Dauphine | PSL
UNIVERSITÉ PARIS

Optimisation en finance

Clément W. Royer

Notes de cours - M2 MIAGE - 2022/2023

- La version la plus récente de ces notes se trouve à l'adresse : **https://www.lamsade.dauphine.fr/∼croyer/ensdocs/OF/PolyOF.pdf**.
- Pour tout commentaire, merci d'envoyer un mail à `clement.royer@lamsade.dauphine.fr`.
- **Historique du document**
  - 2022.12.13 : Correction d'une ambiguïté dans la partie 2.2.2.
  - 2022.12.04 : Version complète.
  - 2022.11.15 : Première version.
- **Objectifs d'apprentissage**
  - Comprendre les caractéristiques des modèles d'optimisation les plus classiques en finance.
  - Connaître les grands principes derrière la résolution de ces modèles et les informations à en tirer.
  - Étudier des exemples de problèmes en finance et leur modélisation via des outils d'optimisation.

Sommaire

Avant-propos

Le but de ce cours est de présenter les classes de problème d'optimisation les plus utilisés en finance, et les algorithmes associés à leur résolution. Il se veut découpé en plusieurs parties suivant le déroulement des séances de cours.

Ce cours n'est pas un cours de mathématiques, mais il repose sur plusieurs concepts élémentaires d'algèbre linéaire et de calcul différentiel. Ceux-ci ne seront pas rappelés en détail en cours, mais sont en revanche détaillés en annexe de ce polycopié. De même, les notations cruciales du polycopié sont décrites ci-après.

Notations

- Les scalaires seront représentés par des lettres minuscules : $a, b, c, \alpha, \beta, \gamma$.
- Les vecteurs seront représentés par des lettres minuscules **en gras** : $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$.
- Les lettres majuscules en gras seront utilisées pour les matrices : $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$.
- Les lettres majuscules cursives seront utilisées pour les ensembles : $\mathcal{A}, \mathcal{B}, \mathcal{C}$.
- La définition d'une nouvelle quantité ou d'un nouvel opérateur sera indiquée par $:=$.
- L'ensemble des entiers naturels sera noté $\mathbb{N}$, l'ensemble des entiers relatifs sera noté $\mathbb{Z}$.
- L'ensemble des réels sera noté $\mathbb{R}$. L'ensemble des réels positifs sera noté $\mathbb{R}_+$ et l'ensemble des réels strictement positifs sera noté $\mathbb{R}_{++}$.
- On notera $\mathbb{R}^n$ l'ensemble des vecteurs à n composantes réelles, et on considèrera toujours que n est un entier supérieur ou égal à 1.
- Un vecteur $\boldsymbol{x} \in \mathbb{R}^n$ sera pensé (par convention) comme un vecteur colonne. On notera $x_i \in \mathbb{R}$ sa i-ème coordonnée dans la base canonique de $\mathbb{R}^n$. On aura ainsi $\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$. Étant donné un vecteur (colonne) $\boldsymbol{x} \in \mathbb{R}^n$, le vecteur ligne correspondant sera noté $\boldsymbol{x}^{\mathrm{T}}$. On aura donc $\boldsymbol{x}^{\mathrm{T}} = [x_1 \ \cdots \ x_n]$ et $[\boldsymbol{x}^{\mathrm{T}}]^{\mathrm{T}} = \boldsymbol{x}$.
- Le produit scalaire entre deux vecteurs de $\mathbb{R}^n$ est défini par $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{v} = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{x} = \sum_{i=1}^{d} x_i v_i$. On définit ensuite la norme d'un vecteur $\boldsymbol{x} \in \mathbb{R}^n$ par $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}$.
- On notera $\mathbb{R}^{m\times n}$ l'ensemble des matrices à m lignes et n colonnes à coefficients réels, où m et n seront des entiers supérieurs ou égaux à 1. Une matrice $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ est dite carrée (dans le cas général, on parlera de matrice rectangulaire).
- On identifiera les vecteurs de $\mathbb{R}^n$ avec les matrices de $\mathbb{R}^{n\times 1}$.
- Étant donnée une matrice $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, on notera $\boldsymbol{A}_{ij}$ le coefficient en ligne i et colonne j de la matrice. La diagonale de $\boldsymbol{A}$ est formée par l'ensembles des coefficients $\boldsymbol{A}_{ii}$ pour $i = 1, \ldots, \min\{m, n\}$. La notation $[\boldsymbol{A}_{ij}]_{\substack{1\leq i\leq m \\ 1\leq j\leq n}}$ sera donc équivalente à $\boldsymbol{A}$. Sans ambigüité sur la taille de la matrice, on notera simplement $[\boldsymbol{A}_{ij}]$.
- Selon les besoins, on utilisera $\boldsymbol{a}_i^{\mathrm{T}}$ pour la i-ème ligne de $\boldsymbol{A}$ ou $\boldsymbol{a}_j$ pour la j-ème colonne de $\boldsymbol{A}$. Selon le cas, on aura donc $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{a}_m^{\mathrm{T}} \end{bmatrix}$ ou $\boldsymbol{A} = [\boldsymbol{a}_1 \ \cdots \ \boldsymbol{a}_n]$.
- Pour une matrice $\boldsymbol{A} = [\boldsymbol{A}_{ij}] \in \mathbb{R}^{m\times n}$, la matrice transposée de $\boldsymbol{A}$, notée $\boldsymbol{A}^{\mathrm{T}}$, est la matrice de $\mathbb{R}^{n\times m}$ telle que
$$\forall i = 1 \ldots m, \ \forall j = 1 \ldots n, \quad \boldsymbol{A}^{\mathrm{T}}_{ji} = \boldsymbol{A}_{ij}.$$
- Pour tout $n \geq 1$, $\mathbf{I}_n$ correspond à la matrice identité $\mathbb{R}^{n\times n}$ (avec 1 sur la diagonale et 0 ailleurs).

Chapitre 1

Introduction à l'optimisation

Le but de ce cours est d'étudier les problèmes d'optimisation issus de la science des données, ainsi que les algorithmes qui leur sont appliqués. Le concept d'optimisation existe bien au-delà de l'apprentissage, mais il est particulièrement important pour ce dernier. De nombreuses tâches d'apprentissage peuvent en effet être formulées comme des problèmes d'optimisation, et ceux-ci peuvent être résolus efficacement par des algorithmes modernes. Ce chapitre présente les principes de base derrière la formulation et l'étude (à la fois mathématique et algorithmique) d'un problème d'optimisation.

1.1 Définition et contexte

1.1.1 Problème et processus d'optimisation

Avant d'être un concept mathématique, un problème d'optimisation peut être verbalisé : on en donne une décomposition ci-dessous.

Définition 1.1.1 *Un problème d'optimisation mathématique est formé par les trois éléments suivants.*

- *Une* **fonction objectif** *: il s'agit du critère qui permet de quantifier la qualité d'une décision. Selon le cas, une meilleure décision peut signifier une valeur de l'objectif plus faible (auquel cas on aura affaire à un problème de* minimisation*) ou plus élevée (il s'agira alors d'un problème de* maximisation*).*
- *Des* **variables de décision** *: ce sont les différents choix qu'il est possible d'effectuer, dont la valeur influence celle de la fonction objectif. On cherche les meilleures valeurs de ces variables relativement à l'objectif.*
- *Des* **contraintes** *: celles-ci spécifient des conditions qui doivent être satisfaites par les variables de décision pour que les valeurs correspondantes soient acceptables.*

Tout processus d'optimisation passe nécessairement par une phase de **modélisation**, durant laquelle on transforme un problème concret en objet mathématique que l'on peut essayer de résoudre. Pour cela, on se base sur des **algorithmes d'optimisation** qui, bien que pouvant être formulés à la main, ont généralement vocation à être implémentés sur un ordinateur. Lorsque l'algorithme appliqué

au problème renvoie une réponse, celle-ci doit être examinée pour savoir si elle fournit une solution satisfaisante au problème : au besoin, le modèle pourra être modifié, et l'algorithme ré-executé. En pratique, le processus d'optimisation peut donc boucler après une phase d'**interprétation** ou de **discussion** avec des experts du domaine d'application dont le problème est issu.

Remarque 1.1.1 *L'optimisation numérique obéit généralement aux principes suivants :*

- *Il n'y a pas d'algorithme universel : certaines méthodes seront très efficaces sur certains types de problèmes, et peu efficaces sur d'autres. L'étude de la structure du problème facilite le choix d'un algorithme.*
- *Il peut y avoir un fort écart entre la théorie et la pratique : le calcul en précision finie peut induire des erreurs d'arrondis qui conduisent à une performance en-deçà de celle prévue par la théorie mathématique.*
- *La théorie guide la pratique, et vice-versa : pour la plupart des problèmes, il est possible de définir des expressions mathématiques qui permettent de vérifier si l'on a trouvé une solution du problème. Celles-ci sont à la base de nombreux algorithmes que nous étudierons. De manière générale, l'étude théorique d'un problème permet des avancées garanties qui surpassent souvent les heuristiques. Réciproquement, la performance de certaines méthodes a amené les chercheurs en optimisation à en proposer une étude théorique, permettant ainsi d'expliquer le comportement pratique.*

1.1.2 Le contexte de la finance

L'optimisation est utilisée en finance comme un outil de modélisation et de résolution efficace de problèmes.

Gestion des risques Durant la dernière décennie et suite à la crise dite des *subprimes* de 2007-2008, la règlementation sur les niveaux de risques autorisés dans les milieux financiers a considérablement évolué. Les opérations financières sont aujourd'hui exécutées En termes d'optimisation, on considèrera ainsi des problèmes sous contraintes de niveau de risque, tels que la maximisation du retour sur investissement.

Gestion des actifs (assets) et des dettes (liabilities) La modélisation de valeurs d'actifs ou de dettes prend en compte l'aléatoire sous-jacent au marché. En termes d'optimisation, on obtient ainsi des problèmes d'optimisation stochastique, où le caractère aléatoire est modélisé par l'introduction de variables aléatoires dont la valeur évolue sur une ou plusieurs périodes.

1.2 Logiciels et optimisation

L'optimisation a connu un essor majeur au cours des années 1980 avec le développement des ordinateurs. Une composante majeure

Les langages de programmation les plus utilisés pour développer des algorithmes d'optimisation étaient historiquemnet `C/C++/Fortran`, notamment pour développer des codes performants. Le développement de prototypes de recherche se fait lui plutôt via des langages interprétés tels que

`Matlab/Octave/Python/Julia`, avec `Python` prenant de plus en plus d'importance de par son utilisation dans les problèmes de sciences des données et apprentissage.

En plus des langages de programmation standard, de nombreux langages ou bibliothèques ont été développés pour modéliser les problèmes d'optimisation. On peut citer notamment `GAMS`, `AMPL`, `CVX`, `Pyomo`, qui ont un champ d'application génériques, ou `MATPOWER` et `PyTorch`, qui sont dédiés à un domaine particulier (respectivement les réseaux d'énergie et l'apprentissage machine).

En finance Dans le contexte de la finance, le recours aux tableurs est prépondérant, ce qui a conduit au développement de solveurs en tant qu'outil interne aux tableurs. C'est ainsi que Microsoft Excel, LibreOffice Calc et Google Sheets possèdent des solveurs, notamment de programmation linéaire. Si certains sont disponibles gratuitement (*open source*), le solveur le plus abouti (Microsoft Excel Solver) requiert une licence d'exploitation[1]

D'autres outils plus bas niveau ont également été développés, parfois avec un spectre plus large d'applications. Le logiciel MATLAB (propriétaire) dispose ainsi d'une boîte à outils de finance très fournie, qui comporte notamment des algorithmes d'optimisation. On s'intéressera dans ce cours à la bibliothèque `cvxpy`, développée à l'origine à l'université de Stanford (États-Unis), et qui dispose de nombreux outils pour résoudre les problèmes de programmation convexe.

1.3 Bases de l'optimisation mathématique

1.3.1 Formulation générale et premières définitions

La transformation d'une description formelle d'un problème d'optimisation en objet mathématique conduit à l'écriture suivante :

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimiser}}\, f(\boldsymbol{x}) \quad \text{s. c.} \quad \boldsymbol{x}\in\mathcal{X}, \tag{1.3.1}$$

où minimiser représente ce que l'on cherche à faire (on peut vouloir minimiser ou maximiser), $\boldsymbol{x}\in\mathbb{R}^n$ est un vecteur regroupant les variables de décision du problème, $f:\mathbb{R}^n\to\mathbb{R}$ est la fonction objectif qui mesure la qualité des décisions, et $\mathcal{X}\subset\mathbb{R}^n$ est l'ensemble des points réalisables ou admissibles, qui vérifient les contraintes posées sur les variables de décision. [2]

Définition 1.3.1 *i) Un point $\boldsymbol{x}\in\mathbb{R}^n$ tel que $\boldsymbol{x}\in\mathcal{X}$ est dit* **admissible**, *ou* **réalisable**.

ii) Un point $\boldsymbol{x}\in\mathbb{R}^n$ tel que $\boldsymbol{x}\notin\mathcal{X}$ est dit **irréalisable**.

iii) Si $\mathcal{X}=\emptyset$, alors le problème (1.3.1) est dit irréalisable.

Définition 1.3.2 *i) La valeur minimale (ou le minimum) du problème (1.3.1) est notée :*

$$\min_{\boldsymbol{x}\in\mathbb{R}^n}\{f(\boldsymbol{x}) s.\ c.\ \boldsymbol{x}\in\mathcal{X}\}. \tag{1.3.2}$$

Lorsque le problème possède une solution, la valeur minimale est la valeur de f en toute solution, et cette valeur est finie. Lorsque le problème est irréalisable, on pose par convention

$$\min_{\boldsymbol{x}\in\mathbb{R}^n}\{f(\boldsymbol{x}) s.\ c.\ \boldsymbol{x}\in\mathcal{X}\}=+\infty.$$

[1] Il est à noter que si un compte passeport Dauphine permet d'accèder à Microsoft Excel, il ne donne pas l'autorisation d'installer des extensions, et donc ne permet pas l'utilisation de Solver.

[2] L'abbréviation *s.c.* signifie "sous la/les contrainte(s)"; en anglais, on utilise *s.t.*, pour *subject to*.

Enfin, si le problème est réalisable mais qu'il n'existe pas de solution, on aura

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \{f(\boldsymbol{x}) s.\ c.\ \boldsymbol{x}\in\mathcal{X}\} = -\infty,$$

et on dit alors que le problème est **non borné**.

ii) L'ensemble des solutions du problème (1.3.1) *est noté*

$$\operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} \{f(\boldsymbol{x}) s.\ c.\ \boldsymbol{x}\in\mathcal{X}\}, \tag{1.3.3}$$

qu'on appelle argument minimal *(ou argmin) du problème. C'est un sous-ensemble de $\mathbb{R}^n$ (et de $\mathcal{X}$) : il s'agit de l'ensemble des points de $\mathcal{X}$ qui donnent la valeur minimale de f. Il est vide si le problème est irréalisable ou réalisable et non borné.*

Remarque 1.3.1 *Le problème* (1.3.1) *n'admet pas forcément de solution (prendre par exemple $d=1$, $\mathcal{F}=\mathbb{R}$ et $f(w)=w$) : dans ce cas, l'argument minimal est l'ensemble vide, et la valeur minimale est $-\infty$.*

On définit de la même manière que précédemment les problèmes de maximisation ainsi que les opérateurs argmax et $\max$. Dans ce cours, on se concentrera sur les problèmes de minimisation, utilisant en cela la propriété ci-dessous.

Proposition 1.3.1 *Pour toute fonction $f:\mathbb{R}^n\to\mathbb{R}$ et tout ensemble $\mathcal{F}\subset\mathbb{R}^n$, résoudre le problème de maximisation*

$$\operatorname*{maximiser}_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \quad s.\ c. \quad \boldsymbol{x}\in\mathcal{F}$$

équivaut à résoudre le problème de minimisation

$$\operatorname*{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} -f(\boldsymbol{x}) \quad s.\ c. \quad \boldsymbol{x}\in\mathcal{F},$$

dans la mesure où

$$\operatorname*{argmax}_{\boldsymbol{x}\in\mathbb{R}^n} \{f(\boldsymbol{x}) s.\ c.\ \boldsymbol{x}\in\mathcal{F}\} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \{-f(\boldsymbol{x}) s.\ c.\ \boldsymbol{x}\in\mathcal{F}\}$$

et

$$\max_{\boldsymbol{x}\in\mathbb{R}^n} \{f(\boldsymbol{x}) s.\ c.\ \boldsymbol{x}\in\mathcal{F}\} = -\min_{\boldsymbol{x}\in\mathbb{R}^n} \{-f(\boldsymbol{x}) s.\ c.\ \boldsymbol{x}\in\mathcal{F}\}.$$

Tout problème de maximisation se ramène donc à un problème de minimisation via une reformulation. Nous verrons d'autres exemples de reformulation, qui est une technique très utilisée pour transformer un problème en un autre problème équivalent mais que l'on saura mieux traiter théoriquement et/ou algorithmiquement.

Dans la majorité de ce cours, et notamment dans le reste de ce chapitre, on se concentrera sur des problèmes d'optimisation sans contraintes pour en caractériser plus précisément les solutions.

1.4 Conclusion du chapitre 1

L'optimisation est un outil mathématique qui permet de modéliser de nombreux problèmes en sciences des données et au-delà. L'optimisation comporte toujours une phase de modélisation, qui consiste à formuler mathématiquement le problème en définissant une fonction objectif, des variables de décisions et des contraintes. Cela permet de placer le problème dans une classe, et ainsi de caractériser ce qu'est une solution de ce problème.

Chapitre 2

Optimisation linéaire en finance

Dans ce chapitre, nous nous intéressons aux problèmes de programmation linéaire. Le développement algorithmique lié à la résolution de ces problèmes, dû notamment à George Dantzig dans les années 1960, a conduit à des avancées majeures dans de nombreux domaines d'application, dont la finance. Les programmes linéaires forment une classe bien comprise de problèmes d'optimisation qui permettent à la fois de modéliser de nombreux problèmes concrets et également de les résoudre avec des algorithmes efficaces, même en présence d'un très grand nombre de variables.

2.1 Bases de la programmation linéaire

Formellement, un programme linéaire consiste à minimiser ou maximiser une fonction linéaire des variables de décision sous la contrainte que ces variables appartiennent à un ensemble décrit par des équations et inéquations linéaires.

Définition 2.1.1 *Un programme linéaire est un problème d'optimisation de la forme*

$$\begin{aligned} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.c.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{D}\boldsymbol{x} \geq \boldsymbol{f}. \end{aligned}$$

avec $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{D} \in \mathbb{R}^{\ell\times n}$, $\boldsymbol{f} \in \mathbb{R}^\ell$, $\boldsymbol{c} \in \mathbb{R}^n$.

Définition 2.1.2 *Un programme linéaire en* **forme standard** *est de la forme*

$$\begin{aligned} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.c.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned}$$

où les contraintes d'intervalle $\boldsymbol{x} \geq \boldsymbol{0}$ *peuvent porter seulement sur un sous-ensemble des coordonnées de* $\boldsymbol{x}$.

Tout programme linéaire peut se mettre sous forme standard en introduisant des variables supplémentaires. Par exemple, la contrainte $x_1 \geq 1$ peut se réécrire comme

$$x_1 - s_1 = 1, s_1 \geq 0.$$

Exemple 2.1.1 (Allocation de fonds (Cornuéjols et al. 2018)) *On considère un ensemble de quatre fonds d'investissement ayant les propriétés suivantes:*

Fonds	*Fonds 1*	*Fonds 2*	*Fonds 3*	*Fonds 4*
Part capitalisation haute	*50%*	*30%*	*25%*	*60%*
Part capitalisation moyenne	*30%*	*10%*	*40%*	*20%*
Part capitalisation basse	*20%*	*60%*	*35%*	*20%*
Retour sur investissement	*10%*	*15%*	*16%*	*8%*

On souhaite allouer 80000€ parmi cet ensemble, de sorte à avoir

- *Au moins 35% de l'allocation en capitalisation haute;*
- *Au moins 30% de l'allocation en capitalisation moyenne;*
- *Au moins 15% de l'allocation en capitalisation basse.*

*On suppose que l'on ne peut investir que positivement dans un fonds (*long-only positions*), et on cherche l'allocation qui maximise le retour sur investissement.*

Pour modéliser cet exemple via un programme linéaire, on commence par identifier les variables : on définit ainsi $\boldsymbol{x} \in \mathbb{R}^4$, où x_i correspond à l'investissement dans le fond i (exprimé en k€). On obtient ainsi le problème de maximisation suivant

$$\begin{array}{rlcl} \textit{maximiser}_{\boldsymbol{x}\in\mathbb{R}^4} & 0.1x_1 + 0.15x_2 + 0.16x_3 + 0.08x_4 & & \\ \textit{s.c.} & 0.5x_1 + 0.3x_2 + 0.25x_3 + 0.6x_4 & \geq & 28 \\ & 0.3x_1 + 0.1x_2 + 0.4x_3 + 0.2x_4 & \geq & 24 \\ & 0.2x_1 + 0.6x_2 + 0.35x_3 + 0.2x_4 & \geq & 12 \\ & x_1 + x_2 + x_3 + x_4 & = & 80 \\ & \boldsymbol{x} & \geq & \mathbf{0}. \end{array} \tag{2.1.1}$$

Le problème (2.1.1) n'est pas en forme standard, mais peut être converti sous cette forme en utilisant la procédure décrite plus haut. Il possède trois contraintes d'inégalité linéaires, une contrainte d'égalité linéaire et quatre contraintes de positivité concaténées sous la forme $\boldsymbol{x} \geq \mathbf{0}$.

2.2 Résolution d'un programme linéaire

La théorie de la programmation linéaire permet à la fois de déterminer si un programme linéaire possède une solution et de donner des conditions qui caractérisent ces solutions si elles existent. Un outil majeur

2.2.1 Dualité et conditions de KKT

Dans cette partie, on se concentre sur le problème suivant, appelé **problème primal**

$$\left\{ \begin{array}{rl} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.c.} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \mathbf{0}, \end{array} \right. \tag{$\mathcal{P}$}$$

On fera les hypothèses simplificatrices suivantes.

Hypothèse 2.2.1
- $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *est de rang plein.*
- *Les contraintes de positivité s'appliquent à tous les* x_i.

Théorème 2.2.1 (Conditions de Karush-Kuhn-Tucker (KKT)) *On considère le problème* $(\mathcal{P})$ *sous l'hypothèse* (2.2.1). *Un point* $\boldsymbol{x}^* \in \mathbb{R}^n$ *est une solution de* $(\mathcal{P})$ *si et seulement si il existe* $\boldsymbol{y}^* \in \mathbb{R}^m$ *et* $\boldsymbol{s}^* \in \mathbb{R}^n$ *tels que* $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{s}^*)$ *vérifie le système*

$$\left\{\begin{array}{lcll} \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}^* + \boldsymbol{s}^* & = & \boldsymbol{c} & \\ \boldsymbol{A}\boldsymbol{x}^* & = & \boldsymbol{b} & \\ \boldsymbol{x}^* & \geq & \boldsymbol{0} & \\ \boldsymbol{s}^* & \geq & \boldsymbol{0} & \\ x_i^* s_i^* & = & 0 & \forall i = 1, \ldots, n. \end{array}\right. \tag{KKT-Lin}$$

Les vecteurs $\boldsymbol{y}^*$ et $\boldsymbol{s}^*$ sont les multiplicateurs de Lagrange associés aux contraintes d'égalité et de positivité. En optimisation, on les appelle variables duales.

Remarque 2.2.1 *En finance, les variables duales associées à une contrainte linéaire sont appelées des* shadow prices, *ou* **prix fictifs**. *Dans le cas d'une contrainte de positivité, le multiplicateur associé sera appelé un* **coût réduit**.

Définition 2.2.1 (Problème dual) *Le problème dual de* $(\mathcal{P})$ *est donné par*

$$\left\{\begin{array}{rl} \text{maximiser}_{\substack{\boldsymbol{y}\in\mathbb{R}^m \\ \boldsymbol{s}\in\mathbb{R}^n}} & \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y} \\ \text{s.c.} & \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c} \\ & \boldsymbol{s} \geq \boldsymbol{0}. \end{array}\right. \tag{$\mathcal{D}$}$$

Proposition 2.2.1 (Conditions d'optimalité) *Si* $(\boldsymbol{y}^*, \boldsymbol{s}^*)$ *est une solution de* $(\mathcal{D})$, *il existe* $\boldsymbol{x}^* \in \mathbb{R}^n$ *tel que* $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{s}^*)$ *vérifie les conditions de KKT.*

L'intérêt de formuler le problème dual réside dans le lien intrinsèque entre ces deux problèmes comme suit.

Théorème 2.2.2 *On considère les problèmes* $(\mathcal{P})$ *et* $(\mathcal{D})$.

1. *Soit les deux problèmes sont irréalisables;*
2. *Soit* $(\mathcal{P})$ *est irréalisable et* $(\mathcal{D})$ *est non borné;*
3. *Soit* $(\mathcal{D})$ *est irréalisable et* $(\mathcal{P})$ *est non borné;*
4. *Soit les deux problèmes sont réalisables.*

La preuve de ce résultat repose sur le résultat suivant d'algèbre linéaire.

Théorème 2.2.3 (Théorème des alternatives) *Soient* $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *et* $\boldsymbol{b} \in \mathbb{R}^m$. *Alors, les alternatives suivantes sont mutuellement exclusives :*

- *Soit* $\exists \boldsymbol{x}, \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$, *soit* $\exists \boldsymbol{y}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} \leq \boldsymbol{0}, \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y} > 0$.

- *Soit* $\exists \boldsymbol{x}, \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}, \boldsymbol{x} \gneqq \boldsymbol{0}$, *soit* $\exists \boldsymbol{y}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} > \boldsymbol{0}$.
- *Soit* $\exists \boldsymbol{x}, \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}, \boldsymbol{x} > \boldsymbol{0}$, *soit* $\exists \boldsymbol{y}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} \gneqq \boldsymbol{0}$.

En plus de résultats sur le caractère réalisable ou non borné des problèmes, on peut aussi lier les solutions de chaque problème.

Théorème 2.2.4 (Dualité faible) *Pour tout* $(\boldsymbol{y}, \boldsymbol{s})$ *admissible pour* $(\mathcal{D})$ *et tout* $\boldsymbol{x}$ *admissible pour* $(\mathcal{P})$, *on a* $\boldsymbol{b}^{\mathrm{T}}\boldsymbol{y} \leq \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$.

Théorème 2.2.5 (Dualité forte) *Si les deux problèmes sont réalisables, alors il existe* $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{s}^*)$ *tel que :*

- $\boldsymbol{x}^*$ *est solution de* $(\mathcal{P})$;
- $(\boldsymbol{y}^*, \boldsymbol{s}^*)$ *est solution de* $(\mathcal{D})$;
- $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{s}^*)$ *vérifie* (KKT);
- $\boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}^* = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^*$.

Pour résoudre un problème linéaire, on peut donc chercher à la fois une solution primale de $(\mathcal{P})$ ainsi qu'une solution duale de $(\mathcal{D})$ en résolvant (KKT-Lin).

2.2.2 Analyse de sensibilité

On s'intéresse maintenant à un usage *a posteriori* des variables duales. Celles-ci permettent en effet de décrire la sensibilité de la valeur optimale d'un problème relativement à une contrainte, en décrivant l'effet d'une perturbation du second membre de ces contraintes.

Théorème 2.2.6 *Soit* y *une variable dual associée à une contrainte d'un programme linéaire. Alors, pour tout* Δ *suffisamment petit en valeur absolue, un changement de second membre dans la contrainte de* Δ *induit un changement de valeur optimale de* $y \cdot \Delta$.

Ainsi, la valeur absolue d'un multiplicateur permet de juger de la sensibilité du problème par rapport à une contrainte. Dans le cas où $y = 0$, par exemple, on sait que la solution ne changera pas si l'on perturbe un peu la contrainte : cela signifie en général que cette contrainte n'est pas critique pour calculer la solution.

Remarque 2.2.2 *En règle générale, l'intervalle de valeurs* Δ *pour lesquelles le résultat du théorème 2.2.6 s'applique n'est pas connu. Cependant, dans le cadre de la programmation linéaire et notamment des algorithmes de type simplexe, il est possible de quantifier ce* Δ.

Une interprétation plus générale de la sensibilité est fournie ci-dessous. On se place comme dans la section précédente dans le cadre d'un programme linéaire en forme standard.

Définition 2.2.2 (Problèmes perturbés) *Partant du problème* $(\mathcal{P})$ *(sous l'hypothèse 2.2.1), on définit pour tous* $\boldsymbol{u} \in \mathbb{R}^m$ *et* $\boldsymbol{v} \in \mathbb{R}^n$*:*

$$p(\boldsymbol{u}, \boldsymbol{v}) := \min_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \middle| \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} + \boldsymbol{u}, \boldsymbol{x} \geq \text{`}\boldsymbol{v} \right\}. \tag{2.2.1}$$

On notera que la valeur $p(\boldsymbol{0}, \boldsymbol{0})$ *correspond à la valeur optimale du problème de départ.*

Théorème 2.2.7 *Si $(\boldsymbol{y}^*, \boldsymbol{s}^*)$ sont les variables duales du problème de départ, alors pour tous $\boldsymbol{u}$ et $\boldsymbol{v}$, on a*

$$p(\boldsymbol{u}, \boldsymbol{v}) \geq p(0,0) + (\boldsymbol{y}^*)^{\mathrm{T}}\boldsymbol{u} + (\boldsymbol{s}^*)^{\mathrm{T}}\boldsymbol{v}. \tag{2.2.2}$$

En finance, le résultat du théorème 2.2.7 s'interprète comme le fait que chaque variable duale (ou multiplicateur) représente un prix d'équilibre pour la ressource représentée par la contrainte. Cela permet d'identifier les effets positifs ou négatifs d'agir sur la contrainte. A titre d'exemple, notons que si $s_i^* \gg 1$ et $v_i > 0$, alors l'équation 2.2.2 indique que la valeur optimale du problème augmentera si l'on requier $x_i \geq v_i$ au lieu de $x_i \geq 0$, ce qui semble logique car on réduit ici l'ensemble des contraintes.

2.3 Algorithmes et logiciels de programmation linéaire

Comme on l'a vu dans la partie précédente, les conditions d'optimalité (dites de KKT) d'un programme linéaire permettent de caractériser ses solutions si elles existent. Cependant, déterminer les solutions du problème directement depuis les conditions de KKT s'avère délicat, car celles-ci forment un système d'équations non linéaires.

On décrit ci-après le fonctionnement de ces méthodes dans le cas d'un programme linéaire en forme standard, et on supposera que ce problème et son dual sont tous deux réalisables. Les conditions d'optimalité pour ce programme, ou de KKT s'écrivent

$$\left\{\begin{array}{lcll} \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{s} & = & \boldsymbol{c} & \\ \boldsymbol{A}\boldsymbol{x} & = & \boldsymbol{b} & \\ \boldsymbol{x} & \geq & \boldsymbol{0} & \\ \boldsymbol{s} & \geq & \boldsymbol{0} & \\ x_i s_i & = & 0 & \forall i = 1, \ldots, n. \end{array}\right. \tag{2.3.1}$$

On décrit ci-après les deux grandes catégories de méthodes qui visent à résoudre ces conditions d'optimalité de manière itérative.

2.3.1 Algorithmes du simplexe

La méthode du simplexe classique revient à une suite $\{\boldsymbol{x}_k, \boldsymbol{y}_k, \boldsymbol{s}_k\}$ vérifiant les conditions de KKT sauf $\boldsymbol{s} \geq \boldsymbol{0}$, et vise à satisfaire cette condition asymptotiquement. La variante duale du simplexe cherche à satisfaire $\boldsymbol{x} \geq \boldsymbol{0}$.

L'idée principale de l'algorithme du simplexe est d'exploiter le fait que l'ensemble réalisable d'un programme linéaire (réalisable) est un polyèdre, et que la solution du problème se trouve nécessairement sur un sommet de ce polyèdre. En ce sommet, toutes les contraintes d'inégalité ne sont généralement pas actives : dans le cadre d'un problème en forme standard ($\mathcal{P}$), cela signifie que certaines coordonnées de la solution seront nulles (contrainte $x_i \geq 0$ active) tandis que d'autres ne le seront pas. Le théorème ci-dessous garantit qu'identifier ces contraintes actives est suffisant pour déterminer la solution.

Théorème 2.3.1 *On considère le problème ($\mathcal{P}$) et son dual sous l'hypothèse 2.2.1. Alors, il existe une partition $\mathcal{B} \cup \mathcal{N} = \{1, \ldots, n\}$ telle que*

i) $|\mathcal{B}| = m$;

ii) la matrice $\boldsymbol{A}_{\mathcal{B}} := [A_{ij}]_{\substack{1\leq i\leq m \\ j\in\mathcal{B}}}$ *est inversible;*

iii) la solution du problème $\boldsymbol{x}^*$ *se décompose selon la partition en*

$$\boldsymbol{x}^*_{\mathcal{B}} = [x_j]_{j\in\mathcal{B}} = -\boldsymbol{A}_{\mathcal{B}}^{-1}\boldsymbol{b}$$

et $\boldsymbol{x}^*_{\mathcal{N}} = \boldsymbol{0}$.

Dans ce cas, une solution duale du problème est donnée par $\boldsymbol{y}^* = \left[\boldsymbol{A}_{\mathcal{B}}^{\mathrm{T}}\right]^{-1}\boldsymbol{c}_{\mathcal{B}}$.

L'ensemble $\mathcal{B}$, qui caractérise un ensemble de contraintes, est appelé une **base** du problème.

L'algorithme du simplexe décrit une manière algorithmique de parcourir l'ensemble des sommets du polyèdre en recherchant la base dans laquelle la solution optimale doit être exprimée. Ce processus termine nécessairement après un nombre fini d'itérations, mais celui-ci peut être très grand si le nombre de contraintes est important.

Principe général L'algorithme commence par déterminer une base $\mathcal{B}$ telle que les propriétés i) et ii) du théorème 2.3.1 soient vérifiées. Si le vecteur $\boldsymbol{x}$ associé vérifie $\boldsymbol{x} \geq \boldsymbol{0}$, on parle alors de solution basique (même s'il ne s'agit pas forcément de la solution optimale, il s'agit d'un point réalisable). Il s'agit ensuite de regarder le vecteur

$$\bar{\boldsymbol{c}} = \boldsymbol{c} - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}_{\mathcal{B}}^{-\mathrm{T}}\boldsymbol{c}_{\mathcal{B}},$$

qui correspond aux variables duales $\boldsymbol{s}$ dans les conditions de KKT. Si les coordonnées de ce vecteur sont positives ou nulles, alors on a obtenu une solution optimale. Sinon, cela signifie qu'il existe un indice de colonne $j \notin \mathcal{B}$ tel que $\bar{\boldsymbol{c}}_j < 0$ et

$$\boldsymbol{c}^{\mathrm{T}}\left(\boldsymbol{x} - \alpha\boldsymbol{A}_{\mathcal{B}}^{-1}\boldsymbol{a}_j\right) < \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$$

pour tout $\alpha > 0$. Cette propriété signifie que l'on ne dispose pas de la bonne base $\mathcal{B}$, et qu'il en existe une meilleure contenant j : l'algorithme du simplexe explique comment échanger j avec un élément de la base.

Variante duale Le principe de méthode du simplexe décrit plus haut est un principe **primal**, qui se base essentiellement sur le calcul d'une solution primale $\boldsymbol{x} \in \mathbb{R}^n$. Il existe également des variantes duales, dans lesquelles on calcule $(\boldsymbol{y}, \boldsymbol{s})$ tels que $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{0}$, $\boldsymbol{s} \geq \boldsymbol{0}$, $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ et $x_i s_i = 0$ pour tout $i = 1, \ldots, n$.

Implémentation L'algorithme du simplexe est notamment utilisé dans l'extension *Solver* de Microsoft Excel. Ce solveur fournit une analyse de sensibilité détaillée, avec notamment un intervalle (plus ou moins interprétable) de valeurs dans lequel l'approximation via les multiplicateurs décrite dans le théorème 2.2.6 est valide. Le solveur de LibreOffice/OpenOffice Calc est également un solveur de type simplexe, ce qui se reconnaît notamment par rapport à la saturation des contraintes en la valeur renvoyée. De même, *Google Solve* et *OpenSolver* (accessibles dans Google Sheets) se basent sur le même genre de techniques, sans pour autant fournir de marges de validité des multiplicateurs. Enfin, le solveur *HIGHS*, qui est maintenant l'algorithme par défaut de simplexe dans la bibliothèque SciPy, est lui aussi basé sur une approche de simplexe.

2.3.2 Algorithmes de points intérieurs

La méthode du simplexe repose sur le fait de parcourir les sommets de l'ensemble des contraintes d'un programme linéaire, qui (lorsqu'il est non vide) est un polyèdre. De par la structure même du programme linéaire, on sait que la solution se trouve sur un sommet, et donc qu'il existe un certain nombre de contraintes d'inégalité satisfaites avec égalité en la solution (c'est le sens du Théorème 2.3.1).

Le principe des méthodes de points intérieurs, comme leur nom l'indique, consiste à passer par l'intérieur de l'ensemble réalisable de sorte à arriver plus vite au sommet solution qu'en parcourant l'ensemble des sommets.

Principe Un algorithme de points intérieurs part des conditions de KKT (KKT-Lin) et considère des points dans l'intérieur de l'ensemble réalisable, c'est-à-dire des triplets de la forme $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s})$ tels que

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c}, \quad \boldsymbol{x} > \boldsymbol{0}, \boldsymbol{y} > \boldsymbol{0}.$$

Un tel triplet est dit **primal-dual admissible**, car il vérifie les contraintes du problème primal et celles du problème dual. En revanche, il ne vérifie pas la dernière condition d'optimalité de (KKT-Lin), car $x_i s_i > 0$ pour tout i. On pose alors

$$\mu = \sum_{i=1}^{n} \frac{x_i s_i}{n}, \tag{2.3.2}$$

qui représente une certaine distance entre le point calculé et une solution des conditions de KKT. Le but d'une méthode de points intérieurs est de calculer une suite de triplets telle que la valeur de μ tende vers 0, et ainsi de devenir de plus en plus proche d'un point optimal. En pratique, on peut calculer pour tout $\mu > 0$ une solution vérifiant les conditions souhaitées en trouvant $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s})$ tels que $x_i > 0$ et $s_i > 0$ pour tout i, et le triplet est solution du système linéaire

$$\begin{aligned} \boldsymbol{A}\boldsymbol{x} &= \boldsymbol{b} \\ \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{s} &= \boldsymbol{c} \\ \boldsymbol{X}\boldsymbol{S}\boldsymbol{e} &= \mu\boldsymbol{e}, \end{aligned} \tag{2.3.3}$$

où $\boldsymbol{X}$ et $\boldsymbol{S}$ sont deux matrices diagonales donc les coefficients diagonaux sont ceux des vecteurs $\boldsymbol{x}$ et $\boldsymbol{s}$, respectivement, et $\boldsymbol{e} \in \mathbb{R}^n$ est le vecteur dont toutes les coordonnées valent 1. On peut ainsi calculer ce type de triplet pour des valeurs décroissantes de μ : c'est ce que font les approches de points intérieures dites primales-duales.

La théorie des points intérieurs est extrêment riche en résultats, en particulier pour des programmes linéaires. Le succès originel des points intérieurs réside dans l'obtention de garanties de convergence en temps polynomial vers une solution approchée du problème (qui dépend de la valeur du paramètre μ défini en (2.3.2)). Ce type de résultat contraste avec ceux existants pour le simplexe, où le temps de calcul peut être exponentiel pour atteindre la solution.

Implémentation `CVX` (MATLAB) et sa variante `CVXPy` en Python est à la fois un langage de modélisation et un outil de résolution basé sur les points intérieurs. Il permet de toujours renvoyer les variables duales d'un problème. Cette librairie permet notamment l'interface avec les meilleurs solveurs commerciaux du marché que sont `Gurobi`, `CPLEX` et `Mosek`. Ceux-ci sont basés sur une approche mixte simplexe/points intérieurs en ce qui concerne la programmation linéaire.

2.4 Application : arbitrage et programmation linéaire

Les problèmes d'arbitrage se posent lorsque l'on cherche à échanger différentes devises. On parle d'**opportunités d'arbitrage** lorsqu'il est possible de réaliser un bénéfice via un cycle d'échanges de devises.

Exemple 2.4.1 *Supposons que l'euro s'échange à 0,9 dollar, et que le dollar s'échange à 1,22 euro. Dans ce cas, pour tout euro échangé en dollar et à nouveau en euro, on obtiendrait 1,098 euro, soit un bénéfice.*

Dans cette section, on s'intéresse à la détection d'opportunités d'arbitrage lorsque l'on considère un ensemble arbitraire de devises.

2.4.1 Formulation en programme linéaire

On considère que l'on dispose d'un tableau $\boldsymbol{A} = [a_{ij}]_{\substack{1\leq i\leq n\\ 1\leq j\leq n}}$ représentant des taux d'échange entre n monnaies. Une unité de monnaie i s'échange ainsi contre $a_{ij} > 0$ unités de monnaie j. On posera $a_{ii} = 1$ pour tout $i = 1, \dots, n$ de sorte à éviter les arbitrages triviaux, et on fait l'hypothèse simplificatrice que le coût de conversion entre devises est nul.

On choisit les quantités suivantes comme variables du problème :

- x_{ij} : quantité de devise i échangée en devise j, que l'on souhaitera toujours positive ou nulle (on s'attend notamment à ce que $x_{ii} = 0$);
- y_k : quantité de devise k obtenue après tous les échanges, définie par

$$y_k = \sum_{i=1}^{n} a_{ik}x_{ik} - \sum_{j=1}^{n} x_{kj},$$

où la première somme correspond à la quantité de devise k obtenue en échangeant d'autres monnaies, et la seconde correspond à la quantité de devise k échangée contre d'autres monnaies. On souhaitera toujours avoir $y_k \geq 0$.

Lorsque $y_k > 0$ pour une devise k donnée, on a une opportunité d'arbitrage. Notre objectif consiste donc à définir un ensemble de transactions (c'est-à-dire de valeurs pour $\{x_{ij}\}_{ij}$ et, par là-même, de valeurs pour $\{y_k\}_k$) telles que l'une des quantités de devise après transaction soit positive. Dans ce qui suit, on se concentrera sur l'existence d'un arbitrage pour la devise 1, et on cherchera donc $\{x_{ij}\}_{ij}$ tels que $y_1 > 0$.

Une première modélisation consiste à maximiser la valeur de y_1, ce qui conduit au problème

$$\begin{cases} \text{maximiser}_{\{x_{ij}\},\{y_k\}} & y_1 & & \\ \text{s.c.} & y_k = \sum_{i=1}^{n} a_{ik}x_{ik} - \sum_{j=1}^{n} x_{kj} & k = 1, \dots, n & \\ & y_k & \geq 0 & k = 1, \dots, n \\ & x_{ij} & \geq 0 & i, j = 1, \dots, n. \end{cases} \tag{2.4.1}$$

Si ce problème est un programme linéaire presque en forme standard (il faudrait minimiser plutôt que maximiser, ce que l'on peut faire en minimisant $-y_1$), il n'est pas pour autant bien posé. En effet, s'il existe effectivement une opportunité d'arbitrage pour la devise 1, alors il existe un point

admissible à ce problème avec $y_1 > 0$. Il en existe même une infinité, et pour tout choix des $\{x_{ij}\}$ qui conduit à un bénéfice de y_1, alors le choix $\{\lambda x_{ij}\}$ où $\lambda > 1$ conduit à $\lambda y_1 > y_1$. Par conséquent, le problème est non borné. Un remède à cela consiste simplement à imposer une borne sur y_1, qui sera une valeur cible pour l'arbitrage. En choisissant cette valeur cible comme étant égale à 1, on obtient

$$\left\{\begin{array}{lll} \text{maximiser}_{\{x_{ij}\},\{y_k\}} & y_1 & \\ \text{s.c.} & y_k = \sum_{i=1}^n a_{ik}x_{ik} - \sum_{j=1}^n x_{kj} \quad k = 1,\dots,n & \\ & y_1 \leq 1 & \\ & y_k \geq 0 \quad k = 1,\dots,n & \\ & x_{ij} \geq 0 \quad i,j = 1,\dots,n. & \end{array}\right. \tag{2.4.2}$$

La résolution du programme linéaire (2.4.2) conduit aux deux cas suivants : soit il n'y a pas d'arbitrage possible, auquel cas la valeur optimale du problème est 0, et $x_{ij} = y_k = 0$ pour tous i, j, k, soit il y a un arbitrage possible, et on aura alors $y_1 = 1$.

2.4.2 Théorème fondamental de l'arbitrage

La notion d'arbitrage peut s'exprimer de manière plus abstraite, mais toujours en lien avec la programmation linéaire. Soit un système composé de m actifs durant une période donnée de $T = 0$ à $T = 1$. On définit le vecteur

$$\boldsymbol{s}_0 = \begin{bmatrix} s_0^1 \\ \vdots \\ s_0^m \end{bmatrix} \in \mathbb{R}^m,$$

qui représente les valeurs des actifs à l'instant $T = 0$ (on considère les valeurs comme pouvant être positives ou négatives). À l'instant $T = 1$, le système peut être dans un état $\omega \in \{\omega_1, \dots, \omega_n\}$, qui définit un nouveau vecteur de valeurs pour les actifs. On pose ainsi

$$\forall j = 1,\dots,n, \qquad \boldsymbol{s}_1(\omega_j) = \begin{bmatrix} s_1^1(\omega_j) \\ \vdots \\ s_1^m(\omega_j) \end{bmatrix} \in \mathbb{R}^m.$$

Une **opportunité d'arbitrage** dans ce contexte est un vecteur $\boldsymbol{y} \in \mathbb{R}^m$ tel que

$$\left\{\begin{array}{lll} \boldsymbol{s}_0^{\mathrm{T}}\boldsymbol{y} & \leq & \boldsymbol{0} \\ \boldsymbol{s}_1(\omega_j)^{\mathrm{T}}\boldsymbol{y} & \geq & \boldsymbol{0} \quad j = 1,\dots,n \\ \boldsymbol{s}_0^{\mathrm{T}}\boldsymbol{y} \neq \boldsymbol{0} & \text{ou} & \boldsymbol{s}_1(\omega_j)^{\mathrm{T}}\boldsymbol{y} \; j = 1,\dots,n. \end{array}\right. \tag{2.4.3}$$

L'existence ou l'absence d'un tel arbitrage est établie par le résultat suivant, appelé en anglais *asset pricing*.

Théorème 2.4.1 (Théorème fondamental de l'arbitrage) *On considère les vecteurs $\boldsymbol{s}_0$, $\boldsymbol{s}_1(\omega_1)$,..., $\boldsymbol{s}_1(\omega_n)$ définis ci-dessus. Alors,*

i) Soit il existe un arbitrage, c'est-à-dire un vecteur $\boldsymbol{y} \in \mathbb{R}^m$ vérifiant les conditions (2.4.3);

ii) Soit il existe $\boldsymbol{x} \in \mathbb{R}^n$ tel que

$$\boldsymbol{s}_0 = \sum_{j=1}^n x_j \boldsymbol{s}_1(\omega_j) \quad \textit{avec} \quad x_j > 0 \forall j = 1,\dots,n. \tag{2.4.4}$$

Le résultat du théorème 2.4.1 peut s'écrire comme un théorème des alternatives similaire au théorème 2.2.3 de la manière suivante :

i) Soit il existe $\boldsymbol{x} \in \mathbb{R}^n$ solution du système

$$\boldsymbol{S}\boldsymbol{x} = \boldsymbol{s}_0, \quad \boldsymbol{S} = [\boldsymbol{s}_1(\omega_1) \ \cdots \ \boldsymbol{s}_1(\omega_n)],$$

avec $\boldsymbol{x} > \boldsymbol{0}$;

ii) Soit il existe $\boldsymbol{y} \in \mathbb{R}^m$ tel que

$$\boldsymbol{0} \neq [\boldsymbol{S} - \boldsymbol{s}_0]^{\mathrm{T}} \boldsymbol{y} \geq \boldsymbol{0}.$$

Ce type de condition n'est pas un problème d'optimisation linéaire, mais est en revanche considéré comme un programme linéaire (résoudre un système d'inégalités et d'égalités linéaires). De fait, en introduisant une fonction objectif constante, il serait possible d'écrire les conditions ci-dessus comme contraintes d'un programme linéaire.

2.5 Conclusion du chapitre 2

La programmation linéaire permet de modéliser un grand nombre de problèmes, parfois grâce à des hypothèses simplificatrices (que l'on cherchera à relâcher dans la suite grâce à des formulations plus complexes). Un des grands avantages des programmes linéaires réside dans leur facilité de résolution, que ce soit par l'algorithme du simplexe ou les approches de points intérieurs. Par ailleurs, l'analyse de ces problèmes au moyen de la théorie de la dualité et des multiplicateurs de Lagrange (ou coûts fictifs/réduits) permet à la fois de caractériser les solutions et d'analyser la sensibilité d'une solution aux différentes contraintes. Ce procédé se révèle essentiel pour ne pas avoir à résoudre un problème de manière répétée.

2.6 Exercices

Exercice 1 : Gestion actifs-passifs

Dans cet exercice, on s'intéresse aux problèmes dits de gestion actifs-passifs[1], que l'on peut modéliser par des programmes linéaires en supposant que le seul risque inhérent au problème réside dans les taux d'intérêt. On souhaite construire un portefeuille qui génère suffisamment de liquidités pour couvrir des besoins (passifs) sur une période de m années consécutives. On note $\ell_i > 0$ le besoin en liquidités pour l'année i.

On suppose que le portefeuille peut être construit à partir de n obligations ayant chacune un prix unitaire et qui génèrent un flot de liquidités sur la période des m années. On notera F_{ij} le flot de liquidités généré par l'obligation j en l'an i.

Année	1	2	$\cdots$	m	
Obligation					Prix unitaire
B_1	F_{11}	F_{21}	$\cdots$	F_{m1}	p_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
B_n	F_{1n}	F_{2n}	$\cdots$	F_{mn}	p_n

[1] Ou *asset and liability management (ALM)* en anglais.

On souhaite déterminer le portefeuille de coût minimal parmi ceux pouvant couvrir les besoins en liquidités sur les m années. On supposera qu'on construit le portefeuille à partir de quantités d'obligations non négatives.

Formuler le problème comme un programme linéaire selon les trois manières suivantes :

a) en supposant que le flot de liquidités à l'année i doit a minima couvrir les besoins sur l'année ℓ_i, et en utilisant des contraintes d'inégalité linéaires. Justifier le choix des inégalités linéaires par rapport aux égalités linéaires.

b) en supposant que le flot de liquidités à l'année i doit a minima couvrir les besoins sur l'année ℓ_i, et en mettant le problème sous forme standard;

c) en supposant qu'un surplus de liquidités peut être reportés d'une année sur l'autre, et en restant en forme standard.

2.6.1 Solution de l'exercice 1

Les variables du problème, concaténées dans un vecteur $\boldsymbol{x} \in \mathbb{R}^n$, représentent les quantités des différentes obligations. L'objectif consiste à minimiser le coût du portefeuille qui est donné par

$$\boldsymbol{p}^{\mathrm{T}}\boldsymbol{x} = \sum_{i=1}^{n} p_i x_i,$$

où $\boldsymbol{p}$ est le vecteur de prix qui concatène $p_1, \dots, p_n$. En termes de contrainte, on sait d'après l'énoncé que l'on cherche à avoir $x_i \geq 0$ pour tout $i = 1, \dots, n$.

a) Pour cette première modélisation, on cherche à couvrir les besoins en liquidité sur chaque année uniquement avec les liquidités obtenues durant cette période. Pour chaque année i, cela s'exprime par la contrainte

$$\sum_{j=1}^{n} F_{ij} x_j \geq \ell_i.$$

Il est à noter que formuler cette contrainte comme une contrainte d'égalité linéaire n'est pas souhaitable, car cela pourrait conduire à ce que le problème soit irréalisable. Au final, on obtient le problème

$$\begin{cases} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \boldsymbol{p}^{\mathrm{T}}\boldsymbol{x} & & \\ \text{s.c.} & \boldsymbol{F}^{\mathrm{T}}\boldsymbol{x} & \geq \boldsymbol{\ell}, & i = 1, \dots, m \\ & \boldsymbol{x} & \geq \boldsymbol{0}, & \end{cases} \tag{2.6.1}$$

où $\boldsymbol{F} \in \mathbb{R}^{n\times m}$ est la matrice des F_{ij} (flots de liquidités) et $\boldsymbol{\ell} = [\ell_i] \in \mathbb{R}^n$ est le vecteur des besoins.

b) Pour mettre le problème en forme standard, on transforme les contraintes d'inégalité en contraintes d'égalité en rajoutant des variables d'écart. On obtient ainsi

$$\begin{cases} \text{minimiser}_{\substack{\boldsymbol{x}\in\mathbb{R}^n \\ \boldsymbol{s}\in\mathbb{R}^m}} & \boldsymbol{p}^{\mathrm{T}}\boldsymbol{x} & \\ \text{s.c.} & \boldsymbol{F}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{s} & = \boldsymbol{\ell} \\ & \boldsymbol{x} & \geq \boldsymbol{0} \\ & \boldsymbol{s} & \geq \boldsymbol{0}. \end{cases} \tag{2.6.2}$$

c) Les contraintes $\boldsymbol{F}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{s} = \boldsymbol{\ell}$ formulées en question b) modélisent de fait la présence éventuelle d'un surplus. En effet, le surplus de liquidités en l'an j est donné par $\sum_{i=1}^{n} F_{ij}x_i - \ell_j = s_j$. Par conséquent, pour modéliser le fait que l'on puisse reporter un surplus d'une année sur l'autre, on remplace la contrainte

$$\sum_{i=1}^{n} F_{ij}x_i - s_j = \ell_j$$

par

$$\sum_{i=1}^{n} F_{ij}x_i - s_j + s_{j-1} = \ell_j$$

lorsque $j \geq 2$ (on ne peut pas effectuer de report la première année).

Exercice 2 : Contraintes de levier

Dans les exemples considérés dans ce chapitre, on a considéré pour simplifier que l'on n'effectue que des investissements positifs *(long-only positions)*. Cependant, en pratique, on autorise un certain niveau d'investissements négatifs (ou *short positions*), qui représentent une réinjection d'un investissement existant dans un autre fonds. Mathématiquement, on modélise cette liberté par l'introduction d'une contrainte dite **de levier** dans la modélisation, de la forme :

$$\sum_{j=1}^{n} \min\{x_j, 0\} \geq -L, \tag{2.6.3}$$

où $L > 0$ est un niveau de levier fixé. Cette contrainte est non linéaire en les variables x_j du fait de l'utilisation de l'opérateur $\min$.

a) En programmation linéaire, l'inégalité (2.6.3) est en général remplacée par le système

$$\begin{cases} \sum_{j=1}^{n} z_j & \geq & -L \\ z_j \leq x_j \quad j = 1, \ldots, n \\ z_j \leq 0 \quad j = 1, \ldots, n, \end{cases} \tag{2.6.4}$$

où $\boldsymbol{z} \in \mathbb{R}^n$ est un vecteur de variables auxiliaires. Pour tout $\boldsymbol{x} \in \mathbb{R}^n$ vérifiant (2.6.3), donner un vecteur $\boldsymbol{z}$ vérifiant (2.6.4).

b) Convertir le système d'inégalités et d'égalités obtenu en question a) en forme standard [2] s'il ne l'est pas déjà.

Solution de l'exercice 2

a) On introduit le vecteur $\boldsymbol{z} \in \mathbb{R}^n$ dont les composantes sont données par $z_j = \min\{x_j, 0\}$. Ce vecteur vérifie bien le système (2.6.4) dès lors que $\boldsymbol{x}$ vérifie (2.6.3).

[2] Par abus de langage, on considèrera qu'un tel système est en forme standard si un programme linéaire avec de telles contraintes le serait.

b) Pour reformuler le système en forme standard, on introduit $2n+1$ variables auxiliaires $\boldsymbol{s} \in \mathbb{R}^{2n+1}$ qui seront contraintes d'être positives ou nulles. On obtient ainsi le système suivant :

$$\begin{cases} \sum_{j=1}^{n} z_j - s_1 & = & -L \\ z_j + s_{j+1} & = & x_j \quad j = 1, \dots, n \\ z_j + s_{j+n+1} & = & 0 \quad j = 1, \dots, n \\ \boldsymbol{s} & \geq & \mathbf{0}, \end{cases}$$

qui est bien en forme standard.

Chapitre 3

Optimisation quadratique en finance

Dans ce chapitre, on cherche à étendre les résultats du chapitre précédent au cas où la fonction objectif n'est plus linéaire mais quadratique en les variables de décision. Comme on le verra, il est possible de développer une théorie de la dualité ainsi que des algorithmes similaires à ceux utilisés pour la programmation linéaire. On s'intéressera tout particulièrement au modèle de Markowitz, qui conduit à des programmes quadratiques spécifiques.

3.1 Bases de la programmation quadratique

Définition 3.1.1 *Un programme quadratique, ou problème d'optimisation quadratique, est un problème d'optimisation de la forme*

$$\begin{array}{rl} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x}+\boldsymbol{g}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.c.} & \boldsymbol{A}\boldsymbol{x}=\boldsymbol{b} \\ & \boldsymbol{C}\boldsymbol{x}\geq\boldsymbol{d}. \end{array}$$

avec $\boldsymbol{H}\in\mathbb{R}^{n\times n}$, $\boldsymbol{g}\in\mathbb{R}^n$, $\boldsymbol{A}\in\mathbb{R}^{m\times n}$, $\boldsymbol{b}\in\mathbb{R}^m$, $\boldsymbol{C}\in\mathbb{R}^{\ell\times n}$, $\boldsymbol{d}\in\mathbb{R}^{\ell}$.

Définition 3.1.2 *Un programme linéaire en* **forme standard** *est de la forme*

$$\begin{array}{rl} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x}+\boldsymbol{g}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.c.} & \boldsymbol{A}\boldsymbol{x}=\boldsymbol{b} \\ & \boldsymbol{x}\geq\boldsymbol{0}, \end{array} \tag{3.1.1}$$

où les contraintes d'intervalle $\boldsymbol{x}\geq\boldsymbol{0}$ *peuvent porter seulement sur un sous-ensemble des coordonnées de* $\boldsymbol{x}$.

En général, une modélisation en programme quadratique est faite de sorte à satisfaire l'hypothèse ci-dessous.

Hypothèse 3.1.1 *La matrice* $\boldsymbol{H}$ *est symétrique et semi-définie positive.*

Un cas particulier de programme quadratique, très important dans la pratique, est décrit dans l'exemple ci-dessous.

Exemple 3.1.1 (Moindres carrés linéaires) *Soient* $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ *et* $\boldsymbol{b}\in\mathbb{R}^m$. *On cherche un vecteur* $\boldsymbol{x}\in\mathbb{R}^n$ *tel que* $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, *ce qui peut ou non être faisable. On considère donc les deux cas suivants.*

a) S'il n'existe pas de solution à l'équation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, on résout

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimiser}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2 = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} + \frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{b}.$$

b) S'il existe plusieurs solutions, on cherche celle de norme minimale en résolvant :

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimiser}} \frac{1}{2}\|\boldsymbol{x}\|^2 = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{I}_n\boldsymbol{x} \quad \textit{s. c.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

où $\boldsymbol{I}_n$ est la matrice identité dans $\mathbb{R}^{n\times n}$.

3.2 Résolution d'un programme quadratique

3.2.1 Résolution théorique

La théorie de la programmation quadratique est très similaire à celle de la programmation linéaire. On peut notamment formuler le dual d'un programme quadratique, et en déduire des conditions d'existence de solutions pour le problème de départ et son dual. On se contente ici de donner les conditions d'optimalité, dites de KKT (Karush-Kuhn-Tucker).

Théorème 3.2.1 *On considère le programme quadratique* (3.1.1) *sous l'hypothèse 3.1.1. Si $\boldsymbol{x}^* \in \mathbb{R}^n$ est solution du problème, alors il existe $\boldsymbol{y}^* \in \mathbb{R}^m$ et $\boldsymbol{s}^* \in \mathbb{R}^n$ tels que*

$$\left\{\begin{array}{lcll} -\boldsymbol{H}\boldsymbol{x}^* + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}^* + \boldsymbol{s}^* & = & \boldsymbol{g} & \\ \boldsymbol{A}\boldsymbol{x}^* & = & \boldsymbol{b} & \\ \boldsymbol{x}^* & \geq & \boldsymbol{0} & \\ \boldsymbol{s}^* & \geq & \boldsymbol{0} & \\ x_i^* s_i^* & = & 0 & \forall i = 1,\dots,n. \end{array}\right. \tag{KKT-Quad}$$

On notera que les conditions (KKT-Quad) sont quasiment identiques à celles pour la programmation linéaire (KKT-Lin). La seule différence réside dans la présence d'un terme en $-\boldsymbol{H}\boldsymbol{x}^*$ dans la première condition, dû au caractère quadratique de la fonction.

3.2.2 Résolution algorithmique

Comme dans le cas des programmes linéaires, la résolution des programmes quadratiques se base sur les conditions de KKT. Les techniques principales de résolution se divisent en deux catégories.

Méthodes de contraintes actives (active set) Ces méthodes peuvent être vues comme une extension de l'algorithme du simplexe à la programmation quadratique. Ces méthodes cherchent à déterminer l'ensemble des contraintes actives en la solution, c'est-à-dire les contraintes d'inégalité qui sont satisfaites avec égalité en la solution. Dans le cas d'un programme quadratique en forme standard pour lequel la solution est $\boldsymbol{x}^*$, on dira que la contrainte $x_i \geq 0$ est active en $\boldsymbol{x}^*$ si $x_i^* = 0$. Une contrainte inactive (pour laquelle $x_i^* > 0$) peut être ignorée pour déterminer la solution.

La méthode des contraintes actives utilise à chaque itération un ensemble d'indices $\mathcal{I}_k \subset \{1,\dots,n\}$, et résout le problème

$$\begin{array}{rl} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x} + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.c.} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & x_i = 0 \quad i \in \mathcal{I}_k. \end{array} \tag{3.2.1}$$

En fonction de la solution, voire de la réalisabilité du problème, le point calculé est conservé, et l'ensemble $\mathcal{I}_k$ est mis à jour en ajoutant ou supprimant un élément, comme pour le simplexe.

Remarque 3.2.1 *Les approches par contraintes actives sont notamment utilisées dans le solveur d'Excel, ainsi que dans des routines de résolution de programmation quadratique présentes dans la bibliothèque SciPy du langage Python.*

Points intérieurs Les approches de points intérieurs pour la programmation quadratique procèdent de manière identique à celles pour la programmation linéaire. Pour un problème en forme standard tel que (3.1.1), il s'agit toujours de calculer une suite $(\boldsymbol{x}_k, \boldsymbol{y}_k, \boldsymbol{s}_k)$ telle que $\boldsymbol{x}_k > \mathbf{0}$ et $\boldsymbol{s}_k > \mathbf{0}$, et de réduire la valeur de $\boldsymbol{x}_k^{\mathrm{T}} \boldsymbol{s}_k$ progressivement. La valeur des prochains itérés est toujours calculée via les deux premières conditions de KKT, exprimées comme un système linéaire en les variables.

Remarque 3.2.2 *Les solveurs de points intérieurs pour la programmation quadratique sont utilisés dans différentes bibliothèques Python, dont* SciPy *et* cvxpy.

Pour terminer cette section, on notera que de nombreux solveurs sont capables d'exploiter une structure particulière d'un programme quadratique. Par exemple, les problèmes aux moindres carrés linéaires (cf l'exemple 3.1.1) bénéficient souvent de solveurs dédiés. La librairie *cvxpy* choisit automatiquement l'algorithme adapté à la structure du problème considéré.

3.3 Application : modèle de Markowitz

3.3.1 Principe

Le modèle de Markowitz, qui valut le prix Nobel d'économie à son auteur en 1990, consiste en la construction d'un portefeuille qui réalise un compromis entre un risque réduit et un rendement moyen élevé.

Le portefeuille peut être défini parmi un ensemble de n actifs qui présentent chacun une part d'aléatoire, et donc un risque, sur une période d'investissement allant d'un instant t_0 à un instant $t > t_0$. On part ainsi de prix fixés à l'instant t_0, que l'on écrit sous la forme d'un vecteur $\boldsymbol{v}_0 = \begin{bmatrix} v_{1,0} \\ \vdots \\ v_{n,0} \end{bmatrix}$ et on cherche à définir un portefeuille en prévision de l'instant t, pour lequel les prix seront les composantes d'un vecteur aléatoire $\boldsymbol{v}_t = \begin{bmatrix} v_{1,t} \\ \vdots \\ v_{n,t} \end{bmatrix}$. La décision de la construction du portefeuille est prise à l'instant t_0, alors que $\boldsymbol{v}_t$ est inconnu. On suppose que l'on connaît néanmoins des statistiques sur le vecteur $\boldsymbol{v}_t$, de sorte à pouvoir optimiser une certaine mesure de qualité du portefeuille.

Mathématiquement, on note $\boldsymbol{h} \in \mathbb{R}^n$ les différentes quantités d'actifs qui composent le portefeuille choisi. Les valeurs du portefeuille à t_0 et à t correspondent à

$$w_0 = \boldsymbol{v}_0^{\mathrm{T}} \boldsymbol{h} \quad \text{et} \quad w = \boldsymbol{v}_t^{\mathrm{T}} \boldsymbol{h}.$$

En pratique, plutôt que de raisonner sur w et w_0, on raisonne sur les retours sur investissements, définis via la normalisation suivante :

$$r_p = \frac{w - w_0}{w_0}, \quad \boldsymbol{r} = \left[r_i = \frac{v_{i,t} - v_{i,0}}{v_{i,0}} \right]_{i=1}^{n}.$$

Par ailleurs, plutôt que de considèrer les quantités d'actifs $\boldsymbol{h}$, on raisonne sur les pourcentages d'actifs présents dans le portefeuille, que l'on concatène sous la forme d'un vecteur $\boldsymbol{x} \in \mathbb{R}^n$ avec

$$x_i = \frac{h_i v_{i,0}}{w_0} \in [0,1].$$

Par construction, on a alors $\sum_{i=1}^n x_i = 1$. De plus, lorsque l'on souhaite maximiser la valeur $w = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{h}$ du portefeuille, cela revient à maximiser

$$r_p = \frac{w - w_0}{w_0} = \frac{(\boldsymbol{v} - \boldsymbol{v}_0)^{\mathrm{T}}\boldsymbol{h}}{w_0} = \boldsymbol{r}^{\mathrm{T}}\boldsymbol{x},$$

où la dernière égalité s'obtient par

$$\frac{(\boldsymbol{v} - \boldsymbol{v}_0)^{\mathrm{T}}\boldsymbol{h}}{w_0} = \frac{\sum_{i=1}^n (v_i - v_{i,0})h_i}{w_0} = \sum_{i=1}^n \frac{(v_i - v_{i,0})}{v_{i,0}} \frac{h_i v_{i,0}}{w_0} = \sum_{i=1}^n r_i^{\mathrm{T}} x_i = \boldsymbol{r}^{\mathrm{T}}\boldsymbol{x}.$$

On se demande alors quels problèmes d'optimisation peuvent être définis si l'on ne connaît que des statistiques sur le vecteur $\boldsymbol{r}$ (et donc sur les valeurs à venir des actifs), et à quel genre de portefeuille on aboutit en résolvant ces problèmes

3.3.2 Problèmes d'optimisation

Les problèmes d'optimisation que nous allons définir se basent sur les statistiques suivantes :

- les moyennes $\{\mu_i = \mathbb{E}[r_i]\}_{i=1,\dots,n}$, ou retours sur investissement moyens (par rapport à $\boldsymbol{v}_0$);
- les variances $\{\sigma_i^2 = \mathbb{E}\left[(r_i - \mathbb{E}[r_i])^2\right]\}_{i=1,\dots,n}$;
- les covariances définies pour tous $(i,j) \in \{1,\dots,n\}$ par

$$\sigma_{ij} = \sigma_{ji} = \frac{\mathbb{E}[(r_i - \mathbb{E}[r_i])(r_j - \mathbb{E}[r_j])]}{\sigma_i \sigma_j}.$$

On notera $\boldsymbol{\mu} = [\mu_i]$ et $\boldsymbol{V} = [\sigma_{ij}]$; on a alors

$$\boldsymbol{\mu} = \mathbb{E}\left[\boldsymbol{r}^{\mathrm{T}}\boldsymbol{x}\right] \quad \text{et} \quad \boldsymbol{V} = \mathbb{E}\left[(\boldsymbol{r} - \mathbb{E}[\boldsymbol{r}])(\boldsymbol{r} - \mathbb{E}[\boldsymbol{r}])^{\mathrm{T}}\right] = \mathbb{E}\left[(\boldsymbol{r} - \boldsymbol{\mu})(\boldsymbol{r} - \boldsymbol{\mu})^{\mathrm{T}}\right].$$

On peut alors formuler les trois problèmes d'optimisation suivants :

a) **Calcul de portefeuille de risque minimal avec retour moyen sur investissement garanti** :

$$\begin{array}{rl} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x} \\ \text{s.c.} & \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x} \geq \bar{\mu} \\ & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1, \end{array} \tag{3.3.1}$$

où $\boldsymbol{e} = [1 \cdots 1]^{\mathrm{T}}$ et $\bar{\mu} \in \mathbb{R}$ représente un retour sur investissement espéré. Ce problème est un programme quadratique.

b) **Calcul de portefeuille à retour moyen sur investissement maximal pour un niveau de risque contrôlé** :

$$\begin{array}{ll} \text{maximiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.c.} & \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x} \leq \frac{\sigma^2}{2} \\ & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1, \end{array} \tag{3.3.2}$$

où σ^2 borne le niveau de risque souhaité. Ce problème possède un objectif linéaire et des contraintes dont l'une est quadratique. Il ne s'agit pas d'un programme quadratique[1]

c) **Approche de Markowitz**

$$\begin{array}{ll} \text{maximiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x} - \frac{\gamma}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x} \\ & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1, \end{array} \tag{3.3.3}$$

où $\gamma > 0$ est un paramètre de pénalité qui permet de contrôler le poids du terme en $-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x}$ dans la maximisation en l'introduisant dans l'objectif et non via une contrainte.

On notera que la dernière formulation (3.3.3) est équivalente au programme quadratique en forme standard :

$$\begin{array}{ll} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \frac{\gamma}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x} - \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x} \\ & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1. \end{array} \tag{3.3.4}$$

La formulation (3.3.4) est un outil de base de calcul de portefeuille qui prend en compte un risque. La valeur de γ représente un compromis entre le retour moyen sur investissement $\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x}$ et la volatilité $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x}$. Lorsque γ tend vers 0, on tend à minimiser $-\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x}$, et donc à maximiser uniquement le retour moyen sur investissement; à l'inverse, lorsque γ tend vers ∞, on tend à minimiser $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x}$ uniquement, et donc à rechercher un portefeuille de risque minimal (cf Section 3.5).

Plus globalement, les modèles définis plus haut peuvent être liés, comme le montre le résultat suivant.

Proposition 3.3.1 *Sous les définitions de cette section, on suppose que $\boldsymbol{V}$ est inversible et que la quantité $\bar{\mu}$ utilisée dans la formulation (3.3.4) vérifie $\bar{\mu} > \frac{\mu^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{e}}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{e}}$. Alors, si $\boldsymbol{x}^*$ est une solution de la formulation (3.3.4), il existe σ^2 et γ (qui dépendent de $\bar{\mu}$) tels que $\boldsymbol{x}^*$ est également une solution des formulations (3.3.3) et (3.3.3).*

Ce raisonnement s'applique aux trois formulations considérées, ainsi qu'à la formulation (3.3.4).

3.3.3 Analyse des modèles et de leurs solutions

Dans cette section,on fait l'hypothèse que la matrice $\boldsymbol{V}$ représentant les covariances des actifs est inversible. Il s'agit alors d'une matrice symétrique définie positive, c'est-à-dire telle que $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x} > 0$ si $\boldsymbol{x} \neq \boldsymbol{0}$. Sous cette hypothèse, on peut montrer que le problème

$$\begin{array}{ll} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x} \\ & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1 \end{array} \tag{3.3.5}$$

[1] Il s'agit cependant d'un programme conique, qui peut lui aussi être résolu par des approches de points intérieurs. En revanche, cette résolution est plus complexe que celle d'un programme quadratique.

possède une unique solution $\boldsymbol{x}^* = \frac{\boldsymbol{V}^{-1}\boldsymbol{e}}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{e}}$ qui vérifie les conditions de KKT

$$\begin{cases} \boldsymbol{V}\boldsymbol{x}^* + \boldsymbol{e}y^* &= 0 \\ \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x}^* &= 1 \end{cases}$$

pour un prix fictif $y^* = -\frac{1}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{e}}$. Le portefeuille ainsi obtenu est noté

$$\boldsymbol{x}_R = \frac{\boldsymbol{V}^{-1}\boldsymbol{e}}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{e}}$$

et s'appelle le **portefeuille de risque minimum**.

Dans le cadre du problème (3.3.4), on montre de manière similaire qu'il existe deux solutions du problème particulières :

- le vecteur $\boldsymbol{x}_R$ est solution, et minimise le risque parmi les solutions;
- le vecteur $\boldsymbol{x}_T = \frac{\boldsymbol{V}^{-1}\boldsymbol{\mu}}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{\mu}}$, appelé **portefeuille tangent** (*tangency portfolio*), maximise le retour moyen sur investissement parmi les solutions.

Le résultat suivant permet de lier ces portefeuilles particuliers aux autres solutions.

Théorème 3.3.1 (Two-fund theorem) *Toute solution du problème* (3.3.4) *avec* $\boldsymbol{V}$ *inversible est une combinaison des portefeuilles de risque minimum et tangent. Pour toute solution* $\boldsymbol{x}^*$*, il existe ainsi* $\alpha \in \mathbb{R}$ *tel que* $\boldsymbol{x}^* = \alpha\boldsymbol{x}_R + (1-\alpha)\boldsymbol{x}_T$.

3.4 Conclusion du chapitre 3

La programmation quadratique offre des possiblités de modélisation plus riches que la programmation linéaire, en pouvant prendre en compte des concepts tels que les corrélations entre couples d'actifs. Des algorithmes efficaces existent encore pour résoudre ces problèmes, notamment en se basant sur leur structure.

L'application majeure de l'optimisation quadratique dans le cadre de la finance est due aux problèmes d'optimisation de portefeuille impliquant un modèle moyenne-variance dit de Markowitz. Dans ce cadre, la programmation quadratique est un outil indispensable pour prendre en compte des aspects de risque via un terme quadratique dans l'objectif.

3.5 Exercices

Exercice 3 : Allocation à volatilité minimale

On considère un problème d'allocation de portefeuille sur trois actifs désignés par $1, 2, 3$ dont le rendement est une variable aléatoire. On définit les quantités suivantes :

- σ_{ij} est la corrélation entre les actifs i et j pour tout couple $(i,j) \in \{1,2,3\}^2$, avec $\sigma_{ii} = 1$ pour tout $i = 1, 2, 3$;
- $\boldsymbol{C} = [\sigma_{ij}] \in \mathbb{R}^{3\times 3}$ est la matrice des corrélations;
- σ_i^2 est la variance de l'actif i, et σ_i son écart-type;

- $\boldsymbol{s} \in \mathbb{R}^3$ est le vecteur des écarts-types;
- $\boldsymbol{V} \in \mathbb{R}^{3\times 3}$ est la matrice de covariance des actifs, donnée par $\boldsymbol{V} = \boldsymbol{C}\boldsymbol{s}\boldsymbol{s}^{\mathrm{T}}$.

Le risque ou la volatilité d'un portefeuille $\boldsymbol{x} \in \mathbb{R}^n$ sera mesuré via la quantité

$$\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x} = \frac{1}{2}\sum_{i=1}^{3} V_{ii}x_i^2 + \sum_{i=1}^{3}\sum_{j\neq i} V_{ij}x_i x_j.$$

Le premier terme de la somme combine les variances relatives aux investissements dans un actif, tandis que le second représente le risque dû aux corrélations entre actifs. Le but de cet exercice est de trouver un portefeuille de risque minimal.

a) Formuler le problème en tant que programme quadratique, en supposant que le portefeuille est exprimé en pourcentages (sommant à 1) et que ses composantes ne peuvent être que positives ou nulles (*long positions*).

b) La résolution numérique de ce problème vue en séance[2] donne comme solution $\boldsymbol{x}^* = \begin{bmatrix} 0.089 \\ 0 \\ 0.91 \end{bmatrix}$ et comme valeur optimale $f^* = 0.0012$. Lorsque les contraintes $\boldsymbol{x} \geq \boldsymbol{0}$ sont supprimées de la modélisation, on obtient en revanche $\boldsymbol{x}^* = \begin{bmatrix} 0.19 \\ -0.14 \\ 0.94 \end{bmatrix}$ et $f^* = 0.0009$, qui est une meilleure valeur de fonction. Pourquoi le fait d'enlever une contrainte permet-il d'obtenir une meilleure valeur de l'objectif ?

Solution de l'exercice 3

a) Le problème s'écrit

$$\begin{array}{rl} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^3} & \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x} \\ \text{s.c.} & x_1 + x_2 + x_3 = 1 \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{array}$$

Note : ce problème est en forme standard.

b) Lorsque l'on supprime la contrainte $\boldsymbol{x} \geq \boldsymbol{0}$, on obtient

$$\begin{array}{rl} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^3} & \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x} \\ \text{s.c.} & x_1 + x_2 + x_3 = 1. \end{array}$$

L'ensemble admissible de ce problème contient (strictement) celui du problème écrit en question a). Par conséquent, la valeur optimale de ce second problème est au moins aussi bonne que celle du premier problème. Comme on s'autorise plus de liberté dans le choix des x_i, on s'attend à ce que la valeur optimale soit strictement meilleure, et c'est ce que l'on observe ici numériquement.

[2] Cf les sources associées.

Exercice 4 : Ratio de Sharpe

Le ratio de Sharpe est une alternative au modèle de Markowitz qui combine le retour moyen et la volatilité en une seule quantité, sous réserve que la matrice de covariance $\boldsymbol{V}$ soit inversible. Avec les mêmes notations que celles de la section 3.3, on considère donc le problème :

$$\begin{array}{rl} \text{maximiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \frac{\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x}}} \\ \text{s.c.} & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1 \end{array} \tag{3.5.1}$$

Ce problème n'est ni un programme linéaire, ni un programme quadratique. Le but de cet exercice est néanmoins de trouver une reformulation équivalente à ce problème en un programme quadratique.

a) On considère le changement de variable $\boldsymbol{z} = \alpha\boldsymbol{x}$ avec $\alpha > 0$ et $\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{z} = 1$. Reformuler le problème (3.5.1) en un problème d'optimisation sur α et $\boldsymbol{z}$ où l'objectif est de la forme $\frac{1}{f(\boldsymbol{z})}$, où f est une fonction que l'on précisera.

b) Pour toute fonction strictement positive f, minimiser f revient à maximiser $\frac{1}{\sqrt{f}}$. Par ailleurs, le cas $\alpha = 0$ peut être incorporé au problème. En déduire alors que le problème (3.5.1) se reformule comme le programme quadratique :

$$\begin{array}{rl} \text{maximiser}_{\substack{\boldsymbol{z}\in\mathbb{R}^n \\ \alpha\in\mathbb{R}}} & \boldsymbol{z}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{z} \\ \text{s.c.} & \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{z} = 1 \\ & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{z} - \alpha = 0 \\ & \alpha \geq 0. \end{array} \tag{3.5.2}$$

c) D'après les conditions de KKT, on sait que $(\boldsymbol{z}^*, \alpha^*)$ est solution du problème (3.5.2) s'il existe $\boldsymbol{y}^* \in \mathbb{R}^2$ et $\boldsymbol{s}^* \in \mathbb{R}$ tels que

$$\begin{aligned} -2\boldsymbol{V}\boldsymbol{z}^* + y_1^*\boldsymbol{\mu} + y_2^*\boldsymbol{e} &= \boldsymbol{0} \\ -y_2^* + s^* &= 0 \\ \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{z}^* &= 1 \\ \boldsymbol{e}^{\mathrm{T}}\boldsymbol{z}^* - \alpha^* &= 0 \\ \alpha^* &\geq 0 \\ \alpha^* s^* &= 0. \end{aligned}$$

Montrer alors que le portefeuille tangent $\boldsymbol{x}^* = \frac{\boldsymbol{V}^{-1}\boldsymbol{\mu}}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{\mu}}$ définit $\boldsymbol{z}^*$ et α^* qui forment une solution du problème (3.5.2).

Solution de l'exercice 4

a) On utilise le changement de variable $\boldsymbol{z} = \alpha\boldsymbol{x}$, qui donne $\boldsymbol{x} = \frac{1}{\alpha}\boldsymbol{z}$. En remplaçant les $\boldsymbol{x}$ dans l'expression de (3.5.1), on obtient :

$$\begin{array}{rl} \text{maximiser}_{\substack{\boldsymbol{z}\in\mathbb{R}^n \\ \alpha\in\mathbb{R}}} & \frac{\frac{1}{\alpha}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{z}}{\sqrt{\frac{1}{\alpha}\boldsymbol{z}^{\mathrm{T}}\boldsymbol{V}\left(\frac{1}{\alpha}\boldsymbol{z}\right)}} \\ \text{s.c.} & \frac{1}{alpha}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{z} = 1 \\ & \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{z} = 1 \\ & \alpha > 0, \end{array}$$

où les deux dernières contraintes proviennent de la définition même de $\boldsymbol{z}$ et α. En simplifiant les α dans la valeur de l'objectif, et en exploitant la contrainte $\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{z} = 1$, on aboutit à

$$\begin{array}{rl} \text{maximiser}_{\substack{\boldsymbol{z}\in\mathbb{R}^n \\ \alpha\in\mathbb{R}}} & \frac{1}{\sqrt{\boldsymbol{z}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{z}}} \\ \text{s.c.} & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{z} - \alpha = 0 \\ & \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{z} = 1 \\ & \alpha > 0, \end{array}$$

b) D'après l'indication donnée dans la question, on sait que le problème obtenu en fin de question a) est équivalent à

$$\begin{array}{rl} \text{minimiser}_{\substack{\boldsymbol{z}\in\mathbb{R}^n \\ \alpha\in\mathbb{R}}} & \boldsymbol{z}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{z} \\ \text{s.c.} & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{z} - \alpha = 0 \\ & \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{z} = 1 \\ & \alpha > 0, \end{array}$$

Il suffit ensuite de transformer la contrainte $\alpha > 0$ en $\alpha \geq 0$ comme demandé par l'énoncé pour arriver à (3.5.2).

c) Il s'agit d'utiliser les conditions de KKT pour obtenir une formule explicite pour α^* et $\boldsymbol{z}^*$ en fonction de celle de $\boldsymbol{x}^*$. On tire tout d'abord de la contrainte $\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{z}^* = 1$ le fait que

$$\alpha^* = \frac{1}{\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x}^*} = \frac{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{\mu}}{\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{\mu}}.$$

En utilisant cette formule dans la définition de $\boldsymbol{z}^* = \alpha^*\boldsymbol{x}^*$, il vient

$$\boldsymbol{z}^* = \frac{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{\mu}}{\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{\mu}}\boldsymbol{x}^* = \frac{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{\mu}}{\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{\mu}}\frac{\boldsymbol{V}^{-1}\boldsymbol{\mu}}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{\mu}} = \frac{\boldsymbol{V}^{-1}\boldsymbol{\mu}}{\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{\mu}}.$$

Il est ensuite aisé de vérifier algébriquement que les conditions de KKT sont vérifiées pour ces valeurs.

Chapitre 4

Optimisation stochastique en finance

Dans les chapitres précédents, nous avons soit négligé l'aléatoire en nous concentrant sur le rendement en moyenne (cf chapitre 2), soit pris en compte cet aléatoire au travers de quantités statistiques (rendement moyen et volatilité, cf chapitre 3 et plus particulièrement la section 3.3).

Dans ce chapitre, on va s'intéresser à des formulations dans lesquelles l'aléatoire est présent. On présente tout d'abord un cadre général, en restant sur les formulations en deux temps, puis on décrit une technique de résolution dans le cadre de l'approche par scénarios. On parle enfin de mesures de risque, et de la façon dont elles généralisent les concepts déjà entrevus.

4.1 Formulation générale

4.1.1 Programmes stochastiques sans et avec recours

On considère une fenêtre de temps $[0, 1]$. On suppose que l'on doit prendre une décision au temps $t = 0$ sous la forme d'un vecteur $\boldsymbol{x} \in \mathbb{R}^n$, mais que la qualité de cette décision est évaluée en fonction d'une variable aléatoire ω, dont la valeur ne sera révélée qu'au temps $t = 1$. Un programme stochastique classique vise à prendre la meilleure solution en moyenne.

Définition 4.1.1 (Programme stochastique) *Un* **programme stochastique** *est un problème d'optimisation de la forme*

$$\begin{array}{rl} \textit{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \mathbb{E}_\omega\left[F(\boldsymbol{x},\omega)\right] \\ \textit{s.c.} & \boldsymbol{x} \in \mathcal{X}, \end{array} \tag{4.1.1}$$

où $\omega \in \Omega$ est une variable aléatoire, $F : \mathbb{R}^n \times \Omega \to \mathbb{R}$ est une fonction et $\mathcal{X} \subset \mathbb{R}^n$.

Un exemple de fonction F est $\boldsymbol{r}^{\mathrm{T}}\boldsymbol{x}$, où $\omega = \boldsymbol{r}$ est un vecteur de rendement aléatoire. Dans ce cas, l'objectif devient de maximiser le rendement moyen.

Il est fréquent qu'une partie de la décision prise à $t = 0$ ne dépende pas de la variable aléatoire. Cela donne lieu à une modélisation particulière du problème d'optimisation, décrite ci-dessous.

Définition 4.1.2 *Un* **programme stochastique avec recours** *est de la forme*

$$\begin{array}{rl} \textit{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & f(\boldsymbol{x}) + \mathbb{E}_\omega\left[Q(\boldsymbol{x},\omega)\right] \\ \textit{s.c.} & \boldsymbol{x} \in \mathcal{X}, \end{array} \tag{4.1.2}$$

où $f : \mathbb{R}^n \to \mathbb{R}$ est une fonction déterministe et $Q : \mathbb{R}^n \times \Omega \to \mathbb{R}$ est une fonction dite de recours donnée par la solution d'un problème d'optimisation :

$$Q(vx, \omega) := \begin{array}{ll} \min_{\boldsymbol{y}(\omega) \in \mathbb{R}^m} & g(\boldsymbol{y}(\omega), \omega) \\ \textit{s. c.} & \boldsymbol{y}(\omega) \in \mathcal{Y}(\boldsymbol{x}, \omega). \end{array} \tag{4.1.3}$$

Le vecteur $\boldsymbol{y}(\omega)$ représente la décision prise au temps $t = 1$, qui dépend de la valeur de la variable aléatoire ω. Cette décision vise à minimiser une fonction de coût $g : \mathbb{R}^m \times \Omega \to \mathbb{R}$ sur un ensemble $\mathcal{Y}(\boldsymbol{x}, \omega) \subset \mathbb{R}^m$. Cet ensemble contient toute dépendance de $\boldsymbol{y}(\omega)$ vis-à-vis de $\boldsymbol{x}$.

Dans le langage de la programmation stochastique, le problème (4.1.2) s'appelle également un **programme stochastique en deux temps** (*two-phase stochastic programming*). La décision de recours (ou dans un second temps) $\boldsymbol{y}(\omega)$ est fixe dès lors que la variable ω a été observée et que $\boldsymbol{x}$ est fixé.

Remarque 4.1.1 *Les notions présentées dans ce chapitre se généralisent sur des temps multiples (on parle en anglais de* multistage stochastic programming*). Dans ce cours, on se contentera du cas plus simple de la programmation en deux temps, qui illustre déjà les défis de modélisation.*

4.1.2 Programme linéaire stochastique en deux temps

Définition 4.1.3 *Un* **programme linéaire stochastique en deux temps** *est donné par*

$$\begin{array}{rll} \textit{minimiser}_{\boldsymbol{x} \in \mathbb{R}^n} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} & + \mathbb{E}_\omega \left[Q(\boldsymbol{x}, \omega) \right] \\ \textit{s.c.} & \boldsymbol{A}\boldsymbol{x} & = \boldsymbol{b} \\ & \boldsymbol{x} & \geq \boldsymbol{0}, \end{array} \tag{4.1.4}$$

où les contraintes d'intervalle $\boldsymbol{x} \geq \boldsymbol{0}$ peuvent porter seulement sur un sous-ensemble des coordonnées de $\boldsymbol{x}$. La fonction Q est définie comme

$$Q(vx, \omega) := \begin{array}{lll} \min_{\boldsymbol{y}(\omega) \in \mathbb{R}^m} & \boldsymbol{q}(\omega)^{\mathrm{T}}\boldsymbol{y}(\omega) & \\ \textit{s. c.} & \boldsymbol{T}(\omega)\boldsymbol{x} + \boldsymbol{N}(\omega)\boldsymbol{y}(\omega) & = \boldsymbol{h}(\omega) \\ & \boldsymbol{y}(\omega) & \geq \boldsymbol{0}, \end{array} \tag{4.1.5}$$

où $\boldsymbol{q}(\omega) \in \mathbb{R}^m$, $\boldsymbol{T}(\omega) \in \mathbb{R}^{\ell \times n}$, $\boldsymbol{N}(\omega) \in \mathbb{R}^{\ell \times m}$, $\boldsymbol{h}(\omega) \in \mathbb{R}^\ell$, et les contraintes d'intervalle $\boldsymbol{y}(\omega) \geq \boldsymbol{0}$ peuvent porter seulement sur un sous-ensemble des coordonnées de $\boldsymbol{y}(\omega)$.

On notera qu'en l'absence de recours, le problème serait un programme linéaire en forme standard. Ici, la fonction objectif dépend d'un second programme linéaire (lui aussi en forme standard). Cependant, on observera que les matrices $\boldsymbol{T}(\omega)$ et $\boldsymbol{N}(\omega)$, ainsi que les vecteurs $\boldsymbol{q}(\omega)$ et $\boldsymbol{h}(\omega)$, peuvent être des fonctions non linéaires de ω.

La définition 4.1.3 met en avant le fait que la décision $\boldsymbol{y}(\omega)$ est prise relativement à la valeur de la variable ω (tandis que la décision $\boldsymbol{x}$ dite "dans un premier temps" n'en dépend pas). Il est courant d'omettre cette dépendance pour obtenir une formulation simplifiée, de la forme suivante :

$$\begin{array}{rl} \text{minimiser}_{\substack{\boldsymbol{x} \in \mathbb{R}^n \\ \boldsymbol{y} \in \mathbb{R}^m}} & \mathbb{E}_\omega \left[\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{q}^{\mathrm{T}}\boldsymbol{y} \right] \\ \text{s.c.} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{T}\boldsymbol{x} + \boldsymbol{N}\boldsymbol{y} = \boldsymbol{h} \\ & \boldsymbol{x} \geq \boldsymbol{0} \\ & \boldsymbol{y} \geq \boldsymbol{0}, \end{array} \tag{4.1.6}$$

dans laquelle il est implicite que $\boldsymbol{q}$, $\boldsymbol{T}$, $\boldsymbol{N}$ et $\boldsymbol{h}$ dépendent de ω.

Exemple 4.1.1 (Problème de distribution) *On achète une quantité $x \geq 0$ d'un bien pour répondre à une demande future incertaine. On considère que le coût d'achat à $t = 0$ par unité vaut c_u, que le coût d'achat à $t = 1$ (en cas de stock insuffisant) est de $c_p > c_u$ et que le coût de stockage à $t = 1$ (en cas de surplus) est de c_s. Si $d > 0$ est la valeur incertaine de la demande à $t = 1$, le coût final de l'achat de quantité x est*

$$F(x, d) = c_u x + c_p \max(d - x, 0) + c_s \max(x - d, 0).$$

La formulation de ce problème en programme stochastique (au sens de la définition 4.1.1) donne

$$\begin{array}{rl} \textit{minimiser}_{x\in\mathbb{R}} & \mathbb{E}_d\left[F(x, d)\right] \\ \textit{s.c.} & x \geq 0, \end{array}$$

tandis que celle avec recours s'écrit

$$\begin{array}{rl} \textit{minimiser}_{x\in\mathbb{R}} & c_u x + \mathbb{E}_d\left[Q(x, d)\right] \\ \textit{s.c.} & x \geq 0, \end{array}$$

avec $Q(x, d) = c_p \max(d - x, 0) + c_s \max(x - d, 0)$. Cette quantité peut s'interpréter comme la solution d'un programme linéaire

$$Q(x, d) = \begin{array}{rl} \min_{y(d),z(d)} & c_p y(d) + c_s z(d) \\ \textit{s. c.} & y(d) \geq d - x \\ & z(d) \geq x - d \\ & y(d) \geq 0 \\ & z(d) \geq 0. \end{array}$$

La reformulation avec dépendance implicite en d, à la manière de (4.1.6), s'écrit :

$$\begin{array}{rl} \textit{minimiser}_{\substack{x\in\mathbb{R} \\ y\in\mathbb{R} \\ z\in\mathbb{R}}} & \mathbb{E}_d\left[c_u x + c_p y + c_s z\right] \\ \textit{s. c.} & x + y \geq d \\ & -x + z \geq -d \\ & x, y, z \geq 0. \end{array}$$

4.2 Approche par scénarios

4.2.1 Principe

L'approche par scénarios est l'une des techniques les plus fréquentes en programmation stochastique. Elle consiste à n'envisager qu'un nombre fini de valeurs possibles pour la variable aléatoire ω, appelés **scénarios**. On définit ainsi des scénarios $\{\omega_1, \ldots, \omega_K\}$ ainsi que leurs probabilités d'occurrence $\{p_1, \ldots, p_K\}$. Par conséquent, ces probabilités vérifient $\sum_{k=1}^{K} p_k = 1$.

L'intérêt de cette approche réside dans le fait qu'il est maintenant possible de réécrire un programme stochastique de manière déterministe, en introduisant une variable pour chaque scénario possible. Dans le cadre d'un programme en deux temps, cela signifie que l'on définit une variable $\boldsymbol{y}_k$ associée au scénario ω_k.

Proposition 4.2.1 *Sous l'hypothèse de K scénarios sur ω, le programme linéaire stochastique en deux temps (4.1.6) s'écrit de manière déterministe comme*

$$\begin{array}{rll} \textit{minimiser}_{\substack{\boldsymbol{x}\in\mathbb{R}^n \\ \boldsymbol{y}_1,\ldots,\boldsymbol{y}_K\in\mathbb{R}^m}} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \sum_{k=1}^K p_k \boldsymbol{q}_k^{\mathrm{T}}\boldsymbol{y}_k & \\ \textit{s.c.} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} & \\ & \boldsymbol{T}_k\boldsymbol{x} + \boldsymbol{N}_k\boldsymbol{y}_k = \boldsymbol{h}_k \quad k = 1,\ldots,K & \\ & \boldsymbol{x} \geq \boldsymbol{0} & \\ & \boldsymbol{y}_k \geq \boldsymbol{0} \quad k = 1,\ldots,K. & \end{array} \tag{4.2.1}$$

La structure très particulière du problème (4.2.1) va faciliter sa résolution. En effet, à $\boldsymbol{x}$ fixé, les $\boldsymbol{y}_k$ peuvent être calculés indépendamment en résolvant k programmes linéaires. C'est le principe de la méthode décrite dans la section suivante.

4.2.2 Décomposition de Benders pour programmes linéaires stochastiques

Dans ce qui suit, on réécrit le problème (4.2.1) comme

$$\begin{array}{rll} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \sum_{k=1}^K P_k(\boldsymbol{x}) & \\ \text{s.c.} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} & \\ & \boldsymbol{x} \geq \boldsymbol{0}, & \end{array} \tag{4.2.2}$$

où chaque valeur $P_k(\boldsymbol{x})$ est la valeur optimale d'un programme linéaire déterminé par $\boldsymbol{x}$:

$$P_k(\boldsymbol{x}) = \begin{array}{rll} \min_{\boldsymbol{y}_k} & p_k \boldsymbol{q}_k^{\mathrm{T}}\boldsymbol{y}_k & \\ \text{s. c.} & \boldsymbol{N}_k\boldsymbol{y}_k = \boldsymbol{h}_k - \boldsymbol{T}_k\boldsymbol{x} & \\ & \boldsymbol{y}_k \geq \boldsymbol{0}. & \end{array} \tag{4.2.3}$$

L'algorithme dit de décomposition de Benders procède itérativement en partant de $\boldsymbol{x}^0 \in \mathbb{R}^n$, typiquement choisi comme

$$\boldsymbol{x}^0 \in \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} | \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0} \right\}.$$

Pour chaque itération i, l'algorithme calcule les valeurs $\{P_k(\boldsymbol{x}^i)\}$ en résolvant les programmes linéaires associés, pour lesquels on suppose que leurs duaux sont toujours réalisables. On se trouve alors dans l'une des situations suivantes.

a) Soit le problème (4.2.3) possède une solution, et dans ce cas il existe $\boldsymbol{u}_k^i$ tel que

$$\forall \boldsymbol{x}, P_k(\boldsymbol{x}) \geq [\boldsymbol{u}_k^i]^{\mathrm{T}}(\boldsymbol{T}_k\boldsymbol{x}^i - \boldsymbol{T}_k\boldsymbol{x}) + P_k(\boldsymbol{x}^i).$$

b) Soit le problème est irréalisable, et il existe alors $\boldsymbol{u}_k^i$ tel que

$$[\boldsymbol{u}_k^i]^{\mathrm{T}}(\boldsymbol{h}_k - \boldsymbol{T}_k\boldsymbol{x}) \leq 0.$$

Dans les deux cas, il s'agit de contraintes linéaires (appelées coupes) que l'on peut rajouter au problème utilisé pour calculer $\boldsymbol{x}^i$ de sorte à obtenir un meilleur $\boldsymbol{x}^{i+1}$. Ce vecteur sera en effet calculé

en résolvant

$$\begin{array}{rl} \text{minimiser}_{\substack{\boldsymbol{x}\in\mathbb{R}^n \\ \boldsymbol{y}_1,\dots,\boldsymbol{y}_K\in\mathbb{R}^m}} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \sum_{k=1}^{K} P_k(\boldsymbol{x}) \\ \text{s.c.} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & P_k(\boldsymbol{x}) \geq [\boldsymbol{u}_k^i]^{\mathrm{T}}(\boldsymbol{T}_k\boldsymbol{x}^i - \boldsymbol{T}_k\boldsymbol{x}) + P_k(\boldsymbol{x}^i) \\ & \quad \forall (i,k) \text{ tel que le problème associé à } P_k(\boldsymbol{x}^i) \text{ ait une solution,} \\ & [\boldsymbol{u}_k^i]^{\mathrm{T}}(\boldsymbol{h}_k - \boldsymbol{T}_k\boldsymbol{x}) \leq 0 \\ & \quad \forall (i,k) \text{ tel que le problème associé à } P_k(\boldsymbol{x}^i) \text{ soit irréalisable,} \\ & \boldsymbol{x} \geq \boldsymbol{0} \\ & \boldsymbol{y}_k \geq \boldsymbol{0} \quad k = 1,\dots,K. \end{array} \tag{4.2.4}$$

L'avantage de cette procédure est qu'elle ne requiert que de résoudre des programmes linéaires (au nombre de contraintes croissant). On peut montrer que ce processus converge vers la solution en un nombre fini d'itérations.

4.3 Mesures de risque

4.3.1 Motivation et définition générique

Dans le chapitre 3, et plus particulièrement dans la section 3.3, nous avons étudié une fonction de coût mêlant la variance d'un vecteur de rendement ainsi que sa moyenne. Ce programme peut s'écrire comme un programme stochastique, où la composante stochastique est le rendement. On obtient ainsi

$$\begin{array}{rl} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & f(\boldsymbol{r}^{\mathrm{T}}\boldsymbol{x}) \\ \text{s.c.} & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1 \end{array} \tag{4.3.1}$$

avec

$$f(t) = \frac{\gamma}{2}\sigma_{\boldsymbol{r}}^2(t) - \mathbb{E}_{\boldsymbol{r}}[t], \quad \sigma_{vr}^2(t) = \mathbb{E}_{\boldsymbol{r}}\left[t^2\right] - \mathbb{E}_{\boldsymbol{r}}[t]^2.$$

A travers le terme de variance, f incorpore une information liée au risque du portefeuille. Plus généralement, les fonctions de ce type sont appelées des **mesures de risque**.

Remarque 4.3.1 *L'utilisation, déjà vue au chapitre (2), du rendement moyen comme mesure conduit à un problème dont le résultat sera* **neutre au risque** *(*risk neutral*), par opposition aux autres mesures de risque, qui elles seront attentives au risque (*risk averse*).*

4.3.2 VaR et CVaR

L'utilisation de la variance présente l'inconvénient de diminuer l'investissement dans des actifs qui peuvent prendre des valeurs extrêmes (et donc avoir une forte variance), même si cela n'arrive qu'avec une faible probabilité. Il est possible de définir d'autres mesures de risque qui soient moins sensibles à cela. C'est notamment le but de la valeur à risque (*value-at-risk*), définie ci-dessous et popularisée par J.P. Morgan.

Définition 4.3.1 *Soit z une valeur aléatoire et $\alpha \in (0,1)$ un niveau de sûreté. La* **valeur à risque α de z** *est la valeur δ telle que*

$$\mathbb{P}(z \geq \delta) = 1 - \alpha. \tag{4.3.2}$$

On note alors $\mathrm{VaR}_\alpha[z] = \delta$.

Exemple 4.3.1 *i) Si* $z \sim \mathcal{N}(\mu, \sigma^2)$*, alors* $\mathrm{VaR}_{0.99}[z] = \mu + 2.33\sigma$.

ii) Si $z \in \{z_1, dots, z_K\}$ *avec* $\mathbb{P}(z = z_k) = p_k$*, alors* $\mathrm{VaR}_\alpha[z] = z_J$*, où* J *est le plus petit entier tel que* $\sum_{k=J}^{K} p_k \geq 1 - \alpha$.

On peut ainsi formuler des problèmes d'optimisation où l'on cherche à minimiser la valeur à risque, ce qui donne généralement des solutions plus robustes aux comportements extrêmes qu'en utilisant la variance. Dans le cas d'une approche par scénarios, on retombe à nouveau sur un programme linéaire.

Valeur à risque conditionnelle L'un des inconvénients de la valeur à risque est qu'il ne s'agit pas d'une mesure de risque cohérente, car $\mathrm{VaR}_\alpha[\boldsymbol{z} + \boldsymbol{y}] \not\leq \mathrm{VaR}_\alpha[\boldsymbol{z}] + \mathrm{VaR}_\alpha[\boldsymbol{y}]$ en général. Cela signifie que diversifier un portefeuille n'apporte pas de robustesse au sens de la valeur à risque. La variante cohérente de la valeur à risque la plus répandue est définie ci-après.

Définition 4.3.2 *Soit* z *une valeur aléatoire et* $\alpha \in (0, 1)$ *un niveau de sûreté. La* **valeur à risque conditionnelle** α **de** z*, notée* $\mathrm{CVaR}_\alpha[z]$*, est définie par*

$$\mathrm{CVaR}_\alpha[z] = \mathbb{E}_z[z | z \geq \mathrm{VaR}_\alpha[z]]. \tag{4.3.3}$$

L'interprétation de la valeur à risque conditionnelle est qu'elle exprime, pour une valeur à risque donnée, de combien on excèdera cette valeur en moyenne. Il s'agit d'une mesure de risque cohérente.

Exemple 4.3.2 *i) Si* $z \sim \mathcal{N}(\mu, \sigma^2)$*, alors* $\mathrm{VaR}_{0.99}[z] = \mu + 2.67\sigma$.

ii) Si $z \in \{z_1, dots, z_K\}$ *avec* $\mathbb{P}(z = z_k) = p_k$*, alors*

$$\mathrm{CVaR}_\alpha[z] = \frac{1}{1-\alpha} \sum_{k=J}^{K} p_k,$$

où J *est le plus petit entier tel que* $\sum_{k=J+1}^{K} p_k \in [1 - \alpha - p_J, 1 - \alpha)$.

Le calcul de la valeur à risque conditionnelle n'est en général pas possible de manière explicite, car il correspond à la résolution d'un programme linéaire stochastique en deux temps, comme le montre le résultat ci-dessous.

Théorème 4.3.1 *Si* z *une valeur aléatoire et* $\alpha \in (0, 1)$*, alors*

$$\begin{array}{rcrl} \mathrm{CVaR}_\alpha[z] & = & \min_\delta \left[\delta + \frac{1}{1-\alpha}\mathbb{E}_z[\max\{z - \delta, 0\}]\right] & \\ & & \min_{\delta,u} & \delta + \frac{1}{1-\alpha}\mathbb{E}_z[u] \\ & = & \textit{s.c.} & u \geq z - \delta \\ & & & u \geq 0. \end{array} \tag{4.3.4}$$

En règle générale, l'utilisation de la valeur à risque conditionnelle conduira donc au moins à une modélisation en programme stochastique avec recours.

4.4 Conclusion du chapitre 4

La prise en compte explicite du caractère aléatoire d'une composante de l'environnement (typiquement, le rendement d'un actif) conduit à des modélisations et donc des problèmes d'optimisation plus complexes. Parmi ceux-ci, les programmes stochastiques en deux temps font partie des classes les mieux comprises sur le plan théorique ainsi que sur le plan algorithmique. Les approches par scénarios permettent également de décrire ces problèmes de manière déterministe. Dans le cas où ces programmes sont aussi linéaires, des approches telles que la décomposition de Benders peuvent exploiter la structure inhérente à la présence de scénarios.

L'un des atouts majeurs de la programmation stochastique est de pouvoir quantifier le risque de certains investissements, de manière plus flexible que ce qui peut être fait avec la variance. Cela est rendu possible grâce à la notion de mesure de risque, dont la valeur à risque et la valeur à risque conditionnelle forment deux importantes instances.

4.5 Exercices

Exercice 5 : Déviation en moyenne absolue

Comme on l'a vu plus haut, l'utilisation de la variance en tant que mesure de risque présente quelques inconvénients, notamment celui d'être sensible aux comportements aberrants, et donc de conduire à des solutions conservatrices. On s'intéresse ici à une autre métrique définie via la **déviation en moyenne absolue**.

Soit $\boldsymbol{r} \in \mathbb{R}^n$ un vecteur de retours sur investissement aléatoire de moyenne $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{r}]$. La déviation en moyenne absolue de $\boldsymbol{r}$ est la fonction $d : \mathbb{R}^n \to \mathbb{R}$ définie par

$$\forall \boldsymbol{x} \in \mathbb{R}^n, \quad d(\boldsymbol{x}) = \mathbb{E}\left[\left|(\boldsymbol{r} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{x}\right|\right]. \tag{4.5.1}$$

a) Supposons que $\boldsymbol{r}$ suive une loi discrète de valeurs $\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_K\}$ avec $\mathbb{P}(\boldsymbol{r} = \boldsymbol{r}_k) = p_k$. Écrire alors le problème

$$\begin{array}{rl} \text{minimiser}_{\boldsymbol{x} \in \mathbb{R}^n} & d(\boldsymbol{x}) \\ \text{s.c.} & \boldsymbol{e}^{\mathrm{T}} \boldsymbol{x} = 1, \end{array} \tag{4.5.2}$$

en fonction de $\{\boldsymbol{r}_k\}_{k=1,\ldots,K}$ et $\{p_k\}_{k=1,\ldots,K}$.

b) En introduisant la décomposition

$$\left\{ \begin{array}{rcl} (\boldsymbol{r}_k - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{x} & = & w_k^+ - w_k^- \\ \left|(\boldsymbol{r}_k - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{x}\right| & = & w_k^+ + w_k^-, \end{array} \right.$$

où w_k^+ et w_k^- sont des quantités positives ou nulles (représentant la partie positive et négative de $(\boldsymbol{r}_k - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{x}$), formuler le problème (4.5.2) comme un programme linéaire.

c) Donner un avantage pratique de résoudre un programme linéaire tel que celui formulé dans cet exercice par rapport à la résolution de programmes quadratiques impliquant la variance.

Solution de l'exercice 5

a) Par définition de l'espérance, on a dans le cas de cette question

$$d(\boldsymbol{x}) = \mathbb{E}\left[|(\boldsymbol{r}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{x}|\right] \quad = \quad \sum_{i=1}^{K} p_k |(\boldsymbol{r}_k - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{x}|.$$

Le problème s'écrit donc

$$\begin{array}{rl} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \sum_{i=1}^{K} p_k |(\boldsymbol{r}_k - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{x}| \\ \text{s.c.} & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1. \end{array}$$

b) En utilisant la décomposition, on peut reformuler le problème en remplaçant l'objectif par

$$\sum_{i=1}^{K} p_k |(\boldsymbol{r}_k - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{x}| = \sum_{i=1}^{K} p_k (w_k^+ - w_k^-)$$

et en rajoutant les contraintes

$$(\boldsymbol{r}_k - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{x} = w_k^+ + w_k^-.$$

On obtient donc le problème

$$\begin{array}{rll} \text{minimiser} \begin{array}{c} \boldsymbol{x}\in\mathbb{R}^n \\ \boldsymbol{w}^+\in\mathbb{R}^K \\ \boldsymbol{w}^-\in\mathbb{R}^K \end{array} & \sum_{i=1}^{K} p_k (w_k^+ - w_k^-) & \\ \text{s.c.} & (\boldsymbol{r}_k - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{x} - w_k^+ - w_k^- = 0 & k = 1, \ldots, K, \\ & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1. & \end{array}$$

c) Un programme linéaire peut se résoudre avec un plus grand nombre de variables qu'un programme quadratique, car il existe des algorithmes très efficaces en grande dimension.

Exercice 6 : Utilité

On considère que l'on dispose d'un capital de $w_0 > 0$ à investir pour construire un portefeuille. En tant que personne, on dispose d'une utilité qui nous est propre représentée par une fonction $U : \mathbb{R} \to \mathbb{R}$, que l'on va chercher à maximiser via la construction du portefeuille. On note $w = w_0(1 + \boldsymbol{r}^{\mathrm{T}}\boldsymbol{x})$ le capital obtenu après investissement, dont on suppose qu'il dépend d'un vecteur de rendement aléatoire.

Le problème de construction de portefeuille d'utilité maximale s'écrit comme suit :

$$\begin{array}{ll} \text{maximiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \mathbb{E}_r\left[U(w)\right] \\ \text{s. c.} & w = w_0(1 + \boldsymbol{r}^{\mathrm{T}}\boldsymbol{x}) \\ & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1. \end{array} \tag{4.5.3}$$

On se concentre dans la suite sur le cas $U : w \mapsto U(w)$.

a) Si $\boldsymbol{r} = \boldsymbol{\mu}$ est déterministe, reformuler le problème (4.5.3) en un programme linéaire. *Utiliser pour cela le fait que maximiser* $\log(f(\boldsymbol{x}))$ *revient à minimiser* $f(\boldsymbol{x})$.

b) Généraliser le résultat précédent dans le cas où $\boldsymbol{r} \in \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_K\}$ avec $\mathbb{P}(\boldsymbol{r} = \boldsymbol{r}_k) = p_k$.

c) Si $\boldsymbol{r}$ suit une loi normale de moyenne $\boldsymbol{\mu}$ et de matrice de covariance $\boldsymbol{V}$, on a alors

$$\mathbb{E}_r\left[U(w)\right] = -\frac{c}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{V}(\boldsymbol{x}-\boldsymbol{\mu})$$

pour $c > 0$. Montrer que dans ce cas, le problème (4.5.3) se ramène à un problème quadratique. Comparer le problème obtenu avec celui de gestion de portefeuille de la forme proposée par Markowitz.

Solution de l'exercice 6

a) Dans le cas d'un vecteur $\boldsymbol{r}$ déterministe, on a

$$\mathbb{E}_r\left[U(w)\right] = U(w_0(1+\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x})) = \log(w_0(1+\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x})).$$

Par conséquent, le problème (4.5.3) devient

$$\begin{array}{ll} \text{maximiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \log(w_0(1+\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x})) \\ \text{s. c.} & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1. \end{array}$$

En utilisant le fait que maximiser un logarithme revient à minimiser la fonction, une reformulation équivalente de ce problème est

$$\begin{array}{ll} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & w_0\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x} \\ \text{s. c.} & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1, \end{array}$$

qui est bien un programme linéaire.

b) Lorsque $\boldsymbol{r}$ suit une distribution discrète, la fonction objectif du problème (4.5.3) s'écrit

$$\begin{aligned} \mathbb{E}_r\left[U(w)\right] &= \mathbb{E}_r\left[U(w_0(1+\boldsymbol{r}^{\mathrm{T}}\boldsymbol{x}))\right] \\ &= \sum_{k=1}^{K} p_k U(w_0(1+\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{x})) \\ &= \log\left(\prod_{k=1}^{K} w_0^{p_k}(1+\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{x})^{p_k}\right). \end{aligned}$$

Par conséquent, en utilisant le fait que maximiser un logarithme revient à minimiser la fonction argument du logarithme, on arrive à

$$\begin{array}{ll} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \prod_{k=1}^{K} w_0^{p_k}(1+\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{x})^{p_k} \\ \text{s. c.} & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1, \end{array}$$

On notera que ce problème n'est un programme linéaire que lorsque $K = 1$ et $p_1 = 1$. [1]

[1] Sinon, il s'agit d'un programme dit géométrique, pour lequel des techniques telles que celles vues en cours peuvent être développées.

c) En introduisant la première contrainte du problème (4.5.3) dans l'objectif, on aboutit au problème

$$\begin{array}{ll} \text{maximiser}_{\boldsymbol{x}\in\mathbb{R}^n} & -\frac{c}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{V}(\boldsymbol{x}-\boldsymbol{\mu}) \\ \text{s. c.} & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x}=1, \end{array}$$

qui est équivalent à

$$\begin{array}{ll} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \frac{c}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{V}(\boldsymbol{x}-\boldsymbol{\mu}) \\ \text{s. c.} & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x}=1. \end{array}$$

Comme multiplier par une constante positive ne change pas les solutions d'un problème d'optimisation, ce dernier problème est aussi équivalent à

$$\begin{array}{ll} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{V}(\boldsymbol{x}-\boldsymbol{\mu}) \\ \text{s. c.} & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x}=1 \end{array}$$

En développant la fonction objectif, on obtient

$$\begin{aligned} \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{V}(\boldsymbol{x}-\boldsymbol{\mu}) &= \frac{1}{2}\left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x}-2\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x}+\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\mu}\right) \\ &= \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x}-\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x}+\frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\mu}. \end{aligned}$$

Le dernier terme étant constant, il n'importe pas dans le processus de minimisation. Au final, on obtient donc le problème

$$\begin{array}{ll} \text{minimiser}_{\boldsymbol{x}\in\mathbb{R}^n} & \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x}-\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x} \\ \text{s. c.} & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x}=1. \end{array}$$

Ce problème ressemble au problème d'allocation de Markowitz (avec $\gamma = 1$). Cependant, au lieu d'avoir un terme en $-\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x}$ qui tend à maximiser le rendement, on a ici un terme en $-\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x}$, qui tend à maximiser une version normalisée du rendement moyen.

Bibliographie

[1] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2011.

[2] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, United Kingdom, 2004.

[3] G. Cornuéjols, J. Peña, and R. Tütüncü. *Optimization Methods in Finance*. Cambridge University Press, Cambridge, United Kingdom, second edition, 2018.

Annexe A

Bases mathématiques

A.1 Éléments d'algèbre linéaire

Les fondements mathématiques de l'optimisation se trouvent dans l'analyse réelle, et en particulier dans le calcul différentiel. Les structures d'algèbre linéaire dans $\mathbb{R}^n$ jouent cependant un rôle prépondérant en optimisation (et en sciences des données). Cette section regroupe les résultats de base qui seront utilisés dans le cours.

Pour approfondir ces notions, on pourra consulter les liens suivants :

- https://www.ceremade.dauphine.fr/∼carlier/polyalgebre.pdf (en français);
- http://vmls-book.stanford.edu/vmls.pdf (Chapitres 1 à 3, en anglais).

A.1.1 Algèbre linéaire vectorielle

On considèrera toujours l'espace des vecteurs $\mathbb{R}^n$ muni de sa structure d'espace vectoriel normé de dimension d. On définit donc les opérations suivantes :

- Pour tous $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, la somme des vecteurs $\boldsymbol{x}$ et $\boldsymbol{y}$ est notée $\boldsymbol{x} + \boldsymbol{y} = [x_i + y_i]_{1 \leq i \leq d}$;
- Pour tout $\lambda \in \mathbb{R}$, on définit $\lambda \boldsymbol{x} \overset{n}{=} \lambda \cdot \boldsymbol{x} = [\lambda x_i]_{1 \leq i \leq d}$. Dans ce contexte, un tel réel λ sera appelé un *scalaire*.

A l'aide de ces opérations, nous pouvons donc construire des **combinaisons linéaires** de vecteurs de $\mathbb{R}^n$ dont le résultat sera un vecteur de $\mathbb{R}^n$, à savoir des vecteurs de la forme $\sum_{i=1}^{p} \lambda_i \boldsymbol{x}_i$, où $\boldsymbol{x}_i \in \mathbb{R}^n$ et $\lambda_i \in \mathbb{R}$ pour tout $i = 1, \ldots, p$.

Pour ce qui est de l'espace des matrices $\mathbb{R}^{n \times d}$, on peut également le munir d'une structure d'espace vectoriel normé de dimension nd. On définit donc les opérations suivantes :

- Pour tous $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n \times d}$, la somme des matrices $\boldsymbol{A}$ et $\boldsymbol{B}$ est notée $\boldsymbol{A} + \boldsymbol{B} = [\boldsymbol{A}_{ij} + \boldsymbol{B}_{ij}]_{\substack{1 \leq i \leq d \\ 1 \leq j \leq d}}$;
- Pour tout scalaire $\lambda \in \mathbb{R}$, on définit $\lambda \boldsymbol{A} \overset{n}{=} \lambda \cdot \boldsymbol{A} = [\lambda \boldsymbol{A}_{ij}]_{\substack{1 \leq i \leq n \\ 1 \leq j \leq d}}$.

Définition A.1.1 *Un ensemble $\mathcal{S} \subseteq \mathbb{R}^n$ vérifiant les conditions*

1. $\mathbf{0}_d \in \mathcal{S}$;

2. $\forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{S},\ \boldsymbol{x} + \boldsymbol{y} \in \mathcal{S}$;

3. $\forall \boldsymbol{x} \in \mathcal{S},\ \forall \lambda \in \mathbb{R},\ \lambda \boldsymbol{x} \in \mathcal{S}$.

s'appelle un sous-espace vectoriel *de* $\mathbb{R}^n$.

Définition A.1.2 *Soient* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p$ p *vecteurs de* $\mathbb{R}^n$. *Le sous-espace engendré par les vecteurs* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p$, *noté* $\mathrm{vect}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p)$, *est le sous-espace vectoriel*

$$\mathrm{vect}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p) \ := \ \left\{ \boldsymbol{x} = \sum_{i=1}^{p} \alpha_i \boldsymbol{x}_i \middle| \alpha_i \in \mathbb{R}\ \forall i \right\}.$$

On rappelle ensuite les différentes propriétes notables des familles de vecteurs.

Définition A.1.3

- *Une famille libre* $\{\boldsymbol{x}_i\}_{i=1}^k$ *de vecteurs de* $\mathbb{R}^n$ *est telle que les vecteurs sont linéairement indépendants : pour tous scalaires* $\lambda_1, \ldots, \lambda_k$ *tels que* $\sum_{i=1}^k \lambda_i \boldsymbol{x}_i = 0$, *alors* $\lambda_1 = \cdots = \lambda_k = 0$. *On a nécessairement* $k \leq n$.
- *Une famille liée est une famille de vecteurs qui n'est pas libre.*
- *Une famille génératrice de vecteurs de* $\mathbb{R}^n$ *est un ensemble de vecteurs* $\{\boldsymbol{x}_i\}$ *tel que le sous-espace engendré par ces vecteurs soit égal à* $\mathbb{R}^n$.
- *Une* base *de* $\mathbb{R}^n$ *est une famille de vecteurs* $\{\boldsymbol{x}_i\}_{i=1}^n$ *qui est à la fois libre et génératrice. Tout vecteur de* $\mathbb{R}^n$ *s'écrit alors de manière unique comme combinaison linéaire des* $\boldsymbol{x}_i$. *Toute base de* $\mathbb{R}^n$ *comporte exactement* n *vecteurs.*

Comme la taille d'une base de $\mathbb{R}^n$ est n, on dit que cet espace vectoriel est de dimension n. En conséquence, la dimension d'un sous-espace vectoriel est au plus n.

Exemple A.1.1 *Tout vecteur* $\boldsymbol{x}$ *de* $\mathbb{R}^n$ *s'écrit* $\boldsymbol{x} = \sum_{i=1}^n x_i \boldsymbol{e}_i$, *où* $\boldsymbol{e}_i = [0 \cdots 0\ 1\ 0 \cdots 0]^{\mathrm{T}}$ *est le* i*-ème vecteur de la base canonique (le coefficient* 1 *se trouvant en* i*-ème position).*

Norme et produit scalaire L'utilisation d'une norme, et du produit scalaire associé, permet de mesurer les distances entre vecteurs, ce qui sera particulièrement utile pour montrer la convergence d'une suite de points générés par un algorithme vers une solution d'un problème d'optimisation.

Définition A.1.4 *La* **norme euclidienne** $\|\cdot\|$ *sur* $\mathbb{R}^n$ *est définie pour tout vecteur* $\boldsymbol{x} \in \mathbb{R}^n$ *par*

$$\|\boldsymbol{x}\| \ := \ \sqrt{\sum_{i=1}^{n} x_i^2}.$$

Remarque A.1.1 *Il s'agit bien d'une norme, car elle vérifie les quatres axiomes d'une norme :*

1. $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n,\ \|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$;
2. $\|\boldsymbol{x}\| = 0 \ \Leftrightarrow \boldsymbol{x} = \boldsymbol{0}_{\mathbb{R}^n}$;
3. $\forall \boldsymbol{x},\ \|\boldsymbol{x}\| \geq 0$;

4. $\forall \boldsymbol{x} \in \mathbb{R}^n, \forall \lambda \in \mathbb{R}, \|\lambda \boldsymbol{x}\| = |\lambda| \|\boldsymbol{x}\|$.

On dira qu'un vecteur $\boldsymbol{x} \in \mathbb{R}^n$ est unitaire si $\|\boldsymbol{x}\| = 1$.

Définition A.1.5 *Pour tous vecteurs $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, le* **produit scalaire** *dérivé de la norme euclidienne est une fonction de $\boldsymbol{x}$ et $\boldsymbol{y}$, notée $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}$, et définie par*

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} := \sum_{i=1}^{n} x_i y_i.$$

Deux vecteurs $\boldsymbol{x}$ et $\boldsymbol{y}$ tels que $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = 0$ seront dits orthogonaux.

On notera en particulier que $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}$. Ce produit scalaire définit donc un "produit" entre un vecteur ligne et un vecteur colonne.

Proposition A.1.1 *Soient $\boldsymbol{x}$ et $\boldsymbol{y}$ deux vecteurs de $\mathbb{R}^n$. Alors, on a les propriétés suivantes :*

i) $\|\boldsymbol{x} + \boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + 2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} + \|\boldsymbol{y}\|^2$;

ii) $\|\boldsymbol{x} - \boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 - 2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} + \|\boldsymbol{y}\|^2$;

iii) $\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 = \frac{1}{4}\left(\|\boldsymbol{x} + \boldsymbol{y}\|^2 + \|\boldsymbol{x} - \boldsymbol{y}\|^2\right)$;

iv) **Inégalité de Cauchy-Schwarz** *:*

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, \qquad \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} \leq \|\boldsymbol{x}\| \|\boldsymbol{y}\|.$$

Remarque A.1.2 *La dernière inégalité est un résultat très important, qui s'utilise aussi bien en algèbre linéaire qu'en analyse fonctionnelle. Comme on le verra, elle apparaît également dans la preuve des inégalités de Taylor, qui sont fondamentales en optimisation.*

A.1.2 Algèbre linéaire matricielle

Sur les espaces matriciels, on définit également un produit matriciel entre matrices de dimensions compatibles. Pour toutes matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ et $\boldsymbol{B} \in \mathbb{R}^{n \times p}$, la matrice produit $\boldsymbol{AB}$ est définie comme la matrice $\boldsymbol{C} \in \mathbb{R}^{m \times p}$ telle que

$$\forall i = 1, \ldots, m, \ \forall j = 1, \ldots, p, \quad \boldsymbol{C}_{ij} = \sum_{k=1}^{n} \boldsymbol{A}_{ik} \boldsymbol{B}_{kj}.$$

Par analogie, le produit d'une matrice $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ avec un vecteur $\boldsymbol{x} \in \mathbb{R}^n$ sera le vecteur $\boldsymbol{y} \in \mathbb{R}^m$ donné par

$$\forall i = 1, \ldots, m, \ y_i = \sum_{j=1}^{n} \boldsymbol{A}_{ij} x_j.$$

Remarque A.1.3 *On notera que la notation du produit scalaire sur $\mathbb{R}^n$ correspond au produit de deux matrices de taillle $1 \times n$ et $n \times 1$, dont le résultat est une matrice 1×1, c'est-à-dire un scalaire.*

Lorsque l'on travaille avec des matrices, on s'intéresse généralement aux sous-espaces définis ci-dessous.

Définition A.1.6 (Sous-espaces matriciels) *Soit une matrice $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, on définit les deux sous-espaces suivants :*

- *Le* noyau *de $\boldsymbol{A}$ est le sous-espace vectoriel*

$$\ker(\boldsymbol{A}) \;:=\; \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}_m\}$$

- *L'*image *de $\boldsymbol{A}$ est le sous-espace vectoriel*

$$\mathrm{Im}(\boldsymbol{A}) \;:=\; \{\boldsymbol{y} \in \mathbb{R}^m \mid \exists \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}\}$$

  La dimension de ce sous-espace vectoriel s'appelle le **rang** *de $\boldsymbol{A}$. On la note $\mathrm{rang}(\boldsymbol{A})$ et on a $\mathrm{rang}(\boldsymbol{A}) \leq \min\{m, n\}$.*

Théorème A.1.1 (Théorème du rang) *Pour toute matrice $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, on a*

$$\dim(\ker(\boldsymbol{A})) + \mathrm{rang}(\boldsymbol{A}) = n.$$

Définition A.1.7 (Normes matricielles) *On définit sur $\mathbb{R}^{m\times n}$ la norme d'opérateur $\|\cdot\|$ et la norme de Frobenius $\|\cdot\|_F$ par*

$$\forall \boldsymbol{A} \in \mathbb{R}^{m\times n}, \begin{cases} \|\boldsymbol{A}\| & := \max\limits_{\substack{\boldsymbol{x}\in\mathbb{R}^n \\ \boldsymbol{x}\neq\boldsymbol{0}_n}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|} = \max\limits_{\substack{\boldsymbol{x}\in\mathbb{R}^n \\ \|\boldsymbol{x}\|=1}} \|\boldsymbol{A}\boldsymbol{x}\| \\ \|\boldsymbol{A}\|_F & := \sqrt{\sum_{\substack{1\leq i\leq m \\ 1\leq j\leq n}} \boldsymbol{A}_{ij}^2}. \end{cases}$$

Définition A.1.8 (Matrice symétrique) *Une matrice carrée $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ est dite symétrique si elle vérifie $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}$.*

L'ensemble des matrices symétriques de taille $n \times n$ sera noté $\mathcal{S}^n$.

Définition A.1.9 (Matrice inversible) *Une matrice carrée $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ est dite inversible s'il existe $\boldsymbol{B} \in \mathbb{R}^{n\times n}$ telle que $\boldsymbol{B}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{B} = \boldsymbol{I}_n$ (où l'on rappelle que $\boldsymbol{I}_n$ désigne la matrice identité de $\mathbb{R}^{n\times n}$).*

Si elle existe, une telle matrice $\boldsymbol{B}$ est unique : elle est appelée **l'inverse de $\boldsymbol{A}$** *et on la note $\boldsymbol{A}^{-1}$.*

Définition A.1.10 (Matrice (semi-)définie positive) *Une matrice carrée $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ symétrique est dite semi-définie positive si*

$$\forall \boldsymbol{x} \in \mathbb{R}^n, \quad \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} \geq 0,$$

ce que l'on notera $\boldsymbol{A} \succeq 0$.

Une telle matrice est dite définie positive lorsque $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} > 0$ pour tout vecteur $\boldsymbol{x}$ non nul, ce que l'on notera $\boldsymbol{A} \succ 0$.

Définition A.1.11 (Matrice orthogonale) *Une matrice carrée $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ est dite orthogonale si $\boldsymbol{P}^{\mathrm{T}} = \boldsymbol{P}^{-1}$.*

Par extension, on dira que $\boldsymbol{Q} \in \mathbb{R}^{m\times n}$ avec $m \leq n$ est orthogonale si $\boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}} = \boldsymbol{I}_m$ (les colonnes de $\boldsymbol{Q}$ sont donc orthonormées dans $\mathbb{R}^m$).

On notera que lorsque $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ est une matrice orthogonale, alors sa transposée $\boldsymbol{Q}^{\mathrm{T}}$ est également orthogonale (ce résultat n'est pas vrai pour une matrice rectangulaire). On utilisera fréquemment la propriété des matrices orthogonales énoncée ci-dessous.

Lemme A.1.1 *Soit une matrice $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ et $\boldsymbol{U} \in \mathbb{R}^{m\times m}$, $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ des matrices orthogonales, respectivement de $\mathbb{R}^{m\times m}$ et $\mathbb{R}^{n\times n}$. On a*

$$\|\boldsymbol{A}\| = \|\boldsymbol{U}\boldsymbol{A}\| = \|\boldsymbol{A}\boldsymbol{V}\| \quad \text{et} \quad \|\boldsymbol{A}\|_F = \|\boldsymbol{U}\boldsymbol{A}\|_F = \|\boldsymbol{A}\boldsymbol{V}\|_F,$$

c'est-à-dire que la multiplication par une matrice orthogonale ne modifie pas la norme d'opérateur.

Par corollaire immédiat du lemme précédent, on note qu'une matrice $\boldsymbol{Q} \in \mathbb{R}^{m\times n}$ orthogonale avec $m \leq n$ vérifie nécessairement $\|\boldsymbol{Q}\| = 1$ et $\|\boldsymbol{Q}\|_F = \sqrt{m}$.

Définition A.1.12 (Valeur propre) *Soit une matrice $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. On dit que $\lambda \in \mathbb{R}$ est une* **valeur propre de $\boldsymbol{A}$** *si*

$$\exists \boldsymbol{v} \in \mathbb{R}^n, \boldsymbol{v} \neq \boldsymbol{0}_n, \quad \boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}.$$

Le vecteur $\boldsymbol{v}$ est appelé un **vecteur propre associé à la valeur propre λ**. *L'ensemble des valeurs propres de $\boldsymbol{A}$ s'appelle le spectre de $\boldsymbol{A}$.*

Le sous-espace engendré par les vecteurs propres associés à la même valeur propre d'une matrice s'appelle un sous-espace propre. Sa dimension correspond à l'ordre de multiplicité de la valeur propre relativement à la matrice.

Proposition A.1.2 *Pour toute matrice $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, on a les propriétés suivantes :*

- *La matrice $\boldsymbol{A}$ possède n valeurs propres complexes mais pas nécessairement réelles.*
- *Si la matrice $\boldsymbol{A}$ est semi-définie positive (respectivement définie positive), alors ses valeurs propres sont réelles positives (respectivement strictement positives).*
- *Le noyau de $\boldsymbol{A}$ est engendré par les vecteurs propres associés à la valeur propre 0.*

Théorème A.1.2 (Théorème spectral) *Toute matrice carrée $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ symétrique admet une décomposition dite* **spectrale** *de la forme :*

$$\boldsymbol{A} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^{-1},$$

où $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ est une matrice orthogonale , dont les colonnes $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$ forment une base orthonormée *de vecteurs propres, et $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times n}$ est une matrice diagonale qui contient les n valeurs propres de $\boldsymbol{A}$ $\lambda_1, \ldots, \lambda_n$ sur la diagonale.*

Il est à noter que la décomposition spectrale n'est pas unique. En revanche, l'ensemble des valeurs propres est unique, que l'on prenne en compte les ordres de multiplicité ou non.

Géométriquement parlant, on voit ainsi que, pour tout vecteur $\boldsymbol{x} \in \mathbb{R}^n$ décomposé dans la base des $\boldsymbol{p}_i$ que l'on multiplie par $\boldsymbol{A}$, les composantes de ce vecteur associées aux plus grandes valeurs propres[1] seront augmentées, tandis que celles associées aux valeurs propres de petite magnitude seront réduites (voire annihilées dans le cas d'une valeur propre nulle).

[1] On parle ici de plus grandes valeurs propres en valeur absolue, ou magnitude.

Lien avec la décomposition en valeurs singulières Soit une matrice rectangulaire $\boldsymbol{A} \in \mathbb{R}^{m \times n}$: dans le cas général, les dimensions de la matrice diffèrent, et on ne peut donc pas parler de valeurs propres de la matrice $\boldsymbol{A}$. On peut en revanche considérer les deux matrices

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} \in \mathbb{R}^{n \times n} \qquad \text{et} \qquad \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} \in \mathbb{R}^{m \times m}.$$

Ces matrices sont symétriques réelles, et par conséquent diagonalisables. On peut se baser sur cette propriété pour obtenir une décomposition en valeurs singulières de $\boldsymbol{A}$ (ou *SVD* d'après l'acronyme anglais).