



HAL
open science

Méthodes d'optimisation et d'apprentissage appliquées à des problèmes de trafic aérien

David Gianazza

► **To cite this version:**

David Gianazza. Méthodes d'optimisation et d'apprentissage appliquées à des problèmes de trafic aérien. Optimisation et contrôle [math.OA]. INP DE TOULOUSE, 2016. tel-01284493

HAL Id: tel-01284493

<https://theses.hal.science/tel-01284493>

Submitted on 7 Mar 2016

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

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



Institut National Polytechnique de Toulouse

Ecole Doctorale Mathématiques, Informatique,
et Télécommunications de Toulouse

Habilitation à diriger des recherches

présentée par

David Gianazza

Enseignant-chercheur à l'Ecole Nationale de l'Aviation Civile

Méthodes d'optimisation et d'apprentissage
appliquées à des problèmes de trafic aérien

Le 18/02/2016

Devant un jury composé de :

M. Eric FERON - Professor of Aerospace Software Engineering at Georgia Tech (Rapporteur)

M. Marc SCHOENAUER - Directeur de recherche à l'INRIA Saclay (Rapporteur)

M. Joseph NOAILLES - Professeur émérite à l'INP Toulouse (Rapporteur)

M. Jean-Marc ALLIOT - Directeur de recherche à l'IRIT, Toulouse

M. Patrick AMESTOY - Professeur INPT-ENSEEIH-IRIT Toulouse

Mme. Michèle SEBAG - Directrice de recherche du CNRS, Université Paris Sud

Remplir son contrat coûte que coûte est la juste sanction endurée par qui n'a pas su le rompre à temps.

Citation attribuée à Louis-René des Forêts.¹

1. À propos de son livre *Ostinato*, qu'il mit 20 ans à écrire.

Remerciements

Je tiens en premier lieu à remercier Eric Feron, Joseph Noailles et Marc Schoenauer, d’avoir accepté de relire le présent mémoire et d’en être rapporteurs. La longueur de ce pavé dépasse assurément les limites communément admises. Je les remercie du fond du cœur de ne pas m’en avoir tenu trop rigueur. Merci également aux autres membres du jury, Jean-Marc Alliot, Patrick Amestoy et Michèle Sebag d’avoir accepté d’assister à la présentation et d’évaluer ce travail.

La vie est faite de rencontres et ma vie professionnelle aurait assurément été tout autre si je n’avais croisé le chemin de Jean-Marc Alliot, Nicolas Durand et Pascal Brisset. C’est à eux que je dois d’être passé du côté obscur de l’informatique, celui où on écrit des algorithmes en plus de simplement les utiliser. Je leur dois beaucoup et je ne les en remercierai jamais assez. La gentillesse et la compétence de Pascal nous manquent à tous toujours autant.

Un grand merci à Patrick Amestoy de m’avoir parrainé pour cette habilitation et aux autres membres de l’équipe APO de l’Institut de Recherche en Informatique de Toulouse pour leur accueil chaleureux et leur sympathie sans faille au long des années. Ils ont assurément contribué à mon ouverture au monde de la recherche académique et à son exigence de qualité. Je remercie également Mathieu Serrurier, de l’équipe ADRIA de l’IRIT, pour nos discussions sur l’apprentissage artificiel et la prévision de trajectoires d’avions.

Je dois mon ancrage dans la réalité opérationnelle, ainsi que le choix de certains de mes sujets de recherche, à mes débuts dans mon autre vie professionnelle, avant de devenir chercheur. Pour cela et pour tous les bons moments que j’ai passé avec eux, je tiens à remercier François Collec, Lisette Buard, Christian Kerlan, Yves Lothou, les anciens de l’équipe 4 et mes anciens collègues du Centre Régional de la Navigation Aérienne Ouest, ainsi que ceux du Service Technique de la Navigation Aérienne.

Je remercie tous ceux qui ont participé aux travaux présentés dans ce mémoire, en espérant n’oublier personne : Richard Alligier, Géraldine Bebout, Claude Chamayou, Florence Cressent, Kévin Guittet, Yi He, Christophe Hurter, Stéphane Puechmorel, Nicolas Saporito, David Szymanski, Gilles Tabard, Benjamin Tissoires, Charlie Vanaret². Merci également à Bernard Brémond, Alain Hérout, François Huchet et Serge Roux pour la collecte et le traitement des données sans lesquelles rien n’aurait été possible, et à Jean-Paul Imbert, David Paquet et à nouveau Serge Roux, pour le support informatique qu’ils apportent chaque jour à nos activités de recherche, support mis en place par Bruno Kriner.

Merci plus particulièrement à Richard Alligier pour ces dernières années de collaboration fructueuse, pour sa curiosité d’esprit et ses soupirs désappointés lorsqu’il n’obtient “que” 45% d’amélioration dans ses résultats.

2. Merci notamment à Charlie pour son inventivité sans cesse renouvelée à me rappeler mon grand âge...

Un merci en forme de “spéciale dédicace”³ à Kévin Guittet d’une part pour m’avoir relancé sur le sujet de la charge de travail et des ouvertures de secteurs, mais aussi pour sa tournure d’esprit, son humour et ses choix circonstanciés de T-shirts à message⁴, de surnoms et d’acronymes.

Mes remerciements vont également à mes autres collègues, actuels ou anciens et que je n’ai pas encore cités, de l’Ecole Nationale de l’Aviation Civile, du Centre d’Etudes de la Navigation Aérienne ou de ses avatars ultérieurs (SDER, DTI/R&D). Merci notamment à Cyril Allignol, Nicolas Archambault, Nicolas Barnier, Eric Blond, Christian Bontemps, Alexandre Gondran, Géraud Granger, Jean-Baptiste Gotteland, Nathalie Lenoir, Pascal Lezaud, Estelle Malavolti, Thomas Rivière, et tous ceux que je ne mentionne pas ici pour ne pas rallonger encore ce document déjà fort long, mais que je remercie pour leur convivialité au quotidien.

Merci surtout à mes amis, ma famille, tout particulièrement à mes filles Typhaine et Auriane et à mon épouse Marie-Pierre dont la patience a été mise à rude épreuve durant la rédaction de ce document. Merci de m’avoir soutenu, et d’avoir supporté aussi longtemps ma sempiternelle excuse : “après mon HDR”! Désormais, je n’ai plus aucune excuse.

3. Ou de cadeau Bonux, pour les plus anciens d’entre nous.

4. T-shirts qu’il a troqués depuis contre un costume plus “corporate”.

Glossaire

ATC Air Traffic Control.

ATM Air Traffic Management.

BADA Base of Aircraft DATA.

CAS Calibrated AirSpeed.

CAUTRA Coordinateur AUTomatique du TRafic Aérien.

CENA Centre d'études de la Navigation Aérienne.

DGAC Direction Générale de l'Aviation Civile.

DSNA Direction des Services de la Navigation Aérienne.

DTI Direction de la Technique et de l'Innovation.

ÉNAC École Nationale de l'Aviation Civile.

ESF Energy Share Factor.

FAA Federal Aviation Administration.

FMS Flight Management System.

IAS Indicated AirSpeed.

ISA International Standard Atmosphere.

LOG Laboratoire d'Optimisation Globale.

NextGen Next Generation Air Transportation System.

OACI Organisation de l'Aviation Civile Internationale.

RMSE Root Mean Square Error.

SESAR Single European Sky ATM Research.

STNA Service Technique de la Navigation Aérienne.

TAS True AirSpeed.

TMA Terminal Manœuvring Area.

Table des matières

Glossaire	v
Table des matières	xii
Introduction	1
1 La gestion du trafic aérien	3
1.1 Introduction	3
1.2 Vocabulaire et unités de mesures	5
1.3 Les missions et les acteurs	5
1.4 Vol à vue et vol aux instruments	6
1.5 Classification des espaces aériens	7
1.6 Organisation de l'espace aérien	7
1.6.1 Régions d'information de vol et blocs fonctionnels d'espace	7
1.6.2 Découpage vertical de l'espace	7
1.6.3 Espaces contrôlés : contrôle en-route, d'approche, d'aérodrome	9
1.6.4 Le réseau de routes aériennes et la sectorisation	9
1.7 Le travail des contrôleurs aériens : la séparation du trafic	11
1.7.1 Normes de séparation, pertes de séparation	11
1.7.2 Détection et résolution des conflits aériens	14
1.7.3 Répartition des tâches entre contrôleurs sur leur poste de travail	15
1.7.4 Les outils du contrôleur	16
1.8 La régulation de la charge de travail des contrôleurs	16
1.8.1 Capacité et demande	17
1.8.2 Charge de travail et complexité ATC	18
1.8.3 La gestion dynamique en temps réel des secteurs aériens	18
1.8.4 La gestion prévisionnelle des ouvertures de secteurs (France et Europe)	19
1.8.5 La gestion des flux de trafic et des capacités	21
1.9 La recherche appliquée, dans le domaine ATM	23
1.9.1 Le contexte français	23
1.9.2 Le contexte international	23
1.10 Thématiques de recherche	24
2 Méthodes d'optimisation, et quelques applications	27
2.1 Résumé des contributions et publications	28
2.2 Introduction	28
2.2.1 Forme générale des problèmes d'optimisation	28

2.2.2	Définitions d'un minimum	29
2.2.3	Caractérisation des méthodes d'optimisation	29
2.2.4	Méthodes d'optimisation employées dans ce mémoire	32
2.3	Méthodes de recherche dans des arbres ou des graphes	33
2.3.1	Principe de la recherche dans des arbres ou des graphes	34
2.3.2	Complétude, optimalité, et complexité des algorithmes	35
2.3.3	Implémentation des algorithmes de recherche	35
2.3.4	Algorithmes de type <i>branch & bound</i>	36
2.3.5	Algorithme A^*	40
2.4	Descente locale sans contraintes, en domaine continu	43
2.4.1	Conditions d'optimalité	43
2.4.2	Principe des méthodes de descente avec <i>line search</i>	44
2.4.3	Choix de la direction de descente	44
2.4.4	Recherche unidimensionnelle pour le choix du pas α_k	45
2.4.5	Propriétés de convergence des méthodes de descente	46
2.5	Métaheuristiques	46
2.5.1	Algorithmes évolutionnaires	47
2.5.2	Métaheuristiques pour l'optimisation en domaine continu	48
2.5.3	Prise en compte des contraintes	50
2.5.4	Métaheuristiques hybrides	51
2.6	Application à quelques problèmes de trafic aérien	51
2.6.1	Réseau de routes-3D séparées	51
2.6.2	Partitionnement optimal de l'espace aérien	52
2.6.3	Résolution de conflits aériens	52
2.7	Conclusion	55
3	Principes et méthodes d'apprentissage supervisé	57
3.1	Notations	58
3.2	Introduction	59
3.3	Notions générales	61
3.3.1	Fonction de perte et minimisation du risque	61
3.3.2	Apprentissage à partir d'un ensemble fini d'exemples	64
3.3.3	La régularisation	66
3.3.4	Le sur-apprentissage	67
3.3.5	Le compromis biais-variance	70
3.3.6	Le fléau de la dimension	71
3.4	Principes d'inférence	72
3.4.1	Choix de représentation	72
3.4.2	Principes d'inférence et choix de la fonction de perte	74
3.4.3	Quel principe d'inférence choisir ?	79
3.5	Cadre méthodologique de la sélection de modèle	80
3.5.1	Estimation empirique de l'erreur de généralisation	81
3.5.2	Les critères d'information	83
3.5.3	Quel méthode de sélection choisir ?	85
3.6	Réduction de la dimensionalité	86
3.6.1	Sélection de variables pertinentes	87
3.6.2	Transformation des variables de base	88

3.7	Les modèles linéaires	92
3.7.1	Les modèles linéaires pour la régression	92
3.7.2	Les modèles linéaires pour la classification	93
3.7.3	Ajustement des poids d'un modèle linéaire	95
3.7.4	Propriétés statistiques	98
3.7.5	Extension à des combinaisons linéaires de fonctions de base	99
3.7.6	Les modèles linéaires généralisés	100
3.8	Réseaux de neurones	101
3.8.1	Le perceptron multi-couches	102
3.8.2	Réseaux de neurones pour la régression	105
3.8.3	Réseaux de neurones pour la classification	107
3.8.4	Calcul du gradient de l'erreur par rétro-propagation	111
3.8.5	Apprentissage du réseau	116
3.8.6	Considérations pratiques sur les réseaux de neurones	118
3.9	Gradient Boosting Machines	119
3.9.1	Ajustement incrémental d'un modèle additif	119
3.9.2	Principe général du <i>gradient boosting</i>	120
3.9.3	<i>Gradient tree boosting</i>	120
3.10	Conclusion	124
4	Application à la gestion prévisionnelle de l'espace aérien	125
4.1	Chronologie et principales publications associées	126
4.2	Travaux antérieurs	127
4.3	Problématique	127
4.4	Travaux préliminaires (avant 2002)	130
4.5	Recherche prospective (2006-2008)	136
4.6	Maquette d'IHM et valorisation (2009-2011)	144
4.7	Discussion et conclusion	150
5	Application aux trajectoires d'avions	153
5.1	Chronologie et principales publications associées	154
5.2	Résumé des travaux réalisés	155
5.2.1	Estimation du vent à partir des trajectoires radar	155
5.2.2	Prévision des trajectoires d'avions en montée	156
5.3	Données disponibles	160
5.3.1	Les différents types de données	160
5.3.2	Deux jeux de données : Mode-C et Mode-S	161
5.3.3	Difficultés du travail sur données réelles	162
5.4	Estimation du vent à partir des trajectoires radar	162
5.4.1	Contexte et travaux antérieurs	162
5.4.2	Relation entre vent et vitesses des avions	164
5.4.3	Formulation du problème d'extraction de vent	164
5.4.4	Résolution par une méthode de moindres carrés	166
5.4.5	Procédure de calcul, critères de qualité	166
5.4.6	Résultats	168
5.4.7	Conclusion sur l'extraction du vent	168
5.5	Prévision de trajectoires d'avions en montée	169

5.5.1	Contexte et travaux antérieurs	169
5.5.2	Le modèle à énergie totale	171
5.5.3	La problématique de la prédiction par les systèmes au sol	173
5.5.4	Estimation de la masse	175
5.5.5	L'apprentissage appliqué à la prévision de trajectoires d'avions	183
5.5.6	Cadre expérimental pour les méthodes d'apprentissage	185
5.5.7	Apprentissage direct de l'altitude future	189
5.5.8	Apprentissage de la masse	192
5.5.9	Apprentissage de la commande de poussée	192
5.5.10	Apprentissage des vitesses-cibles CAS et Mach	196
5.5.11	Résultats comparatifs pour la prévision de l'altitude	198
5.5.12	Généralisation à différents types d'avions et différentes altitudes initiales	199
5.6	Conclusion sur l'exploitation des trajectoires d'avion	201
Conclusion		203
Appendices		205
A Méthodes BFGS et L-BFGS		207
A.1	Principe général des méthodes de Quasi-Newton	207
A.1.1	Approximations quadratiques au voisinage d'un optimum local	208
A.1.2	Approximation itérative de la hessienne ou de son inverse	209
A.2	La méthode BFGS	210
A.2.1	Choix de la matrice initiale H_0	211
A.2.2	Critère d'arrêt	211
A.2.3	Recherche unidimensionnelle (<i>line search</i>)	211
A.2.4	Implantation et complexité de l'algorithme BFGS	212
A.3	<i>Limited-memory</i> BFGS	212
B Théorie de l'information et sélection de modèles		215
B.1	La divergence de Kullback-Leibler	216
B.1.1	Définition	216
B.1.2	Relations avec la log-vraisemblance	216
B.1.3	Lien avec l'information de Fisher et le ratio de vraisemblance	218
B.2	Les critères asymptotiques d'information	220
B.2.1	Principe général	220
B.2.2	Le critère AIC d'Akaike	221
B.2.3	Le critère BIC (<i>Bayesian Information Criterion</i>)	224
C Géodésie		229
C.1	Introduction	229
C.2	Le modèle sphérique	229
C.2.1	Le référentiel lié à la Terre (ECEF)	229
C.2.2	Le référentiel horizontal local (NED)	230
C.2.3	Coordonnées d'un point	230
C.2.4	Vitesse d'un point par rapport au référentiel terrestre	231
C.3	L'ellipsoïde de révolution	232

C.3.1	Définitions de différentes latitudes	232
C.3.2	Quelques caractéristiques utiles de l'ellipse	234
C.3.3	Référentiel ECEF et repères mobiles pour l'ellipsoïde de révolution . . .	235
C.3.4	Coordonnées d'un point	236
C.3.5	Vitesse d'un point par rapport au référentiel terrestre	237
D	Atmosphères ISA et non-ISA pour l'altimétrie	241
D.1	L'atmosphère standard ISA	241
D.1.1	Équation hydrostatique	241
D.1.2	Équation des gaz parfaits	241
D.1.3	Altitude géopotentielle	241
D.1.4	Caractéristiques physiques de l'atmosphère au niveau de la mer	242
D.1.5	Température et gradient vertical de température	242
D.1.6	Pression atmosphérique	243
D.1.7	Altitude géopotentielle en fonction de la pression	243
D.1.8	Densité de l'air	243
D.1.9	Vitesse du son	244
D.2	Les atmosphères non-ISA pour l'altimétrie	244
D.2.1	Les atmosphères non standard	244
D.2.2	Notion d'altitude-pression	244
D.2.3	Équilibre hydrostatique, loi des gaz parfaits, humidité	244
D.2.4	Hypothèses sur le gradient de température	245
D.2.5	Température et "température ISA"	245
D.2.6	Caractéristiques des surfaces de référence	246
D.2.7	Relation entre altitude géopotentielle et d'altitude-pression	246
D.2.8	Pression p en fonction de l'altitude-pression H_p	247
D.2.9	Altitude-pression H_p en fonction de la pression p	247
D.2.10	Densité de l'air ρ en fonction de l'altitude-pression H_p	248
D.2.11	Vitesse du son	248
E	Altimétrie	249
E.1	Moyens de mesure d'altitude à bord de l'avion	249
E.2	Rappel des différentes notions d'altitude	249
E.3	L'altimètre barométrique	250
E.4	Le calage altimétrique en aviation	250
F	La vitesse de l'avion	251
F.1	Moyens de mesure	251
F.2	Quelques définitions de la vitesse	251
F.3	Profils de vitesse	252
G	Équations du modèle avion	255
G.1	Notations	255
G.2	La dynamique du vol	258
G.2.1	Le modèle point-masse	258
G.2.2	Référentiels, et approximation de la terre plate	259
G.2.3	Changements de référentiel	261

G.2.4	Définition de quelques angles	262
G.2.5	Influence du vent	263
G.2.6	Expression des forces dans des référentiels choisis	267
G.2.7	Accélération inertielle dans un référentiel mobile	269
G.2.8	Les équations de la dynamique, en tenant compte du vent	270
G.2.9	Hypothèses simplificatrices usuelles	273
G.3	Modélisation des forces s'appliquant à l'avion	274
G.3.1	Les forces aérodynamiques	274
G.3.2	La poussée des moteurs	275
G.3.3	Le poids de l'avion	277
G.4	La cinématique de l'avion	277
G.4.1	Représentation dans l'espace d'état	277
G.4.2	Les équations du mouvement de l'avion	279
G.4.3	Équations avec la route vraie Ψ_i et l'altitude-pression H_p	281
G.5	Calcul de la trajectoire	285

Introduction

Ce document récapitule une dizaine d'années de recherche appliquée, autour de thématiques concernant la gestion du trafic aérien. Cette activité s'est déroulée dans un contexte professionnel et institutionnel en constante mutation, ce qui n'a pas été sans conséquences sur les travaux que j'ai menés ou encadrés au cours de ces années. La poursuite de cette activité, motivée par des problématiques concrètes de la gestion du trafic aérien, s'est parfois apparentée à une nage à contre-courant dans un océan agité.

La suite de ce mémoire est organisée en cinq chapitres principaux. Le chapitre 1 présente le système actuel de gestion du trafic aérien, et décrit dans la dernière section les thématiques de recherche autour de l'Air Traffic Management (ATM) abordées dans la suite de ce document. Le chapitre 2 présente les méthodes d'optimisation utilisées et quelques applications à des problèmes de trafic aérien, dont notamment la résolution de conflits aériens. Le chapitre 3 est consacré aux principes et aux méthodes de l'apprentissage supervisé. Les deux chapitres suivants sont consacrés à l'application de ces méthodes d'optimisation et d'apprentissage à des problèmes de trafic aérien, en se focalisant sur la gestion prévisionnelle de l'espace aérien dans le chapitre 4, et sur des thématiques autour de la trajectoire de l'avion, dans le chapitre 5.

Les nombreuses annexes regroupent des descriptions plus détaillées, comme celles des méthodes de Quasi-Newton en annexe A, des critères d'information pour la sélection de modèle, en annexe B, et des différents modèles utiles au calcul de trajectoires d'avions : quelques notions de géodésie en annexe C, les modèles atmosphériques pour l'altimétrie en annexe D, quelques rappels sur la mesure de l'altitude (annexe E) et de la vitesse de l'avion (annexe F), et les équations du mouvement de l'avion, dans l'annexe G.

L'ensemble du document (mémoire et annexes) est donc relativement long, essentiellement parce qu'il a été rédigé de façon à être lu avec un minimum de pré-requis. Ce parti-pris peut paraître quelque peu extrémiste, pour un mémoire d'habilitation supposé s'adresser à des spécialistes d'un domaine particulier, et qui devrait donc simplement faire référence à un corpus d'ouvrages et de publications bien connus des lecteurs. Dans le cas présent, ce parti-pris se justifie par le fait que les lecteurs potentiels, si tant est qu'il y en ait plus de quelques-uns, sont peu susceptibles d'être spécialistes de tous les domaines abordés ici : optimisation, apprentissage artificiel, et aéronautique ou ATM.

Les différentes parties peuvent toutefois être lues indépendamment. La plupart des notions exposées dans les chapitres 2 et 3 et les annexes A et B seront plus que familières aux spécialistes en optimisation ou en apprentissage, qui pourront donc les survoler rapidement. De même, les lecteurs familiers du milieu ATM et des modèles et équations pour les trajectoires d'avions pourront ignorer la majorité du chapitre 1, et les annexes C à G.

La seule contribution de ce document, en ces matières, est de chercher à rassembler et présenter de façon aussi claire et complète que possible les différentes notions abordées, afin d'éviter au lecteur néophyte la lecture d'ouvrages de référence assez denses, ou de documen-

tations techniques peu claires.

Les contributions apportées par les travaux présentés dans ce mémoire sont rassemblées dans les sections 2.5.4 et 2.6 du chapitre 2, et dans les chapitres 4 et 5. Un rappel et un résumé des publications concernées sont proposés au début de chaque chapitre concerné. Il s'agit d'un travail collectif, réalisé par des gens passionnés sans la compétence et la persévérance desquels il n'aurait pu voir le jour. Merci à eux tous, et bonne lecture !

Chapitre 1

La gestion du trafic aérien

Sommaire

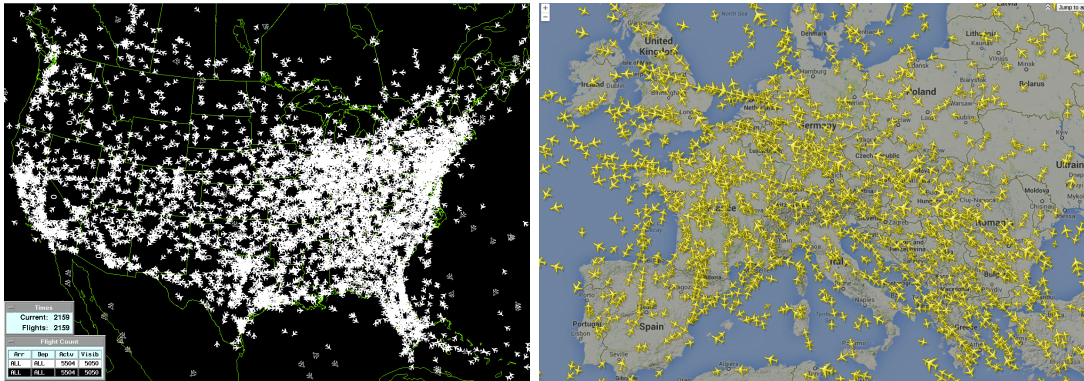
1.1	Introduction	3
1.2	Vocabulaire et unités de mesures	5
1.3	Les missions et les acteurs	5
1.4	Vol à vue et vol aux instruments	6
1.5	Classification des espaces aériens	7
1.6	Organisation de l'espace aérien	7
1.6.1	Régions d'information de vol et blocs fonctionnels d'espace	7
1.6.2	Découpage vertical de l'espace	7
1.6.3	Espaces contrôlés : contrôle en-route, d'approche, d'aérodrome	9
1.6.4	Le réseau de routes aériennes et la sectorisation	9
1.7	Le travail des contrôleurs aériens : la séparation du trafic	11
1.7.1	Normes de séparation, pertes de séparation	11
1.7.2	Détection et résolution des conflits aériens	14
1.7.3	Répartition des tâches entre contrôleurs sur leur poste de travail	15
1.7.4	Les outils du contrôleur	16
1.8	La régulation de la charge de travail des contrôleurs	16
1.8.1	Capacité et demande	17
1.8.2	Charge de travail et complexité ATC	18
1.8.3	La gestion dynamique en temps réel des secteurs aériens	18
1.8.4	La gestion prévisionnelle des ouvertures de secteurs (France et Europe)	19
1.8.5	La gestion des flux de trafic et des capacités	21
1.9	La recherche appliquée, dans le domaine ATM	23
1.9.1	Le contexte français	23
1.9.2	Le contexte international	23
1.10	Thématiques de recherche	24

1.1 Introduction

L'objectif de ce chapitre est de présenter aussi clairement et succinctement que possible le système actuel de gestion du trafic aérien, qu'on abrégera parfois par la suite en utilisant les initiales du terme anglo-saxon ATM. Dans le jargon professionnel, on parle d'ATM pour

faire référence à la gestion du trafic aérien dans sa globalité, et d'Air Traffic Control (ATC) pour l'activité spécifique de contrôle aérien.

Pour des informations plus détaillées, notamment sur le plan réglementaire et sur l'organisation des services, le lecteur pourra se reporter aux documentations [96, 97] de l'Organisation de l'Aviation Civile Internationale (OACI) ou au règlement de la circulation aérienne [149].



Sources FAA pour l'image de gauche, Flightradar24 pour celle de droite.

FIGURE 1.1: Exemples de trafic aérien aux États-Unis (à gauche) et en Europe (à droite).

Ce système ATM/ATC complexe, que nous allons brièvement présenter dans ce chapitre, traite aujourd'hui jusqu'à 33000 vols commerciaux par jour en Europe¹. L'ordre de grandeur est similaire aux États-Unis. La Federal Aviation Administration (FAA) estime le nombre de vols simultanément pris en charge par le NAS (National Airspace System) aux États-Unis entre 4000 et 6000² en heure de pointe. La figure 1.1 illustre le nombre d'avions en vol au-dessus des deux continents.

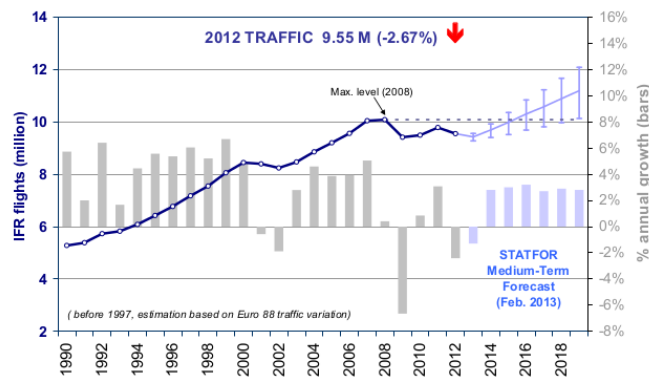


FIGURE 1.2: Évolution du trafic en Europe.

En termes d'évolutions, le trafic aérien est en rapide augmentation dans la région Asie-Pacifique. L'Amérique du Nord et l'Europe voient un tassement de la croissance du trafic,

1. <http://www.sesarju.eu/discover-sesar/why-sesar>

2. <http://www.fly.faa.gov/Products/Information/information.html>

mais les schémas prévisionnels tablent toujours sur une augmentation (voir Figure 1.2).

1.2 Vocabulaire et unités de mesures

Le milieu de l'aviation utilise un vocabulaire et des unités de mesures bien spécifiques, que nous allons introduire ici, avant d'entrer dans la description du système de gestion du trafic aérien. Un glossaire des termes et acronymes est également disponible au début de ce document.

Les altitudes sont exprimées en pieds (ft), ou en niveaux de vol (FL, pour *Flight Level*), avec 1FL = 100ft. L'altitude la plus utilisée dans le milieu de l'aviation est en fait une *altitude pression géopotentielle* (voir annexes D et E) déterminée à l'aide de la pression statique p mesurée à bord de l'avion. Les niveaux de vols sont définis par rapport à une référence qui est la surface isobare 1013,25 hPa.

Les distances sont exprimées en miles nautiques (NM), 1 mile nautique valant 1852 mètres. Les vitesses sont exprimées en noeuds (kts), avec 1 kts qui équivaut à 1 NM/h.

La vitesse de l'avion dans l'air est mesurée par des capteurs de pression dynamique (voir annexe F). La *true airspeed* (TAS) est la vitesse de l'avion dans la masse d'air. La *calibrated airspeed* est équivalente à la *true airspeed* qui serait nécessaire au niveau moyen de la mer, en conditions standard d'atmosphère, pour avoir la même pression dynamique que celle mesurée à bord de l'avion. Si l'on néglige les erreurs d'instruments la *calibrated airspeed* (CAS) est la vitesse lue par le pilote sur ses instruments de bord. Le nombre de Mach est le ratio entre la *true airspeed* (TAS) et la vitesse du son dans l'air. La CAS et le nombre de Mach sont les deux vitesses utilisées par le pilote pour mettre en œuvre son appareil, en suivant par exemple des segments à CAS constante, puis à Mach constant, durant la montée.

L'avion évolue dans une masse d'air elle-même en déplacement par rapport à la surface du globe terrestre. On définit donc également une vitesse par rapport au sol (*ground speed*), exprimée en noeuds.

1.3 Les missions et les acteurs

La mission du système actuel de gestion du trafic aérien est d'assurer un écoulement fluide et sûr du trafic, que ce soit en l'air ou sur les surfaces de roulage des aéroports. Cette mission est assurée par divers organismes qui fournissent un ensemble de services aux usagers de l'espace aérien.

Les usagers peuvent être de natures diverses : circulation aérienne générale, avec des vols commerciaux, des vols de tourisme, des missions de relevés géographiques, de l'aéromodélisme, des drones, ou bien circulation aérienne militaire, avec des vols d'entraînement, des missions spécifiques, etc.

Les services de la circulation aérienne fournis à ces usagers sont de différents types ([148]) :

1. le service du contrôle de la circulation aérienne, qui a pour objet :

- d'empêcher les collisions entre aéronefs ;
- d'empêcher les collisions entre les aéronefs sur l'aire de manœuvre et les obstacles se trouvant sur cette aire ;
- d'accélérer et d'ordonner la circulation aérienne.

2. le service d'information de vol, qui a pour objet de fournir les avis et renseignements utiles à l'exécution sûre et efficace des vols ;
3. le service d'alerte, dont l'objet est d'alerter les organismes appropriés lorsque les aéronefs ont besoin de l'aide des organismes de recherches et de sauvetage, et de prêter à ces organismes le concours nécessaire.

Une différence essentielle entre “service du contrôle” et “service d'information de vol” est liée à la responsabilité de la séparation du trafic. Dans le cadre du contrôle, c'est l'organisme chargé du contrôle aérien, et donc *in fine* le contrôleur sur son poste de travail, qui a la responsabilité d'éviter les collisions entre aéronefs. Pour cela, il donne des instructions de changement de cap ou d'altitude, ou de régulation en vitesse, que les avions doivent impérativement suivre. Lorsque seul le service d'information de vol est fourni, ce sont les pilotes des avions qui ont la responsabilité d'éviter les collisions avec le trafic environnant.

Ces différents services sont fournis aux usagers par des prestataires de services que sont les “ANSP” (*Air Navigation Service Providers*). En France, c'est la Direction des Services de la Navigation Aérienne (DSNA) qui est chargée, sous la tutelle de la Direction Générale de l'Aviation Civile (DGAC), de fournir les services de la circulation aérienne aux aéronefs. Parmi les autres acteurs intervenant dans le système ou interagissant avec lui, on trouve les compagnies aériennes qui opèrent les vols commerciaux dans l'espace aérien, l'armée pour la circulation aérienne militaire, les services de météorologie.

Afin d'éviter des surcharges dans certaines portions de l'espace aérien ou sur certains aéroports, il est de plus en plus nécessaire de réguler et d'organiser le trafic aérien sur une grande échelle. Ce rôle de *Network Manager*³ est aujourd'hui assuré en Europe par un service d'Eurocontrol, une agence européenne inter-étatique. Des organismes similaires existent dans la plupart des autres régions du monde où le trafic est suffisamment dense.

1.4 Vol à vue et vol aux instruments

Les vols peuvent être classés en deux catégories, selon le niveau d'équipement des appareils et les qualifications des pilotes : vol à vue (VFR pour *Visual Flight Rules*), ou vol aux instruments (IFR pour *Instrument Flight Rules*).

Dans le vol à vue (VFR), le pilote doit se maintenir à une certaine distance des obstacles et des nuages et ne voler que dans des conditions météorologiques compatibles avec le vol à vue, notamment en ce qui concerne la visibilité. C'est un type de vol plus adapté à l'aviation de tourisme, où la règle de base pour la séparation entre avions et l'évitement d'obstacle est “voir et éviter”.

Les vols aux instruments (IFR) sont moins contraints par les conditions météorologiques. Du fait qu'ils peuvent voler en conditions météorologiques dégradées, avec une visibilité réduite, les vols IFR bénéficient en règle générale du service de contrôle aérien, qui assure leur séparation avec les autres vols (IFR ou VFR).

3. Le *Network Manager* d'Eurocontrol est l'héritier direct de la CFMU (Central Flow Management Unit) qui gérait en Europe les allocations de créneaux de décollage et de la gestion des flux de trafic

1.5 Classification des espaces aériens

Les espaces aériens sont catégorisés en plusieurs classes (de A à G) qui permettent de distinguer quels services sont fournis à quels types de vols.

Par exemple, dans les espaces de classe A, seuls les vols aux instruments (IFR) sont admis, le service de contrôle est fourni à tous les vols IFR, seuls autorisés, et la séparation est assurée entre tous.

Dans les espaces de classe B, les vols aux instruments (IFR) et à vue (VFR) sont admis ; un service de contrôle est fourni à tous les vols et la séparation est assurée entre tous.

Dans les espaces de classe C, les vols aux instruments (IFR) et à vue (VFR) sont admis ; la séparation n'est assurée qu'entre vols IFR/IFR ou IFR/VFR. Les vols VFR sont séparés des vols IFR et reçoivent des informations de trafic relatives aux autres vols VFR.

On peut trouver les définitions des autres classes d'espaces dans [148], jusqu'à la classe G où seul le service d'information de vol est fourni, pour les vols qui le demandent.

1.6 Organisation de l'espace aérien

1.6.1 Régions d'information de vol et blocs fonctionnels d'espace

L'unité la plus grande pour le découpage de l'espace aérien en unités distinctes est la FIR (*Flight Information Region*). Une FIR est une région d'espace dans laquelle sont fournis les services d'information de vol et d'alerte. Toute portion de l'atmosphère appartient à une FIR. Certains pays du monde voient tout leur espace aérien constituer une unique FIR, voire pour les plus petits, être englobé dans une FIR dépassant leurs frontières. D'autres pays ont divisé leur espace territorial en plusieurs FIRs. C'est le cas de la France, où l'espace est divisé en cinq FIRs.

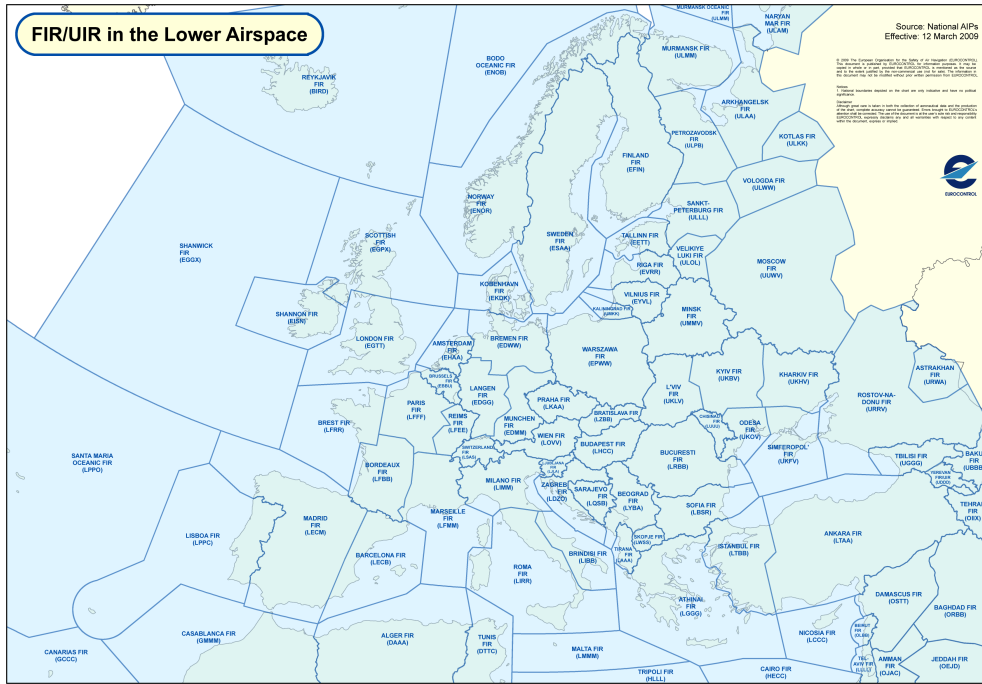
La taille et les limites des FIR résultent de choix faits par les administrations des pays concernés. Certaines FIR sont coupées en deux verticalement. Dans ce cas, la partie inférieure garde le nom de FIR, et la partie supérieure devient une UIR (*Upper Information Region*).

La figure 1.3 montre le découpage de l'espace aérien européen en FIRs.

Au niveau européen, les efforts d'harmonisation des procédures et des moyens du contrôle aérien à travers un package législatif "*Single European Sky*" ont amené à constituer des blocs fonctionnels d'espace (FAB *Functional Airspace Blocks*) regroupant les FIRs de plusieurs espaces aériens nationaux. L'objectif de cette initiative est de fournir dans l'espace européen des services ATM (*Air Traffic Management*) plus homogènes, moins dépendant des frontières nationales. La figure 1.4 montre les blocs fonctionnels d'espace définis en Europe. A l'heure actuelle, ces FABs se traduisent essentiellement par une coopération renforcées des différents ANSPs (*Air Navigation Services Providers*) nationaux.

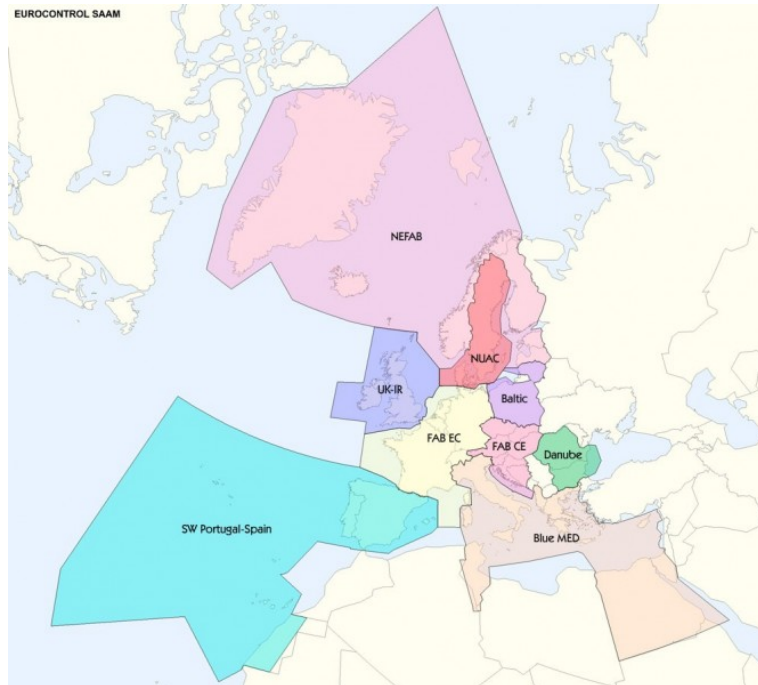
1.6.2 Découpage vertical de l'espace

En Europe, l'espace aérien est découpé verticalement, en un espace inférieur et un espace supérieur. La frontière entre les deux se situe généralement au niveau de vol (ou FL pour *Flight Level*) 195, c'est-à-dire à 19500 pieds d'altitude-pressure au-dessus de l'isobare 1013,25 hPa. Il peut cependant y avoir des variations selon les régions. Par exemple, L'UIR (*Upper Information Region*) du centre de Maastricht qui contrôle l'espace au dessus de la Belgique, du Luxembourg, et du Nord-Ouest de l'Allemagne, commence au FL 245.



Source Eurocontrol

FIGURE 1.3: FIRs en Europe



Source Eurocontrol

FIGURE 1.4: Les FAB (blocs fonctionnels d'espace) en Europe.

Aux États-Unis, il n'y a pas à proprement parler d'UIR, mais les secteurs aériens de l'espace supérieur *en-route* commencent typiquement au FL 240.

En France, comme nous l'avons vu au tout début de cette section (voir Figure 1.3), l'espace aérien inférieur, en-dessous du FL 195, est découpé horizontalement en cinq FIRs (*Flight Information Region*), gérées chacune par un centre de contrôle régional. Dans l'espace supérieur français, il n'est défini qu'une seule région, l'UIR France, même si en pratique les centres régionaux gèrent chacun la portion d'espace supérieur située au-dessus ou à proximité de leur FIR.

1.6.3 Espaces contrôlés : contrôle en-route, d'approche, d'aérodrome

Dans les espaces aériens où sont fournis les services du contrôle, certains volumes d'espace sont plutôt dédiés soit au contrôle des avions autour des aéroports, soit au contrôle des avions en-route entre leur aéroport de départ et celui de destination.

Chaque aéroport est englobé dans une zone de contrôle (CTR) dédiée à la circulation aérienne à proximité immédiate de la piste. Au-dessus de la CTR, et dans avec un rayon plus large autour de l'aéroport, on trouve une autre zone dédiée aux manoeuvres et procédures d'approche ou de départ de l'aéroport. Une telle zone d'approche peut éventuellement concerner plusieurs aéroports, notamment autour des grandes villes. On parle alors de TMA (Terminal Manoeuvring Area, en Europe) ou de TCA (Terminal Control Area, aux USA). La figure 1.5 montre une vue de dessus et une vue en coupe de la TMA Paris, avec les classes d'espaces associées aux volumes d'espace.

Les zones d'approche comme la TMA sont des espaces de transition entre les aéroports et le réseau de route aériennes (*airways*) défini dans l'espace *en-route*. Au final, on distingue donc trois types de contrôle aérien, selon le volume d'espace considéré (voir [149]) :

- le contrôle en route,
- le contrôle d'approche,
- et le contrôle d'aérodrome

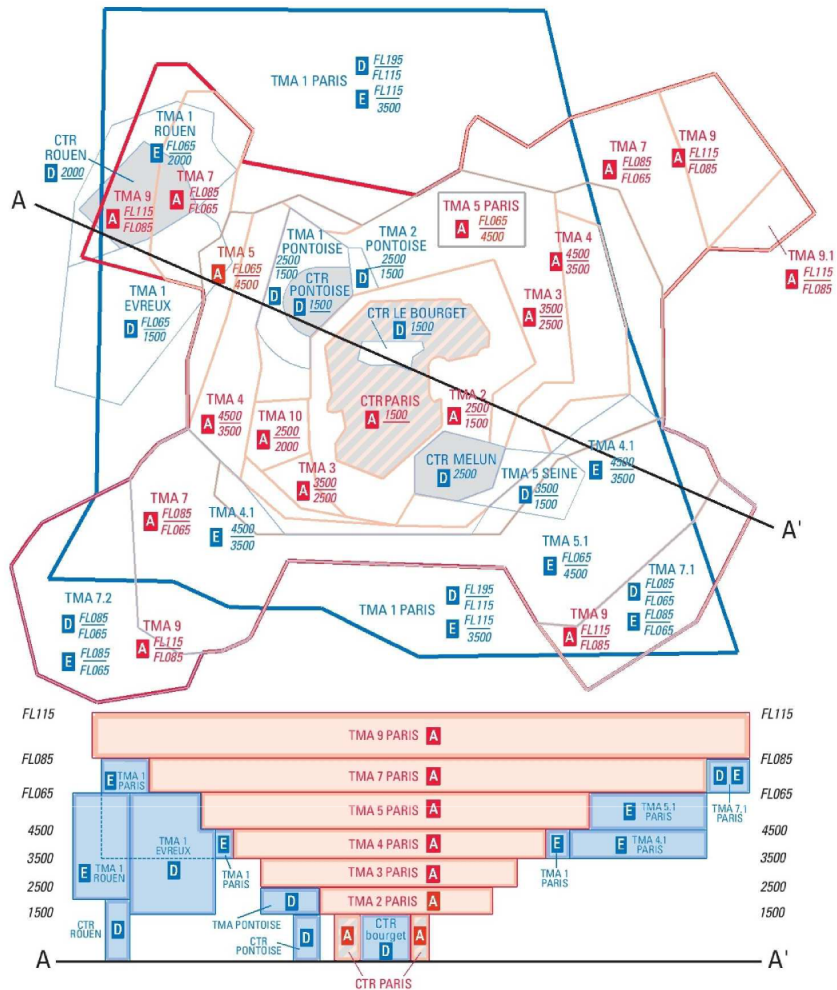
Ces trois types de contrôle sont illustrés sur la figure 1.6, où sont également représentés quelques-uns des moyens techniques de communication, de localisation, ou de radio-navigation indispensables au contrôle aérien. Sont également représentés la vigie d'une tour de contrôle, en charge de la gestion des atterrissages, des décollages, et du roulage au sol des avions, et un centre régional de la navigation aérienne (CRNA), en charge du trafic en-route. Le contrôle d'approche peut quant à lui être localisé sur l'aéroport ou à proximité, pour les plus gros d'entre eux, ou être pris en charge par le centre régional, selon le cas.

1.6.4 Le réseau de routes aériennes et la sectorisation

Les avions volant dans l'espace contrôlé "en route", inférieur ou supérieur, évoluent sur des routes aériennes (*airways*). Le réseau de routes peut être différent selon qu'on est dans l'espace aérien inférieur ou supérieur.

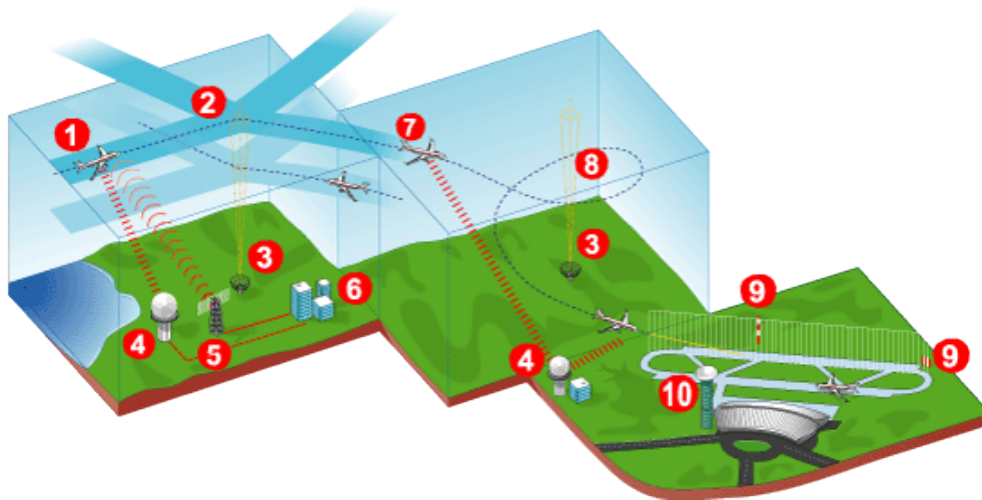
Afin de pouvoir être géré par des opérateurs humains, cet espace est par ailleurs divisé en *secteurs aériens*, qu'on peut voir comme l'unité la plus fine de découpage de l'espace. Ces secteurs sont affectés aux postes de travail des contrôleurs aériens (parfois communément appelés "aiguilleurs du ciel").

La figure 1.7 montre à la fois le réseau de routes aériennes en Europe, et le découpage en secteurs aériens. La figure 1.8 montre quant à elle le réseau de routes aériennes défini dans



Source SIA (Service d'Information Aéronautique)

FIGURE 1.5: Vue de dessus et vue en coupe de la TMA Paris (2011)



Source <http://www.developpement-durable.gouv.fr/Contrôle-aerien,10109.html>

- | | |
|---|--|
| 1 Avion entrant dans un secteur de contrôle | 6 Centre en route de la navigation aérienne |
| 2 Couloir aérien | 7 Avion entrant dans un secteur d'approche |
| 3 Balise radio | 8 Circuit d'attente |
| 4 Radar mono-impulsion | 9 Balise pour l'atterrissage aux instruments (ILS) |
| 5 Émetteur/récepteur radio | 10 Tour de contrôle et radar de contrôle au sol |

FIGURE 1.6: Contrôle en route, contrôle d'approche, et contrôle d'aérodrome

l'UIR France. On y distingue également un certain nombre de zones (TSA, zones D, etc) qui, lorsqu'elles sont actives, sont réservées aux vols militaires d'entraînement ou à d'autres activités de défense aérienne.

L'UIR France est gérée par les cinq centres régionaux français (Aix, Brest, Bordeaux, Paris, Reims).

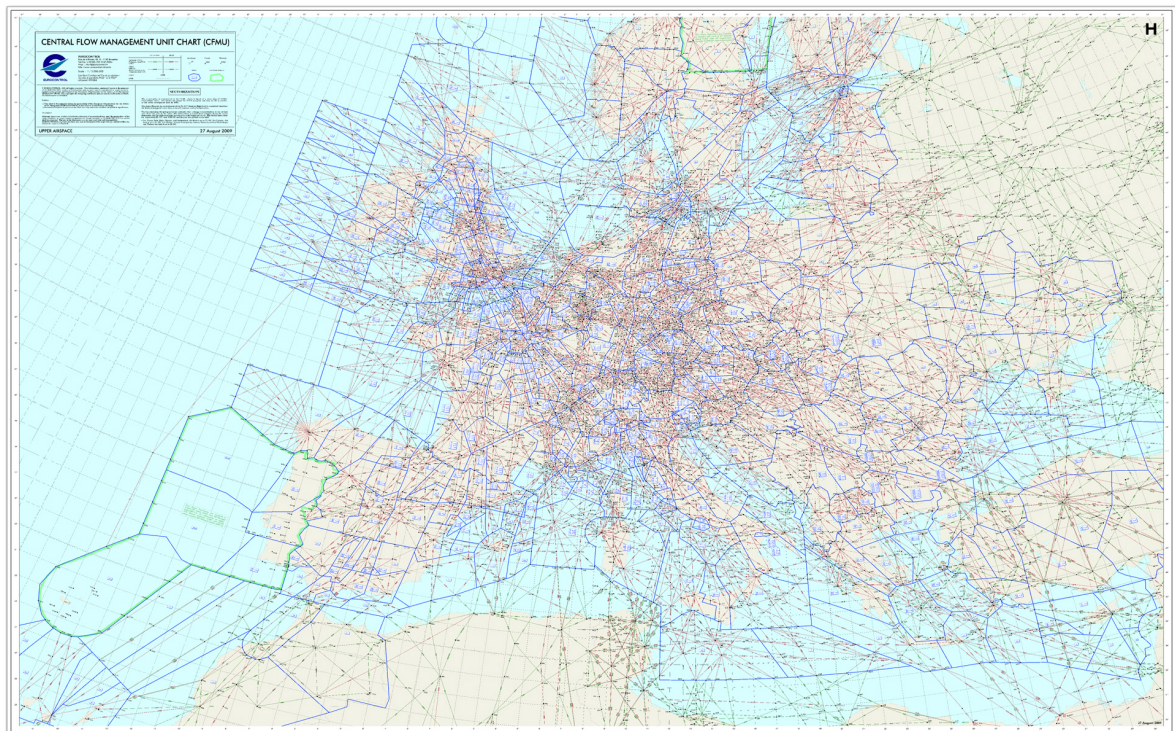
1.7 Le travail des contrôleurs aériens : la séparation du trafic

1.7.1 Normes de séparation, pertes de séparation

Une des tâches essentielles du contrôleur aérien consiste à éviter les collisions entre avions. D'un point de vue réglementaire, il doit s'assurer que les aéronefs maintiennent entre eux une distance minimale, exprimée sous la forme de "séparations standard" ou "normes de séparation". Deux avions doivent toujours être séparés soit latéralement d'au moins une distance δ_ℓ , soit verticalement d'au moins une distance δ_z . Ces deux distances peuvent varier selon le type et le niveau d'équipement de la couverture radar de l'espace aérien considéré.

Typiquement, la norme du minimum de séparation radar dans l'espace en-route européen est de 5 miles nautiques dans le plan horizontal, et de 1000 pieds dans le plan vertical. Ces distances, qui peuvent paraître importantes, tiennent compte des imprécisions des équipements de navigation des avions et des radars, ainsi que des temps de traitement de l'information radar entre la détection proprement dite et l'affichage sur l'écran du contrôleur.

On parle de séparation hors-normes lorsque les distances horizontales et verticale entre deux avions tombent en-dessous de la norme définie réglementairement. En France, une fonc-



Source Eurocontrol

FIGURE 1.7: Routes et secteurs aériens en Europe (2009) dans l'espace supérieur

1.7. LE TRAVAIL DES CONTRÔLEURS AÉRIENS : LA SÉPARATION DU TRAFIC 13



Source SIA (Service d'Information Aéronautique)

FIGURE 1.8: Espace en-route UIR France (2014)

tionnalité “filet de sauvegarde” est définie dans les outils informatiques du contrôleur afin de l’alerter des risques de rapprochement hors-normes. Ces pertes de séparation font l’objet d’un suivi attentif de la part des autorités de la navigation aérienne. La DSNA publie chaque année deux indicateurs (voir [46]) concernant les pertes de séparation : le “HN70” lorsque la séparation est inférieure à 70% de la norme, et le “HN50” lorsqu’elle est inférieure à 50% de la norme. En 2012, le HN70 était de 0.64 pour 100000 vols contrôlé, et il n’y a eu aucune perte de séparation inférieure à 50% de la norme.

La culture de la sécurité est un élément central dans le milieu professionnel de l’Aviation Civile. Lorsqu’ils estiment que la sécurité d’un vol a ou aurait pu être mise en jeu, les agents au sol doivent remplir une fiche de notification d’évènement. Les pilotes, quant à eux, peuvent déposer des comptes-rendus de sécurité, communément appelés “airprox”.

Ces incidents font systématiquement l’objet d’analyses minutieuses, localement et au plan national, afin d’améliorer continuellement les procédures et de garantir la sécurité des vols. Le nombre d’airprox recueillis par la DSNA en 2012 était de 1 pour 100000 vols, pour les vols impliquant au moins un avion IFR et n’impliquant aucun vol militaire, et de 0,3 pour 100000 pour les vols impliquant un avion militaire et un avion civil.

1.7.2 Détection et résolution des conflits aériens

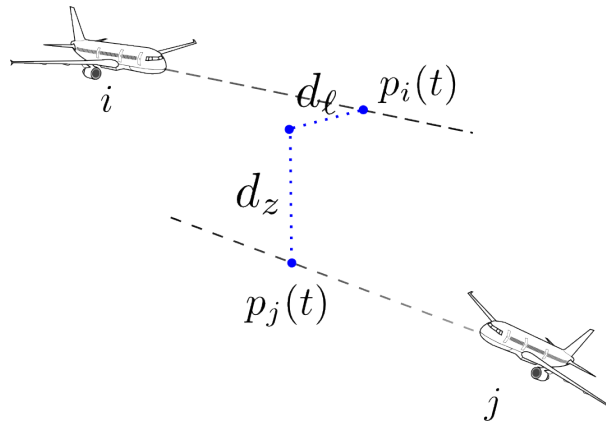


FIGURE 1.9: Conflit aérien

Afin d’éviter les pertes de séparation, les contrôleurs aériens anticipent les trajectoires des avions et donnent des manoeuvres préventives aux avions lorsqu’ils détectent une possible perte de séparation. La notion de *conflit aérien* est donc définie comme une perte de séparation entre deux trajectoires prévues, comme illustré sur la figure 1.9. Formellement, il y a conflit entre deux avions i et j si $\exists t$ tel que :

$$d_l(i, j, t) \leq \delta_l \quad \wedge \quad d_z(i, j, t) \leq \delta_z$$

où δ_l et δ_z sont les normes réglementaires de séparation minimales entre avions (ex. 5 miles nautiques en latéral, 1000 pieds verticalement)

Dans la pratique opérationnelle, cette notion est étendue à plus de deux avions, lorsqu’un avion est en conflit avec un deuxième avion, lui-même en conflit avec un autre, etc. On parle

alors de “conflit à n avions” (voir Figure 1.10), n étant le nombre d’avions impliqués dans la situation conflictuelle. Plus formellement, un certain nombre de publications introduisent la notion de *cluster* (groupe) de conflits comme étant une fermeture transitive de la relation “est en conflit avec”. La notion de conflit reste alors réservée aux pertes potentielles de séparation entre deux avions uniquement, et la notion de *cluster* est utilisée pour les situations conflictuelles impliquant plus de deux avions.

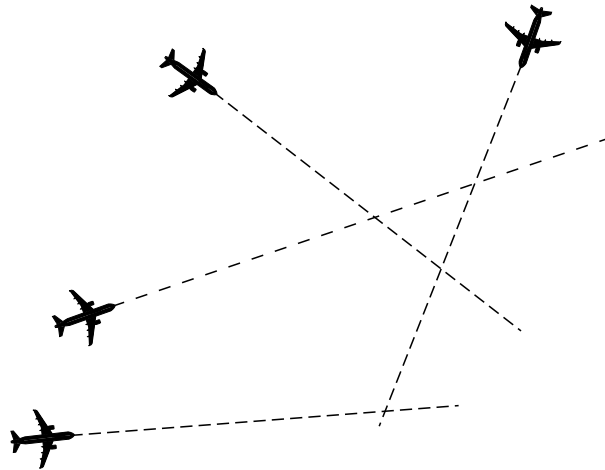


FIGURE 1.10: Conflit à 4 avions (*cluster*).

Les manœuvres données par les contrôleurs pour résoudre les conflits peuvent être de plusieurs types : changement de cap ou d’altitude des avions, stabilisation à un niveau intermédiaire lors d’une montée ou d’une descente, régulation en vitesse, mise en circuit d’attente (à l’approche des aéroports). Aujourd’hui, ces instructions sont transmises aux pilotes par radio.

1.7.3 Répartition des tâches entre contrôleurs sur leur poste de travail

Le contrôleur qui monitorise les avions sur un “écran radar” et qui donne des instructions aux pilotes par radio est appelé “contrôleur radar”. Il est aidé d’un “contrôleur organique” qui pré-détecte les conflits potentiels et coordonne le trafic avec les secteurs adjacents. La nomenclature anglo-saxonne utilise les termes “*radar-controller*” (R-side), “*tactical controller*” ou “*executive controller*” pour le contrôleur radar, et “*data-controller*” (D-side) ou “*planning controller*” pour le contrôleur organique.

En règle générale, les contrôleurs radar et organique opèrent en tandem sur leur poste de travail (voir Figure 1.11), que l’on désigne sous le nom de *position de contrôle*. Avant l’automatisation des transmissions des informations de vol d’un centre à l’autre, une position de contrôle pouvait être armée par trois contrôleurs, un radar, un organique, et un assistant chargé d’aider aux communications téléphoniques.

Avec l’apparition d’outils informatiques d’aide au contrôle, une nouvelle répartition des tâches est envisagée, tant du côté européen dans le cadre du programme Single European Sky ATM Research (SESAR) qu’aux États-Unis dans le cadre du programme Next Generation Air Transportation System (NextGen). Un nouveau rôle, encore à l’étude, est envisagé : celui de *multi-sector planner*, qui pourrait soit jouer un rôle de contrôleur organique pour plusieurs



FIGURE 1.11: Position de contrôle, avec un contrôleur radar et un contrôleur organique

positions radar, soit planifier et organiser le trafic en amont pour plusieurs postes de travail, en s'appuyant sur des prévisions à moyen-terme.

1.7.4 Les outils du contrôleur

Comme illustré sur la figure 1.11, le poste de travail des contrôleurs (position de contrôle) comporte des “écrans radar” permettant d’afficher les positions des avions, la géographie du secteur et les routes principales. On trouve par ailleurs un tableau de “strips”, qui leur permet de manipuler des bandelettes de papier (*strips*) regroupant les informations essentielles relatives à chaque vol. De nombreux pays ont aujourd’hui basculé (ou sont en passe de le faire) vers un système de strips électroniques.

Les autres équipements essentiels sont les chaînes radio et téléphone, le terminal pour la coordination automatique des vols, et les écrans d’affichage des informations météo ou d’autres informations utiles.

A chaque secteur d’espace est affectée une fréquence radio. Les fréquences radio est les lignes téléphoniques peuvent être affectées dynamiquement aux positions de contrôle, ce qui permet de regrouper plusieurs secteurs d’espace sur un même poste de travail.

1.8 La régulation de la charge de travail des contrôleurs

Un élément crucial pour la sécurité du trafic consiste à s’assurer que la charge de travail des contrôleurs aérien n’excède pas certaines limites acceptables. Ces limites déterminent la capacité du système opérationnel à absorber le trafic dans des conditions données (météo, type de trafic, niveau d’équipement, etc). Dans cette section, nous présentons rapidement les différents mécanismes mis en place pour réguler cette charge de travail, en équilibrant au mieux la capacité du système et la demande de trafic.

Ces mécanismes peuvent être mis en place avec plusieurs horizons temporels. On distingue classiquement la planification stratégique (ex. définition des routes, sectorisation de l’espace), la préparation pré-tactique (planification du tour de service, allocation des créneaux de décollage), et la gestion tactique en temps réel du trafic.

1.8.1 Capacité et demande

Les notions de capacité et de demande sont définies comme suit dans la documentation [97] de l'OACI :

Capacité : *“The maximum number of aircraft that can be accommodated in a given time period by the system or one of its components (throughput).”*

Demande : *“The number of aircraft requesting to use the ATM system in a given time period.”*

Notons que ces définitions peuvent couvrir des quantités très différentes selon le contexte dans lequel elles sont définies. Notamment, la notion de capacité ne sera pas la même selon l'échelle du sous-système considéré (secteur aérien, centre de contrôle, bloc fonctionnel d'espace, aéroport) et la période de temps considérée (ex. 1 heure, 1 journée, ou 1 an). En conséquence, on trouve une multitude de définitions formelles (mathématiques) pour la capacité, selon l'utilisation qui en est faite et le contexte dans lequel elle est utilisée.

Parmi les applications faisant usage de la notion de capacité, citons, à l'horizon “stratégique”, la définition des frontières des secteurs ou des blocs d'espace, ou encore l'évaluation et la planification de la performance des systèmes ATC au niveau régional ou européen. Pour les applications “pré-tactiques”, la notion de capacité est employée pour la planification du tour de service, la prévision des ouvertures de secteurs de contrôle, l'allocation de créneaux de décollage aux avions. En temps réel, on peut aussi vouloir détecter les surcharges de travail des contrôleurs, ou ajuster les capacités aéroportuaires en fonction des conditions météorologiques.

Sur ces quelques exemples, on constate une diversité d'objectifs, d'échelles géographiques, et d'horizons temporels dans l'utilisation de la notion de capacité. Une conséquence de cette diversité est que la définition formelle (mathématique) de cette notion est loin d'être unique et consensuelle.

La même remarque vaut pour la demande de trafic, qu'on peut quantifier par divers indicateurs : nombre d'avions présents dans un volume à l'instant t , flux entrant entre t et $t + \Delta t$, nombre moyen d'avions occupant un volume donné sur un intervalle temporel donné, nombre de plans de vol prévus dans la journée, le mois, ou l'année, nombre de décollages et/ou d'atterrissages prévus sur un aéroport, etc.

Nous avons commencé cette section en parlant de réguler la charge de travail des contrôleurs aériens. Or celle-ci n'est pas explicitement mentionnée dans la définition de la capacité, dont on voit bien malgré tout qu'elle correspond à l'idée d'un “nombre maximum” à ne pas excéder sous peine de surcharger les opérateurs du contrôle aérien. La raison en est simple, et tient à la difficulté de quantifier la charge de travail réelle des contrôleurs. Nous verrons dans la section suivante que certains facteurs liés à la complexité du trafic ou du secteur peuvent influencer sur la charge de travail effective des contrôleurs. La prise en compte de ces facteurs est relativement récente. Par le passé, on s'est contenté d'indicateurs simples permettant d'ajuster la variable “trafic” dans les entrées du système ATC.

De fait, les organismes chargés de réguler les flux de trafic s'appuient sur des “capacités déclarées” (voir définition dans [96]) par les organismes de contrôle aérien, pour réguler certains flux en reroutant des vols ou en imposant des délais au décollage. Cette “capacité déclarée” correspond à un compromis jugé acceptable à la fois en termes de retards au décollage imposés aux opérateurs aériens, et en termes de charge de travail pour les organismes de contrôle.

1.8.2 Charge de travail et complexité ATC

La capacité est censée représentée un seuil à ne pas franchir pour la charge de travail des contrôleurs aériens. Cette dernière peut être définie comme le travail physique et mental effectué par le contrôleur pour contrôler le trafic ([113]). Notons toutefois que dans cette même publication [113], Majumdar et Ochieng affirment que le terme “charge de travail du contrôleur” est sujet à confusions, avec une multitude de définitions, de modèles et de mesures proposées dans la littérature.

Dans la pratique, la charge de travail peut varier considérablement à nombre d’avions égal, en fonction de facteurs dynamiques liés au trafic, ou de facteurs statiques comme l’implantation des routes aériennes ou la géométrie du secteur d’espace.

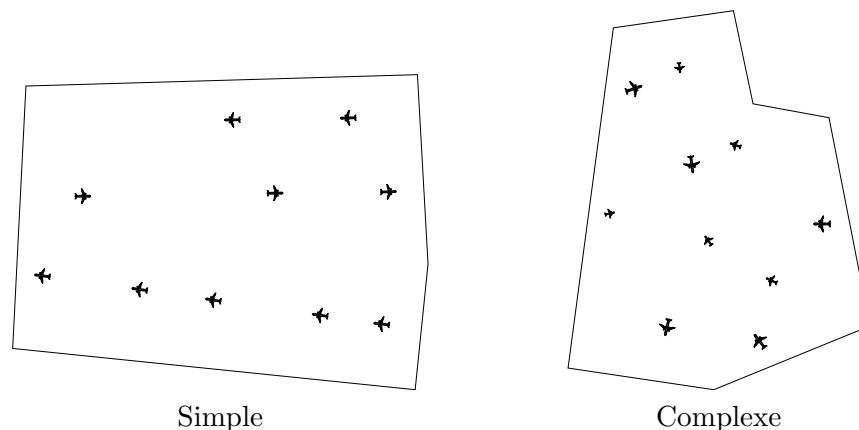


FIGURE 1.12: Une approche intuitive de la complexité ATC

Pour s’en convaincre, considérons la figure 1.12, en supposant que la taille des avions sur ce graphique soit représentative de leur vitesse. Intuitivement, on voit bien que la situation de gauche, avec des avions à la même vitesse évoluant sur des routes qui ne se croisent pas, est plus facile à gérer pour un opérateur humain que la situation de droite, où les avions évoluent à des vitesses différentes avec des trajectoires qui se croisent. Si l’on rajoute en plus la dimension verticale, on peut aussi avoir des avions tous stables à leur niveau de croisière, ou bien en train de monter ou de descendre en croisant un trafic stable.

D’autres facteurs, propres à l’opérateur humain lui-même, ou à son environnement de travail, peuvent influencer le niveau de charge de travail. La figure 1.13, provenant de [122], résume en un schéma les facteurs pouvant affecter la charge de travail du contrôleur. Les auteurs de [122] définissent la complexité du contrôle aérien comme un “concept multidimensionnel incluant des caractéristiques statiques du secteur et des aspects dynamiques du trafic”.

Le lecteur peut se reporter à [87, 122] pour des revues de littérature sur la complexité du contrôle aérien. Nous y reviendrons par ailleurs au chapitre 4.

1.8.3 La gestion dynamique en temps réel des secteurs aériens

Un des moyens pour les contrôleurs aériens de réguler leur charge de travail consiste à jouer sur l’affectation des secteurs aériens aux positions de contrôle. En effet, il est possible

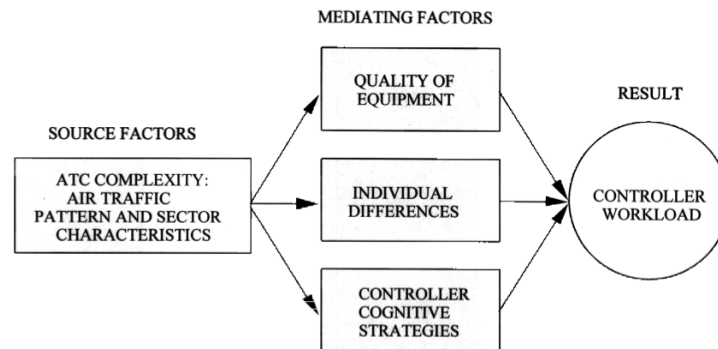


FIGURE FACTORS AFFECTING CONTROLLER WORKLOAD

Source Mogford, R. H., Guttman, J. A., Morrow, S. L., & Kopardekar, P. (1995). *The complexity construct in air traffic control : A review and synthesis of the literature (DOT/FAA/-CT TN95/22)*. Atlantic City, NJ : FAA.

FIGURE 1.13: Facteurs influant sur la charge de travail du contrôleur aérien.

d'affecter un ou plusieurs secteurs d'espace à chaque poste de travail. En supposant qu'un poste de travail gère plusieurs secteurs, il est possible, lorsque la charge de travail augmente, d'ouvrir une autre position de contrôle et d'y affecter une partie des secteurs aériens de la position surchargée, ou bien de transférer ces secteurs à une position déjà ouverte mais moins chargée. A l'inverse, lorsque la charge de travail est faible, on peut regrouper les secteurs de plusieurs positions sur un seul poste de travail.

En supposant qu'on ait découpé l'espace en secteurs suffisamment petit, ce mécanisme de "dégrouperment"⁴/regroupement permet une certaine modularité et une certaine souplesse dans la gestion en temps réel de l'espace aérien. Bien entendu, s'il n'y a qu'un seul secteur affecté à la position de contrôle et s'il est surchargé, il n'est pas possible d'alléger la charge de travail de cette manière.

La solution qui consisterait à découper l'espace en secteurs toujours plus petits, de manière à ce qu'ils ne soient pas surchargés, atteint rapidement une limite : dans un secteur trop petit, le contrôleur radar n'a plus assez de temps ni d'espace pour faire effectuer des manœuvres aux avions, et le contrôleur organique voit sa charge de travail liée à la coordination des vols entrants ou sortants augmenter considérablement.

1.8.4 La gestion prévisionnelle des ouvertures de secteurs (France et Europe)

Afin de mieux comprendre le choix des métriques utilisées jusqu'à récemment pour estimer la capacité du système à gérer un certain volume de trafic, il est utile de remonter dans le temps. Avant 1995, la phase de préparation "pré-tactique" qui avait lieu la veille ou l'avant-veille du jour J consistait simplement à recueillir la liste des vols prévus, et à évaluer la charge de trafic pour chaque tranche horaire ou demi-horaire de la journée à venir. L'ingénieur de permanence opérationnelle sélectionnait les configurations d'ouverture de secteurs compatibles avec le tour de service et qui semblaient les mieux adaptées à la répartition du trafic. Pour chaque secteur de contrôle de chaque configuration d'ouverture, il comparait le

4. Le terme de "dégrouperment" est un néologisme effectivement employé par les contrôleurs aériens.

nombre de vols entrants avec une valeur-seuil (la *capacité* du secteur) pour évaluer les risques de surcharge. Le schéma prévisionnel d'ouverture était alors construit en choisissant la séquence de configurations de secteurs qui semblait la plus adaptée. Ce schéma prévisionnel, ainsi que les surcharges anticipées de trafic, étaient transmises à la CORTA (Cellule d'Organisation et de Régulation du Trafic Aérien), l'organisme chargé de la régulation du trafic français, précurseur de la CFMU (Central Flow Management Unit) européenne créée en 1988 et qui sera pleinement opérationnelle en 1995 avec la mise en place de *Flow Management Positions* (FMP) dans chaque centre de contrôle.

Après 1995, ce sont les opérateurs FMP qui, dans chaque centre de contrôle, sont chargés de la préparation pré-tactique. Ils sont dotés d'outils spécifiques qui permettent d'effectuer informatiquement les tâches précédemment réalisées au moyen d'un listing, d'un papier et d'un crayon. L'IHM d'une version d'études de ce type d'outil informatique est illustrée sur la figure 1.14. Le tableau représentant les ouvertures de secteurs affiche les tranches horaires en colonnes, et les secteurs de contrôle sur les lignes. Chaque case présente deux nombres : le nombre d'avions entrant dans le secteur (à gauche), et la capacité du secteur (à droite).



FIGURE 1.14: Interface SHAMAN pour la visualisation des ouvertures des secteurs de contrôle aérien.

En l'occurrence, la figure 1.14 ne montre pas le schéma prévisionnel d'ouverture des secteurs, mais les ouvertures effectivement réalisées ce jour-là. Les flux entrant sont quant à eux estimés à partir de la demande initiale de trafic, avant régulation. Il est intéressant de regarder de plus près les deux lignes au-dessus du tableau, intitulées "Réalisé" et "Déposé". Ces lignes donnent le nombre de postes de travail effectivement ouverts au cours de la journée (le schéma "réalisé") et le nombre de postes de travail, ou positions de contrôle, qui étaient prévues dans le schéma prévisionnel (celui "déposé" à la CFMU). On constate que le nombre de positions de contrôle prévues est très différent du nombre de positions de contrôle effectivement ouvertes. Par ailleurs, les flux de trafic dans les secteurs réellement ouverts dépassent souvent la valeur-

seuil (capacité) fixée pour ces secteurs. Ce type de comparaison entre la situation prévue et la situation réelle fait immédiatement ressortir la mauvaise qualité du schéma prévisionnel.

Depuis 2011, avec le déploiement des interfaces CHMI/CIFLO pour les FMP, on voit apparaître, dans les outils dédiés à la prévision et la régulation à court-terme, des indicateurs qui n'étaient pas utilisées auparavant, ou alors uniquement pour la gestion en temps réel des secteurs de contrôle. Par ailleurs, plutôt qu'un chiffre unique de capacité, on définit pour chaque secteur plusieurs "monitoring values" pour chaque grandeur représentée. La figure 1.16 illustre ces différentes valeurs-seuils pour l'indicateur "occupancy count".

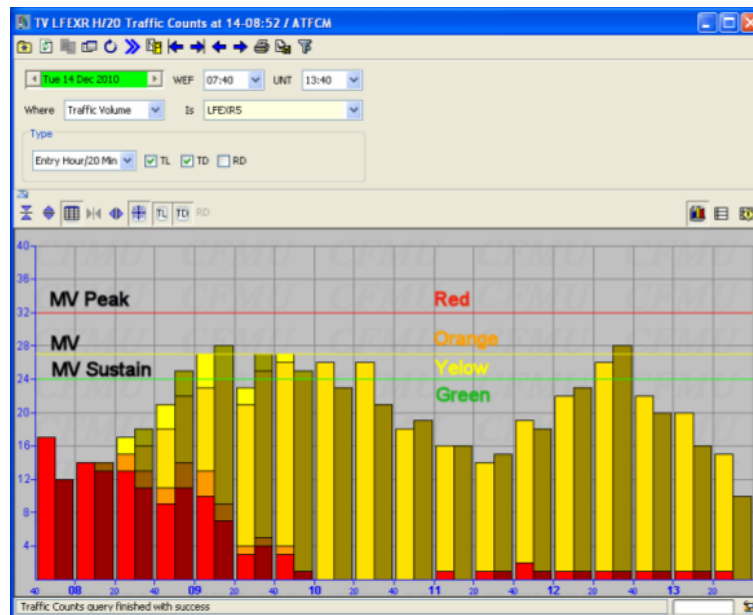


FIGURE 1.15: Les différents types de "Monitoring values" associées à l'indicateur "occupancy count".

Cette évolution récente est sans doute le résultat d'une prise de conscience, par l'ensemble du milieu professionnel, d'une réalité connue depuis longtemps par les contrôleurs aériens, à savoir que les flux entrants et les capacités sont de mauvais indicateurs de la charge de travail réelle. C'est aussi probablement le résultat d'une lente perfusion dans le milieu opérationnel des travaux de recherche autour des notions de complexité ATC, de charge de travail, et de capacité, auxquels contribuent les travaux présentés au chapitre 4.

En ce qui concerne le choix des secteurs de contrôle que l'on prévoit d'ouvrir, la méthode actuelle est purement manuelle : l'opérateur FMP construit son schéma prévisionnel en choisissant parmi des configurations-types prédéfinies. La figure 1.16 donne un exemple de schéma prévisionnel pour le centre de Zagreb (Croatie), construit à partir de telles configurations prédéfinies. C'est à l'opérateur d'estimer quelle configuration est la plus adaptée, en fonction des éléments dont il dispose et de sa propre expérience de situations similaires qu'il a pu rencontrer dans le passé.

1.8.5 La gestion des flux de trafic et des capacités

Lorsqu'on anticipe que les ouvertures de secteurs ne vont pas suffire à équilibrer la charge de travail en évitant les surcharges, il faut mettre en place d'autres types de mesures. Celles-ci

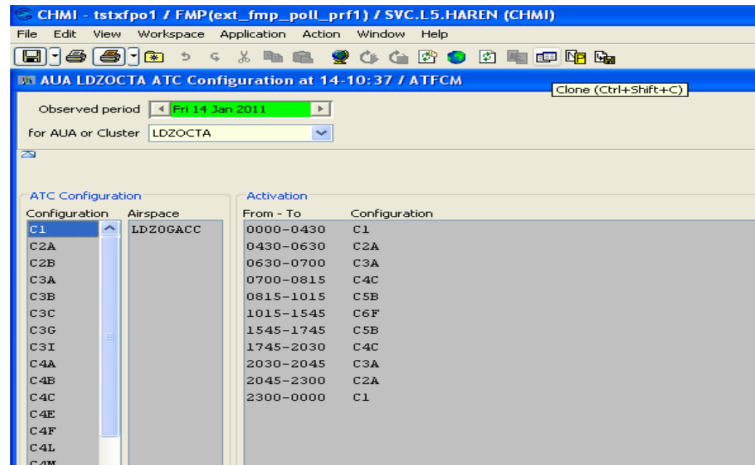


FIGURE 1.16: Interface CHMI/CIFLO pour les FMP (Flow Management Positions).

consistent soit à rerouter des flux de trafic en amont, afin d'éviter le secteur surchargé, soit à retarder les décollages de certains avions pour lisser la charge de trafic dans le secteur.

En général, ces mesures de régulation du trafic doivent être mises en place bien avant l'entrée des vols concernés dans l'espace du centre de contrôle où une surcharge est anticipée. Historiquement, en France, cette régulation du trafic a donc dès le départ été instituée au niveau national plutôt qu'au niveau des centres régionaux. Avec l'augmentation du trafic, le besoin de réguler le trafic s'est généralisé, et c'est rapidement un organisme transnational, Eurocontrol, qui a été chargé des régulations. Au États-Unis, un organisme similaire, l'ATCSCC (*Air Traffic Control Strategic Command Center*) joue ce rôle de coordination et de régulation des vols.

Pour réguler le trafic dans les secteurs surchargés, le *network manager* s'appuie sur les capacités déclarées par les organismes de contrôle. Les capacités des secteurs en-route ou des aéroports peuvent varier au cours du temps, en fonction des aléas météorologiques, de l'activation de zones militaires, de la disponibilité des équipements.

Notons ici quelques différences entre les systèmes en Europe et aux États-Unis. Historiquement en Europe, c'est l'espace en-route qui était le plus saturé, notamment à cause de la fragmentation de l'espace en espaces nationaux, et de l'utilisation simultanée de l'espace par les trafics civils et militaires. Aux États-Unis, les problèmes de saturation concernaient principalement les aéroports et les secteurs d'espace autour des grandes métropoles américaines, avec des capacités beaucoup plus impactées par les phénomènes météorologiques convectifs, notamment sur la côte Est, qu'en Europe de l'Ouest. Ces différences se retrouvent dans les choix des sujets traités par les chercheurs de part et d'autre de l'Atlantique. Depuis les années 1990, et avec l'harmonisation croissante des procédures et des espaces de contrôle européens, ces différences tendent à s'estomper. Aujourd'hui, les aéroports aussi tendent à être saturés en Europe.

1.9 La recherche appliquée, dans le domaine ATM

1.9.1 Le contexte français

La recherche appliquée au trafic aérien est une activité relativement récente au sein des institutions de l'Aviation Civile française. Le développement de systèmes automatisés, avec les diverses versions du Coordinateur AUtomatique du TRafic Aérien (CAUTRA), a longtemps été le sujet central du Centre d'études de la Navigation Aérienne (CENA) qui en a développé les premières versions dans les années 60, puis du Service Technique de la Navigation Aérienne (STNA) qui en a ensuite repris la maîtrise d'ouvrage à partir du CAUTRA 4 vers le milieu des années 70, les développements étant dès lors assurés en partenariat avec des industriels. Après ce transfert, les missions du CENA sont réorientées vers les études à moyen/long terme, notamment sur l'amélioration des outils du contrôle (équipements radars, interfaces homme-machine), la définition de nouveaux systèmes comme l'anti-collision embarquée, les systèmes d'alerte de proximité du sol, et tous types d'études susceptibles d'améliorer le bon fonctionnement du système de gestion du trafic aérien.

Pour autant, la recherche, au sens où on l'entend dans le milieu académique, reste encore longtemps une activité inexistante dans l'administration de la DGAC. A la fin des années 80, celle-ci ne comptait dans ses rangs aucun ingénieur ayant suivi une formation par la recherche dans le cadre de son activité professionnelle. Il faudra attendre les années 90 pour en voir quelques-uns suivre de telles formations, sous l'impulsion de Jean-Marc Garot, chef du CENA à l'époque. La première structure que l'on peut réellement qualifier de laboratoire de recherche apparaît en 1996, avec la création du Laboratoire d'Optimisation Globale (LOG), dirigé par Jean-Marc Alliot, ce laboratoire faisant l'objet d'une convention commune entre le CENA et l'École Nationale de l'Aviation Civile (ÉNAC).

L'activité de recherche au sein de la DGAC a depuis été l'objet de plusieurs réorganisations successives, certaines d'entre elles étant la conséquence de réorganisations globales des services de la DGAC. En 2005, le CENA et le STNA cessent d'exister et sont intégrés dans une nouvelle direction, la Direction de la Technique et de l'Innovation (DTI). Une conséquence indirecte est la dissolution du LOG, qui regroupait un petit nombre de chercheurs, personnels du CENA ou de l'ÉNAC. Dans la nouvelle DTI, l'ancien CENA devient pour quelques temps la Sous-Direction des Etudes et de la Recherche (SDER), qui disparaît en 2006 suite à une nouvelle réorganisation. Le domaine R&D assure alors les missions de recherche et développement de la DTI, avec une implication croissante dans le programme européen SESAR (voir section suivante). La dernière réorganisation, en 2011 voit le domaine R&D disparaître, et être remplacé par un domaine EEI (Etudes européennes et innovation). A cette occasion, les chercheurs titulaires d'une thèse de Doctorat sont transférés à l'ÉNAC, qui regroupe dorénavant l'ensemble des activités de recherche académique appliquée de la DGAC.

1.9.2 Le contexte international

Les programmes SESAR, en Europe, et NextGen aux États-Unis, bien qu'étant parfois présentés comme des programmes de recherche et développement (R&D), sont essentiellement des programmes d'harmonisation et de modernisation des systèmes ATM, à l'échelle continentale. L'objectif est la définition d'un concept opérationnel commun, au niveau mondial, centré sur le concept de trajectoire-4D, et de "business trajectory", et la mise en place d'une gestion plus efficace et plus souple de l'espace aérien, telle que préconisée par la législation du *Single Sky* en Europe, tout en maintenant des objectifs exigeants sur la sécurité des vols.

Sur le plan conceptuel (voir [38] et [156]), la trajectoire-4D serait un volume dans lequel l’avion serait libre de se déplacer, et serait négociée entre les organismes de contrôle et les opérateurs aériens. Le concept ne dit pas par quelles méthodes de telles trajectoires libres, sans conflits, peuvent être calculées.

L’implantation du concept, à travers les phases de développement des programmes SESAR et NextGen, est surtout l’occasion d’obliger les différents partenaires (industriels, compagnies aériennes, fournisseurs de services du contrôle, institutions) à s’associer pour discuter des évolutions du système ATM/ATC, et définir, harmoniser, et développer les systèmes techniques du futur. L’activité de recherche, dans ce contexte, est marginale (ex. *work package E “Innovative research”* de SESAR).

1.10 Thématiques de recherche

Les thématiques de recherche en ATM sont multiples : optimisation du réseau de routes et de la sectorisation, construction des schémas d’ouvertures de secteurs, minimisation des retards aux décollage en respectant les contraintes de capacité, séquençement optimal des arrivées et des départs, résolution de conflits au roulage, prévision des trajectoires d’avions, résolution de conflits aériens, etc.

On perçoit aisément, dans l’énoncé de certains de ces sujets, les problèmes d’optimisation sous-jacents. Pour d’autres, il est peut-être un peu moins évident de distinguer les problèmes d’apprentissage artificiel qu’ils recouvrent, ou dont ils nécessitent la résolution.

Les travaux de recherche que j’ai mené, animé, ou encadrés au cours des dernières années, dans différents services⁵ de la DGAC, portent essentiellement sur les thématiques opérationnelles suivantes :

- la résolution de conflits aériens, abordée dans la section 2.6.3 du chapitre 2 par des métaheuristiques conçues pour l’optimisation en domaine continu (optimisation par essais particuliers, évolution différentielle). Ce sujet n’est cependant pas parmi les thèmes centraux de la recherche présentée dans la suite.
- la gestion prévisionnelle des ouvertures de secteurs aériens. Cette thématique, traitée au chapitre 4, recouvre en fait deux problèmes distincts, avec d’une part la construction d’un modèle fiable de prévision de la charge de travail, et d’autre part la recherche d’une partition optimale de l’espace aérien en secteurs de contrôle, affectés aux postes de travail des contrôleurs. Concernant la charge de travail des contrôleurs, l’originalité de notre démarche aura été d’en construire un modèle en apprenant un réseau de neurones à partir d’indicateurs de complexité du trafic et d’archives d’ouvertures de secteurs. Le partitionnement optimal fait quant à lui appel à des méthodes de recherche dans des arbres ou des graphes, ou à une métaheuristique.
- deux sujets autour de la thématique “trajectoire”, présentés au chapitre 5, avec d’une part l’extraction du vent à partir des enregistrements radars des trajectoires d’avions, et d’autre part la prévision par les systèmes au sol des trajectoires d’avions en montée, à partir d’informations incomplètes. Pour l’extraction du vent, nous avons introduit une modélisation permettant de réduire drastiquement le nombre de variables du problème

5. Ces différents services ou laboratoires sont le LOG, le CENA, la DTI, et le laboratoire MAIAA (Mathématiques Appliquées, Informatique, et Automatique pour l’Aérien) de ÉNAC.

à traiter, et appliqué une méthode de moindres carrés à des trajectoires réelles enregistrées. Pour la prévision de l'altitude des avions en montée, nous avons appliqué des techniques d'apprentissage supervisé pour apprendre les paramètres du modèle physique de l'avion (masse, poussée des moteurs, vitesse-cible), ces paramètres n'étant pas connus des systèmes au sol.

Les méthodes d'optimisation utilisées ou développées durant ces travaux sont présentées au chapitre 2 qui suit, et les principes et méthodes d'apprentissage supervisé au chapitre 3.

Chapitre 2

Méthodes d'optimisation, et quelques applications

Sommaire

2.1	Résumé des contributions et publications	28
2.2	Introduction	28
2.2.1	Forme générale des problèmes d'optimisation	28
2.2.2	Définitions d'un minimum	29
2.2.3	Caractérisation des méthodes d'optimisation	29
2.2.4	Méthodes d'optimisation employées dans ce mémoire	32
2.3	Méthodes de recherche dans des arbres ou des graphes	33
2.3.1	Principe de la recherche dans des arbres ou des graphes	34
2.3.2	Complétude, optimalité, et complexité des algorithmes	35
2.3.3	Implémentation des algorithmes de recherche	35
2.3.4	Algorithmes de type <i>branch & bound</i>	36
2.3.5	Algorithme A^*	40
2.4	Descente locale sans contraintes, en domaine continu	43
2.4.1	Conditions d'optimalité	43
2.4.2	Principe des méthodes de descente avec <i>line search</i>	44
2.4.3	Choix de la direction de descente	44
2.4.4	Recherche unidimensionnelle pour le choix du pas α_k	45
2.4.5	Propriétés de convergence des méthodes de descente	46
2.5	Métaheuristiques	46
2.5.1	Algorithmes évolutionnaires	47
2.5.2	Métaheuristiques pour l'optimisation en domaine continu	48
2.5.3	Prise en compte des contraintes	50
2.5.4	Métaheuristiques hybrides	51
2.6	Application à quelques problèmes de trafic aérien	51
2.6.1	Réseau de routes-3D séparées	51
2.6.2	Partitionnement optimal de l'espace aérien	52
2.6.3	Résolution de conflits aériens	52
2.7	Conclusion	55

Ce chapitre est dédié à la présentation de méthodes d’optimisation, et à leur application à quelques problèmes de trafic aérien. Après une courte introduction sur l’optimisation en général, nous restreindrons le champ à trois types d’approches : les algorithmes de recherche dans des arbres ou des graphes ; les méthodes de descente pour l’optimisation locale sans contraintes en domaine continu ; les métaheuristiques pour les problèmes d’optimisation difficiles, en domaine continu également.

2.1 Résumé des contributions et publications

Les contributions apportées par nos travaux sont succinctement évoquées ou présentées dans la section 2.5.4, pour l’hybridation d’algorithmes évolutionnaires et de méthodes d’intervalles, et dans la section 2.6, pour l’application de méthodes d’optimisation à des problèmes de gestion du trafic aérien, dont notamment la résolution de conflits aériens. Ces travaux ont fait l’objet des publications suivantes :

- 2012

- [165] C. VANARET, D. GIANAZZA, N. DURAND, AND J. GOTTELAND, *Benchmarking Conflict Resolution Algorithms (regular paper)*. Dans : International Conference on Research in Air Transportation (ICRAT 2012), Berkeley, California, 22/05/2012-25/05/2012, ICRAT, (en ligne), mai 2012.
- [166] C. VANARET, D. GIANAZZA, J. GOTTELAND, AND N. DURAND, *Résolution de conflits aériens par un algorithme à évolution différentielle*. Dans : 13e congrès annuel de la Société française de Recherche Opérationnelle et d’Aide à la Décision (ROADEF 2012), Angers, 11/04/2012-13/04/2012.
- [17] J.-M. ALLIOT, N. DURAND, D. GIANAZZA, AND J.-B. GOTTELAND, *Finding and Proving the Optimum : Cooperative Stochastic and Deterministic Search (regular paper)*. Dans : European Conference on Artificial Intelligence (ECAI 2012), Montpellier, France, 27/08/2012-31/08/2012, IOS Press, p. 55-60, septembre 2013.
- [18] J.-M. ALLIOT, J.-B. GOTTELAND, C. VANARET, N. DURAND, AND D. GIANAZZA, *Implementing an interval computation library for OCaml on x86/amd64 architectures (short paper)*. Dans : International Conference on Functional Programming, Copenhagen, Denmark, 10/09/2012-12/09/2012, ACM, (en ligne), septembre 2012.

- 2014

- [15] Chapitre “Application en gestion du trafic aérien”. Dans : Métaheuristiques. Patrick Siarry (Eds.), Eyrolles, 15, p. 413-459, 2014.

2.2 Introduction

2.2.1 Forme générale des problèmes d’optimisation

Dans sa forme la plus générale, un problème de minimisation consiste à trouver, dans un domaine Ω , une valeur x^* minimisant une fonction-objectif $f : \Omega \rightarrow \mathbb{R}$. Les problèmes de

minimisation se formalisent usuellement de la manière suivante :

$$\min_{x \in \Omega} f(x) \tag{2.1}$$

Tout problème de maximisation se ramène facilement à un problème de minimisation, en considérant l'opposé de $f(x)$, et l'on ne considérera donc dans la suite que le cas de la minimisation.

Dans l'équation (2.1), la nature des variables x et du domaine Ω dépend du type de problème d'optimisation traité. En optimisation combinatoire, Ω est un ensemble de valeurs discrètes (ex. chemins dans un graphe), alors qu'en optimisation continue, il s'agira par exemple d'une partie de \mathbb{R}^n , si l'on travaille sur des variables réelles.

Le domaine Ω regroupe l'ensemble des points admissibles. Le domaine admissible peut prendre en compte ou non un certain nombre de contraintes imposées aux solutions, selon qu'on considère un problème d'optimisation sous contraintes, ou sans contrainte.

2.2.2 Définitions d'un minimum

Dans l'idéal, on souhaite trouver un minimum global de la fonction-objectif f , c'est à dire un point répondant à la définition 2.1 suivante.

Définition 2.1 (Minimum global). *Le point $x^* \in \Omega$ est un minimum (ou minimiseur) global de la fonction f si :*

$$\forall x \in \Omega, \quad f(x^*) \leq f(x)$$

Trouver un optimum global peut s'avérer difficile en pratique, selon la nature de la fonction f ou des contraintes définissant le domaine admissible. Il est souvent plus facile d'obtenir un ou plusieurs optima locaux, répondant à la définition 2.2 suivante.

Définition 2.2 (Minimum local). *Le point x^* est un minimum local de la fonction f s'il existe un voisinage $V \subset \Omega$ du point x^* tel que :*

$$\forall x \in V, \quad f(x^*) \leq f(x)$$

On parle de minimum local *strict* si $\forall x \in V$ tel que $x \neq x^*$, $f(x^*) < f(x)$. Un minimum local est dit *isolé* s'il existe un voisinage V de x^* tel que x^* soit le seul minimum local dans V .

Dans le cas particulier où la fonction f est convexe, tout optimum local est également global, et celui-ci est unique (sauf cas dégénéré).

2.2.3 Caractérisation des méthodes d'optimisation

On trouve une très grande variété d'approches et de méthodes pour l'optimisation, et il existe bien des façons, qui se recoupent parfois entre elles, de les caractériser. Très schématiquement, les critères permettant de catégoriser les différentes approches existantes sont liés à trois aspects essentiels : les choix de représentation adoptés lors de la modélisation du problème traité, la nature de l'objectif recherché, et les moyens employés pour y parvenir.

Caractérisation selon les choix de représentation

On peut distinguer différents sous-domaines de l'optimisation selon la nature des variables x , la nature de la fonction-objectif f , et l'existence et la nature des contraintes définissant le domaine Ω .

Selon les cas, la variable x peut être continue, ou discrète, ou un mélange des deux. Le cas classique, pour l'optimisation en domaine continu, consiste à considérer f comme une fonction d'un vecteur $x = (x_1, \dots, x_n)^T$ dont les coordonnées sont des réels. Dans le cas discret, la variable x appartient à un ensemble dénombrable : par exemple celui des vecteurs $x \in \mathbb{N}^n$ de valeurs entières éventuellement restreintes à des valeurs binaires $\{0, 1\}$, ou encore l'ensemble des permutations d'un sous-ensemble de \mathbb{N} , les chemins dans un arbre ou dans un graphe, l'ensemble des partitions d'un ensemble, etc. Certains problèmes peuvent adopter une représentation mixte, combinant des variables réelles et entières par exemple. D'autres choix encore sont possibles, comme dans le cas de l'optimisation stochastique où les variables x du problème sont des variables aléatoires.

La fonction f peut par ailleurs prendre plusieurs formes (linéaire, quadratique, d'expression analytique connue ou non, etc), et vérifier éventuellement un certain nombre de propriétés mathématiques (convexité, continuité, différentiabilité, fonction lipschitzienne ou non, etc) influant sur les conditions d'optimalité et le choix des algorithmes.

Enfin, le domaine Ω des variables x peut être restreint par des contraintes d'appartenance à un ensemble ou un sous-espace donné. Notons que le choix de représentation des variables peut lui-même être vu comme une contrainte : ex. $x \in \mathbb{N}^n$. Classiquement (mais pas obligatoirement), un problème d'optimisation sous contraintes se formalise par l'équation (2.2) suivante, en explicitant les contraintes d'inégalité g_i ou d'égalité h_j .

$$\begin{aligned} \min_{x \in X} f(x) \quad \text{sous les contraintes :} \quad & g_i(x) \leq 0 \quad i = 1, \dots, p \\ & h_j(x) = 0 \quad j = 1, \dots, q \end{aligned} \tag{2.2}$$

L'expression analytique de ces contraintes g_i et h_j peut être connue, ou pas. Tout comme pour la fonction f , elles peuvent vérifier ou non certaines propriétés qui vont influencer les conditions d'optimalité et le choix des méthodes.

De cette multiplicité de cas possibles résulte une variété de termes caractérisant des sous-domaines, ou des méthodes spécifiques d'optimisation :

- optimisation en domaine continu ou discret, ou optimisation stochastique
- optimisation avec ou sans contraintes,
- optimisation de fonctions régulières (*smooth*) ou non,
- optimisation combinatoire,
- optimisation convexe, non convexe, concave sur un domaine convexe, etc.
- programmation linéaire, linéaire en nombre entiers, mixte, non linéaire, quadratique, etc.

Caractérisation selon l'objectif recherché

Les méthodes d'optimisation se différencient également selon qu'on recherche un optimum local ou global, avec ou sans preuve d'optimalité.

Idéalement, lorsqu'on traite un problème d'optimisation, on aimerait trouver avec certitude le ou les optima globaux de la fonction-objectif f . Cet objectif idéal n'est que rarement réalisable en pratique, pour la plupart des problèmes issus du monde réel. Il faut parfois abandonner au moins une de ces deux exigences : globalité, ou certitude d'optimalité.

L'optimisation, en tant que domaine de recherche, trouve ses racines dans les mathématiques, où l'on s'est intéressé en premier lieu à des fonctions convexes pour lesquels les notions de minima locaux ou globaux sont confondues. Avec l'apparition de l'informatique, et la prise en compte de problèmes plus complexes issus du monde réel, le champ de la recherche s'est élargi à l'optimisation, par des méthodes numériques, de fonctions non convexes multimodales. Dans ces cas-là, il est souvent possible d'appliquer tout de même des méthodes numériques initialement prévues pour les fonctions convexes : par exemple une méthode de plus grande pente (*steepest descent*) ou une méthode de Quasi-Newton, pour des fonctions suffisamment régulières. L'optimum trouvé n'est alors pas nécessairement un optimum global, et la qualité de la solution dépend en règle générale du choix d'un point de départ pour la méthode itérative utilisée, comme illustré sur la Figure 2.1. L'ensemble des méthodes développées dans ce cadre est usuellement regroupé sous le label d'*optimisation locale* (voir par exemple [127] pour les aspects théoriques et la description des méthodes numériques d'optimisation locale, avec ou sans contraintes). Notons que, si l'optimalité globale n'est pas garantie, la convergence vers un optimum local est tout de même assurée de façon efficace, sous certaines conditions, par les algorithmes déterministes utilisés en optimisation locale.

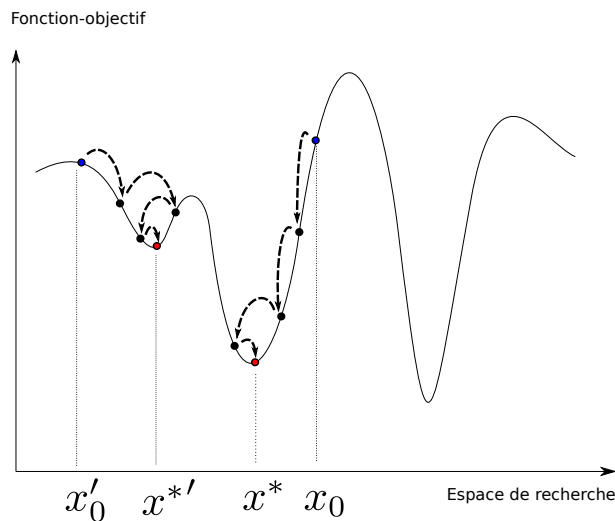


FIGURE 2.1: Illustration de la dépendance des solutions au choix des points initiaux, lors de l'application d'une méthode locale à une fonction non convexe multimodale.

D'autres méthodes, dites d'*optimisation globale*, ont pour objectif spécifique de trouver un optimum global de la fonction-objectif f , dans des cas non convexes. Les conditions d'optimalité globale font appel à l'enveloppe convexe de la fonction f (voir [88]). Dans le cas général, trouver des algorithmes efficaces s'avère bien moins facile que dans le cas convexe. Par exemple, l'optimisation d'une fonction quadratique de forme générale $\frac{1}{2}\mathbf{x}^T H \mathbf{x} + \mathbf{c}^T \mathbf{x}$, sous contraintes linéaires, est un problème NP-difficile [169]. Il existe malgré tout plusieurs classes de problèmes dont les propriétés mathématiques permettent un traitement par des méthodes déterministes fournissant au moins une solution approchée, à défaut d'une solution exacte.

C'est par exemple le cas pour l'optimisation de fonctions concaves sur un domaine convexe, de différences de fonctions convexes, de fonctions quadratiques, etc (voir [92] pour les approches déterministes de l'optimisation globale).

Enfin, des méthodes stochastiques¹ peuvent permettre de traiter des problèmes inaccessibles aux autres approches. Une version basique de ce type de méthode consiste par exemple à tirer au hasard plusieurs points de départ pour une méthode itérative locale (*multi-start local search*). Dans cette catégorie des méthodes itératives stochastiques, les métaheuristiques [157, 15] combinent divers mécanismes pour explorer au mieux l'espace des points admissibles, en introduisant un "biais intelligent" vers de bonnes solutions. L'optimalité globale des solutions trouvées n'est alors pas garantie, ni même souvent l'optimalité locale (pour les versions canoniques de la plupart des métaheuristiques). On dispose parfois de preuves de convergence en probabilité, pour quelques méthodes comme le recuit simulé par exemple, mais ce n'est pas toujours le cas.

Caractérisation selon l'approche employée

Une autre façon de caractériser les méthodes d'optimisation, en dehors de la nature du problème traité ou de l'objectif recherché (optimum local ou global, avec ou sans preuve d'optimalité), a trait à l'approche employée pour traiter le problème. Une méthode analytique n'est possible que dans les cas les plus simples, et pour l'immense majorité des problèmes présentant un intérêt pratique, il faut recourir à des méthodes numériques. Parmi celles-ci, nous avons déjà évoqué deux types d'approches : déterministe ou stochastique. En dehors de ces qualificatifs généraux, la dénomination de certaines méthodes ou classes de méthodes fait plus ou moins explicitement référence à l'approche de résolution employée :

- métaheuristiques
- méthodes par intervalles
- programmation par contraintes
- méthodes hybrides d'optimisation

2.2.4 Méthodes d'optimisation employées dans ce mémoire

Dans la suite de ce chapitre, nous nous contenterons de présenter les principales méthodes utilisées dans les travaux présentés dans le cadre de ce mémoire, c'est à dire les méthodes de recherche dans des arbres ou des graphes, les méthodes de descente locale, et les métaheuristiques.

Des pans importants du domaine de l'Optimisation sont laissés de côté. Pour ce qui concerne les aspects théoriques de l'optimisation sous contraintes, et notamment la dualité lagrangienne et les conditions de Karush-Kuhn-Tucker, qui peuvent éclairer certains points exposés dans le chapitre 3 sur l'apprentissage artificiel, le lecteur peut se reporter à des ouvrages de référence comme [127, 120, 110].

1. Notons qu'il ne faut pas confondre l'optimisation stochastique, au sens où l'on traite un problème dont les variables sont aléatoires, avec les méthodes itératives stochastiques permettant de traiter un problème dont les variables sont déterministes (discrètes ou continues).

Recherche d’optimum dans des arbres ou des graphes

Dans les travaux présentés dans ce mémoire, les méthodes de recherche dans des arbres ou des graphes (section 2.3) sont employées au chapitre 4 pour partitionner de façon optimale l’espace aérien en secteurs de contrôle, en vue d’équilibrer la charge entre les différents postes de travail des contrôleurs aériens, tout en évitant les surcharges excessives.

Méthodes d’optimisation locale

Les méthodes de descente (section 2.4), et notamment celle de plus grande pente et les méthodes de Quasi-Newton BFGS ou L-BFGS, sont utilisées dans le cadre de l’apprentissage des réseaux de neurones présentés au chapitre 3. Ces réseaux de neurones sont utilisés d’une part dans le chapitre 4 pour apprendre un modèle de prédiction de la charge de travail, et dans le chapitre 5, avec d’autres méthodes d’apprentissage présentées au chapitre 3, pour la prévision de trajectoires d’avions en montée. Par ailleurs, la méthode BFGS est également directement employée dans le chapitre 5 pour apprendre le profil de poussée des moteurs lors de la montée. Le détail des méthodes BFGS et L-BFGS, qui ont fait l’objet d’une implantation en Ocaml durant nos travaux², est reporté en annexe A.

Métaheuristiques

Les métaheuristiques présentées dans la section 2.5 ont fait l’objet de quelques-uns de nos travaux, portant notamment sur l’hybridation avec une méthode d’intervalles [17], ou sur l’application d’algorithmes d’essaims particuliers ou d’évolution différentielle à la résolution de conflits aériens (voir section 2.6).

2.3 Méthodes de recherche dans des arbres ou des graphes

Les algorithmes présentés dans cette section appartiennent en fait à une catégorie plus générale de problèmes que celle des problèmes d’optimisation présentés en introduction. L’objectif est ici la “résolution de problèmes”, au sens large, telle qu’on la définit en Intelligence Artificielle (voir [140], chap. 3). Cette “résolution de problème” passe par la formulation du problème à résoudre, en formalisant en premier lieu l’espace d’états dans lequel on évolue (*i.e.* l’univers à explorer), ainsi que le (ou les) but(s) à atteindre dans cet espace d’états. On se munit également de “règles de production”, au sens où de telles règles permettent de générer, à partir de l’état courant, les états suivants à explorer. Typiquement, dans le cas où on manipule explicitement un arbre, un “état” sera un nœud de l’arbre, et la règle de production des états suivants fournit la liste des successeurs du nœud courant dans l’arborescence. Pour un graphe, on considérera les nœuds voisins.

2. Dans le cadre des travaux présentés au chapitre 4, une librairie de réseaux de neurones a été implantée en Ocaml, ainsi que diverses méthodes d’optimisation utilisées pour ajuster les poids : méthode de gradient (plus grande pente à pas fixe), gradient avec momentum, BFGS, L-BFGS, optimisation par essaims particuliers (PSO), évolution différentielles, et quelques hybridations de certaines de ces méthodes. Une partie de ces librairies devait initialement constituer un module spécifique (“calibration du modèle de la charge de travail”) dans la contribution de la DSN aux *work packages* 4.7.1 (“Complexity management in en-route”) et 10.8.1 “Complexity assessment and resolution” du programme européen SESAR, en partenariat avec l’industriel Thales.

Dans ce cadre, les algorithmes recherchent une “solution” (*i.e.* un des buts que l'on s'est fixé) en explorant l'espace d'états, partant d'un état initial. Diverses stratégies d'exploration sont possibles : des stratégies non informées, c'est-à-dire une recherche à l'aveugle (*blind search*) sans tirer parti de connaissances autres que celles formalisées dans le problème ; des stratégies informées (exactes ou approchées) guidées par une heuristique ; ou encore des stratégies faisant appel à une marche au hasard dans l'espace d'états, guidée par une métaheuristique.

Dans de nombreuses applications du monde réel, le problème pour lequel on cherche une solution fait intervenir un critère de coût : on cherche un état final de coût minimum. Par ailleurs, pour être une solution valide, un état doit éventuellement satisfaire certaines contraintes spécifiques au problème traité. C'est en particulier le cas pour le problème du chapitre 4, où l'on cherche un partitionnement optimal de l'espace aérien en respectant certaines contraintes opérationnelles (ex. nombre maximum de postes de travail ouvrables). Sous cette forme, on se retrouve donc dans le cadre des problèmes d'optimisation sous contraintes. C'est la raison pour laquelle nous présentons ici, dans ce chapitre dédié aux méthodes d'optimisation, les techniques de recherche dans des arbres ou des graphes, sans oublier qu'elles peuvent traiter des problèmes plus généraux.

2.3.1 Principe de la recherche dans des arbres ou des graphes

Algorithm 1 Principe général de la recherche dans un arbre.

Require: Etat initial u_0

```

1:  $F \leftarrow \{u_0\}$                                 ▷ Initialiser la frontière avec l'état initial  $u_0$ 
2: loop
3:   if  $F = \emptyset$  then
4:     return FAILURE                                ▷ Echec de la recherche si la frontière est vide
5:   else
6:      $u \leftarrow \text{CHOOSE}(F)$                     ▷ Choisir un nœud  $u$  sur la frontière (i.e. non développé)
7:      $F \leftarrow F \setminus \{u\}$ 
8:     if ISGOAL( $u$ ) then
9:       return  $u$                                   ▷ Si le but est atteint, retourner la solution trouvée
10:    else
11:       $S \leftarrow \text{EXPAND}(u)$                     ▷ Développer  $u$  (i.e. calculer ses successeurs)
12:       $F \leftarrow F \cup S$                           ▷ Ajouter ses successeurs à la frontière  $F$ 
13:    end if
14:  end if
15: end loop

```

Le principe général des algorithmes de recherche dans des arbres est exposé dans l'algorithme 1. Il consiste à développer les nœuds (ou états) situés sur la frontière F de “l'univers connu” des états déjà explorés. Pour cela, on choisit simplement un nœud sur cette frontière, et on le développe, c'est-à-dire qu'on utilise les règles de production pour calculer ses successeurs. Ceux-ci sont ensuite ajoutés à la frontière F .

Les différentes stratégies d'exploration (informées ou non, métaheuristiques) portent sur le choix des fonctions CHOOSE et EXPAND de l'algorithme 1.

Dans le cas d'une recherche dans un graphe, dont le principe est exposé dans l'algorithme 2 qui suit, on s'assure simplement de ne pas rajouter à la frontière des nœuds qui y sont déjà. Pour cela, on mémorise simplement l'ensemble des nœuds déjà développés.

Algorithm 2 Principe général de la recherche dans un graphe.

Require: Etat initial u_0

```

1:  $F \leftarrow \{u_0\}$                                 ▷ Initialiser la frontière avec l'état initial  $u_0$ 
2:  $D \leftarrow \emptyset$                                ▷ L'ensemble des nœuds développés est initialement vide
3: loop
4:   if  $F = \emptyset$  then
5:     return FAILURE                                ▷ Echec de la recherche si la frontière est vide
6:   else
7:      $u \leftarrow \text{CHOOSE}(F)$                        ▷ Choisir un nœud  $u$  sur la frontière (i.e. non développé)
8:      $F \leftarrow F \setminus \{u\}$ 
9:     if ISGOAL( $u$ ) then
10:      return  $u$                                     ▷ Si le but est atteint, retourner la solution trouvée
11:    else
12:       $D \leftarrow D \cup \{u\}$                          ▷ Ajouter  $u$  aux nœuds développés
13:       $S \leftarrow \text{EXPAND}(u)$                        ▷ Développer  $u$  (i.e. calculer ses successeurs)
14:      for all  $v \in S, v \notin D$  do                 ▷ Ajouter à  $F$  les successeurs non déjà développés
15:         $F \leftarrow F \cup \{v\}$ 
16:      end for
17:    end if
18:  end if
19: end loop

```

2.3.2 Complétude, optimalité, et complexité des algorithmes

Un algorithme de recherche est dit complet lorsqu'il garantit de trouver une solution quand il en existe une. Un algorithme est dit optimal si la solution trouvée est optimale (dans le cas où l'optimalité est un objectif du problème traité).

Outre les caractéristiques de complétude et d'optimalité, une caractéristique importante pour évaluer la performance des algorithmes de recherche dans des arbres ou des graphes et leur complexité, en temps ou en espace mémoire.

2.3.3 Implémentation des algorithmes de recherche

File de priorités, et stratégies de contrôle

En pratique, l'implémentation³ de ce type d'algorithme fait appel à différentes structures de données. Une file de priorités peut par exemple être utilisée pour coder la frontière F . Les nœuds sont alors rajoutés dans cette file, avec une priorité qui dépend de la stratégie adoptée.

Le choix de l'ordre de priorité fixe la stratégie de contrôle de l'algorithme. Par exemple, pour la recherche dans des arbres, on adopte une stratégie de type "en largeur d'abord" quand la file de priorité est de type FIFO (*first-in first-out*), c'est à dire si la plus haute priorité est allouée à l'élément le plus ancien dans la file. Si la file est de type LIFO (*last-in first-out*), c'est l'élément le plus récent qui est extrait en priorité, et on effectue une recherche "en profondeur d'abord". Pour une stratégie de type "meilleur en premier", la priorité allouée à un nœud u que l'on souhaite insérer dans la file dépendra d'une évaluation $f(u)$ de ce nœud.

Avec une implantation efficace sous forme d'arbres binaires équilibrés, l'opération d'insertion d'un élément dans la file de priorité, ainsi que l'opération d'extraction, sont d'une complexité en temps $O(\log(n))$, où n est ici la taille des données stockées dans la file.

3. Ce néologisme étant dorénavant dans le Larousse, nous nous permettons de l'utiliser.

Mémorisation des nœuds développés

Dans la description du principe des algorithmes que l'on peut trouver dans la littérature, l'ensemble D des nœuds déjà développés est parfois représenté sous forme d'une liste. Ce n'est pas gênant pour une description du principe d'un algorithme, mais celui-ci ne doit pas pour autant être implanté tel quel, sachant que vérifier si un élément appartient à une liste a un coût de calcul en $O(n)$.

Dans les rares cas où l'on s'intéresse à des problèmes de taille suffisamment petite, on peut envisager de remplacer cette liste par un tableau. Le coût de la vérification est alors en $O(1)$ pour le temps de calcul, et en $O(|G|)$ pour le stockage, $|G|$ étant la taille du graphe.

Dans la très grande majorité des cas, le graphe étudié est trop grand pour être explicité. On se contente de parcourir certains de ces nœuds, grâce aux règles de production de successeurs, qui sont alors calculés au fur et à mesure du parcours. Une solution possible et efficace pour stocker des informations sur les nœuds parcourus est alors d'utiliser par exemple une table de *hash*, ou une autre structure de données à base d'arbres binaires. Les coûts d'insertion et d'extraction d'éléments dans cette structure sont alors en $O(\log(n))$.

Dans le cas où l'on recherche une solution de coût optimal, cette structure peut être utilisée pour stocker également les coûts associés aux nœuds. Si l'on recherche un chemin optimal dans un graphe, il est également utile d'y mémoriser le prédécesseur de chaque nœud, afin de pouvoir comparer les coûts de chemins alternatifs menant à un même nœud, et aussi de reconstruire le chemin parcouru à retourner lorsqu'on atteint un état final.

2.3.4 Algorithmes de type *branch & bound*

Dans cette section et dans la suivante, on se place dans le contexte d'un problème d'optimisation, où l'on associe un coût à chaque solution du problème traité, l'objectif étant de trouver la solution de coût minimum.

On a donc ici un double objectif : d'une part, partant d'un état initial, on cherche à atteindre un état final (une solution), et d'autre part, on souhaite trouver, parmi toutes les solutions possibles, celle de coût minimal.

Principe général des méthodes de *branch and bound*

Le principe du *branch & bound* suppose qu'on puisse estimer, pour chaque nœud parcouru, un minorant du coût de la meilleure solution atteignable à partir de ce nœud (*i.e.* dans le sous-arbre). Si ce minorant est supérieur au coût de la meilleure solution déjà rencontrée auparavant, il est inutile de poursuivre plus avant l'exploration du sous-arbre considéré. Cette stratégie de coupure permet d'explorer beaucoup moins de nœuds que lors d'une exploration exhaustive. Il faut toutefois décider aussi d'une stratégie d'exploration de l'espace d'état (*i.e.* de l'arbre de recherche). Par exemple, on pourra explorer l'arbre "en profondeur d'abord" pour essayer de trouver rapidement une première solution, qu'on espère suffisamment bonne pour pouvoir couper des branches dans la suite de la recherche. Si l'on dispose d'une heuristique qui nous informe plus précisément sur le coût estimé des solutions dans le sous-arbre, on peut également adopter une stratégie d'exploration "meilleur en premier" qui tient compte de ces informations. La priorité associée à chaque nœud inséré dans la file des nœuds à développer dépend de la stratégie d'exploration choisie.

L'algorithme 4 donne le principe général des méthodes de *branch & bound* (souvent traduit par "séparation et évaluation"). Notons que la structure d'arbre n'a pas besoin d'être explicitée

Algorithm 3 Principe des méthodes de *branch and bound* (ici, pour la recherche dans un graphe)

Require: u_0 ▷ Etat initial

- 1: $(best, cost_{best}) \leftarrow (None, +\infty)$
- 2: $p_0 \leftarrow \text{PRIORITY}(u_0)$
- 3: $Q \leftarrow \text{INSERT}(u_0, p_0, \emptyset)$ ▷ Initialiser la file de priorités
- 4: $D \leftarrow \emptyset$ ▷ L'ensemble des nœuds développés est initialement vide
- 5: **loop**
- 6: **if** $Q = \emptyset$ and $best = None$ **then**
- 7: **return** FAILURE ▷ Aucune solution
- 8: **else if** $Q = \emptyset$ and $best \neq None$ **then**
- 9: **return** $(best, cost_{best})$ ▷ Retourner la solution de coût minimum
- 10: **else**
- 11: $(u, Q) \leftarrow \text{EXTRACT}(Q)$ ▷ Extraire le nœud u le plus prioritaire de la file
- 12: **if** ISOLUTION(u) **then**
- 13: $cost_u \leftarrow \text{COST}(u)$ ▷ Si u est un état final, calculer son coût
- 14: **if** $cost_u < cost_{best}$ **then**
- 15: $(best, cost_{best}) \leftarrow (u, cost_u)$ ▷ Mettre à jour la meilleure solution courante
- 16: **end if**
- 17: **else** ▷ Cas où u n'est pas un état final, solution du problème :
- 18: $D \leftarrow D \cup \{u\}$ ▷ Ajouter u aux nœuds développés
- 19: $S \leftarrow \text{EXPAND}(u)$ ▷ Développer u (*i.e.* calculer ses successeurs)
- 20: **for all** $v \in S, v \notin D$ **do**
- 21: $eval_v \leftarrow \text{COSTESTIMATE}(v)$ ▷ Minorant du coût des solutions atteignables
- 22: **if** $eval_v < cost_{best}$ **then**
- 23: $p_v \leftarrow \text{PRIORITY}(v)$
- 24: $Q \leftarrow \text{INSERT}(v, p_v, Q)$
- 25: **end if** ▷ Couper la branche si minorant supérieur à $cost_{best}$
- 26: **end for**
- 27: **end if**
- 28: **end if**
- 29: **end loop**

à travers la définition d'un type. Elle est implicite, dans le choix de la fonction EXPAND qui calcule les descendants d'un nœud quelconque.

Exemple : recherche d'une partition optimale

A titre d'illustration, considérons le problème de la recherche d'une partition optimale d'un ensemble. La construction de toutes les partitions possibles peut être représentée sous forme d'une arborescence, comme illustré sur la Figure 2.2 pour 5 éléments.

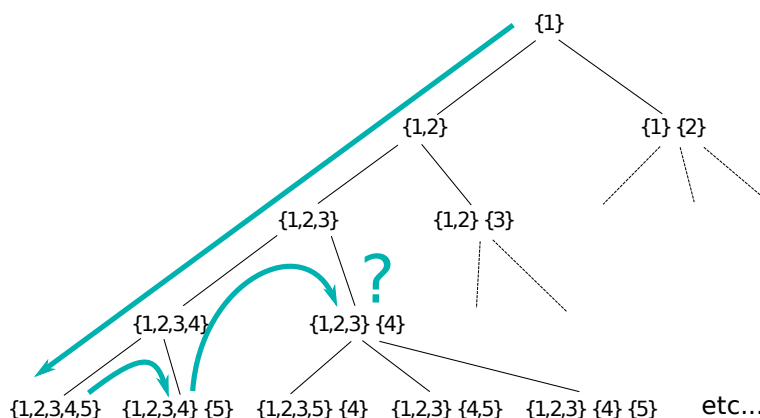


FIGURE 2.2: Construction arborescente de toutes les partitions d'un ensemble à 5 éléments.

Un état solution de notre problème est ici une feuille de l'arbre, c'est-à-dire une partition complète dont on évalue le coût avec la fonction COST (ligne 13 de l'algorithme 4). Les nœuds non terminaux de l'arbre 2.2 sont des états intermédiaires, non solutions, que l'on évalue avec la fonction COSTESTIMATE (ligne 21 de l'algorithme 4).

Supposons qu'on ait choisi une stratégie d'exploration "en profondeur d'abord". L'algorithme commencera alors par descendre jusqu'à la première feuille possible, par exemple ici la partition $\{1, 2, 3, 4, 5\}$, dont il évaluera le coût. Cette feuille étant la première rencontrée, elle devient provisoirement la meilleure solution trouvée. Par la suite, lorsque l'algorithme revient en arrière dans l'arbre (*backtracking*), l'évaluation des nœuds intermédiaires est comparée au coût de cette meilleure solution, ce qui permet le cas échéant d'éviter d'explorer complètement certains sous-arbres. Lorsque l'exploration se poursuit jusqu'à une feuille d'un coût plus faible que celui de la meilleure solution déjà trouvée, cette dernière est remplacée par la nouvelle solution.

La complexité effective de cette recherche arborescente dépend de plusieurs facteurs : la stratégie d'exploration et l'ordre dans lequel sont explorés les états terminaux, et la qualité de l'heuristique minorante COSTESTIMATE. Dans notre exemple, si la première solution rencontrée est celle de coût minimal et si l'heuristique minorante est parfaite, c'est-à-dire si $\text{COSTESTIMATE}(u)$ est égal au coût de la meilleure solution atteignable à partir de u , la complexité en temps est en $O(d)$, où d est la profondeur de l'arbre. A l'extrême opposé, si les partitions que sont les états terminaux de la Figure 2.2 sont explorées par ordre décroissant de coût, on voit qu'il faut développer tout l'arbre avant de trouver la solution de coût minimum. La complexité temporelle est alors en $O(b^d)$, où b est le facteur de branchement.

Branch & bound par intervalles

Dans une tout autre catégorie de problèmes, intéressons nous maintenant à la minimisation d'une fonction-objectif f , à variables réelles $x \in \mathbb{R}^n$ et à valeur dans \mathbb{R} , dans un domaine borné $X_0 \subset \mathbb{R}^n$. La méthode de *Branch & bound* peut être appliquée à l'espace dont les états sont des vecteurs d'intervalles ("boîtes"), de façon à encadrer l'optimum de la fonction f avec une précision donnée. Notons $\mathbb{I} = \{[a, b] \mid a \leq b, (a, b) \in \mathbb{R}^2\}$ l'espace des intervalles dans \mathbb{R} . Un vecteur X d'intervalles appartient alors à \mathbb{I}^n , de dimension n .

Supposons qu'on dispose d'une fonction d'inclusion $F : \mathbb{I}^n \rightarrow \mathbb{I}$, telle que l'image de tout vecteur d'intervalles $X \in \mathbb{I}^n$ soit un intervalle $F(X) \in \mathbb{I}$ qui contienne l'image de X par f . Lorsqu'on dispose d'une expression analytique de la fonction-objectif f , la fonction d'inclusion F peut par exemple être définie comme une extension naturelle de f à l'arithmétique des intervalles, dont les opérations de base sont rappelées dans les équations (2.3) suivantes. L'optimisation globale par analyse d'intervalles est par ailleurs décrite plus en détail dans le livre de Hansen [82].

$$[a, b] + [c, d] = [a + c, b + d] \quad (2.3a)$$

$$[a, b] - [c, d] = [a - d, b - c] \quad (2.3b)$$

$$[a, b] * [c, d] = [\min\{ac, ad, bc, bd\}, \max\{ac, ad, bc, bd\}] \quad (2.3c)$$

$$[a, b]/[c, d] = [a, b] * [1/d, 1/c] \quad \text{si } 0 \notin [c, d] \quad (2.3d)$$

Algorithm 4 Algorithme de *Branch and bound* par intervalles

Require: Domaine initial X_0

```

1:  $p_0 \leftarrow \text{PRIORITY}(X_0)$ 
2:  $Q \leftarrow \text{INSERT}(X_0, p_0, \emptyset)$  ▷ Initialiser la file de priorités
3:  $X_{best} \leftarrow X_0$ 
4:  $f_{best} \leftarrow f(\text{midpoint}(X_0))$ 
5: while  $Q \neq \emptyset$  do
6:    $(X, Q) \leftarrow \text{EXTRACT}(Q)$  ▷ Extraire la boîte  $X$  de la file de priorités  $Q$ 
7:   if not  $[\text{LOWERBOUND}(F(X)) \geq f_{best} \text{ or } \text{STOP}(X, F(X), f_{best})]$  then
8:      $(X_1, X_2) \leftarrow \text{SPLIT}(X)$  ▷ Couper la boîte en deux sous-boîtes
9:     for  $i \in \{1, 2\}$  do
10:       $f_{X_i} \leftarrow f(\text{midpoint}(X_i))$  ▷ Evaluer  $f$  au point milieu de l'intervalle
11:      if  $f_{X_i} < f_{best}$  then ▷ Cas où on améliore le critère  $f$ 
12:         $(X_{best}, f_{best}) \leftarrow (X_i, f_{X_i})$ 
13:      end if
14:       $p_{X_i} \leftarrow \text{PRIORITY}(X_i)$ 
15:       $Q \leftarrow \text{INSERT}(X_i, p_{X_i}, Q)$  ▷ Insérer la sous-boîte dans la file
16:    end for
17:  end if
18: end while
19: return  $(X_{best}, f_{best})$ 

```

Le *Branch & bound* par intervalles, décrit par l'algorithme 4, commence par évaluer f au point milieu du domaine initial $X_0 \in \mathbb{I}$, puis par couper celui-ci en deux sous-boîtes X_1 et X_2 qui sont ensuite évaluées par la fonction d'inclusion F , avant d'être éventuellement réinsérées dans la file de priorité contenant les boîtes à traiter. L'image par F d'une boîte X_i est un

Algorithm 5 Condition d'arrêt du découpage en sous-domaines par le *branch and bound* par intervalles

Require: ϵ_x, ϵ_f

1: **function** STOP($X, F(X), f_{best}$)

2: WIDTH(X) $\leq \epsilon_x$ **or** WIDTH($F(X)$) $\leq \epsilon_f$

3: **end function**

intervalle de réels. Si sa borne inférieure est plus grande que la meilleure valeur déjà trouvée de $f(x)$, la boîte X_i n'est pas réinsérée dans la file car on est certain qu'elle ne contient pas l'optimum. Ce découpage se poursuit récursivement jusqu'à satisfaire un critère d'arrêt portant sur la taille de la boîte ou l'encadrement de l'intervalle (voir algorithme 5 pour la condition d'arrêt du découpage en sous-boîtes). L'algorithme s'arrête lorsque la file est vide.

2.3.5 Algorithme A^*

L'algorithme A^* [84, 85] entre dans la catégorie des méthodes informées, avec une stratégie d'exploration de type "meilleur en premier" guidée par une heuristique h . L'objectif de cet algorithme est de trouver un chemin de coût minimum entre un état initial u_0 et un état solution (on parle encore d'état final, ou terminal). L'heuristique est ici simplement une estimation du meilleur coût possible du chemin restant à parcourir entre l'état courant et un état terminal. Cet algorithme nécessite par ailleurs de définir une fonction $k(u, v)$ retournant le coût de la transition entre deux états successifs u et v .

Principe général de l'algorithme A^*

Le principe de l'algorithme A^* , récapitulé dans l'algorithme 6, consiste à repartir, à chaque itération, du nœud ayant la meilleure estimation (coût du chemin déjà parcouru, plus heuristique). Sous certaines conditions sur l'heuristique h , le premier état terminal atteint est par construction celui pour lequel le chemin entre u_0 et cet état final est de coût minimum.

Un exemple simple de l'application de ce type d'algorithme consiste à trouver le chemin le plus court entre deux villes, dans un réseau routier. L'ensemble des états terminaux est alors réduit à la ville d'arrivée. L'état initial est la ville de départ. Le coût $k(u, v)$ est la distance entre deux villes u et v reliées par un tronçon de route, et l'heuristique $h(v)$ est la distance à vol d'oiseau entre la ville v et la destination.

L'algorithme A^* explore l'arbre (sans pour autant l'explicitier) de tous les chemins possibles entre l'état initial u_0 et un état terminal. A chaque itération, il va chercher dans la frontière F le nœud pour lequel la somme $f(u) = cost(u) + h(u)$ est la plus faible, $cost(u)$ étant le coût du chemin déjà parcouru et $h(u)$ l'estimation (par l'heuristique) du coût du chemin restant à parcourir. Si u c'est un état terminal, la recherche est terminée et on retourne le chemin parcouru. Sinon, le nœud u extrait de F est développé, et chacun de ses descendants v est inséré dans la frontière F , à condition qu'il n'ait jamais été rencontré avant, ou bien, si v a déjà été rencontré, que le chemin allant de u_0 à v et passant par u soit de coût inférieur au meilleur coût précédemment mémorisé pour v . C'est cette condition qui est exprimée par la ligne 12 de l'algorithme 6.

Récapitulatif des notations

u_0 : état initial

T : ensemble de nœuds terminaux (solutions)

D : ensemble des nœuds développés

F : ensemble des nœuds sur la frontière. Ce sont les nœuds générés restant à développer

$h(u)$: fonction heuristique estimant le coût du chemin restant à parcourir entre u et un état final appartenant à T

$cost(u)$: mémorise le meilleur coût trouvé pour le cheminement allant de u_0 à u

$f(u)$: estimation du coût total du chemin passant par u , allant de u_0 à un état terminal.
 $f(u) = cost(u) + h(u)$ est la somme du coût du chemin déjà parcouru et de l'estimation par l'heuristique du coût du chemin restant à parcourir

$k(u, v)$: fonction retournant le coût associé à la transition entre u et v

$parent(v)$: mémorise le parent de v pour lequel le chemin de u_0 à v est de coût le plus faible

Par ailleurs, la stratégie d'exploration de l'espace d'états est fixée par les fonctions suivantes :

- EXPAND : fonction appliquant un ensemble de règles de production $\{p_1, p_2, \dots, p_k\}$ à un état u , pour obtenir un ensemble de successeurs $S = \{v_1, v_2, \dots, v_k\}$. Dans l'exemple du réseau routier, il s'agira simplement des villes voisines de la ville u , reliées à u par un tronçon de route,
- BESTINFRONTIER : fonction retournant le meilleur nœud u de la frontière F , c'est-à-dire celui pour lequel l'estimation $f(u) = cost(u) + h(u)$ du coût total d'un chemin passant par u est le plus faible.
- ADDORREPLACE : ajoute un nœud v à la frontière F , ou le remplace s'il y était déjà, en lui allouant une priorité déterminée par l'estimation $f(v)$.

Algorithm 6 Principe de l'algorithme A^* .

Require: Etat initial u_0

```

1:  $F \leftarrow \{u_0\}$ 
2:  $cost(u_0) \leftarrow 0$ 
3: while  $F \neq \emptyset$  do
4:    $u \leftarrow \text{BESTINFRONTIER}(F)$  ▷ Choisir  $u$  de plus petite valeur de  $f(u)$  dans  $F$ 
5:    $F \leftarrow F \setminus \{u\}$  ▷ Le retirer de la frontière
6:   if  $u \in T$  then ▷ Cas où  $u$  est un état terminal : retourner le chemin
7:     return  $\text{PATH}(u_0, u) = u :: \text{parent}(u) :: \text{parent}(\text{parent}(u)) :: \dots :: u_0$ 
8:   else
9:      $D \leftarrow D \cup \{u\}$  ▷ Ajouter le nœud  $u$  dans l'ensemble des nœuds développés
10:     $S \leftarrow \text{EXPAND}(u)$  ▷ Développer  $u$  (i.e. calculer ses successeurs)
11:    for all  $v \in S$  do
12:      if  $v \notin D \cup F$  or  $cost(v) > cost(u) + k(u, v)$  then
13:         $cost(v) \leftarrow cost(u) + k(u, v)$  ▷ Mémoriser le coût du chemin  $u_0, \dots, u, v$ 
14:         $f(v) \leftarrow cost(v) + h(v)$  ▷ Calculer l'estimation  $f(v)$  déterminant la priorité de  $v$ 
15:         $parent(v) \leftarrow u$  ▷ Mémoriser  $u$  comme parent de  $v$ 
16:         $F \leftarrow \text{ADDORREPLACE}(v, f(v), F)$  ▷ Insérer  $v$  dans la frontière  $F$ 
17:      end if
18:    end for
19:  end if
20: end while
21: return FAILURE ▷ Cas où on a vidé la file, sans trouver de solution

```

L'algorithme 6 décrit le principe de l' A^* , étant entendu qu'une programmation efficace de cet algorithme nécessite quelques aménagements. Typiquement, la frontière F est implantée comme une file de priorité Q , codée par un arbre binaire, et le choix du nœud u de meilleure priorité et le retrait de ce nœud de la file sont combinées en une seule opération d'extraction. Les opérations de stockage en mémoire des meilleurs coûts, des nœuds développés, et du nœud parent peuvent se faire avec une table de *hash*.

Liens entre l' A^* et d'autres méthodes de recherche arborescente

Revenons aux méthodes de type *branch & bound* présentées dans l'algorithme 4 de la section 2.3.4, et considérons le cas particulier où l'on traite des problèmes pour lesquels l'estimation `COSTESTIMATE` peut s'exprimer comme la somme du coût $cost(u)$ du parcours passé jusqu'à u , et de l'heuristique $h(u)$ estimant le coût du chemin restant à parcourir. Supposons par ailleurs que le coût $cost(u)$ soit cumulatif, égal à la somme des coûts individuels $k(u_i, u_{i+1})$ des transitions passées, $[u_0, \dots, u_i, u_{i+1}, \dots, u]$ étant le chemin suivi jusqu'à u . L'algorithme A^* peut alors être vu comme un cas particulier du *branch & bound* de l'algorithme 4, ayant les caractéristiques suivantes :

- $COSTESTIMATE(u) = f(u) = cost(u) + h(u)$, avec une fonction heuristique h ayant les bonnes propriétés (voir plus loin), ce qui permet d'arrêter la recherche à la première solution trouvée, celle-ci étant de coût optimal
- stratégie "meilleur en premier" où la priorité dans la file Q est déterminée par $f(u)$

Par ailleurs, considérons maintenant un cas particulier de l' A^* où l'on adopte une heuristique nulle $h(u) = 0$. La stratégie "meilleur en premier" de l' A^* se réduit alors à prendre dans la file le nœud de coût $cost(u)$ le plus faible, ce qui nous ramène à un algorithme de Dijkstra.

Propriétés de l' A^*

Les propriétés de l'algorithme A^* dépendent notamment de la nature de l'heuristique h estimant le coût du chemin restant à parcourir.

Définition 2.3 (Heuristique admissible (ou minorante)). *Une heuristique est admissible (ou minorante) si pour tout nœud u , $h(u) \leq h^*(u)$, où $h^*(u)$ est le coût minimum pour arriver à un état terminal à partir de u .*

Définition 2.4 (Heuristique consistante (ou monotone)). *Une heuristique est consistante (ou monotone) si pour tout successeur v du nœud u , $h(u) \leq k(u, v) + h(v)$, où $k(u, v)$ est le coût de la transition de u à v .*

Toute heuristique consistante est admissible, mais l'inverse n'est pas vrai.

Théorème 2.1 (Optimalité de l' A^* pour la recherche dans un arbre). *Si l'heuristique h est admissible, alors l'algorithme A^* est optimal (pour une recherche dans un arbre).*

Théorème 2.2 (Optimalité de l' A^* pour la recherche dans un graphe). *Si l'heuristique h est consistante, alors l'algorithme A^* est optimal (pour une recherche dans un graphe).*

Voir [84, 85], ou encore [140], chap. 3, pour la démonstration de ces théorèmes. La complétude de l'algorithme requiert simplement que le facteur de branchement soit fini, et que les coûts $k(u, v)$ ne soient pas nuls.

Les propriétés d’optimalité et de complétude de l’ A^* , pour intéressantes qu’elles soient, ne permettent pas pour autant de traiter avec succès tous les problèmes de chemin optimal dans un arbre ou un graphe avec cet algorithme. En effet, lorsque les conditions d’optimalité sont remplies, l’ A^* développe tous les nœuds u pour lesquels $f(u) < c^*$, où c^* est le coût optimal, et éventuellement certains nœuds pour lesquels $f(u) = c^*$. On peut donc se représenter l’algorithme A^* comme un élargissement itératif des iso-contours de f . La complexité de l’ A^* dépend de l’adéquation entre l’heuristique choisie et la configuration de l’espace d’états.

Pour s’en convaincre, revenons sur l’exemple du plus court chemin dans un réseau routier. En prenant une heuristique nulle, les iso-contours de f sont des cercles concentriques autour de la ville de départ. On va donc développer tous les nœuds à l’intérieur d’un cercle allant s’agrandissant, jusqu’à atteindre la ville de destination. Dans le cas où l’heuristique est la distance à vol d’oiseau, et où le réseau routier est un maillage suffisamment régulier, les iso-contours sont déformés dans la direction de la ville de destination. On voit qu’on peut avoir une complexité exponentielle dans un cas, et linéaire dans l’autre si l’heuristique est parfaite ($h = h^*$). Prendre l’heuristique “distance à vol d’oiseau” ne suffit pas, dans notre exemple, pour garantir une complexité linéaire. Imaginons en effet qu’il y ait un obstacle entre les deux villes, qui provoque une disruption dans le maillage routier (un pays aux frontières fermées, par exemple). L’ A^* va d’abord développer tous les chemins les plus directs menant jusqu’à une impasse (frontière fermée), avant de faire le tour de l’obstacle.

La complexité effective de l’algorithme dépend, dans notre exemple avec un seul état terminal, de l’écart entre h et h^* et du facteur de branchement (*i.e.* le nombre de successeurs de chaque nœud). Comme il est bien difficile de trouver une heuristique parfaite si l’on ne connaît déjà pas le chemin optimal, on se retrouve en règle générale avec une complexité exponentielle.

L’application de l’algorithme A^* à des problèmes de grande taille peut donc s’avérer impraticable, pour les raisons que nous venons d’évoquer, et également pour des raisons de stockage mémoire. Afin d’accélérer la recherche, on se contente alors parfois d’une ϵ -approximation, en choisissant une heuristique non admissible, mais telle que $h(u) \leq h^*(u) + \epsilon$, avec $\epsilon > 0$. Selon la complexité et la taille des problèmes traités, cette stratégie peut toutefois ne pas suffire, et il alors faut passer à d’autres types d’algorithmes comme les métaheuristiques (voir section 2.5), qui ne garantissent plus l’optimalité ni même une approximation bornée de l’optimum.

2.4 Descente locale sans contraintes, en domaine continu

On s’intéresse ici aux fonctions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et on suppose l’existence et la continuité du gradient $\nabla f(x) = \left[\frac{\partial f}{\partial x}(x) \right]$ et éventuellement de la hessienne $\nabla^2 f(x) = \left[\frac{\partial^2 f}{\partial x \partial x^T}(x) \right]$.

Certaines méthodes d’optimisation locale (voir [127]) utilisent le gradient $\nabla f(x)$, et/ou la hessienne $\nabla^2 f(x)$ ou son approximation pour chercher à améliorer itérativement le critère $f(x)$.

2.4.1 Conditions d’optimalité

Théorème 2.3 (CNS d’optimalité locale stricte, cas sans contraintes). *Les conditions nécessaires et suffisantes pour que le point x^* soit un minimum local strict de la fonction f de classe \mathcal{C}^2 sont les suivantes :*

$$\nabla f(x) = 0$$

$\nabla^2 f(x)$ est définie positive.

Théorème 2.4 (Minimum d'une fonction convexe). *Quand f est convexe, tout minimum local strict est un minimum global de f . Si de plus f est différentiable, tout point stationnaire x^* est un optimum global de f .*

2.4.2 Principe des méthodes de descente avec *line search*

Partant d'un point initial x_0 , les méthodes de descente effectuent à chaque itération un pas de longueur α_k dans une direction de descente d_k , telle que $d_k^T \nabla f(x_k) < 0$. L'algorithme 7 résume ces différentes étapes.

Algorithm 7 Principe général des méthodes de descente.

```

1:  $x_0 \leftarrow \text{INIT}()$ 
2: while not(STOP( $x_k$ )) do
3:   Choisir une direction de descente  $d_k$ 
4:   Choisir la longueur du pas  $\alpha_k$ 
5:    $x_{k+1} \leftarrow x_k + \alpha_k d_k$ 
6:    $k \leftarrow k + 1$ 
7: end while
8: return ( $x_k, f(x_k)$ )

```

2.4.3 Choix de la direction de descente

Plus grande pente

$$d_k = -\nabla f(x_k) \tag{2.4}$$

Un exemple classique du type de méthode décrit dans l'algorithme 7 est la descente de plus grande pente, où l'on choisit $d_k = -\nabla f(x_k)$ comme direction de descente, à chaque itération.

Dans sa version la plus basique, le pas α_k est choisi fixe : $\alpha_k = \eta$. Cette approche pose le problème du choix du paramètre η , pour chaque problème considéré. Cependant, même en utilisant un pas variable α_k , ajusté à chaque itération en minimisant $\Phi(\alpha) = f(x_k + \alpha d_k)$ (voir section 2.4.4), la descente de plus grande pente peut s'avérer extrêmement lente à converger, lorsque la fonction-objectif présente une "vallée", comme illustré sur la Figure 2.3.

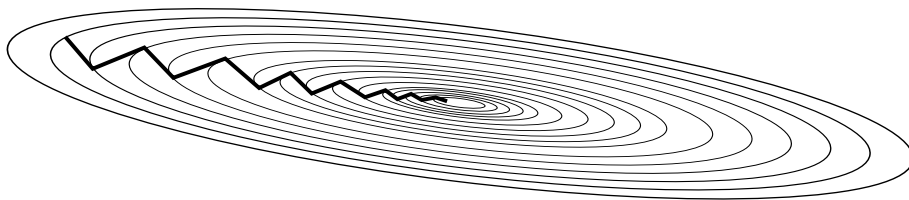


FIGURE 2.3: Illustration des nombreuses itérations de la méthode de plus grande pente.

Ces constatations ont motivé d'autres approches, où la direction de descente d_k est choisie plus judicieusement de façon à mener plus rapidement vers un optimum x^* , local, voire global si f est convexe.

Rappelons ici brièvement les directions de descente des approches les plus classiques : gradients conjugués, méthode de Newton, ou de Quasi-Newton. Le lecteur peut se reporter à [127, 110] pour plus de détails sur ces méthodes.

Gradients conjugués

$$d_k = -\nabla f(x_k) + \beta_k d_{k-1} \quad (2.5)$$

Dans la méthode des gradients conjugués, le facteur β_k du terme correctif est choisi de façon à assurer que les directions d_i successives soient conjuguées.

Direction de Newton

$$d_k = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \quad (2.6)$$

La direction de Newton de l'équation (2.6) s'obtient en effectuant un développement de Taylor du gradient $\nabla f(x_{k+1}) \approx \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k)$. Si la hessienne $\nabla^2 f(x_k)$ est inversible, on en déduit $x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$, ce qui correspond à un pas de longueur 1 dans la direction d_k de l'équation (2.6).

Méthodes de Quasi-Newton

$$d_k = -B_k^{-1} \nabla f(x_k) \quad \text{ou encore} \quad d_k = -H_k \nabla f(x_k) \quad (2.7)$$

Les méthodes de Quasi-Newton mettent à jour itérativement une approximation B_k de la hessienne $\nabla^2 f(x_k)$, en respectant un certain nombre conditions (ex. B_k symétrique, définie positive). Elles peuvent aussi approximer directement l'inverse $[\nabla^2 f(x_k)]^{-1}$ de la hessienne par une matrice H_k .

Parmi les méthodes de Quasi-Newton les plus connues et les plus performantes, citons la méthode BFGS (*Broyden-Fletcher-Goldfarb-Shanno*) et sa variante à mémoire limitée L-BFGS, plus adaptée aux problèmes de grande dimension. Ces deux méthodes sont présentées en détail dans l'annexe A. Elles ont été codées en Ocaml et sont notamment utilisées, dans le cadre des travaux présentés dans ce mémoire, pour ajuster les poids des réseaux de neurones (voir section 3.8 du chapitre 3), qui sont appliqués à des problèmes de trafic aérien dans les chapitres 4 et 5.

2.4.4 Recherche unidimensionnelle pour le choix du pas α_k

Dans l'idéal, on voudrait choisir le pas α_k qui minimise la fonction-objectif f dans la direction d_k , ce qui revient à minimiser la fonction $\Phi : \alpha \rightarrow f(x + \alpha d_k)$.

$$\alpha_k = \arg \min_{\alpha > 0} f(x + \alpha d_k) \quad (2.8)$$

Rechercher ce minimum dans la direction d_k peut s'avérer relativement difficile ou coûteux en temps, dans le cas général. En pratique, on utilise une méthode itérative pour effectuer cette recherche unidimensionnelle. Cette méthode peut être inexacte, auquel cas on se contente de

chercher un pas α_k qui satisfasse certaines conditions garantissant que la méthode de descente multidimensionnelle converge en un nombre fini d'itérations.

Ces conditions portent, directement ou indirectement, sur la longueur minimale du pas et une décroissance suffisamment significative de f (voir [19, 176], [127], chap. 3, ou [110], section 8.5).

Un grand nombre de méthodes de recherche unidirectionnelle (*line search*), non présentées ici, ont été proposées dans la littérature ([124], ou à nouveau [127], chap. 3, [110], chap. 8, ou encore [120], chap. 3) leur principe consistant à rechercher par encadrements successifs un intervalle contenant un pas α_k satisfaisant les conditions requises.

La méthode unidirectionnelle et ses conditions d'arrêt sont généralement choisies de façon à assurer certaines propriétés de convergence (linéaire, super-linéaire, quadratique), de la méthode générale de descente (voir section 2.4.5 suivante).

2.4.5 Propriétés de convergence des méthodes de descente

Rappelons pour terminer les propriétés de convergence des quelques méthodes de descente que nous venons de voir. Au voisinage d'un optimum local x^* , la convergence plus ou moins rapide vers cet optimum x^* est caractérisée en comparant les écarts $\|x - x^*\|$ pour deux itérations successives.

On parlera de convergence linéaire (voir [110], section 7.8, ou [120], section 4.6) lorsque $\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = \alpha < 1$, et de convergence *super-linéaire* lorsque $\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0$. Enfin, la convergence est dite quadratique si $\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} < +\infty$.

On montre, sous certaines conditions sur la fonction f et sur le choix du pas α_k , que la méthode de plus grande pente converge linéairement, la méthode de Newton converge quadratiquement, et les méthodes de quasi-Newton convergent super-linéairement (voir [127], sections 3.3 et 6.4, ou [110]).

2.5 Métaheuristiques

Le terme *métaheuristique* a été introduit par Glover au milieu des années 80, à propos de la Recherche Tabou⁴. Dans [77], Glover décrit sa méthode comme une heuristique de plus haut niveau, pour la résolution de problèmes d'optimisation, conçue pour guider d'autres méthodes afin de sortir du piège de l'optimum local.

Bien d'autres méthodes sont aujourd'hui regroupées sous cette appellation générale, dont un bon nombre existaient avant l'apparition du terme "métaheuristique", comme les stratégies d'évolution ([22]), les algorithmes génétiques ([90, 79]), ou le recuit simulé ([100]). Les métaheuristiques sont une collection de méthodes très diverses ayant pour objectif la résolution de problèmes difficiles pour lesquels les méthodes "classiques" d'optimisation ne fonctionnent plus. Ce sont des algorithmes itératifs, en règle générale stochastiques, dans le sens où l'exploration de l'espace des points admissibles fait appel à une marche au hasard, et qui recherchent un optimum global mais sans garantie de le trouver.

Dans leur principe, ces méthodes échantillonnent itérativement (et plus ou moins aléatoirement) l'espace des points admissibles, en étant guidés par une heuristique qui oriente la

4. *Tabu search may be viewed as a "meta-heuristic" superimposed on another heuristic.* [76]

recherche vers des solutions de bonne qualité. Cet échantillonnage peut faire explicitement appel à des distributions de probabilités, comme dans le cas des algorithmes à estimation de distribution, ou reposer sur d'autres mécanismes, comme dans le cas des algorithmes génétiques où de nouveaux individus sont produits par croisement et mutation.

Toutes ces méthodes s'appuient, à des degrés divers, sur trois types de mécanismes, parfois concurrents : des déplacements aléatoires, où l'on s'autorise à dégrader le critère-objectif f et qui permettent de diversifier la recherche en sortant des optima locaux ; une recherche "gloutonne" visant à intensifier la recherche dans les zones intéressantes en améliorant la fonction-objectif f ; et une archive permettant de mémoriser les zones déjà explorées, afin soit d'éviter d'y retourner (ex. : méthode Tabou), soit au contraire de définir des "attracteurs" indiquant des zones intéressantes à explorer (ex. : meilleurs positions trouvées, pour les essais particulières).

Au-delà de leurs sources d'inspiration diverses (sélection darwinienne, phénomènes physiques, etc), l'efficacité des métaheuristiques tient à la combinaison de ces trois mécanismes, qui fixe un compromis entre la diversification et l'intensification, dans la recherche d'optima. Pour une présentation des métaheuristiques en général, voir l'ouvrage de Talbi [157], ou encore les ouvrages de Siarry *et. al.* [45, 15]. Pour les algorithmes évolutionnaires en particulier, on pourra se référer à [55].

2.5.1 Algorithmes évolutionnaires

Les algorithmes évolutionnaires s'inspirent de l'évolution naturelle et du processus darwinien de sélection et de survie des individus les plus adaptés à leur environnement. Ici, cet environnement est simplement le paysage de la fonction dont on cherche l'optimum. A cause de leur analogie avec le processus naturel, on parle plus volontiers de maximiser l'adaptation (*fitness*) des individus, que de minimiser une fonction-objectif. Cependant, si l'on cherche à minimiser une fonction f , il suffit de choisir comme *fitness* d'un individu x l'opposé de $f(x)$ pour se ramener au problème de minimisation initial.

Le principe général d'un algorithme évolutionnaire, donné par l'algorithme 8, est d'initialiser une population d'une certaine taille, par exemple en tirant des points au hasard dans l'espace de recherche, puis de faire évoluer cette population par sélection, croisement, et mutation de ses individus.

Algorithm 8 Principe général des algorithmes évolutionnaires.

Require: Taille de la population $Popsiz$, probabilités de croisement P_c et de mutation P_m

- 1: $Pop_0 \leftarrow \text{INIT}(popsiz)$
 - 2: **while** not(STOP(k, Pop_k)) **do**
 - 3: Evaluer la valeur brute de l'*adaptation* des éléments de Pop_k
 - 4: Modifier les adaptations brutes (ex. *scaling* et *sharing*)
 - 5: Sélectionner un *pool* de parents, selon les nouvelles adaptations
 - 6: Créer une nouvelle population par croisement, mutation, ou duplication de certains parents
 - 7: $k \leftarrow k + 1$
 - 8: **end while**
 - 9: Retourner les meilleurs éléments de la population
-

Parmi les tous premiers algorithmes de ce type, les algorithmes génétiques introduits par Holland [90] et popularisés par Goldberg [79] adoptent un codage binaire, dans leur version canonique. Avec ce codage, un individu est représenté par une séquence de bits, le croisement

entre deux individus se fait en échangeant une partie des bits de la séquence, et la mutation en modifiant un ou plusieurs bits.

Selon les problèmes traités, le codage des individus peut toutefois prendre d'autres formes. Notamment, dans le cas où on optimise une fonction à variables réelles, on adopte un codage réel et des opérateurs de croisement et de mutation spécifiques (ex. croisement barycentrique).

Au-delà de leur principe de base, assez simple, le bon fonctionnement et l'efficacité des algorithmes évolutionnaires reposent sur de nombreux raffinements, concernant par exemple la définition des opérateurs de mise à l'échelle (*scaling*), ou de rééquilibrage (*sharing*) pour la modification de l'adaptation brute, ou encore le choix du principe de sélection (avec ou sans élitisme, tirage aléatoire des parents), ou le choix des opérateurs de croisement et de mutation. Leurs performances dépendent également de leur bon paramétrage. Ces différents aspects sont détaillés dans [55].

2.5.2 Métaheuristiques pour l'optimisation en domaine continu

Les essais particuliers et l'évolution différentielle, présentés succinctement dans cette section, sont deux autres exemples de métaheuristiques à population. Elles ont initialement été conçues pour l'optimisation de fonctions à variables continues.

Optimisation par essais particuliers (*Particle Swarm Optimization*)

L'optimisation par essais particuliers (PSO), proposée par Kennedy et Eberhart [99, 53, 150], s'inspire du monde du vivant et de ses processus d'auto-organisation, et en particulier des déplacements de groupes d'oiseaux ou de bancs de poissons. L'idée de base est qu'un groupe d'agents munis de règles simples peut faire émerger un "comportement intelligent", au niveau du groupe. Ici, on fixe des règles sur le déplacement individuel des agents, caractérisés par leur position et leur vitesse, en espérant faire émerger un comportement global où l'essaim converge vers un optimum qu'on espère global.

Les étapes du PSO sont décrites dans l'algorithme 9. Les positions et vitesses initiales de l'essaim sont générées aléatoirement. L'algorithme met ensuite à jour les vecteurs vitesse, à chaque itération, en tenant compte d'un facteur d'inertie (paramétré par ω), et de facteurs d'attraction (c_1 et c_2) vers la meilleure position trouvée par la particule considérée, et vers la meilleure position trouvée par l'essaim.

Algorithm 9 *Particle swarm optimization (PSO)*

Require: $popsiz$, ω , c_1 , c_2 ▷ Paramètres de l'algorithme PSO
1: $P_0 \leftarrow \text{INITPOP}(popsiz)$ ▷ Initialiser positions x_i et vitesses v_i , pour chaque individu i de la population
2: **repeat**
3: **for** $i = 1$ to $popsiz$ **do**
4: $v_i \leftarrow \omega v_i + u(0, c_1) \otimes (b_i - x_i) + u(0, c_2) \otimes (g - x_i)$
5: $x_i \leftarrow x_i + v_i$
6: $b_i \leftarrow \text{UPDATEPRIVATEBEST}(b_i, x_i)$ ▷ Meilleur point sur le trajet de la particule i
7: **end for**
8: $g \leftarrow \text{UPDATEGLOBALBEST}(g, b_1, \dots, b_{popsiz})$ ▷ Meilleur point rencontré par l'essaim
9: **until** STOP()
10: **return** g

Dans l’algorithme 9, $u(0, c)$ représente un vecteur aléatoire de dimension n dont les composantes sont tirées au hasard, de manière indépendante, entre 0 et c . L’opération \otimes dénote la multiplication terme à terme entre deux vecteurs. La fonction `UPDATEPRIVATEBEST` évalue la position courante x_i de la particule i , et remplace b_i par x_i si $f(x_i) < f(b_i)$. La fonction `UPDATEGLOBALBEST` met à jour le meilleur point g trouvé par l’essaim.

Dans sa version originale, l’algorithme a tendance à converger trop vite vers un optimum local, du fait de l’attraction vers le meilleur point g trouvé par l’essaim. Une amélioration classique consiste à ne plus utiliser g , le meilleur point trouvé par tout l’essaim, et à le remplacer par plusieurs attracteurs définis par des sous-ensembles de l’essaim. On définit donc un attracteur “local” g_i pour chaque particule i , qui correspond au meilleur point trouvé par un sous-groupe ne comprenant pas i . Ce sous-groupe est défini par une “topologie” fixée de l’essaim. Par exemple, pour une topologie en anneau, on prend le meilleur point trouvé par les particules $i - 1$ et $i + 1$ (modulo le nombre de particules).

Par ailleurs, afin d’éviter que le cumul des modifications successives des vecteurs vitesse ne se traduise par des déplacements beaucoup trop grands, ou au contraire trop petits, on borne la vitesse de chaque particule : $\|v_i\| \in [V_{min}, V_{max}]$. Pour ne pas ajouter de paramètres supplémentaires, le choix de V_{min} et V_{max} se fait automatiquement, en fonction de la taille du domaine dans lequel on cherche un optimum (sachant qu’on ne s’interdit pas nécessairement de sortir provisoirement de ce domaine).

Le réglage des paramètres (ici, $popsize$, ω , c_1 , c_2) est crucial pour les performances de l’algorithme. Pour que la dynamique de l’essaim particulière soit favorable à une convergence sans oscillations excessives vers des attracteurs, les paramètres doivent impérativement se trouver dans certaines limites (voir [36, 162]). A l’intérieur de ces limites, le réglage des paramètres peut être différent selon le problème traité.

Le lecteur peut se référer à [133] pour un panorama plus complet des différents mécanismes et des variantes de l’algorithme PSO, et leur analyse.

Évolution différentielle

L’algorithme d’évolution différentielle (DE : *differential evolution*), introduit par Storn et Price [155] est également un algorithme à population, d’une simplicité remarquable. Partant d’une population initiale de vecteurs x_i générés aléatoirement, il consiste à construire, pour chaque élément x_i , un vecteur candidat y_i , qui vient remplacer x_i dans la population si $f(y_i) < f(x_i)$.

Pour construire le vecteur candidat y_i , on remplace, avec une probabilité CR , chaque coordonnée du vecteur x_i par une combinaison de 3 vecteurs pris au hasard dans la population, en s’assurant qu’il y ait toujours au moins une coordonnée de x_i qui soit modifiée. L’opérateur de recombinaison fait appel à un paramètre F , qui est un facteur d’amplification de la différence entre deux des trois vecteurs. L’algorithme 10 récapitule les étapes de la méthode.

Algorithm 10 Évolution Différentielle (DE)

Require: $popsize, CR, F$ ▷ Paramètres de l'Evolution Différentielle
1: $P_0 \leftarrow \text{INITPOP}(popsize)$ ▷ Initialiser les vecteurs x_i
2: **repeat**
3: **for** $i = 1$ to $popsize$ **do** ▷ Pour chaque individu i de la population
4: $(a, b, c) \leftarrow \text{CHOOSE3}(P)$ ▷ Choisir 3 individus a, b, c dans la population
5: $R \leftarrow \text{RANDOMINT}(dim)$ ▷ Choisir une dimension, pour garantir la modification d'au moins une coordonnée de x_i
6: **for** $j = 1$ to dim **do** ▷ Construire un vecteur y_i candidat
7: $r_j \leftarrow \text{RANDOMFLOAT}([0, 1])$
8: **if** $j = R$ or $r_j < CR$ **then**
9: $y_{ij} \leftarrow x_{aj} + F \times (x_{bj} - x_{cj})$
10: **else**
11: $y_{ij} \leftarrow x_{ij}$
12: **end if**
13: **end for**
14: $x_i \leftarrow \text{BESTOF}(x_i, y_i)$ ▷ Remplacer x_i par y_i si $f(y_i) < f(x_i)$
15: **end for**
16: **until** $\text{STOP}()$
17: Retourner le meilleur élément de la population

Là encore, le choix des paramètres ($popsize$, CR et F) est problème-dépendant et joue sur les performances de l'algorithme. Pour une présentation plus complète, on pourra utilement se reporter au livre de Price, Storn, et Lampinen [135].

2.5.3 Prise en compte des contraintes

Les contraintes peuvent être prises en compte de diverses manières, parmi lesquelles on peut citer les méthodes suivantes (voir [119, 142, 37]) :

- rejeter les individus ne satisfaisant pas les contraintes,
- pénaliser les individus ne satisfaisant pas les contraintes,
- maintenir la faisabilité des solutions, par des représentations ou des opérateurs spécifiques (ex. : croisement barycentrique dans un domaine convexe),
- réparer les solutions non admissibles,
- considérer séparément les solutions faisables et les solutions infaisables (ex. optimisation multi-objectifs, avec un critère pour $f(x)$ et un critère pour la violation des contraintes),
- privilégier l'exploration de la frontière du domaine admissible ([143, 144])

Le choix d'une méthode ou d'une autre dépend largement du problème traité. Dans certains cas, il sera par exemple relativement aisé d'initialiser une population de points satisfaisant les contraintes, puis de maintenir la faisabilité des solutions dans les populations successives, soit par l'utilisation d'opérateurs spécifiques, soit en réparant les solutions. Dans d'autres cas, le simple fait de trouver une solution satisfaisant les contraintes peut s'avérer un problème difficile en soi, indépendamment de l'objectif de trouver une solution optimale. Dans ce dernier cas, on pourra préférer une approche par pénalisation.

2.5.4 Métaheuristiques hybrides

Outre la multitude de métaheuristiques que l'on peut trouver dans la littérature, avec leurs versions canoniques et leurs nombreuses variantes, on trouve aussi des méthodes hybrides [26, 25] combinant soit plusieurs métaheuristiques, soit des métaheuristiques et des méthodes exactes.

On caractérise ces métaheuristiques hybrides selon plusieurs critères (voir [98, 136]), selon le type de méthodes combinées (exactes, métaheuristiques, méthodes spécifiques au problème, etc), comment est réalisée l'hybridation (de bas niveau, ou de haut niveau), l'ordre d'exécution des méthodes (en séquence, entrelacé, ou en parallèle), et selon la stratégie de contrôle adoptée (intégrative, ou collaborative).

Dans ce cadre, nous avons proposé une méthode parallèle coopérative [17, 18] hybridant un algorithme évolutionnaire et une méthode exacte de *branch & bound* par intervalles (voir [82] pour les méthodes d'intervalles, ou la section 2.3.4 pour une courte présentation). Cette approche combine une métaheuristique, capable d'explorer l'espace des points admissibles et de trouver de très bonnes solutions (quoique sans les garantir), avec une méthode exacte par intervalles, capable de réduire le domaine admissible. Cette réduction du domaine admissible se fait par élimination des "boîtes" (vecteurs d'intervalles) dont l'image par une fonction d'inclusion est moins bonne que la meilleure solution trouvée, celle-ci étant mise à jour aussi bien par la métaheuristique que par la méthode d'intervalles. La coopération des deux méthodes permet de prouver l'optimalité de solutions à des problèmes d'optimisation difficiles, pour des fonctions à variables réelles, non convexes, fortement multimodales. Ces travaux préliminaires, où la méthode par intervalles était particulièrement basique, ont par la suite été poursuivis et très largement améliorés par les travaux de thèse de Charlie Vanaret [167, 164].

2.6 Application à quelques problèmes de trafic aérien

Pour clore ce chapitre sur les méthodes d'optimisation, la présente section 2.6 décrit succinctement nos travaux sur l'application de certaines de ces méthodes à quelques problèmes de trafic aérien. Certains de ces travaux sont détaillés dans un chapitre du livre Métaheuristiques [15], coordonné par Patrick Siarry, ou sont abordés plus en profondeur dans d'autres chapitres du présent mémoire (ex. partitionnement optimal en secteurs de contrôle, au chapitre 4). Seuls quelques travaux exploratoires sur l'application à la résolution de conflits aériens de métaheuristiques pour l'optimisation en domaine continu sont donc ici un peu plus développés. Cependant, l'essentiel des contributions faisant l'objet de ce mémoire est présenté aux chapitres 4 et 5.

2.6.1 Réseau de routes-3D séparées

Les travaux présentés dans les publications [71, 69, 70, 62], portaient sur la construction d'un réseau de routes-3D ("tubes") séparées dédiées aux plus gros flux de trafic. La méthode utilisée était un algorithme évolutionnaire, hybridé avec un A^* . Il s'agit d'une hybridation de bas niveau, où l' A^* est intégré dans l'opérateur de mutation de l'algorithme évolutionnaire. Une variante de l'algorithme hybride, incluant une méthode gloutonne, a également été testée.

Ces travaux, essentiellement réalisés durant ma thèse de doctorat, ne sont pas l'objet du présent mémoire, et ne sont donc pas détaillés ici. Ils ne sont rappelés qu'en tant qu'illustration

d'une application à la gestion du trafic aérien des méthodes présentées précédemment dans ce chapitre.

2.6.2 Partitionnement optimal de l'espace aérien

Les méthodes de *branch and bound* présentées dans la section 2.3.4 peuvent s'appliquer directement au partitionnement optimal de l'espace aérien en secteurs de contrôle. Le contexte opérationnel de la gestion des ouvertures de secteurs est décrit dans la section 1.8.4 du chapitre 1. Rappelons simplement que l'espace aérien géré par un centre de contrôle est divisé en volumes élémentaires (secteurs d'espace), qui sont affectés aux postes de travail des contrôleurs aériens. Un poste de travail gère donc un *secteur de contrôle* constitué d'un ou plusieurs *secteurs d'espace*.

Il s'agit donc de construire une partition de l'ensemble des secteurs d'espace, de façon à équilibrer la charge de travail entre les positions de contrôle, tout en respectant un certain nombre de contraintes (respecter le nombre maximum de positions ouvrables, éviter les surcharges, etc). Cette application est très proche de l'exemple de la Figure 2.2, à ceci près que l'on peut en plus couper certaines branches de l'arbre, lors de la construction des partitions, lorsque certains nœuds ne respectent pas les contraintes fixées. Cette application nécessite de plus un modèle de la charge de travail des contrôleurs aériens, lequel peut être appris en utilisant des techniques d'apprentissage supervisé (voir chapitre 3). Nos contributions sur cette application sont détaillées dans le chapitre 4.

2.6.3 Résolution de conflits aériens

Le problème de résolution de conflits aériens, présenté dans la section 1.7.2 du chapitre 1, est un problème fortement combinatoire pour lequel trouver, en un temps raisonnable et avec des hypothèses réalistes, une solution qui soit un optimum global reste très difficile.

On adopte ici une approche centralisée, où les manœuvres sont données par un acteur ayant pleine visibilité sur tout le trafic. Dans ce cadre, les travaux existants présentant des méthodes capables de traiter des instances de l'ordre d'une vingtaine d'avions en conflit, se limitent pour l'essentiel à [130, 30], avec des méthodes de programmation linéaire mixte, et à [51, 52, 49], avec des algorithmes évolutionnaires. Une approche par programmation par contrainte est également proposée, et comparée aux algorithmes évolutionnaires, dans [14].

Dans l'approche par programmation linéaire mixte, comme [130], les hypothèses sur la vitesse des avions sont fortes et peu réalistes dans le cadre d'une utilisation opérationnelle : vitesses constantes pour des résolutions par évitement latéral, ou bien changements de vitesses instantanés pour des résolutions en vitesse. Dans [51], un algorithme génétique fait évoluer une population d'individus encodant des manœuvres de déviation latérales pour les avions initialement en conflit. La modélisation proposée s'appuie sur des variables discrétisées : les temps de début et fin de manœuvres sont choisis avec un pas de temps fixé, et les angles de déviation sont pris parmi des valeurs discrètes. Dans [49], une modélisation similaire est utilisée, avec un algorithme de colonies de fourmis. Les algorithmes évolutionnaires donnent des résultats très convaincants, avec des hypothèses relativement réalistes, permettant notamment la prise en compte d'incertitudes sur les vitesses des avions. Cependant, les équipements aujourd'hui présents dans les centres de contrôle et à bord des avions ne sont pas prévus pour gérer des manœuvres définies en temps.

Dans [166, 165], notre contribution, par rapport aux travaux précédents de [51, 49], a

consisté à adopter une modélisation en distance plutôt qu'en temps, pour les manœuvres des avions, et à modéliser le problème de résolution de conflits comme un problème d'optimisation sous contraintes en domaine continu, plutôt qu'en domaine discret. Nous appliquons à ce problème des métaheuristiques pour l'optimisation en domaine continu : algorithme évolutionnaire à codage réel, optimisation par essaim particulaire, et algorithme à évolution différentielle (voir section 2.5 pour la description de ces méthodes).

Pour cette première approche avec ces méthodes et cette modélisation, nous nous sommes cantonnés à la résolution dans le plan horizontal, pour des avions volant au même niveau de vol, sans incertitudes sur la vitesse.

Modélisation des manœuvres latérales

Les manœuvres latérales des avions sont définies par des distances, que l'on peut traduire directement en points géographiques directement utilisables par les Flight Management System (FMS) actuels, plutôt que par des temps.

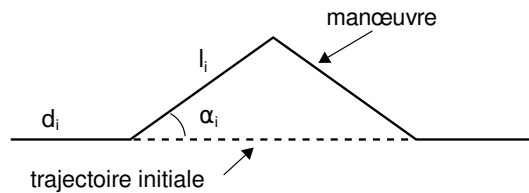


FIGURE 2.4: Modélisation d'une manœuvre latérale, en distance et en angle.

Une manœuvre imposée à un avion i est totalement déterminée par un triplet $x_i = (d_i, \alpha_i, l_i)$ où d_i est la distance (le long de la route initiale) à laquelle débute la manœuvre, α_i est l'angle de déviation, et l_i est la longueur du premier segment de déviation. Le segment de retour se fait par une nouvelle déviation d'angle $(-2\alpha_i)$ jusqu'à interception de la route initiale. La modélisation fournit une suite de points géographiques (Fig. 2.4).

Les modélisations en temps ou en distance sont bien sûr totalement équivalentes, connaissant les vitesses des avions, si l'on ne considère que la trajectoire nominale. Par contre, si l'on considère les incertitudes autour de cette trajectoire nominale, les deux modélisations mènent à des volumes tout à fait différents, comme l'illustre la Figure 2.5.

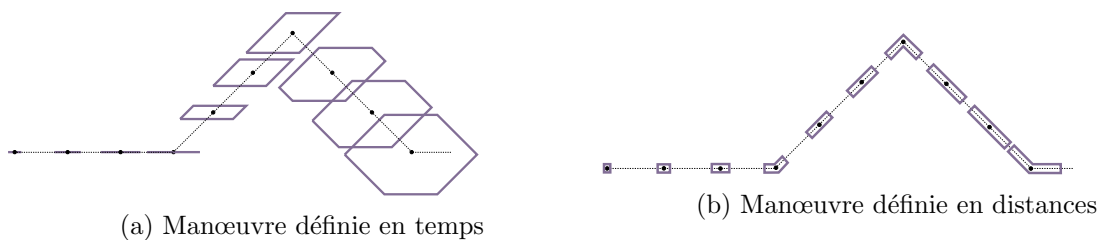


FIGURE 2.5: Zones d'incertitudes, avec modélisation des manœuvres en temps (à gauche) ou en distance (à droite).

Nous n'avons pas pris en compte ces incertitudes dans nos travaux exploratoires, mais c'est ce qui motive ce choix de modélisation en distance. Nous avons supposé que les manœuvres latérales ne seraient plus transmises par radio comme actuellement, mais que les points tournants des manœuvres seraient transmis par liaison de données.

Coût d'une manœuvre latérale

Le coût d'une déviation latérale $x_i = (d_i, \alpha_i, l_i)$ est défini comme suit, où la notation \bar{v} représente la valeur maximale autorisée pour la variable v :

$$\text{cost}(x_i) = \left(\frac{\bar{d}_i - d_i}{\bar{d}_i} \right)^2 + \left(\frac{\alpha_i}{\bar{\alpha}_i} \right)^2 + \left(\frac{l_i}{\bar{l}_i} \right)^2 \quad (2.9)$$

Formalisation du problème de résolution de conflits

Pour n avions, nous avons n manœuvres latérales $x_i = (d_i, \alpha_i, l_i)$ regroupées dans un vecteur $x = (x_1, \dots, x_n)^T$. Le coût total que l'on cherche à minimiser est simplement la somme des coûts des manœuvres individuelles :

$$f(x) = \sum_{i=1}^n \text{cost}(x_i) \quad (2.10)$$

Finalement, le problème de la résolution de conflits peut être formulé de la manière suivante, comme un problème de minimisation sous contraintes :

$$\begin{aligned} \min_{x \in \mathbf{D}} f(x) \quad & \text{sous les contraintes} \\ & \text{dist}(p(x_i, t), p(x_j, t)) > S_h \quad \forall i \neq j, \forall t \end{aligned} \quad (2.11)$$

Dans l'équation 2.11, dist est la distance euclidienne, $p(x_i, t)$ est la position de l'avion i au temps t , étant donnée une manœuvre latérale $x_i = (d_i, \alpha_i, l_i)$ déviant l'avion de sa route initiale. Enfin, S_h est la norme de séparation horizontale, c'est-à-dire la distance minimale qui doit subsister entre deux avions, à tout instant.

Le domaine \mathbf{D} de la variable x est défini par les bornes inférieures $(\underline{d}_i, \underline{\alpha}_i, \underline{l}_i)$, et supérieures $(\bar{d}_i, \bar{\alpha}_i, \bar{l}_i)$ pour chaque déviation latérale $x_i = (d_i, \alpha_i, l_i)$.

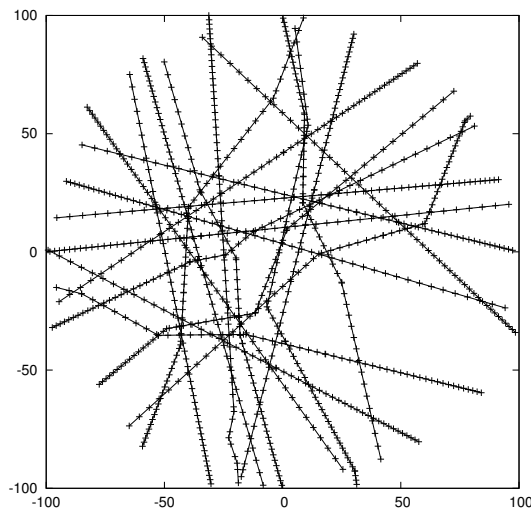


FIGURE 2.6: Exemple de conflit à 20 avions résolu par l'algorithme d'évolution différentielle

Les détails de chacune des méthodes appliquées à ce problème, ainsi que les résultats comparatifs, sont donnés dans [165, 164]. La Figure 2.6 donne un exemple de résolution d'un conflit

à 20 avions par l'algorithme d'évolution différentielle qui, dans ces travaux exploratoires, a donné les meilleurs résultats.

Cette modélisation a par la suite été étendue par Charlie Vanaret, dans le cadre de sa thèse [164], en tenant compte de l'expression de la contrainte de séparation sous forme d'inégalité universellement quantifiée. En appliquant une méthode hybridant un algorithme d'évolution différentielle et un *branch and bound* par intervalles, il trouve des solutions optimales certifiées, pour des cas à deux ou trois avions.

2.7 Conclusion

Ce chapitre était consacré aux méthodes d'optimisation, et plus spécifiquement à la description de méthodes de recherche arborescente, de méthodes d'optimisation locale en domaine continu, et de métaheuristiques, standard ou hybrides. Quelques applications à des problèmes de trafic aérien ont été présentées, dont principalement la résolution de conflits par des métaheuristiques en domaine continu.

Les quelques pistes évoquées dans ce chapitre méritent certainement d'être explorées plus avant. Elle ne constituent cependant pas la contribution principale de ce mémoire. Le chapitre 3 suivant présente les principes de l'apprentissage supervisé, et quelques méthodes, qui sont ensuite appliqués à la gestion prévisionnelle de l'espace aérien, dans le chapitre 4, et à des problématiques autour des trajectoires d'avions, au chapitre 5. Comme nous aurons l'occasion de le voir au chapitre 3, l'optimisation joue par ailleurs un rôle essentiel dans les diverses méthodes d'apprentissage.

Chapitre 3

Principes et méthodes d'apprentissage supervisé

Sommaire

3.1	Notations	58
3.2	Introduction	59
3.3	Notions générales	61
3.3.1	Fonction de perte et minimisation du risque	61
3.3.2	Apprentissage à partir d'un ensemble fini d'exemples	64
3.3.3	La régularisation	66
3.3.4	Le sur-apprentissage	67
3.3.5	Le compromis biais-variance	70
3.3.6	Le fléau de la dimension	71
3.4	Principes d'inférence	72
3.4.1	Choix de représentation	72
3.4.2	Principes d'inférence et choix de la fonction de perte	74
3.4.3	Quel principe d'inférence choisir?	79
3.5	Cadre méthodologique de la sélection de modèle	80
3.5.1	Estimation empirique de l'erreur de généralisation	81
3.5.2	Les critères d'information	83
3.5.3	Quel méthode de sélection choisir?	85
3.6	Réduction de la dimensionalité	86
3.6.1	Sélection de variables pertinentes	87
3.6.2	Transformation des variables de base	88
3.7	Les modèles linéaires	92
3.7.1	Les modèles linéaires pour la régression	92
3.7.2	Les modèles linéaires pour la classification	93
3.7.3	Ajustement des poids d'un modèle linéaire	95
3.7.4	Propriétés statistiques	98
3.7.5	Extension à des combinaisons linéaires de fonctions de base	99
3.7.6	Les modèles linéaires généralisés	100
3.8	Réseaux de neurones	101
3.8.1	Le perceptron multi-couches	102

3.8.2	Réseaux de neurones pour la régression	105
3.8.3	Réseaux de neurones pour la classification	107
3.8.4	Calcul du gradient de l'erreur par rétro-propagation	111
3.8.5	Apprentissage du réseau	116
3.8.6	Considérations pratiques sur les réseaux de neurones	118
3.9	Gradient Boosting Machines	119
3.9.1	Ajustement incrémental d'un modèle additif	119
3.9.2	Principe général du <i>gradient boosting</i>	120
3.9.3	<i>Gradient tree boosting</i>	120
3.10	Conclusion	124

Ce chapitre est consacré à la présentation de l'apprentissage artificiel, et plus particulièrement de l'apprentissage supervisé, ainsi que de quelques-unes de ses méthodes : essentiellement celles que nous avons employées dans les travaux présentés dans les chapitres suivants.

Ce chapitre est relativement long, mais le domaine de l'apprentissage artificiel est particulièrement vaste comme on peut en juger par les quelques 700 pages d'ouvrages de référence comme [86] ou [24], ou les plus de 800 pages de [40]. L'objectif est que le lecteur non spécialiste puisse appréhender le présent document sans avoir à lire ces ouvrages de référence. Ce chapitre ne présente toutefois aucune contribution au domaine, et le lecteur averti pourra donc sans problème passer directement aux chapitres suivants.

3.1 Notations

L'apprentissage artificiel cherche à approximer par un modèle un phénomène observé, l'objectif étant d'utiliser ensuite ce modèle pour faire des prévisions sur ce phénomène. L'apprentissage supervisé suppose l'existence d'un "oracle" dont on observe les réponses à des entrées que nous caractériserons pas la lettre x dans la suite. Les réponses de l'oracle seront quant à elles caractérisées par la lettre y . Le but est d'apprendre un modèle dont les sorties \hat{y} calculées à partir des entrées x soient aussi proches que possible des réponses y de l'oracle.

S'il y a lieu et dans la mesure du possible, on distinguera par des notations distinctes les variables scalaires, les vecteurs, les variables aléatoires, et les matrices ou vecteurs regroupant les exemples observés :

- Par exemple, \mathbf{x} désignera un vecteur regroupant D variables explicatives constituant une entrée : $\mathbf{x} = (x_1, \dots, x_D)^T$.
- La collection $\mathbf{x}_1, \dots, \mathbf{x}_N$ désigne les N vecteurs d'entrée constituant, avec les réponses observées y_1, \dots, y_N l'ensemble d'exemples à partir desquels on cherche à apprendre un modèle de prédiction.
- Lorsqu'on utilisera une représentation basée sur les probabilités, on désignera par X la variable aléatoire dont les \mathbf{x}_n sont des réalisations.
- Les N observations des vecteurs d'entrée de dimension D sont collectées sous la forme d'une matrice, qu'on notera en gras pour la distinguer de la variable aléatoire : par exemple \mathbf{X}_c est la matrice des entrées observées, centrées sur leur moyenne empirique.
- On notera x_{nd} la valeur de l'élément situé sur la $n^{\text{ième}}$ ligne et la $d^{\text{ième}}$ colonne de cette matrice, autrement dit la valeur de la $d^{\text{ième}}$ variable explicative, dans le $n^{\text{ième}}$ exemple observé $\mathbf{x}_n = (x_{n1}, \dots, x_{nd}, \dots, x_{nD})^T$, lequel est situé sur la $n^{\text{ième}}$ ligne de la matrice.

Pour ce qui concerne les probabilités, nous adopterons les notations suivantes :

\mathbb{P} une probabilité

\mathbb{P}_X loi de probabilité d'une variable aléatoire X

p une densité de probabilité

p_X la densité de probabilité associée à la variable X , telle que $\mathbb{P}(X \leq x) = \int_{-\infty}^x p_X(t)dt$

\mathbb{E} espérance mathématique

Lorsque le contexte ne prête pas à confusion, nous utiliserons des notations concises en notant par exemple $\mathbb{P}(y|x)$ la probabilité conditionnelle $\mathbb{P}(Y = y|X = x)$, ceci afin d'alléger les formules présentées. De même, une densité conditionnelle $p_{Y|X=x}(y) = \frac{p_{X,Y}(x,y)}{p_X(x)}$ (où $p_X(x) = \int p_{X,Y}(x,t)dt$) pourra être simplement notée $p(y|x)$.

3.2 Introduction

Qu'est-ce que l'apprentissage artificiel ?

En termes assez généraux, l'apprentissage par induction consiste à trouver les lois générales les plus plausibles qui expliquent des phénomènes particuliers que l'on observe. Un tel processus d'apprentissage suppose donc de collecter des observations du phénomène étudié, et d'en extraire un modèle de prévision qu'on espère suffisamment général pour fournir par la suite des prévisions correctes sur de nouvelles instances.

Les problèmes d'apprentissage artificiel peuvent être de différentes natures : apprentissage supervisé, avec des problèmes de régression ou de classification, apprentissage non supervisé, ou apprentissage par renforcement. Ce dernier s'appuie sur un mécanisme d'essais-erreurs pour choisir une politique d'action qui maximise une espérance de gain. Dans l'apprentissage non supervisé, on cherche soit à regrouper des observations $\{x_1, \dots, x_N\}$ en fonction de leurs similarités, soit à approximer une densité de probabilité à partir de ces observations.

L'apprentissage supervisé

Dans l'apprentissage supervisé, qui nous intéresse plus particulièrement dans ce chapitre, on suppose l'existence d'une relation $y = f(x)$, ou $y = f(x) + \text{epsilon}$ si la réponse de l'oracle est bruitée, entre une entrée $x \in \mathcal{X}$ et une réponse $y \in \mathcal{Y}^1$, laquelle relation n'est pas connue.

On cherche donc à approximer cette relation inconnue, par exemple en cherchant une fonction h prise dans un espace d'hypothèses \mathcal{H} à définir préalablement, telle que h approxime au mieux la fonction f inconnue sur des exemples observés. Il faut donc disposer d'un ensemble d'exemples $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$ donnant les réponses y_n attendues pour les entrées x_n , $n \in \{1, \dots, N\}$.

En adoptant le formalisme des probabilités, les exemples (x_n, y_n) sont des tirages aléatoires selon une loi de probabilité jointe $\mathbb{P}_{X,Y}$ de deux variables aléatoires X et Y . Dans ce cadre, extraire les lois générales les plus plausibles expliquant les observations $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$ revient à estimer la loi jointe $\mathbb{P}_{X,Y}$, ou alternativement la loi conditionnelle $\mathbb{P}_{Y|X=x}$ en tout point x , à partir des données S . Si l'on ne se place pas dans un cadre probabiliste, on peut aussi chercher directement une fonction h^* qui minimise un critère d'erreur empirique sur notre ensemble d'exemples S .

1. x et y peuvent être des vecteurs regroupant plusieurs variables.

Vocabulaire

Pour les variables d'entrée x , on utilisera indifféremment le terme de variables explicatives, variables indépendantes, formes², ou attributs. Pour les variables de sortie y , on utilisera les termes de réponses, de sorties, de variables dépendantes, ou encore de labels ou d'étiquettes lorsque y est une variable catégorielle.

Pour la fonction $h \in \mathcal{H}$ par laquelle on cherche à approximer la “vraie fonction” f , on utilisera le terme d'hypothèse, comme dans [40], ou celui de modèle de prévision, ou plus simplement de modèle, comme dans [86, 24]. L'espace \mathcal{H} est par la suite appelé *espace des hypothèses*.

Problèmes de régression ou de classification

Les problèmes d'apprentissage supervisés sont divisés en deux grandes catégories, selon la nature de la variable réponse. Les problèmes de *régression* s'intéressent au cas où la réponse est continue : $y \in \mathbb{R}$ ou $y = (y_1, \dots, y_P)^T \in \mathbb{R}^P$. Dans un problème de *classification*, on cherche à affecter chaque entrée x à une catégorie, et la variable réponse est donc catégorielle $y \in \{\mathcal{C}_1, \dots, \mathcal{C}_P\}$.

Apprentissage en-ligne ou hors-ligne

Par ailleurs, selon la façon de collecter et d'exploiter les observations, on parlera d'apprentissage *en-ligne* (*online learning*) ou *hors-ligne* (*batch learning*). Dans le premier cas, le modèle de prévision est affiné à chaque observation, après découverte de la “vraie” valeur de la réponse. Dans le deuxième cas, l'ensemble des entrées et des réponses désirées est connu à l'avance, et on l'exploite en entier pour construire le modèle³.

Dans les problèmes que nous avons traités, les données étaient collectées au préalable, et nous nous intéresserons donc exclusivement à l'apprentissage hors-ligne.

Organisation du chapitre

Le présent chapitre commence par exposer les notions générales sur l'apprentissage supervisé (section 3.3), avant d'aborder, dans la section 3.4 les principes d'inférence permettant d'apprendre à partir d'exemples. Le cadre méthodologique de la sélection de modèle est décrit dans la section 3.5, suivie d'une courte présentation des techniques permettant de réduire la dimensionalité des variables explicatives des modèles (section 3.6). Les trois dernières sections entrent dans le détail de quelques méthodes d'apprentissage : les modèles linéaires dans la section 3.7, les réseaux de neurones dans la section 3.8, et enfin les *gradient boosting machines* dans la section 3.9.

2. Au sens de la traduction du terme anglais *feature*.

3. Notons qu'on peut aussi traiter les exemples par blocs, voire séquentiellement, sans pour autant être dans le cadre d'un apprentissage *en-ligne*.

3.3 Notions générales

3.3.1 Fonction de perte et minimisation du risque

L'apprentissage supervisé consiste à sélectionner, dans un espace d'hypothèses \mathcal{H} que l'on se fixe, la fonction $h \in \mathcal{H}$ qui approxime au mieux la fonction f inconnue reliant les entrées x aux sorties y . Pour cela, on définit au préalable une fonction de perte $\ell(y, h(x))$ représentant le coût associé à l'erreur entre la prévision $h(x)$ du modèle et la vraie valeur $y = f(x)$ de la réponse.

Pour une perte ℓ donnée, le modèle de prévision qui généralisera le mieux sur de nouvelles instances de x est la fonction h^* minimisant l'espérance mathématique $\mathbb{E}_{X,Y}[\ell(Y, h(X))]$, où X et Y sont les variables aléatoires produisant les instances x et y . Cette espérance de perte définit le risque associé au choix de l'hypothèse h :

Définition 3.1 (Risque réel). *Étant données une fonction de perte ℓ et une loi de probabilité jointe $\mathbb{P}_{X,Y}$ sur $\mathcal{X} \times \mathcal{Y}$, le risque (ou risque réel) associé au choix de l'hypothèse $h \in \mathcal{H}$ est défini comme l'espérance de la perte ℓ sur $\mathcal{X} \times \mathcal{Y}$:*

$$\mathcal{R}(h) = \mathbb{E}_{X,Y}[\ell(Y, h(X))] = \int_{\mathcal{X} \times \mathcal{Y}} \ell(y, h(x)) \mathbb{P}_{X,Y}(dx, dy) \quad (3.1)$$

En faisant apparaître la loi conditionnelle $\mathbb{P}_{Y|X=x}$, on peut réécrire l'expression (3.1) du risque réel $\mathcal{R}(h)$ comme indiqué dans l'équation (3.2).

$$\mathcal{R}(h) = \mathbb{E}_X \mathbb{E}_{Y|X}[\ell(Y, h(X))] = \int_{\mathcal{X}} \underbrace{\left[\int_{\mathcal{Y}} \ell(y, h(x)) \mathbb{P}_{Y|X=x}(dy) \right]}_{\rho(h,x)} \mathbb{P}_X(dx) \quad (3.2)$$

L'espérance conditionnelle de perte $\mathbb{E}[\ell(Y, h(X)) | X = x]$ correspondant à l'expression entre crochets dans l'intégrale de l'équation (3.2) peut être vue comme une fonction $\rho(h, x)$, donnée par l'équation (3.3) :

$$\rho(h, x) = \mathbb{E}[\ell(Y, h(X)) | X = x] = \int_{\mathcal{Y}} \ell(y, h(x)) \mathbb{P}_{Y|X=x}(dy) \quad (3.3)$$

L'optimum du risque $\mathcal{R}(h)$ est atteint dès lors qu'on minimise $\rho(h, x)$ en chaque point x . Construisons artificiellement la fonction h_B^* , de telle façon qu'elle coïncide en chaque point x avec l'hypothèse h qui minimise $\rho(h, x)$ au point x , comme illustré sur la Figure 3.1.

Plus formellement, la fonction h_B^* est définie point par point, par l'équation (3.4) :

$$\forall x \in \mathcal{X} \quad h_B^*(x) = \arg \min_{h \in \mathcal{H}} \rho(h, x) = \arg \min_{c \in \{h(x) | h \in \mathcal{H}\}} \mathbb{E}_{Y|X=x}[\ell(Y, c)] \quad (3.4)$$

En supposant que $h_B^* \in \mathcal{H}^4$, par construction $\mathcal{R}(h_B^*)$ est le minimum de $\mathcal{R}(h)$, et aucune autre fonction $h \in \mathcal{H}$ ne donnera une meilleure valeur de l'espérance de perte que h_B^* . Cette fonction h_B^* est parfois appelée *estimateur bayésien* dans le cas général, ou *règle de décision de Bayes* en classification, ou encore *fonction de régression*, dans le cas de la régression avec perte quadratique.

4. Ce n'est pas forcément le cas en pratique, mais nous supposons dans cette section, pour la clarté de l'exposé, qu'on peut étendre artificiellement l'espace d'hypothèses \mathcal{H} de façon à ce qu'il contienne l'estimateur bayésien h_B^* .

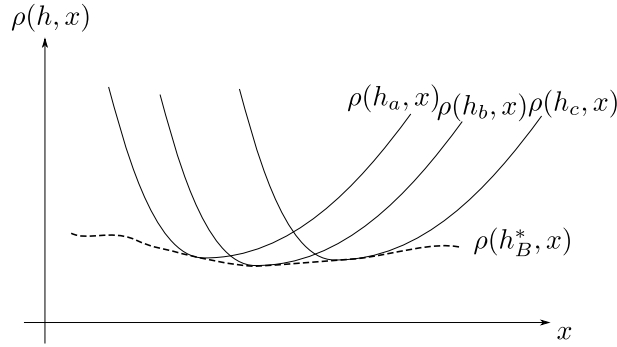


FIGURE 3.1: Minimisation du risque en chaque point x , et définition de l'estimateur bayésien h_B^* .

Illustration pour la classification

Considérons, à titre d'illustration, un problème de classification à P classes, avec une variable réponse catégorielle $y \in \{\mathcal{C}_1, \dots, \mathcal{C}_P\}$. La fonction de perte ℓ est ici simplement représentée par une matrice de perte L , telle que $\ell(y, h(x)) = L_{kj}$ soit le coût de l'erreur faite lorsque l'entrée x , appartenant en réalité à la classe $y = \mathcal{C}_k$, est affectée par notre modèle de prévision à la classe $h(x) = \mathcal{C}_j$.

Pour y pris dans un ensemble fini de valeurs discrètes $\{\mathcal{C}_1, \dots, \mathcal{C}_P\}$, l'espérance conditionnelle apparaissant dans l'équation (3.2) est simplement la somme des pertes pondérées par les valeurs des probabilités conditionnelles d'occurrence des événements $Y = \mathcal{C}_k$ sachant que $X = x$.

L'équation (3.2) donnant $\mathbb{E}[\ell(Y, h(X)) | X = x]$, l'espérance de perte sachant x , devient alors l'équation (3.5), où l'on note $\mathbb{P}(\mathcal{C}_k | x)$ la probabilité $\mathbb{P}(Y = \mathcal{C}_k | X = x)$, afin d'alléger les notations.

$$\rho(h, x) = \sum_{k=1}^P \underbrace{\ell(\mathcal{C}_k, h(x))}_{L_{kj}} \mathbb{P}(\mathcal{C}_k | x) \quad (3.5)$$

La règle de décision de Bayes consiste alors à choisir la classe \mathcal{C}_j telle que la somme des pertes L_{kj} , pondérées par les probabilités d'occurrence de chacune des classes \mathcal{C}_k , soit minimale :

$$\forall x \in \mathcal{X} \quad h_B^*(x) = \arg \min_{j \in \{1, \dots, P\}} \sum_{k=1}^P L_{kj} \mathbb{P}(\mathcal{C}_k | x) \quad (3.6)$$

Pour un problème à seulement deux classes, on a nécessairement $\mathbb{P}(\mathcal{C}_k | x) = 1 - \mathbb{P}(\mathcal{C}_j | x)$ pour $k \neq j$. Avec une perte binaire $L = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, la décision de Bayes se résume alors à choisir, en tout point x , la classe \mathcal{C}_j telle que $1 - \mathbb{P}(\mathcal{C}_j | x)$ soit minimum, c'est-à-dire $\mathbb{P}(\mathcal{C}_j | x)$ maximum :

$$\forall x \in \mathcal{X} \quad h_{\text{Binaire}}^*(x) = \arg \max_{\mathcal{C}_j \in \{\mathcal{C}_1, \mathcal{C}_2\}} \mathbb{P}(\mathcal{C}_j | x) \quad (3.7)$$

Cas de l'existence d'une densité de probabilité

Dans le cas où y est une variable continue, le calcul de l'espérance mathématique se fait par intégration sur le domaine \mathcal{Y} . Si la variable aléatoire $Z = (X, Y)$ admet une densité de probabilité jointe $p_{X,Y}$, on peut écrire $\mathbb{P}_{X,Y}(dx, dy) = p_{X,Y}(x, y) dx dy$, et l'équation (3.1) définissant le risque $\mathcal{R}(h)$ devient l'équation (3.8).

$$\mathcal{R}(h) = \mathbb{E}_{X,Y}[\ell(Y, h(X))] = \int_{\mathcal{X}} \int_{\mathcal{Y}} \ell(y, h(x)) p_{X,Y}(x, y) dy dx \quad (3.8)$$

Soit p_X la densité associée à la variable aléatoire X . Rappelons que si la densité jointe $p_{X,Y}$ existe, alors p_X existe et est définie par $p_X(x) = \int_{\mathcal{Y}} p(x, y) dy$. En introduisant la densité conditionnelle $p_{Y|X=x}$ définie par $p_{Y|X=x}(y) = \frac{p(x,y)}{p_X(x)}$, l'équation (3.2) devient l'équation (3.9).

$$\mathcal{R}(h) = \mathbb{E}_X \mathbb{E}_{Y|X}[\ell(Y, h(X))] = \int_{\mathcal{X}} \underbrace{\left[\int_{\mathcal{Y}} \ell(y, h(x)) p_{Y|X=x}(y) dy \right]}_{\rho(h,x)} p_X(x) dx \quad (3.9)$$

Notons que, par définition des densités de probabilités $\int_{\mathcal{Y}} p_{Y|X=x}(y) dy = 1$

Illustration pour la régression

Considérons maintenant un problème de régression, où la variable réponse $y \in \mathbb{R}$ est réelle. Supposons qu'on ait choisi une perte quadratique $\ell(y, h(x)) = [y - h(x)]^2$. En tenant compte de ce que nous venons de voir sur les densités de probabilités, l'expression (3.4) de l'estimateur bayésien devient l'équation (3.13) suivante, où $p_{Y|X=x}$ est la densité conditionnelle sachant que $X = x$:

$$\forall x \in \mathcal{X} \quad h_B^*(x) = \arg \min_{h \in \mathcal{H}} \rho(h, x) = \arg \min_{c \in \{z=h(x) | h \in \mathcal{H}\}} \int_{\mathcal{Y}} (y - c)^2 p_{Y|X=x}(y) dy \quad (3.10)$$

Notons g_x la fonction telle que $g_x(c) = \int_{\mathcal{Y}} (y - c)^2 p_{Y|X=x}(y) dy$. En supposant réunies toutes les conditions pour pouvoir dériver l'expression sous le signe de l'intégrale, la dérivée de $g_x(c)$ par rapport à c s'exprime comme suit :

$$\begin{aligned} \frac{\partial g_x(c)}{\partial c} &= \frac{\partial}{\partial c} \int_{\mathcal{Y}} (y - c)^2 p_{Y|X=x}(y) dy \\ &= \int_{\mathcal{Y}} \frac{\partial (y - c)^2}{\partial c} p_{Y|X=x}(y) dy \\ &= -2 \int_{\mathcal{Y}} (y - c) p_{Y|X=x}(y) dy \end{aligned} \quad (3.11)$$

Le minimum de la fonction $c \rightarrow g_x(c)$ est obtenu en annulant sa dérivée par rapport à c ⁵,

5. On suppose ici que le point où s'annule la dérivée est bien atteint dans $\{h(x) | h \in \mathcal{H}\}$.

c'est à dire pour c^* tel que :

$$c^* \underbrace{\int_{\mathcal{Y}} p_{Y|X=x}(y) dy}_{=1 \text{ (def. densité)}} = \underbrace{\int_{\mathcal{Y}} y p_{Y|X=x}(y) dy}_{\mathbb{E}(Y|X=x)} \quad (3.12)$$

L'estimateur bayésien pour une régression avec perte quadratique est donc la fonction de régression h_B^* retournant la moyenne conditionnelle des valeurs de y au point x :

$$\forall x \in \mathcal{X} \quad h_B^*(x) = \mathbb{E}(Y|X=x) \quad (3.13)$$

Si l'on choisit une fonction de perte $\ell(y, h(x)) = |y - h(x)|$ au lieu d'une perte quadratique, l'estimateur bayésien est la médiane, et non plus la moyenne, des valeurs de y au point x .

3.3.2 Apprentissage à partir d'un ensemble fini d'exemples

Nous venons de voir, en l'illustrant pour la classification et la régression, qu'on peut théoriquement construire une fonction h_B^* minimisant le risque $\mathcal{R}(h)$ de l'équation (3.1). Le problème, pour obtenir en pratique cet estimateur bayésien, est que la loi jointe $\mathbb{P}_{X,Y}$ utilisée dans cette équation n'est pas connue, pas plus que la loi conditionnelle $\mathbb{P}_{Y|X=x}$ des équations (3.2) et (3.3).

En pratique, il faut donc chercher à minimiser le risque $\mathcal{R}(h)$ en se basant uniquement sur l'ensemble fini d'exemples $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$ à partir duquel on cherche à approximer f . L'objectif d'un algorithme d'apprentissage est donc de sélectionner une hypothèse $h \in \mathcal{H}$ qui approxime au mieux f à partir des données S .

Plusieurs principes d'inférence inductive peuvent être appliqués pour construire un algorithme d'apprentissage. Parmi ceux-ci, celui qui vient le plus naturellement à l'esprit consiste à minimiser le *risque empirique*, donné par la définition 3.2 suivante, où l'espérance de perte est calculée empiriquement en prenant la moyenne des pertes sur l'ensemble S .

Définition 3.2 (Risque empirique). *Étant donné une fonction de perte ℓ et un ensemble $S = \{(x_n, y_n) \mid 1 \leq n \leq N\}$ d'exemples pris dans $\mathcal{X} \times \mathcal{Y}$, le risque empirique est défini comme la moyenne empirique des pertes sur l'ensemble S .*

$$\boxed{\mathcal{R}_{emp}(h, S) = \frac{1}{|S|} \sum_{(x_n, y_n) \in S} \ell(y_n, h(x_n))} \quad (3.14)$$

Le principe inductif de minimisation du risque empirique (MRE) consiste donc simplement à retenir l'hypothèse h_S^* donnée par (3.15) :

$$h_S^* = \arg \min_{h \in \mathcal{H}} \mathcal{R}_{emp}(h, S) \quad (3.15)$$

Pour S fixé, la minimisation du risque empirique revient à minimiser l'erreur suivante :

$$\mathcal{E}(h) = \sum_{(x_n, y_n) \in S} \ell(y_n, h(x_n)) \quad (3.16)$$

Difficultés de la minimisation du risque empirique

Telle quelle, l'approche par minimisation du risque empirique sur l'ensemble S comporte un inconvénient majeur. Supposons que l'espace des hypothèses \mathcal{H} soit un ensemble de fonctions suffisamment complexes. Pour fixer les idées, on peut imaginer que \mathcal{H} est par exemple l'ensemble des fonctions polynomiales, de degré arbitrairement grand. Alors il existe un très grand nombre, voire une infinité, de fonctions $h \in \mathcal{H}$ passant par tous les points de l'ensemble d'apprentissage. Ce phénomène est illustré par la Figure 3.2, où x et y sont de dimensions 1. En l'absence d'informations complémentaires ou d'*a priori* sur la "vraie fonction" f , toutes ces hypothèses h se valent, puisqu'elles expliquent toutes parfaitement les données S .

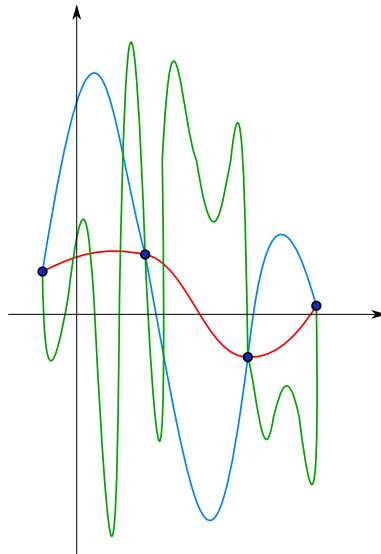


FIGURE 3.2: Quelques exemples de plusieurs fonctions passant par tous les points de l'ensemble d'apprentissage.

Cette difficulté, expliquée ici de façon intuitive, est formalisée par les théorèmes du “*No-Free-Lunch*” pour l'apprentissage, de Wolpert [177].

Pour résumer, on voit que pour pouvoir départager des hypothèses h sélectionnées uniquement sur la base des exemples de S , il nous faut des connaissances ou des *a priori* supplémentaires, hors des données S , sur le phénomène sous-jacent f ayant produit les données S . Ces connaissances ou *a priori*, qu'ils soient implicites ou explicites, dictent la façon dont le passé (les observations collectées dans S) est relié au futur (les nouvelles instances sur lesquelles on effectue une prévision). Ces considérations sont résumées par Bousquet *et al.* [28], par une simple formule :

$$\boxed{\text{Généralisation} = \text{Données} + \text{Connaissances}}$$

Les hypothèses faites sur f peuvent être explicites, comme dans le cas du choix d'une distribution *a priori* dans l'inférence bayésienne, ou implicites, à travers des hypothèses de régularité de la fonction f dans un voisinage local à chaque point (ex. : f constante dans un voisinage défini par les k plus proches voisins). Elles permettent d'imposer des restrictions à

l'espace \mathcal{H} des hypothèses h , par exemple en ordonnant les fonctions h ou en les pondérant selon leur écart présupposé par rapport au comportement effectif de f .

3.3.3 La régularisation

Une des façons de procéder, pour départager des hypothèses h *a priori* équivalentes, consiste à imposer explicitement une certaine régularité à la fonction h . Cette régularité peut être contrôlée à travers un paramètre de régularisation qui pénalise chaque hypothèse selon sa complexité.

La technique de régularisation revient donc à remplacer l'erreur $\mathcal{E}(h)$ de l'équation (3.16) par une fonction d'erreur $\mathcal{E}'(h; \lambda)$ paramétrée par un hyperparamètre λ , et donnée par l'équation (3.17), où $J(h)$ est à définir selon le type de régularisation.

$$\mathcal{E}'(h; \lambda) = \mathcal{E}(h) + \lambda J(h) \quad (3.17)$$

Les régressions linéaires *ridge* ou *lasso*, ou encore la technique du *weight decay* dans les réseaux de neurones, sont des exemples classiques de techniques de régularisation, où le terme $J(h)$ fait simplement appel à la norme L_2 ou L_1 des poids w .

L'expression de l'erreur régularisée $\mathcal{E}'(h; \lambda)$ peut parfois directement découler d'un principe d'inférence choisi, et d'hypothèses sur les distributions de probabilités. La section 3.4.2 montre un exemple où, avec l'hypothèse d'un bruit gaussien et d'une distribution *a priori* gaussienne des poids, le principe de maximum *a posteriori* se traduit par la minimisation d'une erreur quadratique régularisée par un terme $\lambda \sum_{j=1}^D w_j^2$.

Régression linéaire *ridge*

$$h_S^* = \arg \min_{h \in \mathcal{H}} \left[\sum_{i=1}^N \left(y - \sum_{j=1}^D x_{ij} w_j \right)^2 + \lambda \sum_{j=1}^D w_j^2 \right] \quad (3.18)$$

Régression linéaire *lasso*

$$h_S^* = \arg \min_{h \in \mathcal{H}} \left[\sum_{i=1}^N \left(y - \sum_{j=1}^D x_{ij} w_j \right)^2 + \lambda \sum_{j=1}^D |w_j| \right] \quad (3.19)$$

Weight decay dans les réseaux de neurones

$$h_S^* = \arg \min_{h \in \mathcal{H}} \left[\sum_{i=1}^N \ell(y, MLP(w, x)) + \lambda \sum_{j=1}^D w_j^2 \right] \quad (3.20)$$

Dans l'équation (3.20), *MLP* est la fonction réalisée par un réseau de neurones multicouches (*multi-layer perceptron*), détaillé dans la section 3.8 de ce chapitre.

Le problème de la sélection de l'hyperparamètre λ

Notons que l'introduction du paramètre λ ne fait que déplacer le problème du choix de l'hypothèse h parmi plusieurs autres hypothèses *a priori* équivalentes, puisqu'il faut maintenant déterminer quelle valeur choisir pour ce paramètre.

Ce choix peut être fait par validation simple ou croisée à K plis, comme indiqué dans la section 3.5.1.

3.3.4 Le sur-apprentissage

La Figure 3.2 de la section précédente illustre en fait un cas extrême où les modèles h ajustés passent exactement par tous les points de l'ensemble S (apprentissage *par cœur*).

En règle générale, avec des observations (x_n, y_n) de S plus nombreuses et potentiellement entachées d'erreur, le sur-ajustement aux données pourra être moins marqué, et on pourra constater un phénomène de sur-apprentissage qui ressemble plutôt à ce qu'on observe sur la Figure 3.3.

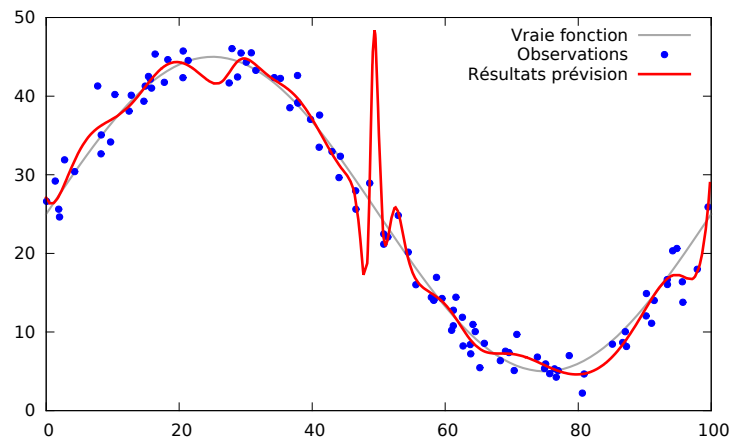


FIGURE 3.3: Exemple de sur-apprentissage, avec un réseau de neurones à une couche cachée de 20 unités.

Dans cet exemple, un réseau de neurones comportant une couche cachée de 20 unités est appris sur un ensemble de 100 observations (points bleus) issues d'une "vraie fonction" f sinusoïdale (courbe grise) bruitée. La courbe rouge est la prévision faite sur 500 autres entrées x n'ayant pas servi à l'apprentissage. Le réseau est appris en minimisant une perte quadratique sur l'ensemble d'apprentissage. Ce modèle, avec 20 unités cachées, est suffisamment "flexible" pour se rapprocher au plus près des points de l'ensemble d'apprentissage, mais il est largement moins performant lorsqu'il s'agit de généraliser la prévision à de nouvelles entrées.

Quantifier à quel point un modèle h est complexe et peut s'adapter aux données n'est pas toujours chose facile. Dans certains cas, cette complexité est en relation directe avec le nombre de paramètres effectifs du modèle. C'est typiquement le cas si h est une combinaison linéaire des entrées x , ou un réseau de neurones : la complexité du modèle sera fonction du nombre de poids à ajuster. Ce n'est toutefois pas toujours le cas et il existe d'autres façon d'évaluer la capacité d'un modèle à s'ajuster aux exemples, comme la dimension de Vapnik-Chervonenkis ([168]), ou la complexité de Rademacher (pour la classification binaire).

Influence de la complexité de h

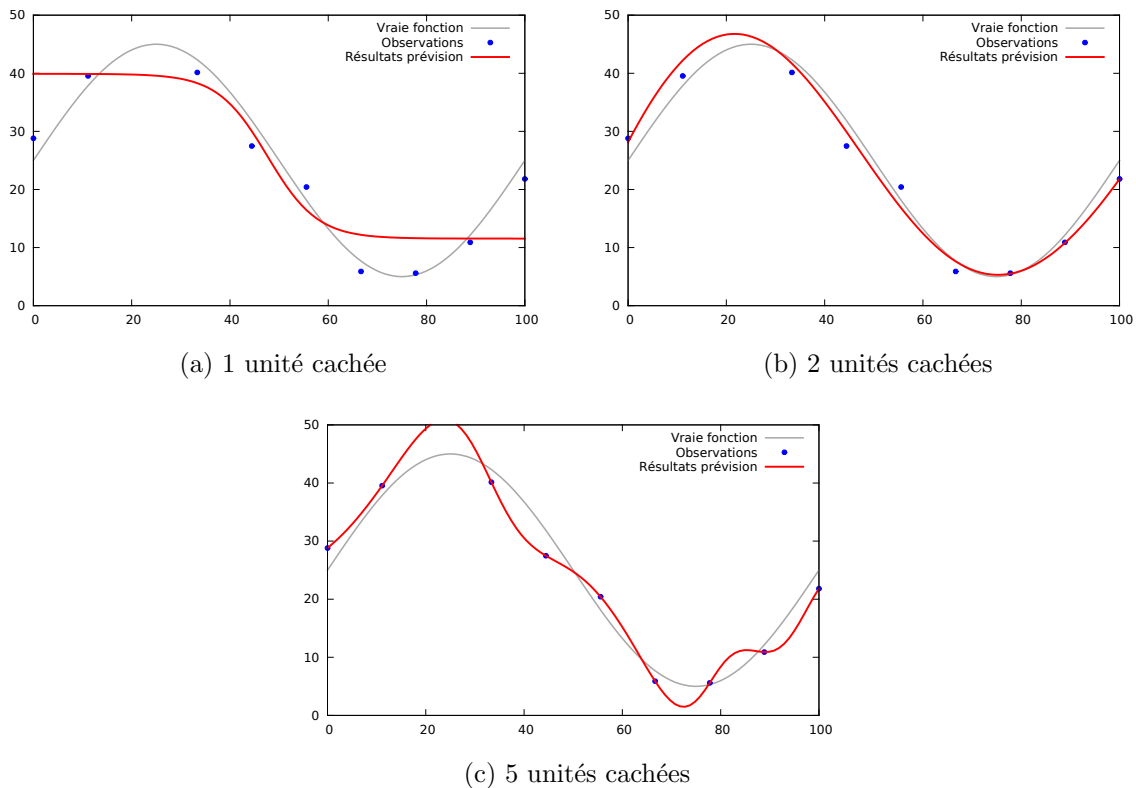


FIGURE 3.4: Réponse (en rouge) d'un réseau de neurones à 1 couche cachée comportant 1, 2, ou 5 unités, appris sur un ensemble d'observations (en bleu) comprenant 10 exemples.

La Figure 3.4 montre les réponses de trois réseaux de neurones à une couche cachée, comprenant respectivement 1, 2, et 5 unités sur la couche cachée. Le nombre de paramètres à ajuster est respectivement de 4, 7, et 16. Dans cet exemple, les 10 entrées x_n sont fixées à l'avance et régulièrement espacées. Seules les réponses y_n comprennent un aléas. Elles sont obtenues en ajoutant un bruit à la fonction f , ici sinusoïdale.

Le modèle le plus simple 3.4a s'ajuste assez mal aux données observées. À l'inverse, le réseau le plus complexe 3.4c passe par tous les points observés. Le meilleur compromis semble être celui de la figure 3.4b. Notons toutefois qu'*in fine*, il s'agit là d'un choix purement subjectif qui correspond à nos présupposés (implicites ou explicites) sur la régularité de la fonction f qu'on cherche à approximer. Si l'on s'en tient aux seules données observées, le vrai phénomène ayant produit les données aurait tout aussi bien pu être plus proche de 3.4c ou de 3.4a.

Influence du nombre d'exemples

On constate sur la Figure 3.4 que le réseau de neurones passe par chacun des 10 points avec seulement 5 unités sur la couche cachée, alors que celui de la Figure 3.3, avec un nombre supérieur de 20 unités cachées, ne passe pas exactement par les 100 points observés, même s'il sur-apprend les données. On devine donc que, selon le nombre d'exemples utilisés pour apprendre le modèle, le phénomène de sur-apprentissage n'aura pas la même ampleur.

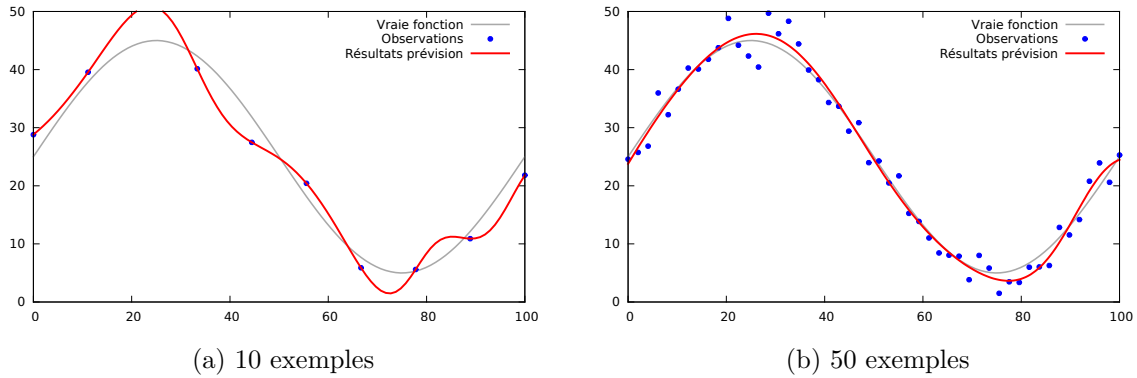


FIGURE 3.5: .

Pour illustrer cette influence du nombre d'exemples, la Figure 3.5 montre la réponse du même réseau de neurones (à 5 unités cachées) sur deux jeux de données, avec 10 exemples (à gauche) ou 50 exemples (à droite). On constate que le sur-apprentissage est d'autant moins marqué que le nombre d'exemples est important.

Pour résumer, intuitivement, le risque de sur-apprentissage diminue avec N grand, N étant est le nombre d'instances de l'ensemble d'apprentissage, et croît avec la complexité γ du modèle (en supposant qu'on dispose d'une telle mesure γ de la complexité).

Évolution des erreurs d'apprentissage et de test selon la complexité du modèle

Pour terminer sur la description du phénomène de sur-apprentissage, illustrons l'évolution typique de l'erreur de prédiction, calculée d'une part sur l'ensemble ayant servi à apprendre le modèle, et d'autre part sur des données séparées n'ayant pas servi à l'apprentissage.

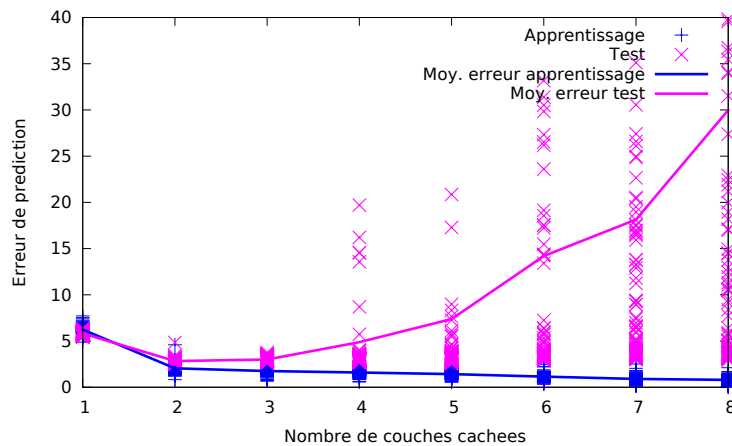


FIGURE 3.6: Évolution des erreurs de prédiction sur l'ensemble d'apprentissage (en bleu) de 20 instances, et sur un ensemble de données de test séparées (en magenta) comprenant 200 instances.

La Figure 3.6 montre l'évolution de ces deux erreurs, pour 100 ensembles d'apprentissage de 20 instances chacun, et 100 ensembles de test, en fonction du nombre d'unités dans la

couche cachée du réseau de neurones, sur le même exemple de sinusoïde que précédemment. L'erreur sur l'ensemble d'apprentissage baisse continuellement avec le nombre de paramètres à ajuster, alors que l'erreur de test commence par baisser pour ensuite augmenter. Par ailleurs, la dispersion des erreurs autour de la moyenne augmente considérablement avec la complexité du modèle.

Cette courbe illustre le compromis biais-variance à trouver entre des modèles simples approximant mal la fonction-cible mais renvoyant des erreurs proches sur les ensembles de d'apprentissage et de test – à gauche de la Figure 3.6 – et des modèles plus complexes capables de mieux s'ajuster aux données observées, mais de forte variance et moins susceptibles de faire des prévisions correctes sur de nouvelles instances – à droite sur la figure.

3.3.5 Le compromis biais-variance

Reprenons les trois mêmes réseaux de neurones de l'exemple de la Figure 3.4, cette fois-ci en considérant 5 tirages aléatoires différents de l'ensemble d'apprentissage. Avec des x_n fixés, cela se traduit simplement par des valeurs différentes des réponses observées y_n . Les différents ensembles d'apprentissage sont reportés sur la Figure 3.4 sous les intitulés ts_1, \dots, ts_5 , ainsi que les réponses des modèles appris sur chacun de ces ensembles.

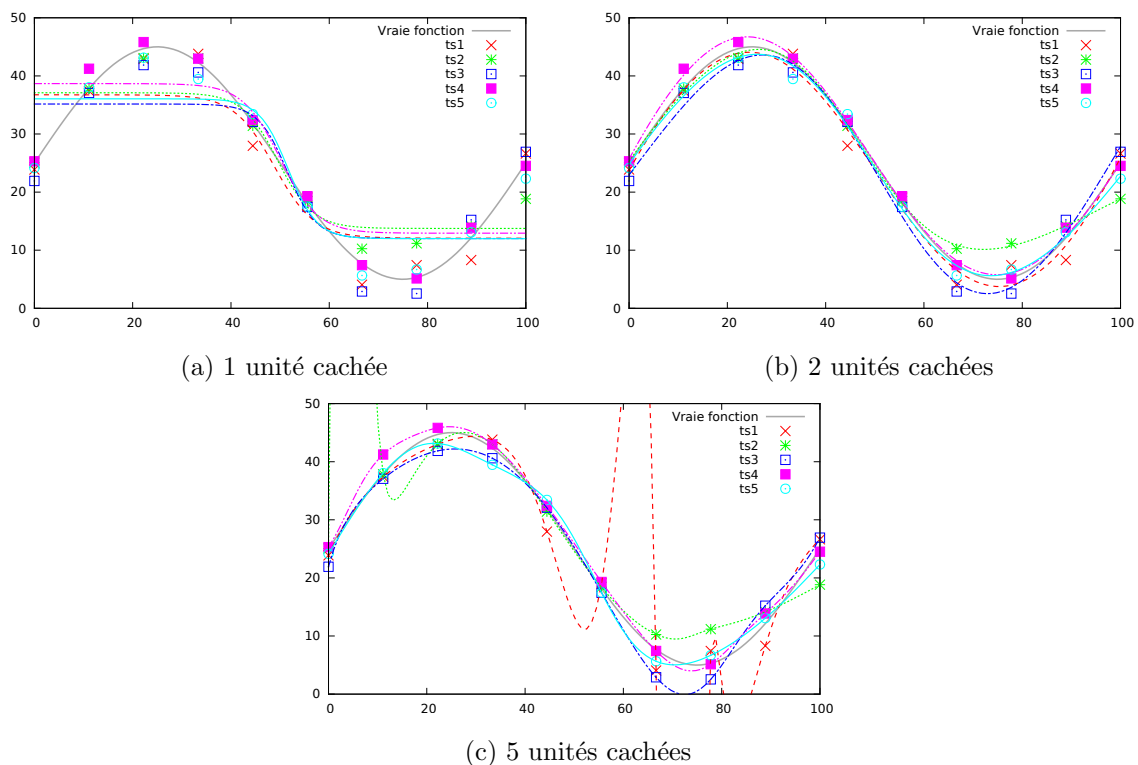


FIGURE 3.7: .

Le modèle le plus simple (3.7a) présente un biais important : il approxime mal la vraie fonction f . Par contre, on a une faible variance des réponses autour de leur moyenne. A contrario, le modèle le plus complexe (3.7c) présente un biais faible, au sens où la moyenne des réponses approxime bien $f(x)$. La variance des réponses des modèles appris sur les ensembles

ts_1, \dots, ts_5 autour de la courbe moyenne est par contre très élevée.

Décomposition de l'erreur de prédiction

Décomposons l'erreur de prédiction en un point x quelconque, en considérant ici une perte quadratique, à titre d'exemple. L'équation (3.21) donne cette décomposition. L'espérance y est calculée sur l'ensemble des valeurs prises par \mathcal{D} , la variable aléatoire dont les ensembles S sont des instances.

$$\begin{aligned} Err(h, x) &= \mathbb{E}_{\mathcal{D}} [h(x) - f(x)]^2 \\ &= \underbrace{\mathbb{E}_{\mathcal{D}} [h(x) - \mathbb{E}_{\mathcal{D}} h(x)]^2}_{\text{Variance}} + \underbrace{\mathbb{E}_{\mathcal{D}} [\mathbb{E}_{\mathcal{D}} h(x) - f(x)]^2}_{\mathbb{E}[\text{Biais}^2]} \end{aligned} \quad (3.21)$$

Le terme de gauche, exprimant les variations de $h(x)$ autour de son espérance mathématique, représente l'erreur d'estimation. Le terme de droite correspond à l'erreur d'approximation (biais). Les modèles les plus simples approximent généralement mal la vraie fonction f , mais présentent une variance faible. À l'inverse, les modèles complexes approximent mieux f en moyenne, en règle générale, mais au prix d'une variance plus forte.

Il y a donc un compromis biais-variance à trouver dans le choix d'une hypothèse $h \in \mathcal{H}$, de façon à approximer correctement f sans faire de sur-apprentissage, afin de pouvoir faire des prévisions correctes sur de nouvelles instances.

3.3.6 Le fléau de la dimension

Pour approximer correctement une fonction inconnue $f : \mathcal{X} \rightarrow \mathcal{Y}$ (ou une distribution inconnue $\mathbb{P}_{X,Y}$) à partir d'exemples $S = \{(x_n, y_n) \mid 1 \leq n \leq N\}$, il est crucial que l'espace des entrées \mathcal{X} soit correctement échantillonné par les instances $\{x_1, \dots, x_N\}$. Supposons, à titre d'illustration, qu'on utilise une méthode des k plus proches voisins pour prédire la sortie y correspondant à une entrée x quelconque, en faisant une somme pondérée des réponses y_{i_1}, \dots, y_{i_k} des k entrées x_{i_1}, \dots, x_{i_k} les plus proches de x . Il est clair qu'en règle générale la prévision sera d'autant moins bonne que ces voisins sont éloignés de x .

Dans les exemples que nous avons vus jusqu'à présent, \mathcal{X} était de dimension 1, ce qui est rarement le cas dans des problèmes réels, où le nombre d'attributs caractéristiques des phénomènes observés est souvent bien plus important. Chaque entrée x_n est donc un vecteur $x_n = (x_{n1}, \dots, x_{nd}, \dots, x_{nD})^T$ de dimension D , où D peut être grand si l'on a beaucoup de variables explicatives.

Quand la dimension D devient grande, il y a une difficulté intrinsèque à échantillonner correctement l'espace \mathcal{X} . Supposons qu'on s'intéresse à un hypercube de côté de longueur 10 dans l'espace \mathcal{X} et qu'on veuille l'échantillonner de manière régulière, en le divisant en hypercubes de côté 1 et en prenant un point dans chaque petit hypercube. Le nombre de points est alors de 10^D et croît donc exponentiellement avec la dimension de l'espace des entrées.

C'est un handicap majeur pour l'application de la méthode des k voisins en dimension élevée : telle quelle, la méthode fonctionnera généralement de moins en moins bien en ajoutant des variables explicatives si on ne collecte pas un nombre d'exemples croissant exponentiellement avec la dimension D .

Sachant que l'information utile n'est généralement pas uniformément répartie dans l'espace \mathcal{X} , certaines méthodes d'apprentissage s'appuient sur des "dictionnaires" de fonctions de base, qu'elles combinent pour explorer des sous-espaces intéressants de l'espace des entrées. C'est le cas par exemple des réseaux de neurones où chaque unité cachée applique une fonction non linéaire $\varphi(z) = 1/(1 + e^{-z})$ (ou \tanh) à la somme pondérée de ses entrées. Malgré tout, le fait d'augmenter la dimension D des entrées augmentera mécaniquement le nombre de paramètres à ajuster, et donc le risque de sur-apprentissage.

D'autres phénomènes, parfois contre-intuitifs, compliquent la tâche des méthodes d'apprentissage dans des espaces en grande dimension (voir [24] pp. 36-37 et [86] pp. 22-24). Par exemple, l'essentiel du volume d'une boule de rayon 1 en dimension D est contenu dans une faible épaisseur proche de la surface de la boule.

Il est donc essentiel, dans un problème d'apprentissage, de ne pas multiplier inutilement les variables explicatives. Si le problème comprend malgré tout un grand nombre de variables, il peut être utile, voire indispensable, de réduire la dimensionalité du problème en sélectionnant les combinaisons de variables les plus utiles, ou en extrayant de nouvelles variables, en nombre inférieur. La sélection et l'extraction de variables explicatives sont présentées dans la section 3.6.

Avant cela, nous allons présenter quelques principes d'inférences (section 3.4), ainsi que le cadre méthodologique de la sélection de modèle (section 3.5), qui cherche à répondre au problème de la recherche d'un bon compromis biais-variance dans le choix du modèle.

3.4 Principes d'inférence

Dans cette section, nous allons essayer de répondre à cette question simple : comment faire apprendre à une machine ? La réponse à cette question s'appuie essentiellement sur le choix et l'application d'un principe d'inférence, issu de la théorie statistique ou de la théorie de l'information.

Nous allons illustrer sur quelques exemples comment cette application d'un principe d'inférence se ramène à un problème de minimisation, où l'on cherche à minimiser une espérance de perte.

Pour fixer les idées, nous allons supposer que les hypothèses $h \in \mathcal{H}$ sont paramétrées par un vecteur w de "poids"⁶ que l'on cherche à ajuster, afin d'obtenir l'hypothèse h_w la meilleure possible. La notion de "meilleure hypothèse" dépend bien sûr du principe d'inférence choisi.

3.4.1 Choix de représentation

Les choix de représentation peuvent parfois être plus ou moins naturels selon le problème d'apprentissage que l'on traite. Le concept qu'on cherche à apprendre à partir des données pourra dans certains cas être une formule logique si on travaille sur des variables booléennes, ou encore une fonction f si on travaille sur des variables numériques, ou bien des probabilités, ou des densités de probabilité.

6. Typiquement, ces poids seront les pondérations d'une combinaison linéaire des variables explicatives dans le cas d'un modèle linéaire, ou bien les poids affectés aux connexions d'un réseau de neurones, par exemple.

Représentation fonctionnelle (approche directe)

Dans le cas d'une représentation fonctionnelle du phénomène ayant produit les données observées $(x_1, y_1), \dots, (x_N, y_N)$, on cherche directement une fonction $h_w \in \mathcal{H}$ qui approxime au mieux la "vraie fonction" f . Cette approximation h_w doit être telle que les sorties calculées $\hat{y}_n = h_w(x_n)$ soient aussi proches que possible des réponses observées y_n , cette notion de proximité étant quantifiée par la fonction de perte $\ell(y, h_w(x))$.

Représentation par une loi conditionnelle (approche "discriminative")

D'autres choix de représentation sont possibles. Typiquement, dans le cas d'une régression avec une réponse $y \in \mathbb{R}$, on peut faire l'hypothèse d'un modèle statistique où la variable réponse y est une réalisation d'une variable aléatoire $Y = h_w(x) + \epsilon$, avec ϵ un bruit aléatoire pour lequel on peut faire une hypothèse de distribution. A titre d'exemple, supposons que ϵ soit une variable aléatoire dont la densité suit une loi gaussienne $\mathcal{N}(0, \sigma^2)$ de moyenne nulle et de variance σ^2 constante (hypothèse d'homoscédasticité) sur l'ensemble des valeurs de x :

$$p(\epsilon) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2} \frac{\epsilon^2}{\sigma^2}\right\} \quad (3.22)$$

Dans ce cas, la probabilité d'observer la réponse y , sachant l'entrée x et pour un vecteur de poids w , est égale à la probabilité d'observer un bruit $\epsilon = y - h_w(x)$ dans les mêmes conditions. On a donc également une loi gaussienne, ici $\mathcal{N}(h_w(x), \sigma^2)$ pour Y sachant x et w :

$$p(y|x, w) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2} (y - h_w(x))^2\right\} \quad (3.23)$$

Représentation par une loi jointe (approche "générative")

Toujours en restant dans une représentation probabiliste, on peut considérer également le vecteur d'entrée x comme une réalisation d'une variable aléatoire X , et s'intéresser à la probabilité jointe d'occurrence de x et y , de densité $p(x, y)$, avec pour objectif de l'approximer par la densité $p(x, y|w)$. Cette densité de probabilité s'exprime comme suit, en supposant que la variable aléatoire X dont les instances x sont des réalisations admette une densité $p(x)$:

$$p(x, y|w) = p(y|x, w)p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} p(x) \exp\left\{-\frac{1}{2\sigma^2} (y - h_w(x))^2\right\} \quad (3.24)$$

L'avantage de cette dernière approche est qu'elle permet de reconstruire des données manquantes en utilisant la distribution jointe $p(x, y|w)$. L'inconvénient est qu'il faut aussi approximer $p(x)$ à partir des données observées.

Les exemples que nous venons de donner pour la régression, avec des variables aléatoires continues, se transposent directement au cas de problèmes de classification où la variable réponse y encode une catégorie, avec par exemple $y \in \{-1, 1\}$ pour un problème à deux classes. Il suffit de remplacer les densités p par des lois de probabilités \mathbb{P} , pour les variables discrètes.

Dans [24], Bishop qualifie ces trois approches de "directe", quand on approxime directement une fonction $y = f(x)$ par une autre fonction $h(x)$, de "discriminative" lorsqu'on cherche à

approximer une densité conditionnelle $p(y|x)$ (pour les variables continues) ou une distribution de probabilité $\mathbb{P}(y|x)$ (pour des variables aléatoire discrètes), et de “générative” quand on cherche à approximer une densité jointe $p(x, y)$ ou une probabilité jointe $\mathbb{P}(x, y)$.

Notons toutefois que, dans l'apprentissage supervisé au sens large, c'est-à-dire non restreint à la reconnaissance de formes (*pattern recognition*), les choix de représentation ne se limitent pas aux seules fonctions ou distributions de probabilités. Selon les problèmes d'apprentissage traités, le concept à apprendre pourra être une expression booléenne, un programme logique, une grammaire, un processus markovien, etc (voir [40]). Dans le cadre du présent mémoire, nous nous limiterons aux choix de représentation que nous venons de présenter : fonction, ou distribution conditionnelle ou jointe.

3.4.2 Principes d'inférence et choix de la fonction de perte

Dans l'approche directe, on approxime une fonction f inconnue par une hypothèse $h \in \mathcal{H}$, en minimisant une perte $\ell(y, h(x))$ dont l'expression peut être choisie arbitrairement.

Dans le cas d'un choix de représentation utilisant les probabilités (conditionnelles, ou jointes), l'expression de la fonction de perte peut découler directement des hypothèses faites sur les distributions de probabilités et du choix d'un principe d'inférence : maximum de vraisemblance, maximum *a posteriori*, inférence bayésienne pure, principe de compression maximale, etc.

A titre d'illustration, voyons comment s'exprime la fonction de perte dans le cas d'un problème de régression, en reprenant l'exemple précédent d'une distribution gaussienne du bruit, de variance constante, et en prenant pour commencer le maximum de vraisemblance comme principe d'inférence. Puis nous verrons le maximum *a posteriori*, avant d'évoquer de manière très succincte l'inférence bayésienne pure, et ce en quoi elle diffère du principe de maximum *a posteriori*.

Maximum de vraisemblance pour un problème de régression

La vraisemblance du modèle h_w est définie, à une constante multiplicative près, comme étant la probabilité (ou la densité de probabilité pour les variables continues) d'observer les données, pour un modèle h_w fixé. Si on adopte l'approche “discriminative” où l'on s'intéresse à la densité conditionnelle, la vraisemblance est donc la probabilité d'observer les réponses y_1, \dots, y_N , connaissant les entrées x_1, \dots, x_N , et pour une valeur fixée du vecteur de poids w .

En supposant un bruit gaussien de variance constante, la vraisemblance s'exprime comme suit⁷, en faisant l'hypothèse que les N observations sont indépendantes :

$$p(y_1, \dots, y_N | x_1, \dots, x_N, w) = \prod_{n=1}^N p(y_n | x_n, w) = \prod_{n=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2\sigma^2} (y_n - h_w(x_n))^2 \right\} \quad (3.25)$$

En passant au logarithme et en prenant l'opposé, on voit que le maximum de la log-

7. Notons, dans l'équation 3.25, que certains auteurs préfèrent séparer par un point-virgule le vecteur w des autres variables, pour bien marquer que les paramètres w sont fixés, et les différencier des variables aléatoires.

vraisemblance est obtenu en minimisant l'expression suivante :

$$\frac{N}{2} \ln(2\pi\sigma^2) + \sum_{n=1}^N \frac{1}{2\sigma^2} (y_n - h_w(x_n))^2 \quad (3.26)$$

L'application du principe de maximum de vraisemblance se ramène donc, sur l'exemple que nous avons pris, à un problème de minimisation. Celui-ci s'exprime formellement par l'équation (3.27) suivante, où l'on voit apparaître une fonction de perte quadratique $\ell(y, h_w(x)) = (y - h_w(x))^2$:

$$\min_{w, \sigma^2} \left[\frac{1}{\sigma^2} \sum_{n=1}^N \underbrace{(y_n - h_w(x_n))^2}_{\ell(y_n, h_w(x_n))} + N \ln(2\pi\sigma^2) \right] \quad (3.27)$$

Notons $\mathcal{E}(w)$ la fonction d'erreur suivante, dépendant uniquement des poids w :

$$\mathcal{E}(w) = \sum_{n=1}^N \ell(y_n, h_w(x_n)) = \sum_{n=1}^N (y_n - h_w(x_n))^2 \quad (3.28)$$

La recherche du vecteur w minimisant l'expression (3.26), pour une valeur quelconque (positive) de σ^2 , se ramène à la minimisation de l'erreur quadratique $\mathcal{E}(w)$, pour notre exemple de régression avec une hypothèse de bruit gaussien. En pratique, la phase d'apprentissage consiste donc ici à trouver un vecteur de poids qui minimise cette erreur, par l'application d'une méthode d'optimisation adéquate.

$$w^* = \hat{w}^{\text{ML}} = \arg \min_w \sum_{n=1}^N (y_n - h_w(x_n))^2 \quad (3.29)$$

Le principe du maximum de vraisemblance, appliqué à notre problème de régression se ramène donc avec les hypothèses choisies à la minimisation du risque empirique $\mathcal{R}_{emp}(h, S) = \frac{1}{N} \sum_{(x_n, y_n) \in S} \ell(y_n, h(x_n))$ de l'équation (3.14) (voir section 3.3.2) avec une fonction de perte ℓ quadratique.

Notons par ailleurs que l'annulation de la dérivée par rapport à σ^2 de l'expression (3.26), prise au point \hat{w}^{ML} , permet facilement de déduire la variance du bruit au maximum de vraisemblance. Pour notre exemple de régression où la réponse est une simple variable réelle $y \in \mathbb{R}$, l'expression de la variance σ^2 est donnée par l'équation (3.30), où $\mathcal{E}(\hat{w}^{\text{ML}})$ est le minimum de l'erreur.

$$\sigma_{\text{ML}}^2 = \frac{1}{N} \mathcal{E}(\hat{w}^{\text{ML}}) \quad (3.30)$$

Rappelons que les équations (3.29) et (3.30) sont obtenues en supposant que la réponse $y \in \mathbb{R}$ est le résultat de l'addition de $h_w(x)$ et d'un bruit ϵ qu'on a ici supposé gaussien, de variance constante en tout point (x, y) . Ces dernières hypothèses ne sont bien entendu pas toujours réalisées pour des données issues de problèmes réels, ce qui n'empêche pas en pratique d'utiliser

une fonction de perte quadratique. D'autres fonctions de perte peuvent cependant s'avérer plus judicieuses, en particulier pour diminuer l'influence des données aberrantes dans le cadre de la régression robuste. Nous avons simplement ici rappelé que le choix d'une fonction de perte quadratique découlait d'un principe fondamental, celui du maximum de vraisemblance, avec les hypothèses choisies.

Maximum *a posteriori* pour un problème de régression

Supposons maintenant qu'on choisisse le principe du maximum *a posteriori* comme principe d'inférence. Ce principe revient à choisir l'hypothèse $h \in \mathcal{H}$ la plus probable (et non pas la plus vraisemblable, au sens que nous avons vu précédemment), au vu des données observées. L'hypothèse h est donc vue ici comme une réalisation d'une variable aléatoire, et non plus comme un modèle fixé, comme dans le maximum de vraisemblance. Pour un modèle paramétré h_w , on choisira donc le vecteur w pour lequel $p(w|S)$ est maximum, $S = \{(x_n, y_n) \mid 1 \leq n \leq N\}$ étant l'ensemble des données observées utilisées pour l'apprentissage.

$$\hat{w}^{\text{MAP}} = \arg \max_w p(w|S) = \arg \max_w p(w \mid x_1, \dots, x_N, y_1, \dots, y_N) \quad (3.31)$$

En appliquant la règle de Bayes, on exprime $p(w|S)$ de la façon suivante :

$$p(w|S) = \frac{p(S|w)p(w)}{p(S)} \quad (3.32)$$

Pour un ensemble d'apprentissage S donné, $p(S)$ restera constant quelle que soit la valeur de w considérée. On peut donc réécrire l'équation (3.31) donnant le maximum de vraisemblance de la façon suivante :

$$\hat{w}^{\text{MAP}} = \arg \max_w p(w|S) = \arg \max_w p(S|w)p(w) \quad (3.33)$$

En passant au logarithme, on obtient l'expression suivante :

$$\hat{w}^{\text{MAP}} = \arg \max_w [\ln p(S|w) + \ln p(w)] \quad (3.34)$$

Dans le cas où l'on s'intéresse à la densité de probabilité conditionnelle $p(y|x)$ plutôt qu'à la densité jointe $p(x, y)$, il est utile d'exprimer la densité de probabilité $p(S|w)$ de la façon suivante, où la dernière ligne tient compte du fait que le tirage des x_n est indépendant du choix de w :

$$\begin{aligned} p(S|w) &= p(x_1, \dots, x_N, y_1, \dots, y_N \mid w) \\ &= p(y_1, \dots, y_N \mid x_1, \dots, x_N, w)p(x_1, \dots, x_N \mid w) \\ &= p(y_1, \dots, y_N \mid x_1, \dots, x_N, w)p(x_1, \dots, x_N) \end{aligned} \quad (3.35)$$

L'équation (3.34) peut être reformulée en fonction de $p(y_1, \dots, y_N \mid x_1, \dots, x_N, w)$, la densité conditionnelle, en tenant compte du fait que $p(x_1, \dots, x_N)$ reste constant :

$$\begin{aligned} \hat{w}^{\text{MAP}} &= \arg \max_w [\ln p(y_1, \dots, y_N \mid x_1, \dots, x_N, w) + \ln p(x_1, \dots, x_N) + \ln p(w)] \\ &= \arg \max_w [\ln p(y_1, \dots, y_N \mid x_1, \dots, x_N, w) + \ln p(w)] \end{aligned} \quad (3.36)$$

Reprenons maintenant notre exemple de régression avec $y \in \mathbb{R}$ une variable réponse réelle, en supposant que y est une réalisation de $Y = h_w(x) + \epsilon$ où ϵ est un bruit aléatoire gaussien de variance constante σ^2 . La densité conditionnelle $p(y_1, \dots, y_N | x_1, \dots, x_N, w)$ est alors celle de l'équation (3.25) que nous avons calculé pour le maximum de vraisemblance. Le principe de maximum *a posteriori* donné par l'équation (3.36) se ramène alors au problème de minimisation suivant :

$$\min_{w, \sigma^2} \left[\frac{1}{\sigma^2} \sum_{n=1}^N \underbrace{(y_n - h_w(x_n))^2}_{\ell(y_n, h_w(x_n))} + N \ln(2\pi\sigma^2) - 2 \ln p(w) \right] \quad (3.37)$$

Faisons l'hypothèse d'une distribution *a priori* normale pour le vecteur aléatoire w de dimension M , telle que donnée par l'équation (3.38), où I_M est la matrice identité et β^2 la variance constante selon chaque composante du vecteur w :

$$\begin{aligned} p(w) &= \mathcal{N}(0_M, \beta^2 I_M) = (2\pi)^{-\frac{M}{2}} |\beta^2 I|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} w^T (\beta^2 I_M)^{-1} w \right\} \\ &= (2\pi\beta^2)^{-\frac{M}{2}} \exp \left\{ -\frac{1}{2\beta^2} w^T w \right\} \end{aligned} \quad (3.38)$$

Le problème de minimisation (3.37) revient alors à minimiser l'expression suivante :

$$\frac{1}{\sigma^2} \sum_{n=1}^N (y_n - h_w(x_n))^2 + \frac{1}{\beta^2} w^T w + N \ln(2\pi\sigma^2) + M \ln(2\pi\beta^2) \quad (3.39)$$

Pour des valeurs données de σ^2 et β^2 , et avec les hypothèses que nous avons faites sur les distributions, le principe du maximum *a posteriori* revient pour notre exemple donc à minimiser l'erreur donnée par l'expression (3.40) suivante, en notant $\lambda = \frac{\sigma^2}{\beta^2}$ le rapport des variances, et où l'on retrouve l'expression d'une perte quadratique à laquelle on a ajouté un terme de régularisation :

$$\mathcal{E}'(w, \lambda) = \sum_{n=1}^N (y_n - h_w(x_n))^2 + \lambda w^T w \quad (3.40)$$

L'expression (3.40) est à rapprocher des expressions (3.18) (régression linéaire *ridge*) et (3.20) (*weight decay* des réseaux de neurones) de la section 3.3.3 sur la régularisation. D'autres choix pour la distribution *a priori* $p(w)$ peuvent mener à d'autres expressions de la pénalité de régularisation.

Dans l'exemple que nous nous sommes donnés, on peut trouver la valeur de la variance σ^2 , pour une valeur fixée de l'hyperparamètre de régularisation λ , en annulant la dérivée de l'expression (3.39) par rapport à σ^2 (en tenant compte du fait que $\beta^2 = \frac{\sigma^2}{\lambda}$). Si $w^* = \hat{w}^{\text{MAP}} = \arg \min_w \mathcal{E}'(w, \lambda)$ est la valeur du vecteur de poids au maximum *a posteriori*, on trouve facilement l'expression (3.41), où N est le nombre d'exemples utilisés pour l'apprentissage et M est le nombre de paramètres à ajuster du modèle h_w (*i.e.* la dimension de w) :

$$\sigma_{\text{MAP}}^2 = \frac{1}{N + M} \mathcal{E}'(\hat{w}^{\text{MAP}}, \lambda) \quad (3.41)$$

Dans le principe de maximum *a posteriori* que nous venons de présenter, la phase d'apprentissage consiste donc à trouver la valeur du vecteur w (dans le cas d'un modèle paramétrique) qui apparaît la plus probable au vu des exemples. La prédiction des réponses y à de nouvelles entrées x sont alors réalisées en utilisant la densité de probabilité conditionnelle $p(y|x, \hat{w}^{\text{MAP}})$

Avec les hypothèses prises dans notre exemple (bruit gaussien, distribution *a priori* gaussienne des poids), nous introduisons deux paramètres supplémentaires σ^2 et β^2 . Le modèle de notre exemple n'est donc pas paramétré uniquement par w , mais par $\theta = (w, \sigma^2, \beta^2)$, ou encore par $\theta = (w, \sigma^2, \lambda)$ avec $\lambda = \frac{\sigma^2}{\beta^2}$. L'hyperparamètre λ peut être soit fixé arbitrairement, soit sélectionné à partir des données, par exemple par validation croisée (voir section 3.5.1). Pour une valeur fixée de λ , la variance σ_{MAP}^2 est donnée par l'équation (3.41), et on en déduit la variance β_{MAP}^2 par la définition de λ . Le modèle prédictif dans notre exemple est alors la densité $p(y|x, \theta^{\text{MAP}}) = p(y|x, \hat{w}^{\text{MAP}}, \sigma_{\text{MAP}}^2, \beta_{\text{MAP}}^2)$.

Inférence bayésienne

Dans le cas de l'inférence bayésienne pure, le modèle de prédiction n'est pas obtenu en considérant une valeur unique du vecteur de poids w ou de tout autre paramètre complémentaire. Le principe de l'inférence bayésienne consiste à faire une "moyenne" sur l'étendue des valeurs des différents paramètres, considérés comme des variables aléatoires munis de distributions *a priori* que l'on révisé au vu des exemples.

Si l'on s'intéresse par exemple à la distribution de probabilité conditionnelle $p(y|x)$ (approche "discriminative") pour prédire la valeur y de la réponse à l'entrée x , on pourra approximer cette distribution inconnue par la distribution suivante, où θ regroupe l'ensemble des paramètres du modèle (poids w , variance σ^2 d'un bruit gaussien, etc) :

$$p(y|x, S) = \int p(y|x, S, \theta) p(\theta|S) d\theta \quad (3.42)$$

La valeur du couple (x, y) ne dépend des exemples passés $S = \{(x_n, y_n) \mid 1 \leq n \leq N\}$ qu'à travers le paramètre θ et on peut donc écrire $p(y|x, S, \theta) = p(y|x, \theta)$. La densité *a posteriori* $p(\theta|S)$ est donnée par l'application de la règle de Bayes (comme pour l'équation (3.32) dans le cas du maximum *a posteriori*) :

$$p(\theta|S) = \frac{p(S|\theta)p(\theta)}{p(S)} \quad (3.43)$$

A la différence du maximum *a posteriori*, on ne peut pas s'abstraire du calcul ou de l'approximation de $p(S)$ dans cette équation, puisqu'on veut obtenir au final l'expression de la densité de probabilité. La quantité $p(S)$ s'obtient en intégrant sur toutes les valeurs possibles de θ , le calcul effectif pouvant reposer sur une approximation numérique :

$$p(S) = \int p(S|\theta)p(\theta) d\theta \quad (3.44)$$

Dans l'inférence bayésienne pure, la quantité apprise à partir des données lors de la phase d'apprentissage est donc la distribution *a posteriori* $p(\theta|S)$, qui est proportionnelle à $p(S|\theta)p(\theta)$ où $p(\theta)$ est la distribution *a priori* que l'on a choisi pour θ .

Le modèle prédictif est la densité $p(y|x, S)$ qu'on peut exprimer par l'équation (3.45) suivante :

$$p(y|x, S) = \frac{1}{p(S)} \int p(y|x, \theta) p(S|\theta) p(\theta) d\theta \quad (3.45)$$

En reprenant brièvement notre exemple, nous avons fait une hypothèse de bruit gaussien de variance σ^2 pour $p(y|x, w)$, et supposé une distribution également gaussienne, de variance β^2 , pour la distribution *a priori* $p(w)$. Dans cet exemple, l'intégration bayésienne doit donc se faire sur l'ensemble des paramètres w , σ^2 et β^2 :

$$p(y|x, S) = \int p(y|x, w, \sigma^2, \beta^2) p(w, \sigma^2, \beta^2 | S) dw d\sigma^2 d\beta^2 \quad (3.46)$$

Le calcul analytique de $p(w, \sigma^2, \beta^2 | S)$ n'est généralement pas possible. En pratique, le calcul effectif de cette quantité peut faire appel à diverses techniques : maximum de vraisemblance de type II, approximation de Laplace, calcul variationnel, méthodes d'échantillonnage.

3.4.3 Quel principe d'inférence choisir ?

Nous venons de présenter trois principes d'inférence, mais il en existe d'autres comme par exemple le principe de compression maximale d'information (*minimum description length*). Parmi ces différents principes d'inférence, lequel choisir ?

Sur le plan théorique, la différence entre ces deux principes de maximum de vraisemblance et de maximum *a posteriori* réside essentiellement dans le fait qu'on adopte un point de vue "fréquentiste" ou "bayésien" dans la sélection du meilleur modèle. Du point de vue "fréquentiste", le modèle est considéré comme fixe, non aléatoire, et on recherche le modèle (fixé) le plus vraisemblable (et non pas le plus probable) en cherchant les paramètres du modèle qui maximisent la probabilité d'observer les données S . En adoptant le point de vue "bayésien", le modèle (ou ici son vecteur de paramètres) est considéré comme une variable aléatoire. Le principe du maximum *a posteriori* revient à chercher le modèle le plus probable (et non pas le plus vraisemblable) au vu des données.

Dans le cas de l'inférence bayésienne pure, on ne cherche pas à sélectionner "le" meilleur modèle possible. Le modèle prédictif réalise en fait une moyenne sur l'ensemble des valeurs possibles des paramètres, cette moyenne étant pondérée par la probabilité *a posteriori* de choisir un vecteur de paramètres donné au vu des exemples observés. La phase d'apprentissage proprement dite consiste à estimer cette probabilité *a posteriori* en révisant, au vu des données observées, la distribution *a priori* dont on s'était initialement doté.

La question du choix du principe d'inférence est loin d'être tranchée dans la littérature. Par exemple, [24] développe systématiquement une approche bayésienne de l'apprentissage pour quasiment toutes les méthodes présentées, alors que [86] n'aborde cette approche que plus marginalement. Le choix d'un principe d'inférence ne repose pas que sur un choix "idéologique" entre approche fréquentiste ou bayésienne. D'un point de vue pratique, l'approche directe où l'on approxime simplement une fonction en minimisant une perte, et les approches cherchant seulement le maximum d'une distribution de probabilité sont indéniablement plus simples que celles reposant sur un calcul effectif d'une distribution de probabilité, jointe ou conditionnelle.

Un autre critère pouvant orienter le choix du praticien est celui de la sélection de modèle, qui nécessite certaines précautions dans le cas des approches directes, ou du maximum de vraisemblance ou *a posteriori*. En effet, nous avons vu sur notre exemple que les principes de maximum de vraisemblance et de maximum *a posteriori* se ramenaient à la minimisation d'un risque empirique, régularisé dans le cas du maximum *a posteriori*.

Nous avons vu dans la section 3.3 que la minimisation du risque empirique est sujette au risque de sur-apprentissage. Si l'espace d'hypothèses \mathcal{H} contient des hypothèses de complexités variées, comme par exemples des modèles linéaires avec différents jeux de variables explicatives en entrée, ou des réseaux de neurones avec différents nombres d'unités cachées, minimiser le risque empirique sur l'ensemble S sans précautions particulières conduit systématiquement à sélectionner l'hypothèse la plus complexe. Le modèle de prédiction ainsi sélectionné généralisera probablement mal sur de nouvelles entrées.

3.5 Cadre méthodologique de la sélection de modèle

Dans la section 3.3.4, nous avons vu l'influence de la complexité du modèle sur le sur-apprentissage, et la nécessité de trouver un bon compromis biais-variance entre des modèles de complexités diverses. Typiquement, on cherchera un tel compromis en considérant une séquence de modèles ordonnés par complexité croissante, cette dernière pouvant être mesurée selon le cas par le nombre de paramètres ajustables, ou bien par la dimension de Vapnik-Chervonenkis, ou encore par la valeur de l'hyperparamètre λ quand on applique une technique de régularisation.

Nous avons vu sur la Figure 3.6 qu'on ne peut pas se baser sur la seule erreur $\mathcal{E}(h)$, calculée sur l'ensemble d'apprentissage S , pour sélectionner h . En effet, cela reviendrait à choisir systématiquement la fonction ayant l'erreur la plus faible, et donc le modèle le plus complexe. Si l'on applique une technique de régularisation en minimisant $\mathcal{E}'(h; \lambda) = \mathcal{E}(h) + \lambda J(h)$, nous avons le même problème pour le choix de λ : ce sera le paramètre correspondant à l'erreur d'apprentissage la plus faible et au modèle le plus complexe qui sera sélectionné.

Pour choisir une hypothèse h dans \mathcal{H} , avec ou sans paramètre de régularisation, il nous faut donc soit borner l'erreur faite par h quand on généralise à de nouvelles données, comme dans le cas de la minimisation du risque structurel de Vapnik ([168]), soit l'estimer empiriquement ou avec un critère théorique (Akaike, Schwartz, etc). L'estimation empirique de l'erreur de généralisation peut se faire en utilisant des données séparées des données d'apprentissage (validation simple ou croisée à K plis), ou en ré-échantillonnant S plusieurs fois par des tirages avec remise (*bootstrap*).

Dans cette section, les techniques de bootstrap ou de minimisation du risque structurel ne sont pas présentées. Le lecteur peut se reporter à [86, 24, 40] pour une présentation de ces méthodes. Seules sont décrites les méthodes de sélection que nous avons employées dans les travaux présentés aux chapitres 4 et 5, c'est-à-dire l'estimation empirique de l'erreur par validation croisée, ou l'estimation par des critères théoriques d'information.

De quel modèle parle-t'on ?

Avant de poursuivre, soulignons l'extrême ambiguïté du mot "modèle" dans la littérature sur l'apprentissage ou les statistiques. Selon le contexte, le terme "modèle" peut désigner une fonction h spécifique prise dans \mathcal{H} ou bien un sous-ensemble de l'espace d'hypothèse \mathcal{H} . A titre d'exemple, on peut désigner comme "modèle de prévision" un réseau de neurones déjà

appris sur un ensemble d'apprentissage et qu'on veut utiliser pour généraliser à de nouvelles entrées. C'est alors d'une fonction h spécifique que l'on parle.

Dans le cadre de la sélection de modèle, c'est par contre la deuxième interprétation qui prévaut. Le terme "modèle" désignera par exemple un réseau de neurones à c unités sur une couche cachée, dont les poids n'ont pas encore été ajustés, et l'objectif de la sélection de modèle sera de déterminer la bonne valeur de c dans le compromis biais-variance à trouver. Dans cet exemple, un "modèle" est donc un sous-espace $\mathcal{H}_c \subset \mathcal{H}$.

Dans ce qui suit, quand on parle de performance en généralisation d'un modèle, c'est donc la performance d'une classe $\mathcal{H}_c \subset \mathcal{H}$ de fonctions h que l'on cherche à évaluer. Le principe de la sélection de modèle consiste à trouver \mathcal{H}_{c^*} dans une séquence $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_C$ qui corresponde au meilleur compromis biais-variance.

3.5.1 Estimation empirique de l'erreur de généralisation

Une approche classique parmi les plus employées pour évaluer la performance en généralisation des modèles consiste à évaluer empiriquement l'erreur de prédiction sur des données indépendantes n'ayant pas servi à l'apprentissage.

La validation simple (*hold-out validation*)

La forme la plus basique de cette approche consiste à séparer l'ensemble de données S en deux, une part S_T servant à l'ajustement du modèle et l'autre S_V pour évaluer l'erreur de généralisation. La Figure 3.8 illustre ce principe de sélection.

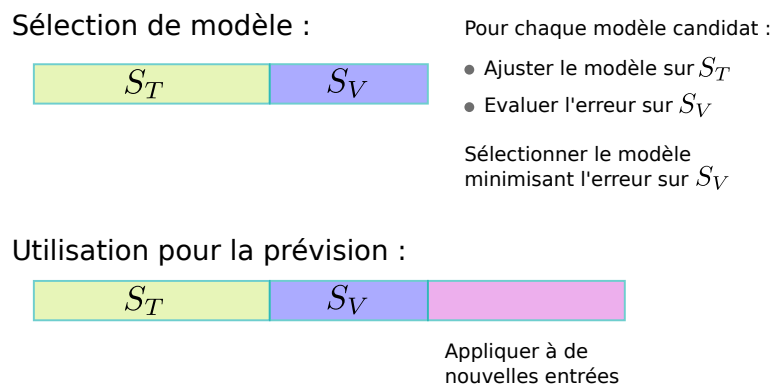


FIGURE 3.8: Principe de la validation simple sur données séparées.

Le "modèle" sélectionné peut ensuite être éventuellement réajusté sur $S = S_T \cup S_V$ avant d'être utilisé comme "modèle" de prévision (fonction h_S^*) sur de nouvelles entrées. Notons que si on veut évaluer la performance de h_S^* en généralisation, il faut disposer d'un troisième ensemble de données, différent de S_T et S_V .

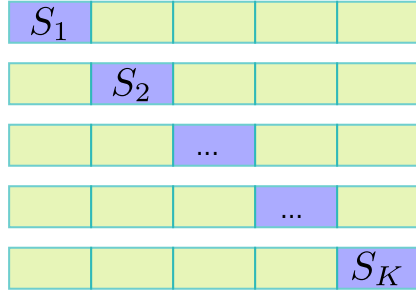
Cette validation simple (*hold-out validation*) est donc relativement coûteuse en données et n'est pas la plus judicieuse lorsqu'on dispose d'un nombre restreint d'exemples.

La validation croisée à K plis

La validation croisée à K plis, dont le principe est exposé sur la Figure 3.9, permet de réutiliser plusieurs fois le même jeu de données, de façon parcimonieuse, pour fournir une

estimation de l'erreur de prédiction sur des données non vues par l'algorithme d'apprentissage. L'ensemble S est divisé en K parties de taille sensiblement égale, de façon que chaque ensemble S_k (pour $1 \leq k \leq K$) contiennent des instances indépendantes de celles des autres sous-ensembles. Notons S_{-k} la réunion des $k - 1$ sous-ensembles autres que S_k : $S_{-k} = S \setminus S_k = \bigcup_{i \neq k} S_i$.

Sélection de modèle :



Pour chaque modèle candidat :

Pour chaque pli k :

- Ajuster le modèle sur $S_{-k} = \bigcup_{i \neq k} S_i$
- Evaluer l'erreur sur S_k

Calculer l'erreur cumulée sur les K plis

Sélectionner le modèle minimisant l'erreur sur les K ensembles de validation S_k

Utilisation pour la prévision :

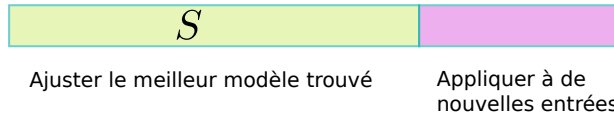


FIGURE 3.9: Principe de la validation croisée à K plis.

Soit \mathcal{A} un algorithme d'apprentissage permettant de choisir une hypothèse $h_T^* = \mathcal{A}(T)$ dans tout ou partie de l'espace d'hypothèses \mathcal{H} , à partir d'un ensemble d'exemples d'apprentissage T . Dans la suite, nous indiquerons $\mathcal{A}_{\mathcal{H}_c}$ cet algorithme d'apprentissage lorsqu'il effectue cette sélection dans le sous-espace \mathcal{H}_c .

La validation croisée à K plis consiste, pour chaque "modèle" candidat \mathcal{H}_c et chaque pli k , à apprendre un "modèle de prévision" $h_c^* = \mathcal{A}_{\mathcal{H}_c}(S_{-k})$ sur l'ensemble S_{-k} , puis à évaluer son erreur $Err(h_c^*, S_k)$ sur l'ensemble S_k n'ayant pas été utilisé pour l'apprentissage. L'ensemble des erreurs $Err(\mathcal{A}_{\mathcal{H}_c}(S_{-k}), S_k)$ faites par les K modèles ajustés est alors cumulée, pour évaluer l'erreur de généralisation du modèle \mathcal{H}_c , donnée par l'équation (3.47). Remarquons au passage que cette procédure permet aussi d'estimer empiriquement la variance de l'erreur de prévision sur les K sous-ensembles de données.

$$Err_{CV_K}(\mathcal{H}_c) = \frac{1}{|S|} \sum_{k=1}^K Err(\mathcal{A}_{\mathcal{H}_c}(S_{-k}), S_k) \quad (3.47)$$

On choisit le modèle \mathcal{H}_{c^*} dont l'erreur sur l'ensemble des S_k est minimale. Le modèle sélectionné est ensuite ajusté sur l'ensemble des données de S , pour pouvoir servir comme modèle de prévision $h_S^* = \mathcal{A}_{\mathcal{H}_{c^*}}(S)$ sur de nouvelles entrées.

Notons que l'ensemble de la procédure de validation croisée à K plis que nous venons de décrire est elle-même un algorithme d'apprentissage, puisqu'elle permet de sélectionner une hypothèse h_S^* dans l'espace \mathcal{H} à partir d'un ensemble de données S . Avec les notations

adoptées dans cette section, on peut le formaliser sous la forme de l'équation (3.48) suivante :

$$\begin{aligned} CV_K(\mathcal{A}, \mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_C) : \quad & \bigcup_{N=0}^{\infty} (\mathcal{X} \times \mathcal{Y})^N \quad \rightarrow \quad \mathcal{H} = \bigcup_{c=1}^C \mathcal{H}_c \\ S = \{(x_n, y_n)_{1 \leq n \leq N}\} \quad & \rightarrow \quad h_S^* \end{aligned} \quad (3.48)$$

Les validations croisées imbriquées

L'algorithme d'apprentissage \mathcal{A} dans l'équation (3.48) peut lui-même faire appel à une validation croisée à K' plis, et on parle alors de double validation croisée imbriquée. C'est typiquement le cas lorsqu'on imbrique un processus de sélection des variables d'entrée avec un processus de sélection d'un paramètre représentant la complexité des hypothèses, comme par exemple un hyperparamètre de régularisation (voir [86], Chap. 7).

L'algorithme \mathcal{A} imbriqué dans CV_K peut aussi être une validation simple sur données séparées, plutôt qu'une validation croisée, ou encore il peut s'agir d'un algorithme itératif dont le critère d'arrêt peut inclure une validation sur données séparées, comme dans certains réseaux de neurones, par exemple.

Choix du paramètre K de la validation croisée

La question se pose de savoir quelle valeur de K faut-il choisir pour la validation croisée. La plus grande valeur possible est $K = N$, où N est le nombre d'exemples dans l'ensemble initial S . Dans ce cas, l'erreur de prédiction est évaluée sur un seul exemple, à chacun des N plis. Ce type de validation (*leave-one-out*) donne une variance de l'erreur plus importante que pour des valeurs plus faibles de K .

Pour K plus petit, la validation croisée à K plis est susceptible de surestimer l'erreur de généralisation, dans le cas où l'algorithme \mathcal{A} produit des modèles de prévision significativement moins bons avec $N - \frac{N}{K}$ exemples qu'avec N exemples. Un bon compromis semble être de choisir K entre 5 et 10 (voir [101]).

Les techniques d'évaluation empirique de l'erreur de généralisation que nous venons de voir sont relativement coûteuses en temps de calcul, et peuvent nécessiter un nombre d'exemples assez important, en particulier dans le cas de validations croisées imbriquées. Une alternative à cette estimation empirique est d'utiliser un critère théorique d'information.

3.5.2 Les critères d'information

Plutôt que de chercher à estimer l'erreur de généralisation sur des données indépendantes, séparées des données d'apprentissage, comme dans le cas de la validation croisée présentée précédemment, on peut utiliser un critère théorique d'information, calculé à partir de l'ensemble des données S également utilisées pour l'apprentissage.

Dans cette section, on suppose que les hypothèses $h = h_\theta$ sont paramétrées par un vecteur θ . On considère une collection de sous-espaces $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_C$, chaque sous-espace \mathcal{H}_c regroupant les hypothèses h_{θ_c} dont le vecteur de paramètres θ_c est de même dimension k_c . L'objectif de la sélection de modèle est alors de choisir $c \in \{1, \dots, C\}$ et de trouver l'hypothèse $h_{\hat{\theta}_c} \in \mathcal{H}_c$ qui approxime au mieux f .

On fait par ailleurs ici un choix de représentation où f , le vrai phénomène ayant produit les données observées, est représentée par une densité de probabilité, ainsi que les hypothèses h_θ .

Choix de représentation

Dans ce qui suit, le phénomène f ayant produit les données observées est modélisé comme une densité de probabilité, que l'on notera p_f . La variable aléatoire dont les observations z_1, \dots, z_N sont des réalisations est notée Z . Plusieurs choix de représentation sont possibles, pour la variable Z :

- a) $Z = (X, Y)$ si p_f est la densité jointe $p_{X,Y|f}$. Dans ce cas, chaque observation $z = (x, y)$ d'une entrée et de la réponse associée est donc un tirage aléatoire d'une variable $Z = (X, Y)$, selon la loi de densité jointe $p_f = p_{X,Y|f}$.
- b) $Z = Y$, lorsque les entrées x sont fixées, si p_f est la densité conditionnelle $p_{Y|X,f}$. Dans ce cas, pour une valeur donnée de x , la réponse observée $z = y$ est un tirage aléatoire de $Z = Y$, selon la loi conditionnelle $p_f = p_{Y|X,f}$.

Le choix de représentation de f par une densité jointe ou conditionnelle n'a pas d'incidence sur l'expression des critères d'information présentés dans la suite, et nous parlerons donc de la variable Z et de la densité p_f sans préciser ce choix de représentation, sauf si nécessaire.

On recherche une hypothèse $h \in \mathcal{H}$ qui approxime au mieux le phénomène f ayant produit les données observées. Cette hypothèse h est donc ici également représentée par une densité de probabilité p_h , qui pourra être soit jointe soit conditionnelle selon le choix de représentation fait pour p_f .

Forme générale d'un critère d'information

La forme générale d'un critère théorique d'information est donnée par l'équation (3.49), où le facteur 2 apparaît pour des raisons "historiques", et où $S = \{z_1, \dots, z_N\}$ est l'ensemble des données observées.

$$IC = -2 \ln \mathcal{L}(h_{\hat{\theta}_c}; S) + P_c = \sum_{n=1}^N \ln p_{h_{\hat{\theta}_c}}(z_n) + P_c \quad (3.49)$$

Dans l'expression (3.49), la quantité $\ln \mathcal{L}(h_{\hat{\theta}_c}; S)$ est le maximum de la fonction de log-vraisemblance empirique $\theta_c \rightarrow \sum_{n=1}^N \ln p_{h_{\theta_c}}(z_n)$ pour les θ_c pris dans le sous-espace de dimension k_c de l'espace des paramètres correspondant au sous-espace d'hypothèses \mathcal{H}_c . Ce maximum de vraisemblance empirique est obtenu pour $\theta_c = \hat{\theta}_c$, dont la valeur dépend de l'échantillon de données S : $\hat{\theta}_c = \hat{\theta}_c(S)$. La pénalité P_c ajoutée à $-2 \ln \mathcal{L}(h_{\hat{\theta}_c}; S)$ est déterminée de façon à corriger le biais introduit en approximant une mesure théorique de la divergence entre le modèle et la vraie distribution (typiquement, la divergence de Kullback-Leibler) par une estimation empirique obtenue à partir du seul ensemble S .

De nombreuses façons de calculer cette pénalité P_c ont été proposées, et nous nous contenterons ici de décrire deux des critères asymptotiques les plus employés, à savoir le critère AIC d'Akaike, et le critère BIC de Schwartz. Une description plus complète de ces critères et de leurs liens avec la divergence de Kullback-Leibler et la théorie de l'information peut être trouvée dans l'annexe B.

Le critère AIC

Le critère d'information proposé par Akaike en 1973 [6] cherche à minimiser la divergence de Kullback-Leibler (voir annexe B) en faisant deux hypothèses fortes :

- i le vrai phénomène f ayant produit les données observées fait partie de l'ensemble \mathcal{H} des hypothèses testées : $\exists \theta^* \in \Theta, \quad f = h_{\theta^*} \in \mathcal{H}$,
- ii pour chaque modèle candidat \mathcal{H}_c , l'estimation de la “meilleure” valeur possible $\hat{\theta}_c$ du paramètre θ_c est obtenue par maximum de vraisemblance.

Avec k_c le nombre de paramètres à ajuster (la dimension de θ_c) pour les hypothèses du modèle \mathcal{H}_c , et N le nombre d'exemples dans l'ensemble S , le critère AIC s'exprime comme suit :

$$AIC = -2 \ln \mathcal{L}(h_{\hat{\theta}_c}; S) + 2k_c = -2 \sum_{n=1}^N \ln p_{h_{\hat{\theta}_c}}(z_n) + 2k_c \quad (3.50)$$

Le critère BIC

Le critère BIC (Bayesian Information Criterion), introduit par Schwartz [146], sélectionne le modèle dont la probabilité *a posteriori*, au vu des données, est la plus grande. Il s'exprime comme suit, avec N le nombre d'exemples dans S , et k_c la dimension du vecteur de paramètres à ajuster :

$$BIC = -2 \ln \mathcal{L}(h_{\hat{\theta}_c}; S) + k_c \ln N = -2 \sum_{n=1}^N \ln p_{h_{\hat{\theta}_c}}(z_n) + k_c \ln N \quad (3.51)$$

Dans l'équation (B.34), le vecteur $\hat{\theta}_c$ correspond au *maximum a posteriori* $\hat{\theta}_c = \hat{\theta}_c^{\text{MAP}}$, mais comme le BIC est un critère asymptotique où l'on suppose N grand, on approxime par $\hat{\theta}_c^{\text{MAP}}$ par le maximum de vraisemblance $\hat{\theta}_c = \hat{\theta}_c^{\text{ML}}$.

3.5.3 Quel méthode de sélection choisir ?

Nous n'avons présenté dans cette section 3.5 qu'une petite partie des méthodes existantes pour la sélection de modèle, soit par estimation empirique de l'erreur, soit par critère d'information.

Les deux critères asymptotiques d'information, AIC et BIC, parmi les plus largement employés pour la sélection de modèles, sont tous deux d'autant plus valables que N est grand, mais de nombreux autres critères existent comme l'AIC_c qui corrige un biais supplémentaire lorsque le nombre d'exemples N n'est pas suffisamment grand devant la dimension k_c du vecteur de paramètres. Les critères AIC et BIC s'appuient tous deux sur des fondements théoriques solides (voir annexe B), mais avec des approches statistiques différentes (fréquentiste pour l'AIC, bayésienne pour le BIC), et il est parfois difficile de savoir lequel choisir en fonction du problème traité.

Pour N plus grand que 7, le critère BIC pénalise plus fortement les hypothèses complexes que le critère AIC, de la même façon que la validation croisée à 10 plis pénalise plus fortement que la validation croisée à N plis (*leave-one-out*). Stone [154] a d'ailleurs montré l'équivalence asymptotique entre le critère AIC et la validation croisée à N plis.

Sur le plan théorique, la subtile différence entre la validation croisée et les critères théoriques d'information porte essentiellement sur le type d'erreur qu'on cherche à évaluer. Dans la validation croisée, on estime empiriquement l'erreur de prédiction sur des données à part

(*extra-sample error* dans [86]). Dans le cas des critères d'information, c'est l'erreur de prédiction sur données *i.i.d* qui est évaluée : on suppose qu'on peut faire des re-tirages aléatoires (virtuels) de nos données, indépendants et selon une même loi p_f . Il est donc en théorie possible de retirer (virtuellement) des échantillons déjà présents dans l'ensemble $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$ dont on dispose en pratique. A l'extrême, lorsqu'on fait un choix de représentation où $Z = Y$ et qu'on s'intéresse à la densité conditionnelle $p_{Y|X=x}$, on considère des retirages aléatoires des réponses Y_1, \dots, Y_N , pour le même jeu d'entrées x_1, \dots, x_N , fixées. C'est d'ailleurs le parti-pris de certains auteurs qui présentent les critères d'information comme une estimation de cette erreur sur le même échantillon (*in-sample prediction error*, dans [86]). En pratique, le choix entre estimation empirique par validation croisée ou utilisation d'un critère théorique peut dépendre de considérations plus prosaïques, sachant que la validation croisée est plus coûteuse en temps de calcul et en données.

Les différentes méthodes présentées, qu'elles soient empiriques ou théoriques, pénalisent plus ou moins fortement les hypothèses complexes : la validation croisée à N plis et le critère AIC auront tendance à sélectionner des hypothèses plus complexes que la validation croisée à 5 ou 10 plis ou le critère BIC. Le principe du rasoir d'Occam peut nous laisser penser que l'hypothèse la plus simple expliquant nos données est toujours la meilleure, mais ce n'est pas forcément toujours le cas, selon le problème traité. Le choix de la méthode de sélection est donc largement orienté par des *a priori* ou des connaissances externes aux données.

3.6 Réduction de la dimensionalité

Nous avons vu dans la section 3.3.4 que le risque de sur-apprentissage est d'autant plus important que le nombre N d'exemples est faible et que la complexité du modèle est élevée. Pour un certain nombre de méthodes paramétriques, la complexité du modèle dépend du nombre de paramètres à ajuster. Or, ce nombre est généralement d'autant plus grand que le nombre de variables explicatives est grand. Une conséquence directe, dans ce cas, est que plus on prend de variables explicatives en entrée, plus on risque le sur-apprentissage.

Pour illustrer ce propos, considérons simplement un modèle linéaire où la réponse y à un vecteur d'entrée $x = (x_1, \dots, x_D)^T$ de dimension D est une combinaison linéaire $y = \beta_0 + \sum_{d=1}^D \beta_d x_d$. On voit que plus la dimension D du vecteur d'entrée augmente, plus la dimension du vecteur β de paramètres à ajuster augmente. On a donc un risque de sur-apprentissage lorsque le nombre d'exemples N n'est pas suffisamment grand devant D , le nombre de variables explicatives en entrée.

Par ailleurs, nous avons évoqué le "*fléau de la dimension*" dans la section 3.3.6, avec notamment la difficulté intrinsèque à échantillonner correctement un espace multi-dimensionnel de dimension D grande. Il est donc important, lorsque c'est possible, de réduire la dimensionnalité de l'espace des variables d'entrée, afin d'une part de réduire les risques de sur-apprentissage, et d'autre part de permettre une plus grande efficacité des algorithmes d'apprentissage utilisés.

Pour cela, on peut chercher à sélectionner les variables les plus pertinentes, afin de n'utiliser qu'un sous-ensemble restreint des variables initiales. On peut aussi effectuer au préalable une transformation de ces variables, afin d'identifier les dépendances existantes et la dimensionnalité intrinsèque des données.

3.6.1 Sélection de variables pertinentes

Si l'idée d'utiliser un sous-ensemble de variables pertinentes paraît simple, sa réalisation se heurte à quelques difficultés, la principale étant d'identifier quelles sont les variables pertinentes. Plusieurs approches sont proposées dans la littérature, qu'on peut catégoriser de la façon suivante : les méthodes "filtre" où on calcule un score individuel à chaque variable afin d'éliminer celles ayant un score trop faible, les méthodes "symbiose" qui cherchent à identifier les meilleures combinaisons possibles de variables, et les méthodes intégrées où la sélection de variables fait partie intégrante de l'algorithme d'apprentissage, comme dans les arbres de régression par exemple.

Calcul d'un score pour chaque variable

Pour les méthodes "filtre", dans le cas de variables explicatives réelles et si $y \in \mathbb{R}$, on peut par exemple choisir comme score alloué à chaque variable explicative x_d ($1 \leq d \leq D$) la corrélation linéaire empirique $\frac{\text{Cov}(x_d, y)}{\sigma_{x_d} \sigma_y}$ entre x_d et la variable réponse y . L'avantage des méthodes "filtre" est que le calcul des scores est en $O(D)$ par rapport au nombre de variables considérées. Ce type d'approche présente toutefois plusieurs inconvénients. Tout d'abord, ce score spécifique fait l'hypothèse d'une relation linéaire entre variables explicatives et variable réponse. Plus généralement, et quel que soit le score choisi, celui-ci est calculé individuellement pour chaque variable. Deux variables redondantes de scores élevé pourront donc être sélectionnées, alors qu'une seule aurait pu suffire. Enfin, il est bien connu que des combinaisons de variables prises ensemble peuvent avoir un pouvoir explicatif fort, tout en ayant de mauvais scores lorsqu'elles sont prises individuellement.

Exploration des sous-ensembles de variables

Les méthodes de type "symbiose" s'intéressent aux sous-ensembles de variables, plutôt qu'aux variables prises individuellement, et calculent leur pertinence par rapport au problème posé. Typiquement, la pertinence est estimée en appliquant un même algorithme d'apprentissage à différents sous-ensembles de variables d'entrée, et en estimant l'erreur de généralisation de chaque modèle ajusté, par validation croisée ou par un critère d'information (voir section 3.5).

La Figure 3.10 donne pour exemple les combinaisons possibles avec 4 variables explicatives. Pour D grand, le nombre de sous-ensembles (2^D) peut rapidement rendre impraticable l'exploration exhaustive de toutes les combinaisons possibles. On a alors recours à une stratégie d'exploration des sous-ensembles, par exemple de type *forward selection* (voir Figure 3.10) quand on ajoute incrémentalement des variables dans un ensemble initialement vide, ou encore *backward elimination* quand on retire incrémentalement des variables d'un ensemble initialement plein, ou bien encore à d'autres stratégies faisant par exemple appel à une exploration aléatoire des combinaisons possibles, guidée par une heuristique.

Supposons qu'on dispose d'un critère J permettant d'évaluer la pertinence d'un sous-ensemble d'attributs explicatifs. La méthode de sélection des attributs consiste, pour chaque sous-espace \mathcal{X}_d de dimension $d \leq D$ considéré durant l'exploration des combinaisons possibles, à choisir le sous-ensemble maximisant le critère J .

$$S_d(\mathcal{X}_d) = \arg \max_{X \in \mathcal{X}_d} J(X) \tag{3.52}$$

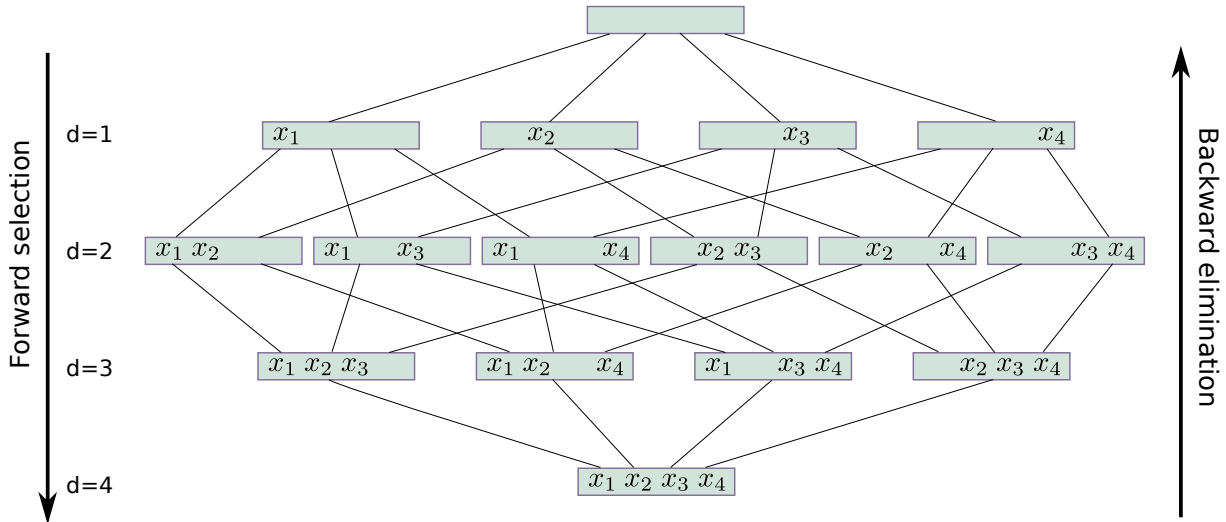


FIGURE 3.10: Sous-ensembles possibles d'un ensemble de 4 variables.

Notons que cette méthode peut s'appliquer soit directement aux variables explicatives, soit aux variables issues d'une transformation de ces variables initiales (voir un peu plus loin).

Méthodes intégrées

Dans certaines méthodes, la sélection des variables explicatives fait partie intégrante de l'algorithme d'apprentissage. C'est par exemple le cas des algorithmes construisant des arbres de régression ou de classification (ex. CART [29]), ou dans les méta-méthodes comme les arbres boostés dans GBM (*gradient boosting machines* [59]), qui font appel à de tels algorithmes (voir section 3.9). De tels arbres de régression ou de classification sont construits en ajoutant incrémentalement des variables explicatives aux nœuds développés, en subdivisant à chaque fois l'espace en deux sous-régions, selon un seuil à déterminer pour chaque nouvelle variable ajoutée. La valeur de la réponse $h(x)$ du modèle est choisie constante dans chaque région de l'espace, comme étant par exemple la moyenne des réponses y_i observées dans cette région, dans le cas de la régression. Les arbres sont dans un premier temps développés jusqu'à ce qu'un critère d'arrêt soit atteint, basé sur le nombre de points présents dans chaque région d'espace, avant de leur appliquer dans un deuxième temps une technique de *pruning* permettant de les réduire. Dans ce type d'algorithme, la sélection des variables est partie intégrante de la méthode.

On peut ranger dans la même catégorie la régression linéaire *lasso* [160], où la régularisation avec une norme L_1 est susceptible d'annuler certains des facteurs β_d de la combinaison linéaire des entrées $\sum_{d=1}^D \beta_d x_d$. Les variables x_d ayant un coefficient nul sont alors *de facto* éliminées de l'ensemble des variables pertinentes.

3.6.2 Transformation des variables de base

Une autre approche, éventuellement complémentaire de la sélection de variables pertinentes que nous venons de présenter, consiste à appliquer une transformation à l'espace des variables explicatives, afin d'identifier les variables latentes du problème d'apprentissage traité, ou encore de déterminer la dimensionalité intrinsèque des données en entrée.

L'analyse en composantes principales (ACP)

Un exemple classique de transformation des données d'entrée est l'analyse en composantes principales [131], ou transformée de Karhunen-Loève, qui consiste à effectuer un changement de base en recherchant les directions orthogonales successives expliquant au mieux la variance des données d'entrée. Les données sont constituées de N vecteurs $\{x_1, \dots, x_N\}$, où chaque vecteur x_n ($1 \leq n \leq N$) est de dimension D : $x_n = (x_{n1}, \dots, x_{nd}, \dots, x_{nD})^T$.

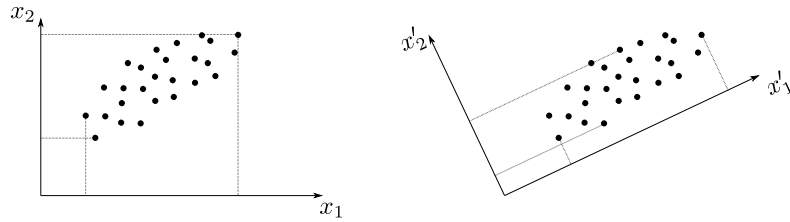


FIGURE 3.11: Illustration de l'intérêt d'un changement de repère.

A titre d'illustration et d'explication intuitive de l'intérêt de ces changements de base pour la sélection de variables, considérons la Figure 3.11 où l'on représente N points dans un espace de dimension 2. Sur la partie gauche de la figure, on constate que lorsqu'on projette les points sur chacun des axes x_1 et x_2 , la dispersion des données (ici grossièrement représentée par les valeurs extrémales) est sensiblement identique dans chacune des dimensions. Par contre, sur la partie droite, la dispersion (et donc la variance) est bien plus grande selon l'axe x'_1 que selon l'axe x'_2 . En supposant que la dispersion selon x'_2 soit très faible, on peut tout à fait envisager de supprimer une dimension et de ne représenter nos données que par leurs valeurs selon x'_1 , la dimension jugée suffisamment représentative de nos données.

Revenons maintenant à un espace des variables explicatives de dimension D quelconque, et supposons qu'on recherche une direction représentée par un vecteur u normé ($u^T u = 1$) telle que la variance des données projetées sur cette direction soit maximale.

La moyenne des données x_1, \dots, x_N est donnée par l'équation (3.53a), et la variance des données projetées sur u est donnée par l'équation (3.53b) :

$$\bar{x} = \frac{1}{N} \sum_{n=1}^N x_n \quad (3.53a)$$

$$\frac{1}{N} \sum_{n=1}^N (u^T x_n - u^T \bar{x})^2 = u^T \Sigma u \quad (3.53b)$$

Dans l'équation (3.53b), Σ est la matrice des covariances :

$$\Sigma = \frac{1}{N} \sum_{n=1}^N (x_n - \bar{x})(x_n - \bar{x})^T \quad (3.54)$$

Maximiser la variance $u^T \Sigma u$ des données projetées sur u s'exprime formellement comme un problème d'optimisation sous contraintes, de la façon suivante :

$$\max_{u \in \mathcal{X}^D} u^T \Sigma u \quad \text{sous les contraintes} \quad u^T u = 1 \quad (3.55)$$

En introduisant un coefficient multiplicateur de Lagrange, ce problème peut être reformulé comme un problème d'optimisation sans contraintes :

$$\max_{\mathbf{u} \in \mathcal{X}^D} [\mathbf{u}^T \Sigma \mathbf{u} + \lambda(1 - \mathbf{u}^T \mathbf{u})] \quad (3.56)$$

Le maximum de $\mathbf{u}^T \Sigma \mathbf{u} + \lambda(1 - \mathbf{u}^T \mathbf{u})$ doit être un point stationnaire. L'annulation de la dérivée par rapport à λ permet de retrouver l'expression des contraintes $\mathbf{u}^T \mathbf{u} = 1$. En dérivant par rapport à \mathbf{u} , on obtient l'équation suivante, qui exprime que \mathbf{u} doit être un vecteur propre de la matrice Σ , de valeur propre λ :

$$\Sigma \mathbf{u} = \lambda \mathbf{u} \quad (3.57)$$

En multipliant à gauche par \mathbf{u}^T et sachant que $\mathbf{u}^T \mathbf{u} = 1$, on voit que la variance $\mathbf{u}^T \Sigma \mathbf{u}$ est égale à la valeur propre λ , au point stationnaire. La variance est donc maximisée en choisissant le vecteur propre \mathbf{u}_1 correspondant à la plus grande valeur propre λ_1 de Σ . En répétant ce raisonnement au sous-espace orthogonal à \mathbf{u}_1 , on peut choisir un second vecteur \mathbf{u}_2 , orthogonal à \mathbf{u}_1 , qui maximise la variance des données projetées sur ce sous-espace. Il s'agit du vecteur propre de seconde plus grande valeur propre λ_2 . En répétant ainsi incrémentalement M fois l'opération, avec $M \leq D$, on obtient une série de vecteurs propres $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M$, de valeurs propres $\lambda_1, \lambda_2, \dots, \lambda_M$.

Choisir $M = D$ revient à effectuer une rotation des axes de coordonnées, pour les aligner à la base des vecteurs propres de Σ . La projection d'un vecteur $\mathbf{x} = (x_1, \dots, x_D)^T$ de dimension D , quelconque, sur l'axe défini par un vecteur propre \mathbf{u}_d est le produit scalaire $\mathbf{x}^T \mathbf{u}_d$. L'analyse en composantes principales, avec $M = D$, revient donc à remplacer les coordonnées (x_1, \dots, x_D) de \mathbf{x} par des variables transformées (x'_1, \dots, x'_D) , avec $x'_d = \mathbf{x}^T \mathbf{u}_d$, pour tout $d \in \{1, \dots, D\}$. Dans ce cadre, l'ACP permet d'identifier les variables latentes, sous forme de combinaisons linéaires des variables d'entrée, qui expliquent au mieux la variance des données. On ne réduit donc pas la dimensionalité des variables d'entrées si on choisit $M = D$, mais on peut cependant appliquer une méthode de sélection (voir section 3.6.1) sur les variables transformées $x'_d = \mathbf{x}^T \mathbf{u}_d$.

L'ACP permet aussi de réduire directement la dimensionalité des données d'entrée en les compressant. Cette méthode consiste à effectuer le changement de base présenté précédemment, pour chacun des vecteurs $\mathbf{x}_1, \dots, \mathbf{x}_N$ de l'ensemble de données, mais en ne transformant les coordonnées de chaque \mathbf{x}_n que pour les M premières dimensions correspondant aux valeurs propres les plus grandes, avec cette fois-ci $M < D$. Les $D - M$ coordonnées suivantes sont prises constantes, $\forall n \in \{1, \dots, N\}$:

$$\tilde{\mathbf{x}}_n = \sum_{d=1}^M \underbrace{(\mathbf{x}_n^T \mathbf{u}_d)}_{x'_{nd}} \mathbf{u}_d + \sum_{d=M+1}^D b_d \mathbf{u}_d \quad (3.58)$$

On montre (voir [24] pp. 563-565) qu'un choix optimal pour les constantes b_d est de prendre $b_d = \bar{\mathbf{x}}^T \mathbf{u}_d$, où $\bar{\mathbf{x}}$ est le vecteur moyen de l'équation (3.53a), si on choisit le critère J suivant pour caractériser l'erreur de compression :

$$J = \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2 \quad (3.59)$$

En considérant l'équation (3.53b) appliquée aux D vecteurs propres de la matrice des covariances Σ , et en se rappelant que $\mathbf{u}_d^T \Sigma \mathbf{u}_d = \lambda_d$, la variance totale des données peut être décomposée comme suit :

$$\frac{1}{N} \sum_{n=1}^N \sum_{d=1}^D (\mathbf{x}_n^T \mathbf{u}_d - \bar{\mathbf{x}}^T \mathbf{u}_d)^2 = \sum_{d=1}^D \mathbf{u}_d^T \Sigma \mathbf{u}_d = \sum_{d=1}^D \lambda_d \quad (3.60)$$

Avec les constantes $b_d = \bar{\mathbf{x}}^T \mathbf{u}_d$, les données compressées s'écrivent sous la forme suivante :

$$\tilde{\mathbf{x}}_n = \sum_{d=1}^M (\mathbf{x}_n^T \mathbf{u}_d) \mathbf{u}_d + \sum_{d=M+1}^D (\bar{\mathbf{x}}^T \mathbf{u}_d) \mathbf{u}_d \quad (3.61)$$

On montre facilement que le critère J s'exprime alors comme suit :

$$J = \frac{1}{N} \sum_{n=1}^N \sum_{d=M+1}^D (\mathbf{x}_n^T \mathbf{u}_d - \bar{\mathbf{x}}^T \mathbf{u}_d)^2 = \sum_{d=M+1}^D \mathbf{u}_d^T \Sigma \mathbf{u}_d = \sum_{d=M+1}^D \lambda_d \quad (3.62)$$

En considérant (3.60) et (3.62), on voit que remplacer \mathbf{x}_n par la donnée compressée $\tilde{\mathbf{x}}_n$ de l'équation (3.61) revient à considérer comme négligeables les $D - M$ termes de variance la plus faible dans l'équation (3.60).

En pratique, le choix de M peut être guidé par le rapport entre la variance expliquée par les M premières composantes et la variance totale, et on peut considérer que la dimensionalité intrinsèque des données est la valeur de M pour laquelle ce rapport est jugé suffisamment grand. Cette réduction de dimensionalité ($M < D$) permet de n'utiliser que les M premières composantes principales (*i.e.* les coordonnées transformées $x'_d = \mathbf{x}^T \mathbf{u}_d$) comme nouvelles variables explicatives de l'algorithme d'apprentissage utilisé pour construire un modèle de prédiction de la variable réponse y .

Autres transformations possibles

Nous venons de présenter la forme classique de l'ACP, mais il existe de nombreuses autres transformations possibles dont l'objectif est de réduire la dimensionalité des données d'entrée. Par exemple, l'ACP peut-être étendue dans un cadre probabiliste, ou à des transformations non linéaires des variables d'entrée (voir [24], section 12.2).

L'ACP dans sa forme canonique est une transformation linéaire des variables initiales. Il est également possible d'effectuer des transformations non linéaires, par exemple en utilisant un réseau de neurones auto-associatif à 3 couches cachées, où la réponse en sortie du réseau est égale à l'entrée \mathbf{x} , et où la couche cachée du milieu est de dimension $M < D$, comme illustré sur la Figure 3.12.

Une fois ce réseau auto-associatif appris sur les seules données de l'espace \mathcal{X} des variables explicatives, la moitié gauche de ce réseau (de la première couche à la couche cachée du milieu, incluse) peut être utilisée pour transformer les données d'entrée. Les données de dimension $M < D$ peuvent alors être utilisées comme nouvelles variables explicatives pour construire un modèle de prédiction de la variable réponse y .

Pour plus de détails sur ces autres approches de la réduction de dimensionalité, évoquées très succinctement ici, le lecteur est invité à se reporter à [23], pp. 314-317, ou [24], section 12.4.

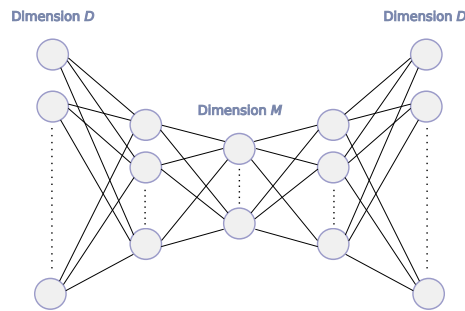


FIGURE 3.12: Exemple de réseau auto-associatif pour la réduction de dimensionalité.

Ceci clôt la présentation des principes et du cadre méthodologique de l'apprentissage. Les sections suivantes sont consacrées à la description de quelques méthodes employées dans les chapitres 4 et 5 : les modèles linéaires, les réseaux de neurones, et les *gradient boosting machines* (GBM).

3.7 Les modèles linéaires

Cette section est une courte introduction aux modèles linéaires pour la régression ou la classification. Son objectif n'est pas d'être exhaustive (voir [58] pour une présentation plus complète), mais d'apporter un éclairage permettant de mieux appréhender des modèles plus complexes présentés dans les sections suivantes.

3.7.1 Les modèles linéaires pour la régression

Cas où $y \in \mathbb{R}$

On s'intéresse ici au cas où la réponse est une variable réelle $y \in \mathbb{R}$. Les variables d'entrée peuvent être multiples, et sont représentées par un vecteur $\mathbf{x} = (x_1, \dots, x_D)^T$ de dimension D .

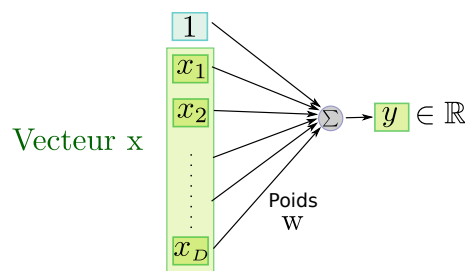


FIGURE 3.13: Modèle linéaire pour la régression.

L'espace des hypothèses \mathcal{H} correspondant à l'ensemble des combinaisons linéaires des entrées s'exprime par l'équation (3.63), paramétrée par les poids w_d ($d = 0, \dots, D$).

$$\mathcal{H} = \{h_w : \mathcal{X} \rightarrow \mathcal{Y}, \quad h_w(\mathbf{x}) = w_0 + \sum_{d=1}^D w_d x_d\} \quad (3.63)$$

La Figure (3.13) donne le schéma du modèle linéaire de base. Le biais w_0 ajouté à la combinaison linéaire $\sum_{d=1}^D w_d x_d$ a pour objectif de permettre de recaler, si besoin, les valeurs prises par la somme pondérée des entrées par rapport aux valeurs prises par la variable réponse.

En introduisant artificiellement une dimension supplémentaire x_0 n'apparaissant pas dans les variables de départ, et en imposant $x_0 = 1$, le biais w_0 peut être considéré comme un poids comme les autres. En notant \mathcal{X}_e l'espace des vecteurs $\mathbf{x} = (x_0, x_1, \dots, x_D)^T$ de dimension $D+1$ tels que $x_0 = 1$, l'équation (3.63) peut être réécrite de façon plus compacte en l'équation (3.64).

$$\mathcal{H} = \{h_w : \mathcal{X}_e \rightarrow \mathcal{Y}, \quad h_w(\mathbf{x}) = \mathbf{w}^T \mathbf{x}\} \quad (3.64)$$

L'utilité du passage en dimension $D+1$, avec l'espace \mathcal{X}_e apparaîtra plus évidente lorsque nous aborderons le cas des problèmes de classification, avec l'interprétation géométrique des classifieurs linéaires.

Cas où $y \in \mathbb{R}^P$

Le cas où la variable réponse est un vecteur de variables réelles $y = (y_1, \dots, y_P)^T$ peut éventuellement être ramené, dans le cas des modèles linéaires, à une juxtaposition de P modèles linéaires renvoyant chacun une composante y_p , pour $p = 1, \dots, P$. Ce choix est tout à fait justifié si on utilise une approche directe où le concept à apprendre est une fonction (linéaire) $h_w : \mathbb{R}^D \rightarrow \mathbb{R}^P$, et où on minimise une fonction de perte arbitrairement choisie, sans faire d'hypothèses supplémentaires.

On peut aussi choisir une représentation où la réponse y est la somme de $h_w(\mathbf{x})$ et d'un vecteur de bruit ϵ , de dimension P (voir section 3.4.1). Dans ce cas, la modélisation de ce bruit peut faire intervenir une matrice de covariance Σ . Si cette matrice de covariance est diagonale, on se ramène à nouveau à P modèles linéaires juxtaposés, mais il peut dans certains cas être intéressant de modéliser une matrice non diagonale. C'est typiquement le cas si ϵ est un bruit gaussien multi-dimensionnel de matrice de covariance Σ quelconque.

$$p(\epsilon) = \mathcal{N}(0_P, \Sigma) = (2\pi)^{-\frac{P}{2}} |\Sigma|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \epsilon^T \Sigma^{-1} \epsilon \right\} \quad (3.65)$$

Il n'est plus possible dans de tels cas de considérer indépendamment les différentes composantes du vecteur $y = h_w(\mathbf{x}) + \epsilon$.

3.7.2 Les modèles linéaires pour la classification

Cas du problème à deux classes, avec $y \in \{-1, 1\}$

Se plonger dans un espace de dimension $D+1$, comme nous l'avons fait pour introduire l'équation (3.64) du modèle linéaire pour la régression, permet une interprétation géométrique éclairante dans le cas où on s'intéresse à un problème de classification à deux catégories, où la réponse $y \in \{-1, 1\}$ est binaire.

En effet, l'équation $\mathbf{w}^T \mathbf{x} = 0$ définit un hyperplan passant par l'origine, dans l'espace de dimension $D+1$ des vecteurs $\mathbf{x} = (x_0, x_1, \dots, x_D)^T$. Cet hyperplan de dimension D et de vecteur directeur $\mathbf{w} = (w_0, w_1, \dots, w_D)^T$ intersecte l'hyperplan d'équation $x_0 = 1$. L'orientation

du vecteur w permet éventuellement, si le problème est linéairement séparable, de séparer les données observées x_1, \dots, x_N ⁸ en deux groupes distincts selon leur classe d'appartenance. La Figure 3.14 montre, en dimension $D = 2$, un exemple d'un tel hyperplan séparant les exemples appartenant à deux classes distinctes, ici représentées par les couleurs ocre et bleue.

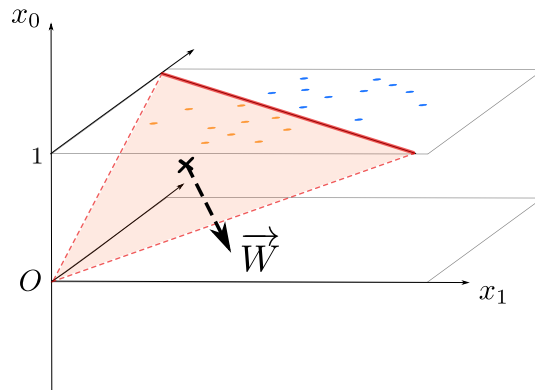


FIGURE 3.14: Hyperplan passant par l'origine en dimension $D + 1$.

Pour les problèmes non linéairement séparables, on cherche à placer l'hyperplan de façon à discriminer au mieux les deux classes. On parle alors d'hyperplan discriminant.

En se plaçant dans l'espace de dimension $D + 1$, les points x_1, \dots, x_N de l'ensemble d'exemples $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$ sont tous situés dans l'hyperplan d'équation $x_0 = 1$, de dimension D . L'appartenance à une des deux classes est codée par une étiquette $y_n = -1$ ou $y_n = +1$.

Pour un modèle linéaire de prédiction de poids w donnés, l'appartenance d'un point x quelconque à une des deux classes s'exprime géométriquement par le fait que ce point est situé d'un côté ou de l'autre ($w^T x \geq 0$ ou $w^T x < 0$) de l'hyperplan discriminant d'équation $w^T x = 0$.

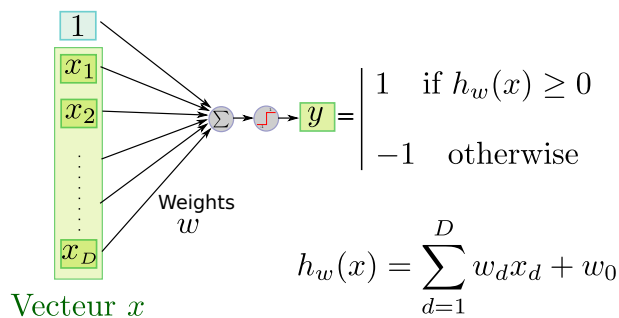


FIGURE 3.15: Modèle linéaire pour un problème de classification à deux catégories.

Pour calculer la réponse y à un vecteur d'entrée x , il suffit donc d'appliquer une fonction échelon à la somme pondérée des entrées, ce qu'on peut représenter par le schéma de la Figure (3.15). Avec le codage choisi pour la réponse, et en notant H la fonction échelon telle que $H(x) = -1$ si $x < 0$ et $H(x) = 1$ si $x \geq 0$, l'espace des hypothèses \mathcal{H} pour un modèle

8. Chaque point x_n est un vecteur de dimension D : $x_n = (x_{n1}, \dots, x_{nD})^T$.

linéaire pour le problème de classification binaire s'exprime donc comme suit :

$$\mathcal{H} = \{h_w : \mathcal{X}_e \rightarrow \mathcal{Y}, \quad h_w(\mathbf{x}) = H(\mathbf{w}^T \mathbf{x})\} \quad (3.66)$$

Un exemple classique d'algorithme de classification s'appuyant sur un tel modèle linéaire pour un problème à deux classes est l'algorithme du perceptron introduit au milieu des années 50 par Frank Rosenblatt [138]. Cet algorithme ne fonctionne correctement que s'il existe effectivement un hyperplan séparateur (*i.e.* les données sont linéairement séparables). L'hyperplan obtenu n'est par ailleurs pas toujours très bien placé par rapport au données.

De ce point de vue, les séparateurs à vaste marge (SVM, ou *Support Vector Machines*) de Vapnik peuvent être vus comme une façon de placer un hyperplan discriminant en maximisant la marge le séparant des points des deux classes.

Après une période d'enthousiasme initial⁹, les perceptrons connurent une période d'oubli relatif suite au livre de Minsky et Papert [121], sous le prétexte qu'ils sont incapables d'apprendre une fonction logique simple comme le "OU exclusif". Les réseaux de neurones, et particulièrement les perceptrons multi-couches, présentés plus loin dans la section 3.8, permirent entre autres de remédier à cet inconvénient.

Cas des problèmes à plus de 2 classes

Les problèmes de classification à P classes ($P > 2$) peut être traité en considérant P fonctions discriminantes $h_p(\mathbf{x}) = \mathbf{w}_p^T \mathbf{x}$, où $\mathbf{x} \in \mathcal{X}_e$ appartient à l'espace étendu à la dimension $D + 1$ en rajoutant la coordonnée $x_0 = 1$.

Un point \mathbf{x} est alors affecté à la classe \mathcal{C}_k si $\forall p \in \{1, \dots, P\}, \quad h_k(\mathbf{x}) \geq h_p(\mathbf{x})$. L'hyperplan discriminant entre deux classes quelconques \mathcal{C}_i et \mathcal{C}_j est défini par l'équation :

$$(\mathbf{w}_i - \mathbf{w}_j)^T \mathbf{x} = 0 \quad (3.67)$$

3.7.3 Ajustement des poids d'un modèle linéaire

Jusqu'à présent, nous avons simplement décrit les espaces d'hypothèses correspondant à des modèles linéaires, en adoptant une représentation fonctionnelle de ces modèles, sans présenter d'algorithme permettant concrètement d'ajuster les poids \mathbf{w} des combinaisons linéaires.

De façon générale, le choix de l'algorithme d'apprentissage dépend naturellement à la fois du type de modèle considéré, et de la fonction-objectif minimisée. Nous allons nous contenter ici de présenter la méthode des moindres carrés, et la régression ridge, dans le cas d'une variable réponse réelle $y \in \mathbb{R}$. Ces méthodes correspondent à un choix d'une perte quadratique, régularisée ou non.

Régression linéaire par la méthode des moindres carrés

La méthode des moindres carrés consiste à minimiser la fonction d'erreur donnée par l'équation (3.68) suivante. Elle revient à minimiser le risque empirique associé à une perte

9. Dans un article du 7 juillet 1957, le New York Times annonçait le perceptron de Rosenblatt comme étant l'embryon d'un calculateur électronique qui serait bientôt capable de marcher, parler, voir, écrire, se reproduire, et être conscient de son existence.

quadratique $\ell(y, h_w(x)) = (y - h_w(x))^2$, en faisant l'hypothèse d'un modèle h_w linéaire.

$$\mathcal{E}(w) = \sum_{n=1}^N \left(y_n - w_0 - \sum_{d=1}^D w_d x_{nd} \right)^2 \quad (3.68)$$

Introduisons la matrice \mathbf{X} de l'équation (3.69) suivante, dont les N lignes sont les N vecteurs $\mathbf{x}_1, \dots, \mathbf{x}_N$ de l'ensemble d'exemples, plongés dans \mathcal{X}_e . Chaque vecteur \mathbf{x}_n a donc pour coordonnées $(1, x_{n1}, \dots, x_{nd}, \dots, x_{nD})$ dans l'espace \mathcal{X}_e des variables explicatives, étendu à la dimension $D + 1$ par l'ajout de la coordonnée $x_0 = 1$.

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & \dots & x_{1D} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \dots & x_{nD} \\ \vdots & \vdots & & \vdots \\ 1 & x_{N1} & \dots & x_{ND} \end{pmatrix} \quad (3.69)$$

Notons \mathbf{y} le vecteur regroupant les N observations de la réponse : $\mathbf{y} = (y_1, \dots, y_N)^T$. Les notations en gras sont ici utilisées pour caractériser les quantités collectant les entrées ou les réponses observées dans un vecteur ou une matrice. Avec ces notations, l'équation (3.68) peut se réécrire sous la forme plus concise de l'équation (3.70) suivante :

$$\mathcal{E}(w) = (\mathbf{y} - \mathbf{X}w)^T (\mathbf{y} - \mathbf{X}w) \quad (3.70)$$

En dérivant \mathcal{E} par rapport à w , on obtient le gradient de l'erreur :

$$\frac{\partial \mathcal{E}}{\partial w} = -2\mathbf{X}^T (\mathbf{y} - \mathbf{X}w) \quad (3.71)$$

La matrice des dérivées secondes s'obtient en dérivant à nouveau :

$$\frac{\partial^2 \mathcal{E}}{\partial w \partial w^T} = 2\mathbf{X}^T \mathbf{X} \quad (3.72)$$

Si $\mathbf{X}^T \mathbf{X}$ est de rang $D + 1$, inversible, on obtient le minimum de l'erreur \mathcal{E} en annulant le gradient (3.71). Ce minimum unique est atteint pour la valeur suivante de w :

$$\hat{w}^{\text{LS}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (3.73)$$

Régression linéaire *ridge*

Étant donné un hyperparamètre de régularisation λ fixé, on cherche ici un vecteur de poids $w = (w_0, w_1, \dots, w_D)^T$ qui minimise la fonction d'erreur suivante :

$$\mathcal{E}'(w_0, \dots, w_D) = \sum_{n=1}^N \left(y_n - w_0 - \sum_{d=1}^D w_d x_{nd} \right)^2 + \lambda \sum_{d=1}^D w_d^2 \quad (3.74)$$

Cette fonction d'erreur régularisée est simplement la somme de l'erreur des moindres carrés (3.68) et d'une pénalité ici proportionnelle à $\sum_{d=1}^M w_d^2$. Notons que w_0 est exclus de ce terme de régularisation. En effet, rappelons que w_0 sert à recaler la somme pondérée des entrées, par rapport aux valeurs observées pour la réponse. Or il n'y a aucune raison de changer la façon d'ajuster le modèle si l'on décale d'une constante c toutes les valeurs des réponses observées y_n , ce qui serait le cas si on introduisait w_0 dans le terme de régularisation : la somme des carrés des poids serait modifiée, avec une valeur différente de w_0 , ce qui correspondrait *de facto* à choisir un modèle avec un équilibre biais-variance différent. Il faut donc traiter w_0 un peu différemment des autres poids, dans la régression *ridge*.

Pour trouver les poids minimisant l'erreur (3.74), on centre en premier lieu les variables d'entrée sur leurs moyennes, c'est-à-dire qu'on remplace les entrées x_{nd} par $x_{nd} - \bar{x}_d$, où $\bar{x}_d = \frac{1}{N} \sum_{n=1}^N x_{nd}$ est la moyenne empirique pour la variable explicative x_d . L'erreur \mathcal{E}' se réécrit alors de la façon suivante, en introduisant une nouvelle expression w'_0 pour le biais :

$$\mathcal{E}'(w'_0, w_1, \dots, w_D) = \sum_{n=1}^N \left(y_n - w'_0 - \sum_{d=1}^D w_d (x_{nd} - \bar{x}_d) \right)^2 + \lambda \sum_{d=1}^D w_d^2 \quad (3.75)$$

avec $w'_0 = w_0 + \sum_{d=1}^D w_d \bar{x}_d$

La valeur du biais w'_0 est déterminée en annulant la dérivée de l'erreur par rapport à w'_0 , ce qui donne facilement la valeur suivante :

$$\widehat{w}_0^{\text{ridge}} = \bar{y} = \frac{1}{N} \sum_{n=1}^N y_n \quad (3.76)$$

On en déduit la valeur ajustée de w_0 :

$$\widehat{w}_0^{\text{ridge}} = \bar{y} - \sum_{d=1}^D w_d \bar{x}_d \quad (3.77)$$

Notons \mathbf{y}_c le vecteur regroupant les valeurs centrées des N réponses, et \mathbf{X}_c la matrice des valeurs centrées des N observations des vecteurs \mathbf{x} d'entrée, en ne considérant que les coordonnées dans l'espace \mathcal{X} de dimension D (*i.e.* sans ajouter la coordonnée $x_{\cdot 0} = 1$) :

$$\mathbf{X}_c = \begin{pmatrix} x_{11} - \bar{x}_1 & \dots & x_{1D} - \bar{x}_D \\ \vdots & & \vdots \\ x_{N1} - \bar{x}_1 & \dots & x_{ND} - \bar{x}_D \end{pmatrix} \quad \mathbf{y}_c = \begin{pmatrix} y_1 - \bar{y} \\ \vdots \\ y_N - \bar{y} \end{pmatrix} \quad (3.78)$$

En remplaçant w'_0 dans l'équation (3.75) par sa valeur ajustée $\widehat{w}_0^{\text{ridge}} = \bar{y}$, on trouve aisément l'expression suivante de la fonction d'erreur, où $\mathbf{w}_c = (w_1, \dots, w_D)^T$ est le vecteur de dimension D regroupant les poids, à l'exception du biais w_0 :

$$\mathcal{E}'(\widehat{w}_0^{\text{ridge}}, \mathbf{w}_c) = (\mathbf{y}_c - \mathbf{X}_c \mathbf{w}_c)^T (\mathbf{y}_c - \mathbf{X}_c \mathbf{w}_c) + \lambda \mathbf{w}_c^T \mathbf{w}_c \quad (3.79)$$

Le gradient et la matrice des dérivées secondes de l'erreur par rapport aux poids w_c s'expriment alors par les équations (3.80) et (3.81) :

$$\frac{\partial \mathcal{E}'}{\partial w_c} = -2\mathbf{X}_c^T (\mathbf{y}_c - \mathbf{X}_c w_c) + 2\lambda w_c = 2(\mathbf{X}_c^T \mathbf{X}_c + \lambda I_D) w_c - 2\mathbf{X}_c^T \mathbf{y}_c \quad (3.80)$$

$$\frac{\partial^2 \mathcal{E}'}{\partial w_c \partial w_c^T} = 2(\mathbf{X}_c^T \mathbf{X}_c + \lambda I_D) \quad (3.81)$$

Si $\mathbf{X}_c^T \mathbf{X}_c + \lambda I_D$ est de rang D , inversible, on obtient le minimum de l'erreur \mathcal{E}' en annulant le gradient de l'équation (3.80). Le minimum est alors atteint pour la valeur suivante de w_c :

$$\widehat{w}_c^{\text{ridge}} = (\mathbf{X}_c^T \mathbf{X}_c + \lambda I_D)^{-1} \mathbf{X}_c^T \mathbf{y}_c \quad (3.82)$$

3.7.4 Propriétés statistiques

Dans ce qui précède, pour ajuster les poids, nous avons implicitement adopté une approche directe (voir section 3.4.1) : nous avons simplement minimisé une fonction d'erreur afin d'approximer la "vraie fonction" f ayant produit les réponses observées $y_n = f(x)$ par un modèle linéaire $h_w(x) = w^T x$.

Pour aller plus loin, on peut modéliser la réponse comme une variable aléatoire Y vérifiant (3.83), où ϵ est un bruit aléatoire de moyenne nulle ($\mathbb{E}(\epsilon) = 0$) et de variance σ^2 constante. Chaque réponse observée y est alors une réalisation de Y .

$$Y = h_w(x) + \epsilon \quad (3.83)$$

Propriétés des moindres carrés ordinaires

L'estimateur des moindres carrés \widehat{w}^{LS} est un estimateur linéaire : $\widehat{w}^{\text{LS}} = \mathbf{M}\mathbf{y}$ avec $\mathbf{M} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$. Il a par ailleurs les propriétés suivantes.

$$\mathbb{E}(\widehat{w}^{\text{LS}}) = \mathbb{E}(\mathbf{M}\mathbf{y}) = \mathbf{M}\mathbb{E}(\mathbf{y}) = \mathbf{M}\mathbf{X}\mathbf{w} = \mathbf{w} \quad (3.84)$$

C'est donc un estimateur "sans biais", au sens où, si le vrai phénomène ayant produit les données observées est bien linéaire, de la forme $Y = w^T x + \epsilon$ et vérifiant les hypothèses énoncées impliquant notamment $\mathbb{E}(Y|x) = w^T x = x^T w$, alors l'espérance mathématique $\mathbb{E}(x^T \widehat{w}^{\text{LS}} | x)$ des réponses obtenues en utilisant l'estimateur des moindres carrés est bien égale à la vraie valeur $w^T x$. Cet estimateur est donc sans biais uniquement si l'on se place dans le cadre de l'*identification*. Si l'on est par contre dans le cadre de l'*approximation*, où l'on cherche à approximer la "vraie fonction" f (ou plus généralement le phénomène réel ayant produit les données) sans que celle-ci ne fasse nécessairement partie de l'espace d'hypothèses \mathcal{H} , alors il peut bien entendu subsister un biais, et nous verrons avec les propriétés de la régression *ridge* qu'il peut même être bénéfique d'augmenter ce biais pour diminuer en contrepartie la variance du modèle obtenu.

La matrice des covariances $\mathbb{V}(\widehat{\mathbf{w}}^{\text{LS}})$ de l'estimateur des moindres carrés s'exprime de la façon suivante, sachant que $\mathbf{X}^T \mathbf{X}$, et donc $(\mathbf{X}^T \mathbf{X})^{-1}$, sont des matrices symétriques et que $\mathbb{V}(\mathbf{y}) = \sigma^2 \mathbf{I}_N$:

$$\mathbb{V}(\widehat{\mathbf{w}}^{\text{LS}}) = \mathbb{V}(\mathbf{M}\mathbf{y}) = \mathbf{M} \mathbb{V}(\mathbf{y}) \mathbf{M}^T = \sigma^2 \mathbf{M} \mathbf{M}^T = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1} \quad (3.85)$$

D'après le théorème de Gauss-Markov, l'estimateur des moindres carrés est l'estimateur linéaire "sans biais" de variance minimale (voir par exemple [58], chap. 9, pour la démonstration). Comme nous l'avons déjà évoqué, on peut toutefois avoir intérêt à utiliser des estimateurs ayant un biais supérieur, afin d'obtenir une variance encore plus faible.

Enfin, si l'on fait de plus l'hypothèse que le bruit ϵ est gaussien $p(\epsilon) = \mathcal{N}(0, \sigma^2)$ et non auto-corrélé, alors l'estimateur des moindres carrés ordinaires correspond au maximum de vraisemblance (voir section 3.4.2).

Propriétés de la régression *ridge*

Avec l'hypothèse d'un bruit ϵ d'espérance nulle, l'espérance de $\widehat{\mathbf{w}}_c^{\text{ridge}}$ est la suivante, avec $\mathbf{S}_\lambda = \mathbf{X}_c^T \mathbf{X}_c + \lambda \mathbf{I}_D$:

$$\mathbb{E}(\widehat{\mathbf{w}}_c^{\text{ridge}} | \lambda) = \mathbb{E}(\mathbf{S}_\lambda^{-1} \mathbf{X}_c^T \mathbf{y}_c) = \mathbb{E}(\mathbf{S}_\lambda^{-1} \mathbf{X}_c^T \mathbf{X}_c \widehat{\mathbf{w}}_c^{\text{LS}}) = \mathbf{S}_\lambda^{-1} \mathbf{X}_c^T \mathbf{X}_c \mathbf{w}_c = \mathbf{w}_c - \lambda \mathbf{S}_\lambda^{-1} \mathbf{w}_c \quad (3.86)$$

L'estimateur *ridge* introduit donc un biais supplémentaire, qui n'existait pas pour l'estimateur des moindres carrés. Si par ailleurs le bruit est de variance constante σ^2 , la matrice des variances-covariances de $\widehat{\mathbf{w}}_c^{\text{ridge}}$ est alors la suivante :

$$\begin{aligned} \mathbb{V}(\widehat{\mathbf{w}}_c^{\text{ridge}} | \lambda) &= \mathbb{V}(\mathbf{S}_\lambda^{-1} \mathbf{X}_c^T \mathbf{X}_c \widehat{\mathbf{w}}_c^{\text{LS}}) = \mathbf{S}_\lambda^{-1} \mathbf{X}_c^T \mathbf{X}_c \mathbb{V}(\widehat{\mathbf{w}}_c^{\text{LS}}) (\mathbf{S}_\lambda^{-1} \mathbf{X}_c^T \mathbf{X}_c)^T \\ &= \mathbf{S}_\lambda^{-1} \mathbf{X}_c^T \mathbf{X}_c \sigma^2 (\mathbf{X}_c^T \mathbf{X}_c)^{-1} \mathbf{X}_c^T \mathbf{X}_c \mathbf{S}_\lambda^{-1} = \sigma^2 \mathbf{S}_\lambda^{-1} \mathbf{X}_c^T \mathbf{X}_c \mathbf{S}_\lambda^{-1} \end{aligned} \quad (3.87)$$

La variance qu'on aurait obtenue sans régularisation ($\lambda = 0$) peut s'exprimer par l'équation (3.88) suivante, où P est une matrice orthogonale et $\text{Diag}(\lambda_1^{-1}, \dots, \lambda_D^{-1})$ la matrice diagonale dont les éléments sont les inverses des valeurs propres $\lambda_1, \dots, \lambda_D$ de $\mathbf{X}_c^T \mathbf{X}_c$.

$$\mathbb{V}(\widehat{\mathbf{w}}_c^{\text{ridge}} | \lambda = 0) = \sigma^2 (\mathbf{X}_c^T \mathbf{X}_c)^{-1} = P^T \text{Diag}(\lambda_1^{-1}, \dots, \lambda_D^{-1}) P \quad (3.88)$$

La variance avec régularisation de l'équation (3.87) peut alors s'exprimer comme suit, où l'on constate qu'augmenter le biais, avec la valeur de l'hyperparamètre λ , permet de diminuer la variance :

$$\mathbb{V}(\widehat{\mathbf{w}}_c^{\text{ridge}} | \lambda) = P^T \text{Diag} \left(\frac{\lambda_1}{(\lambda_1 + \lambda)^2}, \dots, \frac{\lambda_D}{(\lambda_D + \lambda)^2} \right) P \quad (3.89)$$

3.7.5 Extension à des combinaisons linéaires de fonctions de base

L'application des modèles linéaires peut être largement étendue en remplaçant les entrées du modèle linéaire par des transformations non linéaires de ces entrées. Le modèle consiste alors en une combinaison linéaire de fonctions de base Φ_m , $m = 1, \dots, M$, appliquées au vecteur $\mathbf{x} \in \mathcal{X}$ de variables d'entrée.

Les fonctions de base ϕ_m peuvent être de différentes formes : gaussiennes, sigmoïdes, sinusoïdales, polynomiales, polynomiales par morceaux (splines), etc. L'indice m indique un choix de paramètres différents pour chacune des fonctions ϕ_m , ou bien une forme différente dans le cas de fonctions non paramétriques. Par exemple, dans le cas de fonctions de base gaussiennes $\phi_m(\mathbf{x}) = \exp\{-\frac{\|\mathbf{x}-\mu_m\|^2}{2\sigma_m^2}\}$, les valeurs de la moyenne μ_m et de la variance σ_m^2 diffèrent selon les fonctions ϕ_m . Elles doivent être ajustées au préalable par un apprentissage non supervisé sur les seules variables d'entrée, avant d'ajuster dans un deuxième temps les poids de la combinaison linéaire.

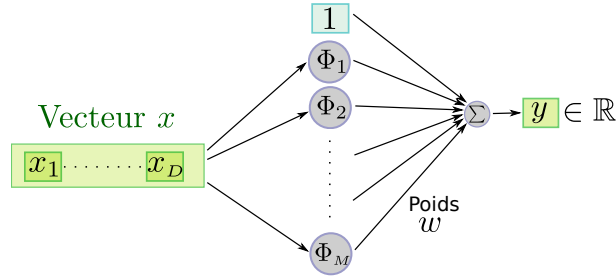


FIGURE 3.16: Combinaison linéaire de fonctions de base, dans le cas où $y \in \mathbb{R}$.

Comme dans les modèles linéaires simples présentés précédemment, on rajoute un biais w_0 afin de pouvoir recalibrer les valeurs prises par la somme $\sum_{m=1}^M \phi_m(\mathbf{x})$, par rapport aux valeurs prises par la réponse y . En posant $\phi_0(\mathbf{x}) = 1$, et en notant $\phi(\mathbf{x})$ le vecteur $(\phi_0(\mathbf{x}), \dots, \phi_M(\mathbf{x}))^T$ l'espace d'hypothèses des combinaisons linéaires de fonctions de base s'exprime par l'équation (3.90) suivante, où \mathcal{X} est l'espace des variables explicatives de dimension D . On n'a pas besoin ici d'étendre l'espace des variables explicatives à la dimension $D + 1$ en ajoutant une coordonnée $x_0 = 1$: on ajoute ici une coordonnée ϕ_0 à l'espace des fonctions de base.

$$\mathcal{H} = \{h_w : \mathcal{X} \rightarrow \mathcal{Y}, \quad h_w(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})\} \quad (3.90)$$

avec $\phi(\mathbf{x}) = (\phi_0(\mathbf{x}), \dots, \phi_M(\mathbf{x}))^T$ où $\phi_0(\mathbf{x}) = 1$

La Figure 3.16 illustre ce type de modèle, dans le cas d'une variable réponse réelle $y \in \mathbb{R}$. Dans ce cas, les poids w de l'équation (3.90) sont regroupés dans un vecteur $\mathbf{w} = (w_0, w_1, \dots, w_M)^T$.

Dans le cas où la réponse est un vecteur $\mathbf{y} = (y_1, \dots, y_P)^T \in \mathbb{R}^P$, l'expression $\mathbf{w}^T \phi(\mathbf{x})$ de l'équation (3.90) peut simplement être remplacée par $\mathbf{W}\mathbf{x}$, où \mathbf{W} est une matrice de poids à P lignes et $M + 1$ colonnes (si l'on a $M + 1$ fonctions de base, ϕ_0 incluse). Par ailleurs, les modèles linéaires pour la classification présentés dans la section 3.7.2 peuvent aisément se généraliser aux combinaisons linéaires de fonctions de base.

3.7.6 Les modèles linéaires généralisés

Dans certains cas, et en particulier pour traiter des problèmes de classification, il peut être utile d'appliquer une fonction non linéaire à la combinaison linéaire des fonctions de base (ou à la somme pondérée des variables d'entrée, dans le cas de variables non transformées). L'espace d'hypothèses, pour ces modèles linéaires généralisés, s'exprime par l'équation (3.91) suivante :

$$\mathcal{H} = \{h_w : \mathcal{X}_e \rightarrow \mathcal{Y}, \quad h_w(\mathbf{x}) = \varphi(\mathbf{w}^T \phi(\mathbf{x}))\} \quad (3.91)$$

Parmi ces méthodes, citons la *régression logistique*, qui malgré son nom est en fait une méthode de classification pour les problèmes à deux classes. Dans cette méthode, la fonction non linéaire φ appliquée à la combinaison linéaire $w^T x$ est la fonction sigmoïde logistique $\varphi(x) = 1/(1 + \exp\{-x\})$, qu'on retrouvera dans les réseaux de neurones. Le modèle utilisé pour la régression logistique est de fait un réseau de neurones sans couche cachée.

Avec l'introduction de la fonction φ non linéaire, l'ajustement des poids ne peut plus se faire par la simple résolution analytique d'un système linéaire, comme dans la section 3.7.3. On utilise donc une méthode itérative (*iterative reweighted least squares*), avec une mise à jour des poids par la formule de Newton-Raphson. La fonction d'erreur étant convexe, cette méthode converge vers un minimum global unique, ce qui ne sera pas le cas pour les réseaux de neurones que nous allons voir dans la section suivante.

3.8 Réseaux de neurones

L'histoire des réseaux connexionnistes est intimement liée à l'histoire, mouvementée, de l'Intelligence Artificielle. Les premières ébauches de tels réseaux remontent au modèle booléen du cerveau humain proposé par McCulloch et Pitts [117] en 1943. Le perceptron de Rosenblatt [138] au milieu des années 50, et des approches similaires comme l'ADALINE (*adaptive linear neuron*) de Widrow et Hoff [175] dans les années 60, connurent ensuite un engouement alimenté par l'optimisme ambiant sur la possibilité de construire effectivement à plus ou moins court terme une intelligence artificielle. Les réseaux connexionnistes connurent ensuite un relatif abandon suite à la publication en 1969 du livre de Minsky et Papert [121] et la "controverse du XOR"¹⁰, qui fut le prétexte d'une réorientation des budgets vers une approche symbolique de l'intelligence artificielle, et vers le développement de systèmes experts. Au milieu des années 80, avec la (re-)découverte de la rétro-propagation par plusieurs équipes, notamment par Rumelhart et McClelland, mais aussi Werbos, ou LeCun, les perceptrons multi-couches permirent une renaissance de l'approche connexionniste.

Les réseaux de neurones, initialement inspirés par le fonctionnement des neurones et des synapses du cerveau, sont aujourd'hui totalement replacés dans le contexte des méthodes d'apprentissage statistique, en concurrence avec d'autres approches comme les séparateurs à vaste marge (SVM, *support vector machines*), issus des importants travaux de Vapnik [168]. Les SVM ont connu ces dernières années des succès certains, récemment remis en jeu par de nouvelles approches des réseaux connexionnistes permettant l'apprentissage de réseaux profonds ("*deep learning*").

En laissant de côté les références au monde biologique, les réseaux de neurones de type *feed-forward* que nous allons présenter plus en détail dans cette section peuvent être vus comme une composition de fonctions de régression logistique, arrangées en couches successives (voir section 3.8.1), une fonction spécifique pouvant par ailleurs s'appliquer à la sortie de la dernière couche cachée, selon le type de problème traité, comme nous le verrons dans les sections 3.8.2 et 3.8.3. Le terme de *perceptron multi-couches*, consacré par l'usage, est donc en fait impropre, puisque ce n'est pas une fonction échelon discontinue qui est appliquée, comme dans le cas du perceptron (3.66), mais une fonction sigmoïdale continue.

L'algorithme de rétro-propagation, que nous détaillerons dans la section 3.8.4, permet un

10. Les perceptrons simples étant des séparateurs linéaires, ils ne peuvent pas apprendre une fonction logique comme le "OU exclusif" (XOR).

calcul efficace du gradient de l'erreur¹¹, rendu plus facile par la continuité de la fonction de transfert. Les méthodes d'optimisation utilisées pour minimiser l'erreur, que nous verrons dans la section (3.8.5), peuvent selon le cas utiliser ou non le gradient calculé par rétro-propagation, pour ajuster les poids du réseau de neurones.

3.8.1 Le perceptron multi-couches

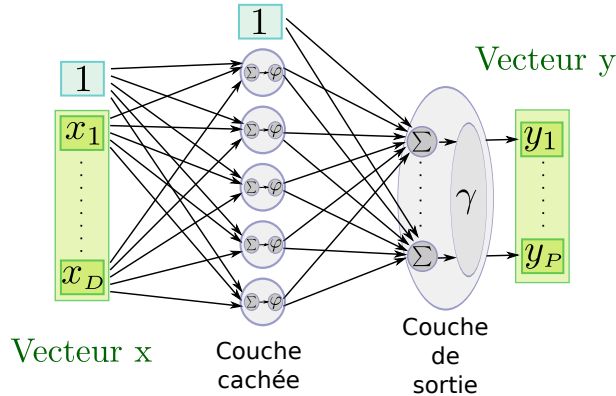


FIGURE 3.17: Exemple de réseau de neurones à une couche cachée.

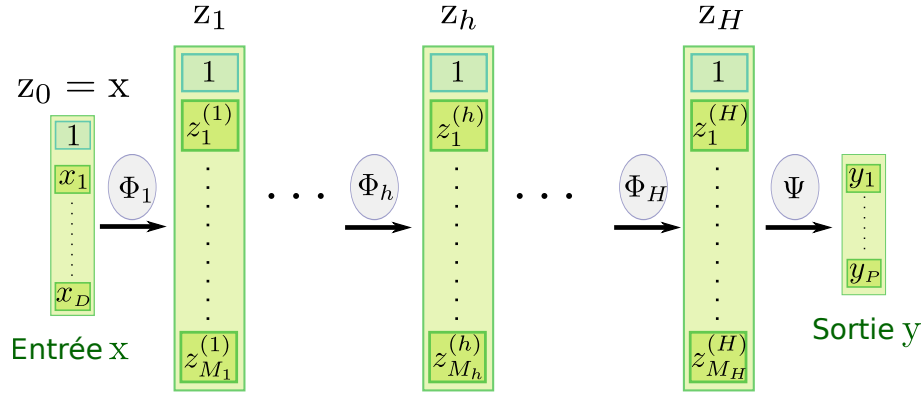
Un réseau de neurones de type *feed-forward* est classiquement représenté par un schéma comme celui de la Figure 3.17, qui illustre un réseau à une couche cachée. Chaque connexion de ce réseau est affectée d'un poids. Chaque unité d'une couche cachée réalise une somme, pondérée par les poids des connexions entrantes, de ses entrées. Une fonction non linéaire φ (ex. sigmoïde) est alors appliquée à cette somme pondérée, pour obtenir la sortie de l'unité. Le traitement réalisé par la couche de sortie peut être légèrement différent : on fait également une somme pondérée des entrées, mais on applique ensuite une fonction γ au vecteur de ces sommes pondérées calculées par les unités de la couche de sortie. La fonction γ peut prendre différentes formes selon le problème traité (classification, ou régression).

Pour la suite de cette section, il est utile d'adopter une représentation un peu différente de celle de la Figure 3.17, en considérant le réseau *feed-forward* comme une composition de modèles linéaires généralisés. L'espace d'hypothèses \mathcal{H} de ces réseaux de neurones peut être décrit de façon générale par l'équation 3.92 suivante, où Φ_1, \dots, Φ_H et Ψ sont des fonctions vectorielles s'appliquant à des vecteurs z_0, \dots, z_H :

$$\mathcal{H} = \{h_w : \mathcal{X}_e \rightarrow \mathcal{Y}, \quad h_w(x) = \Psi \circ \Phi_H \circ \dots \circ \Phi_1(x)\} \quad (3.92)$$

Les fonctions vectorielles Φ_1, \dots, Φ_H et Ψ correspondent aux traitements successifs réalisés sur le flux de données, par les couches cachées et par la couche de sortie, comme illustré sur la Figure 3.18. Ces traitements comprennent des sommations pondérées des entrées, et l'application de fonctions non linéaires, comme nous allons le détailler dans la suite.

11. Le gradient de l'erreur est le vecteur des dérivées partielles de l'erreur par rapport aux poids affectés aux connexions du réseau.

FIGURE 3.18: Forme générale d'un perceptron multi-couche de type *feed-forward*.

Couche d'entrée

Le perceptron multi-couches de type *feed-forward* est alimenté en entrée par le vecteur $\mathbf{x} = (x_1, \dots, x_D)$ de dimension D . Pour noter de façon similaire les unités de biais (fournissant des valeurs constantes égales à 1) et les autres unités du réseau, on passe en dimension $D + 1$ en considérant que le vecteur des entrées est $\mathbf{x} = (1, x_1, \dots, x_D)^T$.

Couches cachées

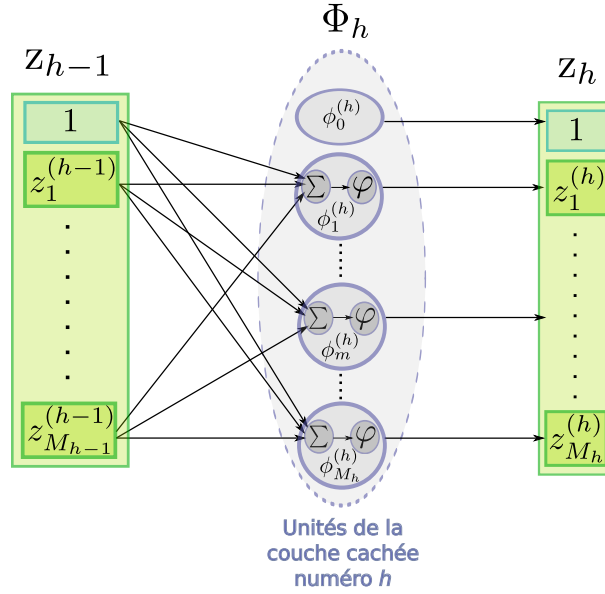
Le réseau comprend par ailleurs une ou plusieurs couches cachées, numérotées de 1 à H , chaque couche comprenant M_h unités. Chaque couche cachée h ($1 \leq h \leq H$) est alimentée par un vecteur $\mathbf{z}_{h-1} = (z_0^{(h-1)}, z_1^{(h-1)}, \dots, z_{M_{h-1}}^{(h-1)})^T$ fourni par la couche précédente (voir Figure 3.19), avec la convention de notation $\mathbf{z}_0 = \mathbf{x}$. Le vecteur en sortie de la couche h est $\mathbf{z}_h = \Phi_h(\mathbf{z}_{h-1})$, avec :

$$\mathbf{z}_h = \begin{pmatrix} z_0^{(h)} \\ z_1^{(h)} \\ \vdots \\ z_{M_h}^{(h)} \end{pmatrix} = \Phi_h(\mathbf{z}_{h-1}) = \begin{pmatrix} \phi_0^{(h)}(\mathbf{z}_{h-1}) \\ \phi_1^{(h)}(\mathbf{z}_{h-1}) \\ \vdots \\ \phi_{M_h}^{(h)}(\mathbf{z}_{h-1}) \end{pmatrix} \quad (3.93)$$

Chaque unité k de la couche cachée h calcule $z_k^{(h)} = \phi_k^{(h)}(\mathbf{z}_{h-1})$ (la $k^{\text{ième}}$ coordonnée du vecteur \mathbf{z}_h) en réalisant les opérations suivantes :

$$\begin{cases} z_0^{(h)} = \phi_0^{(h)}(\mathbf{z}_{h-1}) = 1 & \text{pour l'unité de biais } k = 0 \\ z_k^{(h)} = \phi_k^{(h)}(\mathbf{z}_{h-1}) = \varphi(a_k^{(h)}) & \forall k \in \{1, \dots, M_h\} \end{cases} \quad (3.94)$$

Dans l'équation (3.94), la quantité $a_k^{(h)}$ est la somme pondérée des entrées de l'unité k de la couche cachée h , et φ est une fonction non linéaire (fonction sigmoïde, ou tangente

FIGURE 3.19: Couche cachée d'un perceptron multi-couche de type *feed-forward*.

hyperbolique, typiquement) :

$$a_k^{(h)} = \sum_{i=0}^{M_{h-1}} w_{ik}^{(h-1)} z_i^{(h-1)} \quad \text{si } z_{h-1} = \left(z_0^{(h-1)}, \dots, z_{M_{h-1}}^{(h-1)} \right)^T \quad (3.95a)$$

$$\varphi(a) = \frac{1}{1 + e^{-a}} \quad \text{ou} \quad \varphi(a) = \tanh(a) \quad (3.95b)$$

Couche de sortie

La dernière couche du réseau, appelée couche de sortie, applique une fonction Ψ au vecteur z_H fourni par la dernière couche cachée. Le vecteur de sortie du réseau est donc :

$$\hat{y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_P \end{pmatrix} = \Psi(z_H) = \gamma(a) \quad \text{où } a = \begin{pmatrix} a_1 \\ \vdots \\ a_P \end{pmatrix} \quad (3.96)$$

$$\text{avec } \forall p \in \{1, \dots, P\} \quad a_p = \sum_{i=0}^{M_H} w_{ip}^{(H)} z_i^{(H)} \quad \text{sachant que } z_H = \left(z_0^{(H)}, \dots, z_{M_H}^{(H)} \right)^T$$

Selon la fonction d'erreur que l'on cherche à minimiser, qui peut elle-même éventuellement découler d'un principe d'inférence choisi, la fonction γ appliquée au vecteur des sommes pondérées a_p calculées par la couche de sortie pourra prendre différentes formes. Par exemple, dans le cas de la régression avec une perte quadratique, un choix cohérent sur le plan théorique et permettant un calcul facile du gradient de l'erreur¹², par rétro-propagation dans le réseau,

12. L'erreur est ici vue comme une fonction des poids w , et on cherche ses dérivées partielles par rapport à ces poids, pour pouvoir appliquer une méthode d'optimisation utilisant le gradient.

consiste à prendre pour γ la fonction identité :

$$\gamma : \begin{pmatrix} a_1 \\ \vdots \\ a_P \end{pmatrix} \longrightarrow \begin{pmatrix} a_1 \\ \vdots \\ a_P \end{pmatrix} \quad (3.97)$$

Dans le cas d'un classifieur minimisant l'entropie croisée, pour P classes exclusives, on pourra choisir pour γ la fonction *softmax* :

$$\gamma : \begin{pmatrix} a_1 \\ \vdots \\ a_P \end{pmatrix} \longrightarrow \begin{pmatrix} \gamma_1(\mathbf{a}) \\ \vdots \\ \gamma_P(\mathbf{a}) \end{pmatrix} = \begin{pmatrix} \frac{\exp\{a_1\}}{\sum_{p=1}^P \exp\{a_p\}} \\ \vdots \\ \frac{\exp\{a_P\}}{\sum_{p=1}^P \exp\{a_p\}} \end{pmatrix} \quad (3.98)$$

Principe général de l'apprentissage du réseau de neurones

L'apprentissage du réseau de neurones se fait en ajustant les poids w du réseau de façon à minimiser l'erreur entre la sortie calculée \hat{y} et la sortie observée y , cette erreur étant évaluée sur un ensemble d'exemples $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$ d'entrées et de réponses observées ou désirées. Les données d'apprentissage sont ici considérées comme fixes, et l'erreur en sortie du réseau est vue comme une fonction du vecteur des poids.

L'apprentissage consiste donc à choisir le vecteur w^* minimisant l'erreur $\mathcal{E}(w)$, évaluée au point w sur l'ensemble d'exemples S :

$$w^* = \arg \min_{w \in \Theta} \mathcal{E}(w) \quad (3.99)$$

Contrairement aux modèles linéaires, il n'est pas possible pour les réseaux de neurones d'obtenir par le calcul une forme analytique de cet optimum. On utilise donc des méthodes itératives permettant de l'approximer. Historiquement, ce sont des méthodes utilisant le gradient $\nabla \mathcal{E}(w)$ de l'erreur, c'est-à-dire le vecteur des dérivées partielles par rapport aux poids, qui ont permis la re-émergence des réseaux connexionnistes, dans le milieu des années 1980, grâce à un calcul efficace du gradient par rétro-propagation de l'erreur, que nous présentons dans la section 3.8.4. D'autres méthodes n'utilisant pas le gradient, comme par exemple les métaheuristiques à population, ont depuis été appliquées à l'apprentissage des réseaux de neurones.

La fonction d'erreur $\mathcal{E}(w)$ peut bien entendu être arbitrairement choisie par le praticien, mais ce choix peut être éclairé ou motivé par l'application formelle d'un principe d'inférence. Les deux sections suivantes illustrent l'application du principe de maximum de vraisemblance pour les réseaux de neurones appliqués à des problèmes de régression, et dans le cadre de problèmes de classification à plus de deux classes.

3.8.2 Réseaux de neurones pour la régression

Considérons un modèle où la variable réponse $y \in \mathbb{R}^P$ est un vecteur de dimension P donné par l'équation suivante, avec ϵ un vecteur représentant un bruit aléatoire ajouté à la

réponse calculée par le modèle.

$$y = h_w(x) + \epsilon \quad (3.100)$$

Dans cette équation, toutes les quantités sont des vecteurs :

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_P \end{pmatrix} \quad h_w(x) = \begin{pmatrix} h_1(x) \\ \vdots \\ h_P(x) \end{pmatrix} \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_P \end{pmatrix} \quad (3.101)$$

Faisons l'hypothèse d'un bruit centré gaussien multi-dimensionnel $\mathcal{N}(0_P, \sigma^2 \mathbf{I}_P)$, avec des composantes indépendantes, de variance σ^2 constante et identique dans toutes les dimensions de x . En d'autres termes, la densité de probabilité de ϵ est égale au produit de densités normales de même variance σ^2 :

$$p(\epsilon) = \prod_{p=1}^P \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2} \frac{\epsilon_p^2}{\sigma^2}\right\} = (2\pi\sigma^2)^{-\frac{P}{2}} \exp\left\{-\frac{1}{2} \frac{\epsilon^T \epsilon}{\sigma^2}\right\} \quad (3.102)$$

Par conséquent, la densité conditionnelle de la probabilité d'observer la réponse y , connaissant l'entrée x et le vecteur w de paramètres, suit une loi $\mathcal{N}(h_w(x), \sigma^2 \mathbf{I}_P)$ également gaussienne, centrée sur $h_w(x)$, et l'on a :

$$\begin{aligned} p(y|x, w) &= (2\pi\sigma^2)^{-\frac{P}{2}} \exp\left\{-\frac{1}{2\sigma^2} (y - h_w(x))^T (y - h_w(x))\right\} \\ &= (2\pi\sigma^2)^{-\frac{P}{2}} \exp\left\{-\frac{1}{2\sigma^2} \|y - h_w(x)\|^2\right\} \end{aligned} \quad (3.103)$$

La vraisemblance du modèle h_w au vu d'observations $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$ d'entrées et de réponses associées est définie, à un facteur multiplicatif quelconque près, par la probabilité d'observer les données S , étant donné le modèle h_w fixé. Pour des variables aléatoires continues admettant une densité, et en supposant les N observations tirées de manière indépendante selon une même loi $p_{h_w}(x, y)$, elle s'exprime comme suit :

$$\mathcal{L}(h_w; S) = \prod_{n=1}^N p_{h_w}(x_n, y_n) = \prod_{n=1}^N p(y_n|x_n, w) p_X(x_n) \quad (3.104)$$

La log-vraisemblance s'obtient en passant au logarithme :

$$\ln \mathcal{L}(h_w; S) = \sum_{n=1}^N \ln p(y_n|x_n, w) + \sum_{n=1}^N \ln p_X(x_n) \quad (3.105)$$

Supposons qu'on cherche à ajuster les poids w en maximisant la vraisemblance (et donc la log-vraisemblance). Dans ce cadre, le terme le plus à droite de l'équation (3.105) reste

constant, quel que soit le vecteur de poids choisi, et on n'en tient pas compte dans l'apprentissage du réseau de neurones. Le terme dépendant des poids peut s'exprimer, avec l'hypothèse gaussienne de l'équation (3.103), de la façon suivante :

$$\sum_{n=1}^N \ln p(y_n | x_n, w) = -\frac{1}{2\sigma^2} \sum_{n=1}^N \|y - h_w(x)\|^2 - \frac{NP}{2} \ln(2\pi\sigma^2) \quad (3.106)$$

Le terme le plus à droite de cette équation est également indépendant de w , ainsi que la variance du bruit σ^2 . Il suffit donc, si on choisit comme principe d'inférence le principe du maximum de vraisemblance et si on fait l'hypothèse d'un bruit gaussien, de minimiser l'erreur quadratique suivante :

$$\mathcal{E}(w) = \frac{1}{2} \sum_{n=1}^N \|y_n - h_w(x_n)\|^2 = \frac{1}{2} \sum_{n=1}^N \sum_{p=1}^P (y_{np} - h_p(x_n))^2 \quad (3.107)$$

Soit $w^* = \arg \min_{w \in \Theta} \mathcal{E}(w)$ la valeur de w minimisant l'erreur $\mathcal{E}(w)$. En dérivant l'équation (3.106) par rapport à σ et en annulant cette dérivée, on obtient la valeur de la variance du bruit au maximum de vraisemblance :

$$\sigma^2 = \frac{1}{NP} \mathcal{E}(w^*) \quad (3.108)$$

Nous savions déjà, d'après l'équation (3.13) de la section 3.3.1, que l'estimateur bayésien dans le cas d'une régression avec perte quadratique est la fonction renvoyant l'espérance conditionnelle $\mathbb{E}(Y|X = x)$, ici égale à $h_w(x)$ avec nos hypothèses. L'hypothèse d'un bruit gaussien se rajoutant à $h_w(x)$ nous permet de caractériser la variance de ce bruit, en fonction de l'erreur observée.

En pratique, le bruit ϵ n'est pas forcément gaussien, ce qui n'empêche pas d'utiliser un réseau de neurones avec une perte quadratique. Les résultats seront en tout état de cause indiscernables de ceux obtenus en faisant l'hypothèse d'un bruit gaussien, si la distribution réelle a une espérance conditionnelle $\mathbb{E}(Y|X = x)$ et une variance identiques à ceux de la distribution gaussienne dont nous avons fait l'hypothèse.

Nous avons ici illustré l'application du maximum de vraisemblance comme principe d'inférence. On montre facilement que le choix du *maximum a posteriori* conduit avec les mêmes hypothèses à minimiser la somme d'une erreur quadratique et d'un terme de régularisation dépendant du vecteur des poids. D'autres types de fonctions d'erreur, que nous ne détaillerons pas ici, peuvent aussi s'avérer intéressants. Notamment, dans le cadre de la régression robuste, on préférera minimiser une erreur en valeur absolue plutôt qu'une erreur quadratique, auquel cas la sortie du réseau pourra s'interpréter comme la médiane des valeurs de y , conditionnellement à x .

3.8.3 Réseaux de neurones pour la classification

Considérons un problème de classification, où l'on cherche à affecter les entrées x à une des classes $\mathcal{C}_1, \dots, \mathcal{C}_P$, celles-ci étant supposées mutuellement exclusives. Sachant que x est

toujours affecté à une des P classes, et qu'il ne peut pas appartenir simultanément à 2 classes différentes (*i.e.* les événements $x \in \mathcal{C}_p$ sont deux à deux disjoints) on a l'expression suivante :

$$\mathbb{P} \left(x \in \bigcup_{p=1}^P \mathcal{C}_p \right) = \sum_{p=1}^P \mathbb{P}(x \in \mathcal{C}_p) = 1 \quad (3.109)$$

Si on modélise l'appartenance à une classe par une variable aléatoire \mathcal{C} , l'entrée x étant elle-même une réalisation d'une variable aléatoire X , on peut exprimer cette condition par $\sum_{p=1}^P \mathbb{P}(\mathcal{C} = \mathcal{C}_p | X = x) = 1$. Dans ce qui suit, on utilisera des notations abrégées en écrivant plus simplement :

$$\sum_{p=1}^P \mathbb{P}(\mathcal{C}_p | x) = 1 \quad (3.110)$$

Choix du codage pour P classes mutuellement exclusives

Un codage intéressant pour ce type de problème de classification consiste à prendre comme variable réponse un vecteur de valeurs binaires $y = (y_1, \dots, y_P)^T$ tel que $y_p = 1$ si $x \in \mathcal{C}_p$ et $y_p = 0$ sinon, $\forall p \in \{1, \dots, P\}$. Pour un vecteur x dont on sait avec certitude qu'il appartient à la classe \mathcal{C}_p , la réponse sera donc codée par le vecteur y suivant :

$$y = (0, \dots, 0, \underset{p^{\text{ième comp.}}}{1}, 0, \dots, 0)^T \quad (3.111)$$

Avec ce codage, on a donc :

$$\sum_{p=1}^P y_p = 1 \quad (3.112)$$

Sachant que $\mathbb{P}(\mathcal{C}_p | x) = 1$ quand on est sûr que x appartient à la classe \mathcal{C}_p et que $\mathbb{P}(\mathcal{C}_k | x) = 0$ si $k \neq p$, on peut identifier chaque composante y_p du vecteur de sortie y à une probabilité $\mathbb{P}(\mathcal{C}_p | x)$, dont les valeurs 0 ou 1, dans le cas d'une réponse observée, traduisent une certitude sur l'appartenance à l'une des classes.

Interprétation des sorties du réseau

Un intérêt immédiat de ce codage est donc que, sous certaines conditions, les sorties \hat{y}_p calculées par le réseau peuvent être interprétées comme étant les probabilités conditionnelles d'appartenance aux classes \mathcal{C}_p , sachant l'entrée x :

$$\hat{y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_P \end{pmatrix} = \begin{pmatrix} \mathbb{P}(\mathcal{C}_1 | x) \\ \vdots \\ \mathbb{P}(\mathcal{C}_P | x) \end{pmatrix} \quad (3.113)$$

Pour cela, les sorties calculées \hat{y}_p doivent rester toujours comprises entre 0 et 1, et leur somme doit être égale à 1 :

$$\forall p \in \{1, \dots, P\}, \quad 0 \leq \hat{y}_p \leq 1 \quad (3.114a)$$

$$\sum_{p=1}^P \hat{y}_p = 1 \quad (3.114b)$$

Choix de la fonction d'erreur $\mathcal{E}(\mathbf{w})$

Compte-tenu des valeurs binaires des composantes de la réponse observée, la probabilité pour une entrée \mathbf{x} donnée d'obtenir la réponse $\mathbf{y} = (0, \dots, 0, 1, 0, \dots, 0)^T$ où la $p^{\text{ième}}$ composante est égale à 1 peut s'exprimer de la façon suivante, où $\delta_{kp} = 1$ si $k = p$, et 0 sinon :

$$\mathbb{P}(\mathbf{y}|\mathbf{x}) = \prod_{k=1}^P \mathbb{P}(\mathcal{C}_k|\mathbf{x})^{\delta_{kp}} = \prod_{k=1}^P \mathbb{P}(\mathcal{C}_k|\mathbf{x})^{y_k} \quad (3.115)$$

L'équation précédente peut être rapprochée de la loi binômiale, qu'elle généralise au cas où l'on a plus de deux classes. Comme on souhaite interpréter les composantes \hat{y}_k du vecteur de sortie $\hat{\mathbf{y}}$ calculé par le réseau comme étant les probabilités $\mathbb{P}(\mathcal{C}_k|\mathbf{x})$, cela nous conduit à écrire l'équation suivante :

$$\mathbb{P}(\mathbf{y}|\mathbf{x}, \mathbf{w}) = \prod_{k=1}^P (\hat{y}_k)^{y_k} \quad (3.116)$$

Pour N observations indépendantes $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ la probabilité d'observer y_1, \dots, y_N étant données les entrées $\mathbf{x}_1, \dots, \mathbf{x}_N$ doit donc s'exprimer comme suit :

$$\mathbb{P}(y_1, \dots, y_N | \mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{w}) = \prod_{n=1}^N \mathbb{P}(y_n | \mathbf{x}_n, \mathbf{w}) = \prod_{n=1}^N \prod_{p=1}^P (\hat{y}_{np})^{y_{np}} \quad (3.117)$$

La vraisemblance du modèle $h_{\mathbf{w}}$ au vu des données $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ est définie par l'équation suivante, en faisant apparaître la dépendance de $\hat{\mathbf{y}}$ aux poids \mathbf{w} :

$$\begin{aligned} \mathcal{L}(h_{\mathbf{w}}; S) &= \prod_{n=1}^N \mathbb{P}_{h_{\mathbf{w}}}(\mathbf{x}_n, y_n) = \prod_{n=1}^N \mathbb{P}(y_n | \mathbf{x}_n, \mathbf{w}) p_X(\mathbf{x}_n) \\ &= \prod_{n=1}^N \prod_{p=1}^P (\hat{y}_{np})^{y_{np}} p_X(\mathbf{x}_n) \end{aligned} \quad (3.118)$$

La log-vraisemblance s'obtient en passant au logarithme :

$$\ln \mathcal{L}(h_{\mathbf{w}}; S) = \sum_{n=1}^N \sum_{p=1}^P y_{np} \ln \hat{y}_{np}(\mathbf{w}) + \sum_{n=1}^N \ln p_X(\mathbf{x}_n) \quad (3.119)$$

Le terme le plus à droite restant constant quel que soit le vecteur de poids w , appliquer le principe du maximum de vraisemblance revient donc à maximiser le terme $\sum_{n=1}^N \sum_{p=1}^P y_{np} \ln \hat{y}_{np}(w)$. Cela revient encore à minimiser l'erreur suivante, en soustrayant la constante $\sum_{n=1}^N \sum_{p=1}^P y_{np} \ln y_{np}$ et en passant à l'opposé :

$$\mathcal{E}(w) = - \sum_{n=1}^N \sum_{p=1}^P y_{np} \ln \frac{\hat{y}_{np}(w)}{y_{np}} \quad (3.120)$$

On reconnaît dans l'expression de cette erreur une entropie croisée, à rapprocher de la divergence de Kullback-Leibler (voir annexe B).

Choix de la fonction γ appliquée par la couche de sortie

Il est possible de satisfaire les conditions (3.114a) et (3.114b) pour que les \hat{y}_p puissent être interprétées comme des probabilités en choisissant comme fonction γ appliquée par la couche de sortie, la fonction *softmax* suivante :

$$\gamma : \begin{pmatrix} a_1 \\ \vdots \\ a_P \end{pmatrix} \longrightarrow \begin{pmatrix} \gamma_1(a) \\ \vdots \\ \gamma_P(a) \end{pmatrix} = \begin{pmatrix} \frac{\exp\{a_1\}}{\sum_{p=1}^P \exp\{a_p\}} \\ \vdots \\ \frac{\exp\{a_P\}}{\sum_{p=1}^P \exp\{a_p\}} \end{pmatrix} \quad (3.121)$$

En faisant ce choix, toutes les composantes du vecteur $\hat{y} = \gamma(a)$ sont comprises entre 0 et 1, et la somme des composantes vaut 1.

Pour justifier un peu plus le choix de cette fonction *softmax*, Bishop [23] montre que, pour un vecteur de poids w fixé, cette fonction apparaît naturellement dans l'expression de $\mathbb{P}(\mathcal{C}_k | z_H)$, où z_H est le vecteur qui alimente la couche de sortie, dès lors que $p(z_H | \mathcal{C}_k)$ appartient à la famille exponentielle. En effet, en appliquant la règle de Bayes et sachant que $p(z_H) = \sum_{p=1}^P p(z_H, \mathcal{C}_p)$, on peut écrire :

$$\mathbb{P}(\mathcal{C}_k | z_H) = \frac{p(z_H | \mathcal{C}_k) \mathbb{P}(\mathcal{C}_k)}{p(z_H)} = \frac{p(z_H | \mathcal{C}_k) \mathbb{P}(\mathcal{C}_k)}{\sum_{p=1}^P p(z_H, \mathcal{C}_p)} = \frac{p(z_H | \mathcal{C}_k) \mathbb{P}(\mathcal{C}_k)}{\sum_{p=1}^P p(z_H | \mathcal{C}_p) \mathbb{P}(\mathcal{C}_p)} \quad (3.122)$$

Supposons que la densité $p(z_H | \mathcal{C}_k)$ appartienne à la famille des distributions exponentielles :

$$p(z_H | \mathcal{C}_k) = \alpha(z_H) \beta(\eta_k) \exp\{\eta_k^T u(z_H)\} \quad (3.123)$$

L'équation (3.122) s'exprime alors de la façon suivante :

$$\mathbb{P}(\mathcal{C}_k | z_H) = \frac{\exp\{\eta_k^T u(z_H) + \ln \beta(\eta_k) + \ln \mathbb{P}(\mathcal{C}_k)\}}{\sum_{p=1}^P \exp\{\eta_p^T u(z_H) + \ln \beta(\eta_p) + \ln \mathbb{P}(\mathcal{C}_p)\}} \quad (3.124)$$

En identifiant l'équation (3.124) avec la composante $\frac{\exp\{a_k\}}{\sum_{p=1}^P \exp\{a_p\}}$ de la fonction *softmax*, et sachant que $a_k = \sum_{i=0}^{M_H} w_{ik}^{(H)} z_i^{(H)}$, on voit que $\mathbb{P}(\mathcal{C}_k | z_H)$ est une composante de la fonction *softmax* dès lors qu'on choisit les paramètres de la manière suivante :

$$\begin{aligned} u(z_H) &= z_H \\ \eta_k &= (w_{1k}^{(H)}, \dots, w_{M_H k}^{(H)})^T \\ \ln \beta(\eta_k) + \ln \mathbb{P}(\mathcal{C}_k) &= w_{0k}^{(H)} \end{aligned} \quad (3.125)$$

3.8.4 Calcul du gradient de l'erreur par rétro-propagation

Nous venons de voir dans les deux sections précédentes comment obtenir l'expression d'une fonction d'erreur $\mathcal{E}(\mathbf{w})$ à partir d'un principe d'inférence (ici le maximum de vraisemblance), appliqué soit dans cadre d'un problème de régression, soit d'un problème de classification. Nous allons maintenant voir comment calculer le gradient $\nabla \mathcal{E}(\mathbf{w})$ de cette erreur, en l'illustrant sur les exemples des fonctions d'erreurs pour la régression et la classification que nous venons de voir dans les sections précédentes.

Dans le cas le plus général, on suppose qu'on dispose d'une fonction $\mathcal{E}(\mathbf{w})$ quelconque permettant d'évaluer l'erreur entre les sorties $\hat{y} = h_{\mathbf{w}}(\mathbf{x})$ calculées par réseau de neurones et les réponses y désirées, sur un ensemble d'exemples $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$.

L'erreur $\mathcal{E}(\mathbf{w})$ est vue ici comme une fonction du vecteur \mathbf{w} de poids à ajuster, les données S étant considérées comme fixes. On la suppose cumulative, dans le sens où l'erreur totale est la somme des erreurs commises sur chaque exemple (\mathbf{x}_n, y_n) , pour $n \in \{1, \dots, N\}$. Notons $E_n(\mathbf{w})$ l'erreur commise sur le $n^{\text{ième}}$ exemple. On a donc :

$$\mathcal{E}(\mathbf{w}) = \sum_{n=1}^N E_n(\mathbf{w}) \quad (3.126)$$

On cherche à calculer le gradient $\nabla \mathcal{E}(\mathbf{w})$ de l'erreur, c'est-à-dire le vecteur des dérivées partielles par rapport aux coordonnées w_l du vecteur de poids $\mathbf{w} = (w_1, \dots, w_L)$. Dans la description que nous avons faite du réseau de neurones, dans la section 3.8.1, les poids $w_{ij}^{(h-1)}$ sont indicés par i , le numéro de l'unité de départ dans la couche précédente $h-1$, et j , le numéro de l'unité d'arrivée dans la couche h ¹³, mais il est évident qu'on peut ranger ces poids dans un vecteur en définissant une bijection entre l'espace des indices (h, i, j) et \mathbb{N} . Dans la suite, utilisera indifféremment les notations w_l , l étant l'indice d'une connection du réseau, ou $w_{ij}^{(h)}$, selon les besoins.

Le gradient $\nabla \mathcal{E}(\mathbf{w})$ regroupe donc les dérivées partielles, rangées dans un vecteur selon l'ordre choisi pour les connections du réseau. L'erreur étant cumulative, on a :

$$\nabla \mathcal{E}(\mathbf{w}) = \begin{pmatrix} \frac{\partial \mathcal{E}}{\partial w_1} \\ \dots \\ \frac{\partial \mathcal{E}}{\partial w_L} \end{pmatrix} = \sum_{n=1}^N \nabla E_n(\mathbf{w}) = \begin{pmatrix} \sum_{n=1}^N \frac{\partial E_n}{\partial w_1} \\ \dots \\ \sum_{n=1}^N \frac{\partial E_n}{\partial w_L} \end{pmatrix} \quad (3.127)$$

13. Pour H couches cachées numérotées de 1 à H , on prend la convention que la couche 0 est la couche d'entrée, et $H+1$ celle de sortie.

Intéressons nous de plus près aux dérivées de l'erreur E_n commise sur un seul exemple $(\mathbf{x}_n, \mathbf{y}_n)$. Considérons une unité j sur la couche cachée h . La dérivée de l'erreur peut se décomposer comme suit :

$$\frac{\partial E_n}{\partial w_{ij}^{(h-1)}} = \underbrace{\frac{\partial E_n}{\partial a_j^{(h)}}}_{\delta_j^{(h)}} \underbrace{\frac{\partial a_j^{(h)}}{\partial w_{ij}^{(h-1)}}}_{z_i^{(h-1)}} \quad (3.128)$$

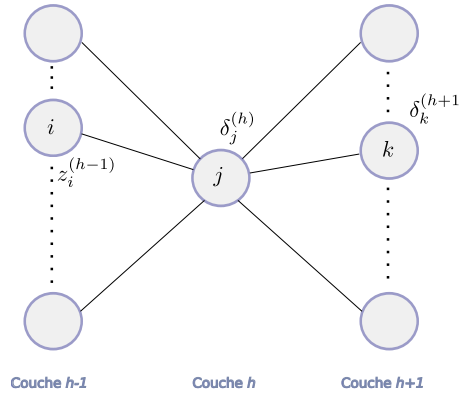


FIGURE 3.20: Quantités $z_i^{(h-1)}$ et $\delta_j^{(h)}$.

La quantité $a_j^{(h)}$ est la somme pondérée des entrées sur l'unité j de la couche h :

$$a_j^{(h)} = \sum_{i=0}^{M_{h-1}} w_{ij}^{(h-1)} z_i^{(h-1)} \quad (3.129)$$

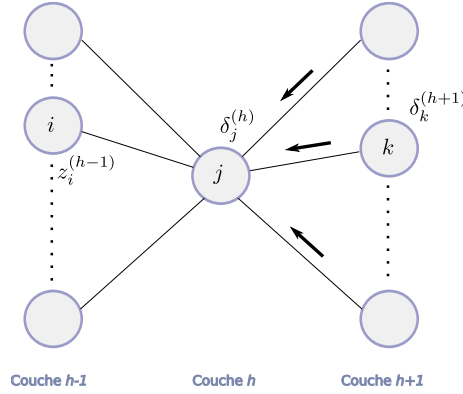
Sa dérivée par rapport à $w_{ij}^{(h-1)}$ est donc :

$$\frac{\partial a_j^{(h)}}{\partial w_{ij}^{(h-1)}} = z_i^{(h-1)} \quad (3.130)$$

En notant $\delta_j^{(h)} = \frac{\partial E_n}{\partial a_j^{(h)}}$, la dérivée partielle de l'erreur E_n par rapport à $w_{ij}^{(h-1)}$ s'écrit donc simplement comme suit :

$$\frac{\partial E_n}{\partial w_{ij}^{(h-1)}} = \delta_j^{(h)} z_i^{(h-1)} \quad (3.131)$$

Les quantités $z_i^{(h-1)}$ sont calculées en parcourant le réseau dans son sens naturel (“vers l’avant”), lors du calcul de $\hat{\mathbf{y}}_n = h_w(\mathbf{x}_n)$. Il reste donc à obtenir les quantités $\delta_j^{(h)}$, de la manière suivante.

FIGURE 3.21: Rétropropagation des δ .

Calcul des δ_j pour une couche cachée

La grande idée de la rétro-propagation consiste à calculer les $\delta_j^{(h)}$ en parcourant le réseau dans l'autre sens ("vers l'arrière") comme illustré sur la Figure 3.21, en remarquant que la quantité $\delta_j^{(h)}$ associée à l'unité j de la couche cachée h dépend des $\delta_k^{(h+1)}$, pour $k \in \{1, \dots, M_{h+1}\}$, de la couche suivante $h + 1$.

$$\delta_j^{(h)} = \frac{\partial E_n}{\partial a_j^{(h)}} = \sum_{k=1}^{M_{h+1}} \underbrace{\frac{\partial E_n}{\partial a_k^{(h+1)}}}_{\delta_k^{(h+1)}} \frac{\partial a_k^{(h+1)}}{\partial a_j^{(h)}} \quad (3.132)$$

Le terme $\frac{\partial a_k^{(h+1)}}{\partial a_j^{(h)}}$ s'obtient en considérant l'expression des sommes pondérées $a_k^{(h+1)}$:

$$a_k^{(h+1)} = \sum_{j=0}^{M_h} w_{jk}^{(h)} z_j^{(h)} = \sum_{j=0}^{M_h} w_{jk}^{(h)} \varphi(a_j^{(h)}) \quad (3.133)$$

On en déduit la dérivée partielle :

$$\frac{\partial a_k^{(h+1)}}{\partial a_j^{(h)}} = w_{jk}^{(h)} \varphi'(a_j^{(h)}) \quad (3.134)$$

L'expression finale des $\delta_j^{(h)}$ pour les couches cachées est donc donnée par l'équation (3.135) suivante :

$$\delta_j^{(h)} = \varphi'(a_j^{(h)}) \sum_{k=1}^{M_{h+1}} w_{jk}^{(h)} \delta_k^{(h+1)} \quad (3.135)$$

Calcul des δ_p pour la couche de sortie

Pour la couche de sortie, les quantités $\delta_p = \frac{\partial E_n}{\partial a_p}$ se calculent directement, pour $p \in \{1, \dots, P\}$, en considérant l'expression du vecteur de sortie, qui est ici \hat{y}_n et qui dépend fonctionnellement à travers γ du vecteur $\mathbf{a} = (a_1, \dots, a_p, \dots, a_P)^T$ (rappelons que les a_p sont les sommes pondérées données par l'équation (3.96)).

$$\hat{y}_n = \begin{pmatrix} \hat{y}_{n1} \\ \vdots \\ \hat{y}_{nP} \end{pmatrix} = \gamma(\mathbf{a}) = \begin{pmatrix} \gamma_1(\mathbf{a}) \\ \vdots \\ \gamma_P(\mathbf{a}) \end{pmatrix} = \begin{pmatrix} \gamma_1(a_1, \dots, a_P) \\ \vdots \\ \gamma_P(a_1, \dots, a_P) \end{pmatrix} \quad (3.136)$$

La $k^{\text{ième}}$ coordonnée du vecteur de sortie \hat{y}_n calculé par le réseau de neurones en réponse au vecteur d'entrée \mathbf{x}_n , est notée \hat{y}_{nk} . Tout choix raisonnable de la fonction d'erreur E_n fera intervenir l'écart, mesuré par une fonction quelconque, entre la composante \hat{y}_{nk} calculée par le réseau et la valeur observée y_{nk} , pour $k \in \{1, \dots, P\}$. Cette dépendance peut s'exprimer par l'équation suivante, où E_n est ici vue comme fonction de $\hat{y} = h_w(\mathbf{x})$, pour une valeur donnée de w :

$$E_n = E_n(\hat{y}) = E_n(\hat{y}_{n1}, \dots, \hat{y}_{nk}, \dots, \hat{y}_{nP}) \quad (3.137)$$

Deux exemples classiques sont l'erreur quadratique $E_n = \frac{1}{2} \sum_{p=1}^P (\hat{y}_{np} - y_{np})^2$ et l'entropie croisée $E_n = - \sum_{p=1}^P y_{np} \ln \left(\frac{\hat{y}_{np}}{y_{np}} \right)$, que nous détaillerons un peu plus loin.

Le calcul de $\delta_p = \frac{\partial E_n}{\partial a_p}$ pour une unité p de la couche de sortie doit donc tenir compte de la dépendance de E_n aux composantes $\hat{y}_{nk} = \gamma_k(a_1, \dots, a_P)$, pour $k \in \{1, \dots, P\}$, qui dépendent chacune de a_p , dans le cas général :

$$\begin{aligned} \delta_p = \frac{\partial E_n}{\partial a_p} &= \sum_{k=1}^P \frac{\partial E_n}{\partial \hat{y}_{nk}} \frac{\partial \hat{y}_{nk}}{\partial a_p} \quad \text{dans le cas général où } \hat{y}_{nk} = \gamma_k(a_1, \dots, a_P) \\ &= \sum_{k=1}^P \frac{\partial E_n}{\partial \hat{y}_{nk}} \frac{\partial \gamma_k(a_1, \dots, a_p, \dots, a_P)}{\partial a_p} \end{aligned} \quad (3.138)$$

Dans le cas particulier où chaque composante \hat{y}_{nk} ne dépend que la composante a_k du vecteur \mathbf{a} (comme par exemple quand γ est la fonction identité), δ_p s'exprime plus simplement de la façon suivante :

$$\begin{aligned} \delta_p = \frac{\partial E_n}{\partial a_p} &= \frac{\partial E_n}{\partial \hat{y}_{np}} \frac{\partial \hat{y}_{np}}{\partial a_p} \quad \text{si } \hat{y}_{nk} = \gamma_k(a_k) \text{ ne dépend que de } a_k, \forall k \in \{1, \dots, P\} \\ &= \frac{\partial E_n}{\partial \hat{y}_{np}} \frac{\partial \gamma_p(a_p)}{\partial a_p} \end{aligned} \quad (3.139)$$

Le calcul des dérivées partielles $\frac{\partial E_n}{\partial \hat{y}_{nk}}$ dépend directement de l'expression choisie pour la fonction d'erreur (voir les exemples un peu plus loin). La dérivée partielle $\frac{\partial \hat{y}_{np}}{\partial a_p}$ se déduit quant à elle directement de l'expression choisie pour la fonction vectorielle γ appliquée au vecteur $\mathbf{a} = (a_1, \dots, a_P)^T$ des sommes pondérées calculées par la couche de sortie.

Exemple où E_n est quadratique et $\gamma = \text{identité}$

Supposons par exemple que E_n soit donnée par la fonction suivante :

$$E_n(\mathbf{w}) = \frac{1}{2} \sum_{p=1}^P (\hat{y}_{np} - y_{np})^2 \quad (3.140)$$

L'expression de sa dérivée partielle par rapport à \hat{y}_{np} est alors :

$$\frac{\partial E_n}{\partial \hat{y}_{np}} = \hat{y}_{np} - y_{np} \quad (3.141)$$

Prenons pour γ la fonction identité. Chaque fonction composante γ_p , ne dépend alors que de la composante a_p du vecteur \mathbf{a} , $\forall p \in \{1, \dots, P\}$:

$$\gamma_p(\mathbf{a}) = \gamma_p(a_1, \dots, a_P) = \gamma_p(a_p) = a_p \quad (3.142)$$

La dérivée γ_p par rapport à a_p est égale à 1, et on en déduit la valeur de δ_p , d'après l'équation (3.139) :

$$\delta_p = \hat{y}_{np} - y_{np} \quad \text{pour la couche de sortie} \quad (3.143)$$

Exemple où E_n est basée sur l'entropie croisée et où $\gamma = \text{softmax}$

Supposons qu'on ait pris la fonction suivante comme fonction d'erreur :

$$E_n(\mathbf{w}) = - \sum_{p=1}^P y_{np} \ln \left(\frac{\hat{y}_{np}}{y_{np}} \right) \quad (3.144)$$

Les dérivées partielles s'obtiennent par l'expression suivante :

$$\frac{\partial E_n}{\partial \hat{y}_{nk}} = - \frac{y_{nk}}{\hat{y}_{nk}} \quad \forall k \in \{1, \dots, P\} \quad (3.145)$$

Dans le cas où γ est la fonction *softmax*, les fonctions composantes γ_k dépendent chacune de toutes les composantes a_1, \dots, a_P du vecteur \mathbf{a} :

$$\gamma_k(\mathbf{a}) = \gamma_k(a_1, \dots, a_P) = \frac{\exp\{a_p\}}{\sum_{i=1}^P \exp\{a_i\}} \quad (3.146)$$

Les dérivées partielles s'expriment comme suit, avec $\delta_{kp} = 1$ si $k = p$ et 0 sinon, et sachant que $\hat{y}_{nk} = \gamma_k(\mathbf{a})$, pour tout $k \in \{1, \dots, P\}$:

$$\frac{\partial \gamma_k(\mathbf{a})}{\partial a_p} = \delta_{kp} \gamma_p(\mathbf{a}) - \gamma_k(\mathbf{a}) \gamma_p(\mathbf{a}) = \delta_{kp} \hat{y}_{np} - \hat{y}_{nk} \hat{y}_{np} \quad (3.147)$$

En utilisant l'équation (3.138), et sachant qu'on a $\sum_{k=1}^P y_{nk} = 1$, on en déduit l'expression des δ_p pour la couche de sortie :

$$\begin{aligned}
\delta_p &= \frac{\partial E_n}{\partial a_p} = \sum_{k=1}^P \frac{\partial E_n}{\partial \hat{y}_{nk}} \frac{\partial \gamma_k(\mathbf{a})}{\partial a_p} = \frac{\partial E_n}{\partial \hat{y}_{np}} \frac{\partial \gamma_p(\mathbf{a})}{\partial a_p} + \sum_{k \neq p} \frac{\partial E_n}{\partial \hat{y}_{nk}} \frac{\partial \gamma_k(\mathbf{a})}{\partial a_p} \\
&= \left(-\frac{y_{np}}{\hat{y}_{np}} \right) \hat{y}_{np} (1 - \hat{y}_{np}) + \sum_{k \neq p} \left(-\frac{y_{nk}}{\hat{y}_{nk}} \right) (-\hat{y}_{nk} \hat{y}_{np}) \\
&= -y_{np} (1 - \hat{y}_{np}) + \hat{y}_{np} \sum_{k \neq p} y_{nk} = -y_{np} (1 - \hat{y}_{np}) + \hat{y}_{np} (1 - y_{np}) \\
&= \hat{y}_{np} - y_{np}
\end{aligned} \tag{3.148}$$

Résumé du calcul du gradient

Le calcul du gradient $\nabla E_n(\mathbf{w})$ de l'erreur E_n pour un exemple (\mathbf{x}_n, y_n) peut se résumer comme suit :

1. Une passe de calcul "vers l'avant" pour calculer :
 - les vecteurs $\mathbf{a}^{(h)}$ et $\mathbf{z}^{(h)}$ des couches cachées en utilisant les équations (3.94), et (3.95),
 - les vecteurs \mathbf{a} et \mathbf{y} de la couche de sortie, avec les équations (3.96).
2. Une passe de calcul "vers l'arrière" pour obtenir :
 - les $\delta_p = \frac{\partial E_n}{\partial a_p}$ de la couche de sortie, en appliquant, selon la fonction γ de la couche de sortie, l'équation (3.138) ou (3.139) à la fonction d'erreur $\mathcal{E}(\mathbf{w})$ choisie.
 - les $\delta_j^{(h)} = \varphi'(a_j^{(h)}) \sum_{k=1}^{M_{h+1}} w_{jk}^{(h)} \delta_k^{(h+1)}$ (3.135) des couches cachées.
3. Calcul du gradient $\frac{\partial E_n}{\partial w_{ij}^{(h-1)}} = \delta_j^{(h)} z_i^{(h-1)}$ (3.131)

Le gradient de l'erreur totale $\mathcal{E}(\mathbf{w}) = \sum_{n=1}^N E_n(\mathbf{w})$ pour les N exemples $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ est ensuite calculé en faisant la somme $\nabla \mathcal{E}(\mathbf{w}) = \sum_{n=1}^N \nabla E_n(\mathbf{w})$ (3.127).

3.8.5 Apprentissage du réseau

L'apprentissage du réseau de neurones consiste à ajuster les poids \mathbf{w} de façon à minimiser l'erreur $\mathcal{E}(\mathbf{w})$ évaluée au point \mathbf{w} sur les exemples $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$. On recherche donc un vecteur \mathbf{w}^* tel que :

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \Theta} \mathcal{E}(\mathbf{w}) \tag{3.149}$$

L'optimum global vérifiant (3.149) (ou les optima, s'ils sont multiples) n'est pas accessible au calcul analytique. On fait donc appel à un calcul itératif permettant d'approcher au mieux \mathbf{w}^* . On peut distinguer les méthodes d'optimisation utilisées pour ce calcul selon qu'elles utilisent ou non le gradient $\nabla \mathcal{E}(\mathbf{w})$, celui-ci pouvant être calculé efficacement par la méthode de rétro-propagation présentée dans la section 3.8.4 précédente.

De nombreuses présentations, encore aujourd’hui, regroupent sous un même terme de “rétro-propagation” à la fois le calcul du gradient et la mise à jour des poids par la formule (3.150) correspondant à la méthode de plus grande pente initialement utilisée dans les travaux de Rumelhart *et al.* [139]. Dans ce document, les deux notions (calcul du gradient, et mise à jour des poids) sont distinguées l’une de l’autre et le terme de rétro-propagation fait uniquement référence au calcul du gradient, indépendamment de la méthode d’optimisation qui utilise ou non le gradient calculé par rétro-propagation.

Méthodes de descente de gradient

La méthode la plus basique, originellement associée avec la rétro-propagation du gradient, est la méthode de plus grande pente à pas fixe, où la mise à jour des poids à chaque itération est donnée par l’équation (3.150), η étant la taille du pas effectué dans la direction de descente.

Nous avons vu, dans la section 2.4 du chapitre 2, que le choix de la plus grande pente comme direction de descente peut conduire à un nombre d’itérations important. La convergence vers un optimum local peut de plus être très lente. Dans le cas d’un pas fixe η , elle peut même être remise en cause, pour peu que le taux d’apprentissage η soit mal choisi.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla \mathcal{E}(\mathbf{w}) \quad (3.150)$$

Une tentative d’amélioration de la méthode de plus grande pente consiste à corriger la mise à jour des poids en ajoutant un terme de “*momentum*”, proportionnel à la valeur du gradient de l’itération précédente, soit encore à la différence entre le vecteur \mathbf{w}_t l’itération courante et celui \mathbf{w}_{t-1} de la précédente. L’équation (3.151) donne la mise à jour avec *momentum* :

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla \mathcal{E}(\mathbf{w}) + \mu (\mathbf{w}_t - \mathbf{w}_{t-1}) \quad (3.151)$$

Méthodes de Quasi-Newton (avec utilisation du gradient)

D’autres méthodes s’appuient sur une approximation quadratique de la fonction d’erreur, obtenue par un développement de Taylor à l’ordre 2 au voisinage du point \mathbf{w}^* correspondant à l’optimum, et une approximation itérative de la hessienne ou de son inverse. Ces méthodes de Quasi-Newton sont brièvement présentées à la section 2.4 du chapitre 2, et abordées en détail dans l’annexe A qui présente les méthodes BFGS et L-BFGS. Nous n’y reviendrons donc pas ici.

L’étude de Webb *et al.* [173] comparant différentes méthodes utilisant le gradient conclut que la simple descente de gradient, avec ou sans momentum, n’est pas recommandée car c’est la plus inefficace. Parmi les méthodes testées, celles de Quasi-Newton, et en particulier BFGS, semblent être les méthodes de choix pour l’apprentissage des réseaux de neurones, avec toutefois un certain coût de stockage qui peut faire préférer des approches plus économes, comme les gradients conjugués, pour les problèmes de grande dimension.

Apprentissage par métaheuristiques, sans utilisation du gradient

La motivation initiale pour utiliser des méthodes avec gradient pour l’apprentissage des réseaux de neurones tient à des raisons historiques. Elles ont été développées et étudiées avant les autres. Elles garantissent par ailleurs l’optimalité de la solution trouvée lorsque

la fonction-objectif est convexe et suffisamment régulière, et que la méthode de *linesearch* satisfait certaines conditions de décroissance à chaque itération (voir chap. 2, section 2.4).

Cependant, on peut remarquer que la fonction-objectif a ici peu de chances d'être convexe, et présente très certainement de nombreux optima locaux au vu des symétries possibles dans les poids du réseaux. Les méthodes itératives utilisant le gradient vont donc mener à une solution dont l'optimalité globale n'est pas garantie, et qui peut être plus ou moins bonne selon le point initial choisi au hasard.

De nombreuses approches alternatives aux méthodes avec gradient, et notamment des approches métaheuristiques, ont été proposées dans la littérature pour optimiser la structure ou les valeurs des poids d'un réseau de neurones : algorithmes génétiques [108], essaims particuliers [80], colonies de fourmis [27], évolution différentielle [152], etc.

Ces métaheuristiques n'utilisent pas le gradient de l'erreur pour chercher un optimum. Elles ne garantissent pas non plus l'optimalité globale de la solution. Elles s'appuient généralement sur une exploration aléatoire de l'espace de recherche, biaisée par une heuristique qui permet de guider la recherche vers des solutions de bonne qualité (voir chap. 2). Elles présentent l'avantage d'une meilleure exploration de l'espace des poids, puisqu'elles considèrent une population au lieu d'un seul vecteur w et qu'elles s'appuient sur des mécanismes de diversification de cette population, en plus de mécanismes d'intensification orientant vers de bonnes solutions. L'inconvénient de ces méthodes est qu'elles sont plus coûteuses en temps de calcul que des méthodes comme BFGS ou L-BFGS, lesquelles trouvent souvent des solutions de très bonne qualité en un temps très court.

3.8.6 Considérations pratiques sur les réseaux de neurones

Les méthodes que nous venons d'évoquer ne garantissent pas de trouver un vecteur de poids optimal $w^* = \arg \min_{w \in \Theta} \mathcal{E}(w)$. Contrairement aux modèles linéaires pour lesquels on obtient une solution analytique unique, le résultat de l'apprentissage du réseau de neurones n'est qu'une solution approchée, qu'on espère de bonne qualité. Cette solution a un caractère aléatoire, même pour les méthodes itératives déterministes de descente de gradient ou de Quasi-Newton, puisque la valeur initiale de w est tirée au hasard.

Le résultat peut donc être influencé par un grand nombre de facteurs comme la performance intrinsèque de la méthode d'optimisation utilisée, la qualité de son implémentation, le choix du critère d'arrêt avec ou sans *early stopping*, la façon de choisir les valeurs initiales des poids, etc. Le pré-processing et la normalisation des entrées jouent également un rôle crucial dans les performances des réseaux de neurones. l'objectif de cette section 3.8 n'est pas de faire une présentation exhaustive des réseaux de neurones, laquelle peut se trouver dans des ouvrages de référence comme [23, 137, 24], et nous ne détaillerons donc pas ici ces points pourtant essentiels sur le plan pratique.

Signalons, pour terminer, une des limitations des réseaux de neurones classiques, tels que nous les avons présentés dans cette section. Le calcul du gradient par rétro-propagation se heurte à une difficulté lorsque le nombre de couches cachées devient important : l'erreur rétro-propagée diminue exponentiellement avec le nombre de couches [89]. Ce problème, connu sous le nom de *vanishing gradient problem*, rend l'apprentissage de réseaux profonds exagérément lent, voire impossible. Il apparaît également dans l'apprentissage des réseaux récurrents que l'on peut développer en réseaux de type *feed-forward* comportant un nombre important de couches cachées. Les réseaux connexionnistes ont récemment connu un regain d'intérêt avec le développement de techniques de *deep learning* [21], où les couches cachées sont d'abord

apprises de façon non supervisée, sans chercher à calculer un gradient d’erreur pour l’ensemble du modèle.

3.9 Gradient Boosting Machines

Cette section est une courte introduction aux techniques de *boosting* (stimulation, “do-ping”), qui permettent de construire incrémentalement un modèle de prédiction performant à partir de modèles beaucoup plus simples et moins performants (*weak learner*). La sous-section 3.9.1 décrit le principe général de l’ajustement incrémental d’un modèle additif, auquel peuvent se ramener les techniques de *boosting*. Le principe général du *gradient boosting*, présenté dans la sous-section 3.9.2, est une amélioration du *boosting* qui utilise la direction de plus grande pente (l’opposé du gradient de la fonction de perte). Finalement, la sous-section 3.9.3 présente l’application du *gradient boosting* aux arbres de régression et de classification.

3.9.1 Ajustement incrémental d’un modèle additif

Dans sa forme la plus générale, décrite dans [86] chap. 10, le *boosting* peut être vue comme une forme d’ajustement incrémental d’un modèle additif de la forme suivante, où les fonctions de base $\Phi_m = \Phi(\cdot, \gamma_m)$ comportent des paramètres ajustables γ_m :

$$h_M(\mathbf{x}) = \sum_{m=1}^M \beta_m \Phi(\mathbf{x}, \gamma_m) \quad (3.152)$$

L’algorithme 11 décrit le principe de cet ajustement incrémental, auquel peuvent se ramener les techniques de *boosting*.

Algorithm 11 Ajustement incrémental d’un modèle additif.

- 1: Initialiser $h_0 : \mathbf{x} \rightarrow 0$
- 2: **for** $m = 1$ to M **do**
- 3: Calculer les paramètres (β_m, γ_m) :

$$(\beta_m, \gamma_m) = \arg \min_{\beta, \gamma} \sum_{n=1}^N \ell(y_n, h_{m-1}(x_n) + \beta \Phi(x_n, \gamma))$$

- 4: Mettre à jour l’itération courante du modèle :

$$h_m : \mathbf{x} \rightarrow h_{m-1}(\mathbf{x}) + \beta_m \Phi(\mathbf{x}, \gamma_m)$$

- 5: **end for**
- 6: Renvoyer le modèle final de prédiction :

$$h_M : \mathbf{x} \rightarrow \sum_{m=1}^M \beta_m \Phi(\mathbf{x}, \gamma_m)$$

Par exemple, la méthode AdaBoost est équivalente à l’algorithme 11 avec une perte exponentielle $\ell(y, h(\mathbf{x})) = \exp\{-yh(\mathbf{x})\}$, et avec des classifieurs binaires comme fonctions de base (voir [86]).

3.9.2 Principe général du *gradient boosting*

Le *gradient boosting* [59, 60] est une amélioration de l'algorithme 11, dont l'idée est de choisir la fonction de base $\gamma \rightarrow \Phi(x, \gamma)$ la plus proche possible, en chaque point x_n de l'ensemble d'exemples, de la direction de plus grande pente $-\frac{\partial \ell(y, v)}{\partial v} \Big|_{v=h_{m-1}(x_n)}$. Une fois choisie cette direction de descente, le paramètre β est choisi par une méthode de recherche unidirectionnelle (*linesearch*) dans cette direction.

L'algorithme 12 décrit le principe général du *gradient boosting*, pour des fonctions de base Φ_m quelconques, dépendant de paramètres γ_m .

Algorithm 12 Algorithme de *gradient boosting*.

- 1: Initialiser $h_0 : x \rightarrow \arg \min_v \sum_{n=1}^N \ell(y_n, v)$
- 2: **for** $m = 1$ to M **do**
- 3: Evaluer l'opposé du gradient $-\frac{\partial \ell(y, v)}{\partial v} \Big|_{v=h_{m-1}(x)}$ en chaque point x_n :

$$g_n = -\frac{\partial \ell(y, v)}{\partial v} \Big|_{v=h_{m-1}(x_n)} \quad , \quad n = 1, \dots, N$$

- 4: Calculer γ_m donnant la direction de descente la plus proche de la plus grande pente :

$$\gamma_m = \arg \min_{\beta, \gamma} \sum_{n=1}^N [g_n - \beta \Phi(x, \gamma)]^2$$

- 5: Chercher le pas optimal β_m dans la direction de descente :

$$\beta_m = \arg \min_{\beta} \sum_{n=1}^N \ell(y_n, h_{m-1}(x_n) + \beta \Phi(x_n, \gamma_m))$$

- 6: Mettre à jour l'itération courante du modèle :

$$h_m : x \rightarrow h_{m-1}(x) + \beta_m \Phi(x, \gamma_m)$$

- 7: **end for**

- 8: Renvoyer le modèle final de prédiction :

$$h_M : x \rightarrow h_0(x) + \sum_{m=1}^M \beta_m \Phi(x, \gamma_m)$$

3.9.3 *Gradient tree boosting*

Dans [59, 60], Friedman applique le principe du *gradient boosting* à des arbres de régression ou de classification.

Les arbres de régression ou de classification

Les arbres de régression ou de classification [29] partitionnent l'espace \mathcal{X} des variables explicatives en régions \mathcal{R}_j , $j \in \{1, \dots, J\}$, et affectent à chaque région R_j une fonction spécifique c_j , où $\hat{y} = c_j(x)$ est la réponse donnée par le modèle pour tout $x \in R_j$.

Le partitionnement en régions R_j peut être représenté par un arbre où chaque nœud non terminal mémorise une coupure de l'espace \mathcal{X} , le plus souvent selon une des P dimensions de $\mathbf{x} = (x_1, \dots, x_P)^T$. Dans l'implantation la plus courante de ce type de méthode, un nœud est caractérisé par une seule variable x_p et une valeur de coupure s , le sous-arbre de gauche correspondant aux valeurs de $x_p \leq s$ et celui de droite à $x_p > s$. Les feuilles de l'arbre sont les régions R_j , auxquelles sont affectées les fonctions c_j , souvent prises à valeur constante.

Pour un arbre T donné, la réponse du modèle de prédiction h_T à une entrée \mathbf{x} quelconque s'exprime alors par l'équation (3.153), où $\mathbb{1}_{R_j}$ est la fonction indicatrice d'appartenance de \mathbf{x} à R_j , et où $|T|$ est le nombre de feuilles (*i.e.* de régions).

$$h_T(\mathbf{x}) = \sum_{j=1}^{|T|} c_j(\mathbf{x}) \mathbb{1}_{R_j}(\mathbf{x}) \quad (3.153)$$

L'expression de c_j pour chaque région R_j dépend de la tâche d'apprentissage et de la fonction de perte que l'on cherche à minimiser. Typiquement, dans le cas d'une régression avec une perte quadratique, on prendra $c_j(\mathbf{x}) = \mathbb{E}(y_n | \mathbf{x}_n \in R_j) = \frac{1}{N_j} \sum_{\mathbf{x}_n \in R_j} y_n$, constante sur la région R_j . Dans le cas de la classification, on prendra la classe la plus représentée dans la région R_j .

Trouver une partition optimale minimisant un critère $\sum_{n=1}^N \ell(y_n, h_T(\mathbf{x}_n))$ est un problème fortement combinatoire, qu'on contourne en utilisant une heuristique gloutonne. Le principe consiste à développer l'arbre en choisissant à chaque itération une variable x_p et une valeur de coupure s , de la façon suivante.

Pour une région R_j correspondant à une feuille de l'arbre, on définit les régions $R_{j1}(p, s)$ et $R_{j2}(p, s)$ par :

$$\begin{aligned} R_{j1}(p, s) &= \{\mathbf{x} = (x_1, \dots, x_P) \in R_j \mid x_p \leq s\} \\ R_{j2}(p, s) &= \{\mathbf{x} = (x_1, \dots, x_P) \in R_j \mid x_p > s\} \end{aligned} \quad (3.154)$$

Il est relativement facile d'examiner le coût d'une coupure selon l'une des variables x_p , en choisissant un critère de coût comme le suivant :

$$\text{cost}(p, s) = \sum_{\mathbf{x}_n \in R_{j1}(p, s)} \ell(y_n, c_{j1}(\mathbf{x})) + \sum_{\mathbf{x}_n \in R_{j2}(p, s)} \ell(y_n, c_{j2}(\mathbf{x})) \quad (3.155)$$

Pour une feuille correspondant à une région R_j , l'obtention d'un couple (x_p, s) minimisant le coût $\text{cost}(p, s)$ de l'équation (3.155) nécessite donc simplement de parcourir la liste des variables x_1, \dots, x_P , de calculer pour chacune la valeur optimale de coupure, puis de choisir la dimension p donnant la meilleure coupure.

La question se pose toutefois de savoir quelles régions R_j faut-il développer selon ce principe? Une stratégie trop gloutonne, où l'on ne développerait que les feuilles améliorant significativement le critère de perte, n'est pas judicieuse. En effet, une coupure apparemment sans intérêt immédiat, où le critère n'est que peu amélioré suite à cette première coupure, peut parfaitement mener à un gain significatif par des coupures supplémentaires. À l'inverse, développer toutes les feuilles pour produire un arbre très gros mène clairement à un modèle de prédiction trop complexe, qui risque de sur-apprendre les exemples.

Une stratégie classique consiste à procéder en deux étapes successives. La première est une expansion *top-down* à partir d'un arbre initial comprenant une seule feuille représentant tout

le domaine \mathcal{X} . Toutes les feuilles sont développées jusqu'à obtenir un arbre T_0 d'une certaine taille J , suffisamment importante pour que h_{T_0} soit sur-ajusté aux données. La deuxième étape est un élagage (*pruning*) de l'arbre T_0 , dont l'objectif est d'obtenir un arbre présentant un bon compromis entre le critère de perte et la taille de l'arbre.

Le fait de passer par une première étape d'expansion permet de calculer un score pour chaque nœud de l'arbre développé T_0 . Ce score peut être par exemple le coût d'une re-substitution d'un nœud non terminal par une feuille. Dans la méthode du *weakest link pruning* le coût de re-substitution d'un nœud u par une feuille f_u est donné par l'équation (3.156), où E est une fonction permettant d'évaluer n'importe quel sous-arbre, et où T_u est le sous-arbre de racine u :

$$g(u) = \frac{E(f_u) - E(T_u)}{|T_u| - 1} \quad (3.156)$$

Pour mieux comprendre le choix de la fonction de score de l'équation (3.156), considérons la fonction C_α suivante :

$$C_\alpha(T) = E(T) + \alpha|T| \quad (3.157)$$

Cette fonction C_α exprime un compromis entre le coût total de l'arbre $E(T)$ et sa complexité, ici évaluée par son nombre de feuilles $|T|$. Ce compromis est fixé par l'hyperparamètre α . On peut par ailleurs faire le lien entre la fonction de compromis coût-complexité C_α et le score g , et justifier au passage le choix de l'expression (3.156) pour g , en remarquant les équivalences suivantes :

$$\begin{aligned} g(u) \leq \alpha &\Leftrightarrow C_\alpha(f_u) \leq C_\alpha(T_u) \\ g(u) > \alpha &\Leftrightarrow C_\alpha(f_u) > C_\alpha(T_u) \end{aligned} \quad (3.158)$$

On montre que pour α fixé, on obtient un arbre T_α optimal pour le critère C_α en parcourant récursivement T_0 à partir des feuilles et en remontant jusqu'à la racine, en remplaçant les nœuds pour lesquels $g(u) \leq \alpha$ par des feuilles. Les démonstrations sont détaillées dans [137] p. 223.

La fonction E peut typiquement être choisie égale au critère de perte ayant servi durant la phase d'expansion de l'arbre : une feuille R_j aura pour coût $E(\{R_j\}) = \sum_{x_n \in R_j} \ell(y_n, c_j(x_n))$ et le coût d'un arbre sera $E(T) = \sum_{R_j \in T} E(\{R_j\})$. Cette fonction de coût E s'exprime alors par l'équation suivante, où l'on reconnaît le critère empirique de perte qu'on cherche à minimiser en construisant la partition de \mathcal{X} en régions :

$$E(T) = \sum_{j=1}^{|T|} \sum_{x_n \in R_j} \ell(y_n, c_j(x_n)) \quad (3.159)$$

Avec ce choix pour E , choisir $\alpha = 0$ revient à sélectionner l'arbre T_0 de risque empirique minimal, c'est à dire le modèle sur-ajusté. A l'inverse, fixer une valeur élevée pour α conduit à sélectionner des arbres de taille réduite, c'est-à-dire des modèles simples approximant mal le vrai phénomène ayant produit les données observées. Le choix de l'hyperparamètre α détermine donc un compromis biais-variance, et il a un rôle analogue au paramètre λ de la régularisation (voir section 3.3.3). Le choix de la valeur de l'hyperparamètre α peut se faire par exemple par validation croisée (voir section 3.5.1).

Gradient boosting appliqué aux arbres de régression ou de classification

Le *Gradient tree boosting* consiste à choisir des arbres de régression ou de classification h_T , dont l'expression est donnée par l'équation (3.153), comme fonctions de base Φ dans l'algorithme 12. Les fonctions $c_j(x)$ de chaque région R_j sont choisies constantes $c_j(x) = c_j$.

L'algorithme du *gradient tree boosting* prend une forme légèrement modifiée en remarquant que la mise à jour du modèle (ligne 6 de l'algorithme 12) s'écrit maintenant de la façon suivante :

$$h_m(x) = h_{m-1}(x) + \beta_m h_T(x) = h_{m-1}(x) + \sum_{j=1}^{J_m} \underbrace{\beta_m c_j}_{\gamma_{jm}} \mathbb{1}_{R_j}(x) \quad (3.160)$$

L'algorithme 12 est donc modifié en remplaçant l'ajustement incrémental du seul terme $\beta_m h_T(x)$, par l'ajustement des J_m termes $\gamma_{jm} \mathbb{1}_{R_j}(x)$.

Algorithm 13 Algorithme de *gradient tree boosting*.

-
- 1: Initialiser $h_0 : x \rightarrow \arg \min_v \sum_{n=1}^N \ell(y_n, v)$
 - 2: **for** $m = 1$ to M **do**
 - 3: Evaluer l'opposé du gradient $-\frac{\partial \ell(y, v)}{\partial v} \Big|_{v=h_{m-1}(x)}$ en chaque point x_n :

$$g_n = -\frac{\partial \ell(y, v)}{\partial v} \Big|_{v=h_{m-1}(x_n)} \quad , \quad n = 1, \dots, N$$

- 4: Ajuster un arbre de régression ou de classification de la forme $h_T(x) = \sum_{j=1}^{J_m} a_j \mathbb{1}_{R_j}(x)$, comprenant J_m feuilles, sur les g_n en minimisant la somme des carrés des écarts :

$$h_{T_m} = \arg \min_{\{h_T / |T|=J_m\}} \sum_{n=1}^N \left[g_n - \sum_{j=1}^{J_m} a_j \mathbb{1}_{R_j}(x) \right]^2$$

- 5: Calculer, pour chaque région R_j de l'arbre T_m , le γ_{mj} optimal pour la perte ℓ :

$$\gamma_{mj} = \arg \min_{\gamma} \sum_{x_n \in R_j} \ell(y_n, h_{m-1}(x) + \gamma) \quad , \quad j = 1, \dots, J_m$$

- 6: Mettre à jour l'itération courante du modèle :

$$h_m : x \rightarrow h_{m-1}(x) + \sum_{R_j \in T_m} \gamma_{mj} \mathbb{1}_{R_j}(x)$$

- 7: **end for**

- 8: Renvoyer le modèle final de prédiction :

$$h_M : x \rightarrow h_0(x) + \sum_{m=1}^M \sum_{R_j \in T_m} \gamma_{mj} \mathbb{1}_{R_j}(x)$$

Friedman introduit plusieurs améliorations à son algorithme de base 13, avec un paramètre de “*shrinkage*” pour le terme additif, et un re-tirage aléatoire d'un échantillon de données à chaque itération (*stochastic gradient boosting*). Le “*shrinkage*” consiste à remplacer la mise à

jour du modèle (ligne 6 de l'algorithme 13) par la mise à jour suivante, où ν est le paramètre de “*shrinkage*” ($0 < \nu \leq 1$) :

$$h_m : \mathbf{x} \rightarrow h_{m-1}(\mathbf{x}) + \nu \sum_{R_j \in T_m} \gamma_{mj} \mathbb{1}_{R_j}(\mathbf{x}) \quad (3.161)$$

Dans [86], chap. 10, il est recommandé de choisir ν petit ($\nu < 0.1$), de prendre la taille des arbres $J_m = J$ constante avec une valeur à choisir en fonction du problème (usuellement $4 \leq J \leq 8$), et de choisir le nombre d'itérations M du boosting par *early stopping*¹⁴. La technique en deux étapes, avec une expansion jusqu'à une taille J grande suivie d'un élagage, utilisée pour les arbres de régression ou de classification standards est ici inutile, dans le cas des arbres boostés. La complexité du modèle est contrôlée par le nombre d'itérations et le paramètre de “*shrinkage*”, pour une taille J donnée. Les hyperparamètres J , ν , et M peuvent également être sélectionnés par validation croisée (voir section 3.5.1).

3.10 Conclusion

Ce chapitre était dédié d'une part à la présentation des notions, des principes, et du cadre méthodologique de l'apprentissage, et d'autre part à la description des quelques méthodes d'apprentissage utilisées dans les travaux présentés aux chapitres 4 et 5. De nombreux points ont été laissés de côté, dont notamment la théorie de l'apprentissage statistique, et les nombreuses autres méthodes qui auraient pu être testées sur les problèmes traités dans les chapitres suivants. L'objectif n'était pas d'être exhaustif, ce qui aurait encore rallongé ce chapitre, déjà long, mais de décrire le plus clairement possible les méthodes que nous avons effectivement employées.

Les réseaux de neurones pour la classification, présentés dans la section 3.8.3 du présent chapitre, sont utilisés dans le chapitre 4 pour l'apprentissage d'un modèle de prévision de la charge de travail des contrôleurs aériens. Le travail de thèse de Richard Alligier sur la prévision de l'altitude des avions en montée, présenté au chapitre 5, fait appel aux modèles linéaires de la section 3.7 du présent chapitre, aux réseaux de neurones pour la régression (section 3.8.2), et aux *gradient boosting machines* (section 3.9).

Les chapitres suivants donnent le détail de l'application de ces méthodes d'apprentissage, et des méthodes d'optimisation présentées au chapitre 2, aux problèmes *ATM* que nous avons traités : la gestion prévisionnelle de l'espace aérien, au chapitre 4, et l'extraction du vent ou l'apprentissage de modèles pour la prévision de trajectoires à partir de données enregistrées, dans le chapitre 5.

14. L'*early stopping* consiste à tester des valeurs de M croissantes, en évaluant l'erreur de prédiction sur un ensemble séparé, et à s'arrêter à la valeur de M pour laquelle l'erreur est minimale.

Chapitre 4

Application à la gestion prévisionnelle de l'espace aérien

Sommaire

4.1	Chronologie et principales publications associées	126
4.2	Travaux antérieurs	127
4.3	Problématique	127
4.4	Travaux préliminaires (avant 2002)	130
4.5	Recherche prospective (2006-2008)	136
4.6	Maquette d'IHM et valorisation (2009-2011)	144
4.7	Discussion et conclusion	150

Ce chapitre décrit les recherches sur la prévision de la charge de travail des contrôleurs et des ouvertures de secteurs aériens, que j'ai mené dans différents services de la DGAC : au Laboratoire d'Optimisation Globale ÉNAC/CENA, au Centre d'Etudes de la Navigation Aérienne, puis au domaine R&D de la Direction de la Technique et de l'Innovation de la DSNA (Direction des Services de la Navigation Aérienne).

Cette démarche de recherche, motivée par mon expérience personnelle dans un centre de contrôle aérien en début de carrière, a débuté par quelques travaux préliminaires sous la direction de Jean-Marc Alliot, avant et pendant ma thèse. L'objectif de ces travaux était de calculer un schéma prévisionnel d'ouverture de secteurs, sur la base des indicateurs existants (flux entrants, capacités des secteurs).

Elle s'est poursuivie plus tard par une phase de recherche prospective, avec la modélisation la charge de travail des contrôleurs aériens par un réseau de neurones. L'idée était de mettre en relation des indicateurs de la complexité du trafic et des secteurs, et des observations sur les ouvertures effectives des secteurs aériens, pour construire par apprentissage un modèle de charge. Cette recherche s'est faite en collaboration avec Kévin Guittet, du Laboratoire d'Economie et d'Econométrie de l'Aérien de l'ÉNAC, pour la définition et le codage des indicateurs de complexité ATC.

La phase de valorisation de ces travaux a ensuite été l'occasion pour moi de rassembler une équipe multidisciplinaire comprenant des experts du contrôle aérien, des chercheurs en interface homme-machine, et des ingénieurs de recherche, avec pour objectif la réalisation d'une maquette d'IHM permettant la démonstration des algorithmes proposés et de leur

usages possibles par des opérateurs ATM. Je tiens à remercier Géraldine Beboux, Claude Chamayou, Florence Cressent, Christophe Hurter, Nicolas Saporito, David Szymanski, Gilles Tabard, Benjamin Tissoires pour ce travail collectif.

L'ensemble de ces travaux ont donné lieu à deux rapports techniques [81, 74], huit publications dans des conférences internationales avec comité de relecture ([68, 67, 72, 73, 63, 64, 66, 141]), une publication dans un journal international ([65]), et un partenariat industriel avec Thales, dans le cadre du work package 10.8.1 du programme SESAR.

4.1 Chronologie et principales publications associées

- **Avant 2002**, travaux préliminaires (Laboratoire d'Optimisation Globale ÉNAC/CENA), partitionnement optimal en fonction des capacités secteurs déclarées :
 - [68] D. GIANAZZA, J. M. ALLIOT, AND G. GRANGER. *Optimal combinations of air traffic control sectors using classical and stochastic methods*. In Proceedings of the 2002 International Conference on Artificial Intelligence, IC-AI'02, Las Vegas, 2002.
 - [67] D. GIANAZZA AND J. M. ALLIOT. *Optimization of air traffic control sector configurations using tree search methods and genetic algorithms*. In Proceedings of the 21st IEEE/AIAA Digital Avionics Systems Conference, 2002.
- **2002-2006** : en suspens.
- **2006-2008**, recherche prospective (DSNA/DTI/R&D/POM), utilisation d'indicateurs de complexité du trafic :
 - [72] D. GIANAZZA AND K. GUITTET. *Evaluation of air traffic complexity metrics using neural networks and sector status*. In Proceedings of the 2nd International Conference on Research in Air Transportation. ICRAT, 2006.
 - [73] D. GIANAZZA AND K. GUITTET. *Selection and evaluation of air traffic complexity metrics*. In Proceedings of the 25th IEEE/AIAA Digital Avionics Systems Conference. DASC, 2006.
 - [63] D. GIANAZZA. *Airspace configuration using air traffic complexity metrics*. In Proceedings of the 7th USA/Europe Seminar on Air Traffic Management Research and Development, 2007. Best paper of "Dynamic Airspace Configuration" track.
 - [64] D. GIANAZZA. *Smoothed traffic complexity metrics for airspace configuration schedules*. In Proceedings of the 3rd International Conference on Research in Air Transportation. ICRAT, 2008.
- **2009-2011**, conception participative (avec experts ATC et IHM) et développement d'une maquette d'interface homme-machine :
 - [66] D. GIANAZZA, C. ALLIGNOL, AND N. SAPORITO. *An efficient airspace configuration forecast*. In Proceedings of the 8th USA/Europe Air Traffic Management R & D Seminar, 2009.
 - [141] N. SAPORITO, C. HURTER, D. GIANAZZA, AND G. BEBOUX. *A participatory design for the visualization of airspace configuration forecasts*. In Proceedings of the 4th International Conference on Research in Air Transportation, 2010.
 - [65] D. GIANAZZA. *Forecasting workload and airspace configuration with neural networks and tree search methods*. Artificial Intelligence Journal, Elsevier, 174(7-8) :530–549, may 2010.

Valorisation dans les projets européens SESAR :

- SESAR 4.7.1 Complexity management in en-route
 - SESAR 10.8.1 Complexity assessment and resolution
 - et plus marginalement SESAR 7.5.4 Dynamic airspace configuration
- **Fin 2011** : interruption du développement des logiciels à la DSNA/DTI, pour cause de départ des participants suite à la réorganisation simultanée du domaine R&D de la DSNA/DTI, et de la recherche à la DGAC (Direction Générale de l'Aviation Civile)

4.2 Travaux antérieurs

On trouve peu de travaux avant les années 2000 sur les schémas prévisionnels d'ouverture de secteurs aériens, en dehors de ceux de Verlhac et Manchon à Eurocontrol. Dans [170], ils proposent trois modèles de programmation entière, mixte, de programmation quadratique pour optimiser les schémas d'ouverture.

L'objectif est d'obtenir un schéma prévisionnel de meilleure qualité que celui transmis par les centres de contrôle à la Central Flow Management Unit (voir 1.8 pour le contexte opérationnel). Dans la méthode proposée dans [170], le schéma initial transmis par les centres sert uniquement à déterminer le nombre de postes de travail à ouvrir à chaque instant de la journée. Dans le calcul du schéma optimisé, la configuration pour chaque créneau d'une demi-heure est prise parmi les configurations possibles ayant exactement le même nombre de secteurs ATC que celui du schéma initial. Par ailleurs, l'ensemble des configurations possibles est restreint à un sous-ensemble prédéfini de configurations usuelles. La résolution autorise la réaffectation du trafic excédentaire (au-dessus de la capacité des secteurs considérés) à des créneaux horaires ultérieurs, simulant ainsi de façon macroscopique l'allocation de créneaux de décollage faite par la CFMU.

Les principales critiques que l'on peut faire à cette approche sont les suivantes : elle n'explore pas l'ensemble de toutes les partitions possibles de l'espace ; Le nombre de postes de travail est considéré comme une contrainte dure, alors que, dans un souci d'efficacité et quand le trafic est faible, on pourrait envisager d'ouvrir moins de positions de contrôle que ce que prévoyait le tour de service ; Enfin, elle ne répond pas à la question : quel est le nombre de postes de travail nécessaires, et quel est le schéma optimal, pour un trafic donné, considéré comme fixé ?

4.3 Problématique

La gestion quotidienne d'une salle de contrôle aérien consiste à affecter les secteurs d'espace aux postes de travail (ou *positions de contrôle*) des contrôleurs aériens. L'ensemble des secteurs d'espace affectés à un même poste de travail constituent ce qu'on appelle un *secteur de contrôle* (ou *secteur ATC*).

Dans le cadre de la prévision à court-moyen terme, la problématique consiste à prévoir à l'avance combien de postes de travail seront nécessaires pour traiter le trafic prévu et quels secteurs de contrôle risquent d'être surchargés, afin de mettre en place des mesures préventives (ajustement de tour de service, régulation du trafic).

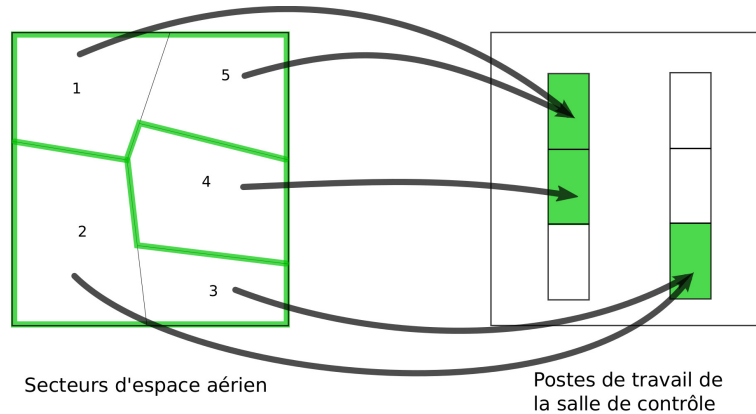


FIGURE 4.1: Affectation des secteurs d'espace aux postes de travail.

Motivations de la démarche de recherche

Les principaux inconvénients de la pratique opérationnelle, en ce qui concerne les aspects liés à la prévision, peuvent être résumés en quelques points. Tout d'abord, étant donné l'aspect combinatoire des problèmes de partitionnement, il est impraticable de définir de manière statique toutes les configurations possibles de secteurs (*i.e.* toutes les façon d'affecter des secteurs d'espace aux postes de travail), sauf pour de petites instances. Ensuite, il n'y a pas de critère quantifié permettant de juger si une configuration de secteurs est meilleure qu'une autre.

Les autres inconvénients concernent le choix des variables et la façon de les mesurer. Jusqu'à récemment, les flux entrants étaient évalués sur des tranches horaires fixes. Avec ce système, la variable flux n'est mesurée qu'à des instants discrets, relativement espacés dans la journée : toutes les heures ou toutes les demi-heures, typiquement. Par exemple, on compte le nombre d'avions qui entrent dans un secteur entre 9h et 10h, puis entre 10h et 11h, mais qu'en est-il entre 9h30 et 10h30? Le défaut majeur des pratiques présentées au chapitre 1, section 1.8, concerne cependant l'estimation de la charge de travail et la détection des surcharges, qui s'appuient sur les flux entrants et les capacités, avec les inconvénients suivants :

- ces variables, faciles à calculer et héritées des pratiques passées (comptage, papier+crayon), ne sont pas suffisamment pertinentes,
- le paramétrage des valeurs-seuils (capacités ou *monitoring values*) doit être particularisé pour chaque secteur de contrôle.

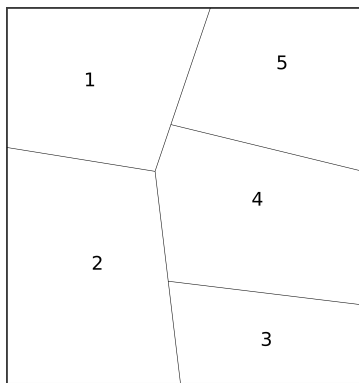
La démarche présentée dans ce chapitre a pour motivation principale de pallier à ces divers inconvénients. Afin de fournir une prévision plus réaliste des ouvertures de secteurs, on souhaite donc :

- explorer toutes les configurations possibles de secteurs,
- évaluer la charge de travail à chaque minute de la journée,
- associer un coût à chaque configuration de secteurs, afin de pouvoir les comparer pour sélectionner la meilleure,

Enfin, on souhaite utiliser des indicateurs qui soient réellement représentatifs de la charge de travail, et dont le choix serait basé sur des observables et non sur un choix arbitraire d'une valeur-seuil pour chaque secteur.

Une des idées originales du travail de recherche mené sur cette thématique a été d'utiliser les archives d'ouvertures de secteurs comme observables, pour apprendre un réseau de neurones qui sera ensuite utilisé comme modèle de prévision de la charge de travail. Une fois calibré, ce modèle peut alors servir à détecter les surcharges de travail (ou les sous-charges), et à quantifier le coût associé à une configuration de secteurs.

Formalisation des problèmes



Regroupements autorisés :

$a : \{2, 3\}$

$b : \{3, 4\}$

$c : \{4, 5\}$

$d : \{1, 5\}$

$e : \{1, 2, 3, 4, 5\}$

$s : \textit{singleton}$

FIGURE 4.2: Un exemple de secteurs d'espace aérien d'un bloc fonctionnel.

Les figures 4.2 et 4.1 illustrent ce partitionnement de l'espace en secteurs de contrôle, sur un exemple avec cinq secteurs d'espace et une liste de regroupements autorisés.

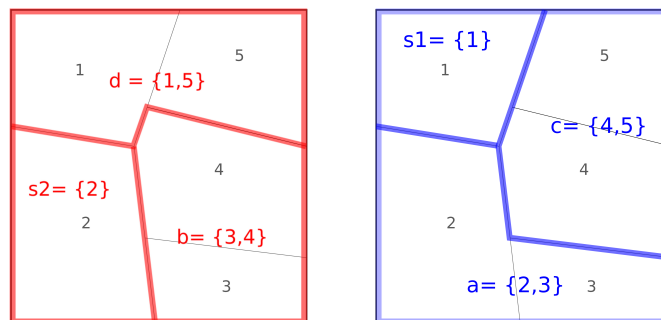


FIGURE 4.3: Autres partitions possibles de l'espace aérien en secteurs de contrôle.

Le partitionnement de l'espace aérien en secteurs de contrôle varie au cours de la journée, en fonction de la charge de travail ressentie sur les positions de contrôle. La figure 4.3 illustre quelques alternatives possibles à la partition présentée sur la figure 4.1. Certaines contraintes opérationnelles sont également à prendre en compte : tour de service, nombre maximum de postes de travail disponibles, règles de transfert de secteurs d'espace d'une position à l'autre, liste des regroupements autorisés de secteurs.

L'objectif premier de cette gestion dynamique des secteurs est d'éviter les surcharges, qui mettent en jeu la sécurité des vols. Lorsqu'un poste de travail est surchargé, une partie des secteurs d'espace qu'il gère est transférée à un autre poste de travail, afin d'en alléger la charge. Lorsque ce n'est pas possible, il faut envisager des mesures de régulation du trafic (délais au décollage, re-routages). De telles situations doivent alors être anticipées suffisamment tôt pour pouvoir mettre en œuvre ces mesures. Afin de gérer l'espace à moindre coût, on souhaite également ouvrir un minimum de postes de travail, et éviter les sous-charges.

L'affectation dynamique des secteurs d'espace aux postes de travail permet d'ajuster en temps réel la sectorisation en fonction de la charge de travail, selon les critères que nous venons d'évoquer. Cependant, il manque aujourd'hui encore des outils de prévision fiables pour les ouvertures de secteurs, qui permettraient d'anticiper avec plus de précision l'évolution future des charges de travail et du partitionnement de l'espace en secteurs de contrôle. De tels outils doivent s'appuyer sur deux éléments essentiels : une estimation fiable de la charge de travail des contrôleurs, et un algorithme permettant d'obtenir un partitionnement optimal de l'espace.

Difficulté du problème de partitionnement et choix des méthodes

Le problème de partitionnement optimal de l'espace est fortement combinatoire : le nombre total de partitions possibles est égal au nombre de Bell. Toutefois, les contraintes opérationnelles, et notamment la restriction à certains regroupements autorisés, réduisent la combinatoire du problème.

Pour des instances relativement modestes et suffisamment contraintes, on peut donc espérer appliquer avec succès une méthode de recherche arborescente (voir section 2.3 du chapitre 2), pour explorer de façon déterministe l'ensemble des partitions possibles. Par contre, ce type de méthode risque de ne plus être applicable pour des blocs fonctionnels contenant un nombre important de secteurs, ou avec moins de contraintes sur les regroupements autorisés. Dans ce dernier cas, un partitionnement optimal ou presque optimal peut être recherché par une métaheuristique (voir chap. 2, section 2.5).

4.4 Travaux préliminaires (avant 2002) : partitionnement optimal en fonction des capacités secteurs déclarées

- [68] D. Gianazza, J. M. Alliot, and G. Granger. Optimal combinations of air traffic control sectors using classical and stochastic methods. In *Proceedings of the 2002 International Conference on Artificial Intelligence, IC-AI*, Las Vegas, 2002.
- [67] D. Gianazza and J. M. Alliot. Optimization of air traffic control sector configurations using tree search methods and genetic algorithms. In *Proceedings of the 21st IEEE/AIAA Digital Avionics Systems Conference*, 2002.

Les travaux présentés dans ces publications sont antérieurs à ma thèse et ne sont présentés dans ce mémoire d'habilitation qu'au titre de travaux préliminaires ayant donné lieu à une démarche plus aboutie après ma thèse.

Résumé

Les hypothèses faites dans ces travaux préliminaires correspondent à la pratique opérationnelle du moment :

- un secteur de contrôle est constitué d'un ou plusieurs secteurs d'espace affectés à un même poste de travail. On se limite aux regroupements autorisés en opérationnel.
- un secteur de contrôle est considéré comme surchargé dès que le flux entrant d'avions devient supérieur à une valeur-seuil, appelée *capacité du secteur*,
- on tient compte des capacités déclarées par les centres de contrôle, pour chacun des secteurs de contrôle,

Les innovations par rapport à la pratique opérationnelle, sont les suivantes :

- on calcule une partition optimale de l'espace aérien en secteurs de contrôle, plutôt que de choisir manuellement parmi quelques configurations prédéfinies de secteurs,
- on explore toutes les partitions possibles construites à partir des regroupements autorisés, au lieu de se restreindre à quelques partitions prédéfinies.

Les deux publications [68, 67] explorent deux approches pour la construction de partitions optimales de l'espace aérien, l'une par recherche arborescente (*depth first* et *best first*), l'autre par un algorithme évolutionnaire. Les différentes approches sont comparées sur des instances réelles de secteurs aériens et de trafic français. La première publication [68] décrit les algorithmes proposés, et la deuxième [67] examine les bénéfices de l'optimisation, à la fois sur les délais au décollage imposés aux avions, et sur l'utilisation des ressources, en termes d'ouverture de postes de travail.

Ces travaux préliminaires ont permis de montrer que, pour des instances de taille modeste, le partitionnement optimal de l'espace peut être obtenu par des méthodes exactes de recherche arborescente. En pratique, ces méthodes se sont montrées les plus efficaces sur les instances traitées (les secteurs des cinq centres de contrôle aérien français).

L'emploi d'une métaheuristique reste toutefois une bonne alternative, au prix de temps de calcul nettement plus longs, dans l'hypothèse où l'augmentation future du nombre de secteurs ou de regroupements autorisés rendrait impraticable les méthodes de recherche arborescente.

En comparant les schémas d'ouvertures calculés par nos algorithmes avec les schémas prévisionnels effectivement déposés par les cinq centres de contrôle français, on montre par ailleurs que l'optimisation des ouvertures de secteurs permet un gain de l'ordre de 70% des délais au décollage (en ne considérant toutefois que l'espace français), avec une diminution de l'ordre de 20% de l'utilisation des ressources.

Ces bons résultats apparents sont toutefois largement à relativiser. En effet, si on compare les schémas prévisionnels, quels qu'ils soient (calculés ou déposés), avec les ouvertures de postes de travail effectivement réalisées¹, on constate de fortes différences. En pratique, les contrôleurs ouvrent largement moins de positions de contrôle, en moyenne, que ce qu'indiquent les schémas prévisionnels.

Ceci indique que le choix des variables (flux entrants) et des capacités de secteurs, qui étaient pourtant ceux utilisés en opérationnel à l'époque, ne permet pas une estimation fiable de la charge de travail. Avant de présenter les travaux sur la recherche d'indicateurs plus pertinents, décrivons d'abord succinctement les méthodes que nous venons d'évoquer.

Une approche par algorithme génétique

Dans [67, 68], un algorithme génétique [78, 118] est utilisé pour construire une partition optimale de l'espace aérien en secteurs de contrôle.

1. Cette comparaison n'a pas été publiée.

Dans cette approche, chaque individu de la population est une configuration de secteurs de contrôle, c'est-à-dire une partition de l'ensemble des secteurs d'espace. A chaque itération, l'algorithme génétique sélectionne un pool de parents, qui sont ensuite recombinaisonnés par croisement et mutation pour produire une nouvelle population. Les individus les moins adaptés sont éliminés, et remplacés par d'autres tirés au hasard dans le reste de la population. De nombreux raffinements sont possibles dans le choix des opérateurs de sélection et de remplacement, de mise à l'échelle des critères d'adaptation (*scaling, sharing*). Le lecteur pourra se référer au chapitre 3 de [55] pour plus de détails.

Dans [67, 68], l'opération de mutation d'un individu (*i.e.* une configuration de secteurs) consiste à choisir au hasard un des secteurs de contrôle et un de ses voisins, puis à repartitionner les secteurs d'espace composant ces deux secteurs de contrôle. Ce repartitionnement est aléatoire, mais restreint à un résultat entre un et trois secteurs maximum. Les nouveaux secteurs viennent remplacer les deux secteurs initiaux de l'individu muté.

L'opération de croisement de deux parents consiste à remplacer une partie des secteurs de contrôle de chaque parent par des secteurs provenant de l'autre parent. Le résultat ne forme généralement pas une partition complète de l'espace, et il faut donc le compléter en choisissant aléatoirement parmi les secteurs de contrôle compatibles avec la configuration incomplète.

Le critère d'adaptation (*fitness*) tient compte, par ordre de priorité décroissante, des surcharges élevées, du nombre de postes de travail ouverts, des sous-charges importantes, et enfin des surcharges et sous-charges dans les marges de tolérance. Pour un secteur de contrôle donné, la charge est évaluée en fonction de l'écart entre le flux entrant dans le secteur et un seuil maximal autorisé (la *capacité* du secteur), en utilisant les valeurs utilisées en opérationnel. Les valeurs brutes du critère d'adaptations sont modifiées par un opérateur de *clusterized sharing*, puis par *sigma truncation* ([78], ou [55] p.59), afin de laisser aux individus les moins adaptés une meilleure chance de se reproduire et de permettre ainsi une exploration plus large de l'espace d'états. Pour l'opérateur de *sharing*, la difficulté consiste à définir une distance entre des partitions différentes de l'ensemble des secteurs d'espace. La pseudo-distance choisie est analogue à la distance de Hamming, à la différence près que les suites de symboles pour lesquelles on comptabilise les différences n'ont pas la même longueur.

Une stratégie élitiste permet de préserver les meilleurs individus lors de la constitution de la nouvelle population, pour l'itération suivante. Cette nouvelle population est par ailleurs constituée des individus résultant des croisements et mutations, complétée ensuite selon le mécanisme du *stochastic remainder without replacement* (voir [55]).

Méthodes de recherche arborescente

Deux stratégies de recherche arborescente sont également présentées dans [67, 68]. L'une explore en profondeur (*depth first*) l'arbre permettant de construire les partitions de l'espace. Ce principe de recherche arborescente est illustré sur la Figure 4.4, sur notre exemple à cinq secteurs. L'autre méthode de recherche arborescente explore en premier le nœud ayant la meilleure estimation du coût total du cheminement entre la racine et une feuille.

La Figure 4.4 reprend l'exemple à 5 secteurs d'espace introduit au début du chapitre. Les secteurs élémentaires d'espaces sont numérotés de 1 à 5. Chaque nœud de l'arbre est une partition en construction. Les feuilles sont des partitions complètes de l'espace en secteurs de contrôle. Les regroupements autorisés sont rappelés en haut à gauche et dénotés par des lettres, la lettre 's' étant une notation générique pour un singleton (secteur de contrôle constitué d'un seul secteur d'espace).

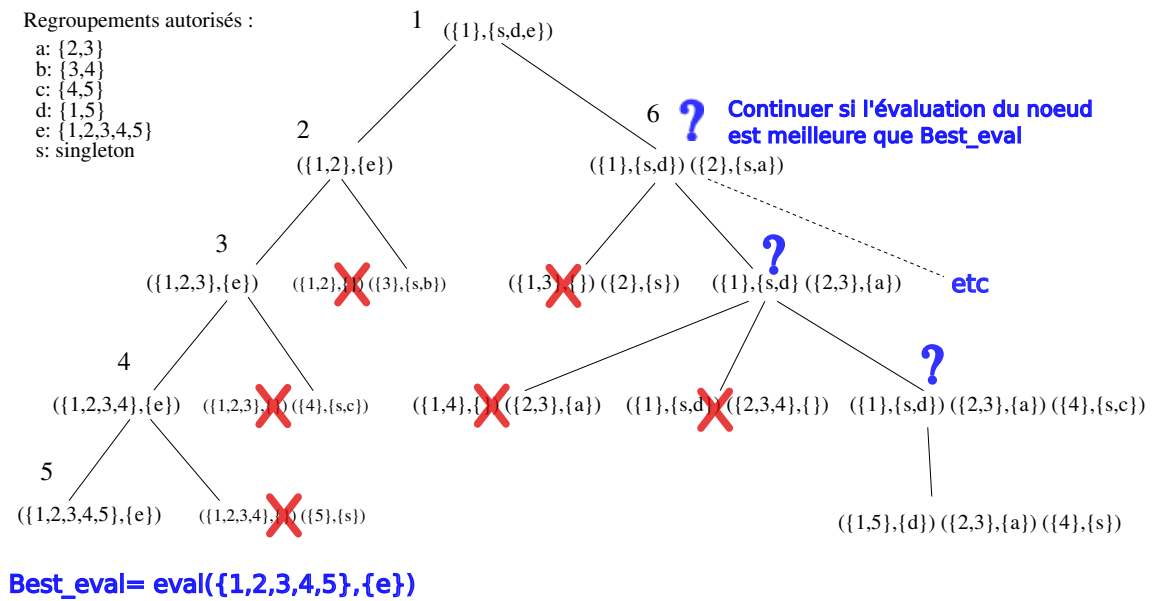


FIGURE 4.4: Recherche d'une partition optimale par un algorithme de recherche arborescente.

L'exploration de toutes les partitions valides se fait en examinant successivement chacun des secteurs élémentaires, soit en cherchant à les ajouter dans un groupe existant, soit en constituant un nouveau groupe contenant ce seul secteur. Pour chaque groupe en construction, on tient à jour la liste des regroupements autorisés compatibles avec la partition en construction.

Dans notre exemple, on démarre à la racine avec le groupe $\{1\}$ contenant le seul secteur d'espace 1. Les regroupements autorisés compatibles avec ce groupe sont le singleton $s = \{1\}$, et les regroupements $d = \{1, 5\}$ et $e = \{1, 2, 3, 4, 5\}$. La branche de gauche sous la racine correspond à l'ajout du secteur 2 dans le groupe existant, pour obtenir $\{1, 2\}$. On voit que s et d ont été supprimés de la liste des regroupements compatibles, puisqu'ils ne contiennent pas le secteur 2. La branche de droite sous la racine correspond à l'ajout d'un nouveau groupe $\{2\}$ à la partition en construction. On voit que la liste des regroupements compatibles associée au groupe $\{1\}$ est maintenant $\{s, d\}$ au lieu de $\{s, d, e\}$ à la racine. En effet, le regroupement $e = \{1, 2, 3, 4, 5\}$ n'est plus compatible avec la partition en construction, puisque 1 et 2 y sont dans des groupes séparés.

L'exploration de l'arbre se poursuit de manière similaire jusqu'à atteindre des partitions complètes (les feuilles). Les nœuds pour lesquels un des groupes a une liste de regroupements compatibles vide ne sont pas développés. Ce sont ceux avec une croix rouge sur la Figure 4.4. On voit que le fait de se restreindre aux seuls regroupements autorisés permet de réduire la combinatoire de notre problème de partitionnement.

Dans le cas d'une recherche en profondeur d'abord de type *Branch & Bound*, on mémorise la meilleure partition trouvée (en bas à gauche dans notre exemple). On évalue chaque nœud exploré et on compare cette évaluation à celle de la meilleure partition. Les fonctions d'évaluation pour les nœuds et les feuilles doivent être telles que l'évaluation d'un nœud soit toujours meilleure que les évaluations des feuilles contenues dans le sous-arbre correspondant. Avec cette hypothèse, on peut s'éviter d'explorer plus avant les nœuds moins bien évalués que

la meilleure feuille déjà trouvée. En effet, toutes les feuilles atteignables à partir de ce nœud auront nécessairement une évaluation moins bonne.

Une recherche arborescente de type *meilleur en premier* a également été testée. En pratique, l'heuristique employée n'étant pas très bonne, c'est le *Branch & Bound* avec une stratégie *depth-first* qui s'est avéré le plus efficace, à condition d'explorer l'arbre comme sur la Figure 4.4, en privilégiant en priorité les configurations les plus regroupées.

Ces méthodes de recherche arborescente sont testées sur des instances réelles, sur les secteurs aériens des cinq centres de contrôle français. Les résultats montrent que, sur des instances de cette taille et avec les contraintes opérationnelles existantes sur les regroupements possibles, l'optimum global est atteignable par ce type de méthodes en un temps très court (quelques secondes au maximum, sur un Pentium IV 1.8 GHz).

Dans [67, 68], la recherche arborescente avec les stratégies "en profondeur d'abord" et "meilleur en premier" est comparée à l'algorithme génétique présenté en section 4.4. Avec 220 éléments de population évoluant sur 300 générations, et avec une probabilité de croisement de 0.6 et de mutation de 0.2, l'algorithme génétique retrouve l'optimum global dans la quasi-totalité des cas. Les temps de calcul sont toutefois beaucoup plus longs (plusieurs minutes).

Enseignements tirés des travaux préliminaires

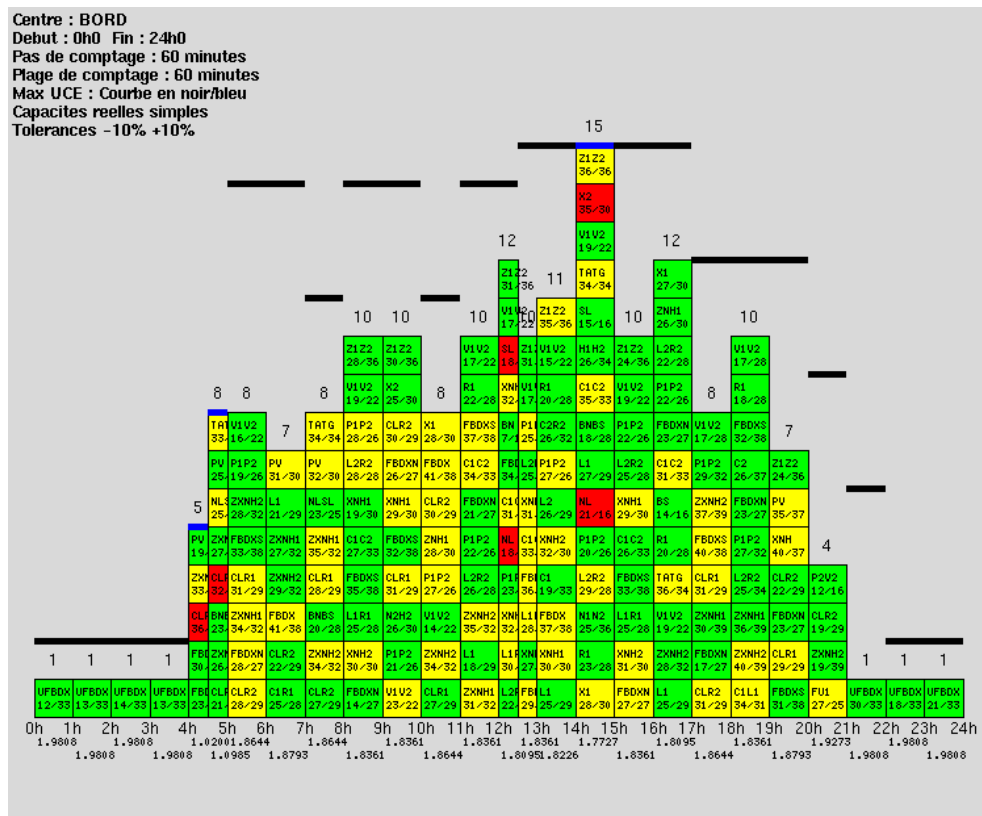


FIGURE 4.5

Les Figures 4.5 et 4.6 résument à elles seules les enseignements que l'on a pu tirer des travaux préliminaires basés sur les métriques utilisées en opérationnel (flux entrants et capa-

comme contrainte le nombre de postes de travail effectivement ouverts ce jour-là.

La comparaison des deux figures, 4.5 et 4.6, amène plusieurs commentaires. On constate tout d'abord que le nombre de positions de contrôle effectivement ouvertes, de l'ordre de 8 à 9 en pointe, est généralement très inférieur aux contraintes du tour de service (14 à 15 en pointe sur la Figure 4.5). Paradoxalement sur la Figure 4.5, un très grand nombre de secteurs ATC se retrouvent en surcharge, au sens où le flux entrant excède la capacité (carrés rouges).

Sachant que le tour de service prévoyait suffisamment de contrôleurs pour pouvoir ouvrir un plus grand nombre de postes de travail, on peut en déduire que le modèle basé sur les flux entrant et les capacités n'est pas représentatif de la charge de travail réellement ressentie par les contrôleurs. Ces considérations nous ont amené à rechercher des indicateurs et un modèle plus pertinents pour la charge de travail des contrôleurs aériens, en s'appuyant sur des indicateurs de complexité ATC.

4.5 Recherche prospective (2006-2008) : utilisation d'indicateurs de complexité ATC

- [72] D. Gianazza and K. Guittet. Evaluation of air traffic complexity metrics using neural networks and sector status. In *Proceedings of the 2nd International Conference on Research in Air Transportation*. ICRAT, 2006.
- [73] D. Gianazza and K. Guittet. Selection and evaluation of air traffic complexity metrics. In *Proceedings of the 25th IEEE/AIAA Digital Avionics Systems Conference*. DASC, 2006.
- [63] D. Gianazza. Airspace configuration using air traffic complexity metrics. In *Proceedings of the 7th USA/Europe Seminar on Air Traffic Management Research and Development*, 2007. Best paper of "Dynamic Airspace Configuration" track.
- [64] D. Gianazza. Smoothed traffic complexity metrics for airspace configuration schedules. In *Proceedings of the 3rd International Conference on Research in Air Transportation*. ICRAT, 2008.

Résumé

Dans les travaux préliminaires d'avant 2002 (voir section précédente), le choix des variables utilisées en opérationnel (flux entrants et capacités) ne permettait pas une estimation fiable de la charge de travail. Dans les travaux présentés ici [72, 73, 63, 64], on cherche des indicateurs plus pertinents pour prévoir la charge de travail des contrôleurs aériens, dans le cadre d'une application à la prévision des ouvertures de secteurs aériens.

Deux premières publications [72, 73] s'intéressent à la sélection de métriques de complexité ATC comme variables explicatives de la charge de travail, en utilisant un réseau de neurones comme modèle de prévision. Ces métriques sont prises parmi les nombreux indicateurs de complexité ATC proposés dans la littérature. Un apprentissage supervisé permet d'apprendre le réseau de neurones sur une base d'exemples regroupant des instances associant les réponses attendues, ici une quantification de la charge de travail effective, à des entrées données, qui sont ici des mesures des indicateurs de complexité ATC. L'originalité de notre travail est d'avoir choisi une quantification objective de la charge de travail s'appuyant sur l'état observé des secteurs de contrôle (nous y reviendrons plus loin). Les deux publications suivantes [63, 64]

intègrent ce modèle de prévision de la charge de travail dans l'algorithme de partitionnement optimal qui calcule les configurations de secteurs.

En rentrant un peu plus dans le détail, dans [72] nous appliquons dans un premier temps une analyse en composantes principales (ACP²) à 27 indicateurs choisis dans la littérature, et calculés à chaque minute d'une journée de trafic pour tous les secteurs de contrôle aériens français. L'objectif de cet ACP est de réduire la dimensionalité de l'espace des entrées de notre modèle de prévision, en regroupant dans une même composante les variables explicatives corrélées entre elles. Dans un deuxième temps, ces composantes principales résultant de l'ACP sont utilisées en entrée du réseau de neurones, en y ajoutant éventuellement le volume V du secteur. Différentes combinaisons des variables d'entrées (composantes ou volume V) sont testées. Une procédure de sélection basée sur le critère d'Akaike (AIC), calculé sur l'ensemble d'apprentissage et sur un ensemble de test, permet de déterminer quelle combinaison explique au mieux la charge de travail sur notre base d'exemples. L'utilisation du volume du secteur et des quatre premières composantes principales permet d'obtenir un taux de classifications correctes de l'ordre de 82 % sur l'état du secteur de contrôle (la variable réponse de notre modèle).

Dans [73], nous cherchons à nous abstraire de la phase d'extraction de variables via ACP, en sélectionnant un jeu de variables explicatives parmi les indicateurs initiaux (non transformés). L'objectif est de rendre plus compréhensibles pour un opérationnel les variables d'entrée de notre modèle, quitte à perdre un peu d'information utile en entrée de celui-ci. Pour cela, nous cherchons d'abord les composantes les plus pertinentes, de la même manière que dans [72] mais en utilisant un critère d'information légèrement différent (BIC). Nous identifions ensuite les variables prépondérantes dans chaque composante, avant d'appliquer finalement une procédure de type *forward selection* sur cet ensemble de variables. Ceci nous permet d'identifier 6 variables explicatives simples expliquant au mieux la charge de travail : le volume du secteur V , le nombre d'avions dans le secteur Nb , la moyenne des vitesses verticales des avions dans le secteur avg_vs , les flux entrants F_{15} et F_{60} à des horizons temporels de 15 minutes et 60 minutes respectivement, et le nombre *inter_hori* de croisements de trajectoires dans le plan horizontal avec un angle supérieur à 20 degrés. Ces 6 variables $\{V, Nb, avg_vs, F_{60}, F_{15}, inter_hori\}$, utilisées directement en entrée de notre modèle de prévision, donnent un taux de classifications correctes analogue à celui obtenu en utilisant les composantes principales, et s'avèrent plus simple à mettre en œuvre et à comprendre pour un utilisateur opérationnel.

Dans [63], le modèle de prévision de la charge de travail est utilisé, avec l'algorithme de *Branch & Bound* décrit dans la section 4.4 (voir Figure 4.4), pour construire un schéma prévisionnel d'ouverture de secteurs de contrôle. La Figure 4.7 montre à titre d'illustration l'évolution du nombre de postes de travail au cours d'une journée (différente de celle choisie pour calibrer le modèle), selon que l'on considère les ouvertures de secteurs du schéma prévisionnel déposé par l'opérateur FMP (*Flow Management Position*), celles calculées par notre modèle, ou celles effectivement réalisées ce jour-là au centre de contrôle de Reims.

Dans [64], on s'intéresse à l'influence du lissage des variables d'entrée sur le taux de classifications correctes, et sur le réalisme de la prévision du nombre de postes de travail à ouvrir pour traiter un trafic donné. Avec les choix faits précédemment pour nos variables, notre modèle de charge, et notre fonction de décision de reconfiguration, un lissage des variables sur une fenêtre glissante d'une trentaine de minutes semble être le meilleur compromis.

2. L'analyse en composantes principales (ACP) est également connue sous le nom de transformation de Karhunen-Loève.

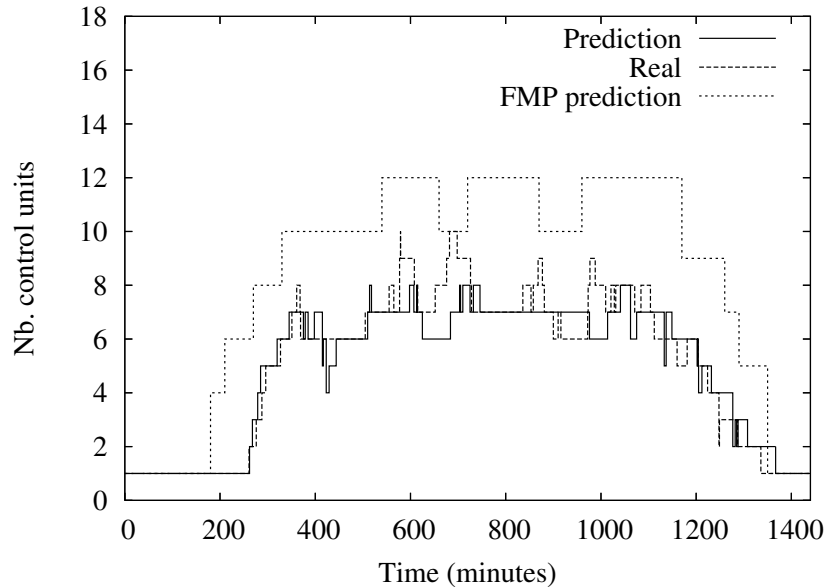


FIGURE 4.7: Évolution du nombre de postes de travail pour le centre de contrôle de Reims (journée du 2 juin 2003). Trois courbes sont représentées, correspondant aux ouvertures des positions de contrôle prévues par la FMP, calculées par le modèle, ou effectivement réalisées.

Les éléments essentiels de l’approche employée dans les publications que nous venons de résumer [72, 73, 63, 64] sont décrits plus en détail un peu plus loin dans cette section. Avant cela, commençons par un très court état de l’art des travaux antérieurs.

Un bref état de l’art

La prévision de la charge de travail suppose l’existence d’une relation fonctionnelle $y = f(x)$ entre une variable réponse y représentative de la charge et un vecteur x de variables explicatives regroupant les indicateurs que l’on juge pertinents pour prévoir celle-ci. A défaut d’une modélisation précise du fonctionnement du cerveau humain, on ne dispose pas de la “vraie” fonction f .

Naturellement, cette charge de travail est directement liée au nombre d’avions que le contrôleur doit gérer sur son poste de travail. L’approche classique utilisée en opérationnel (voir section 1.8.2 du chapitre 1), consiste donc simplement à compter les avions et à chercher à évaluer le nombre maximum d’avions que peut gérer le contrôleur humain. Cette approche est toutefois très limitée, car elle ne tient pas compte de la complexité du trafic ou du secteur (qu’on regroupe sous le terme “complexité ATC”), laquelle peut faire varier fortement la charge de travail ressentie par le contrôleur, à nombre d’avions égal.

Une approche couramment utilisée dans le milieu ATC/ATM pour modéliser la charge de travail consiste à adopter un modèle simplifié des tâches réalisées par le contrôleur aérien, et de leur durée moyenne. Parmi les travaux basés sur cette approche, citons [57] dans le cadre du projet COCA (*Complexity and Capacity Analysis*) d’Eurocontrol, ou encore [174] du côté américain.

De nombreux autres travaux sur la charge de travail des contrôleurs s’appuient sur une démarche d’apprentissage supervisé (voir [40]) ou de modélisation statistique. Comme on

ne connaît pas la “vraie” fonction f , il s’agit alors de sélectionner une hypothèse h parmi un ensemble de fonctions \mathcal{H} appelé espace d’hypothèses (voir le chapitre 3 sur l’apprentissage), en choisissant celle qui donne les meilleures performances sur un ensemble S d’exemples d’entrées et de réponses attendues. Dans la suite, nous noterons y la réponse attendue (collectée dans les exemples), en la distinguant de la sortie \hat{y} calculée par le modèle.

Pour le choix de la variable réponse supposée quantifier la charge de travail effective du contrôleur, on trouve dans la littérature des mesures physiologiques [20] (rythme cardiaque, pression sanguine, résistance électrique de la peau, taux de cortisol et d’adrénaline), des mesures oculaires [5] (fréquence et durée des clignements, fréquence et distance des saccades, diamètre pupillaire), l’occupation de la fréquence radio [114] (durée et fréquence des échanges, type d’instructions données aux avions, etc), ou encore des notes subjectives données par les contrôleurs [153, 103, 106, 102, 125, 115, 34, 33] en temps réel ou à posteriori, sur du trafic réel ou simulé.

Le choix des variables explicatives possibles est également très vaste, et nous n’entrerons pas ici dans le détail des nombreux indicateurs de complexité ATC qui ont pu être proposés dans la littérature (voir [87, 122]).

Concernant le choix du modèle, bon nombre de publications autour du concept de *densité dynamique* [153, 103, 106, 102, 125, 115] font l’hypothèse d’une relation linéaire entre indicateurs de complexité ATC et charge de travail. Quelques autres [34, 33, 5] font l’hypothèse d’un modèle non linéaire et utilisent un réseau de neurones.

Choix de la variable réponse

L’originalité de notre approche, par rapport aux travaux précédents, réside principalement dans le choix de la variable réponse. Plutôt que de choisir une note subjective attribuée par le contrôleur aérien ou d’autres mesures difficiles à collecter, nous utilisons l’état observé des secteurs ATC en faisant l’hypothèse que les décisions d’affectation des secteurs d’espace aux postes de travail sont liées la charge de travail effectivement ressentie par les contrôleurs.

En effet, dans la pratique opérationnelle, lorsque les contrôleurs (radar et organique) en charge d’un secteur ATC commencent à être surchargés, on décide lorsque c’est possible de réaffecter les secteurs d’espace de façon à former deux secteurs ATC plus petits (voire plus) et donc moins chargés, affectés chacun à une position de contrôle distincte. En jargon ATC, on parle de “*dégrouper*” le secteur de contrôle. A l’inverse, la décision de regrouper deux secteurs (ou plus) sur un même poste de travail indique que la charge sur chacun de ces secteurs, pris séparément, est jugée trop faible.

La variable réponse que l’on a choisie correspond donc à l’état de regroupement du secteur de contrôle :

- *regroupé* : charge faible, secteur fusionné avec d’autres,
- *ouvert* (ou “*armé*”) : domaine d’exploitation normale du secteur,
- “*dégrouper*”³ : charge forte, secteur découpé en plusieurs secteurs plus petits.

Les autres états ne sont pas utilisés, car on ne peut rien en inférer concernant la charge de travail. Considérons l’exemple d’un secteur ATC, noté a , et composé des secteurs d’espace 2 et 3. Supposons que ce secteur de contrôle $a = \{2, 3\}$ ne soit pas “armé” tel quel (*i.e.* affecté à un poste de travail). Supposons que l’on observe que les secteurs ATC $s_2 = \{2\}$ et $b = \{3, 4\}$

3. Que le lecteur nous pardonne pour l’usage de ce néologisme, utilisé dans le milieu opérationnel.

soient eux effectivement ouverts. Dans cette configuration, on voit que les secteurs d'espace composant a ne sont ni regroupés dans un ensemble plus vaste, ni séparés en secteurs ATC distincts. On ne peut donc rien en déduire sur ce qu'aurait été la charge de travail effective sur le secteur de contrôle a s'il avait été ouvert.

Par contre, si on observe l'ouverture des secteurs ATC $s_2 = \{2\}$ et $s_3 = \{3\}$, on peut légitimement en déduire que le secteur $a = \{2, 3\}$ aurait été surchargé s'il avait été ouvert tel quel. De même, si on constate l'ouverture du secteur $e = \{1, 2, 3, 4, 5\}$, on peut penser que le secteur $a = \{2, 3\}$ aurait été en sous-charge s'il avait été ouvert.

L'encodage de la variable réponse y se fait sous forme d'un vecteur à trois composantes (voir section 3.8.3 du chapitre 3 pour la justification de ce choix d'encodage), comme récapitulé dans le tableau 4.1. La figure 4.8 donne par ailleurs un exemple, pour une configuration donnée de la salle de contrôle, des encodages des états des secteurs ATC.

TABLE 4.1: Encodage de l'état du secteur ATC, et hypothèse de charge de travail associée

Etat du secteur ATC	Encodage	Charge de travail
<i>Regroupé</i>	$y = (1, 0, 0)^T$	sous-charge
<i>Ouvert</i>	$y = (0, 1, 0)^T$	charge normale
<i>“Dégroupé”</i>	$y = (0, 0, 1)^T$	charge excessive
Autre	donnée non utilisée	charge inconnue

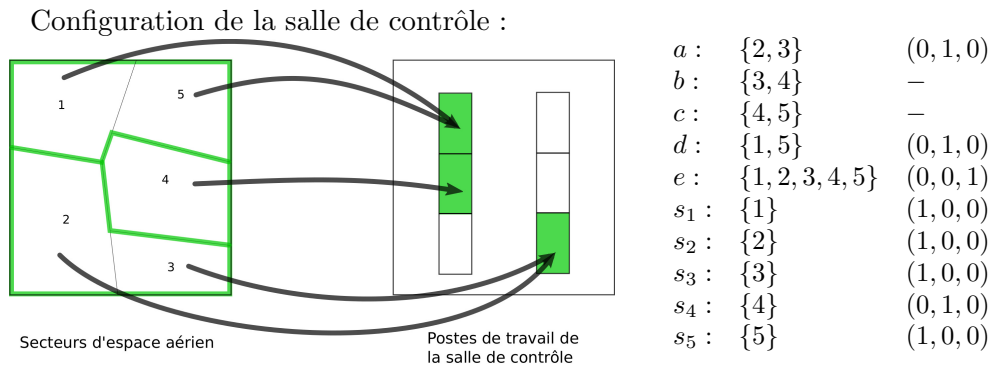


FIGURE 4.8: Exemple d'encodage des états des secteurs de contrôle.

Un réseau de neurones pour l'estimation de la charge de travail

Un réseau de neurones permet de calculer les probabilités (p_1, p_2, p_3) d'être dans un des trois états possibles correspondant à des charges de travail respectives faible, normale, ou trop élevée. Dans la suite, on identifiera plus explicitement ce triplet en le notant $(p_{low}, p_{normal}, p_{high})$. Les entrées du réseau sont des indicateurs de complexité du trafic, calculés à partir des trajectoires des avions, ou des indicateurs portant sur la géométrie du secteur (volume).

Les réseaux de neurones sont présentés en détail dans la section 3.8 du chapitre 3 regroupant les principes et les méthodes de l'apprentissage supervisé. La Figure 3.18 de la section 3.8.1 illustre le type de réseau de neurones que nous avons utilisé (perceptron multi-couche de type *feed-forward*).

Dans notre application, le vecteur de sortie est $y = (p_{low}, p_{normal}, p_{high})^T$, de dimension $P = 3$. La fonction *softmax* est appliquée à la couche de sortie, et assure que $p_{low} + p_{normal} + p_{high} = 1$.

Lors de la phase d'apprentissage, les poids du réseau de neurones sont ajustés sur une base d'exemples, constitués de N couples (x_n, y_n) , $n = 1, \dots, N$, associant variables explicatives et réponse attendue. Ici, les exemples sont élaborés à partir du trafic enregistré et d'archives des affectations de secteurs d'espace aux postes de travail, provenant des cinq centres de contrôle aérien français.

La phase d'apprentissage du réseau de neurones fait appel à une méthode d'optimisation pour minimiser l'erreur en sortie du réseau, celle-ci étant une fonction des poids affectés aux connexions. Dans notre cas, on cherche le vecteur de poids w^* qui minimise l'entropie croisée sur nos N exemples (voir la section 3.8.3 pour la justification de ce choix), donnée par l'équation 4.1 suivantes, où y_{np} désigne la $p^{\text{ième}}$ composante du $n^{\text{ième}}$ exemple de réponse y_n , et \hat{y}_{np} et la $p^{\text{ième}}$ composante du vecteur de sortie calculé par le réseau de neurones, pour l'entrée x_n correspondante :

$$\mathcal{E}(w) = - \sum_{n=1}^N \sum_{p=1}^P y_{np} \ln \left(\frac{\hat{y}_{np}}{y_{np}} \right) \quad (4.1)$$

Les choix faits ici, notamment pour la fonction *softmax* appliquée à la sortie de la dernière couche cachée et la fonction \mathcal{E} que l'on choisit de minimiser, correspondent sur le plan théorique au principe du *maximum de vraisemblance*, dans le cas d'un problème de classification à plus de deux classes (voir la section 3.8.3 du chapitre 3, ou encore [23], section 6.9, ou [47]), et à l'entropie croisée dans la théorie de l'information (voir [159]). Nous n'avons pas employé de technique de régularisation, comme le *weight decay* dans le cas des réseaux de neurones (voir chap. 3, section 3.3.3). Ce type de technique, qu'on peut raccorder sur le plan théorique au *maximum a posteriori* (voir chap. 3, section 3.4.2, ou encore l'article de M. E. Tipping [161] sur les principes d'inférence) n'a pas semblé apporter d'amélioration significative dans notre application, où nous disposons d'un nombre d'exemples suffisamment grand pour éviter les problèmes de sur-apprentissage.

Dans le cadre de nos travaux, les toutes premières expérimentations ont été réalisées avec le logiciel SNNS, pour rapidement passer au package `nnet` développé par Brian Ripley [137], dans l'environnement *R* (<http://www.r-project.org/>). J'ai ensuite développé une version Ocaml d'un réseau de neurones utilisant au choix BFGS, L-BFGS, ou une optimisation par essaim particulaires (PSO) ou évolution différentielle (DE) pour l'apprentissage. Les résultats présentés dans [72, 73, 64] sur l'estimation de la charge de travail ont été obtenus avec un réseau calibré par la méthode BFGS.

Construction des schémas prévisionnels d'ouverture des secteurs

L'algorithme 14 permet de construire le schéma prévisionnel d'ouverture des secteurs de contrôle. Ce schéma prévisionnel est une succession de configurations de secteurs, entre un temps initial t_0 et un temps final $t_0 + horizon$ correspondant à l'horizon de prévision fixé par l'utilisateur. Rappelons qu'une configuration de secteurs est une partition de l'espace aérien (soit encore une partition de l'ensemble des secteurs élémentaires d'espace) en secteurs ATC affectés aux postes de travail gérés par les contrôleurs aériens. Chaque secteur ATC (ou secteur de contrôle) est donc composé d'un ou plusieurs secteurs élémentaires d'espace.

Algorithm 14 Algorithme de construction du schéma prévisionnel d'ouvertures de secteurs.

```

function GET_SECTOR_CONFIGURATIONS( $t_0, horizon$ )
   $t \leftarrow t_0$ 
   $configs \leftarrow \emptyset$ 
   $c \leftarrow \text{INIT}(t_0, S, A)$   $\triangleright$  Ex. la nuit, tous secteurs sur une même position de contrôle.
  while  $t \leq t_0 + horizon$  do
    if OVERLOADS_OR_UNDERLOADS( $c, t$ ) then  $\triangleright$  Voir algorithme 15
       $c \leftarrow \text{RECONFIGURE}(c, t, S, A)$   $\triangleright$  Rech. arborescente (voir Fig. 4.4)
    end if
     $configs[t] \leftarrow c$ 
     $t \leftarrow t + \delta t$ 
  end while
  return  $configs$   $\triangleright$  Retourner le tableau des configurations successives de secteurs.
end function

```

Algorithm 15 Fonction de décision de reconfiguration.

```

function OVERLOADS_OR_UNDERLOADS( $c, t$ )
   $r \leftarrow false$ 
   $l \leftarrow \text{ATC\_SECTORS}(c)$ 
  while  $l \neq \emptyset$  and  $r = false$  do
     $sectATC \leftarrow \text{GET\_ELEMENT}(l)$ 
     $l \leftarrow l \setminus \{sectATC\}$ 
     $(p_{low}, p_{normal}, p_{high}) \leftarrow \text{EVAL\_WORKLOAD}(sectATC, t)$   $\triangleright$  Réseau de neurones
     $p_{max} \leftarrow \text{MAX}(p_{low}, p_{normal}, p_{high})$ 
    if  $(p_{max} = p_{low}$  and  $1 - p_{low} < \alpha)$  or  $(p_{max} = p_{high}$  and  $1 - p_{high} < \beta)$  then
       $r \leftarrow true$ 
    end if
  end while
  return  $r$ 
end function

```

Le principe de l'algorithme est simple : partant d'une configuration initiale à t_0 , on évalue à chaque instant t la configuration courante c , en regardant si certains secteurs sont en surcharge excessive ou en sous-charge. Si c'est le cas, on repartitionne tout ou partie de l'ensemble S des secteurs d'espace, en tenant compte de l'ensemble A des regroupements autorisés, et éventuellement de la configuration courante c (selon la façon choisie de repartitionner). On stocke les partitions successives dans une table indexée par le temps, qui est retournée en résultat de la fonction `GET_SECTOR_CONFIGURATION` de l'algorithme 14.

L'évaluation de la configuration courante est décrite dans l'algorithme 15. La fonction `OVERLOADS_OR_UNDERLOADS` examine chaque secteur ATC de la configuration c , pour évaluer sa charge de travail et décider s'il faut ou non repartitionner. Elle fait appel au réseau de neurones décrit précédemment. L'appel à `EVAL_WORKLOAD` retourne un triplet de probabilités $(p_{low}, p_{normal}, p_{high})$. Le critère de décision consiste à examiner si p_{high} ou p_{low} est suffisamment proche de 1, pour au moins un des secteurs ATC de la configuration.

Le partitionnement optimal de l'espace en secteurs de contrôle, réalisé par la fonction `RECONFIGURE` dans l'algorithme 14, est assuré par une recherche arborescente analogue au *Branch&Bound* présenté en section 4.4 (voir Figure 4.4), mais dans laquelle l'évaluation des nœuds et des feuilles fait appel au réseau de neurones qui prévoit la charge de travail sur chaque secteur ATC.

Critère de coût d'une configuration de secteurs

Une configuration de secteurs est constituée de plusieurs secteurs de contrôle. Le réseau de neurones nous permet d'évaluer la charge de travail sur chacun de ces secteurs, mais il nous faut combiner ces informations en un critère de coût associé à la configuration complète (*i.e.* une feuille de l'arbre exploré par le *Branch&Bound*). Il nous faut également une estimation minorante du coût, pour une partition en cours de construction, lorsqu'on évalue un nœud dans le *Branch&Bound* (voir l'algorithme de *Branch&Bound*, chap. 2, section 2.3).

Pour évaluer une configuration, on considère le nombre de secteurs ATC qu'elle contient, et les valeurs des probabilités p_{low} , p_{normal} , et p_{high} sur chacun de ces secteurs de contrôle. Considérons la configuration $[(\{1\}, s); (\{2, 3\}, a); (\{4, 5\}, c)]$, en reprenant les notations des exemples de ce chapitre. Cette configuration serait parfaitement équilibrée si nous avions $(p_{low}, p_{normal}, p_{high})$ égal à $(0, 1, 0)$ pour les secteurs ATC a , c , et $s = \{1\}$. Une situation idéale comme celle-ci n'arrive jamais en pratique, et on a généralement des secteurs ATC pour lesquels $p_{normal} < 1$, avec éventuellement des surcharges ou des sous-charges sur certains secteurs. L'objectif du partitionnement optimal reste toutefois de se rapprocher le plus possible de la situation idéale, tout en respectant les contraintes opérationnelles.

Le critère de coût a évolué au cours des publications, notamment pour prendre en compte certaines contraintes comme le nombre maximum de positions ouvrables. Le principe reste toutefois toujours le même, à savoir que l'on doit pouvoir comparer deux configurations candidates selon différents critères, considérés par ordre de priorité. Une fois prises en compte les contraintes intangibles (ex. nombre maximum de postes de travail), on souhaite en premier lieu éviter les surcharges de travail, dangereuses pour la sécurité du trafic. Pour des configurations sans surcharges, on privilégiera celles ouvrant le moins de positions de contrôle. A nombre de postes de travail égal, on cherchera à équilibrer au mieux la charge, en minimisant les sous-charges et en cherchant à obtenir des configurations où p_{normal} est aussi proche de 1 que possible dans tous les secteurs ATC.

4.6 Maquette d'IHM et valorisation dans des projets européens (2009-2011)

- [66] D. Gianazza, C. Allignol, and N. Saporito. An efficient airspace configuration forecast. In *Proceedings of the 8th USA/Europe Air Traffic Management R & D Seminar*, 2009.
- [141] N. Saporito, C. Hurter, D. Gianazza, and G. Bebox. A participatory design for the visualization of airspace configuration forecasts. In *Proceedings of the 4th International Conference on Research in Air Transportation*, 2010.
- [65] D. Gianazza. Forecasting workload and airspace configuration with neural networks and tree search methods. *Artificial Intelligence Journal, Elsevier*, 174(7-8) :530–549, may 2010.

Résumé

A partir de 2009, les travaux autour de la prévision de la charge de travail et des ouvertures de secteurs vont progressivement impliquer un plus grand nombre d'intervenants : experts contrôleurs aériens, experts en IHM, programmeurs, et industriels.

La publication [66] présente une première version expérimentale d'une maquette d'IHM, dont le processus de conception participative est décrit dans [141]. Dans [66], on s'intéresse par ailleurs à l'impact des aléas de la prévision de trafic sur la prévision des ouvertures de secteurs. Les travaux précédents calibraient le modèle de charge de travail (le réseau de neurones) sur du trafic observé constitué d'enregistrements radar. Dans le cadre d'un outil prévisionnel, on ne dispose que du trafic planifié. Les résultats de [66] montrent que les écarts entre trafic planifié et trafic réel induisent un écart dans la prévision du nombre de postes de travail. Malgré tout, le nombre prévu de positions de contrôle reste bien plus réaliste que celui du schéma prévisionnel FMP. Cependant, comme la prévision du trafic est d'autant plus précise que l'on se place à court-terme, le cadre d'utilisation des modèles et algorithmes proposés serait finalement plutôt celui d'une prévision en temps réel plutôt que pré-tactique (la veille ou l'avant-veille).

La démarche de conception participative présentée dans [141] permet d'impliquer des experts du monde opérationnel dans la réalisation d'une maquette d'IHM, et plus globalement dans une réflexion sur les utilisations possibles des algorithmes proposés. Le groupe de travail comprend trois contrôleurs aériens, dont un ancien chef de salle, quatre concepteurs d'IHM, dont trois chercheurs, et le concepteur des algorithmes (qui est, incidemment, l'auteur de ce document). Outre l'utilisation dans un contexte pré-tactique (1 à 2 jours avant le jour J) par l'opérateur FMP (*Flow Management Position*) qui motivait à l'origine cette recherche, le groupe identifie une utilisation possible dans un outil tactique de gestion prévisionnelle en temps réel, avec un horizon de prévision à quelques heures. Un tel outil pourrait aider à anticiper les surcharges de travail, et à évaluer l'incidence de mesures correctives sur la charge de travail et sur la configuration de la salle de contrôle. La conception se fait initialement sur papier, et est ensuite poursuivie par un développement itératif en Flex de l'IHM. Le retour des experts opérationnels est positif.

La dernière publication [65] récapitule l'ensemble de la démarche autour de la prévision de la charge de travail et des ouvertures de secteurs, et décrit la version la plus à jour des algorithmes en prenant notamment en compte les contraintes sur le nombre maximum de positions ouvrables.

“Brainstorming” et maquette papier

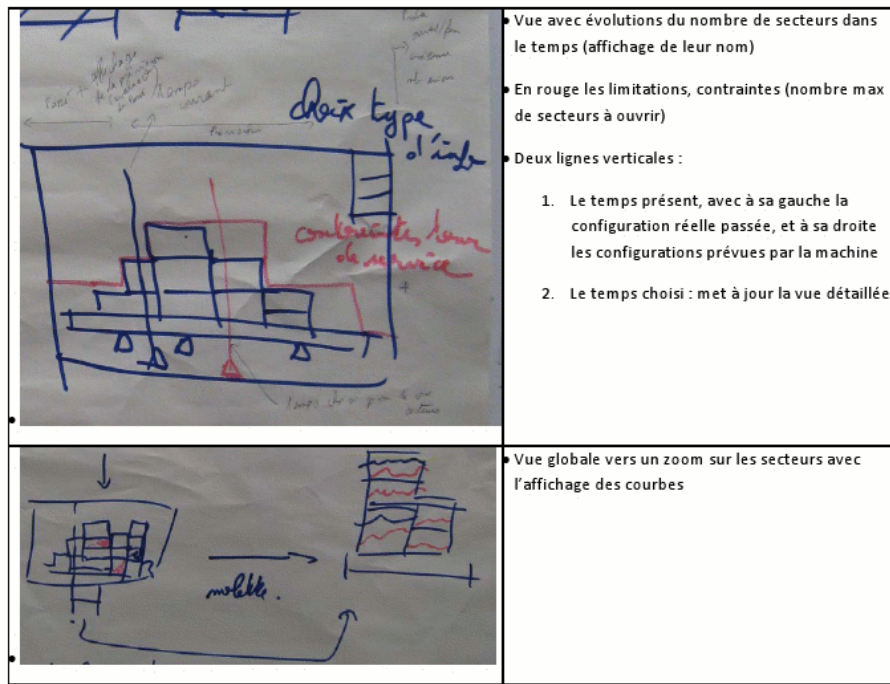


FIGURE 4.9

La réflexion autour d'une utilisation possible des modèles et algorithmes de prévision de charge de travail et de prévision des ouvertures de secteurs fait rapidement apparaître le besoin de hiérarchiser l'information présentée à l'utilisateur. Les experts opérationnels expriment d'une part l'intérêt d'une vue globale, avec une *timeline* et un codage couleur simple pour repérer les surcharges, et d'autre part le besoin de pouvoir “zoomer” sur des configurations ou des secteurs spécifiques pour afficher des informations plus détaillées. Une première esquisse “papier” issue de cette réflexion est présentée sur la Figure 4.9.

Dans un second temps apparaît le besoin de représenter l'évolution des secteurs au cours du temps : lesquels sont la cause d'une décision de repartitionnement ? Comment sont recombinaisonnés les secteurs ? La Figure 4.10 illustre les idées pour représenter ces transitions d'une configuration à l'autre.

Maquette d'IHM

La vue globale fournie par la maquette d'IHM développée en Flex est illustrée sur la Figure 4.11. L'axe des temps du schéma prévisionnel est représenté en bas de la figure, avec au-dessus des empilements de pavés de couleur représentant chacun un secteur de contrôle affecté à un poste de travail. Les contraintes sur le nombre maximum de positions de contrôle ouvrables à chaque instant de la journée sont représentées par des barres de couleur rouge sombre et des nombres.

Le codage couleur des pavés est le suivant. Un pavé plein de couleur rouge indique une surcharge persistante dans le secteur correspondant, et donc une situation potentiellement dangereuse pour la sécurité du trafic aérien. La couleur verte correspond à une charge de

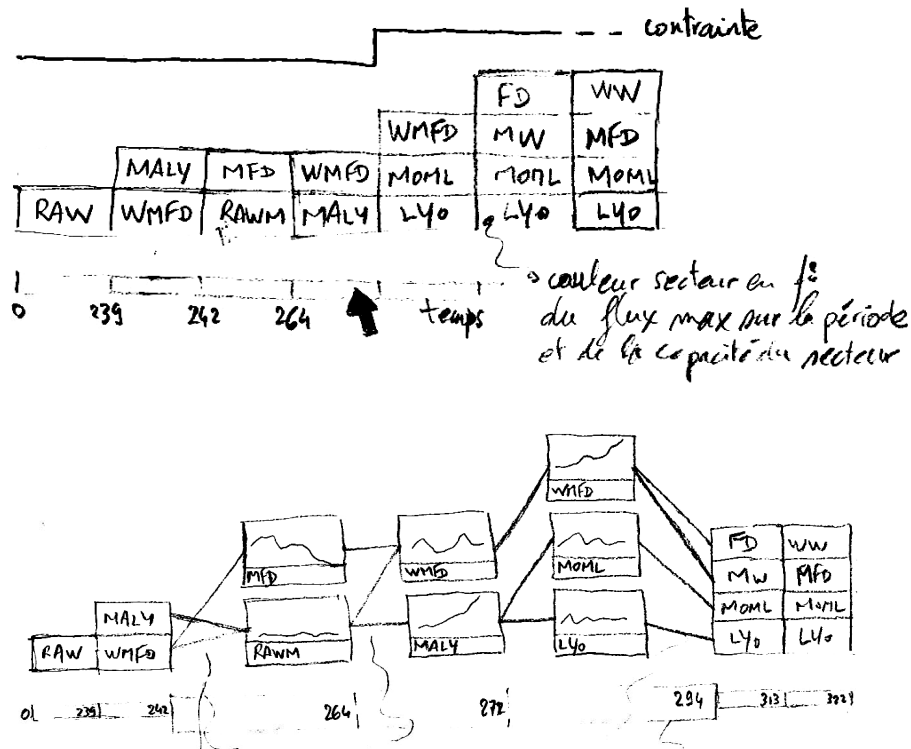


FIGURE 4.10

travail normale, et la couleur bleue à une sous-charge. Les petits demi-disques de couleur situés sur le bord des pavés indiquent quels secteurs sont la cause d'un repartitionnement de l'espace aérien. Ceux de couleur rouges indiquent que c'est un début de surcharge qui a provoqué une reconfiguration des secteurs. Ceux de couleur bleue signifient que c'est une sous-charge qui est à l'origine de la reconfiguration.

Un survol d'une configuration donnée par la souris provoque, comme illustré sur la Figure 4.12, l'affichage d'informations contextuelles telles que le nombre de secteurs ATC et leurs noms, et les heures de début et de fin de la configuration. Un clic sur la configuration fait entrer l'utilisateur dans une vue plus détaillée, illustrée sur la Figure 4.13.

L'utilisateur peut alors cliquer sur les configurations et secteurs de son choix, et afficher ainsi une vue développée des transitions entre configurations, et éventuellement de l'évolution de la charge de travail dans les secteurs sélectionnés. La Figure 4.14 montre un exemple d'une telle vue. L'évolution des trois probabilités p_{low} , p_{normal} , et p_{high} (avec le même codage couleur que pour les pavés) est représentée pour quelques secteurs. On voit que la reconfiguration prévue à 06h26 est causée par un début de surcharge du secteur RFOQ, lequel se retrouve "dégrouper" en ROQ et FBRT dans la configuration suivante. De même à 08h30, le secteur RGA est dégroupé en deux secteurs plus petits suite à un début de surcharge.

Dans cette première itération de la maquette, l'utilisateur ne peut interagir que de façon limitée avec l'IHM, uniquement pour afficher des informations pré-calculées. Il était prévu ensuite, dans le cadre de projets européens du programme SESAR, de réaliser un prototype pré-industriel plus complet, connecté à un vrai prédicteur de trajectoires, et de faire à la demande les calculs des indicateurs de complexité ATC et les prévisions de charge de travail

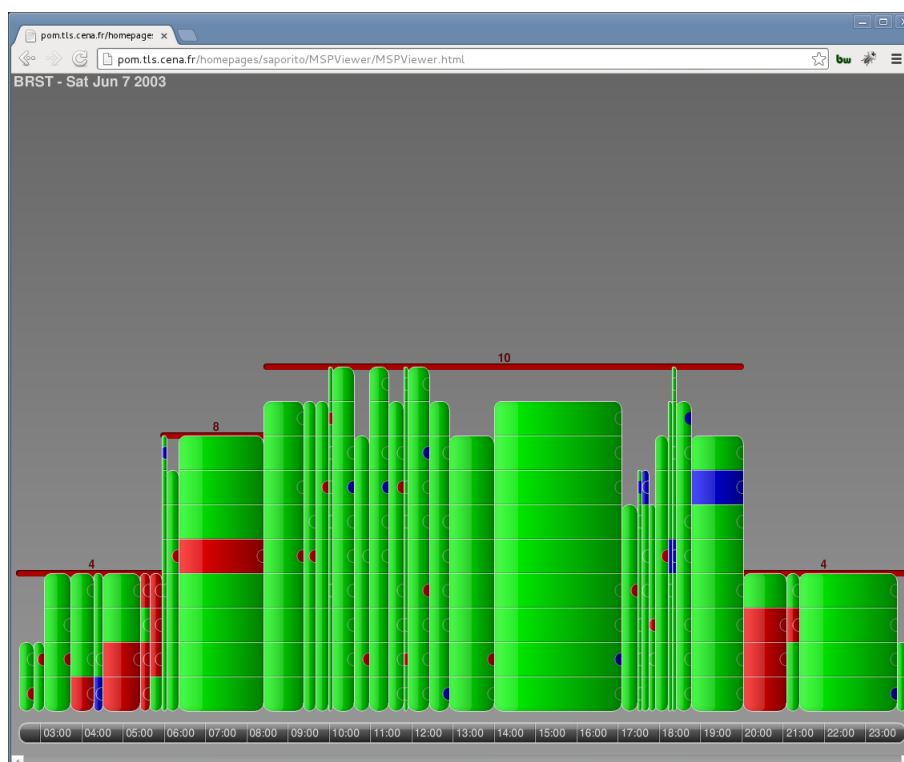


FIGURE 4.11: Vue globale du schéma prévisionnel d'ouverture des secteurs de contrôle aérien.

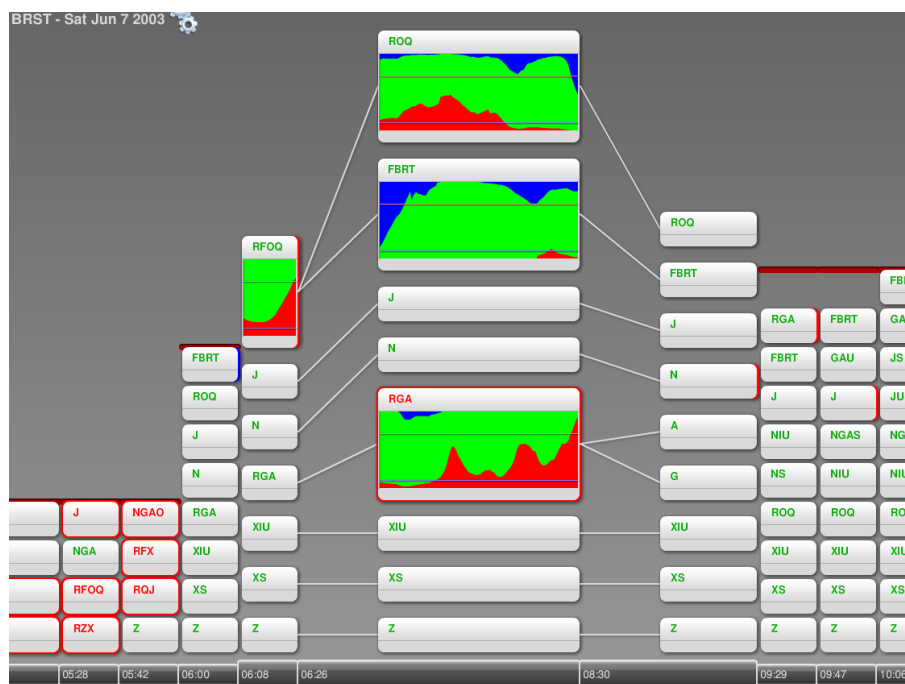


FIGURE 4.14: Zoom sur une configuration de secteurs.

et d'ouvertures de secteurs.

Valorisation dans les projets européens SESAR

Le programme SESAR (Single European Sky Air Traffic Management Research) est le volet technologique du “Ciel Unique Européen” lancé par la Commission européenne. Les objectifs-clés du Ciel Unique Européen sont de restructurer l'espace aérien pour gérer au mieux les flux de trafic (en rendant les frontières nationales les plus transparentes possibles), et d'augmenter la capacité et l'efficacité du système européen de gestion du trafic aérien. Les objectifs chiffrés, particulièrement ambitieux, sont d'augmenter la capacité d'un facteur 3, d'améliorer la sécurité d'un facteur 10, tout en diminuant de 10% les impacts environnementaux et de 50% le coût des services aux usagers (ici, les compagnies aériennes).

Sur le plan technologique, le programme SESAR vise à créer un “changement de paradigme”, supporté par des innovations et des technologies de pointe, pour aider à atteindre les objectifs du Ciel Unique Européen. Contrairement à ce que son nom pourrait laisser penser, SESAR n'est pas à proprement parler un programme de recherche⁴, même s'il comprend quelques projets de “recherche innovatrice”. Son objectif principal est l'augmentation de la capacité et de la sécurité du système européen de gestion du trafic aérien, avec en corollaire l'harmonisation des procédures, des méthodes de contrôle, et des environnements techniques des différents fournisseurs de services du contrôle aérien en Europe. Après une première phase de définition (2006-2008), la phase de développement (2008-2014) pilotée par la SESAR Joint Undertaking doit produire des prototypes pré-industriels des futurs systèmes devant être ensuite produits industriellement et mis en service lors de la phase de déploiement (2015-2025).

4. Cette opinion n'engage que l'auteur du présent document.

La SESAR Joint Undertaking (SJU) est une structure juridique créée par le conseil de l'UE, instituant un partenariat entre la Communauté Européenne, Eurocontrol, et les principaux industriels et opérateurs du secteurs :

- prestataires de services de navigation aérienne : DSNA (France), DFS (Allemagne), ENAV (Italie), NORACON (Europe du Nord et Autriche), AENA (Espagne) et NATS (Royaume-Uni),
- aéroports : SEAC (consortium de 6 grands aéroports), AENA, et NORACON
- fournisseurs d'équipements au sol : Frequentis, Indra, Natmig, SELEX Sistemi Integrati et Thales,
- fabricants d'avions : Airbus et Alenia Aeronautica,
- fabricants d'équipements embarqués : Honeywell et Thales.

Dans ce cadre, les travaux présentés dans ce chapitre ont constitué la proposition de la DSNA France dans trois des projets du programme SESAR :

WP 4.7.1 : “*Complexity management in en-route*”

WP 10.8.1 : “*Complexity assessment and resolution*”

WP 7.5.4 : “*Dynamic airspace configuration*”

Le *Work Package 4.7.1* traite de l'utilisation opérationnelle d'indicateurs de complexité ATC, et le WP 10.8.1 de la réalisation de prototypes pré-industriels des outils informatiques associés. Le sujet du WP 7.5.4 est la configuration dynamique de l'espace aérien, afin de s'adapter au mieux la capacité des centres de contrôle à la demande de trafic.

En particulier, nous avons un partenariat avec l'industriel Thales, leader du WP 10.8.1, pour la réalisation en commun d'un prototype pré-industriel devant permettre de calculer les indicateurs de complexité ATC et d'en déduire une prévision des charges de travail et des ouvertures de secteurs.

En tant que chef du pôle *planification, optimisation, modélisation* de la DSNA/DTI/R&D, j'ai été brièvement *focal point* pour ces projets, dans la phase initiale d'élaboration de la proposition DSNA, avant de déléguer cette tâche à deux ingénieurs chargés spécifiquement de la gestion de ces projets.

Par ailleurs, dans le cadre du partenariat industriel avec Thales et du step 1 du WP 10.8.1, nous étions chargés de la réalisation de trois modules de code : un outil *off-line* de calibration du modèle de prévision de la charge de travail, un module “*complexity assessment*” devant calculer les indicateurs de complexité ATC à partir des positions des avions fournies par les prédicteurs de Thales, et un module “*complexity display*” comprenant l'interface homme-machine permettant d'afficher les prévisions de charge de travail et d'ouvertures de secteurs. Ce partenariat industriel et cette valorisation des travaux de recherche dans les projets SESAR ont été pour moi l'occasion d'encadrer concrètement le travail des deux ingénieurs chargés de développer les modules que nous avions à fournir.

4.7 Discussion et conclusion

Nous avons vu que le problème d'affectation des secteurs d'espace aérien aux postes de travail était un problème *a priori* fortement combinatoire, mais dont la difficulté est amoindrie par l'existence de contraintes opérationnelles (liste de regroupements autorisés, contraintes

dans les transferts de secteurs, etc). Nous avons également vu qu'une prévision réaliste des regroupements de secteurs ne peut se faire sans une estimation fiable de la charge de travail.

Pour des problèmes de partitionnement fortement combinatoires, les métaheuristiques sont souvent la seule option possible : elles utilisent une marche au hasard dans l'espace des solutions possibles, guidée par une heuristique susceptible d'orienter la recherche vers les meilleures solutions. Pour des instances de faible taille, le partitionnement optimal de l'espace peut être obtenu par des méthodes exactes de recherche arborescente. En pratique, pour notre application, ces méthodes exactes se sont finalement montrées les plus efficaces sur les instances traitées (les secteurs des cinq centres de contrôle aérien français).

Pour ce qui concerne le modèle de prévision de la charge de travail, nous avons clairement privilégié dans notre démarche la "lisibilité" du modèle, peut-être un peu au détriment de sa performance. C'est notamment le cas pour la sélection des variables d'entrée. Il nous a paru important d'isoler un sous-ensemble de quelques variables pertinentes simples, clairement compréhensibles pour un utilisateur opérationnel et faciles à coder et à utiliser indépendamment des autres variables candidates.

Plusieurs améliorations sont envisageables, tant pour le modèle de prévision de la charge de travail que pour l'algorithme de prévision des ouvertures de secteurs. Par exemple, nous avons choisi de lisser les indicateurs en entrée pour contrer les trop fortes variations des valeurs de sortie, lesquelles variations provoquaient des décisions trop fréquentes de repartitionner l'espace. Il aurait pu être intéressant d'utiliser un modèle qui prenne en entrée une succession temporelle des valeurs des indicateurs, voire un modèle récurrent où la sortie est rebouclée vers les variables d'entrée. Le choix des variables explicatives candidates aurait pu être étendu à d'autres indicateurs, comme le nombre de conflits potentiels entre trajectoires prévues, à calculer par simulation.

Pour la prévision des ouvertures de secteurs, il était prévu de la rendre plus réaliste en tenant compte des règles opérationnelles de reconfiguration qui limitent les transitions autorisées (ex. AB-C \rightarrow A-BC). D'autres règles de décision auraient pu être envisagées pour déclencher le recalcul du partitionnement de l'espace. Nous avons fait le choix d'un paramétrage des seuils sur les probabilités en sortie du réseau de neurones, et d'un lissage des variables en entrée, pour éviter de déclencher trop fréquemment des repartitionnements de l'espace, mais on aurait aussi pu choisir d'imposer une durée minimale d'ouverture de secteur ATC, et avoir un critère de fréquence sur la surcharge qui évite de déclencher une reconfiguration lors d'une surcharge ponctuelle.

Sur l'ensemble de la démarche, nous avons progressé de façon incrémentale, en explorant parfois des impasses ou en palliant à des problèmes rencontrés au fur et à mesure. Il faudrait sans doute, sur le plan méthodologique, reprendre et améliorer chacun des éléments de nos algorithmes, en élargissant l'analyse à toute une année de trafic pour mieux paramétrer nos algorithmes.

Les approfondissements évoqués ci-dessus n'ont pas été réalisés, suite à une réorganisation, en 2011, de l'activité de Recherche & Développement de la DSNA, et plus généralement des activités de recherche à la DGAC. Les personnels DSNA titulaires d'un doctorat ont été réaffectés à l'École Nationale de l'Aviation Civile, et les autres personnels ont par la suite été relocalisés sur le site de la Direction de la Technique et de l'Innovation. A cette occasion, nombre d'entre eux ont demandé des mutations sur d'autres postes, parmi lesquels les ingénieurs du pôle dont j'étais responsable avant d'être transféré à l'ÉNAC. Suite à ces réorganisations, la contribution de la DSNA France dans les trois projets SESAR a dû être réorientée, et le partenariat industriel avec Thales abandonné.

L'ensemble des travaux que j'ai présenté dans ce chapitre m'auront malgré tout permis de mener sur plusieurs années une démarche complète de recherche appliquée, partant de l'exploration de quelques idées initiales, en passant par une recherche plus approfondie, et presque jusqu'à l'application industrielle et opérationnelle.

Chapitre 5

Application aux trajectoires d'avions

Sommaire

5.1	Chronologie et principales publications associées	154
5.2	Résumé des travaux réalisés	155
5.2.1	Estimation du vent à partir des trajectoires radar	155
5.2.2	Prévision des trajectoires d'avions en montée	156
5.3	Données disponibles	160
5.3.1	Les différents types de données	160
5.3.2	Deux jeux de données : Mode-C et Mode-S	161
5.3.3	Difficultés du travail sur données réelles	162
5.4	Estimation du vent à partir des trajectoires radar	162
5.4.1	Contexte et travaux antérieurs	162
5.4.2	Relation entre vent et vitesses des avions	164
5.4.3	Formulation du problème d'extraction de vent	164
5.4.4	Résolution par une méthode de moindres carrés	166
5.4.5	Procédure de calcul, critères de qualité	166
5.4.6	Résultats	168
5.4.7	Conclusion sur l'extraction du vent	168
5.5	Prévision de trajectoires d'avions en montée	169
5.5.1	Contexte et travaux antérieurs	169
5.5.2	Le modèle à énergie totale	171
5.5.3	La problématique de la prédiction par les systèmes au sol	173
5.5.4	Estimation de la masse	175
5.5.5	L'apprentissage appliqué à la prévision de trajectoires d'avions	183
5.5.6	Cadre expérimental pour les méthodes d'apprentissage	185
5.5.7	Apprentissage direct de l'altitude future	189
5.5.8	Apprentissage de la masse	192
5.5.9	Apprentissage de la commande de poussée	192
5.5.10	Apprentissage des vitesses-cibles CAS et Mach	196
5.5.11	Résultats comparatifs pour la prévision de l'altitude	198
5.5.12	Généralisation à différents types d'avions et différentes altitudes initiales	199
5.6	Conclusion sur l'exploitation des trajectoires d'avion	201

La démarche de recherche présentée dans ce chapitre est centrée sur l'application de méthodes d'estimation ou d'apprentissage supervisé à des jeux de données enregistrées de trajectoires d'avions. Deux idées distinctes sont explorées. La première consiste à extraire des trajectoires observées le vent subi par les avions. La deuxième idée consiste à extraire des trajectoires les valeurs de certains paramètres du modèle physique de l'avion, inconnues des systèmes au sol, avec pour objectif d'améliorer la prévision des trajectoires en montée.

Le travail sur l'estimation du vent s'est fait en collaboration avec Christophe Hurter et Gennady et Natalia Andrienko, pour la partie IHM et visu-analytics, Stéphane Puechmorel pour la définition des critères de qualité et l'implantation en langage C de la version définitive de l'algorithme, et Richard Alligier, dont j'encadrerais par ailleurs la thèse portant sur la prévision de trajectoires, pour le pré-processing des données radar et météorologiques. L'essentiel de ma contribution a porté sur la méthode d'extraction du vent à partir des données radar. Ce travail a donné lieu à deux publications dans des journaux [93, 94]. Les auteurs ne sauraient trop remercier Serge Roux, ingénieur à la DSNA/DTI puis au support à la recherche à l'ÉNAC, pour la collecte des données radar qui sont à la base des travaux présentés dans ce chapitre.

Les travaux sur la prévision des trajectoires d'avions ont été réalisés dans le cadre de la thèse de Richard Alligier ([7]) que j'ai encadrée à 100%, sous la tutelle de Nicolas Durand, directeur de thèse. Ils ont donné lieu à quatre publications dans des conférences internationales avec comité de relecture [8, 9, 13, 12], et à deux publications dans des journaux [10, 11]. La thèse de Richard Alligier [7] a par ailleurs été proposée par le jury pour le prix de thèse Léopold Escande.

5.1 Chronologie et principales publications associées

- 2012

- [8] R. ALLIGIER, D. GIANAZZA AND N. DURAND. *Energy Rate Prediction Using an Equivalent Thrust Setting Profile (regular paper)*. In ICRAT 2012, Proceedings of the 5th International Conference on Research in Air Transportation, May 2012, Berkeley, United States.

- 2013

- [9] R. ALLIGIER, D. GIANAZZA, AND N. DURAND. *Ground-based estimation of aircraft mass, adaptive vs. least squares method*. In Proceedings of the 10th USA/Europe Air Traffic Management R & D Seminar, 2013.
- [10] R. ALLIGIER, D. GIANAZZA, AND N. DURAND. *Learning the aircraft mass and thrust to improve the ground-based trajectory prediction of climbing flights*. Transportation Research Part C : Emerging Technologies, 2013.

- 2014

- [13] R. ALLIGIER, D. GIANAZZA, M. GHASEMI HAMED, AND N. DURAND. *Comparison of Two Ground-based Mass Estimation Methods on Real Data (regular paper)*. In Proceedings of the International Conference on Research in Air Transportation (ICRAT), Istanbul, 26/05/2014-30/05/2014. May 2014. ICRAT.

- [93] C. HURTER, R. ALLIGIER, D. GIANAZZA, S. PUECHMOREL, G. ANDRIENKO, & N. ANDRIENKO. *Wind parameters extraction from aircraft trajectories*. Computers, Environment and Urban Systems, 2014, vol. 47, p. 28-43.
- [94] C. HURTER, S. CONVERSY, D. GIANAZZA, AND AC TELEA. *Interactive image-based information visualization for aircraft trajectory analysis*. Transportation Research Part C : Emerging Technologies, 2014.
- [7] R. ALLIGIER. *Apprentissage artificiel appliqué à la prévision de trajectoire d'avion*. 2014. Thèse de doctorat.

- 2015

- [11] R. ALLIGIER, D. GIANAZZA, N. DURAND. *Machine learning and mass estimation methods for ground-based aircraft climb prediction*. IEEE Transactions on Intelligent Transportation Systems, 2015, vol. PP, issue 99, p.1-12.
- [12] R. ALLIGIER, D. GIANAZZA, AND N. DURAND. *Machine Learning Applied to Airspeed Prediction During Climb*. In Proceedings of the 11th USA/Europe Air Traffic Management R & D Seminar, 2015.

5.2 Résumé des travaux réalisés

5.2.1 Estimation du vent à partir des trajectoires radar

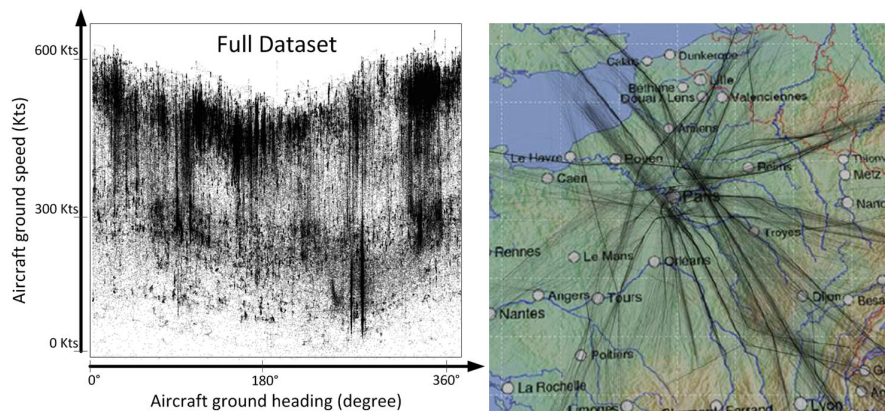


FIGURE 5.1: Une journée de trafic au-dessus de la région parisienne (vue de droite), et tracé de la vitesse-sol en fonction de la direction de la route suivie (à gauche). Les formes sinusoidales émergeant de la représentation des données traduisent l'influence du vent sur la vitesse-sol des avions.

Un des deux axes d'études autour de l'exploitation des données de trajectoires a été l'extraction du vent à partir des trajectoires enregistrées par les radars au sol. L'idée à l'origine de ces travaux est extrêmement simple : en traçant pour un grand nombre d'avions le module de la vitesse-sol en fonction de sa direction, on voit émerger des courbes plus ou moins sinusoidales, comme illustré sur la Figure 5.1. La partie droite de la figure montre les traces radar des avions ayant survolé la région parisienne, et la partie gauche trace la vitesse par rapport au sol (en ordonnée) en fonction de la direction de la route suivie (en abscisse), pour chaque plot radar enregistré.

Dans [93, 94], nous proposons d'une part une méthode permettant d'ajuster une ou plusieurs courbes sinusoïdales sur un nuage de points afin d'en déduire l'intensité et la direction du vent, et d'autre part une procédure interactive permettant à l'utilisateur de filtrer les données de trajectoires, d'évaluer la pertinence du vent extrait par la méthode automatique, et de corriger manuellement les résultats.

Le vent obtenu est comparé à celui fourni par Météo France d'une part, et également, pour le jeu de données du radar Mode-S expérimental de Toulouse, au vent calculé à partir d'informations transmises par l'avion par liaison de données Mode-S (voir section 5.3.1). Les résultats montrent une bonne approximation du vent pour les niveaux de vol en espace supérieur où les avions volent à des vitesses-air sensiblement identiques, et dans des zones de trafic suffisamment denses, avec des avions volant dans des directions diverses.

Les perspectives d'utilisation opérationnelles restent toutefois limitées. Avec le déploiement généralisé des radars Mode-S ou des liaisons de données ADS-B, l'enrichissement des modèles météo par les informations récoltées à bord des avions (voir [41]) semble une piste bien plus prometteuse pour améliorer la précision du vent observé ou prévu. Le méthode que nous proposons ne peut dans ce cadre être envisagée que comme un système alternatif à bas coût pour les zones mal couvertes par les modèles météo et le réseau de liaisons de données.

5.2.2 Prédiction des trajectoires d'avions en montée

Le deuxième axe de recherche autour de la thématique "trajectoire de l'avion" concerne la prédiction de l'altitude de l'avion en montée, par des prédicteurs situés au sol ne disposant que d'informations incomplètes ou imprécises. Pour améliorer cette prédiction, l'idée centrale explorée durant la thèse de Richard Alligier consiste à estimer ou apprendre les valeurs inconnues des paramètres du modèle physique de l'avion, à partir de trajectoires observées. Le court résumé qui suit s'appuie sur un regroupement thématique des différentes publications (conférences ou journaux) issues du travail de thèse de Richard Alligier [7], et ne suit donc pas nécessairement l'ordre chronologique de la section 5.1.

Estimation de la masse de l'avion à partir des points passés

- [9] R. ALLIGIER, D. GIANAZZA, AND N. DURAND. *Ground-based estimation of aircraft mass, adaptive vs. least squares method*. In Proceedings of the 10th USA/Europe Air Traffic Management R & D Seminar, 2013.
- [13] R. ALLIGIER, D. GIANAZZA, M. GHASEMI HAMED, AND N. DURAND. *Comparison of Two Ground-based Mass Estimation Methods on Real Data (regular paper)*. In Proceedings of the International Conference on Research in Air Transportation (ICRAT), Istanbul, 26/05/2014-30/05/2014. May 2014. ICRAT.

L'idée explorée dans ces deux publications [9, 13], a en fait été introduite dans une publication antérieure [8] sur l'apprentissage du profil de poussée, que nous verrons un peu plus loin. Cette idée consiste à estimer la masse d'un avion pendant sa phase de montée, à partir des informations extraites des points passés de sa trajectoire (variations d'altitudes, de vitesse, etc) et des informations météorologiques (vent, température). En fixant le profil de poussée, le paramètre masse est ajusté de façon à minimiser l'erreur entre la puissance modélisée et les variations observées d'énergie cinétique et potentielle de l'avion. Cette nouvelle méthode, proposée dans [8], est améliorée dans [9] pour tenir compte de la consommation de carburant. Elle minimise l'erreur quadratique entre la prédiction du modèle et l'observation,

sur un segment de points passés de la trajectoire en tirant parti de la forme polynomiale de la puissance modélisée (vue comme une fonction de la masse) par le modèle Base of Aircraft DATA (BADA) d'Eurocontrol.

Simultanément à nos premiers travaux [8] en 2012, Schultz, Thipphavong et Erzberger ont proposé une autre méthode [145], basée sur un mécanisme adaptatif. Cette méthode ajuste incrémentalement la masse, selon l'erreur constatée entre la puissance modélisée et la variation d'énergie observée, et en tenant compte d'un paramètre auto-adaptatif de sensibilité.

L'objectif des deux publications [9, 13] est de comparer notre méthode de moindres carrés avec la méthode adaptative de Schultz *et al.*, d'abord sur des données simulées, dans [9], puis sur des données réelles, dans [13]. Dans [9], la robustesse des deux approches aux erreurs artificiellement introduites sur les variables d'état de l'avion est évaluée en ajoutant un bruit paramétrable aux valeurs prises par différentes variables. Les deux méthodes se montrent capables de retrouver avec une très bonne précision les masses ayant servi au calcul des données simulées, avec des performances légèrement meilleures pour la méthode minimisant l'erreur quadratique.

La publication [13] compare également les deux méthodes, mais cette fois-ci sur des données réelles. Dans ce cadre, on fait l'hypothèse d'une montée à poussée *max climb* et on suppose parfaitement connu le profil des vitesses lors de la montée, afin de n'évaluer que l'influence de l'estimation de la masse. Sur données réelles, la méthode adaptative de Schultz *et al.* [145] se montre très légèrement plus performante, suggérant que la méthode de moindres carrés est moins robuste aux valeurs aberrantes. Comparées au modèle BADA utilisé avec la masse de référence, les deux méthodes montrent un gain de l'ordre de 45 % sur la Root Mean Square Error (RMSE) de l'altitude prévue à 10 minutes. Notons que ce gain n'est pas représentatif d'une vraie prévision, puisque la comparaison se fait ici en utilisant le profil observé de vitesse en montée, qui n'est pas connu *a priori* dans le cadre d'une prévision réelle.

Apprentissage de la loi de poussée des moteurs

- [8] R. ALLIGIER, D. GIANAZZA AND N. DURAND. *Energy Rate Prediction Using an Equivalent Thrust Setting Profile (regular paper)*. In ICRAT 2012, Proceedings of the 5th International Conference on Research in Air Transportation, May 2012, Berkeley, United States.
- [10] R. ALLIGIER, D. GIANAZZA, AND N. DURAND. *Learning the aircraft mass and thrust to improve the ground-based trajectory prediction of climbing flights*. Transportation Research Part C : Emerging Technologies, 2013.

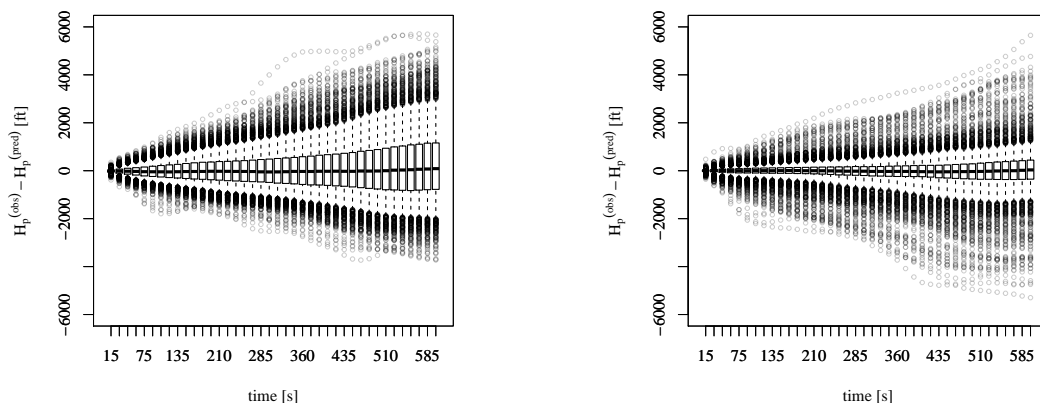
Le sujet principal des publications [8] et [10] est l'apprentissage d'un profil de poussée commun à tous les avions de même type au départ d'une même *Terminal Manœuvring Area (TMA)*. On fait ici l'hypothèse que les montées d'avions de même type possèdent des caractéristiques communes, que l'on cherche à apprendre à partir d'exemples.

Dans [8], le profil de poussée est appris sur une base d'exemples de trajectoires, avec une méthode *ad-hoc* basée sur un algorithme BFGS (voir chap. 2, section 2.4, et l'annexe A). A chaque itération de la méthode BFGS, l'évaluation de la fonction-objectif fait appel à la méthode d'estimation de masse évoquée précédemment, pour chacun des avions de la base d'exemples. Le profil retenu est celui qui minimise l'erreur de prévision sur la variation d'énergie pour l'ensemble des exemples de trajectoires. Une fois appris, le profil de poussée peut ensuite être utilisé, toujours avec la méthode d'estimation de la masse, pour prévoir les

variations d'énergie d'un nouvel avion du même type, dont on observe le début de la montée au départ de la même TMA. Les entrées du modèle de prévision sont alors simplement les premiers points de la nouvelle trajectoire et les informations météo. Les résultats montrent que, par rapport au modèle BADA de référence, la méthode d'estimation des masses apporte à elle seule une réduction de 45 % sur la RMSE des variations d'énergie sur un horizon de 10 minutes, pour un Airbus A320 en montée de la région parisienne passant le niveau de vol FL180. L'utilisation du profil de poussée appris apporte un gain supplémentaire de 5 %, portant à 50 % la réduction de la RMSE sur la variation d'énergie.

La publication [10] dans le journal *Transportation Research Part C* reprend en détail les modèles et algorithmes introduits dans [8, 9] et montre, sur des données réelles, l'apport de l'estimation de la masse et de l'apprentissage du profil de poussée pour la prévision des altitudes futures de l'avion. Par rapport à [8] où les résultats concernaient uniquement l'erreur sur la variation d'énergie aux points "futurs" (observés) des trajectoires enregistrées, nous faisons dans [10] une vraie prédiction des points futurs, en tenant par ailleurs compte du gradient de vent (ce qui n'était pas le cas dans [8]). Cette prévision nécessite, en plus de la masse et du profil de poussée, de faire une hypothèse sur les intentions en vitesse de l'avion. On suppose par exemple que l'avion suit un profil standard de montée à vitesse Calibrated AirSpeed (CAS) constante, puis à nombre de Mach constant (voir l'annexe F sur les différentes vitesses de l'avion).

Sur un jeu de trajectoires suivant des profils CAS/Mach constants l'utilisation de la masse estimée et de la loi de poussée apprise sur les exemples améliore de 41 à 48 % la RMSE sur l'altitude prévue à 10 minutes, par rapport aux profils de poussée BADA (*max climb* ou *reduced power*) combinés avec la masse de référence BADA. La Figure 5.2 montre les erreurs d'altitude à chaque temps t dans un horizon de prévision de 10 minutes maximum, pour une montée standard à poussée *max climb* avec la masse de référence BADA (à gauche), ou avec la masse estimée et en suivant le profil de poussée appris sur les exemples (à droite).



(a) Poussée $BADA_{max\ climb}$ et masse m_{BADA} de référence. (b) Masse estimée $m_{estimated}$ et loi de poussée $f(\cdot|\theta^*)$ apprise sur les exemples.

FIGURE 5.2: Erreurs d'altitude pour des temps futurs entre 15 s et 10 mn.

Sur la Figure 5.2, l'axe des abscisses représente le temps t dans l'horizon de prévision, et l'axe des ordonnées représente la différence entre l'altitude observée sur la trajectoire réelle et celle prévue au temps t par le modèle. Les boîtes au centre des cônes d'incertitude représentent les distances inter-quartiles 0.25/0.75, et les moustaches autour de chaque boîte sont définies de façon à laisser 5 % des données en-dessous de la frontière inférieure, et 5 % au-dessus de la frontière supérieure. On voit que les cônes d'incertitude sont mieux centrés autour de zéro et plus étroits, en utilisant notre méthode.

Apprentissage d'un modèle fournissant la masse de l'avion

- [11] R. ALLIGIER, D. GIANAZZA, N. DURAND. *Machine learning and mass estimation methods for ground-based aircraft climb prediction*. IEEE Transactions on Intelligent Transportation Systems, 2015, vol. PP, issue 99, p.1-12.

L'idée est ici de construire dans un premier temps des exemples associant des variables explicatives extraites des points "passés" à une variable réponse qui est la masse ajustée sur les points "futurs" de chaque trajectoire de notre base d'exemples, puis dans un deuxième temps d'apprendre un modèle qui prévoit cette masse ajustée, en utilisant une méthode d'apprentissage supervisé (voir chapitre 3). Les points "passés" sont simplement ceux antérieurs au temps t_0 du passage d'une altitude donnée, correspondant au moment auquel on souhaite réaliser la prévision des altitudes futures. Les points "futurs" sont donc les points observés situés au-dessus de cette altitude.

Une fois appris, ce modèle peut être utilisé sur de nouvelles trajectoires, pour fournir une masse qui est ensuite utilisée avec le modèle physique de l'avion pour prévoir l'altitude. Dans la phase d'apprentissage comme dans la phase de prévision, on fixe la commande de poussée en faisant ici une hypothèse de poussée *max climb*.

Cette approche est comparée, sur 9 types d'avions différents, avec les méthodes d'estimation de la masse à partir des seuls points passés, et avec la méthode de référence utilisant la masse BADA par défaut. En supposant connu le profil de vitesses afin d'examiner uniquement l'influence de la masse, l'apprentissage supervisé de cette masse permet d'améliorer l'erreur RMSE sur l'altitude prévue à 10 minutes de 58 % à 92 % selon le type d'avion, par rapport à la méthode de référence. L'amélioration reste significative (de 28 % à 52 %) si l'on compare aux résultats obtenus avec les masses estimées à partir des seuls points passés.

En se plaçant dans un cadre plus réaliste pour la prévision, c'est-à-dire avec un profil BADA par défaut pour la vitesse-cible suivie par l'avion pour les points futurs de la trajectoires, les résultats varient selon la catégorie d'avion et selon que le réalisme de l'hypothèse sur la vitesse. Pour les 6 types d'avions ayant un profil CAS/Mach par défaut assez proche de la réalité, le gain apporté par la méthode d'apprentissage de la masse va de 46 % à 86 %, par rapport à la référence, et de 17 % à 49 %, par rapport aux méthodes d'estimation. Parmi les 3 types d'avions posant problème, la mauvaise modélisation de la vitesse annule le bénéfice de l'apprentissage ou de l'estimation de la masse pour un seul d'entre eux (E145). Pour les deux autres, on constate tout de même une amélioration, même si la vitesse est mal modélisée. Pour le Fokker 100, les résultats avec les méthodes d'estimation ou d'apprentissage sont équivalents, et 30 % meilleurs que la référence. Pour le Boeing 747-400, l'utilisation de l'apprentissage supervisé réduit la RMSE de 85 % par rapport à la référence, et de 30 % par rapport à la méthode d'estimation de la masse.

Apprentissage de modèles fournissant le profil de vitesse

- [12] R. ALLIGIER, D. GIANAZZA, AND N. DURAND. *Machine Learning Applied to Airspeed Prediction During Climb*. In Proceedings of the 11th USA/Europe Air Traffic Management R & D Seminar, 2015.

Dans [12], la même démarche d'apprentissage supervisé est appliquée, mais avec les vitesses-cibles CAS et Mach comme variables réponses. On suppose que la montée s'effectue à vitesse CAS constante, puis à nombre de Mach constant (voir annexe F). Afin de construire les exemples pour l'apprentissage, les valeurs CAS et Mach sont ajustées sur les points "futurs" des exemples de trajectoires. Des modèles retournant ces variables réponses sont ensuite appris, ici avec la méthode GBM (voir chap. 3, section 3.9) sur les exemples ainsi construits.

Les valeurs de CAS et de Mach retournées par ces modèles sont ensuite utilisées avec le modèle physique de l'avion, à la place des valeurs par défaut du modèle BADA habituellement utilisées dans les prédicteurs actuels.

La comparaison avec la méthode de référence utilisant les valeurs par défaut de BADA, sur 9 types d'avions différents, montre une réduction de l'erreur RMSE sur la prévision du profil de vitesse allant de 36 % à 79 %, ce dernier gain étant atteint pour le type d'avion E145 qui posait problème dans [11]. La combinaison des méthodes d'apprentissage des vitesses-cibles et de la méthode d'apprentissage de la masse introduite dans [11] permet d'améliorer la RMSE sur altitude prévue à 10 minutes d'au moins 45 %, par rapport à la méthode BADA de référence. Ce gain atteint 87 % pour le type B772.

Ceci conclut ce résumé sur la démarche de recherche autour de la thématique "trajectoires des avions". La suite du chapitre présente ces travaux de façon plus détaillée, après une courte description, dans la section 5.3, des données ayant servi dans ces travaux. La section 5.4 décrit la méthode d'estimation du vent à partir des données de trajectoires, et la section 5.5 donne plus en détails sur la prévision de trajectoires d'avions en montée, laquelle est par ailleurs traitée de manière plus fouillée dans la thèse de Richard Alligier [7].

5.3 Données disponibles

5.3.1 Les différents types de données

Les données disponibles sont de plusieurs types : enregistrements radar, plans de vol déposés par les compagnies, données météorologiques.

Données radar Mode-C et Mode-S

Pour la gestion du trafic aérien, on distingue principalement deux technologies de radars : le radar primaire et le radar secondaire. Le radar primaire émet une onde électromagnétique, qui est réfléchiée par l'avion. Cet écho radar est capté par les stations-sol et permet de situer l'avion en azimut et en distance, voire en élévation pour certains radars primaires militaires. Ce type de radar, qui ne nécessite aucune réponse de la part de l'avion, est aujourd'hui essentiellement utilisé pour des besoins militaires. Le radar secondaire émet également une onde électromagnétique, mais celle-ci interroge un "transpondeur" situé à bord de l'avion. En réponse, cet équipement transmet un signal qui permet de localiser l'avion en azimut et en distance, et qui contient par ailleurs un certain nombre d'informations utiles. C'est ce type

de technologie qui est aujourd’hui très majoritairement utilisé pour les besoins du contrôle aérien.

La nature et la quantité d’informations transmises de l’avion vers le sol dépendent des équipements à bord de l’avion, des types de radars déployés. Dans le cas du radar Mode-C, l’avion transmet essentiellement son identité, sous forme d’un “code transpondeur”, et son altitude-pression numérisée avec une granularité de 100 pieds.

Le radar Mode-S est une amélioration du radar Mode-C. L’altitude-pression de l’avion est numérisée plus finement, avec une granularité de 25 pieds. Le Mode-S EHS (*enhanced*) permet de récupérer d’autres informations sur l’état de l’avion, comme sa vitesse-air, son nombre de Mach, son angle d’inclinaison, etc. Notons toutefois que les informations aujourd’hui transmises ne sont que très partielles. La masse de l’avion n’y figure pas, ni les paramètres de la poussée des moteurs. Seul l’état courant de l’avion (position, vitesse, etc, à l’instant présent) est transmis, mais pas les intentions sur la trajectoire future (vitesse-cible, loi de poussée future).

Données plan de vol

Un plan de vol contient un certain nombre d’informations essentielles, comme l’identifiant commercial du vol, l’aéroport de départ et celui d’arrivée, la route suivie, l’heure de départ prévue, le niveau de croisière demandé, etc. Plusieurs instances du plan de vol peuvent être archivées, à différents moments du vol : une version initiale, une autre qui tient compte des mesures éventuelles de régulation du trafic, et une version finale mise à jour selon le déroulement réel du vol, archivée après la fin de celui-ci.

Données météorologiques

Les données météorologiques sont issues du modèle de prévision ALADIN opérationnel à Météo-France. Ce modèle est couplé au modèle ARPEGE qui couvre l’ensemble du globe avec une résolution variable. Le modèle ALADIN est constitué d’une grille couvrant toute la France. Les mailles de cette grille sont espacées de 0.1 degré en latitude pour les mailles Est/Ouest et en longitude pour les mailles Nord/Sud. Verticalement, les grilles sont situées sur des isobares qui ne sont pas espacées de manière régulière. Toutes les 6 heures, les prévisions météorologiques sont mises à jour. Concernant nos fichiers, on a 10 isobares ; l’isobare la plus basse est l’isobare 1 000 hPa et la plus haute est l’isobare 250 hPa. On dispose de deux échéances de prévisions, l’échéance zéro et l’échéance à 3 heures. L’échéance zéro correspond aux grandeurs prévues en chaque nœud de la grille à la date de prévision. L’échéance à 3 heures correspond aux grandeurs prévues en chaque nœud de la grille à la date de prévision plus 3 heures.

5.3.2 Deux jeux de données : Mode-C et Mode-S

On dispose de deux jeux de données pour tester l’estimation des paramètres, et pour construire les exemples utilisés dans les méthodes d’apprentissage : un jeu de trajectoires Mode-C de la région parisienne, et un jeu de trajectoires Mode-S du radar expérimental de Toulouse.

Le jeu de données Mode-C contient deux mois (juillet 2006 et janvier 2007) d’enregistrements radar du Centre en Route de la Navigation Aérienne (CRNA) Nord, à raison d’un plot radar toutes les 1 à 3 secondes. Ces enregistrements radars sont complétés par les plans

de vol correspondants et les données météorologiques des journées enregistrées. Ces dernières permettent d'associer à chaque plot radar un vent et une température en interpolant dans une grille météorologique. Ceci permet de calculer des informations complémentaires qui ne sont pas disponibles directement dans les données Mode-C, comme la vitesse-air, la CAS, ou le nombre de Mach. Ces données étant calculées à partir d'une prévision des champs de vent et de température, elles sont certainement moins précises que ce qui aurait pu être mesuré à bord de l'avion.

Le jeu de données Mode-S est constitué de 245 jours d'enregistrements du radar expérimental Mode-S de la Direction de la Technique et de l'Innovation (DTI) de Toulouse, répartis entre février 2011 et décembre 2012. Pour chaque trajectoire, on observe en moyenne un plot toutes les 4 secondes, avec parfois des «trous» pour lesquels une série de plots est manquante. Ces «trous», dûs à une limitation du débit de la liaison entre le radar et le site de l'ÉNAC, limitent grandement le nombre de trajectoires exploitables. Comme pour les trajectoires Mode-C, chaque plot est daté et contient la position et la vitesse de l'avion. À ces informations viennent s'ajouter d'autres grandeurs issues des équipements à bord de l'avion : l'inclinaison, le cap magnétique, le taux de virage, la vitesse verticale inertielle, la vitesse verticale barométrique, la vitesse sol, la True AirSpeed (TAS), l'Indicated AirSpeed (IAS) et le Mach . Dans ce qui suit, on considère l'IAS égale à la CAS.

5.3.3 Difficultés du travail sur données réelles

La collecte et le pré-traitement de ces données réelles ont constitué des écueils dans le déroulement de la thèse de Richard Alligier ([7]). En début de thèse, nous disposions d'un jeu initial de données Mode-C, qui avait été fourni par l'ancien service R&D de la DTI, déjà pré-traitées, mais qu'il a fallu reconstituer entièrement en cours de thèse. En effet, la dérivée de la vitesse donnait des valeurs numériques aberrantes avec la technique qui avait été choisie précédemment pour lisser les trajectoires brutes. Or l'accélération est un élément essentiel dans l'équation (5.10) qui est à la base du travail présenté ici. Ensuite, la collecte, sur une période plus étendue que pour les données Mode-C, de données Mode-S que nous espérions *a priori* plus précises et plus riches s'est compliquée par la nécessité de traiter les plots manquants dûs aux problèmes de débit sur un raccordement au réseau opérationnel que nous ne maîtrisions pas.

Nous ne détaillerons pas ici le traitement et l'analyse de ces données, qui peuvent être trouvés dans [7]. Au final, nous disposons donc de deux jeux de trajectoires exploitables, Mode-C et Mode-S, utilisés dans le cadre de la prévision de trajectoires pour tester les méthodes d'estimation et d'apprentissage supervisé que nous allons décrire dans la section suivante, et aussi pour l'estimation du vent décrite en section 5.4.

5.4 Estimation du vent à partir des trajectoires radar

5.4.1 Contexte et travaux antérieurs

Les avions volent dans une masse d'air, elle-même en déplacement au-dessus de la surface du globe terrestre. Cette simple constatation montre l'importance cruciale de connaître le vent pour pouvoir naviguer d'un point à un autre dans une machine volante, et pour prévoir correctement la trajectoire future de l'avion.

Prévoir précisément les trajectoires futures est devenu de plus en plus nécessaire, avec l'apparition de concepts opérationnels basés sur les trajectoires-4D [38, 156], et avec l'introduction d'outils automatisés de détection et de résolution des conflits de trajectoires. Les algorithmes de détection et de résolution, qui opèrent sur un horizon de prévision à moyen-terme de l'ordre de 10 à 20 minutes, sont particulièrement sensibles aux incertitudes de trajectoires (voir [50]), et une des causes importantes de ces incertitudes est due à la prévision du vent. Aujourd'hui, les modèles météorologiques existants ne réalisent pas de prévision aux échelles de temps considérées (de l'ordre du quart d'heure). La meilleure solution à ce jour semble être de prendre le vent connu à l'instant t_0 où l'on fait la prévision de trajectoire, et de supposer ce vent constant sur l'intervalle de prévision de la trajectoire future.

Estimer numériquement le vent présent reste toutefois un problème compliqué. Les modèles d'assimilation des météorologues doivent être alimentés par des mesures effectuées par divers moyens (ballons-sondes, radar profileurs de vent, etc.). Ces mesures échantillonnent l'état de l'atmosphère avec une fréquence relativement peu élevée (notamment pour les ballons-sondes), en un nombre assez faible de points au regard du volume de l'atmosphère. Les données recueillies doivent ensuite être traitées par des modèles météorologiques complexes, avec des temps de traitement parfois longs. Aujourd'hui, la fréquence de rafraîchissement de l'information de vent dans les centres de contrôle aérien est de l'ordre de toutes les 3 à 6 heures, au mieux 1 heure au U.S.A.

Dans les travaux présentés dans cette section 5.4, les avions sont utilisés comme des capteurs passifs afin d'estimer le vent dans les volumes d'atmosphère qu'ils traversent. Chercher à extraire le vent des trajectoires d'avions n'est pas une idée nouvelle. Dans [91], Hollister, Bradford, et Welch estiment le vent à partir de la trajectoire d'un avion volant en cercle, en considérant la variance de la vitesse-sol. Dans [42], Delahaye et Puechmorel utilisent un filtre de Kalman étendu pour estimer le vent à partir de trajectoires simulées. La méthode est basée soit sur l'observation d'un seul avion en croisière effectuant au moins deux virages, soit sur deux avions effectuant un virage chacun. La vitesse de chaque avion dans la masse d'air est supposée constante, ainsi que le taux de virage. Le calcul s'appuie sur le "triangle des vitesses" (voir Figure 5.3), la relation vectorielle reliant la vitesse-air, la vitesse-sol, et le vent. Une approche par filtrage particulière est proposée par Lymperopoulos et Lygeros dans [111] pour le problème de prévision des trajectoires de plusieurs avions, en modélisant l'erreur sur le vent comme une variable aléatoire. Une méthode séquentielle de Monte Carlo est utilisée pour affiner l'erreur sur le vent, et améliorer la prévision des trajectoires. Les vitesses-air sont supposées connues et la méthode est testée sur des données simulées uniquement.

Contrairement à [91, 42], notre approche ne se focalise pas sur quelques avions en virage. Nous proposons au contraire d'exploiter la masse des données de trajectoires, avec de nombreux avions volant dans des directions diverses, pour estimer le vent. Pour cela, nous supposons l'existence de catégories d'avions volant à des vitesses-air identiques. Cependant la vitesse-air des avions n'a pas à être connue, comme c'était le cas dans [111]. Par ailleurs, notre étude porte sur des données réelles, et non simulées comme dans [42, 111].

Dans notre modélisation, nous proposons quelques hypothèses simplificatrices raisonnables qui permettent de réduire drastiquement le nombre d'inconnues dans la formulation du problème d'extraction de vent, que nous traitons ensuite par une méthode de moindres carrés ordinaires. Les sections qui suivent décrivent la modélisation et la méthode utilisée, ainsi que les résultats obtenus.

5.4.2 Relation entre vent et vitesses des avions

Pour la formulation mathématique du problème d'extraction du vent à partir des trajectoires radar, nous noterons V_{g_i} la vitesse de l'avion i par rapport au sol (*ground speed*), et V_{a_i} sa vitesse par rapport à la masse d'air (*true airspeed*). Dans ce qui suit, nous supposons que les avions sont stabilisés à leur niveau de vol, ou que leur pente de montée ou de descente est suffisamment faible pour que l'on puisse considérer que la projection $V_{a_i/x_h, y_{h_i}}$ de la vitesse-air dans le plan horizontal soit approximativement égale à V_{a_i} . L'angle entre la route suivie par l'avion i et la direction du Nord géographique est noté Ψ_{g_i} . Le cap de l'avion, c'est-à-dire l'angle entre son axe longitudinal et le Nord, est noté Ψ_{a_i} . On fait ici l'hypothèse que l'avion n'est pas en dérapage et effectue un vol symétrique. Nous noterons W l'intensité du vent, et Ψ_w sa direction.

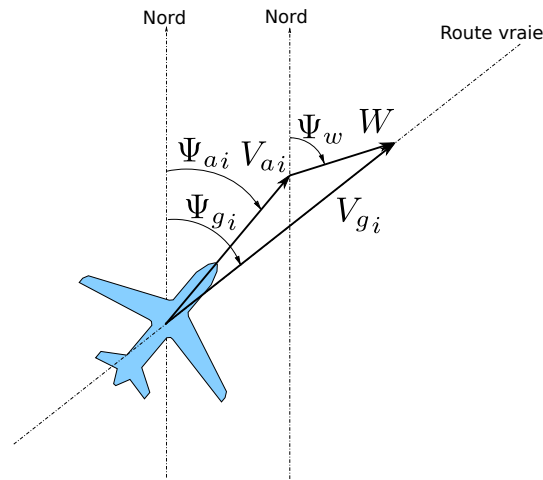


FIGURE 5.3: Triangle des vitesses.

La relation entre la vitesse-air, la vitesse-sol, et le vent est illustrée sur la Figure 5.3 et s'exprime simplement par la relation vectorielle donnée par l'équation (5.1) suivante :

$$\vec{V}_{g_i} = \vec{V}_{a_i} + \vec{W} \quad (5.1)$$

En projetant les vecteurs sur l'axe de la route vraie suivie par l'avion, nous obtenons l'équation (5.2) :

$$V_{g_i} = V_{a_i} \cos(\Psi_{a_i} - \Psi_{g_i}) + W \cos(\Psi_w - \Psi_{g_i}) \quad (5.2)$$

Dans cette équation, les quantités V_{g_i} et Ψ_{g_i} peuvent être observées par les mesures radar des positions successives de chaque avion i . Par contre, les quantités V_{a_i} , Ψ_{a_i} , W , et Ψ_w concernant la vitesse-air de chaque avion et le vent ne sont pas connues, si l'on considère l'avion comme un capteur passif qui ne transmet aucune information au sol.

5.4.3 Formulation du problème d'extraction de vent

Notre objectif est de déduire l'intensité W et la direction Ψ_w du vent, à partir d'un ensemble de mesures $(\Psi_{g_i}(t), V_{g_i}(t))$ effectuées à différents temps t dans une période donnée, et

pour plusieurs avions i . On fait l'hypothèse que tous les vols d'une même catégorie, c'est-à-dire ayant des performances aérodynamiques et de motorisation similaires, volent à une vitesse-air moyenne identique. Bien entendu, en réalité, il existe des disparités au sein d'une même catégorie, sachant que les compagnies peuvent choisir différentes vitesses de croisière en fonction de leur politique de gestion des coûts. On supposera tout de même que, statistiquement, la vitesse-air est sensiblement identique pour une même catégorie d'avions, même s'il peut y avoir une certaine dispersion des vitesses.

Par ailleurs, pour des avions volant à des vitesses élevées, l'angle de dérive ($\Psi_{a_i} - \Psi_{g_i}$) est relativement petit. Pour un avion volant à $V_a = 450$ nœuds, avec un vent de travers de 70 nœuds, l'erreur faite en considérant que $\cos(\Psi_{a_i} - \Psi_{g_i}) \approx 1$ est d'environ 1 % de la vitesse-air. Cette erreur est en tout état de cause très vraisemblablement inférieure à la dispersion des vitesses au sein d'une même catégorie d'avions.

En tenant compte de ces remarques, on peut simplifier l'équation (5.2) exprimant la relation entre les vitesses et le vent, en remplaçant le terme $V_{a_i} \cos(\Psi_{a_i} - \Psi_{g_i})$ par une unique vitesse moyenne \bar{V} , pour tous les vols i appartenant à une même catégorie. Pour un vol i dont la vitesse-sol mesurée par le radar à l'instant t est $V_{g_i}(t)$ et la route vraie est $\Psi_{g_i}(t)$, on obtient alors une formulation simplifiée de notre problème d'extraction de vent, en considérant que W et Ψ_w restent constant sur l'intervalle de temps choisi pour les observations :

$$V_{g_i}(t) = \bar{V} + W \cos(\Psi_w - \Psi_{g_i}(t))$$

avec $\bar{V} \in \{\bar{V}_k | k \in \{1, \dots, c\}\}$ la vitesse moyenne dans la catégorie de l'avion i . (5.3)

La variable \bar{V} peut être considérée comme une inconnue de notre problème, au même titre que W et Ψ_w . Le seul pré-requis est que chaque vol i doit être affecté à une des classes existantes $\{1, \dots, c\}$, qui doivent être déterminées à l'avance, en fonction par exemple de la vitesse de croisière "théorique" de l'avion. Cette vitesse théorique de croisière peut être obtenue par un modèle des performances des moteurs et des performances aérodynamiques de l'avion. Un exemple d'un tel modèle, extrait de données fournies par les constructeurs, est le modèle BADA d'Eurocontrol ([3]).

Notre modélisation du problème d'extraction du vent, donnée par l'équation (5.3), peut être linéarisée en introduisant deux nouvelles variables $W_X = W \cos \Psi_w$ et $W_Y = W \sin \Psi_w$, et sachant que $\cos(\Psi_w - \Psi_{g_i}(t)) = \cos \Psi_w \cos \Psi_{g_i}(t) + \sin \Psi_w \sin \Psi_{g_i}(t)$:

$$V_{g_i}(t) = \bar{V} + W_X \cos \Psi_{g_i}(t) + W_Y \sin \Psi_{g_i}(t) \quad (5.4)$$

Dans cette équation, les valeurs numériques pour $V_{g_i}(t)$, $\cos \Psi_{g_i}(t)$ et $\sin \Psi_{g_i}(t)$ sont données par l'observation du vecteur vitesse-sol, par des mesures radar. On voit donc que l'équation (5.4) est linéaire par rapport aux variables \bar{V} , W_X , et W_Y .

Cette simplification de notre formulation initiale réduit drastiquement le nombre d'inconnues dans notre problème. Avec l'équation (5.2), les variables Ψ_{a_i} étaient des inconnues du problème. Nous avons donc au moins une nouvelle variable pour chaque segment rectiligne de trajectoire de chaque avion. Avec les approximations faites en négligeant l'angle de dérive et en supposant l'existence une vitesse moyenne commune, nous avons réduit le nombre d'inconnues à 3 variables W_X , W_Y , et \bar{V} , si l'on ne considère qu'une seule catégorie d'avions, ou à $c + 2$ variables W_X , W_Y , $\bar{V}_1, \dots, \bar{V}_c$ si l'on considère simultanément c catégories d'avions.

Pour un nombre suffisamment grand d'observations ($\Psi_{g_i}(t), V_{g_i}(t)$), notre problème d'extraction du vent se réduit donc à résoudre un système surdéterminé d'équations linéaires (5.4).

5.4.4 Résolution par une méthode de moindres carrés

Considérons un volume d'espace \mathcal{A} sur un intervalle de temps $[t_1, t_2]$, en supposant que le vent reste constant dans ce volume-4D. Supposons, en faisant ici abstraction des indices i des différents avions observés, qu'on collecte N observations $\{(\Psi_{g_j}, V_{g_j}) | j \in \{1, \dots, N\}\}$ des vitesses-sol V_g et des directions Ψ_g suivies par les avions traversant le volume \mathcal{A} pendant la période de temps considérée.

Supposons pour l'instant que tous les avions appartiennent à une même catégorie, de vitesse-air moyenne \bar{V} . Si la qualité et la quantité des données collectées est suffisante, c'est-à-dire si on a une dispersion correcte des valeurs de Ψ_g , les inconnues W , Ψ_w , et \bar{V} peuvent être calculées à partir des N instances de l'équation (5.4) correspondant aux N mesures radar des vitesses-sol et directions des avions.

En règle générale, N est bien supérieur au nombre d'inconnues, et il nous faut donc résoudre un système surdéterminé. Pour un tel système, il n'est généralement pas possible de trouver de solutions qui satisfassent exactement les N équations. On fait donc la distinction entre $\widehat{V}_{g_j} = \bar{V} + W_X \cos \Psi_{g_j}(t) + W_Y \sin \Psi_{g_j}(t)$, la vitesse-sol calculée par notre modèle linéaire (5.4), et V_{g_j} , la valeur observée de la vitesse-sol, et on recherche une solution qui minimise une fonction des erreurs $\epsilon_j = \widehat{V}_{g_j} - V_{g_j}$.

Dans [93], nous résolvons ce système linéaire par une méthode de moindres carrés ordinaire. On minimise donc la fonction de perte quadratique donnée par l'équation (5.5) :

$$E(W_X, W_Y, \bar{V}) = \sum_{j=1}^N \epsilon_j^2 = \sum_{j=1}^N (V_{g_j} - W_X \cos \Psi_{g_j} - W_Y \sin \Psi_{g_j} - \bar{V})^2 \quad (5.5)$$

Le minimum de cette erreur est obtenu en résolvant le système linéaire obtenu en annulant les dérivées partielles de $E(W_X, W_Y, \bar{V})$.

$$\frac{\partial E(W_X, W_Y, \bar{V})}{\partial W_X} = 0 \quad \frac{\partial E(W_X, W_Y, \bar{V})}{\partial W_Y} = 0 \quad \frac{\partial E(W_X, W_Y, \bar{V})}{\partial \bar{V}} = 0 \quad (5.6)$$

Quand la matrice associée à ce système est inversible, on obtient une solution $(\widehat{W}_X, \widehat{W}_Y, \widehat{\bar{V}})$ qui minimise l'erreur quadratique $E(W_X, W_Y, \bar{V})$. Cette solution est valable lorsque la matrice est bien conditionnée.

Au final, le vent est obtenu par les équations (5.7), en se rappelant que $W_X = W \cos \Psi_w$ et $W_Y = W \sin \Psi_w$.

$$\begin{aligned} W &= \sqrt{\widehat{W}_X^2 + \widehat{W}_Y^2} \\ \Psi_w &= \arctan_2\left(\frac{\widehat{W}_Y}{\widehat{W}_X}\right) \end{aligned} \quad (5.7)$$

5.4.5 Procédure de calcul, critères de qualité

La procédure d'extraction automatique du vent consiste à estimer celui-ci en un point $P(x, y, z)$ à un temps t , en appliquant la méthode des moindres carrés sur des données collectées dans un voisinage géographique et temporel du point-4D considéré. Selon les choix faits sur le filtrage des données, et en particulier sur la taille du voisinage du point P et la durée considérée, la qualité des données utilisées pour calculer le vent peut s'avérer insuffisante.

Nos expérimentations préliminaires ont fait ressortir un certain nombre de problèmes potentiels : nombre insuffisant d'observations, dispersion insuffisante des angles des directions

des routes, solutions multiples très différentes mais d'évaluation très proche, présence de points aberrants (ex. : nuages de points dûs à des avions en montée).

Chacun de ces problèmes est susceptible de fausser l'estimation, et nous avons donc proposé d'une part deux critères permettant d'évaluer la qualité de l'estimation, et d'autre part une procédure interactive permettant à l'utilisateur d'évaluer lui-même le résultat, et éventuellement de le corriger ou de l'invalider. En effet, le champ de vent peut avoir certaines régularités spatiales et temporelles que nous n'avons pas pris en compte dans notre modélisation, et qui peuvent être repérées par l'œil humain. Par exemple, si sur une grille régulière de points voisins le vent estimé est dans la direction du Nord-ouest, à l'exception d'un point où il pointe vers le Sud, on peut penser que le vent est mal estimé en ce point.

Les deux critères objectifs que nous avons proposés sont d'une part un seuil sur l'erreur quadratique moyenne calculée après régression, et d'autre part un critère d'entropie reflétant la dispersion des directions des vitesses. Le seuil sur l'erreur résiduelle permet de s'assurer que la dispersion des points autour de la courbe sinusoïdale obtenue après régression n'est pas trop importante. Une entropie suffisamment grande est quant à elle une condition suffisante pour que la matrice à inverser soit bien conditionnée.

L'entropie est calculée comme indiqué dans l'équation (5.8), en considérant la distribution des directions des vitesses, réparties dans n paquets partitionnant l'intervalle $[0^\circ, 360^\circ]$.

$$Ent = - \sum_i P_i \ln P_i \quad (\text{avec la convention } 0 \ln 0 = 0) \quad (5.8)$$

Dans cette équation, P_i est la probabilité empirique (en utilisant des histogrammes normalisés) que la direction de la vitesse-sol soit dans le $i^{\text{ième}}$ paquet.

En introduisant ces critères, la procédure automatique pour le calcul du vent est alors la suivante :

1. Filtrer les données, en espace et en temps, dans le voisinage de $P(x, y, z, t)$,
2. Évaluer la qualité des données filtrées (critère d'entropie),
3. Si le critère d'entropie est bon, appliquer la méthode des moindres carrés pour ajuster une courbe sinusoïdale et en déduire le vent,
4. Évaluer la qualité de la régression pour la courbe ajustée (critère d'erreur quadratique moyenne)
5. Si un des critères de qualité (entropie, ou erreur après régression) n'est pas satisfaisant, revenir à l'étape 1 et filtrer les données dans un voisinage plus large (en-deçà d'une taille maximum fixée) de $P(x, y, z, t)$, puis répéter la procédure.

Après plusieurs essais, les valeurs numériques des critères ont été empiriquement fixés à 1,7 pour l'entropie (avec 20 paquets), et à 0,35 pour le seuil sur l'erreur quadratique moyenne.

La procédure semi-automatique consiste simplement à autoriser une étape supplémentaire d'ajustement manuel de la courbe sinusoïdale par l'utilisateur, via une interface graphique (voir [93]). D'autres interfaces ont également été proposées pour visualiser la répartition spatiale des résultats et l'évolution dynamique du vent, afin de permettre à l'utilisateur d'évaluer la qualité du vent estimé, de le corriger, ou éventuellement de réitérer la procédure de calcul avec d'autres paramètres.

5.4.6 Résultats

Dans nos expérimentations préliminaires, nous avons utilisé le jeu de données radar Mode-C de la région parisienne (voir section 5.3.2), en comparant le vent calculé par notre méthode au vent fourni par Météo France. Le jeu de données Mode-S du radar expérimental de Toulouse, collecté plus tard, est enrichi des données recueillies à bord de l'avion et transmises au sol via la liaison de données Mode-S (voir section 5.3.1). Ces données Mode-S permettent de reconstruire le vent subi par l'avion, à partir de la vitesse-air mesurée à bord, et de la vitesse-sol. Nos résultats les plus complets portent donc sur ce deuxième jeu de données, où nous comparons les trois méthodes d'obtention du vent. Dans les résultats présentés ici, nous noterons **LS** notre méthode des moindres carrés, **MTO** la méthode de Météo France, et **Mode-S** la méthode de calcul à partir des données radar enrichies Mode-S.

La Figure 5.4 montre, sous forme de boîtes à moustaches¹, les résultats des comparaisons deux à deux des méthodes LS, MTO, et Mode-S. Les données sont sélectionnées entre les niveaux de vol *FL350* et *FL400*, sur un intervalle d'une heure.

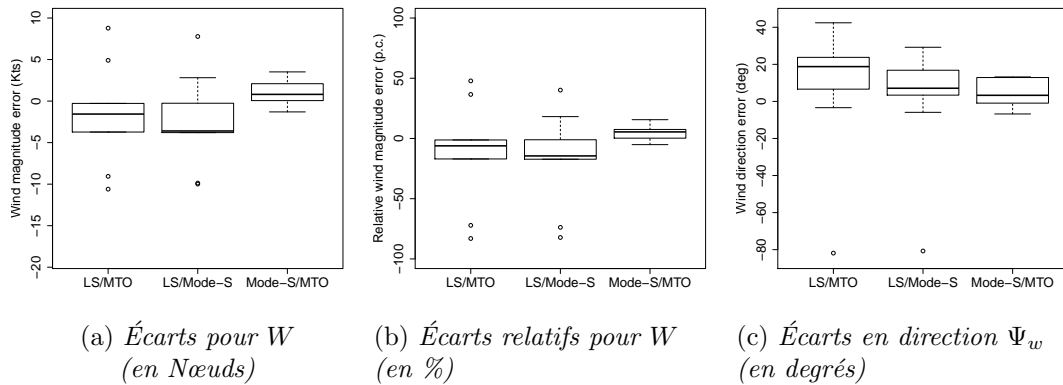


FIGURE 5.4: Résultats comparatifs des différentes méthodes d'obtention du vent (LS, MTO, et Mode-S) – avec les écarts absolus de l'intensité (en Nœuds) du vent W (à gauche), les écarts relatifs en % pour W (au milieu), et les écarts de direction, en degrés (à droite) – entre les niveaux *FL350* et *FL400*, pour une grille 4x4 et une fenêtre temporelle d'une heure (données Mode-S de la région de Toulouse).

Les résultats de la Figure 5.4 montrent que notre estimation du vent par une méthode de moindres carrés est proche des valeurs fournies par Météo France et du vent recalculé à partir des données enrichies Mode-S, du moins pour la tranche d'altitude considérée, et en collectant une heure d'enregistrements radar.

5.4.7 Conclusion sur l'extraction du vent

La conclusion sur notre méthode d'extraction du vent est mitigée. Notre approche, où les avions sont considérés comme des capteurs passifs, nous permet effectivement d'estimer le

1. Ces boîtes à moustaches sont obtenues par la fonction `boxplot` de l'environnement R pour le calcul statistique, avec son paramétrage par défaut. La boîte elle-même représente l'intervalle inter-quartile. La ligne à l'intérieur de la boîte donne la médiane, et les moustaches représentent soit la valeur extremum, soit au plus 1,5 fois la distance inter-quartile.

vent à partir de simples enregistrements radar. Nos expérimentations montrent cependant qu'il faut disposer d'un nombre suffisant de données, avec une certaine dispersion des directions des vols, pour pouvoir obtenir une estimation correcte.

Il y a donc un compromis à trouver dans le choix du volume d'espace et de l'intervalle de temps à considérer pour filtrer les données. En effet, on veut pouvoir obtenir suffisamment de données, avec une bonne dispersion, tout en gardant une estimation pertinente par rapport au point $P(x, y, z)$ et au temps t auquel on se place. Sélectionner un intervalle de temps et un volume d'espace trop grands ne nous donnerait qu'un vent moyen sur la zone et la période considérées, sans intérêt pour une utilisation réclamant une estimation relativement précise au point P . A l'inverse, un volume-4D trop petit peut ne contenir que trop peu de données, de faible qualité, qui ne nous permettent pas d'estimer le vent.

En pratique, le domaine d'application de notre méthode semble restreint à des zones où le trafic est relativement dense, avec des avions volant dans des directions diverses, et en espace aérien supérieur (plus spécifiquement entre les niveaux de vol $FL300$ et $FL400$). Dans ces tranches d'altitudes, la vitesse-air de croisière de jets commerciaux est relativement homogène, et les avions volent suffisamment vite pour que nos hypothèses simplificatrices soient valables.

Le meilleur compromis pour la taille de l'intervalle de temps semble être autour de 1 à 2 heures. Au vu des échelles géographiques et temporelles choisies, il y a donc fort à parier que notre méthode capture assez mal les changements météorologiques brutaux, comme le passage d'un front.

Avec une granularité géographique et temporelle assez grossière, notre méthode approxime cependant assez bien le vent dans les zones denses, et semble tout à fait comparable au vent fourni par Météo France. Rappelons que celui-ci est rafraîchi toutes les 3 à 6 heures, en opérationnel. Au vu des efforts fournis pour développer, alimenter, et exécuter les modèles météorologiques, la méthode que nous proposons est comparativement simple et peu coûteuse.

Toutefois, le déploiement généralisé des liaisons de données Mode-S permet d'ores-et-déjà d'utiliser les avions en vol comme des capteurs actifs, et non plus passifs comme dans notre approche. Les avions transmettent au sol des mesures de vent et de température qui peuvent être utilisées pour alimenter les modèles météorologiques (voir [41]), avec des gains significatifs dans l'estimation du vent. Dans ce contexte, la méthode que nous proposons ne peut être au mieux qu'une alternative à bas coût, à utiliser dans les zones géographiques où les modèles météorologiques sont peu précis et où les équipements permettant les liaisons de données sol-bord ne sont pas déployées.

Nous n'avons donc pas exploré plus avant cette voie de recherche, même si nous pensons que la méthode aurait pu être améliorée, notamment par l'utilisation de méthodes robustes.

5.5 Prévision de trajectoires d'avions en montée

5.5.1 Contexte et travaux antérieurs

La prévision des trajectoires d'avions a toujours été un élément-clé pour bon nombre d'applications, que ce soit à bord de l'avion ou dans les systèmes au sol. C'est encore plus vrai ces dernières années, avec l'introduction de concepts opérationnels basés sur les trajectoires-4D dans les programmes Européen SESAR [38] et américain NextGen [156].

On pourrait penser qu'il suffit de demander à l'avion de transmettre aux systèmes au sol sa trajectoire prévue, calculée à bord, pour obtenir la prévision la plus précise possible. Les liaisons de données sol-bord existantes ou en cours de déploiement pourraient être utilisées

à cet effet, moyennant quelques aménagements des formats de données. Bien que ce soit en partie vrai, il n'est cependant pas possible à ce jour de se passer complètement des systèmes de prédiction de trajectoires situés au sol. En effet, les algorithmes les plus récents proposés pour résoudre certains problèmes difficiles de gestion du trafic aérien nécessitent de calculer et de tester un grand nombre de trajectoires alternatives pour chaque avion. Dans le contexte actuel, il n'est pas envisageable de demander aux équipements de bord de l'avion de calculer et transmettre toutes ces trajectoires.

Pour illustration de tels algorithmes requérant des calculs fréquents et nombreux de trajectoires alternatives, citons [134] où une méthode itérative de Quasi-Newton est utilisée pour trouver des trajectoires minimisant les nuisances sonores au départ d'un aéroport. Un autre exemple est [32] où une méthode de Monte Carlo est utilisée pour estimer le risque de conflits entre trajectoires, dans un environnement stochastique. Quelques-uns des outils les plus récents développés pour la gestion et le contrôle du trafic aérien sont capables de détecter et résoudre les conflits aériens (voir [104] pour une revue de littérature). Pour trouver des solutions aux conflits de trajectoires, les méthodes les plus performantes s'appuient par exemple sur des méthodes *Mixed Integer Programming* ([130]), des algorithmes génétiques ([16, 48]), des algorithmes de colonies de fourmis ([52]), d'évolution différentielle ou d'essaims particuliers ([165]). Pour être efficaces, ces méthodes requièrent une prévision rapide et fiable des trajectoires futures, et une capacité à tester un grand nombre de trajectoires candidates. Ces exigences interdisent l'usage exclusif des prévisions faites à bord, bien que celles-ci soient certainement les plus précises que l'on puisse obtenir.

La plupart des prédicteurs au sol s'appuient sur un modèle physique de la dynamique de l'avion en vol. Dans la version simplifiée généralement utilisée, tel le modèle BADA d'Eurocontrol, l'avion est assimilé à son centre de masse, auquel s'applique la deuxième loi de Newton reliant l'accélération à la résultante des forces s'exerçant sur l'avion : poussée des moteurs, forces aérodynamiques (traînée, portance), et poids. Un tel modèle est formulé comme un ensemble d'équations différentielles et algébriques, que l'on doit intégrer sur un intervalle de temps pour calculer les positions successives de l'avion, connaissant son état initial (masse, position, vitesse, inclinaison, etc.), les conditions atmosphériques (vent, température), et les "intentions avion" (route suivie, loi de poussée des moteurs, vitesses-cibles).

Malheureusement, un grand nombre de ces données sont aujourd'hui inconnues des systèmes au sol, ou au minimum entachées d'incertitude. La masse de l'avion est considérée comme une donnée sensible par certaines compagnies aériennes, puisqu'elle reflète le taux de remplissage de leurs avions. La commande de poussée des moteurs, nominale, économique, ou autre, selon la position des manettes de gaz, n'est pas transmises au sol, ni le profil des vitesses-cibles envisagées pour le vol. Il y a par ailleurs des incertitudes sur les données météorologiques. Le problème des paramètres inconnus, tels que la masse, la loi de poussée, ou les vitesses-cibles pendant les phases évolutives, est particulièrement sensible pour la prévision des trajectoires en montée (voir illustration un peu plus loin, sur la Figure 5.5), où l'avion n'est pas stabilisé en vitesse, ni en altitude.

Certaines études ([1, 2, 39]) s'intéressent aux bénéfiques potentiels apportés par des données plus précises que les valeurs par défaut généralement utilisées. D'autres travaux proposent de formaliser les échanges sol-bord en un langage AIDL (*Aircraft Intent Description Language*) décrivant les intentions de pilotage de l'avion. Notre parti pris, dans les travaux présentés dans la suite de ce chapitre, a été de ne prendre en compte que les données aujourd'hui disponibles au sol, sans attendre la mise en place, dans un futur plus ou moins éloigné, d'un transfert des données indispensables à une prévision plus précise.

Nous nous intéressons ici à la prévision des trajectoires en montée, avec un horizon temporel de l'ordre de 10 minutes adapté aux applications de détection et de résolution des conflits aériens. L'idée est d'appliquer soit des méthodes d'estimation, soit des méthodes d'Apprentissage Artificiel, pour calculer des valeurs des paramètres inconnus qui permettent une prévision plus précise des altitudes futures de l'avion.

Quelques travaux antérieurs ([172, 171, 151, 112, 145]) se sont déjà intéressés à l'estimation des paramètres inconnus, en utilisant exclusivement les points passés de la trajectoire en cours. D'autres, plus rares ([56, 158, 61]), proposent une approche de type apprentissage supervisé où on apprend, à partir d'une base d'exemples de trajectoires enregistrées, un modèle statistique prévoyant directement les positions ou altitudes futures.

La démarche adoptée dans le cadre de la thèse de Richard Alligier [7] se situe entre ces deux approches. L'idée centrale consiste à estimer plus finement les paramètres inconnus du modèle physique de l'avion, soit directement à partir des points passés de la trajectoire, soit en apprenant à partir d'une base d'exemples des modèles permettant de prédire les valeurs de ces paramètres. Dans ce dernier cas, nous utiliserons donc des techniques d'apprentissage supervisé, non pas pour obtenir un modèle prévoyant directement les positions ou altitudes futures, mais pour obtenir plusieurs modèles prévoyant chacun un des paramètres manquants du modèle physique (masse, loi de poussée, vitesses-cibles).

5.5.2 Le modèle à énergie totale

Pour prévoir le comportement d'avions volant dans une masse d'air en mouvement autour du globe terrestre, il faut disposer d'un certain nombre de modèles mathématiques, physiques, ou calculatoires, et notamment d'un modèle avion (dynamique du vol, forces appliquées, consommation de carburant, enveloppe de vol), d'une modélisation des intentions de pilotage (réglage de la poussée des moteurs, répartition entre accélération et taux de montée), d'un modèle de l'atmosphère (équilibre fluide-statique, loi des gaz parfaits, atmosphère standard ISA, relations entre altitude géodésique et altitude-pressure géopotentielle), et d'un modèle du "globe" terrestre (ellipsoïde de révolution).

Nous n'allons pas décrire ici l'intégralité de ces modèles. Le lecteur pourra se reporter aux annexes C pour quelques éléments de géodésie, D pour la description des modèles atmosphériques, E pour une courte description des principes d'altimétrie en aviation, F pour la (ou les) vitesse(s) de l'avion, et surtout à l'annexe G pour une présentation complète des équations du mouvement de l'avion tenant compte de l'influence du gradient de vent.

Rappelons simplement ici l'équation des modèles dits "masse-énergie" ou "à énergie totale", qui va principalement nous intéresser dans la suite de ce chapitre. L'équation (5.9), extraite du système (G.74), exprime la relation entre l'accélération longitudinale $\dot{V}_a = \frac{dV_a}{dt}$, le taux de montée ou de descente $\dot{H}_p = \frac{dH_p}{dt}$, et les autres variables du modèle. C'est cette équation qui est utilisée dans l'immense majorité des prédicteurs-sol de trajectoires, généralement sous une forme encore plus simplifiée². Elle régit les aspects longitudinaux et verticaux de la dynamique de l'avion.

$$V_a \dot{V}_a + \left(\frac{T}{T_{ISA}} \right) g_0 \dot{H}_p = \left(\frac{Thr - D}{m} \right) V_a - \dot{\vec{W}} \cdot \vec{V}_a + g W_{vp} \quad (5.9)$$

2. Notamment, les effets du gradient de vent sont souvent négligés.

Dans cette équation, V_a est la vitesse de l'avion relativement à la masse d'air. Cette *vitesse-air* est souvent notée TAS (*true airspeed*, voir annexe F) dans la littérature. H_p est l'*altitude-pressure* de l'avion (voir section E.2 de l'annexe E pour les définitions d'altitudes). Les autres variables sont la masse m de l'avion, la poussée des moteurs Thr (*thrust*), la traînée aérodynamique D (*drag*) provoquée par le déplacement de l'avion dans l'air, et le vecteur vent \vec{W} , avec une composante vers le haut W_{Up} . T est la température extérieure, $T_{ISA} = T - \Delta T$ est la température qu'on obtiendrait à la même pression atmosphérique dans des conditions standard International Standard Atmosphere (ISA) (voir section D.2.5). Enfin g_0 est l'accélération due à la gravité terrestre au niveau moyen de la mer. Le lecteur pourra utilement se reporter à la première section de l'annexe G qui récapitule toutes les notations utilisées pour les équations du modèle avion.

Les principales hypothèses et approximations faites dans l'équation (5.9) sont les suivantes :

- toutes les forces s'appliquent au centre de masse,
- la poussée des moteurs s'exerce dans l'axe de la vitesse-air,
- la composante latérale de la force aérodynamique est négligeable,
- pour le calcul de l'accélération, le référentiel lié au plan horizontal local est considéré comme galiléen et les effets de la rotation de la terre sont négligés,
- le gradient de température est une fonction linéaire par morceaux de l'altitude-pressure, avec la même loi qu'en atmosphère standard ISA, mais avec des conditions initiales éventuellement différentes à l'isobare de référence (écart ΔT).

On observe que le terme $V_a \dot{V}_a$ est la dérivée de l'énergie cinétique spécifique (*i.e.* ramenée à l'unité de masse³) calculée avec une hypothèse de vent nul. Le terme $\frac{T}{T_{ISA}} g_0 \dot{H}_p$ est par ailleurs égal à $g \dot{h}$ quand on l'exprime en fonction de l'altitude géodésique h , et c'est donc la dérivée de l'énergie potentielle spécifique.

On peut donc reformuler l'équation (5.9) en identifiant d'un côté la puissance développée par les forces de poussée et de traînée, et de l'autre la variation d'énergie et l'effet du vent, pour obtenir l'équation (5.10).

$$\boxed{\underbrace{\frac{(Thr - D)V_a}{m}}_{\text{Puissance spécifique}} = \underbrace{V_a \frac{dV_a}{dt} + g_0 \left(\frac{T}{T - \Delta T} \right) \frac{dH_p}{dt}}_{\text{Variation d'énergie spécifique}} + \underbrace{\frac{d\vec{W}}{dt} \cdot \vec{V}_a - gW_{Up}}_{\text{Effet du vent}}} \quad (5.10)$$

En supposant la poussée Thr fixée, il faut rajouter à l'équation (5.10) une loi de commande qui fixe la répartition entre la puissance allouée à la montée et celle allouée à l'accélération longitudinale, si l'on veut effectivement calculer l'accélération \dot{V}_a et le taux de montée \dot{H}_p . Dans l'annexe G, c'est la pente-air γ_a de la trajectoire qui est choisie comme commande pour fixer cette répartition (voir section G.4.1).

La commande γ_a peut être remplacée par une autre, l'Energy Share Factor (ESF) qui est simplement définie, par l'équation (5.11), comme le ratio de la variation d'énergie potentielle

3. On fait l'hypothèse que l'éjection de masse \dot{m} est suffisamment lente pour ne pas influencer la dynamique de l'avion. On la considère soit comme négligeable dans la dérivée de l'énergie cinétique, soit comme faisant partie intégrante de la force de propulsion.

sur la variation d'énergie totale :

$$ESF = \frac{g_0 \left(\frac{T}{T-\Delta T} \right) \frac{dH_p}{dt}}{V_a \frac{dV_a}{dt} + g_0 \left(\frac{T}{T-\Delta T} \right) \frac{dH_p}{dt}} = \left[1 + \left(\frac{T - \Delta T}{T} \right) \left(\frac{V_a}{g_0} \right) \left(\frac{dV_a}{dH_p} \right) \right]^{-1} \quad (5.11)$$

En pratique, on aura différentes formes de cette expression de l'ESF et différentes lois de commande $ESF = ESF(M)$, où M est le nombre de Mach à l'instant t , et qui s'expriment différemment en fonction du mode opératoire de l'avion : montée à vitesse corrigée (CAS) constante, montée à nombre de Mach constant, à taux de montée fixé, etc. Ces lois sont décrites dans [129], pour le modèle BADA (*Base of Aircraft Data*) d'Eurocontrol.

Notons qu'il existe plusieurs mode de guidage dans le plan vertical : à vitesse CAS ou Mach constante, à taux de montée ou de descente (ROCD) constant, à pente constante, à ESF constant. La loi que suit l'*Energy Share Factor* est différente selon le mode choisi, et on peut, pour chaque mode, remplacer la commande ESF par une commande encore plus simple de type $CAS = \text{constante}$ par exemple (ou $ROCD = \text{constante}$, etc).

5.5.3 La problématique de la prédiction par les systèmes au sol

Connaissant l'état de l'avion à t_0 (position, vitesse, etc), on peut chercher à prévoir les positions futures de l'avion en s'appuyant sur un modèle physique de l'avion, tel que décrit dans la section précédente et dans l'annexe G .

Pour les évolutions verticales et l'accélération longitudinale, on s'intéressera essentiellement à l'équation (5.10), qu'on complétera par une loi de commande portant soit sur l'ESF (5.11), soit sur un profil de vitesses-cibles que doit suivre l'avion (ex. : segments à CAS ou Mach constants).

On voit immédiatement que l'utilisation de ces équations nécessite de connaître un certain nombre de paramètres, tels que la masse de l'avion m , la loi de commande de la poussée des moteurs Thr , et les intentions en vitesse (ou alternativement la loi de l'ESF). Ces paramètres sont connus du pilote, mais ne sont malheureusement pas disponibles pour les systèmes de contrôle au sol, à ce jour. Malheureusement, la masse et le profil de vitesse sont des données jugées sensibles par les compagnies aériennes car la masse donne une indication sur le taux de remplissage de l'avion, et les CAS/Mach cibles sur leur stratégie de gestion des coûts⁴. Certaines compagnies sont donc réticentes à transmettre ces données à des tiers, et les discussions sur le transfert par liaison de données de ces informations ne semblent pas en voie d'aboutir.

La Figure 5.5 illustre l'utilisation du modèle physique pour la prédiction d'une trajectoire, et les paramètres manquants.

Dans ce contexte, l'ensemble des prédicteurs-sol utilise aujourd'hui des valeurs par défaut pour la masse de l'avion, le profil de poussée, et les intentions en vitesse. Un des modèles les plus employés, le modèle BADA d'Eurocontrol, fournit ces valeurs par défaut en complément du modèle physique comprenant la modélisation des forces s'appliquant à l'avion. La Figure 5.6 illustre le processus de prévision de la trajectoire s'appuyant sur le modèle et les données fournies par BADA.

4. Le choix des valeurs de CAS et Mach cibles dépend du *cost index* de la compagnie, qui est un ratio entre coût du temps de transport et coût d'opération de l'avion (carburant, etc)

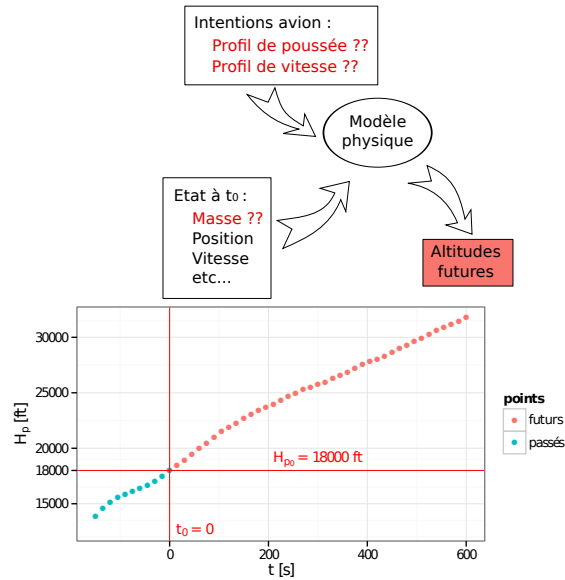


FIGURE 5.5: Problématique de la prévision de trajectoire (cas d'une montée). Les points bleus avant t_0 représentent la trajectoire passée, et les points rouge après t_0 la trajectoire future que l'on cherche à prévoir.

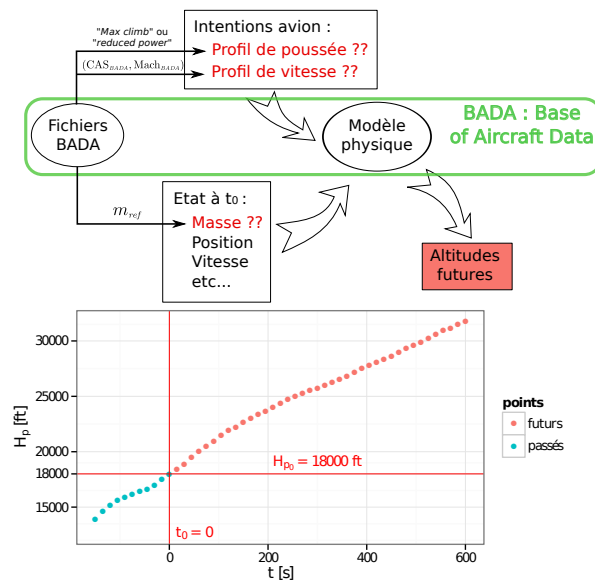


FIGURE 5.6: La prédiction avec le modèle BADA.

On conçoit aisément que cette prévision faite avec des valeurs par défaut, qui peuvent être très différentes des valeurs réelles, est assortie d'une très forte incertitude. C'est particulièrement le cas pour la montée de l'avion, où la masse et le profil de poussée ont une forte influence, et où les intentions en vitesse sont difficiles à déduire des observations, contrairement au vol stabilisé en croisière où la vitesse est sensiblement constante.

Jusqu'à récemment, cette imprécision n'était pas vraiment problématique puisque la prévision de trajectoire était essentiellement utilisée pour la desserte des "strips" (bande de papier portant les informations de vol) sur les postes de travail des contrôleurs. Le seul risque dans ce contexte était de prendre des marges d'incertitude trop grandes et de fournir des informations inutiles à quelques positions de contrôle.

Ce n'est plus le cas depuis quelques années, avec l'apparition d'algorithmes pour la détection et la résolution de conflits aériens pour lesquels la précision des trajectoires prédites est un élément essentiel pour la validité des solutions calculées, et donc pour la sécurité des vols et l'acceptabilité par les personnels opérationnels des nouveaux outils d'aide au contrôle.

Pour améliorer la qualité de la prévision, plusieurs approches sont envisageables. La première consiste à estimer les valeurs inconnues de certains paramètres du modèle physique à partir de l'observation des points passés de la trajectoire. Une deuxième approche consiste à les apprendre à partir d'une base d'exemples de trajectoires passées. Une troisième approche consiste, au vu du nombre de paramètres inconnus, à se passer totalement du modèle physique de l'avion et à apprendre directement la trajectoire en ajustant un modèle mathématique sur une base d'exemples.

5.5.4 Estimation de la masse

Le principe d'estimation de la masse consiste à ajuster la variable m de façon à minimiser l'erreur entre le terme de gauche de l'équation (5.10), issu d'un modèle dont m est une variable, et le terme de droite de la même équation qui peut être obtenu à partir des points passés de la trajectoire observée. La méthode d'estimation utilise donc le modèle physique et les points passés, comme illustré sur la Figure 5.7, pour calculer une masse qui est ensuite utilisée pour le calcul des points futurs de la trajectoire.

Pour simplifier nos équations dans la suite, notons Q le membre de droite de l'équation (5.10) :

$$Q = \underbrace{V_a \frac{dV_a}{dt} + g_0 \left(\frac{T}{T - \Delta T} \right) \frac{dH_p}{dt}}_{\text{Variation d'énergie spécifique}} + \underbrace{\frac{d\vec{W}}{dt} \cdot \vec{V}_a - gW_{V_p}}_{\text{Effet du vent}} \quad (5.12)$$

Connaissant le vecteur d'état de l'avion pour la trajectoire passée, et disposant des informations météorologiques de vent et de température, nous pouvons calculer cette quantité Q en chaque point passé de la trajectoire.

Le terme de gauche de l'équation (5.10) est quant à lui simplement l'excès de puissance des moteurs disponible pour la montée ou l'accélération, divisée par la masse. On parlera dans la suite d'excès de puissance spécifique, ou plus simplement de puissance spécifique :

$$P_{es} = \frac{Power}{m} = \frac{(Thr - D)V_a}{m} \quad (5.13)$$

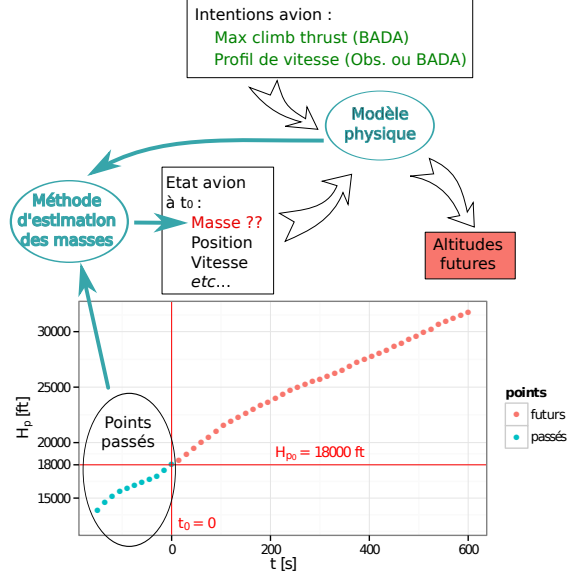


FIGURE 5.7: Estimation de la masse.

Cette puissance modélisée dépend de la masse m , qui apparaît au dénominateur et également dans la modélisation de la traînée D . Elle dépend aussi du choix fait pour la commande c de poussée $Thr = Thr(c, V_a, H_p, T)$ (G.58). En considérant par ailleurs l'expression des forces aérodynamiques (G.50), (G.53), et (G.51), on constate qu'on peut exprimer les forces de poussée et de traînée sous la forme suivante, en fonction des variables d'état V_a , H_p , T^5 , Φ , de la commande de poussée c , et de la masse m :

$$Thr = f_1(c, H_p, V_a, T) \quad (5.14a)$$

$$D = f_2(H_p, V_a, T) + m^2 f_3(H_p, V_a, T, \Phi) \quad (5.14b)$$

Pour une commande de poussée c fixée, et avec ces expressions (5.14) pour les forces, la puissance $Power(m) = (Thr - D)V_a$ s'exprime comme un polynôme de degré 2 en fonction de la masse m . L'écart entre la puissance spécifique modélisée P_{es} et la quantité observée Q représentant la dérivée de l'énergie totale peut être exprimée comme une fonction rationnelle (un ratio de deux polynômes) de m , en tout point de la trajectoire passée.

$$P_{es} - Q = \frac{-m^2 V_a f_3(H_p, V_a, T, \Phi) - mQ + V_a [f_1(c, H_p, V_a, T) - f_2(H_p, V_a, T)]}{m} \quad (5.15)$$

Dans la suite, on notera $P(m)$ le polynôme qui apparaît au numérateur de (5.15) :

$$\begin{aligned} P(m) &= Power(m) - mQ \\ &= -m^2 V_a f_3(H_p, V_a, T, \Phi) - mQ + V_a [f_1(c, H_p, V_a, T) - f_2(H_p, V_a, T)] \end{aligned} \quad (5.16)$$

5. On choisit les variables H_p et T sachant que la densité de l'air ρ qui apparaît dans l'expression des forces aérodynamiques dépend directement de la température et de la pression (et donc de l'altitude-pression).

L'écart entre la puissance modélisée et l'observation s'écrit alors simplement :

$$P_{es} - Q = \frac{P(m)}{m} \quad (5.17)$$

Une méthode de type “moindres carrés”

Considérant n points de la trajectoire passée, on peut chercher à estimer la masse en minimisant l'erreur quadratique suivante, où m_i est la masse au point i de la trajectoire :

$$\begin{aligned} \mathcal{E}(m_1, \dots, m_n) &= \sum_{i=1}^n \left(\frac{Power_i(m_i)}{m_i} - Q_i \right)^2 \\ &= \sum_{i=1}^n \left(\frac{P_i(m_i)}{m_i} \right)^2 \end{aligned} \quad (5.18)$$

La relation entre les valeurs successives de la masse aux différents points de la trajectoire est contrainte par la consommation de carburant, dont l'équation peut s'exprimer comme suit :

$$\dot{m} = \frac{dm}{dt} = -f_{nom} = -\eta Thr \quad (5.19)$$

La consommation spécifique η est donnée dans le modèle BADA comme une fonction de la vitesse-air V_a (voir équations 3.9.1 et 3.9.2 dans [129]). La consommation nominale $f_{nom} = \eta Thr$ varie au cours du temps et peut s'écrire comme une fonction de la commande de poussée c et des valeurs prises par les variables d'état au temps t :

$$f_{nom}(t) = f_{nom}(c(t), V_a(t), H_p(t), T(t)) \quad (5.20)$$

On peut donc exprimer la masse au point i en fonction de la masse m_n au dernier point du segment de trajectoire observé, en intégrant la consommation de carburant, comme exprimé par l'équation (5.21).

$$m_i = m_n + \int_{t_i}^{t_n} f_{nom}(c(t), V_a(t), H_p(t), T(t)) dt \quad (5.21a)$$

$$\simeq m_n + \sum_{k=i}^{n-1} \frac{f_{nom}(t_{k+1}) + f_{nom}(t_k)}{2} (t_{k+1} - t_k) \quad (5.21b)$$

$$= m_n + \delta_i \quad (5.21c)$$

Au final, en prenant en compte la consommation, on obtient une expression de l'erreur \mathcal{E} qui dépend uniquement de la masse m_n au dernier point observé et des quantités $\delta_i =$

$\sum_{k=i}^{n-1} \frac{f_{nom}(t_{k+1})+f_{nom}(t_k)}{2} (t_{k+1}-t_k)$ qu'on peut calculer à partir des données disponibles en chaque point i précédent.

$$\mathcal{E}(m_n) = \sum_{i=1}^n \left(\frac{P_i(m_n + \delta_i)}{(m_n + \delta_i)} \right)^2 \quad (5.22)$$

La masse est estimée en minimisant l'erreur $\mathcal{E}(m_n)$. Le minimum de cette fonction sur $]0; +\infty[$ satisfait l'équation qui annule la dérivée : $\mathcal{E}'(m) = 0$. Quand on réduit au même dénominateur cette équation, résoudre $\mathcal{E}'(m) = 0$ se ramène à trouver les racines d'un polynôme de degré $3(n-1) + 4$. Pour simplifier le calcul, on approxime la fonction d'erreur $\mathcal{E}(m_n)$ en remplaçant les quantités δ_i par leur moyenne sur l'ensemble des points i . On définit l'erreur \mathcal{E}_{approx} par les équations ci-dessous :

$$\delta_{avg} = \frac{1}{n} \sum_{i=1}^n \delta_i \quad (5.23a)$$

$$\mathcal{E}_{approx}(m_n) = \sum_{i=1}^n \left(\frac{P_i(m_n + \delta_i)}{m_n + \delta_{avg}} \right)^2 \quad (5.23b)$$

Comme la variation relative de la masse est faible dans la fenêtre temporelle considérée (quelques minutes de vol), l'erreur relative due à l'approximation est également faible. Dans [9], on montre que l'utilisation de \mathcal{E}_{approx} sur une trajectoire simulée permet de retrouver à la centaine de grammes près la masse avec laquelle la trajectoire a été générée.

La dérivée de l'erreur $\mathcal{E}_{approx}(m_n)$ par rapport à la masse m_n s'exprime de la façon suivante :

$$\mathcal{E}'_{approx}(m_n) = \frac{2}{(m_n + \delta_{avg})^3} \sum_{i=1}^n P_i(m_n + \delta_i) [P'_i(m_n + \delta_i)(m_n + \delta_{avg}) - P_i(m_n + \delta_i)] \quad (5.24)$$

Le minimum de la fonction d'erreur vérifie donc l'équation polynomiale suivante, qui n'est plus que de degré 4 (nous avons vu précédemment que les P_i sont de degré 2) :

$$\sum_{i=1}^n P_i(m_n + \delta_i) [P'_i(m_n + \delta_i)(m_n + \delta_{avg}) - P_i(m_n + \delta_i)] = 0 \quad (5.25)$$

La masse estimée \hat{m}_n s'obtient en résolvant l'équation (5.25). En pratique, plutôt qu'une méthode analytique potentiellement sujette à des instabilités numériques, nous utilisons la méthode numérique implantée dans la *GNU Scientific Library* qui effectue une décomposition QR de la matrice de Frobenius associée au polynôme pour trouver ses racines [54]. Parmi les quatre solutions réelles potentiellement isolées, on sélectionne celle minimisant \mathcal{E}_{approx} .

Une méthode robuste

Afin de rendre l'estimation de la masse plus robuste aux valeurs aberrantes, on utilise une fonction d'erreur qui pénalise les grands écarts entre le modèle et l'observation.

$$\mathcal{E}_{robuste}(m_n) = \sum_{i=1}^n L_r \left(\frac{P_i(m_n + \delta_i)}{m_{avg}(m_n)} \right) \quad (5.26a)$$

$$\text{avec } L_r(\varepsilon) = \left(\sqrt{1 + \frac{\varepsilon^2}{r}} - 1 \right) r \quad (5.26b)$$

La fonction de perte de pseudo-Huber L_r est proche de la fonction valeur absolue (à un facteur multiplicatif près) pour de grandes valeurs de ε , et se comporte comme une fonction quadratique pour des valeurs plus petites. Elle a par ailleurs l'avantage, par rapport à la fonction de Huber originale, d'être C^∞ .

L'erreur $\mathcal{E}_{robuste}$ est ici minimisée en utilisant un algorithme de Newton par intervalles ([83, 163]), qui s'appuie sur l'arithmétique des intervalles ([123]). Cette méthode permet de borner rigoureusement tous les zéros de la dérivée $\mathcal{E}'_{robuste}$.

Résultats sur l'estimation de la masse

La méthode de moindres carrés est comparée dans [9] à la méthode adaptative de Schultz *et al.* ([145]), sur des données simulées avec un bruit aléatoire sur les variables d'état de l'avion. Cette comparaison est étendue dans [13] à des données réelles. Dans [7], les trois méthodes (moindres carrés, robuste, et adaptative) sont évaluées et comparées sur des données simulées et sur deux jeux de données réelles issus d'enregistrements radars Mode-C de la région parisienne, et du radar expérimental Mode-S de Toulouse.

Notons que l'évaluation des résultats sur données réelles pose problème, puisqu'on ne dispose pas de la vraie masse de l'avion. Les performances des différentes méthodes ne peuvent donc être évaluées qu'au regard de la qualité de la prédiction de trajectoire obtenue avec la masse estimée.

Pour résumer, les résultats sur données simulées donnent l'avantage à la méthode robuste, suivie de la méthode de moindres carrés, la méthode adaptative s'avérant la plus sensible aux bruitages des variables. Notons cependant que l'ensemble des trois méthodes donne de très bons résultats et permet de retrouver la masse initiale utilisée pour produire les données artificielles, avec une RMSE sur l'erreur relative de l'ordre de quelques pour-cents (voir [9, 7]).

L'étude sur données simulées a par ailleurs mis en évidence que la précision de l'estimation est variable, avec un écart-type plus grand lorsque l'atmosphère est suffisamment chaude (voir Figure 5.8). En effet, on observe un seuil sur l'écart ΔT aux conditions ISA, au-delà duquel la poussée modélisée par BADA décroît. Comme la masse estimée dépend d'une hypothèse sur la poussée, on conçoit que si la poussée modélisée est entachée d'erreur, alors la masse estimée l'est également.

Sur les données réelles, les trois méthodes sont aussi comparées à une méthode naïve par dichotomie et avec la prédiction obtenue avec la masse de référence (BADA) de l'avion. Les tableaux 5.1 et 5.2, issus de [7], donnent les résultats des différentes méthodes sur les données Mode-C et Mode-S respectivement.

Dans l'ensemble de ces résultats, on considère que le profil des vitesses-cibles est parfaitement connu. La "prévision" de trajectoire est donc faite en suivant le profil de vitesse-air

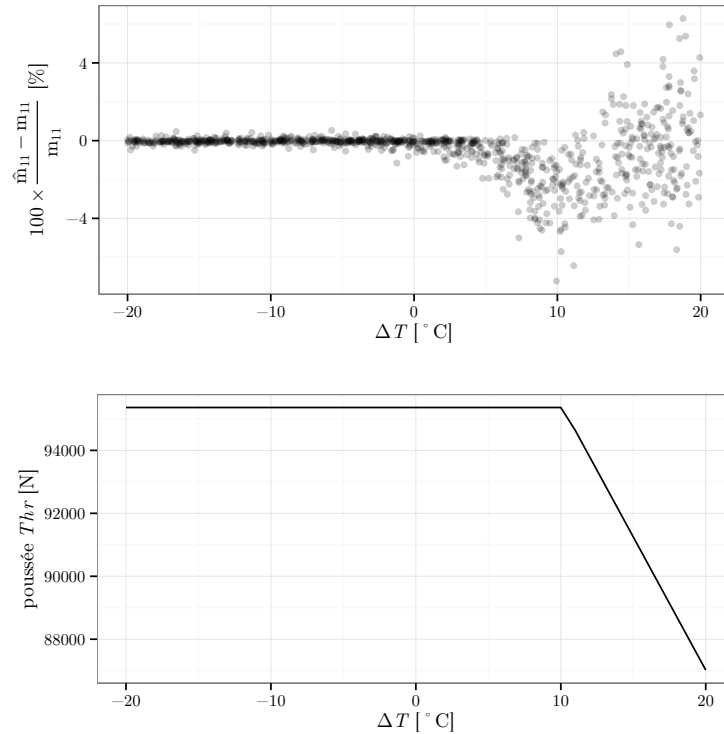


FIGURE 5.8: Erreur relative sur la masse estimée (en haut) et modélisation de la poussée des moteurs (en bas) en fonction des conditions de température. Données simulées pour un Airbus A320, avec un bruit gaussien d'écart-type $\sigma_T = 5$ K sur la température. \hat{m}_{11} est la masse estimée au onzième point en appliquant la méthode des moindres carrés aux onze premiers points.

TABLE 5.1: Erreur sur l'altitude prévue à $t_0 + 10$ minutes avec la masse estimée en supposant le profil de vitesse connu $V_a(t) = V_a^{(obs)}(t)$ et une poussée nominale *max climb* des moteurs, pour le jeu de données de trajectoires Mode-C A320.

méthode	moy [ft]	écart-type [ft]	moy abs [ft]	rmse [ft]	max abs [ft]
<i>mass_{ref}</i>	-82.4	1479	1168	1481	5493
naïve	-524	651	628	836	6874
adaptative	-303	685	582	749	5535
moindres carrés	-532	653	631	843	6033
robuste (pseudo-Huber)	-503	595	603	779	6843

TABLE 5.2: Erreur sur l'altitude prévue à $t_0 + 10$ minutes avec la masse estimée en supposant le profil de vitesse connu $V_a(t) = V_a^{(obs)}(t)$ et une poussée nominale *max climb* des moteurs, pour le jeu de données de trajectoires Mode-S A320.

méthode	moy [ft]	écart-type [ft]	moy abs [ft]	rmse [ft]	max abs [ft]
<i>mass_{ref}</i>	-810	1133	1079	1392	5531
naïve	-598	602	673	849	6574
adaptative	-551	554	651	781	4171
moindres carrés	-584	606	663	842	7150
robuste (pseudo-Huber)	-571	550	645	792	6371

observé pour les points futurs $V_a(t) = V_a^{(obs)}(t)$, en utilisant la masse \hat{m}_{11} estimée sur les 11 points passés ou présent, et en supposant la poussée des moteurs fixée à la valeur nominale en montée $Thr_{\max \text{ climb}}$ donnée par le modèle BADA. La prévision réalisée n'est donc pas réaliste, car dans le contexte réel le profil de vitesse ne serait pas connu, mais elle nous permet tout de même d'évaluer comparativement les différentes estimations de la masse.

Les tableaux 5.1 et 5.2 montrent d'une part tout l'intérêt d'estimer la masse de l'avion en utilisant les points passés de la trajectoire. On constate une amélioration de l'ordre de 40 % à 50 % de la RMSE sur l'erreur d'altitude au bout de 10 minutes de montée, par rapport à la prévision faites avec la masse standard de référence BADA.

Parmi les méthodes testées, les méthodes robuste et adaptative donnent les meilleurs résultats, avec peut-être un léger avantage pour cette dernière. La méthode adaptative privilégiant les derniers points observés, on peut penser qu'une simple pondération des observations en fonction de leur ancienneté, dans la méthode robuste, donnerait des résultats au moins aussi bons. Ceci reste encore à étudier.

Les limites de l'estimation de la masse

Malgré l'intérêt évident d'estimer la masse à partir des points passés, on constate sur les tableaux 5.1 et 5.2 que les erreurs maximum en valeur absolue (dernière colonne) restent grandes. A titre d'illustration, les Figures 5.9 et 5.10 tracent l'évolution des erreurs de prévision de la puissance spécifique pour deux trajectoires particulières, l'une du jeu de données Mode-C et l'autre du jeu de données Mode-S, pour lesquelles l'erreur est maximale.

La Figure 5.9 montre un cas où la puissance semble mal modélisée pour les points passés ($t \leq 0$) de la trajectoire, avec pour résultat un mauvais ajustement de la masse. A l'inverse, sur la Figure 5.10, la puissance modélisée et l'observation Q sont très proches pour les points passés, mais on a un décrochage pour les points futurs, avec un écart important entre $t = 0$ et $t = 400$ secondes.

Cette mauvaise modélisation de la puissance est très vraisemblablement due, pour des erreurs aussi grandes, à des variations de la commande de poussée qui peuvent intervenir avant ou après $t = 0$. On voit donc qu'il ne suffit pas de disposer d'une méthode d'estimation de la seule masse m . Il faut s'intéresser simultanément à la masse et à la loi de poussée. Dans la suite, on tentera d'améliorer nos hypothèses sur la loi de poussée en cherchant à extraire certaines régularités d'une base d'exemples de trajectoires.

Par ailleurs, plutôt que d'estimer simplement la variable masse sur les points passés, on peut aussi chercher à construire un modèle de prédiction d'une variable masse m_{futur} qui donnerait le meilleur ajustement de la puissance modélisée à la quantité observée Q , sur

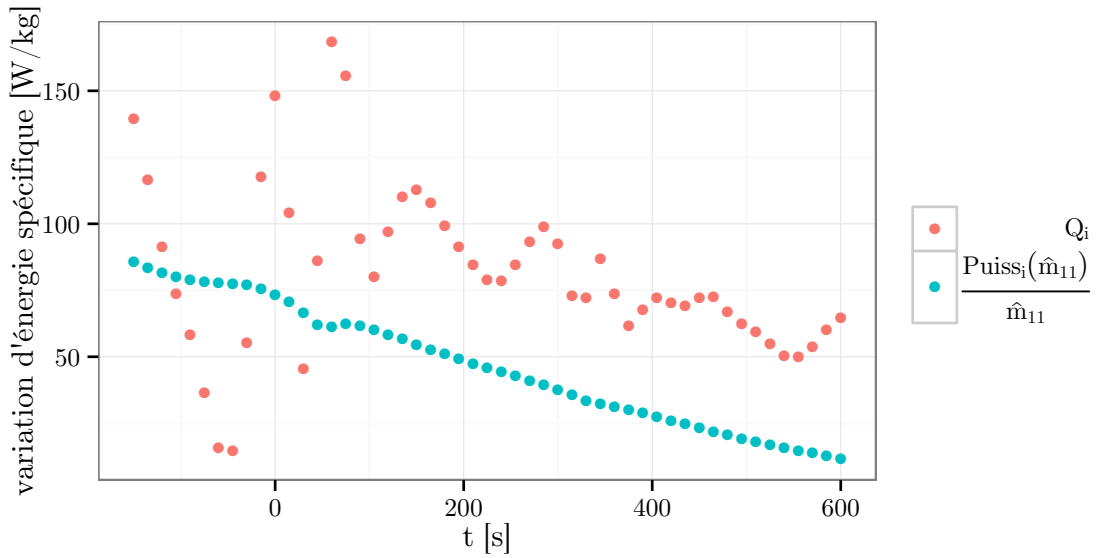


FIGURE 5.9: Puissance spécifique $\frac{Power_i(\hat{m}_{11})}{\hat{m}_{11}}$ calculée et quantité observée Q_i pour la dérivée de l'énergie spécifique avec l'effet du vent, pour la trajectoire Mode-C avec laquelle on constate le plus d'erreur avec la méthode des moindres carrés. \hat{m}_{11} est la masse est estimée à $t = 0$ sur les onze premiers points de la trajectoire ($t \leq 0$).

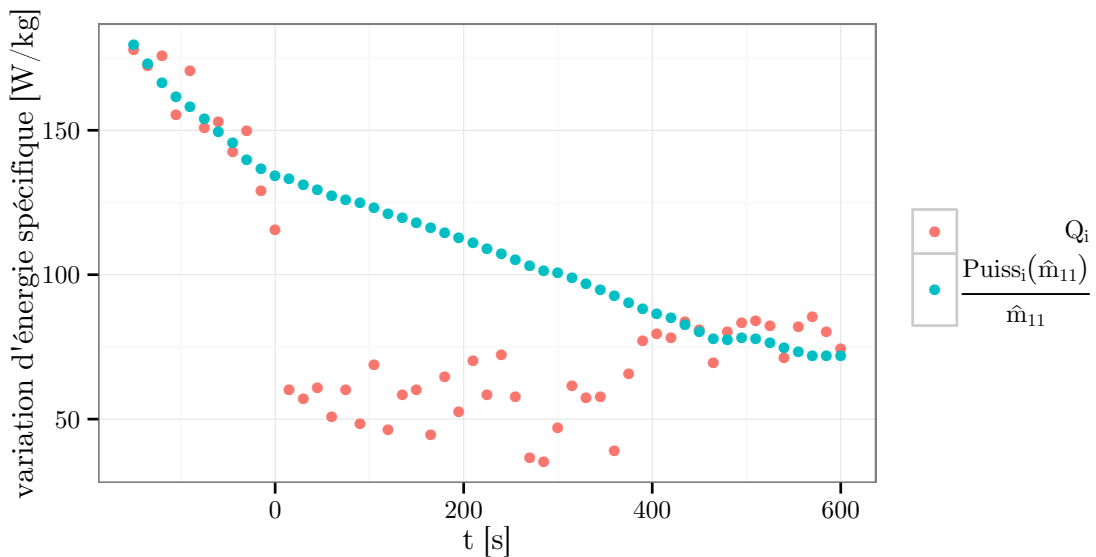


FIGURE 5.10: Puissance spécifique $\frac{Power_i(\hat{m}_{11})}{\hat{m}_{11}}$ calculée et quantité observée Q_i pour la dérivée de l'énergie spécifique avec l'effet du vent, pour la trajectoire Mode-S qui surestime le plus l'altitude à $t = 600$ s en estimant la masse avec la méthode des moindres carrés. \hat{m}_{11} est la masse estimée à $t = 0$ sur les onze premiers points de la trajectoire ($t \leq 0$).

les points futurs. On peut alors espérer que l'utilisation d'une base d'exemples permette de mieux prendre en compte certains paramètres dont l'influence n'apparaît pas immédiatement quand on ne considère que les points passés d'une unique trajectoire. Par exemple, on sait que la quantité embarquée de carburant, et donc la masse de l'avion, croît avec la distance à parcourir. C'est typiquement le genre de relation qu'on peut chercher à extraire d'exemples passés, en introduisant par exemple la distance à parcourir parmi les variables explicatives de notre modèle de prévision de la masse.

Enfin, nous ne nous sommes intéressés jusqu'à présent qu'à l'estimation de la masse, en ajustant la puissance modélisée à l'observation. Ceci permet d'obtenir un modèle de prévision de la puissance excédentaire, modèle que l'on peut appliquer aux points futurs. Cependant, prévoir l'altitude future de l'avion, plutôt que simplement sa puissance, nécessite de connaître aussi la loi de répartition entre taux de montée et accélération. Cette répartition fixe le profil de vitesse prévu pour la montée. On peut aussi, comme nous allons le voir dans la suite, chercher à extraire cette loi des vitesses-cibles de notre base d'exemples en utilisant des méthodes classiques d'apprentissage artificiel.

5.5.5 L'apprentissage appliqué à la prévision de trajectoires d'avions

Par opposition à la simple estimation de paramètres à partir des seuls points passés de la trajectoire en cours, les méthodes d'Apprentissage Artificiel cherchent à extraire un modèle de prévision à partir d'une base d'exemples comprenant un grand nombre de trajectoires diverses dont on pense qu'elles possèdent *a priori* des caractéristiques communes. Dans notre cas, on considérera tous les avions de même type décollant d'une TMA donnée (Paris, ou Toulouse). Une fois appris sur les exemples, le modèle est ensuite utilisé pour fournir la valeur de paramètres permettant de prédire une nouvelle trajectoire.

Plus formellement, le principe de l'apprentissage supervisé (voir chap. 3) consiste donc à ajuster un modèle $y = h(x)$ sur un jeu de données $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$ où les y_n est des exemples de réponse attendue aux entrées x_n . L'ajustement du modèle se fait en minimisant l'erreur entre les sorties calculées \hat{y}_n et les réponses attendues y_n , selon une fonction de perte à choisir.

Dans notre cas, le vecteur de variables explicatives x en entrée du modèle regroupe les informations disponibles à l'instant t_0 de la prévision, comme par exemple les altitudes et vitesses sur les points passés de la trajectoire ($t \leq t_0$). La variable réponse y pourra être un des paramètres manquants du modèle physique de l'avion. On peut également être tenté de faire totalement abstraction de ce modèle, pour prévoir directement l'altitude future avec un modèle statistique, auquel cas la réponse y est simplement cette altitude future.

Apprentissage des paramètres du modèle physique

La Figure 5.11 illustre les différentes lois et paramètres que l'on peut chercher à apprendre à partir d'une base d'exemples de trajectoires : la loi de commande de la poussée des moteurs, le profil de vitesses-cibles prévu pour la montée, et une masse dont on espère qu'elle donne la meilleure prédiction possible de la puissance spécifique sur les points futurs.

La thèse de Richard Alligier a exploré ces trois aspects, soit en appliquant différentes méthodes classiques d'apprentissage artificiel pour construire des modèles prédisant la masse ou le profil de vitesse, soit en développant une méthode *ad-hoc* combinant un algorithme BFGS et une méthode d'estimation de masse (voir section précédente) pour trouver une loi

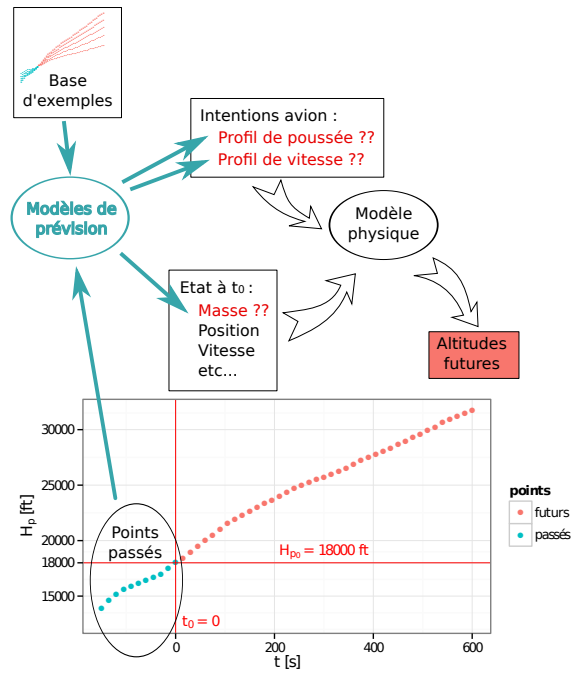


FIGURE 5.11: Apprentissage des paramètres du modèle physique.

de poussée minimisant un critère d'erreur sur la base d'exemples.

Apprentissage direct de l'altitude future

Dans le cas de l'apprentissage direct de l'altitude future, on n'utilise plus du tout le modèle physique de l'avion mais un modèle statistique dont la sortie est l'altitude future, comme illustré sur la Figure 5.12.

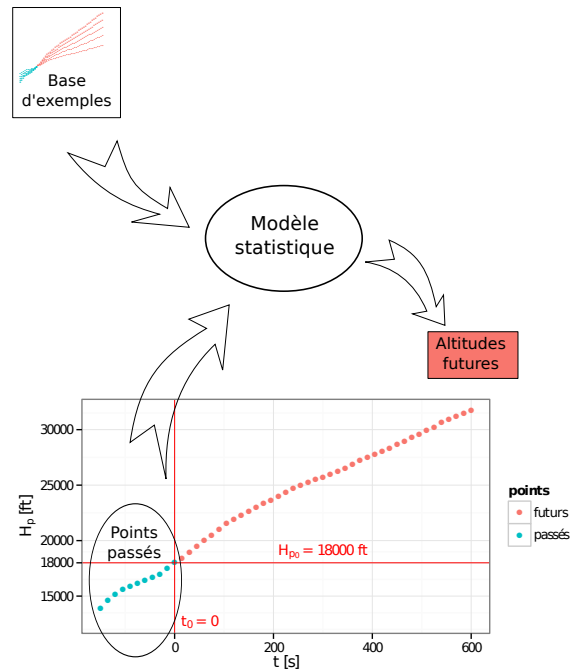


FIGURE 5.12: Apprentissage direct des trajectoires.

5.5.6 Cadre expérimental pour les méthodes d'apprentissage

Les méthodes utilisées et leurs paramètres

L'éventail des méthodes d'apprentissage est extrêmement vaste. En première approche dans le travail de thèse de Richard Alligier, cinq méthodes ont été testées : trois méthodes linéaires (MLR-FS, Ridge, PCR), une approche non linéaire par réseau de neurones (NNet), et une méthode de boost d'arbres de régression (GBM : *Gradient Boosting Machine*). MLR-FS est une régression linéaire multiple avec une sélection des variables explicatives de type *forward selection* (voir [86], section 3.3.2). PCR est une méthode de régression sur composantes principales. Les modèles linéaires, les réseaux de neurones, et les méthodes de *gradient boosting* sont présentées plus en détail dans les sections 3.7, 3.8, et 3.9 du chapitre 3. Les logiciels utilisés sont des packages de l'environnement R pour le calcul statistique : `caret`, `nnet`, et `gbm`.

Le tableau 5.3 récapitule les différentes méthodes d'apprentissage utilisées, et les grilles d'hyper-paramètres associés.

La sélection d'hyper-paramètres se fait soit par validation simple (*holdout validation*), soit par validation croisée à 10 plis (voir section 3.5.1 du chapitre 3), comme décrit par les fonctions `TUNEGRID` et `TUNEGRIDCV` dans l'algorithme 16. Supposons qu'on ait un algorithme d'apprentissage \mathcal{A}_λ dont les hyper-paramètres sont regroupés dans un vecteur λ . Soit par ailleurs T l'ensemble de données d'apprentissage à partir desquelles \mathcal{A}_λ cherche à apprendre un modèle. La sélection d'hyper-paramètres consiste à chercher le vecteur λ^* tel que le modèle produit par \mathcal{A}_{λ^*} donne les meilleurs résultats possibles lorsqu'il généralise sur de nouvelles entrées. La fonction `TUNEGRID` réserve 20 % des données de l'ensemble T initial pour estimer cette erreur en généralisation. Les 80 % restants sont utilisés pour ajuster le

TABLE 5.3: Algorithmes d'apprentissage utilisés et grilles d'hyper-paramètres associés.

méthode	grille d'hyper-paramètres
MLR-FS _k	$k = \llbracket 2; \min(120, n_{var}) \rrbracket$
Ridge _λ	$\lambda = 10^{\llbracket -5; 1 \rrbracket} \cup 0.5 \times 10^{\llbracket -5; 0 \rrbracket}$
PCR _k	$k = \llbracket 2; \min(120, n_{var}) \rrbracket$
NNet _(n,λ)	$n = \{2, 3, 4, 5, 6\}$
	$\lambda = \{0.1, 0.2, 0.3, 0.4, 0.5\}$
GBM _(m,J,ν)	$m = \{1000, 1500, 2000\}$
	$J = \{3, 5, 10, 15\}$
	$\nu = \{0.001, 0.0025, 0.005, 0.01, 0.025, 0.05\}$

Algorithm 16 Algorithmes de sélection d'hyper-paramètres pour un algorithme \mathcal{A}_λ d'hyper-paramètre λ opérant sur un ensemble d'apprentissage T . Pour *TuneGrid*, le critère pour sélectionner le meilleur hyper-paramètre est l'erreur en validation Err_{val} sur 20 % des exemples que l'on a mis de côté. Pour *TuneGridCV*, le critère est l'erreur en validation obtenue par validation croisée sur 10 plis.

```

1: function TUNEGRID( $\mathcal{A}_\lambda, grid$ )[ $T$ ]
2:    $(T_T, T_V) \leftarrow partition_{(80\%, 20\%)}(T)$ 
3:    $\lambda^* \leftarrow \underset{\lambda \in grid}{argmin} Err_{val}(\mathcal{A}_\lambda, T_T, T_V)$ 
4:   return  $\mathcal{A}_{\lambda^*}[T]$ 
5: end function

1: function TUNEGRIDCV( $\mathcal{A}_\lambda, grid$ )[ $T$ ]
2:    $\lambda^* \leftarrow \underset{\lambda \in grid}{argmin} CV_{10}(\mathcal{A}_\lambda, T)$ 
3:   return  $\mathcal{A}_{\lambda^*}[T]$ 
4: end function

```

modèle pour un jeu donné d'hyper-paramètres λ . On teste une grille de valeurs pour λ , et on retient la valeur λ^* ayant la meilleure erreur sur les 20 % mis de côté.

La validation croisée à 10 plis de TUNEGRIDCV procède de la même façon, à ceci près qu'on divise T en dix sous-ensembles, chacun servant à son tour à évaluer l'erreur de généralisation, le reste des données servant à apprendre un modèle avec A_λ . On choisit la valeur λ^* pour laquelle l'erreur de généralisation évaluée sur les 10 ensembles de validation est la meilleure. Cette procédure est donc plus coûteuse en temps de calcul, mais elle est plus fiable et utilise plus judicieusement et de façon plus économe le jeu de données T . Notons que cet ensemble T peut lui-même être issu d'une validation croisée de niveau supérieur. On parle alors de double validation croisée imbriquée.

Dans notre cas, les résultats présentés dans les sections qui suivent sont obtenus par une validation simple imbriquée dans une validation croisée à 10 plis CV10(TUNEGRID) pour le jeu de données Mode-C, et par une double validation croisée imbriquée CV10(TUNEGRIDCV) pour les données Mode-S, pour lesquelles nous avons moins d'échantillons.

Variables explicatives

Les variables explicatives en entrée des modèles de prévision sont récapitulées dans le tableau 5.4. Elles sont regroupées selon leur provenance et leur nature : le groupe r regroupe des variables réelles extraites des enregistrements radar des trajectoires, le groupe w contient les informations météorologiques fournies par Météo France, le groupe p les variables réelles tirées des plans de vol, le groupe c les variables catégorielles provenant de ces mêmes plans de vol, le groupe \hat{m} les masses estimées sur les points passés en utilisant les méthodes présentées en 5.5.4, et enfin le groupe t ne contient qu'une variable, la variante de l'appareil, disponible uniquement dans les adresses Mode-S pour un seul de nos deux jeux de données. Différentes combinaisons de ces variables sont testées en entrée des méthodes d'apprentissage, dont certaines intègrent par ailleurs une méthode d'extraction ou de sélection des variables d'entrée.

Variables réponses et construction des exemples

Rappelons qu'on souhaite apprendre à partir des données plusieurs modèles qui fournissent en sortie soit la masse de l'avion, soit son profil de poussée, ou bien les vitesses-cibles (CAS ou Mach) qu'il cherchera à suivre durant la montée. Ce sont donc là les variables réponses de chacun de nos modèles.

Malheureusement, ni la masse réelle de l'avion, ni le profil de poussée qu'il a effectivement suivi, ni les vitesses-cibles ne sont disponibles dans nos données enregistrées. On ne dispose que d'informations de trajectoires.

Pour l'apprentissage de la masse et des vitesses-cibles, la stratégie adoptée a consisté à ajuster du mieux possible les variables réponses manquantes sur les points "futurs" des trajectoires de nos exemples. Pour le modèle prédisant la masse, la variable réponse $\hat{m}_{11}^{[11:51]}$ est ajustée sur les 41 points postérieurs à un temps de référence $t_0 = 0$ correspondant au passage d'une altitude donnée (voir Figure 5.5), en utilisant la méthode d'estimation présentée en 5.5.4. Pour les modèles prédisant les vitesses-cibles CAS et Mach, on ajuste sur ces mêmes points un couple $(\widehat{cas}, \widehat{M})$ qui minimise l'erreur (RMSE) par rapport à la vitesse-air observée $e^{[11:51]}_{vitesse}(cas, M)$. L'erreur $e^{[i:j]}_{vitesse}$ est donnée par l'équation (5.27), où $f_{(cas, M)}$ est la

TABLE 5.4: Variables explicatives utilisées par les méthodes d'apprentissage.

Groupe	Variables	Description
r	$H_p, \frac{dH_p}{dt}, \frac{d^2H_p}{dt^2}$ $V_g, \frac{dV_g}{dt}$ $V_a, \frac{dV_a}{dt}$ V_{aXY} ΔT W $W_{along}, \frac{dW_{along}}{dt}$ W_{across} θ_c $CAS, \frac{dCAS}{dt}$ $Mach, \frac{dMach}{dt}$ $1/r_{sol}$ $1/r_{air}$ ϕ $e = V_a \frac{dV_a}{dt} + g_0 \frac{T}{T - \Delta T} \frac{dH_p}{dt}$ $ew = e + \vec{W} \cdot \vec{V}_a$	altitude géopotentielle et ses dérivées vitesse sol et sa dérivée vitesse air et sa dérivée vitesse air dans le plan (X,Y) différentiel de température (cf. [129]) vent vent selon V_{aXY} , et sa dérivée vent perpendiculaire à V_{aXY} angle de dérive Calibrated Air Speed et sa dérivée nombre de Mach et sa dérivée courbure par rapport au sol courbure par rapport à l'air angle de roulis variation d'énergie spécifique variation d'énergie spécifique corrigée de l'effet du vent
w	$\Delta T (weather\ grid)$ $W_{along} (weather\ grid)$	différentiel de température sur la grille pour différent H_p vent selon TAS_{XY} sur la grille pour différent H_p
p	RFL Speed distance	niveau de vol demandé dans le plan de vol vitesse demandée dans le plan de vol distance entre les deux aéroports
c	AO DEP ARR	compagnie aérienne aéroport de départ aéroport d'arrivée
\hat{m}	\hat{m}_{LS} \hat{m}_{AD}	masse estimée par méthode des moindres carrés sur les points passés masse estimée par méthode adaptative sur les points passés
t	t	variante de l'appareil

fonction calculant la vitesse-air à partir des CAS et Mach cibles, en fonction de l'altitude et de la température, et où V_{ak} est la vitesse-air observée au point k :

$$e^{[i:j]}_{vitesse}(cas, M) = \sqrt{\frac{1}{j-i+1} \sum_{k=i}^j (f_{(cas, M)}(H_{pk}, T_k) - V_{ak})^2} \quad (5.27)$$

Dans le cas de l'apprentissage du profil de poussée des moteurs, on ne cherche pas à reconstruire artificiellement la variable réponse manquante. La stratégie choisie consiste à ajuster directement une loi de poussée paramétrée de façon à minimiser l'erreur entre la puissance modélisée et les variations d'énergie observées sur les points des trajectoires de nos exemples. La méthode *ad-hoc* utilisée pour réaliser cet ajustement est présentée dans la section 5.5.9.

Dans tous les cas, pour la construction de nos exemples, les variables explicatives présentées dans le tableau 5.4 sont calculées à partir des informations disponibles à t_0 , et notamment des points "passés" de chaque segment de montée. Les variables réponses $\widehat{m}_{11}^{[11:51]}$, \widehat{cas} , et \widehat{M} , ainsi que la variation d'énergie observée utilisée pour apprendre la loi de poussée, sont quant à elles calculées sur les points "futurs", postérieurs à t_0 , de chaque trajectoire.

Dans le cas de l'apprentissage direct de l'altitude future, la variable réponse y est simplement l'altitude $H_{p(t=600s)}$ de l'avion 10 minutes après t_0 . Dans ce cas, on dispose déjà des exemples $H_p^{obs}(t=600s)$ de l'altitude réellement observée, dans les enregistrements de trajectoires.

Dans ce qui suit, l'étude comparative des différentes méthodes appliquées à l'apprentissage de divers paramètres (masse, poussée, etc.), est présentée dans les sections 5.5.7 à 5.5.11. Cette étude comparative porte sur un seul type d'avion, l'Airbus A320, et sur deux jeux d'exemples extraits des données Mode-C et Mode-S. Chaque trajectoire est échantillonnée toutes les 15 secondes, avec un temps de référence $t_0 = 0$ correspondant au passage du niveau de vol $FL180$. Pour chaque trajectoire, on extrait quand c'est possible un segment de montée comprenant 10 points "passés", 1 point courant (à t_0), et 40 points "futurs" après t_0 .

La section 5.5.12 généralise à plusieurs types d'avions et à différentes altitudes initiales, en ne considérant que les données Mode-C, pour lesquelles nous avons le plus de trajectoires. Celles-ci sont échantillonnées avec un temps de référence quelconque, identique pour tous les avions, et on sélectionne pour chaque trajectoire tous les segments possibles de 51 points en montée dont le point courant (le 11^{ième}) est au-dessus d'une altitude minimale donnée.

5.5.7 Apprentissage direct de l'altitude future

Les tableaux 5.5 et 5.6 montrent, pour les données Mode-C et Mode-S respectivement, les résultats comparatifs pour la prévision directe de l'altitude future à $t_0 + 10$ minutes par un modèle statistique, sans utiliser le modèle physique de l'avion. La première ligne de chaque tableau indique pour référence la prévision réalisée avec le modèle BADA d'Eurocontrol et ses paramètres par défaut. Des résultats statistiques plus détaillés et complets sont disponibles dans [7]. Dans l'ensemble de cette section, nous ne présenterons que la RMSE obtenue avec les différentes méthodes, pour plus de clarté.

On constate que les méthodes d'apprentissage améliorent la RMSE de l'altitude d'environ 40 à 45 % pour les données Mode-C, et jusqu'à 50 % pour les données Mode-S.

Parmi les méthodes testées, les trois modèles linéaires MLR-FS, Ridge, et PCR montrent des performances très proches. Le réseau de neurones NNet et la méthode GBM basée sur des arbres de régression boostés sont un peu plus performants.

TABLE 5.5: Résultats des modèles statistiques prévoyant directement l'altitude H_p à $t_0 + 10$ minutes, pour les données Mode-C.

Variables	Méthode	RMSE $\left(H_p^{pred} - H_p^{obs}\right)_{(t=600\text{ s})}$ [ft]	
-	$BADA_{ref}$	1595	
r	MLR-FS	995	} $\simeq 978$ (-38 %)
r	Ridge	991	
r	PCR	990	
r	NNet	965	
r	GBM	948	
$\hat{m}r$	MLR-FS	980	} $\simeq 970$ (-39 %)
$\hat{m}r$	Ridge	983	
$\hat{m}r$	PCR	980	
$\hat{m}r$	NNet	961	
$\hat{m}r$	GBM	939	
$\hat{m}rw$	MLR-FS	914	} $\simeq 911$ (-42 %)
$\hat{m}rw$	Ridge	918	
$\hat{m}rw$	PCR	916	
$\hat{m}rw$	NNet	903	
$\hat{m}rw$	GBM	904	
$\hat{m}prw$	MLR-FS	905	} $\simeq 900$ (-43 %)
$\hat{m}prw$	Ridge	910	
$\hat{m}prw$	PCR	909	
$\hat{m}prw$	NNet	892	
$\hat{m}prw$	GBM	885	
$c\hat{m}prw$	GBM	859	} = 859 (-46 %)

Pour ce qui concerne les différents jeux de variables explicatives testés en entrée des modèles de prévision, les résultats sont à différencier selon les données. Pour les données Mode-C, chaque nouveau groupe de données semble améliorer la prévision. Les gains les plus significatifs sont apportés par les données météorologiques (w), et les données catégorielles (c) issues des plans de vol (compagnie aérienne, aéroport de destination). Pour les données Mode-S, l'ajout des variables explicatives relatives à la masse estimée (\hat{m}) ou aux variables réelles issues de plan de vol (p) ne semblent pas améliorer la prévision. Seules les données catégorielles c et la variante de motorisation t extraite de l'adresse Mode-S de l'avion apportent un gain.

Les meilleurs résultats sont obtenus avec la méthode GBM et le jeu le plus complet de variables.

TABLE 5.6: Résultats des modèles statistiques prévoyant directement l'altitude H_p à $t_0 + 10$ minutes, pour les données Mode-S.

Variables	Méthode	RMSE	
		$(H_p^{pred} - H_p^{obs})_{(t=600 \text{ s})}$	[ft]
<i>BADA_{ref}</i>		1693	
<i>r</i>	MLR-FS	883	} $\simeq 878$ (-48 %)
<i>r</i>	Ridge	895	
<i>r</i>	PCR	884	
<i>r</i>	NNet	880	
<i>r</i>	GBM	848	
$\hat{m}r$	MLR-FS	877	} $\simeq 873$ (-48 %)
$\hat{m}r$	Ridge	880	
$\hat{m}r$	PCR	885	
$\hat{m}r$	NNet	867	
$\hat{m}r$	GBM	856	
$\hat{m}pr$	MLR-FS	877	} $\simeq 875$ (-48 %)
$\hat{m}pr$	Ridge	881	
$\hat{m}pr$	PCR	888	
$\hat{m}pr$	NNet	878	
$\hat{m}pr$	GBM	849	
$\hat{c}mpr$	GBM	841	} = 841 (-50 %)
$\hat{c}mprt$	GBM	830	} = 830 (-50 %)

5.5.8 Apprentissage de la masse

Les résultats pour l'apprentissage de la masse $\hat{m}_{11}^{[11:51]}$ sont présentés dans les tableaux 5.7 et 5.8. Pour alléger la présentation des résultats, on n'a représenté que quelques-uns des jeux de variables explicatives, les résultats complets pouvant être consultés dans [7].

La première ligne donne l'erreur (RMSE) constatée entre la masse BADA de référence et la masse ajustée sur chacune des trajectoires de nos exemples. Les trois lignes suivantes donnent la RMSE obtenue avec les trois méthodes estimant la masse uniquement à partir des points passés (voir section 5.5.4). Les cinq lignes suivantes donnent les résultats des modèles appris sur nos exemples, avec les cinq méthodes d'apprentissage testées (voir section 5.5.6).

On constate que les méthodes d'estimation fournissent une masse plus proche de $\hat{m}_{11}^{[11:51]}$ que ne l'est la masse de référence de BADA : le gain en RMSE est d'environ 45 à 50 %. Ce même gain passe à 60 à 70 % lorsqu'on utilise les méthodes d'apprentissage.

TABLE 5.7: Résultats comparatifs, pour les données Mode-C, sur l'erreur entre la masse estimée ou apprise \tilde{m}_{11} et la masse ajustée de la base d'exemples $\hat{m}_{11}^{[11:51]}$.

Variables	Méthode	RMSE $(\tilde{m}_{11} - \hat{m}_{11}^{[11:51]})$ [kg]	
-	<i>mass_{ref}</i>	4704	
-	adaptative	2186	} $\simeq 2300$ (-51 %)
-	quadratique	2558	
-	pseudo-Huber	2332	
\hat{m}_{prw}	MLR-FS	1498	
\hat{m}_{prw}	Ridge	1504	
\hat{m}_{prw}	PCR	1508	
\hat{m}_{prw}	NNet	1377	} $\simeq 1330$ (-71 %)
$c\hat{m}_{prw}$	GBM	1303	

Légende :

paramètre de référence	
méthode d'estimation	
méthode d'apprentissage	

5.5.9 Apprentissage de la commande de poussée

Principe général de la méthode

L'apprentissage de la loi de poussée des moteurs ne se fait pas avec les méthodes classiques d'apprentissage, et mérite quelques explications. Rappelons que pour le calcul de la puissance $Power(c, m) = (cThr_{max\ climb} - D(m))V_a$, il nous faut connaître d'une part la masse m qui intervient dans le calcul de la traînée D , et d'autre part les valeurs $C = (c_1, \dots, c_n)$ prises par le coefficient de poussée c en chaque point de la trajectoire.

Une des difficultés pour l'apprentissage de la loi de poussée est qu'on a potentiellement une infinité de couples (m, C) qui peuvent expliquer une même trajectoire de montée. Intuitivement, on comprend bien qu'un avion plus lourd avec une poussée des moteurs plus importante peut suivre une trajectoire identique à celle d'un avion plus léger avec une poussée plus faible. Si l'on considère une unique trajectoire, il faut donc nécessairement fixer un des deux paramètres pour pouvoir éventuellement ajuster l'autre. On remarquera d'ailleurs

TABLE 5.8: Résultats comparatifs, pour les données Mode-S, sur l'erreur entre la masse estimée ou apprise \tilde{m}_{11} et la masse ajustée de la base d'exemples $\hat{m}_{11}^{[11:51]}$.

Variables	Méthode	RMSE	
		$(\tilde{m}_{11} - \hat{m}_{11}^{[11:51]})$ [kg]	
-	<i>mass_{ref}</i>	3950	
-	adaptative	2071	} $\simeq 2190$ (-44 %)
-	quadratique	2346	
-	pseudo-Huber	2153	
\hat{m}_{pr}	MLR-FS	1471	} $\simeq 1535$ (-61 %)
\hat{m}_{pr}	Ridge	1561	
\hat{m}_{pr}	PCR	1546	
\hat{m}_{pr}	NNet	1563	
$c\hat{m}_{prt}$	GBM	1342	} = 1342 (-66 %)

Légende :

paramètre de référence	
méthode d'estimation	
méthode d'apprentissage	

que, dans la section 5.5.4, nous faisons une hypothèse sur le profil de poussée (*max climb*, typiquement).

L'idée de l'approche présentée ici est de trouver la meilleure hypothèse possible pour la loi de poussée, en l'apprenant sur une base d'exemples de trajectoires. On suppose que le coefficient de poussée appliqué en chaque point de la trajectoire est une fonction $c(X; \theta)$ de l'état X de l'avion, paramétrée par θ . On fait l'hypothèse que cette loi est commune à tous les avions de même type de notre base d'exemple. Le principe de l'apprentissage de la loi de poussée consiste à ajuster θ de façon à minimiser l'erreur entre la puissance modélisée $Power(c, m)$ et la variation d'énergie observée, sur l'ensemble des trajectoires utilisées pour l'apprentissage.

Minimisation de l'erreur

La minimisation de l'erreur, vue comme une fonction de θ , est réalisée par une méthode de Quasi-Newton (BFGS) qui utilise le gradient de l'erreur pour guider itérativement la recherche vers un optimum *a priori* local. Pour chaque valeur candidate de θ , l'évaluation de l'erreur $\mathcal{E}(\theta)$ nécessite d'estimer la masse m de chacun des avions de l'ensemble d'apprentissage, en utilisant une des méthodes présentées en section 5.5.4 avec une hypothèse de poussée $c(\cdot; \theta)$ correspondant à la valeur candidate de θ .

En reprenant les notations de la section 5.5.4, nous noterons Q la quantité observée, décrite dans l'équation (5.12), correspondant au cumul de la variation d'énergie spécifique et de l'effet du vent. Notons par ailleurs $P(c, m) = Power(c, m) - mQ$ l'écart entre puissance modélisée et variation d'énergie, comme dans l'équation (5.16). L'erreur pour un avion k est alors donnée par l'équation (5.28), analogue à (5.22) de la section 5.5.4 à ceci près qu'elle inclut le vecteur $C_k(\theta) = (c(X_{k,1}; \theta), \dots, c(X_{k,n}; \theta))$ des coefficients de poussée $c(X_{k,i}; \theta)$ calculés en chaque point i de la trajectoire de l'avion k :

$$\mathcal{E}_k(C_k(\theta), m_k) = \sum_{i=1}^n \left(\frac{P_{k,i}(c(X_{k,i}; \theta), m_k + \delta_{k,i})}{m_k + \delta_{k,avg}} \right)^2 \quad (5.28)$$

Dans cette formule, on reprend les notations précédemment introduites dans la section 5.5.4 : $\delta_{k,i}$ est le carburant consommé entre le point i et le point n , $\delta_{k,avg}$ est la moyenne des δ_i sur les n points considérés et m_k est la masse de l'avion k au point n . La masse m_k est toutefois inconnue.

Pour une hypothèse θ donnée sur la poussée, le minimum de $\mathcal{E}_k(C_k(\theta), m_k)$ est atteint pour la masse $m_k^*(\theta)$, donnée par l'équation (5.29), et qu'on peut calculer par une des méthodes d'estimation de la masse décrites en section 5.5.4.

$$m_k^*(\theta) = \underset{m \in]-\delta_{avg}; +\infty[}{\operatorname{argmin}} \mathcal{E}_k(C, m) \quad (5.29)$$

L'erreur totale sur l'ensemble des avions de notre ensemble d'apprentissage, pour une valeur de θ donnée, s'exprime alors par l'équation (5.30) :

$$\mathcal{E}_{total}(\theta) = \sum_{k=1}^K \mathcal{E}_k(C_k(\theta), m_k^*(\theta)), \quad (5.30)$$

avec $C_k(\theta) = (c(X_{k,1}; \theta), \dots, c(X_{k,n}; \theta))$.

Sachant que l'on dispose de l'expression (5.16) du polynôme P en fonction de la masse, et que par ailleurs la masse m_k^* satisfait l'équation $\frac{\partial \mathcal{E}_k}{\partial m}(C_k, m_k^*) = 0$, on peut calculer le gradient de l'erreur totale :

$$\frac{\partial \mathcal{E}_{total}}{\partial \theta_j}(\theta) = \sum_{k=1}^K \frac{\partial \mathcal{E}_k}{\partial \theta_j}(\theta) \quad (5.31)$$

avec :

$$\begin{aligned} \frac{\partial \mathcal{E}_k}{\partial \theta_j}(\theta) &= \frac{\partial \mathcal{E}_k}{\partial m}(C_k, m_k^*) \frac{\partial m_k^*}{\partial \theta_j} + \sum_{i=1}^n \frac{\partial \mathcal{E}_k}{\partial c_{k,i}}(C_k, m_k^*) \frac{\partial c_{k,i}}{\partial \theta_j} \\ &= \sum_{i=1}^n \frac{\partial \mathcal{E}_k}{\partial c_{k,i}}(C_k, m_k^*) \frac{\partial c}{\partial \theta_j}(x_{k,i}; \theta) \\ &= 2 \sum_{i=1}^n \frac{P_{k,i}(c(x_{k,i}; \theta), m_k^* + \delta_{k,i})}{m_k^* + \delta_{k,avg}} \frac{V_{a,k,i} Thr_{k,i}}{m_k^* + \delta_{k,avg}} \frac{\partial c}{\partial \theta_j}(x_{k,i}; \theta). \end{aligned} \quad (5.32)$$

Les expressions de la fonction d'erreur et de son gradient sont utilisées dans la méthode BFGS pour trouver une valeur θ^* du vecteur de paramètres qui minimise l'erreur. La loi de poussée $c(\cdot; \theta^*)$ représente alors la meilleure loi de poussée que l'on a pu trouver, qui minimise l'erreur sur l'ensemble de nos exemples d'apprentissage.

Choix de la fonction paramétrique c

Plusieurs fonctions paramétrées ont été testées dans le cadre de la thèse de Richard Alligier. Elles sont décrites dans les équations (5.33), (5.34), et (5.35), cette dernière permettant de tenir compte de la variante de motorisation de l'avion, présente dans les informations radar Mode-S.

$$c_{H_p}(X; \theta) = \theta_0 + \theta_1 H_p + \theta_2 H_p^2 + \theta_3 H_p^3 + \theta_4 H_p^4 \quad (5.33)$$

$$c_t(X; \theta) = \theta_0 + \theta_1 t + \theta_2 t^2 + \theta_3 t^3 + \theta_4 t^4 \tag{5.34}$$

$$\begin{aligned} c_{\text{variante}[c]}(X; \theta) = & c(X; \theta_{\llbracket 0:4 \rrbracket}) \\ & + \delta_{\text{variante}, \{211, 212\}}(X) c(X; \theta_{\llbracket 5:9 \rrbracket}) \\ & + \delta_{\text{variante}, \{214, 216\}}(X) c(X; \theta_{\llbracket 10:14 \rrbracket}) \end{aligned} \tag{5.35}$$

avec $\delta_{\text{variante}, \text{num}}(X) = \begin{cases} 1 & \text{si } \text{variante}(X) \in \text{num} \\ 0 & \text{sinon} \end{cases}$

Résultats sur l'apprentissage de la poussée

Les tableaux 5.9 et 5.10 donnent les résultats sur l'apprentissage de la loi de poussée des moteurs, pour les données Mode-C et Mode-S respectivement. Dans le but d'alléger la présentation de ces résultats, nous avons omis de reporter les résultats pour les lois c_t et $c_{\text{variante}[c]}$, sensiblement équivalents à ceux obtenus pour c_{H_p} et $c_{\text{variante}[c_{H_p}]}$.

Comme on ne dispose pas de la vraie loi de poussée dans nos exemples, ni même d'exemples reconstruits par estimation, comme pour la masse ou les vitesses-cibles, l'évaluation des méthodes porte uniquement sur le résultat final, c'est-à-dire la prévision de l'altitude future. Afin de nous concentrer uniquement sur l'apport de l'apprentissage de la loi de poussée, nous avons supposé que le profil de vitesses-cibles était parfaitement connu, en prenant $V_a = V_a^{\text{obs}}(t)$ en tout point de la trajectoire future. Les résultats sont comparés d'une part à ceux la méthode BADA avec les paramètres de référence, et d'autre part aux résultats avec le profil de poussée BADA *max climb* combiné avec une méthode d'estimation de masse (voir section 5.5.4).

TABLE 5.9: Résultats comparatifs, pour les données Mode-C, sur l'erreur RMSE de l'altitude prévue à 10 mn, en utilisant diverses combinaisons de masses et de lois de poussée des moteurs, et en supposant le profil de vitesses parfaitement connu ($V_a = V_a^{\text{obs}}(t)$).

Masse	Poussée	Vitesse	RMSE $(H_p^{\text{pred}} - H_p^{\text{obs}})_{(t=600 \text{ s})}$ [ft]	
<i>mass_{ref}</i>	<i>max climb</i>	$V_a^{\text{obs}}(t)$	1481	
adaptative	<i>max climb</i>	$V_a^{\text{obs}}(t)$	749	} $\simeq 790$ (-46 %)
quadratique	<i>max climb</i>	$V_a^{\text{obs}}(t)$	843	
pseudo-Huber	<i>max climb</i>	$V_a^{\text{obs}}(t)$	779	
adaptative	c_{H_p}	$V_a^{\text{obs}}(t)$	828	} = 828 (-44 %)
quadratique	c_{H_p}	$V_a^{\text{obs}}(t)$	671	} $\simeq 640$ (-56 %)
pseudo-Huber	c_{H_p}	$V_a^{\text{obs}}(t)$	614	

Légende :

paramètre de référence	
méthode d'estimation	
méthode d'apprentissage	
valeur observée	

Pour les données Mode-C (tableau 5.9), on constate que l'utilisation de la masse estimée seule améliore d'environ 46 % la RMSE sur l'altitude prévue à 10 minutes, par rapport à la méthode de référence. L'utilisation de la loi de poussée apprise sur les exemples améliore

encore la RMSE de 10 % supplémentaires, portant le gain à 56 % environ, sauf lorsque la loi apprise est combinée avec la masse estimée par la méthode adaptative de Schultz *et al.* ([145]). Avec cette méthode, l'apport du profil appris de poussée ne semble pas significatif, pour les données Mode-C.

TABLE 5.10: Résultats comparatifs, pour les données Mode-S, sur l'erreur RMSE de l'altitude prévue à 10 mn, en utilisant diverses combinaisons de masses et de lois de poussée des moteurs, et en supposant le profil de vitesses parfaitement connu ($V_a = V_a^{obs}(t)$).

Masse	Poussée	Vitesse	RMSE $(H_p^{pred} - H_p^{obs})_{(t=600\text{ s})}$ [ft]
$mass_{ref}$	$max\ climb$	$V_a^{obs}(t)$	1392
adaptative	$max\ climb$	$V_a^{obs}(t)$	781
quadratique	$max\ climb$	$V_a^{obs}(t)$	842
pseudo-Huber	$max\ climb$	$V_a^{obs}(t)$	792
adaptative	c_{H_p}	$V_a^{obs}(t)$	651
quadratique	c_{H_p}	$V_a^{obs}(t)$	629
pseudo-Huber	c_{H_p}	$V_a^{obs}(t)$	570
adaptative	$c_{variante}[c_{H_p}]$	$V_a^{obs}(t)$	585
quadratique	$c_{variante}[c_{H_p}]$	$V_a^{obs}(t)$	582
pseudo-Huber	$c_{variante}[c_{H_p}]$	$V_a^{obs}(t)$	517

Légende :

paramètre de référence	
méthode d'estimation	
méthode d'apprentissage	
valeur observée	

Les ordres de grandeurs pour les résultats sur les données Mode-S (tableau 5.10) sont sensiblement équivalents : 42 % de gain en utilisant la seule masse estimée, 56 % en combinant masse estimée et loi de poussée apprise c_{H_p} , avec cette fois-ci une amélioration quelle que soit la méthode d'estimation de la masse. Les meilleurs résultats (61 %) sont obtenus avec la loi $c_{variante}[c_{H_p}]$ qui tient compte de la variante de motorisation de chaque avion, disponible dans les données Mode-S.

5.5.10 Apprentissage des vitesses-cibles CAS et Mach

Pour l'apprentissage du profil de vitesse suivi durant la montée, on utilise à nouveau des méthodes standards, comme pour l'apprentissage de la masse. Les résultats complets, avec une comparaison avec les \widehat{cas} et \widehat{M} ajustés sur les trajectoires de nos exemples, peuvent être trouvés dans la thèse de Richard Alligier [7]. Nous présentons ici une vue simplifiée de ces résultats, en ne montrant que l'influence de l'apprentissage sur le profil final de vitesse-air, que l'on appelle aussi TAS dans le milieu de l'aviation, et pour quelques combinaisons de variables d'entrée. Cette loi de vitesse-cible pour la *true airspeed* se calcule à partir de la vitesse calibrée CAS et du nombre de Mach par une formule $V_{acible} = f_{(cas,M)}(H_p, T)$ que l'on peut retrouver dans [129] (voir aussi l'annexe D).

Les tableaux 5.11 et 5.12 présentent les résultats pour les données Mode-C et Mode-S respectivement. La première ligne de chaque tableau donne l'erreur RMSE entre la vitesse-air $V_{acible} = f_{(\widehat{cas}, \widehat{M})}$ calculée avec les paramètres $(\widehat{cas}, \widehat{M})$ ajustés les exemples, et la vitesse-

air observée V_a^{obs} . Avec la modélisation choisie pour le profil de vitesse (CAS constante puis Mach constant au-dessus de l'altitude de conjonction) on sait qu'on ne pourra pas faire mieux que l'erreur reportée sur la première ligne, quelle que soit la méthode d'apprentissage. Les méthodes d'apprentissage sont également comparées au profil de vitesse obtenu avec les CAS et Mach moyens observés (2^{ième} ligne des tableaux), ou avec la CAS observée à l'instant $t = 0$ de la prévision (3^{ième} ligne). La quatrième ligne montre l'erreur en vitesse obtenue avec les CAS et Mach de référence du modèle BADA.

TABLE 5.11: pour les trajectoires Mode-C.

Variables	Méthode	RMSE
		$f(cas, M) - V_a^{obs}$ [kts]
-	$f(\widehat{cas}, \widehat{M})$	9.29
-	$f(\overline{cas}, \overline{M})$	23.8
-	$f(cas(t=0), \overline{M})$	19
-	$f(cas_{BADA}, M_{BADA})$	23.7
$\hat{m}prw$	MLR-FS	15.6
$\hat{m}prw$	Ridge	15.6
$\hat{m}prw$	PCR	15.8
$\hat{m}prw$	NNet	15.5
$\hat{m}prw$	GBM	15.2
$c\hat{m}prw$	GBM	15.1

TABLE 5.12: pour les trajectoires Mode-S.

Variables	Méthode	RMSE
		$f(cas, M) - V_a^{obs}$ [kts]
-	$f(\widehat{cas}, \widehat{M})$	8.32
-	$f(\overline{cas}, \overline{M})$	21.9
-	$f(cas(t=0), \overline{M})$	18.1
-	$f(cas_{BADA}, M_{BADA})$	27.1
$\hat{m}prw$	MLR-FS	14.6
$\hat{m}prw$	Ridge	14.6
$\hat{m}prw$	PCR	14.4
$\hat{m}prw$	NNet	14
$\hat{m}prw$	GBM	14
$c\hat{m}pr$	GBM	14
$c\hat{m}prt$	GBM	14.1

On constate sur les deux tableaux 5.11 et 5.12 que l'utilisation des vitesses-cibles CAS et Mach apprises améliorent d'au moins 35% la RMSE sur le profil de vitesse-air, si l'on compare aux paramètres CAS et Mach de référence du modèle BADA. L'amélioration reste significative si on compare au calcul fait avec les valeurs moyennes ou initiales observées. Les meilleurs résultats sont obtenus avec les méthodes NNet et GBM.

5.5.11 Résultats comparatifs pour la prévision de l'altitude

Dans les sections précédentes, nous avons examiné l'apport de méthodes d'apprentissage sur chacun des paramètres (masse, poussée, vitesses-cibles), pris individuellement. Dans cette section, nous comparons diverses combinaisons des méthodes de référence, d'estimation, ou d'apprentissage des différents paramètres, pour la prévision de l'altitude à 10 minutes.

Les tableaux 5.13 pour les données Mode-C, et 5.14 pour les données Mode-S, ne présentent que les combinaisons des méthodes ayant donné les meilleurs résultats. L'objectif est de comparer le modèle BADA renseigné par les paramètres appris ou estimés avec le modèle BADA utilisé avec les valeurs de référence (1^{ière} ligne des tableaux), mais aussi avec le modèle purement statistique de prévision de l'altitude à 10 minutes (dernière ligne) dont les résultats sont donnés en section 5.5.7.

TABLE 5.13: Résultats comparatifs, pour les données Mode-C, sur la RMSE de l'altitude prévue à 10 minutes (en pieds).

Modèle	Masse	Poussée	Vitesse	RMSE $(H_p^{pred} - H_p^{obs})_{(600\text{ s})}$ [ft]	
$BADA_{ref}$	$mass_{ref}$	max	ref	1595	
BADA renseigné	pseudo-Huber	max	GBM	989	} = 989 (-37 %)
	pseudo-Huber	c_{H_p}	GBM	935	} = 935 (-41 %)
	GBM	max	GBM	826	} = 826 (-48 %)
statistique	prévision directe de H_p par GBM			859	} = 859 (-46 %)

Légende :

paramètre de référence	
méthode d'estimation	
méthode d'apprentissage	

TABLE 5.14: Résultats comparatifs, pour les données Mode-S, sur la RMSE de l'altitude prévue à 10 minutes (en pieds).

Modèle	Masse	Poussée	Vitesse	RMSE $(H_p^{pred} - H_p^{obs})_{(600\text{ s})}$ [ft]	
$BADA_{ref}$	$mass_{ref}$	max	ref	1693	
BADA renseigné	pseudo-Huber	max	GBM	955	} = 955 (-43 %)
	pseudo-Huber	c_{H_p}	GBM	851	} \simeq 830 (-50 %)
	pseudo-Huber	$c_{variante}$	GBM	821	
	GBM	max	GBM	766	} = 766 (-54 %)
statistique	prévision directe de H_p par GBM			830	} = 830 (-50 %)

Légende :

paramètre de référence	
méthode d'estimation	
méthode d'apprentissage	

Les gains en RMSE apportés soit par les paramètres appris ou estimés, soit par l'utilisation d'un modèle purement statistique, sont tout à fait considérables : de l'ordre de 40 à 55 % d'amélioration sur la prévision de l'altitude. Pour les données Mode-C comme pour les données Mode-S, les meilleurs résultats sont obtenus en apprenant la masse et le profil de vitesse par la méthode GBM, en faisant une hypothèse de poussée *max climb*. Le modèle physique BADA

renseigné avec ces paramètres appris est donc bien meilleur que le même modèle physique utilisé avec les paramètres de référence, et ses performances sont un peu supérieures à celles du modèle statistique.

L'avantage du modèle BADA renseigné, par rapport à l'approche purement statistique, est que même si une des hypothèses faites sur un des paramètres du modèle est fautive, le modèle physique lui-même reste valable et utilisable (au prix d'imprécisions éventuelles), quelle que soit la loi de commande de poussée ou la loi de vitesses-cibles qu'on lui fournit en entrée. Si on constate qu'un des paramètres ou une des lois de commande (poussée, vitesse-cible) semblent erronées, on peut toujours se ramener aux paramètres par défaut, ou à d'autres paramètres estimés par des méthodes différentes, sans changer le modèle physique de base. Un modèle statistique doit être appris sur des données suffisamment représentatives du phénomène observé. Il faudrait donc, pour une réelle mise en œuvre opérationnelle d'un modèle prévoyant directement l'altitude, pouvoir apprendre des modèles différenciés selon, par exemple, le type de profil de vitesses-cible (CAS/Mach constants, ou bien taux de montée constant, etc). L'utilisation d'un modèle physique BADA, renseigné par les paramètres appris, est donc comparativement plus facile à mettre en œuvre.

5.5.12 Généralisation à différents types d'avions et différentes altitudes initiales

Dans tout ce qui précède, l'étude portait sur un seul type d'avion (Airbus A320), et une seule altitude initiale : le temps $t_0 = 0$ correspondait au passage du niveau de vol *FL180*. De plus, on ne s'intéressait qu'à la prévision de l'altitude à 10 minutes. Dans cette section, les résultats sont étendus à plusieurs types d'avions et à différentes altitudes initiales, et on cherche à caractériser l'incertitude sur la prévision d'altitude sur tout l'horizon de prévision, entre $t_0 = 0$ et $t_0 + 10$ minutes.

La construction des exemples pour l'apprentissage diffère légèrement de celle des exemples précédents pour l'A320. Ici, les trajectoires sont échantillonnées toutes les 15 secondes en prenant pour référence une date fixe quelconque, et non plus à partir d'un temps t_0 correspondant au passage du niveau de vol *FL180*. A partir de chaque trajectoire échantillonnée, on construit ensuite autant de segments de montée que possible, chaque segment comprenant 10 points "passés", 1 point courant, et 40 points "futurs". On ne retient que les segments dont le point courant à t_0 est au-dessus d'une altitude fixée. Par défaut, on choisit le niveau de vol *FL180*, mais pour certains types d'avions pour lesquels il y a très peu de segments en montée de 10 minutes après ce niveau de vol, on retiendra tous les segments dont le point courant est au-dessus du niveau *FL150*.

Chaque trajectoire étant susceptible de fournir plusieurs segments en montée, on a pris soin, lors de la construction des ensembles d'apprentissage et de validation, que tous les segments d'une même trajectoire se retrouvent soit dans l'un, soit dans l'autre, afin de garantir l'indépendance des deux ensembles.

Une autre différence, par rapport à ce qui a été présenté précédemment, est que l'altitude z_0 à t_0 fait maintenant partie des entrées des modèles de prévision.

La Figure 5.13 présente les cônes d'incertitude pour l'altitude prévue entre t_0 et $t_0 + 10$ minutes, pour les données Mode-C des avions au départ de la région Parisienne. Ces résultats sont obtenus, pour 6 types d'avions aux caractéristiques différentes, en appliquant la méthode GBM avec les variables $c\hat{m}prw$ plus l'altitude z_0 en entrée, pour apprendre la masse $\hat{m}_{11}^{[11:51]}$ et les vitesses cibles CAS et Mach. Pour chaque type d'avion, la Figure 5.13 montre

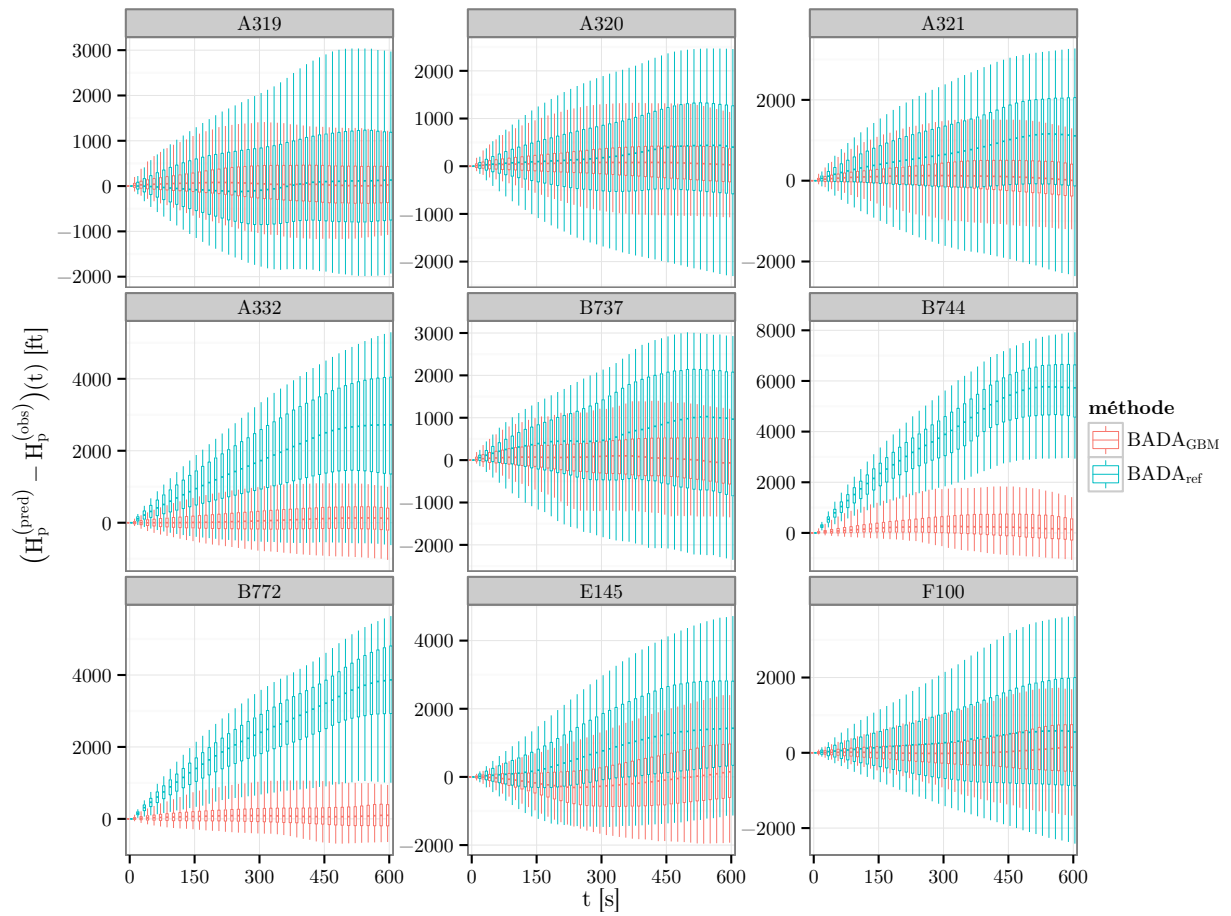


FIGURE 5.13: Prédiction de l'altitude future pour plusieurs types d'avion.

le cône d'incertitude obtenu avec $BADA_{ref}$, en bleu, et celui obtenu avec BADA renseigné par les paramètres appris par la méthode GBM, en rouge. Chaque cône est constitué d'un ensemble de boîtes à moustaches, pour chaque temps t entre t_0 et $t_0 + 10$ minutes. Ces boîtes à moustaches représentent les quantiles 5 %, 25 %, 50 % (la valeur centrale), 75 %, et 95 %. Visuellement, on voit en fait deux cônes par couleur (bleue ou rouge), imbriqués l'un dans l'autre. Le cône le plus étroit contient 50 % des données, le plus large en contient 90 %.

On constate sur la Figure 5.13 que les cônes d'incertitude de la méthode $BADA_{GBM}$ sont plus étroits que ceux de la méthode de référence $BADA_{ref}$. Pour les types A332, B772, et B744, la médiane reste beaucoup mieux centrée avec $BADA_{GBM}$, sur l'ensemble de la prévision. Les cônes obtenus avec $BADA_{ref}$ sont par contre très excentrés. Ces trois types d'avions correspondent à des long-courriers. Au départ de la région parisienne, ces vols sont généralement soit des trans-atlantiques très chargés en carburant, soit des vols internationaux vers des destinations lointaines. La masse par défaut $mass_{ref}$ de BADA sous-estime donc assez largement la masse réelle de ces avions, et le gain apporté par l'apprentissage des paramètres du modèle physique de l'avion est ici particulièrement flagrant.

Les résultats numériques (voir [7]) montrent que la RMSE de l'erreur de prévision d'altitude est améliorée d'au moins 45 % en utilisant $BADA_{GBM}$ plutôt que $BADA_{ref}$, quel que soit le type d'avion considéré. Pour les avions particulièrement mal modélisés par les valeurs BADA par défaut, ce gain peut monter jusqu'à plus de 85 %.

5.6 Conclusion sur l'exploitation des trajectoires d'avion

A travers ces travaux de recherche sur l'extraction du vent d'une part, et la prévision de l'altitude de l'avion en montée, nous avons montré tout l'intérêt d'appliquer des méthodes d'estimation ou d'apprentissage supervisé aux données de trajectoires.

Notamment, les gains obtenus sur la précision de la prévision d'altitude sont substantiels, si l'on compare à la méthode de référence aujourd'hui largement utilisée en opérationnel. Nous avons par ailleurs montré que l'apprentissage supervisé de modèles fournissant les paramètres manquants du modèle physique de l'avion (masse, poussée, vitesses-cibles), et l'utilisation de données supplémentaires (ex. distance à parcourir), non prises en compte dans la seule dynamique de l'avion, permettaient d'améliorer significativement la prévision, par rapport à la méthode de référence, mais aussi par rapport aux méthodes estimant ces paramètres à partir des seuls points passés de la trajectoire en cours.

Ces résultats nous semblent ouvrir des perspectives intéressantes en termes d'utilisation opérationnelle, les gains obtenus justifiant à nos yeux les efforts de collecte, de traitement de données, et de calibration des modèles, indispensables à ce type d'approche.

Conclusion

La gestion du trafic aérien est un domaine passionnant, en constante évolution, qui doit faire face à un trafic globalement en croissance ininterrompue depuis les débuts de l'aviation commerciale. Le système ATM n'a cessé de repousser les limites de sa capacité à traiter ce trafic, en adoptant des solutions opérationnelles et techniques innovantes : automatisation croissante de certaines tâches comme le traitement des plans de vol ou la coordination entre secteurs de contrôle, améliorations de la chaîne de traitement radar et des visualisations, mise en place de liaisons de données sol-bord, diminution des normes de séparation latérales et verticales, refonte du réseau de routes, aménagements des frontières des espaces aériens, mise en place d'organismes de coordination pour la gestion des flux de trafic, etc. Plus récemment, c'est essentiellement sur une harmonisation à l'échelle continentale des concepts opérationnels et des systèmes techniques qu'ont portés les efforts, à travers de grands programmes comme NextGen aux États-Unis, ou SESAR en Europe. Ces efforts de modernisation ont mobilisé des budgets importants, avec pour objectif une amélioration de la performance globale et de la sécurité du système ATM, afin de rendre un meilleur service aux compagnies aériennes et *in fine* aux passagers. Dans ce contexte, les travaux de recherche récapitulés dans ce mémoire sont susceptibles d'apporter des améliorations significatives sur des points-clés du système, et en particulier sur les deux suivants : la modélisation de la charge de travail des contrôleurs aériens et la gestion prévisionnelle des ouvertures de secteurs ATC d'une part, et la prévision de trajectoires d'avions par les systèmes au sol d'autre part.

Pour ce qui concerne la charge de travail et la prévision des ouvertures de secteurs aériens, les travaux récapitulés au chapitre 4 montrent qu'une méthode de recherche arborescente combinée avec un réseau de neurones permet une amélioration significative de la fiabilité de la prévision du nombre de postes de travail nécessaires pour traiter un trafic donné (voir Figure 4.7). La méthode de recherche arborescente explore l'ensemble des partitions possibles de l'espace aérien en secteurs ATC en tenant compte de contraintes opérationnelles diverses. Elle diffère en cela de la pratique opérationnelle actuelle, qui s'appuie sur un nombre restreint de configurations pré-définies de secteurs. Le réseau de neurones, appris à partir d'archives de trafic et d'ouvertures de secteurs, modélise la charge de travail effective des contrôleurs. Il intervient dans le critère de décision de reconfiguration de l'espace, et dans le critère de coût associé à une configuration de secteurs ATC. La maquette d'interface homme-machine montrant les ouvertures prévues et l'évolution de la charge dans chaque secteur a été conçue en concertation étroite avec des experts ATC. Elle a été perçue positivement par ceux-ci, et a permis de montrer, sur des données statiques, comment les algorithmes proposés pouvaient permettre d'anticiper quels secteurs ATC allaient être surchargés. Une anticipation correcte de telles surcharges est indispensable pour mettre en place à temps des mesures de régulation du trafic adaptées à la situation. L'ensemble de ces travaux a constitué la contribution initiale de la DSNA française à trois projets ("*work packages*") du programme européen SESAR de

modernisation des systèmes ATM et ATC : “*Complexity management in en-route*” (WP 4.7.1), “*Complexity assessment and resolution*” (WP 10.8.1), et “*Dynamic airspace configuration*” (WP 7.5.4). Dans ce cadre, un partenariat avec l’industriel Thalès a été mis en place pour le développement d’un prototype pré-industriel qui devait permettre la démonstration et la validation opérationnelle de nos algorithmes, en les connectant au prédicteur de trajectoires de l’industriel.

Sur la prévision des trajectoires d’avions par des systèmes au sol, si l’on résume en quelques mots le travail de thèse de Richard Alligier présenté au chapitre 5, on obtient une amélioration d’au moins 45% de la précision de l’altitude prévue à 10 minutes, par rapport au modèle BADA aujourd’hui largement utilisé en opérationnel. Ce gain est obtenu en appliquant des techniques d’apprentissage supervisé pour construire des modèles estimant les valeurs des paramètres manquants du modèle physique de l’avion. Cette amélioration monte jusqu’à plus de 85%, pour les avions particulièrement mal modélisés par les valeurs par défaut de BADA. Sachant que les concepts opérationnels développés dans les programmes NextGen et SESAR s’appuient tous sur un concept de trajectoire-4D de grande précision devant notamment alimenter les outils d’aide à la détection et à la résolution de conflits aériens, on conçoit l’importance de disposer effectivement d’une prévision précise des trajectoires. Ce n’est pas le cas à ce jour, en particulier pour la phase de montée.

Les travaux de recherche présentés dans ce mémoire s’appuient, en termes de modélisation, sur des hypothèses du monde d’aujourd’hui, sans extrapoler sur la mise en place à plus ou moins brève échéance de nouvelles fonctionnalités venant opportunément résoudre un certain nombre de difficultés dans la résolution des problèmes traités. Cette approche pragmatique et cet ancrage dans l’opérationnel, que je dois en partie à mes débuts de carrière dans un centre de contrôle aérien, se justifie par ailleurs par le fait que ces recherches ont été réalisées ou ont débuté au sein de la DSNA, le fournisseur français des services de la navigation aérienne. La dernière réorganisation des activités de recherche à la DGAC, avec leur regroupement à l’ÉNAC, est l’occasion de m’intéresser un peu plus aux méthodes, comme nous avons pu le faire avec l’hybridation de métaheuristiques et de méthodes d’intervalles, tout en gardant les thématiques ATM comme fil conducteur de ma recherche.

Appendices

Annexe A

Méthodes BFGS et L-BFGS

A.1 Principe général des méthodes de Quasi-Newton

Les méthodes de BFGS et L-BFGS font partie des méthodes de Quasi-Newton, qui sont elles-mêmes des méthodes de descente dont le principe est rappelé dans la section 2.4 du chapitre 2.

Dans les méthodes de Quasi-Newton, la fonction-objectif f est approximée localement au voisinage de l'optimum x^* (local, si f n'est pas convexe) par un modèle quadratique. Ce modèle fait appel à une matrice B_k , qui est une approximation de la matrice hessienne $\nabla^2 f(x_k)$. La fonction quadratique est strictement convexe si B_k est définie positive. Une alternative classique consiste à approximer directement l'inverse de la hessienne $[\nabla^2 f(x)]^{-1}$ par une matrice H_k . Cette approximation permet d'obtenir une direction de descente $d_k = -H_k \nabla f(x_k)$ (ou bien $d_k = -B_k^{-1} \nabla f(x_k)$ si l'on approxime la hessienne) menant plus rapidement vers l'optimum supposé.

Algorithm 17 Principe général des méthodes de Quasi-Newton.

```
1:  $x_0 \leftarrow \text{INIT}(\ )$ 
2:  $H_0 \leftarrow \text{INITINVHESSIAN}(\ )$ 
3: while not(STOP( $k, x_k$ )) do
4:    $d_k = -H_k \nabla f(x_k)$  ▷ Détermination de la direction de descente
5:    $\alpha_k \leftarrow \text{LINESEARCH}(f, d_k)$  ▷ Sélection d'un pas satisfaisant les conditions de Wolfe
6:    $x_{k+1} \leftarrow x_k + \alpha_k d_k$ 
7:    $H_{k+1} \leftarrow \text{UPDATE}(H_k, x_{k+1}, x_k)$  ▷ Dans le cas où on approxime l'inverse de la hessienne
8:    $k \leftarrow k + 1$ 
9: end while
10: return ( $x_k, f(x_k)$ )
```

Chaque itération de l'algorithme consiste alors à déterminer cette direction de descente, puis à effectuer un pas de longueur α_k dans cette direction. Pour garantir les propriétés de convergence de la méthode, le pas α_k sélectionné doit vérifier un certain nombre de conditions (longueur minimale, décroissance de f dans la direction d_k). Pour les méthodes de Quasi-Newton, il s'agit des conditions de Wolfe. Ces différentes étapes sont résumées dans l'algorithme 17.

Le reste de cette annexe détaille les différents points que nous venons d'évoquer, ainsi que les méthodes reposant sur la mise à jour de *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) de la

matrice H_k (ou B_k), généralement reconnue comme la plus performante.

A.1.1 Approximations quadratiques au voisinage d'un optimum local

Développement de Taylor en x^*

L'approximation de la fonction-objectif f au voisinage du point x^* s'obtient par un développement de Taylor, donné par l'équation (A.1) suivante :

$$f(x) \approx f(x^*) + (x - x^*)^T \nabla f(x^*) + \frac{1}{2} (x - x^*)^T \nabla^2 f(x^*) (x - x^*) \quad (\text{A.1})$$

Le point x^* étant stationnaire ($\nabla f(x^*) = 0$), l'approximation au voisinage de l'optimum local se réduit à l'équation (A.2) :

$$f(x) \approx f(x^*) + \frac{1}{2} (x - x^*)^T \nabla^2 f(x^*) (x - x^*) \quad (\text{A.2})$$

En dérivant l'équation (A.2) et en tenant compte du fait que $\nabla f(x^*) = 0$, on obtient la relation (A.3) suivante entre le gradient de f au point x et la hessienne au point x^* :

$$\nabla f(x) \approx \nabla^2 f(x^*) (x - x^*) \quad (\text{A.3})$$

Si la hessienne $\nabla^2 f(x^*)$ est définie positive, on peut alors écrire l'équation suivante :

$$x^* \approx x - [\nabla^2 f(x^*)]^{-1} \nabla f(x) \quad (\text{A.4})$$

On voit que, dans ces conditions idéales et en supposant qu'on connaisse l'inverse de la hessienne $[\nabla^2 f(x^*)]^{-1}$, un seul pas de longueur 1 dans la direction de Newton $d = -[\nabla^2 f(x^*)]^{-1} \nabla f(x)$ permet d'atteindre l'optimum x^* . Ces considérations justifient notamment d'essayer en premier lieu un pas de longueur 1 dans les algorithmes de *line search* chargés de déterminer la longueur α_k du pas à réaliser dans la direction de descente $d_k = -H_k \nabla f(x_k)$, où H_k est une approximation de l'inverse de la hessienne.

Dans la réalité, on ne connaît pas la hessienne $\nabla^2 f(x^*)$, et on ne dispose que des informations collectées à l'itération courante k et aux itérations passées.

Développement de Taylor en x_k

En faisant à nouveau un développement de Taylor, cette fois-ci en considérant deux itérations successives x_k et x_{k+1} , on obtient la relation suivante :

$$\begin{aligned} f(x_{k+1}) &= f(x_k) + (x_{k+1} - x_k)^T \nabla f(x_k) \\ &\quad + \frac{1}{2} (x_{k+1} - x_k)^T \nabla^2 f(x_k) (x_{k+1} - x_k) \\ &\quad + o(\|x_{k+1} - x_k\|^2) \end{aligned} \quad (\text{A.5})$$

Une approximation du premier ordre du gradient $\nabla f(x_{k+1})$ permet d'obtenir l'équation (A.6) suivante :

$$\nabla f(x_{k+1}) - \nabla f(x_k) \approx \nabla^2 f(x_k) (x_{k+1} - x_k) \quad (\text{A.6})$$

Afin de rendre plus lisibles les équations qui suivent, adoptons les notations suivantes :

$$p_k = x_{k+1} - x_k \quad (\text{A.7a})$$

$$q_k = \nabla f(x_{k+1}) - \nabla f(x_k) \quad (\text{A.7b})$$

Faisons momentanément l'hypothèse que $\nabla^2 f(x_k) = F$ est constante, au moins dans le voisinage des points qui nous intéressent. Avec cette hypothèse et les notations adoptées, l'équation (A.6) s'exprime simplement par l'équation (A.8) suivante :

$$q_k \approx F p_k \quad (\text{A.8})$$

Supposons qu'on connaisse $n + 1$ points x_0, \dots, x_n tels que les directions $p_k = x_{k+1} - x_k$ soient linéairement indépendantes, pour $k = 0, \dots, n - 1$. On peut alors construire la matrice P dont les vecteurs colonnes sont les directions p_0, \dots, p_{n-1} , et la matrice Q dont les colonnes sont les q_0, \dots, q_{n-1} . La hessienne F est alors déterminée de manière unique par l'équation suivante :

$$F = Q P^{-1} \quad (\text{A.9})$$

Dans la pratique, on n'attendra pas de collecter $n + 1$ points pour estimer F . On cherchera à l'approximer dès les premières itérations, en utilisant les informations sur le gradient collectées à chaque pas de calcul.

Plutôt que d'approximer F , on peut aussi chercher à approximer son inverse. Si F est inversible, la direction $p_k = x_{k+1} - x_k$ s'exprime en fonction de l'inverse de la hessienne et de la différence des gradients $q_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ de la façon suivante :

$$p_k \approx F^{-1} q_k \quad (\text{A.10})$$

A.1.2 Approximation itérative de la hessienne ou de son inverse

Les méthodes de Quasi-Newton approximent itérativement la hessienne par une matrice B_k symétrique, définie positive, ou son inverse par une matrice H_k . L'inverse F^{-1} de la hessienne est approximée par une matrice H_{k+1} vérifiant la condition suivante, analogue à (A.10), mais valable pour toutes les itérations i précédant l'itération courante :

$$p_i \approx H_{k+1} q_i \quad , \forall i \in \{0, \dots, k\} \quad (\text{A.11})$$

Pour $k < n$, la détermination d'une matrice H_{k+1} vérifiant (A.11) n'est pas unique. On choisit donc celle qui satisfait un certain nombre de conditions, comme par exemple d'être définie positive si H_k l'est aussi, et d'être la plus proche possible de H_k , au sens d'une norme matricielle choisie (ex. norme de Frobenius pondérée).

Le principe de l'approximation itérative de F^{-1} consiste donc, à partir d'une matrice initiale H_0 définie positive, à mettre à jour l'approximation à chaque itération en calculant H_{k+1} en fonction de H_k , p_k , et q_k . Les formules de mise à jour les plus connues sont celles de *Davidon-Fletcher-Powell* (DFP), et celles de *Broyden-Fletcher-Goldfarb-Shanno* (BFGS).

La formule BFGS est généralement reconnue comme donnant les meilleurs résultats. Elle est donnée par l'équation (A.12) suivante (voir [110] ou [43] pour le calcul de l'expression) :

$$\begin{aligned} \mathbf{H}_{k+1}^{\text{BFGS}} &= \left(\mathbf{I} - \frac{\mathbf{p}_k \mathbf{q}_k^T}{\mathbf{q}_k^T \mathbf{p}_k} \right) \mathbf{H}_k \left(\mathbf{I} - \frac{\mathbf{q}_k \mathbf{p}_k^T}{\mathbf{q}_k^T \mathbf{p}_k} \right) + \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{q}_k^T \mathbf{p}_k} \\ &= \mathbf{H}_k + \left(\frac{1 + \mathbf{q}_k^T \mathbf{H}_k \mathbf{q}_k}{\mathbf{q}_k^T \mathbf{p}_k} \right) \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{q}_k^T \mathbf{p}_k} - \frac{\mathbf{p}_k \mathbf{q}_k^T \mathbf{H}_k + \mathbf{H}_k \mathbf{q}_k \mathbf{p}_k^T}{\mathbf{q}_k^T \mathbf{p}_k} \end{aligned} \quad (\text{A.12})$$

A.2 La méthode BFGS

L'équation (A.4) pourrait nous laisser penser que si l'on dispose d'une bonne approximation \mathbf{H}_k de la hessienne $\nabla^2 f(\mathbf{x}^*)$, supposée constante, égale à \mathbf{F} , au voisinage de l'optimum local \mathbf{x}^* , alors il suffit de faire un pas $\mathbf{x}^* = \mathbf{x}_k - \mathbf{H}_k \nabla f(\mathbf{x}_k)$ pour trouver l'optimum \mathbf{x}^* .

En pratique, le choix du point initial \mathbf{x}_0 ne nous place pas nécessairement dans un voisinage de \mathbf{x}^* . De plus, l'approximation quadratique que l'on a fait pour obtenir (A.4) reste une approximation, par nature inexacte. Enfin, pour la grande majorité des problèmes que l'on pourra rencontrer, l'hypothèse d'une hessienne constante, que nous avons faite pour justifier l'approximation \mathbf{H}_k de \mathbf{F}^{-1} , n'est généralement pas vérifiée.

Malgré tout, la direction $\mathbf{d}_k = -\mathbf{H}_k \nabla f(\mathbf{x}_k)$ reste dans tous les cas une direction de descente quand \mathbf{H}_k est définie positive. En effet, $\nabla f(\mathbf{x}_k)^T \mathbf{d}_k = -\nabla f(\mathbf{x}_k)^T \mathbf{H}_k \nabla f(\mathbf{x}_k) < 0$. Le principe général des méthodes de Quasi-Newton, dont la méthode BFGS, consiste donc à effectuer à chaque itération un pas de longueur α_k dans cette direction \mathbf{d}_k .

L'algorithme 18 récapitule les différentes étapes de la méthode BFGS. Le point initial \mathbf{x}_0 est généralement choisi au hasard. La matrice initiale \mathbf{H}_0 est choisie définie positive, afin que les approximations successives \mathbf{H}_k le soient aussi, garantissant ainsi qu'on suit bien une direction de descente à chaque itération. Pour s'assurer que la méthode BFGS converge super-linéairement vers un optimum local en un nombre fini d'itérations, le pas α_k doit vérifier les conditions de Wolfe (voir Wolfe [176], ou [127], p. 33).

Algorithm 18 Principe de la méthode BFGS.

- 1: $k \leftarrow 0$
- 2: $\mathbf{x}_0 \leftarrow \text{INIT}(\)$
- 3: $\mathbf{H}_0 \leftarrow \text{INITINVHESSIAN}(\)$
- 4: **while** not(STOP(k, \mathbf{x}_k)) **do**
- 5: $\mathbf{d}_k = -\mathbf{H}_k \nabla f(\mathbf{x}_k)$ ▷ Détermination de la direction de descente
- 6: $\alpha_k \leftarrow \text{LINESEARCH}(f, \mathbf{d}_k)$ ▷ Sélection d'un pas satisfaisant les conditions de Wolfe
- 7: $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{d}_k$
- 8: $\mathbf{p}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$
- 9: $\mathbf{q}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$
- 10: Mettre à jour l'approximation de l'inverse de la hessienne

$$\begin{aligned} \mathbf{H}_{k+1}^{\text{BFGS}} &= \mathbf{V}_k^T \mathbf{H}_k \mathbf{V}_k + \rho_k \mathbf{p}_k \mathbf{p}_k^T \\ \text{avec } \rho_k &= \frac{1}{\mathbf{q}_k^T \mathbf{p}_k} \quad , \quad \mathbf{V}_k = \mathbf{I} - \rho_k \mathbf{q}_k \mathbf{p}_k^T \end{aligned} \quad (\text{A.13})$$

- 11: $k \leftarrow k + 1$
 - 12: **end while**
 - 13: Retourner ($\mathbf{x}_k, f(\mathbf{x}_k)$)
-

Il peut arriver que la recherche unidimensionnelle ne permette pas de trouver un pas α_k améliorant significativement la fonction f dans la direction d_k choisie. Dans ces cas-là, on réinitialise la valeur courante de H_k à H_0 et on redémarre l'algorithme 18 à partir du point courant x_k . Avec H_0 proportionnelle à la matrice identité, cela revient à prendre une direction de plus grande pente.

A.2.1 Choix de la matrice initiale H_0

La matrice initiale H_0 doit être définie positive. Un choix usuel consiste à prendre $H_0 = \gamma I$. Le paramètre γ peut éventuellement être choisi par l'utilisateur, en fonction du problème traité. Il peut aussi être fixé automatiquement, selon de la courbure observée au point initial.

Une heuristique efficace (voir [127], p. 143, ou [75]) consiste à faire le premier pas de descente avec la matrice identité $H_0 = I$ (dans la direction de plus grande pente), puis de calculer \hat{H}_0 selon la formule (A.14) suivante, qui sert ensuite dans le calcul de la première mise à jour H_1^{BFGS} de l'approximation de l'inverse de la hessienne.

$$\hat{H}_0 = \frac{q_0^T p_0}{q_0^T q_0} I \quad (\text{A.14})$$

A.2.2 Critère d'arrêt

Le critère d'arrêt (STOP dans l'algorithme 18) peut prendre plusieurs formes. Il peut porter sur la valeur du gradient, avec une expression du type $\|\nabla f(x_k)\| \leq \epsilon$, ou bien sur deux valeurs successives d'évaluations de f , avec un critère du type $\left| \frac{f(x_k) - f(x_{k+1})}{f(x_k)} \right| \leq \text{reltol}$. Dans ce dernier cas, on complète le critère d'arrêt, pour les cas où $f(x)$ converge vers un minimum égal à 0, par un critère absolu du type $|f(x_k)| \leq \text{abstol}$. Par ailleurs, en règle générale, on se limite à un nombre maximum d'itérations.

A.2.3 Recherche unidimensionnelle (*line search*)

La sélection de la longueur α_k du pas effectué dans la direction d_k se fait par une méthode itérative, en minimisant la fonction $\Phi : \alpha \rightarrow f(x + \alpha_k d_k)$. Contrairement à la méthode des gradients conjugués, il n'est pas nécessaire pour les méthodes de Quasi-Newton d'obtenir une valeur très précise du minimum de cette fonction.

La recherche unidimensionnelle peut donc être inexacte. Cependant son critère d'arrêt doit vérifier certaines conditions pour assurer les propriétés de convergence super-linéaire de la méthode de Quasi-Newton au voisinage d'un optimum local x^* . Le pas α_k sélectionné doit satisfaire les conditions de Wolfe, données par les équations (A.15) suivantes :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^T d_k \quad (\text{A.15a})$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2 \nabla f(x_k)^T d_k \quad (\text{A.15b})$$

Ces conditions expriment des exigences de pentes minimales et maximales, fixées par les coefficients c_1 et c_2 , que doivent satisfaire les portions de la courbe $\Phi(\alpha)$ pour que le pas α soit admissible.

Dans notre implantation en Ocaml des méthodes BFGS et L-BFGS, nous avons utilisé un algorithme de bisection qui teste récursivement des intervalles de valeurs de α jusqu'à satisfaire les conditions de Wolfe.

A.2.4 Implantation et complexité de l'algorithme BFGS

Implanter tel quel l'algorithme 18 n'est pas très judicieux. En effet, la mise à jour de l'approximation de l'inverse de la hessienne (équation (A.13)), implique la multiplication de matrices, dont la complexité algorithmique est en $O(n^3)$, où n est la dimension des vecteurs x_k .

Une implantation plus efficace consiste à ne pas calculer explicitement cette mise à jour. On calcule directement la direction de descente (ligne 6 de l'algorithme 18), qui fait intervenir un produit matrice-vecteur, ce qui peut se faire avec une complexité de calcul est alors $O(n^2)$, à partir de toutes les valeurs successives des p_k et q_k , ce qui implique de les stocker en mémoire.

A.3 *Limited-memory* BFGS

Le calcul efficace de la direction de descente s'appuie en fait sur le développement de H_k donné ici pour une profondeur m quelconque ($m \leq k$) par la formule (A.16) suivante, et sur la multiplication de cette expression par le vecteur $\nabla f(x)$.

$$\begin{aligned}
H_k &= V_{k-1}^T \dots V_{k-m}^T H_{k-m} V_{k-m} \dots V_{k-1} \\
&+ \rho_{k-m} V_{k-1}^T \dots V_{k-m+1}^T P_{k-m} P_{k-m}^T V_{k-m+1} \dots V_{k-1} \\
&+ \rho_{k-m+1} V_{k-1}^T \dots V_{k-m+2}^T P_{k-m+1} P_{k-m+1}^T V_{k-m+2} \dots V_{k-1} \\
&+ \dots \\
&+ \rho_{k-1} P_{k-1} P_{k-1}^T
\end{aligned} \tag{A.16}$$

En prenant $m = k$, on voit que ce développement permet d'exprimer directement H_k à partir de H_0 et de toutes les valeurs des vecteurs p_i et q_i , ce qui nécessite de les avoir stockées en mémoire. Le calcul efficace du produit $H_k \nabla f(x_k)$ se fait alors selon le principe de l'algorithme 19 présenté un peu plus loin (en prenant $k = m$ et $H_0^{(k)} = H_0$). Son inconvénient majeur est qu'il est coûteux en mémoire.

L'idée de la méthode à mémoire limitée, dénommée L-BFGS et proposée par Nocedal [126, 109], consiste à arrêter ce développement à une profondeur m fixée, et à remplacer H_{k-m} par une nouvelle matrice "initiale" $H_0^{(k)}$ proportionnelle à la matrice identité. Cette approche se justifie par le fait que les itérations les plus récentes sont plus porteuses d'information sur la hessienne à la position courante que les plus anciennes. L'avantage de la méthode est qu'elle ne nécessite que le stockage d'au plus m valeurs des couples (p_i, q_i) . La matrice $H_0^{(k)}$ est choisie de la même manière que dans l'équation (A.14), en tenant compte de la courbure locale en x_{k-m} (si $k > m$). L'équation (A.17) donne l'expression de la matrice $H_0^{(k)}$:

$$H_0^k = \begin{cases} \gamma_{k-m} I & \text{si } k > m \\ H_0 = \gamma_0 I & \text{si } k \leq m \end{cases} \quad \text{avec } \gamma_i = \frac{q_i^T p_i}{q_i^T q_i} \tag{A.17}$$

Le calcul de la direction de descente se fait alors de façon efficace (en $O(mn)$) et relativement peu coûteuse en mémoire, en appliquant l'algorithme 19. La méthode L-BFGS complète est quant à elle décrite dans l'algorithme 20. C'est cette méthode que nous avons implantée en Ocaml, et utilisée dans le cadre des travaux présentés dans ce document, notamment pour ajuster les poids des réseaux de neurones (voir section 3.8) du chapitre 3).

Algorithm 19 Calcul efficace de la direction de descente $d_k = -H_k \nabla f(x_k)$

```

1: function COMPUTEDIRECTION( $m, H_0^{(k)}$ )
2:    $q \leftarrow \nabla f(x_k)$ 
3:   for  $i = k - 1$  to  $\max(k - m, 0)$  do
4:      $a_i \leftarrow \rho_i p_i^T q$ 
5:      $q \leftarrow q - a_i q_i$ 
6:   end for
7:    $r \leftarrow H_0^{(k)} q$ 
8:   for  $i = \max(k - m, 0)$  to  $k - 1$  do
9:      $b \leftarrow \rho_i q_i^T r$ 
10:     $r \leftarrow r + p_i(a_i - b)$ 
11:  end for
12:  Retourner le résultat  $-H_k \nabla f(x_k) = -r$ 
13: end function

```

Algorithm 20 Algorithme de la méthode L-BFGS.

```

1: Choisir  $m$ 
2:  $x_0 \leftarrow \text{INIT}(\ )$ 
3:  $H_0 \leftarrow \text{INITINVHESSIAN}(\ )$ 
4: while not(STOP( $k, x_k$ )) do
5:    $H_0^{(k)} \leftarrow \text{INITINVHESSIANK}(k, m)$  ▷ Equation (A.17)
6:    $d_k = \text{COMPUTEDIRECTION}(m, H_0^{(k)})$  ▷ Algorithme 19
7:    $\alpha_k \leftarrow \text{LINESEARCH}(f, d_k)$  ▷ Choix d'un pas satisfaisant les conditions de Wolfe
8:    $x_{k+1} \leftarrow x_k + \alpha_k d_k$ 
9:    $p_k = x_{k+1} - x_k$ 
10:   $q_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ 
11:  if  $k > m$  then
12:    Supprimer  $(p_{k-m}, q_{k-m})$  de la mémoire
13:  end if
14:  Stocker  $(p_k, q_k)$  dans la mémoire
15:   $k \leftarrow k + 1$ 
16: end while
17: Retourner  $(x_k, f(x_k))$ 

```

Annexe B

Théorie de l'information et sélection de modèles

Cette annexe présente le principe général de l'utilisation de critères théoriques d'information pour la sélection de modèles, et détaille deux des critères les plus employés : l'AIC d'Akaike [6], et le BIC de Schwartz [146]. Les démonstrations données pour ces deux critères ne prétendent pas à la plus totale rigueur mathématique, l'objectif étant d'en transmettre l'idée de façon concise.

Rappelons que, dans le cadre de l'apprentissage supervisé, on cherche à approximer un phénomène f inconnu ayant produit des données $S = \{(x_n, y_n)_{1 \leq n \leq N}\}$ par une hypothèse h connue, prise dans un espace d'hypothèses \mathcal{H} . Dans ce cadre, S est un ensemble d'exemples où les x sont des entrées et les y les réponses observées à ces entrées. Dans la suite, on notera parfois S^N l'ensemble S , pour rappeler le fait qu'il s'agit d'un ensemble de taille N .

Dans la sélection de modèles, on considère une collection de sous-espaces $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_C$ de complexités généralement croissantes. L'objectif est alors de déterminer $c^* \in \{1, \dots, C\}$ et de trouver l'hypothèse $h_{c^*} \in \mathcal{H}_{c^*}$ qui approxime au mieux f .

Dans ce qui suit, le phénomène f ayant produit les données observées est modélisé comme une densité de probabilité, que l'on notera p_f . La variable aléatoire dont les observations sont des réalisations est notée Z . Plusieurs choix de représentation sont possibles, pour la variable Z :

- a) $Z = (X, Y)$ si p_f est la densité jointe $p_{X,Y|f}$. Dans ce cas, chaque observation $z = (x, y)$ d'une entrée et de la réponse associée est donc un tirage aléatoire d'une variable $Z = (X, Y)$, selon la loi de densité jointe $p_f = p_{X,Y|f}$. La relation entre la probabilité jointe et la densité, sachant que les données sont produites par f est alors la suivante :

$$\mathbb{P}(X \leq x, Y \leq y|f) = \int_{-\infty}^x \int_{-\infty}^y p_f(u, v) du dv$$

- b) $Z = Y$, lorsque les entrées x sont fixées, si p_f est la densité conditionnelle $p_{Y|X,f}$. Dans ce cas, pour une valeur donnée de x , la réponse observée $z = y$ est un tirage aléatoire de $Z = Y$, selon la loi conditionnelle $p_f = p_{Y|X,f}$. La relation entre probabilité et densité conditionnelles est alors :

$$\mathbb{P}(Y \leq y|X = x, f) = \int_{-\infty}^y p_{Y|X=x,f}(t) dt$$

Le choix de représentation de f par une densité jointe ou conditionnelle n'a pas d'influence sur l'expression des critères d'information présentés dans la suite, et nous parlerons donc de la variable Z et de la densité p_f sans préciser ce choix de représentation, sauf si nécessaire.

On recherche une hypothèse $h \in \mathcal{H}$ qui approxime au mieux le phénomène f ayant produit les données observées. Cette hypothèse h est donc ici également représentée par une densité de probabilité p_h , qui pourra être soit jointe soit conditionnelle selon le choix de représentation fait pour p_f .

B.1 La divergence de Kullback-Leibler

B.1.1 Définition

La divergence de Kullback-Leibler [105] est l'écart entre les deux densités p_h et p_f , sous la loi p_f :

$$\begin{aligned} D_{KL}(p_f||p_h) &= \left(- \int_{\mathcal{Z}} p_f(z) \ln p_h(z) dz \right) - \left(- \int_{\mathcal{Z}} p_f(z) \ln p_f(z) dz \right) \\ &= - \int_{\mathcal{Z}} p_f(z) \ln \frac{p_h(z)}{p_f(z)} dz \end{aligned} \quad (\text{B.1})$$

Dans l'équation (B.1), le terme $-\int p_f(z) \ln p_f(z) dz$ est l'entropie différentielle de p_f . En théorie de l'information, c'est une borne inférieure de la quantité d'information nécessaire pour transmettre l'état de la variable aléatoire Z de densité p_f . Le lecteur peut se reporter à [24], pp. 48-58, pour une courte présentation de la théorie de l'information et de la notion d'entropie.

La divergence de Kullback-Leibler représente donc la quantité d'information supplémentaire à encoder pour transmettre Z , lorsqu'on approxime p_f par p_h . De façon plus concise, elle peut s'exprimer par l'équation (B.2) où \mathbb{E}_{p_f} est l'espérance sous la loi p_f .

$$\begin{aligned} D_{KL}(p_f||p_h) &= \mathbb{E}_{p_f} (\ln p_f(Z)) - \mathbb{E}_{p_f} (\ln p_h(Z)) \\ &= \mathbb{E}_{p_f} \left(- \ln \frac{p_h(Z)}{p_f(Z)} \right) \end{aligned} \quad (\text{B.2})$$

Lorsqu'on compare différentes densités p_h pour une loi p_f fixée, mais inconnue, le terme $\mathbb{E}_{p_f}(\ln p_f(Z))$ peut être vu comme une constante. On s'intéresse donc essentiellement au terme $\mathbb{E}_{p_f}(\ln p_h(Z))$. Celui-ci n'est pas calculable, en règle générale, puisque l'espérance \mathbb{E}_{p_f} est sous la vraie distribution p_f , inconnue. On peut cependant chercher à estimer empiriquement $\mathbb{E}_{p_f}(\ln p_h(Z))$ à partir d'un ensemble d'observations $S^N = \{z_n | 1 \leq n \leq N\}$ de taille N finie, ce qui introduit un biais que les critères théoriques d'information cherchent à corriger.

B.1.2 Relations avec la log-vraisemblance

Dans les équations (B.1) et (B.2) définissant la divergence de Kullback-Leibler, la quantité $p_h(z)$ représente la (densité de) probabilité d'observer la donnée z sachant qu'on a fait l'hypothèse h . En d'autres termes, c'est la vraisemblance de l'hypothèse h au vu de la donnée z , par définition de la vraisemblance. Le terme $\ln p_h(z)$ est donc la log-vraisemblance de h au vu de la donnée z .

Une estimation empirique de $\mathbb{E}_{p_f}(\ln p_h(Z))$ à partir de $S^N = \{z_1, \dots, z_N\}$ peut s'obtenir en considérant $\ln \mathcal{L}(h; S^N)$, la log-vraisemblance de h au vu de S^N , et on a la convergence asymptotique suivante (en probabilité ou presque sûrement selon les hypothèses), en supposant $\mathbb{E}_{p_f}(\ln p_h(Z))$ bornée :

$$\frac{1}{N} \ln \mathcal{L}(h; S^N) = \frac{1}{N} \sum_{n=1}^N \ln p_h(z_n) \xrightarrow{N \rightarrow \infty} \mathbb{E}_{p_f}(\ln p_h(Z)) \quad (\text{B.3})$$

En appliquant le même raisonnement à $\ln p_f(z)$, et en considérant la définition (B.2) de $D_{KL}(p_f||p_h)$, on a l'équation (B.4) faisant intervenir le rapport des vraisemblances (en supposant $D_{KL}(p_f||p_h)$ bornée) :

$$-\frac{1}{N} \ln \left(\frac{\mathcal{L}(h; S^N)}{\mathcal{L}(f; S^N)} \right) = -\frac{1}{N} \ln \left(\frac{\prod_{n=1}^N p_h(z_n)}{\prod_{n=1}^N p_f(z_n)} \right) \xrightarrow{N \rightarrow \infty} D_{KL}(p_f||p_h) \quad (\text{B.4})$$

En considérant les instances z_1, \dots, z_N comme des réalisations d'un N -uplet de variables aléatoires (Z_1, \dots, Z_N) , indépendantes et de même loi p_f , on voit que l'espérance de la moyenne empirique de la log-vraisemblance, sur tous les tirages possibles de Z_1, \dots, Z_N , est égale à l'espérance $\mathbb{E}_{p_f} \ln p_h(Z)$:

$$\mathbb{E}_{Z_1, \dots, Z_N} \left[\frac{1}{N} \ln \mathcal{L}(h; Z_1, \dots, Z_N) \right] = \mathbb{E}_{p_f} \ln p_h(Z) \quad (\text{B.5})$$

Pour une hypothèse h fixée, dont le choix ne dépend pas des données, on peut donc écrire, en nommant \mathcal{D} la variable aléatoire (Z_1, \dots, Z_N) dont S^N est une réalisation, pour plus de concision :

$$\mathbb{E}_{\mathcal{D}}(\ln \mathcal{L}(h; \mathcal{D})) = N \mathbb{E}_{p_f} \ln p_h(Z) \quad (\text{B.6})$$

$$\mathbb{E}_{\mathcal{D}} \left(-\ln \left(\frac{\mathcal{L}(h; \mathcal{D})}{\mathcal{L}(f; \mathcal{D})} \right) \right) = N D_{KL}(p_f||p_h) \quad (\text{B.7})$$

Plaçons nous maintenant dans le cas où $h = h_\theta$ est paramétrée par θ , et supposons qu'on dispose d'une méthode permettant d'obtenir une valeur "optimale" $\hat{\theta}$ de ce vecteur de paramètres à partir des données. Le critère d'optimalité peut par exemple être celui du maximum de vraisemblance. Dans ce contexte, le choix de l'hypothèse $h_{\hat{\theta}}$ dépend des données puisque $\hat{\theta}$ en dépend. Les équations (B.6) et (B.7) ne sont donc plus valables telles quelles. Il faut en effet les moyenner sur l'éventail des valeurs prises par $\hat{\theta}$ selon les tirages possibles des données.

En considérant la valeur prise par θ comme un réalisation d'une variable aléatoire θ , et en considérant l'ensemble de données utilisé pour calculer $\hat{\theta}$ comme une réalisation d'une variable aléatoire \mathcal{D}' , on peut alors écrire :

$$\mathbb{E}_{\mathcal{D}'} \mathbb{E}_{\mathcal{D}} \left(\ln \mathcal{L}(h_\theta; \mathcal{D}) \mid_{\theta = \hat{\theta}(\mathcal{D}')} \right) = N \mathbb{E}_{\mathcal{D}'} \mathbb{E}_{p_f} \ln p_{h_{\hat{\theta}(\mathcal{D}')}}(Z) \quad (\text{B.8})$$

$$\mathbb{E}_{\mathcal{D}'} \mathbb{E}_{\mathcal{D}} \left(-\ln \left(\frac{\mathcal{L}(h_{\theta}; \mathcal{D})}{\mathcal{L}(f; \mathcal{D})} \right) \Big|_{\theta = \hat{\theta}(\mathcal{D}')} \right) = N \mathbb{E}_{\mathcal{D}'} (D_{KL}(p_f || p_{h_{\hat{\theta}(\mathcal{D}')}})) \quad (\text{B.9})$$

Dans les équations B.8 et B.9, il est commode de distinguer par des notations différentes la variable aléatoire \mathcal{D}' intervenant dans $\hat{\theta}(\mathcal{D}')$ et la variable \mathcal{D} représentant l'ensemble de données utilisé dans le calcul de la log-vraisemblance, quand bien même les réalisations des deux variables correspondent à des tirages aléatoires de (Z_1, \dots, Z_N) selon une même loi.

L'équation (B.9) peut se réécrire comme suit, sachant que f ne dépend pas de $\hat{\theta}$:

$$\begin{aligned} N \mathbb{E}_{\mathcal{D}'} (D_{KL}(p_f || p_{h_{\hat{\theta}(\mathcal{D}')}})) &= \mathbb{E}_{\mathcal{D}'} \mathbb{E}_{\mathcal{D}} \left(-\ln \mathcal{L}(h_{\hat{\theta}(\mathcal{D}')}; \mathcal{D}) + \ln \mathcal{L}(f; \mathcal{D}) \right) \\ &= \mathbb{E}_{\mathcal{D}'} \mathbb{E}_{\mathcal{D}} \left(-\ln \mathcal{L}(h_{\hat{\theta}(\mathcal{D}')}; \mathcal{D}) \right) - \mathbb{E}_{\mathcal{D}} \left(-\ln \mathcal{L}(f; \mathcal{D}) \right) \end{aligned} \quad (\text{B.10})$$

Tout l'enjeu de la définition d'un critère asymptotique d'information comme l'AIC d'Akaike consiste à corriger le biais introduit en approximant $\mathbb{E}_{\mathcal{D}'} \mathbb{E}_{\mathcal{D}} \left(\ln \mathcal{L}(h_{\hat{\theta}(\mathcal{D}')}; \mathcal{D}) \right)$ par la log-vraisemblance $\ln \mathcal{L}(h_{\hat{\theta}(S^N)}; S^N)$ maximisée sur le seul ensemble S^N de taille finie.

B.1.3 Lien avec l'information de Fisher et le ratio de vraisemblance

Reconsidérons l'équation (B.2), en nous plaçant dans le cas d'hypothèses $h = h_{\theta}$ dépendant d'un paramètre θ . La divergence de Kullback-Leibler pour une valeur quelconque de θ , fixée de façon indépendante des données, est donnée par l'équation suivante :

$$D_{KL}(p_f || p_h) = \mathbb{E}_{p_f} \left(-\ln \frac{p_{h_{\theta}}(Z)}{p_f(Z)} \right) \quad (\text{B.11})$$

Notons θ^* la valeur optimale de θ , au sens où θ^* minimise la divergence de Kullback-Leibler sur l'ensemble Θ des valeurs possibles de θ :

$$\theta^* = \arg \min_{\theta \in \Theta} D_{KL}(p_f || p_{h_{\theta}}) = \arg \min_{\theta \in \Theta} \mathbb{E}_{p_f} (-\ln p_{h_{\theta}}(Z)) \quad (\text{B.12})$$

Par conséquent, θ^* maximise $\mathbb{E}_{p_f} (\ln p_{h_{\theta}}(Z))$, l'espérance de la log-vraisemblance de l'hypothèse h_{θ} au vu des données Z distribuées selon la loi p_f inconnue. Le gradient de la fonction $\theta \rightarrow \mathbb{E}_{p_f} (\ln p_{h_{\theta}}(Z))$ s'annule donc au point θ^* . En tenant compte du fait que les dérivées à l'ordre 1 s'annulent, le développement de Taylor limité à l'ordre 2 de cette fonction, pour des valeurs de θ proches de θ^* , s'exprime alors comme suit :

$$\mathbb{E}_{p_f} (\ln p_{h_{\theta}}(Z)) = \mathbb{E}_{p_f} (\ln p_{h_{\theta^*}}(Z)) - \frac{1}{2} (\theta - \theta^*)^T \mathcal{I}(\theta^*) (\theta - \theta^*) + o \left(\|\theta - \theta^*\|_{\mathcal{I}(\theta^*)}^2 \right) \quad (\text{B.13})$$

Dans l'équation (B.13), la matrice $\mathcal{I}(\theta^*)$ est la matrice d'information de Fisher donnée par l'équation (B.14) où θ_i et θ_j dénotent les $i^{\text{ième}}$ et $j^{\text{ième}}$ composantes du vecteur θ . Cette matrice est de dimension $k \times k$, si k est la dimension de θ .

$$\mathcal{I}(\theta^*) = \mathbb{E}_{p_f} \left[-\frac{\partial^2 \ln p_{h_{\theta}}(Z)}{\partial \theta \partial \theta^T} \Big|_{\theta = \theta^*} \right] = \left(\mathbb{E}_{p_f} \left[-\frac{\partial^2 \ln p_{h_{\theta}}(Z)}{\partial \theta_i \partial \theta_j} \Big|_{\theta = \theta^*} \right] \right)_{(i,j) \in \{1, \dots, k\}^2} \quad (\text{B.14})$$

Avec $\mathcal{I}(\theta^*)$ symétrique, définie positive, la forme quadratique $(\theta - \theta^*)^T \mathcal{I}(\theta^*) (\theta - \theta^*)$ définit une norme, et on note $\|\theta - \theta^*\|_{\mathcal{I}(\theta^*)}^2$ cette quantité.

Convergence asymptotique des estimateurs de maximum de vraisemblance

En considérant maintenant l'équation (B.7), avec des données \mathcal{D} comprenant N instances de Z tirées dans la même distribution p_f , on peut décomposer cette expression en introduisant $\theta^* = \arg \min_{\theta \in \Theta} D_{KL}(p_f \| p_{h_\theta})$:

$$\begin{aligned} N D_{KL}(p_f \| p_{h_\theta}) &= \mathbb{E}_{\mathcal{D}} \left(-\ln \left(\frac{\mathcal{L}(h_\theta; \mathcal{D})}{\mathcal{L}(f; \mathcal{D})} \right) \right) \\ &= \mathbb{E}_{\mathcal{D}} \left(\ln \left(\frac{\mathcal{L}(h_{\theta^*}; \mathcal{D})}{\mathcal{L}(f; \mathcal{D})} \right) \right) - \mathbb{E}_{\mathcal{D}} \left(\ln \left(\frac{\mathcal{L}(h_\theta; \mathcal{D})}{\mathcal{L}(h_{\theta^*}; \mathcal{D})} \right) \right) \end{aligned} \quad (\text{B.15})$$

Cette équation montre la relation de la divergence de Kullback-Leibler avec le rapport des vraisemblances, en particulier avec l'espérance $\mathbb{E}_{\mathcal{D}} \left(-\ln \left(\frac{\mathcal{L}(h_\theta; \mathcal{D})}{\mathcal{L}(h_{\theta^*}; \mathcal{D})} \right) \right)$. Malheureusement, avec un ensemble fini de données, on ne dispose que d'une estimation empirique de $-\ln \left(\frac{\mathcal{L}(h_\theta; S^N)}{\mathcal{L}(h_{\theta^*}; S^N)} \right)$ sur cet ensemble de données S^N .

En multipliant par deux et en injectant dans $\mathbb{E}_{\mathcal{D}} \left(-\ln \left(\frac{\mathcal{L}(h_\theta; \mathcal{D})}{\mathcal{L}(h_{\theta^*}; \mathcal{D})} \right) \right)$ l'équation (B.6) qui nous dit que $\mathbb{E}_{\mathcal{D}} (\ln \mathcal{L}(h_\theta; \mathcal{D})) = N \mathbb{E}_{p_f} \ln p_{h_\theta}(Z)$ pour une hypothèse h_θ fixée, et en combinant avec (B.13), on obtient l'expression suivante :

$$\begin{aligned} \mathbb{E}_{\mathcal{D}} \left(-2 \ln \left(\frac{\mathcal{L}(h_\theta; \mathcal{D})}{\mathcal{L}(h_{\theta^*}; \mathcal{D})} \right) \right) &= \mathbb{E}_{\mathcal{D}} \left(2 \ln \mathcal{L}(h_{\theta^*}; \mathcal{D}) \right) - \mathbb{E}_{\mathcal{D}} \left(2 \ln \mathcal{L}(h_\theta; \mathcal{D}) \right) \\ &= 2N \mathbb{E}_{p_f} \left(\ln p_{h_{\theta^*}}(Z) \right) - 2N \mathbb{E}_{p_f} \left(\ln p_{h_\theta}(Z) \right) \\ &= N(\theta - \theta^*)^T \mathcal{I}(\theta^*) (\theta - \theta^*) + o \left(N \|\theta - \theta^*\|_{\mathcal{I}(\theta^*)}^2 \right) \\ &= \sqrt{N}(\theta - \theta^*)^T \mathcal{I}(\theta^*) \sqrt{N}(\theta - \theta^*) + o \left(\|\sqrt{N}(\theta - \theta^*)\|_{\mathcal{I}(\theta^*)}^2 \right) \\ &= \|\sqrt{N}(\theta - \theta^*)\|_{\mathcal{I}(\theta^*)}^2 + o \left(\|\sqrt{N}(\theta - \theta^*)\|_{\mathcal{I}(\theta^*)}^2 \right) \end{aligned} \quad (\text{B.16})$$

Si l'on reprend maintenant le logarithme du rapport des vraisemblances, sans prendre cette fois-ci son espérance mais en considérant une instance particulière $S^N = \{z_1, \dots, z_N\}$ de \mathcal{D} , on obtient l'expression suivante, en multipliant par deux et en faisant un développement limité en θ^* , en tenant compte du fait que les dérivées à l'ordre 1 s'annulent en θ^* :

$$\begin{aligned} -2 \ln \left(\frac{\mathcal{L}(h_\theta; S^N)}{\mathcal{L}(h_{\theta^*}; S^N)} \right) &= 2 \ln \mathcal{L}(h_{\theta^*}; S^N) - 2 \ln \mathcal{L}(h_\theta; S^N) \\ &= (\theta - \theta^*)^T J_N(\theta^*) (\theta - \theta^*) + o \left(\|\theta - \theta^*\|_{J_N(\theta^*)}^2 \right) \\ &= \sqrt{N}(\theta - \theta^*)^T \left(\frac{1}{N} J_N(\theta^*) \right) \sqrt{N}(\theta - \theta^*) + o \left(\|\theta - \theta^*\|_{J_N(\theta^*)}^2 \right) \end{aligned} \quad (\text{B.17})$$

La forme quadratique $(\theta - \theta^*)^T J_N(\theta^*)(\theta - \theta^*)$ est notée $\|\theta - \theta^*\|_{J_N(\theta^*)}^2$, et $J_N(\theta^*)$ est la matrice $J_N(\theta)$ de l'opposé des dérivées secondes de la log-vraisemblance sur l'ensemble S^N , au point θ^* :

$$J_N(\theta) = -\frac{\partial^2 \sum_{n=1}^N \ln p_{h_\theta}(z_n)}{\partial \theta \partial \theta^T} = \left(-\frac{\partial^2 \sum_{n=1}^N \ln p_{h_\theta}(z_n)}{\partial \theta_i \partial \theta_j} \right)_{(i,j) \in \{1, \dots, k\}^2} \quad (\text{B.18})$$

Sous certaines conditions sur la fonction $\theta \rightarrow \mathcal{L}(h_\theta; S^N)$ et l'espace Θ , il existe une suite de points $(\hat{\theta}_N)_{N \in \mathbb{N}}$ de Θ où un maximum local de $\mathcal{L}(h_\theta; S^N)$ est atteint, et qui converge en probabilité vers θ^* . On considérera par la suite que les conditions sont réunies pour que l'on ait :

$$\hat{\theta}_N \xrightarrow[N \rightarrow \infty]{\mathcal{P}} \theta^* \quad \text{et} \quad \frac{1}{N} J_N(\theta) \xrightarrow[N \rightarrow \infty]{\mathcal{P}} \mathcal{I}(\theta) \quad \text{au voisinage de } \theta^* \quad (\text{B.19})$$

Pour alléger les notations, on omettra par la suite l'indice N dans $\hat{\theta}_N$, et on notera l'estimateur de maximum de vraisemblance $\hat{\theta}$, ou $\hat{\theta}^{\text{ML}}$ (pour *maximum likelihood*) s'il y a lieu de le distinguer d'autres estimateurs comme le *maximum a posteriori*. On pourra encore le noter $\hat{\theta}^{\text{ML}}(S^N)$ ou $\hat{\theta}(S^N)$ pour souligner la dépendance de l'estimateur au jeu de données S^N .

Cas d'estimateurs $\hat{\theta}_c$ dans des espaces de dimension restreinte $k_c \leq k$

Pour l'élaboration de critères d'information, dans le cadre de la sélection de modèle, on considérera des sous-espaces Θ_c de Θ , de dimension k_c inférieure à la dimension k de θ^* , et on s'intéressera, dans le cas du critère d'Akaike, aux estimateurs par maximum de vraisemblance θ_c dans chacun de ces sous-espaces, l'objectif étant d'obtenir une valeur asymptotique du biais introduit en approximant $\mathbb{E}_{\mathcal{D}'} \mathbb{E}_{\mathcal{D}} \left(\ln \mathcal{L}(h_{\hat{\theta}_c(\mathcal{D}')}); \mathcal{D} \right)$ par la log-vraisemblance empirique $\ln \mathcal{L}(h_{\hat{\theta}_c(S^N)}; S^N)$ maximisée sur le seul ensemble S^N de taille finie.

Lorsque $k_c = k$, on supposera réunies les conditions pour la convergence asymptotique de $\hat{\theta}_N$ (que l'on notera simplement $\hat{\theta}$ dans la suite) vers θ^* et de $\frac{1}{N} J_N(\theta)$ quand $N \rightarrow \infty$, que nous venons de présenter précédemment dans cette section.

On notera θ_c^* la valeur de $\theta_c \in \Theta_c$ qui minimise la divergence de Kullback-Leibler, dans le sous-espace Θ_c de dimension restreinte k_c . En se plaçant dans l'espace vectoriel muni du produit scalaire $\langle \cdot, \cdot \rangle_{\mathcal{I}(\theta^*)}$ et de la métrique $\|\cdot\|_{\mathcal{I}(\theta^*)}$ définis par la matrice d'information de Fisher, on considérera que θ_c^* est approximativement la projection de θ^* sur le sous-espace de dimension k_c inférieure à k .

B.2 Les critères asymptotiques d'information

B.2.1 Principe général

Étant donné une collection de modèles $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_C$ de complexités différentes, on cherche à déterminer $c \in \{1, \dots, C\}$ et à trouver $h_c^* \in \mathcal{H}_{c^*}$ qui approxime au mieux f .

On suppose ici que chaque hypothèse h de chaque modèle candidat \mathcal{H}_c est paramétrée par un vecteur $\theta_c \in \Theta_c$ de dimension k_c . Pour un certain nombre de méthodes paramétriques d'apprentissage, la complexité d'une hypothèse $h = h_{\theta_c}$ est directement liée au nombre de

paramètres k_c à ajuster. On suppose qu'on dispose d'une méthode permettant d'ajuster ces paramètres (par exemple par maximum de vraisemblance) à partir d'un ensemble d'observations S^N , et d'obtenir le vecteur $\hat{\theta}_c$ qui estime au mieux θ_c^* , la valeur optimale de θ_c pour le modèle \mathcal{H}_c .

La forme générale d'un critère d'information permettant de comparer différents modèles \mathcal{H}_c est alors donnée par l'équation (B.20), où le facteur 2 apparaît pour des raisons "historiques" (définition de la déviance statistique), et où $\hat{\theta}_c$ est une estimation (par exemple par maximum de vraisemblance) du paramètre θ_c^* de la meilleure hypothèse possible dans \mathcal{H}_c :

$$IC = -2 \ln \mathcal{L}(h_{\hat{\theta}_c}; S^N) + P_c \quad (\text{B.20})$$

Pour certains de critères d'information, comme le critère d'Akaike (AIC), la pénalité P_c est obtenue en considérant la convergence asymptotique du biais introduit lorsqu'on estime empiriquement sur S^N la divergence de Kullback-Leibler exprimée en fonction d'un rapport de vraisemblances, en faisant l'hypothèse que $f = h_{\theta^*} \in \mathcal{H}$ appartient à l'espace des hypothèses. D'autres, comme le BIC, utilisent une approche bayésienne et maximisent la probabilité *a posteriori* du modèle \mathcal{H}_c , après observation des données S^N .

La sélection de modèle avec un critère d'information consiste alors simplement à choisir le modèle \mathcal{H}_{c^*} qui minimise le critère IC choisi.

B.2.2 Le critère AIC d'Akaike

Le critère d'information proposé par Akaike en 1973 [6] cherche à minimiser la divergence de Kullback-Leibler en faisant deux hypothèses fortes :

- i le vrai phénomène f ayant produit les données observées fait partie de l'ensemble \mathcal{H} des hypothèses testées : $\exists \theta^* \in \Theta, \quad f = h_{\theta^*} \in \mathcal{H}$,
- ii pour chaque modèle candidat \mathcal{H}_c , l'estimation de la "meilleure" valeur possible $\hat{\theta}_c$ du paramètre θ_c est obtenue par maximum de vraisemblance.

Pour un sous-espace \mathcal{H}_c donné, l'ajustement de θ_c par maximum de vraisemblance consiste à choisir la valeur $\hat{\theta}_c = \hat{\theta}_c^{\text{ML}}(S^N)$ qui maximise la vraisemblance $\mathcal{L}(h_{\theta_c}; S^N) = \prod_{n=1}^N p_{h_{\theta_c}}(z_n)$, soit encore la log-vraisemblance comme indiqué dans l'équation (B.21).

$$\hat{\theta}_c = \arg \max_{\theta_c \in \Theta_c} [\ln \mathcal{L}(h_{\theta_c}; S^N)] = \arg \max_{\theta_c \in \Theta_c} \left[\sum_{n=1}^N \ln p_{h_{\theta_c}}(z_n) \right] \quad (\text{B.21})$$

Pour $\theta_c = \hat{\theta}_c$, la log-vraisemblance maximisée sur l'ensemble S^N est donc :

$$\ln \mathcal{L}(h_{\hat{\theta}_c}; S^N) = \sum_{n=1}^N \ln p_{h_{\hat{\theta}_c}}(z_n) \quad (\text{B.22})$$

Avec k_c le nombre de paramètres à ajuster (la dimension de θ_c) pour les hypothèses du modèle \mathcal{H}_c , et N le nombre d'exemples, le critère d'information d'Akaike (AIC), s'exprime comme suit :

$$\boxed{AIC = -2 \ln \mathcal{L}(h_{\hat{\theta}_c}; S^N) + 2k_c = -2 \sum_{n=1}^N \ln p_{h_{\hat{\theta}_c}}(z_n) + 2k_c} \quad (\text{B.23})$$

Esquisse de démonstration

Soit θ^* la valeur de θ qui minimise la divergence de Kullback-Leibler $D_{KL}(p_f||p_{h_\theta})$. On peut, sans perte de généralité, considérer la collection de modèles $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_C$ telle que θ^* soit de dimension $k = k_C$, la dimension du modèle \mathcal{H}_C le plus complexe de la collection, en prenant k_C suffisamment grand pour cela¹.

Avec les hypothèses i) et ii) prises par Akaike, et en se plaçant en dimension $k_c \leq k$ et en considérant l'hypothèse $h_{\hat{\theta}_c}$ obtenue par maximum de vraisemblance, l'équation (B.10) peut se réécrire comme suit :

$$\begin{aligned}
2N \mathbb{E}_{\mathcal{D}'}(D_{KL}(p_f||p_{h_{\hat{\theta}_c(\mathcal{D}')}})) &= 2N \mathbb{E}_{\mathcal{D}'}(D_{KL}(p_{h_{\theta^*}}||p_{h_{\hat{\theta}_c(\mathcal{D}')}})) \\
&= \mathbb{E}_{\mathcal{D}'} \mathbb{E}_{\mathcal{D}} \left(-2 \ln \frac{\mathcal{L}(h_{\hat{\theta}_c(\mathcal{D}')}; \mathcal{D})}{\mathcal{L}(h_{\theta^*}; \mathcal{D})} \right) \\
&= \mathbb{E}_{\mathcal{D}'} \mathbb{E}_{\mathcal{D}} \left(-2 \ln \frac{\mathcal{L}(h_{\hat{\theta}_c(\mathcal{D}')}; \mathcal{D})}{\mathcal{L}(h_{\theta_c^*}; \mathcal{D})} - 2 \ln \frac{\mathcal{L}(h_{\theta_c^*}; \mathcal{D})}{\mathcal{L}(h_{\theta^*}; \mathcal{D})} \right) \quad (\text{B.24}) \\
&= \mathbb{E}_{\mathcal{D}'} \mathbb{E}_{\mathcal{D}} \left(-2 \ln \frac{\mathcal{L}(h_{\hat{\theta}_c(\mathcal{D}')}; \mathcal{D})}{\mathcal{L}(h_{\theta_c^*}; \mathcal{D})} \right) + \mathbb{E}_{\mathcal{D}} \left(-2 \ln \frac{\mathcal{L}(h_{\theta_c^*}; \mathcal{D})}{\mathcal{L}(h_{\theta^*}; \mathcal{D})} \right)
\end{aligned}$$

En combinant avec (B.16), on obtient :

$$2N \mathbb{E}_{\mathcal{D}'}(D_{KL}(p_{h_{\theta^*}}||p_{h_{\hat{\theta}_c(\mathcal{D}')}})) \approx \underbrace{\mathbb{E}_{\mathcal{D}'} \|\sqrt{N}(\hat{\theta}_c - \theta_c^*)\|_{\mathcal{I}(\theta_c^*)}^2}_{\text{Variance}} + \underbrace{\|\sqrt{N}(\theta_c^* - \theta^*)\|_{\mathcal{I}(\theta^*)}^2}_{\text{Biais } \delta} \quad (\text{B.25})$$

Dans cette équation, le terme le plus à droite est le biais entre l'optimum θ_c^* obtenu quand on se restreint à la dimension k_c et le vrai optimum θ^* en dimension k . Le terme à gauche peut être vu comme représentatif de la variance de l'erreur aléatoire $(\hat{\theta}_c - \theta_c^*)$. La forme quadratique $\|\sqrt{N}(\hat{\theta}_c - \theta_c^*)\|_{\mathcal{I}(\theta_c^*)}^2$ est une somme de k_c carrés, qu'on peut voir comme réalisation d'une somme de carrés de variables aléatoires indépendantes. En supposant réunies les conditions pour qu'elle reste bornée et que $\sqrt{N}(\hat{\theta}_c - \theta_c^*)$ soit asymptotiquement distribué comme une loi normale², alors $\|\sqrt{N}(\hat{\theta}_c - \theta_c^*)\|_{\mathcal{I}(\theta_c^*)}^2$ est distribué comme une loi du $\chi_{k_c}^2$ avec k_c degrés de liberté, et son espérance vaut k_c . On peut donc écrire l'équation suivante, valable asymptotiquement pour N grand, et grand devant k_c , et où le biais δ reste encore à déterminer :

$$2N \mathbb{E}_{\mathcal{D}'}(D_{KL}(p_{h_{\theta^*}}||p_{h_{\hat{\theta}_c(\mathcal{D}')}})) \approx \delta + k_c \quad (\text{B.26a})$$

$$\text{où } \delta = \mathbb{E}_{\mathcal{D}} \left(-2 \ln \frac{\mathcal{L}(h_{\theta_c^*}; \mathcal{D})}{\mathcal{L}(h_{\theta^*}; \mathcal{D})} \right) \approx \|\sqrt{N}(\theta_c^* - \theta^*)\|_{\mathcal{I}(\theta^*)}^2 \quad (\text{B.26b})$$

1. Dans le cadre des critères asymptotiques, on suppose que la complexité du modèle \mathcal{H}_C devant contenir l'hypothèse optimale h_{θ^*} ne croît pas indéfiniment avec la taille des données. Des approches plus récentes ([116]) s'intéressent à des critères non asymptotiques où la taille du modèle peut croître avec les données.

2. Cf. propriétés de convergence du maximum de vraisemblance.

La seule quantité dont on dispose pour approximer $\mathbb{E}_{\mathcal{D}} \left(-2 \ln \frac{\mathcal{L}(h_{\theta_c^*}; \mathcal{D})}{\mathcal{L}(h_{\theta^*}; \mathcal{D})} \right)$ est la quantité $-2 \ln \left(\frac{\mathcal{L}(h_{\hat{\theta}_c}; S^N)}{\mathcal{L}(h_{\hat{\theta}}; S^N)} \right)$ évaluée empiriquement sur l'ensemble S^N en approximant θ_c^* par $\hat{\theta}_c = \hat{\theta}_c(S^N)$ et θ^* par $\hat{\theta} = \hat{\theta}(S^N)$. Ces approximations introduisent un biais supplémentaire, qu'Akaike estime asymptotiquement en effectuant deux développements de Taylor similaires à celui effectué dans la section B.1.3 pour obtenir l'équation (B.17).

Un premier développement de Taylor de $\theta \rightarrow -2 \ln \mathcal{L}(h_{\theta}; S^N)$ en $\hat{\theta}$, nous donne l'équation suivante, où J_N est définie par l'équation (B.18) :

$$-2 \ln \mathcal{L}(h_{\theta}; S^N) \approx -2 \ln \mathcal{L}(h_{\hat{\theta}}; S^N) + (\theta - \hat{\theta})^T J_N(\hat{\theta})(\theta - \hat{\theta}) \quad (\text{B.27})$$

Un deuxième développement en $\hat{\theta}_c$ donne, de manière similaire, et en supposant ici $\hat{\theta}_c$ de dimension k , mais avec les $k - k_c$ dernières coordonnées égales à 0 :

$$-2 \ln \mathcal{L}(h_{\theta}; S^N) \approx -2 \ln \mathcal{L}(h_{\hat{\theta}_c}; S^N) + (\theta - \hat{\theta}_c)^T J_N(\hat{\theta}_c)(\theta - \hat{\theta}_c) \quad (\text{B.28})$$

En faisant la différence des deux équations, en choisissant $\theta = \theta_c^*$ et en supposant que toutes les conditions soient réunies pour qu'on puisse approximer $\frac{1}{N} J_N(\hat{\theta})$ et $\frac{1}{N} J_N(\hat{\theta}_c)$ par $\mathcal{I}(\theta^*)$ on en déduit :

$$\begin{aligned} -2 \ln \left(\frac{\mathcal{L}(h_{\hat{\theta}_c}; S^N)}{\mathcal{L}(h_{\hat{\theta}}; S^N)} \right) &\approx \|\sqrt{N}(\theta_c^* - \hat{\theta})\|_{\mathcal{I}(\theta^*)}^2 - \|\sqrt{N}(\theta_c^* - \hat{\theta}_c)\|_{\mathcal{I}(\theta^*)}^2 \\ &= N \|\theta_c^* - \hat{\theta}\|_{\mathcal{I}(\theta^*)}^2 - N \|\theta_c^* - \hat{\theta}_c\|_{\mathcal{I}(\theta^*)}^2 \end{aligned} \quad (\text{B.29})$$

On réintroduit le biais $\delta = N \|\theta_c^* - \theta^*\|_{\mathcal{I}(\theta^*)}^2$ dans le premier terme à droite de l'égalité, en utilisant les propriétés de la métrique $\|\cdot\|_{\mathcal{I}(\theta^*)}$ et du produit scalaire $\langle \cdot, \cdot \rangle_{\mathcal{I}(\theta^*)}$ définis par la matrice d'information de Fisher :

$$N \|\theta_c^* - \hat{\theta}\|_{\mathcal{I}(\theta^*)}^2 = \underbrace{N \|\theta_c^* - \theta^*\|_{\mathcal{I}(\theta^*)}^2}_{\delta} + N \|\theta^* - \hat{\theta}\|_{\mathcal{I}(\theta^*)}^2 + 2N \langle \theta_c^* - \theta^*, \theta^* - \hat{\theta} \rangle_{\mathcal{I}(\theta^*)} \quad (\text{B.30})$$

En considérant le produit scalaire comme négligeable devant les autres termes (voir [6], où Akaike redéfinit θ_c^* comme la projection orthogonale de θ^* sur le sous-espace de dimension k_c), on obtient l'équation suivante en combinant (B.29) et (B.30) :

$$-2 \ln \left(\frac{\mathcal{L}(h_{\hat{\theta}_c}; S^N)}{\mathcal{L}(h_{\hat{\theta}}; S^N)} \right) \approx \underbrace{N \|\theta_c^* - \theta^*\|_{\mathcal{I}(\theta^*)}^2}_{\delta} + \underbrace{N \|\theta^* - \hat{\theta}\|_{\mathcal{I}(\theta^*)}^2}_{\chi_k^2} - \underbrace{N \|\theta_c^* - \hat{\theta}_c\|_{\mathcal{I}(\theta^*)}^2}_{\chi_{k_c}^2} \quad (\text{B.31})$$

En remarquant que $N \|\theta^* - \hat{\theta}\|_{\mathcal{I}(\theta^*)}^2$ est asymptotiquement distribuée comme une loi χ_k^2 d'espérance k , et $N \|\theta_c^* - \hat{\theta}_c\|_{\mathcal{I}(\theta^*)}^2$ comme une loi $\chi_{k_c}^2$ d'espérance k_c , on en déduit la valeur asymptotique de δ :

$$\delta \approx \mathbb{E}_{\mathcal{D}} \left(-2 \ln \left(\frac{\mathcal{L}(h_{\hat{\theta}_c(\mathcal{D})}; \mathcal{D})}{\mathcal{L}(h_{\hat{\theta}(\mathcal{D})}; \mathcal{D})} \right) \right) + k_c - k \quad (\text{B.32})$$

En injectant cette valeur dans l'équation (B.26a), on obtient finalement :

$$2N \mathbb{E}_{\mathcal{D}'}(D_{KL}(p_{h_{\theta^*}} || p_{h_{\hat{\theta}_c(\mathcal{D}')}})) \approx \mathbb{E}_{\mathcal{D}} \left(-2 \ln \left(\frac{\mathcal{L}(h_{\hat{\theta}_c(\mathcal{D})}; \mathcal{D})}{\mathcal{L}(h_{\hat{\theta}(\mathcal{D})}; \mathcal{D})} \right) \right) + 2k_c - k \quad (\text{B.33})$$

En pratique, pour la sélection de modèle, on dispose d'une réalisation $S^N = \{z_1, \dots, z_N\}$ de la variable \mathcal{D} représentant les données, et on souhaite comparer différents modèles \mathcal{H}_c de paramètres θ_c de dimensions k_c différentes ($k_c \leq k$). Dans ce contexte, le terme k reste constant ainsi que la log-vraisemblance $\mathcal{L}(h_{\hat{\theta}(S^N)}; S^N)$. Pour choisir le modèle minimisant la divergence de Kullback-Leibler sous les hypothèses i) et ii), le principe de sélection selon le critère d'Akaike consiste à prendre celui pour lequel $-2 \ln \mathcal{L}(h_{\hat{\theta}_c(S^N)}; S^N) + 2k_c$ est minimal.

B.2.3 Le critère BIC (*Bayesian Information Criterion*)

Le critère BIC (Bayesian Information Criterion) de Schwarz [146] est obtenu en se plaçant dans un cadre bayésien, où le modèle \mathcal{H}_c lui-même est considéré comme une réalisation d'une variable aléatoire H , et où chaque vecteur de paramètres θ est également une réalisation d'une variable aléatoire θ , de densité conditionnelle $p_{\theta|H=\mathcal{H}_c}$, qu'on notera plus simplement $p_{\theta|\mathcal{H}_c}$. Les modèles sont distribués selon une loi *a priori* \mathbb{P}_H , souvent choisie uniforme.

Le critère BIC (Bayesian Information Criterion) cherche à sélectionner le modèle dont la probabilité *a posteriori*, au vu des données, est la plus grande. Soulignons la différence avec le critère d'Akaike, qui sélectionne quant à lui le modèle dont la *vraisemblance* est la plus grande, celle-ci étant la probabilité d'observer les données en supposant fixé le modèle considéré³.

Avec k_c la dimension du vecteur de paramètres θ_c à ajuster, et $\ln \mathcal{L}(h_{\hat{\theta}_c}; S)$ la log-vraisemblance pour un modèle de paramètre estimé $\hat{\theta}_c$, le critère BIC s'exprime comme suit, S étant l'ensemble de données considéré, de taille N :

$$\boxed{BIC = -2 \ln \mathcal{L}(h_{\hat{\theta}_c}; S) + k_c \ln N = -2 \sum_{n=1}^N \ln p_{h_{\hat{\theta}_c}}(z_n) + k_c \ln N} \quad (\text{B.34})$$

Dans l'équation (B.34), le vecteur $\hat{\theta}_c$ correspond au *maximum a posteriori* $\hat{\theta}_c = \hat{\theta}_c^{MAP}$. Pour N suffisamment grand (puisqu'on parle ici d'un critère asymptotique), on peut l'approximer par le maximum de vraisemblance $\hat{\theta}_c = \hat{\theta}_c^{ML}$.

Démonstration

La construction du critère BIC proposée ici s'inspire partiellement de [24] pp. 213-217, [137] pp. 63, et [107], en essayant de rendre explicites toutes les étapes du calcul.

Étant donné un ensemble d'observations S^N de taille N , considéré comme une réalisation d'une variable aléatoire \mathcal{D} , le critère BIC cherche à sélectionner le modèle maximisant la probabilité *a posteriori* $\mathbb{P}(H = \mathcal{H}_c | \mathcal{D} = S^N)$. En appliquant la formule de Bayes, on obtient :

$$\mathbb{P}(H = \mathcal{H}_c | \mathcal{D} = S^N) = \frac{\mathbb{P}(\mathcal{D} = S^N | H = \mathcal{H}_c) \mathbb{P}_H(\mathcal{H}_c)}{\mathbb{P}(\mathcal{D} = S^N)} \quad (\text{B.35})$$

3. Le critère d'Akaike est généralement présenté comme émanant d'une approche fréquentiste, où le modèle n'est pas une variable aléatoire

Dans la suite, on allégera les notations en omettant d'expliciter les variables aléatoires et en écrivant par exemple $\mathbb{P}(\mathcal{H}_c|S^N)$ pour la probabilité *a posteriori* du modèle \mathcal{H}_c , ayant observé les données S^N .

L'objectif est donc de choisir le modèle le plus vraisemblable au vu des données, ce qui s'exprime comme suit, sachant que le dénominateur $\mathbb{P}(S^N)$ dans l'équation (B.35) reste identique quel que soit le modèle considéré :

$$\mathcal{H}_{c^*} = \arg \max_{\mathcal{H}_1, \dots, \mathcal{H}_C} \mathbb{P}(\mathcal{H}_c|S^N) = \arg \max_{\mathcal{H}_1, \dots, \mathcal{H}_C} \mathbb{P}(S^N|\mathcal{H}_c)\mathbb{P}_H(\mathcal{H}_c) \quad (\text{B.36})$$

Les observations z_1, \dots, z_N constituant S^N étant supposées indépendantes, un réflexe naturel consisterait à écrire $\mathbb{P}(S^N|\mathcal{H}_c) = \prod_{n=1}^N \mathbb{P}(Z_n = z_n|\mathcal{H}_c)$, ce qui n'a pas de sens pour des variables aléatoires Z_n continues admettant une densité. En effet, dans le cas continu, la probabilité d'observer *exactement* une valeur donnée z est nulle : $\mathbb{P}(Z \in [z, z]) = \int_z^z p(z)dz$, si p est la densité de Z . On écrira donc plutôt, en choisissant Δ petit, Δ étant un intervalle de \mathbb{R} ou un pavé de \mathbb{R}^2 selon le choix de représentation pour Z :

$$\mathbb{P}(S^N|\mathcal{H}_c) = \prod_{n=1}^N \mathbb{P}(Z_n \in [z_n, z_n + \Delta]|\mathcal{H}_c) \approx \Delta^N \prod_{n=1}^N p_{Z_n|\mathcal{H}_c}(z_n) \quad (\text{B.37})$$

Avec Δ et N fixés, maximiser la probabilité $\mathbb{P}(S^N|\mathcal{H}_c)$ revient donc à maximiser la densité $p_{\mathcal{D}|\mathcal{H}_c}(S^N) = \prod_{n=1}^N p_{Z_n|\mathcal{H}_c}(z_n)$, où l'on reconnaît l'expression de la vraisemblance du modèle \mathcal{H}_c au vu des données S^N , pour des variables aléatoires continues admettant une densité.

L'équation (B.36) est donc équivalente à l'équation (B.38) suivante, où $p_{\mathcal{D}|\mathcal{H}_c}$ est la densité conditionnelle de la variable aléatoire \mathcal{D} dont S^N est une réalisation, sachant que le modèle est $H = \mathcal{H}_c$. Par analogie avec les probabilités, de nombreux auteurs notent $p(S^N|\mathcal{H}_c)$ la valeur $p_{\mathcal{D}|\mathcal{H}_c}(S^N)$ de cette densité au point $\mathcal{D} = S^N$.

$$\mathcal{H}_{c^*} = \arg \max_{\mathcal{H}_1, \dots, \mathcal{H}_C} [p_{\mathcal{D}|\mathcal{H}_c}(S^N)\mathbb{P}_H(\mathcal{H}_c)] \quad (\text{B.38})$$

La valeur de la densité $p_{\mathcal{D}|\mathcal{H}_c}(S^N) = p(S^N|\mathcal{H}_c)$ peut s'exprimer en intégrant sur toutes les valeurs possibles du paramètre θ_c , puis en faisant apparaître la densité conditionnelle $p_{\theta|\mathcal{H}_c}$ par application de la formule de Bayes sur $p_{\mathcal{D}|\theta_c, \mathcal{H}_c}(S^N)$:

$$p_{\mathcal{D}|\mathcal{H}_c}(S^N) = \int_{\Theta_c} p_{\mathcal{D}, \theta|\mathcal{H}_c}(S^N, \theta_c) d\theta_c = \int_{\Theta_c} p_{\mathcal{D}|\theta_c, \mathcal{H}_c}(S^N) p_{\theta|\mathcal{H}_c}(\theta_c) d\theta_c \quad (\text{B.39})$$

La connaissance de la dimension du vecteur θ_c suffit à déterminer que l'hypothèse h_{θ_c} a été choisie dans \mathcal{H}_c et on omettra donc par la suite de faire apparaître \mathcal{H}_c dans les conditions, en écrivant que :

$$p_{\mathcal{D}|\mathcal{H}_c}(S^N) = \int_{\Theta_c} p_{\mathcal{D}|\theta_c}(S^N) p_{\theta|\mathcal{H}_c}(\theta_c) d\theta_c \quad (\text{B.40})$$

On reconnaît dans $p_{\mathcal{D}|\theta_c}(S^N)$ la vraisemblance $\mathcal{L}(h_{\theta_c}; S^N)$ de l'hypothèse h_{θ_c} au vu des données S^N . En supposant les observations z_1, \dots, z_N de S^N indépendantes et identiquement distribuées, selon la même loi de densité $p_{h_{\theta_c}}$, on peut écrire :

$$p_{\mathcal{D}|\mathcal{H}_c}(S^N) = \int_{\Theta_c} \underbrace{\left(\prod_{n=1}^N p_{h_{\theta_c}}(z_n) \right)}_{\mathcal{L}(h_{\theta_c}; S^N)} p_{\theta|\mathcal{H}_c}(\theta_c) d\theta_c \quad (\text{B.41})$$

Cette quantité n'étant généralement pas directement calculable, on peut l'approximer en faisant intervenir $\hat{\theta}_c$, la valeur de θ_c où la fonction $\theta_c \rightarrow \mathcal{L}(h_{\theta_c}; S^N) p_{\theta|\mathcal{H}_c}(\theta_c)$ atteint son maximum, appelé *maximum a posteriori* (MAP). Ce point $\hat{\theta}_c$ est différent du maximum de vraisemblance évoqué précédemment et intervenant dans le critère d'Akaike, mais il tend asymptotiquement vers celui-ci pour de grandes valeurs de N , car $p_{\theta|\mathcal{H}_c}(\theta_c)$ reste constant quand N augmente ce qui n'est pas le cas de $\mathcal{L}(h_{\theta_c}; S^N)$. Par conséquent, en passant au logarithme, le maximum de $\theta_c \rightarrow \ln \mathcal{L}(h_{\theta_c}; S^N) + \ln p_{\theta|\mathcal{H}_c}(\theta_c)$ sera sensiblement identique au maximum de $\theta_c \rightarrow \ln \mathcal{L}(h_{\theta_c}; S^N)$ pour N suffisamment grand.

Le principe de l'approximation de l'intégrale de l'équation (B.41) est le suivant, en faisant momentanément abstraction des indices c , pour plus de clarté. On introduit la fonction $L(\theta)$ définie par l'équation (B.42) suivante, et qui atteint son minimum en $\hat{\theta}$:

$$L(\theta) = -\frac{1}{N} [\ln \mathcal{L}(h_{\theta}; S^N) + \ln p_{\theta|\mathcal{H}_c}(\theta)] = -\frac{1}{N} \left[\sum_{n=1}^N \ln p_{h_{\theta_c}}(z_n) + \ln p_{\theta|\mathcal{H}_c}(\theta) \right] \quad (\text{B.42})$$

Un développement de Taylor à l'ordre 2 de cette fonction au point $\hat{\theta}$ où elle atteint son minimum donne, sachant que le gradient s'annule en ce point :

$$L(\theta) \approx L(\hat{\theta}) + \frac{1}{2}(\theta - \hat{\theta})^T C_N(\hat{\theta})(\theta - \hat{\theta}) \quad (\text{B.43})$$

Dans cette équation, $C_N(\hat{\theta})$ est la matrice de dimension $k_c \times k_c$ des dérivées secondes $\left[\frac{\partial^2 L(\theta)}{\partial \theta_i \partial \theta_j}(\hat{\theta}) \right]_{(i,j) \in \{1, \dots, k_c\}^2}$ au point $\hat{\theta}$, et où θ_i et θ_j dénotent les $i^{\text{ième}}$ et $j^{\text{ième}}$ composantes du vecteur θ . En supposant que $C_N(\hat{\theta}) = \Sigma^{-1}$ soit une matrice inversible et en supposant que θ suive une loi sensiblement gaussienne au voisinage de $\hat{\theta}$, on peut alors écrire :

$$e^{-N L(\theta)} \approx e^{-N L(\hat{\theta})} e^{-\frac{1}{2}(\theta - \hat{\theta})^T (N \Sigma^{-1})(\theta - \hat{\theta})} = e^{-N L(\hat{\theta})} \left(\frac{2\pi}{N} \right)^{\frac{k_c}{2}} |\Sigma|^{\frac{1}{2}} p_{\mathcal{N}}(\hat{\theta}, \frac{\Sigma}{N})(\theta) \quad (\text{B.44})$$

Dans cette équation, $p_{\mathcal{N}}(\hat{\theta}, \frac{\Sigma}{N})$ est une densité de probabilité suivant une loi gaussienne multidimensionnelle $\mathcal{N}(\hat{\theta}, \frac{\Sigma}{N})$ de moyenne $\hat{\theta}$ et de matrice de covariance $\frac{\Sigma}{N}$, de déterminant $|\frac{\Sigma}{N}| = (\frac{1}{N})^{k_c} |\Sigma|$.

$$p_{\mathcal{N}}(\hat{\theta}, \frac{\Sigma}{N})(\theta) = (2\pi)^{-\frac{k_c}{2}} \left| \frac{1}{N} \Sigma \right|^{-\frac{1}{2}} e^{-\frac{1}{2}(\theta - \hat{\theta})^T (N \Sigma^{-1})(\theta - \hat{\theta})} \quad (\text{B.45})$$

L'intégrale sur Θ_c de cette densité vaut 1, et on en déduit l'approximation de Laplace de l'intégrale de $e^{-NL(\theta)}$:

$$\int_{\Theta_c} e^{-NL(\theta)} d\theta \approx \left(\frac{2\pi}{N}\right)^{\frac{k_c}{2}} |\Sigma|^{\frac{1}{2}} e^{-NL(\hat{\theta})} \quad (\text{B.46})$$

En développant $e^{-NL(\hat{\theta})} = \mathcal{L}(h_{\hat{\theta}}; S^N) p_{\theta|\mathcal{H}_c}(\hat{\theta})$ et en revenant à l'équation (B.41), on en déduit l'approximation de $p_{\mathcal{D}|\mathcal{H}_c}(S^N)$, en réintroduisant les indices c pour bien signifier qu'on a un *maximum a posteriori* par modèle \mathcal{H}_c :

$$p_{\mathcal{D}|\mathcal{H}_c}(S^N) = \int_{\Theta_c} e^{-NL(\theta_c)} d\theta_c \approx \left(\frac{2\pi}{N}\right)^{\frac{k_c}{2}} |\Sigma|^{\frac{1}{2}} \mathcal{L}(h_{\hat{\theta}_c}; S^N) p_{\theta|\mathcal{H}_c}(\hat{\theta}_c) \quad (\text{B.47})$$

Choisir le meilleur modèle *a posteriori* au vu des données en utilisant l'équation (B.38) revient alors à écrire :

$$\begin{aligned} \mathcal{H}_{c^*} &= \arg \max_{\mathcal{H}_1, \dots, \mathcal{H}_C} [p_{\mathcal{D}|\mathcal{H}_c}(S^N) \mathbb{P}_H(\mathcal{H}_c)] \\ &\approx \arg \max_{\mathcal{H}_1, \dots, \mathcal{H}_C} \left[\left(\frac{2\pi}{N}\right)^{\frac{k_c}{2}} |\Sigma|^{\frac{1}{2}} \mathcal{L}(h_{\hat{\theta}_c}; S^N) p_{\theta|\mathcal{H}_c}(\hat{\theta}_c) \mathbb{P}_H(\mathcal{H}_c) \right] \end{aligned} \quad (\text{B.48})$$

En passant au logarithme, on obtient :

$$\mathcal{H}_{c^*} \approx \arg \max_{\mathcal{H}_1, \dots, \mathcal{H}_C} \left[\ln \mathcal{L}(h_{\hat{\theta}_c}; S^N) - \frac{k_c}{2} \ln(N) + \frac{k_c}{2} \ln(2\pi) - \frac{1}{2} \ln |\Sigma| + \ln p_{\theta|\mathcal{H}_c}(\hat{\theta}_c) + \ln \mathbb{P}_H(\mathcal{H}_c) \right] \quad (\text{B.49})$$

En règle générale, on suppose que les modèles $\mathcal{H}_1, \dots, \mathcal{H}_C$ sont équiprobables et donc que $\ln \mathbb{P}_H(\mathcal{H}_c)$ est une constante. Par ailleurs, pour N grand, les termes prépondérants dans le membre de droite de l'équation (B.49) sont les deux premiers, les autres étant d'ordre 1.

En multipliant (B.49) par 2 et en supposant N suffisamment grand, on voit donc qu'il suffit de minimiser $-2 \ln \mathcal{L}(h_{\hat{\theta}_c}; S) + k_c \ln N$, où on retrouve l'expression du critère BIC, pour sélectionner le modèle dont la probabilité *a posteriori*, au vu des données, est la plus grande.

Dans l'expression de la log-vraisemblance $\ln \mathcal{L}(h_{\hat{\theta}_c}; S)$, le vecteur $\hat{\theta}_c$ correspond au *maximum a posteriori* $\hat{\theta}_c = \hat{\theta}_c^{MAP}$, mais nous avons vu que pour N grand, on peut l'approximer par le maximum de vraisemblance $\hat{\theta}_c = \hat{\theta}_c^{ML}$.

Annexe C

Géodésie

C.1 Introduction

Cette annexe est un très court résumé de quelques-unes des notions de géodésie utiles au calcul des trajectoires d'avions volant au-dessus de la surface terrestre. Pour une documentation plus complète, le lecteur pourra se référer au livre de Michel Capderou ([31]) sur les satellites, et plus particulièrement aux deux premiers chapitres pour la géométrie de l'ellipse et la géodésie, ainsi qu'au troisième chapitre du livre de Dominic J. Diston [44].

On dispose de plusieurs modèles possibles de la surface terrestre parmi lesquels le géoïde, l'ellipsoïde de révolution, et la sphère. Le géoïde est défini comme la surface équipotentielle du champ de pesanteur épousant le niveau moyen des océans. Le calcul de cette surface équipotentielle n'est pas simple et on l'approxime généralement par une surface plus facile à calculer, l'ellipsoïde de révolution, où la Terre est considérée comme une sphère aplatie aux pôles, chaque section dans un plan méridien étant une ellipse de paramètres a pour le grand axe dans le plan équatorial, et b pour le petit axe entre les pôles. Les paramètres a et b de l'ellipse sont constants, quel que soit le méridien. Pour certaines applications ou certains calculs, on peut parfois se contenter d'un modèle sphérique de la Terre.

Nous allons brièvement décrire le modèle sphérique et l'ellipsoïde de révolution dans la suite de cette annexe, en commençant par la sphère pour des raisons de simplicité.

C.2 Le modèle sphérique

C.2.1 Le référentiel lié à la Terre (ECEF)

La Terre est ici modélisée comme une sphère de rayon R_T . On définit un référentiel fixé à la Terre (ECEF pour *Earth centered, earth fixed* en anglais), dont l'axe z_e passe par les pôles et est orienté du pôle sud vers le pôle nord. L'axe x_e est choisi arbitrairement de façon à passer par le méridien de Greenwich. L'axe y_e vient compléter ce système d'axe de façon à former un repère orthonormé, d'origine O le centre de la Terre.

Dans la suite, on notera $(\vec{i}_e, \vec{j}_e, \vec{k}_e)$ le repère orthonormé ECEF lié à la Terre.

Dans ce système, la position d'un point P est donnée par sa latitude, sa longitude, et sa distance au centre de la Terre (ou plus simplement son altitude h au-dessus de la surface du globe). La latitude, que nous noterons μ dans la suite, est l'angle entre le plan équatorial x_eOy_e et \vec{OP} . La longitude, notée λ , est l'angle entre le plan méridien dans lequel se situe P

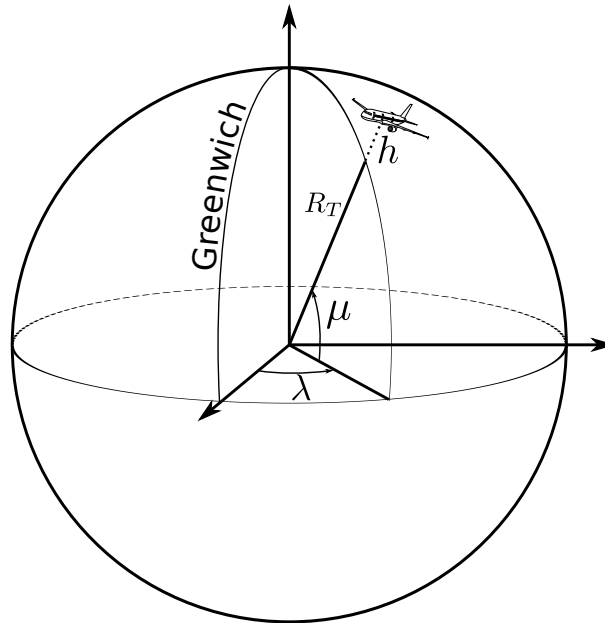


FIGURE C.1: Modèle sphérique de la Terre.

et le plan méridien de référence $x_e O z_e$. La Figure C.1 illustre ce modèle de la sphère terrestre.

C.2.2 Le référentiel horizontal local (NED)

Pour un point P donné, situé sur la surface terrestre ou à proximité, on définit également un autre système d'axes, que nous appellerons référentiel horizontal local, ou NED (*North, East, Down*), lié au point P . Dans ce système d'axes noté $x_h y_h z_h$ l'axe x_h pointe vers le Nord, y_h vers l'Est, et z_h vers le centre de la Terre, comme représenté sur la Figure C.2.

On y associe une base orthonormée de vecteurs $(\vec{i}_h, \vec{j}_h, \vec{k}_h)$.

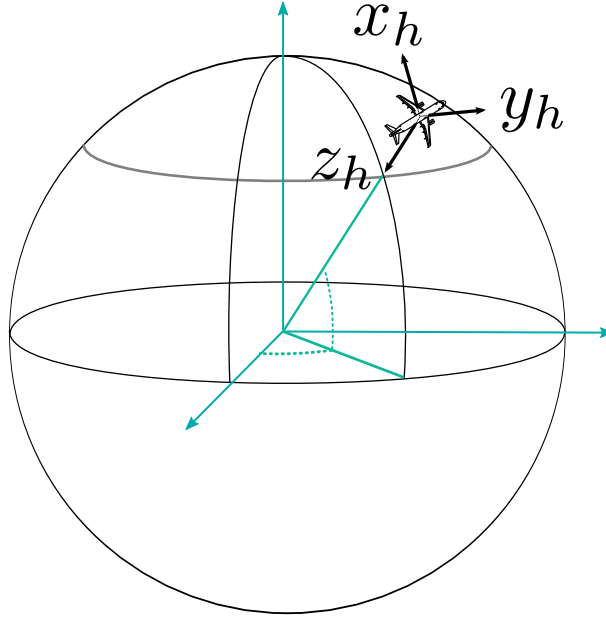
C.2.3 Coordonnées d'un point

Les coordonnées cartésiennes, dans le repère ECEF lié à la Terre, d'un point P situé à une altitude h sont données par l'équation (C.1) :

$$\{\overrightarrow{OP}\}_{ECEF} \begin{cases} x_e = (R_T + h) \cos \mu \cos \lambda \\ y_e = (R_T + h) \cos \mu \sin \lambda \\ z_e = (R_T + h) \sin \mu \end{cases} \quad (\text{C.1})$$

Dans le repère NED, où le vecteur unitaire \vec{k}_h pointe de P vers le centre de la Terre O , le vecteur position s'exprime simplement par l'équation (C.2), et les coordonnées par (C.3).

$$\overrightarrow{OP} = -(R_T + h) \vec{k}_h \quad (\text{C.2})$$

FIGURE C.2: Le repère horizontal local NED (*North, East, Down*) pour une terre sphérique.

$$\{\overrightarrow{OP}\}_{NED} \begin{cases} x_h = 0 \\ y_h = 0 \\ z_h = -(R_T + h) \end{cases} \quad (\text{C.3})$$

C.2.4 Vitesse d'un point par rapport au référentiel terrestre

Soit $Ox_my_mz_e$ le repère orthonormé direct tel que x_mOz_e soit le plan méridien contenant la position P du mobile. Ce repère est simplement obtenu par rotation du référentiel terrestre ECEF d'axes $Ox_e y_e z_e$ d'un angle λ autour de l'axe des pôles z_e . Soit $(\vec{i}_m, \vec{j}_m, \vec{k}_e)$ la base orthonormale associée au référentiel fixé au plan méridien passant par P .

La vitesse d'un mobile par rapport au référentiel ECEF, considéré comme fixe, s'exprime de la façon suivante en tenant compte de la rotation $\vec{\Omega}_{Ox_my_mz_e/ECEF} = \dot{\lambda} \vec{k}_e$ du repère $Ox_my_mz_e$ autour de l'axe Oz_e

$$\frac{d\overrightarrow{OP}}{dt} /_{ECEF} = \frac{d\overrightarrow{OP}}{dt} /_{Ox_my_mz_e} + \vec{\Omega}_{Ox_my_mz_e/ECEF} \wedge \overrightarrow{OP} \quad (\text{C.4})$$

La dérivée $\frac{d\overrightarrow{OP}}{dt} /_{Ox_my_mz_e}$ s'obtient simplement en dérivant (C.2) en supposant $Ox_my_mz_e$ fixe. Avec cette hypothèse, on se place dans le plan méridien et on ne tient compte que du mouvement de rotation de \vec{k}_h dans ce plan, lorsqu'on calcule sa dérivée :

$$\frac{d\overrightarrow{OP}}{dt} /_{Ox_my_mz_e} = -(R_T + h) \frac{d\mu}{dt} \frac{d\vec{k}_h}{d\mu} - \dot{h} \vec{k}_h = \dot{\mu}(R_T + h) \vec{i}_h - \dot{h} \vec{k}_h \quad (\text{C.5})$$

En combinant (C.5) et (C.4), la vitesse par rapport au référentiel ECEF s'exprime alors sous une forme simple dans la base de vecteurs $(\vec{v}_h, \vec{j}_h, \vec{k}_h)$ associée au repère horizontal local NED.

$$\begin{aligned} \frac{d\vec{OP}}{dt} /_{ECEF} &= \dot{\mu}(R_T + h)\vec{v}_h - \dot{h}\vec{k}_h + \left(\dot{\lambda}\vec{k}_e\right) \wedge \left(- (R_T + h)\vec{k}_h\right) \\ &= \dot{\mu}(R_T + h)\vec{v}_h + \dot{\lambda}(R_T + h)\cos\mu\vec{j}_h - \dot{h}\vec{k}_h \end{aligned} \quad (C.6)$$

En adoptant la convention de notation, un peu lourde mais explicite, où “/ECEF” signifie “par rapport au référentiel terrestre” et $\{.\}_{NED}$ signifie “dans le système de coordonnées NED” (*North, East, Down*), les coordonnées de la vitesse par rapport à la Terre sont données par l'équation (C.6).

$$\left\{ \frac{d\vec{OP}}{dt} /_{ECEF} \right\}_{NED} = \begin{pmatrix} \dot{\mu}(R_T + h) \\ \dot{\lambda}(R_T + h)\cos\mu \\ -\dot{h} \end{pmatrix} \quad (C.7)$$

C.3 L'ellipsoïde de révolution

Une approximation plus précise que la sphère de la surface terrestre est l'ellipsoïde de révolution. Dans ce modèle, chaque section dans un plan méridien est une ellipse de grand axe a et de petit axe b . Les paramètres a et b de l'ellipse sont les mêmes dans tous les plans méridiens. La surface approximée de la Terre est donc déterminée par une seule ellipse, que l'on fait tourner autour de son petit axe confondu avec l'axe des pôles.

Dans la suite, on notera e l'excentricité de l'ellipse, qui vérifie l'équation(C.8) suivante :

$$e^2 = 1 - \frac{b^2}{a^2} \quad (C.8)$$

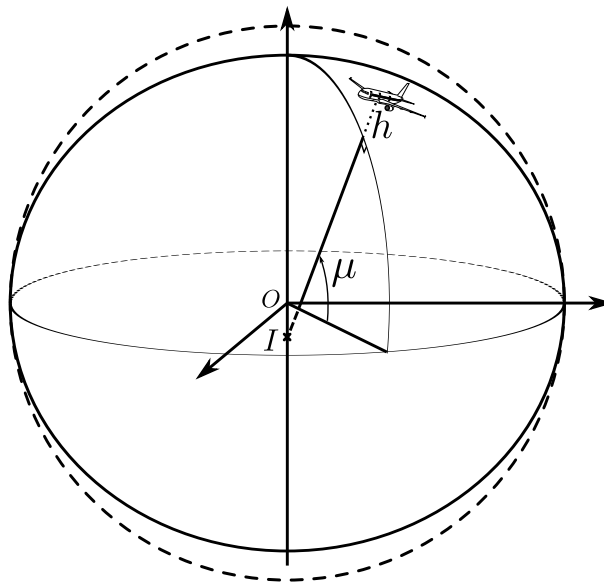
On rappelle en (C.9) l'équation de l'ellipse, en notant z le petit axe, confondu avec l'axe des pôles, et x le grand axe de l'ellipse :

$$\frac{x^2}{a^2} + \frac{z^2}{b^2} = 1 \quad (C.9)$$

C.3.1 Définitions de différentes latitudes

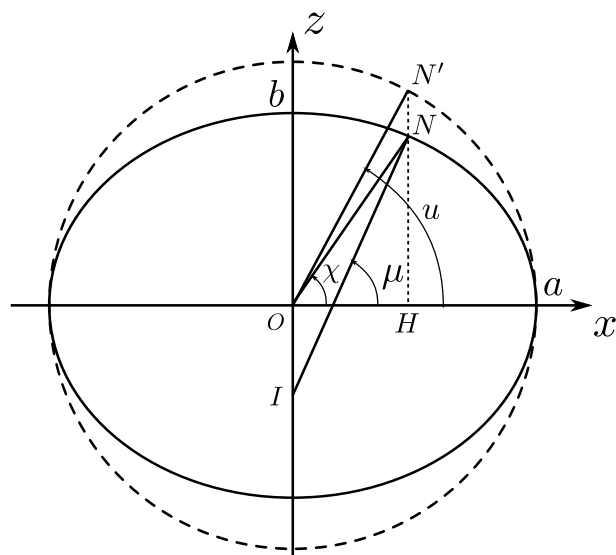
Historiquement, la latitude était calculée par des mesures d'angles entre le plan horizontal local et la direction d'étoiles de référence (l'étoile polaire par exemple). Le plan horizontal est déterminé comme étant perpendiculaire à la direction du fil à plomb. Cette direction est donc la normale à la surface équipotentielle représentée par le géoïde local. Cette latitude mesurée “sur le terrain” est appelée *latitude géographique* (ou astronomique).

On appelle *latitude géodésique* l'angle entre le plan équatorial et la normale, au point N considéré, à la surface de l'ellipsoïde de révolution (voir FigureC.3). Notons que cette normale ne passe pas par le centre O , mais coupe l'axe polaire en un point I . La distance IN est appelé “grande normale” et notée \mathcal{N} .

FIGURE C.3: Latitude (μ) et altitude (h) géodésiques sur un ellipsoïde de révolution.

La *latitude géocentrique* est l'angle entre le plan équatorial et l'axe allant du centre de la Terre O au point N considéré. Enfin, on utilise pour les besoins du calcul une *latitude paramétrée* ou *latitude réduite*, qui correspond à la latitude géocentrique d'un point sur un cercle de centre O de rayon a , lequel point est obtenu en suivant la parallèle à l'axe Oz passant par le point N (voir Figure C.4).

Dans la suite du document, on notera μ la latitude géodésique, χ la latitude géocentrique, et u la latitude paramétrée.

FIGURE C.4: Latitude géocentrique (χ), géodésique (μ), et paramétrée (u).

C.3.2 Quelques caractéristiques utiles de l'ellipse

Relations entre latitudes géodésique, géocentrique, et paramétrée

Rappelons que l'ellipse est une transformation affine, d'axe Ox de direction Oz et de rapport $\frac{b}{a}$, du cercle de centre O et de rayon a . En d'autres termes, les points de l'ellipse sont obtenus à partir du cercle de centre O et de rayon a en appliquant un facteur multiplicatif $\frac{b}{a}$ à la coordonnée z .

Par conséquent, sur la figure C.4, on observe que $\tan \mu = \frac{HN}{OH} = \frac{b}{a} \frac{HN'}{OH}$. On en déduit de façon immédiate la relation (C.10) entre la latitude géocentrique χ et la latitude paramétrée u .

$$\tan \chi = \frac{b}{a} \tan u \quad (\text{C.10})$$

Par ailleurs, en différenciant (C.9), on obtient la pente de la tangente à l'ellipse au point N :

$$\frac{dz}{dx} = -\frac{b^2}{a^2} \frac{x}{z} \quad (\text{C.11})$$

La direction de la normale est alors donnée par :

$$\tan \mu = \frac{a^2}{b^2} \frac{z}{x} \quad (\text{C.12})$$

D'après la définition de la latitude géocentrique, on voit par ailleurs que :

$$\tan \chi = \frac{z}{x} \quad (\text{C.13})$$

On en déduit la relation entre la latitude géocentrique χ et la latitude géodésique μ :

$$\boxed{\tan \chi = \frac{b^2}{a^2} \tan \mu} \quad (\text{C.14})$$

En combinant (C.10) et (C.14), on déduit facilement la relation entre la latitude paramétrée u et la latitude géodésique μ :

$$\tan u = \frac{b}{a} \tan \mu \quad (\text{C.15})$$

Coordonnées d'un point de l'ellipse

Le point N' de la Figure C.4 est sur un cercle de rayon a . Ces coordonnées s'expriment simplement en fonction de la latitude paramétrée :

$$\left. \begin{array}{l} \overrightarrow{ON'} \end{array} \right\}_{Oxz} \left| \begin{array}{l} x = a \cos u \\ z = a \sin u \end{array} \right. \quad (\text{C.16})$$

Le point N de l'ellipse s'obtient par une transformation affine du point N' , de rapport $\frac{b}{a}$. La coordonnée z de N est donc simplement $\frac{b}{a}(a \sin u)$, et on a donc :

$$\{\overrightarrow{ON}\}_{Oxz} \left| \begin{array}{l} x = a \cos u \\ z = b \sin u \end{array} \right. \quad (\text{C.17})$$

Par ailleurs, en considérant la latitude géodésique μ et la grande normale $\mathcal{N} = IN$, on observe que $x = \mathcal{N} \cos \mu$. En tenant compte de cette expression et de (C.15), la coordonnée z se réécrit de la façon suivante en fonction de la latitude géodésique :

$$z = b \sin u = \frac{b}{a} \tan u (a \cos u) = \frac{b^2}{a^2} \tan \mu (\mathcal{N} \cos \mu) = \mathcal{N} \frac{b^2}{a^2} \sin \mu \quad (\text{C.18})$$

En introduisant l'excentricité (C.8), on obtient finalement :

$$\{\overrightarrow{ON'}\}_{Oxz} \left| \begin{array}{l} x = \mathcal{N} \cos \mu \\ z = \mathcal{N}(1 - e^2) \sin \mu \end{array} \right. \quad (\text{C.19})$$

Expression de la grande normale

L'expression de la grande normale \mathcal{N} se retrouve simplement à partir de (C.19) et de l'équation de l'ellipse (C.9) :

$$\begin{aligned} & \frac{x^2}{a^2} + \frac{z^2}{b^2} = 1 \\ \Leftrightarrow & \frac{\mathcal{N}^2 \cos^2 \mu}{a^2} + \frac{\mathcal{N}^2 (1 - e^2)^2 \sin^2 \mu}{b^2} = 1 \\ \Leftrightarrow & \mathcal{N}^2 \left[1 - \sin^2 \mu + \frac{a^2}{b^2} (1 - e^2)^2 \sin^2 \mu \right] = a^2 \\ \Leftrightarrow & \mathcal{N}^2 [1 - \sin^2 \mu + (1 - e^2) \sin^2 \mu] = a^2 \\ \Leftrightarrow & \mathcal{N}^2 (1 - e^2 \sin^2 \mu) = a^2 \end{aligned} \quad (\text{C.20})$$

On en déduit l'expression de la grande normale :

$$\boxed{\mathcal{N} = \frac{a}{\sqrt{1 - e^2 \sin^2 \mu}}} \quad (\text{C.21})$$

C.3.3 Référentiel ECEF et repères mobiles pour l'ellipsoïde de révolution

Dans le cas de l'ellipsoïde de révolution, on utilise un référentiel terrestre ECEF (*Earth Centered Earth Fixed*) similaire à celui du modèle sphérique. Son système d'axes est ici noté $Ox_e y_e z_e$, avec une base de vecteurs orthonormés $(\vec{i}_e, \vec{j}_e, \vec{k}_e)$. Pour les besoins du calcul, on utilisera également un repère orthonormé $(\vec{i}_m, \vec{j}_m, \vec{k}_e)$ obtenu par une rotation d'angle λ autour de l'axe des pôles z_e . Les vecteurs \vec{i}_m et \vec{k}_e sont donc dans le plan méridien passant par N (et P). On notera $Ox_m y_m z_e$ le système d'axe associé.

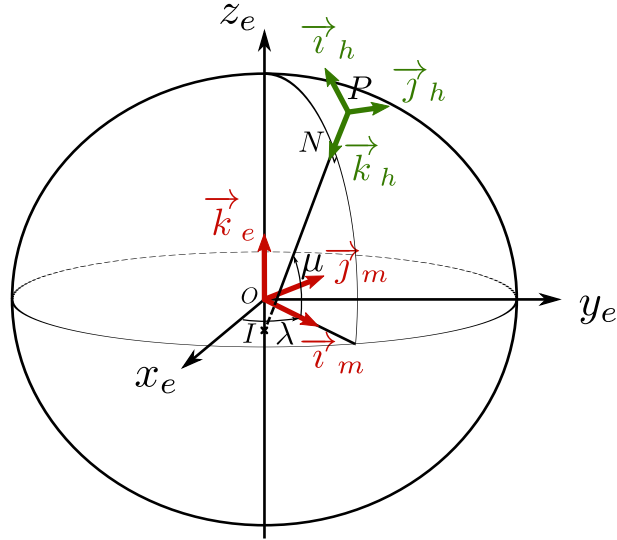


FIGURE C.5: Référentiel terrestre et repères mobiles pour l'ellipsoïde de révolution.

Le repère horizontal local NED (*North, East, Down*), de base orthonormée $(\vec{i}_h, \vec{j}_h, \vec{k}_h)$, est similaire à celui du modèle sphérique, sauf que la verticale déterminée par le vecteur \vec{k}_h ne passe plus vers le centre de la Terre mais est définie comme la normale à la surface de l'ellipsoïde. Le vecteur \vec{i}_h pointe vers le Nord, et \vec{j}_h vers l'Est. Ces différents repères et systèmes d'axes sont décrits sur la Figure C.5.

C.3.4 Coordonnées d'un point

La position d'un point P situé à une altitude h au dessus de la surface de l'ellipsoïde de révolution peut être décomposée comme suit (voir Figure C.5) :

$$\vec{OP} = \vec{ON} + \vec{NP} \quad (\text{C.22})$$

Dans cette expression, le point N (nadir) est le point sur la surface de l'ellipsoïde situé à la verticale du point P .

Coordonnées du nadir N

$$\{\vec{ON}\}_{Ox_my_mz_e} \left| \begin{array}{l} x_m = \mathcal{N} \cos \mu \\ y_m = 0 \\ z_e = \mathcal{N}(1 - e^2) \sin \mu \end{array} \right. \quad (\text{C.23})$$

$$\{\vec{ON}\}_{ECEF} \left| \begin{array}{l} x_e = \mathcal{N} \cos \mu \cos \lambda \\ y_e = \mathcal{N} \cos \mu \sin \lambda \\ z_e = \mathcal{N}(1 - e^2) \sin \mu \end{array} \right. \quad (\text{C.24})$$

Dans ces équations, \mathcal{N} est la "grande normale" au point N , donnée par l'équation (C.21).

Coordonnées du point P à l'altitude h

$$\{\overrightarrow{OP}\}_{Ox_my_mz_e} \left| \begin{array}{l} x_m = (\mathcal{N} + h) \cos \mu \\ y_m = 0 \\ z_e = [\mathcal{N}(1 - e^2) + h] \sin \mu \end{array} \right. \quad (\text{C.25})$$

$$\{\overrightarrow{OP}\}_{ECEF} \left| \begin{array}{l} x_e = (\mathcal{N} + h) \cos \mu \cos \lambda \\ y_e = (\mathcal{N} + h) \cos \mu \sin \lambda \\ z_e = [\mathcal{N}(1 - e^2) + h] \sin \mu \end{array} \right. \quad (\text{C.26})$$

C.3.5 Vitesse d'un point par rapport au référentiel terrestre

$$\frac{d\overrightarrow{ON}}{dt} /_{Ox_my_mz_e} = \dot{\mu} \left(\frac{d\mathcal{N}}{d\mu} \cos \mu - \mathcal{N} \sin \mu \right) \vec{v}_m + (1 - e^2) \dot{\mu} \left(\frac{d\mathcal{N}}{d\mu} \sin \mu + \mathcal{N} \cos \mu \right) \vec{k}_e \quad (\text{C.27})$$

$$\text{avec } \frac{d\mathcal{N}}{d\mu} = ae^2 \sin \mu \cos \mu (1 - e^2 \sin^2 \mu)^{-\frac{3}{2}} \quad (\text{C.28})$$

$$\begin{aligned} \frac{d\overrightarrow{ON}}{dt} /_{Ox_my_mz_e} &= \dot{\mu} \left[ae^2 \sin \mu \cos^2 \mu (1 - e^2 \sin^2 \mu)^{-\frac{3}{2}} - a (1 - e^2 \sin^2 \mu)^{-\frac{1}{2}} \sin \mu \right] \vec{v}_m \\ &\quad + \dot{\mu} (1 - e^2) \left[ae^2 \sin^2 \mu \cos \mu (1 - e^2 \sin^2 \mu)^{-\frac{3}{2}} + a (1 - e^2 \sin^2 \mu)^{-\frac{1}{2}} \cos \mu \right] \vec{k}_e \\ &= \dot{\mu} \frac{a}{(1 - e^2 \sin^2 \mu)^{\frac{3}{2}}} \sin \mu \left[e^2 \cos^2 \mu - (1 - e^2 \sin^2 \mu) \right] \vec{v}_m \\ &\quad + \dot{\mu} \frac{a(1 - e^2)}{(1 - e^2 \sin^2 \mu)^{\frac{3}{2}}} \cos \mu \left[e^2 \sin^2 \mu + 1 - e^2 \sin^2 \mu \right] \vec{k}_e \\ &= \dot{\mu} \frac{a(1 - e^2)}{(1 - e^2 \sin^2 \mu)^{\frac{3}{2}}} \left[-\sin \mu \vec{v}_m + \cos \mu \vec{k}_e \right] \\ &= \dot{\mu} \frac{a(1 - e^2)}{(1 - e^2 \sin^2 \mu)^{\frac{3}{2}}} \vec{v}_h \end{aligned} \quad (\text{C.29})$$

$$\frac{d\overrightarrow{ON}}{dt} /_{Ox_my_mz_e} = \dot{\mu} R_M \vec{v}_h \quad (\text{C.30})$$

où R_M est le rayon de courbure méridien (voir Figure C.6) :

$$\boxed{R_M = \frac{a(1 - e^2)}{(1 - e^2 \sin^2 \mu)^{\frac{3}{2}}}} \quad (\text{C.31})$$

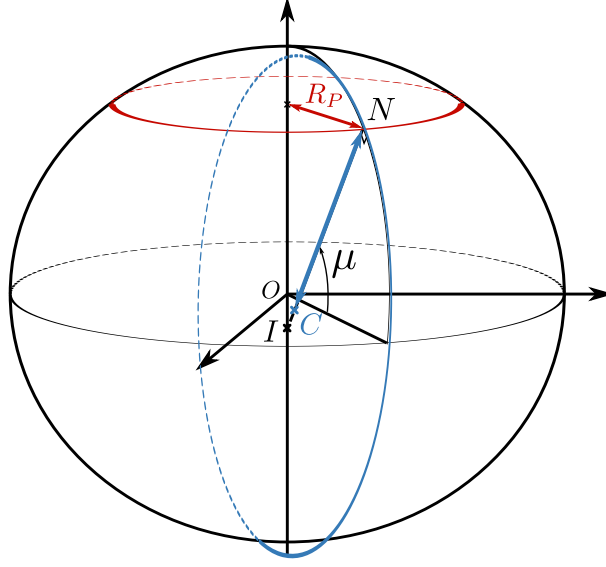


FIGURE C.6: Rayons du cercle parallèle (R_P) et rayon de courbure méridien (R_M) sur l'ellipsoïde de révolution.

On a par ailleurs :

$$\begin{aligned} \frac{d\overrightarrow{NP}}{dt} /_{Ox_my_mz_e} &= -\dot{h}\vec{k}_h - h\frac{d\vec{k}_h}{dt} = -\dot{h}\vec{k}_h - h\dot{\mu}\frac{d\vec{k}_h}{d\mu} \\ &= h\dot{\mu}\vec{v}_h - \dot{h}\vec{k}_h \end{aligned} \quad (\text{C.32})$$

On en déduit :

$$\frac{d\overrightarrow{OP}}{dt} /_{Ox_my_mz_e} = \dot{\mu}(R_M + h)\vec{v}_h - \dot{h}\vec{k}_h \quad (\text{C.33})$$

En tenant compte de la rotation $\vec{\Omega}_{Ox_my_mz_e/ECEF} = \dot{\lambda}\vec{k}_e$ du repère $Ox_my_mz_e$ autour de l'axe Oz_e , on obtient :

$$\begin{aligned} \frac{d\overrightarrow{OP}}{dt} /_{ECEF} &= \frac{d\overrightarrow{OP}}{dt} /_{Ox_my_mz_e} + \vec{\Omega}_{Ox_my_mz_e/ECEF} \wedge \overrightarrow{OP} \\ &= \dot{\mu}(R_M + h)\vec{v}_h - \dot{h}\vec{k}_h + \dot{\lambda}\vec{k}_e \wedge (\overrightarrow{ON} + \overrightarrow{NP}) \\ &= \dot{\mu}(R_M + h)\vec{v}_h + \dot{\lambda}(\mathcal{N} + h)\cos\mu\vec{j}_h - \dot{h}\vec{k}_h \end{aligned} \quad (\text{C.34})$$

En introduisant $R_P = \mathcal{N} \cos \mu$, le rayon du parallèle passant par le point N (voir Figure C.6), on obtient finalement l'expression (C.35) :

$$\left\{ \frac{d\overrightarrow{OP}}{dt} /_{ECEF} \right\}_{NED} = \begin{pmatrix} \dot{\mu}(R_M + h) \\ \dot{\lambda}(\mathcal{N} + h)\cos\mu \\ -\dot{h} \end{pmatrix} = \begin{pmatrix} \dot{\mu}(R_M + h) \\ \dot{\lambda}(R_P + h\cos\mu) \\ -\dot{h} \end{pmatrix} \quad (\text{C.35})$$

Dans cette équation, le rayon de courbure méridien R_M est donné par (C.31), et le rayon R_P du cercle parallèle est donné par l'équation (C.36) suivante :

$$\boxed{R_P = \mathcal{N} \cos \mu = \frac{a \cos \mu}{\sqrt{1 - e^2 \sin^2 \mu}}} \quad (\text{C.36})$$

Annexe D

Atmosphères ISA et non-ISA pour l'altimétrie

Les éléments présentés dans cette annexe sont pour l'essentiel tirés du manuel OACI (Organisation de l'Aviation Civile Internationale) de l'atmosphère-type [95], et du modèle révisé de l'atmosphère pour le modèle de performances BADA (*Base of Aircraft Data*) d'Eurocontrol [132].

D.1 L'atmosphère standard ISA

D.1.1 Équation hydrostatique

$$-dp = \rho g dh \tag{D.1}$$

où ρ est la densité de l'air.

D.1.2 Équation des gaz parfaits

$$p = \rho RT \tag{D.2}$$

Dans cette équation, R est la constante spécifique pour l'air sec ($R = 287,05287 \text{ m}^2/K.s^2$), et T est la température de l'air.

D.1.3 Altitude géopotentielle

Dans l'équation (D.1), l'accélération due à la gravité terrestre $g = g(h)$ varie avec l'altitude h . Selon la modélisation choisie pour le globe terrestre, l'expression de $g(h)$ peut être plus ou moins complexe.

Pour simplifier l'expression des équations du modèle atmosphérique, on introduit une nouvelle grandeur, l'altitude géopotentielle H , définie par l'équation (D.3), où g_0 est une

valeur de référence pour l'accélération due à la gravité $g_0 = 9,80665 \text{ m/s}^2$ (valeur prise au niveau moyen de la mer à une latitude de référence)

$$gdh = g_0 dH \quad (\text{D.3})$$

D.1.4 Caractéristiques physiques de l'atmosphère au niveau de la mer

Les principales caractéristiques de l'atmosphère ISA au niveau moyen de la mer sont rappelées dans le tableau D.1.

Accélération due à la gravité	$g_0 = 9,80665$	[m/s ²]
Pression atmosphérique	$p_0 = 101325$	[Pa]
Température	$T_0 = 288,15$	[K]
Densité atmosphérique	$\rho_0 = 1,225$	[kg/m ³]
Vitesse du son	$a_0 = 340,294$	[m/s]

TABLE D.1: Valeurs standards des principaux paramètres, au niveau moyen de la mer

D.1.5 Température et gradient vertical de température

La température est une fonction linéaire par morceaux de l'altitude géopotentielle.

$$T = T_b + \beta_b(H - H_b) \quad (\text{D.4})$$

Les différentes couches d'atmosphère, avec les valeurs du gradient de température jusqu'à l'altitude de 80 km, sont décrites dans la table D.2. Notons que l'équation (D.4) peut être utilisée pour des altitudes négatives, avec les paramètres de la couche 0.

N° couche b	Altitude géopotentielle H_p , [km]	Limite inf. T_b , [K]	Gradient β_b , [K/km]	Nom
0	0	288.15	-6.5	troposphère
1	11	216.65	0	stratosphère
2	20	216.65	+1.0	stratosphère
3	32	228.65	+2.8	stratosphère
4	47	270.65	0	stratosphère
5	51	270.65	-2.8	mésosphère
6	71	214.65	-2.0	mésosphère

TABLE D.2: Profil de température, jusqu'à la mésopause (80 km).

L'aviation commerciale est concernée par les deux première couches, dans la troposphère et le début de la stratosphère. La troposphère et la stratosphère sont séparées par une surface isobare, la tropopause, à 11 km d'altitude.

D.1.6 Pression atmosphérique

L'équation (D.2) nous donne une expression de la densité de l'air $\rho = \frac{p}{RT}$ que l'on peut remplacer dans l'équation (D.1), en introduisant l'altitude géopotentielle (D.3).

$$dp = -\rho g dh = -\rho g_0 dH = -\frac{g_0}{RT} p dH \quad (\text{D.5})$$

En tenant compte de l'expression de la température (D.4) dans la couche d'altitude (numérotée b) où l'on se situe, on obtient :

$$\frac{dp}{p} = -\frac{g_0}{RT} dH = -\left(\frac{g_0}{R}\right) \frac{dH}{T_b + \beta_b(H - H_b)} \quad (\text{D.6})$$

Ceci conduit à deux expressions possibles de la pression en fonction de l'altitude géopotentielle, selon la valeur du gradient de température :

$$\beta_b \neq 0 \quad \ln \frac{p}{p_b} = -\frac{g_0}{R\beta_b} \ln \left(\frac{T_b + \beta_b(H - H_b)}{T_b} \right) \quad (\text{D.7a})$$

$$\beta_b = 0 \quad \ln \frac{p}{p_b} = -\frac{g_0}{RT_b} (H - H_b) \quad (\text{D.7b})$$

$$\beta_b \neq 0 \quad p = p_b \left[\frac{T_b + \beta_b(H - H_b)}{T_b} \right]^{-\frac{g_0}{R\beta_b}} \quad (\text{D.8a})$$

$$\beta_b = 0 \quad p = p_b \exp \left[-\frac{g_0}{RT_b} (H - H_b) \right] \quad (\text{D.8b})$$

D.1.7 Altitude géopotentielle en fonction de la pression

Inversement, en repartant de (D.7a), on exprime facilement l'altitude géopotentielle H en fonction de la pression atmosphérique.

$$\beta_b \neq 0 \quad H = H_b + \frac{T_b}{\beta_b} \left[\left(\frac{p}{p_b} \right)^{-\frac{R\beta_b}{g_0}} - 1 \right] \quad (\text{D.9a})$$

$$\beta_b = 0 \quad H = H_b - \frac{RT_b}{g_0} \ln \left(\frac{p}{p_b} \right) \quad (\text{D.9b})$$

D.1.8 Densité de l'air

La densité de l'air s'exprime simplement à partir de l'équation (D.2).

$$\rho = \frac{p}{RT} \quad (\text{D.10})$$

Elle s'exprime en fonction de l'altitude géopotentielle en remplaçant la pression et la température par leur expressions (D.8a) et (D.4). On trouve alors l'équation (D.11a), où

$\rho_b = \frac{p_b}{RT_b}$ est la densité de l'air à la base de la couche d'altitude considérée.

$$\beta_b \neq 0 \quad \rho = \rho_b \left[\frac{T_b + \beta_b(H - H_b)}{T_b} \right]^{-\frac{g_0}{R\beta_b} - 1} \quad (\text{D.11a})$$

$$\beta_b = 0 \quad \rho = \rho_b \exp \left[-\frac{g_0}{RT_b}(H - H_b) \right] \quad (\text{D.11b})$$

D.1.9 Vitesse du son

La vitesse du son dans l'air est donnée par l'équation suivante, où $\kappa = 1,4$ pour l'air :

$$a = \sqrt{\kappa RT} \quad (\text{D.12})$$

D.2 Les atmosphères non-ISA pour l'altimétrie

D.2.1 Les atmosphères non standard

En règle générale, l'atmosphère réelle ne satisfait pas les hypothèses de l'atmosphère standard ISA. Une atmosphère peut être non-ISA de bien des manières différentes. Tout d'abord, elle n'est pas toujours constituée que d'air sec. Par ailleurs, les conditions de température et de pression au niveau de la mer et/ou le gradient de température peuvent être différents de ceux définis par l'atmosphère standard.

Pour la mesure de l'altitude des avions, on fait toutefois des hypothèses relativement simples sur l'atmosphère, qui font appel à la notion d'altitude-pression, et que nous allons détailler dans la suite de cette section.

D.2.2 Notion d'altitude-pression

Soit p la pression à l'altitude géopotentielle H , dans l'atmosphère non-ISA modélisée dans cette section. L'altitude-pression géopotentielle (ou simplement *altitude-pression*) est définie comme l'altitude géopotentielle à laquelle on mesurerait la pression p si l'atmosphère était standard.

L'altitude-pression est notée H_p . Notons que, par définition, l'isobare $p = p_0 = 1013,25 \text{ hPa}$ correspond à une altitude-pression nulle ($H_p = 0$)

D.2.3 Équilibre hydrostatique, loi des gaz parfaits, humidité

Pour les besoins d'altimétrie, on suppose que l'atmosphère non-ISA que l'on modélise dans cette section est à l'équilibre hydrostatique, et qu'elle suit la loi des gaz parfaits. L'humidité de l'air n'est pas prise en compte, comme dans le modèle ISA. Les équations (D.1) et (D.2) qui avaient été faites pour le modèle ISA restent donc valables, avec la même constante spécifique R pour l'air.

D.2.4 Hypothèses sur le gradient de température

On fait les hypothèses suivantes concernant le profil de température, pour ce qui concerne les besoins en altimétrie :

- les couches d'atmosphère sont définies en altitude-pression H_p , et non plus en altitude géopotentielle H^1 . Pour les besoins de l'aviation, on peut se contenter de décrire seulement les deux couches les plus basses : la troposphère, et la stratosphère, séparées par la tropopause située à l'altitude-pression donnée par (D.13) :

$$H_{p,trop} = 11000 \quad m \quad (D.13)$$

- la température dépend linéairement de l'altitude-pression (i.e. l'altitude que l'on observerait si l'atmosphère était ISA), avec le gradient suivant :

$$\frac{dT}{dH_p} = \begin{cases} -6.5 \text{ K/m} & \text{pour } H_p < H_{p,trop} \\ 0 \text{ K/m} & \text{pour } H_p \geq H_{p,trop} \end{cases} \quad (D.14)$$

Dans la suite du document, on notera β la constante numérique du gradient de température dans la troposphère :

$$\beta = -6.5 \text{ K/m} \quad (D.15)$$

D.2.5 Température et “température ISA”

Soit T la température en un point donné de l'atmosphère situé à une altitude-pression H_p . On notera T_{ISA} la température que l'on aurait constaté à la même altitude-pression (et donc à la même pression) si l'atmosphère avait été ISA.

Selon les hypothèses faites dans cette section, dans chaque couche d'atmosphère, la température est une fonction linéaire de l'altitude-pression, avec un gradient de température identique pour T et T_{ISA} . En conséquence, l'écart entre T et T_{ISA} reste constant quelle que soit l'altitude-pression H_p , à la verticale d'un point géographique donné. On notera ΔT cet écart.

$$T = T_{ISA} + \Delta T \quad (D.16)$$

En prenant pour référence dans la troposphère l'isobare $p = p_0$, les profils de température pour T et T_{ISA} s'expriment comme suit, l'altitude-pression de la tropopause étant donnée par (D.13) :

$$T = T_0 + \Delta T + \beta H_p \quad \text{pour } H_p < H_{p,trop} \quad (D.17a)$$

$$T_{trop} = T_0 + \Delta T + \beta H_{p,trop} \quad \text{à la tropopause} \quad (D.17b)$$

$$T = T_{trop} \quad \text{pour } H_p \geq H_{p,trop} \quad (D.17c)$$

$$T_{ISA} = T_0 + \beta H_p \quad \text{pour } H_p < H_{p,trop} \quad (D.18a)$$

$$T_{ISA,trop} = T_0 + \beta H_{p,trop} = 216,65 \text{ K} \quad \text{à la tropopause} \quad (D.18b)$$

$$T_{ISA} = T_{ISA,trop} \quad \text{pour } H_p \geq H_{p,trop} \quad (D.18c)$$

1. Notons que les deux notions se confondent pour l'atmosphère ISA

D.2.6 Caractéristiques des surfaces de référence

L'isobare $p = p_0 = 1013.25 \text{ hPa}$ (StdRef).

Par définition de l'altitude-pression, l'isobare *StdRef* est à l'altitude-pression $H_p = 0$. Sa température (en un point géographique donné) peut différer des conditions ISA d'un écart ΔT .

$$p_{StdRef} = p_0 \quad (\text{D.19a})$$

$$T_{StdRef} = T_0 + \Delta T \quad (\text{D.19b})$$

$$T_{ISA, StdRef} = T_0 \quad (\text{D.19c})$$

$$H_{p, StdRef} = 0 \quad (\text{D.19d})$$

$$(\text{D.19e})$$

Le niveau moyen de la mer (MSL).

Par définition, le niveau moyen de la mer est à l'altitude géopotentielle $H = 0$, et à l'altitude géodésique $h = 0$. La pression diffère d'un écart Δp de la pression p_0 en conditions ISA au niveau de la mer. L'altitude-pression du niveau moyen de la mer $H_{p,MSL}$ s'obtient simplement en remplaçant l'altitude géopotentielle H par l'altitude-pression H_p dans l'équation (D.9b) valable pour l'atmosphère ISA, et en l'appliquant à la troposphère en prenant pour base l'isobare $p = p_0$.

$$p_{MSL} = p_0 + \Delta p \quad (\text{D.20a})$$

$$T_{MSL} = T_0 + \Delta T + \beta H_{p,MSL} = T_{ISA,MSL} + \Delta T \quad (\text{D.20b})$$

$$T_{ISA,MSL} = T_0 + \beta H_{p,MSL} \quad (\text{D.20c})$$

$$H_{p,MSL} = \frac{T_0}{\beta} \left[\left(\frac{p_{MSL}}{p_0} \right)^{\frac{g_0}{\beta R}} - 1 \right] \quad (\text{D.20d})$$

$$H_{MSL} = 0 \quad (\text{D.20e})$$

D.2.7 Relation entre altitude géopotentielle et d'altitude-pression

En combinant (D.1) et (D.2), on voit qu'une variation de pression dp dans l'atmosphère non-ISA correspond à une variation d'altitude géopotentielle dH , selon l'équation (D.21) suivante.

$$dp = -\frac{p}{RT} g_0 dH \quad (\text{D.21})$$

La même variation de pression dans une atmosphère ISA correspondrait à une variation d'altitude-pression dH_p , satisfaisant l'équation (D.22).

$$dp = -\frac{p}{RT_{ISA}} g_0 dH_p \quad (\text{D.22})$$

En divisant l'expression (D.21) par (D.22), on obtient la relation entre variation d'altitude géopotentielle et variation d'altitude-pression, donnée par l'équation (D.23) suivante.

$$\frac{dH}{dH_p} = \frac{T}{T_{ISA}} \quad (\text{D.23})$$

La relation (D.24) entre H et H_p s'obtient en intégrant l'équation (D.23) en tenant compte de (D.17) et (D.18) pour l'expression des températures T et T_{ISA} .

$$H = H_p - H_{p,MSL} + \frac{\Delta T}{\beta} \ln \left(\frac{T_0 + \beta H_p}{T_{ISA,MSL}} \right) \quad \text{pour } H_p < H_{p,trop} \quad (\text{D.24a})$$

$$H_{trop} = H_{p,trop} - H_{p,MSL} + \frac{\Delta T}{\beta} \ln \left(\frac{T_{ISA,trop}}{T_{ISA,MSL}} \right) \quad (\text{D.24b})$$

$$H = H_{trop} + \frac{T_{ISA,trop}}{T_{ISA,MSL}} (H_p - H_{p,trop}) \quad \text{pour } H_p \geq H_{p,trop} \quad (\text{D.24c})$$

avec

$$H_{p,MSL} = \frac{T_0}{\beta} \left[\left(\frac{p_{MSL}}{p_0} \right)^{\frac{g_0}{\beta R}} - 1 \right] \quad (\text{D.25a})$$

$$H_{p,trop} = 11000 \text{ m} \quad (\text{D.25b})$$

$$T_{ISA,MSL} = T_0 + \beta H_{p,MSL} \quad (\text{D.25c})$$

$$T_{ISA,trop} = T_0 + \beta H_{p,trop} \quad (\text{D.25d})$$

$$(\text{D.25e})$$

D.2.8 Pression p en fonction de l'altitude-pression H_p

Les équations (D.8a) sont transposables directement, en remplaçant H par H_p et T par T_{ISA} . Pour la troposphère et la stratosphère, on obtient alors les expressions (D.26a) :

$$p = p_0 \left[\frac{T_0 + \beta H_p}{T_0} \right]^{-\frac{g_0}{R\beta}} \quad \text{pour } H_p < H_{p,trop} \quad (\text{D.26a})$$

$$p_{trop} = p_{ISA,trop} = p_0 \left[\frac{T_0 + \beta H_{p,trop}}{T_0} \right]^{-\frac{g_0}{R\beta}} \quad (\text{D.26b})$$

$$p = p_{ISA,trop} \exp \left[-\frac{g_0}{RT_{ISA,trop}} (H_p - H_{p,trop}) \right] \quad \text{pour } H_p \geq H_{p,trop} \quad (\text{D.26c})$$

D.2.9 Altitude-pression H_p en fonction de la pression p

$$H_p = \frac{T_0}{\beta} \left[\left(\frac{p}{p_0} \right)^{-\frac{R\beta}{g_0}} - 1 \right] \quad \text{pour } p \geq p_{ISA,trop} \quad (\text{D.27a})$$

$$H_p = H_{p,trop} - \frac{RT_{ISA,trop}}{g_0} \ln \left(\frac{p}{p_{ISA,trop}} \right) \quad \text{pour } p < p_{ISA,trop} \quad (\text{D.27b})$$

avec

$$p_{ISA,trop} = p_0 \left[\frac{T_{ISA,trop}}{T_0} \right]^{-\frac{g_0}{R\beta}} = p_0 \left[\frac{T_0 + \beta H_{p,trop}}{T_0} \right]^{-\frac{g_0}{R\beta}} \quad (\text{D.28})$$

D.2.10 Densité de l'air ρ en fonction de l'altitude-pressure H_p

$$\rho = \frac{p_0}{T_0 + \Delta T + \beta H_p} \left[\frac{T_0 + \beta H}{T_0} \right]^{-\frac{g_0}{R\beta}} \quad \text{pour } H_p < H_{p,\text{trop}} \quad (\text{D.29a})$$

$$\rho_{\text{trop}} = \frac{p_{ISA,\text{trop}}}{R(T_{ISA,\text{trop}} + \Delta T)} \quad (\text{D.29b})$$

$$\rho = \frac{p_{ISA,\text{trop}}}{R(T_{ISA,\text{trop}} + \Delta T)} \exp \left[-\frac{g_0}{RT_{ISA,\text{trop}}} (H_p - H_{p,\text{trop}}) \right] \quad \text{pour } H_p \geq H_{p,\text{trop}} \quad (\text{D.29c})$$

D.2.11 Vitesse du son

La vitesse du son dans l'atmosphère non-ISA modélisée ici est toujours donnée par $a = \sqrt{\kappa R T}$ (D.12), la température T étant donnée par (D.17).

Annexe E

Altimétrie

E.1 Moyens de mesure d'altitude à bord de l'avion

Il existe plusieurs façons de mesurer l'altitude à bord d'un avion :

- l'altimètre barométrique donne l'altitude-pression à partir d'une mesure de la pression statique extérieure.
- le radioaltimètre donne la hauteur par rapport au relief sous l'avion,
- le système de positionnement par satellites (GPS) fournit l'altitude géodésique par rapport au niveau moyen de la mer sur un ellipsoïde de révolution qui est une approximation du géoïde terrestre.

Il est à noter que ces trois types d'équipement mesurent des quantités fondamentalement différentes, souvent confondues sous un même terme générique d'"altitude" :

- une différence de pression par rapport à une isobare de référence, pour l'altitude-pression de l'altimètre, que l'on transforme en "*altitude-pression géopotentielle*" en faisant certaines hypothèses sur l'atmosphère terrestre (voir annexe D),
- une hauteur géométrique par rapport au relief du géoïde, pour le radioaltimètre,
- une hauteur géométrique par rapport au MSL (*Mean Sea Level*) de l'ellipsoïde.

Le radioaltimètre est en particulier utilisé pour les phases d'approche et d'atterrissage, pour la tenue d'une pente dans le plan vertical, ou pour fournir des alertes de proximité du sol.

Le GPS se base sur une approximation de la surface MSL, considérée comme un ellipsoïde de révolution. Cette approximation peut introduire une erreur entre -100 m et 70 m par rapport au géoïde terrestre (cf note OACI [4]). La mesure d'altitude du GPS est par ailleurs moins précise que le positionnement horizontal, avec une erreur typiquement trois fois plus importante. Dans des conditions correctes de couverture satellite, on peut s'attendre à une erreur de l'ordre de 30 à 50 mètres. Actuellement, le GPS est essentiellement utilisé comme source secondaire d'estimation de l'altitude, pour estimer l'erreur du système altimétrique de l'avion. Il n'est pas utilisé pour les besoins du contrôle du trafic aérien (ATC : *Air Traffic Control*).

E.2 Rappel des différentes notions d'altitude

On définit pour nos besoins trois altitudes distinctes :

- l’altitude géodésique, notée h . C’est la hauteur géométrique par rapport au niveau moyen de la mer sur l’ellipsoïde de révolution qui approxime la surface terrestre. Par extension, dans ce document, on utilisera la même définition si la Terre est considérée comme une sphère, ou approximée par un plan.
- l’altitude géopotentielle, notée H . C’est l’altitude résultant d’une mise à l’échelle pour compenser la variation de la gravité terrestre avec l’altitude. Elle est telle que $gdh = g_0 dH$, et $H_{MSL} = h_{MSL} = 0$
- l’altitude-pression géopotentielle, notée H_p que l’on raccourcira en *altitude-pression* dans ce document. Il s’agit de l’altitude géopotentielle que l’on obtiendrait à la même pression atmosphérique, si l’atmosphère était standard (ISA).

C’est cette dernière altitude (l’altitude-pression), qui est mesurée à bord de l’avion et transmise aux systèmes au sol pour les besoins ATC. Traditionnellement en aviation, cette altitude-pression est donnée en pieds (ft) ou en niveaux de vol (FL : *Flight Level*) correspondant à des incréments de 100 pieds.

E.3 L’altimètre barométrique

L’altimètre barométrique de l’avion se base sur une mesure de la pression atmosphérique statique pour estimer la hauteur géopotentielle par rapport à une surface isobare de référence. Le ou les altimètres situés dans le cockpit doivent donc être calés sur une pression de référence.

L’indication d’altitude-pression est par ailleurs transmise aux systèmes de contrôle au sol, soit suite à une interrogation par un radar secondaire (Mode-C ou Mode-S), soit par ADS-B. L’information d’altitude-pression transmise au sol de cette façon est toujours calculée avec la pression standard $p_0 = 1013,25 \text{ hPa}$ comme référence. Les systèmes au sol effectuent ensuite une éventuelle correction d’altitude, en prenant en compte les conditions de pression du jour. L’altitude-pression transmise par l’avion est donc calculée en utilisant l’équation (D.27).

E.4 Le calage altimétrique en aviation

On utilise plusieurs calages altimétriques en aviation :

- le QFE : l’altimètre est calé sur la pression atmosphérique mesurée sur l’aérodrome,
- le QNH : l’altimètre est calé sur la pression calculée du niveau moyen de la mer, en faisant l’hypothèse d’une atmosphère standard pour ce calcul.
- l’isobare $p_0 = 1013,25 \text{ hPa}$. L’avion vole alors “en niveaux de vol” (FL : *Flight Level*) au-dessus de l’isobare de référence. Par exemple, le FL350 correspond à 35000 pieds d’altitude-pression au-dessus de l’isobare $p = p_0$.

Annexe F

La vitesse de l'avion

F.1 Moyens de mesure

A bord, la vitesse de l'avion par rapport à la masse d'air est calculée à partir de mesures de la pression dynamique et de la pression atmosphérique statique. D'autres instruments comme le GPS apportent également des informations sur la vitesse de l'avion par rapport au sol.

Au sol, les systèmes de contrôle aérien calculent la vitesse horizontale de l'avion par rapport à la surface terrestre à partir des positions radar successives de l'avion. Le taux de montée ou de descente est calculé à partir des altitudes-pressions transmises par le transpondeur Mode-C ou Mode-S de l'avion.

F.2 Quelques définitions de la vitesse

TAS (*true airspeed*) : la vitesse vraie de l'avion par rapport à l'air dans lequel il évolue.

CAS (*calibrated airspeed*) : la vitesse-air que l'on aurait au niveau du sol avec la même pression dynamique que celle mesurée à bord de l'avion. C'est la vitesse indiquée par les instruments de bord de l'avion, corrigée des erreurs d'instruments. On parlera aussi de "vitesse corrigée", ou "vitesse calibrée".

Nombre de Mach : le rapport entre la TAS et la vitesse du son dans l'air.

Vitesse-sol (*ground speed*) : la vitesse horizontale de l'avion par rapport à la surface terrestre.

ROCD (*Rate of climb/descent*) : le taux de montée ou de descente. C'est la dérivée $\frac{dH_p}{dt}$ de l'altitude-pression par rapport au temps. Notons que le taux de montée ou de descente (ROCD) et la vitesse verticale $\frac{dh}{dt}$, où h est l'altitude géodésique, sont des quantités différentes.

Pour les formules de conversion entre les vitesses TAS, CAS, et nombre de Mach, le lecteur peut se reporter à la documentation du modèle BADA (Base of Aircraft Data) d'Eurocontrol ([128], ou versions ultérieures). La relation entre la vitesse-air et la vitesse-sol dépend quant à elle du vent (voir G.2.5 de l'annexe G).

F.3 Profils de vitesse

Pour la gestion de la vitesse, le mode d'opération de l'avion est basé sur les vitesses calculées par les instruments de bord : la vitesse CAS, ou le nombre de Mach. Typiquement, pour une montée standard en l'absence de contraintes particulières, l'avion suivra une phase d'accélération initiale jusqu'à atteindre une vitesse CAS_1 qu'il maintiendra constante jusqu'à une certaine altitude (typiquement le niveau $FL100$, soit 10000 pieds au-dessus de l'isobare $p = p_0$), au-delà de laquelle il volera à une vitesse CAS_2 jusqu'à l'altitude de conjonction CAS/Mach à partir de laquelle il volera à Mach constant. La formule de calcul de l'altitude de conjonction CAS/Mach est décrite en [128].

Le profil des vitesses-cibles pour ce type de montée dite à CAS ou Mach constant est illustré sur la figure (F.1). Lorsque l'avion vole à CAS constant ou à Mach constant en changeant d'altitude, la vitesse-air TAS évolue en fonction de l'altitude-pressure, comme illustré sur cette figure.

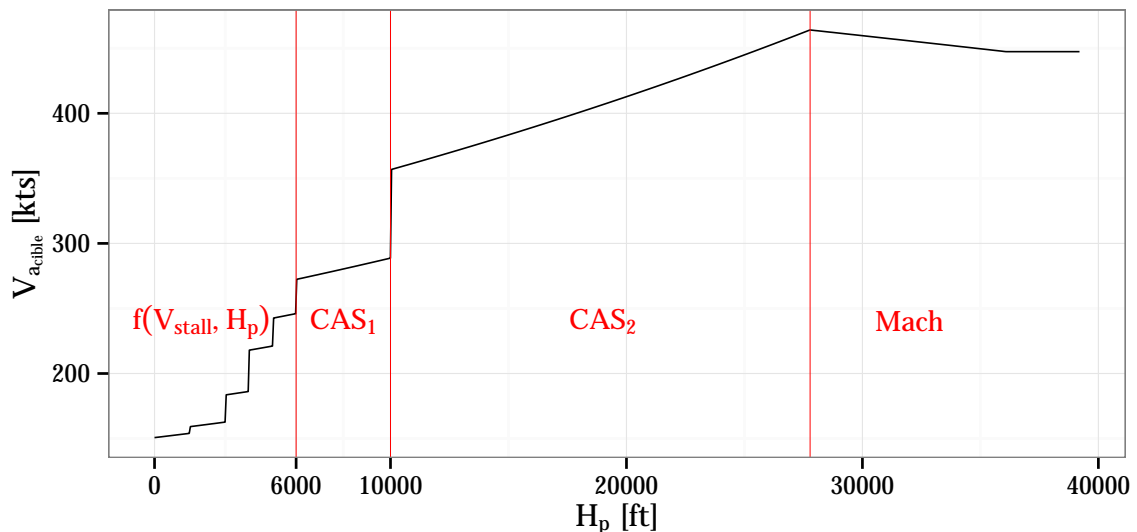


FIGURE F.1: Profil typique pour les segments de vitesse en montée. Le schéma donne l'évolution de la vitesse-air TAS (ici notée V_a) correspondant aux différents segments de vitesses-cibles à CAS constant ou Mach constant donnés par le modèle BADA. Les phases d'accélération ou de décélération entre ces segments discontinus ne sont pas ici représentées.

La vitesse CAS_1 correspond à une limitation de vitesse (généralement 250 nœuds en Europe) mise en place en-dessous d'une certaine altitude (FL100). Le choix des vitesses CAS_2 , et $Mach$ dépend pour chaque avion des choix fait par la compagnie aérienne en matière d'équilibre entre le coût attribué aux retards (ou plus généralement au temps de voyage) et le coût d'opération de l'avion, notamment en carburant. Ce ratio, appelé *cost index*, est propre à chaque compagnie. C'est un des éléments-clés, avec le taux de remplissage, de la stratégie commerciale des compagnies aériennes. Il n'est donc pas divulgué à l'extérieur de la compagnie.

La croisière de l'avion se fait à la vitesse (CAS ou Mach) correspondant à l'optimum du *cost index* choisi par la compagnie. La descente vers l'aéroport se fait à partir du *top of*

Annexe G

Équations du modèle avion

G.1 Notations

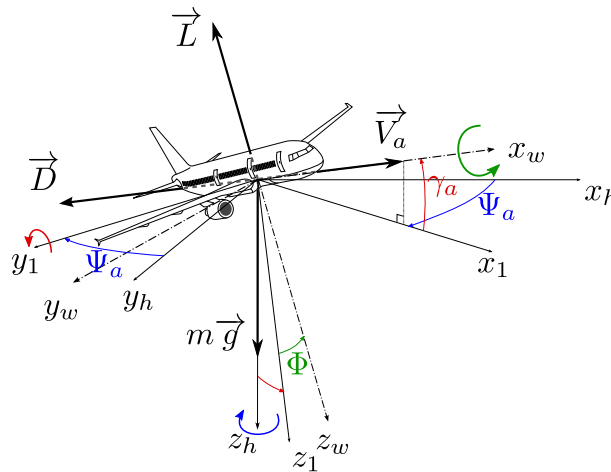


FIGURE G.1: Rotations et angles d'Euler (Ψ_a, γ_a, Φ), orientés dans le sens trigonométrique, et avec des repères orthonormés directs.

\vec{A}	Force aérodynamique, qu'on décompose usuellement en trois forces : la traînée \vec{D} , dans la direction du vent relatif subi par l'avion (c.à.d. dans la direction opposée à la vitesse air \vec{V}_a), la portance \vec{L} , orthogonale à \vec{D} et dans le plan de symétrie de l'avion, et \vec{C} , la composante latérale. Si P est le point où s'applique la résultante des forces aérodynamiques, les coordonnées de \vec{A} dans $Px_w y_w z_w$ sont donc simplement $(-D, -C, -L)$.
c	Commande de poussée des moteurs.
C	Composante latérale de la force aérodynamique \vec{A} .
CAS	Vitesse calibrée (<i>Calibrated Airspeed</i>)
C_D	Coefficient de traînée aérodynamique.
C_L	Coefficient de portance aérodynamique.
D	Traînée aérodynamique (<i>drag</i>), i.e. composante longitudinale de la force aérodynamique.

ESF	<i>Energy share factor</i> . Facteur de répartition entre variation d'énergie cinétique et variation d'énergie potentielle. Typiquement, lors d'une montée, ce facteur fixe la répartition de la puissance disponible entre la montée proprement dite et l'accélération longitudinale de l'avion.
g	Accélération due à la gravitation terrestre.
g_0	Accélération gravitationnelle au niveau moyen de la mer.
h	Altitude géodésique.
H	Altitude géopotentielle.
H_p	Altitude pression géopotentielle, ou simplement altitude pression. C'est l'altitude géopotentielle que l'on aurait à la même pression atmosphérique dans une atmosphère standard ISA.
ISA	Atmosphère standard internationale (<i>International Standard Atmosphere</i>).
L	Portance aérodynamique (<i>Lift</i>).
m	Masse de l'avion.
M	Nombre de Mach.
MSL	Niveau moyen de la mer (<i>Mean Sea Level</i>)
p	Pression atmosphérique
p_{MSL}	Pression atmosphérique au niveau moyen de la mer (MSL)
S	Aire des surfaces portantes de l'avion.
T	Température de l'air.
T_{ISA}	Température de l'air que l'on aurait à la même pression dans une atmosphère standard ISA.
Thr	Poussée des moteurs (<i>thrust</i>)
\vec{U}	Vecteur de commande de la trajectoire. Typiquement, $\vec{U} = [Thr, \gamma_a, \Phi]$, ou bien $\vec{U} = [c, \gamma_a, \Phi]$ si la poussée $Thr = Thr(c, V_a, H_p, T)$ est elle-même fonction d'une commande c (manette des gaz).
\vec{V}_a	Vitesse de l'avion relativement à la masse d'air. Son module V_a est noté TAS (<i>true airspeed</i>) dans la littérature.
$\vec{V}_{a/x_h y_h}$	Projection de la vitesse air dans le plan horizontal local.
\vec{V}_g	Vitesse horizontale de l'avion par rapport au sol. V_g est parfois notée GS (<i>Ground speed</i>) dans la littérature. La direction du vecteur \vec{V}_g dans le plan horizontal local $x_h y_h$ est donnée par l'angle Ψ_i . Notons que \vec{V}_g est la projection du vecteur vitesse \vec{V}_i dans le plan $x_h y_h$.
\vec{V}_i	Vitesse de l'avion. \vec{V}_i est le vecteur vitesse dans un référentiel inertiel. Notons que, pour tout ce qui concerne l'application du principe fondamental de la dynamique ($m \vec{a} = \sum \vec{F}$), on considère comme fixe (inertiel) le repère horizontal local $x_h y_h z_h$ muni pour origine d'un point fixe par rapport à la Terre ¹ .

1. Pour la commodité des calculs, on pourra prendre le point fixe par rapport à la Terre, confondu avec la position du centre de gravité de l'avion.

\vec{W}	Le vecteur vent (<i>wind</i>). En météorologie, les coordonnées du vecteur vent en un point P sont données dans un repère xyz où x pointe vers l'Est, y vers le Nord, et z vers le haut. Pour éviter toute confusion avec le repère $x_h y_h z_h$ d'orientation différente employé dans ce document, nous utiliserons des noms explicites pour les trois composantes du vent ($W_{North}, W_{East}, W_{Up}$), respectivement vers le Nord, l'Est, ou le haut. Les calculs intermédiaires pourront toutefois faire appel aux composantes du vent dans un repère choisi, par exemple $(W_{x_h}, W_{y_h}, W_{z_h})$
x, y	Coordonnées cartésiennes dans un plan horizontal.
\vec{X}	Vecteur d'état de l'avion. Par exemple, $\vec{X} = [\lambda, \mu, h, V_a, \Psi_a, m]$ si la position est repérée en longitude et latitude, ou bien $\vec{X} = [x, y, h, V_a, \Psi_a, m]$ si la position est repérée en coordonnées cartésiennes dans un plan.
$x_w y_w z_w$	Repère lié à la vitesse air \vec{V}_a . L'axe x_w est dans la direction de \vec{V}_a . Notons que c'est la direction opposée au vent relatif subi par l'avion, ce qui justifie l'appellation anglaise <i>wind-axis system</i> . L'axe z_w est dans le plan de symétrie de l'avion, orienté vers le bas ² , et y_w est orienté vers l'aile droite de façon à former un repère orthonormé direct (voir Figure G.4).
$x_b y_b z_b$	Repère fixé à la structure de l'avion (<i>body-axis system</i>). L'axe x_b est fixé au fuselage et pointe vers l'avant. Ici, on choisit x_b tel que si l'angle entre x_b et la direction du vol dans le flux d'air est nul, alors la portance est nulle ³ . L'axe z_b est orthogonal à x_b , dans le plan de symétrie de l'avion, et pointant vers le bas. L'axe y_b , orienté vers l'aile droite, vient compléter le repère orthonormé direct.
$x_h y_h z_h$	Repère horizontal local (ou repère topographique). L'axe z_h est dans la direction verticale et pointe <i>vers le bas</i> . Le plan $x_h y_h$ est le plan horizontal local, à la position courante de l'avion. L'axe x_h pointe vers le Nord géographique, et y_h vers l'Est (voir Figure G.5).
$x_g y_g z_h$	Repère lié à la vitesse sol horizontale \vec{V}_g . On passe de $x_h y_h z_h$ à $x_g y_g z_h$ par une rotation d'angle Ψ_i autour de z_h (voir Figure G.8).
λ	Longitude géodésique.
μ	Latitude géodésique.
γ_a	Pente de la trajectoire dans la masse d'air. C'est l'angle entre le plan horizontal local et le vecteur vitesse air \vec{V}_a .
γ_i	Pente de la trajectoire par rapport à l'horizontale. C'est l'angle entre \vec{V}_g (dans le plan horizontal) et \vec{V}_i .
θ_d	Angle de la dérive due au vent. $\theta_d = \Psi_i - \Psi_a$
Φ	Angle d'inclinaison de l'avion (<i>bank angle</i> en anglais)
Ψ_a	Direction de la vitesse air \vec{V}_a .
Ψ_i	Direction de la route vraie suivie par l'avion (i.e. direction du vecteur vitesse \vec{V}_i).
Ψ_W	Direction du vecteur vent \vec{W} .
ρ	Densité de l'air.

2. On suppose que l'avion n'est pas en vol-dos.

3. Si on choisit un autre axe que celui correspondant à la portance nulle, comme par exemple l'axe longitudinal du fuselage, il faut éventuellement rajouter un angle fixe de calage dans les équations.

- Δp Écart de pression au niveau moyen de la mer (MSL) par rapport à la pression atmosphérique standard en conditions ISA ($p_0 = 1013.25 \text{ hPa}$). On a $p_{MSL} = p_0 + \Delta p$.
- ΔT Écart entre la température à l'isobare $p_0 = 1013.25$ et la température que l'on aurait à la même pression, dans une atmosphère ISA. Notons qu'avec le modèle d'atmosphère non-ISA utilisé pour l'altimétrie, cet écart reste constant quelle que soit l'altitude-pression H_p (voir annexe D). On a donc $T = T_{ISA} + \Delta T$, quelle que soit l'altitude-pression⁴.

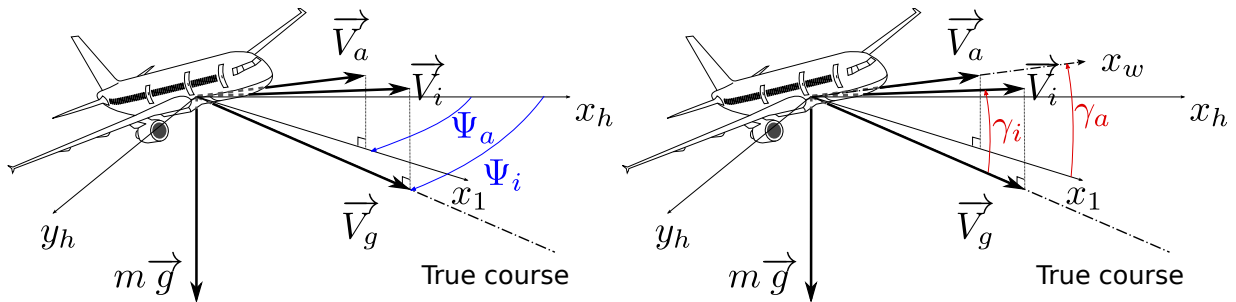


FIGURE G.2: Angles Ψ et γ (avec x_h pointant vers le Nord).

G.2 La dynamique du vol

Le terme de “dynamique du vol” s’entend dans cette section comme l’expression de l’accélération en fonction des forces s’appliquant à l’avion. Les modèles utilisés dans la très grande majorité des prédicteurs-sol ne modélisent pas la rotation de l’avion autour de son centre d’inertie ni l’effet des commandes de vol sur les surfaces portantes. On suppose également que le vol est symétrique et les virages coordonnés (pas de “dérapage” latéral de l’avion), et que la résultante des forces aérodynamiques s’applique au centre de gravité de l’avion.

G.2.1 Le modèle point-masse

Toute la masse de l’avion est ramenée en un point, son centre d’inertie, auquel s’applique les forces aérodynamiques, l’attraction terrestre, et la poussée des moteurs, comme illustré sur la Figure G.3. La deuxième loi de Newton $m\vec{a}_i = \sum \vec{F}$ se traduit alors par l’équation (G.1), où le point au-dessus d’une quantité dénote sa dérivée par rapport au temps, et l’indice i dénote que la grandeur est calculée dans un référentiel inertiel.

$$m\dot{\vec{V}}_i = \overrightarrow{Thr} + \vec{A} + m\vec{g} \quad (\text{G.1})$$

La force aérodynamique \vec{A} est usuellement décrite comme la somme de trois composantes : la traînée \vec{D} (*drag*), qui est parallèle à la vitesse air \vec{V}_a (*true airspeed*) et dans la direction

4. Si l’on se place sur l’isobare de référence $p_0 = 1013.25$ où l’altitude-pression H_p est par définition égale à 0, on a $T_{H_p=0} = T_0 + \Delta T$. Si on se place au niveau de la mer MSL, qui ne coïncide généralement pas avec l’isobare $p = p_0$, on a $T_{MSL} = T_{ISA, MSL} + \Delta T = T_0 + \beta H_{p,MSL} + \Delta T$ où β est le gradient de température dans la troposphère.

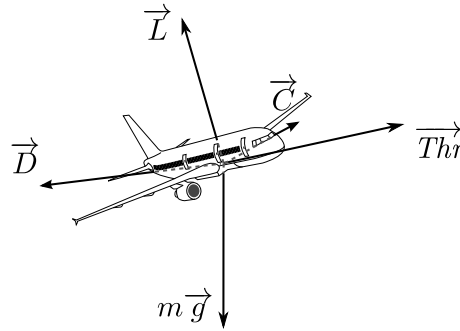


FIGURE G.3: Forces s'appliquant au centre d'inertie de l'avion.

opposée, la portance \vec{L} (*lift*), perpendiculaire à la traînée et dans le plan de symétrie de l'avion, et une force latérale \vec{C} , orthogonale à \vec{L} et \vec{D} . Avec cette décomposition, l'équation (G.1) devient l'équation (G.2).

$$m\vec{\dot{V}}_i = \vec{Thr} + \vec{D} + \vec{L} + m\vec{g} \quad (\text{G.2})$$

Dans les équations (G.1) et (G.2), la variation de la masse de l'avion au cours du temps est incluse dans la force de poussée. Avec les systèmes de propulsion équipant les avions, l'influence de la variation de masse $\dot{m} = \frac{dm}{dt}$ sur la dynamique du vol est de toute façon négligeable : on n'est pas dans le cas d'une fusée où l'éjection de masse participe à l'accélération.

Pour ce qui concerne les équations complètes du mouvement de l'avion, la consommation de carburant doit cependant être prise en compte, au moment où on intègre les équations pour prédire la trajectoire future. La consommation de carburant pour les jets et les turboprops, pour les phases de montée et de croisière, est de la forme décrite dans l'équation (G.3). Dans cette équation, $\eta = \eta(V_a)$ est la consommation spécifique, par unité de poussée, de carburant. Elle dépend de la vitesse air.

$$\dot{m} = -\eta Thr \quad (\text{G.3})$$

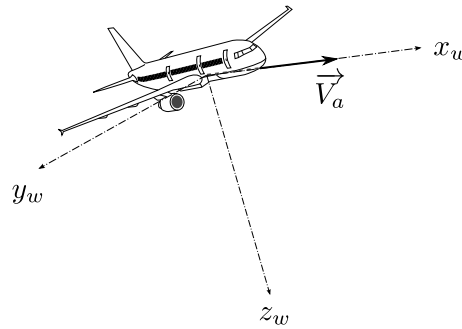
Pour les engins à pistons, la consommation de carburant est donnée par des constantes, différentes pour chaque phase de vol.

G.2.2 Référentiels, et approximation de la terre plate

La vitesse air pointe dans la direction dont provient le vent relatif que subit l'avion. Le vecteur vitesse air détermine un axe x_w , qui lorsqu'on le complète en un référentiel orthogonal $x_w y_w z_w$ dont l'axe z_w pointe vers le dessous de l'avion, permet d'écrire la force aérodynamique sous la forme donnée dans l'équation G.4.

$$\vec{A} = \begin{pmatrix} -D \\ -C \\ -L \end{pmatrix}_{x_w y_w z_w} \quad (\text{G.4})$$

Ce "référentiel air" (*wind-axis system*) n'est pas fixé à la structure de l'avion : la direction de la vitesse air peut varier selon l'orientation de la structure par rapport au déplacement.

FIGURE G.4: le référentiel “air” (*wind-axis system*).

La force de poussée s'exerce quant à elle toujours dans une direction fixée par rapport au fuselage de l'avion. On définit donc un autre référentiel $x_b y_b z_b$, fixe par rapport au fuselage. Le poids s'exerce dans la direction du centre de gravité de la terre, et on définit un référentiel $x_h y_h z_h$ tel que z_h pointe vers ce centre, x_h vers le Nord, et y_h vers l'Est, comme illustré sur la figure G.5.

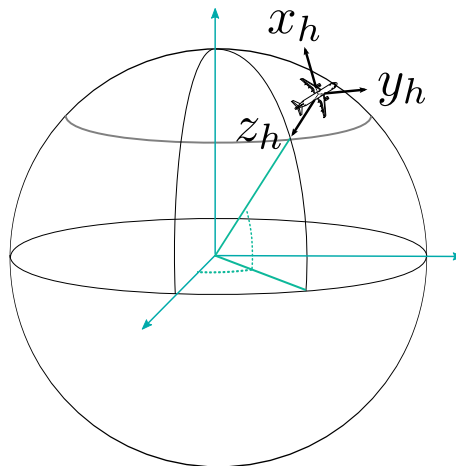


FIGURE G.5: Le référentiel horizontal local.

Étant donné l'ordre de grandeur des vitesses pour les vols commerciaux, on considère que l'accélération de Coriolis due à la rotation de la terre peut être négligée pour tout ce qui concerne le calcul de l'accélération. Cette approximation est dite “approximation de la Terre plate”. Pour exprimer sous forme scalaire les équations de la dynamique du vol, on considérera comme fixe le référentiel constitué du repère $x_h y_h z_h$ et ayant pour origine un point fixé par rapport à la Terre (éventuellement confondu avec la position du centre de masse de l'avion pour simplifier les calculs).

Afin d'exprimer l'équation vectorielle (G.2) sous forme d'équations scalaires, il est également utile de plonger la seconde loi de Newton dans un système de coordonnées attaché à la vitesse-sol de l'avion \vec{V}_g . Cette vitesse est la projection dans le plan horizontal local de la vitesse inertielle \vec{V}_i . On note $x_g y_g z_h$ ce système de coordonnées.

G.2.3 Changements de référentiel

Supposons que l'on connaisse les coordonnées d'un vecteur \vec{v} dans un référentiel F_1 d'axes $x_1y_1z_1$, et que l'on cherche la transformation $T_{2,1}$ à appliquer à ces coordonnées pour obtenir les coordonnées du vecteur \vec{v} dans un autre référentiel F_2 , d'axes $x_2y_2z_2$.

$$\begin{pmatrix} v_{x_2} \\ v_{y_2} \\ v_{z_2} \end{pmatrix} = T_{1 \rightarrow 2} \begin{pmatrix} v_{x_1} \\ v_{y_1} \\ v_{z_1} \end{pmatrix} \quad (\text{G.5})$$

On peut envisager plusieurs approches pour calculer $T_{1 \rightarrow 2}$: un calcul direct de la rotation d'un référentiel à l'autre en exprimant les coordonnées des vecteurs de base de F_1 dans le nouveau système F_2 , ou une succession de trois rotations suivant les angles d'Euler, ou encore une approche par quaternions. Cette dernière est l'approche préférentielle dans les applications d'asservissement de trajectoire ou dans les simulateurs de vol, car elle évite certaines singularités apparaissant dans les équations obtenues avec les angles d'Euler.

Toutefois, les prédicteurs de trajectoires utilisés dans ce domaine du contrôle du trafic aérien sont très peu susceptibles de rencontrer de telles singularités. On utilise donc généralement les trois rotations successives autour des axes, avec les angles d'Euler. L'ordre des axes peut être choisi arbitrairement, mais on commence usuellement par une rotation R_3 d'angle θ_z autour du troisième axe z_1 , qui transforme le système de départ en un système $x'y'z'$, avec $z' = z_1$. La matrice de cette rotation est donnée par l'équation (G.6).

$$R_3(\theta_z) = \begin{pmatrix} \cos(-\theta_z) & \cos(\frac{\pi}{2} - \theta_z) & \cos(-\frac{\pi}{2}) \\ \cos(\frac{\pi}{2} + \theta_z) & \cos(-\theta_z) & \cos(\frac{\pi}{2}) \\ \cos(\frac{\pi}{2}) & \cos(-\frac{\pi}{2}) & \cos(0) \end{pmatrix} = \begin{pmatrix} \cos \theta_z & \sin \theta_z & 0 \\ -\sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{G.6})$$

La deuxième rotation R_2 d'angle θ_y autour de l'axe y' nous donne un système d'axes $x''y''z''$, avec $x'' = x_2$ et $y'' = y'$. La matrice de cette rotation est donnée par l'équation (G.7).

$$R_2(\theta_y) = \begin{pmatrix} \cos(-\theta_y) & \cos(\frac{\pi}{2}) & \cos(\frac{\pi}{2} + \theta_y) \\ \cos(-\frac{\pi}{2}) & \cos(0) & \cos(-\frac{\pi}{2}) \\ \cos(\frac{\pi}{2} - \theta_y) & \cos(\frac{\pi}{2}) & \cos(-\theta_y) \end{pmatrix} = \begin{pmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{pmatrix} \quad (\text{G.7})$$

La dernière rotation, R_1 d'angle θ_x autour de l'axe x_2 permet de passer de $x_2y'z''$ à $x_2y_2z_2$. Sa matrice est donnée par l'équation (G.8).

$$R_1(\theta_x) = \begin{pmatrix} \cos(0) & \cos(\frac{\pi}{2}) & \cos(-\frac{\pi}{2}) \\ \cos(-\frac{\pi}{2}) & \cos(-\theta_x) & \cos(\frac{\pi}{2} - \theta_x) \\ \cos(\frac{\pi}{2}) & \cos(\frac{\pi}{2} + \theta_x) & \cos(-\theta_x) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & \sin \theta_x \\ 0 & -\sin \theta_x & \cos \theta_x \end{pmatrix} \quad (\text{G.8})$$

La transformation $T_{1 \rightarrow 2}$ s'exprime finalement sous forme d'une multiplication de matrices $T_{1 \rightarrow 2} = R_1(\theta_x)R_2(\theta_y)R_3(\theta_z)$, décrite par l'équation (G.9).

$$T_{1 \rightarrow 2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & \sin \theta_x \\ 0 & -\sin \theta_x & \cos \theta_x \end{pmatrix} \begin{pmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{pmatrix} \begin{pmatrix} \cos \theta_z & \sin \theta_z & 0 \\ -\sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{G.9})$$

En développant l'équation (G.9), on obtient finalement l'équation (G.10).

$$T_{1 \rightarrow 2} = \begin{pmatrix} \cos \theta_y \cos \theta_z & \cos \theta_y \sin \theta_z & -\sin \theta_y \\ \begin{pmatrix} \sin \theta_x \sin \theta_y \cos \theta_z \\ -\cos \theta_x \sin \theta_z \end{pmatrix} & \begin{pmatrix} \sin \theta_x \sin \theta_y \sin \theta_z \\ +\cos \theta_x \cos \theta_z \end{pmatrix} & \sin \theta_x \cos \theta_y \\ \begin{pmatrix} \cos \theta_x \sin \theta_y \cos \theta_z \\ +\sin \theta_x \sin \theta_z \end{pmatrix} & \begin{pmatrix} \cos \theta_x \sin \theta_y \sin \theta_z \\ -\sin \theta_x \cos \theta_z \end{pmatrix} & \cos \theta_x \cos \theta_y \end{pmatrix} \quad (\text{G.10})$$

G.2.4 Définition de quelques angles

Les angles aérodynamiques

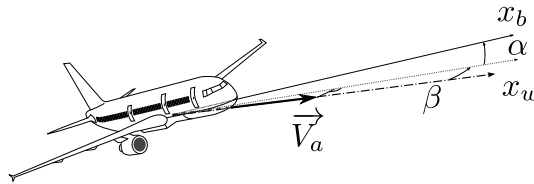


FIGURE G.6: Vue 3D des angles aérodynamiques α et β .

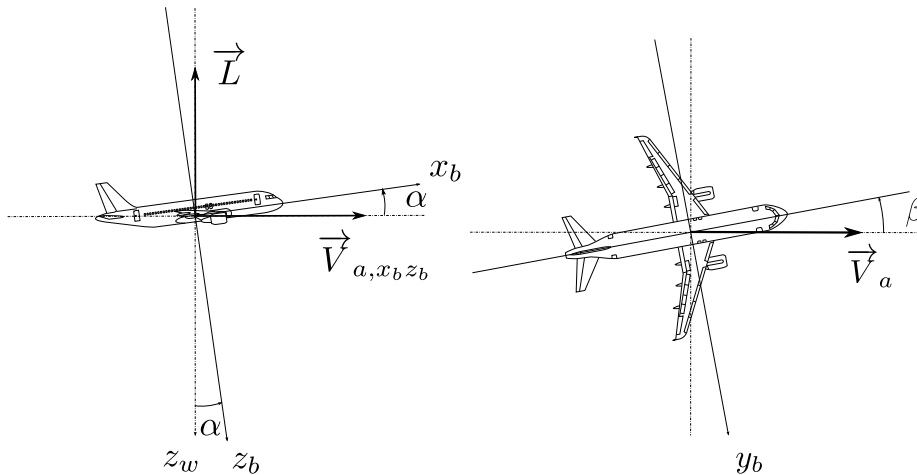


FIGURE G.7: Angle d'attaque α , et angle de dérapage β .

L'orientation de l'avion relativement au système d'axes $x_w y_w z_w$ associé à la vitesse air est définie par deux angles : l'angle d'attaque α et l'angle de dérapage β .

En considérant les deux systèmes d'axes $x_w y_w z_w$, attaché à la vitesse air \vec{V}_a , et $x_b y_b z_b$, attaché au fuselage de l'avion, l'angle d'attaque α est l'angle entre la projection de \vec{V}_a sur le plan de symétrie $x_b z_b$ et l'axe x_b . Comme z_w et z_b sont tous les deux dans le plan de symétrie, α est simplement l'angle entre z_w et z_b . Lorsqu'on modélise la portance en fonction de l'angle d'attaque α , il est pratique de choisir l'axe x_b , fixé au fuselage, tel que la portance soit nulle pour $\alpha = 0$. Avec cette convention, l'angle d'attaque est positif quand le vent relatif vient de dessous le plan $x_b y_b$.

L'angle de dérapage β est l'angle entre la vitesse air \vec{V}_a et le plan de symétrie de l'avion, mesuré dans le plan $x_w y_w$. Il est positif quand le vent relatif provient de la droite du plan de symétrie.

Angles dans le plan horizontal

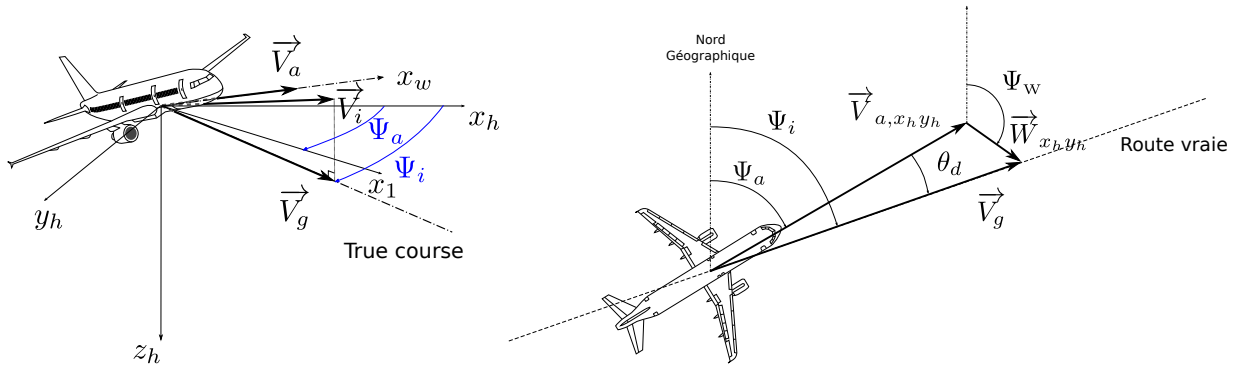


FIGURE G.8: Angles dans le plan horizontal.

Par convention, dans le milieu aéronautique, la direction du vecteur vitesse air ou de la vitesse sol de l'avion est comptée positivement dans le sens horaire, avec le zéro dans la direction du Nord géographique. Comme tous les autres angles dans ce document utilisent la convention trigonométrique classique, il est pratique de considérer le référentiel $x_h y_h z_h$ où x_h pointe vers le Nord, y_h vers l'Est, et z_h vers le bas. Avec cette convention, l'angle d'Euler Ψ mesurant la rotation autour de z_h (pointant vers le bas) nous donne directement l'angle de la direction de l'avion dans le plan horizontal avec la convention aéronautique de sens horaire.

On note Ψ_a l'angle mesurant la direction de la projection de \vec{V}_a sur le plan horizontal $x_h y_h$, et Ψ_i la direction du vecteur vitesse sol \vec{V}_g (c'est-à-dire la projection de la vitesse inertielle \vec{V}_i sur le plan horizontal).

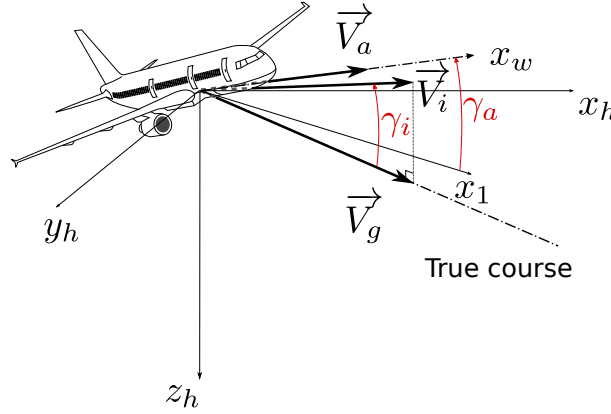
L'angle de dérive θ_d (*drift angle*) est alors défini comme l'angle entre la vitesse air horizontale et la vitesse sol : $\theta_d = \Psi_i - \Psi_a$. Ces angles sont illustrés sur la Figure G.8

Angles dans le plan vertical

La pente de la trajectoire de l'avion est mesurée par un angle γ , avec un indice différent selon que l'on considère la pente "air" ou la pente "sol". On note donc γ_a l'angle d'élévation du vecteur vitesse air \vec{V}_a au-dessus du plan horizontal, et γ_i l'angle entre \vec{V}_g et \vec{V}_i . Ces deux angles sont illustrés sur la Figure G.9.

G.2.5 Influence du vent

L'effet du vent sur la dynamique du vol ne doit pas être négligé, et est pris en compte d'une part dans l'équation (G.11a) pour le calcul des vitesses, et d'autre part dans l'équation (G.11b) pour ce qui concerne la dynamique du vol (l'expression de l'accélération en fonction des forces).

FIGURE G.9: Pente “air” γ_a et pente “sol” γ_i .

$$\vec{V}_i = \vec{V}_a + \vec{W} \quad (\text{G.11a})$$

$$\vec{a}_i = \dot{\vec{V}}_i = \dot{\vec{V}}_a + \dot{\vec{W}} \quad (\text{G.11b})$$

Le triangle des vitesses

L'équation (G.11a) est communément appelée “triangle des vitesses”, ou “triangle du vent” (*wind triangle*) chez les anglo-saxons. En projetant cette équation $\vec{V}_i = \vec{V}_a + \vec{W}$ sur un axe vertical orienté vers le haut, on obtient l'équation (G.12) :

$$\dot{h} = V_a \sin \gamma_a - W_{z_h} = V_a \sin \gamma_a + W_{Up} \quad (\text{G.12})$$

La projection du triangle des vitesses dans le plan horizontal local $Px_h y_h$ donne le système (G.13) :

$$\begin{cases} V_g \cos \Psi_i = V_a \cos \gamma_a \cos \Psi_a + W_{North} \\ V_g \sin \Psi_i = V_a \cos \gamma_a \sin \Psi_a + W_{East} \end{cases} \quad (\text{G.13})$$

Calcul de la dérive

En notant W_{\perp} la composante de vent horizontale perpendiculaire à la route vraie de l'avion, on remarque en observant la figure G.10 que :

$$W_{\perp} = V_a \cos \gamma_a \sin \theta_d = W_{x_h y_h} \sin(\Psi_w - \Psi_i) \quad (\text{G.14})$$

En projetant la vitesse-air et le vent sur la route vraie, on obtient par ailleurs l'égalité suivante :

$$\begin{aligned} V_g &= V_a \cos \gamma_a \cos \theta_d + W_{x_h y_h} \cos(\Psi_w - \Psi_i) \\ &= V_a \cos \gamma_a \cos \theta_d + W_{x_h y_h} (\cos \Psi_w \cos \Psi_i + \sin \Psi_w \sin \Psi_i) \\ &= V_a \cos \gamma_a \cos \theta_d + W_{North} \cos \Psi_i + W_{East} \sin \Psi_i \end{aligned} \quad (\text{G.15})$$

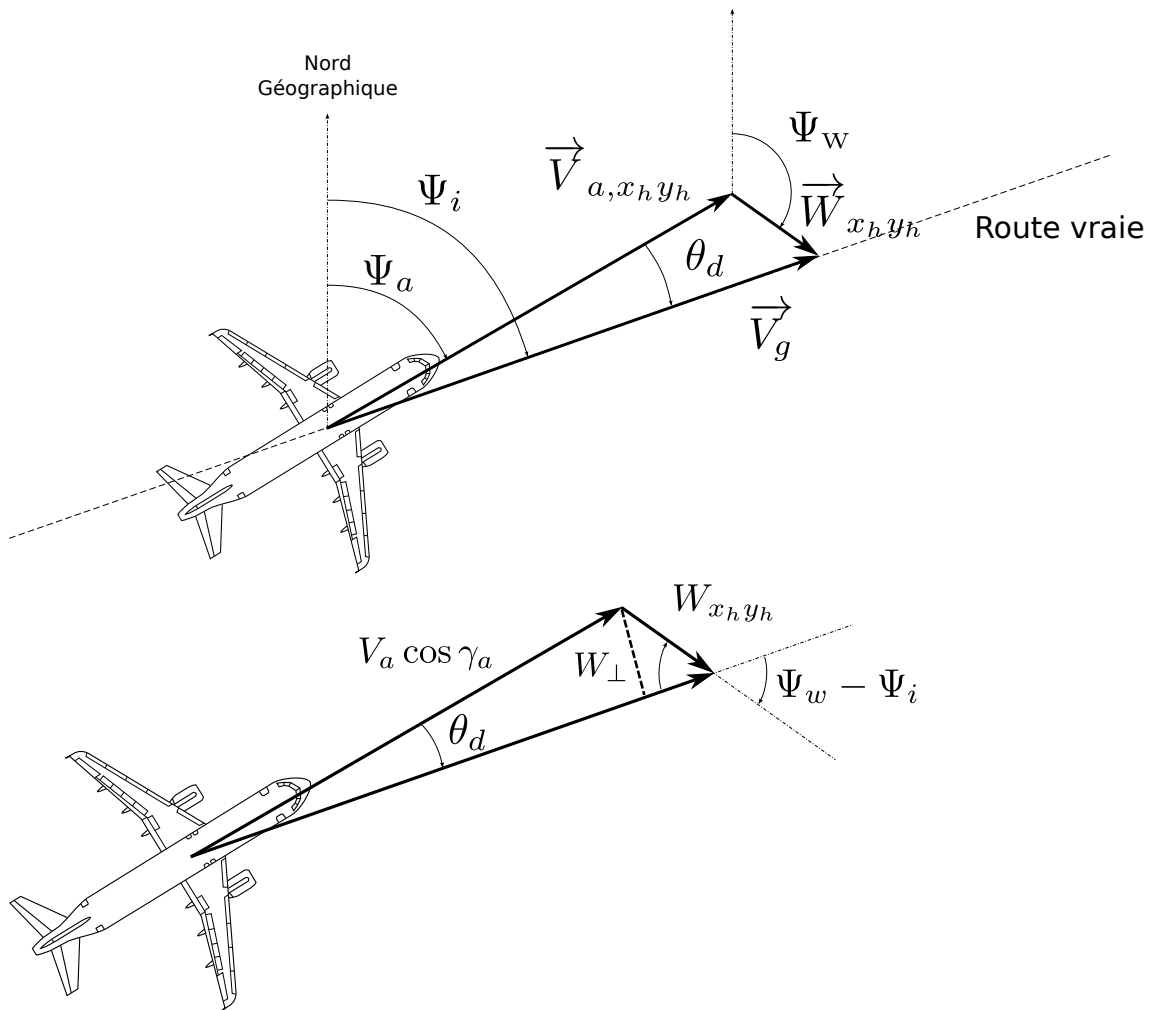


FIGURE G.10: Le triangle des vitesses.

On en déduit l'expression de $\sin \theta_d$ et $\cos \theta_d$ en fonction des autres grandeurs :

$$\begin{aligned} \sin \theta_d &= \frac{W_{x_h y_h} \sin(\Psi_w - \Psi_i)}{V_a \cos \gamma_a} = \frac{W_{x_h y_h}}{V_a \cos \gamma_a} (\sin \Psi_w \cos \Psi_i - \cos \Psi_w \sin \Psi_i) \\ &= \frac{W_{East} \cos \Psi_i - W_{North} \sin \Psi_i}{V_a \cos \gamma_a} \end{aligned} \quad (\text{G.16})$$

$$\cos \theta_d = \frac{V_g - W_{North} \cos \Psi_i - W_{East} \sin \Psi_i}{V_a \cos \gamma_a} \quad (\text{G.17})$$

Expression de la dérivée du vent

Soit \vec{W} le champ de vent dans l'atmosphère terrestre. Un avion situé à la position P au temps t subira un vent $\vec{W} = \vec{W}(P, t)$. Le calcul de la dérivée $\dot{\vec{W}} = \frac{d}{dt} \vec{W}$ fait donc intervenir le déplacement de l'avion dans le champ de vent, en plus de l'éventuelle variation de ce champ au cours du temps.

En se plaçant dans un référentiel horizontal local dont les axes sont $x_h y_h z_h$ et en considérant celui-ci comme fixe⁵, les composantes de la dérivée du vent sont alors données par l'équation (G.18).

$$\begin{cases} \dot{W}_{x_h} = \dot{x} \frac{\partial W_{x_h}}{\partial x} + \dot{y} \frac{\partial W_{x_h}}{\partial y} + \dot{z} \frac{\partial W_{x_h}}{\partial z} + \frac{\partial W_{x_h}}{\partial t} \\ \dot{W}_{y_h} = \dot{x} \frac{\partial W_{y_h}}{\partial x} + \dot{y} \frac{\partial W_{y_h}}{\partial y} + \dot{z} \frac{\partial W_{y_h}}{\partial z} + \frac{\partial W_{y_h}}{\partial t} \\ \dot{W}_{z_h} = \dot{x} \frac{\partial W_{z_h}}{\partial x} + \dot{y} \frac{\partial W_{z_h}}{\partial y} + \dot{z} \frac{\partial W_{z_h}}{\partial z} + \frac{\partial W_{z_h}}{\partial t} \end{cases} \quad (\text{G.18})$$

En introduisant le gradient du champ vectoriel de vent $\nabla \vec{W} = \mathbf{grad} \vec{W}$ comme étant la jacobienne des dérivées partielles par rapport aux coordonnées spatiales, on peut exprimer le système d'équations (G.18) sous forme d'une équation vectorielle (G.19).

$$\frac{d}{dt} \vec{W}(P, t) = (\nabla \vec{W}) \vec{V}_i + \frac{\partial \vec{W}}{\partial t} \quad (\text{G.19})$$

où

$$\nabla \vec{W} = \mathbf{grad} \vec{W} = \begin{pmatrix} \frac{\partial W_{x_h}}{\partial x} & \frac{\partial W_{x_h}}{\partial y} & \frac{\partial W_{x_h}}{\partial z} \\ \frac{\partial W_{y_h}}{\partial x} & \frac{\partial W_{y_h}}{\partial y} & \frac{\partial W_{y_h}}{\partial z} \\ \frac{\partial W_{z_h}}{\partial x} & \frac{\partial W_{z_h}}{\partial y} & \frac{\partial W_{z_h}}{\partial z} \end{pmatrix}$$

En exprimant la vitesse inertielle \vec{V}_i comme la somme du vecteur vitesse-sol \vec{V}_g dans le plan horizontal et de la vitesse verticale \vec{V}_z , on obtient (G.20). L'avantage de cette formulation

5. Rappelons que l'on fait l'"approximation de la Terre plate" pour tout ce qui concerne le calcul de l'accélération.

est que la vitesse-sol et la vitesse verticale sont des quantités directement observables par les systèmes de contrôle au sol.

$$\dot{\vec{W}} = \frac{d}{dt} \vec{W}(P, t) = (\nabla \vec{W}) (\vec{V}_g + \vec{V}_z) + \frac{\partial \vec{W}}{\partial t} \quad (\text{G.20})$$

D'un point de vue pratique, il peut être utile d'exprimer $\dot{\vec{W}}$ en fonction de la vitesse-sol V_g , de la route vraie Ψ_i , et de la vitesse verticale \dot{h} . Sachant que l'axe z_h est orienté vers le bas et que h est comptée positivement vers le haut, une variation dz le long de l'axe z_h correspondra à une variation $-dh$ de l'altitude géodésique. Par ailleurs, on supposera que le champ de vent est donné selon ses composantes "vers le Nord", "vers l'Est", et "vers le haut", avec par conséquent $[W_{North}, W_{East}, W_{Up}] = [W_{x_h}, W_{y_h}, -W_{z_h}]$. En tenant compte de ces éléments, le système (G.21) donne les coordonnées de $\dot{\vec{W}}$ dans $Px_h y_h z_h$.

$$\begin{cases} \dot{W}_{x_h} = \frac{\partial W_{North}}{\partial x} V_g \cos \Psi_i + \frac{\partial W_{North}}{\partial y} V_g \sin \Psi_i + \frac{\partial W_{North}}{\partial h} \dot{h} + \frac{\partial W_{North}}{\partial t} \\ \dot{W}_{y_h} = \frac{\partial W_{East}}{\partial x} V_g \cos \Psi_i + \frac{\partial W_{East}}{\partial y} V_g \sin \Psi_i + \frac{\partial W_{East}}{\partial h} \dot{h} + \frac{\partial W_{East}}{\partial t} \\ \dot{W}_{z_h} = - \left[\frac{\partial W_{Up}}{\partial x} V_g \cos \Psi_i + \frac{\partial W_{Up}}{\partial y} V_g \sin \Psi_i + \frac{\partial W_{Up}}{\partial h} \dot{h} + \frac{\partial W_{Up}}{\partial t} \right] \end{cases} \quad (\text{G.21})$$

G.2.6 Expression des forces dans des référentiels choisis

L'expression la plus naturelle des forces est celle donnée dans l'équation (G.22). La poussée des moteurs est exprimée dans le référentiel fixé à l'avion, en supposant que la poussée s'exerce dans une direction définie par un angle ϵ_t par rapport à l'axe de référence x_b lié au fuselage de l'avion. La force aérodynamique, avec ses composantes que sont la traînée, la portance, et la force latérale, est exprimée dans le référentiel lié à la vitesse air. Les coordonnées du poids sont données dans le référentiel horizontal local.

$$\{\vec{Thr}\}_{x_b y_b z_b} = \begin{pmatrix} Thr \cos \epsilon_t \\ 0 \\ Thr \sin \epsilon_t \end{pmatrix}_{x_b y_b z_b} \quad \{\vec{A}\}_{x_w y_w z_w} = \begin{pmatrix} -D \\ -C \\ -L \end{pmatrix}_{x_w y_w z_w} \quad \{m\vec{g}\}_{x_h y_h z_h} = \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix}_{x_h y_h z_h} \quad (\text{G.22})$$

Commençons par exprimer la poussée dans le référentiel "air", par une transformation faisant passer du référentiel "avion" $x_b y_b z_b$ au référentiel "air" $x_w y_w z_w$. Notons $(Thr_{x_w}, Thr_{y_w}, Thr_{z_w})$ les coordonnées du vecteur \vec{Thr} dans le système d'axes $x_w y_w z_w$ lié à la vitesse air \vec{V}_a .

La poussée s'exerce dans le plan de symétrie de l'avion, avec un angle ϵ_t par rapport à l'axe x_b lié au fuselage, lui-même orienté dans une direction définie par l'angle d'attaque α et l'angle de dérapage β , par rapport à x_w . En notant $\alpha_t = \alpha + \epsilon_t$, nous obtenons donc l'équation (G.23) pour la poussée exprimée dans le référentiel "air".

$$\{\vec{Thr}\}_{x_w y_w z_w} = R_3(-\beta) R_2(-\alpha_t) \begin{pmatrix} Thr \\ 0 \\ 0 \end{pmatrix}_{x_b y_b z_b} = \begin{pmatrix} Thr \cos \beta \cos \alpha_t \\ Thr \sin \beta \cos \alpha_t \\ Thr \sin \alpha_t \end{pmatrix} \quad (\text{G.23})$$

Coordonnées des forces dans $x_w y_1 z_1$

Les coordonnées de la somme des deux vecteurs $\overrightarrow{Thr} + \overrightarrow{A}$ dans le système $x_w y_1 z_1$ s'obtiennent par une rotation d'angle $(-\Phi)$ à partir du système d'axes (x_w, y_w, z_w) , en utilisant la matrice de rotation présentée en section G.2.3 (avec l'équation (G.8), où $\theta_x = -\Phi$).

$$\{\overrightarrow{Thr} + \overrightarrow{A}\}_{x_w y_1 z_1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi & \cos \Phi \end{pmatrix} \begin{pmatrix} Thr_{x_w} - D \\ Thr_{y_w} - C \\ Thr_{z_w} - L \end{pmatrix} \quad (\text{G.24})$$

Les coordonnées du poids dans $x_w y_1 z_1$ sont obtenues à partir des coordonnées dans $x_1 y_1 z_h$ par une rotation d'angle γ_a autour de l'axe y_1 , décrite dans l'équation (G.25).

$$\{m\vec{g}\}_{x_w y_1 z_1} = \begin{pmatrix} \cos \gamma_a & 0 & -\sin \gamma_a \\ 0 & 1 & 0 \\ \sin \gamma_a & 0 & \cos \gamma_a \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} \quad (\text{G.25})$$

Coordonnées des forces dans $x_g y_g z_h$

Les coordonnées de la somme des deux vecteurs $\overrightarrow{Thr} + \overrightarrow{A}$ dans le système $x_g y_g z_h$ s'obtiennent en appliquant trois rotations successives transformant le système initial de coordonnées $x_w y_w z_w$ en $x_g y_g z_h$. On a d'abord une rotation d'angle $(-\Phi)$ autour du premier axe (x_w) pour ramener z_w à la verticale, puis une rotation d'angle $(-\gamma_a)$ autour de y_1 pour ramener x_1 dans le plan horizontal, et enfin une rotation d'angle θ_d pour s'aligner sur la vitesse sol, où $\theta_d = \Psi_i - \Psi_a$ est l'angle de dérive. Cette transformation est décrite par l'équation (G.26). La matrice de transformation de $R_3(\theta_d)R_2(-\gamma_a)R_1(-\Phi)$ est développée dans l'équation (G.27).

$$\{\overrightarrow{Thr} + \overrightarrow{A}\}_{x_g y_g z_h} = R_3(\theta_d)R_2(-\gamma_a)R_1(-\Phi) \begin{pmatrix} -D \\ -C \\ -L \end{pmatrix} \quad (\text{G.26})$$

$$\begin{aligned} R_3(\theta_d)R_2(-\gamma_a)R_1(-\Phi) &= \begin{pmatrix} \cos \theta_d & \sin \theta_d & 0 \\ -\sin \theta_d & \cos \theta_d & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \gamma_a & 0 & \sin \gamma_a \\ 0 & 1 & 0 \\ -\sin \gamma_a & 0 & \cos \gamma_a \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi & \cos \Phi \end{pmatrix} \\ &= \begin{pmatrix} \cos \theta_d \cos \gamma_a & \begin{pmatrix} \cos \theta_d \sin \gamma_a \sin \Phi \\ + \sin \theta_d \cos \Phi \end{pmatrix} & \begin{pmatrix} \cos \theta_d \sin \gamma_a \cos \Phi \\ - \sin \theta_d \sin \Phi \end{pmatrix} \\ -\sin \theta_d \cos \gamma_a & \begin{pmatrix} -\sin \theta_d \sin \gamma_a \sin \Phi \\ + \cos \theta_d \cos \Phi \end{pmatrix} & \begin{pmatrix} -\sin \theta_d \sin \gamma_a \cos \Phi \\ - \cos \theta_d \sin \Phi \end{pmatrix} \\ -\sin \gamma_a & \cos \gamma_a \sin \Phi & \cos \gamma_a \cos \Phi \end{pmatrix} \end{aligned} \quad (\text{G.27})$$

Les coordonnées du poids dans $x_g y_g z_h$ sont les mêmes que dans $x_h y_h z_h$, et sont rappelées dans l'équation (G.28).

$$\{m\vec{g}\}_{x_g y_g z_h} = \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} \quad (\text{G.28})$$

G.2.7 Accélération inertielle dans un référentiel mobile

Afin de transformer (G.2) en trois équations scalaires, il est utile d'exprimer l'accélération inertielle dans un référentiel mobile, lié par exemple à la vitesse air \vec{V}_a ou à la vitesse sol \vec{V}_g . En notant F_i le référentiel inertiel (par exemple $x_h y_h z_h$ avec pour origine un point fixé à la Terre, selon l'hypothèse de platitude de la terre), et F le référentiel mobile, et en notant $\{\vec{V}\}_F$ (resp. $\{\vec{A}\}_F$) les coordonnées du vecteur vitesse (resp. de l'accélération) dans le référentiel F , l'accélération inertielle s'écrit comme indiqué dans l'équation (G.29), avec le produit vectoriel noté \wedge .

$$\vec{a}_i = \{\vec{V}\}_F + \vec{\Omega}_{F/F_i} \wedge \{\vec{V}\}_F \quad (\text{G.29})$$

L'accélération dans $x_w y_1 z_1$

Supposons que l'on prenne comme référentiel mobile $x_w y_1 z_1$, le référentiel lié à la vitesse air \vec{V}_a et tel que z_1 soit dans un plan vertical (G.1). Le référentiel $x_w y_1 z_1$ est obtenu à partir du référentiel horizontal local $x_h y_h z_h$ par une rotation d'angle Ψ_a autour de z_h , puis d'une rotation d'angle γ_a autour de y_1 . Les angles d'Euler dans ce cas sont donc (Ψ_a, γ_a, Φ) présentés sur la Figure G.1. La vitesse angulaire $\vec{\Omega}_{x_w y_1 z_1 / x_h y_h z_h}$ du référentiel mobile est donnée par l'équation (G.30).

$$\begin{aligned} \vec{\Omega}_{x_w y_1 z_1 / x_h y_h z_h} &= \dot{\Psi}_a \vec{k} + \dot{\gamma}_a \vec{j}_1 \\ &= \dot{\Psi}_a (-\sin \gamma_a \vec{i}_w + \cos \gamma_a \vec{k}_1) + \dot{\gamma}_a \vec{j}_1 \\ &= \begin{pmatrix} -\dot{\Psi}_a \sin \gamma_a \\ \dot{\gamma}_a \\ \dot{\Psi}_a \cos \gamma_a \end{pmatrix}_{x_w y_1 z_1} \end{aligned} \quad (\text{G.30})$$

L'accélération inertielle dans le référentiel mobile $x_w y_1 z_1$ est alors donnée par l'équation (G.31), obtenue en combinant les équations (G.11b), (G.30) et (G.29).

$$\{\vec{a}_i\}_{x_w y_1 z_1} = \begin{pmatrix} \dot{V}_a \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -\dot{\Psi}_a \sin \gamma_a \\ \dot{\gamma}_a \\ \dot{\Psi}_a \cos \gamma_a \end{pmatrix} \wedge \begin{pmatrix} V_a \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \dot{\vec{W}} \cdot \vec{i}_w \\ \dot{\vec{W}} \cdot \vec{j}_1 \\ \dot{\vec{W}} \cdot \vec{k}_1 \end{pmatrix} \quad (\text{G.31})$$

L'accélération dans $x_g y_g z_h$

Si on choisit $x_g y_g z_h$ (voir Figure G.1) comme référentiel mobile, avec x_g la direction du vecteur vitesse sol \vec{V}_g et Ψ_i la direction de ce vecteur par rapport au Nord (voir les figures G.2 et G.1), on obtient l'équation (G.32). Le système d'axes $x_g y_g z_h$ est obtenu à partir de $x_h y_h z_h$

par une rotation d'angle Ψ_i autour de z_h . La vitesse angulaire du référentiel mobile est $\dot{\Psi}_i \vec{k}$.

$$\begin{aligned} \{\vec{a}_i\}_{x_g y_g z_h} &= \frac{d(V_g \vec{v}_g)}{dt} + \dot{V}_{z_h} \vec{k} \\ &= \begin{pmatrix} \dot{V}_g \\ 0 \\ \dot{V}_{z_h} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \dot{\Psi}_i \end{pmatrix} \wedge \begin{pmatrix} V_g \\ 0 \\ V_{z_h} \end{pmatrix} = \begin{pmatrix} \dot{V}_g \\ V_g \dot{\Psi}_i \\ \dot{V}_{z_h} \end{pmatrix} \end{aligned} \quad (\text{G.32})$$

G.2.8 Les équations de la dynamique, en tenant compte du vent

Dans cette section, on souhaite extraire de (G.2) trois équations différentielles scalaires qui expriment la dynamique du vol, en tenant compte de l'effet du vent sur celle-ci. Le choix des variables intervenant dans les équations du modèle est ici guidé par des considérations pratiques, et par l'utilisation que l'on veut faire du modèle. Pour la gestion du trafic aérien par des systèmes au sol, il est par exemple plus intéressant d'utiliser Ψ_i , la direction de la vitesse sol, plutôt que Ψ_a , la direction de la vitesse air. Toutefois, utiliser simplement les angles d'Euler (Ψ_i, γ_i, Φ) et exprimer l'équation vectorielle (G.2) dans un système d'axes attaché à la vitesse inertielle \vec{V}_i n'est pas très pratique : l'expression des forces dans ce système est compliquée.

On choisit donc une solution mixte, en considérant deux référentiels mobiles, l'un attaché à \vec{V}_g , et l'autre à \vec{V}_a , pour trouver trois équations différentielles scalaires donnant les lois de $\dot{V}_a, \dot{\gamma}_a$, et $\dot{\Psi}_i$.

Expressions de $\dot{V}_a, \dot{\gamma}_a$, et $\dot{\Psi}_i$ en appliquant la 2^{de} loi de Newton dans $x_w y_1 z_1$

Avec l'expression de l'accélération inertielle donnée par l'équation (G.31), la deuxième loi de Newton appliquée au centre de masse de l'avion devient l'équation (G.33).

$$\begin{pmatrix} \dot{V}_a \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -\dot{\Psi}_a \sin \gamma_a \\ \dot{\gamma}_a \\ \dot{\Psi}_a \cos \gamma_a \end{pmatrix} \wedge \begin{pmatrix} V_a \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \dot{\vec{W}} \cdot \vec{v}_w \\ \dot{\vec{W}} \cdot \vec{j}_1 \\ \dot{\vec{W}} \cdot \vec{k}_1 \end{pmatrix} = \frac{\{\overrightarrow{Thr} + \vec{A}\}_{x_w y_1 z_1}}{m} + \{\vec{g}\}_{x_w y_1 z_1} \quad (\text{G.33})$$

Notons W_{x_h}, W_{y_h} et W_{z_h} les coordonnées du vent dans $x_h y_h z_h$, assimilé à un repère fixe. Nous avons $\vec{W} = W_{x_h} \vec{v} + W_{y_h} \vec{j} + W_{z_h} \vec{k}$, et la dérivée du vent s'exprime par $\dot{\vec{W}} = \dot{W}_{x_h} \vec{v} + \dot{W}_{y_h} \vec{j} + \dot{W}_{z_h} \vec{k}$. Les coordonnées de $\dot{\vec{W}}$ dans $x_w y_1 z_1$ sont alors obtenues par deux rotations successives d'angles Ψ_a et γ_a , comme exprimé dans l'équation (G.34).

$$\{\dot{\vec{W}}\}_{x_w y_1 z_1} = \begin{pmatrix} \cos \gamma_a & 0 & -\sin \gamma_a \\ 0 & 1 & 0 \\ \sin \gamma_a & 0 & \cos \gamma_a \end{pmatrix} \begin{pmatrix} \cos \Psi_a & \sin \Psi_a & 0 \\ -\sin \Psi_a & \cos \Psi_a & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{W}_{x_h} \\ \dot{W}_{y_h} \\ \dot{W}_{z_h} \end{pmatrix} \quad (\text{G.34})$$

En développant cette équation, on obtient l'équation (G.35).

$$\{\dot{\vec{W}}\}_{x_w y_1 z_1} = \begin{pmatrix} \dot{W}_{x_h} \cos \gamma_a \cos \Psi_a + \dot{W}_{y_h} \cos \gamma_a \sin \Psi_a - \dot{W}_{z_h} \sin \gamma_a \\ -\dot{W}_{x_h} \sin \Psi_a + \dot{W}_{y_h} \cos \Psi_a \\ \dot{W}_{x_h} \sin \gamma_a \cos \Psi_a + \dot{W}_{y_h} \sin \gamma_a \sin \Psi_a + \dot{W}_{z_h} \cos \gamma_a \end{pmatrix} \quad (\text{G.35})$$

En développant l'équation matricielle (G.31), et en utilisant les équations (G.24) et (G.25) donnant les coordonnées des forces dans $x_w y_1 z_1$, et l'équation (G.35) pour la dérivée du vent, on obtient les équations (G.36), (G.37) et (G.38).

$$\dot{V}_a + \dot{W}_{x_h} \cos \gamma_a \cos \Psi_a + \dot{W}_{y_h} \cos \gamma_a \sin \Psi_a - \dot{W}_{z_h} \sin \gamma_a = \frac{Thr_{x_w} - D}{m} - g \sin \gamma_a \quad (\text{G.36})$$

$$\dot{\Psi}_a V_a \cos \gamma_a - \dot{W}_{x_h} \sin \Psi_a + \dot{W}_{y_h} \cos \Psi_a = \frac{1}{m} \left[(Thr_{y_w} - C) \cos \Phi - (Thr_{z_w} - L) \sin \Phi \right] \quad (\text{G.37})$$

$$\begin{aligned} & -\dot{\gamma}_a V_a + \dot{W}_{x_h} \sin \gamma_a \cos \Psi_a + \dot{W}_{y_h} \sin \gamma_a \sin \Psi_a + \dot{W}_{z_h} \cos \gamma_a \\ & = \frac{1}{m} \left[(Thr_{y_w} - C) \sin \Phi + (Thr_{z_w} - L) \cos \Phi \right] + g \cos \gamma_a \end{aligned} \quad (\text{G.38})$$

Si l'on considère que la poussée s'exerce dans l'axe de la vitesse air et que la composante latérale de la force aérodynamique est nulle, les équations (G.36), (G.37) et (G.38) se simplifient en (G.39), qui donne les équations de la dynamique du vol en tenant compte de l'effet du vent.

$$\dot{V}_a = \frac{Thr - D}{m} - g \sin \gamma_a - \left[\dot{W}_{x_h} \cos \gamma_a \cos \Psi_a + \dot{W}_{y_h} \cos \gamma_a \sin \Psi_a - \dot{W}_{z_h} \sin \gamma_a \right] \quad (\text{G.39a})$$

$$\dot{\Psi}_a = \frac{1}{V_a \cos \gamma_a} \left[\frac{L \sin \Phi}{m} + \dot{W}_{x_h} \sin \Psi_a - \dot{W}_{y_h} \cos \Psi_a \right] \quad (\text{G.39b})$$

$$\dot{\gamma}_a = \frac{L \cos \Phi}{m V_a} - \frac{g \cos \gamma_a}{V_a} + \frac{1}{V_a} \left[\dot{W}_{x_h} \sin \gamma_a \cos \Psi_a + \dot{W}_{y_h} \sin \gamma_a \sin \Psi_a + \dot{W}_{z_h} \cos \gamma_a \right] \quad (\text{G.39c})$$

Élimination de γ_a et Ψ_a dans l'expression de \dot{V}_a

Connaissant le champ de vent et ses variations selon les coordonnées spatiales et temporelle, on peut utiliser (G.21) pour remplacer \dot{W}_{x_h} , \dot{W}_{y_h} , et \dot{W}_{z_h} dans (G.36), (G.37), et (G.38). Cependant, les termes exprimant l'effet du vent contiendront les variables ψ_a et γ_a issus de la rotation du changement d'axes.

On peut chercher des formes alternatives à ces équations, de façon à exprimer l'effet du vent sur la dynamique de l'avion en fonction de paramètres plus facilement observables par les systèmes au sol (V_g , Ψ_i , la vitesse verticale), et en éliminant autant que possible les paramètres liés au système d'axe de la vitesse air.

Par exemple, on peut éliminer Ψ_a et γ_a de l'équation (G.36) ou (G.39a) en repartant de la première ligne de (G.33) et en écrivant $\vec{W} \cdot \vec{v}_w = \frac{\vec{W} \cdot \vec{V}_a}{V_a}$.

$$V_a \dot{V}_a + \vec{W} \cdot \vec{V}_a = V_a \left[\frac{Thr_{x_w} - D}{m} - g \sin \gamma_a \right] \quad (\text{G.40})$$

L'équation (G.12) nous permet d'éliminer γ_a de nos équations, en remplaçant $V_a \sin \gamma_a$ par $\dot{h} - W_{Up}$ dans (G.40). En utilisant (G.13) (G.12), et (G.21) pour exprimer $\vec{W} \cdot \vec{V}_a$, on obtient finalement l'équation (G.41).

$$\boxed{V_a \dot{V}_a + g(\dot{h} - W_{Up}) = \left(\frac{Thr_{x_w} - D}{m} \right) V_a - \vec{W} \cdot \vec{V}_a} \quad (\text{G.41})$$

où

$$\begin{aligned} \vec{W} \cdot \vec{V}_a = & \left[\frac{\partial W_{North}}{\partial x} V_g \cos \Psi_i + \frac{\partial W_{North}}{\partial y} V_g \sin \Psi_i + \frac{\partial W_{North}}{\partial h} \dot{h} + \frac{\partial W_{North}}{\partial t} \right] (V_g \cos \Psi_i - W_{North}) \\ & + \left[\frac{\partial W_{East}}{\partial x} V_g \cos \Psi_i + \frac{\partial W_{East}}{\partial y} V_g \sin \Psi_i + \frac{\partial W_{East}}{\partial h} \dot{h} + \frac{\partial W_{East}}{\partial t} \right] (V_g \sin \Psi_i - W_{East}) \\ & + \left[\frac{\partial W_{Up}}{\partial x} V_g \cos \Psi_i + \frac{\partial W_{Up}}{\partial y} V_g \sin \Psi_i + \frac{\partial W_{Up}}{\partial h} \dot{h} + \frac{\partial W_{Up}}{\partial t} \right] (\dot{h} - W_{Up}) \end{aligned} \quad (\text{G.42})$$

Expression de $\dot{\Psi}_i$ en appliquant la 2nde loi de Newton dans $x_g y_g z_h$

En utilisant (G.32) pour l'accélération inertielle dans $x_g y_g z_h$, la deuxième loi de Newton devient l'équation (G.43) :

$$\begin{pmatrix} \dot{V}_g \\ V_g \dot{\Psi}_i \\ \dot{V}_{z_h} \end{pmatrix} = \frac{\{\vec{Thr} + \vec{A}\}_{x_g y_g z_h}}{m} + \{\vec{g}\}_{x_g y_g z_h} \quad (\text{G.43})$$

L'équation scalaire gouvernant $\dot{\Psi}_i$ est obtenue en considérant la projection sur y_g de l'équation (G.43). En développant les calculs, en utilisant les équations (G.26), (G.27) et (G.28), nous obtenons l'équation (G.44).

$$\begin{aligned} mV_g \dot{\Psi}_i = & D \sin \theta_d \cos \gamma_a + C(\sin \theta_d \sin \gamma_a \sin \Phi - \cos \theta_d \cos \Phi) + L(\sin \theta_d \sin \gamma_a \cos \Phi + \cos \theta_d \sin \Phi) \\ & - Thr_{x_w} \sin \theta_d \cos \gamma_a \\ & + Thr_{y_w} (-\sin \theta_d \sin \gamma_a \sin \Phi + \cos \theta_d \cos \Phi) \\ & - Thr_{z_w} (\sin \theta_d \sin \gamma_a \cos \Phi + \cos \theta_d \sin \Phi) \end{aligned} \quad (\text{G.44})$$

Si l'on considère que la poussée s'exerce dans l'axe de la vitesse air et que la composante latérale de la force aérodynamique est nulle, on obtient l'équation (G.45).

$$\dot{\Psi}_i = \frac{1}{mV_g} \left[L(\sin \theta_d \sin \gamma_a \cos \Phi + \cos \theta_d \sin \Phi) - (Thr - D) \sin \theta_d \cos \gamma_a \right] \quad (\text{G.45})$$

Dans cette équation, θ_d est l'angle entre les vitesses air et sol $\theta_d = \Psi_i - \Psi_a$. Pour calculer cet angle, il faut soit conserver Ψ_a dans notre jeu de variables, soit calculer $\sin \theta_d$ et $\cos \theta_d$ en

utilisant le triangle des vitesses et les équations (G.16) et (G.17). Dans ce cas, on voit que la variable γ_a reste présente dans notre équation.

En pratique, on souhaite utiliser l'équation (G.45) pour calculer les débuts de virage entre segments successifs de la route de l'avion. Pour les avions stables en croisière ($\gamma_a = 0$), on obtient l'équation (G.46).

$$\dot{\Psi}_i = \frac{1}{mV_g} \left[L \sin \Phi \frac{V_g - W_{North} \cos \Psi_i - W_{East} \sin \Psi_i}{V_a} - (Thr - D) \frac{W_{East} \cos \Psi_i - W_{North} \sin \Psi_i}{V_a} \right] \quad (G.46)$$

G.2.9 Hypothèses simplificatrices usuelles

Rappelons d'abord que la principale simplification, que nous avons déjà évoquée précédemment, consiste à négliger l'influence de la rotation de la Terre et à considérer comme fixe le référentiel horizontal local, pour tout ce qui concerne la dynamique du vol, c'est-à-dire l'expression de l'accélération. Bien entendu, nous tiendrons compte de la rotation et de la rotondité de la Terre dans les autres équations du mouvement, notamment pour le calcul de la vitesse et de la position de l'avion, dans la section G.4.

Lorsqu'on traite de vols commerciaux ou d'aviation de loisir à l'exception de la voltige, on considère généralement que la poussée s'applique dans l'axe de la vitesse air et que la composante latérale de la force aérodynamique est nulle. Nous avons déjà évoqué ces simplifications et leurs conséquences sur les équations de la dynamique dans les sections précédentes.

Une autre simplification usuelle, que nous n'avons pas encore vue, consiste à considérer que la pente de la trajectoire varie très faiblement au cours du temps. Autrement formulé, on considère que $\dot{\gamma}_a = \frac{d\gamma_a}{dt} \approx 0$. En appliquant cette hypothèse simplificatrice à l'équation (G.39c), nous pouvons calculer la valeur de la portance L , donnée par l'équation (G.47).

$$L \approx \frac{m}{\cos \Phi} \left[g \cos \gamma_a - \underbrace{\left(\dot{W}_{x_h} \sin \gamma_a \cos \Psi_a + \dot{W}_{y_h} \sin \gamma_a \sin \Psi_a + \dot{W}_{z_h} \cos \gamma_a \right)}_{\text{Influence du vent } (\vec{W} \cdot \vec{j}_1)} \right] \quad (G.47)$$

Dans toutes les publications que nous avons pu lire, l'influence du vent dans cette équation est également négligée, ce qui nous donne l'équation (G.48) pour l'estimation de la portance :

$$L \approx \frac{mg \cos \gamma_a}{\cos \Phi} \quad (G.48)$$

En utilisant l'approximation (G.48) pour la portance, on peut, si on le souhaite, éliminer L et la variable γ_a de l'équation (G.45) pour obtenir une expression de $\dot{\Psi}_i$ qui reste valable pour les vols évolutifs⁶, mais qui ne fasse plus apparaître les variables Ψ_a et γ_a . Pour cela, on utilise (G.16) et (G.17) pour remplacer $\sin \theta_d \cos \gamma_a$ et $\cos \theta_d \cos \gamma_a$ par des expressions

6. Contrairement à (G.46) qui n'est valable que pour les vols stables en croisière.

ne dépendant que de la vitesse sol, du vent, et de V_a , et on remarque d'après (G.40) que $\frac{Thr-D}{m} - g \sin \gamma_a = \dot{V}_a + \frac{\vec{W} \cdot \vec{V}_a}{V_a}$.

$$\begin{aligned}
\dot{\Psi}_i &\approx \frac{1}{mV_g} \left[\frac{mg \cos \gamma_a}{\cos \Phi} (\sin \theta_d \sin \gamma_a \cos \Phi + \cos \theta_d \sin \Phi) - (Thr - D) \sin \theta_d \cos \gamma_a \right] \\
&= \frac{1}{V_g} \left[g \cos \gamma_a \cos \theta_d \tan \Phi - \left(\frac{Thr - D}{m} - g \sin \gamma_a \right) \sin \theta_d \cos \gamma_a \right] \\
&= \frac{1}{V_g} \left[g \cos \gamma_a \cos \theta_d \tan \Phi - \left(\dot{V}_a + \frac{\vec{W} \cdot \vec{V}_a}{V_a} \right) \sin \theta_d \cos \gamma_a \right] \\
&= \frac{1}{V_g V_a} \left[g \tan \Phi (V_g - W_{North} \cos \Psi_i - W_{East} \sin \Psi_i) - \left(\dot{V}_a + \frac{\vec{W} \cdot \vec{V}_a}{V_a} \right) (W_{East} \cos \Psi_i - W_{North} \sin \Psi_i) \right]
\end{aligned} \tag{G.49}$$

Dans cette expression (G.49) de $\dot{\Psi}_i$, le terme $\frac{\vec{W} \cdot \vec{V}_a}{V_a}$ faisant intervenir la dérivée du vent est donné par l'équation (G.42).

Remarquons toutefois, en regardant les équations (G.40), (G.49) et (G.42), que le fait d'éliminer Ψ_a et γ_a de nos équations pour ne conserver que V_a et des variables observables directement par les systèmes au sol (V_g, Ψ_i, h) a introduit des interdépendances plus complexes entre \dot{V}_a, \dot{h} et $\dot{\Psi}_i$. Ceci ne facilitera sans doute pas la résolution de notre système d'équations différentielles. Ce point sera discuté dans la section G.5.

G.3 Modélisation des forces s'appliquant à l'avion

G.3.1 Les forces aérodynamiques

Les forces aérodynamiques exercées par le déplacement relatif de l'air sur la surface de l'avion sont proportionnelles à la surface portante S et à la pression dynamique $q = \frac{1}{2} \rho V_a^2$, où $\rho = \rho(h)$ est la densité de l'air, qui dépend de l'altitude h de l'avion, et V_a est la vitesse air.

Pour des applications de gestion de trafic aérien, on néglige la composante latérale C de la force aérodynamique, pour ne s'intéresser qu'à la portance L (*lift*) et à la traînée D (*drag*). Celles-ci s'expriment sous la forme des équations (G.50) en utilisant des coefficients sans dimension : C_L pour le coefficient de portance, et C_D pour le coefficient de traînée.

$$L = C_L \frac{\rho V_a^2 S}{2} \tag{G.50a}$$

$$D = C_D \frac{\rho V_a^2 S}{2} \tag{G.50b}$$

La relation entre les coefficients de traînée et de portance, dite courbe polaire d'Eiffel, est de forme quadratique. Elle est donnée dans l'équation (G.51).

$$C_D = c_{D0} + c_{D2} C_L^2 \tag{G.51}$$

Les coefficients c_{D0} et c_{D2} sont fonction de la phase de vol, qu'on peut modéliser comme une variable discrète. En effet, la forme du profil de l'aile est modifiée par la sortie des volets ou des traînées, et on n'a donc pas les mêmes coefficients selon qu'on est en configuration lisse, ou avec un ou plusieurs crans de volets, ou lorsque le train d'atterrissage est sorti.

Par ailleurs, on peut approximer C_L en combinant (G.50a) et (G.48), ce qui nous donne (G.52). Rappelons que l'équation (G.48) est une simplification de (G.38), en supposant que la variation de la pente au cours du temps est très faible ($\dot{\gamma}_a \approx 0$), que la poussée s'exerce sensiblement dans l'axe de la vitesse air ($Thr_{y_w} \approx 0$ et $Thr_{z_w} \approx 0$), et qu'on néglige la force aérodynamique latérale C et l'influence du vent dans (G.38).

$$L = C_L \frac{\rho V_a^2 S}{2} \approx \frac{mg \cos \gamma_a}{\cos \Phi} \quad (\text{G.52})$$

Ceci nous permet d'obtenir l'équation (G.53) donnant l'expression du coefficient de portance C_L , et son approximation pour des pentes γ_a faibles.

$$C_L = \frac{2mg \cos \gamma_a}{\rho S V_a^2 \cos \Phi} \underset{\cos \gamma_a \approx 1}{\approx} \frac{2mg}{\rho S V_a^2 \cos \Phi} \quad (\text{G.53})$$

En pratique on utilisera donc (G.48) pour calculer la portance L , puis (G.53) et (G.51) pour calculer C_L et C_D , ce qui permet finalement d'obtenir la traînée D en utilisant (G.50b).

G.3.2 La poussée des moteurs

Généralités sur la propulsion

La poussée est générée par un système de propulsion. Un tel système est une machine produisant un certain travail, transféré à un fluide accéléré vers l'arrière du système de propulsion. Par exemple, les fusées (non utilisées pour les avions commerciaux) produisent le fluide qu'elles éjectent à partir de propellants liquides ou solides. Comme autre exemple de système de propulsion, on trouve les engins à hélices, entraînés par un moteur à pistons ou par une turbine à gaz (turboprop). On trouve également les turbojets, qui brûlent et éjectent un mélange d'air et de carburant, et les turbofans, où une part du flux d'air est mixée avec le carburant, brûlée, et éjectée, alors que le reste est simplement accéléré par une hélice.

La force de poussée est dirigée dans la direction opposée au fluide éjecté. Son intensité dépend essentiellement de la quantité de fluide éjectée et de sa vitesse d'éjection. Supposons, pour simplifier, que le flux d'air soit parallèle à l'axe du système de propulsion. Avant l'entrée dans le système, la vitesse relative de l'air est V_a et sa pression est p_a . A la sortie, la vitesse du gaz éjecté est V_e , et sa pression p_e .

Considérons une masse Δm_a d'air accélérée par le système de propulsion pendant un temps Δt . Soit Δm_f la masse de carburant injectée, brûlée, puis éjectée vers l'arrière pendant ce laps de temps Δt . Écrivons la conservation de la quantité de mouvement du mélange d'air et de carburant, en tenant compte du fait que la vitesse initiale du carburant dans le référentiel de l'avion est nulle, et en introduisant une force due à la différence entre pression d'admission et pression d'éjection. On obtient l'équation (G.54), où A_e est l'aire de la section de la tuyère d'éjection.

$$Thr + (p_a - p_e)A_e = \frac{[\Delta m_a(V_e - V_a) + \Delta m_f(V_e - 0)]}{\Delta t} \quad (\text{G.54})$$

Notons que l'équation (G.54) n'est valable qu'à des vitesses assez faibles pour pouvoir négliger les effets de compressibilité et de viscosité du fluide éjectés. A des vitesses élevées, ces effets doivent être pris en compte. L'objet de cette section étant de donner un éclairage rapide sur les phénomènes physiques mis en jeu pour produire une force de poussée, nous ne détaillerons pas ces effets.

En passant à la limite et en introduisant $\dot{m}_a = \frac{dm_a}{dt}$ et $\dot{m}_f = \frac{dm_f}{dt}$ les flux massiques d'air et de carburant, nous obtenons l'équation générale de la poussée (G.55).

$$Thr = \dot{m}_a(V_e - V_a) + \dot{m}_f V_e + (p_e - p_a)A_e \quad (\text{G.55})$$

Pour les moteurs équipant les vols commerciaux, le flux de carburant \dot{m}_f est largement inférieur au flux d'air \dot{m}_a . Par ailleurs, les tuyères d'éjection sont conçues de telle façon que $p_e \approx p_a$. Avec ces approximations, on voit que la poussée dépend essentiellement du flux d'air \dot{m}_a , et de la vitesse relative d'éjection des gaz accélérés ($V_e - V_a$).

$$Thr \approx \dot{m}_a(V_e - V_a) \quad (\text{G.56})$$

En considérant l'équation (G.56), on voit qu'on peut produire de la poussée soit en accélérant une grande quantité d'air, soit en éjectant des gaz à haute vitesse. Typiquement, les engins à hélice utilisent la première approche, alors que les turbojets s'appuient sur la seconde. Les turbofans (turboréacteurs à double flux), qui équipent aujourd'hui la quasi-totalité des jets commerciaux, combinent les deux approches, avec une partie du flux d'air qui "bypass" le cœur du moteur pour être propulsé par une hélice⁷.

Modélisation de la force de poussée

L'équation (G.56) nous donne le principe général pour générer de la poussée en éjectant un fluide, ici un mélange d'air et de gaz brûlés. Cependant, elle ne nous suffit pas pour modéliser comment la puissance développée par le moteur de l'avion est transmise au fluide éjecté vers l'arrière du système propulsif pour donner une force de poussée.

L'efficacité de ce transfert de puissance dépend du type de système propulsif : hélice entraînée par un moteur à pistons, hélice entraînée par une turbine (turboprop), ou turboréacteur (turbojet ou turbofan). Pour chaque catégorie, il peut par ailleurs dépendre de la vitesse air V_a et/ou de la densité et de la température de l'air, qui dépendent elles-mêmes de l'altitude h . L'expression la plus générale de la poussée en fonction de ces paramètres et de la commande de poussée c choisie par le pilote est de la forme de l'équation (G.57).

$$Thr = Thr(c, V_a, h) \quad (\text{G.57})$$

Pour une commande c donnée, la dépendance entre la poussée et la vitesse V_a et/ou l'altitude h peut être relativement complexe. Elle peut par exemple dépendre du type d'hélice (pas variable ou fixe), ou du type de système d'alimentation en air⁸.

Pour trouver un modèle utilisable de la poussée, on passe donc généralement par des modèles paramétriques dont les coefficients sont ajustés sur des données observées, pour chaque type d'engin propulsif.

7. L'équation de poussée pour les turbofans s'écrit d'ailleurs souvent en utilisant un "by-pass rate" (BPR).

8. Certains engins sont équipés d'un "turbocharger" qui peut maintenir constante la densité de l'air alimentant le moteur, du moins jusqu'à une altitude critique au-delà de laquelle elle décroît.

Par ailleurs, ces modèles paramétriques sont généralement donnés en fonction de l'altitude-pression H_p plutôt que l'altitude géodésique h , et font intervenir les conditions de température T de l'atmosphère :

$$Thr = Thr(c, V_a, H_p, T) \quad (\text{G.58})$$

L'exemple de la poussée *max climb* du modèle BADA

A titre d'exemple, le modèle BADA (Base of Aircraft Data, version 3.9, [128]) développé par Eurocontrol, définit trois modèles différents pour la poussée maximum en montée (*max climb*) en conditions standard d'atmosphère (ISA), donnés par les équations (G.59). Dans ces équations, H_p est l'altitude-pression géopotentielle de l'avion (voir annexe D).

$$\text{Jet : } (Thr_{\max \text{ climb}})_{ISA} = C_{T_c,1} \left(1 - \frac{H_p}{C_{T_c,2}} + C_{T_c,3} H_p^2 \right) \quad (\text{G.59a})$$

$$\text{Turboprop : } (Thr_{\max \text{ climb}})_{ISA} = \frac{C_{T_c,1}}{V_a} \left(1 - \frac{H_p}{C_{T_c,2}} \right) + C_{T_c,3} \quad (\text{G.59b})$$

$$\text{Piston : } (Thr_{\max \text{ climb}})_{ISA} = C_{T_c,1} \left(1 - \frac{H_p}{C_{T_c,2}} \right) + \frac{C_{T_c,3}}{V_a} \quad (\text{G.59c})$$

Lorsque l'atmosphère n'est pas dans les conditions standard de température, la poussée donnée par (G.59) est corrigée par un coefficient multiplicatif (voir [128] et révisions ultérieures) qui dépend de l'écart de température ΔT par rapport aux conditions ISA. Les coefficients de tous ces modèles sont ajustés sur des données constructeur pour chaque type de moteur d'avion de chacune des trois catégories.

G.3.3 Le poids de l'avion

Le poids de l'avion, appliqué au centre de gravité, s'exprime simplement comme $m \vec{g}$, où m est la masse de l'avion et $g = g(h)$ est l'accélération due à la gravité terrestre, qui dépend de l'altitude géodésique h .

La valeur standard pour l'accélération due à la gravité au niveau moyen de la mer est $g_0 = 9,80665 m.s^{-2}$. Dans la réalité, la gravité g est le résultat des effets de la gravitation et de l'accélération centrifuge due à la rotation de la terre, et dépend donc de la latitude et de la distance au centre de la terre. Pour le calcul pratique de g , on ne considérera que les effets de la gravitation et on se contentera d'une approximation sphérique du globe terrestre. La valeur de g en fonction de l'altitude géodésique h est alors donnée par (G.60), où $R_T = 6325766m$.

$$g(h) = g_0 \left(\frac{R_T}{R_T + h} \right)^2 \quad (\text{G.60})$$

G.4 La cinématique de l'avion

G.4.1 Représentation dans l'espace d'état

Choix de la commande

La trajectoire de l'avion dépend de choix faits par le pilote. Ces choix sont modélisés par une commande, c'est-à-dire un vecteur de paramètres dont les valeurs sont déterminées par le

pilote. Dans le cas d'un simulateur d'avion, cette commande pourra comprendre par exemple la position des ailerons de direction et de profondeur, et celle de la manette des gaz.

Pour les besoins liés à la gestion du trafic aérien, la commande de la trajectoire de l'avion est largement simplifiée en ne considérant que trois paramètres : la poussée des moteurs Thr , la pente-air γ_a , et l'angle d'inclinaison de l'avion Φ . Les deux premiers paramètres déterminent les composantes longitudinale et verticale du déplacement de l'avion, et le dernier sa composante latérale. On rassemble les différents paramètres de commande dans un vecteur de commande $\vec{U} = [c, \gamma_a, \Phi]$, où c est la commande de poussée des moteurs. La poussée dépend de la commande et de certains paramètres d'état de l'avion : $Thr(c, V_a, H_p, T)$, où H_p est l'altitude-pression et T la température.

Notons tout de suite que le choix du vecteur de commande pourra être remis en question. Par exemple, en l'absence de contraintes particulières, l'avion en montée suit généralement un profil de vitesse composé de segments successifs à vitesse calibrée (CAS) constante, ou à nombre de Mach constant. Dans un tel cas, on voit bien que c'est la loi $CAS(t)$ ou $Mach(t)$ qui commande la montée, et non pas $\gamma_a(t)$. Dans l'immédiat, nous allons cependant supposer que l'avion est commandé en pente-air γ_a , pour expliciter les équations du mouvement.

Choix du vecteur d'état

La position de l'avion au-dessus de la Terre peut être exprimée de différentes manières selon le modèle choisi pour représenter la surface terrestre : représentation sphérique, ellipsoïde de révolution, ou projection sur un plan. Dans une représentation plane, elle sera donnée par des coordonnées (x, y) dans le plan, et par l'altitude h au-dessus du plan. Sur une sphère ou un ellipsoïde de révolution, la position sera donnée par la longitude λ et la latitude géodésique μ , et l'altitude géodésique h .

En plus de sa position, il nous faut connaître à chaque instant la vitesse de l'avion pour pouvoir calculer sa trajectoire. En supposant le vent connu, la vitesse par rapport à la surface terrestre est parfaitement déterminée par les trois paramètres de la vitesse air (V_a, γ_a, Ψ_a) . L'un de ces paramètres, la pente-air γ_a , fait partie du vecteur de commande que l'on a choisi. L'évolution des deux autres paramètres au cours du temps est donnée par les lois de la dynamique de l'avion. Un autre paramètre important dans les équations de la dynamique de l'avion est la masse m , qui évolue également au cours du temps en fonction de la consommation de carburant.

Au final, pour un vecteur de commande $\vec{U} = [c, \gamma_a, \Phi]$ donné, où c est la commande de poussée, l'état de l'avion à un instant t sera déterminé par le vecteur d'état $\vec{X} = [\lambda, \mu, h, V_a, \Psi_a, m]$ dans le cas où la Terre est modélisée par une sphère ou un ellipsoïde de révolution, ou $\vec{X} = [x, y, h, V_a, \Psi_a, m]$ dans le cas d'une représentation plane.

Comme on souhaite contrôler latéralement la trajectoire de façon à suivre des segments de routes fixés par rapport à la surface terrestre, l'orientation Ψ_a de la projection du vecteur vitesse-air \vec{V}_a dans le plan horizontal local n'est par toujours la variable la plus utile à manipuler. Une alternative consiste à choisir Ψ_i , l'orientation de la vitesse-sol \vec{V}_g , comme variable d'état à la place de Ψ_a .

Par ailleurs, la variable utile pour représenter l'altitude de l'avion est l'altitude-pression géopotentielle H_p , et non l'altitude géodésique h . Les évolutions verticales sont données par le taux de montée et de descente $\dot{H}_p = \frac{dH_p}{dt}$ (ROCD⁹) plutôt que la vitesse verticale \dot{h} . Nous

9. ROCD : *Rate of Climb or Descent*.

présenterons donc également les équations du mouvement de l'avion en considérant un vecteur d'état $\vec{X} = [\lambda, \mu, H_p, V_a, \Psi_i, m]$ (ou $\vec{X} = [x, y, H_p, V_a, \Psi_i, m]$ dans le plan).

G.4.2 Les équations du mouvement de l'avion

Cette section rassemble les équations du mouvement de l'avion, obtenues d'une part en considérant l'expression de la vitesse comme dérivée du vecteur de position, et d'autre part en utilisant les équations (G.39) de la dynamique du vol et (G.3) de la consommation de carburant.

Sur un ellipsoïde de révolution

$$\begin{aligned}\vec{U} &= [c, \gamma_a, \Phi] \\ \vec{X} &= [\lambda, \mu, h, V_a, \Psi_a, m]\end{aligned}\tag{G.61}$$

$$\begin{cases} \dot{\mu} = \frac{V_a \cos \gamma_a \cos \Psi_a + W_{North}}{R_M + h} \\ \dot{\lambda} = \frac{V_a \cos \gamma_a \sin \Psi_a + W_{East}}{R_P + h \cos \mu} \\ \dot{h} = V_a \sin \gamma_a + W_{Up} \\ \dot{V}_a = \frac{Thr - D}{m} - g \sin \gamma_a - \left[\dot{W}_{North} \cos \gamma_a \cos \Psi_a + \dot{W}_{East} \cos \gamma_a \sin \Psi_a + \dot{W}_{Up} \sin \gamma_a \right] \\ \dot{\Psi}_a = \frac{1}{V_a \cos \gamma_a} \left[\frac{L \sin \Phi}{m} + W_{North} \sin \Psi_a - W_{East} \cos \Psi_a \right] \\ \dot{m} = -\eta Thr \end{cases}\tag{G.62}$$

Dans les équations (G.62), R_M est le rayon de courbure de l'ellipse dans le plan méridien, au point de coordonnées (λ, μ) , et R_P est le rayon de courbure dans le plan parallèle passant également par ce point. Les expressions de ces rayons de courbure sont donnés dans les équations (G.63) et (G.64).

$$R_M = \frac{a(1 - e^2)}{(1 - e^2 \sin^2 \mu)^{\frac{3}{2}}}\tag{G.63}$$

$$R_P = \frac{a \cos \mu}{(1 - e^2 \sin^2 \mu)^{\frac{1}{2}}}\tag{G.64}$$

Notons que les paramètres W_{North} , W_{East} , et W_{Up} dépendent de la position (λ, μ, h) de l'avion dans le champ de vent qui varie par ailleurs au cours du temps. Si l'on note \vec{W} le champ de vent, on doit donc rajouter l'équation (G.65) aux équations (G.62), pour déterminer le mouvement de l'avion.

$$[W_{North}, W_{East}, W_{Up}] = \vec{W}(\lambda, \mu, h, t)\tag{G.65}$$

Par ailleurs, les forces de poussée, de traînée, et de portance dépendent aussi d'un certain nombre paramètres, comme la température extérieure, la vitesse, ou l'altitude de l'avion. Il faut donc ajouter aussi aux équations différentielles (G.62) des équations algébriques décrivant ces relations, pour pouvoir déterminer complètement le mouvement de l'avion.

$$L = L(\vec{U}, \vec{X}) = C_L \frac{\rho V_a^2 S}{2} \approx \frac{mg \cos \gamma_a}{\cos \Phi} \quad (\text{G.66a})$$

$$D = D(\vec{U}, \vec{X}) = C_D \frac{\rho V_a^2 S}{2} \quad (\text{G.66b})$$

$$Thr = Thr(\vec{U}, \vec{X}) = Thr(c, V_a, h) \quad (\text{G.66c})$$

$$g = g(\vec{U}, \vec{X}) = g(\lambda, \mu, h) \quad (\text{G.66d})$$

$$\left\{ \begin{array}{l} h \in [0, h_{max}] \\ V_a \in [V_{a,min}, V_{a,max}] \\ \gamma_a \in [\gamma_{a,min}, \gamma_{a,max}] \\ \Phi \in [0, Phi_{max}] \\ m \in [m_{min}, m_{max}] \\ \dot{V}_a \in [\dot{V}_{a,min}, \dot{V}_{a,max}] \\ |\dot{\gamma}_a| \leq \dot{\gamma}_{a,max} \\ |\dot{\Phi}| \leq \dot{\Phi}_{max} \\ \dot{m} \in [\dot{m}_{min}, \dot{m}_{max}] \\ Thr \in [Thr_{min}, Thr_{max}] \end{array} \right. \quad (\text{G.67})$$

Sur une sphère

$$\begin{aligned} \vec{U} &= [c, \gamma_a, \Phi] \\ \vec{X} &= [\lambda, \mu, h, V_a, \Psi_a, m] \end{aligned} \quad (\text{G.68})$$

$$\left\{ \begin{array}{l} \dot{\mu} = \frac{V_a \cos \gamma_a \cos \Psi_a + W_{North}}{R_T + h} \\ \dot{\lambda} = \frac{V_a \cos \gamma_a \sin \Psi_a + W_{East}}{(R_T + h) \cos \mu} \\ \dot{h} = V_a \sin \gamma_a + W_{Up} \\ \dot{V}_a = \frac{Thr - D}{m} - g \sin \gamma_a - \left[\dot{W}_{North} \cos \gamma_a \cos \Psi_a + \dot{W}_{East} \cos \gamma_a \sin \Psi_a + \dot{W}_{Up} \sin \gamma_a \right] \\ \dot{\Psi}_a = \frac{1}{V_a \cos \gamma_a} \left[\frac{L \sin \Phi}{m} + W_{North} \sin \Psi_a - \dot{W}_{East} \cos \Psi_a \right] \\ \dot{m} = -\eta Thr \end{array} \right. \quad (\text{G.69})$$

Dans un plan

$$\begin{aligned}\vec{U} &= [c, \gamma_a, \Phi] \\ \vec{X} &= [x, y, h, V_a, \Psi_a, m]\end{aligned}\quad (\text{G.70})$$

$$\begin{cases} \dot{x} = V_a \cos \gamma_a \cos \Psi_a + W_{North} \\ \dot{y} = V_a \cos \gamma_a \sin \Psi_a + W_{East} \\ \dot{h} = V_a \sin \gamma_a + W_{Up} \\ \dot{V}_a = \frac{Thr - D}{m} - g \sin \gamma_a - \left[\dot{W}_{North} \cos \gamma_a \cos \Psi_a + \dot{W}_{East} \cos \gamma_a \sin \Psi_a + \dot{W}_{Up} \sin \gamma_a \right] \\ \dot{\Psi}_a = \frac{1}{V_a \cos \gamma_a} \left[\frac{L \sin \Phi}{m} + \dot{W}_{North} \sin \Psi_a - \dot{W}_{East} \cos \Psi_a \right] \\ \dot{m} = -\eta Thr \end{cases} \quad (\text{G.71})$$

G.4.3 Équations avec la route vraie Ψ_i et l'altitude-pression H_p

Avant l'apparition des moyens satellites de positionnement, le seul moyen d'estimer l'altitude de l'avion était basé sur une mesure de la pression atmosphérique. Aujourd'hui encore, toutes les instructions, contraintes, et consignes données à l'avion en ce qui concerne son évolution verticale sont données en *altitude pression* géopotentielle (ou simplement "altitude pression"). Cette altitude pression correspond à une altitude géopotentielle calculée en faisant l'hypothèse de conditions atmosphériques standard (voir annexes D et E).

En notant h l'altitude géodésique, H l'altitude géopotentielle, et H_p l'altitude pression géopotentielle, on exprime facilement le terme $g\dot{h}$, où $g = g(h)$ dépend de l'altitude géodésique h , sous la forme de l'équation (G.72).

$$g\dot{h} = g \frac{dh}{dt} = g_0 \frac{dH}{dt} = g_0 \frac{dH}{dH_p} \frac{dH_p}{dt} = g_0 \left(\frac{T}{T_{ISA}} \right) \frac{dH_p}{dt} = g_0 \left(\frac{T}{T - \Delta T} \right) \frac{dH_p}{dt} \quad (\text{G.72})$$

Dans cette équation, g_0 désigne l'accélération due à la gravité terrestre, mesurée au niveau moyen de la mer, T est la température à la position et à l'altitude de l'avion, et $T - \Delta T = T_{ISA}$ est la température qu'on obtiendrait au même point dans une atmosphère standard ISA. L'équation (G.72) nous permet d'éliminer $g = g(h)$ et la vitesse verticale \dot{h} de nos équations, en y faisant apparaître $\dot{H}_p = \frac{dH_p}{dt}$, le taux de montée ou de descente (ROCD, pour *Rate of Climb or Descent*).

L'élimination de h des première et deuxième équations des systèmes (G.62), (G.69), et (G.71) est plus problématique, puisqu'il faut intégrer (G.72) en connaissant $g(h)$ et la température en fonction de l'altitude pression, afin d'obtenir la relation entre h et H_p . Pour cela, on intègre (G.60) pour exprimer la relation entre altitude géodésique h et altitude géopotentielle H (en faisant l'hypothèse d'une terre sphérique), et on utilise (D.24) qui donne H en fonction de l'altitude pression H_p , pour finalement obtenir $h = h(H_p, \Delta T, p_{MSL})$. Notons qu'il faut connaître les conditions de température du jour ΔT et la pression atmosphérique au niveau moyen de la mer p_{MSL} pour pouvoir faire ce calcul.

Par ailleurs, dans certains cas, on sera plus intéressé par l'évolution de Ψ_i , la direction de la vitesse-sol, que par l'évolution de Ψ_a , la direction de la vitesse-air. C'est par exemple le cas lorsqu'on souhaite contrôler latéralement la trajectoire afin de suivre des segments de route successifs entre l'aéroport de départ et de destination. L'équation donnant $\dot{\Psi}_i$ nous est donnée par (G.45). Tenir compte de Ψ_i nous impose de rajouter les relations algébriques entre la vitesse-sol, la vitesse-air, et le vent dans nos équations. Au final, on obtient un système d'équations différentielles et algébriques, qui peut prendre différentes formes selon qu'on considère le mouvement de l'avion sur un ellipsoïde de révolution, sur une sphère, ou dans un plan.

Équations du mouvement sur un ellipsoïde de révolution

Commande et vecteur d'état :

$$\begin{aligned}\vec{U} &= [c, \gamma_a, \Phi] \\ \vec{X} &= [\lambda, \mu, H_p, V_a, \Psi_i, m]\end{aligned}\tag{G.73}$$

Données météorologiques :

- le champ de vent \vec{W} , permettant d'obtenir le vent $\vec{W} = \vec{W}(P, t)$ en tout point P au temps t , et de calculer le gradient du vent.
- ΔT , la différence de température entre l'atmosphère du jour et l'atmosphère standard,
- p_{MSL} , la pression atmosphérique au niveau moyen de la mer.

Données des modèles terrestres :

- excentricité e (ou aplatissement f), et grand axe a pour le calcul de R_P et R_M à la position courante de l'avion,
- rayon terrestre R_T , car on fait l'hypothèse d'un champ géopotentiel autour d'une terre sphérique pour le calcul de l'altitude géodésique h en fonction de l'altitude géopotentielle H .

$$\left\{ \begin{aligned}\dot{\mu} &= \frac{V_g \cos \Psi_i}{R_M + h(H_p, \Delta T, p_{MSL})} \\ \dot{\lambda} &= \frac{V_g \sin \Psi_i}{R_P + h(H_p, \Delta T, p_{MSL}) \cos \mu} \\ \frac{g_0}{g} \left(\frac{T}{T_{ISA}} \right) \dot{H}_p &= V_a \sin \gamma_a + W_{Up} \\ V_a \dot{V}_a + g_0 \left(\frac{T}{T_{ISA}} \right) \dot{H}_p &= \left(\frac{Thr - D}{m} \right) V_a - \dot{\vec{W}} \cdot \vec{V}_a + g W_{Up} \\ \dot{\Psi}_i &= \frac{1}{m V_g} \left[L(\sin \theta_d \sin \gamma_a \cos \Phi + \cos \theta_d \sin \Phi) - (Thr - D) \sin \theta_d \cos \gamma_a \right] \\ \dot{m} &= -\eta Thr\end{aligned}\right.\tag{G.74}$$

Dans ces équations, θ_d est l'angle entre les vitesses air et sol $\theta_d = \Psi_i - \Psi_a$.

$$\begin{aligned} \vec{W} \cdot \vec{V}_a = & \\ & \left[\frac{\partial W_{North}}{\partial x} V_g \cos \Psi_i + \frac{\partial W_{North}}{\partial y} V_g \sin \Psi_i + \frac{\partial W_{North}}{\partial h} \frac{g_0}{g} \left(\frac{T}{T_{ISA}} \right) \dot{H}_p + \frac{\partial W_{North}}{\partial t} \right] (V_g \cos \Psi_i - W_{North}) \\ & + \left[\frac{\partial W_{East}}{\partial x} V_g \cos \Psi_i + \frac{\partial W_{East}}{\partial y} V_g \sin \Psi_i + \frac{\partial W_{East}}{\partial h} \frac{g_0}{g} \left(\frac{T}{T_{ISA}} \right) \dot{H}_p + \frac{\partial W_{East}}{\partial t} \right] (V_g \sin \Psi_i - W_{East}) \\ & + \left[\frac{\partial W_{Up}}{\partial x} V_g \cos \Psi_i + \frac{\partial W_{Up}}{\partial y} V_g \sin \Psi_i + \frac{\partial W_{Up}}{\partial h} \frac{g_0}{g} \left(\frac{T}{T_{ISA}} \right) \dot{H}_p + \frac{\partial W_{Up}}{\partial t} \right] \left(\frac{g_0}{g} \left(\frac{T}{T_{ISA}} \right) \dot{H}_p - W_{Up} \right) \end{aligned} \quad (G.75)$$

$$\begin{cases} V_g \cos \Psi_i = V_a \cos \gamma_a \cos \Psi_a + W_{North} \\ V_g \sin \Psi_i = V_a \cos \gamma_a \sin \Psi_a + W_{East} \end{cases} \quad (G.76)$$

Pour obtenir la relation $h = h(H_p, \Delta T, p_{MSL})$ entre l'altitude géodésique h et l'altitude-pression H_p , on exprime d'une part la relation entre h et H , l'altitude géopotentielle, et d'autre part la relation entre altitude géopotentielle H et altitude-pression H_p . La première relation s'obtient en intégrant (G.60), en faisant l'hypothèse d'un champ géopotentiel autour d'une terre sphérique de rayon R_T . La deuxième dépend des conditions atmosphériques et fait intervenir ΔT , l'écart à la température standard $T_0 = 288.15 \text{ K}$ à l'isobare $p = p_0 = 1013.15 \text{ hPa}$, et la pression p_{MSL} au niveau de la mer.

$$h = \frac{R_T H}{R_T - H} \quad (G.77)$$

$$\begin{aligned} H &= H_p - H_{p,MSL} + \frac{\Delta T}{\beta} \ln \left(\frac{T_0 + \beta H_p}{T_{ISA,MSL}} \right) && \text{pour } H_p < H_{p,trop} \\ H_{trop} &= H_{p,trop} - H_{p,MSL} + \frac{\Delta T}{\beta} \ln \left(\frac{T_{ISA,trop}}{T_{ISA,MSL}} \right) && \\ H &= H_{trop} + \frac{T_{ISA,trop}}{T_{ISA,MSL}} (H_p - H_{p,trop}) && \text{pour } H_p \geq H_{p,trop} \end{aligned} \quad (G.78)$$

avec

$$H_{p,MSL} = \frac{T_0}{\beta} \left[\left(\frac{p_{MSL}}{p_0} \right)^{\frac{g_0}{\beta R}} - 1 \right] \quad (G.79a)$$

$$H_{p,trop} = 11000 \text{ m} \quad (G.79b)$$

$$T_{ISA,MSL} = T_0 + \beta H_{p,MSL} \quad (G.79c)$$

$$T_{ISA,trop} = T_0 + \beta H_{p,trop} \quad (G.79d)$$

$$(G.79e)$$

Dans ces équations, β est le gradient de température dans la troposphère, et R est la constante spécifique des gaz, pour l'air sec. La pression au niveau moyen de la mer peut être exprimée par un écart aux conditions standard ISA : $p_{MSL} = p_0 + \Delta p$.

Équations du mouvement sur une sphère

$$\begin{aligned}\vec{U} &= [c, \gamma_a, \Phi] \\ \vec{X} &= [\lambda, \mu, H_p, V_a, \Psi_i, m]\end{aligned}\tag{G.80}$$

$$\left\{ \begin{aligned}\dot{\mu} &= \frac{V_g \cos \Psi_i}{R_T + h(H_p, \Delta T, p_{MSL})} \\ \dot{\lambda} &= \frac{V_g \sin \Psi_i}{[R_T + h(H_p, \Delta T, p_{MSL})] \cos \mu} \\ \frac{g_0}{g} \left(\frac{T}{T_{ISA}} \right) \dot{H}_p &= V_a \sin \gamma_a + W_{Up} \\ V_a \dot{V}_a + g_0 \left(\frac{T}{T_{ISA}} \right) \dot{H}_p &= \left(\frac{Thr - D}{m} \right) V_a - \dot{\vec{W}} \cdot \vec{V}_a + g W_{Up} \\ \dot{\Psi}_i &= \frac{1}{m V_g} \left[L(\sin \theta_d \sin \gamma_a \cos \Phi + \cos \theta_d \sin \Phi) - (Thr - D) \sin \theta_d \cos \gamma_a \right] \\ \dot{m} &= -\eta Thr\end{aligned}\right.\tag{G.81}$$

Dans un plan

$$\begin{aligned}\vec{U} &= [c, \gamma_a, \Phi] \\ \vec{X} &= [x, y, H_p, V_a, \Psi_i, m]\end{aligned}\tag{G.82}$$

$$\left\{ \begin{aligned}\dot{x} &= V_g \cos \Psi_i \\ \dot{y} &= V_g \sin \Psi_i \\ \frac{g_0}{g} \left(\frac{T}{T_{ISA}} \right) \dot{H}_p &= V_a \sin \gamma_a + W_{Up} \\ V_a \dot{V}_a + g_0 \left(\frac{T}{T_{ISA}} \right) \dot{H}_p &= \left(\frac{Thr - D}{m} \right) V_a - \dot{\vec{W}} \cdot \vec{V}_a + g W_{Up} \\ \dot{\Psi}_i &= \frac{1}{m V_g} \left[L(\sin \theta_d \sin \gamma_a \cos \Phi + \cos \theta_d \sin \Phi) - (Thr - D) \sin \theta_d \cos \gamma_a \right] \\ \dot{m} &= -\eta Thr\end{aligned}\right.\tag{G.83}$$

Rappel des hypothèses principales

Dans toutes les équations de cette section, on considère que :

- toutes les forces s'appliquent au centre de masse,
- la poussée des moteurs s'exerce dans l'axe de la vitesse air,
- la composante latérale de la force aérodynamique est négligeable,
- les effets de la rotation de la terre sur l'accélération de l'avion sont négligeables.

Pour les équations impliquant l'altitude-pression H_p , on fait l'hypothèse supplémentaire que le gradient de température est une fonction linéaire par morceaux de l'altitude-pression, avec la même loi qu'en atmosphère standard ISA, mais avec des conditions initiales éventuellement différentes à l'isobare de référence (écart ΔT).

G.5 Calcul de la trajectoire

Pour un état initial \vec{X}_0 et une loi de commande $\vec{U}(t)$ donnés, on calcule la trajectoire de l'avion en intégrant les équations du mouvement entre t_0 et un temps final t_f .

Les équations développées dans les sections G.4.2 (ou G.4.3, avec un vecteur d'état différent) forment un système d'équations différentielles et algébriques dont la résolution analytique est impraticable.

On utilise donc des techniques de calcul numérique, dont la plus simple est la méthode d'Euler. Dans cette méthode, le calcul est fait itérativement avec un pas de temps δt , en partant de $t = t_0$. L'état $\vec{X}(t)$ étant connu à l'instant t , on calcule $\dot{\vec{X}}(t) = \frac{d\vec{X}}{dt}$ à l'instant t pour en déduire l'état suivant $\vec{X}(t + \delta t)$.

$$\begin{cases} \vec{X}(t_0) = \vec{X}_0 \\ \dot{\vec{X}}(t) = F(\vec{X}(t), \vec{U}(t)) \\ \vec{X}(t + \delta t) = \vec{X}(t) + \dot{\vec{X}}(t)\delta t \end{cases} \quad (\text{G.84})$$

Un calcul plus précis peut être obtenu par une méthode de Runge Kutta, où la dérivée $\dot{\vec{X}}$ est estimée plus précisément en faisant une somme pondérée de valeurs prises à des points intermédiaires entre t et $t + \delta t$.

Notons par ailleurs que les équations différentielles peuvent ne pas se présenter sous la forme $\dot{\vec{X}}(t) = F(\vec{X}(t), \vec{U}(t))$, mais sous une forme plus générale exprimant une relation fonctionnelle entre les différentes grandeurs (G.85) :

$$F'(\dot{\vec{X}}(t), \vec{X}(t), \vec{U}(t)) = 0 \quad (\text{G.85})$$

C'est typiquement le cas lorsqu'on exprime l'effet du vent en fonction des dérivées \dot{H}_p et \dot{V}_a qui apparaissent aussi dans la dérivée du vecteur d'état $\dot{\vec{X}}(t)$ (voir équations (G.74) et (G.75)). Dans ce cas, l'intégration du système d'équations différentielles peut nécessiter la résolution d'un système d'équations reliant les différentes variables.

Une autre stratégie de calcul, plutôt que de résoudre le système évoqué, consiste à exprimer les équations du mouvement sous la forme (G.84) et à approximer les quantités \dot{H}_p et \dot{V}_a présents dans les termes de droite représentant l'effet du vent par leur valeurs à $t - 1$. Dans ce cas, on néglige l'effet du vent pour le premier point à t_0 .

Bibliographie

- [1] *Study of the acquisition of data from aircraft operators to aid trajectory prediction calculation*, tech. rep., EUROCONTROL Experimental Center, 1998. *Cité p.* 170
- [2] *Adapt2. aircraft data aiming at predicting the trajectory. data analysis report*, tech. rep., EUROCONTROL Experimental Center, 2009. *Cité p.* 170
- [3] *Base of aircraft data (BADA) aircraft performance modelling report*, tech. rep., EUROCONTROL, 2009. *Cité p.* 165
- [4] *Use of barometric altitude and geometric altitude information in ADS-B message for ATC applications*, tech. rep., International Civil Aviation Organization (ICAO), 2012. *Cité p.* 249
- [5] U. AHLSTROM, *Subjective workload ratings and eye movement activity measures*, technical report DOT/FAA/CT- 05/32, Federal Aviation Administration William J. Hughes Technical Center, Atlantic City, December 2005. *Cité p.* 139
- [6] H. AKAIKE, *Information theory and an extension of the maximum likelihood principle*, in Selected Papers of Hirotugu Akaike, Springer, 1998, pp. 199–213. *Cité p.* 84, 215, 221, 223
- [7] R. ALLIGIER, *Apprentissage artificiel appliqué à la prévision de trajectoire d’avion*, PhD thesis, Université de Toulouse, 2014. *Cité p.* 154, 155, 156, 160, 162, 171, 179, 189, 192, 196, 201
- [8] R. ALLIGIER, D. GIANAZZA, AND N. DURAND, *Energy Rate Prediction Using an Equivalent Thrust Setting Profile (regular paper)*, in International Conference on Research in Air Transportation (ICRAT), Berkeley, California, 22/05/12-25/05/12, ICRAT, may 2012. *Cité p.* 154, 156, 157, 158
- [9] R. ALLIGIER, D. GIANAZZA, AND N. DURAND, *Ground-based estimation of aircraft mass, adaptive vs. least squares method*, in Proceedings of the 10th USA/Europe Air Traffic Management R & D Seminar, 2013. *Cité p.* 154, 156, 157, 158, 178, 179
- [10] R. ALLIGIER, D. GIANAZZA, AND N. DURAND, *Learning the aircraft mass and thrust to improve the ground-based trajectory prediction of climbing flights*, Transportation Research Part C : Emerging Technologies, (2013). *Cité p.* 154, 157, 158
- [11] R. ALLIGIER, D. GIANAZZA, AND N. DURAND, *Machine learning and mass estimation methods for ground-based aircraft climb prediction*, IEEE Transactions on Intelligent Transportation Systems, PP (2015), pp. 1–12. *Cité p.* 154, 155, 159, 160
- [12] R. ALLIGIER, D. GIANAZZA, AND N. DURAND, *Machine learning applied to airspeed prediction during climb*, in Proceedings of the 11th USA/Europe Air Traffic Management R & D Seminar, 2015. *Cité p.* 154, 155, 160

- [13] R. ALLIGIER, D. GIANAZZA, M. G. HAMED, AND N. DURAND, *Comparison of Two Ground-based Mass Estimation Methods on Real Data (regular paper)*, in International Conference on Research in Air Transportation (ICRAT), Istanbul, 26/05/2014-30/05/2014, <http://www.icrat.org>, May 2014, ICRAT, p. (on line). *Cité p.* 154, 156, 157, 179
- [14] C. ALLIGNOL, N. BARNIER, N. DURAND, AND J.-M. ALLIOT, *A new framework for solving en-routes conflicts*, in ATM 2013, 10th USA/Europe Air Traffic Management Research and Development Seminar, 2013, pp. pp-1. *Cité p.* 52
- [15] J. ALLIOT, S. AUPETIT, S. B. HAMIDA, I. BOUSSAID, G. CAPOROSSI, M. CLERC, L. DEROUSI, N. DURAND, D. GIANAZZA, J. GOTTELAND, N. GRANGEON, N. MONMARCHÉ, S. NORRE, A. PÉROWSKI, C. PRODHON, C. PRINS, P. SIARRY, M. SLIMANE, E. TAILLARD, C. VANARET, AND M. VASQUEZ, *Métaheuristiques : Recruits simulé, recherche avec tabous, recherche à voisinages variables, méthodes GRASP, algorithmes évolutionnaires, fournis artificielles, essais particuliers et autres méthodes d'optimisation*, Editions Eyrolles, 2014. Sous la direction de P. Siarry. *Cité p.* 28, 32, 47, 51
- [16] J. ALLIOT, H. GRUBER, AND M. SCHOENAUER, *Using genetic algorithms for solving ATC conflicts*, in Proceedings of the Ninth Conference on Artificial Intelligence Application, IEEE, 1993. *Cité p.* 170
- [17] J.-M. ALLIOT, N. DURAND, D. GIANAZZA, AND J.-B. GOTTELAND, *Finding and proving the optimum : Cooperative stochastic and deterministic search*, in Proceedings of the European Conference on Artificial Intelligence (ECAI 2012), 2012. *Cité p.* 28, 33, 51
- [18] J.-M. ALLIOT, N. DURAND, D. GIANAZZA, AND J.-B. GOTTELAND, *Implementing an interval computation library for ocaml on x86/amd64 architectures (short paper)*, in International Conference on Functional Programming, Copenhagen, Denmark, ACM SIGPLAN, 2012. *Cité p.* 28, 51
- [19] L. ARMIJO ET AL., *Minimization of functions having lipschitz continuous first partial derivatives*, Pacific Journal of mathematics, 16 (1966), pp. 1-3. *Cité p.* 46
- [20] S. ATHÈNES, P. AVERTY, S. PUECHMOREL, D. DELAHAYE, AND C. COLLET, *Atc complexity and controller workload : Trying to bridge the gap*, in Proceedings of the International Conference on HCI in Aeronautics, 2002, pp. 56-60. *Cité p.* 139
- [21] Y. BENGIO, A. COURVILLE, AND P. VINCENT, *Representation learning : A review and new perspectives*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35 (2013), pp. 1798-1828. *Cité p.* 118
- [22] H.-G. BEYER AND H.-P. SCHWEFEL, *Evolution strategies—a comprehensive introduction*, Natural computing, 1 (2002), pp. 3-52. *Cité p.* 46
- [23] C. M. BISHOP, *Neural networks for pattern recognition*, Oxford University Press, 1996. ISBN : 0-198-53864-2. *Cité p.* 91, 110, 118, 141
- [24] C. M. BISHOP ET AL., *Pattern recognition and machine learning*, vol. 4, springer New York, 2006. *Cité p.* 58, 60, 72, 73, 79, 80, 90, 91, 118, 216, 224
- [25] C. BLUM, J. PUCHINGER, G. R. RAIDL, AND A. ROLI, *Hybrid metaheuristics in combinatorial optimization : A survey*, Applied Soft Computing, 11 (2011), pp. 4135-4151. *Cité p.* 51

- [26] C. BLUM, A. ROLI, AND M. SAMPELS, *Hybrid metaheuristics : an emerging approach to optimization*, vol. 114, Springer Science & Business Media, 2008. *Cité p.* 51
- [27] C. BLUM AND K. SOCHA, *Training feed-forward neural networks with ant colony optimization : an application to pattern classification*, in Fifth International conference on Hybrid Intelligent Systems, 2005. *Cité p.* 118
- [28] O. BOUSQUET, S. BOUCHERON, AND G. LUGOSI, *Introduction to statistical learning theory*, in Advanced Lectures on Machine Learning, Springer, 2004, pp. 169–207. *Cité p.* 65
- [29] L. BREIMAN, J. FRIEDMAN, C. J. STONE, AND R. A. OLSHEN, *Classification and regression trees*, CRC press, 1984. *Cité p.* 88, 120
- [30] S. CAFIERI, P. BRISSET, AND N. DURAND, *A mixed-integer optimization model for air traffic deconfliction*, in Proceedings of the Toulouse Global Optimization Workshop, 2010, pp. 27–30. *Cité p.* 52
- [31] M. CAPDEROU, *Satellites : de Kepler au GPS*, Springer, 2011. *Cité p.* 229
- [32] G. CHALOULOS, E. CRÜCK, AND J. LYGEROS, *A simulation based study of subliminal control for air traffic management*, Transportation Research Part C, 18 (2010), pp. 963–974. *Cité p.* 170
- [33] G. CHATTERJI AND B. SRIDHAR, *Measures for air traffic controller workload prediction*, in Proceedings of the First AIAA Aircraft Technology, Integration, and Operations Forum, 2001. *Cité p.* 139
- [34] G. B. CHATTERJI AND B. SRIDHAR, *Neural network based air traffic controller workload prediction*, in American Control Conference, 1999. Proceedings of the 1999, vol. 4, IEEE, 1999, pp. 2620–2624. *Cité p.* 139
- [35] H. CHERNOFF, *On the distribution of the likelihood ratio*, The Annals of Mathematical Statistics, (1954), pp. 573–578. *Cité p.*
- [36] M. CLERC AND J. KENNEDY, *The particle swarm-explosion, stability, and convergence in a multidimensional complex space*, Evolutionary Computation, IEEE Transactions on, 6 (2002), pp. 58–73. *Cité p.* 49
- [37] C. A. C. COELLO AND A. CARLOS, *A survey of constraint handling techniques used with evolutionary algorithms*, Lania-RI-99-04, Laboratorio Nacional de Informática Avanzada, (1999). *Cité p.* 50
- [38] S. CONSORTIUM, *Milestone Deliverable D3 : The ATM Target Concept*, tech. rep., 2007. *Cité p.* 24, 163, 169
- [39] R. A. COPPENBARGER, *Climb trajectory prediction enhancement using airline flight-planning information*, in AIAA Guidance, Navigation, and Control Conference, 1999. *Cité p.* 170
- [40] A. CORNUÉJOLS AND L. MICLET, *Apprentissage artificiel : concepts et algorithmes*, Editions Eyrolles, 2011. *Cité p.* 58, 60, 74, 80, 138
- [41] S. DE HAAN AND A. STOFFELEN, *Assimilation of high-resolution Mode-S wind and temperature observations in a regional NWP model for nowcasting applications*, Weather and Forecasting, 27 (2012), pp. 918–937. *Cité p.* 156, 169
- [42] D. DELAHAYE AND S. PUECHMOREL, *Tas and wind estimation from radar data*, in Digital Avionics Systems Conference, 2009. DASC'09. IEEE/AIAA 28th, IEEE, 2009, pp. 2–B. *Cité p.* 163

- [43] J. E. DENNIS, JR AND J. J. MORÉ, *Quasi-newton methods, motivation and theory*, SIAM review, 19 (1977), pp. 46–89. *Cité p.* 210
- [44] D. J. DISTON, *Computational Modelling and Simulation of Aircraft and the Environment : Volume 1-Platform Kinematics and Synthetic Environment*, John Wiley & Sons, 2009. *Cité p.* 229
- [45] J. DRÉO, A. PÉTROWSKI, P. SIARRY, AND E. TAILLARD, *Métaheuristiques pour l'optimisation difficile*, Eyrolles, 2003. ISBN : 2-212-11368-4. *Cité p.* 47
- [46] DSNA, *Rapport d'activité de la direction des services de la navigation aérienne*, 2012. *Cité p.* 14
- [47] R. A. DUNNE AND N. A. CAMPBELL, *On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function*, in Proceedings of the Eighth Australasian Conference on Neural Networks, 1997. *Cité p.* 141
- [48] N. DURAND, *Optimisation de trajectoires pour la résolution de conflits aériens en route*, PhD thesis, Institut National Polytechnique de Toulouse, 1996. *Cité p.* 170
- [49] N. DURAND AND J. ALLIOT, *Un algorithme de colonie de fourmis pour résoudre des conflits aériens*, in ROADEF, 2010. *Cité p.* 52
- [50] N. DURAND, J. ALLIOT, AND G. GRANGER, *A statistical analysis of the influence of vertical and ground speed errors on conflict probe*, in Proceedings of ATM2001, 2001. *Cité p.* 163
- [51] N. DURAND, J. ALLIOT, AND J. NOAILLES, *Automatic aircraft conflict resolution using genetic algorithms*, in Proceedings of the Symposium on Applied Computing, Philadelphia, ACM, 1996. *Cité p.* 52
- [52] N. DURAND AND J.-M. ALLIOT, *Ant colony optimization for air traffic conflict resolution*, in 8th USA/Europe Air Traffic Management Research and Development Seminar, 2009. *Cité p.* 52, 170
- [53] R. C. EBERHART AND J. KENNEDY, *A new optimizer using particle swarm theory*, in Proceedings of the sixth international symposium on micro machine and human science, vol. 1, New York, NY, 1995, pp. 39–43. *Cité p.* 48
- [54] A. EDELMAN AND H. MURAKAMI, *Polynomial roots from companion matrix eigenvalues*, Mathematics of Computation, 64 (1995), pp. 763–776. *Cité p.* 178
- [55] A. EIBEN AND J. SMITH, *Introduction to Evolutionary Computing*, Springer, 2003. ISBN : 3-540-40184-9. *Cité p.* 47, 48, 132
- [56] Y. L. FABLEC, *Prévision de trajectoires d'avions par réseaux de neurones*, PhD thesis, Institut National Polytechnique de Toulouse, 1999. *Cité p.* 171
- [57] G. FLYNN, A. BENKOUAR, AND R. CHRISTIEN, *Adaptation of workload model by optimisation algorithms*, tech. rep., Eurocontrol, 2005. *Cité p.* 138
- [58] J. FOX, *Applied regression analysis, linear models, and related methods.*, Sage Publications, Inc, 1997. *Cité p.* 92, 99
- [59] J. H. FRIEDMAN, *Greedy function approximation : a gradient boosting machine*, Annals of statistics, (2001), pp. 1189–1232. *Cité p.* 88, 120
- [60] J. H. FRIEDMAN, *Stochastic gradient boosting*, Computational Statistics & Data Analysis, 38 (2002), pp. 367–378. *Cité p.* 120

- [61] M. GHASEMI HAMED, *Méthodes non-paramétriques pour la prévision d'intervalles avec haut niveau de confiance : application à la prévision de trajectoires d'avions*, PhD thesis, Thèse doctorat informatique de l'INPT, 2014. *Cité p.* 171
- [62] D. GIANAZZA, *Algorithme évolutionnaire et A* pour la séparation en 3D des flux de trafic aérien*, Journal Européen des Systèmes automatisés, numéro spécial "Métaheuristiques pour l'optimisation difficile", 38 (2004), pp. 1065–1095. *Cité p.* 51
- [63] D. GIANAZZA, *Airspace configuration using air traffic complexity metrics*, in Proceedings of the 7th USA/Europe Seminar on Air Traffic Management Research and Development, 2007. best paper of "Dynamic Airspace Configuration" track. *Cité p.* 126, 136, 137, 138
- [64] D. GIANAZZA, *Smoothed traffic complexity metrics for airspace configuration schedules*, in Proceedings of the 3rd International Conference on Research in Air Transportation, ICRAAT, 2008. *Cité p.* 126, 136, 137, 138, 141
- [65] D. GIANAZZA, *Forecasting workload and airspace configuration with neural networks and tree search methods*, Artificial Intelligence Journal, Elsevier, 174 (2010), pp. 530–549. *Cité p.* 126, 144
- [66] D. GIANAZZA, C. ALLIGNOL, AND N. SAPORITO, *An efficient airspace configuration forecast*, in Proceedings of the 8th USA/Europe Air Traffic Management R & D Seminar, 2009. *Cité p.* 126, 144
- [67] D. GIANAZZA AND J. M. ALLIOT, *Optimization of air traffic control sector configurations using tree search methods and genetic algorithms*, in Proceedings of the 21st Digital Avionics Systems Conference, 2002. *Cité p.* 126, 130, 131, 132, 134
- [68] D. GIANAZZA, J. M. ALLIOT, AND G. GRANGER, *Optimal combinations of air traffic control sectors using classical and stochastic methods*, in Proceedings of the 2002 International Conference on Artificial Intelligence, 2002. *Cité p.* 126, 130, 131, 132, 134
- [69] D. GIANAZZA AND N. DURAND, *Separating air traffic flows by allocating 3d-trajectories*, in Proceedings of the 23rd Digital Avionics Systems Conference, 2004. *Cité p.* 51
- [70] D. GIANAZZA AND N. DURAND, *Assessment of the 3d-separation of air traffic flows*, in Proceedings of the 6th USA/Europe Seminar on Air Traffic Management Research and Development, 2005. *Cité p.* 51
- [71] D. GIANAZZA, N. DURAND, AND N. ARCHAMBAULT, *Allocating 3d-trajectories to air traffic flows, using a* and genetic algorithms*, in Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, 2004. *Cité p.* 51
- [72] D. GIANAZZA AND K. GUITTET, *Evaluation of air traffic complexity metrics using neural networks and sector status*, in Proceedings of the 2nd International Conference on Research in Air Transportation, ICRAAT, 2006. *Cité p.* 126, 136, 137, 138, 141
- [73] D. GIANAZZA AND K. GUITTET, *Selection and evaluation of air traffic complexity metrics*, in Proceedings of the 25th Digital Avionics Systems Conference, DASC, 2006. *Cité p.* 126, 136, 137, 138, 141
- [74] D. GIANAZZA AND K. GUITTET, *Réseaux de neurones appliqués aux indicateurs de complexité et aux regroupements de secteurs aériens*, note NR06-517, DSN/DTI/SDER, Avril 2006. *Cité p.* 126

- [75] J. C. GILBERT AND C. LEMARÉCHAL, *Some numerical experiments with variable-storage quasi-newton algorithms*, Mathematical programming, 45 (1989), pp. 407–435. *Cité p.* 211
- [76] F. GLOVER, *Future paths for integer programming and links to artificial intelligence*, Computers & operations research, 13 (1986), pp. 533–549. *Cité p.* 46
- [77] F. GLOVER, *Tabu search : A tutorial*, Interfaces, 20 (1990), pp. 74–94. *Cité p.* 46
- [78] D. GOLDBERG, *Genetic Algorithms*, Addison Wesley, 1989. ISBN : 0-201-15767-5. *Cité p.* 131, 132
- [79] D. GOLDBERG, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989. *Cité p.* 46, 47
- [80] V. GUDISE AND G. VENAYAGAMOORTHY, *Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks*, in Proceedings of the 2003 IEEE Swarm Intelligence Symposium, 2003. *Cité p.* 118
- [81] K. GUITTET AND D. GIANAZZA, *Analyse descriptive des indicateurs de complexité du trafic aérien à partir des données IMAGE et COURAGE*, note NR05-905, DSN/DTI/SDER, Décembre 2005. *Cité p.* 126
- [82] E. HANSEN AND G. W. WALSTER, *Global optimization using interval analysis : revised and expanded*, vol. 264, CRC Press, 2003. *Cité p.* 39, 51
- [83] E. R. HANSEN AND R. I. GREENBERG, *An interval newton method*, Applied Mathematics and Computation, 12 (1983), pp. 89–98. *Cité p.* 179
- [84] P. E. HART, N. J. NILSSON, AND B. RAPHAEL, *A formal basis for the heuristic determination of minimum cost paths*, Systems Science and Cybernetics, IEEE Transactions on, 4 (1968), pp. 100–107. *Cité p.* 40, 42
- [85] P. E. HART, N. J. NILSSON, AND B. RAPHAEL, *Correction to a formal basis for the heuristic determination of minimum cost paths*, ACM SIGART Bulletin, (1972), pp. 28–29. *Cité p.* 40, 42
- [86] T. HASTIE, R. TIBSHIRANI, AND J. H. FRIEDMAN, *The Elements of Statistical Learning*, Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2001. *Cité p.* 58, 60, 72, 79, 80, 83, 86, 119, 124, 185
- [87] B. HILBURN, *Cognitive complexity in air traffic control, a litterature review*, tech. rep., Eurocontrol experimental centre, 2004. *Cité p.* 18, 139
- [88] J.-B. HIRIART-URRUTY, *Conditions for global optimality*, in Handbook of global optimization, Springer, 1995, pp. 1–26. *Cité p.* 31
- [89] S. HOCHREITER, Y. BENGIO, P. FRASCONI, AND J. SCHMIDHUBER, *Gradient flow in recurrent nets : the difficulty of learning long-term dependencies*, 2001. *Cité p.* 118
- [90] J. HOLLAND, *Adaptation in Natural and Artificial Systems*, University of Michigan press, 1975. *Cité p.* 46, 47
- [91] W. HOLLISTER, E. BRADFORD, AND J. WELCH, *Using aircraft radar tracks to estimate wind aloft*, The Lincoln Laboratory Journal, 2 (1989), pp. 555–565. *Cité p.* 163
- [92] R. HORST AND H. TUY, *Global optimization : Deterministic approaches*, Springer Science & Business Media, 1996. *Cité p.* 32

- [93] C. HURTER, R. ALLIGIER, D. GIANAZZA, S. PUECHMOREL, G. ANDRIENKO, AND N. ANDRIENKO, *Wind parameters extraction from aircraft trajectories*, Computers, Environment and Urban Systems, 47 (2014), pp. 28–43. *Cité p.* 154, 155, 156, 166, 167
- [94] C. HURTER, S. CONVERSY, D. GIANAZZA, AND A. TELEA, *Interactive image-based information visualization for aircraft trajectory analysis*, Transportation Research Part C : Emerging Technologies, (2014). *Cité p.* 154, 155, 156
- [95] INTERNATIONAL CIVIL AVIATION ORGANIZATION, *Manual of the ICAO standard atmosphere, Third Edition*, 1993. *Cité p.* 241
- [96] INTERNATIONAL CIVIL AVIATION ORGANIZATION, *Annex 11 to the Convention on International Civil Aviation, Air Traffic Services, Thirteenth Edition*, 2001. *Cité p.* 4, 17
- [97] INTERNATIONAL CIVIL AVIATION ORGANIZATION, *Manual on Air Traffic Management System Requirement, Doc 9882, First Edition*, 2008. *Cité p.* 4, 17
- [98] L. JOURDAN, M. BASSEUR, AND E.-G. TALBI, *Hybridizing exact methods and metaheuristics : A taxonomy*, European Journal of Operational Research, 199 (2009), pp. 620–629. *Cité p.* 51
- [99] J. KENNEDY, *Particle swarm optimization*, in Encyclopedia of Machine Learning, Springer, 2010, pp. 760–766. *Cité p.* 48
- [100] S. KIRKPATRICK, C. GELATT, AND M. VECCHI, *Optimization by simulated annealing*, Science, 220 (1983), pp. 671–680. *Cité p.* 46
- [101] R. KOHAVI ET AL., *A study of cross-validation and bootstrap for accuracy estimation and model selection*, in Ijcai, vol. 14, 1995, pp. 1137–1145. *Cité p.* 83
- [102] P. KOPARDEKAR, *Dynamic density : A review of proposed variables*, FAA WJHTC internal document. overall conclusions and recommendations, Federal Aviation Administration, 2000. *Cité p.* 139
- [103] P. KOPARDEKAR AND S. MAGYARITS, *Measurement and prediction of dynamic density*, in Proceedings of the 5th USA/Europe Air Traffic Management R & D Seminar, 2003. *Cité p.* 139
- [104] J. K. KUCHAR AND L. C. YANG, *A review of conflict detection and resolution modeling methods*, Intelligent Transportation Systems, IEEE Transactions on, 1 (2000), pp. 179–189. *Cité p.* 170
- [105] S. KULLBACK AND R. A. LEIBLER, *On information and sufficiency*, The annals of mathematical statistics, (1951), pp. 79–86. *Cité p.* 216
- [106] I. V. LAUDEMAN, S. G. SHELDEN, R. BRANSTROM, AND C. L. BRASIL, *Dynamic density : An air traffic management metric*, tech. rep., 1999. *Cité p.* 139
- [107] E. LEBARBIER AND T. MARY-HUARD, *Le critère bic : fondements théoriques et interprétation. rapport de recherche n 5315*, 2004. *Cité p.* 224
- [108] F. LEUNG, H. LAM, S. LING, AND P. TAM, *Tuning of the structure and parameters of a neural network using an improved genetic algorithm*, IEEE transactions on Neural Networks, 14 (2003), pp. 79–88. *Cité p.* 118
- [109] D. C. LIU AND J. NOCEDAL, *On the limited memory bfgs method for large scale optimization*, Mathematical programming, 45 (1989), pp. 503–528. *Cité p.* 212

- [110] D. G. LUENBERGER AND Y. YE, *Linear and nonlinear programming*, vol. 116, Springer Science & Business Media, 2008. *Cité p.* 32, 44, 46, 210
- [111] I. LYMPEROPOULOS AND J. LYGEROS, *Sequential monte carlo methods for multi-aircraft trajectory prediction in air traffic management*, International Journal of Adaptive Control and Signal Processing, 24 (2010), pp. 830–849. *Cité p.* 163
- [112] I. LYMPEROPOULOS, J. LYGEROS, AND A. L. VISINTINI, *Model based aircraft trajectory prediction during takeoff*, in AIAA Guidance, Navigation and Control Conference and Exhibit, Keystone, Colorado, aug 2006. *Cité p.* 171
- [113] A. MAJUMDAR AND W. Y. OCHIENG, *Factors affecting air traffic controller workload : Multivariate analysis based on simulation modeling of controller workload*, Transportation Research Record : Journal of the Transportation Research Board, 1788 (2002), pp. 58–69. *Cité p.* 18
- [114] C. A. MANNING, S. H. MILLS, C. M. FOX, E. M. PFLEIDERER, AND H. J. MORGILKA, *Using air traffic control taskload measures and communication events to predict subjective workload*, tech. rep., DTIC Document, 2002. *Cité p.* 139
- [115] A. J. MASALONIS, M. B. CALLAHAM, AND C. R. WANKE, *Dynamic density and complexity metrics for realtime traffic flow management*, in Proceedings of the 5th USA/Europe Air Traffic Management R & D Seminar, 2003. *Cité p.* 139
- [116] P. MASSART, *Concentration inequalities and model selection*, vol. 10, Springer, 2007. *Cité p.* 222
- [117] W. S. MCCULLOCH AND W. PITTS, *A logical calculus of the ideas immanent in nervous activity*, The bulletin of mathematical biophysics, 5 (1943), pp. 115–133. *Cité p.* 101
- [118] Z. MICHALEWICZ, *Genetic algorithms+data structures=evolution programs*, Springer-Verlag, 1992. ISBN : 0-387-55387-. *Cité p.* 131
- [119] Z. MICHALEWICZ, D. DASGUPTA, R. G. LE RICHE, AND M. SCHOENAUER, *Evolutionary algorithms for constrained engineering problems*, Computers & Industrial Engineering, 30 (1996), pp. 851–870. *Cité p.* 50
- [120] M. MINOUX, *Programmation mathématique : théorie et algorithmes*, vol. 1, Dunod Paris, 1983. *Cité p.* 32, 46
- [121] M. L. MINSKY AND S. A. PAPERT, *Perceptrons - Expanded Edition : An Introduction to Computational Geometry*, MIT press Boston, MA :, 1987. *Cité p.* 95, 101
- [122] R. MOGFORD, J. A. GUTTMAN, S. L. MORROW, AND P. KOPARDEKAR, *The complexity construct in air traffic control : A review and synthesis of the literature*, tech. rep., FAA Technical Center : Atlantic City, 1995. *Cité p.* 18, 139
- [123] R. E. MOORE, *Interval Analysis*, Prentice-Hall, 1966. *Cité p.* 179
- [124] J. J. MORÉ AND D. J. THUENTE, *Line search algorithms with guaranteed sufficient decrease*, ACM Transactions on Mathematical Software (TOMS), 20 (1994), pp. 286–307. *Cité p.* 46
- [125] A.-. NAS ADVANCED CONCEPTS BRANCH, *An evaluation of dynamic density metrics using RAMS*, technical report (draft) DOT/FAA/CT-TN, Federal Aviation Administration, April 2001. *Cité p.* 139
- [126] J. NOCEDAL, *Updating quasi-newton matrices with limited storage*, Mathematics of computation, 35 (1980), pp. 773–782. *Cité p.* 212

- [127] J. NOCEDAL AND S. WRIGHT, *Numerical Optimization*, Springer series in operations research and financial engineering, Springer Science+Business Media, LLC., 2006. *Cité p.* 31, 32, 43, 44, 46, 210, 211
- [128] A. NUIC, *User manual for base of aircraft data (bada) rev.3.9*, tech. rep., EUROCONTROL, avril 2011. *Cité p.* 251, 252, 277
- [129] A. NUIC, *User manual for base of aircraft data (bada) revision 3.10*, tech. rep., EUROCONTROL, avril 2012. *Cité p.* 173, 177, 188, 196
- [130] L. PALLOTTINO, E. M. FERON, AND A. BICCHI, *Conflict resolution problems for air traffic management systems solved with mixed integer programming*, Intelligent Transportation Systems, IEEE Transactions on, 3 (2002), pp. 3–11. *Cité p.* 52, 170
- [131] K. PEARSON, *Liii. on lines and planes of closest fit to systems of points in space*, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2 (1901), pp. 559–572. *Cité p.* 89
- [132] D. POLES, *Revision of atmosphere model in bada aircraft performance model*, tech. rep., EUROCONTROL, 2010. *Cité p.* 241
- [133] R. POLI, J. KENNEDY, AND T. BLACKWELL, *Particle swarm optimization*, Swarm intelligence, 1 (2007), pp. 33–57. *Cité p.* 49
- [134] X. PRATS, V. PUIG, J. QUEVEDO, AND F. NEJJARI, *Multi-objective optimisation for aircraft departure trajectories minimising noise annoyance*, Transportation Research Part C, 18 (2010), pp. 975–989. *Cité p.* 170
- [135] K. PRICE, R. M. STORN, AND J. A. LAMPINEN, *Differential evolution : a practical approach to global optimization*, Springer Science & Business Media, 2006. *Cité p.* 50
- [136] G. R. RAIDL, *A unified view on hybrid metaheuristics*, in Hybrid Metaheuristics, Springer, 2006, pp. 1–12. *Cité p.* 51
- [137] B. D. RIPLEY, *Pattern recognition and neural networks*, Cambridge University Press, 1996. ISBN : 0-521-46086-7. *Cité p.* 118, 122, 141, 224
- [138] F. ROSENBLATT, *The perceptron : a probabilistic model for information storage and organization in the brain.*, Psychological review, 65 (1958), p. 386. *Cité p.* 95, 101
- [139] D. E. RUMELHART, G. E. HINTON, AND R. J. WILLIAMS, *Learning internal representations by error propagation*, tech. rep., University of California, Institute for Cognitive Science, San Diego, September 1985. *Cité p.* 117
- [140] S. RUSSELL AND P. NORVIG, *Artificial intelligence : A modern approach author*, Prentice Hall, 2009. *Cité p.* 33, 42
- [141] N. SAPORITO, C. HURTER, D. GIANAZZA, AND G. BEBOUX, *A participatory design for the visualization of airspace configuration forecasts*, in Proceedings of the 4th International Conference on Research in Air Transportation, 2010. *Cité p.* 126, 144
- [142] M. SCHOENAUER AND Z. MICHALEWICZ, *Evolutionary algorithms for constrained parameter optimization problems*, in Evolutionary Computation, 1996. *Cité p.* 50
- [143] M. SCHOENAUER AND Z. MICHALEWICZ, *Boundary operators for constrained parameter optimization problems*, in Proceedings of the Seventh International Conference on Genetic Algorithms, ICGA, 1997. *Cité p.* 50

- [144] M. SCHOENAUER AND Z. MICHALEWICZ, *Sphere operators and their applicability for constrained parameter optimization problems*, in Evolutionary Programming VII, 1998. *Cité p.* 50
- [145] C. SCHULTZ, D. THIPPHAVONG, AND H. ERZBERGER, *Adaptive trajectory prediction algorithm for climbing flights*, in AIAA Guidance, Navigation, and Control (GNC) Conference, August 2012. *Cité p.* 157, 171, 179, 196
- [146] G. SCHWARZ ET AL., *Estimating the dimension of a model*, The annals of statistics, 6 (1978), pp. 461–464. *Cité p.* 85, 215, 224
- [147] S. G. SELF AND K.-Y. LIANG, *Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions*, Journal of the American Statistical Association, 82 (1987), pp. 605–610. *Cité p.*
- [148] SERVICE DE L'INFORMATION AÉRONAUTIQUE, *Annexe 2 à l'arrêté du 3 mars 2006 modifié (Services de la circulation aérienne)*, décembre 2006. *Cité p.* 5, 7
- [149] SERVICE DE L'INFORMATION AÉRONAUTIQUE, *Arrêté du 6 juillet 1992 relatif aux procédures pour les organismes rendant les services de la circulation aérienne aux aéronefs de la circulation aérienne générale (RCA/3), modifié par l'arrêté du 29 octobre 2013*, Octobre 2013. *Cité p.* 4, 9
- [150] Y. SHI AND R. EBERHART, *A modified particle swarm optimizer*, in Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, IEEE, 1998, pp. 69–73. *Cité p.* 48
- [151] G. L. SLATER, *Adaptive improvement of aircraft climb performance for air traffic control applications*, in Proceedings of the 2002 IEEE International Symposium on Intelligent Control, IEEE conference publications, October 2002. *Cité p.* 171
- [152] A. SLOWIK AND M. BIALKO, *Training of artificial neural networks using differential evolution algorithm*, in Conference on Human System Interactions, 2008. *Cité p.* 118
- [153] B. SRIDHAR, K. S. SHETH, AND S. GRABBE, *Airspace complexity and its application in air traffic management*, in Proceedings of the 2nd USA/Europe Air traffic Management R&D Seminar. *Cité p.* 139
- [154] M. STONE, *An asymptotic equivalence of choice of model by cross-validation and akaike's criterion*, Journal of the Royal Statistical Society. Series B (Methodological), (1977), pp. 44–47. *Cité p.* 85
- [155] R. STORN AND K. PRICE, *Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces*, Journal of global optimization, 11 (1997), pp. 341–359. *Cité p.* 49
- [156] H. SWENSON, R. BARHYDT, AND M. LANDIS, *Next Generation Air Transportation System (NGATS) Air Traffic Management (ATM)-Airspace Project*, tech. rep., National Aeronautics and Space Administration, 2006. *Cité p.* 24, 163, 169
- [157] E.-G. TALBI, *Metaheuristics : from design to implementation*, vol. 74, John Wiley & Sons, 2009. *Cité p.* 32, 47
- [158] K. TASTAMBEKOV, S. PUECHMOREL, D. DELAHAYE, AND C. RABUT, *Aircraft trajectory forecasting using local functional regression in sobolev space*, Transportation Research Part C : Emerging Technologies, 39 (2014), pp. 1 – 22. *Cité p.* 171
- [159] J. TAYLOR AND M. PLUMBLEY, *Information theory and neural networks*, 1993. *Cité p.* 141

- [160] R. TIBSHIRANI, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society. Series B (Methodological), (1996), pp. 267–288. *Cité p.* 88
- [161] M. E. TIPPING, *Bayesian inference : An introduction to principles and practice in machine learning*, in Advanced lectures on machine Learning, Springer, 2004, pp. 41–62. *Cité p.* 141
- [162] I. C. TRELEA, *The particle swarm optimization algorithm : convergence analysis and parameter selection*, Information processing letters, 85 (2003), pp. 317–325. *Cité p.* 49
- [163] P. VAN HENTENRYCK, *Numerica : a modeling language for global optimization*, in Proceedings of the Fifteenth international joint conference on Artificial intelligence - Volume 2, IJCAI'97, 1997, pp. 1642–1647. *Cité p.* 179
- [164] C. VANARET, *Hybridation d'algorithmes évolutionnaires et de méthodes d'intervalles pour l'optimisation de problèmes difficiles*, PhD thesis, Thèse doctorat informatique de l'INPT, 2015. *Cité p.* 51, 54, 55
- [165] C. VANARET, D. GIANAZZA, N. DURAND, AND J. GOTTELAND, *Benchmarking conflict resolution algorithms (regular paper)*, in International Conference on Research in Air Transportation (ICRAT), Berkeley, California, 22/05/12-25/05/12, <http://www.icrat.org>, may 2012, ICRAT, p. (on line). *Cité p.* 28, 52, 54, 170
- [166] C. VANARET, D. GIANAZZA, J. GOTTELAND, AND N. DURAND, *Résolution de conflits aériens par un algorithme à évolution différentielle*, tech. rep., 13e congrès annuel de la Société française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2012), 2012. *Cité p.* 28, 52
- [167] C. VANARET, J.-B. GOTTELAND, AND N. DURAND, *Hybridation de programmation par contraintes sur intervalles et d'algorithmes évolutionnaires pour l'optimisation globale-cooperation of evolutionary algorithms and interval constraint programming*, in ROADEF 2013, 14ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision, 2015. *Cité p.* 51
- [168] V. N. VAPNIK, *An overview of statistical learning theory*, Neural Networks, IEEE Transactions on, 10 (1999), pp. 988–999. *Cité p.* 67, 80, 101
- [169] S. A. VAVASIS, *Complexity issues in global optimization : a survey*, in Handbook of global optimization, Springer, 1995, pp. 27–41. *Cité p.* 31
- [170] C. VERLHAC AND S. MANCHON, *Optimization of opening schemes*, in Proceedings of the fourth USA/Europe Air Traffic Management R&D Seminar, 2001. *Cité p.* 127
- [171] A. WARREN, *Trajectory prediction concepts for next generation air traffic management*, in 3rd USA/Europe ATM R&D Seminar, June 2000. *Cité p.* 171
- [172] A. WARREN AND Y. EBRAHIMI, *Vertical path trajectory prediction for next generation atm*, in Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE, vol. 2, oct-7 nov 1998, pp. F11/1 –F11/8 vol.2. *Cité p.* 171
- [173] A. WEBB, D. LOWE, AND M. D. BEDWORTH, *A comparison of nonlinear optimisation strategies for feed-forward adaptive layered networks*, tech. rep., DTIC Document, 1988. *Cité p.* 117
- [174] J. D. WELCH, J. W. ANDREWS, B. D. MARTIN, AND B. SRIDHAR, *Macroscopic workload model for estimating en route sector capacity*, in Proc. of 7th USA/Europe ATM Research and Development Seminar, Barcelona, Spain, 2007. *Cité p.* 138

- [175] B. WIDROW, M. E. HOFF, ET AL., *Adaptive switching circuits.*, (1960). *Cité p.* 101
- [176] P. WOLFE, *Convergence conditions for ascent methods*, SIAM review, 11 (1969), pp. 226–235. *Cité p.* 46, 210
- [177] D. H. WOLPERT, *The supervised learning no-free-lunch theorems*, in *Soft Computing and Industry*, Springer, 2002, pp. 25–42. *Cité p.* 65