations des taux d'intérêt, sur le prix du marché (la juste valeur). Les instruments financiers de couverture les plus utilisés en pratique sont les swaps de taux standard (*Plain Vanilla Swaps*) ([6], [8], [11]).

En effet, l'ajout séquentiel de ces instruments, aux portefeuilles financiers, offre aux gestionnaires la possibilité de modifier le prix du marché de leurs portefeuilles, ce qui leur permet de contrôler son niveau du risque et de l'adapter avec leurs besoins de financements.

Dans la littérature, nous avons trouvé une variété de problèmes de couverture. Ces problèmes ont été traités de manières différentes selon le type du modèle. On trouve quelques modèles stochastiques en temps continu dans ([33], [34], [36], [37]). Cependant, il est difficile de trouver des articles scientifiques ou spécialisées traitant ces problèmes par des modèles déterministes (voir [29]). Par ailleurs, dans la plupart des modèles qu'on a rencontré, le sous-problème de couverture a été modélisé comme un problème d'optimisation non-linéaire : « quadratique à contraintes linéaires ». Afin de le résoudre, plusieurs méthodes d'optimisation de type gradient ont été proposées.

Le problème de la couverture comptable a été déjà étudié par (A. Rossi, G. Bichisao et F. Campolongo) [29]. En comparaison, l'originalité de notre approche réside dans l'application d'une méthode duale basée sur les techniques de lagrangien augmenté ([15], [16], [17], [19]) pour la résolution du sous-problème. La mise en œuvre de cette méthode constitue une innovation technique dans la résolution du problème.

Dans ce mémoire, le problème est formulé comme un problème d'optimisation combinatoire dont le sous-problème est un problème d'optimisation continue, plus précisément : « un problème quadratique à contraintes linéaires d'inégalité ».

L'objectif de ce mémoire est de :

- Appliquer les méthodes duales d'optimisation, basées sur les techniques de lagrangien augmenté pour la résolution du problème.
- Élaborer un programme informatique pour la résolution effective du problème.

Les méthodes de lagrangien augmenté peuvent être obtenues en combinant les méthodes de dualité lagrangienne avec les méthodes de pénalité.

Le principe d'une méthode de dualité ([13], [15], [16], [19]) consiste à ramener la résolution du problème initial appelé « primal », à un autre problème d'optimisation plus facile à résoudre, appelé par opposition, problème « dual ».

Le principe d'une méthode de pénalité ([16], [19]) consiste à ramener la résolution du problème initial à une suite de problèmes d'optimisation sans contraintes.

Le contenu de ce mémoire est composé de quatre chapitres et d'une conclusion générale.

Le premier chapitre, présente quelques définitions sur les instruments financiers, en particuliers, les obligations (similaires aux prêts / emprunts) et les swaps de taux d'intérêt. Les principaux concepts de la valorisation et les mesures systématiques de risque de ces instruments sont rappelés ainsi que leur développement mathématique qui va nous être utile tout au long de ce mémoire.

Le deuxième chapitre est consacré à une étude succincte des méthodes de dualité et celles de pénalité, nous y rappelons les méthodes numériques de minimisation sans contraintes. Ensuite nous proposons la résolution du problème quadratique à contraintes linéaires d'inégalité, en appliquant une méthode duale basée sur les techniques de lagrangien augmenté. L'algorithme d'*Uzawa* à pas fixe, utilisant ces techniques, est implémenté et testé sur un exemple.

Dans le chapitre trois, nous présentons la formulation mathématique du problème de la couverture comptable, ensuite nous présentons aussi les méthodes de résolution, une méthode exacte pour les exemples de petite taille du problème, et une méthode approchée, basée sur une heuristique gloutonne, et qui est plus adaptée aux exemples de grande taille.

Le chapitre quatre est consacré à l'implémentation des techniques d'amélioration permettant d'accélérer la résolution du problème. Nous donnons aussi des résultats numériques qui ont été obtenus en testant ces techniques sur un exemple type du problème. Un listing complet en langage de programmation C++ est présenté en annexe-D.

Enfin, nous terminons notre travail par une conclusion générale.

CHAPITRE 1

QUELQUES NOTIONS SUR LES INSTRUMENTS FINANCIERS

1.1 Introduction

Un actif ou un instrument financier génère des flux monétaires (Cash flows) futures. Lorsque ces flux sont parfaitement connus, on parle d'actifs à revenu fixe (*Fixed-Income-Securities*), et cela désigne typiquement les obligations à taux fixe.

Les instruments financiers ([1], [5], [6], [7]) sont très divers et leur complexité s'accroît de jour en jour. Il est plus difficile de dresser une typologie de ces instruments. Dans ce chapitre, nous discutons des actifs de base (les obligations), et des produits dérivés (les swaps de taux d'intérêt). Nous présentons aussi leurs définitions et leur développement mathématique ([1], [2], [3]) qu'il est nécessaire de rappeler avant d'aborder le problème de la couverture comptable.

1.2 Obligation classique

Les institutions financières ont deux possibilités pour lever des fonds afin de financer leurs investissements commerciaux à long terme, elles peuvent emprunter ces fonds et / ou émettre des obligations.

Définition 1.1 : Une obligation est une dette. En effet, quand un investisseur achète une obligation, il prête en réalité une somme d'argent à l'émetteur de l'obligation et celui-ci contracte une dette. Par conséquent, l'émetteur (ou le vendeur) est emprunteur, et l'investisseur (ou l'acheteur) est prêteur. Comme l'indique la figure suivante :

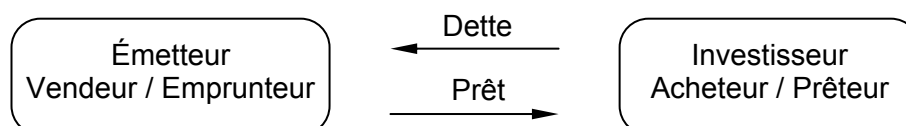


Figure 1.1 : Un prêt pour l'investisseur est une dette pour l'émetteur.

Généralement, le prix d'achat de l'obligation correspond à l'argent que l'investisseur prête à l'émetteur. Et comme dans la plupart des prêts, lorsqu'on achète une obligation, l'emprunteur verse à l'acheteur des intérêts à la fin de chaque période pendant toute la

durée de vie du prêt. Ensuite, à l'échéance fixée, l'emprunteur rembourse le prêt. Cette opération peut être représenté sur la flèche du temps (voir la figure (1.2)).

Définitions 1.2 :

- Le prix payé pour l'obligation, noté V_n , est appelé valeur nominale, ou montant principal.
- La durée de vie d'un prêt, notée T , s'appelle son échéance (ou maturité).
- Les intérêts du prêt payés par l'emprunteur, notés C , s'appellent les coupons.

Remarque 1.1 : Les obligations sont aussi connues sous le nom des titres à revenu fixe, car la plupart des obligations rapportent un revenu régulier au prêteur, qui correspond au taux d'intérêt du prêt.

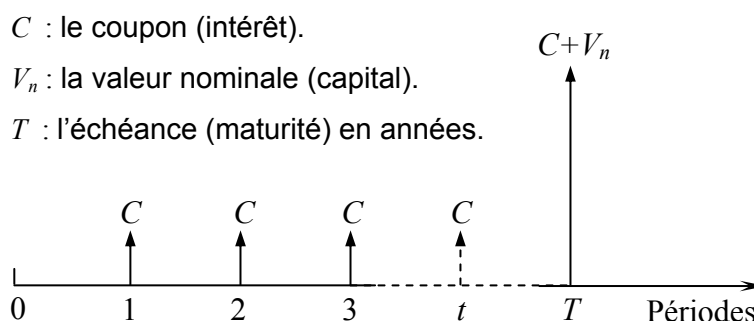


Figure 1.2 : Représentation des flux monétaires d'une obligation ordinaire sur la flèche du temps.

1.2.1 Flux monétaires et modalités de remboursement du capital

Les flux monétaires (ou, Cash-flows en anglais) à revenu fixe sont des flux d'intérêts et de remboursements du capital, Ces flux sont connus et fixés à l'avance, plus précisément, l'émetteur du titre (i.e. l'emprunteur) n'a pas le droit de modifier le montant des flux versés au détenteur de titre. Il s'agit d'un contrat où l'emprunteur s'engage à rembourser le capital emprunté ainsi que des intérêts selon des modalités et une règle de fixation décidées initialement et intangible par la suite.

A l'émission du titre, les conditions de paiement des intérêts sont précisées (les coupons) et le remboursement (l'amortissement) du capital. On note F_t le flux versé au détenteur du titre à la date t , il est calculé par la formule suivante :

$$F_t = F_t^{(1)} + F_t^{(2)} \quad (t = 1, \dots, T) \quad (1.1)$$

où, $F_t^{(1)}$ et $F_t^{(2)}$: l'amortissement du capital et l'intérêt (le coupon), versés à la date t .

1) Remboursement in fine (Obligation ordinaire)

Le capital est remboursé en bloc à l'échéance ($t=T$).

- Flux de capital à la date t : $F_t^{(1)} = \begin{cases} 0 & , (t=1, \dots, T-1) \\ V_n & , t=T \end{cases}$
- Flux d'intérêt à la date t : $F_t^{(2)} = C$, ($t=1, \dots, T$)

$$\text{Le flux monétaire à la date } t \text{ est : } F_t = \begin{cases} C & , (t=1, \dots, T-1) \\ C + V_n & , t=T \end{cases} \quad (1.2)$$

2) Amortissement constant

Le capital est remboursé en T parties égales.

- Flux de capital à la date t : $F_t^{(1)} = \frac{V_n}{T}$ à toutes les dates ($t=1, \dots, T$)
- Flux d'intérêt à la date t : $F_t^{(2)} = k \times V_n \times (1 - \frac{t-1}{T})$, ($t=1, \dots, T$)

$$\text{Le flux monétaire à la date } t \text{ est : } F_t = \frac{V_n}{T} + k \times V_n \times (1 - \frac{t-1}{T}) \text{ , } (t=1, \dots, T) \quad (1.3)$$

3) Annuité constante

L'amortissement du capital est calculé de telle façon que les flux : F_t ($t=1, \dots, T$) sont tous égaux à une constante A . $F_t = A$, ($t=1, \dots, T$)

Le montant constant A , (capital + intérêt) est lié au capital initial V_n , au taux nominal k et au nombre de période T , par la relation :

$$A = \frac{k \times V_n}{1 - (1+k)^{-T}} \quad (1.4)$$

La séquence de flux $[-V_n, \underbrace{A, A, \dots, A}_{T \text{ fois}}]$

- Flux de capital à la date t : $F_t^{(1)} = V_n \times k \times \left[\frac{(1+k)^{(-T+t-1)}}{1 - (1+k)^{-T}} \right]$, ($t=1, \dots, T$)
- Flux d'intérêt à la date t : $F_t^{(2)} = V_n \times k \times \left[\frac{1 - (1+k)^{(-T+t-1)}}{1 - (1+k)^{-T}} \right]$, ($t=1, \dots, T$)

4) Remboursement zéro-coupon

Un seul flux comprenant remboursement du capital + intérêt :

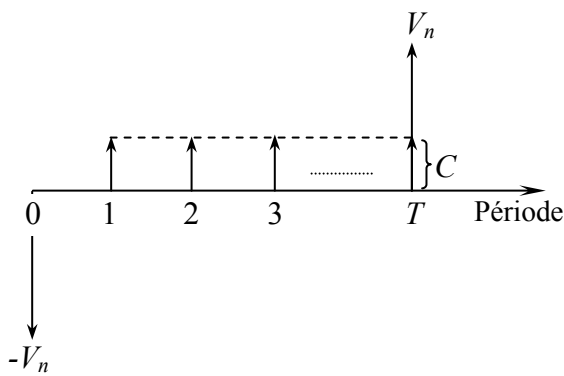
La séquence de flux : $[-V_n, V_n + C]$, (Les deux flux tombent respectivement en 0 et T),

avec : $C = V_n ((1+k)^T - 1)$.

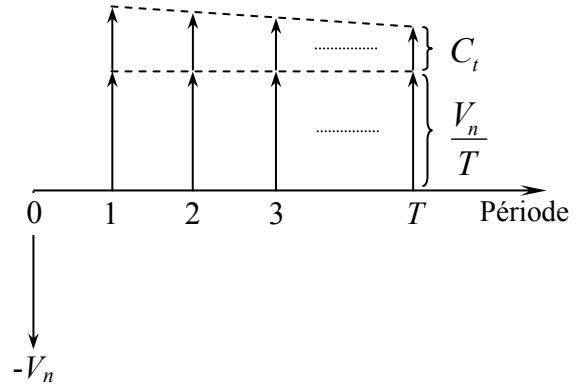
Le flux monétaire à la date T est :

$$F_T = V_n \times (1+k)^T \tag{1.5}$$

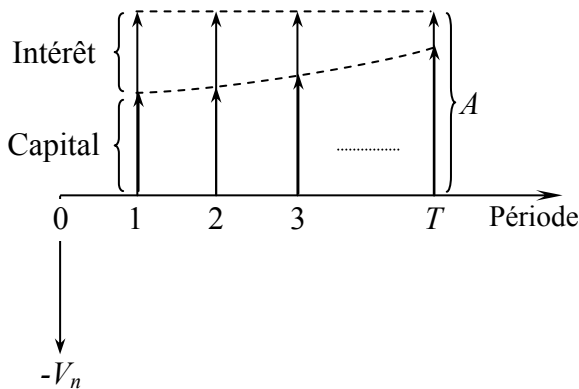
Les différentes modalités de remboursement d'une obligation sont représentées sur la figure ci-dessous :



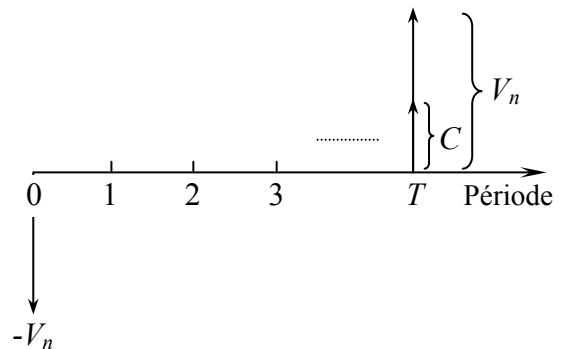
1) Remboursement In fine



2) Amortissement Constant



3) Annuité constante



(4) Remboursement zéro-coupon

Figure 1.3 : Différentes modalités de remboursement d'une obligation

1.2.2 Actualisation des flux monétaires

On applique ici, la technique de l'intérêt composé, un flux positif est une rentrée d'argent (revenu), un flux négatif est une sortie d'argent (paiement), et par souci de simplicité, on évalue les flux monétaires sur une base annuelle (i.e. la période est l'année).

Définition 1.2 : L'actualisation est une méthode qui permet de calculer la valeur actuelle en ($t = 0$) d'un cash-flow F_t à recevoir dans l'avenir en t .

On suppose que l'on peut prêter ou emprunter à toute échéance avec le même taux d'intérêt r .

(a) Considérons le flux (un seul flux) suivant :

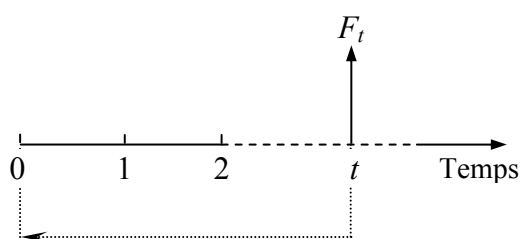


Figure 1.4.a : Actualisation d'un seul flux monétaire

La valeur actuelle d'un flux de paiement futur est :

$$VA = \frac{F_t}{(1+r)^t} \quad (1.6)$$

où, F_t : le flux monétaire versé à la date t , et r : le taux d'actualisation.

(b) Considérons la séquence de flux suivante (plusieurs flux) :

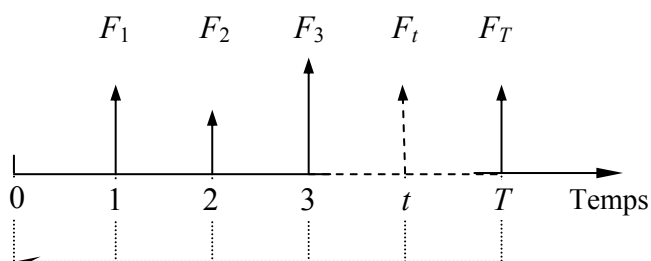


Figure 1.4.b : Actualisation d'une série de flux monétaires

La valeur actuelle d'une séquence de flux de paiements futurs est la somme des valeurs actuelles de chacun des flux :

$$VA = \sum_{t=1}^T \frac{F_t}{(1+r)^t} \quad (1.7)$$

où, F_t : le flux monétaire versé à la date t ; r : le taux d'actualisation.

Définition 1.3 : Le prix d'une obligation est la somme des valeurs actuelles (en $t = 0$) des flux monétaires futurs, et il est donné par la formule suivante :

$$P(r) = \sum_{t=1}^T \frac{F_t}{(1+r)^t} \quad (1.8)$$

où, P : le prix de l'obligation en ($t = 0$).

F_t : flux monétaire versé à la date t .

T : l'échéance (la durée de vie de l'obligation).

r : le taux d'actualisation.

Les flux monétaires d'une obligation ordinaire (remboursement in fine) sont calculés par le schéma suivant :

$$\begin{cases} F_t = C = k \times V_n, & t = (1, \dots, T-1) \\ F_T = C + V_r, & t = T \end{cases}$$

Le prix de l'obligation en fonction du taux d'actualisation r s'écrit :

$$P(r) = \frac{C}{(1+r)^1} + \frac{C}{(1+r)^2} + \dots + \frac{C}{(1+r)^{T-1}} + \frac{C+V_r}{(1+r)^T} \quad (1.9)$$

qui est une fonction non linéaire décroissante en r .

où, P : prix (ou, la valeur actuelle) de l'obligation.

V_n : valeur nominale (ou, montant principal)

V_r : valeur de remboursement (généralement $V_r \geq V_n$)

C : valeur du coupon.

r : taux de rendement annuel.

t : période du paiement annuelle.

T : échéance (ou, maturité) en nombre d'année.

1.2.3 Taux de rendement interne

Définition 1.4 : La valeur actuelle nette d'une obligation, notée VAN , est la différence entre la somme des cash-flows, actualisés à la date ($t = 0$) et le capital investi (Prix d'achat) P_0 :

$$VAN(r) = -P_0 + \sum_{t=1}^T \frac{F_t}{(1+r)^t} \quad (1.10)$$

Remarque 1.2 : Pour qu'une obligation soit rentable, il faut que la VAN soit positive pour un taux d'actualisation r donné, et elle est d'autant plus rentable que la VAN est grande.

Définition 1.5 : Le taux de rendement interne (ou taux actuariel), noté TRI , est un taux d'actualisation qui annule la valeur actuelle nette (i.e. $VAN(TRI) = 0$).

Il s'agit de la solution de l'équation : $-P_0 + \sum_{t=1}^T \frac{F_t}{(1+r)^t} = 0$, qui est une équation non

linéaire en r . L'existence et l'unicité du TRI sont assurées si les conditions (i) et (ii) sont vérifiées :

$$\text{i) } P_0 \geq 0; \quad F_t \geq 0, (t=1 \dots T)$$

$$\text{ii) } -P_0 + \sum_{t=1}^T F_t \geq 0$$

Remarque 1.3 : Le calcul précis du TRI est souvent laborieux. Il faut donc recourir aux méthodes itératives telles que : la méthode de (Dichotomie, Point fixe, Newton, ..., etc.).

Le TRI peut être calculé par l'algorithme du point fixe suivant :

Algorithme 1.1

Début

- a) Poser $k = 0$, $r^k = 10\%$, $w = 1/(T \times P_0)$, $\varepsilon > 0$ (ε : la précision requise sur la VAN)
- b) Si : $|VAN(r^k)| < \varepsilon$, aller à l'étape (d).
- c) $r^{k+1} = r^k + w \times VAN(r^k)$, $k \leftarrow k + 1$, retourner en (b).
- d) Poser : $\hat{r} \leftarrow r^k$, (la solution approchée est \hat{r}). Arrêter.

Fin

Précisons le vocabulaire utilisé :

- Si la valeur d'émission et la valeur nominale sont égales ($V_e = V_n$), on dit que l'obligation est émise au pair. Le taux actuariel de l'obligation est égal au taux d'intérêt nominal, (voir la figure 1.5).
- Si la valeur d'émission est inférieure à la valeur nominale ($V_e < V_n$) on dit que l'obligation est émise en dessous du pair. Le taux actuariel de l'obligation est supérieur au taux d'intérêt nominal, (voir la figure 1.5).
- Si la valeur d'émission est supérieure à la valeur nominale ($V_e > V_n$) on dit que l'obligation est émise au dessus du pair. Le taux actuariel de l'obligation est inférieur au taux d'intérêt nominal, (voir la figure 1.5).

La courbe suivante, représente l'évolution de prix en fonction du taux d'actualisation.

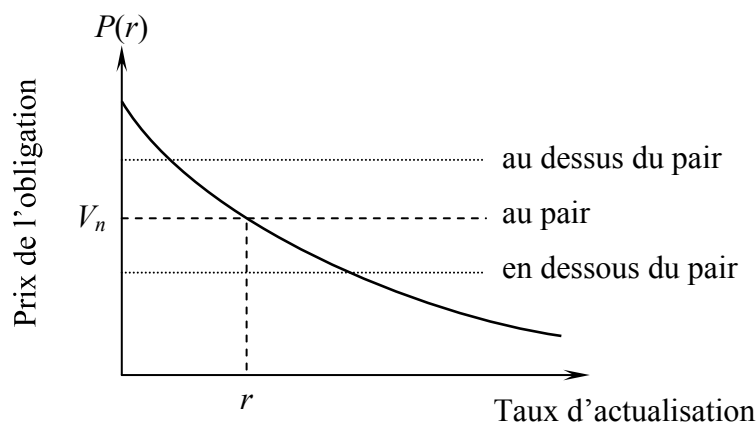


Figure 1.5 : Évolution de prix d'une obligation ordinaire en fonction du taux d'actualisation.

Exemple 1.1 : « Obligation émise en dessous du pair ($V_e < V_n$) »

Considérons une obligation de 5 ans d'échéance, de 1000 € de nominal, le taux nominal est de 6%, et la valeur d'émission (prix d'achat) est : $P_0 = 980$ €.

Les flux monétaires de l'obligation sont représentés dans le tableau suivant :

Année	0	1	2	3	4	5
Flux monétaire	-980	60	60	60	60	1060

Tableau 1.1 : Répartition des flux monétaires d'obligation.

La valeur actuelle nette de l'obligation en fonction du taux de d'actualisation r est :

$$VAN(r) = -980 + \frac{60}{(1+r)^1} + \frac{60}{(1+r)^2} + \frac{60}{(1+r)^3} + \frac{60}{(1+r)^4} + \frac{1060}{(1+r)^5}$$

Le taux de rendement interne est calculé par l'algorithme (1.1) :

$$VAN(r) = 0 \Rightarrow TRI \approx 6,48\%$$

On remarque bien que, le TRI (6,48%) est supérieur au taux d'intérêt nominal (6%).

1.3 Risque de taux d'intérêt

Tout instrument financier comporte un ensemble de risques. Le risque le plus significatif attendu par les porteurs d'obligations est le risque de taux d'intérêt. Le prix d'une obligation va toujours dans le sens inverse d'un changement des taux d'intérêt (voir figure 1.6). Si on conserve une obligation jusqu'à échéance, cela n'a pas d'importance, mais si l'on veut vendre avant échéance, une baisse du prix signifiera une perte en capital. Ce risque de perte en capital est appelé le risque de marché ou de taux d'intérêt.

Exemple 1.2 : Reprenons l'exemple (1.1), et nous calculons la valeur actuelle nette (VAN) pour différents taux de rendement :

Rendement interne (TRI en %)	Valeur actuelle (VA en €)	Valeur actuelle nette (VAN en €)
5,48	1022,17	+ 42,17
6,48	980,00	00,00
7,48	940,05	- 39,95

Tableau 1.2 : Évaluation de la VAN pour différents taux de rendement

Interprétation :

D'après le tableau ci-dessus, le taux de rendement interne est : $\hat{r} \approx 6,48\%$. Si le taux de rendement augmente brusquement à 7,48 %, la réalisation de cette opération revient à décaisser 980 € et recevoir immédiatement 940,05 € en contrepartie, l'opération est donc perdante, et la VAN mesure cette perte en capital ($VAN = -39,95$ €).

1.3.1 Mesures systématiques de risque de taux

Le risque de marché est le risque relié à une variation des taux d'intérêt et de l'impact de cette variation sur le prix des obligations.

Les mesures de risque de taux les plus utilisées sont : la Variation, la Sensibilité, la Duration et la Convexité.

La Variation :

Une variation infinitésimale « dr » du taux d'intérêt se traduit par une variation « dP » de la valeur de l'obligation obtenue par la dérivation de l'équation (1.8)

$$\frac{dP}{dr} = \frac{-1 \times F_1}{(1+r)^2} + \frac{-2 \times F_2}{(1+r)^3} + \dots + \frac{-T \times F_T}{(1+r)^{T+1}} = -\frac{1}{(1+r)} \sum_{t=1}^T \frac{t \times F_t}{(1+r)^t} \quad (1.11)$$

La quantité : $-\frac{dP}{dr}$ s'appelle la «Variation».

La Sensibilité :

La Sensibilité est la variation en pourcentage du prix P induite par une variation du taux d'intérêt r , et elle est donnée par :

$$S = -\frac{1}{P} \times \frac{dP}{dr} = \frac{1}{(1+r)} \times \frac{1}{P} \times \sum_{t=1}^T \frac{t \times F_t}{(1+r)^t} \quad (1.12)$$

La variation $\frac{\Delta P}{P}$ induite par une variation Δr non infinitésimale du taux est appréciée à

l'aide de la relation approximative : $\frac{\Delta P}{P} \approx -S \times \Delta r$

La formule suivante, fournit une approximation acceptable (d'ordre 2 en Δr) de la Sensibilité d'une obligation.

$$S \approx \frac{1}{P(r)} \left[\frac{P(r - \Delta r) - P(r + \Delta r)}{2 \cdot \Delta r} \right] \quad (1.13)$$

Remarque 1.4 : Pour de petites variations de r , la Sensibilité donne une bonne estimation de la variation du prix des obligations. Cette estimation se dégrade au fur et à mesure que la variation Δr augmente. En effet, pour les grandes fluctuations Δr , la Sensibilité sous estime le nouveau prix, du fait de la forme convexe de la relation entre le prix et le rendement à l'échéance.

La Duration : est la mesure la plus courante de la sensibilité d'une obligation. Elle est définie comme étant la durée de vie effective de l'obligation ou encore la durée moyenne pondérée, pour récupérer entièrement le capital et les paiements d'intérêt, et elle est donnée par la formule :

$$D = \frac{1}{P(r)} \sum_{t=1}^T \frac{t \times F_t}{(1+r)^t} \quad (1.14)$$

Le coefficient de pondération relatif à l'échéance t est $w_t = \frac{1}{P(r)} \left[\frac{F_t}{(1+r)^t} \right]$

On vérifie bien que : $\sum_{t=1}^T w_t = 1$, alors : $D = \sum_{t=1}^T t \times w_t$

La *Duration* correspond donc à la position moyenne (ou le Centre de Gravité) des flux actualisés sur la flèche du temps ci-dessous.

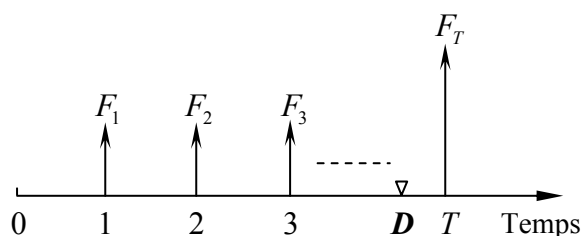


Figure 1.6 : Position de la *Duration* (D) sur la flèche du temps.

Dérivation de la convexité

Le développement en série de *Taylor* de la variation du prix d'une obligation (ΔP) due à une variation du taux de rendement (Δr) s'écrit :

$$\Delta P = \frac{dP}{dr} \Delta r + \frac{1}{2} \frac{d^2 P}{dr^2} (\Delta r)^2 + \dots$$

donc, la variation relative (en pourcentage) du prix P s'écrit approximativement :

$$\frac{\Delta P}{P} \approx \left[\frac{1}{P} \frac{dP}{dr} \right] \Delta r + \frac{1}{2} \left[\frac{1}{P} \frac{d^2 P}{dr^2} \right] (\Delta r)^2$$

ou encore :

$$\frac{\Delta P}{P} \approx -S \times \Delta r + \frac{1}{2} C \times (\Delta r)^2 \quad (1.15)$$

où, C désigne la convexité de l'obligation, et elle est donné par la formule :

$$C = \frac{1}{P} \frac{d^2 P}{dr^2} \quad (1.16)$$

La formule (1.15), présente la relation fondamentale qui permet d'utiliser conjointement la sensibilité et la convexité pour prévoir le taux de variation du prix d'une obligation suite à une variation du taux de rendement à l'échéance. L'expression de la convexité est donnée par :

$$C = \frac{1}{(1+r)^2 P(r)} \sum_{t=1}^T t(t+1) \frac{F_t}{(1+r)^t} \quad (1.17)$$

La formule suivante, fournit une approximation acceptable de la convexité

$$C \approx \frac{1}{P(r)} \left[\frac{P(r+\Delta r) - 2P(r) + P(r-\Delta r)}{(\Delta r)^2} \right] \quad (1.18)$$

L'importance de la convexité :

Soient A et B deux obligations ayant la même *Sensibilité*. Si l'obligation B est plus convexe que l'obligation A , les investisseurs préfèrent l'obligation B à A puisque :

- Lorsque les taux baissent, le prix de B augmente plus que celui de A . (voir figure 1.7)
- Lorsque les taux augmentent, le prix de B baisse moins que celui de A . (voir figure 1.7)

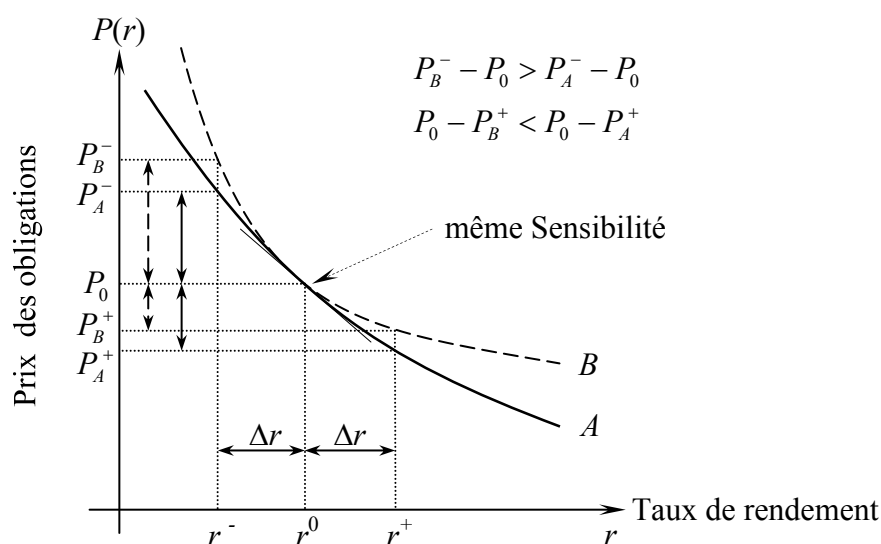


Figure 1.7 : Comparaison de deux obligations ayant la même *Sensibilité* et de *Convexité* différente.

1.4 Portefeuille obligataire

Définition 1.6 : Un portefeuille (ou panier) obligataire est un ensemble d'obligations.

Pour un portefeuille P composé de N obligations, la valeur actuelle du portefeuille est égale à la somme des valeurs actuelles des obligations qui le compose :

$$V_p = \sum_{i=1}^N V_i \quad (1.19)$$

où, V_p est la valeur actuelle du portefeuille, V_i est la valeur actuelle de l'obligation i .

La *Duration* d'un portefeuille obligataire est égale à la somme pondérée des *Durations* des obligations qui le compose :

$$D_p = \sum_{i=1}^N w_i D_i \quad \text{avec} \quad w_i = \frac{V_i}{V_p} \quad (i=1, \dots, N) \quad (1.20)$$

où : les w_i sont les coefficients de pondération tel que $\sum_{i=1}^N w_i = 1$, D_i est la *Duration* de l'obligation i , et D_p est la *Duration* du portefeuille.

De même pour la *Convexité* : $C_p = \sum_{i=1}^N w_i C_i \quad (1.21)$

où : C_i est la *Convexité* de l'obligation i . et C_p est la *Convexité* du portefeuille.

1.4.1 Taux de rendement d'un portefeuille

Afin de déterminer le rendement à échéance, il faut projeter tous les flux monétaires de l'ensemble du portefeuille et procéder à l'actualisation de ces flux. Le rendement à l'échéance du portefeuille est le taux tel que, la valeur actuelle des flux monétaires du portefeuille est égale à la valeur marchande de ce portefeuille.

Exemple 1.3 : Soit un portefeuille P composé de deux obligations A et B avec les caractéristiques suivantes :

Obligation	Prix	Taux nominal	Échéance
A	995 €	6 %	3 ans
B	1000 €	5 %	2 ans

Tableau 1.3 : Les caractéristiques du portefeuille.

Pour calculer le taux de rendement, on projette les flux des deux obligations A et B dans le portefeuille P , puis on procède à l'actualisation.

Le tableau suivant, illustre le calcul du taux de rendement du portefeuille :

Année	0	1	2	3	TRI	$Duration$
Obligation A	-995	60	60	1060	6,188	2,83
Obligation B	-1000	50	1050	/	5,000	1,95
Portefeuille P	-1995	110	1110	1060	r_p	D_p

Tableau 1.4 : Le taux de rendement d'un portefeuille composé de deux obligations.

Le taux de rendement (TRI) du portefeuille est :

$$-1995 + \frac{110}{(1+r_p)^1} + \frac{1110}{(1+r_p)^2} + \frac{1060}{(1+r_p)^3} = 0 \Rightarrow r_p \approx 5,704\%$$

où : r_p est le taux de rendement du portefeuille.

La $Duration$ D_p du portefeuille est : $D_p = \frac{995 \times D_A + 1000 \times D_B}{995 + 1000} \approx 2,39$ années

où, D_A : la $Duration$ de l'obligation A , et D_B : la $Duration$ de l'obligation B .

1.5 La courbe des taux d'intérêt

Dans les paragraphes précédents on a supposé que le taux d'actualisation est unique, c'est-à-dire, la courbe de taux est plate, mais en pratique, cette courbe est très rarement plate.

Définition 1.7 : La courbe des taux à l'instant t est l'ensemble de taux d'intérêt, notés $r(t, T)$ ou $r_T(t)$ pour les différentes échéances T , (T variant de 0 à 30 ans).

Notation 1.1 : Le taux $r(0, T)$ est noté simplement : r_T , indiquant, le taux d'aujourd'hui à ($t = 0$) pour l'échéance T .

La courbe des taux peut être représentée graphiquement comme sur la figure (1.8), où l'axe des abscisses correspond aux différentes échéances T , et l'axe des ordonnées aux taux prévalant sur le marché.

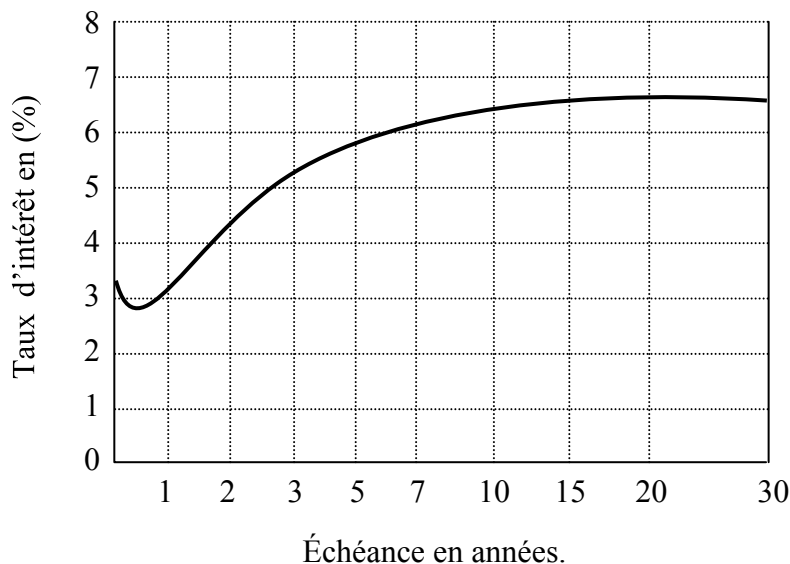


Figure 1.8 : Exemple de courbe des taux d'intérêt.

1.5.1 Taux de marché au comptant

Les taux de marché au comptant, dit aussi « *Spot* », sont les taux moyens constatés sur le marché sur un type particulier d'instrument et pour une certaine durée. Ils sont dits au comptant car valables pour une mise en place immédiate.

Deux catégories de taux existent : les taux à court terme et les taux à long terme.

- Les taux à court terme vont jusqu'à un an, les plus connus sont, l'EURIBOR (*European Interbank Offered rate*) pour l'euro, et le LIBOR (*London Interbank Offered rate*) pour les autres devises. Ils sont calculés quotidiennement en effectuant la moyenne des taux prêteurs d'un certain nombre de banques, dites de référence, pour des échéances de un à douze mois.
- Les taux à long terme, supérieurs à un an, dont les plus connus sont, les BTAN (*Bons du Trésor à intérêts annuels*) à échéance 2 à 5 ans, ils produisent un coupon annuel, et les OAT (*Obligation Assimilables du Trésor*) : échéance de 7 à 25 ans. Ils sont calculés à partir des obligations émises par l'Etat.

Ces taux sont régulièrement publiés par des agences d'information financière (ou les Benchmarks). Ils sont exprimés sous forme d'une courbe des taux.

1.5.2 Taux zéro-coupons

Définition 1.8 : Une obligation zéro coupon (ou *Strip*) est une obligation qui ne verse qu'un seul flux monétaire à l'échéance T . Aucun coupon n'est versé pendant sa durée de vie.

Le taux zéro-coupon est implicitement défini dans la relation suivante :

$$B_t = \frac{1}{(1+r_t)^t}, \quad (t=1, \dots, T) \quad (1.22)$$

où :

B_t : prix de marché d'aujourd'hui d'une obligation zéro-coupon délivrant 1 euro à la date t . On appelle aussi B_t , le facteur d'actualisation pour la maturité t .

r_t : taux de rendement d'aujourd'hui de l'obligation zéro-coupon délivrant 1 euro en t .

Toute obligation peut être décomposée en une somme d'obligations zéro-coupons d'échéances différentes. La valeur de l'obligation globale est égale à la somme des valeurs des obligations zéro coupon qui la compose. Alors, la valeur actuelle de l'obligation s'écrit plus justement :

$$V(r_1, r_2, \dots, r_T) = \sum_{t=1}^T \frac{F_t}{(1+r_t)^t} = \sum_{t=1}^T B_t \times F_t \quad (1.23)$$

où, r_t : le taux zéro-coupon pour l'échéance t .

F_t : le cash-flow versé à la date t

B_t : le facteur d'actualisation pour l'échéance t .

Pour évaluer correctement une obligation, il suffit donc de connaître les taux zéro-coupons associés aux maturités de chacun des flux de l'obligation. Malheureusement, ces taux n'existent pas sur le marché pour un continuum de maturité, il est donc nécessaire de reconstruire cette courbe. La méthode classique utilisée par les gestionnaires est la méthode récurrente ([1], [3], [8]) qui calcule les taux zéro-coupons de proche en proche. Le principe est le suivant :

On calcule le taux zéro-coupon r_i , ($i=1, \dots, T$) en considérant une obligation in fine, au pair de durée T années et en écrivant sa valeur :

$$\sum_{t=1}^{i-1} \frac{s_t}{(1+r_t)^t} + \frac{1+s_i}{(1+r_i)^i} = 100\% \quad , \quad (i=1, \dots, T) \quad (1.24)$$

où, s_i : le taux au comptant (*Spot*) pour l'échéance i .

r_i : le taux zéro-coupon pour l'échéance i .

Le système linéaire (1.24) permet d'exprimer, r_i en fonction de s_i et de r_1, r_2, \dots, r_{i-1} , par la relation de récurrence suivante :

$$\begin{cases} B_1 = \frac{1}{1+s_1} \\ B_i = \frac{1-s_i \sum_{t=1}^{i-1} B_t}{(1+s_i)} \quad (i=2, \dots, T) \end{cases} \quad (1.25)$$

En suite, les taux zéros-coupon sont calculés par la formule :

$$r_t = \left(\frac{1}{B_t} \right)^{1/T} - 1 \quad (t=1, \dots, T) \quad (1.26)$$

où, r_t : le taux zéro-coupon pour l'échéance t .

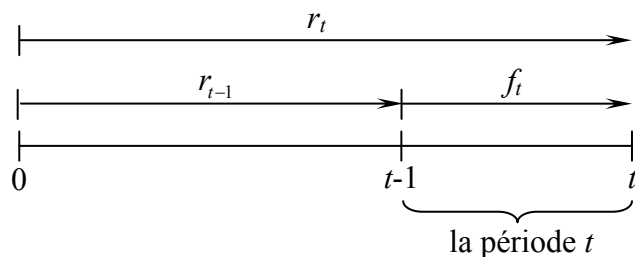
B_t : le facteur d'actualisation pour l'échéance t .

T : la maturité(ou échéance) de l'obligation.

La connaissance de la courbe des taux zéro-coupons permet aussi de déduire deux autres courbes très utilisées en pratique : la courbe des taux à terme et la courbe des taux de rendement au pair.

1.5.3 Taux à terme

Les taux de marché au comptant (*Spot*) sont des taux de placement immédiat pour ces échéances, or il peut être nécessaire de connaître par exemple le taux de début de la période t jusqu'à la fin de cette période (voir figure 1.9). Ce qui est utile pour déterminer la courbe des taux variables implicites, dite : taux à terme (ou *forward*). Cette courbe permet de valoriser les obligations à taux variables.



r_t : taux zéro-coupon pour la maturité t

f_t : taux *forward* de la période t

Figure 1.9 : Relation entre le taux zéro-coupon et *forward*.

Cette méthode consiste à calculer à partir de la courbe des taux zéro-coupons, les taux à terme par la formule suivante :

$$\begin{cases} f_1 = r_1 \\ f_t = \frac{(1+r_t)^t}{(1+r_{t-1})^{t-1}} - 1 = \frac{B_{t-1}}{B_t} - 1 \quad (t = 2, \dots, T) \end{cases} \quad (1.27)$$

où, r_t : le taux zéro-coupon pour la maturité t .

f_t : le taux à terme (*forward*) de la période t

B_t : le facteur d'actualisation pour la maturité t .

Propriété 1.1 [6] : Les taux à termes (*forward*) et les taux zéro-coupons sont liés par la relation suivante :

$$\sum_{t=1}^T \frac{f_t}{(1+r_t)^t} + \frac{1}{(1+r_T)^T} \stackrel{\text{def}}{=} \sum_{t=1}^T B_t \times f_t + B_T = 1 \quad (1.28)$$

1.5.4 Taux de rendement au pair

Une obligation au pair, est une obligation dont le taux de coupon est identique au taux de rendement actuariel, c'est-à-dire qui vaut 100 (100% du montant nominal de l'obligation).

Désignant par r_t , le taux zéro-coupon pour la maturité t , le taux de rendement au pair p_T pour la maturité T est calculé comme suit :

$$p_T \sum_{t=1}^T B_t + B_T = 1 \Rightarrow p_T = \frac{1 - B_T}{\sum_{t=1}^T B_t} \quad (1.29)$$

La courbe associée à la maturité T le taux p_T , est utilisée afin de déterminer le niveau du coupon lors de l'émission d'une obligation au pair.

1.5.5 Taux de rendement à maturité

Les marchés définissent également la notion de taux de rendement actuariel y qui est implicitement défini dans la formule suivante :

$$P_0 = \frac{F_1}{(1+y)^1} + \frac{F_2}{(1+y)^2} + \dots + \frac{F_T}{(1+y)^T} \quad (1.30)$$

Il s'agit de rechercher le taux d'actualisation y , qui permet exactement d'égaliser la valeur actuelle des flux avec le prix initial P_0 . Le calcul du taux de rendement actuariel se fait par des méthodes itératives telles que la méthode de (Newton, Point fixe, ..., etc).

1.6 Valorisation d'obligation et risque de taux d'intérêt

Toute obligation peut être décomposée en une somme d'obligations zéro-coupons d'échéances différentes. La valeur de l'obligation globale est égale à la somme des valeurs des obligations zéro coupon qui la compose. Alors, la valeur actuelle d'une obligation, s'écrit :

$$\begin{aligned} V(r_1, r_2, \dots, r_T) &= \sum_{t=1}^T \frac{C_t}{(1+r_t)^t} + \frac{M}{(1+r_T)^T} \\ &= \sum_{t=1}^T C_t \times B_t + M \times B_T \end{aligned} \quad (1.31)$$

où, V : le prix (la valeur actuelle) de l'obligation prévalant sur le marché.

B_t : le facteur d'actualisation, avec : $B_t = (1+r_t)^{-t}$ (r_t : le taux zéro-coupon)

M : la valeur nominale (ou le principal).

C_t : le coupon (l'intérêt) à la date t .

Pour évaluer une obligation, il suffit donc de connaître les taux zéro-coupon associés aux maturités de chacun des flux de l'obligation.

1.6.1 Obligation à taux fixe

Les obligations à taux fixe ont un coupon qui est constant. En général, le coupon versé en t ans, résulte d'une formule telle que :

$$C_t = C = k \times M, \quad (t = 1, \dots, T) \quad (1.32)$$

où : M est la valeur nominale ; k est le taux nominal (fixe).

La valeur actuelle de l'obligation s'écrit :

$$V(r_1, r_2, \dots, r_T) = \sum_{t=1}^T \frac{M \times k}{(1+r_t)^t} + \frac{M}{(1+r_T)^T} \quad (1.33)$$

qui est une fonction à plusieurs variables en (r_1, r_2, \dots, r_T) .

La sensibilité de la valeur actuelle aux variations de la courbe des taux est exprimée par le vecteur gradient :

$$\left[\frac{\partial V(r_1, r_2, \dots, r_T)}{\partial r_t} \right], \quad (t = 1, \dots, T) \quad (1.34)$$

Généralement, le calcul analytique du vecteur gradient est compliqué. On donne ici une approximation du gradient en différences finies d'ordre 2 en h :

$$\frac{\partial V}{\partial r_t} \approx \left[\frac{V(r_1, \dots, r_t + h, \dots, r_T) - V(r_1, \dots, r_t - h, \dots, r_T)}{2h} \right], \quad (t = 1, \dots, T) \quad (1.35)$$

où : h est un scalaire choisi suffisamment petit ($h \rightarrow 0$).

1.6.2 Obligation à taux variable

Les obligations à taux variable ont un coupon qui varie en fonction du niveau des taux prévalant sur le marché. En général, le coupon versé en t ans, résulte d'une formule telle que :

$$C_t = M \times f_t \quad (t = 1, \dots, T) \quad (1.36)$$

où, M : la valeur nominale ; f_t : le taux *forward* de la période t .

La valeur actuelle de l'obligation s'écrit :

$$V(r_1, r_2, \dots, r_T) = \sum_{t=1}^T \frac{M \times f_t}{(1+r_t)^t} + \frac{M}{(1+r_T)^T} \quad (1.37)$$

En utilisant la propriété (1.1), la valeur V s'écrit simplement :

$$V(r_1, r_2, \dots, r_T) = M \times \underbrace{\left(\sum_{t=1}^T B_t f_t + B_T \right)}_{=1} = M \quad (1.38)$$

La valeur actuelle V est indépendante des taux (r_1, r_2, \dots, r_T) . Donc, le vecteur gradient est nul :

$$\frac{\partial V}{\partial r_t} = 0 \quad , \quad (t=1, \dots, T) \quad (1.39)$$

par conséquent, la valeur actuelle est insensible aux variations des taux. En d'autre terme, les coupons futurs sont ajustés au taux de marché, lors d'une baisse des taux d'intérêts, les gains sur le compte du capital sont compensés par les pertes sur le compte des intérêts recevables.

Remarque 1.5 : Pour l'investisseur, le choix entre le taux fixe et le taux variable est fondé sur ses anticipations de taux d'une part, et sur l'exposition au risque de taux qu'il est prêt à accepter d'autre part. Si l'investisseur prévoit une hausse des taux, il préférera les placements à taux variable, en effet, s'il a raison, la valeur de ses titres ne diminuera pas. En revanche, en cas d'anticipation d'une baisse des taux, il privilégiera le taux fixe.

1.7 Swap de taux d'intérêt

Dans cette section, nous présentons l'instrument financier : swap de taux d'intérêt, nous donnons ici une définition précise. Celle-ci nécessite d'abord la terminologie suivante :

Définition 1.9 : Un produit dérivé est un instrument financier dont la valeur est dérivée (d'où son nom) de la valeur d'un autre actif financier de base.

Le produit dérivé étudié ici, est le swap de taux d'intérêt.

Définition 1.10 : Un swap (du mot anglais signifiant échange) de taux d'intérêt, ou IRS (*Interest Rate Swap*) est un contrat de gré à gré aux termes duquel, deux parties conviennent d'échanger des flux d'intérêt portant sur un capital nominal, suivant une durée déterminée à l'avance.

L'opération swap consiste, pour chaque partie, à contracter simultanément un prêt et un emprunt dans la même devise pour un montant nominal. Le capital nominal est fictif en ce sens qu'il n'est généralement pas échangé, son seul but est de servir de référence pour le calcul des intérêts (l'échange porte uniquement sur des flux d'intérêts). Les deux termes de l'opération sont appelés branches ou jambes du swap.

Le swap le plus commun, dit « *Plain Vanilla Swap* » (taux fixe contre taux variable), présente la structure suivante :

- Une partie verse un taux d'intérêt fixe (emprunt à taux fixe) et reçoit un taux d'intérêt variable (prêt à taux variable) sur un montant notionnel. On l'appelle « prêteur de swap » ou « payeur du taux fixe ». (voir figure ci-dessous).

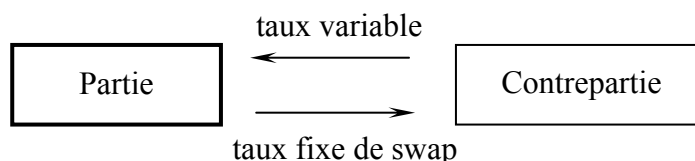


Figure 1.10.a : Échange des flux monétaires d'un swap prêteur.

- L'autre partie reçoit un taux fixe (prêt à taux fixe) et verse un taux variable (emprunt à taux variable) sur le même nominal. On l'appelle « emprunteur de swap » ou « receveur du taux fixe ». (voir figure ci-dessous).

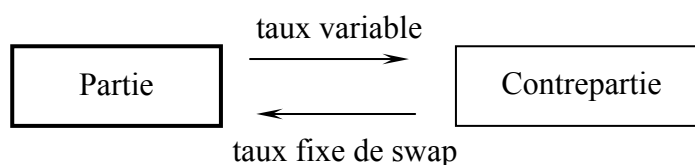


Figure 1.10.b : Échange des flux monétaires d'un swap emprunteur.

1.7.1 Valorisation

La principale méthode d'évaluation des swaps de taux d'intérêt est la méthode obligataire ([1], [6]). Un swap est comparable à deux obligations : l'une détenue (flux positifs) et l'autre émise (flux négatifs). La méthode consiste à déterminer la valeur actuelle des flux générés par chacune des deux obligations (ou jambes), la différence des flux actualisés des deux jambes constitue la valeur du swap.

Notation 1.2 : Le swap prêteur et emprunteur sont notés respectivement S_p, S_e .

Le tableau suivant, illustre l'échange d'intérêts d'un swap prêteur :

Années	0	$t = (1, \dots, T-1)$	T
Emprunt (Taux fixe)	$+M$	$-M \times k$	$-M \times k - M$
Prêt (Taux variable)	$-M$	$+M \times f_t$	$+M \times f_T + M$
Swap prêteur (S_p)	0	$M \times (f_t - k)$	$M \times (f_T - k)$

Tableau 1.5 : L'échange d'intérêts d'un swap prêteur.

La valeur actuelle du swap prêteur est donnée par :

$$V(S_p) = \sum_{t=1}^T B_t \times M \times (f_t - k) \quad (1.40)$$

où, M : la valeur nominale (ou le principal).

f_t : le taux *forward* (variable) de la période t .

k : le taux nominale de la jambe fixe.

B_t : le facteur d'actualisation de la période t .

T : la maturité du swap S .

La valeur actuelle, d'un swap emprunteur et celle d'un swap prêteur son liées comme suit :

$$V(S_e) = -V(S_p).$$

La valeur du marché d'un swap à l'instant ($t = 0$) où il est fixé, est nulle. En effet, la valeur actualisée de la jambe variable est égale à la valeur actualisée de la jambe fixe.

Définition 1.11 : Le taux de swap, en ($t = 0$), est le taux de la jambe fixe pour lequel la valeur actuelle du swap soit nulle.

Le taux de swap, noté k_T^{swap} , peut être calculé par la formule suivante :

$$k_T^{\text{swap}} = \frac{\sum_{t=1}^T B_t \times f_t}{\sum_{t=1}^T B_t} \quad (1.41)$$

où, f_t : le taux *forward* (variable) de la période t .

B_t : le facteur d'actualisation pour la maturité t .

Définition 1.12 : La courbe associée à la maturité T , le taux k_T^{swap} s'appelle la courbe des taux de swap à l'instant ($t = 0$).

Exemple 1.4 : On considère un swap de taux (prêteur) dont les caractéristiques sont :

Valeur nominale : 1000 € ; Maturité : 4 ans ; Taux de swap : $k_4^{\text{swap}} = 5,47\%$.

L'évaluation de swap à l'instant ($t = 0$) est illustré dans le tableau suivant :

Période (en année)		1	2	3	4
Taux d'actualisation (zéro-coupon) en (%)		5,00	5,10	5,30	5,50
Taux à terme (<i>forward</i>) en (%)		5,00	5,20	5,70	6,10
Taux fixe (taux de swap) en (%)		5,47	5,47	5,47	5,47
Flux de la jambe variable	Total	50,00	52,00	57,01	1061,02
Flux actualisé de la jambe variable	1000,00	47,62	47,08	48,83	856,48
Flux de la jambe fixe		- 54,75	- 54,75	- 54,75	-1054,75
Flux actualisé de la jambe fixe	-1000,00	- 52,14	- 49,56	- 46,89	- 851,41
Flux actualisé du swap	0,00	- 4,52	- 2,49	+ 1,94	+ 5,07

Tableau 1.6 : Évaluation d'un swap prêteur par la méthode obligatoire

1.7.2 Calcul de sensibilité

La sensibilité de la valeur actuelle d'un swap aux variations de la courbe des taux est exprimée par le vecteur gradient (ou *Key rate Duration*) de dimension T , qui est donné par :

$$\begin{aligned} \frac{\partial V(S_p)}{\partial r} &= - \frac{\partial V(J_F)}{\partial r} \\ \frac{\partial V(S_e)}{\partial r} &= + \frac{\partial V(J_F)}{\partial r} \end{aligned} \quad (1.42)$$

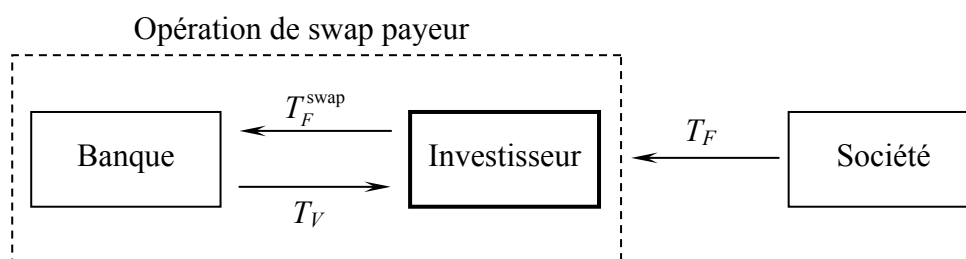
où, S_p : le swap prêteur, S_e : le swap emprunteur, et J_F : la jambe fixe du swap.

Utilité d'un swap de taux d'intérêt

Généralement, le swap de taux est utilisé pour gérer le risque de taux. Le swap prêteur bénéficie de la hausse de la courbe des taux, donc il peut être utilisé pour pallier les éventuels inconvénients d'une hausse des taux, alors que le swap emprunteur possède les qualités inverses et peut être utilisé comme une couverture contre le risque de baisse des taux.

Exemple 1.5 (de couverture)

Un investisseur qui est financé par une société au moyen de billet à taux fixe (T_F), et qui craint une augmentation future des taux d'intérêt. L'investisseur peut atténuer le risque de taux d'intérêt en contractant un swap de taux, au terme duquel, il paye un taux fixe (T_F^{swap}) et reçoit un taux variable (T_V). De cette façon, l'investisseur convertit son revenu à taux fixe, en revenu à taux variable et immobilise le niveau actuel des taux d'intérêt. Cette opération peut être schématisé sur la figure suivante :



T_F : le taux fixe. T_V : le taux variable. T_F^{swap} : le taux fixe de swap

Figure 1.11 : Échange d'un revenu à taux fixe en revenu à taux variable.

1.8 Reconstitution de la courbe des taux zéro-coupons

La connaissance de la courbe des taux zéro-coupons est très importante en pratique, car cela permet de :

- 1) évaluer les produits de taux délivrant des flux futurs, par exemple, les obligations à taux fixe et les obligations à taux variable, les swaps, ..., etc.
- 2) dériver les autres courbes implicites, par exemple, la courbe des taux *forward*, et la courbe des taux de rendement au pair, ..., etc.
- 3) représenter les différentes déformations de cette courbe dans le temps.

La reconstitution de la courbe des taux d'intérêt est nécessaire, car il n'existe pas suffisamment d'obligation zéro-coupons cotées sur le marché, par conséquent, il n'est pas possible d'obtenir ces taux à n'importe quelle date pour actualiser les flux monétaires. Plusieurs approches sont proposées afin de reconstituer cette courbe. Certains sont employés comme des fonctions d'interpolation (linéaire, cubique), tandis que d'autres sont des modèles dynamiques ([1], [8], [11]), à priori spécifiés. On distingue deux grandes classes de modèles :

- Les modèles type *Nelson* et *Siegel* (1987) fondés sur une modélisation des taux zéro-coupons.
- Les modèles à splines. Ils sont fondés sur une modélisation de la fonction d'actualisation. Les plus connus sont les splines polynomiaux (Mc Culloch (1971,1975)) et les splines exponentielles (*Vasicek et Fong* (1982)).

L'avantage de ces modèles, tient à leur grande flexibilité, qui leur permet de reconstruire toutes les formes de courbe rencontrées sur le marché. Leur principe est, pour un portefeuille d'obligations à coupons, il s'agit de la minimisation de l'écart quadratique entre les prix du marché et les prix reconstitués à l'aide d'un modèle des taux zéro-coupon ou de la fonction d'actualisation.

1.8.1 Scénarios de déformation de la courbe des taux

Pour simuler la courbe des taux de référence, ainsi que ces déformations futures, une étude statistique en ACP (*Analyse en Composantes Principales*) [8], effectuée sur un historique de courbes des taux (zéro-coupons), a montré que trois facteurs expliquent la quasi-totalité (plus de 95%) des déformations de ces courbes. Ces trois facteurs s'interprètent comme suit :

- 1) Un premier facteur qui présente une forme quasiment plate et qui provoque des déplacements parallèles de la courbe des taux. Ce facteur s'interprète comme un facteur de niveau. (voir figure (1.12.a) et (1.12.b)).
- 2) Un deuxième facteur qui rend compte des inversions de la courbe autour d'un point. Ce facteur s'interprète comme un facteur de pente car il a une influence contraire sur les taux longs et sur les taux courts (rotation). (voir figure (1.12.c)).
- 3) Un troisième facteur, qui agit différemment sur trois compartiments de la courbe des taux, et qui s'interprète comme un facteur de courbure. (voir figure (1.12.d)).

La figure suivante, illustre les principales déformations futures de la courbe des taux.

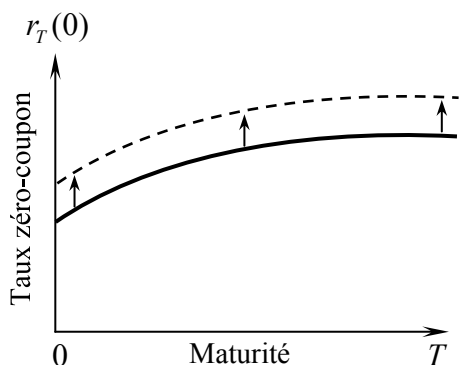


Figure 1.12.a : Déplacement parallèle :
(vers le haut).

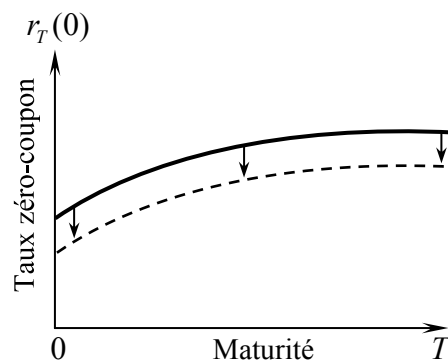


Figure 1.12.b : Déplacement parallèle :
(vers le bas).

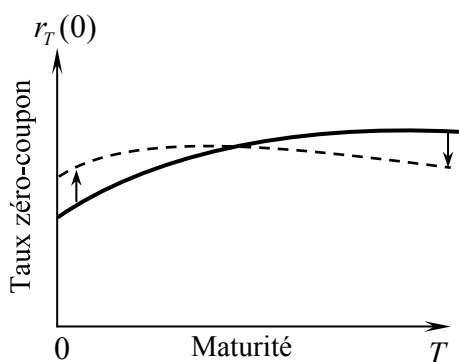


Figure 1.12.c : Déformation de rotation
(ou d'inversion)

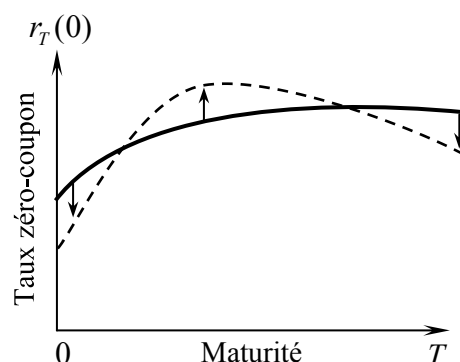


Figure 1.12.d : Déformation de courbure
sur trois compartiments

— Courbe des taux de référence

- - - Déformation de la courbe de référence

Figure 1.12 : Les principales déformations futures de la courbe des taux.

1.9 Conclusion

Dans ce chapitre, nous avons présenté les différentes notions sur les instruments financiers, en particulier, les obligations et les swaps de taux d'intérêt, et nous avons donné des exemples sur la valorisation de prix de ces instruments et sur le calcul de leur sensibilité par rapport aux fluctuations des taux d'intérêt. Ces notions sont indispensables pour aborder le problème de la couverture comptable. Il est considéré comme un problème d'optimisation combinatoire (voir chapitre 3), et qui est résolu par des méthodes duales d'optimisation que nous allons les présenter dans le chapitre 2.

CHAPITRE 2

MÉTHODES DUALES D'OPTIMISATION NON LINÉAIRE

2.1 Introduction

Depuis quelques années, les méthodes duales suscitent beaucoup d'intérêt pour la résolution des problèmes d'optimisation sous contraintes, tant par la simplicité de leur mise en œuvre que par les avantages qu'elle offre par rapport aux méthodes (directes) primales ([13], [16],[19]). Dans ce chapitre, nous présentons les méthodes de dualité et les méthodes de pénalité. Nous rappelons, auparavant, les principaux concepts de l'optimisation (continue) non-linéaire, ainsi que les méthodes numériques de minimisation sans contraintes. Ensuite nous abordons les méthodes duales fondées sur les techniques de lagrangien augmenté. Nous appliquons ces techniques sur un exemple de problème quadratiques à contraintes linéaires d'inégalité. L'algorithme de type d'*Uzawa* à pas constant, est implémenté et testé sur cet exemple.

2.2 Problème de l'optimisation non linéaire

Le problème d'optimisation non linéaire en dimension finie, consiste à chercher le minimum d'une fonction f dans un ensemble X , il s'agit de déterminer un point $\hat{x} \in X$ tel que : $f(\hat{x}) \leq f(x), \forall x \in X$. On dit alors que \hat{x} est solution optimale (globale) du problème de minimisation sur X .

Le problème peut s'écrire sous la forme suivante :

$$(P_X) \begin{cases} \text{Min}_x f(x) \\ x \in X \end{cases}$$

avec :

x : est un vecteur de composantes (x_1, x_2, \dots, x_n) , qui sont les inconnues du problème.

f : est la fonction objectif (ou économique), qui est une fonction continue non linéaire.

X : est l'ensemble des solutions réalisables (ou admissibles), qui est un sous-ensemble de \mathfrak{R}^n ($X \subseteq \mathfrak{R}^n$), continu.

2.3 Notions fondamentales

Dans les problèmes d'optimisation non linéaire, la notion de la convexité joue un rôle très important. En effet, pour la plupart des algorithmes, la convergence vers un optimum global ne pourra être démontrée qu'avec des hypothèses de convexité.

Définition 2.1 : Un ensemble $X \subset \mathfrak{R}^n$ est dit convexe si pour tout couple $(x, y) \in X^2$ et $\forall \lambda \in [0, 1]$ on a : $\lambda x + (1 - \lambda)y \in X$.

Cette définition peut s'interpréter en disant que le segment reliant x et y doit être contenu entièrement dans X .

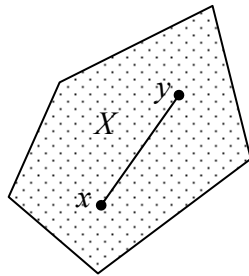


Figure 2.1.a Ensemble convexe

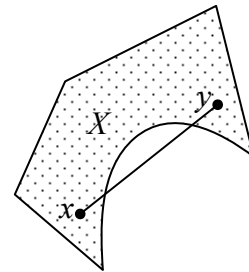


Figure 2.1.b Ensemble non-convexe

Définition 2.2 : On dit qu'une fonction $f : X \rightarrow \mathfrak{R}$, définie sur un ensemble convexe X , est convexe si elle vérifie : $\forall (x, y) \in X^2, \forall \lambda \in [0, 1], f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$

f est dite strictement convexe si :

$$\forall (x, y) \in X^2, x \neq y, \forall \lambda \in]0, 1[, f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y)$$

Dans le cas unidimensionnel ($n = 1$), cette définition s'interprète bien géométriquement : Le graphe de la fonction est toujours en dessous du segment reliant les points $(x, f(x))$ et $(y, f(y))$.

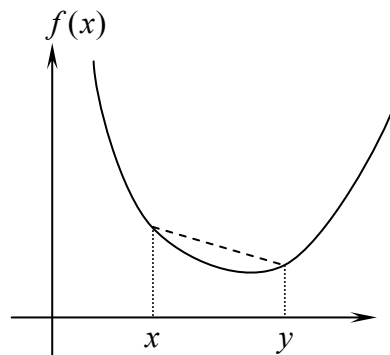


Figure 2.2.a : Fonction convexe

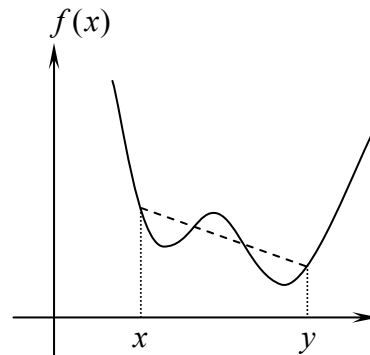


Figure 2.2.b : Fonction non-convexe

Caractérisation de la convexité en terme du gradient

Dans le cas où la fonction f n'est supposée qu'une fois différentiable, on a le résultat suivant :

Théorème 2.1 [19]. Soit $f : X \subset \mathfrak{R}^n \rightarrow \mathfrak{R}$ une fonction une fois différentiable, alors f est convexe si et seulement si : $f(y) \geq f(x) + \nabla f(x)(y - x), \forall (x, y) \in X^2$

La fonction f est strictement convexe si et seulement si :

$$f(y) > f(x) + \nabla f(x)(y - x), \forall (x, y) \in X^2, x \neq y.$$

L'interprétation géométrique de ce résultat quand ($n = 1$) : Le graphe d'une fonction convexe se trouve au-dessus de la tangente en un point donné.

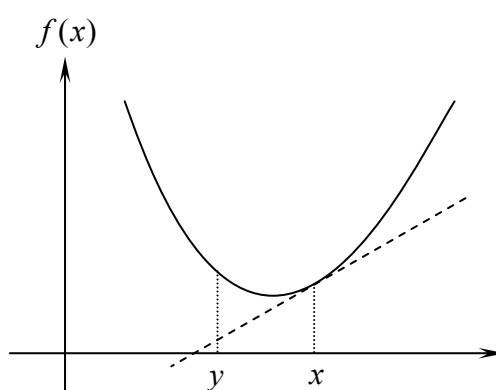


Figure 2.3 : Caractérisation de la convexité en terme du gradient.

Caractérisation de la convexité en terme du hessien

Théorème 2.2 [19]. Soit $f : X \subset \mathfrak{R}^n \rightarrow \mathfrak{R}$ une fonction deux fois différentiable, alors f est convexe si et seulement si : $\nabla^2 f(x) \geq 0, \forall x \in X$, et strictement convexe si : $\nabla^2 f(x) > 0, \forall x \in X$

Définition 2.3 : Une fonction f est dite concave si la fonction $-f$ est convexe.

Résultats d'existence et d'unicité

On donne deux résultats d'existence d'une solution au problème (P_X) ; auparavant nous avons besoin de la définition d'un ensemble compact :

Définition 2.4 : Un ensemble $X \subset \mathfrak{R}^n$ est dit compact, si de toute suite $\{x_k\}$, où $x_k \in X, \forall k$ on peut extraire une sous-suite convergente.

Théorème 2.3 [19]. Un ensemble $X \subset \mathfrak{R}^n$ est compact si et seulement si il est fermé et borné.

Théorèmes généraux d'existence et d'unicité

Théorème 2.4 [19]. *Si $f : X \subset \mathfrak{R}^n \rightarrow \mathfrak{R}$ est continue et si de plus X est un ensemble compact, alors le problème (P_X) admet une solution optimale $\hat{x} \in X$, qui vérifie donc $f(\hat{x}) \leq f(x), \forall x \in X$.*

Le second résultat est moins général car il considère le cas particulier $X \equiv \mathfrak{R}^n$:

Théorème 2.5 [19]. *Soit $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$ une fonction continue sur \mathfrak{R}^n . Si $\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$, alors (P_X) admet une solution optimale \hat{x} .*

En général, l'unicité résulte de la propriété de convexité (de f et de X).

Théorème 2.6 [19] : *Soit $f : X \subset \mathfrak{R}^n \rightarrow \mathfrak{R}$ strictement convexe sur X convexe. Le minimum de f sur X , s'il existe, est unique.*

2.4 Optimisation sans contraintes

Définition 2.5 : Le problème (P_X) est dit, sans contraintes, si $X \equiv \mathfrak{R}^n$, et on note :

$$(P') \quad \begin{cases} \text{Min}_x f(x) \\ x \in \mathfrak{R}^n \end{cases}$$

2.4.1 Conditions d'optimalité

Condition nécessaire

Dans le cas où f est différentiable, on a le résultat suivant :

Théorème 2.7 [19]. *Soit $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$ différentiable et \hat{x} vérifiant $f(\hat{x}) \leq f(x), \forall x \in \mathfrak{R}^n$, alors on a nécessairement : $\nabla f(\hat{x}) = 0$*

Condition nécessaire et suffisante

La condition de gradient nul devient suffisante dans le cas où f est convexe :

Théorème 2.8 [19]. *Soit $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$ convexe et différentiable. Si \hat{x} vérifie $\nabla f(\hat{x}) = 0$, alors on a $f(\hat{x}) \leq f(x), \forall x \in \mathfrak{R}^n$*

Lorsque la fonction n'est pas convexe, on ne peut donner qu'une condition nécessaire et suffisante d'optimalité locale.

Définition 2.6 : On appelle \hat{x} un minimum local de f , s'il existe $\delta > 0$ tel que :

$$f(\hat{x}) \leq f(x), \quad \forall x, \|x - \hat{x}\| \leq \delta$$

Dans le cas où f est deux fois différentiable on peut alors donner le résultat suivant :

Théorème 2.9 [19]. Soit $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$: deux fois différentiable, Si $\begin{cases} \nabla f(\hat{x}) = 0 \\ \nabla^2 f(\hat{x}) > 0 \end{cases}$, alors \hat{x} est un minimum local de f .

2.4.2 Méthodes numériques de gradient

On considère le problème d'optimisation (P_X) dans le cas où $X \equiv \mathfrak{R}^n$:

$$(P') \quad \begin{cases} \text{Min } f(x) \\ x \\ x \in \mathfrak{R}^n \end{cases}$$

et, on suppose que f est continue et à dérivées premières continues.

Pratiquement, toutes les méthodes d'optimisation sans contraintes dans \mathfrak{R}^n consiste à rechercher un point \bar{x} stationnaire tel que : $\nabla f(\bar{x}) = 0$. Ce problème est équivalent à la résolution du système d'équations non linéaires. Les méthodes les plus couramment utilisées, sont les méthodes de gradient.

Principe général des méthodes de gradient

On génère une suite de points x^0, x^1, \dots, x^k convergeant vers un optimum local de f . À chaque étape k , x^{k+1} est défini par : $x^{k+1} = x^k + \alpha_k d_k$ où α_k est le pas de déplacement ($\alpha_k > 0$), et d_k est une direction de déplacement qui peut être soit :

- L'anti-gradient de f en x^k : $d_k = -\nabla f(x^k)$.
- Calculée à partir du gradient $\nabla f(x^k)$.
- Choisie de façon plus ou moins arbitraire à condition que ce soit une direction de descente c'est -à-dire que : $\nabla f^t(x^k).d_k < 0$.

Méthode du gradient à pas fixe

Dans cette méthode, le pas α_k est choisi constant à chaque itération d'où l'algorithme suivant :

Algorithme 2.1 : ($\alpha_k = \alpha \quad \forall k \geq 0$)

Début

(a) Poser : $k = 0, \varepsilon > 0, \alpha > 0$, choisir un point de départ x^0 .

(b) À l'itération k , calculer : $d_k = -\nabla f(x^k)$

Faire : $x^{k+1} = x^k + \alpha d_k$

(c) Test d'arrêt : Si $\|d_k\| < \varepsilon$, arrêter. Sinon faire : $k \leftarrow k + 1$, et retourner en (b).

Fin.

Méthode de la plus forte pente (Steepest descent)

Dans cette méthode, le pas α_k est choisi de façon à minimiser la fonction : $\alpha \rightarrow f(x^k - \alpha \nabla f(x^k))$, sur l'ensemble des ($\alpha \geq 0$).

Algorithme 2.2

Début

(a) Poser : $k = 0, \varepsilon > 0$, choisir un point de départ x^0 .

(b) À l'itération k , calculer : $d_k = -\nabla f(x^k)$

Rechercher α_k tel que : $f(x^k + \alpha_k d_k) = \underset{\alpha \geq 0}{\text{Min}} f(x^k + \alpha d_k)$

Faire : $x^{k+1} = x^k + \alpha_k d_k$

(c) Test d'arrêt : Si $\|d_k\| < \varepsilon$, arrêter. Sinon faire : $k \leftarrow k + 1$, et retourner en (b).

Fin.

Remarque 2.1 : Si, f est continûment différentiable, tout ce qu'on peut dire c'est que l'on obtient un point \bar{x} stationnaire de f , c'est-à-dire $\nabla f(\bar{x}) = 0$.

- Si, f est deux fois continûment différentiable en \bar{x} , et que $\nabla^2 f(\bar{x})$ est définie positif, alors \bar{x} est un minimum local de f .
- Si, f est convexe différentiable en \bar{x} , alors \bar{x} est minimum global de f .

Vitesse de convergence des méthodes de gradient

La rapidité de convergence des méthodes de gradient dépende des valeurs propres de la matrice : $\nabla^2 f(x)$ (le hessien de la fonction f en point x).

Définition 2.7 : Le conditionnement d'une matrice A symétrique définie positive, noté $\text{cond}(A)$, est le rapport de la plus grande à la plus petite valeur propre de la matrice A .

Ce rapport est toujours supérieur à 1, (voir annexe-B). Plus le conditionnement du hessien est proche de un (i.e. $\text{cond} \approx 1$), plus la convergence est rapide. Plus le conditionnement est grand (i.e. $\text{cond} \gg 1$), plus la convergence est lente.

Définition 2.8 : La courbe de niveau d'une fonction f au niveau β , est le sous-ensemble noté : $C_\beta(f)$ (éventuellement vide), tel que : $C_\beta(f) \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n / f(x) = \beta\}$

Les ensembles de niveaux ressemblent plus ou moins à des ellipsoïdes, qui ont une excentricité plus ou moins forte en fonction du conditionnement, le cas idéal étant constitué par les fonctions du type $f : x \mapsto r\|x\|^2$, avec : ($r > 0$), dont les ensembles de niveaux sont sphériques où le conditionnement vaut 1, (voir la figure ci-dessous) :

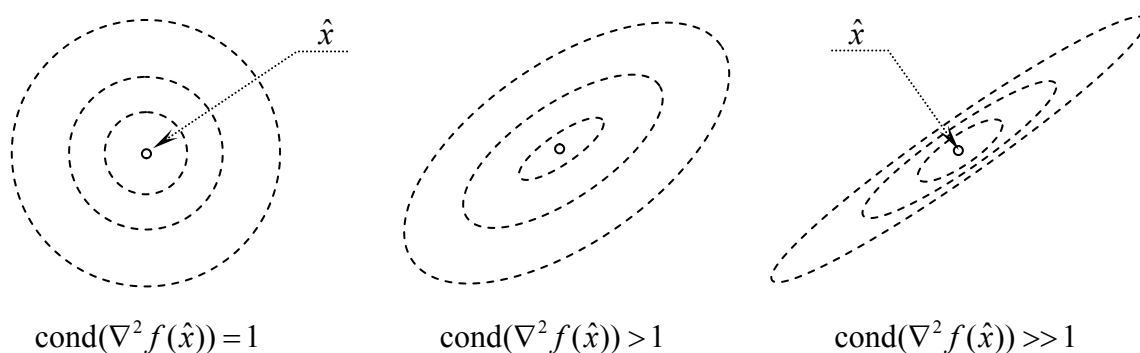


Figure 2.4 : Différentes formes de courbes de niveau d'une fonction f à deux variables.

Remarque 2.2 : Pour plus de détails sur la convergence des méthodes de gradient nous renvoyons le lecteur aux ouvrages ([14], [15], [16], [17], [19]).

2.5 Optimisation avec contraintes

Définition 2.9 : Le problème d'optimisation (P_X) est dit, avec contraintes, si $X \subset \mathfrak{R}^n$.

En général, l'ensemble X est défini par : $X = \{x \in \mathfrak{R}^n / g_i(x) \leq 0, (i = 1, \dots, m)\}$.

On considère le problème (P) suivant :

$$(P) \quad \begin{cases} \text{Min}_x f(x) \\ g_i(x) \leq 0 \quad i \in I = \{1, 2, \dots, m\} \\ x \in \mathfrak{R}^n \end{cases}$$

Toutes les fonctions f et g_i ($i = 1, \dots, m$) sont supposées continues différentiables.

2.5.1 Les conditions nécessaires d'optimalité de Kuhn et Tucker

Le théorème suivant est fondamental et donne, une condition nécessaire d'optimalité locale pour le problème (P) .

Théorème 2.10 [19]. (*Kuhn et Tucker*). Une condition nécessaire pour que x soit un optimum local de (P) , est qu'il existe des nombres $\lambda_i \geq 0$ ($i \in I$) appelés multiplicateurs de Kuhn et Tucker, tel que :

$$(KT) \quad \begin{cases} \nabla f(x^0) + \sum_{i=1}^m \lambda_i \nabla g_i(x^0) = 0 \\ \lambda_i g_i(x^0) = 0 \quad i \in I \end{cases} \quad (2.1)$$

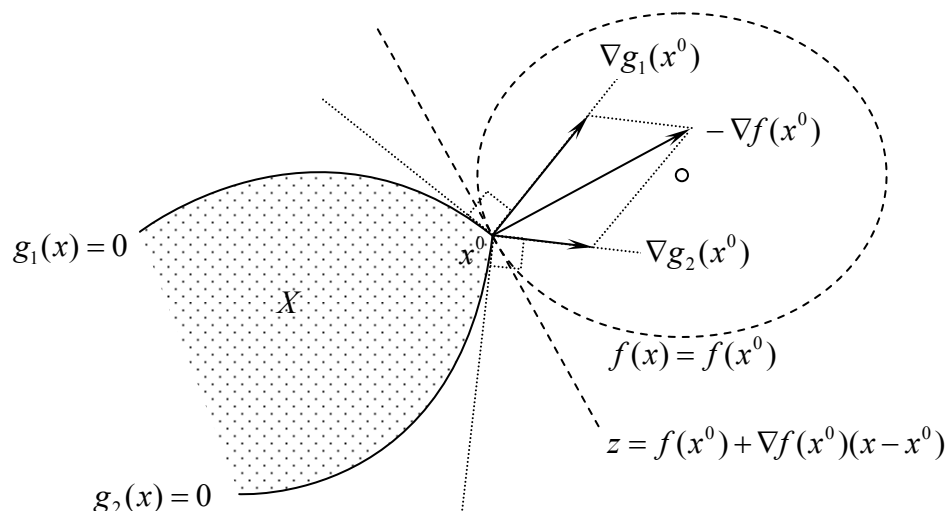
Définition 2.10 : On appelle contraintes saturées en $x^0 \in X$ l'ensemble des contraintes g_i tel que, $g_i(x^0) = 0$, et on note : $I^0 = \{i \in I / g_i(x) = 0\}$

Définition 2.11 : On dit que $x^0 \in X$ est un point régulier si les gradients des contraintes saturées $[\nabla g_i(x^0)]_{i \in I^0}$ sont linéairement indépendants.

Dans le cas où x^0 est un point régulier, le vecteur λ de *Kuhn Tucker* est unique.

Interprétation géométrique des conditions de *Kuhn et Tucker*

Les conditions de *Kuhn et Tucker* peuvent s'interpréter géométriquement de la façon suivante : Pour que x^0 soit un optimum local il faut que le vecteur $-\nabla f(x^0)$ fasse un angle aigu avec le gradient $\nabla g_i(x^0)$ de chaque contraintes saturées ($i \in I^0$), autrement dit $-\nabla f(x^0)$ doit s'exprimer comme combinaison linéaire à coefficient λ_i positifs des $\nabla g_i(x^0)$ pour $i \in I^0$. Comme l'indique la figure ci-dessous :



au point x^0 , les contraintes saturées sont g_1, g_2

Figure 2.5 : Illustration des conditions de *Kuhn et Tucker* sur un exemple à deux dimensions.

2.5.2 Condition suffisante d'optimalité

Considérons le problème (P).

$$(P) \begin{cases} \text{Min } f(x) \\ x \in \mathbb{R}^n \\ g_i(x) \leq 0 \quad i \in I = \{1, 2, \dots, m\} \end{cases}$$

Fonction de Lagrange (lagrangien)

Associons à chaque contrainte : g_i , ($i \in I$) un nombre réel ($\lambda_i \geq 0$) appelé multiplicateur de Lagrange. La fonction de Lagrange associée au problème (P) est par définition :

$$L(x, \lambda) = f(x) + \sum_{i \in I} \lambda_i g_i(x) \quad , \quad x \in \mathbb{R}^n \text{ et } \lambda \in \mathbb{R}^{m+} \quad (2.2)$$

Point-selle du lagrangien

Soit : $\bar{x} \in \mathfrak{R}^n$ et $\bar{\lambda} \geq 0$, on dit que $(\bar{x}, \bar{\lambda})$ est point-selle de $L(x, \lambda)$ si :

$$\forall \lambda \in \mathfrak{R}^{m+} \quad L(\bar{x}, \lambda) \leq L(\bar{x}, \bar{\lambda}) \leq L(x, \bar{\lambda}) \quad \forall x \in \mathfrak{R}^n \quad (2.3)$$

La propriété caractéristique du point-selle

Théorème 2.11 [19]. Soit $\bar{x} \in \mathfrak{R}^n$ et $\bar{\lambda} \geq 0$, $(\bar{x}, \bar{\lambda})$ est un point-selle pour $L(x, \lambda)$ si et seulement si :

$$\begin{aligned} \text{a) } & L(\bar{x}, \bar{\lambda}) = \underset{x \in \mathfrak{R}^n}{\text{Min}} L(x, \bar{\lambda}) \\ \text{b) } & g_i(\bar{x}) \leq 0 \quad , (i \in I) \\ \text{c) } & \bar{\lambda}_i g_i(\bar{x}) = 0 \quad , (i \in I) \end{aligned} \quad (2.4)$$

Suffisance de la condition de point-selle

Théorème 2.12 [19]. Si $(\bar{x}, \bar{\lambda})$ est un point-selle de $L(x, \lambda)$, alors \bar{x} est un optimum global de (P) .

Une image à deux dimensions (une variable x , une variable λ) d'un point-selle est donnée à la figure suivante :

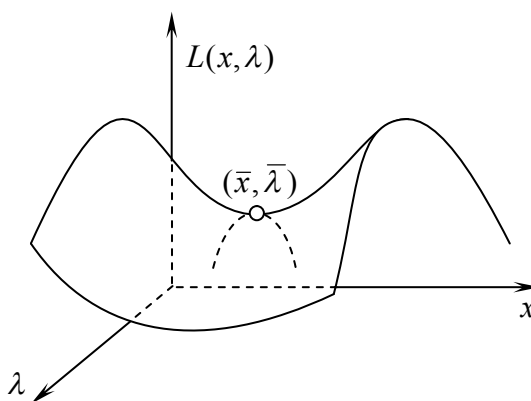


Figure 2.6 : Illustration de la notion de point-selle

2.5.3 Dualité lagrangienne classique

Considérons le problème suivant :

$$(P) \begin{cases} \text{Min}_x f(x) \\ g_i(x) \leq 0 \quad (i=1, \dots, m) \\ x \in \mathfrak{R}^n \end{cases}$$

La fonction de Lagrange est :

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i g_i(x), \quad (x \in \mathfrak{R}^n, \lambda \in \mathfrak{R}^{m+}) \quad (2.5)$$

Le problème (P) peut être résolu si l'on sait déterminer un point-selle de la fonction de Lagrange L .

Définissons pour $\lambda \geq 0$ la fonction $w: \mathfrak{R}^{m+} \rightarrow \mathfrak{R}$ par :

$$w(\lambda) = \text{Min}_{x \in \mathfrak{R}^n} L(x, \lambda) \quad (2.6)$$

La recherche d'un point-selle, lorsqu'il existe, peut se faire en résolvant le problème suivant :

$$(D) \begin{cases} \text{Max}_{\lambda} w(\lambda) \\ \lambda \in \mathfrak{R}^{m+} \end{cases}$$

(D) est appelé le problème Dual de (P) . Par opposition (P) est appelé le problème Primal, w est la fonction duale.

Propriété 2.1 : Pour tout $\lambda \in \mathfrak{R}^{m+}$, la valeur de la fonction duale $w(\lambda)$ est un minorant de l'optimum absolu $f(\hat{x})$ de (P) , autrement dit, si $w(\hat{\lambda})$ est la valeur optimale du problème dual : $\forall \lambda \geq 0: w(\lambda) \leq w(\hat{\lambda}) \leq f(\hat{x})$

Propriété 2.2 :

a) Si, le problème (P) admet un point-selle $(\hat{x}, \hat{\lambda})$ alors, on a :

$$\text{Max}(D) = w(\hat{\lambda}) = f(\hat{x}) = \text{Min}(P)$$

b) S'il existe \bar{x} solution de (P) et $\bar{\lambda} \geq 0$ tels que : $w(\bar{\lambda}) = f(\bar{x})$ alors (P)

admet un point-selle, et $(\bar{x}, \bar{\lambda})$ est un tel point-selle.

Propriété 2.3 : La fonction duale w est fonction concave de λ .

Propriété 2.4 : $[g_i(x(\lambda))]_{i \in I}$ est un gradient de la fonction duale w au point λ .

2.6 Méthodes de dualité

Les méthodes de dualité ([16], [19]) consistent à résoudre le problème dual (D) en utilisant essentiellement la concavité de la fonction duale w et la propriété (2.4), qui permet de déduire un gradient pour la fonction w .

2.6.1 La méthode d'Uzawa

La méthode d'Uzawa ([16], [19]) utilise une méthode de gradient projeté classique pour résoudre le problème dual. À chaque étape, la fonction de Lagrange $(x, \lambda) \mapsto L(x, \lambda)$ est minimisée en x et maximisée en λ , d'où l'algorithme suivant :

Algorithme 2.3 (d'Uzawa)

Début

- (a) Poser : $k = 0, \lambda^0 \in \mathfrak{R}^{m+}, x^0 \in \mathfrak{R}^n, \varepsilon > 0$
- (b) Déterminer : x^{k+1} solution du problème $\text{Min}_{x \in \mathfrak{R}^n} L(x, \lambda^k)$ en partant de x^k
- (c) Si, $\| \nabla_x L(x^{k+1}, \lambda^k) \| < \varepsilon$, alors : arrêter.
- (d) Calculer : $\lambda^{k+1} = \max(0, \lambda^k + \rho_k g(x^{k+1}))$
- (e) Faire : $k \leftarrow k + 1$, et retourner en (b)

Fin.

Remarque 2.3 : Dans cet algorithme, les pas de déplacement ρ_k sont choisis a priori : il s'agit alors d'une méthode de gradient à pas déterminé (fixe), appliquée à la fonction duale. Les pas peuvent être choisis de façon à maximiser $w(\lambda)$ à partir de λ^k dans la direction $\nabla w(\lambda^k) = g(x^{k+1})$: il s'agit de la méthode de la plus forte pente appliquée au problème dual.

2.6.2 La méthode d'Arrow-Hurwicz

La méthode d'Arrow-Hurwicz ([16], [19]) est très voisine de la précédente, la différence réside dans le fait qu'à chaque étape k , on ne recherche pas le minimum en x de la fonction de Lagrange $L(x, \lambda^k)$ mais, on se contente d'un déplacement (suivant un

pas prédéterminé) dans la direction : $-\nabla_x L(x^k, \lambda^k)$. La méthode consiste à se déplacer alternativement dans l'espace des variables primales et dans l'espace des variables duales.

Algorithme 2.4 (d'Arrow-Hurwicz)

Début

- (a) Poser : $k = 0$ et $\lambda^0 \in \mathfrak{R}^{m+}$, $x^0 \in \mathfrak{R}^n$, $\varepsilon > 0$
- (b) Déterminer : x^{k+1} par $x^{k+1} = x^k - \alpha_k \cdot \nabla_x L(x^k, \lambda^k)$
- (c) Si, $\| \nabla_x L(x^{k+1}, \lambda^k) \| < \varepsilon$, alors : arrêter.
- (d) Calculer : $\lambda^{k+1} = \max(0, \lambda^k + \rho_k g(x^{k+1}))$
- (e) Faire : $k \leftarrow k + 1$, et retourner en (b)

Fin.

2.6.3 Conditions de convergence des méthodes de dualité

Les méthodes de dualité lagrangienne convergent vers une solution optimale du problème (P) sous les conditions suivantes :

Théorème 2.13 [19] : Si f est strictement convexe, si les g_i ($i = 1, \dots, m$) sont convexes, et si X est d'intérieur non vide, l'existence de \hat{x} est équivalente à l'existence de $\hat{\lambda}$ et on a :

$$w(\hat{\lambda}) = L(\hat{x}, \hat{\lambda}) = f(\hat{x}) \quad (2.7)$$

Remarque 2.4 : Il existe des cas, où il existe un point-selle et les conditions précédentes ne sont pas vérifiées. En plus, quand il n'y a pas de point selle, on peut faire appel à des techniques où on utilise un lagrangien augmenté qu'on va le voir par la suite.

2.7 Méthodes de pénalité

Principe général.

Considérons le problème (P). Soit $h(\mathfrak{R} \rightarrow \mathfrak{R})$ la fonction définie par :

$$\begin{cases} h(y) = 0 & \text{si : } y \leq 0 \\ h(y) = +\infty & \text{si : } y > 0 \end{cases}$$

Considérons le problème sans contraintes (problème pénalisé) :

$$(PP) \begin{cases} \text{Min}_x \varphi(x) = f(x) + H(x) \\ x \in \mathfrak{R}^n \end{cases}$$

où la fonction H , appelée fonction de pénalisation est définie par :

$$\forall x \in \mathfrak{R}^n : H(x) = \sum_{i=1}^m h(g_i(x))$$

L'optimum de $\varphi(x)$ ne peut être atteint en un point n'appartenant pas à X , (car : $\varphi(x) = +\infty$), d'autre part, pour tout $x \in X$, $H(x) = 0$, et par conséquent : $\forall x \in X, \varphi(x) = f(x)$

Résoudre (P) est équivalent à la résolution du problème pénalisé (PP) , qui est un problème d'optimisation sans contraintes.

Remarque 2.5 : L'approche précédente n'est pas directement applicable en pratique, car la fonction de pénalisation H est discontinue par construction, alors la résolution de (PP) est un problème difficile pour lequel on ne peut utiliser aucune des méthodes d'optimisation sans contrainte connues telles que les méthodes de gradient.

2.7.1 Méthode de pénalité extérieure

La difficulté mentionnée ci-dessus, peut être évitée en utilisant des fonctions de pénalisation continues et différentiables. Ainsi, dans la méthode dite de « pénalité extérieure », on peut prendre :

$$\begin{cases} h(y) = 0 & \text{si : } y \leq 0 \\ h(y) = y^2 & \text{si : } y > 0 \end{cases}$$

en posant :

$$H(x) = \frac{1}{2} \sum_{i=1}^m h(g_i(x)) = \frac{1}{2} \sum_{i=1}^m [g_i^+(x)]^2$$

où, $\forall x \in \mathfrak{R}^n$, $g_i^+(x) = \max(0, g_i(x))$, ($i = 1, \dots, m$).

le problème (P) est remplacé par le problème pénalisé :

$$(PP) \begin{cases} \text{Min } \varphi(x, r) = f(x) + r H(x) , (r > 0) \\ x \in \mathfrak{R}^n \end{cases}$$

où : r est appelé coefficient de pénalité , H est la fonction de pénalisation quadratique.

Notation : Dans ce qui suit, on note par : $\bar{x}(r)$ un minimum de $\varphi(x, r)$ pour r donné.

Le choix du coefficient de pénalité

Le choix d'une valeur appropriée du coefficient de pénalité r résulte d'un compromis : d'une part, r doit être suffisamment grand pour que le point $\bar{x}(r)$ obtenu soit proche de l'ensemble X (i.e. $H(\bar{x}(r))$, soit suffisamment faible), d'autre part, si r est choisi trop grand, la fonction φ peut être mal conditionnée, d'où les difficultés numériques dans la recherche de l'optimum sans contraintes. Ceci explique pourquoi, les méthodes de pénalité sont généralement mises en œuvre sous forme itérative de la façon suivante : On commence par choisir un coefficient de pénalité r_1 de valeur pas trop élevée, puis on résout le problème sans contrainte :

$$\text{Min}_{x \in \mathfrak{R}^n} \varphi(x, r_1) = f(x) + r_1 H(x)$$

Soit $\bar{x}(r_1)$ le point obtenu. Si la quantité $H(\bar{x}(r_1))$ est suffisamment faible, alors $\bar{x}(r_1)$ est une bonne approximation de l'optimum, et les calculs sont terminés. Sinon on choisit un coefficient $r_2 > r_1$, et on résout le nouveau problème sans contrainte :

$$\text{Min}_{x \in \mathfrak{R}^n} \varphi(x, r_2) = f(x) + r_2 H(x)$$

On obtient un nouveau point, $\bar{x}(r_2)$ et ainsi de suite.

2.7.3 Conditions de convergence des méthodes de pénalité

Les méthodes de pénalité ([16], [19]) convergent vers une solution optimale du problème (P) sous des conditions très peu restrictives, comme nous l'indique le théorème suivant :

Théorème 2.14 [19]. *Soit, H une fonction de pénalisation vérifiant :*

- $H(x) \geq 0, \forall x \in \mathfrak{R}^n$
- $H(x) = 0 \Leftrightarrow x \in X = \{x / g_i(x) \leq 0, (i = 1, \dots, m)\}$
- H est continue.

On suppose que f est continue, X est fermé, et que l'une des deux conditions suivantes est vérifiée :

- i) $f(x) \rightarrow +\infty$, quand $\|x\| \rightarrow +\infty$
- ii) X est borné et $H(x) \rightarrow +\infty$, quand $\|x\| \rightarrow +\infty$

Alors, lorsque le coefficient de pénalité r tend vers $+\infty$:

- *La suite $\bar{x}(r)$ admet au moins un point d'accumulation, et tout point d'accumulation de cette suite est une solution optimale (globale) du problème (P).*
- $H(\bar{x}(r)) \rightarrow 0$

Algorithme 2.5

Début

- (a) Poser : $k = 0$, Choisir : $r_0 = 1$ et $x^0 \in \mathfrak{R}^n, \varepsilon > 0$
- (b) Déterminer : x^{k+1} solution du problème $\text{Min}_{x \in \mathfrak{R}^n} \varphi(x, r_k)$ en partant de x^k
- (c) Si, $\|\nabla_x \varphi(x^{k+1}, r_k)\| < \varepsilon$, alors arrêter.
- (d) Choisir : $r_{k+1} > r_k$
- (e) Faire : $k \leftarrow k + 1$, et retourner en (b)

Fin.

2.7.3 Approximation des multiplicateurs de Kuhn et Tucker à l'optimum

Une caractéristique intéressante des méthodes de pénalité, est qu'elles fournissent de bonnes approximations des multiplicateurs optimaux de *Kuhn et Tucker*.

Il est prouvé dans [19] que si, $\bar{x}(r)$ est un point régulier on a :

$$\lim_{r \rightarrow +\infty} r \cdot g^+(\bar{x}(r)) = \hat{\lambda} \quad (2.8)$$

2.8 Le lagrangien augmenté

2.8.1 Motivation

Nous avons décrit deux types de méthodes permettant de remplacer le problème d'optimisation avec contraintes (P) par la résolution d'une séquence de problèmes d'optimisation sans contraintes (P').

- 1) Les méthodes de pénalité (extérieure) présente l'inconvénient que, pour obtenir un point $\bar{x}(r)$ réalisable, il faut faire tendre r vers $+\infty$; alors $\varphi(x, r)$ devient une fonction mal conditionnée sur laquelle les méthodes de gradient ont une convergence lente.
- 2) Les méthodes de dualité lagrangienne présentent l'inconvénient que, si λ^k converge vers l'optimum de dual $\hat{\lambda}$, les solution successives x^k obtenues ne convergent vers une solution primale que si $L(x, \lambda)$ a un minimum unique en x dans un voisinage $\hat{\lambda}$, (c'est le cas si : $L(x, \lambda)$ est strictement convexe en x).

Amélioration de la convergence

Les inconvénients mentionnés précédemment, peuvent être en partie éliminés en combinant les deux approches (Pénalité + Dualité), et on propose de résoudre une séquence de problèmes sans contraintes de la forme :

$$[\text{Max}]_{\lambda \in \mathfrak{R}^{m+}} [\text{Min}]_{x \in \mathfrak{R}^n} L_r(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \frac{r}{2} \sum_{i=1}^m g_i^+(x)^2, \quad (r > 0) \quad (2.9)$$

où : L_r est le lagrangien augmenté associé au problème (P).

2.8.2 Méthodes des multiplicateurs

Les méthodes des multiplicateurs ([13], [16], [19]) combinent les deux approches (Dualité + Pénalité). L'idée est la suivante :

Considérons, le lagrangien augmenté (2.9) :

- 1) Les méthodes de pénalité procurent de bonnes approximations du multiplicateur de Lagrange $\hat{\lambda}$ (pour r suffisamment grand), comme le montre la formule suivante :

$$\lim_{r \rightarrow +\infty} \lambda_r = r g^+(\bar{x}(r)) = \hat{\lambda} \quad (2.10)$$

- 2) Si le multiplicateur de Lagrange $\hat{\lambda}$ est connu, les méthodes de dualité donnent la solution \hat{x} (L est strictement convexe en x) en une seule étape en résolvant le problème suivant :

$$\hat{x} = \underset{x \in \mathfrak{R}^n}{\operatorname{argmin}} L(x, \hat{\lambda}) \quad (2.11)$$

Algorithme 2.6

Début

- (a) Poser : $k = 0$, $\lambda^0 \in \mathfrak{R}^{m+}$, $x^0 \in \mathfrak{R}^n$ et $r > r_0 \geq 0$, $\varepsilon > 0$
- (b) Déterminer : x^{k+1} solution du problème $\underset{x \in \mathfrak{R}^n}{\operatorname{Min}} L_r(x, \lambda^k)$ en partant de x^k
- (c) Si, $\| \nabla_x L_r(x^{k+1}, \lambda^k) \| < \varepsilon$, alors arrêter.
- (d) Mise à jour : $\lambda^{k+1} = \max(0, \lambda^k + r g(x^{k+1}))$
- (e) Faire : $k \leftarrow k + 1$, et retourner en (b)

Fin.

2.8.3 Méthodes de lagrangien augmenté

Les méthodes de lagrangien augmenté ([15], [17], [19], [23]) peuvent être considérées comme une amélioration des méthodes de dualité lagrangienne classique.

Considérons le problème pénalisé :

$$(P_r) \quad \begin{cases} \text{Min}_{x \in \mathfrak{R}^n} f(x) + \frac{r}{2} \|g^+(x)\|_2^2, & (r > 0) \\ g_i(x) \leq 0, & (i = 1, \dots, m) \end{cases}$$

Le problème dual augmenté s'écrit :

$$(D_r) \quad \begin{cases} \text{Max}_{\lambda} w_r(\lambda) = \text{Max}_{\lambda} [\text{Min}_{x \in \mathfrak{R}^n} L_r(x, \lambda)] \\ \lambda \in \mathfrak{R}^{m+} \end{cases}$$

La fonction duale augmentée $w_r(\lambda)$ est maximisée par l'algorithme du gradient projeté :

Algorithme 2.7 (d'Uzawa : lagrangien augmenté classique)

Début

- (a) Poser : $k = 0$, $\lambda^0 \in \mathfrak{R}^{m+}$, $x^0 \in \mathfrak{R}^n$ et $r > r_0 \geq 0$, $\varepsilon > 0$
- (b) Déterminer : x^{k+1} solution du problème $\text{Min}_{x \in \mathfrak{R}^n} L_r(x, \lambda^k)$ en partant de x^k
- (c) Si, $\|\nabla_x L_r(x^{k+1}, \lambda^k)\| < \varepsilon$, alors : arrêter.
- (d) Calculer : $\lambda^{k+1} = \max(0, \lambda^k + \rho_k g(x^{k+1}))$
- (e) Faire : $k \leftarrow k + 1$, et retourner en (b)

Fin.

Remarque 2.6 : Si le pas de déplacement ρ_k est choisi constant tel que, $\rho_k = r$: alors, il s'agit d'une méthode des multiplicateurs.

Le lagrangien augmenté de Rockefellar

Nous présentons ici, une méthode duale très efficace, basée sur le lagrangien augmenté de Rockefellar [23].

On considère dans \mathfrak{R}^n , le problème classique :

$$(P) \quad \begin{cases} \text{Min } f(x), & x \in \mathfrak{R}^n \\ g(x) \leq 0 \end{cases}$$

il peut évidemment s'écrire sous la forme :

$$(\tilde{P}_r) \begin{cases} \text{Min}_{x,y} f(x), & (x, y) \in \mathfrak{R}^n \times \mathfrak{R}^m \\ g(x) = y \\ y \leq 0 \end{cases}$$

où, y est une variable (vectorielle $y \in \mathfrak{R}^m$) artificielle.

Le lagrangien augmenté associé au problème (\tilde{P}_r) s'écrit :

$$L_r(x, y, \lambda) = f(x) + \lambda'(g(x) - y) + \frac{r}{2} \|g(x) - y\|_2^2 \quad (2.12)$$

La minimisation de (2.12) en y , peut être menée explicitement, puisque le problème est quadratique en y , avec un hessien diagonal qui est défini positif, et que l'on n'a que des contraintes de bornes sur la variable y .

plus précisément, $\nabla_y L_r(x, y, \lambda) = 0 \Leftrightarrow r(y - g(x)) - \lambda = 0$, d'où : $y = g(x) + \frac{\lambda}{r}$

et comme le hessien $\nabla_y^2 L_r(x, y, \lambda) = r \times I_{(m \times m)}$ est défini positif donc y est un minimum global, en suite on projette y sur l'orthant négatif, on trouve :

$$\bar{y} = \min(0, g(x) + \frac{\lambda}{r}) \Rightarrow g(x) - \bar{y} = \max(g(x), -\frac{\lambda}{r})$$

En remplaçant le terme $g(x) - \bar{y}$ par sa valeur dans (2.12), on obtient le lagrangien augmenté de *Rockefeller*, il se note \mathcal{L} .

$$\mathcal{L}_r(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i \max(g_i(x), -\frac{\lambda_i}{r}) + \frac{r}{2} \sum_{i=1}^m \left[\max(g_i(x), -\frac{\lambda_i}{r}) \right]^2 \quad (2.13)$$

ou encore :

$$\mathcal{L}_r(x, \lambda) = \text{Min}_{y \leq 0} L_r(x, y, \lambda) \quad (2.14)$$

le problème dual augmenté s'écrit :

$$(\tilde{D}_r) \begin{cases} \text{Max}_{\lambda} w_r(\lambda) = \text{Max}_{\lambda} [\text{Min}_{x \in \mathfrak{R}^n} \mathcal{L}_r(x, \lambda)] \\ \lambda \in \mathfrak{R}^m \text{ (libre)}, r > 0 \end{cases}$$

On peut utiliser une méthode de gradient pour maximiser la fonction duale augmentée, cela donne le schéma itératif suivant :

$$\begin{cases} \lambda^0 \in \mathfrak{R}^m \\ \lambda^{k+1} = \lambda^k + \rho_k \nabla w_r(\lambda^k) \end{cases}$$

où, les multiplicateurs λ_i , ($i = 1, \dots, m$) sont libres (i.e. non contraint au signe).

Algorithme 2.8 (d'Uzawa : lagrangien augmenté de *Rockefeller*)

Début

- (a) Poser : $k = 0, \lambda^0 \in \mathfrak{R}^m, x^0 \in \mathfrak{R}^n$ et $r > r_0 \geq 0, \varepsilon > 0$
- (b) Calculer : $y^k = \min(0, g(x^k) + \frac{\lambda^k}{r})$
- (c) Déterminer : x^{k+1} solution du problème $\text{Min}_{x \in \mathfrak{R}^n} L_r(x, y^k, \lambda^k)$ en partant de x^k
- (d) Si, $\| \nabla_x L_r(x^{k+1}, y^k, \lambda^k) \| < \varepsilon$, alors : arrêter.
- (e) Calculer : $\lambda^{k+1} = \lambda^k + \rho_k (g(x^{k+1}) - y^k)$
- (f) Faire : $k \leftarrow k + 1$, et retourner en (b)

Fin.

2.9 Application au problème quadratique à contraintes linéaires d'inégalité

La résolution du problème quadratique constitue la partie la plus importante dans la résolution globale du problème de la couverture comptable (voir chapitre 3). Donc, un algorithme efficace (rapide) doit être appliqué.

Dans cette section, nous appliquons, la méthode du lagrangien augmenté due à *Rockefeller* [23], pour la résolution du problème quadratique suivant :

$$(Q) \begin{cases} \text{Min}_x \frac{1}{2} \|Ax - b\|_2^2, & x \in \mathfrak{R}^n \\ 0 \leq x \leq d \\ \alpha \leq Cx \leq \beta \end{cases}$$

où : $A \in \mathfrak{R}^{m \times n}, b \in \mathfrak{R}^m, C \in \mathfrak{R}^{p \times n}, d \in \mathfrak{R}^n$, et $\alpha \in \mathfrak{R}^p, \beta \in \mathfrak{R}^p, (\alpha_i \leq \beta_i, i = 1, \dots, p)$

Principe de la méthode :

La méthode consiste à reformuler le problème de minimisation (Q) en relaxant les contraintes à l'aide des multiplicateurs de Lagrange, donc de transformer ce problème en celui de la recherche du point selle d'un lagrangien augmenté. Elle est décrite comme suit :

Le problème peut s'écrire :

$$(\tilde{Q}) \begin{cases} \text{Min}_x \frac{1}{2} \|Ax - b\|^2 \\ x = y & \text{----- (1)} \\ Cx = s & \text{----- (2)} \\ x \in \mathfrak{R}^n, 0 \leq y \leq d, \alpha \leq s \leq \beta \end{cases}$$

on dualise et on pénalise les deux contraintes (1) et (2), le lagrangien augmenté associé au problème \tilde{Q} s'écrit :

$$L_r(x, y, s, \lambda, \mu) = \frac{1}{2} \|Ax - b\|^2 + \lambda'(x - y) + \frac{r}{2} \|x - y\|^2 + \mu'(Cx - s) + \frac{r}{2} \|Cx - s\|^2 \quad (2.15)$$

avec : $x \in \mathfrak{R}^n, 0 \leq y \leq d, \alpha \leq s \leq \beta$ et $\lambda \in \mathfrak{R}^n, \mu \in \mathfrak{R}^p$

où, r : le coefficient de pénalité ($r > 0$)

λ : le multiplicateur de Lagrange associé à la contrainte (1)

μ : le multiplicateur de Lagrange associé à la contrainte (2)

La minimisation du lagrangien augmenté (2.15) en (x, y, s) peut être déterminée en deux étapes :

a) la première étape :

Pour (x, λ, μ) fixés, la minimisation du lagrangien (2.15) en variables (y, s) peut être résolue analytiquement par la résolution du problème suivant :

$$\begin{cases} \text{Min}_{y,s} \left[\frac{r}{2} y^t y - (r x + \lambda)^t y + \frac{r}{2} s^t s - (r Cx + \mu)^t s \right] \\ 0 \leq y \leq d \text{ et } \alpha \leq s \leq \beta, \text{ avec : } y \in \mathfrak{R}^n, s \in \mathfrak{R}^p \end{cases}$$

la solution (y, s) est obtenue par les formules suivantes :

$$\begin{cases} y = \max(0, \min(x + \frac{\lambda}{r}, d)) \\ s = \max(\alpha, \min(Cx + \frac{\mu}{r}, \beta)) \end{cases} \quad (2.16)$$

b) la deuxième étape :

Pour (y, s, λ, μ) fixés, la minimisation du lagrangien (2.15) en x peut être résolue : par la résolution du problème suivant :

$$\begin{cases} \text{Min}_x \left[\frac{1}{2} x^t [A^t A + r(I_{n \times n} + C^t C)] x - [A^t b + r y - \lambda + C^t (r s - \mu)]^t x \right] \\ x \in \mathfrak{R}^n \end{cases}$$

la solution x est obtenue numériquement par la résolution du système linéaire suivant :

$$[A^t A + r(I_{n \times n} + C^t C)] x = A^t b + r y - \lambda + C^t (r s - \mu) \quad (2.17)$$

où : $I_{n \times n}$ est la matrice identité (unité) de taille $(n \times n)$.

2.9.1 Algorithme d'Uzawa à pas constant

La méthode précédente, appliquée au problème (Q) peut être décrite par l'algorithme suivant :

Algorithme 2.9

Début

(a) Poser : $k = 0$, $\lambda^0 \in \mathfrak{R}^n$, $\mu^0 \in \mathfrak{R}^p$, $x^0 \in \mathfrak{R}^n$ et $r > r_0 \geq 0$, $\varepsilon > 0$

(b) Calculer :

$$\begin{cases} y^k = \max(0, \min(x^k + \frac{\lambda^k}{r}, d)) \\ s^k = \max(\alpha, \min(Cx^k + \frac{\mu^k}{r}, \beta)) \end{cases}$$

(c) Résoudre le système linéaire :

$$[A^t A + r(I_{n \times n} + C^t C)] x^{k+1} = A^t b - \lambda^k + r y^k + C^t (r s^k - \mu^k)$$

(d) Si, $\max(\|x^{k+1} - y^k\|, \|Cx^{k+1} - s^k\|) < \varepsilon$, alors : arrêter.

(e) Calculer :

$$\begin{cases} \lambda^{k+1} = \lambda^k + \rho(x^{k+1} - y^k) \\ \mu^{k+1} = \mu^k + \rho(Cx^{k+1} - s^k) \end{cases}$$

(f) Faire : $k \leftarrow k + 1$, et retourner en (b)

Fin.

Remarque 2.7 : La tâche la plus coûteuse dans l'algorithme (2.9) est la résolution du système linéaire (2.17) à l'étape (c). La matrice symétrique : $[A^t A + r(I_{n \times n} + C^t C)]$ est définie positive ($r > 0$), et à chaque itération k , on doit résoudre un système linéaire. Il est convenable de la factoriser en *Cholesky* (sans racine carrée) [21], et de résoudre deux systèmes triangulaires à chaque itération.

Convergence de l'algorithme (2.9)

Pour les problèmes quadratiques à contraintes linéaires, nous avons le résultat théorique suivant, qui a été démontré par : *Gabay -Mercier* [12].

Théorème 2.15 [17] : *Si, les contraintes sont linéaires, l'algorithme converge si :*

$$0 < \rho < 2r.$$

Remarque 2.8 : *Rockafellar* [23] a défini la méthode du lagrangien augmenté pour des programmes convexes et démontré sa convergence globale et linéaire. Pour plus de détails, nous renvoyons le lecteur à la référence [24].

2.9.2 Complexité de l'algorithme d'Uzawa à pas constant

Généralités sur la complexité algorithmique

Dans ce paragraphe, nous abordons brièvement la problématique de la comparaison des algorithmes.

Beaucoup de problèmes sont de nature à pouvoir être résolus par une variété d'algorithmes différents. Il se pose alors le problème du choix de meilleur algorithme.

La notion de complexité d'un algorithme ([18], [26]) est utilisée pour comparer les performances des algorithmes. Le principal critère de comparaison est le temps d'exécution. Afin d'obtenir des résultats qui soient indépendants de la performance d'un ordinateur ou d'un compilateur particulier, on mesure le temps en comptant le nombre d'opérations élémentaires exécutées.

La notion de base en complexité des algorithmes est la définition de la taille du problème.

Définition 2.12 : La taille du problème, notée n , est la quantité de données qui doivent être traitées pour résoudre le problème en question.

Par la suite, on tentera d'exprimer le temps nécessaire au déroulement de l'algorithme comme une fonction de la taille du problème.

Définition 2.13 : On appelle complexité d'un algorithme, le nombre d'opérations élémentaires qui est nécessaire pour résoudre un problème de taille n . Cette complexité est donc proportionnelle au temps d'exécution effectif de l'ordinateur.

Nombre d'opérations dans le pire des cas

Pour compter le nombre d'opérations élémentaires, la mesure la plus utilisée est le « pire cas » : il s'agit d'une borne supérieure du nombre d'opérations nécessaires pour résoudre un problème de taille n dans sa configuration la plus défavorable pour l'algorithme étudié. Dans ce genre de calcul, on s'intéresse davantage à évaluer la performance de l'algorithme pour les grandes tailles que pour les petites.

L'approche utilisée pour calculer la complexité d'un algorithme est basée sur deux concepts principaux, la notion mathématique de proportionnalité et celle de comportement asymptotique.

Définition 2.14 : Étant données deux fonctions $f, g : \mathbb{N} \rightarrow \mathbb{N}$, on dit que f est $O(g)$ s'il existe une constante $c > 0$ telle que : $f(n) \leq c \times g(n)$ pour tous $n \in \mathbb{N}$.

Définition 2.15 : Une fonction est dite polynomiale si elle est $O(g)$ et si g est un polynôme en n . Ou encore s'il existe deux constantes c et k telles que : $f(n) \leq c n^k$ pour tous $n \in \mathbb{N}$.

Définition 2.16 : Un algorithme est dit « polynomial » si le nombre d'opérations élémentaires nécessaires pour résoudre un exemple du problème de taille n est une fonction polynomiale en n . Un algorithme est considéré « efficace » si, seulement si, il est polynomial.

a) Calcul de la taille du problème

La taille du problème quadratique (Q), notée M , peut être estimée comme suit :

$$M = O((m + p) \times n) \quad (2.18)$$

où,

n : le nombre de colonnes de la matrice A (nombre des inconnus).

m : le nombre de lignes de la matrice A .

p : le nombre de contraintes.

b) Complexité temporelle de l'algorithme (2.9)

Le calcul exact de la complexité est difficile (voire impossible dans certains cas). Seul le terme dominant (significatif) de la formule exprimant la complexité est retenu. On ignore également le coefficient multiplicateur constant du terme dominant.

Dans le tableau (2.1), nous passons en revue les différentes étapes de l'algorithme afin d'évaluer son temps d'exécution. Pour ce faire, nous attribuons un coût en temps à chaque instruction, et nous comptons le nombre d'exécutions de chacune des instructions.

Instructions	Coût	Fréquence
$A^t A + r (I_{n \times n} + C^t C)$	$O((m + p) \times n^2)$	1 (fois)
Factorisation de <i>Cholesky</i>	$O(n^3)$	
$y_i = \max(0, \min(x_i, d_i)) \quad (i = 1, \dots, n)$	$O(n)$	N_{iter} (fois) : le nombre d'itérations effectuées par l'algorithme
$s_i = \max(\alpha_i, \min([Cx]_i, \beta_i)) \quad (i = 1, \dots, p)$	$O(p \times n)$	
$A^t b - \lambda + r y + C^t (r s - \mu)$	$O((m + p) \times n)$	
Résolution du système linéaire factorisé	$O(n^2)$	
$\max(\ Cx - s\ _\infty, \ x - y\ _\infty) < \varepsilon$	$O(p \times n)$	
$\lambda_i = \lambda_i + \rho (x_i - y_i) \quad (i = 1, \dots, n)$	$O(n)$	
$\mu_i = \mu_i + \rho ([Cx]_i - s_i) \quad (i = 1, \dots, p)$	$O(p \times n)$	

Tableau 2.1 : Illustration du calcul de la complexité temporelle de l'algorithme d'*Uzawa* à pas fixe (2.9).

En vertu du tableau (2.1), la complexité temporelle de l'algorithme peut s'écrire ainsi :

$$N_{\text{oper}} = O((m + p + n) \times n^2) + N_{\text{iter}} \times O((m + p + n) \times n) \quad (2.19)$$

où :

n : le nombre de colonnes de la matrice A (le nombre des inconnus).

m : le nombre de lignes de la matrice A .

p : le nombre de contraintes.

N_{iter} : le nombre d'itérations effectuées par l'algorithme.

2.9.3 Exemple et résultats numériques

Nous considérons l'exemple suivant :

$$(E) \begin{cases} \text{Min}_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2 \\ \text{sous :} \\ 0 \leq x_j \leq 1, \quad (j = 1, \dots, 3) \\ 1,6 \leq \sum_{j=1}^3 x_j \leq 2,5 \end{cases}$$

où :

$$A = \begin{pmatrix} 5 & 2 & 3 \\ 5 & 3 & 3 \\ 5 & 2 & 3 \\ 5 & 3 & 2 \\ 5 & 2 & 2 \\ 5 & 3 & 2 \end{pmatrix} \quad b = \begin{pmatrix} 10 \\ 11 \\ 10 \\ 10 \\ 9 \\ 10 \end{pmatrix}$$

La solution est :

$$\hat{x} = (1,00 \quad 0,75 \quad 0,75)^t$$

Le coût optimal vaut alors :

$$\frac{1}{2} \|A\hat{x} - b\|^2 = 4,75$$

Les contraintes sont satisfaites :

$$\begin{cases} 0 \leq \hat{x}_j \leq 1, \quad (j = 1, \dots, 3) \\ \sum_{j=1}^3 \hat{x}_j = 2,5 \end{cases}$$

Nous avons expérimenté l'algorithme (2.9) sur cet exemple en utilisant les paramètres suivants :

- Le nombre maximal d'itérations, noté MaxIter, est fixé à 1000 ; (MaxIter = 1000).
- La précision requise sur les contraintes $[x = y, Cx = s]$, notée Tol, est de l'ordre de $(\varepsilon = 10^{-6})$, où : $\text{Tol} = \max(\|x - y\|_\infty, \|Cx - s\|_\infty)$
- Le test d'arrêt utilisé dans l'algorithme est : Si, $[(N_{\text{iter}} > \text{MaxIter}) \text{ ou } (\text{Tol} < \varepsilon)]$, arrêter.

2.9.4 Influence de choix du pas ρ sur le nombre d'itérations

La mise en œuvre de l'algorithme (2.9) est très simple, la difficulté essentielle est de déterminer la valeur optimale du paramètre ρ pour une valeur fixée de r . Pour cela, nous étudions expérimentalement l'influence du paramètre ρ sur le nombre effectif d'itérations, N_{iter} fourni par l'algorithme.

Les tableaux ci-dessous, représentent pour deux valeurs fixées du coefficient de pénalité r , le nombre d'itérations effectuées par l'algorithme en fonction de ρ .

$r = 30$	ρ	0	3	6	9	12	18	30	42	51	56	59	60
	N_{iter}	1000	415	204	133	97	61	35	33	34	69	288	1000

Tableau 2.2 : Le nombre d'itérations (N_{iter}) en fonction du paramètre ρ , (cas : $r = 30$)

$r = 50$	ρ	0	5	10	15	20	30	50	60	75	90	98	100
	N_{iter}	1000	290	140	90	65	37	27	35	34	43	227	1000

Tableau 2.3 : Le nombre d'itérations (N_{iter}) en fonction du paramètre ρ , (cas : $r = 50$)

La figure (2.7) représente, pour une valeur fixé du coefficient de pénalisation r , la variation du nombre d'itérations effectuées par l'algorithme en fonction de ρ .

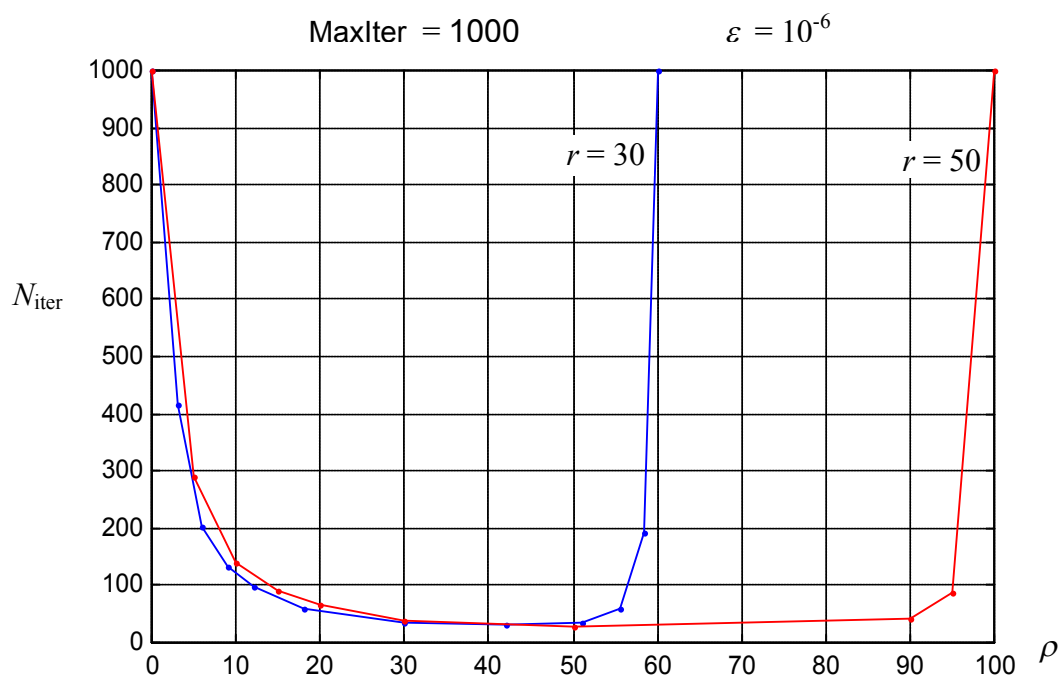


Figure 2.7 : Variation du nombre d'itérations (N_{iter}) en fonction de ρ .

La figure (2.7) montre l'importance de choix du paramètre ρ :

- L'algorithme converge si : $0 < \rho < 2r$. (voir le théorème (2.15)).
- Une estimation de bon choix du paramètre ρ est de prendre ($\bar{\rho} = r$).
- Le nombre d'itérations N_{iter} , dépend faiblement des variations de valeur du coefficient r (r assez grand) d'où la robustesse de l'algorithme.

Accélération de la convergence

Théoriquement, la convergence est d'autant plus rapide que r est grand, mais pratiquement, le choix de r est un problème délicat, du fait que la fonction du lagrangien augmenté peut être très mal conditionnée dans ce cas, d'où les difficultés numériques ([17],[19]). Afin d'accélérer la convergence, pour r donné, le choix optimal de ρ peut être ajusté numériquement.

Nous tentons ici, d'estimer la valeur optimale du paramètre ρ en fonction du coefficient de pénalisation r .

En fixant la précision du calcul à ($\varepsilon = 10^{-6}$), le nombre d'itérations : N_{iter} dépend essentiellement de deux paramètres (r et ρ), il peut s'exprimer par le modèle suivant :

$$N_{iter} = \phi(r, \rho) \quad (2.20)$$

où : ϕ est une fonction qui représente l'algorithme « d'Uzawa à pas constant » (2.9).

Pour différentes valeurs fixées de r , on calcule numériquement la valeur optimale de ρ . Il s'agit, alors de résoudre le problème d'optimisation suivant :

$$\left[\begin{array}{l} \hat{N}_{iter} = \underset{\rho}{\text{Min}} \phi(r, \rho) \\ 0 \leq \rho \leq 2r \end{array} \right. \quad (*)$$

qui est un problème d'optimisation non différentiable. Pour le résoudre, nous utilisons la commande de minimisation « fmin » fournie du logiciel de calcul numérique « Matlab® ».

Les résultats obtenus sont donnés dans le tableau (2.4).

Notons : $H_r = [A^t A + r(I_{n \times n} + C^t C)]$.

r	5	10	20	30	50	80	100
$\hat{\rho}$	7,41	14,82	26,76	28,60	55,69	84,98	103,64
\hat{N}_{iter}	125	61	33	32	26	26	27
$\text{cond}_2(H_r)$	34,83	21,87	13,36	10,57	7,98	6,49	5,99

Tableau 2.4 : ρ optimal (expérimental) en fonction de r .

D'après le tableau ci-dessus, on constate effectivement que :

- La convergence s'améliore rapidement lorsque r croît.
- Il n'est pas nécessaire de faire tendre r vers l'infini pour résoudre le problème.
- La valeur optimale de ρ est légèrement supérieure à celle du coefficient de pénalité r .

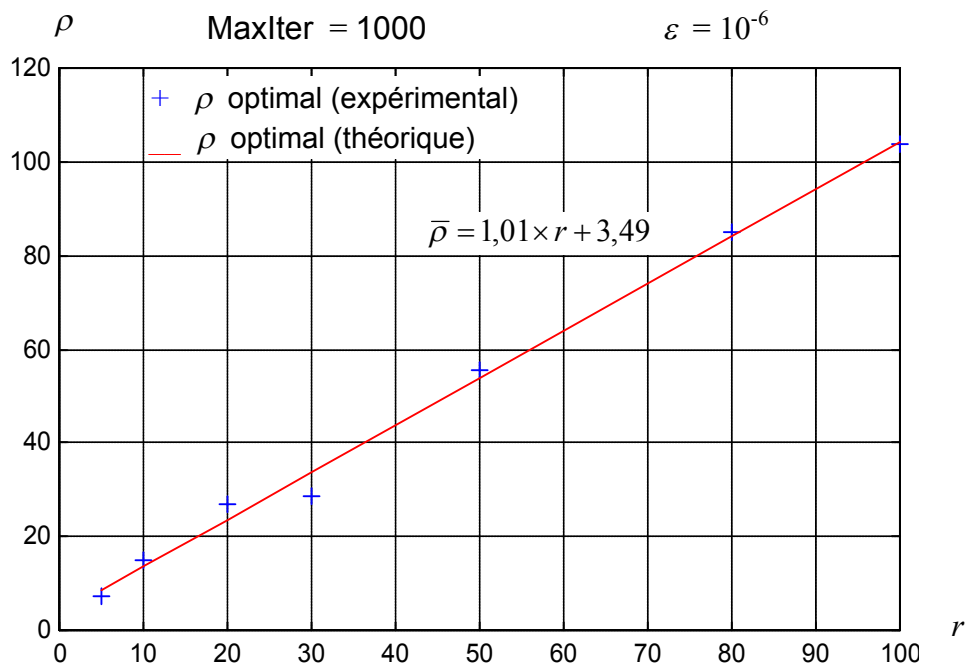


Figure 2.8 : ρ optimal théorique et expérimental en fonction de r

Sur la figure (2.8), on constate que la valeur optimale (expérimentale) de ρ obéit au modèle de régression linéaire suivant :

$$\bar{\rho} = 1,01 \times r + 3,49 \quad (2.21)$$

avec un coefficient de détermination, noté R^2 , ($R^2 = 99,81\%$), traduisant la validité du modèle linéaire, (i.e. la part de variation de $\hat{\rho}$ expliquée par le modèle de régression $\bar{\rho}$).

D'après le modèle linéaire (2.21), on constate que, pour un coefficient de pénalisation r fixé, la valeur optimale (théorique) du paramètre ρ est légèrement supérieure à la valeur de r . De plus, en vertu du théorème (2.15), le paramètre ρ peut être choisi par la formule suivante :

$$\rho = w \times r \quad (2.22)$$

où , r : le coefficient de pénalité ($r > 0$), qu'il est inutile de faire tendre vers l'infini.

w : un paramètre positif, tel que : $1 \leq w < 2$.

2.10 Conclusion

Dans ce chapitre, nous avons présenté des méthodes d'optimisation duales, basée sur les techniques de lagrangien augmenté pour la résolution des problèmes d'optimisation sous contraintes.

Pour la résolution de problèmes quadratiques à contraintes linéaires d'inégalité, l'algorithme d'*Uzawa* à pas constant (2.9) apparaît très performant. Cependant, cette performance est dépendante de deux facteurs qui contrôlent l'algorithme (r et ρ).

La convergence de l'algorithme est d'autant plus rapide que r est grand, mais pratiquement, le choix de r est un problème délicat. Tandis que, pour r donné, le choix optimal du ρ peut être estimé. Pour cela, nous avons étudié numériquement l'impact de choix du paramètre ρ pour un coefficient de pénalisation r fixé, et nous avons remarqué qu'un choix approprié de ce paramètre joue un rôle important, car il peut réduire énormément le nombre d'itérations effectuées par l'algorithme.

La prévision optimale (théorique) du paramètre ρ est un problème délicat. De ce fait, nous avons pu établir empiriquement une relation linéaire qui peut exister entre le coefficient de pénalité r et le paramètre optimal (approché) ρ .

CHAPITRE 3

PROBLÈME DE LA COUVERTURE COMPTABLE

3.1 Introduction

Ce chapitre a pour objet, une présentation de la modélisation du problème et aux méthodes de résolution adaptées.

En premier lieu, nous présentons l'objectif et le principe de l'opération de couverture, ensuite nous abordons la formulation mathématique du problème.

Nous présentons aussi les méthodes de résolution. Tout d'abord, une méthode exacte pour les exemples de petite taille du problème, ensuite une méthode approchée, fondée sur une heuristique de type gloutonne, et qui est plus adaptée aux exemples de plus grande taille. Cette heuristique est tirée du rapport [29].

Présentation générale

La réglementation prudentielle bancaire joue un rôle essentiel afin de mesurer et de contrôler la qualité des couvertures, elle impose des normes visant à maintenir le niveau du capital que doit détenir la banque (Capital Adequacy) sur chaque portefeuille financier en fonction du risque associé. Dans ce travail, nous adoptons la norme : IAS39 ([32], [35]).

La norme IAS39 (*International Accounting Standard*), est une norme internationale qui impose les principes de comptabilisation et d'évaluation des instruments financiers. L'application de la norme est importante pour les institutions financières, elle leur permet de réduire le capital requis et de libérer des fonds propres afin de générer d'autres actifs, d'autre part, cette norme impose des règles strictes à respecter et qui sont : la comptabilisation et l'évaluation de tous les instruments financiers à leur juste valeur (prix du marché), la documentation de toutes les couvertures et la vérification en permanence leur efficacité.

Selon la norme IAS39, une opération de couverture est celle par laquelle une institution financière se protège contre un risque de variation du prix par une deuxième transaction. Les instructions suivantes doivent être respectées.

- Couvrir signifie, désigner certaines transactions comme instruments de couverture, les variations de prix de ces transactions doivent absolument compenser celles de l'élément couvert.
- Un élément couvert (Hedged Item) est un actif, un passif, ou une transaction future qui expose l'institution financière à un risque au niveau de l'évaluation.
- Un instrument de couverture (Hedging Item) est un instrument dérivé, un actif ou un passif qui compense les variations de prix attendues de l'élément couvert.
- Il est permis d'attribuer deux ou plusieurs instruments de couverture à un élément couvert, et il est aussi permis d'attribuer un instrument de couverture à plusieurs éléments couverts.

Dans ce travail, nous nous intéressons à la couverture comptable sous la norme de l'IAS39. Les éléments couverts sont : les obligations (ou les prêts), et les instruments de couverture utilisés sont : les swaps de taux d'intérêt standard (*Plain Vanilla Swaps*).

La couverture comptable, consiste à réduire au minimum l'exposition aux variations de prix de la totalité ou d'une partie d'un portefeuille financier qui est attribuable à un risque de taux d'intérêt. D'après la norme IAS39, les conditions suivantes doivent être strictement respectées :

- L'efficacité de la couverture doit pouvoir être évaluée de manière fiable.
- La couverture doit être efficace sur toute la période couverte.

Appréciation de l'efficacité de la couverture.

Une couverture est dite efficace, si elle est efficace au début de la couverture et pendant toute sa durée de vie, le gestionnaire financier peut s'attendre à ce que les variations de prix de l'élément couvert et de l'instrument de couverture se compensent presque intégralement. C'est le cas lorsque l'évolution des variations de prix se situe entre 80 % et 120 %.

3.2 Position et modélisation du problème

Dans ce paragraphe, nous posons le problème de la couverture comptable dans le cas d'un seul swap, ensuite nous allons l'étendre au cas de plusieurs swaps afin de traiter les problèmes réels.

Avant, d'entamer la modélisation du problème, nous rappelons l'objectif et le principe de la couverture comptable.

Objectif

La couverture comptable vise à atténuer le risque engendré par les fluctuations des taux d'intérêt sur la valeur du marché d'un portefeuille financier, et cela revient à réduire au minimum l'amplitude des variations de prix de ce portefeuille, (voir la figure suivante).

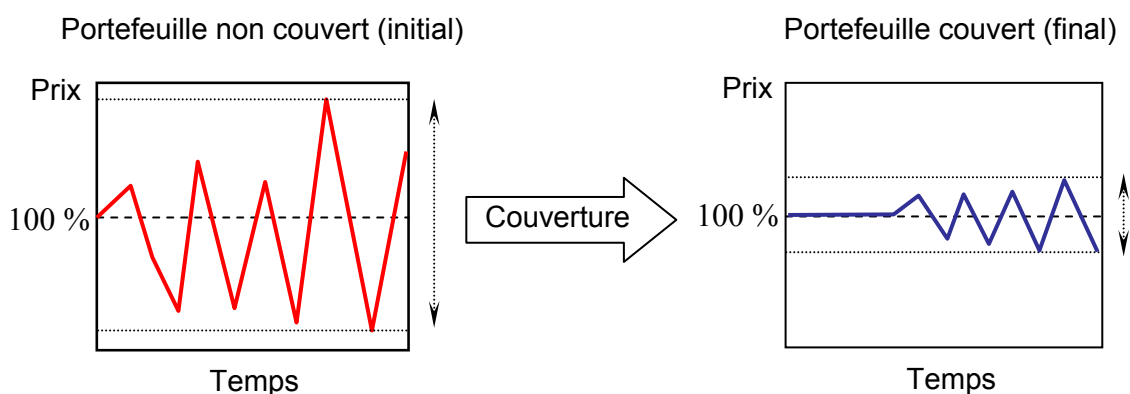


Figure 3.1 : Réduction de l'amplitude des variations de prix du portefeuille.

Principe

L'opération de la couverture consiste à amener, pour un horizon de temps donné, le portefeuille initial à un nouveau portefeuille dont le prix est moins sensible aux changements des taux. Cela, peut être fait en ajoutant des instruments de couverture (produits dérivés) au portefeuille initial. En effet, ces instruments permettent de modifier et d'optimiser (minimiser) l'amplitude des variations de prix du nouveau portefeuille. Cela peut être schématisé par la figure (3.2).

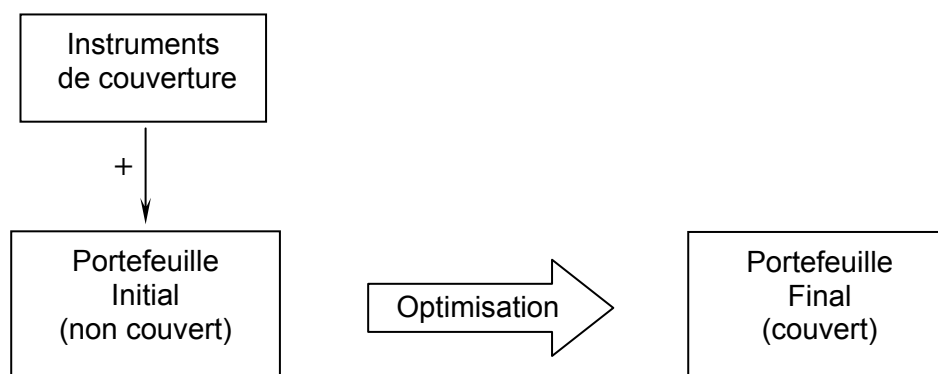


Figure 3.2 : Schéma conceptuel de la couverture comptable.

Modélisation

Soient :

- $\{L_1, L_2, \dots, L_k\}$ un portefeuille de prêts à taux fixe (éléments couverts).
- S un swap payeur (instrument de couverture).

Le swap S peut couvrir la totalité ou une partie du prêt L_j , ($j = 1, \dots, k$). L'instrument couvert (ou structuré), noté H , peut donc s'écrire :

$$H = \sum_{j=1}^k x_j L_j \quad (3.1)$$

où, x_j : indique la portion couverte du prêt L_j , ($j = 1, \dots, k$).

Notons, I : l'instrument synthétique qui est composé du swap S et de l'instrument couvert H . Il s'écrit comme suit :

$$I = S + \sum_{j=1}^k x_j L_j \quad (3.2)$$

et, $V(Y, r)$: la juste valeur d'un instrument Y à l'instant ($t = 0$), elle est donnée par :

$$V(Y, r) = \sum_{t=1}^T \frac{F_t(Y)}{(1+r_t)^t} \quad (3.3)$$

où, r : le vecteur (r_1, r_2, \dots, r_T) désigne la courbe des taux de référence.

T : la maturité (ou l'échéance) de l'instrument financier Y .

$F_t(Y)$: le flux monétaire de l'instrument financier Y , délivré à la date t .

La juste valeur de l'instrument synthétique I à l'instant ($t = 0$) s'écrit :

$$V(I, r) = V(S, r) + \sum_{j=1}^k x_j V(L_j, r) \quad (3.4)$$

On rappelle que le prix du swap S peut s'écrire ainsi :

$$V(S, r) = V(J_V, r) - V(J_F, r) \quad (3.5)$$

où : J_V, J_F sont la jambe variable, et la jambe fixe du swap S .

alors, le prix de l'instrument I s'écrit plus justement :

$$V(I, r) = \underbrace{V(J_V, r)}_{\approx 100\%} + \underbrace{\left[-V(J_F, r) + \sum_{j=1}^k x_j V(L_j, r) \right]}_{\text{variations en \%}} \quad (3.6)$$

la valeur de la jambe variable du swap (J_V) est très proche du pair, (i.e. $\approx 100\%$ de la valeur nominale du swap), donc, elle est très peu sensible aux changements des taux.

Nous cherchons à réduire au minimum l'amplitude des variations de prix de l'instrument I .

La valeur du marché de l'instrument I , pour une variation infinitésimale (dr) de la courbe des taux de référence, est donnée par le développement de *Taylor* :

$$V(I, r + dr) = V(I, r) + \nabla_r V(I, r) \cdot dr + O(\|dr\|^2) \quad (3.7)$$

en négligeant les termes d'ordre supérieurs, l'écart de la variation de prix de l'instrument I est estimé comme suit :

$$|V(I, r + dr) - V(I, r)| \approx |\nabla_r V(I, r) \cdot dr|$$

cet écart peut être majoré par l'inégalité suivante :

$$|\nabla_r V(I, r) \cdot dr| \leq \|\nabla_r V(I, r)\|_2 \cdot \|dr\|_2 \quad (3.8)$$

D'après (3.8), on peut ramener la réduction de l'amplitude des variations de prix, à celle de la norme euclidienne suivante, (appelée aussi *Key Rate Duration*) : $\|\nabla_r V(I, r)\|_2^2$ et qui est indépendante des variations des taux d'intérêt : (dr).

Le vecteur gradient de l'instrument synthétique I s'écrit :

$$\nabla V_r(I, r) = \nabla V_r(S, r) + \sum_{j=1}^k x_j \nabla V_r(L_j, r)$$

où, encore :

$$\nabla V_r(I, r) = \underbrace{\nabla V_r(J_V, r)}_{\approx 0} + \left[-\nabla V_r(J_F, r) + \sum_{j=1}^k x_j \nabla V_r(L_j, r) \right] \quad (3.9)$$

Notons, $\eta(\cdot) \equiv \nabla_r V(\cdot, r)$, le problème revient à réduire au minimum l'écart quadratique de la sensibilité de l'instrument : I . Cet écart est défini comme suit :

$$F(x) = \left\| \eta(S) + \sum_{j=1}^k x_j \eta(L_j) \right\|_2^2 \quad (3.10)$$

Le problème consiste à chercher, les portions des prêts : \hat{x}_j , ($j = 1, \dots, k$) telles que l'écart quadratique (3.10) soit minimum. Cette opération doit tenir compte des conditions d'efficacité qui sont établies par la norme : IAS39.

L'efficacité de la couverture comptable

En matière de l'efficacité, la norme IAS39 impose des règles prudentielles strictes dont le but est de limiter le risque. Elle exige que les variations de prix de l'instrument de couverture S et celles de l'instrument couvert H se compensent presque intégralement (i.e. presque la même amplitude, mais de signe opposé). Pour cela, la norme a adopté le ratio suivant, afin de qualifier les couvertures efficaces.

$$R_{eff} := \frac{|V(S, r + dr) - V(S, r)|}{\sum_{j=1}^k |V(L_j, r + dr) - V(L_j, r)| x_j}, \quad (dr \in \mathfrak{R}^{T_{max}}) \quad (3.11)$$

où, dr : un changement infinitésimal de la courbe des taux de référence.

T_{max} : la durée de vie de couverture.

Une couverture est considérée efficace si le ratio (3.11), vérifie la double inégalité suivante, pendant toute la durée de couverture :

$$80\% \leq R_{eff} \leq 120\% \quad , \quad (\forall dr \in \mathfrak{R}^{T_{max}}) \quad (3.12)$$

En pratique, la condition (3.12) ne peut être vérifiée que pour des changements des taux d'intérêt, notés (Δr) , supposés connus à l'avance. Par la suite, ces changements sont appelés : « scénarios ».

Dans ce qui suit, nous ne considérons que trois scénarios, les plus constatés sur le marché monétaire. Ils se notent comme suit :

($i = 1$) : le premier scénario, désigne le changement parallèle de la courbe des taux vers le haut de (+1%) : $\Delta^{(1)}r_t = +1\%$, ($t = 1, \dots, T_{\max}$).

($i = 2$) : le deuxième scénario, désigne le changement parallèle de la courbe des taux vers le bas de (-1%) : $\Delta^{(2)}r_t = -1\%$, ($t = 1, \dots, T_{\max}$).

($i = 3$) : le troisième scénario, désigne un changement d'inversion (ou de rotation) de la courbe des taux ; $\Delta^{(3)}r_t = \text{rotation}(t)$, ($t = 1, \dots, T_{\max}$). Ce scénario est anticipé par le gestionnaire.

La figure suivante, représente les différents scénarios considérés.

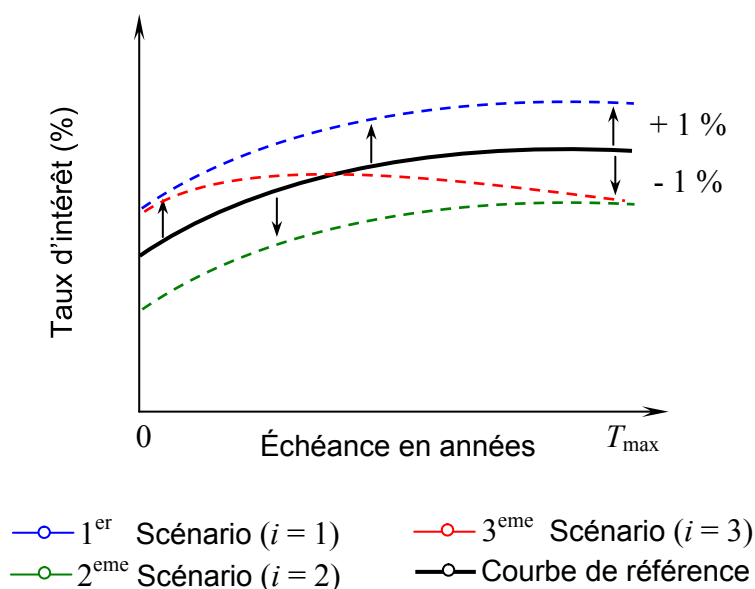


Figure 3.3 : Les différents scénarios de la courbe des taux d'intérêt.

Le ratio (3.11) peut être ramené au ratio suivant :

$$R_{eff}^i := \frac{|V^i(S,r) - V(S,r)|}{\sum_{j=1}^k |V^i(L_j,r) - V(L_j,r)| x_j}, \quad (i=1,\dots,3) \quad (3.13)$$

où : $V^i(Y,r)$ désigne la juste valeur de l'instrument Y , selon le scénario i . Elle est donnée par la formule :

$$V^i(Y) = \sum_{t=1}^T \frac{F_t(Y)}{(1+r_t^{(i)})^t}, \quad (i=1,\dots,3)$$

où, $r_t^{(i)}$: le taux d'actualisation de la période t , selon le scénario i .

avec : $r_t^{(i)} = r_t + \Delta^{(i)} r_t$, $(i=1,\dots,3)$ et $(t=1,\dots,T)$

la condition (3.12) devient :

$$lb^i \leq R_{eff}^i \leq ub^i, \quad (i=1,\dots,3) \quad (3.14)$$

où, lb^i et ub^i , $(i=1,\dots,3)$: sont les bornes inférieures et supérieures, du ratio d'efficacité, pour chaque scénario i , $(i=1,\dots,3)$. Elles sont égales, respectivement, à 80% et 120%.

Les conditions non linéaires (3.14) peuvent se ramener aux conditions linéaires de la manière suivante :

posons : $a^i = |V^i(S) - V(S)|$, $(i=1,\dots,3)$

et : $C_j^i = |V^i(L_j) - V(L_j)|$, $(i=1,\dots,3)$, $(j=1,\dots,k)$

on obtient alors : $\frac{a^i}{ub^i} \leq \sum_{j=1}^k C_j^i x_j \leq \frac{a^i}{lb^i}$, $(i=1,\dots,3)$ (3.15)

D'après la norme IAS39, si l'ensemble de contraintes (3.15) n'est pas satisfait, la couverture devient inefficace et doit être rejetée.

Notre objectif, est de réduire au minimum l'écart quadratique (3.10), tout en respectant les contraintes suivantes :

(1) les contraintes de disponibilité :

$$0 \leq x_j \leq d_j, (j = 1, \dots, k).$$

où, d_j : la portion disponible du $j^{\text{ième}}$ prêt, (non couverte par le swap S).

(2) les contraintes d'efficacité :

$$\frac{a^i}{ub^i} \leq \sum_{j=1}^k C_j^i x_j \leq \frac{a^i}{lb^i}, \text{ pour chaque scénario : } i, (i = 1, \dots, 3)$$

Ainsi, le problème de la couverture comptable peut être formulé sous la forme suivante :

$$(SQ) \left\{ \begin{array}{l} \text{Min}_{x \in \mathbb{R}^k} F(x) = \left\| \eta(S) + \sum_{j=1}^k x_j \eta(L_j) \right\|_2^2 \\ \text{sous les contraintes :} \\ 0 \leq x_j \leq d_j, (j = 1, \dots, k) \\ \frac{a^i}{ub^i} \leq \sum_{j=1}^k C_j^i x_j \leq \frac{a^i}{lb^i}, (i = 1, \dots, 3) \end{array} \right.$$

Finalement, on aboutit au modèle mathématique de la couverture comptable, qui est un problème d'optimisation quadratique à contraintes linéaires d'inégalités. Afin de le résoudre, nous proposons d'appliquer l'algorithme d'Uzawa à pas constant (2.9), (voir chapitre 2).

3.3 Extension du problème

3.3.1 Motivation

En pratique, on est souvent amené à couvrir des portefeuilles de grande taille (le nombre de prêts couverts, noté n , pouvant être très grand, de l'ordre de 100.000). En plus, un swap donné ne peut couvrir qu'un nombre limité de prêts (un sous-portfeuille). Donc, il reste plusieurs prêts, dans le portefeuille, qui ne seront pas couverts. Afin de pallier cette lacune, on peut ajouter d'autres swaps au portefeuille en vue de couvrir le plus grand nombre possible de prêts. Le problème peut être posé à nouveau comme suit :

Considérons : un ensemble de prêts $L = \{L_1, L_2, \dots, L_n\}$, et un ensemble de swaps $S = \{S_1, S_2, \dots, S_m\}$. Nous cherchons à couvrir le portefeuille de prêts L , en utilisant les instruments swaps S .

3.3.2 Procédure de sélection

Nous supposons, qu'un swap donné puisse couvrir au maximum k prêts ($k \ll n$), donc, il y a plusieurs manières de construire un sous-portefeuille (instrument couvert) de taille k . Pour le premier swap, il existe C_n^k sous-portefeuilles de taille k , formés d'un portefeuille qui est composé de n prêts. Nous cherchons à sélectionner un sous-portefeuille dont les variations de prix compensent au mieux celles du swap.

La figure suivante, illustre la sélection optimale d'un sous-portefeuille, pour un seul swap donné.

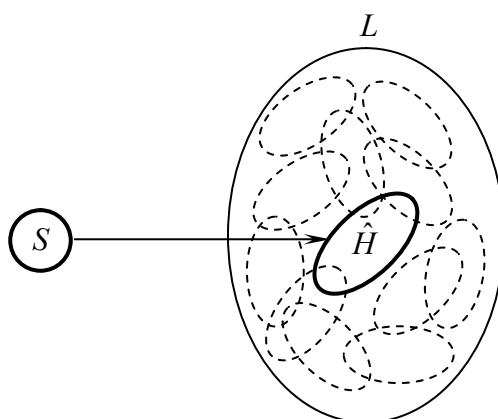


Figure 3.4 : Sélection optimale du sous-portefeuille \hat{H} , pour un swap S .

3.3.3 Généralisation de la procédure de sélection

La procédure précédente peut être généralisée pour les autres swaps disponibles. Le problème peut être décrit comme suit :

Nous cherchons, pour chaque swap S_i , ($i=1, \dots, m$), un sous-portefeuille \hat{H}_i , ($i=1, \dots, m$), composé de k prêts, dont l'amplitude des variations de prix compense au mieux celle des variations de prix du swap. Ceci peut être illustré sur la figure (3.5).

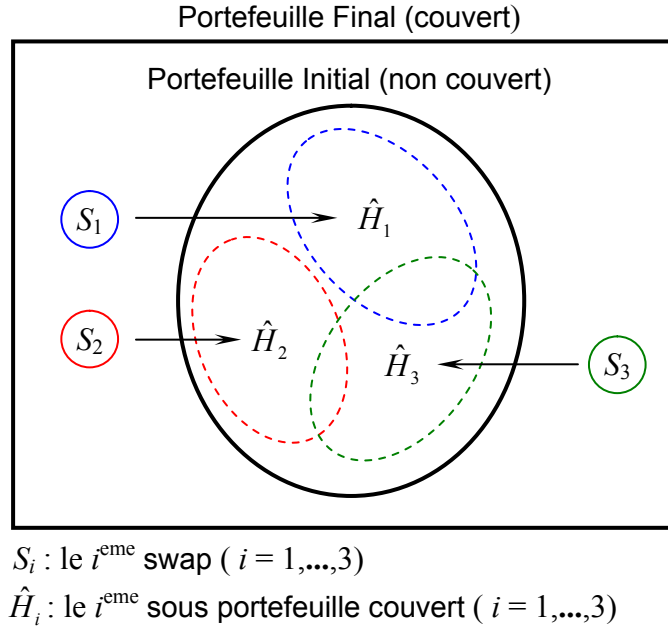


Figure 3.5 : Exemple de couverture d'un portefeuille de prêts avec trois swaps.

3.4 Formulation du problème

De ce qui précède, nous avons remarqué que pour un swap donné S , le problème peut être formulé par le modèle mathématique suivant :

$$[\text{Min}]_{e \in E[L_d(S)]} [\text{Min}]_{x \in X(S,e)} F(S, e, x) = \left\| \eta(S) + \sum_{j \in e} x_j \eta(L_j) \right\|_2^2$$

avec :

$$X(S, e) = \begin{cases} 0 \leq x_j \leq d_j(S) & , (j \in e) \\ \frac{a^i(S)}{ub^i} \leq \sum_{j \in e} C_j^i x_j \leq \frac{a^i(S)}{lb^i} & , (i = 1, \dots, 3) \\ x \in \mathfrak{R}^{|e|} \end{cases}$$

où,

- $L_d(S)$: désigne l'ensemble de prêts disponibles pour le swap S , ($L_d(S) \subseteq L$).
- e : désigne une k -combinaison : ($e \subset \{1, 2, \dots, n\}$, tel que : $|e| = k$, avec : $k < n$).
- $E[L_d(S)] = \{\text{toutes les } k\text{-combinaisons de prêts engendrées par l'ensemble } L_d(S)\}$, cet espace est fini, sa taille peut être très importante, mais elle est bornée par : C_n^k , (i.e. $|E[L_d(S)]| \leq C_n^k$).

- $d_j(S)$: la portion disponible de chaque prêt L_j , ($j \in e$), non couverte par le swap S .
- x_j : la portion couverte de chaque prêt L_j , ($j \in e$).
- $a^i(S)$: la variation absolue de prix du swap S engendrée par le scénario i , ($i = 1, \dots, 3$).
- C_j^i : la variation absolue de prix du prêt L_j , ($j \in e$), engendrée par le scénario i , ($i = 1, \dots, 3$)
- lb^i et ub^i : la borne inférieure et supérieure du ratio d'efficacité, pour chaque scénario i , ($i = 1, \dots, 3$).

Le problème ainsi formulé est un problème d'optimisation combinatoire, pour le résoudre nous appliquons les méthodes suivantes :

3.5 Méthodes de résolution

Dans ce paragraphe, nous décrivons les méthodes de résolutions utilisées. En premier temps, nous présentons ces méthodes pour un seul swap, ensuite nous allons les généraliser pour le cas de plusieurs swaps.

3.5.1 Méthode exacte

Le principe essentiel d'une méthode exacte, consiste généralement à énumérer de manière implicite ou explicite l'ensemble de solutions admissibles, afin de déterminer la solution optimale pour un exemple du problème de taille raisonnable.

Les méthodes exactes les plus intéressantes qui existent dans la littérature, sont celles qui examinent l'espace de recherche d'une manière implicite. Parmi ces méthodes, on trouve les techniques de séparation et évaluation progressive (*SEP*) ([18], [19]), qui nécessitent le calcul d'une borne inférieure de la fonction objective afin d'énumérer implicitement l'espace de recherche et de localiser la solution optimale.

Pour notre problème, il est difficile d'avoir une telle borne, de ce fait la méthode exacte qui peut être appliquée aux exemples de petites tailles, est la méthode classique¹ suivante :

¹ Parfois on dit : la méthode naïve (ou triviale), à cause de l'énumération explicite de l'espace de recherche.

Principe de la méthode :

L'idée est de dresser pour un swap donné, la liste de $N = C_n^k$ k -combinaisons de prêts, ensuite d'associer à chaque k -combinaison son coût réduit (sensibilité quadratique minimale) et enfin, de choisir une k -combinaison de coût réduit minimum dans cette liste. Ceci peut être schématisé par la figure suivante :

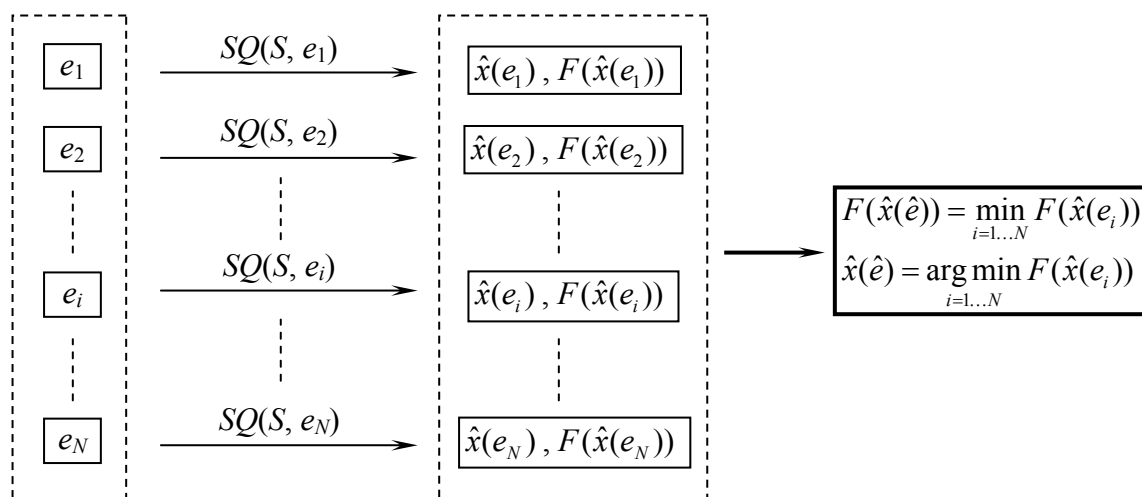


Figure 3.6 : Schéma illustrant le principe de la méthode exacte, pour un seul swap.

Les grandes lignes de la méthode (pour un seul swap), peuvent se traduire par l'algorithme suivant.

Algorithme 3.1

Début

- Générer toutes les k -combinaisons possibles, prises dans l'ensemble de prêts L , ($|L| = n$). (i.e. construire l'espace de recherche $E[L_d(S)]$, $|E[L_d(S)]| = C_n^k$).
- À chaque k -combinaison, calculer le vecteur de portions : $\hat{x}(S, e)$ en résolvant le sous-problème quadratique :

$$\text{Min}_{x \in X(S, e)} F(S, e, x)$$

- Sélectionner une k -combinaison de coût réduit minimum, $\hat{e}(S)$ en résolvant le problème :

$$\text{Min}_{e \in E[L_d(S)]} F(S, e, \hat{x}(S, e))$$

Fin

Complexité de l'algorithme (3.1)

Le principe de la méthode exacte est simple, le problème est que :

- 1) Pour n assez grand, la réponse de l'ordinateur peut prendre un temps d'exécution très important. En effet, lorsque n augmente (k est fixé), le nombre Binomial : C_n^k , peut devenir très vite un entier assez important, ce qui est un sérieux obstacle, tant au niveau du temps d'exécution que de l'espace requis pour stocker les k -combinaisons en mémoire vive de la machine.

À titre illustratif, le nombre de k -combinaisons générées au pire cas est : $C_n^{\lfloor n/2 \rfloor}$, comme l'indique l'inégalité suivante, ($\lfloor \alpha \rfloor$, désigne la partie entière de α) :

$$C_n^k \leq C_n^{\lfloor n/2 \rfloor}, \text{ pour tout : } k \in \{0, 1, \dots, n\}.$$

son développement asymptotique (voir annexe-C), est donné par l'estimation suivante :

$$C_n^{\lfloor n/2 \rfloor} \underset{n \rightarrow +\infty}{\approx} \sqrt{\frac{2}{\pi}} \times \frac{2^n}{\sqrt{n}} \quad (3.16)$$

où, n : désigne le nombre de prêts; et, π : désigne le nombre 3,14159...

En vertu de l'estimation (3.16), le nombre $C_n^{\lfloor n/2 \rfloor}$ augmente de manière exponentielle en fonction du nombre de prêts n .

- 2) L'évaluation de l'écart quadratique (3.10) est coûteuse en temps de calcul. En effet, à chaque k -combinaison, nous devons résoudre le problème quadratique SQ , ce qui nécessite l'utilisation d'un algorithme efficace (rapide). Pour cela, on propose d'appliquer l'algorithme d'Uzawa à pas constant (2.9), (voir chapitre 2), où sa complexité est évaluée par la formule suivante :

$$N_{\text{oper}}^{\text{Uzawa}} = O((T_{\text{max}} + k) \times k^2) + N_{\text{iter}} \times O((T_{\text{max}} + k) \times k) \quad (3.17)$$

où, k : la taille du sous portefeuille (élément couvert).

T_{max} : la durée de vie de la couverture (en années).

N_{iter} : le nombre d'itérations effectuées par l'algorithme d'Uzawa.

Le nombre total d'opérations élémentaires de l'algorithme (3.1) peut être estimé comme suit :

$$N_{\text{oper}} = O\left(C_n^k \times N_{\text{oper}}^{\text{Uzawa}}(k, T_{\text{max}}, N_{\text{maxiter}})\right) \quad (3.18)$$

où,

k : la taille du sous portefeuille.

C_n^k : le nombre de toutes les combinaisons à k éléments prises d'un ensemble à n éléments. (le coefficient binomial).

$N_{\text{oper}}^{\text{Uzawa}}$: la complexité temporelle de l'algorithme d'*Uzawa* à pas fixe.

N_{maxiter} : le nombre maximal d'itérations effectuées par l'algorithme d'*Uzawa*.

T_{max} : la durée de vie de la couverture (en années).

D'après (3.18), la complexité temporelle de l'algorithme (3.1) est très importante.

3.5.2 Méthode approchée

Motivation

Nous venons de voir que la complexité de l'algorithme (3.1) est très importante, elle dépend fortement du nombre de prêts à couvrir. Pour les exemples de grande taille du problème, il n'est pas possible de déterminer une solution exacte en temps raisonnable, donc on s'est tourné vers une méthode approchée.

Le but d'une méthode approchée, est de trouver une solution de bonne qualité en un temps de calcul raisonnable sans garantir l'optimalité de la solution obtenue. Ces méthodes sont fondées principalement sur diverses heuristiques¹ souvent spécifiques à un type de problème.

La méthode approchée que nous allons présenter, fait partie des méthodes constructives.

Une méthode constructive consiste à construire pas à pas une solution. Partant d'une solution partielle initialement vide, elle cherche à étendre à chaque étape la solution partielle de l'étape précédente. Pour cela, elle fait intervenir des heuristiques pour

¹ Le mot « heuristique » vient du grec *heurein* (découvrir) et qualifie tout ce qui sert à la découverte, à l'invention et à la recherche.

effectuer un choix approprié pour obtenir une nouvelle solution partielle. Ce processus se répète jusqu'à ce que l'on obtienne une solution complète.

Définition 3.1: Une heuristique gloutonne est une méthode constructive. Elle effectue, à chaque étape, le choix le plus avantageux selon un critère défini a priori, en général, basé sur le coût (la fonction objective).

L'idée de la méthode approchée

La première idée est d'effectuer, à chaque swap, une recherche incomplète dans l'espace de k -combinaisons, en limitant le nombre de combinaisons à examiner. Les k -combinaisons sont générées par tirage au sort (i.e. aléatoirement) dans l'espace de recherche. Cette façon de faire, peut accélérer considérablement le processus de résolution, mais elle présente l'inconvénient de produire des k -combinaisons de mauvaises qualités.

Afin d'améliorer la qualité des k -combinaisons, il est possible d'effectuer une recherche incomplète dans un espace réduit qui est engendré par un sous-ensemble de prêts spécifique, appelé « ensemble de prêts homogènes ». La construction de cet espace est basée sur le critère suivant :

Nous introduisons le rapport du temps moyen de sensibilité, noté TAS , (*Time Average Sensitivity*), qui est donné par la formule suivante :

$$TAS(Y) = \frac{\sum_{t=1}^T t \times |\eta_t(Y)|}{\sum_{t=1}^T |\eta_t(Y)|} \quad (3.19)$$

où, Y : désigne l'instrument financier utilisé (Y peut être un prêt ou un swap).

$\eta_t(Y)$: la sensibilité de l'instrument Y au point t de la courbe des taux.

L'utilisation de ce critère est justifiée par le fait qu'un prêt et un swap ayant le même TAS , leurs variations de prix se compensent presque totalement (i.e. presque la même amplitude mais de signe opposé).

Nous définissons aussi une distance, notée D , entre un swap S et un prêt $l, (l \in L)$ par :

$$D(S, l) = |TAS(S) - TAS(l)| \quad (3.20)$$

Par la suite, nous utilisons cette distance pour déterminer un sous-ensemble de prêts dit : « homogène », noté \tilde{L} .

Remarque 3.1 : La distance (3.20) n'est pas une métrique au sens mathématique. En effet, il se peut que deux instruments aient le même rapport TAS , (i.e. D est nulle), mais ils ne sont pas identiques.

Définition 3.2 : Les prêts homogènes sont ceux qui sont les plus proches du swap S en terme de la distance D .

Le sous-ensemble de prêts homogènes, relatif à un swap S , est défini comme suit :

$$\tilde{L}(S) = \{ l \in L_d(S) / D(S, l) < \varepsilon \} \quad (3.21)$$

où : ε est un paramètre positif choisi à l'avance par le gestionnaire.

Notons, $E[\tilde{L}(S)]$: l'espace de recherche réduit de k -combinaisons, engendré par le sous-ensemble de prêts homogènes $\tilde{L}(S)$.

Notre objectif revient à chercher la meilleure k -combinaison dans $E(\tilde{L}(S))$. Le coût d'une recherche complète dans cet espace est :

$$|E(\tilde{L}(S))| = C_K^k$$

où : $K = |\tilde{L}(S)|$, désigne le nombre de prêts homogènes, choisi a priori par le gestionnaire.

de ce fait, il n'est pas utile d'examiner complètement l'espace de recherche $E(\tilde{L}(S))$, mais on se contente d'énumérer un échantillon de k -combinaisons (de taille limitée), généré par tirage au sort dans cet espace.

La figure suivante, illustre la sélection du meilleur sous-portefeuille \hat{H} qui est couvert par le swap S .

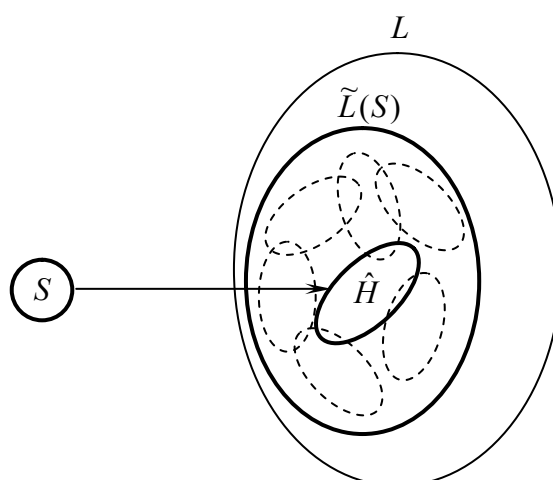


Figure 3.7 : Sélection du meilleur sous-portefeuille \hat{H} pour le swap S .

Les grandes lignes de la méthode approchée, pour un seul swap, sont traduites par l'algorithme suivant :

Algorithme 3.2

Données

- $L = \{L_1, L_2, \dots, L_n\}$: un ensemble de prêts (éléments couverts).
- S : un swap, (un seul instrument de couverture).
- $d_j \leftarrow 100\%$, $(j = 1, \dots, n)$, (au début, tous les prêts sont totalement disponibles).

Début

- (a) Déterminer, l'ensemble de prêts homogènes $\tilde{L}(S)$ de taille K , ($|\tilde{L}(S)| = K$).
- (b) Générer, un échantillon de k -combinaisons, de taille NC dans $E[\tilde{L}(S)]$, ($k < K$).
- (c) Pour chaque k -combinaison : $e \in \{e_1, e_2, \dots, e_{NC}\}$, calculer le vecteur de portions $\hat{x}(S, e_i)$, en résolvant le sous-problème quadratique :

$$\text{Min}_{x \in X(s, e_i)} F(s, e_i, x) \quad , \quad (i = 1, \dots, NC)$$
 Fin pour.
- (d) Sélectionner, une k -combinaison de coût réduit minimal : $\hat{e}(S)$, en résolvant le problème :

$$\text{Min}_{i \in \{1, \dots, NC\}} F(s, e_i, \hat{x}(s, e_i))$$

Fin.

Principe général de la méthode approchée

Dans la section précédente, nous avons considéré la méthode de résolution dans le cas d'un seul swap. Ici, nous présentons la généralisation de la méthode approchée, ce qui se fait de la façon suivante :

- 1) La méthode est constructive, on ajoute séquentiellement les instruments swaps au portefeuille de prêts initial, et à chaque swap (étape), nous cherchons à sélectionner un instrument structuré (sous-portefeuille) dont les variations de prix compensent au mieux celles du swap. La couverture de l'instrument structuré, par le swap en cours, se réalise en minimisant l'écart quadratique de la sensibilité entre ces deux instruments.
- 2) Ensuite, on doit mettre à jours la disponibilité de chaque prêt de l'instrument structuré précédemment sélectionné. Les prêts qui sont totalement couverts (i.e. les prêts ayant une disponibilité inférieure à 10%) par le swap en cours sont exclus de la prochaine étape.
- 3) On répète ce processus jusqu'à l'épuisement de tous les swaps, ou quand la sensibilité du portefeuille n'est plus significativement améliorée avec l'ajout d'un nouveau swap.

Finalement, on obtient un nouveau portefeuille dont le prix du marché est moins sensible, aux variations des taux d'intérêt, que le portefeuille initial (non couvert).

Les grandes lignes de la méthode, peuvent être traduites par l'algorithme suivant :

Algorithme 3.3

Données

- $L = \{L_1, L_2, \dots, L_n\}$: un ensemble de prêts (éléments couverts).
- $S = \{S_1, S_2, \dots, S_m\}$: un ensemble de swaps (instruments de couverture).
- $d_j \leftarrow 100\%$, $(j = 1, \dots, n)$: (au début, tous les prêts sont totalement disponibles).

Début

- Pour chaque swap $s \in S$, faire :

(a) Déterminer, l'ensemble de prêts homogènes $\tilde{L}(s)$ de taille K , ($|\tilde{L}(s)| = K$).

(b) Générer, un échantillon de k -combinaisons, de taille $NC(s)$ dans $E[\tilde{L}(s)]$, ($k < K$).

(c) Pour chaque k -combinaison : $e \in \{e_1, e_2, \dots, e_{NC(s)}\}$, calculer le vecteur de portions $\hat{x}(s, e_i)$ en résolvant le sous-problème quadratique :

$$\text{Min}_{x \in X(s, e_i)} F(s, e_i, x) \quad , \quad (i = 1, \dots, NC(s))$$

Fin pour.

(d) Sélectionner, la k -combinaison de coût réduit minimal, $\hat{e}(s)$ en résolvant le problème :

$$\text{Min}_{i \in \{1, \dots, NC(s)\}} F(s, e_i, \hat{x}(s, e_i))$$

(e) Mise à jour de la disponibilité de chaque prêt sélectionné :

$$d_j \leftarrow d_j - \hat{x}_j(s, \hat{e}(s)) \quad , \quad (j \in \hat{e}(s))$$

- Fin pour.

Fin.

La solution obtenue à l'issue de l'algorithme (3.3) ne peut être qu'une approximation de la solution optimale recherchée, et que sa qualité découle essentiellement de choix du paramètre K (nombre de prêts homogènes), plus K est grand, plus la chance de trouver une solution optimale est élevée.

Complexité de l'algorithme (3.3).

La complexité algorithmique est exprimée comme une fonction de la taille du jeu de données du problème.

Notons, M : la taille du problème, qui est la quantité de données qui doivent être traitées pour résoudre le problème. Elle est estimée comme suit :

$$M = O(T_{\max} \times (n + m)) \quad (3.22)$$

où, n : le nombre de prêts (éléments couverts).

m : le nombre de swaps (instruments de couverture).

T_{\max} : la durée de couverture (en année).

Le nombre d'opérations élémentaires effectuées par l'algorithme, est estimé par la formule suivante :

$$N_{\text{oper}} = O\left(m \times NC_{\text{max}} \times N_{\text{oper}}^{\text{Uzawa}}(k, T_{\text{max}}, N_{\text{maxiter}})\right) \quad (3.23)$$

avec :

$$NC_{\text{max}} = \max_{i \in \{1, \dots, m\}} [NC(S_i)]$$

où,

m : le nombre de swaps.

NC_{max} : le nombre maximal de k -combinaisons examinées.

$N_{\text{oper}}^{\text{Uzawa}}$: la complexité de l'algorithme d'*Uzawa* à pas fixe.

T_{max} : la durée de couverture(en année).

k : la taille du sous portefeuille.

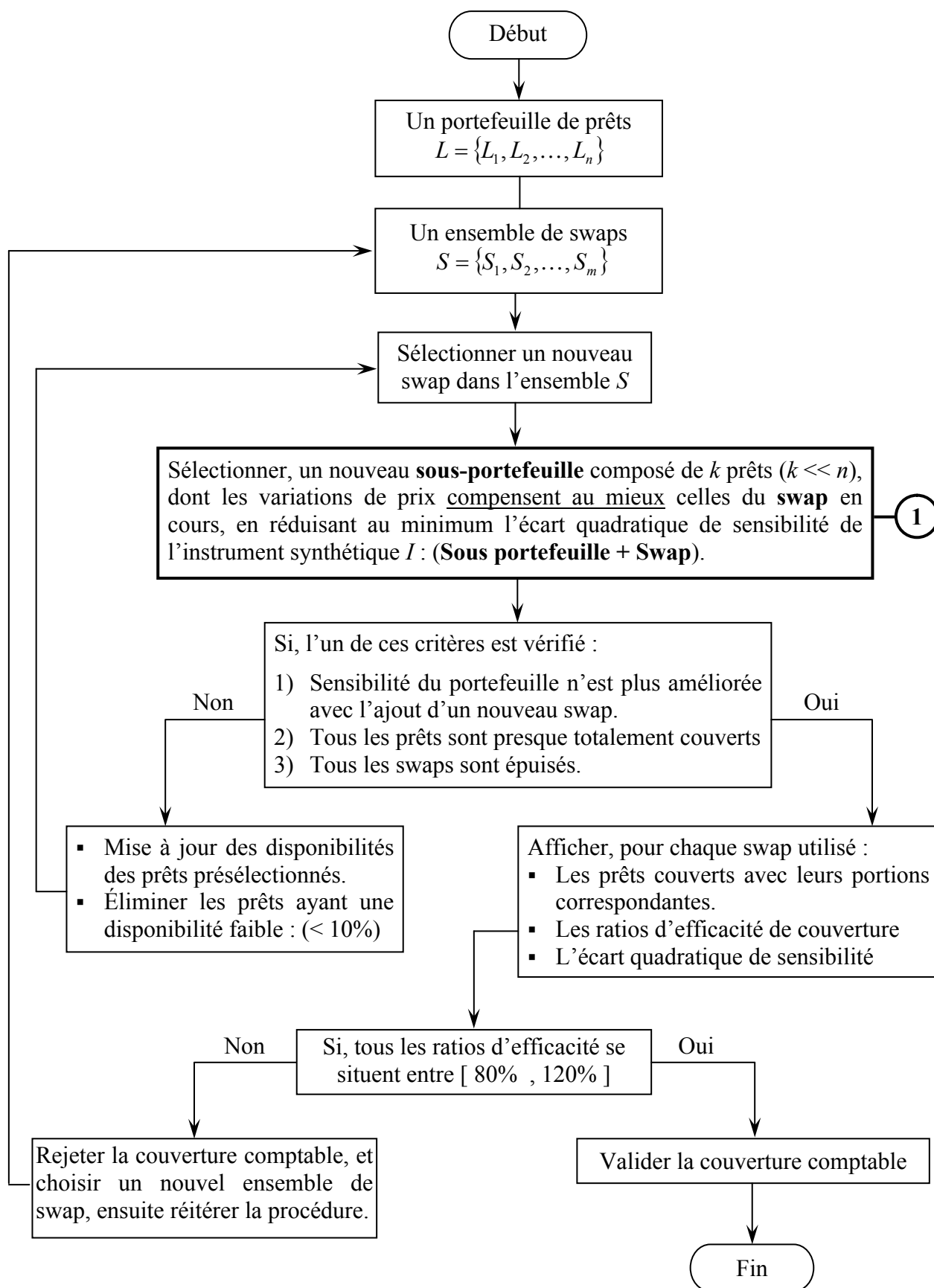
N_{maxiter} : le nombre maximal d'itérations effectuées par l'algorithme d'*Uzawa*.

D'après (3.23), la complexité de l'algorithme est polynomiale, ce qui justifie son efficacité (rapidité) pour la résolution des exemples de plus grande taille du problème.

Organigramme de la couverture comptable

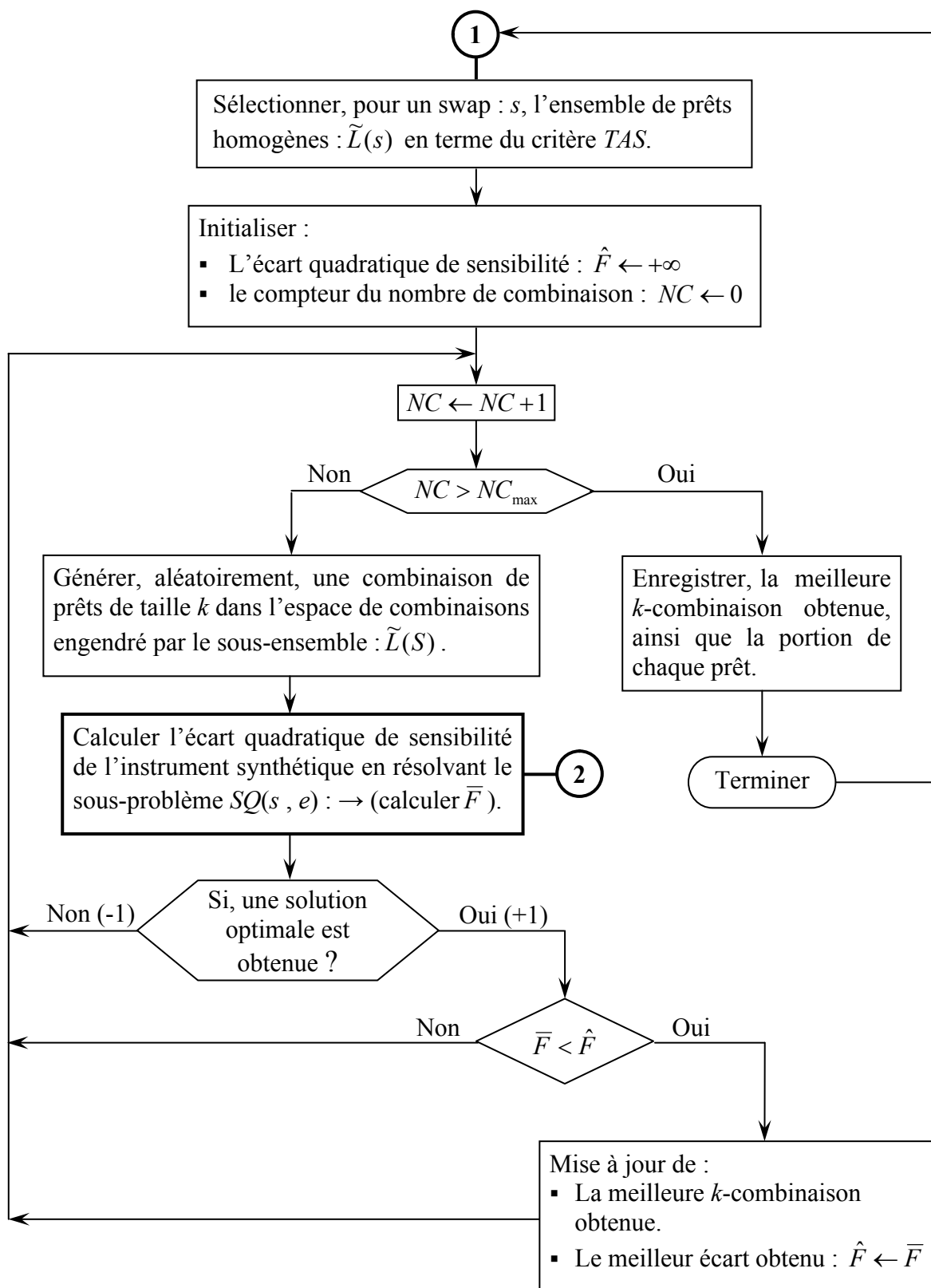
Afin de montrer les différentes étapes de l'opération de la couverture comptable, nous récapitulons le principe général de la méthode approchée à travers l'organigramme suivant :

Organigramme de la couverture comptable



Suite ...

Suite de l'organigramme



Suite ...

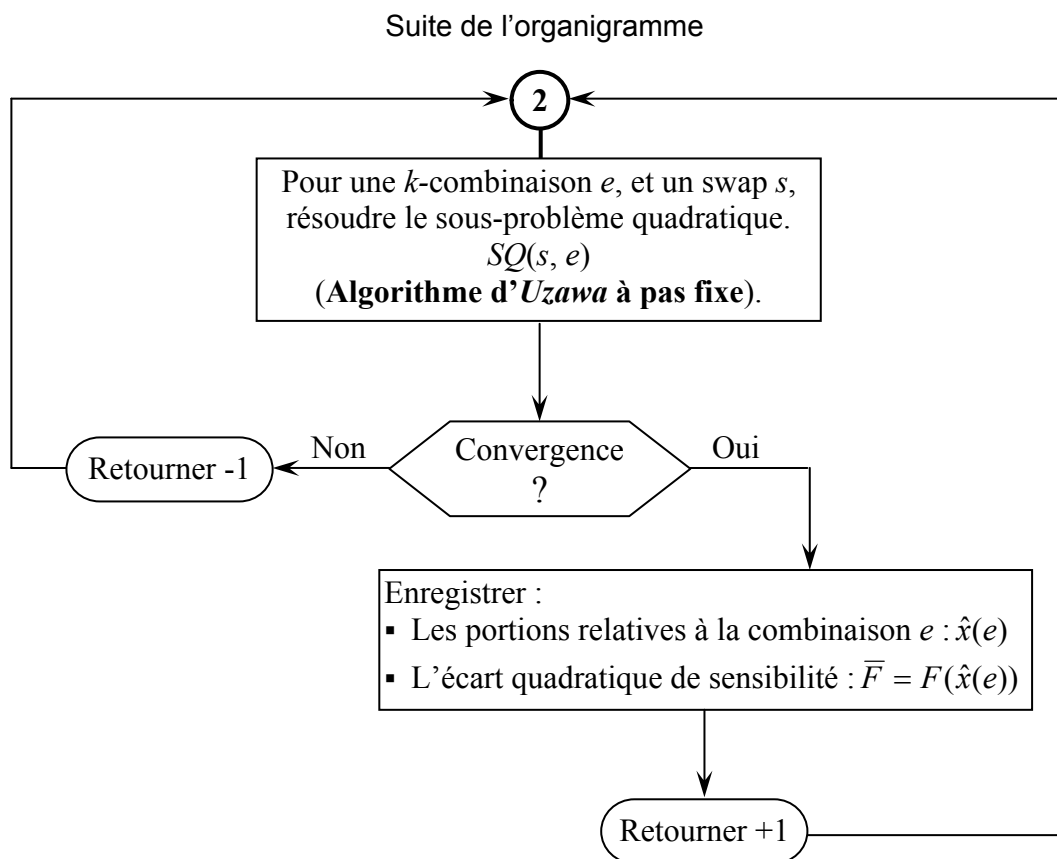


Figure 3.8 : Organigramme de la couverture comptable.

3.6 Conclusion

Dans ce chapitre, nous avons présenté la formulation mathématique du problème de la couverture comptable sous la norme de l'IAS39, ainsi que les méthodes de résolution.

La modélisation mathématique du problème, nous a conduit à une méthode de résolution complexe. Il apparaît clairement que la méthode exacte est limitée par ces exigences en temps de calcul et en espace mémoire, ce qui justifie le recours à une méthode approchée.

Nous avons appliqué une méthode approchée, basée sur une heuristique gloutonne, et nous avons constaté que la difficulté de résolution du problème provient non seulement de sa grande taille, mais aussi de la résolution du sous-problème quadratique (SQ). Pour le résoudre, nous avons proposé l'application de l'algorithme d'*Uzawa* à pas constant (2.9).

Afin de résoudre le problème de la couverture comptable, d'une manière plus efficace (rapide), nous avons implémenté des techniques d'amélioration qui permettent d'accélérer le processus de résolution. Ces techniques feront l'objet du chapitre 4.

CHAPITRE 4

IMPLÉMENTATION ET TESTS

4.1 Introduction

Dans ce chapitre, nous abordons l'implémentation de la méthodologie décrite au chapitre précédent et aux tests expérimentaux. Nous décrivons, dans un premier temps, des techniques d'amélioration permettant d'accélérer la résolution du problème, et le choix de divers paramètres qui contrôlent la qualité et l'efficacité de la couverture comptable. Ensuite, en utilisant ces techniques, nous donnons un programme informatique pour la résolution effective du problème.

De nombreux tests numériques que nous avons effectués, ont permis d'avérer les avantages de l'utilisation de l'algorithme d'*Uzawa* à pas fixe (2.9).

Nous présentons aussi les meilleurs résultats obtenus sur un exemple type du problème.

4.2 Techniques d'amélioration

Il est possible d'améliorer la rapidité de la résolution du problème de la couverture comptable en utilisant des techniques, simples à mettre en œuvre, décrites ci-dessous.

4.2.1 Réduction du nombre de contraintes du sous-problème (SQ)

Les deux premières contraintes, du sous problème quadratique (SQ) (voir chapitre 3), peuvent être pénalisées au niveau de la fonction objective : en introduisant la quantité de la sensibilité totale qui est engendrée par les deux déplacements parallèles de la courbe des taux d'intérêt à $\pm 1\%$ (vers le haut ou vers le bas).

La sensibilité totale est calculée par la formule suivante :

$$\sum_{t=1}^{T_{\max}} \eta_t, \quad (T_{\max} : \text{la durée de couverture en année}).$$

le nouveau vecteur de sensibilité, noté η^P , est de dimension $(T_{\max} + 1)$. Il s'écrit comme suit :

$$\eta^P = (\eta, \eta_{T_{\max}+1})^t = \left(\underbrace{\eta_1, \eta_2, \dots, \eta_T}_{(1, \dots, T_{\max})}, \underbrace{P \times \sum_{t=1}^{T_{\max}} \eta_t}_{(T_{\max}+1)} \right)^t$$

où, $\eta_{T_{\max}+1}^P$: représente la sensibilité totale multipliée par un facteur P .

$$\eta_{T_{\max}+1}^P = P \times \sum_{t=1}^{T_{\max}} \eta_t$$

avec, P est un paramètre choisi par l'utilisateur, comme on va le voir par la suite.

Le sous-problème de minimisation de la sensibilité quadratique (SQ) devient :

$$SQP \left\{ \begin{array}{l} [\text{Min}]_{x \in \mathfrak{R}^k} F_P(x) = \left\| \eta^P(S) + \sum_{j=1}^k x_j \eta^P(L_j) \right\|_2^2 \\ \text{sous les contraintes :} \\ 0 \leq x_j \leq d_j \quad (j = 1, \dots, k) \\ \frac{a^3}{ub^3} \leq \sum_{j=1}^k C_j^3 x_j \leq \frac{a^3}{lb^3} \quad (\text{une seule contrainte}) \end{array} \right.$$

où : l'exposant (3) désigne le scénario d'inversion (ou de rotation).

De cette façon de faire, nous avons ramené la résolution du problème (SQ) « à trois contraintes » à celle du problème pénalisé (SQP) « à une seule contrainte ».

Choix du facteur P :

Le choix du facteur P dépend du scénario choisi, en effet :

- les grandes valeurs du facteur P provoquent une meilleure couverture contre les changements parallèles de la courbe des taux (scénario 1 et 2).
- les petites valeurs de P provoquent une meilleure couverture contre l'inversion (ou la rotation) de la courbe des taux (scénario 3).

Alors, le gestionnaire peut agir sur ce facteur, pour établir le portefeuille de robustesse voulu dans le 1^{er} et le 2^{ème} scénario (déplacements parallèles) ou dans le 3^{ème} scénario d'inversion (ou de rotation).

4.2.2 Construction des pools de prêts

Le nombre important de prêts influe sur le déroulement de la résolution du problème. Afin de réduire ce nombre, on peut traiter les prêts sous forme de pools (i.e. groupes de prêts), pour cela, nous utilisons la stratégie *Bullet* [6] qui consiste à regrouper les prêts selon le critère de Duration, ainsi les prêts ayant une Duration équivalente (proche), font parti du même pool.

La *Duration*, notée DUR , d'un instrument financier Y est donnée par la formule suivante:

$$DUR(Y) = \frac{\sum_{t=1}^T t \times F_t(Y) B_t}{\sum_{t=1}^T F_t(Y) B_t} \quad (4.1)$$

où : $F_t(Y)$: le flux monétaire de l'instrument Y , versé à la date futur t .

T : la maturité (ou échéance) de l'instrument Y .

B_t : le facteur d'actualisation pour la maturité t .

Elle peut être interprétée comme le centre de gravité des flux monétaires actualisés.

La construction des pools de prêts, en terme de la Duration, peut être réalisée par l'algorithme suivant :

Algorithme 4.1 (construction des pools)

Début

- 1) Subdiviser l'intervalle du temps $[0, T_{\max}]$ en np sous intervalles, $]t_{i-1}, t_i]$, ($i=1, \dots, np$)
- 2) Ordonner les prêts selon l'ordre croissant en terme de la Duration (DUR).
- 3) Déterminer, le $i^{\text{ème}}$ pool, de la façon suivante :

$$L_i = \{ l \in L : t_{i-1} < DUR(l) \leq t_i \}, (i=1, \dots, np)$$

où, $t_0 = 0$ et $t_{np} = T_{\max}$; (T_{\max} : la durée de couverture).

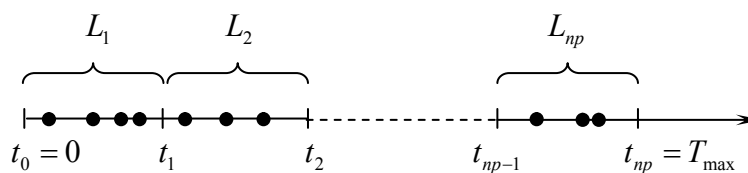
L : l'ensemble de prêts élémentaires $L = \{l_1, l_2, \dots, l_n\}$.

np : le nombre de pools.

L_i : le $i^{\text{ème}}$ pool construit dans l'ensemble L .

Fin.

La figure suivante, illustre la construction des pools de prêts.



(le symbole • indique la position d'un prêt élémentaire sur la flèche du temps).

Figure 4.1 : Construction des pools de prêts en terme de la *Duration*.

4.2.3 Classement de swaps

D'après de nombreuses expériences effectuées par notre programme, nous avons constaté que le classement dont les swaps sont ajoutés au portefeuille, influe à la fois : sur la qualité de la couverture et aussi bien sur la rapidité du programme.

Dans le but d'accélérer l'exécution du programme, nous avons classé les swaps selon l'ordre décroissant de leurs prix, afin d'éliminer au maximum les prêts qui sont susceptible d'être totalement couvert par le swap en cours. Cette façon de faire, peut sensiblement réduire le nombre de prêts à couvrir pour les prochains swaps. Malgré l'inconvénient que ce classement présente, en produisant une couverture de moins bonne qualité, il reste jusqu'ici le plus performant (au sens du temps d'exécution).

4.3 Méthodologie de choix de quelques paramètres

Le choix des paramètres effectifs : (K, k, NC) qui contrôlent la couverture est une tâche délicate. Dans cette section, nous proposons une méthodologie de choix pour l'adaptation de ces paramètres. Ce choix résulte d'un compromis entre la qualité de la couverture et l'efficacité (la rapidité) de résolution.

- a) En résolvant le problème, la qualité de la k -combinaison obtenue dépend essentiellement du paramètre K (nombre de prêts homogènes). En effet, plus K est grand plus la qualité de la solution est bonne, mais plus le temps d'exécution est important. La stratégie que nous avons adoptée est la suivante : au début de la couverture, le paramètre K est choisi suffisamment grand. De plus, ce paramètre dépend du nombre de prêts disponibles. Dans notre implémentation, il est choisi variable par la formule suivante :

$$K = \min(n_d(S), K_{\max})$$

où, $n_d(S)$: le nombre de prêts disponibles pour le swap S .

K_{\max} : le nombre maximal de prêts homogènes (fixé à l'avance avec : $K_{\max} \leq n$).

b) Nous avons également choisi le paramètre k (la taille de l'élément couvert) par la formule suivante :

$$k = \min(K, k_{\max})$$

où, K : le nombre effectif de prêts homogènes, disponibles pour le swap en cours.

k_{\max} : la taille maximale de la k -combinaison (fixée à l'avance avec : $k_{\max} \leq T_{\max}$).

c) Pour chaque swap, le nombre effectif de k -combinaisons générées, est choisi par la formule suivante :

$$NC = \min(C_K^k, NC_{\max})$$

où, C_K^k : le nombre Binomial.

NC_{\max} : le nombre maximal de k -combinaisons (fixé à l'avance).

4.4 Programme de la couverture comptable

Nous donnons ici, le listing de notre programme en utilisant les techniques évoquées dans les sections précédentes. Ce listing est écrit dans un pseudo-langage de programmation qui peut être traduit dans n'importe quel langage évolué. Un listing plus détaillé en langage C++ est présenté en annexe-D.

Données :

- L'ensemble de prêts : $L = \{L_1, L_2, \dots, L_n\}$.
- L'ensemble de swaps : $S = \{S_1, S_2, \dots, S_m\}$.
- La courbe des taux (zéro-coupon) : $z_T, (T = 1, \dots, T_{\max})$.
- k : la taille de la combinaison de prêts couverts.
- K : le nombre de prêts homogènes.
- NC_{\max} : le nombre maximal de k -combinaisons.
- d_{\min} : la disponibilité minimale d'un prêt.
- P : le facteur de pénalité de deux scénarios de déplacement parallèles.

Initialisation :

- Les disponibilités des prêts, $d_j = 100\%$, ($j = 1, \dots, n$), (tous les prêts sont disponibles).
- Marquage de prêts disponibles : $f_i = vrai$, ($j = 1, \dots, n$).

Corps du programme :

- Calculer : les vecteurs de sensibilité : $\eta(L_j)$, ($j = 1, \dots, n$) et $\eta(S_i)$, ($i = 1, \dots, m$).
- Calculer : les vecteurs TAS : $TAS(L_j)$, ($j = 1, \dots, n$) et $TAS(S_i)$, ($i = 1, \dots, m$).
- Ordonner : les swaps selon l'ordre décroissant de leurs prix du marché.
- Calculer : $C_j = |V^3(L_j) - V(L_j)|$, ($j = 1, \dots, n$), l'exposant ³ désigne le 3^{eme} scénario.

Pour tout, i allant de 1 jusqu'à m (i.e. pour chaque swap $s \in \{S_1, S_2, \dots, S_m\}$)

- Initialiser l'écart quadratique de la sensibilité : $\hat{F} = +\infty$.
- Calculer les distances : $D_j = |TAS(L_j) - TAS(S_i)|$, ($j = 1, \dots, n$).
- Sélectionner le sous-ensemble $\tilde{L}(S_i)$ de taille K , de la façon suivante :
 - Ordonner les prêts disponibles selon l'ordre croissant de la distance D_j , ($j = 1, \dots, n$)
 - Prendre les K premiers prêts, (i.e. les prêts homogènes de taille K).

Pour tout, $c = 1$ jusqu'à NC , (NC : nombre de k -combinaisons à examiner)

Générer, aléatoirement, une k -combinaison : e dans le sous-ensemble $E[\tilde{L}(S_i)]$.

Résoudre le sous problème quadratique (SQP).

La solution est : $[\bar{F}, \bar{e}, \bar{x}]$.

où, \bar{F} : la valeur de l'écart quadratique de la sensibilité, \bar{e} : la k -combinaison, et \bar{x}_j , ($j \in \bar{e}$) : les portions des prêts obtenues.

Si ($\bar{F} < \hat{F}$) alors, changer la solution : $[\hat{F}, \hat{e}, \hat{x}] \leftarrow [\bar{F}, \bar{e}, \bar{x}]$.

Fin pour.

La meilleure solution obtenue est : $[\hat{F}, \hat{e}, \hat{x}]$

Pour tout $j \in \hat{e}$

Mise à jour du vecteur de disponibilité : $d_j \leftarrow d_j - \hat{x}_j$

Si ($d_j < d_{\min}$) alors, poser : $f_j = faux$ (i.e. le prêt L_j est indisponible).

Fin pour.

Fin pour.

Fin du programme.

4.5 Mise en œuvre informatique

Pour la résolution du problème en temps réel, nous testons notre programme sur l'exemple suivant :

4.5.1 Exemple test

Nous cherchons à couvrir un portefeuille de prêts bancaires à taux fixe, de montant nominal : 490 M€ (millions d'euro), d'échéance : 10 ans, et qui est composé de 11 pools de prêts. Pour ce faire, nous considérons un portefeuille de montant nominal : 395 M€ composé de 3 pools de swaps payeurs. Les caractéristiques de ces instruments sont reportées dans les tableaux ci-dessous.

Éléments couverts				
Prêt	Montant (M€)	Taux fixe (%)	Amortissement	Duration (ans)
L_1	60	6,51	Bullet	7,09
L_2	70	6,47	Bullet	6,57
L_3	40	6,43	Bullet	6,04
L_4	35	6,40	Bullet	5,51
L_5	45	6,30	Linéaire	4,98
L_6	30	6,20	Bullet	3,87
L_7	40	6,25	Bullet	4,43
L_8	40	6,30	Bullet	3,34
L_9	30	6,55	Bullet	7,60
L_{10}	40	6,15	Bullet	2,75
L_{11}	60	6,35	Annuité	4,95
Total	490			

Tableau 4.1 : Caractéristiques des prêts

Instruments de couverture				
Swap	Montant (M€)	Taux fixe (%)	Amortissement	Duration (ans)
S_1	160	6,47	Bullet	6,57
S_2	135	6,36	Bullet	5,51
S_3	100	6,24	Bullet	4,43
Total	395			

Tableau 4.2 : Caractéristiques des swaps

4.5.2 Initialisation des paramètres du programme

Dans nos expérimentations, nous initialisons les paramètres de notre programme par les valeurs suivantes :

- a) les valeurs des paramètres qui contrôlent la couverture comptable, sont reportées dans le tableau suivant :

Paramètre	Désignation	Valeur
n	Nombre de prêts	11
m	Nombre de swaps	3
T_{\max}	Durée de vie de couverture (en année)	10
P	Facteur de pénalisation des contraintes	10
lb	Borne inférieure du rapport d'efficacité	80 %
ub	Borne supérieure du rapport d'efficacité	120 %
k_{\max}	Taille maximale de k -combinaisons	6
K_{\max}	Nombre maximal de prêts homogènes	10
NC_{\max}	Nombre maximal de combinaisons	350

Tableau 4.3 : Initialisation des paramètres du programme.

- b) les valeurs des paramètres de l'algorithme d'*Uzawa* à pas fixe, sont reportées dans le tableau suivant :

Paramètre	Désignation	Valeur
MaxIter	Nombre maximal d'itérations	1000
r	Coefficient de pénalité	100
w	Paramètre multiplicatif ($\rho = w \times r$)	1,4
ε	Précision requise sur les contraintes :	10^{-6}

Tableau 4.4 : Initialisation des paramètres de l'algorithme d'*Uzawa* à pas fixe.

4.5.3 Simulation de la courbe des taux d'intérêt

Afin d'évaluer les titres financiers à revenu fixe (connu), l'estimation de la courbe des taux (zéro-coupon) est nécessaire. Plusieurs modèles ont été considérés pour estimer cette courbe. Nous adoptons le modèle dû à *Nelson* et *Siegel* ([1], [6]). Son avantage porte sur sa grande flexibilité qui lui permet de reconstruire presque toutes les formes de courbes rencontrées sur le marché monétaire.

Ce modèle est donné par la formule suivante :

$$\hat{z}_T = a + b \left[\frac{1 - e^{-\lambda T}}{\lambda T} \right] + c \left[\frac{1 - e^{-\lambda T}}{\lambda T} - e^{-\lambda T} \right], \quad (T = 0, \dots, T_{\max}) \quad (4.2)$$

où, \hat{z}_T : le taux zéro-coupon en T ans, obtenu à partir du modèle théorique (4.2).

a : facteur de niveau. b : facteur de rotation.

c : facteur de courbure. λ : paramètre d'échelle.

Méthode d'estimation

La méthode consiste à ajuster ce modèle sur la courbe des taux (zéro-coupon), observés sur le marché à ($t = 0$) en minimisant la somme des écarts quadratiques entre taux (zéro-coupon) observés et taux (zéro-coupon) théoriques.

L'estimation des paramètres (a, b, c, λ) du modèle, est réalisée par la résolution du problème d'optimisation non linéaire suivant, au sens des Moindres Carrés classique :

$$\text{Min} \sum_{T=1}^{T_{\max}} [z_T - \hat{z}_T(a, b, c, \lambda)]^2 \quad (4.3)$$

où, z_T : le taux zéro-coupon en T ans, observé sur le marché (taux expérimental).

\hat{z}_T : le taux zéro-coupon en T ans, obtenu à partir du modèle (taux théorique).

Pour la résolution de ce problème, l'algorithme de *Gauss-Newton*, pour les Moindres Carrées, peut être appliqué (voir annexe-A).

Pour simuler la courbe des taux de référence, nous avons utilisé les valeurs suivantes :

Paramètre	a	b	c	λ
Valeur	0,07	- 0,02	0,01	0,3

Tableau 4.5 : Valeurs estimées des paramètres de la courbe des taux de référence.

Le tableau suivant, donne les valeurs estimées des taux : \hat{z}_T , ($T = 1, \dots, T_{\max}$).

T (année)	1	2	3	4	5	6	7	8	9	10
\hat{z}_T en (%)	5,40	5,70	5,93	6,12	6,26	6,37	6,46	6,53	6,59	6,63

Tableau 4.6 : Valeurs estimées des taux d'intérêt de référence.

La figure suivante, présente l'évolution des taux de référence, en fonction de l'échéance.

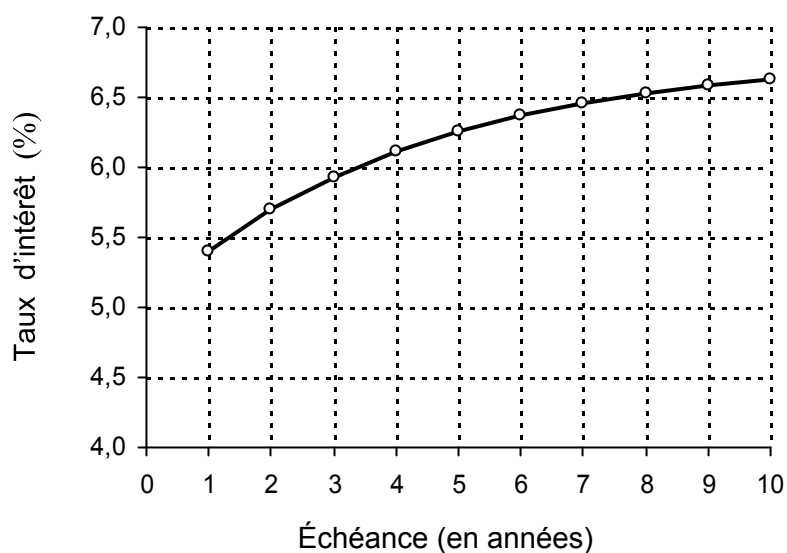


Figure 4.2 : La courbe des taux d'intérêt de référence.

Déformations de la courbe des taux

Afin de simuler les différentes déformations de la courbe de référence, nous considérons les trois scénarios suivants :

- 1) 1^{er} scénario : désigne un déplacement parallèle vers le haut de (+1%).
- 2) 2^{eme} scénario : désigne un déplacement parallèle vers le bas de (-1%).
- 3) 3^{eme} scénario : désigne un mouvement de rotation (ou d'inversion).

Les valeurs estimées des paramètres du modèle (4.2) sont données dans le tableau suivant :

Paramètres	a	b	c	λ
La courbe de référence	0,07	-0,02	0,01	0,3
1 ^{er} Scénario : ($i = 1$)	0,08	-0,02	0,01	0,3
2 ^{eme} Scénario : ($i = 2$)	0,06	-0,02	0,01	0,3
3 ^{eme} Scénario : ($i = 3$)	0,055	0,01	-0,005	0,25

Tableau 4.7 : Valeurs estimées des paramètres pour les différents scénarios de la courbe des taux.

Les différents scénarios sont représentés sur la figure suivante :

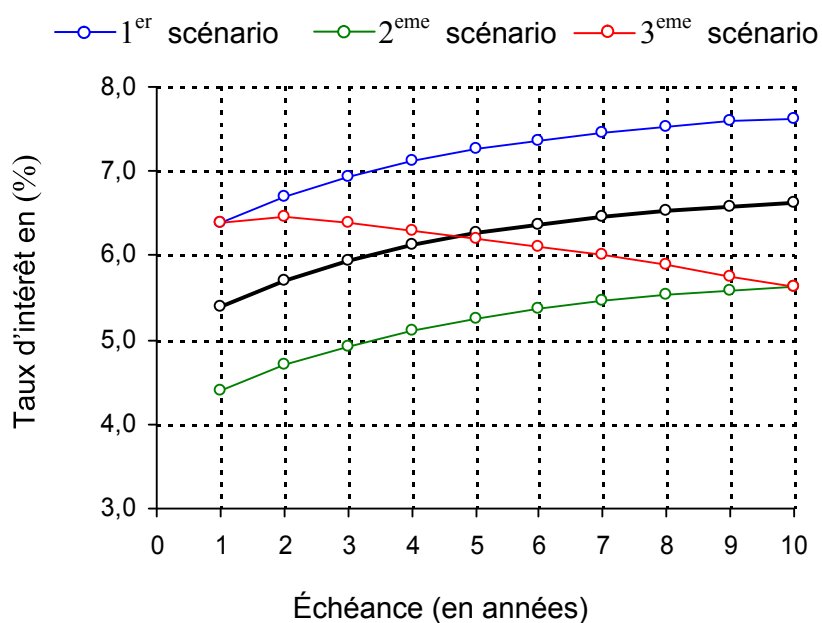


Figure 4.3 : Les différents scénarios de la courbe des taux.

4.6. Résultats expérimentaux

Dans cette section, nous donnons les meilleurs résultats expérimentaux, relatifs à la résolution de l'exemple test que nous avons présenté dans la sous-section (4.5.1).

4.6.1 Les sensibilités des meilleurs instruments synthétiques

Dans le tableau (4.9), chaque swap couvre la meilleure k -combinaison obtenue. Les écarts de la sensibilité des instruments synthétiques sont calculés. Le paramètre P est initialisé à ($P = 10$) et le paramètre k_{\max} est fixé à ($k_{\max} = 6$). Pour chaque scénario, les ratios d'efficacité sont calculés, Les bornes inférieures et supérieures de ces ratios sont fixées à 80% et 120 %.

Dans le tableau (4.10), on présente les résultats complémentaires du tableau (4.9). Pour chaque swap, nous donnons les portions des prêts obtenues : \hat{x}_j , ($j = 1, \dots, 6$).

Le vecteur de sensibilité, noté : η , pour chaque instrument synthétique : \hat{I} , est calculé par la formule suivante :

$$\eta(\hat{I}) := \eta(S) + \sum_{j \in \hat{I}(S)} \hat{x}_j \eta(L_j)$$

Les résultats obtenus sont reportés dans le tableau suivant :

t	$\eta_t(\hat{I}_1)$	$\eta_t(\hat{I}_2)$	$\eta_t(\hat{I}_3)$
1	0,04	0,01	0,03
2	0,08	0,06	0,09
3	0,11	19,87	45,73
4	0,13	-36,83	45,66
5	-76,08	-37,04	-180,01
6	-78,23	-140,01	348,02
7	0,19	352,13	-135
8	359,65	-12,12	-27,22
9	-80,17	-50,05	-26,01
10	-80,80	-49,91	-25,97
$\sum(\cdot)$	44,92	46,11	45,32
$\ \cdot\ _2$	392,70	389,70	421,92

$\sum(\cdot)$: désigne la somme totale ; et $\|\cdot\|_2$: désigne la norme euclidienne.

Tableau 4.8 : Les sensibilités des instruments synthétiques en chaque période t (en années).

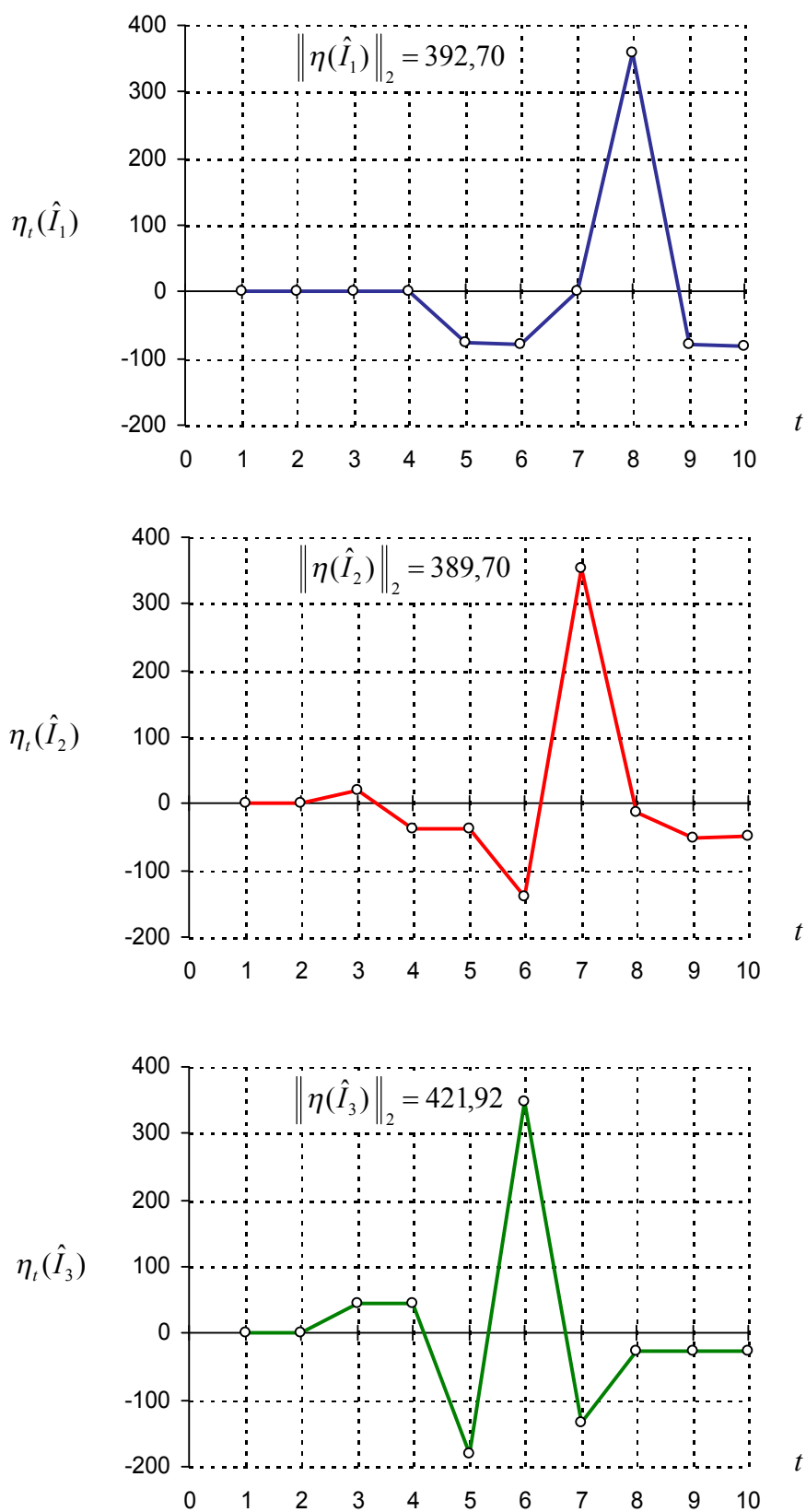


Figure 4.4 : Les sensibilités des instruments synthétiques en chaque période t (en années).

États de sortie du programme

Swap	La meilleure 6-combinaison : $\hat{e}(\cdot)$	L'écart de la sensibilité.	Rapport d'efficacité R_{eff} en (%)	
			Scénario	valeur
S_1	$\hat{e}(S_1) =$ $\{ 1, 5, 4, 10, 2, 9 \}$	$\ \eta(\hat{I}_1) \ _2 = 392,70$	Scénario	valeur
			1	102,79
			2	95,06
			3	104,84
S_2	$\hat{e}(S_2) =$ $\{ 6, 5, 1, 3, 9, 7 \}$	$\ \eta(\hat{I}_2) \ _2 = 389,70$	Scénario	valeur
			1	104,04
			2	96,92
			3	106,85
S_3	$\hat{e}(S_3) =$ $\{ 10, 8, 4, 11, 3, 1 \}$	$\ \eta(\hat{I}_3) \ _2 = 421,92$	Scénario	valeur
			1	111,48
			2	104,10
			3	114,76

Tableau 4.9 : La valeur de l'écart de la sensibilité pour chaque swap, et du ratio d'efficacité pour chaque scénario.

	S_1	S_2	S_3	Total (%)	Disponibilité (%)	TAS
L_1	30,01	25,18	0,00	55,19	44,81	8,20
L_2	100,00	-	-	100,00	/	7,50
L_3	-	52,05	0,00	52,05	47,95	6,84
L_4	73,73	-	0,00	73,73	26,27	6,24
L_5	66,20	33,80	-	100,00	/	5,72
L_6	-	95,81	-	95,81	4,19	5,05
L_7	-	100,00	-	100,00	/	5,30
L_8	-	-	21,46	21,46	78,54	5,07
L_9	54,74	43,08	-	97,82	2,18	8,94
L_{10}	0,00	-	52,85	52,85	47,15	5,45
L_{11}	-	-	100,00	100,00	/	6,58
TAS	7,50	6,24	5,30			

(le rapport TAS est exprimé en années).

Tableau 4.10 : Les portions des prêts couverts en (%) pour chaque swap.

4.6.2 Effet du paramètre P sur la qualité de la couverture

Nous étudions ici, l'effet du paramètre P sur l'écart de la sensibilité (la fonction objective).

Pour différentes valeurs fixées du paramètre P , et pour chaque swap S , on calcule la valeur suivante :

$$\hat{F}_P^{1/2}(S) = \left\| \eta^P(S) + \sum_{j \in \hat{e}(S)} \hat{x}_j \eta^P(L_j) \right\|_2$$

Les résultats obtenus sont reportés dans le tableau suivant :

P	0	5	10	15	20	30
$\hat{F}_P^{1/2}(S_1)$	243,11	506,82	596,64	662,11	698,32	701,03
$\hat{F}_P^{1/2}(S_2)$	315,27	476,01	603,72	648,03	687,23	685,13
$\hat{F}_P^{1/2}(S_3)$	367,19	523,97	619,20	685,62	705,15	666,98

Tableau 4.11 : Valeurs des meilleurs écarts de la sensibilité, pour chaque swap en fonction du paramètre P . (l'exposant $\frac{1}{2}$ désigne la racine carrée.)

Pour chaque swap, la valeur de l'écart de la sensibilité en fonction du paramètre P , est représentée sur la figure suivante :

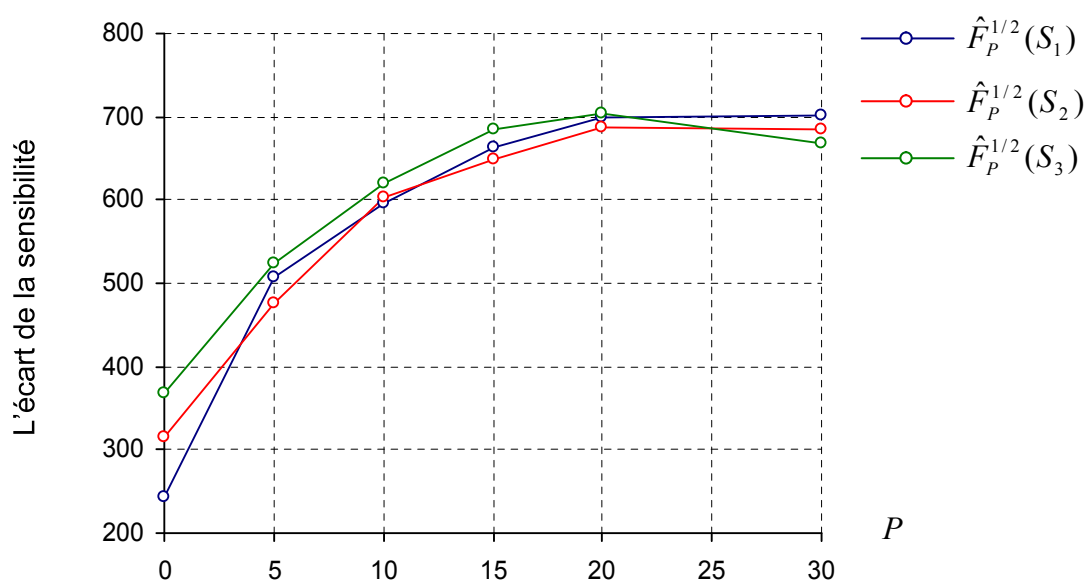


Figure 4.5 : Évolution de l'écart de la sensibilité, pour chaque swap, en fonction du paramètre P .

Adaptation du paramètre P

Dans le tableau (4.12), on calcule les ratios d'efficacité pour le premier swap, et cela pour chaque scénario. L'intervalle d'efficacité est fixé à [80 % , 120%].

P	5	10	20
1 ^{er} Scénario : ($i = 1$)	<u>129,43</u>	102,79	99,97
2 ^{eme} Scénario : ($i = 2$)	<u>74,26</u>	95,06	101,22
3 ^{eme} Scénario : ($i = 3$)	100,86	104,84	<u>130,13</u>

Tableau 4.12 : La valeur du ratio d'efficacité R_{eff}^i , pour chaque scénario i , ($i = 1, \dots, 3$) en fonction du paramètre P . (Les valeurs soulignées ne sont pas comprises dans l'intervalle [80% , 120%]).

D'après le tableau ci-dessus, la valeur ($P = 10$) semble la plus adaptée.

4.6.3 Effet des paramètres k_{max} et NC_{max} sur le temps d'exécution

Nous étudions ici, l'effet des paramètres k_{max} et NC_{max} sur le temps d'exécution (CPU : Central Processing Unit) effectué par notre programme.

Nous testons différentes valeurs du paramètre k_{max} sur notre programme, et on mesure le temps d'exécution correspondant. Ici, NC_{max} est fixé ($NC_{max} = 350$). Les résultats obtenus sont reportés dans le tableau (4.13).

k_{max}	3	4	5	6	7	8
Temps-CPU en (sec)	0,38	0,49	0,56	0,58	0,52	0,47

Tableau 4.13 : Effet de k_{max} sur le temps d'exécution effectif, avec ($NC_{max} = 350$).

De même, nous étudions l'effet du paramètre NC_{max} sur le temps d'exécution. Pour différentes valeurs fixées de NC_{max} , on mesure le temps d'exécution (CPU) correspondant. Ici, k_{max} est fixé ($k_{max} = 6$). Les résultats obtenus sont reportés dans le tableau (4.14).

NC_{max}	150	250	350	450
Temps-CPU en (sec)	0,13	0,41	0,58	1,07

Tableau 4.14 : Effet de NC_{max} sur le temps d'exécution effectif, avec ($k_{max} = 6$).

Les figures suivantes, présentent l'évolution du temps d'exécution (CPU) en fonction du paramètre k_{\max} et NC_{\max} .

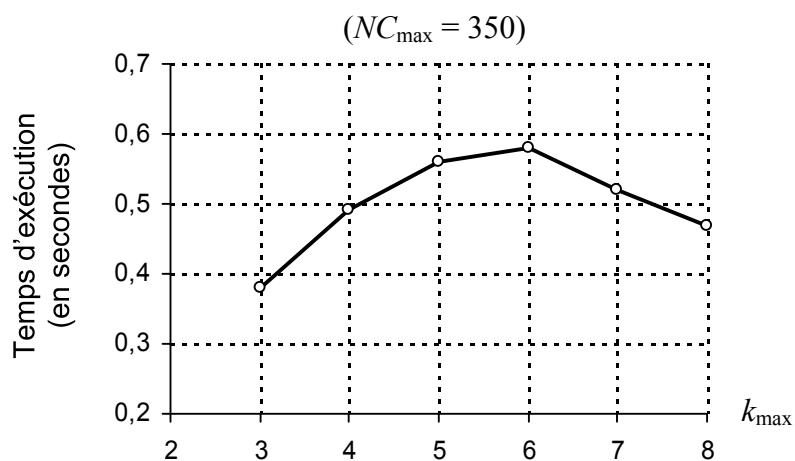


Figure 4.6 : Évolution du temps d'exécution en fonction du paramètre k_{\max} .

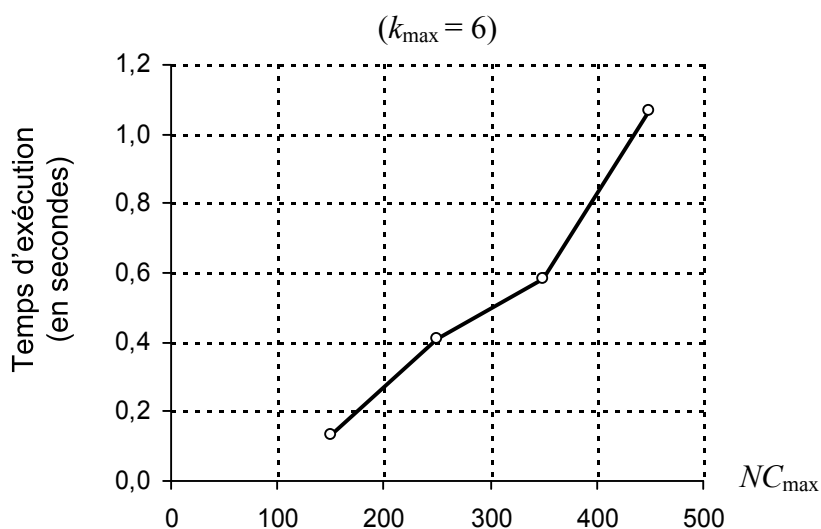


Figure 4.7 : Évolution du temps d'exécution en fonction du paramètre NC_{\max} .

Le programme informatique que nous avons développé (voir annexe-D), est codé en langage C++, et nous n'avons lui incorporé aucun sous-programme ou une routine de calcul, auxiliaire provenant des bibliothèques de calcul scientifique.

Les tests sont effectués sur une machine moyenne PC (Pentium III) dont les caractéristiques sont données dans le tableau ci-dessous :

PROCESSEUR	Intel Celeron (TM)
FRÉQUENCE DU CPU	800 MHZ
RAM	128 Mo
ENVIRONNEMENT	Windows Xp2000+
LANGAGE	Borland C++ (Ver : 5+)

Tableau 4.15 : Caractéristiques de la machine.

4.7 Conclusion

Dans la course à la performance exigée par le contexte économique actuel, la rapidité de la prise de décision peut s'avérer primordiale.

Dans ce chapitre, nous avons présenté des techniques d'amélioration qui permettent d'accélérer la résolution du problème. En utilisant ces techniques, nous avons développé et implémenté un programme informatique avec lequel nous avons testé l'effet du paramètre P sur la qualité de la couverture comptable, ainsi que l'effet des paramètres (k_{\max} et K_{\max}) sur le temps d'exécution du programme.

Dans notre étude des différentes techniques d'amélioration, nous avons constaté qu'elles sont bénéfiques sur les exemples qu'on a traités durant notre travail. L'une des difficultés d'utilisation du programme, réside dans le choix de divers paramètres qui le contrôlent : ($r, w, \text{MaxIter}, P, K_{\max}, k_{\max}, \text{NC}_{\max}, \dots$). Dans notre implémentation, tous ces paramètres ont des valeurs par défaut et peuvent être modifiés pendant la phase d'initialisation du programme.

Grâce à la performance de l'algorithme d'*Uzawa* à pas constant (2.9), et sa mise en oeuvre dans la résolution du problème, les résultats numériques obtenus affirment l'efficacité de cette approche.

CONCLUSION

Tout instrument financier comporte un risque et un rendement. Dans le cadre de la gestion Actif / Passif (ALM : *Asset Liability Management*), le risque le plus considéré par les gestionnaires financiers est le risque engendré par les fluctuations des taux d'intérêts. L'exposition à ce risque est susceptible d'entraîner des variations de prix du marché (juste valeur) de leurs portefeuilles financiers (ensembles de prêts ou emprunts).

La gestion du risque offre plusieurs approches pour réduire ce risque. La couverture comptable, est une approche qui permet aux banques de pouvoir justifier l'utilisation des produits dérivés (Swaps) dans un but de couverture.

La couverture comptable permet aux institutions financières de se prémunir contre les effets des fluctuations des taux d'intérêt sur le prix du marché de leurs portefeuilles financiers. Pour cela, ces institutions font souvent appel à des contreparties qui s'engagent à effectuer des opérations d'échange de taux d'intérêt. Les instruments de couverture les plus utilisés dans ce genre d'opérations, sont les swaps de taux d'intérêt (IRS : Interest Rate Swap). En effet, l'ajout séquentiel de ces instruments au portefeuille financier, offre aux gestionnaires la possibilité de modifier le prix du marché de leurs portefeuilles, ce qui leur permet de contrôler son niveau de risque et de l'ajuster.

Dans ce mémoire, nous nous sommes intéressés au problème de la couverture comptable, dont l'objectif est de réduire au minimum la sensibilité de prix du marché d'un portefeuille financier aux variations futures des taux d'intérêt. Ce problème peut être formulé comme un problème d'optimisation combinatoire, dont le sous-problème est un problème quadratique à contraintes linéaires d'inégalité. La structure combinatoire du problème nécessite l'utilisation d'un algorithme performant pour la résolution du sous-problème.

Le travail que nous avons réalisé au cours de ce mémoire, d'une part, nous a permis d'appliquer les méthodes duales d'optimisation en utilisant les techniques de lagrangien augmenté et de proposer un algorithme performant pour la résolution du sous-problème quadratique. D'autre part, ce travail nous a permis aussi d'élaborer un programme informatique pour la résolution effective du problème.

En premier lieu, nous avons cherché un algorithme performant pour la résolution du sous-problème. Parmi les algorithmes d'optimisation qu'on a testés, l'algorithme d'*Uzawa* à pas constant s'est montré numériquement très compétitif grâce à son efficacité (rapidité) et sa robustesse. De ce fait, nous l'avons évalué la complexité temporelle, et nous l'avons proposé pour la résolution du sous-problème quadratique.

Ensuite, nous avons remarqué que la difficulté de la résolution du problème provient essentiellement de sa grande taille. Afin de le résoudre d'une manière efficace, nous avons appliqué une méthode de résolution approchée, fondée sur une heuristique de type gloutonne.

Nous avons développé un programme informatique pour la résolution du problème en temps réel, et nous avons aussi implémenté des techniques d'amélioration permettant d'accélérer le processus de résolution.

Les résultats obtenus ont montré que :

- l'utilisation de la méthode duale, basée sur les techniques de lagrangien augmenté, s'avère efficace. En effet, l'emploi de l'algorithme d'*Uzawa* à pas constant a pour avantage de résoudre rapidement le sous-problème quadratique. L'intérêt de cette méthode est affirmé dans les exemples du problème où la grande taille rend difficile l'emploi des méthodes directes (ou primales) d'optimisation.
- grâce à l'utilisation de l'heuristique gloutonne qui est fondée sur le critère TAS (Time Average Sensitivity), les solutions produites par notre programme, sont de bonne qualité même si elles ne sont pas optimales.
- les techniques d'amélioration que nous avons implémentées, ont permis de réduire considérablement le temps de calcul effectué par le programme.

Ces résultats nous conduisent à envisager plusieurs perspectives afin d'améliorer la robustesse d'un portefeuille couvert. Ces idées s'articulent autour des points suivants :

- Généralement, afin de couvrir un portefeuille donné, on peut utiliser d'avantage de swaps. Toutefois, ces instruments ne seront pas tous disponibles à la date de couverture ($t = 0$). Donc, il est intéressant de proposer un algorithme qui

détermine le nombre minimal de swaps à utiliser pour assurer une meilleure couverture au portefeuille financier.

- Nous avons remarqué que le classement des ajouts séquentiels de swaps au portefeuille, influe sur la qualité de couverture. Il est intéressant de proposer un algorithme à implanter qui détermine un classement adéquat pour ces ajouts.
- À notre connaissance, il n'existe pas actuellement de logiciel commercial fournissant de meilleurs résultats pour la résolution du problème de la couverture comptable. Un progiciel, utilisant notre programme comme routine de calcul, fera l'objet d'un travail ultérieur.

Le travail que nous avons réalisé dans ce mémoire, nous a permis d'approfondir nos connaissances théoriques et pratiques dans le domaine d'optimisation non-linéaire et d'acquérir des connaissances dans les domaines du calcul actuariel : le pricing des instruments financiers et la gestion du risque de taux d'intérêt.

Nous espérons que ce modeste travail contribue à l'avancement de la recherche dans ce domaine, et que les gestionnaires financiers trouvent un outil d'aide à la décision, afin de protéger leurs portefeuilles contre le risque de taux d'intérêt.

APPENDICE

LISTE DES SYMBOLES ET DES ABRÉVIATIONS

Ensemble :

\mathfrak{R}	Ensemble des nombres réels.
$x \in X$	x est un élément de l'ensemble X .
$x \notin X$	x n'est pas élément de X .
$ X $	Cardinal de X : Le nombre d'éléments de l'ensemble X .
$A \subset X$	A est strictement contenu dans X .
$A \subseteq X$	A est contenu dans X .
$A \cup B$	Réunion de sous-ensembles A et B
$A \cup \{x\}, A + x$	Réunion du sous-ensemble A et du sous ensemble formé par l'élément x .
$\exists x :$	Il existe x tel que.
$\forall x \in X$	Quel que soit x élément de X .
$A \times B$	Produit cartésien de A et de B .
$[a, b]$	Intervalle fermé : $\{x / x \in \mathfrak{R}, a \leq x \leq b\}$ où a, b sont des réels ($a \leq b$)
$]a, b[$	Intervalle ouvert : $\{x / x \in \mathfrak{R}, a < x < b\}$ où a, b sont des réels ($a < b$)
$\{x^k\}_{k \geq 0}$	Suite d'éléments x^k , pour $k = 0, 1, 2, \dots$
$\{x^k\}_{k \in I}$	Suite d'éléments x^k , k parcourant l'ensemble d'indices I .

Vecteurs et matrices :

\mathfrak{R}^n	Produit cartésien de l'ensemble \mathfrak{R} .
\mathfrak{R}^{n+}	L'orthant positif de l'espace \mathfrak{R}^n .
$x = (x_1, x_2, \dots, x_n)^t$	Le vecteur de \mathfrak{R}^n (vecteur colonne).
x^t	Transposé du vecteur x , (vecteur ligne).
$x^t \cdot y$ ou (x, y)	Produit scalaire des vecteurs x et y .
$\ x\ _p$	Norme du vecteur x , où ($p = 1, 2, \infty$).
$x \geq y$	Chaque composante de x est supérieure ou égale à la composante correspondante de y .
$A = [a_{ij}]_{i=1, m}^{j=1, n}$	Matrice à m lignes et n colonne

AB	Produit de la matrice A par la matrice B .
A'	Matrice transposée de la matrice A
$\det(A)$	Déterminant de la matrice (Carrée) A .
$\text{rang}(A)$	Rang de la matrice A .
$\text{cond}_p(A)$	Conditionnement de la matrice A , $\text{cond}_p(A) = \ A\ _p \ A^{-1}\ _p$
$A \geq 0$	La matrice A est définie positive $\forall d \in \mathfrak{R}^n, d'Ad \geq 0$

Fonctions :

$\nabla f(x)$	Si, f est une fonction ($\mathfrak{R}^n \rightarrow \mathfrak{R}$) des variables (x_1, x_2, \dots, x_n) , $\nabla f(x)$ est le gradient de la fonction f au point x
$\text{grad}_x(f), \frac{\partial f}{\partial x}(x)$	Notation équivalente pour désigner le gradient de f au point x . $\nabla f(x)$
$\nabla_x f(x, y)$	Si f est une fonction ($\mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}$) des variables (x_1, x_2, \dots, x_n) , et (y_1, y_2, \dots, y_m) . $\nabla_x f(x, y)$ désigne le gradient de f en x
$\frac{\partial f}{\partial x}(x, y)$	Notation équivalente pour désigner le gradient de f au point x . $\nabla_x f(x, y)$
$\nabla^2 f(x)$	Si, f est une fonction ($\mathfrak{R}^n \rightarrow \mathfrak{R}$) des variables (x_1, x_2, \dots, x_n) , $\nabla^2 f(x)$ désigne le hessien de f en x : c'est-à-dire une matrice réelle symétrique dont le terme (i, j) est : $\frac{\partial^2 f}{\partial x_i \partial x_j}(x)$.
$\nabla_x^2 f(x, y)$	Si, f est une fonction ($\mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}$) des variables (x_1, x_2, \dots, x_n) , et (y_1, y_2, \dots, y_m) . $\nabla_x^2 f(x, y)$ désigne le hessien de f en x : c'est-à-dire la $(n \times n)$ matrice : $\left[\frac{\partial^2 f}{\partial x_i \partial x_j}(x, y) \right]$ (les dérivées partielles étant évaluées au point (x, y)).
$\frac{\partial g}{\partial x}(x)$	Si, g est une fonction ($\mathfrak{R}^n \rightarrow \mathfrak{R}^m$) des variables (x_1, x_2, \dots, x_n) , $\frac{\partial g}{\partial x}(x)$ désigne le jacobien de g au point x .
$\text{Min}_{x \in X}[f(x)]$	La valeur minimale de la fonction f dans l'ensemble X .
$\text{Max}_{x \in X}[f(x)]$	$\text{Max}_{x \in X}[f(x)] = -\text{Min}_{x \in X}[-f(x)]$
$\min(a, b)$	Le minimum de deux nombres réels a et b .
$\max(a, b)$	Le maximum de deux nombres réels a et b .
$C_n^k = \frac{n!}{k!(n-k)!}$	Le nombre de combinaisons à k éléments prises parmi n éléments. (Le nombre Binomial).

RÉFÉRENCES

- [1] D. Heath, A. Jarrow, A. Morton, Bond pricing and the term structure of interest rate: A new methodology, *Econometrica*, 1992.
- [2] E. Favor, *Mathématiques financières et calcul d'optimisation*. Dunod, Paris, 1991
- [3] G. Demange, J.C. Rochet, *Méthodes Mathématiques de la Finance*, *Econometrica*, 1992.
- [4] J. Hull, A. White, Pricing Interest Rate Derivate Securities, *The Review of Financial Studies*. 3n.4 pp (1990) 573-592.
- [5] J. P. Bouchaud, M. Potters, *Théorie des risques financiers*, Alea. Saclay, 1997.
- [6] L. Martellini, P. Priaulet et S. Priaulet, *Fixed-Income Securities : Valuation, Risk, Management and Portfolio Strategies*, Wiley, 2003.
- [7] M. Dothan, *Prices in Financial Markets*, Oxford University Press N.Y, 1990.
- [8] N. El Karoui, *Couverture des risques dans les marchés financiers*, Cours de DEA. École Polytechnique, CMAP, 91128 Palaiseau Cedex, 2003/2004.
- [9] P. Artzner, F. Delbaen, J. M. Eber, D. Heath, Coherent Measures of Risk, *Mathematical Finance* 9 (3), (1999) 203-228.
- [10] P. Poncet, R. Portait, S. Hayat, *Mathématiques financières : Évaluation des actifs et analyse du risque*. Dalloz, Paris, 1993.
- [11] S. Priaulet, L. Martellini et P. Priaulet, *Produits de taux d'intérêt : Méthodes dynamiques d'évaluation et de couverture*, *Economica*, 2000.
- [12] D. Gabay, B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximations, *comp. and Math. with Application*, 2, (1976), 1, pp. 17-40
- [13] D. P. Bertsekas, *Constrained optimization and lagrange multiplier methods*, Academic Press, 1982.
- [14] G. Cohen, *Convexité et Optimisation*. École Nationale des Ponts et Chaussées et INRIA, 2000.
- [15] G. Cohen, *Optimisation de grands systèmes*. Cours de DEA. Cermics-École nationale des Ponts et Chaussées et INRIA, 2003.
- [16] J.C. Culioli, *Introduction à l'optimisation*, École nationale supérieure des mines de paris, Centre Automatique et Système, Fontainebleau, Ellipse, 1994.
- [17] M. Fortin et R. Glowinski, *Méthodes de lagrangien augmenté, Application à la résolution de problèmes aux limites*, *Méthodes Mathématiques de l'informatique* 9, Dunod, Paris, 1982.
- [18] M. Garey, S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, 1979.

- [19] M. Minoux, Programmation mathématique, Théorie et algorithmes, Tome 1 et 2 Collection Technique et Scientifique des Télécommunications, Dunod, Paris, 1983.
- [20] P. G. Ciarlet, Introduction à l'analyse numérique matricielle et à l'optimisation, Collection Mathématiques appliquées pour la maîtrise, Masson, Paris, 1988.
- [21] P. Lascaux et R. Théodor, Analyse Numérique Matricielle appliquée à l'art de l'ingénieur, Masson, 1984.
- [22] R. Glowinsky, P. L. Tallec, An Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics, SIAM, Philadelphia, 1989.
- [23] R. T. Rockafellar, Convex Analysis, Princeton Univ. Press, 1970.
- [24] R. T. Rockafellar, A dual approach to solving nonlinear programming problems by unconstrained optimization, Math. Program. 5 (1973) 354-373.
- [25] R. T. Rockafellar, Augmented lagrange multiplier functions and duality in non convex programming, SIAM J. Control 12 (1974) 268-285.
- [26] V. V. Vazirani, Approximation Algorithms, Springer-Verlag, 2000.
- [27] W. Queck, The convergence factor of preconditioned algorithms of the Arrow-Hurwicz type, SIAM. J. Numer. Anal. 26 (1989) 1016-1030.
- [28] A. Benkaci, S. Manseur, « Problème de la Couverture Comptable en Finance: Modélisation et Techniques de Résolution », exposé au séminaire du département de mathématiques de l'université Saâd Dahlab. Blida, 14 mars 2009.
- [29] A. Rossi, G. Bichisao and F. Campolongo, Hedge Accounting Within IAS39, European Investment Bank, Economic and Financial Report, 2002/02.
- [30] C. Aubin et G. Gil , Impact des nouvelles normes internationales sur les états financiers des banques, Revue d'Économie Financière, N° 71, août, pp (2003) 99-108.
- [31] F. Biagini, P. Guasoni, M. Pratelli, Mean-Variance Hedging for Stochastic Volatility Model, Mathematical Finance 10 (2003) 109-123.
- [32] H. Bahaji, IAS39 et comptabilité de couverture en juste valeur : simulation des conséquences sur les fonds propres prudentiels des établissements de crédit, mémoire de DEA 124. (2004). « Comptabilité, Décision, Contrôle ».
- [33] I. Karatzas, I. Pikovsky. Anticipating portfolio optimisation, Adv. Appl. Prob. 28 (1996) 1095-1122.
- [34] J. Amending, D. Becherer, M. Schweizer. A Monetary Value for Initial Information in Portfolio Optimisation, Finance and Stochastic 7 (2003) 29-46.
- [35] J.P. Morgan, A consistent framework for assessing hedge effectiveness under IAS39 and FAS133, HEAT™ Technical Document, 2003.
- [36] S. Benninga, Financial Modeling. MIT Press, Cambridge, Massachusetts.1997.
- [37] S. Pliska, A stochastic calculus model of continuous trading, optimal portfolios, Math of Operations Research. 11, n.2, pp (1986) 371-382.
- [38] S. Manseur, A. Benkaci, A. Mataoui, Optimisation de la couverture de portefeuille dans la norme IAS39, Financial & Economic Engineering Symposium FEES-2K8, Agadir, Maroc, 20-21, juin, 2008.

ANNEXE-A :**ALGORITHME DE GAUSS-NEWTON
POUR LES MOINDRES CARRÉES**

Soit, le problème d'optimisation non linéaire de la forme :

$$\text{Min}_p G(p) = \frac{1}{2} \|r - F(p)\|_2^2$$

où : $r \in \mathfrak{R}^n$ et $p \in \mathfrak{R}^m$ avec ($n > m$)

$F : \mathfrak{R}^m \rightarrow \mathfrak{R}^n$ est une fonction continue est différentiable.

On cherche à résoudre le problème par la méthode de *Gauss-Newton*.

La méthode consiste à minimiser à chaque étape la fonction quadratique suivante :

$$G(p) \approx G(p^k) + \nabla G(p^k) \cdot (p - p^k) + \frac{1}{2} (p - p^k)^T \nabla^2 G(p^k) \cdot (p - p^k)$$

On résolve en p le système linéaire suivant :

$$\nabla^2 G(p^k) \cdot (p - p^k) = -\nabla G(p^k)$$

- Le gradient de la fonction G au point p^k est : $\nabla G(p^k) = F(p^k)^T \cdot [F(p^k) - r]$
- Le calcul du hessien $\nabla^2 G(p^k)$ est très coûteux, on se contente de l'approximation suivante : $\nabla^2 G(p^k) \approx \nabla F(p^k)^T \cdot \nabla F(p^k)$

Algorithme :**Début**

- Connaissant : $r \in \mathfrak{R}^n$, $F(p)$; F est une fonction ($F : \mathfrak{R}^m \rightarrow \mathfrak{R}^n$)
- Choix d'un estimé $p^0 \in \mathfrak{R}^m$ de \hat{p} , et une tolérance : $\varepsilon > 0$
- Poser : $k = 0$

Itérations :

- 1) Si, $\|\nabla F(p^k)^T \cdot [r - F(p^k)]\| < \varepsilon$, poser : $\hat{p} = p^k$, arrêter.

Sinon, calculer d^k solution du système linéaire :

$$[\nabla F(p^k)^T \cdot \nabla F(p^k)] \cdot d^k = \nabla F(p^k)^T \cdot [r - F(p^k)]$$

- 2) $p^{k+1} = p^k + d^k$
- 3) $k \leftarrow k + 1$, aller à l'étape (1)

Fin.

ANNEXE-B :**CONDITIONNEMENT D'UNE MATRICE**

Lorsque on effectue des calculs sur ordinateur, le résultat obtenu est une estimation du calcul exact du fait de l'arithmétique finie de l'ordinateur. Connaissant ce phénomène il est important de pouvoir estimer la validité de la solution obtenue calculée en précision finie. Ce problème peut être traité grâce à l'idée du conditionnement.

Définition :

Le conditionnement d'une matrice carrée A , est le nombre réel positif $\text{cond}_p(A)$ défini par :

$$\text{cond}_p(A) = \|A\|_p \|A^{-1}\|_p$$

où : $\|\cdot\|_p$ est une norme matricielle subordonnée : $(\|A\|_p = \max_{\|x\|_p=1} \|Ax\|_p, \forall A \in \mathfrak{R}^{n \times n})$.

Quelques propriétés :

- $\text{cond}_p(A) \geq 1$
- $\text{cond}_p(\alpha A) = \text{cond}_p(A), \forall \alpha \in \mathfrak{R}$
- $\text{cond}_2(A) = 1$, Pour A orthogonale.
- $\text{cond}_2(A) = \frac{\lambda_{\max}}{\lambda_{\min}}$, Pour A symétrique définie positive.
(où, $\lambda_{\max}, \lambda_{\min}$: valeurs propres extrêmes de la matrice A).

Système linéaire mal conditionné

Soit, le système linéaire : $Ax = b$; considérons le système perturbé : $A(x + \delta x) = b + \delta b$.

L'estimation de l'écart δx en fonction de la perturbation δb est donnée par l'inégalité suivante :

$$\frac{\|\delta x\|}{\|x\|} \leq \text{cond}(A) \frac{\|\delta b\|}{\|b\|} \quad (*)$$

Le conditionnement est une quantité qui exprime la sensibilité de la solution, aux perturbations des données. En vertu de (*), plus $\text{cond}(A)$ est grand, plus la solution est sensible aux perturbations des données.

ANNEXE-C :**COMBINAISONS****Définition**

Soit : E un ensemble fini à n élément : $E = \{1, 2, \dots, n\}$, on appelle combinaison de k éléments pris parmi n éléments dans E , tout sous-ensemble de E à k éléments.

- Les combinaisons sont sans répétition.
- Une combinaison ne teint pas en compte l'ordre de ces éléments.
- Le nombre de combinaisons à k éléments prises parmi n éléments est :

$$C_n^k = \frac{n!}{k!(n-k)!} = \frac{n \times (n-1) \times \dots \times (n-k+1)}{k!}$$

Quelques propriétés

- 1) Par convention : $C_n^k = 0$, pour tout $k < 0$ ou $k > n$
- 2) Pour tout, $k \in \{0, 1, \dots, n\}$, $C_n^k = C_n^{n-k}$
- 3) Pour tout, $k \in \{1, 2, \dots, n-1\}$, $C_n^k = C_{n-1}^{k-1} + C_{n-1}^k$
- 4) Le nombre de toute les combinaison est : $\sum_{k=0}^n C_n^k = 2^n$
- 5) Pour n fixé, $\forall k \in \{0, 1, \dots, n\}$, $C_n^k \leq \begin{cases} C_n^{n/2} & \text{si, } n \text{ est pair.} \\ C_n^{(n+1)/2} = C_n^{(n-1)/2} & \text{si, } n \text{ est impair.} \end{cases}$
- 6) Pour k fixé, $\forall n \geq k$, $C_n^k \leq \frac{n^k}{k!}$

Comportement asymptotique

Le nombre Binomial C_n^k (pour n suffisamment grand) est estimé par la formule suivante :

Posons : $n = p \times k$, où : k et p sont des nombres entiers, ($k \geq 1$, $p \geq 2$)

$$C_{p \times k}^k \approx \frac{1}{\sqrt{2\pi}} \times \frac{(p-1)^k}{\sqrt{k}} \times \sqrt{\frac{p}{p-1}} \times \left(\frac{p}{p-1}\right)^{p \times k}$$

Ce développement asymptotique est obtenu en utilisant la commande « asympt » fournie du logiciel de calcul formel « Maple® ».

ANNEXE-D :

PROGRAMME DE LA COUVERTURE COMPTABLE
DANS LA NORME DE L'IAS39

```

#include <stdio.h>
#include <sys\timeb.h>
#include <time.h>
#include <math.h>
#include <stdlib.h>

//Quelques macros utiles
#define INF 1e+12
#define ZERO 1e-12
#define VRAI 1
#define FAUX 0

//Initialisation des paramètres "Algorithme d'Uzawa"

const double lb = 0.80 , // La borne inférieure du rapport d'efficacité
              ub = 1.20 , // La borne inférieure du rapport d'efficacité
              Tol = 1e-6 , // Tolérance: La précision requise sur les contraintes
              r = 100 , // Coefficient de pénalité
              w = 1.4 , // Le paramètre multiplicatif
              dmin = 0.1 ; // la disponibilité minimale
const int MaxIter = 500 ; // Nombre maximal des itérations

/* Les fonctions */

double alea()
//génération d'un nombre aléatoire dans L'intervalle (0,1)
{ static prem = VRAI; time_t t;
  if (prem){srand((unsigned)time(&t)); prem = FAUX;}
  return ((double)rand()/(RAND_MAX+1.));
}

double Time()
//Calcul du temps d'exécution effectué par le programme
{ struct timeb tb; ftime(&tb); return tb.time + 0.001*tb.millitm ;
}

int *NewPerm(int n)
//Initialisation d'une nouvelle combinaison
{ int *X = new int[n];X-=1; for(int i=1;i<=n;i++) X[i]=i; return X;
}

double *NewVector(int n)
//Création d'un nouveau vecteur de type réel et de taille n
{ double *X = new double[n]; X-=1;
  for(int i=1;i<=n;i++) X[i]=0; return X ;
}

int *NewIVector(int n)
//Création d'un nouveau vecteur de type entier et de taille n
{ int *X =new int[n];X-=1;
  for(int i=1;i<=n;i++) X[i]=0; return X;
}

double **NewMatrix(int m,int n)
//Création d'une nouvelle matrice de type double et de taille (m.n)
{ double **A; int i,j;
  A=new double*[m];A-=1;
  for(i=1;i<=m;i++){A[i]=new double[n];A[i]-=1;}
  for(j=1;i<=m;j++) for(i=1;i<=n;i++)A[j][i]=0; return A;
}

```

```

double Courbe(double t)
//Simulation de la courbe des taux : Modèle "de Nelson et Siegel"
{ static double a = 0.07,b = -0.02,c = 0.01,lambda = 0.3,t1,t2;
  t1=exp(-lambda*t);t2=(1-t1)/(lambda*t); return a+b*t2+c*(t2-t1);}
double rot(double t)
//Simulation : scénario de rotation (Modèle "de Nelson et Siegel")
{ static double a = 0.065,b = -0.01,c = 0.005,lambda = 0.4,t1,t2;
  t1=exp(-lambda*t);t2=(1-t1)/(lambda*t);
  return a+b*t2+c*(t2-t1);
}

void Pb_Quad(double **A,double *B,double*C,double*d,int *J,int k,int m,
             double **a,double*b,double *Ck,double *dk)
//Construction du sous-problème quadratique SPQ
{ int i,j,q; double s;
  for(i=1;i<=k;i++)
    for(j=1;j<=k;j++)
      for(s=0.,q=1;q<=m;q++) s+=A[J[i]][q]*A[J[j]][q];a[j][i]=s;}
  for(i=1;i<=k;i++)
    {for(s=0.,q=1;q<=m;q++) s+=A[J[i]][q]*B[q]; b[i]=s; }
  for(i=1;i<=k;i++){Ck[i]=C[J[i]];dk[i]=d[J[i]];}
}

double EvalF(double **A,double *x,double *b,int *J,int m,int n)
// Evaluation de la fonction objective (sensibilité quadratique)
{ int i,j; double s,ss;
  for(ss=0,i=1;i<=m;i++)
    {for(s=0,j=1;j<=n;j++)s+=A[J[j]][i]*x[j];s-=b[i];ss+=s*s;}
  return sqrt(ss);
}

void Residu(double **L,double *x,double*R,int *J,int m,int n)
// Calcul du résidu (erreur)
{ int i,j; double s;
  for(i=1;i<=m;i++)
    {for(s=0,j=1;j<=n;j++)s+=L[J[j]][i]*x[j];
      R[i]=s;}
}

void Ratios(double *R,double **C,double *X,double a,int *I,int k)
// Calcul des ratios (efficacité de la couverture)
{ int j,i; double s;
  for(i=1;i<=3;i++)
    {for(s=0,j=1;j<=k;j++)s+=C[I[j]][i]*X[j];R[i]=s>0?a/s:0;}
}

int nLoans(int *f,int n)
// Calcul du nombre de prêts non couverts
{int i,s=0; for(i=1;i<=n;i++)s+=f[i];return s;
}

int Cholesky(double **A,int n,double eps)
// Factorisation de cholesky A = L'.D.L
//où L est une matrice triangulaire supérieure
{ int k,j,i; double d;
  for(k=1;k<n;k++)
    for(j=k+1;j<=n;j++){if(fabs(A[k][k])< eps) return -1;
      d=A[j][k]/A[k][k];for(i=j ;i<=n;A[i+1][j]-=A[i][k]*d); A[j][k]=d;}
  return 1;
}

void Cholesky_Solve(double**A,double*X,int n)
// Résolution du système linéaire [L'.D.L] X = b
{int i,j; double S;
  for(i=2;i<=n;i++) {for(S=0.0,j=1;j<i;S+=A[i][j]*X[j++]);X[i]-=S;}
  for(i=1;i<=n;i++)X[i]/=A[i][i];
  for(i=n-1;i>0;i--){for(S=0.0,j=i+1;j<=n;S+=A[j][i]*X[j++]);X[i]-=S;}
}

```

```

void NewCombinaison(int *P,int *J,int n)
// génération d'une nouvelle combinaison de taille n
{ int t,i,j;
  for(i=1;i<=n;i++)P[i]=J[i];
  for(i=n;i>1;i--){j=i*alea()+1;t= P[i];P[i]=P[j];P[j]=t;}
}

void Cash_Flows(double**F,double *Nominal,double *Taux, int *Peak,int m,int n)
// Calcul de flux monétaires actualisés des instruments financiers
{ int i,j;
  for(j=1;j<=n;++j)
  {for(i=1;i<=m;++i)F[j][i]=0.01*Nominal[j]*Taux[j];
  F[j][Peak[j]]+=Nominal[j]*pow(1+0.01*Taux[j],Peak[j]-m);}
}

double Amor(double Nominal,double Taux,int m)
// Type d'Amortissement " annuité constante"
{ return Nominal*Taux*0.01/(1-pow(1+0.01*Taux,-m));
}

void Valeur_M(double **VM,double **C,double**F,double *Z,int m,int n)
// Calcul de la variation de valeur du marché des instruments financiers
// Pour chaque scénario
{ int i,j; double S;
  double *R = NewVector(m);for(i=1;i<=m;++i)R[i]=rot(i);
  for(j=1;j<=n;j++)
  {for(S=0,i=1;i<=m;++i)S+=F[j][i]*pow(1+Z[i],-i);VM[j][1]=S;}
  for(j=1;j<=n;j++)
  {for(S=0,i=1;i<=m;++i)S+=F[j][i]*pow(1.01+Z[i],-i);VM[j][2]=S;}
  for(j=1;j<=n;j++)
  {for(S=0,i=1;i<=m;++i)S+=F[j][i]*pow(0.99+Z[i],-i);VM[j][3]=S;}
  for(j=1;j<=n;j++)
  {for(S=0,i=1;i<=m;++i)S+=F[j][i]*pow(1+R[i],-i);VM[j][4]=S;}
  for(i=1;i<=3;i++)
  for(j=1;j<=n;j++)C[j][i]=fabs(VM[j][1]- VM[j][i+1]);
}

void Sens(double **ETA,double **F,double*Z,double P,int m,int n)
// Calcul du vecteur de sensibilité (Le vecteur ETA)
{ int i,j; double S1,S2;
  for(i=1;i<=m;i++)
  for(j=1;j<=n;j++)ETA[j][i]=i*F[j][i]*pow(1+Z[i],-i-1);

  // Calcul du rapport TAS
  for(j=1;j<=n;j++)
  {for(S2=0,S1=0,i=1;i<=m;i++){S2+=ETA[j][i]; S1+=i*ETA[j][i];}
  ETA[j][m+1]=P*S2; ETA[j][m+2]=S1/S2;}
}

void TTM(double *DUR,double **F,double*Z,int m,int n,int f)
// Calcul de la duration des instruments financiers.
{ int i,j; double S1,S2,S;
  for(j=1;j<=n;j++)
  {for(S1=0,S2=0,i=1;i<=m;i++){S=F[j][i]*pow(1+Z[i],-i-f);S2+=i*S;S1+=S;}
  DUR[j]=S2/S1;}
}

int Select_K(double *a,int *flag,int*J,int n,int K)
// Tri croissant d'un vecteur de taille n (TriShell) de complexité O(n^(3/2))
{ int i,j,h,t; double v;
  static int *I=NewPerm(n);for(i=1;i<=n;i++)I[i]=i;
  for(h=1;h<=n;h=3*h+1);
  do
  {h/=3;
  for (i=h+1;i<=n ;i++)
  { v = a[i]; t=I[i]; j = i;
  while (a[j-h]>v)
  { a[j] = a[j-h];I[j]=I[j-h]; j-= h; if(j <=h ) break;}
  a[j] = v; I[j] = t;}
}
}

```

```

    } while(h>1);
    for(j=0,i=1;i<=K;i++)if(flag[I[i]])J[++j]=I[i];
    return j;
}

int Minimiser(double **A,double *b,double *C,double *d,double *X,double dmin,
             int n,double alpha,double beta,double r,int MaxIter, double Tol)

// Algorithme D'UZAWA à pas fixe
// Résolution du sous-problème quadratique (SPQ)

{int i, j, iter=0;
 double som, P=0, S;
 static double *Q=NewVector(n),*Y=NewVector(n);
 for(i=1;i<=n;i++) for(j=1;j<=n;j++)A[i][j]+= r*(C[i]*C[j]);
 for(i=1;i<=n;i++)A[i][i]+=r;
 if(Cholesky(A,n,ZERO)==-1)return(-2);
 for(i=1;i<=n;i++)X[i]=d[i];
 for(som=0.0,i=1;i<=n;i++)som+=C[i]*X[i];
 do{
 iter++;
 S=som+P/r; S=(S<alpha)? alpha :(S>beta)? beta : S;
 for(i=1;i<=n;i++){Y[i]=X[i]+Q[i]/r;
 Y[i]=(Y[i]<dmin)?dmin:(Y[i]>d[i]?d[i]:Y[i]);}
 for(i=1;i<=n;i++) X[i]= b[i]-Q[i]+r*Y[i]+(r*S-P)*C[i];
 Cholesky_Solve(A,X,n);
 //for(som=0,i=1;i<=n;i++)som+=fabs(X[i]-Y[i]);som/=n;
 for(som=0,i=1;i<=n;i++)som+=C[i]*X[i];som=fabs(som-S);
 if(som < Tol) break;
 for(som=0,i=1;i<=n;i++) som+=C[i]*X[i]; P+=w*r*(som-S);
 for(i=1;i<=n;i++)Q[i]+=w*r*(X[i]-Y[i]);}
 while(iter < MaxIter);
 for(i=1;i<=n;i++)X[i]=(X[i]<dmin)?dmin:(X[i]>d[i]?d[i]:X[i]);
 for(som=0,i=1;i<=n;i++)som+=C[i]*X[i];
 return (S>beta)|| (S<alpha)?-1:(iter<MaxIter)?iter:0;
}

int Hedge(double **SL,double*SS,int *flag,double*C, double*d,double a,
          int MaxComb,int K,int kmax,double*X,double &FX,int *I, int nL,int m)
// Résolution du problème (un seul Swap)
{ int i,j;
 static double *D=NewVector(nL);
 for(j=1;j<=nL;j++)D[j] = fabs(SL[j][m+2]-SS[m+2]);
 static int *J=NewPerm(nL);
 int Card_K = Select_K(D,flag,J,nL,K);
 if(Card_K<1) return(0);
 int k = Card_K > kmax ? kmax :Card_K ;
 int NC = Card_K >k ? 250*(Card_K-k):1; NC=(NC>MaxComb)?MaxComb:NC;
 static double **A=NewMatrix(k,k), *b=NewVector(k);
 static double *Ck=NewVector(k), *dk=NewVector(k);
 static double *Y=NewVector(k),FY; FX=INF;
 static int *Imin=NewPerm(k);
 static int prem = VRAI;

 for(int _k=1;_k<=NC;_k++)

{ if (prem){for(i=1;i<=k;i++)Imin[i]=J[i];prem = FAUX;}
 else NewCombinaison(Imin,J,Card_K);
 Pb_Quad(SL,SS,C,d,Imin,k, m+1,A,b,Ck,dk);
 j=Minimiser(A,b,Ck,dk,Y,0,k,a/ub,a/lb,r,MaxIter,Tol);
 if(j<=0)continue;
 FY = EvalF(SL,Y,SS,Imin,m+1,k);
 if(FY<FX){for(i=1;i<=k;i++){I[i]=Imin[i];X[i]=Y[i];} FX = FY;}
}
prem = VRAI;

return j>0 ? k : -1 ;
}

```

```

void Global_hedge(double **SL,double**SS,int *flag,double*C, double*d,
                double *a,int MaxComb,int K,int kmax,int *k,double**X,
                double *FX,int **I, int m,int nL,int nS)
// Résolution du problème (plusieurs Swaps)
{int i,j;
for(j=1;j<=nS;j++)
{k[j]=Hedge(SL,SS[j],flag,C,d,a[j],MaxComb,K,kmax,X[j],FX[j],I[j],nL,m);
if(k[j]<1)continue;
// Marquage de prêts non disponibles
for(i=1;i<=k[j];i++){d[I[j][i]]-=X[j][i];flag[I[j][i]]=d[I[j][i]]>dmin;}
}
}

void afficher(double**A,int n,int m,const char *str)
// Fonction d'affichage
{int i,j;
for(i=1;i<=m;i++)
{for(j=1;j<=n;j++) printf(str,A[j][i]);printf("\n");
printf("\n");
}
// *** Début du programme ***

void main()
{ // Initialisation des paramètres

const int m = 10 , // La durée de vie de la couverture
nL = 11 , // nombre de prêts(Loans)
nS = 3 , // Nombre de Swaps
kmax = 6 , // La taille maximale de combinaison
K = 11 , // Le nombre maximal de prêt homogènes
MaxComb = 450 ; // Le nombre maximal de combinaisons
const double P = 10 ; // Le facteur de pénalisation

//Corps du programme

int i,j;
double *Taux_L,*Taux_S,*Nominal_L,*Nominal_S;
int *Piq_L,*Piq_S;
double *Z = NewVector(m);for(i=1;i<=m;+i)Z[i]=Courbe(i);
Nominal_L = NewVector(nL);Nominal_S = NewVector(nS);
Taux_L = NewVector(nL); Taux_S = NewVector(nS);
Piq_L = NewIVector(nL); Piq_S = NewIVector(nS);

//Les prêts
Nominal_L[1]=60;Nominal_L[2]=70;Nominal_L[3]=40;Nominal_L[4]=35;
Nominal_L[5]=45;Nominal_L[6]=30;Nominal_L[7]=40;Nominal_L[8]=40;
Nominal_L[9]=30;Nominal_L[10]=40;
Taux_L[1]=6.51;Taux_L[2]=6.466;Taux_L[3]=6.43;Taux_L[4]=6.4;Taux_L[5]=6.3;
Taux_L[6]=6.20;Taux_L[7]=6.25; Taux_L[8]=6.30;Taux_L[9]=6.55;Taux_L[10]=6.15;

Piq_L[1]=9;Piq_L[2]=8;Piq_L[3]=7;Piq_L[4]=6;Piq_L[5]=5;Piq_L[6]=3;
Piq_L[7]=4;Piq_L[8]=2;Piq_L[9]=10;Piq_L[10]=1;

// Les swaps
Nominal_S[1]=160; Nominal_S[2]=135; Nominal_S[3]=100;
Taux_S[1]=6.467; Taux_S[2]=6.362; Taux_S[3]=6.235;
Piq_S[1]=8; Piq_S[2]=6; Piq_S[3]=4;

double **FL,**FS;FL=NewMatrix(nL,m);FS=NewMatrix(nS,m);
double A = Amor(60,6.35,10); Nominal_L[11]=60;
for(i=1;i<=m;+i) FL[nL][i]=A;
Cash_Flows(FL,Nominal_L,Taux_L,Piq_L,m,nL-1);
Cash_Flows(FS,Nominal_S,Taux_S,Piq_S,m,nS);
printf("\nLa courbe des taux(zéro-coupon en %)\n");printf("\n");
for(i=1;i<=m ;i++)printf("%7d",i); printf("\n");
for(i=1;i<=m ;i++)printf("%7.2f",100*Z[i]); printf("\n\n");
printf("-----\n");
printf("\nLes cash-flows des prets\n");printf("\n");
for(i=1;i<=nL ;i++)printf("%7d",i); printf("\n\n");
}

```

```

afficher (FL,nL,m,"%7.2f");
printf(" -----\n");
printf("\nLes cash-flows des Swaps\n");printf("\n");
for(i=1;i<=nS ;i++)printf("%7d",i); printf("\n\n");
afficher (FS,nS,m,"%7.2f");

double ** VM_L,** VM_S,**C_L,**C_S;
VM_L=NewMatrix(nL,4);VM_S= NewMatrix(nS,4);
C_L =NewMatrix(nL,3);C_S = NewMatrix(nS,3);
Valeur_M(VM_L,C_L,FL,Z,m,nL);
Valeur_M(VM_S,C_S,FS,Z,m,nS);

printf("\n Les variations de prix en chaque scénario C[i][j]\n");
printf("\n");
for(i=1;i<=nL ;i++)printf("%7d",i); printf("\n\n");
afficher (C_L,nL,3,"%7.2f");
printf(" -----\n");
printf("\n les variations du prix des swaps en chaque scénario:a[i]\n");
printf("\n");
for(i=1;i<=nS ;i++)printf("%7d",i); printf("\n\n");
afficher (C_S,nS,3,"%7.2f");

double **ETA_L = NewMatrix(nL,m+2);
double **ETA_S = NewMatrix(nS,m+2);

Sens (ETA_L,FL,Z,P,m,nL); Sens (ETA_S,FS,Z,P,m,nS);
printf(" -----\n");
printf("\nLes Vecteurs de sensibilité des prêts\n");printf("\n");
for(i=1;i<=nL ;i++)printf("%7d",i); printf("\n\n");
afficher (ETA_L,nL,m+1,"%7.1f");
printf(" -----\n");
printf("\nLes Vecteurs de sensibilité des swaps\n");printf("\n");
for(i=1;i<=nS ;i++)printf("%10d",i); printf("\n\n");
afficher (ETA_S,nS,m+1,"%10.2f");

//Minimisation
double *C,**X,*d,*a,*FX;

//Allocation mémoire
X = NewMatrix(nS,kmax); FX = NewVector(nS);
d = NewVector(nL); C = NewVector(nL); a = NewVector(nS);

int **I=new int*[nS];I--=1; for(i=1;i<=nS;i++){I[i]=new int[kmax];I[i]--=1;}
for(j=1;j<=nS;j++) for(i=1;i<=kmax;i++) I[j][i]=0;
int *k=new int[nS];k--=1; for(i=1;i<=nS;i++)k[i]=1;

for(j=1;j<=nL;j++) C[j]=C_L[j][3];
for(j=1;j<=nS;j++) a[j]=C_S[j][3];

// Au début, tous les prêts sont disponibles
for(j=1;j<=nL;j++) d[j]=1.0; //d[j]=100%
int *flag = NewPerm(nL); for(i=1;i<=nL;i++) flag[i] = VRAI;

double t = Time();

Global_hedge (ETA_L,ETA_S,flag,C,d,a,MaxComb,K,kmax,k,X,FX,I,m,nL,nS);
t = Time()-t;

double *R = NewVector(3);

for(j=1;j<=nS;j++) //Pour chaque swap

{if(k[j]<=1) continue;
printf("\n Id Nominal Taux(%) TAS\n");
printf("SWAP_%d%10.2f%11.2f%13.2f\n",j,Nominal_S[j],Taux_S[j],ETA_S[j][m+2]);
printf("\nLa sensibilité quadratique : %10.2f\n ",FX[j]);
printf("\nRatios\n"); Ratios(R,C_L,X[j],a[j],I[j],k[j]);
}

```



```

for(i=1;i<=3;i++) printf("R[%d]=%6.2f\n",i,100*R[i]);

printf("\n");
printf("  Id          Nominal    Portions(%)   TAS\n\n");
for(i=1;i<=k[j];i++)
printf("  Loan_ %2d%10.2f %10.2f%10.2f\n",I[j][i],
      Nominal_L[I[j][i]], 100*X[j][i],ETA_L[I[j][i]][m+2]);

printf("\n-----\n");
printf(" nombre de loans disponibles %2d\n",nLoans(flag,nL)); printf("\n");
for(i=1;i<=nL;i++) if(flag[i])printf("Loan_ %2d%10.2f\n",i,100*d[i]);
printf("\n Le Temps d'exécution est : %5.2f Seconde(s)\n",t);
double *H = new double[m] ;H-=1;
Residu(ETA_L,X[1],H,I[1],m,k[1]);
printf("\n Composition    Swap_1    Residu\n");
for(i=1;i<=m;i++)
printf(" %10.2f%10.2f%10.2f\n ",H[i],ETA_S[1][i],ETA_S[1][i]-H[i]);
double *DUR = new double[nL]; DUR-=1;
printf("\n");
TTM(DUR, FL,Z, m,nL,0);
for(i=1;i<=nL;i++)printf("%8.3f\n",DUR[i]);
printf("\n");
TTM(DUR, FS,Z,m,nS,0);
for(i=1;i<=nS;i++)printf("%8.3f\n",DUR[i]);
printf("\n\n");
TTM(DUR, FL,Z, m,nL,1);
for(i=1;i<=nL;i++)printf("%8.3f\n",DUR[i]);
printf("\n");
TTM(DUR, FS,Z,m,nS,1);
for(i=1;i<=nS;i++)printf("%8.3f\n",DUR[i]);

} /* fin du programme */

```

FIN DU PROGRAMME.