

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE DE LA PRODUCTION AUTOMATISÉE
M.Ing.

PAR
Philippe LAMONTAGNE

MODÉLISATION SPATIO-TEMPORELLE ORIENTÉE PAR PATRONS AVEC UNE
APPROCHE BASÉE SUR INDIVIDUS

MONTRÉAL, LE 24 NOVEMBRE 2009

© Philippe Lamontagne, 2009

PRÉSENTATION DU JURY
CE MÉMOIRE A ÉTÉ ÉVALUÉ
PAR UN JURY COMPOSÉ DE

M. Jacques-André Landry, directeur de mémoire
Département de génie de la production automatisée à l'École de technologie supérieure

Mme Lael Parrott, codirectrice de mémoire
Département de géographie à l'université de Montréal

M. Mohamed Cheriet, président du jury
Département de génie de la production automatisée à l'École de technologie supérieure

M. Robert Sabourin, membre du jury
Département de génie de la production automatisée à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 17 NOVEMBRE 2009

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

J'aimerais remercier en premier lieu mon directeur de recherche Jacques-André Landry et ma codirectrice Lael Parrott sans l'initiative desquels ce projet n'aurait pas été possible. J'ai pu constater dans mon incursion dans le monde de la recherche l'extraordinaire chance que j'ai eu d'être supervisé par eux. Leur ouverture d'esprit et la confiance qu'ils m'ont accordée m'ont permis de découvrir une passion dans l'enseignement et la modélisation de système spatio-temporel. Je tiens à exprimer toute ma reconnaissance pour leur dévouement, leur rigueur et la qualité des commentaires et suggestions dont ils m'ont fait part tout au long de ma maîtrise.

J'aimerais aussi remercier l'extraordinaire équipe impliquée dans ce projet multidisciplinaire, alliant l'ingénierie, l'océanographie et la géomatique. Grâce à vous, Clément Chion, Cristiane Albuquerque et Samuel Turgeon, ce projet a été possible et réalisé dans une atmosphère des plus conviviale et stimulante.

Le projet a été possible grâce aux nombreux partenaires impliqués dans ce projet : le GREMM, Parc Marin Saguenay-Saint-Laurent, Parcs Canada, Pêches et Océans Canada et l'Université de Calgary. Grâce à leur participation active et au partage de leurs données exceptionnelles, nous avons pu créer un modèle des mammifères marins en utilisant les meilleures connaissances disponibles dans le parc marin.

Enfin, j'aimerais remercier les organismes subventionnaires CRSNG, FQRNT et l'École de technologie supérieure, qui m'ont permis de me concentrer pendant deux ans sur la réalisation de ce projet.

MODÉLISATION SPATIO-TEMPORELLE ORIENTÉE PAR PATRONS AVEC UNE APPROCHE BASÉE SUR INDIVIDUS

LAMONTAGNE, Philippe

RÉSUMÉ

Nous proposons un modèle spatialement explicite avec une approche basée sur individus qui simule les déplacements spatio-temporels des rorquals communs et des bélugas durant la saison touristique dans une portion de l'estuaire du Saint-Laurent. Ce modèle est une composante d'un outil d'aide à la décision, en cours de développement par notre groupe de recherche, pour tester différents scénarios de zonage et de règlementation dans le parc marin.

Nous avons utilisé l'approche de modélisation orientée par patrons afin de reproduire les patrons de trajectoires, de répartition spatiale et de taille de groupe pour chaque mammifère. Ces patrons sont extraits d'une large base de données contenant des suivis d'individus obtenus par radio télémétrie et stations d'observation terrestre ainsi que plusieurs observations à partir de bateaux d'excursions entre 1994 et 2007 fournies par le groupe de recherche et d'éducation sur les mammifères marins. Nous avons implémenté et testé cinq différents modèles de déplacement des mammifères marins basés sur la théorie de la marche aléatoire. Pour comparer les patrons observés et simulés, nous avons utilisé la statistique de Kolmogorov-Smirnov, l'erreur moyenne absolue normalisée et la métrique d'efficacité de modèle. Cette analyse a démontré que les mouvements simulés ne reproduisent pas correctement tous les patrons extraits précédemment.

Ainsi, nous proposons un nouvel algorithme, basé sur la marche aléatoire corrélée avec une minimisation de la moyenne des biais normalisés, qui sélectionne un mouvement en trouvant un compromis entre les patrons conflictuels : angle de braquage, distribution spatiale et taille de groupe. Ce compromis est calculé par une fonction objectif pondérée qui cherche à minimiser le biais entre les patrons observés et simulés. Cet algorithme simule adéquatement les patrons de mouvement extraits de la base de données, en plus de respecter les contraintes de répartition spatiale et de taille de groupe. Cet algorithme sera intégré dans notre modèle global basé sur individus qui tente de reproduire les patrons de déplacement des rorquals communs et des bélugas en plus du trafic maritime dans l'estuaire du Saint-Laurent.

A SPATIALLY-EXPLICIT INDIVIDUAL-BASED MODEL USING THE PATTERN-ORIENTED MODELING APPROACH

LAMONTAGNE, Philippe

ABSTRACT

We present a spatially-explicit individual-based model to simulate spatiotemporal movements of fin and beluga whales during the tourist season in a part of the St. Lawrence estuary in Canada. This model is a component of a decision support system under development by our research group to test various zoning and regulatory scenarios in the marine park.

As part of this individual-based model, we used the pattern-oriented modeling approach to reproduce trajectories, spatial distribution and group size patterns of individual whales. These patterns were extracted from a large dataset of individual tracking data obtained via radio-telemetry, shore-based stations and observations from whale-watching boats between 1994 and 2007 provided by the group for research and education on marine mammals. We implemented and tested five different models of marine mammal movements based on variations of the random walk theory. To compare the observed and simulated patterns, we used the Kolmogorov-Smirnov statistic, the normalized mean absolute error and the model efficiency metric. This analysis showed that the simulated movements did not correctly reproduce all of the previously extracted patterns.

We therefore propose a new algorithm based on a correlated random walk with mean normalized bias minimization, which selects a movement by trading-off conflicting patterns: turning angle, spatial distribution and group size. This trade-off is computed by constructing a single aggregate objective function that minimizes the bias between simulated and observed patterns. This algorithm simulates individual trajectories that adequately match the movement patterns extracted from the database while at the same time meeting the constraints of spatial distribution and group size. The algorithm will be integrated into a global individual-based model aimed at reproducing the displacement patterns of fin and beluga whales as well as boat traffic in the St. Lawrence estuary.

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 ÉTAT DE L'ART.....	5
1.1 Introduction aux modèles.....	5
1.1.1 Processus de modélisation	6
1.1.2 Les utilités d'un modèle.....	7
1.2 Modélisation de systèmes écologiques complexes	9
1.2.1 L'approche basée sur individus	10
1.2.2 Représentation spatio-temporelle.....	11
1.2.3 La validation des modèles écologiques complexes	12
1.3 Approche de modélisation orientée par patrons.....	13
1.3.1 Les patrons	14
1.3.2 Les étapes de modélisation orientée par patrons.....	16
1.4 Approches de modélisation de populations animales	18
1.4.1 Représentation des individus	19
1.4.2 Facteurs influençant les individus.....	22
1.4.3 Processus décisionnel des individus	26
1.5 Conclusion	36
CHAPITRE 2 DÉMARCHE DE MODÉLISATION.....	37
2.1 Étape I : Formuler la question.....	38
2.2 Étape II : Poser les hypothèses.....	39
2.3 Étape III : Définir la structure du modèle	40
2.4 Étape IV : Implémenter le modèle	45
2.5 Étape V : Analyser le modèle	48
CHAPITRE 3 ALGORITHMES DE DÉPLACEMENT DES INDIVIDUS	51
3.1 Algorithme de marche aléatoire.....	52
3.2 Algorithme de marche aléatoire corrélée.....	53
3.3 Algorithme de marche aléatoire corrélée avec indice de résidence	55
3.4 Algorithme de marche aléatoire corrélée avec influence sociale.....	61
3.5 Algorithme de marche aléatoire corrélée avec indice de résidence et influence sociale.....	64
3.6 Algorithme de minimisation de la moyenne des biais normalisés.....	66
CHAPITRE 4 EXTRACTION ET ANALYSE DES PATRONS	72
4.1 Données de l'étude.....	72
4.2 Extraction des patrons observés.....	73
4.2.1 Patrons de trajectoire.....	74
4.2.2 Patron de répartition.....	82
4.2.3 Patron de socialisation	87
4.3 Extraction des patrons simulés.....	90

4.4	Comparaison des patrons observés et simulés	93
CHAPITRE 5 RÉSULTATS ET DISCUSSION		100
5.1	Évaluation des algorithmes de déplacement	100
5.1.1	Marche aléatoire (MA)	100
5.1.2	Marche aléatoire corrélée (MAC).....	103
5.1.3	Marche aléatoire corrélée avec indice de résidence (MAC-R).....	106
5.1.4	Marche aléatoire corrélée avec influence sociale (MAC-S).....	110
5.1.5	Marche aléatoire corrélée avec indice de résidence et influence sociale (MAC-RS).....	118
5.1.6	Marche aléatoire corrélée avec minimisation de la moyenne des biais normalisés (MAC-MMBN)	121
5.2	Temps de simulation	124
5.3	Discussion générale des approches utilisées.....	126
CONCLUSION.....		128
RECOMMANDATIONS		131
ANNEXE I DONNÉES GÉOGRAPHIQUES DISPONIBLES		133
ANNEXE II RÉPARTITION SPATIALE DES ESPÈCES EN FONCTION DES FILTRES DE PRÉTRAITEMENT		136
ANNEXE III CARTE DE RÉPARTITION DES ESPÈCES		139
ANNEXE IV ÉVALUATION DES PARAMÈTRES.....		142
ANNEXE V SCRIPTS D'EXTRACTION DES PATRONS SIMULÉS.....		153
ANNEXE VI SCRIPTS D'EXTRACTION DU PATRON DE RÉPARTITION		160
ANNEXE VII SCRIPTS DE COMPILATION DES PATRONS SIMULÉS DANS R		166
BIBLIOGRAPHIE		171

LISTE DES TABLEAUX

	Page
Tableau 1.1	Les utilisateurs et les rôles d'un modèle.....8
Tableau 2.1	Liste des variables d'état d'un mammifère marin.....41
Tableau 2.2	Les valeurs enregistrées pour chaque individu49
Tableau 4.1	Autocorrélation de l'angle de braquage r_θ en fonction la distance seuil, x, en mètres77
Tableau 4.2	Patrons de trajectoire extraits des données observées.....79
Tableau 4.3	Tests de qualité d'ajustement des fonctions de répartition des patrons de trajectoire avec une approche paramétrique.....80
Tableau 4.4	Tests de qualité d'ajustement des fonctions de répartition des patrons de trajectoire avec une approche non paramétrique.....82
Tableau 4.5	Stratégies de prétraitement des données AOM.....84
Tableau 4.6	Nombre d'observations AOM conservées en fonction du filtre85
Tableau 4.7	Patron observé de la taille de groupe en fonction de la marée.....90
Tableau 4.8	Exemple d'un tableau de compilation des résultats pour une expérimentation.....97
Tableau 5.1	Résultat de l'expérimentation avec l'algorithme MA.....101
Tableau 5.2	Résultat de l'expérimentation avec l'algorithme MA.....102
Tableau 5.3	Résultat de l'expérimentation avec l'algorithme MAC104
Tableau 5.4	Résultat de l'expérimentation avec l'algorithme MAC105
Tableau 5.5	Résultat de l'expérimentation avec l'algorithme MAC-R108
Tableau 5.6	Résultat de l'expérimentation avec l'algorithme MAC-R108
Tableau 5.7	Paramètres utilisés pour l'algorithme MAC-S.....111
Tableau 5.8	Équation et statistiques de la régression multilinéaire112
Tableau 5.9	Résultat de l'expérimentation avec l'algorithme MAC-S116

Tableau 5.10	Résultat de l'expérimentation avec l'algorithme MAC-S	116
Tableau 5.11	Résultat de l'expérimentation avec l'algorithme MAC-RS.....	119
Tableau 5.12	Résultat de l'expérimentation avec l'algorithme MAC-RS.....	119
Tableau 5.13	Résultat de l'expérimentation avec l'algorithme MAC-MMBN	122
Tableau 5.14	Résultat de l'expérimentation avec l'algorithme MAC-MMBN	123
Tableau 5.15	Temps d'exécution des expérimentations en fonction des algorithmes et ratio de temps d'exécution par rapport à la marche aléatoire simple.....	125
Tableau 5.16	Tableau récapitulatif des patrons reproduits en fonction de l'algorithme de déplacement utilisé.....	127

LISTE DES FIGURES

	Page
Figure 1.1	Processus de modélisation.7
Figure 1.2	Processus de modélisation d'un système complexe.....7
Figure 1.3	Processus de modélisation orienté par patrons.13
Figure 1.4	Voisinage d'un automate cellulaire.19
Figure 1.5	Structure d'un algorithme génétique simple.....34
Figure 2.1	Cycle de modélisation.....37
Figure 2.2	Modèle conceptuel du processus de déplacement d'un mammifère marin.43
Figure 2.3	Diagramme de classe simplifié du modèle.47
Figure 2.4	Séquence d'exécution d'un pas de temps.48
Figure 2.5	Processus de modélisation utilisé dans ce projet50
Figure 4.1	Exemple d'une trajectoire à gauche et de cette même trajectoire simplifiée à droite à l'aide de la méthode d'échantillonnage variable.....75
Figure 4.2	Illustration de la distance maximale x.76
Figure 4.3	Exemple d'une trajectoire avec trois positions.78
Figure 4.4	Fonction de répartition discrétisée de la vitesse du rorqual commun avec un pas de 0.01 et son intervalle de confiance de $\alpha_{0.05}$81
Figure 4.5	Boîte à moustache de la taille de groupe du rorqual commun en fonction des marées.89
Figure 4.6	Processus d'extraction des patrons simulés.92
Figure 4.7	Exemple du calcul de la statistique de Kolmogorov-Smirnov. La flèche représente l'endroit où il y a la plus grande différence entre les deux fonctions de répartition.95
Figure 4.8	Processus de compilation des résultats inter-réplication.96
Figure 4.9	Exemple d'un graphique de type radar avec les six patrons.98

Figure 5.1	Résultat de l'expérimentation avec l'algorithme MA.....	102
Figure 5.2	Résultat de l'expérimentation avec l'algorithme MAC.....	105
Figure 5.3	Résultat de l'expérimentation avec l'algorithme MAC-R.....	109
Figure 5.4	Diagramme d'un contrôleur de type proportionnel.....	113
Figure 5.5	Diagramme d'un contrôleur proportionnel intégral dérivé.....	114
Figure 5.6	Résultat de l'expérimentation avec l'algorithme MAC-S.....	117
Figure 5.7	Résultat de l'expérimentation avec l'algorithme MAC-RS.....	120
Figure 5.8	Résultat de l'expérimentation avec l'algorithme MAC-MMBN.....	123

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AOM	Activités d'observation en mer
BOI	Blocs d'observation instantanée
EM	Efficiencce du modèle
EMAN	Erreur moyenne absolue normalisée
KS	Statistique de Kolmogorov-Smirnov
MA	Marche aléatoire
MAC	Marche aléatoire corrélée
MAC-R	Marche aléatoire corrélée avec indice de résidence
MAC-S	Marche aléatoire corrélée avec influence social
MAC-RS	Marche aléatoire corrélée avec indice de résidence et influence sociale
MAC-MMBN	Marche aléatoire corrélée avec minimisation de la moyenne des biais normalisés
PMSSL	Parc Marin Saguenay-Saint-Laurent
SIGHAP	Système d'information pour la gestion de l'habitat du poisson
ZPM	Zone de Protection Marine

LISTE DES SYMBOLES ET UNITÉS DE MESURE

α	Rayon d'évitement entre les individus (mètre)
β	Rayon d'interaction entre les individus (mètre)
m	Distance d'un mouvement (mètre)
n	Nombre de mouvements
ψ	Cosinus moyen de l'angle de braquage
ρ	Indice de résidence
τ	Durée d'un mouvement (seconde)
v	Vitesse d'un mouvement (mètre/seconde)
x	Distance de seuil d'échantillonnage (mètre)
ω	Facteur de socialisation
ζ	Nombre réel aléatoire pigé dans l'intervalle [0,1]
$B_\psi(\theta)$	Biais normalisé de la fréquence d'occurrence de l'angle de braquage
$B_u(\theta)$	Biais normalisé de répartition de l'espèce
$c_j(t)$	Vecteur de position de l'individu i au temps t
$E_g(\theta)$	Erreur absolue normalisée de la taille de groupe
\bar{g}_{int}	Moyenne de la taille de groupe avec un point de vue externe
\bar{g}_{ext}	Moyenne de la taille de groupe avec un point de vue interne
$\Psi(x)$	Fonction de répartition de l'angle de braquage des mouvements
$s_j(t)$	Vecteur de vitesse de l'individu i au temps t
$T(x)$	Fonction de répartition de la durée des mouvements
$v_j(t)$	Vecteur de direction de l'individu i au temps t
$V(x)$	Fonction de répartition de la vitesse des mouvements

INTRODUCTION

La modélisation est utilisée dans une multitude de disciplines, tant en science pure pour simuler une réaction nucléaire qu'en science sociale pour comprendre l'interaction entre des personnes. Quel que soit le domaine de recherche, l'utilisation de modèles donne une formidable flexibilité aux chercheurs afin de se créer une plateforme expérimentale pour tester leurs hypothèses de recherche ou simuler leurs systèmes dans différents contextes.

La littérature scientifique abordant l'utilisation de modèles est en pleine expansion dans le domaine de l'écologie. Un nombre grandissant d'articles présentent leur modèle, une représentation simplifiée du système ciblé, comme piste de solution afin d'étudier des systèmes écologiques complexes. De nos jours, certains modèles étudient spécifiquement l'évolution des forêts (Parrott et Lange, 2004), la dynamique d'une population animale (Bergman, Schaefer et Luttich, 2000) ou encore, les interactions complexes entre des populations et leur environnement (Christensen et Walters, 2004)

Un projet de modélisation d'envergure a été amorcé en 2006 pour développer un modèle du trafic maritime et des mammifères marins dans le Parc Marin Saguenay-Saint-Laurent. L'objectif de ce projet est de développer un outil d'aide à la décision pour tester différents scénarios de réglementation et de zonage dans le parc marin. Ce projet est le fruit d'une collaboration entre l'École de technologie supérieure, l'Université de Montréal et l'Université de Calgary en partenariat avec le groupe de recherche et d'éducation sur les mammifères marins (GREMM), le Parc Marin Saguenay-Saint-Laurent, Parcs Canada et Pêches et Océans Canada.

L'une des raisons d'être de ce projet est de mieux comprendre et quantifier l'exposition des mammifères marins exercée par l'industrie touristique. Au cours de la saison touristique qui s'étale de juin à septembre, les rorquals communs qui visitent l'estuaire du Saint-Laurent subissent une exposition soutenue par les excursionnistes : dans un rayon de 2000 mètres, les rorquals communs ont subi une présence moyenne de 6.4 bateaux en 1995 et 5.6 bateaux en

1996 (Michaud et Girard, 1997a). Une recherche évaluant l'impact des activités d'observation en mer sur le comportement des rorquals communs conclut que l'exposition de ces mammifères modifie le comportement en plongée profonde qui est susceptible d'entraîner des coûts énergétiques importants (Michaud et Girard, 1997b). En plus des rorquals communs, il y a une population de bélugas, une espèce en péril, qui subit indirectement la pression induite par l'industrie touristique d'observation des grands mammifères marins. Même s'ils ne sont pas ciblés directement par les capitaines grâce à leur statut d'espèce en péril, l'espèce est touchée par l'intensification du trafic maritime dans son habitat et les nombreuses observations opportunistes effectuées par les excursionnistes lors de leur rencontre. Ainsi, l'un des sous-objectifs de ce projet d'envergure est de développer un outil pour quantifier l'exposition actuelle des mammifères marins et de mesurer l'impact potentiel des changements de réglementation et de zonage dans le parc marin sur leur exposition.

Ce mémoire se penche sur une problématique spécifique de ce projet : comment modéliser les déplacements des mammifères marins afin d'intégrer ce module dans l'outil d'aide à la décision? Une infinité de facteurs attirent ces grands mammifères marins dans l'estuaire du Saint-Laurent chaque année. Recueillir toutes les informations nécessaires pour définir précisément l'écosystème entourant les mammifères marins est impossible à réaliser. Mais au-delà des limitations au niveau des données, reproduire exactement le système implique la compréhension et la prédiction des mouvements effectués par des mammifères évolués et sociaux. Évidemment, ce projet ne prétend pas vouloir déchiffrer le processus cognitif des mammifères. Nous devons définir le niveau de détail à inclure dans le modèle, afin d'inclure suffisamment d'éléments pour répondre rigoureusement à la problématique sans complexifier inutilement le modèle ou poser des hypothèses sans fondement.

Pour répondre à cette problématique, l'objectif de ce mémoire est de développer un modèle basé sur individus et spatialement explicite des mammifères marins qui reproduit leurs déplacements dans le territoire d'étude. L'approche de modélisation orientée par patrons est utilisée pour structurer la démarche de développement. Ainsi, le choix des patrons utilisés

dans la démarche définit indirectement le niveau de détail à inclure dans le modèle. Par la suite, plusieurs modèles sont développés afin de tenter de répondre à la problématique. Finalement, l'utilisation des patrons permet d'analyser quantitativement les performances des différents modèles en comparant tous les patrons simulés par le modèle aux patrons observés dans la réalité. Cette phase de validation permet de sélectionner le meilleur modèle qui reproduit adéquatement tous les patrons observés.

Ce mémoire présente cette démarche de modélisation, de l'énoncé de la problématique jusqu'à la sélection du modèle final, en cinq chapitres.

Le premier chapitre présente l'état de l'art en modélisation de populations animales avec une approche basée sur individus. Nous présentons une introduction aux modèles ainsi que les subtilités entre les systèmes simples et les systèmes complexes. Par la suite, nous abordons la démarche de modélisation orientée par patrons et les différentes stratégies proposées dans la littérature en modélisation de populations animales.

Le second chapitre introduit notre projet avec l'approche de modélisation orientée par patrons. En premier lieu, nous définissons clairement la problématique, les objectifs et les hypothèses de recherche spécifique à ce projet. Par la suite, nous expliquons les modèles conceptuel et informatique développés pour répondre à cette problématique. Finalement, la présentation du processus d'analyse et de validation du modèle permet de boucler le cycle de modélisation.

Le chapitre III présente les six algorithmes de déplacement des individus implémentés dans le modèle. Une brève introduction situe chaque algorithme dans la littérature scientifique et détaille son implémentation. Finalement, les avantages et limitations théoriques de chaque algorithme sont discutés.

Le processus d'extraction et d'analyse des patrons est expliqué dans le chapitre IV. Nous présentons les données utilisées et la démarche pour extraire les patrons observés et les

patrons simulés. Pour analyser les performances de chaque modèle, la dernière section propose trois métriques pour décrire la similarité entre deux patrons.

Le chapitre V présente les résultats des expérimentations. Chaque algorithme de déplacement est simulé et le processus de validation compare les patrons simulés à ceux observés pour tester si le modèle répond aux objectifs. Par la suite, nous discutons du temps de traitement et recommandons un algorithme dans le but de l'implémenter dans le modèle global qui sera livré aux utilisateurs finaux.

Finalement, la conclusion synthétise l'ensemble du projet et présente les contributions de ce modèle dans le domaine de la modélisation de systèmes écologiques. Une série de recommandations concluent ce mémoire en proposant des pistes de recherche afin d'améliorer et de faire évoluer le modèle proposé.

CHAPITRE 1

ÉTAT DE L'ART

L'objectif de ce chapitre est de présenter l'état de l'art en modélisation de populations animales avec une approche basée sur individus. En premier lieu, une brève introduction permet de définir les notions de base liées aux modèles, de présenter le processus de modélisation d'un système et d'énumérer les principales utilités des modèles. La seconde section définit les différences entre le modèle conceptuel des systèmes traditionnels et le modèle conceptuel des systèmes complexes ainsi que leurs impacts dans le processus de modélisation. À partir de ces subtilités, la troisième section présente un cadre unifié de modélisation des systèmes complexes basé sur les patrons. Finalement, la quatrième section présente les différentes stratégies de modélisation des espèces animales que l'on retrouve dans la littérature.

1.1 Introduction aux modèles

Coquillard (Coquillard et Hill, 1997) propose une excellente définition d'un modèle : « Un modèle est une abstraction qui simplifie le système étudié en ignorant de nombreuses caractéristiques du système réel étudié, pour se focaliser sur les aspects qui intéressent le modélisateur et qui définissent la problématique du modèle ». Selon cette définition, c'est au modélisateur de définir le niveau d'abstraction de son modèle en fonction du domaine d'application et du niveau de précision désiré du modèle. Les variables sélectionnées doivent être choisies judicieusement, car la complexité augmente exponentiellement au nombre de variables, tandis que les gains tirés d'un modèle plafonnent en supposant que les variables sont incorporées au modèle dans un ordre d'importance décroissant (Coquillard et Hill, 1997; Grimm et Railsback, 2005).

Un modèle peut servir à analyser ou expliquer le fonctionnement d'un système. Cependant, les propriétés dynamiques du système ne sont généralement pas explicites dans un modèle. Ainsi, la simulation sert à faire évoluer un modèle dans le temps afin d'observer les

caractéristiques dynamiques du système. De plus, la simulation est utilisée afin de comprendre l'état d'équilibre du système lorsque les méthodes analytiques de résolution en sont incapables, ce qui est généralement le cas lorsque le système est spatialement explicite (Parker et al., 2003).

1.1.1 Processus de modélisation

Le processus de modélisation est un exercice itératif qui boucle de l'analyse de la réalité jusqu'à la validation du modèle par rapport aux observations réelles. Dès 1979, un formalisme de normalisation du processus de modélisation a été proposé par la *Society for Computer Simulation* (Schlesinger, 1979). Ce formalisme, toujours d'actualité, propose la création d'un modèle conceptuel et d'un modèle informatique tel qu'illustré à la figure 1.1.

Le modèle conceptuel décrit la réalité à l'aide de différents médiums tels que des équations, schémas ou des relations de causes à effets. Cette description est obtenue en analysant la réalité et en retenant les divers phénomènes pertinents aux yeux du modélisateur (Schlesinger, 1979). Dans le cas d'un système complexe, le modèle conceptuel peut décrire de manière hiérarchique les différents éléments dans le système et les relations entre eux. Dans ce cas, un modèle représentationnel sera nécessaire afin de détailler chacun des éléments primaires du modèle conceptuel. Chaque élément est alors décrit de manière indépendante et possède ses propres attributs, états et règles de transitions. C'est la combinaison et l'interaction entre les éléments primaires du modèle qui permet de faire émerger le comportement global du système (Parrott et Kok, 2000).

Finalement, le modèle informatique implémente le modèle conceptuel de Schlesinger ou le modèle représentationnel de Parrott et Kok. Dans les deux cas, c'est à ce niveau que le modèle est encodé sous la forme d'un programme informatique. Une vérification du modèle doit être effectuée afin de s'assurer de la concordance entre le modèle conceptuel et le modèle informatique. Finalement, le modèle est simulé et les résultats sont comparés aux données réelles afin de valider son niveau de précision.

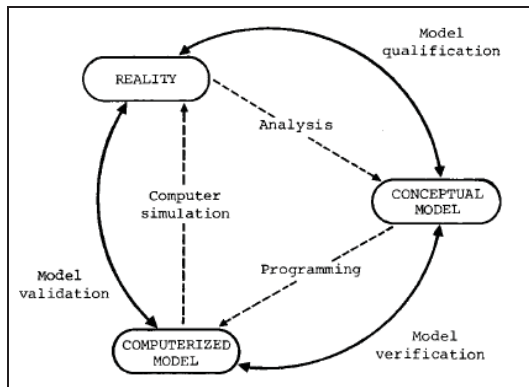


Figure 1.1 Processus de modélisation.

Tirée de Schlesinger (1979, p.103)

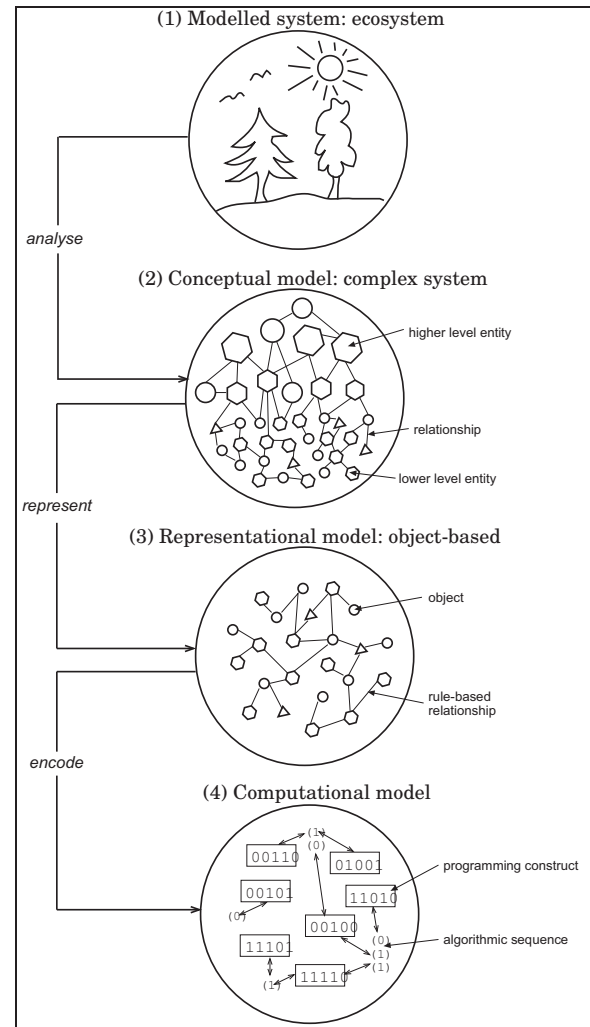


Figure 1.2 Processus de modélisation d'un système complexe.

Tirée de Parrott et Kok (2000)

1.1.2 Les utilités d'un modèle

Un modèle peut jouer plusieurs rôles en fonction du niveau de participation du modélisateur ou de l'utilisateur (Parker et al., 2003). Dans le cadre d'un projet typique de modélisation dans le but de créer un outil d'aide à la décision afin de mieux protéger une ressource, quatre types d'utilisateurs exploitent le modèle de manières différentes comme le présente le tableau 1.1.

Tableau 1.1 Les utilisateurs et les rôles d'un modèle

Niveau de participation	Acteurs	Rôle du modèle
Conception	Chercheurs en modélisation de systèmes.	Faciliter la compréhension de l'interaction entre les entités et la dynamique du modèle.
Modification	Chercheurs et spécialistes du système.	Faciliter la compréhension de la dynamique du système présent.
Utilisation	Spécialistes et gestionnaires du système.	Outil d'aide à la décision fiable pour les gestionnaires.
Visualisation	Gestionnaires et public.	Compréhension, vulgarisation du système et sensibilisation du public.

Lors de la conception du modèle, celui-ci est un instrument scientifique afin de valider la compréhension du système par l'expert. Dans ce cas, le modèle sert à tester des hypothèses scientifiques en validant le modèle avec les données provenant du système réel.

Un modèle est aussi un instrument scientifique pour les experts qui l'utilisent et le modifient. Le modèle sert à valider des propriétés dynamiques du système lorsque l'expert modifie légèrement le système et analyse les réactions à ces divers stimuli.

Il est aussi possible de considérer un modèle comme outil de gestion. L'objectif est de proposer des aides fiables à la gestion raisonnée du système étudié. Un exemple concret d'utilisation d'un modèle comme un outil de gestion est le modèle proposé par Anwar (Anwar et al., 2007). Ce modèle propose un outil d'aide à la décision aux gestionnaires quant aux divers impacts écologiques et économiques prévus en fonction de la réglementation et du zonage d'une aire protégée.

Finalement, un modèle est aussi un outil de communication. Avec une interface réaliste, la simulation d'un modèle est un moyen simple et efficace de visualiser les caractéristiques dynamiques d'un système.

1.2 Modélisation de systèmes écologiques complexes

Un système est constitué d'une collection d'objets en interactions (Coquillard et Hill, 1997). Globalement, un système peut être qualifié de complexe lorsque le comportement global du système est plus que la somme du comportement de ses entités. Cette inégalité est due aux liens hiérarchiques et aux interactions entre les individus qui ne sont pas prévisibles en analysant chaque entité de manière individuelle. Ainsi, cette approche proposée par les systèmes complexes confronte directement la philosophie réductionniste de Descartes (Schank, 2001).

L'émergence est un concept fondamental des systèmes complexes qui les distinguent des systèmes simples. Les phénomènes émergents d'un système sont définis comme des caractéristiques dynamiques qui ne peuvent être prédites par l'analyse de chaque élément pris individuellement (Parker et al., 2003) et qui créent cette inégalité entre la somme des comportements des composantes individuelles du système et le comportement du système dans son ensemble. Ces caractéristiques peuvent prendre la forme d'un patron spatial, temporel ou spatio-temporel généré par l'auto-organisation des éléments. Ainsi, l'analyse de systèmes complexes et des phénomènes émergents se concentre sur l'organisation des entités dans le système ainsi que le développement de nouvelles caractéristiques dans cette organisation (Parrott et Kok, 2000). Certains auteurs situent les systèmes complexes à la frontière entre un système organisé et le chaos. En effet, il se crée dans un système complexe une certaine organisation interne entre les différentes entités. Cependant, cette organisation est hautement dépendante de l'historique du système puisqu'une petite différence à l'état initial du système peut engendrer une dynamique du système totalement différente (Judson, 1994).

1.2.1 L'approche basée sur individus

La représentation théorique d'un système complexe peut se faire sous deux formes différentes. La première consiste à écrire la description mathématique détaillée du système. Cependant, l'imprécision de ces modèles traditionnels nécessite de poser des hypothèses simplificatrices qui ne respectent pas deux aspects fondamentaux de la biologie : tous les individus sont différents et les interactions entre les individus prennent place localement. Afin de pallier à ces lacunes, une seconde représentation propose une description mathématique simple, abstraite et générale qui détaille chaque entité du système et les relations entre ces entités (Judson, 1994). Cette seconde approche basée sur individus est appelée ascendante ou « bottom-up » dans le jargon de l'écologie en s'inspirant du principe de modéliser les éléments en bas de l'arbre hiérarchique et d'analyser l'émergence de comportements dans la collectivité en haut de l'arbre hiérarchique.

L'utilisation de l'approche basée individus est suggérée dans différentes situations. Le premier cas est lorsque chaque individu a plusieurs attributs d'importance qui déterminent son comportement. Dans ce cas, l'utilisation de courbes de distribution statistique des attributs des individus n'est pas suffisante pour modéliser correctement le système. Le second cas est lorsque la taille de la population est petite. Dans ce cas, la taille de l'échantillon est insuffisante afin d'effectuer des analyses statistiques significatives. Finalement, le troisième cas où l'approche basée sur individus est nécessaire est lorsque l'hypothèse d'une population parfaitement connectée n'est pas adéquate. Dans ce cas, une approche de modélisation globale du système ne permet pas d'exploiter les propriétés et l'organisation locales des individus.

Les avantages de l'approche basée sur individus sont nombreux. En premier lieu, la modélisation des individus est beaucoup plus simple et intuitive que celle d'un écosystème entier agrégé. De plus, les données physiologiques et comportementales provenant des études sur le terrain sont directement exploitables dans une approche basée sur individus où chaque individu est potentiellement différent (DeAngelis, 2001). Finalement, l'élaboration

de ce type de modèle permet de générer un comportement complexe du système en utilisant moins d'hypothèses simplificatrices irréalistes en plus d'aider à la compréhension de la dynamique du système et des interactions entre les individus lors de sa conception (Judson, 1994).

1.2.2 Représentation spatio-temporelle

La représentation spatiale est généralement explicite dans les modèles de systèmes complexes puisque les interactions entre les individus prennent place localement. Dans ce cas, il faut choisir une résolution spatiale équilibrée puisqu'une résolution trop fine entraîne un biais statistique par un manque d'échantillons alors qu'une résolution trop grande réduit la précision du modèle. Finalement, il faut tenir compte de la résolution des données réelles disponibles avant de fixer la résolution spatiale du modèle (Parker et al., 2003). Ces données géographiques sont généralement discrétisées sous la forme matricielle. Lorsque les données nécessitent des résolutions différentes, il est possible de superposer ces matrices avec des échelles différentes (Bennett et Tang, 2006).

La granularité du temps est une variable importante lors de la simulation. Généralement fixe tout au long de la simulation, une règle du pouce simple propose : « Le pas de temps à retenir est celui des événements pris en compte dans la modélisation et intervenant le plus fréquemment dans le système réel » (Coquillard et Hill, 1997). Cependant, cette même règle limite l'écart de fréquence entre les événements fréquents et rares à 10^6 afin de limiter le temps de calcul et éviter les erreurs d'échantillonnage par la précision arithmétique des ordinateurs.

Pour lier les résolutions spatiale et temporelle, le critère de Courant-Friedrichs-Lewy (CFL) décrit à l'équation 1.1 permet de déterminer si le ratio entre la précision spatiale et temporelle est adéquat. Dans cette équation, Δt est le pas de temps, Δx_c la résolution spatiale et c la vitesse la plus rapide des phénomènes d'intérêts dans le modèle (Bennett et Tang, 2006).

$$\Delta t \leq \frac{\Delta x_c}{c} \quad (1.1)$$

Le critère CFL a été proposé initialement dans le domaine de la météorologie et de l'hydrologie en 1928 pour s'assurer que l'information ne voyage pas plus vite dans l'espace que dans le temps.

1.2.3 La validation des modèles écologiques complexes

Dans le processus de modélisation proposé par Schlesinger, la validation du modèle compare le résultat de la simulation du modèle informatique à la réalité. Évidemment, une simulation ne peut reproduire parfaitement la réalité. C'est au modélisateur et aux utilisateurs de fixer le niveau de précision du modèle par rapport à la réalité en fonction du domaine d'application du modèle (Schlesinger, 1979).

Cependant, la validation d'un modèle complexe est un exercice beaucoup plus difficile dans la mesure où il est impossible de prédire à long terme l'état exact d'un système non linéaire. Ce type de modèle est très sensible aux conditions initiales du système et à sa séquence des nombres aléatoires puisqu'il utilise généralement la méthode de Monte-Carlo. Ainsi, il est impératif d'effectuer plusieurs répliques de chaque simulation en modifiant ces deux paramètres afin de calculer la moyenne et la variance des phénomènes observés (DeAngelis, 2001).

Très peu de techniques sont proposées dans la littérature afin de valider adéquatement un modèle de systèmes complexes. Des approches simples de compilation des attributs et de validation visuelle de la densité de la population sont des techniques encore couramment utilisées dans la littérature. Une approche beaucoup plus complète, la modélisation orientée par patrons (pattern-oriented modeling), propose un cadre général de modélisation et de validation plus souple et plus apte à capter l'essence d'un système complexe.

1.3 Approche de modélisation orientée par patrons

L'approche de modélisation orientée par patrons propose un cadre unifié de modélisation spécialement adapté pour les systèmes complexes. La particularité de cette approche provient de l'attention particulière apportée à la validation tout au long du processus de modélisation. Puisque le comportement de chaque individu est impossible à prédire dans un système complexe, l'approche par modélisation orientée par patrons permet de s'affranchir des attributs intrinsèques des entités en quantifiant à l'aide de patrons l'organisation globale des entités dans le temps et l'espace.

Le processus de modélisation orienté par patrons illustré à la Figure 1.3, présenté par Wiegand, comporte quatre étapes itératives (Wiegand et al., 2003). En premier lieu, les données provenant du système étudié sont collectées et compilées. Par la suite, les paramètres du modèle sont déterminés de manière empirique et s'enchaîne une phase de calibration en comparant les patrons réels aux patrons simulés. Finalement, des prédictions secondaires sont dérivées du modèle afin de valider sa cohérence.

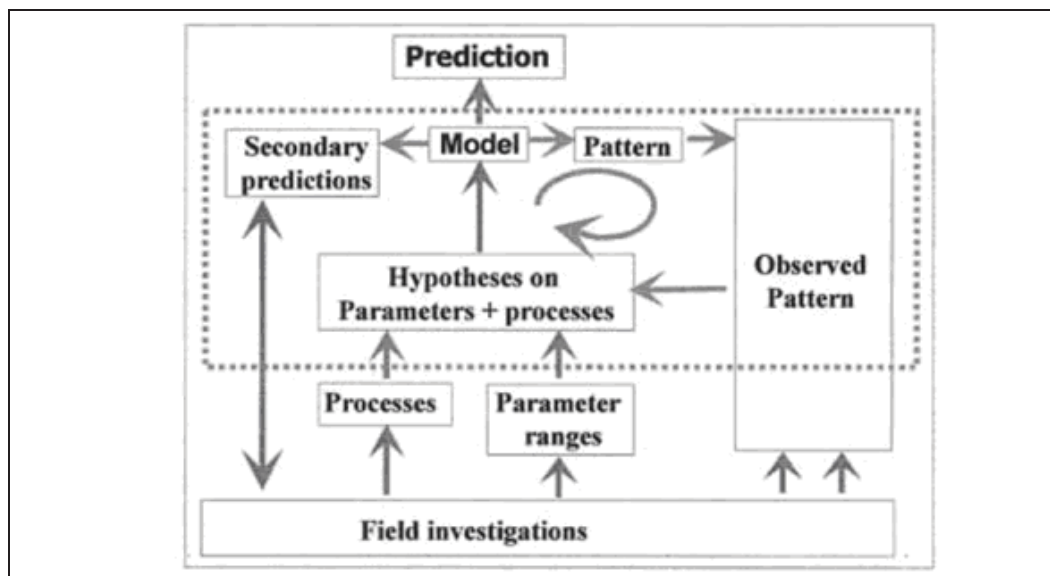


Figure 1.3 Processus de modélisation orienté par patrons.

Tirée de Wiegand (2003, p.209)

1.3.1 Les patrons

Dans le domaine de l'écologie, les patrons sont des caractéristiques identifiables directement dans la nature ou dans les données extraites de celle-ci. Les patrons peuvent être identifiés sous la forme d'une série temporelle de données, une distribution statistique de données ou encore une distribution spatiale. En fait, n'importe quel ensemble de données dont la variation n'est pas aléatoire est susceptible d'être un patron et indique la présence d'un processus sous-jacent significatif qui cause cette variation (Wiegand et al., 2003). L'utilisation d'un seul patron n'est pas suffisante afin de décrire correctement l'ensemble des interactions d'un modèle. L'utilisation simultanée de patrons à différentes échelles de temps, d'espaces et de niveaux hiérarchiques permet de lier les processus dans les différentes échelles et ainsi comprendre la dynamique du système complexe (Grimm et al., 2005).

1.3.1.1 Les patrons non spatialement explicites

En modélisation basée sur individus, la validation de données non spatialement explicites consiste à compiler les attributs de chaque individu et de comparer ces résultats à la population réelle avec différents outils statistiques (Judson, 1994; Parker et al., 2003). Cette technique couramment utilisée (Bennett et Tang, 2006; Gimblett, Richards et Itami, 1997; Huse et Giske, 1998; Little et al., 2004; Morales et al., 2005) a l'avantage d'être très simple à valider puisque les données provenant du terrain sont généralement sous ce format.

Cependant, la compilation des attributs décrit difficilement les propriétés dynamiques et l'auto-organisation des entités qui sont l'essence même des systèmes complexes. Ainsi, la valeur moyenne de l'attribut est généralement tracée en fonction du temps afin de voir émerger un patron de l'évolution de l'attribut dans le temps.

1.3.1.2 Les patrons spatio-temporels

Lorsqu'un modèle est spatialement explicite, la densité des individus dans l'environnement est un indicateur significatif de son auto-organisation. Une approche couramment utilisée

consiste à valider visuellement une carte de densité de la population (Alderman et Hinsley, 2007; Bennett et Tang, 2006; Huse et Giske, 1998; Little et al., 2004). Afin de représenter l'évolution spatio-temporelle de la densité d'une population, une superposition de plusieurs cartes de densité à différents moments de la simulation permet de bien visualiser les patrons émergents du déplacement des individus en fonction du temps (Huse et Giske, 1998). Au-delà de la validation visuelle, quelques approches basées sur les statistiques spatiales permettent de comparer quantitativement les patrons de densité d'un modèle (Parker et al., 2003).

En plus de la densité, des patrons caractérisant la texture sont utilisés afin de décrire l'organisation des entités (Parker et al., 2003). Un exemple frappant de cette utilisation de texture est le modèle de Wootton décrivant l'organisation spatiale des moules de Tatoosh (Wootton, 2001). Dans ce cas précis, le patron de texture est défini comme une distribution de fréquence de la longueur de l'espace entre deux moules sur un transect aléatoire.

Finalement, un dernier exemple de patron spatio-temporel est l'utilisation de la fréquence de rencontre ou de collision en fonction de l'emplacement. Le système RBSim de Gimblet utilise ce type de patron afin de modéliser les risques de rencontre des touristes le long d'un sentier récréotouristique (Gimblett, Richards et Itami, 1997). Dans le cas d'un environnement en deux dimensions, une carte de densité représentant la fréquence de rencontre doit être utilisée au lieu d'une courbe de distribution dans le cas d'un sentier à une seule dimension.

1.3.1.3 Les patrons de mouvement

Enfin, les patrons de mouvement permettent de décrire les déplacements des individus indépendamment de leur position géographique. Parmi ce groupe de patrons, la distance de déplacement quadratique R_n^2 (*squared displacement*) compile la distance euclidienne parcourue quotidiennement par chaque individu. Ce patron a permis par exemple de

quantifier numériquement les phases d'exploitation et d'exploration d'un herbivore à la recherche de pâturage (Morales et al., 2005).

Un second patron de mouvement utilisé dans les modèles basée sur individus est le home range. Ce patron mesure la surface du territoire utilisée par l'individu en fonction du plus petit polygone convexe englobant l'ensemble de la région utilisée. De plus, des données statistiques concernant la composition ou les attributs du territoire peuvent aussi être compilées. Cette stratégie a permis de compiler des patrons intéressants d'utilisation et de sélection du territoire pour les élans du parc de Yellowstone (Forester et al., 2007).

1.3.2 Les étapes de modélisation orientée par patrons

Tel que mentionné, le processus de modélisation orientée par patrons, reformulé par Wiegand (Wiegand et al., 2003), comprend quatre étapes qui sont détaillées dans cette section.

1.3.2.1 Étape 1 – Compiler et agréger les données

La première étape de la modélisation orientée par patrons consiste à compiler les données obtenues sur le terrain sous différentes échelles afin de détecter les patrons caractérisant le système. Le modélisateur doit apporter une attention particulière lors de la sélection des patrons puisqu'ils représentent la fondation du modèle.

Les hypothèses du modèle sont posées afin de reproduire les patrons sélectionnés. Ainsi, cette approche force à structurer le modèle avec les éléments et les relations qui sont présents dans la réalité et qui sont explicitement contenus dans les patrons. En plus de structurer le modèle, le choix des patrons impose des échelles spatiale et temporelle cohérentes qui permettent d'agréger suffisamment le comportement imprévisible de chaque entité pour voir émerger les phénomènes complexes. Ce niveau de détail idéal se situe entre la modélisation simpliste d'un seul patron et la modélisation trop précise qui utilise toutes les variables possibles collectées sur le terrain (Grimm et al., 2005). Selon Grimm, un modèle utilisant plusieurs patrons observés dans différentes échelles est plus susceptible de contenir ce niveau

de détail idéal. Cette notion de compromis est aussi rapportée en illustrant que la connaissance tirée d'un modèle est une parabole inversée alors que la complexité informatique est exponentielle par rapport à la complexité du modèle (Coquillard et Hill, 1997).

1.3.2.2 Étape 2 – Déterminer les paramètres du modèle

La seconde étape consiste à déterminer les paramètres du modèle. Cette étape, identique à un modèle conventionnel, fixe les valeurs des paramètres par défaut du modèle. En plus de la valeur par défaut, il est nécessaire de déterminer pour chaque paramètre une plage de valeurs potentielles qui sera éventuellement balayée dans la phase subséquente de calibrage.

1.3.2.3 Étape 3 – Comparer les patrons observés et simulés

Dans cette étape, toutes les hypothèses de modèle sont simulées dans l'ensemble de la plage de valeurs des paramètres. Les patrons observés servent de point de référence pour évaluer les patrons simulés. Ainsi, une analyse statistique quantitative permet de comparer la ressemblance de chaque patron par rapport à la réalité et de déterminer les paramètres du modèle qui reproduisent le mieux l'ensemble des patrons.

Cette phase de calibration peut facilement être représentée comme un problème d'optimisation multicritère, où la problématique consiste à optimiser simultanément la ressemblance d'un ensemble de patrons en fonction de plusieurs paramètres. Wiegand et al. (Wiegand et al., 2003) suggèrent l'utilisation d'un front de Pareto de type POMAC spécialement adapté pour les modèles écologiques, tandis qu'une approche basée sur les algorithmes génétiques peut s'avérer une alternative intéressante (Schank, 2001).

Cette étape de comparaison ne garantit pas que le modèle reproduise le mécanisme générant ces patrons. La phase de calibration est susceptible de masquer des problèmes au niveau du modèle et de laisser transparaître un modèle répondant parfaitement aux attentes (Railsback, 2001). Sans éliminer complètement ce risque, l'utilisation d'une multitude de patrons dans

les différentes échelles spatio-temporelles et hiérarchiques assure de structurer de manière réaliste le modèle et de réduire le risque de reproduire les patrons par un processus irréaliste (Grimm et al., 2005).

1.3.2.4 Étape 4 – Effectuer des prédictions secondaires

Finally, the last step of the pattern-based modeling approach consists of performing secondary predictions from the model. These predictions can be unused patterns during the modeling phase or simply field data compared to simulated data. This maneuver has two objectives: increase the credibility of the model and improve its comprehension.

Secondary predictions serve as a validation base to ensure the model faithfully represents the process. By analogy with a learning process, patterns are the training data while secondary predictions are the validation data. Thus, a poor representation of the mechanism behind the complex system or a bias in the model parameters can be detected during the modeling process. This method allows validating the structure of the model, which is difficult to quantify, and the model's credibility is thus enhanced.

In addition to increasing credibility, secondary predictions allow increasing the comprehension of the model. Thus, it is possible to test secondary predictions at scales outside those used during modeling to understand the model's dynamics and its limits. Eventually, observing different secondary predictions can also lead to parameter modification or the creation of new patterns during the next modeling cycle (Wiegand et al., 2003).

1.4 Approches de modélisation de populations animales

Animal population modeling is a subject regularly addressed in the literature of ecological modeling. This section briefly presents the different approaches

pour simuler les animaux dans les modèles basés sur individus. En premier lieu, les méthodes de représentation des individus sont présentées. Par la suite, les facteurs influençant les individus sont abordés. Finalement, les approches utilisées pour modéliser le processus décisionnel des individus concluent ce chapitre.

1.4.1 Représentation des individus

La stratégie de représentation des individus dépend principalement de leur dynamique spatiale. Lorsque les individus sont statiques spatialement, la représentation par automates cellulaires est généralement privilégiée. Cependant, la représentation basée sur individus ou agents est beaucoup plus flexible lorsque les entités se déplacent dans leur environnement.

1.4.1.1 Les automates cellulaires

Les automates cellulaires sont un ensemble d'automates disposés sous la forme d'une grille. Chaque automate est une machine à états finis simulée dans une base de temps discret. Les règles de transition déterministes d'un automate sont basées sur l'état des automates avoisinants. Généralement, le voisinage est défini par une grille carrée, voisinage de Moore, ou une grille à la forme d'un losange, voisinage de Von Neumann, adjacent à l'automate.

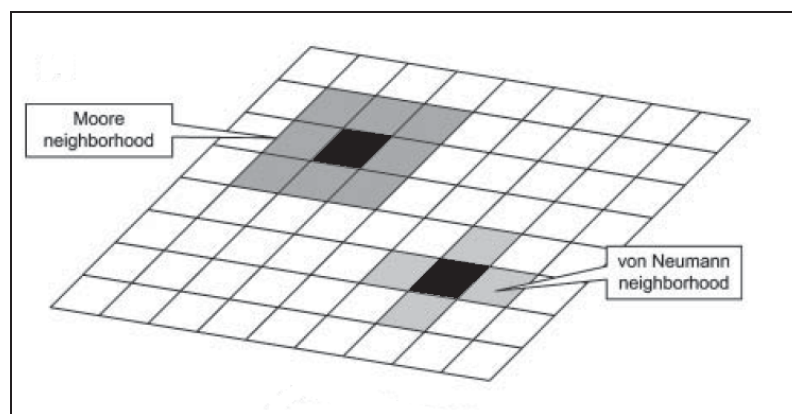


Figure 1.4 Voisinage d'un automate cellulaire.

Adaptée de Torrens et Benenson (2005, p.385)

À partir de ce voisinage simple de l'automate cellulaire, plusieurs variantes peuvent s'y greffer. La notion de voisinage par exemple peut être modifiée afin d'établir des liens non locaux ou d'obtenir une connectivité variable entre les automates afin de représenter la distance ou un coût entre ceux-ci. L'approche généralisée de l'automate cellulaire permet d'avoir un voisinage flexible basé sur une connectivité, proximité ou distance entre les automates (Torrens et Benenson, 2005).

Malgré cette composition très simple des automates cellulaires, il est possible de voir émerger des patrons et des comportements complexes dans ces modèles. La simulation d'un modèle composé d'automates cellulaires peut avoir quatre comportements distincts : une forme homogène, une structure stable ou périodique, un comportement chaotique non périodique ou une structure complexe localisée (Wolfram, 1984). Dans le premier et deuxième cas, l'automate cellulaire tombe dans un bassin d'attraction (Wuensche, 1994). Cette théorie propose qu'un système dynamique déterministe discret tel qu'un automate cellulaire converge vers un état ou un cycle d'états stables lorsque les règles de transition et la connectivité sont homogènes. Le quatrième comportement, la structure complexe localisée, indique la présence d'un système complexe par la sensibilité à son état initial et l'émergence de patrons non prévisibles lors de l'analyse d'un seul automate. Le jeu de la vie (Gardner, 1970) est sans aucun doute la démonstration la plus connue qui reproduit exactement cette émergence de patrons dans un système complexe constitué d'un automate cellulaire très simple.

Les automates cellulaires sont largement utilisés afin de modéliser l'environnement dans le domaine de l'écologie tel que la composition des forêts ou l'utilisation du terrain en fonction du climat (Parker et al., 2003; Parrott et Kok, 2000). Cependant, les automates cellulaires peuvent aussi représenter des espèces animales statiques comme la composition des lits de moules (Wootton, 2001). Dans ce cas, des règles simples de propagation permettent la création et l'agrandissement des colonies. Dans ce type de modèle plus complexe, des événements stochastiques peuvent être injectés en cours de simulation afin de perturber le

système comme l'effet des vagues des océans sur les moules ou des feux de broussaille dans une forêt.

L'inconvénient majeur de cette approche est l'incapacité des automates à bouger dans un environnement spatialement explicite. Dans ce cas, il est nécessaire d'utiliser une nouvelle approche de modélisation plus souple, les modèles basé sur individus spatialement explicites.

1.4.1.2 Les agents et individus

Contrairement aux automates cellulaires, les individus sont des entités beaucoup plus souples qui peuvent se déplacer et interagir avec leur environnement. Chaque individu est une entité indépendante qui possède ses propres attributs, son propre processus décisionnel et une capacité d'interagir avec les autres individus ou son environnement (Parrott et Kok, 2000). La majorité des projets qui modélisent des populations animales utilisent ce type de représentation. Lorsque les individus posent des actions au meilleur de leurs connaissances afin d'atteindre un but, on dit alors qu'ils sont autonomes et cognitifs.

L'autonomie

Pour qu'un individu soit considéré comme autonome, il doit connaître un ensemble d'actions potentielles à poser et généralement un objectif à atteindre. En fonction de ce but à atteindre, l'individu analyse ses attributs internes, son environnement et les caractéristiques des autres individus avoisinants (Parker et al., 2003). Par la suite, les réactions créées par les actions de l'individu sont propagées dans son environnement et dans ses attributs internes.

La cognition

La cognition est l'ensemble des activités mentales, incluant la volonté, la perception, la projection et l'action, impliquées dans la relation de l'individu avec son environnement. Une structure cognitive est présente afin de lier l'objectif de l'individu avec son environnement à partir de son comportement (Parker et al., 2003). Le niveau de cognition peut varier grandement d'un modèle à l'autre. Une convention permet de préciser le niveau de cognition

selon l'appellation des entités du système (Parrott et Kok, 2000). Un modèle basé *individus* signifie un niveau de cognition faible des entités. Dans ce cas, la prise de décision de l'individu est réactive et statique en fonction de son environnement et de son objectif à atteindre. Cependant, un modèle basé *agents* indique la présence d'un processus cognitif beaucoup plus évolué. Dans ce cas, l'agent peut être capable d'apprendre comment interagir avec l'environnement afin d'atteindre son but. Le processus cognitif est ainsi adaptatif à son environnement et permet de mieux représenter le processus d'évolution d'un animal ou d'un humain.

Ce processus cognitif complexe peut être implémenté sous la forme d'un mécanisme d'optimisation ou un mécanisme d'évolution. Dans le premier cas, une optimisation par programmation dynamique peut calculer le comportement d'un agent en anticipant les états futurs du système (Parker et al., 2003). Cependant, ce calcul d'optimisation hautement complexe est biologiquement peu plausible dans la réalité. Dans le cas de l'implantation d'un mécanisme évolutif des agents, l'objectif est de déterminer les actions qui mèneront potentiellement vers une solution sans nécessairement être optimale. Les algorithmes génétiques ou d'apprentissage par renforcement sont les deux stratégies généralement utilisées afin d'implanter ce mécanisme évolutif. Les algorithmes évolutifs sont très coûteux en temps de calcul et la représentation des agents est généralement simplifiée afin de faire évoluer convenablement une population d'agents sur quelques milliers de générations (Parrott et Kok, 2000).

1.4.2 Facteurs influençant les individus

L'autonomie et la cognition de l'individu nécessitent de connaître le contexte dans lequel il évolue afin d'effectuer les bonnes actions pour atteindre ses objectifs. Généralement, ce sont les attributs individuels, les attributs spatiaux et les influences sociales qui sont modélisés et qui peuvent potentiellement influencer le comportement des individus.

1.4.2.1 Les attributs individuels

En premier lieu, les attributs individuels sont généralement utilisés dans les approches basées sur individus. La présence d'attributs permet d'exploiter au maximum la flexibilité des modèles basés sur individus comparativement aux modèles analytiques qui exigent que tous les individus soient identiques ou suivent une distribution statistique. L'âge ou la masse corporelle sont deux exemples d'attributs, spécifiques à chaque individu, couramment utilisés dans les modèles écologiques (Bennett et Tang, 2006; Huse et Giske, 1998; Morales et al., 2005).

L'attribut interne d'un individu peut, au-delà des propriétés physiologiques, qualifier la classe comportementale de celui-ci. Bien que ce soit applicable aux populations animales (Bennett et Tang, 2006), cette approche est plus souvent utilisée pour les humains. Un ouvrage intéressant dans ce domaine est le modèle de Gimblett simulant les interactions et les sources de conflit des différents utilisateurs d'un parc naturel (Gimblett, Richards et Itami, 1997). Un sondage effectué préalablement dans le parc a permis de déterminer différents types d'utilisateur dans le parc et de dresser un profil propre à chacun. Ainsi, lors de l'initialisation du modèle, un profil d'utilisateur est assigné à chaque individu et permet de reproduire la diversité des comportements et des conflits dans le parc. De manière plus simpliste, Little propose un modèle simulant l'exploitation des ressources maritimes (Little et al., 2004). L'attribut interne décrivant la catégorie de capitaine de bateau, stochastique ou cartésien, permet de classer en deux grandes familles leur profil comportemental.

1.4.2.2 Les attributs spatiaux

L'influence de l'environnement sur le comportement des individus permet d'exploiter la deuxième force de l'approche basée sur individus comparativement aux modèles analytiques : les interactions prennent place localement. Ainsi, les attributs locaux de l'environnement sont aussi responsables de l'émergence de patrons et doivent être inclus dans le modèle pour qu'il devienne spatialement explicite. Les attributs classiques dans les modèles écologiques sont généralement la densité des proies, des prédateurs ou la présence

de nourriture (Bennett et Tang, 2006; Forester et al., 2007; Huse et Giske, 1998; Little et al., 2004; Morales et al., 2005).

En plus de ces propriétés statiques, l'interaction des individus avec leur environnement change aussi la dynamique du modèle. Dans un modèle simulant le déplacement des capelans (Huse et Giske, 1998), une étude de sensibilité a démontré que la navigation des poissons est différente lorsque la nourriture consommée est retirée de l'environnement. Bennett explore aussi cette notion de rétroaction avec l'environnement afin d'émerger un patron de distribution spatiale des élans en partie causée par la compétition des différents individus pour les ressources communes (Bennett et Tang, 2006).

Finalement, la topologie dans laquelle évoluent les individus influence leur comportement. Alderman propose que la topologie de l'environnement modifie la portée visuelle des individus et induise directement une modification de son comportement. Ainsi, cette portée visuelle dynamique d'un individu est une propriété émergente provenant de la topologie de l'environnement combinée à la portée visuelle maximale d'un individu (Alderman et Hinsley, 2007). La capacité visuelle est aussi abordée dans le modèle de Gimblett. Celui-ci modifie la capacité visuelle à détecter la présence d'autres individus en fonction de la topologie et du type de végétation. Ainsi, la portée visuelle émergente utilisée dans ce modèle modifie le nombre de contacts visuels comptabilisés entre les utilisateurs (Gimblett, Richards et Itami, 1997).

1.4.2.3 Les influences sociales

Le comportement d'individus sociaux est évidemment influencé par la proximité des autres membres de la même espèce. Dans le cas de bancs de poissons ou de nuées d'oiseaux, les individus ont tendance à respecter certaines règles collectives. Les individus ont tendance à éviter les individus trop près d'eux, de créer une cohésion au groupe en s'alignant aux individus autour d'eux et en se rapprochant des individus trop éloignés (Couzin, 2007; Couzin et al., 2002). Ainsi, il n'existe aucun contrôle centralisé du groupe puisque la

cohésion du groupe est générée par la sommation de l'influence de chaque individu sur son voisinage. Le transfert de l'information s'effectue strictement par le couplage fort entre les individus avoisinants. Ainsi, un changement de direction causé par la présence d'un prédateur ou de nourriture est amplifié et propagé comme une onde dans l'ensemble du groupe.

L'implantation de ce type d'algorithme est très simple et s'effectue en trois étapes (Couzin et al., 2005). En utilisant le vecteur de position $c_j(t)$ et de vitesse $v_j(t)$ des individus avoisinants, il est possible de déterminer le vecteur de direction $d_i(t + \Delta t)$ désiré par l'individu afin de respecter les règles décrites précédemment.

La première étape consiste à conserver une distance minimale entre l'individu i et les autres individus j avoisinants. S'il y a un individu dans ce rayon, alors il y a une manœuvre d'évitement avec l'équation 1.2.

$$d_i(t + \Delta t) = - \sum_{j \neq i} \frac{c_j(t) - c_i(t)}{|c_j(t) - c_i(t)|} \quad (1.2)$$

S'il n'y a aucun individu dans le rayon d'évitement de l'agent i , la seconde étape consiste à rapprocher et aligner l'individu i avec les j individus présents dans son rayon d'interaction β avec l'équation 1.3.

$$d_i(t + \Delta t) = \sum_{j \neq i} \frac{c_j(t) - c_i(t)}{|c_j(t) - c_i(t)|} + \sum_{j=1} \frac{v_j(t)}{|v_j(t)|} \quad (1.3)$$

Finalement, si l'individu i désire naviguer dans une direction précise notée par le vecteur unitaire g , alors le vecteur calculé à la 2^e étape est ramené sous la forme unitaire \hat{d} et le vecteur désiré g est pondéré en fonction du paramètre ω . Un ω près de zéro accorde plus d'importance à maintenir la cohésion du groupe, près de 1 balance les deux vecteurs alors

qu'un ω supérieur à 1 donne plus d'importance à la direction désirée de l'individu selon l'équation 1.4.

$$d_i'(t + \Delta t) = \frac{\hat{d}_i(t + \Delta t) + \omega g_i}{|\hat{d}_i(t + \Delta t) + \omega g_i|} \quad (1.4)$$

Ainsi, l'implantation de cet algorithme nécessite la paramétrisation de trois variables. Le rayon d'évitement α , le rayon de socialisation β et le facteur de pondération de la direction désirée ω . Dans un autre modèle plus complexe (Couzin et al., 2002), des paramètres limitant la vitesse angulaire et le champ de vision sont aussi pris en considération dans l'algorithme.

1.4.3 Processus décisionnel des individus

À partir des facteurs influençant l'individu, un processus décisionnel doit déterminer les actions à poser afin d'atteindre son but. Dans certains cas, cette prise de décision peut être hiérarchique. Le concept clé de la modélisation par patrons est l'utilisation de plusieurs patrons dans un système complexe sous plusieurs échelles spatio-temporelles différentes. Dans l'étude des déplacements de capelans (Huse et Giske, 1998) ou des élans (Morales et al., 2005), la prise de décision s'effectue de manière hiérarchique en deux étapes. Le premier choix détermine l'activité de l'individu. Dans le cas des poissons, le choix d'activité est de se nourrir localement ou de migrer, alors que pour les élans le choix est de se nourrir, se déplacer localement ou d'explorer le territoire. Par la suite, le second choix des deux modèles consiste à déterminer la direction du mouvement afin d'effectuer l'activité désirée. Cette prise de décision hiérarchique est aussi présente dans un second modèle modélisant le déplacement des élans (Bennett et Tang, 2006). Dans ce cas, une représentation multiéchelles des attributs de l'environnement a amené une prise de décision hiérarchique de la direction de navigation des individus. Dans ce cas particulier, l'individu détermine globalement sa direction de migration et navigue localement en pondérant ce vecteur de direction globale avec un vecteur de direction local décrivant la présence de nourriture. Le

point commun à ses trois modèles est l'utilisation explicite d'une structure hiérarchique dans le processus décisionnel des agents. Dans chacun de ces cas, plusieurs patrons ont été observés dans la réalité pour ensuite être transposés dans l'architecture interne du modèle (Grimm et al., 2005).

Il existe plusieurs techniques afin de déterminer quelle action effectuer en fonction de l'influence des attributs internes, de l'environnement et des individus avoisinants. Toutes ces techniques, de la simple réaction déterministe aux stratégies évolutives adaptatives, sont utilisées dans la littérature pour modéliser des espèces animales ou végétales.

1.4.3.1 Les automates finis

Les automates finis, ou machines à états finis, peuvent représenter une multitude d'objets biologiques ou inertes. Un automate est caractérisé principalement par son état interne qui est sélectionné à partir d'un ensemble fini d'états potentiels. L'état de l'automate évolue dans le temps grâce aux fonctions de transition qui utilisent l'état actuel de l'automate et des entrées externes afin de déterminer son état futur. Ainsi, les automates finis proposent une excellente structure générique pour modéliser des individus caractérisés par un seul état en effectuant des opérations complexes à partir de l'information environnante pour modifier leur comportement.

Formellement, un automate A est défini par un ensemble d'états S et un ensemble de règles de transition T . Chaque règle de transition calcule l'état futur de l'automate S_{t+1} en utilisant l'état actuel de l'automate S_t et des informations externes I_t (Torrens et Nara, 2007):

$$\begin{aligned} A &\sim (S, T) \\ T &: (S_t, I_t) \rightarrow S_{t+1} \end{aligned} \tag{1.5}$$

L'automate cellulaire est un cas particulier de l'automate fini. Dans ce cas, plusieurs automates finis immobiles sont arrangés sous la forme d'une grille et sont influencés par

l'état des automates avoisinants, généralement définis par le voisinage de Moore ou Von Neumann introduit à la section 1.4.1.1. Cependant, il existe des approches beaucoup plus complexes qui exploitent au maximum le potentiel des automates finis. Le modèle GAS (Torrens et Nara, 2007) par exemple propose une approche adaptative. En plus des attributs standards, les automates sont géoréférencés et des règles de mouvement déterminent leur déplacement. De plus, des règles définissant les changements de relation entre les automates permettent de rendre cette approche adaptative à son environnement. Cependant, cette représentation mobile des automates reste complexe et la solution la plus utilisée consiste à confiner l'automate fini dans un automate cellulaire afin de représenter l'environnement ou le terrain de manière statique (Parrott et Kok, 2000).

1.4.3.2 Règles déterministes

Un autre moyen plus simple et flexible de simuler le comportement d'un individu avec une capacité cognitive limitée est d'utiliser un ensemble de règles déterministes. Cette représentation est suffisante pour plusieurs modèles complexes basés sur individus, dans la mesure où c'est l'interaction entre les individus et leur environnement qui est analysée. Les règles peuvent être inférées à partir d'hypothèses de recherche, comme le processus de navigation des oiseaux proposé par Alderman (Alderman et Hinsley, 2007). Dans cette étude, trois algorithmes de déplacement déterministes sont simulés afin d'analyser la sensibilité du modèle en fonction de la présence ou l'absence de règles impliquant la topologie et la portée visuelle des oiseaux. Visuellement très intuitifs, les algorithmes sont illustrés sous la forme d'un diagramme de décision. Ainsi, à chaque pas de temps, les individus utilisent le diagramme afin de déterminer leur prochaine direction de navigation en fonction de la topologie du terrain et de la qualité des habitats environnants.

Dans un autre contexte, l'ensemble de règles déterministes peut être construit à partir d'études rigoureuses afin de construire un modèle reproduisant le plus fidèlement possible la réalité. Pour modéliser le comportement d'humains par exemple, une étude sociologique a été conduite dans le canyon de Broken Arrow afin de mieux comprendre la motivation des

usagers (Gimblett, Richards et Itami, 1997). Les utilisateurs du parc ont été invités à répondre à un sondage concernant les lieux visités, l'appréciation des sites et le niveau d'achalandage. Finalement, à partir de ces résultats, des ensembles de règles comportementales ont été extraits sous la forme d'heuristiques afin de répliquer exactement dans le modèle les réactions individuelles des usagers.

1.4.3.3 Décisions stochastiques

Dans une approche basée sur individus, les décisions stochastiques sont généralement générées par la méthode de Monte-Carlo. Globalement, cette méthode propose d'utiliser un nombre aléatoire généré en cours de simulation avec une distribution de probabilités afin de sélectionner le comportement ou l'action de l'individu (Judson, 1994). La présence de phénomènes stochastiques empêche de prédire les événements au niveau des individus sans nécessairement compromettre la convergence de la dynamique globale du système. Une phase de validation rigoureuse avec plusieurs réplifications est nécessaire afin de bien discerner les événements aléatoires et les patrons dynamiques en cours de simulation.

Les décisions stochastiques peuvent se présenter sous plusieurs formes différentes dans un modèle. La forme classique consiste à calculer un niveau d'attraction a_i pour chaque événement e_i d'un ensemble de N événements et ensuite piger un événement en convertissant ce niveau d'attraction sous la forme d'une probabilité $P(e_i)$.

$$P(e_i) = \frac{a_i}{\sum_N a_n} \quad (1.6)$$

La fonction du niveau d'attraction peut être un calcul complexe introduisant la notion de mémoire et d'expérience de l'individu (Bennett et Tang, 2006). Dans ce cas, l'individu devient un agent adaptatif qui utilise son expérience afin d'influencer ses prises de décision futures.

Plusieurs variantes à la prise de décision stochastique utilisent la méthode de Monte Carlo afin de contrôler les actions des individus. La marche aléatoire corrélée et les réseaux bayésiens sont deux stratégies probabilistes qui en sont dérivées.

1.4.3.4 Marche aléatoire corrélée

La marche aléatoire corrélée, *correlated random walk (CRW)*, est une technique probabiliste qui consiste à simuler le déplacement des animaux par une série de mouvements discrets linéaires. Chaque mouvement est défini par deux variables spatiales et une variable temporelle, soit la direction, la distance et la durée. Afin de simplifier le modèle en éliminant la corrélation entre la distance et la durée et en ajoutant implicitement la corrélation entre la direction actuelle et prochaine, les trois variables du mouvement sont converties sous la forme : vitesse, durée et changement de direction (Turchin, 1991). À chacune de ces trois variables de mouvements est associée une distribution de fréquence. Ainsi, la création d'un mouvement consiste à échantillonner à l'aide de la méthode de Monte Carlo chacune de ces trois variables de manière indépendante. En cours de simulation, les individus enchaînent les mouvements toujours générés par le même processus.

L'objectif d'utiliser des distributions de fréquences caractérisant le mouvement comparativement à un mouvement strictement aléatoire est la possibilité de calculer la densité de la population du système en équilibre en fonction de la position dans l'espace. Le concept d'indice de résidence ρ , calculé à partir des caractéristiques des mouvements observés dans deux zones, est proportionnel à la densité de la population $\tilde{\mu}$ observée dans le système en équilibre (Turchin, 1991) :

$$\frac{\tilde{\mu}_1}{\tilde{\mu}_2} = \frac{\rho_1}{\rho_2} \quad (1.7)$$

Le calcul de l'indice de résidence est basé strictement sur des données facilement mesurables à partir de suivis de trajectoires de quelques individus. Turchin propose un calcul par étapes afin de déterminer l'indice de résidence (Turchin, 1991) que nous présentons ici:

La première étape consiste à discrétiser les trajectoires observées des individus en segments linéaires, séparés en fonction des zones d'intérêts. Chaque segment doit être strictement contenu dans une seule zone. Par la suite, les caractéristiques des mouvements sont compilées pour chaque zone j : la longueur moyenne des mouvements μ_j , la longueur au carré moyenne M_j , la durée moyenne du mouvement τ_j et la moyenne du cosinus du changement de direction ψ_j .

La seconde étape permet de calculer certaines variables intermédiaires avec l'équation 1.8.

$$\begin{aligned}\gamma &= \frac{M}{\mu^2} \\ \varphi_j &= \gamma + (2 - \gamma)\psi_j \\ \alpha &= \frac{2\gamma(\mu_2 - \mu_1)}{(2 - \gamma)(\varphi_1\mu_2 - \varphi_2\mu_1)}\end{aligned}\tag{1.8}$$

Finalement, la troisième étape permet de calculer l'indice de résidence avec l'équation 1.9.

$$\rho_j = \tau_j \varphi_j^\alpha \mu_j^{-\left(\frac{2-2\gamma+\alpha}{2-\gamma}\right)}\tag{1.9}$$

Le modèle de marche aléatoire corrélée décrit par Turchin comporte certaines limitations sur le comportement des individus. En premier lieu, les interactions entre les individus doivent être négligeables afin de pouvoir estimer correctement la densité de la population. Puisque la distribution statistique de changement de direction est centrée à zéro, les individus ne doivent pas effectuer des mouvements circulaires ou de spirales. De plus, les individus ne doivent pas être influencés par le compas, car les changements de direction du modèle ne sont pas calculés en fonction d'une direction absolue. Finalement, l'absence d'une attraction lointaine ou de phénomène de migration est nécessaire pour exploiter la puissance de ce modèle.

1.4.3.5 Réseau bayésien

Le réseau bayésien est aussi une approche de modélisation probabiliste mais qui s'appuie sur la loi de Bayes. Les nœuds du réseau représentent les variables étudiées et les liens entre les nœuds sont les relations de dépendance quantifiées à l'aide de probabilités conditionnelles (Pearl, 1996). Le réseau bayésien donne un modèle probabiliste complet, c'est-à-dire la probabilité de tous les événements possibles est définie par les valeurs de l'ensemble des variables. Bien que cette combinaison d'événements soit exponentielle, l'approche de Bayes permet de réduire ce nombre par la représentation des probabilités conditionnelles locales entre la variable et ses parents. Ainsi, la sémantique globale d'un réseau bayésien pour les variables x_i et leurs parents pa_i est (Pearl et Russell, 2003):

$$P(x_1, \dots, x_n) = \prod_i P(x_i | pa_i) \quad (1.10)$$

Les réseaux bayésien possèdent deux grands avantages comparativement aux réseaux de neurones. Premièrement, il est possible d'effectuer des inférences de manière bidirectionnelle de la causalité des variables. Par exemple, l'inférence bidirectionnelle permet à la fois de connaître la probabilité de pluie si le sol est mouillé ou la probabilité que le sol soit mouillé s'il y a de la pluie. De plus, ce mécanisme d'inférence est transparent à l'utilisateur. Ainsi, le réseau bayésien n'est plus une boîte noire comme le réseau de neurones, mais un outil d'expérimentation et de compréhension du système (Pearl et Russell, 2003).

Avec ces propriétés, le réseau bayésien est une approche intéressante pour la création d'outils d'aide à la décision. Un cas concret du processus de modélisation d'un outil d'aide à la décision pour une application d'aménagement adaptatif est proposé par Nyberg et al (Nyberg, Marcot et Sulyma, 2006). Dans ce cas, la conception du réseau bayésien sert de ligne directrice afin de faire le point sur les connaissances actuelles sur le domaine, les hypothèses simplificatrices et les incertitudes. Ensuite, le réseau se transforme en outil technique afin d'effectuer des tests de sensibilité des différentes variables et d'évaluer les

conséquences de décisions alternatives. Cette expérimentation empirique est possible grâce à la propriété de l'inférence bidirectionnelle qui permet de spécifier certaines variables et de résoudre l'ensemble du réseau afin d'arriver à cet état. Finalement, il ne reste qu'à lier les variables du réseau à une fonction de coût pour obtenir un réseau bayésien capable d'optimiser un problème borné de contraintes en spécifiant la valeur de certaines variables (Nyberg, Marcot et Sulyma, 2006).

Ce modèle de raisonnement est aussi exploitable dans un modèle basé sur individus. Afin de comprendre le comportement des pêcheurs, un réseau bayésien a été utilisé afin de simuler l'exploitation des ressources halieutiques (Little et al., 2004). Les variables de ce réseau incluent, pour chaque parcelle de l'espace discrétisé, l'expérience antérieure de pêche, sa distance par rapport au port et le nombre d'individus observés ayant exploités cet endroit. À partir de ces variables, le modèle calcule les probabilités que l'individu exploite chaque parcelle de l'environnement. Ce réseau bayésien ne peut être résolu statiquement, car l'expérience et l'observation des autres bateaux changent avec le temps. Ainsi, une simulation avec un pas de temps discret s'impose pour permettre aux bateaux d'interagir entre eux et avec leur environnement et ainsi créer un système complexe plus réaliste.

1.4.3.6 Algorithme évolutionnaire

Contrairement aux réseaux bayésiens, les algorithmes évolutionnaires permettent aux individus de s'adapter explicitement à leur environnement en fonction de leurs expériences antérieures. Généralement, un algorithme évolutionnaire mimique l'évolution naturelle des organismes en modifiant leurs règles comportementales au fil des générations. La sélection naturelle favorise l'élitisme et la convergence vers un comportement approprié. Les algorithmes évolutionnaires nécessitent la simulation de plusieurs milliers de générations compte tenu de l'évolution pseudo-aléatoire de la population. Ainsi, le calcul d'adéquation (*fitness*) de chaque individu doit être relativement rapide afin de faire converger la population dans un temps raisonnable (Parrott et Kok, 2000).

L'algorithme génétique générique, couramment utilisé dans le domaine de l'optimisation, consiste à faire évoluer une population en fonction d'une mesure d'adéquation représentant la fonction objectif du problème. Cet algorithme itératif, illustré à la figure 1.5, sélectionne les meilleurs individus dans la population, modifie les gènes de ces individus par des opérateurs de mutation et de croisement puis les évalue. Dans le domaine de l'écologie, Morales propose un modèle multi-agents adaptatif évoluant le comportement des élans afin d'optimiser leurs chances de survie et de reproduction (Morales et al., 2005). Dans ce cas, la mesure d'adéquation utilisée est le produit de la réserve de gras de l'agent et la probabilité de survie face aux prédateurs.

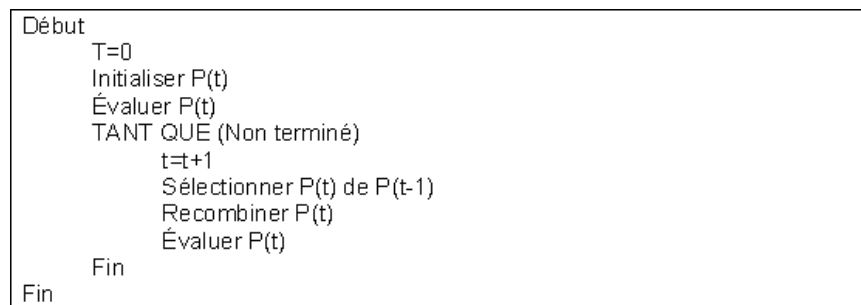


Figure 1.5 Structure d'un algorithme génétique simple.

Adaptée de (Herrera, Lozano et Verdegay, 1998)

Un deuxième type d'algorithme génétique utilise de manière explicite les propriétés de reproduction et de sélection naturelle directement présentes dans la biologie. Le modèle de Huse par exemple simule le processus naturel de reproduction afin de faire évoluer une population de capelan (Huse et Giske, 1998). Dans ce cas, la mesure d'adéquation n'est pas implémentée dans le modèle. Par l'absence de cette fonction d'évaluation, il est impossible d'utiliser une fonction de sélection afin de choisir les meilleurs individus pour créer la nouvelle génération. Ces deux étapes importantes de l'algorithme génétique sont remplacées par la sélection naturelle et la capacité de reproduction des poissons. Seuls les opérateurs de croisement et de mutation sont empruntés de l'algorithme génétique lors de la reproduction des individus. Cette stratégie évolutive se rapporte directement au domaine de la vie artificielle. Les algorithmes évolutionnaires sont des méthodes d'optimisation très flexibles

qui permettent de faire évoluer n'importe quelle séquence de gènes pour chaque individu d'une population. Il existe une multitude de manières de codifier le processus cognitif des individus dans une séquence de gènes. Une méthode biologiquement plausible consiste à utiliser un réseau de neurones et d'encoder les poids synaptiques dans les gènes des individus (Huse et Giske, 1998; Morales et al., 2005). Cependant, la flexibilité de ces algorithmes permettrait de faire évoluer plusieurs autres méthodes telles que des arbres de décision, des règles d'inférence, des seuils de décision ou des ensembles de règles.

1.4.3.7 Théorie des graphes

Finally, le comportement adaptatif d'un animal peut être implémenté à l'aide d'une mémoire spatiale de son environnement. L'objectif de cette approche est de mémoriser les expériences positives et négatives pour chaque emplacement. Bennett représente cette mémoire sous la forme d'un graphe bidirectionnel avec un poids sur chaque arête qualifiant l'appréciation des expériences antérieures (Bennett et Tang, 2006). Ainsi, le graphe G est la carte cognitive de l'individu, les nœuds V sont des emplacements géographiques, les arcs E sont des chemins directionnels entre deux emplacements et les poids W sont les forces d'attraction relative basées sur l'expérience sur chaque arc.

$$G = (V, E, W) \quad (1.11)$$

L'adaptation de la carte cognitive, initialisée à zéro, est effectuée par renforcement avec un apprentissage hebbien. La modification de la carte est appliquée à la fin d'une génération à partir de la fonction d'évaluation de l'individu. Tous les poids associés aux arcs visités par l'individu sont renforcés par la constante η_1 , privilégiant les arcs lorsque l'individu termine avec une fonction d'adéquation élevée et pénalisant les arcs lorsque l'adéquation est faible. Cependant, les nœuds non visités, mais avoisinant le passage d'un individu, sont inhibés par la constante η_2 alors que le poids des autres arcs reste identique. En posant comme hypothèse que la relation entre l'adéquation de l'individu et le poids associé à l'arc est linéaire, la modification des poids avec un apprentissage hebbien est :

$$\delta w_{ij|t} = \begin{cases} \eta_1 \cdot \text{Fitness}(\text{Agent}_{k|t}) & \text{si } v_i \in V_p \text{ et } v_j \in V_p \\ -\eta_2 \cdot \text{Fitness}(\text{Agent}_{k|t}) & \text{si } v_i \in V_p \text{ et } v_j \notin V_p \\ 0 & \text{si } v_i \notin V_p \text{ et } v_j \notin V_p \end{cases} \quad (1.12)$$

En cours de simulation, les individus utilisent leur carte cognitive afin de déterminer leur prochaine direction de navigation. Une méthode déterministe permet de diriger directement les individus vers les endroits où ils ont eu les meilleures expériences. Pour favoriser simultanément l'exploration et l'exploitation de l'espace, une méthode probabiliste de navigation peut diriger les individus proportionnellement aux poids d'attraction des arcs à la portée de l'individu, permettant ainsi de simuler l'imperfection sensorielle et les erreurs de navigation (Bennett et Tang, 2006).

1.5 Conclusion

Ce chapitre a permis de présenter l'état de l'art dans le domaine de la modélisation écologique. Une introduction aux modèles ainsi qu'aux modèles basés sur individus a bien situé les défis de la modélisation dans un contexte écologique. Par la suite, cette revue de la littérature a présenté l'approche de modélisation orientée par patrons et les stratégies de modélisation de populations animales les plus couramment utilisées dans le domaine de la modélisation de systèmes écologiques. Dans le prochain chapitre, nous appliquons concrètement ces différentes approches de modélisation afin de répondre à notre problématique de recherche.

CHAPITRE 2

DÉMARCHE DE MODÉLISATION

« Mere realism is a poor guideline for modeling. Modeling must be guided by a problem or question about real system, not by the system itself. »

-Volker Grimm et Steven Railsback, 2005

Ce chapitre présente la démarche utilisée pour développer le modèle des mammifères marins avec l'approche de modélisation orientée par patrons. Bien que ce processus soit généralement itératif, ce chapitre est structuré de manière linéaire en présentant séquentiellement les étapes du processus de modélisation, illustré à la figure 2.1, proposé par Grimm (Grimm et Railsback, 2005).

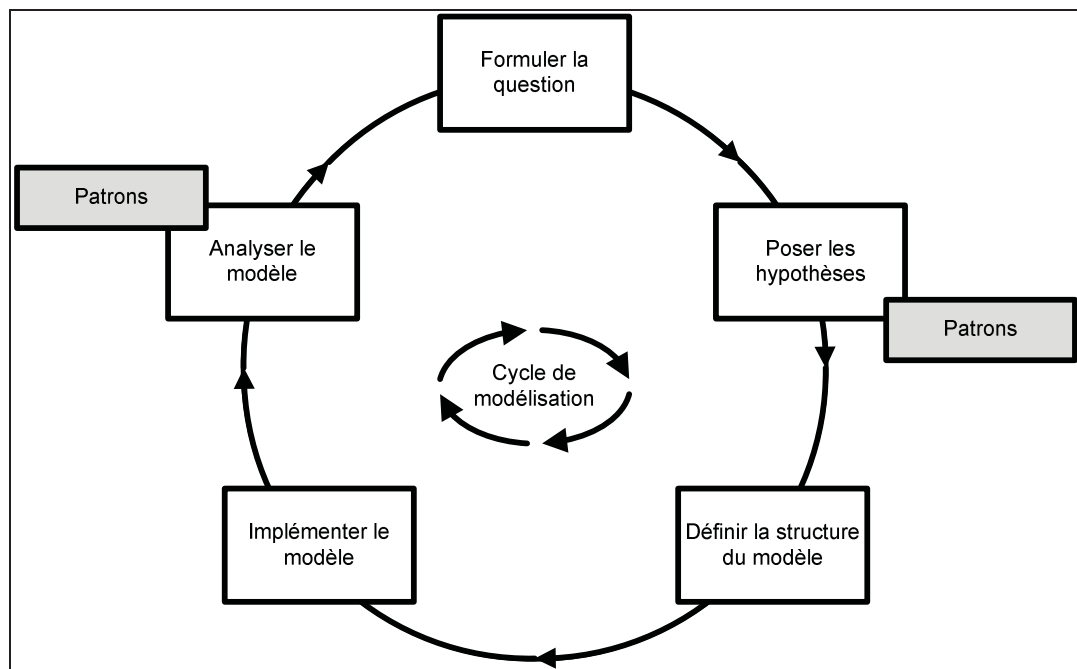


Figure 2.1 Cycle de modélisation.

Adaptée de Grimm et Railsback (2005, p.105)

La première étape du cycle de modélisation définit clairement la problématique à laquelle le modèle tente de répondre. À partir du problème bien défini, la seconde étape consiste à poser les hypothèses de recherche et élaborer le premier modèle conceptuel. À ce stade, les patrons que doit reproduire le modèle sont identifiés. La prochaine étape permet de structurer le modèle conceptuel en convertissant sous la forme quantitative les hypothèses posées précédemment. À ce niveau, tous les paramètres et les variables du modèle devraient être définis. Par la suite, la quatrième étape consiste à implémenter le modèle conceptuel, généralement sous la forme d'un programme informatique, et ainsi obtenir le modèle informatique. Finalement, la dernière étape permet d'analyser et surtout de valider le modèle à partir des patrons extraits.

2.1 Étape I : Formuler la question

Formuler la question à laquelle le modèle doit répondre est la première étape essentielle d'un processus de modélisation structuré. En plus de justifier le développement du modèle, formuler la question clairement permet de fixer la portée de l'étude en précisant explicitement la structure et les éléments à inclure dans le modèle (Grimm et Railsback, 2005).

Dans le cadre de ce projet, la question qui motive la création du modèle est : « *Comment représenter les déplacements spatio-temporels des rorquals communs et des bélugas dans le Parc Marin Saguenay-Saint-Laurent au cours de la saison touristique ?* »

La portée de l'étude est définie par le verbe utilisé dans la question : représenter. Ce projet n'a pas la prétention d'expliquer ou de comprendre la raison des déplacements des mammifères marins, mais seulement de simuler leurs mouvements. Ce projet s'inscrit dans une recherche beaucoup plus large qui cherche à comprendre l'effet de la réglementation à l'intérieur du parc marin sur le trafic maritime et les mammifères marins. Un modèle simple et prédictible des déplacements des mammifères marins est essentiel afin de ne pas poser d'hypothèses incertaines et ainsi miner la crédibilité du modèle global.

En plus de préciser la portée de l'étude, la question précise la structure du modèle et les éléments à y incorporer. La structure doit inclure les dimensions spatiale et temporelle, limitée dans l'espace à l'intérieur du Parc Marin Saguenay-Saint-Laurent et circonscrite dans le temps pendant la saison touristique. Les éléments à incorporer dans le modèle sont les rorquals communs et bélugas ainsi que l'environnement du parc marin.

2.2 Étape II : Poser les hypothèses

De manière générale, une hypothèse est une « proposition visant à fournir une explication vraisemblable d'un ensemble de faits, et qui doit être soumise au contrôle de l'expérience ou vérifiée dans ses conséquences. » (*Le Larousse des noms communs*, 2008) Cette définition très généraliste cadre très bien avec l'approche de modélisation orientée par patrons, où *l'ensemble de faits* correspond aux patrons observés du système alors que les *conséquences* sont les patrons prédits par le modèle. Dans ce contexte de modélisation, poser une hypothèse consiste à tenter de représenter un mécanisme sous-jacent qui génère les patrons observés dans le système réel.

Ce travail propose six algorithmes de déplacement différents. Cinq d'entre eux tentent de représenter le déplacement spatio-temporel des mammifères marins en utilisant ou hybridant divers modèles proposés dans la littérature. Finalement, le sixième algorithme constitue une tentative nouvelle de représentation de ces déplacements. Le chapitre III décrit en détail la théorie derrière chacun des algorithmes testés :

1. Marche aléatoire (MA);
2. Marche aléatoire corrélée (MAC);
3. Marche aléatoire corrélée avec indice de résidence (MAC-R);
4. Marche aléatoire corrélée avec influence sociale (MAC-S);
5. Marche aléatoire corrélée avec indice de résidence et influence sociale (MAC-RS);
6. Marche aléatoire corrélée avec minimisation de la moyenne des biais normalisés (MAC-MMBN).

Six hypothèses seront testées, une pour chaque algorithme. Dans chaque cas, l'hypothèse nulle H_0 est que l'algorithme de déplacement simule adéquatément les patrons spatio-temporels de déplacement des mammifères dans la région d'étude.

En fonction de l'utilisation finale du modèle, il a été convenu que le modèle doit reproduire trois catégories de patrons : les patrons de trajectoire, les patrons de répartition et les patrons de socialisation. Le chapitre IV décrit en détail l'ensemble des patrons ainsi que les protocoles utilisés afin d'extraire ces informations à partir des données observées dans le système réel et simulées par le modèle.

2.3 Étape III : Définir la structure du modèle

La question posée à l'étape I et les hypothèses de travail proposées à l'étape II définissent indirectement et de manière qualitative les éléments à incorporer dans le modèle. La troisième étape du processus de modélisation consiste à identifier explicitement les variables d'état et les processus essentiels afin de modéliser correctement le système. Pour simplifier la représentation, il est d'usage de dessiner les liens entre les variables d'état et les processus dynamiques du système qui modifient ces variables. Le schéma ainsi créé donne le premier modèle conceptuel, qui sera par la suite implémenté à l'étape suivante (Grimm et Railsback, 2005).

Les variables définissent l'état d'un système figé dans le temps. Dans un modèle basé sur individus, ces variables d'état sont les attributs de chaque individu qui évolue dans le système. L'objectif est de conserver strictement les variables essentielles pour modéliser correctement le système, sans le complexifier inutilement. Puisque ce projet s'intéresse à la représentation spatio-temporelle des déplacements des mammifères, les attributs essentiels des individus sont leurs emplacements géographiques en trois dimensions et les propriétés du mouvement en cours. Le taux d'oxygénation de l'individu est le dernier attribut inclus dans le modèle afin de simuler le cycle respiratoire des mammifères. Le tableau 2.1 résume l'ensemble des variables d'état de chaque individu. Dans un modèle plus évolué, une

variable définissant l'état comportemental serait une contribution intéressante pour explorer ce domaine qui est au-delà de la portée de ce projet.

Tableau 2.1 Liste des variables d'état d'un mammifère marin

Variable	Unité	Description
Espèce		Numéro de l'espèce
Emplacement	UTM	Emplacement géographique
Profondeur	Mètre	Profondeur de plongée
Direction	Radian	Direction du déplacement en deux dimensions
Vitesse	Mètre/seconde	Vitesse de déplacement
Oxygène	%	Pourcentage d'oxygène restant avant de faire surface

Définir la structure du modèle implique aussi de fixer les échelles spatiale et temporelle. L'échelle d'une dimension comporte deux éléments : le grain et l'étendue. Le grain définit la taille du plus petit élément discernable, alors que l'étendue définit les limites de la dimension couvertes par le modèle. Pour la dimension temporelle, le grain de simulation est fixé à une minute. Ce grain extrêmement petit est motivé par le besoin de simuler grossièrement le cycle respiratoire des mammifères. L'étendue de la dimension temporelle est limitée par la durée de la saison touristique, généralement entre le mois de juin et le mois de septembre. Cependant, l'étendue temporelle est un paramètre très flexible afin de simuler facilement de plus petites portions de la saison touristique.

La dimension spatiale est représentée dans le modèle sous la forme d'une grille référencée géographiquement. Le grain de la dimension spatiale correspond à la grandeur d'une cellule de la grille. Celle-ci est fixée arbitrairement à 100x100 mètres pour conserver un juste équilibre entre la précision des données géographiques et l'utilisation de la mémoire. Dans un même ordre d'idée, l'étendue spatiale correspond à la grandeur totale de la grille. Afin d'éviter les effets nuisibles des rebords, l'étendue spatiale se doit d'être plus grande que la

zone de l'étude. Ainsi, la grille comporte 1983x2205 cellules pour une étendue totale de 43725 km².

À partir de ce point, la structure statique du modèle est totalement définie. Pour donner vie au modèle, il faut ajouter les processus dynamiques qui modifient les variables d'état en cours de simulation. Dans ce projet, deux processus déterminent le prochain mouvement du mammifère, soit une règle déterministe de plongée et un algorithme de déplacement. Ces deux processus dynamiques détaillés ci-dessous reliés avec les variables d'état permettent de dessiner le premier modèle conceptuel du système illustré à la figure 2.2.

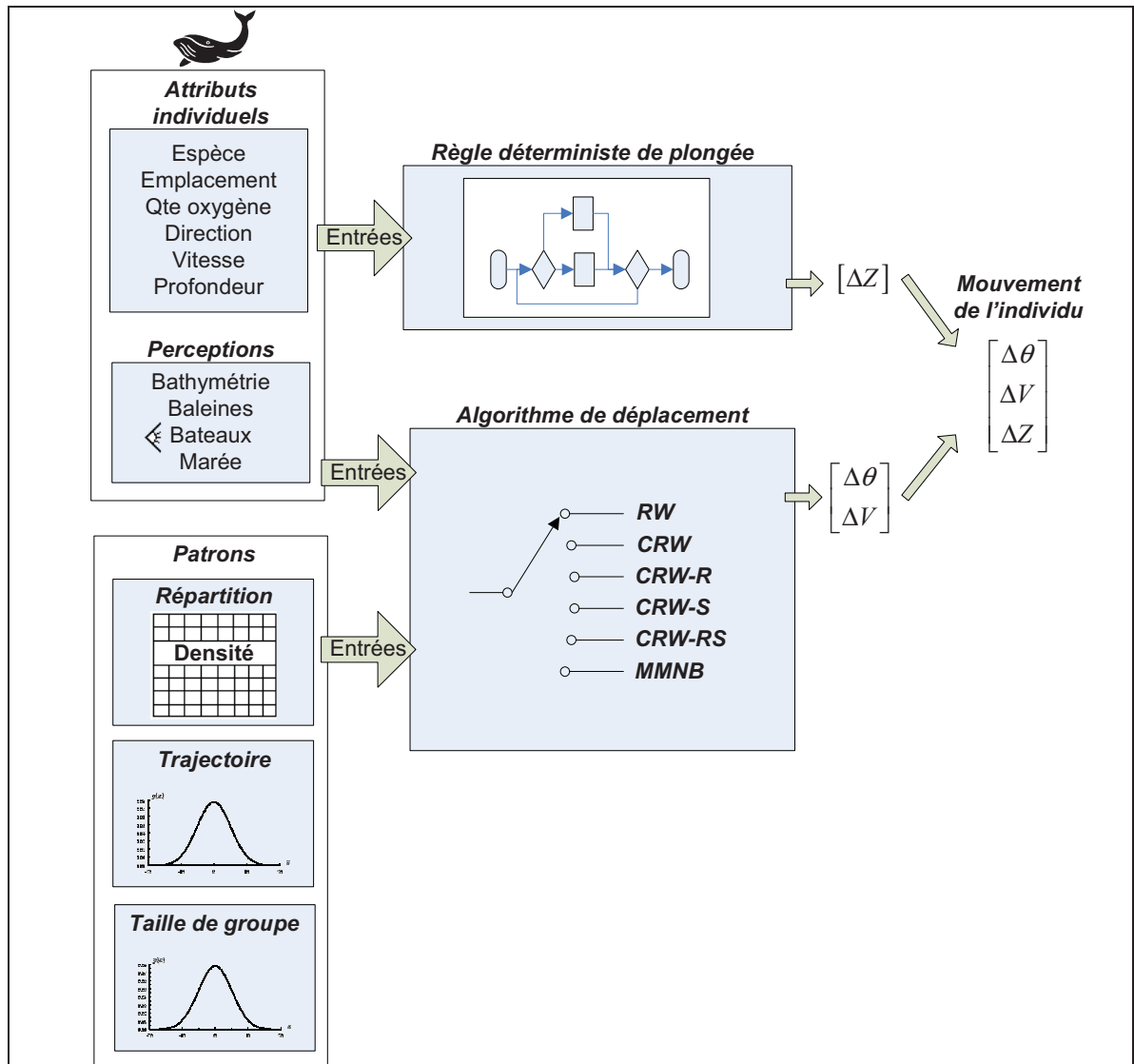


Figure 2.2 Modèle conceptuel du processus de déplacement d'un mammifère marin.

Règle déterministe de plongée

L'intervalle respiratoire est un processus qui n'est que grossièrement abordé dans le modèle. Ce processus est trop important pour l'ignorer, car c'est précisément la présence en surface des mammifères marins qui permet à l'industrie récréotouristique d'offrir des croisières d'observation aux baleines. Cependant, le mécanisme sous-jacent qui crée le cycle respiratoire est inutilement complexe pour ce projet.

Ainsi, la profondeur de chaque individu est générée de manière déterministe à l'aide de règles simples afin de reproduire le plus fidèlement possible son cycle de respiration. Chaque espèce perd ou gagne un pourcentage fixe d'oxygène s'il est en cours de plongée ou en surface. Ces deux constantes sont calculées selon l'équation 2.1.

$$\begin{aligned}\Delta\%Oxygène_{Plongée} &= \frac{-1}{Durée_{Plongée}} \cdot 100\% \mid profondeur_{Individu} < 0 \\ \Delta\%Oxygène_{Surface} &= \frac{1}{Durée_{Surface}} \cdot 100\% \mid profondeur_{Individu} = 0\end{aligned}\tag{2.1}$$

Les mammifères vont à la surface lorsque leur taux d'oxygène est égal ou en dessous de zéro. Une fois à la surface, ils y restent jusqu'à ce que leur taux d'oxygène soit de 1 et replongent par la suite. Le fait que les règles déterministes de plongée soient encapsulées dans un sous-modèle distinct permettra de complexifier graduellement le cycle respiratoire des individus dans une prochaine version du modèle.

Algorithme de déplacement individuel

En cours de simulation, les individus doivent déterminer les mouvements à effectuer. Pour implémenter et intégrer facilement n'importe quelle règle de déplacement dans le modèle, un sous-modèle abstrait impose une structure rigide.

La sortie de ce sous-modèle est le mouvement à effectuer pour le prochain intervalle de temps, défini par un angle de braquage $\Delta\theta$ et une vitesse de déplacement V . Cependant, les entrées de ce sous-modèle varient en fonction de la complexité de l'algorithme à implémenter. Pour un algorithme simple tel que la marche aléatoire, les seules entrées nécessaires sont la vitesse et la durée moyennes des déplacements. À l'opposé, l'algorithme de minimisation de la moyenne des biais normalisés nécessite la taille moyenne des groupes en fonction de la marée, la carte de répartition de l'espèce et les fonctions de distribution cumulative des vitesses, des durées et des angles de braquage des mouvements observés et simulés.

2.4 Étape IV : Implémenter le modèle

À partir du modèle conceptuel dessiné sous la forme d'un schéma, l'étape de l'implémentation consiste à convertir ce modèle sous la forme d'un programme informatique. L'objectif de cette conversion est d'exploiter la rapidité des ressources informatiques afin de donner vie au modèle en simulant l'évolution des individus dans l'espace et dans le temps.

Plusieurs raisons nous ont amenés à utiliser le langage de programmation Java dans le cadre de ce projet. En premier lieu, Java est un langage de programmation orientée objet. La structure imposée par un langage orienté objet permet de traduire intuitivement un modèle conceptuel basé sur individus sous la forme d'un programme informatique. Ainsi, chaque classe représente une espèce, alors que chaque instance de la classe représente un individu dans le système. Pour compléter l'analogie, les variables d'état du modèle conceptuel sont les attributs des classes et finalement les processus dynamiques qui modifient les variables d'état sont les méthodes des classes. Bien qu'il existe plusieurs autres langages de programmation orientée objet, Java a l'avantage d'être un langage qui est compatible avec la majorité des systèmes d'exploitation. Cette particularité est très importante, car elle permet d'exécuter le modèle sans modification ou compilation entre le centre de recherche, les partenaires et les machines de calcul qui utilisent respectivement les systèmes d'exploitation OSX, Windows et possiblement Linux.

Il existe plusieurs plateformes logicielles qui facilitent l'implémentation d'un modèle basé sur individus. Ces plateformes, généralement des logiciels libres maintenus par des groupes de recherche, sont des ensembles de fonctions ou même de programmes qui implémentent toutes les fonctionnalités récurrentes d'un modèle à l'autre. L'utilisation d'une plateforme de modélisation est une pratique largement répandue et encouragée. L'utilisation de celle-ci permet d'accélérer le temps de développement, de réduire le risque d'erreurs de programmation et de standardiser les différents modèles dans la communauté scientifique. Actuellement, quelques plateformes s'imposent dans le développement de modèles basés sur individus : NetLogo, Swarm, Mason et Repast (Railsback, Lytinen et Jackson, 2006). Dans

le cadre de ce projet, la plateforme Repast Symphony (North et al., 2005) fondée sur le langage Java a été sélectionnée. Les principales fonctionnalités utilisées de cette plateforme sont les modules d'ordonnancement d'exécution des individus, d'enregistrement des données en cours de simulation, d'exécution de simulations en lot et de visualisation des individus en trois dimensions ou dans l'environnement WorldWind développé par la NASA.

À partir de la plateforme Repast Symphony, il reste à implémenter les classes des différentes espèces de mammifères marins pour compléter le modèle informatique. Afin de structurer adéquatement le modèle basé sur individus, la notion d'héritage a été utilisée. La classe mère, *Agent*, est une classe abstraite qui contient toutes les fonctionnalités pour créer un individu fonctionnel qui se déplace dans le modèle. Au second niveau hiérarchique, la classe *Whale* est aussi une classe abstraite qui hérite de la classe *Agent*. À ce niveau, les attributs d'un mammifère marin sont définis ainsi que les algorithmes pour calculer le prochain mouvement. Finalement au troisième niveau hiérarchique, les classes instanciables *Whale_Beluga* et *Whale_Rorqual_Commune* héritent de la classe *Whale* et contiennent les paramètres de déplacement propre à chaque espèce. Pour bien illustrer nos propos, la figure 2.3 présente le diagramme de classe simplifié du modèle et la figure 2.4 montre la séquence d'exécution et l'interaction entre les classes à chaque pas de temps.

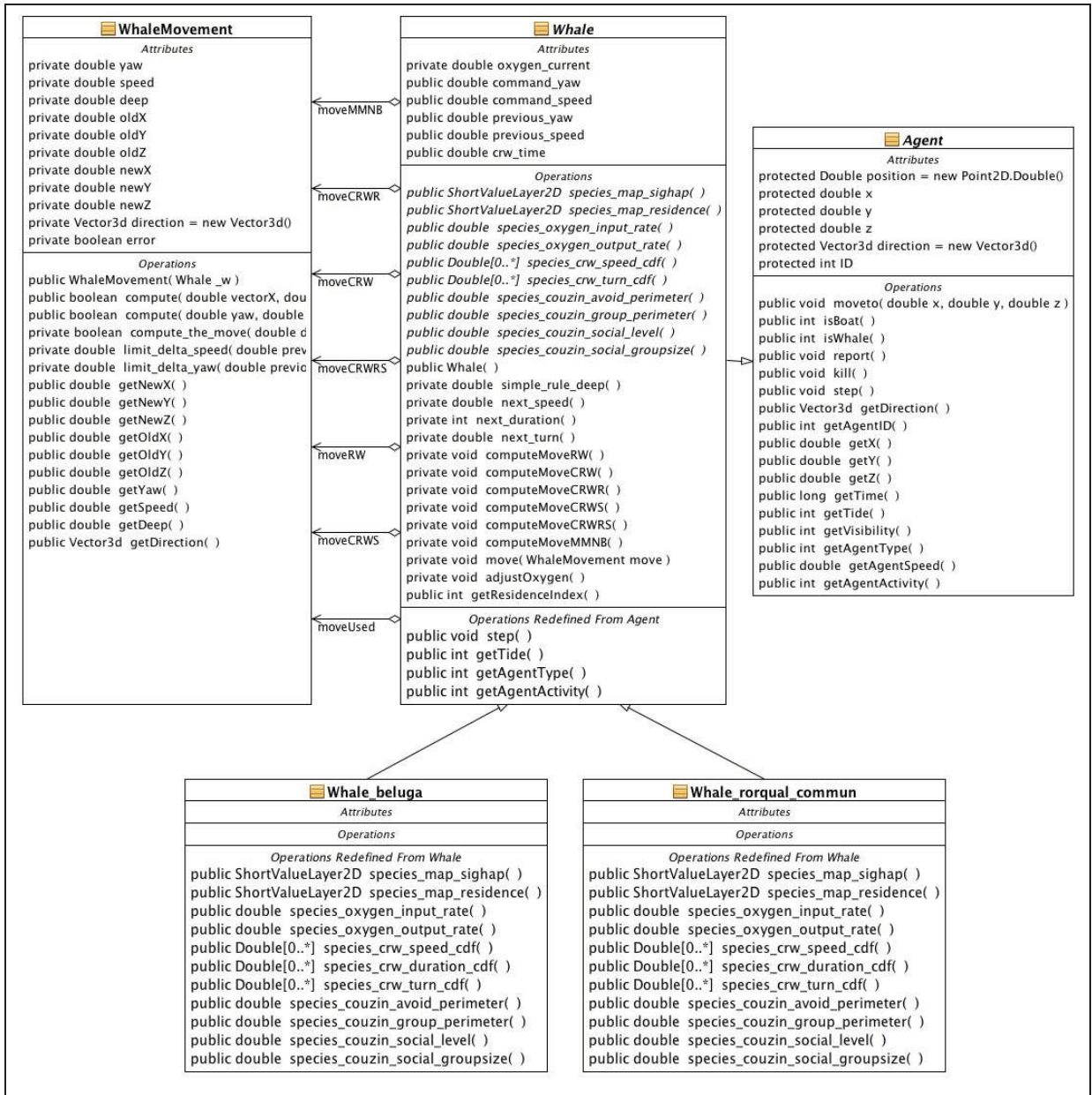


Figure 2.3 Diagramme de classe simplifié du modèle.

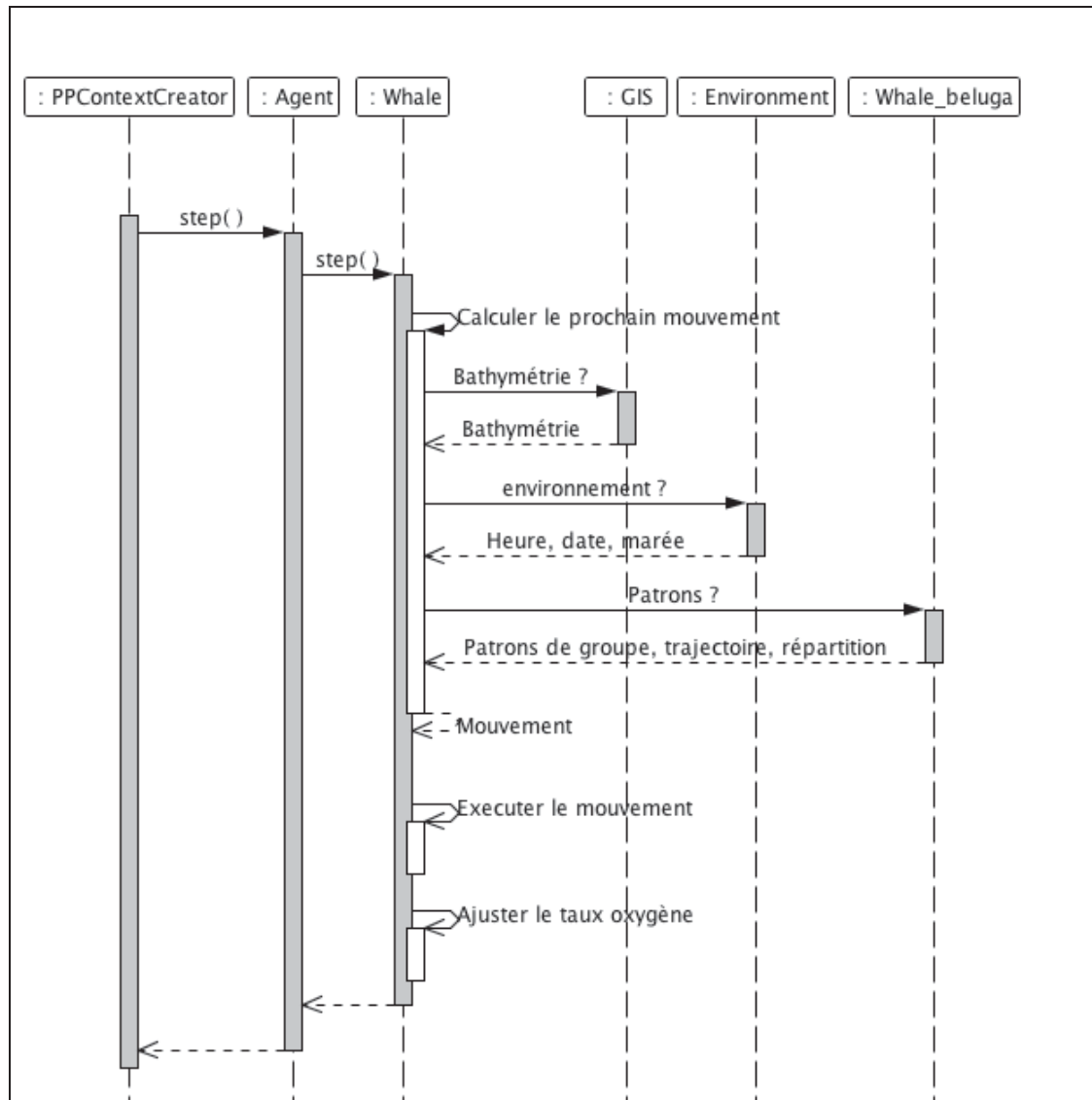


Figure 2.4 Séquence d'exécution d'un pas de temps.

2.5 Étape V : Analyser le modèle

L'analyse des modèles consiste à évaluer la capacité des différents algorithmes de déplacement à répondre à la question initiale du modèle (Grimm et Railsback, 2005). Pour structurer cette phase d'analyse, l'approche de modélisation orientée par patrons compare systématiquement tous les patrons spatio-temporels des mammifères observés et simulés.

Le déroulement des expérimentations est rigoureusement identique pour les six algorithmes afin de ne pas biaiser les résultats. La durée de simulation est fixée à 30 jours avec un grain d'une minute. La durée d'un mois permet aux individus de se déplacer partout dans leur environnement et évite de donner trop d'importance à l'état initial du modèle. De plus, 24 réplifications sont effectuées pour chaque simulation afin de calculer l'intervalle de confiance avec le test t de Student. Le nombre d'individus à l'intérieur du modèle est de 100 bélugas et de 35 rorquals communs. Ces nombres découlent de discussions informelles avec les différents partenaires du projet. Cependant, le nombre d'individus à l'intérieur du modèle est un paramètre à fixer par l'utilisateur lors de la simulation d'un scénario de gestion.

En cours de simulation, les données enregistrées chaque minute par chaque individu sont décrites dans le tableau 2.2.

Tableau 2.2 Les valeurs enregistrées pour chaque individu

Variable	Type	Description
Expérimentation	Entier	Numéro d'identification de l'expérimentation
Réplication	Entier	Numéro d'identification de la réplication
Tick	Entier	Numéro de l'itération de la simulation
Agent_Type	Entier	L'espèce de baleine
Agent_ID	Entier	Numéro d'identification de l'individu
X_UTM	Double	Position géographique sur l'axe des x
Y_UTM	Double	Position géographique sur l'axe des y
Groupe	Entier	Nombre d'individus de la même espèce environnant
Marée	Entier	Type de la marée (montante, haute, descendante, basse)

Le grain temporel très petit de l'ordre de la minute conjugué à une étendue temporelle de 24x30 jours de simulation nécessite de prendre en considération l'espace de stockage

nécessaire pour chaque expérimentation. En estimant la taille d'un enregistrement à 56 octets, l'équation 2.2 prédit un fichier 7.3 gigaoctets par expérimentation.

$$24 \text{ rép.} \times 30 \text{ jours} \times 135 \text{ ind.} \times 1440 \text{ min.} = 139.968.000 \text{ enregistrements}$$

$$139.968.000 \text{ enregistrements} \times 56 \frac{\text{octets}}{\text{enregistrement}} \cong 7475 \text{ Mo} \quad (2.2)$$

À la fin d'une expérimentation, le temps d'exécution est noté pour décrire grossièrement la complexité de chaque algorithme. Par la suite, les patrons observés, extraits à la section 4.2, sont comparés aux patrons simulés, extraits à la section 4.3. Le processus d'évaluation, décrit à la section 4.4, compare systématiquement chaque paire de patrons pour évaluer leur similarité. Finalement, l'analyse et la discussion de ces résultats sont l'objet du chapitre V. La figure 2.5 illustre le processus de modélisation complet utilisé dans le cadre de ce projet.

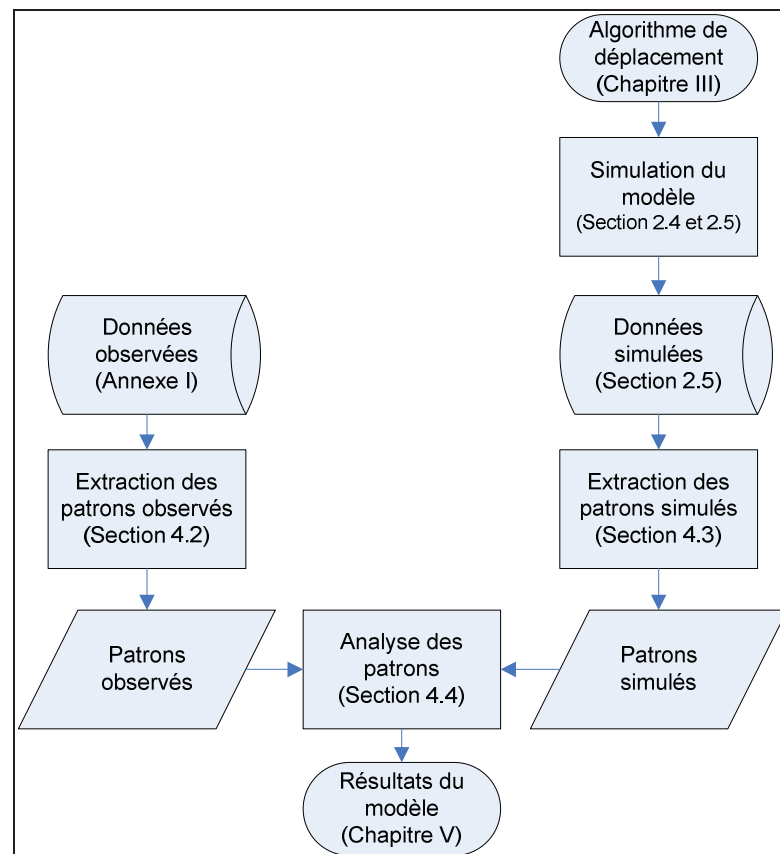


Figure 2.5 Processus de modélisation utilisé dans ce projet

CHAPITRE 3

ALGORITHMES DE DÉPLACEMENT DES INDIVIDUS

« If the simplest model does not reproduce an essential pattern, it has to be replaced by a model with a more adequate structure »

-Weigland, 2003

Ce chapitre présente les six algorithmes de déplacement des individus implémentés dans le modèle. La présentation de chaque algorithme est divisée en trois sections. En premier lieu, une brève introduction situe l'algorithme dans la littérature scientifique. Par la suite, l'implémentation de l'algorithme est détaillée textuellement suivie de son pseudo-code dans un encadré. Finalement, les avantages et les limitations théoriques de l'algorithme sont discutés.

La méthode de Monte Carlo est régulièrement utilisée dans tous les algorithmes employés dans ce projet. Puisqu'il existe un flou autour de cette méthode, nous définissons ici la méthode de Monte Carlo comme l'utilisation d'une distribution de probabilités d'un phénomène et d'un générateur de nombres aléatoires dans le but de piger une valeur stochastique (Judson, 1994).

Un autre élément commun à tous les algorithmes de déplacement est la gestion des frontières (Codling, Plank et Benhamou, 2008). Il existe deux types de frontières, soit les frontières naturelle et artificielle. Les frontières naturelles sont les berges qui délimitent la zone navigable de l'environnement. Cependant, les frontières artificielles sont des cartes dans le modèle qui délimitent l'habitat généralement utilisé en fonction de l'espèce. Conceptuellement, ces frontières sont dites réfléchives et non absorbantes. Ce type de frontière permet à un mammifère qui rencontre une frontière de changer de direction pour l'éviter, à l'opposé d'une frontière absorbante qui signifierait que le mammifère s'échoue sur la berge.

3.1 Algorithme de marche aléatoire

L'ancêtre de la marche aléatoire est le mouvement Brownien, créé initialement pour modéliser de manière continue dans l'espace et le temps le mouvement de minuscules particules dans le fluide d'un grain de pollen (Nelson, 1967). Par la suite, la marche aléatoire a été définie comme une formulation discrétisée dans le temps et l'espace du mouvement Brownien. Bien qu'il existe plusieurs déclinaisons, l'algorithme de marche aléatoire conventionnel est non biaisée et non corrélée. L'absence de biais signifie que l'individu n'a pas de préférence d'habitat ou de trajectoire migratrice. Dans un autre ordre d'idée, l'absence de corrélation signifie qu'il n'y a aucune persistance dans la direction de l'individu.

Implémentation de l'algorithme

L'algorithme de la marche aléatoire non biaisée et non corrélée est très simple à implémenter. Puisque la direction du mouvement est indépendante des directions précédentes et équiprobable dans toutes les directions, les individus n'ont pas à mémoriser les caractéristiques des mouvements précédents. Ainsi, il suffit de piger un nombre aléatoire dans l'intervalle $[0, 2\pi]$ pour déterminer la direction du prochain mouvement. La durée et la vitesse du mouvement sont tout aussi simples à déterminer. Bien que les développements théoriques de la marche aléatoire utilisent généralement une distance de déplacement fixe à chaque pas de temps, Turchin propose une approche alternative plus réaliste en utilisant la méthode de Monte Carlo (Turchin, 1998). À partir de la table des centiles, un nombre aléatoire entre le 1^{er} et le 100^e centile est pigé pour obtenir une vitesse et une durée de mouvement réalistes et non corrélées avec le mouvement précédent. À partir de ces trois caractéristiques, le prochain mouvement est entièrement défini et l'individu effectue le mouvement pendant la durée déterminée avant de piger un nouveau mouvement.

```

SI DUREE <= 0 ALORS
    DIRECTION =  $2\pi \cdot \xi$ 
    VITESSE =  $CENTILE_{Vitesse}(\xi)$ 
    DUREE =  $CENTILE_{Duree}(\xi)$ 
SINON
    DUREE = DUREE - 1
FIN SI

```

Avantages et limitations

L'avantage principal de cet algorithme est sa simplicité. Généralement nommé le modèle nul, c'est une stratégie recommandée d'amorcer le cycle de modélisation avec un modèle très simplifié tel que la marche aléatoire (Grimm et Railsback, 2005). Cette simplicité permet d'implémenter et de tester rapidement un premier modèle puisque les résultats sont facilement compréhensibles. Finalement, ce modèle nul permet d'avoir un modèle de référence simple pour comparer par la suite les modèles beaucoup plus complexes.

Par contre, ce premier modèle comporte de nombreuses limitations. L'algorithme de marche aléatoire utilise peu de patrons comme source d'information. Il utilise les patrons de vitesse et de durée du mouvement, mais ignore les patrons de l'angle de braquage, la taille de groupe et la répartition de l'espèce. Ainsi, il est peu probable que tous ces patrons observés soient répliqués en cours de simulation.

3.2 Algorithme de marche aléatoire corrélée

La marche aléatoire corrélée est une évolution naturelle de l'algorithme de la marche aléatoire simple. La marche aléatoire corrélée complexifie l'algorithme précédent en prenant compte de la corrélation à court terme de la direction d'un individu. Généralement, cette corrélation se traduit par une trajectoire avec des mouvements dans une direction similaire, connue sous le nom de persistance (Codling, Plank et Benhamou, 2008). Pour mettre en relation la direction entre deux mouvements consécutifs, la notion de direction absolue de la

marche aléatoire est remplacée par la notion d'angle de braquage. Ainsi, la direction du prochain mouvement est calculée en additionnant la direction précédente à un angle de braquage pigé de manière stochastique.

Implémentation de l'algorithme

L'implémentation de l'algorithme de marche aléatoire corrélée reprend plusieurs éléments de l'algorithme précédent. En effet, la vitesse et la durée du prochain mouvement sont pigées en utilisant la méthode de Monte Carlo. Cependant, l'angle de braquage qui remplace la direction absolue n'est pas pigé à partir d'une distribution uniforme $[0, 2\pi]$ comme son prédécesseur. L'algorithme utilise plutôt la méthode de Monte Carlo avec une distribution de probabilités du cosinus de l'angle de braquage. La fonction cosinus permet d'ignorer le sens de rotation et d'obtenir une valeur numérique identique pour un angle de braquage de 10 ou de 350 degrés. Cependant, puisque la fonction $\cos^{-1}()$ retourne toujours la portion positive, l'angle de braquage pigé est multiplié par la fonction $signum()$ qui retourne une valeur unitaire négative une fois sur deux en utilisant un second nombre aléatoire. Ainsi, l'algorithme de marche aléatoire corrélée définit entièrement le prochain mouvement en retournant l'angle de braquage, la vitesse et la durée de celui-ci.

```

SI DUREE <= 0 ALORS
    ANGLE =  $signum(\xi_1 - 0.5) \cdot \cos^{-1}(CENTILE_{Psi}(\xi_2))$ 
    VITESSE =  $CENTILE_{Vitesse}(\xi)$ 
    DUREE =  $CENTILE_{Duree}(\xi)$ 
SINON
    DUREE = DUREE - 1
FIN SI

```

Avantages et limitations

La simplicité de ce second algorithme amène son lot d'avantages et de limitations. L'avantage majeur de l'algorithme de marche aléatoire corrélée est qu'il utilise les trois

patrons de trajectoires observés : la vitesse, la durée et l'angle de braquage. Ainsi, l'enchaînement des mouvements est plus réaliste puisqu'il conserve la propriété de persistance de la direction propre à chaque espèce. Une autre propriété intéressante de la marche aléatoire corrélée qui n'est pas abordée dans ce mémoire est la distance moyenne quadratique (mean squared displacement) parcourue par les individus. Cette distance est dérivable à partir de la moyenne de l'angle de braquage et permettrait de calculer un nouveau patron afin de valider le modèle.

Cependant, bien que cet algorithme utilise les trois propriétés de la trajectoire, il faut supposer que la distribution des angles soit symétrique. Cette supposition est nécessaire puisque la fonction cosinus élimine les informations sur le sens de rotation (Turchin, 1998). Une seconde limitation de cet algorithme est qu'il ne tient pas compte des individus environnants. Ainsi, il est impossible d'implémenter directement le patron de la taille de groupe pour les espèces grégaires. Finalement, il est impossible d'utiliser une carte de répartition comme source d'informations supplémentaires afin de distribuer correctement les individus dans l'espace, contrairement aux algorithmes présentés dans les sections suivantes.

3.3 Algorithme de marche aléatoire corrélée avec indice de résidence

L'algorithme de marche aléatoire corrélée peut être complexifié en ajoutant le concept d'indice de résidence (Turchin, 1998). L'indice de résidence permet de résumer l'effet des conditions environnementales sur les mouvements des individus. Cet indice est proportionnel à la répartition des individus d'une région par rapport à l'autre. Ce concept, dérivé du modèle de Fokker-Planck, prédit que les individus s'accumuleront aux endroits où la mobilité est faible, contrairement aux endroits dont la densité d'individus est faible et la mobilité est élevée. Cet algorithme a la force de démontrer théoriquement ce concept intuitif de la diffusion. Ainsi, les endroits où les mammifères pratiquent la nage rapide et directionnelle ont une densité plus faible qu'un endroit où ils nagent de manière lente et aléatoire.

Conceptuellement, l'approche proposée est très intéressante car elle permet de faire le lien entre les mouvements des individus et leur répartition spatiale en utilisant deux approches diamétralement opposées. D'un côté, les modèles de diffusion théoriques utilisent les paramètres de la marche aléatoire pour prédire l'indice de résidence alors que les modèles basés sur individus peuvent facilement simuler le déplacement par la marche aléatoire et calculer empiriquement l'indice de résidence. Pour illustrer ces propos, une preuve de concept a été développée par Turchin (Turchin, 1991) qui simule le déplacement de la coccinelle et prédit correctement, à partir des mouvements locaux, la répartition spatiale des insectes. Pour notre projet, le raisonnement inverse est nécessaire. La carte de répartition est une composante connue de l'algorithme et les caractéristiques des mouvements doivent être ajustées en conséquence pour reproduire simultanément les patrons de mouvement et de répartition de chaque espèce.

Implémentation de l'algorithme

L'algorithme de marche aléatoire corrélée avec indice de résidence a plusieurs fonctions en commun avec les deux algorithmes précédents. La vitesse du prochain mouvement est déterminée de la même manière que la marche aléatoire, en utilisant la méthode de Monte Carlo avec une distribution de probabilités des vitesses pour chaque espèce. La direction du prochain mouvement est déterminée par un angle de braquage de la même manière que l'algorithme de marche aléatoire corrélée, par la méthode de Monte Carlo. Cependant, la durée du prochain mouvement est calculée en utilisant la notion d'indice de résidence.

Pour déduire l'équation de la durée du mouvement en fonction de l'indice de résidence, il faut revenir à la formule originale de l'indice de résidence (Turchin, 1998). À noter qu'il y a plusieurs suppositions, décrites dans la section limitations, nécessaires afin d'obtenir l'équation simplifiée 3.1.

$$\rho = \frac{\tau}{m_1^{2-\psi}} \quad (3.1)$$

À partir de cette équation, il est possible d'isoler la distance m_1 d'un mouvement en fonction de l'indice de résidence ρ , la vitesse v et l'angle de braquage ψ en substituant la durée τ au quotient de la distance par la vitesse ($\tau = m_1 / v$).

$$m_1 = (\rho \cdot v)^{\left(\frac{1}{\psi-1}\right)} \quad (3.2)$$

Finalement, l'équation désirée 3.3 de la durée en fonction de l'indice de résidence est trouvée en utilisant l'équation précédente de la distance du mouvement et la même relation ($\tau = m_1 / v$).

$$\tau = \frac{m_1}{v} = \frac{(\rho \cdot v)^{\left(\frac{1}{\psi-1}\right)}}{v} \quad (3.3)$$

À partir de cette nouvelle équation, la stratégie pour déterminer les trois caractéristiques d'un prochain mouvement est définie. Cependant, il reste à déterminer l'indice de résidence maximale pour convertir la carte de répartition sous la forme d'une carte d'indice de résidence nécessaire dans l'algorithme.

```

SI DUREE <= 0 ALORS
    ANGLE =  $signum(0.5 - \xi_1) \cdot \cos^{-1}(CENTILE_{Psi}(\xi_2))$ 
    VITESSE =  $CENTILE_{Vitesse}(\xi)$ 
    DUREE =  $\frac{(\rho \cdot v)^{\left(\frac{1}{\psi-1}\right)}}{v}$ 
SINON
    DUREE = DUREE - 1
FIN SI

```

Calcul de l'indice de résidence maximal

L'indice de résidence est un indice relatif de la répartition de l'espèce. Une zone dont l'indice de résidence est de $\rho = 4$ contient théoriquement deux fois plus d'individus qu'une zone avec un indice de résidence de $\rho = 2$. Cependant, puisque cet indice est relatif, il est nécessaire de fixer une valeur pour convertir la carte de répartition dans ce nouveau référentiel. L'objectif est de fixer la valeur de l'indice de résidence maximal ρ_{\max} pour convertir la densité des individus $u_{(x,y)}$ à un endroit spécifique sous la forme d'indice de résidence $\rho_{(x,y)}$ à partir de l'équation 3.4.

$$\rho_{(x,y)} = \frac{\mu_{(x,y)} \rho_{\max}}{\mu_{\max}} \quad (3.4)$$

La limite maximale de l'indice de résidence ρ_{\lim} est contrainte par la durée minimale de déplacement. Le déplacement doit avoir une durée minimale de t_{step} secondes afin que le mammifère n'effectue qu'un seul mouvement à l'intérieur d'une unité de temps. Puisque la durée du mouvement est minimale lorsque l'indice de résidence est maximal, alors il est possible de transformer l'équation 3.3 sous la forme :

$$t_{step} = \frac{(\rho_{\lim} \cdot v)^{\left(\frac{1}{\psi-1}\right)}}{v} \quad (3.5)$$

À partir de l'équation 3.5, la limite supérieure de l'indice de résidence peut être isolée. Ainsi, cette valeur est fixée en fonction du cosinus moyen de braquage ψ , la vitesse moyenne v et la durée de déplacement minimale t_{step} tel que :

$$\rho_{\lim} = \frac{t_{step}}{(t_{step} \cdot v)^{2-\psi}} \quad (3.6)$$

L'indice de résidence maximal ρ_{\max} est ainsi contraint entre 0 et ρ_{\lim} . L'objectif est de trouver la valeur exacte de ρ_{\max} qui permet de minimiser l'erreur des patrons de la distance des mouvements simulés et observés. Cet objectif peut être abordé sous la forme d'un problème d'optimisation non linéaire, avec une fonction objectif, une variable de décision et des contraintes.

La fonction objectif de ce problème, l'équation 3.7, tente de minimiser l'erreur absolue de la moyenne de la distance des mouvements observés et simulés.

La variable de décision ρ_{\max} , définie à l'équation 3.8, fixe la valeur maximale de l'indice de résidence. Cette constante sert à convertir la carte de répartition de chaque espèce en carte de résidence avec l'équation 3.4.

Les cinq contraintes sont définies à l'équation 3.9. La première contrainte fixe la vitesse moyenne des mammifères v à celle observée dans les suivis de trajectoires. Similairement, la seconde contrainte fixe le cosinus moyen de l'angle de braquage ψ à celui extrait des données observées. La troisième contrainte calcule l'indice de résidence limite ρ_{\lim} . Cet indice de résidence est calculé à partir de la durée minimale de déplacement de t_{step} afin qu'un mammifère effectue un seul nouveau mouvement par unité de temps. Par la suite, la quatrième contrainte impose que l'indice de référence maximal soit inférieur à cet indice de référence limite. Finalement, la cinquième contrainte calcule la longueur de déplacement moyenne m_1 en fonction de la variable ρ_{\max} . L'intégration de la probabilité qu'un mouvement soit initié à cet indice de résidence $P(\rho)$ multiplié par la longueur moyenne de mouvement pour cet indice de résidence $m(\rho)$ est calculée de manière numérique compte tenu de la nature discrète de la carte de répartition.

Fonction objectif :

$$\text{Minimiser } E_m = |m_{patron} - m| \quad (3.7)$$

Variable de décision :

$$\rho_{max} \mid \rho_{max} \in \mathbb{R} \quad (3.8)$$

Contraintes :

$$\begin{aligned} v &= \bar{v}_{patron} \\ \psi &= \bar{\psi}_{patron} \\ \rho_{limit} &= \frac{t_{step}}{(t_{step} \cdot v)^{2-\psi}} \\ \rho_{max} &\leq \rho_{limit} \\ m &= \int_0^{\rho_{max}} P(\rho) \cdot m(\rho) \cdot \partial\rho \end{aligned} \quad (3.9)$$

Avantages et limitations

L'avantage principal de l'algorithme de marche aléatoire corrélée avec indice de résidence par rapport à ses prédécesseurs est qu'il tente de reproduire simultanément les patrons de la trajectoire et le patron de répartition spatiale de l'espèce. Conceptuellement, c'est une approche très intéressante qui s'inspire de l'approche ascendante (*bottom-up*) de la modélisation orientée par patrons. Ainsi, l'algorithme tente de reproduire un patron global, la répartition spatiale, à partir de patrons locaux qui sont les caractéristiques de mouvement.

Toutefois, cette approche a plusieurs limitations qui lui sont propres ou transmises par son prédécesseur la marche aléatoire corrélée. Ainsi, la distribution des angles de braquage doit être symétrique autour de zéro et la direction des mouvements ne doit pas être affectée ni par une attraction lointaine ni par la direction du compas. De plus, l'interaction entre les individus n'est pas prise en compte au cours du processus décisionnel.

En plus de ces limitations, il a été nécessaire de poser deux nouvelles suppositions pour obtenir l'équation simplifiée de l'indice de résidence. La première supposition est que le degré de persistance des mouvements ψ ne change pas en fonction de l'indice de résidence, mais uniquement la durée et la fréquence des mouvements. La seconde supposition implique que la distribution de distance des mouvements soit exponentielle. Cette forme spécifique de la distribution permet de simplifier grandement l'équation jusqu'à l'obtention de la forme de l'équation 3.1.

3.4 Algorithme de marche aléatoire corrélée avec influence sociale

Parmi les trois premiers algorithmes de déplacement présentés, une limitation récurrente qui n'a pas encore été abordée est l'interaction entre les individus. L'interaction entre les mammifères est facilement observable chez les espèces grégaires et il en résulte un patron de taille de groupe spécifique pour chaque espèce.

Pour modéliser ces interactions et tenter de reproduire le patron de taille de groupe, l'algorithme proposé par Couzin est utilisé (Couzin et al., 2005). Celui-ci propose de rapprocher les vecteurs de position et d'aligner les vecteurs de direction entre les individus environnants afin de créer un groupe compact et qui se dirige dans la même direction. Cet algorithme peut être utilisé simultanément avec n'importe quel algorithme de déplacement puisqu'il modifie un mouvement individuel en ajoutant une interaction sociale. Cette approche incrémentale est utilisée dans les deux prochains algorithmes pour ajouter une influence sociale dans la marche aléatoire corrélée et la marche aléatoire corrélée avec indice de résidence.

Implémentation de l'algorithme

L'algorithme de marche aléatoire corrélée avec influence sociale est un algorithme qui s'implémente en trois étapes. En premier lieu, le mouvement de l'individu est déterminé sans tenir compte des individus dans son voisinage. Par la suite, un second mouvement est calculé, indépendamment du premier, pour regrouper les individus environnants.

Finalement, la dernière étape consiste à combiner les deux mouvements en fonction du comportement plus grégaire ou solitaire de l'individu.

La première étape permet de déterminer le mouvement désiré de l'individu. Que ce soit pour éviter une région de prédation, suivre une trajectoire migratrice ou se diriger vers une source de nourriture, le mouvement désiré est celui qui serait effectué si l'individu n'avait aucune interaction sociale. Pour cet algorithme, le mouvement individuel utilisé est la marche aléatoire corrélée, décrit à la section 3.3, qui retourne un vecteur unitaire de direction g_i et la vitesse h_i du mouvement.

La seconde étape consiste à calculer le mouvement social, à l'opposé du mouvement désiré, qui permet de regrouper les individus environnants sans prendre en compte les préférences propres à chaque individu. Les informations perçues par les individus sont le vecteur de position $c_j(t)$, le vecteur de direction $v_j(t)$ et la vitesse $s_j(t)$ des individus j à l'intérieur du rayon d'interaction β . L'individu i doit conserver une distance minimale α avec les autres mammifères j avoisinants. S'il y a un individu dans ce rayon, alors il y a une manœuvre d'évitement décrite à l'équation 3.10.

$$d_i(t + \Delta t) = - \sum_{j \neq i} \frac{c_j(t) - c_i(t)}{|c_j(t) - c_i(t)|} \quad (3.10)$$

Cependant, s'il n'y a aucun mammifère dans le rayon d'évitement de l'individu i , alors celui-ci se rapproche et s'aligne avec les j individus présents dans son rayon d'interaction β selon l'équation 3.11.

$$d_i(t + \Delta t) = \sum_{j \neq i} \frac{c_j(t) - c_i(t)}{|c_j(t) - c_i(t)|} + \sum_{j=1}^n \frac{v_j(t)}{|v_j(t)|} \quad (3.11)$$

$$s_i(t + \Delta t) = \frac{1}{n} \sum_{j=1}^n |s_j(t)|$$

Finalement, la troisième étape consiste à combiner les mouvements individuel et social avec l'équation 3.12. Un facteur de socialisation ω pondère l'importance accordée au mouvement individuel par rapport au mouvement social. Un ω près de zéro accorde plus d'importance à maintenir la cohésion du groupe, près de 1 accorde la même importance aux deux vecteurs alors qu'un ω supérieur à 1 donne plus d'importance à la direction désirée de l'individu.

$$d_i'(t + \Delta t) = \frac{\hat{d}_i(t + \Delta t) + \omega g_i}{|\hat{d}_i(t + \Delta t) + \omega g_i|} \quad (3.12)$$

$$s_i'(t + \Delta t) = \frac{s_i(t + \Delta t) + \omega h_i}{\omega + 1}$$

Enfin, l'algorithme retourne la vitesse $s_i'(t + \Delta t)$ et le vecteur de direction $d_i'(t + \Delta t)$ du prochain mouvement. Pour terminer l'implémentation avec la même structure algorithmique, le vecteur de direction est transformé sous la forme d'un angle de braquage.

```
SI DUREE <= 0 ALORS
```

$$\text{ANGLE} = \text{DIRECTION} - \tan^{-1} \left(\frac{d_{i(x)}'(t + \Delta t)}{d_{i(y)}'(t + \Delta t)} \right)$$

$$\text{VITESSE} = s_i'(t + \Delta t)$$

$$\text{DUREE} = \text{CENTILE}_{\text{Duree}}(\xi)$$

```
SINON
```

$$\text{DUREE} = \text{DUREE} - 1$$

```
FIN SI
```

Avantages et limitations

Conceptuellement, l'algorithme de marche aléatoire corrélée avec influence sociale a l'avantage d'utiliser une approche ascendante. Il tente de modéliser un mécanisme sous-jacent, l'interaction sociale, qui permettrait de reproduire le patron de taille de groupe. De plus, l'algorithme implémente la notion d'attraction et de répulsion qui sera très utile dans un

prochain cycle de modélisation pour simuler l'interaction entre les mammifères et les bateaux.

Malgré ces avantages, il y a aussi plusieurs limitations dans cet algorithme. La paramétrisation de cet algorithme est problématique car il comporte plusieurs paramètres que l'on peut difficilement extraire des données : le rayon d'interaction avec les individus environnants, la distance d'évitement d'un autre individu ou la vitesse angulaire maximale en fonction du type de mammifère. Le paramètre le plus significatif, le facteur de socialisation ω , doit aussi être déterminé afin de reproduire correctement le patron de taille de groupe malgré la présence de plusieurs autres variables telles que la taille de la population qui influence la taille de groupe. Enfin, l'interaction avec les autres individus modifie la trajectoire et empêche à l'algorithme de contrôler pleinement le patron de trajectoire de l'individu contrairement à la marche aléatoire corrélée simple.

3.5 Algorithme de marche aléatoire corrélée avec indice de résidence et influence sociale

L'algorithme de marche aléatoire corrélée avec indice de résidence et influence sociale est la dernière approche qui utilise tout ce qui a été développé jusqu'à maintenant. Sans ajouter aucune nouvelle stratégie, cet algorithme reprend les points forts des trois derniers algorithmes pour tenter de reproduire tous les patrons simultanément. Ainsi, les éléments repris sont l'algorithme de marche aléatoire corrélée pour reproduire la trajectoire, l'indice de résidence afin de contrôler la distribution spatiale et l'influence sociale pour regrouper les individus.

Implémentation de l'algorithme

Pour intégrer tous ces éléments, la structure de cet algorithme est identique à la marche aléatoire corrélée avec influence sociale décrite précédemment. Ainsi, un mouvement individuel et un mouvement social sont calculés pour ensuite être combinés.

Le mouvement individuel est déterminé à partir de l'algorithme de marche aléatoire corrélée avec indice de résidence décrit à la section 3.3. Sans détailler à nouveau ce processus, la vitesse et l'angle de braquage sont calculés en utilisant la méthode de Monte Carlo. Cependant, l'équation de la durée du mouvement utilise la notion d'indice de résidence pour déterminer si l'individu effectue un long mouvement linéaire pour sortir d'un habitat hostile ou de brefs mouvements pour explorer un habitat hospitalier.

Les deux étapes suivantes sont en tous points identiques à l'algorithme de marche aléatoire corrélée avec influence sociale décrit à la section 3.4. L'individu calcule le mouvement social qui tente de rapprocher les vecteurs de position et d'aligner les vecteurs de direction avec les individus environnants. Finalement, le vecteur de direction $d_i'(t + \Delta t)$ et la vitesse de déplacement $s_i'(t + \Delta t)$ sont calculés en pondérant l'importance du mouvement individuel par un facteur de socialisation.

```
SI DUREE <= 0 ALORS
```

$$\text{ANGLE} = \text{DIRECTION} - \tan^{-1} \left(\frac{d_{i(x)}'(t + \Delta t)}{d_{i(y)}'(t + \Delta t)} \right)$$

$$\text{VITESSE} = s_i'(t + \Delta t)$$

$$\text{DUREE} = \frac{(\rho \cdot v)^{\left(\frac{1}{\psi-1}\right)}}{v}$$

```
SINON
```

$$\text{DUREE} = \text{DUREE} - 1$$

```
FIN SI
```

Avantages et limitations

La force de cet algorithme réside encore dans son approche ascendante. Il tente d'expliquer les deux patrons globaux qui sont la taille de groupe et la répartition spatiale en modifiant strictement les paramètres locaux des mouvements.

Cependant, mélanger tous les algorithmes dans une même fonction a aussi des inconvénients. La répartition spatiale prédite par les équations de diffusion n'est plus valable car elle est conditionnelle à l'absence d'interaction entre les individus. De plus, cette interaction sociale entre les individus modifie le patron de trajectoire initialement contrôlé par la marche aléatoire corrélée. Ainsi, la plus grande limitation de cet algorithme est qu'il est impossible de prédire théoriquement le niveau de reproduction des patrons.

3.6 Algorithme de minimisation de la moyenne des biais normalisés

Dans les cinq algorithmes présentés précédemment, le modèle tente de reproduire les patrons de répartition et de taille de groupe de manière indirecte. Dans un premier temps, ces deux patrons sont tout simplement ignorés par les algorithmes de marche aléatoire (3.1) et de marche aléatoire corrélée (3.2). Par la suite, le concept d'indice de résidence (3.3) est introduit pour tenter de reproduire de manière indirecte le patron de répartition en modifiant la durée des déplacements en fonction de la densité observée. De l'autre côté, l'algorithme de marche aléatoire corrélée avec influence sociale (3.4) tente de simuler le patron de taille de groupe, toujours de manière indirecte, en paramétrant le taux de socialisation de chaque espèce. Finalement, l'algorithme de marche aléatoire corrélée avec indice de résidence et influence sociale (3.5) réunit ces deux stratégies pour tenter de reproduire simultanément ces deux patrons.

Contrairement aux algorithmes précédents, l'approche par minimisation de la moyenne des biais normalisés consiste à reproduire les patrons de trajectoire, de répartition spatiale et de taille de groupe en calculant directement leur fréquence d'occurrence. Ainsi, une mémoire collective est nécessaire pour chaque espèce afin de conserver l'ensemble des mouvements effectués depuis le début de la simulation. À partir de ces informations, le prochain mouvement est déterminé avec l'objectif de minimiser la différence entre les fonctions de répartition des patrons observés et simulés.

Cet algorithme utilise avec subtilité les notions d'erreur et de biais. Ainsi, un biais consiste à soustraire une valeur simulée par la valeur observée comparativement à une erreur qui est la valeur absolue de cette soustraction.

Implémentation de l'algorithme

Tout comme l'algorithme de type marche aléatoire, le déplacement des individus est composé d'une séquence de mouvements définis par une vitesse, une durée et un angle de braquage. La vitesse et la durée du prochain mouvement sont deux attributs simples à déterminer car ils ont peu d'influence sur les autres patrons de trajectoire, de taille de groupe ou de répartition. Ainsi, la vitesse et la durée sont pigées aléatoirement à partir de leur fonction empirique de centile selon le même processus que la marche aléatoire, où ξ est un nombre aléatoire pigé entre 1 et 100.

$$\begin{aligned} v_i &= CENTILE_{Vitesse}(\xi) \\ t_i &= CENTILE_{Duree}(\xi) \end{aligned} \quad (3.13)$$

Cependant, la direction du prochain mouvement est beaucoup plus complexe à déterminer puisqu'il influence simultanément trois patrons : l'angle de braquage, la taille de groupe et la répartition spatiale de l'espèce. De plus, le choix de l'angle de braquage est complexifié par l'interdépendance du mouvement des autres individus afin de calculer la taille de groupe par exemple. Ainsi, la recherche de l'angle de braquage idéal est très couteuse en temps de traitement. Puisque cette opération est à recommencer à chaque mouvement, le domaine de recherche est réduit arbitrairement à 20 possibilités d'angle de braquage uniformément distribué dans l'intervalle $[0, 2\pi]$ qui sont analysées de manière exhaustive.

La fonction objectif de ce problème d'optimisation, défini à l'équation 3.14, somme le biais de fréquence d'occurrence du patron de répartition, le biais de l'angle de braquage ainsi que l'erreur de la taille de groupe. L'objectif est de déterminer l'angle de braquage qui minimise l'erreur globale car un biais négatif indique une sous-représentation de la fréquence d'occurrence en cours de simulation comparativement aux réelles fréquences d'occurrence

observées. Ainsi, l'angle de braquage qui génère l'erreur pondérée minimale est conservé pour amorcer le prochain mouvement de l'individu.

$$\text{Minimiser } E(\theta) = w_\psi B_\psi(\theta) + w_u B_u(\theta) + w_g E_g(\theta) \quad (3.14)$$

Le biais normalisé de la fréquence d'occurrence de l'angle de braquage $B_\psi(\theta)$, défini à l'équation 3.15, consiste à effectuer la différence entre la fréquence d'occurrence de l'angle de braquage θ en cours de simulation $\varphi_{\theta,sim}$ et celle observée dans les données $\varphi_{\theta,obs}$. Le résultat est normalisé par la fréquence d'occurrence observée afin de pondérer équitablement l'importance de ce biais par rapport aux deux autres patrons suivants. Logiquement, l'angle de braquage le plus intéressant à sélectionner est celui dont le biais est minimal puisqu'il indique que cet angle est sous-représenté dans la simulation comparativement aux données observées.

$$B_\psi(\theta) = \frac{\varphi_{\theta,sim} - \varphi_{\theta,obs}}{\varphi_{\theta,obs}} = \frac{\frac{n_{\theta,sim}}{n_{sim}} - \varphi_{\theta,obs}}{\varphi_{\theta,obs}} \quad (3.15)$$

Le biais normalisé de répartition de l'espèce $B_u(\theta)$, défini à l'équation 3.16, consiste à comparer la fréquence de présence de l'espèce pour chaque emplacement visité au cours du prochain mouvement. Ainsi, en fonction de la durée du mouvement T et de son l'angle de braquage θ , le biais est calculé en effectuant la moyenne pour toutes les emplacements visités de l'individu de la différence entre le taux d'occupation simulé $u_{p(t),sim}$ à l'emplacement $p(t)$ comparativement aux données observées $u_{p(t),obs}$. Ce biais est normalisé en le divisant par la moyenne du taux d'occupation afin de pondérer équitablement l'importance du résultat. Encore une fois, minimiser le biais signifie sélectionner l'angle de braquage afin de se diriger vers des emplacements avec une présence sous-représentée en cours de simulation comparativement aux cartes de répartition extraites à la section 4.2.

$$B_u(\theta) = \frac{1}{T} \sum_{t=1}^T \frac{u_{p(t),sim} - u_{p(t),obs}}{u_{obs}} = \frac{1}{T} \sum_{t=1}^T \frac{\frac{n_{p(t),sim} - u_{p(t),obs}}{n_{sim}}}{\frac{1}{size_{SIGHAP}}} = \frac{size_{SIGHAP}}{T} \sum_{t=1}^T \left(\frac{n_{p(t),sim} - u_{p(t),obs}}{n_{sim}} \right) \quad (3.16)$$

Finalement, l'erreur absolue normalisée de la taille de groupe E_g , définie à l'équation 3.17, utilise la taille de groupe projetée du mouvement qui est amorcé avec un angle θ et une durée T . Cette taille de groupe projetée est comparée à la taille de groupe moyenne observée en fonction de la marée $\overline{g_{obs,marée}}$. Afin d'accélérer le temps de traitement, un sous-ensemble d'individus A susceptibles d'entrer en contact avec le mammifère étudié est créé. Cet ensemble est constitué de tous les individus de la même espèce qui se trouve à être à une distance inférieure à la somme de deux fois le rayon de socialisation β et deux fois la distance parcourue par le mouvement projeté. Cette fonction pose comme supposition qu'un individu connaît l'emplacement et la direction des autres individus environnants de son espèce. À partir de ce sous-ensemble d'individus, il est possible de calculer la taille de groupe moyenne en fonction de l'angle de braquage avec la fonction binaire $d(a,t)$ qui est égale à 1 lorsque l'individu a est à une distance inférieure à β par rapport à l'individu étudié au temps t , sinon égale à 0. La valeur absolue de l'erreur de la taille de groupe simulée par rapport à celle observée est normalisée par la taille de groupe observée afin de pondérer équitablement cette erreur par rapport aux deux biais précédents. Dans la même logique, l'objectif est de fixer un angle de braquage qui minimise l'erreur absolue normalisée de la taille de groupe $E_g(\theta)$.

$$E_g(\theta) = \frac{\left| \left(\frac{1}{T} \sum_{a \in A} \sum_{t=1}^T d(a,t) \right) - \overline{g_{obs,marée}} \right|}{\overline{g_{obs,marée}}} \quad | \quad d(a,t) = \begin{cases} 1 & \sqrt{(x(t) - a_x(t))^2 + (y(t) - a_y(t))^2} < r \\ 0 & \text{Sinon} \end{cases} \quad (3.17)$$

Enfin, les trois paramètres w_ψ , w_u et w_g pondèrent la contribution de chaque biais et de l'erreur dans le calcul de l'erreur globale. Ces paramètres permettent de raffiner l'importance accordée à chacun des patrons pour terminer l'implémentation de l'algorithme.

```

SI DUREE <= 0 ALORS
  POUR  $\theta = -180 : 18 : 162$ 
     $\theta = \theta + 18\xi$ 
    Resultat =  $w_\psi B_\psi(\theta) + w_u B_u(\theta) + w_g E_g(\theta)$ 
    SI Resultat < Meilleur_Resultat
      Meilleur_Theta =  $\theta$ 
      Meilleur_Resultat = Resultat
    FIN SI
  FIN POUR

  ANGLE = Meilleur_Theta
  VITESSE =  $CENTILE_{Vitesse}(\xi)$ 
  DUREE =  $CENTILE_{Duree}(\xi)$ 
SINON
  DUREE = DUREE - 1
FIN SI

```

Avantages et limitations

L'avantage principal de l'algorithme de minimisation de la moyenne des biais normalisés est de contrôler facilement les mouvements dans le but de respecter les patrons. La formulation du problème sous la forme d'une erreur pondérée permet de déterminer le mouvement idéal malgré les situations où des patrons sont en contradictions les uns envers les autres. De plus, cette approche est très flexible car elle permet d'ajouter facilement des patrons supplémentaires en ajoutant simplement des termes dans l'équation de l'erreur pondérée.

Cependant, cette approche est conceptuellement à l'opposée de l'approche ascendante, *bottom-up*, privilégiée dans ce projet. Ainsi, l'algorithme ne modélise pas les mécanismes sous-jacents pour faire émerger les patrons globaux de répartition et de taille de groupe. À l'opposé, l'algorithme utilise plutôt les patrons globaux pour contraindre les mouvements locaux à reproduire ces patrons.

Au-delà de l'approche conceptuelle, cet algorithme comporte aussi des limitations techniques. Pour limiter le temps de traitement, l'angle de braquage est approximé avec un intervalle de 18 degrés. Malgré cet artifice, cet algorithme risque d'avoir un temps de traitement très élevé puisqu'il projete à chaque itération 20 mouvements en tenant compte du déplacement des individus environnants avant de sélectionner le meilleur mouvement à effectuer.

CHAPITRE 4

EXTRACTION ET ANALYSE DES PATRONS

Ce chapitre présente l'ensemble des manipulations de données pour extraire et analyser les patrons selon l'approche de modélisation orientée par patrons. Dans un premier temps, les données disponibles pour ce projet sont introduites. Par la suite, ce chapitre présente la démarche pour extraire les patrons observés et les patrons simulés. Finalement, la dernière section propose trois métriques pour décrire la similarité entre deux patrons.

4.1 Données de l'étude

Quatre sources d'information sont utilisées pour construire ce modèle. Chaque source d'information est introduite brièvement ainsi que sa provenance. L'annexe i contient les cartes illustrant les différentes données géographiques disponibles.

Carte de bathymétrie

Une carte de bathymétrie couvrant l'aire de l'étude est disponible via le Service hydrographique du Canada (SHC). Cette carte est enregistrée avec une structure matricielle et une résolution de 100 mètres par 100 mètres.

Système d'information pour la gestion de l'habitat du poisson (SIGHAP)

Le système d'information pour la gestion de l'habitat du poisson, géré par le Ministère des Pêches et Océans Canada, contient de l'information sur la présence des mammifères marins dans l'estuaire du Saguenay-Saint-Laurent (Ministères des Pêches et des Océans, 2004). Ces données, illustrées dans l'annexe i à la figure I-1, permettent de créer une carte de présence ou d'absence pour le béluga et le rorqual commun.

Suivis d'individus

La base de données géographiques contient 80 suivis de mammifères, illustrés dans l'annexe i à la figure I-2, qui ont été effectués à l'aide de balises télémétriques de type VHF. Ces 25

suivis de rorquals communs (Michaud et Girard, 1997a) et 55 suivis de bélugas (Michaud) totalisent plus de 3656 observations. Les données intégrées dans la base de données géographiques sont le numéro du suivi, l'heure et la position géographique de chaque observation.

Activités d'observation en mer (AOM)

L'étude des activités d'observation en mer a été initiée en 1994 et ses objectifs principaux sont de caractériser les AOM et leur utilisation du territoire (Michaud et al., 2008). Cette étude échantillonne les excursions en mer offertes sur une période de 15 semaines entre la mi-juin et la fin septembre dans une aire d'étude incluant le Parc Marin Saguenay-Saint-Laurent (PMSSL) et la zone de protection marine (ZPM) Estuaire du Saint-Laurent proposée.

Parmi les quatre protocoles d'échantillonnage utilisés par les AOM, les blocs d'observation instantanés (BOI) fournissent de l'information supplémentaire sur la présence et la répartition des mammifères. Ces BOI sont une série d'informations relatives aux activités en mer qui sont notées à un intervalle de 10 minutes. Les données consignées par les BOI sont l'heure, l'activité, la position et la vitesse du bateau ainsi que les conditions d'observation. En plus de ces données relatives à l'observateur, le nombre de bateaux et de mammifères marins dans un rayon de 2000 mètres est disponible pour ce projet ainsi que l'espèce ciblée lorsque le bateau est en activité d'observation. La base de données géographiques contient plus de 27658 blocs d'observation instantanés pour un total de 31976 mammifères observés. Ces données sont illustrées à la figure I-3 de l'annexe i.

4.2 Extraction des patrons observés

À partir de ces données, il y a suffisamment d'informations pour extraire les patrons de déplacement des mammifères. Ces patrons sont de trois natures différentes : les patrons de trajectoire, le patron de taille de groupe et le patron de répartition spatiale. Bien que le processus d'extraction soit différent pour chacune de ces catégories de patron, ils partagent certains points en commun. La première étape est de sélectionner la bonne source

d'information. Par la suite, une étape de prétraitement des données permet d'éliminer les informations superflues. Ensuite, les caractéristiques du patron sont extraites pour être finalement compilées dans un format permettant de comparer facilement sa ressemblance à un patron de même nature.

4.2.1 Patrons de trajectoire

Les patrons de trajectoire tentent de définir numériquement le type et l'enchaînement des mouvements effectués par les mammifères. Quatre caractéristiques ont été retenues pour qualifier les trajectoires : la vitesse, la durée, la distance et l'angle de braquage. Les données provenant des suivis d'individus avec une balise télémétrique de type VHF contiennent exactement ce type d'informations. En reliant les observations pour chaque suivi de manière chronologique, il est possible de reconstruire les trajectoires en deux dimensions et d'en extraire ses patrons

Simplification des trajectoires

Idéalement, les données provenant de suivis d'individus sont des traces continues. Dans le but d'extraire certaines caractéristiques, ces trajectoires doivent être discrétisées en une série de mouvements généralement linéaires (Turchin, 1998). Cependant, les données disponibles provenant des suivis d'individus sont implicitement discrétisées à chaque intervalle respiratoire dû au protocole d'échantillonnage. Pour rester cohérent avec la démarche d'analyse des patrons proposée par Turchin, ces suivis sont considérés comme des tracés continus et seront discrétisés de manière explicite dans cette section.

La discrétisation d'une trajectoire continue peut s'effectuer selon deux techniques différentes en fonction des objectifs de l'analyse. La première technique échantillonne la trajectoire avec une durée ou une longueur fixe. Cette méthode est appropriée dans le cas où c'est la magnitude des angles de braquage qui est étudiée. Cependant, un échantillonnage fixe conserve un certain degré d'autocorrélation dans l'angle de braquage et la longueur des

déplacements qui est nuisible à l'extraction des caractéristiques de la marche aléatoire corrélée. Ainsi, cette technique ne sera pas utilisée dans le cadre de ce projet.

La seconde technique, illustrée à la figure 4.1, échantillonne la trajectoire avec une durée et une longueur variables. Cette dernière permet d'extraire les caractéristiques de la théorie de diffusion-approximation qui suggère que la variation de la durée et de la longueur des déplacements change considérablement la distribution spatiale de l'espèce. De plus, les caractéristiques de la marche aléatoire, qui sont la durée, la longueur et l'angle de braquage, sont directement compilables à partir de cette technique d'échantillonnage.

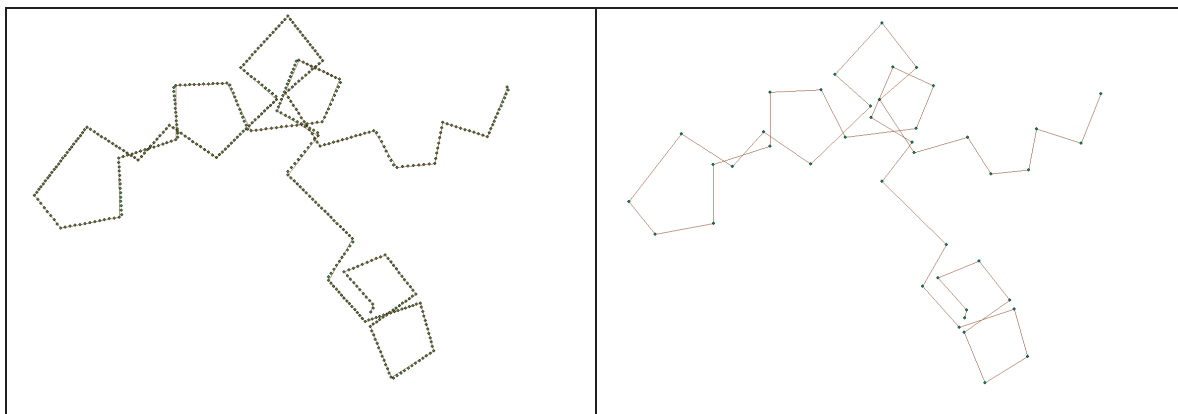


Figure 4.1 Exemple d'une trajectoire à gauche et de cette même trajectoire simplifiée à droite à l'aide de la méthode d'échantillonnage variable.

Afin d'effectuer cet échantillonnage variable, Turchin propose une approche éprouvée et généralement implantée par défaut dans les systèmes d'information géographique. Ainsi, n positions peuvent être agrégées en un mouvement si les $n-1$ positions intermédiaires ne sont pas à plus de x mètres de la ligne connectant la première et dernière position. La figure 4.2 illustre le calcul de la distance du point intermédiaire P_1 par rapport aux points initial et final P_0 et P_2 .

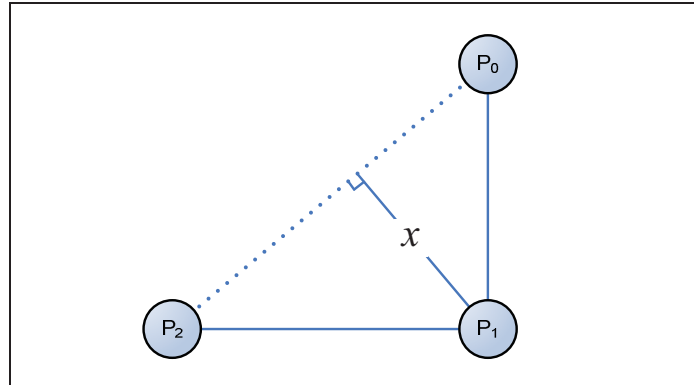


Figure 4.2 Illustration de la distance maximale x .

La valeur de la distance x est déterminée de manière empirique, en augmentant la valeur jusqu'à ce que les phénomènes de sur-échantillonnage disparaissent. De manière concrète, le sur-échantillonnage des trajectoires se détecte par la présence d'autocorrélation positive de l'angle de braquage si un virage est échantillonné plusieurs fois ou une autocorrélation négative si la trajectoire oscille lorsque la résolution est trop fine.

Le coefficient de corrélation de Pearson, défini à l'équation 4.1, est utilisé afin d'estimer l'autocorrélation d'une caractéristique c de mouvements pour deux déplacements consécutifs.

$$r_{c_t, c_{t+1}} = \frac{n \sum c_t c_{t+1} - \sum c_t \sum c_{t+1}}{\sqrt{n \sum c_t^2 - (\sum c_t)^2} \sqrt{n \sum c_{t+1}^2 - (\sum c_{t+1})^2}} \quad (4.1)$$

Pour valider la signifiante du résultat, le coefficient d'autocorrélation est testé à l'aide du test t , où la distribution t possède $n - 2$ degrés de liberté. Le test statistique est bidirectionnel car l'autocorrélation entre les mouvements peut être positive ou négative. Le tableau suivant montre l'autocorrélation de l'angle de braquage r_θ en fonction de la distance seuil x de simplification de la trajectoire.

Tableau 4.1 Autocorrélation de l'angle de braquage r_θ en fonction de la distance seuil x

x	Rorqual commun			Béluga		
	n	ψ	r_θ	n	ψ	r_θ
10	1501	0.35	-0.04	1672	0.53	-0.10*
20	1392	0.32	-0.04	1428	0.46	-0.11*
40	1209	0.25	-0.03	1268	0.41	-0.11*
60	1097	0.20	-0.03	1143	0.37	-0.10*
80	992	0.17	-0.03	1028	0.33	-0.14*

* = Coefficient d'autocorrélation significatif avec $\alpha = 0.05$

À partir de ces résultats, la discrétisation des trajectoires avec une distance seuil de 10 mètres est suffisante, puisque le coefficient d'autocorrélation r_θ reste relativement constant en augmentant le seuil jusqu'à 80 mètres. Cependant, augmenter cette valeur de seuil approche le cosinus moyen de l'angle de braquage ψ vers zéro, indiquant qu'on tend vers une trajectoire sursimplifiée de type marche aléatoire comparativement à une marche aléatoire corrélée avec un seuil fixé à 10 mètres. Une fois la trajectoire simplifiée avec une distance seuil de 10 mètres, il reste 1501 observations de rorquals communs et 1672 observations de bélugas provenant des suivis d'individus par VHF.

Extraction des patrons des trajectoires

À partir des trajectoires simplifiées, la prochaine étape consiste à extraire les caractéristiques des mouvements. Pour éviter d'analyser des mouvements avec certaines caractéristiques manquantes, les deux premières positions de chaque trajectoire sont éliminées. Par la suite, quatre caractéristiques sont extraites pour chaque observation en utilisant sa position courante et ses deux positions précédentes. La figure 4.3 et les équations 4.2 à 4.5 illustrent les opérations trigonométriques à effectuer pour calculer la durée τ , la longueur m , la

vitesse v et le cosinus de l'angle de braquage ψ associés à la position P_2 , à partir des trois positions P_0 , P_1 et P_2 aux temps $T(P_0)$, $T(P_1)$ et $T(P_2)$.

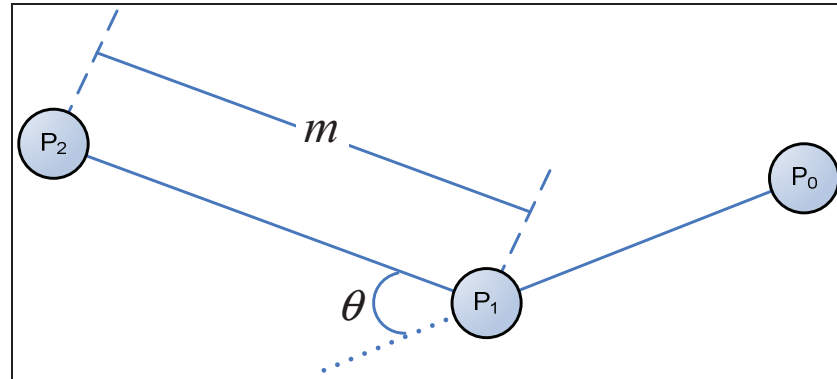


Figure 4.3 Exemple d'une trajectoire avec trois positions.

$$\tau_2 = T(P_2) - T(P_1) \quad (4.2)$$

$$m_2 = \sqrt{(y_{p2} - y_{p1})^2 + (x_{p2} - x_{p1})^2} \quad (4.3)$$

$$v_2 = \frac{m_2}{\tau_2} \quad (4.4)$$

$$\theta_2 = \pi - \arccos \left(\frac{\left((y_{p2} - y_{p0})^2 + (x_{p2} - x_{p0})^2 \right) - \left((y_{p1} - y_{p0})^2 + (x_{p1} - x_{p0})^2 \right) - m_2^2}{-2 \cdot \sqrt{(y_{p1} - y_{p0})^2 + (x_{p1} - x_{p0})^2} \cdot m_2} \right) \quad (4.5)$$

$$\psi_2 = \cos(\theta_2)$$

Les quatre caractéristiques sont compilées au tableau 4.2 en calculant leur moyenne et leur intervalle de confiance pour obtenir les patrons de trajectoire. L'intervalle de confiance de la moyenne est calculé à l'aide de la technique de *bootstrap* avec correction du biais accéléré (BCa) avec 10000 itérations, permettant de ne pas assumer la distribution normale des données (Efron et Tibshirani, 1986).

Tableau 4.2 Patrons de trajectoire extraits des données observées

Patron	Unité	Rorqual commun		Béluga	
		\bar{x}_{obs}	IC	\bar{x}_{obs}	IC
τ	(h)	0.1396	[0.1341 ; 0.1487]	0.1454	[0.1249 ; 0.1957]
m	(m)	577.0858	[550.9614 ; 606.4553]	658.6246	[616.2113 ; 721.2966]
v	(m/s)	1.3262	[1.2718 ; 1.3806]	1.6419	[1.5918 ; 1.6919]
ψ	-	0.3520	[0.3158 ; 0.3865]	0.5278	[0.4969 ; 0.5570]

Estimation des fonctions de répartition des patrons de trajectoire

Compiler les patrons de trajectoire sous la forme de moyenne et d'écart-type n'est pas suffisant pour le modèle. L'ensemble des algorithmes de déplacement décrit dans le chapitre III utilise les fonctions de répartition des patrons afin de piger les caractéristiques du prochain mouvement. Pour approximer ces fonctions de répartition, l'approche paramétrique était privilégiée pour sa simplicité d'implémentation dans le modèle. Cependant, l'approche paramétrique s'est avérée inutilisable avec ces distributions des données non conventionnelles et l'approche non paramétrique est donc utilisée.

L'approche paramétrique consiste à trouver la bonne distribution de probabilités qui permet de modéliser les trois fonctions de répartition extraites. Cinq distributions de probabilités standards ont été testées : normale, gamma, log-normale, weibull et exponentielle. Les paramètres pour chaque distribution de probabilités sont estimés à l'aide du critère de maximum de vraisemblance implémenté dans Matlab. Par la suite, un test de qualité d'ajustement du Chi-carré, avec un intervalle de confiance fixé à 95%, valide la similarité entre la distribution théorique et les données observées. Le tableau 4.3 regroupe les résultats des 30 tests de qualité d'ajustement du Chi-carré. Malheureusement, très peu de distributions de probabilités reproduisent, avec un intervalle de confiance fixé à $\alpha_{0.05}$, les fonctions de répartition caractérisant les trajectoires des mammifères.

Tableau 4.3 Tests de qualité d'ajustement des fonctions de répartition des patrons de trajectoire avec une approche paramétrique

	Rorqual commun			Béluga		
	$T(x)$	$V(x)$	$\Psi(x)$	$T(x)$	$V(x)$	$\Psi(x)$
Normale	-	-	-	-	-	-
Gamma	*	-	-	-	-	-
Lognormal	*	-	-	-	-	-
Weibull	-	-	-	-	-	-
Exponentielle	-	-	-	-	-	-

(*) = Test de qualité d'ajustement du Chi-carré significatif à $\alpha_{0.05}$

(-) = Test de qualité d'ajustement du Chi-carré non significatif à $\alpha_{0.05}$

Puisque l'approche paramétrique n'est pas satisfaisante, les fonctions de répartition utilisées dans le modèle seront représentées à l'aide d'une approche non paramétrique. Les fonctions de répartition empiriques sont calculées directement à partir des caractéristiques extraites des suivis VHF. Par la suite, chaque fonction empirique est discrétisée en 100 cases équidistantes entre 0 et 1 sur l'axe des ordonnées pour créer l'équivalent d'une fonction de centile telle qu'illustrée à la figure 4.4.

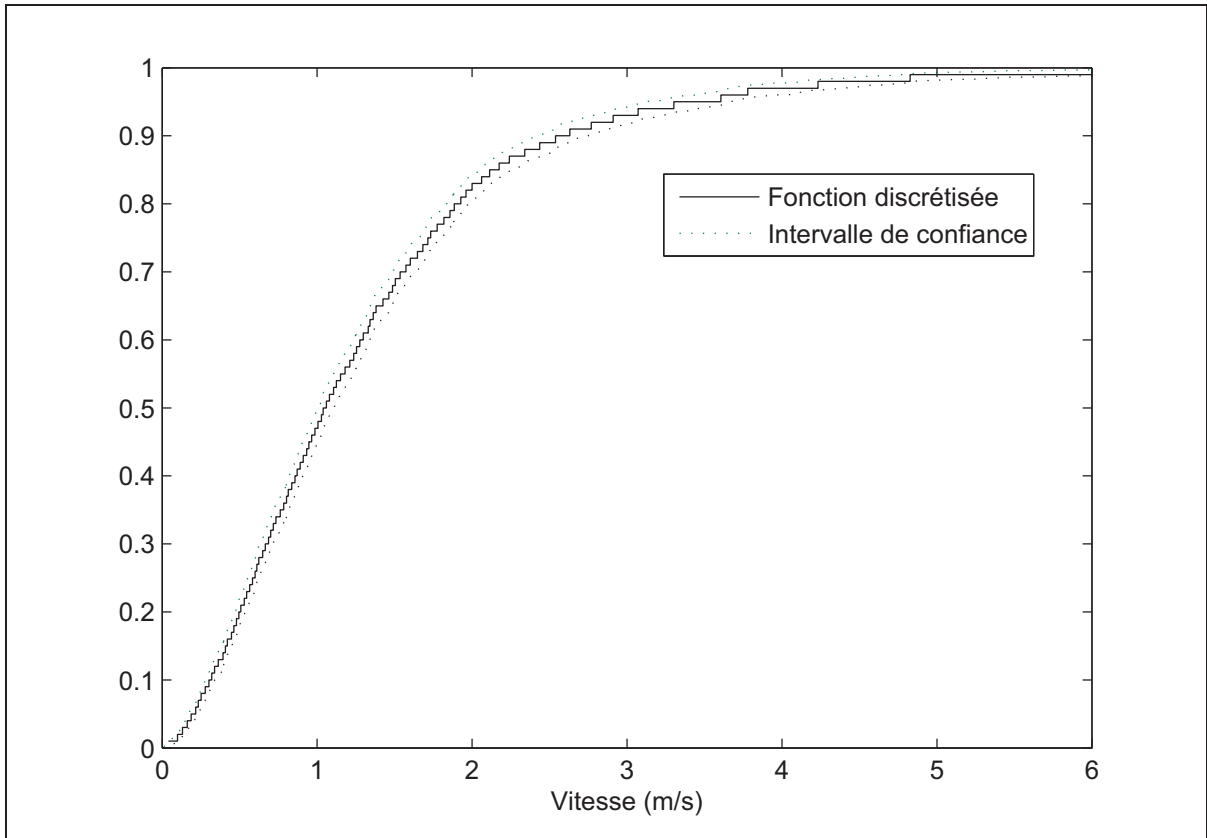


Figure 4.4 Fonction de répartition discrétisée de la vitesse du rorqual commun avec un pas de 0.01 et son intervalle de confiance de $\alpha_{0.05}$.

La qualité d'ajustement de ces fonctions empiriques discrétisées par rapport aux fonctions de répartition extraites est testée en utilisant le test de Kolmogorov-Smirnov avec un intervalle de confiance fixé à 95%. Le tableau 4.4 regroupe les résultats des six tests d'ajustement. La discrétisation de la fonction en 100 cases s'est avérée efficace puisque les six tests statistiques indiquent que les deux courbes sont similaires à un intervalle de confiance de 95%.

Tableau 4.4 Tests de qualité d'ajustement des fonctions de répartition des patrons de trajectoire avec une approche non paramétrique

	Rorqual commun			Béluga		
	$T(x)$	$V(x)$	$\Psi(x)$	$T(x)$	$V(x)$	$\Psi(x)$
Centile	*	*	*	*	*	*

* = Test d'ajustement avec Kolmogorov-Smirnov significatif à $\alpha_{0.05}$

4.2.2 Patron de répartition

La deuxième catégorie de patron à extraire est le patron de répartition spatiale des espèces dans l'aire d'étude. Les données enregistrées sous la forme de BOI, blocs d'observation instantanés, sont une source intéressante d'informations afin de dresser cette carte de distribution. Cependant, ces données sont biaisées par plusieurs facteurs spatio-temporels intrinsèques à ce type d'échantillonnage. Ainsi, certaines stratégies de prétraitement des données AOM sont appliquées sur les données brutes. Par la suite, la carte de répartition est créée par une estimation par noyau et reformatée sous une forme permettant de la comparer facilement à d'autres cartes similaires.

Les stratégies de prétraitement

Le protocole d'échantillonnage des données AOM n'est pas conçu pour dresser une carte de répartition des espèces. Ces données contiennent de nombreux biais comme l'effort d'échantillonnage ou les préférences des espèces par les excursionnistes. Les stratégies de prétraitement ont pour but de diminuer ces biais mais n'ont pas la prétention de créer une carte avec une absence totale de biais d'échantillonnage.

Un premier biais d'échantillonnage dans les données AOM provient des différents types d'observation : les observations ciblées et les observations opportunistes. Les observations sont dites ciblées lorsqu'il y a une intention de la part du capitaine d'observer une espèce en particulier. Dans le cas contraire, l'observation est dite opportuniste car les passagers

peuvent profiter d'observations initialement non prévues par le capitaine. Puisque la trajectoire du bateau est influencée par l'espèce ciblée, il est raisonnable de croire que l'emplacement des observations ciblées par un bateau soit représentatif de la répartition des espèces ciblées. Cependant, ce raisonnement n'est pas valide avec les observations opportunistes. Dans ce cas, l'emplacement des observations opportunistes est biaisé par une surreprésentation des zones susceptibles de trouver des espèces cibles. L'élimination de toutes les observations opportunistes afin de conserver strictement les observations ciblées est une stratégie efficace afin d'éliminer le biais d'effort d'échantillonnage non uniforme des différentes espèces de mammifères marins.

Un second biais d'échantillonnage peut être introduit en cours d'excursion lorsqu'un capitaine observe un individu sur une longue période. Dans ce cas, le protocole d'échantillonnage enregistre toutes les dix minutes la présence du même individu. Ces données redondantes peuvent introduire un biais dans la densité de répartition des espèces. Un moyen simple et efficace d'éliminer cette source de biais est de conserver uniquement l'observation médiane par excursion.

Ces deux stratégies d'extraction des données AOM sont combinables. Ainsi, il y a quatre filtres potentiels, identifiés dans le tableau 4.5, afin d'éliminer ces deux sources de biais dans les données. Idéalement, ces deux filtres seraient utilisés conjointement. Cependant, il faut prendre en considération la perte de données que provoquent ces filtres et le fait que ce ne sont pas toutes les espèces qui sont ciblées par les capitaines.

Tableau 4.5 Stratégies de prétraitement des données AOM

Filtre #	Observation médiane seulement	Espèces ciblées seulement
Filtre 1	Oui	Oui
Filtre 2	Oui	Non
Filtre 3	Non	Oui
Filtre 4	Non	Non

Le premier filtre est le plus restrictif en conservant uniquement une observation par excursion sur l'espèce ciblée. Ce type de filtre est idéal pour les espèces ciblées et observées couramment. Pour les espèces moins populaires mais observées régulièrement, le deuxième filtre est approprié. Il permet d'éliminer le biais d'échantillonnage multiple d'un seul individu tout en conservant un nombre suffisant d'observations en relaxant la contrainte des espèces ciblées. Le troisième filtre s'occupe du cas inverse où une espèce est très populaire mais rarement observée. Dans ce cas, le biais d'échantillonnage non uniforme de l'espace est éliminé en conservant strictement les observations ciblées mais la contrainte d'observation médiane est relaxée afin de conserver un maximum d'observations. Finalement, le quatrième filtre relaxe les deux contraintes afin de conserver tous les points en acceptant en contrepartie la présence des deux biais d'échantillonnage. Ce filtre est utile en dernier recours lorsque l'espèce n'est ni ciblée ni observée régulièrement. L'ensemble des cartes de répartition pour chaque espèce et chaque filtre est présenté à l'annexe ii aux tableaux II-1 et II-2, alors que le tableau 4.6 résume le nombre d'observations et le pourcentage d'observations conservées en fonction du filtre utilisé.

Tableau 4.6 Nombre d'observations AOM conservées en fonction du filtre

Filtre #	Rorqual commun		Béluga	
	n	%	n	%
Filtre 1	748	12.38	122	4.61
Filtre 2	853	14.12	250	9.46
Filtre 3	4686	77.54	613	23.18
Filtre 4	6043	100.00	2644	100.00

Le choix du filtre dépend de la popularité de l'espèce et de son abondance. Dans le cas du rorqual commun, cette espèce est généralement ciblée par les capitaines lorsqu'il n'y a pas de rorqual à bosse à proximité. De plus, le nombre considérable d'observations de cette espèce permet d'utiliser uniquement l'observation médiane de chaque excursion. Ainsi, les deux filtres (#1) sont sélectionnés afin d'éliminer les deux biais d'échantillonnage identifiés précédemment.

Cependant, dans le cas du béluga, cette espèce est très peu ciblée lors des excursions par le manque d'intérêt des capitaines et par la réglementation du parc marin. De plus, le nombre insuffisant d'observations empêche d'utiliser strictement les observations médianes de chaque excursion. Ainsi, aucun filtre (#4) n'est sélectionné afin d'utiliser au maximum toutes les observations opportunistes permettant de dresser une carte de répartition adéquate de l'espèce.

Malgré l'application de ces deux filtres, un biais important reste présent dans les données AOM. L'effort d'échantillonnage des mammifères marins est biaisé puisqu'il y a beaucoup plus d'excursions échantillonnées provenant du port de Tadoussac que du port des Escoumins par exemple. Ce biais cause une surestimation de la répartition des espèces à proximité des ports échantillonnés fréquemment, principalement près de l'embouchure du Saguenay-Saint-Laurent. Pour uniformiser cet effort d'échantillonnage, le nombre corrigé

d'individus $n'_{e,p,l}$ est pondéré selon l'effort de chaque port comme le démontre l'équation 4.6. Le nombre $n_{e,p,l}$ représente le nombre initial d'individus observés de l'espèce e , à la localisation l , par une excursion provenant du port p , alors que la variable d_p représente le nombre de départs d'excursion à partir du port p .

$$n'_{e,p,l} = n_{e,p,l} \frac{1}{d_p} \quad (4.6)$$

Processus de création de la carte de répartition spatiale

La carte de répartition est construite à partir des données filtrées et pondérées $n'_{e,p,l}$ provenant des AOM. Ces données sont converties sous la forme d'une carte de répartition par trois opérations géographiques.

1. La première opération consiste à effectuer une estimation par noyau (kernel density) pour estimer la densité des observations $n'_{e,p,l}$. Le rayon de recherche, fixé à 4000 mètres, est très large dans le but de créer une carte de répartition très lisse. Cette opération retourne une matrice de 1983x2205 cellules avec une résolution de 100 mètres qui couvre l'ensemble de la zone de l'étude.
2. L'application d'une estimation par noyau étend la présence de mammifères au-delà des zones navigables. La seconde opération consiste à supprimer toutes les observations de mammifères à l'extérieur des zones navigables du Saguenay-Saint-Laurent.
3. Enfin, la dernière opération consiste à normaliser la carte de répartition en multipliant chaque élément de la matrice par la constante κ calculée à l'équation 4.7. L'objectif de cette opération est d'obtenir une carte de répartition dont la sommation de tous les éléments soit égale à un.

$$\kappa = \frac{1}{\sum_x \sum_y \text{Repartition}(x, y)} \quad (4.7)$$

Enfin, le patron de répartition spatiale, pour chaque espèce, est enregistré sous la forme d'une matrice standardisée. Avec une dimension de 1983 par 2205 cellules, une résolution de 100 mètres et la sommation des éléments égale à un, ce format standard de matrice permettra de comparer facilement différents patrons de répartition entre eux. Les cartes de répartition utilisées sont disponibles à l'annexe iii aux figures III-1 et III-2.

4.2.3 Patron de socialisation

Enfin, la troisième catégorie de patron à extraire est le patron de socialisation. Les données utilisées pour calculer la taille des groupes sont les observations AOM entre 1994 et 2007. Ces données, échantillonnées aux 10 minutes, contiennent le nombre de mammifères présents dans un rayon de 2000 mètres autour du bateau d'observation. Le nombre total d'individus à chaque observation donne une taille de groupe à grande échelle alors qu'un second champ précise la taille de chaque sous-groupe. Deux individus ou plus sont considérés comme un sous-groupe s'ils respectent les conditions suivantes : nager dans la même direction, avoir un cycle respiratoire relativement synchronisé et se situer à une distance maximale d'une longueur de corps. Compte tenu de la résolution grossière de 100 mètres du simulateur, le patron de taille des groupes à grande échelle sera utilisé.

Définition de la taille de groupe

La taille de groupe est un patron simple en apparence mais qui nécessite d'être défini en détail afin d'éviter plusieurs confusions récurrentes. La taille moyenne des groupes peut être compilée à partir d'un point de vue externe ou interne au groupe. D'un point de vue externe, l'expérimentateur note la taille de chaque groupe. Par la suite, la taille de groupe moyenne \bar{g}_{ext} , définie à l'équation 4.8, est calculée comme une simple moyenne arithmétique en effectuant une sommation de la taille de chaque groupe n_g divisé par le nombre total de groupes observés G .

$$\bar{g}_{ext} = \frac{\sum_{g=1}^G n_g}{G} \quad (4.8)$$

Cependant, il est aussi possible de calculer la taille d'un groupe d'un point de vue interne à celui-ci. Dans ce cas, l'expérimentateur note la taille de groupe entourant chaque individu. La taille de groupe d'un point de vue interne \bar{g}_{int} , définie à l'équation 4.9, peut être calculée comme la moyenne arithmétique de la taille de groupe n_i pour tous les individus I ou alors sous une forme plus compacte lorsque la taille pour chaque groupe n_g est connue.

$$\bar{g}_{int} = \frac{\sum_{i=1}^I n_i}{I} = \frac{\sum_{g=1}^G n_g^2}{\sum_{g=1}^G n_g} \quad (4.9)$$

Afin de clarifier ces deux formalismes, prenons l'exemple de deux groupes de 1 et de 5 individus. Alors, pour la même situation, la taille de groupe \bar{g}_{ext} est alors de 3 et la taille de groupe \bar{g}_{int} est de 4.33.

$$\bar{g}_{ext} = \frac{1+5}{2} = 3$$

$$\bar{g}_{int} = \frac{(1) + (5+5+5+5+5)}{6} = \frac{1^2 + 5^2}{1+5} = 4.33$$

La ligne directrice de ce chapitre est d'extraire des patrons afin de développer un modèle basé sur individu. Ainsi, l'utilisation du patron de la taille de groupe d'un point de vue interne \bar{g}_{int} correspond très bien à la philosophie du modèle. De plus, le coût de traitement informatique est beaucoup plus faible puisqu'il ne sera pas nécessaire de diviser la population en plusieurs sous-groupes afin de pouvoir compiler ce patron lors de la phase de validation.

L'influence de la marée

Selon l'expérience des biologistes et des capitaines d'excursions (R. Michaud, communication personnelle), la marée modifie de manière significative le patron de la taille de groupe des rorquals communs puisqu'ils se regroupent à marée haute. Pour valider cette règle du pouce, une analyse de la variance (ANOVA) compare la taille de groupe en fonction de la marée.

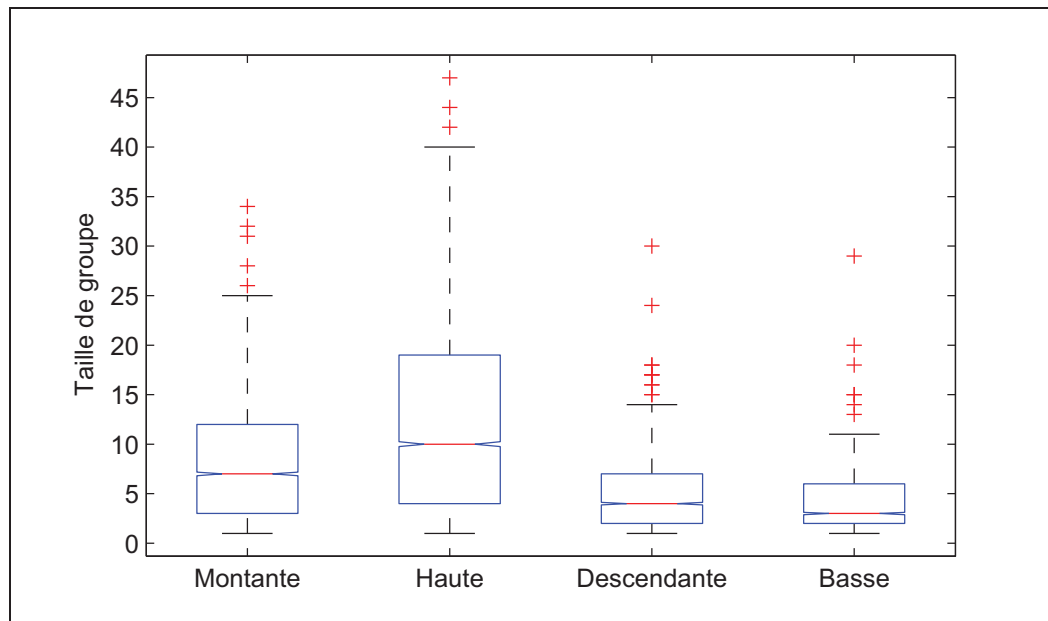


Figure 4.5 Boîte à moustache de la taille de groupe du rorqual commun en fonction des marées.

Cette analyse confirme que la marée modifie de manière significative le patron de taille de groupe des rorquals communs avec un intervalle de confiance fixé à 95%. Une analyse similaire a été effectuée pour valider l'influence de la marée pour le béluga. Pour cette espèce, la marée n'est pas une variable significative qui modifie la taille de groupe. À partir de ces informations, le patron de taille de groupe est compilé au tableau 4.7 séparément pour chaque type de marée dans le cas du rorqual commun et globalement sans tenir compte de la marée pour le béluga.

Tableau 4.7 Patron observé de la taille de groupe en fonction de la marée

Espèce	Marée	n_{ext}	\bar{g}_{ext}	$\sigma_{\bar{g}_{ext}}$	n_{int}	\bar{g}_{int}	$\sigma_{\bar{g}_{int}}$
Rorqual commun	*	6043	3.63	4.49	21941	9.19	8.65
	Montante	1712	3.73	4.30	6390	8.68	7.16
	Haute	1854	4.82	6.14	8939	12.64	10.34
	Descendante	1205	2.97	2.81	3574	5.62	4.83
	Basse	1272	2.39	2.14	3038	4.31	4.13
Béluga	*	2644	2.74	3.76	7248	7.91	14.49

Cette table conclut le processus d'extraction des patrons observés dans les données AOM et des suivis d'individus. À partir de ce point, les patrons de trajectoires sont compilés sous la forme de fonctions de répartition, le patron de distribution spatiale sous la forme d'une matrice normalisée et le patron de taille de groupe sous la forme d'une taille moyenne en fonction de la marée.

4.3 Extraction des patrons simulés

La ligne directrice de la modélisation orientée par patrons est de simuler adéquatement l'ensemble des patrons observés dans le système réel. Ainsi, les patrons de trajectoire, de répartition et de socialisation provenant de la simulation doivent également être compilés en utilisant rigoureusement le même processus d'extraction afin de pouvoir les comparer.

L'implémentation du processus de compilation doit avoir la capacité de gérer des fichiers avec une taille de plusieurs giga-octets, compte tenu de la résolution temporelle très fine du modèle conjuguée à une durée de simulation très longue et un nombre considérable d'individus à l'intérieur de celui-ci. Ainsi, les outils utilisés couramment dans ce projet, ESRI Arcgis et MathWorks Matlab 32-bit, ne peuvent traiter directement le fichier de sortie du simulateur. Pour exploiter la puissance des multiples processeurs, une architecture de

compilation des patrons en parallèle et en cascade a été implémentée en utilisant principalement des commandes de bases de UNIX.

Globalement, le script d'extraction des patrons simulés exécute deux processus distincts afin d'optimiser l'utilisation des ressources. Le premier processus est responsable de compiler les patrons de trajectoire et de socialisation, alors que le second processus compile le patron de répartition. La figure 4.6 illustre l'ensemble du processus d'extraction des patrons simulés. Les lignes pointillées représentent les fichiers virtuels qui ne sont jamais créés grâce à l'utilisation des tubes entre les processus UNIX.

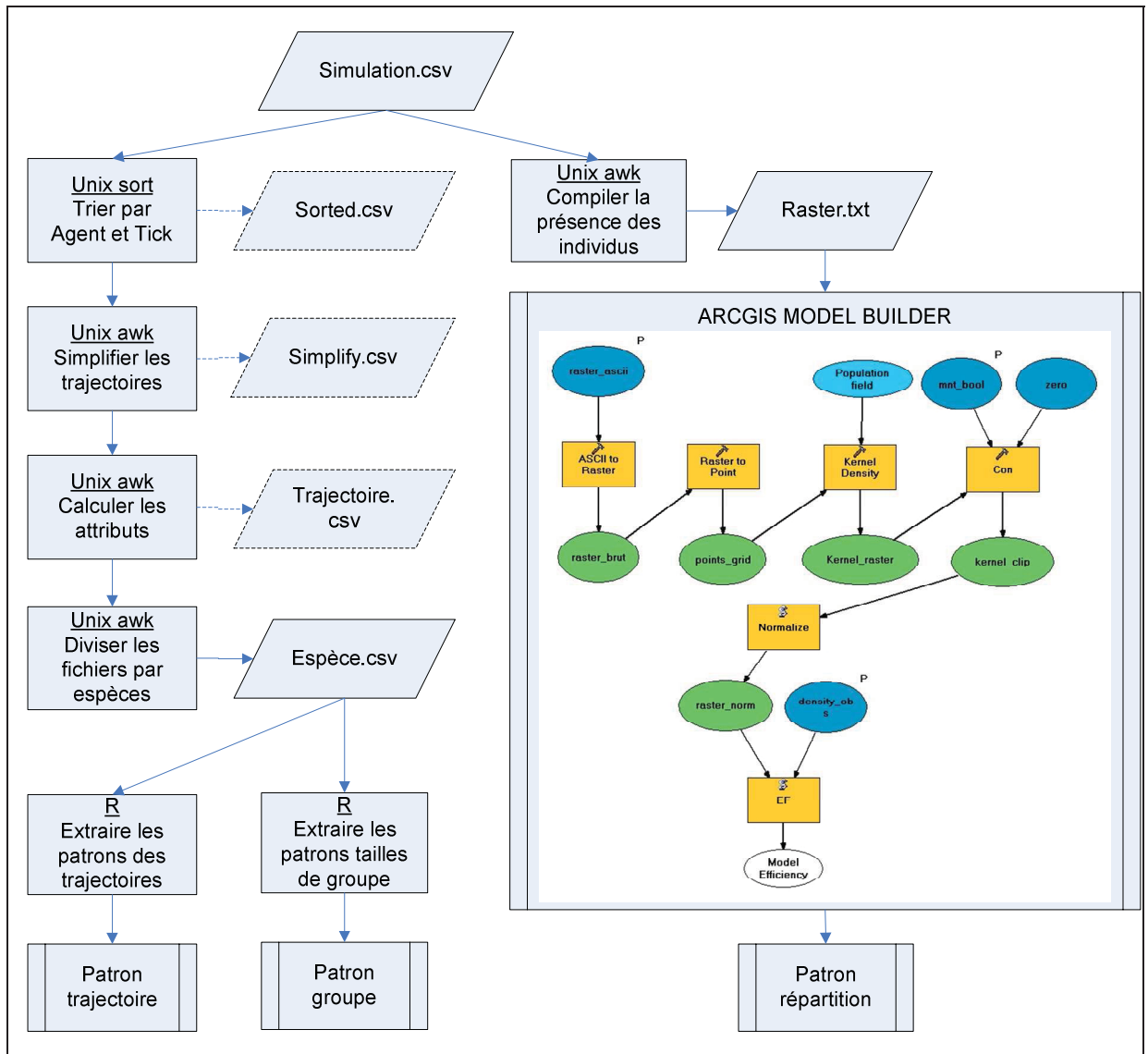


Figure 4.6 Processus d'extraction des patrons simulés.

Le processus d'extraction des patrons de trajectoire et de taille de groupe est situé à la gauche de la figure. Celui-ci exécute une commande de tri suivi d'une série de programmes AWK reliés par des tubes. Ainsi, tous les programmes AWK sont exécutés en parallèle et la vitesse d'exécution est limitée par le programme le plus lent. En premier lieu, le fichier est trié afin de regrouper tous les enregistrements d'un individu par ordre chronologique. Par la suite, les programmes AWK simplifient les trajectoires, calculent les caractéristiques de la trajectoire et séparent les informations pour chaque espèce. Finalement, les statistiques de toutes les

caractéristiques des trajectoires et de la taille de groupe sont compilées avec le logiciel R pour être comparées ultérieurement.

Le second processus, à droite dans la figure 4.6, extrait le patron de répartition en deux phases. Un script AWK compte le nombre de présences des individus pour chaque cellule avec une résolution de 100 mètres et crée une matrice de dénombrement. Par la suite, cette matrice est convertie sous la forme d'une carte matricielle dans Arcgis. Cette carte subit la série de transformations géographiques telle que présentée à la section 4.2.2 pour estimer la répartition d'une espèce. À la suite de ces opérations, cette carte de répartition simulée possède la même dimension et résolution que la carte de répartition observée afin de pouvoir les juxtaposer et calculer leur niveau de similarité dans la prochaine étape de validation des patrons.

Globalement, la complexité lors de l'extraction des patrons simulés est linéaire dans la majorité des opérateurs utilisés. L'exception à cette règle est l'opérateur de triage de type *merge-sort* dont la complexité algorithmique s'exprime par $O(n \cdot \log(n))$.

4.4 Comparaison des patrons observés et simulés

L'approche de modélisation orientée par patrons propose de comparer systématiquement les patrons observés et simulés pour déterminer la précision d'un modèle. Trois mesures de similarité sont utilisées pour quantifier cette ressemblance. Par la suite, un test de Student permet de calculer la variation inter-répliques de ces mesures. Finalement, tous ces résultats sont compilés sous la forme d'un tableau et d'un graphique de type radar, décrit plus loin.

Les mesures de similarité

Trois indicateurs de performance sont utilisés afin de valider la correspondance entre les patrons simulés et observés. Pour comparer deux ensembles de données, l'erreur moyenne absolue normalisée ainsi que la statistique de Kolmogorov-Smirnov sont utilisées.

Cependant, pour comparer la ressemblance entre deux matrices, la mesure d'efficacité du modèle (*model efficiency*) est privilégiée.

L'erreur moyenne absolue normalisée (EMAN) est une statistique qui compare deux groupes de données. Cette statistique calcule l'erreur absolue entre la moyenne des données prédites et observées pour la durée entière de la simulation. Cette erreur est normalisée en divisant par la moyenne des données observées, ce qui permet d'obtenir une métrique sans dimensions.

$$EMAN = \frac{|\bar{x}_{sim} - \bar{x}_{obs}|}{\bar{x}_{obs}} \quad (4.10)$$

La statistique de Kolmogorov-Smirnov (KS), illustrée à la figure 4.7 et définie à l'équation 4.11, est une mesure non paramétrique qui exprime la plus grande différence en valeur absolue entre les distributions de répartition de deux groupes de données. Cette statistique a pour but de s'assurer que les patrons simulés respectent les réelles fonctions de répartition extraites lors de l'analyse des patrons. La statistique de Kolmogorov-Smirnov est complémentaire à l'erreur moyenne absolue normalisée. La valeur de l'EMAN valide la similarité des valeurs centrales des données alors que la valeur de KS valide la similitude de leur dispersion.

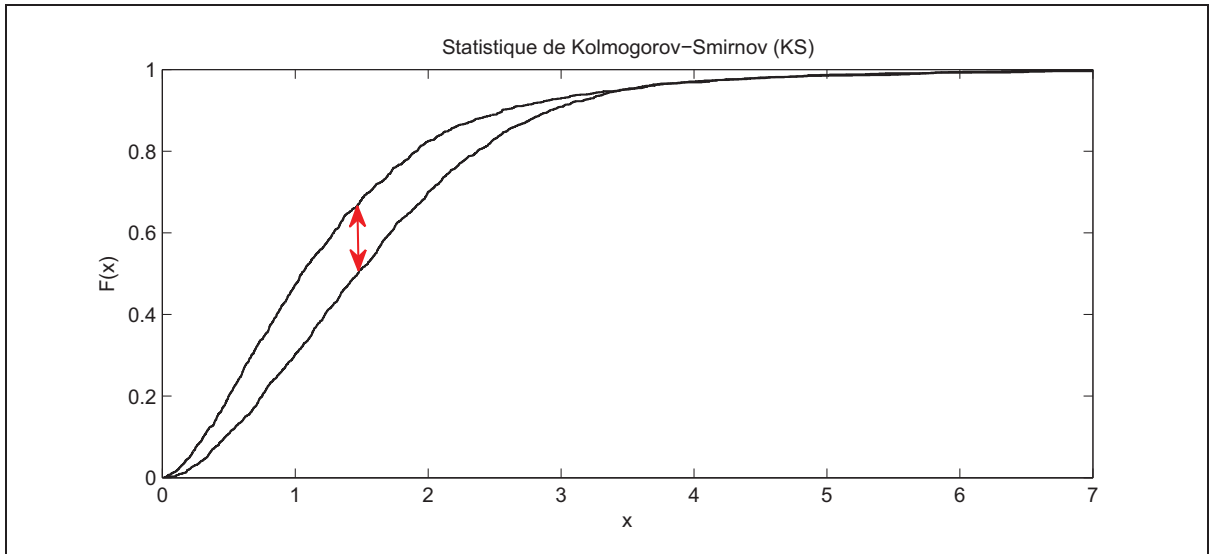


Figure 4.7 Exemple du calcul de la statistique de Kolmogorov-Smirnov. La flèche représente l'endroit où il y a la plus grande différence entre les deux fonctions de répartition.

La valeur retournée par cette fonction est comprise dans l'intervalle $[0;1]$. Une valeur près de zéro indique que les deux fonctions de répartition sont très similaires.

$$KS = \max_x (|F_o(x) - F_s(x)|) \quad (4.11)$$

Finalement, la mesure d'efficacité du modèle (EM) indique le pourcentage de la variance expliquée dans le modèle. Tout comme le coefficient de détermination r^2 , cette statistique calcule la déviation des informations simulées p_i à partir des informations observées o_i en fonction de la dispersion des données observées. L'efficacité du modèle est utilisée spécifiquement pour comparer deux cartes avec une structure matricielle où chaque cellule est une information i . L'efficacité du modèle retourne une valeur comprise dans l'intervalle $[-\infty;1]$, où la valeur 1 signifie que les deux cartes sont rigoureusement identiques.

$$EM = 1 - \frac{\sum_{i=1}^n (p_i - o_i)^2}{\sum_{i=1}^n (o_i - \bar{o})^2} \quad (4.12)$$

Mesure de la variabilité inter-réplication

Lors de la simulation d'un scénario, ces statistiques pour comparer la similarité entre les patrons observés et simulés sont calculées pour chaque réplication. Cependant, il y a une variabilité entre chaque réplication inhérente aux processus stochastiques qui sont simulés. Pour définir clairement les performances d'un modèle et sa variation inter-réplication, les mesures sont exprimées sous la forme d'une moyenne et d'un intervalle de confiance. Le test de Student est utilisé pour permettre de calculer l'intervalle de confiance malgré un faible nombre de réplifications. La figure 4.8 résume le processus de compilation des résultats pour un patron. Les mesures EMAN et KS sont calculées indépendamment pour chaque réplication avant de compiler la moyenne et l'intervalle de confiance de ceux-ci. Cependant, la statistique d'efficacité du modèle EM n'est pas calculée à chaque réplication, mais uniquement pour l'ensemble du scénario, dû à une limitation du temps de traitement avec le logiciel Arcgis.

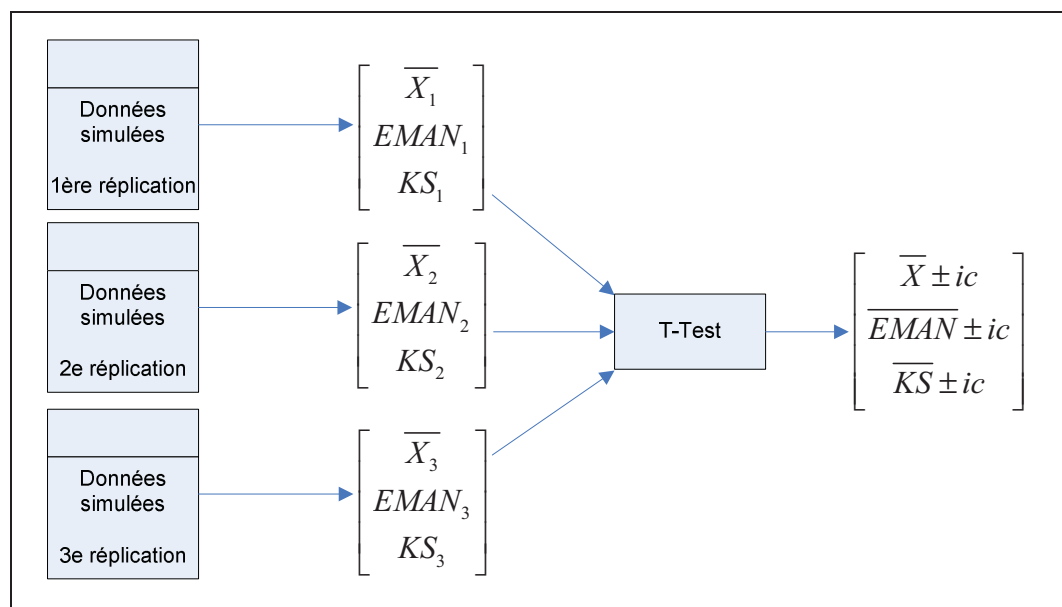


Figure 4.8 Processus de compilation des résultats inter-réplication.

Compilation des résultats

Les mesures de similitude décrites dans cette section ne sont pas applicables à tous les patrons. Les quatre patrons de trajectoire et les cinq patrons de socialisation retournent des vecteurs de données. Ainsi, les statistiques EMAN et KS sont utilisées pour évaluer la similitude de ces patrons entre les deux groupes de données observées et simulées. Dans le cas du patron de répartition avec sa structure matricielle, la mesure d'efficacité du modèle est privilégiée pour évaluer sa similarité. Le tableau 4.8 récapitule l'ensemble de métriques utilisées pour évaluer une simulation et sera utilisé, au chapitre V, pour présenter et analyser les résultats.

Tableau 4.8 Exemple d'un tableau de compilation des résultats pour une expérimentation

<i>Catégorie</i>	<i>Patron</i>	<i>Symbole</i>	$\bar{X} \pm ic$	$\overline{EMAN} \pm ic$	$\overline{KS} \pm ic$	<i>EM</i>
Trajectoire	Distance	m	X	X	X	
	Durée	τ	X	X	X	
	Vitesse	v	X	X	X	
	Angle	ψ	X	X	X	
Social	Taille de groupe*	g	X	X	X	
	- Marée montante	$g_{montante}$	X	X	X	
	- Marée haute	g_{haute}	X	X	X	
	- Marée descendante	$g_{descendante}$	X	X	X	
	- Marée basse	g_{basse}	X	X	X	
Répartition	Répartition spatiale	u				X

Tous les patrons utilisent les équations 4.10 à 4.12 pour calculer les statistiques à l'exception du patron de la taille de groupe g . Ce patron général sert à regrouper les statistiques des quatre patrons spécifiques de la taille de groupe à chaque marée. Ainsi, la statistique g_{EMAN}

est la moyenne arithmétique des quatre statistiques $g_{montante_{EMAN}}$, $g_{haute_{EMAN}}$, $g_{descendante_{EMAN}}$, $g_{basse_{EMAN}}$ et la statistique g_{KS} est calculée avec la même technique.

Puisqu'une image vaut mille mots, un graphique de type radar similaire à la figure 4.9 est utilisé pour illustrer grossièrement l'information contenue dans le tableau récapitulatif. Ce graphique possède six axes et résume la performance des six patrons : distance, durée, vitesse, angle de braquage, taille de groupe et répartition spatiale. Les valeurs utilisées sont la statistique de KS pour résumer les patrons de trajectoire, l'EMAN pour le patron global de la taille de groupe et l'EM pour le patron de répartition spatiale.

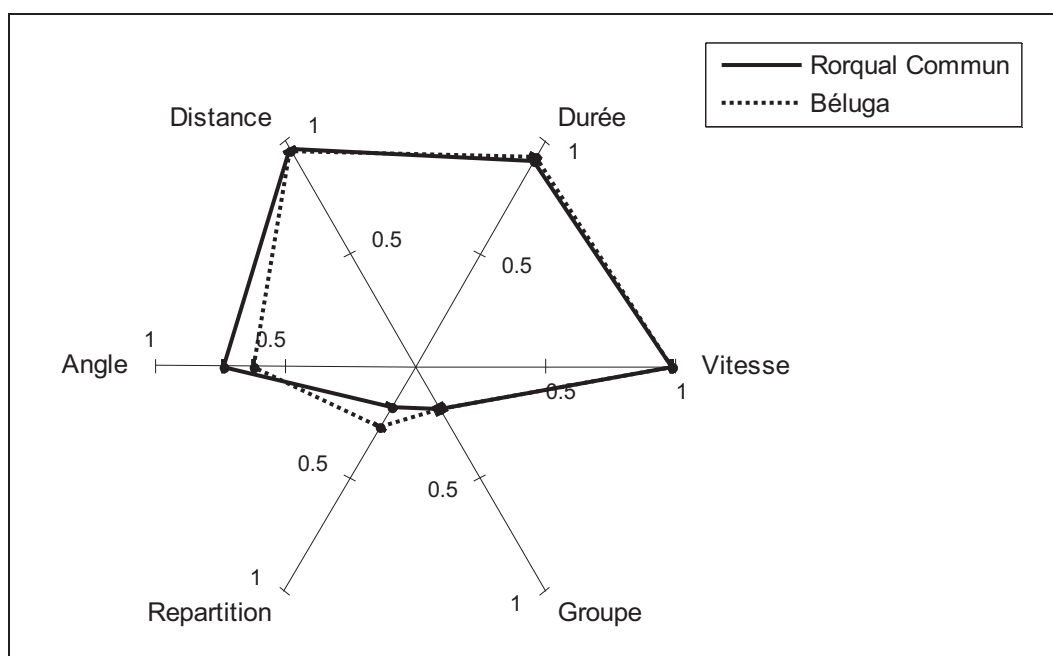


Figure 4.9 Exemple d'un graphique de type radar avec les six patrons.

Afin de standardiser les axes, deux petites manipulations mathématiques sont effectuées. Un opérateur maximum s'assure que la valeur sur chaque axe soit dans l'intervalle $[0,1]$. La seconde opération, la soustraction de la statistique KS et EMAN du nombre 1, inverse l'interprétation pour qu'une valeur près de 1 signifie une très grande similarité entre les

patrons simulés et observés. Les opérations exactes utilisées pour standardiser les six axes sont présentées aux équations 4.13 à 4.18.

$$Angle = 1 - \psi_{KS} \quad (4.13)$$

$$Distance = 1 - m_{KS} \quad (4.14)$$

$$Durée = 1 - \tau_{KS} \quad (4.15)$$

$$Vitesse = 1 - v_{KS} \quad (4.16)$$

$$Groupe = \max(0, 1 - g_{EMAN}) \quad (4.17)$$

$$Repartition = \max(0, u_{EM}) \quad (4.18)$$

Dans ce chapitre, nous avons défini les outils d'extraction, de comparaison et de visualisation des patrons à partir des données observées et simulées. Le prochain chapitre utilise ces outils afin d'évaluer les six algorithmes introduits au chapitre III.

CHAPITRE 5

RÉSULTATS ET DISCUSSION

L'objectif de ce chapitre est de compiler et d'analyser les résultats des expérimentations dans le but de confirmer ou d'infirmer les hypothèses de recherche. Ainsi, six expérimentations sont exécutées et chacune d'entre elles utilise un algorithme de déplacement différent. Par la suite, le processus de validation compare les patrons observés et simulés dans le but de tester les hypothèses de recherche.

Ce chapitre se divise en trois sections. En premier lieu, les résultats de chaque expérimentation sont compilés et analysés de manière individuelle. Par la suite, une brève section analyse le temps de traitement nécessaire pour exécuter les algorithmes. Finalement, la dernière section recommande un algorithme et discute globalement des avantages et des limitations qui ont mené vers ce choix.

5.1 Évaluation des algorithmes de déplacement

Dans cette section, les six algorithmes de déplacement sont évalués de manière indépendante. Une étape préliminaire est généralement nécessaire afin de fixer certains paramètres des algorithmes. Par la suite, l'expérimentation finale est exécutée avec le protocole décrit à la section 2.5 et les résultats sont compilés sous la forme d'un tableau et d'un graphique. Finalement, une discussion présente l'analyse des résultats obtenus.

5.1.1 Marche aléatoire (MA)

Évaluation des paramètres

Le déplacement des individus est explicitement délimité par les berges du Saint-Laurent et du Fjord du Saguenay. Pour gérer cette limitation physique de l'habitat, deux stratégies potentielles ont été analysées afin de gérer les cas d'exception. La première stratégie consiste à accepter les collisions avec les berges. Lors d'une collision, le mouvement en

cours est arrêté et un nouveau mouvement est amorcé. La seconde stratégie étudiée ajoute une notion préventive en tentant d'éviter les collisions avec les berges. Lors de l'initialisation d'un mouvement, un test logique s'assure que la position finale du mouvement reste dans l'habitat du mammifère. Si ce test s'avère faux, la seconde stratégie consiste à recommencer la pige de tous les attributs du mouvement.

Ces deux stratégies ont été implémentées et le protocole expérimental ainsi que les résultats sont disponibles à l'annexe iv aux tableaux IV-1 et IV-2. À partir de ces résultats, il devient évident que la stratégie de gestion des frontières influence la similarité des patrons de durée et de distance des mouvements. La stratégie réactive termine abruptement les mouvements lorsque l'individu sort de son habitat, ce qui cause la présence de mouvement trop court en termes de distance et de temps. À l'opposé, l'approche préventive permet d'éviter ce cas d'exception en complétant la majorité des mouvements amorcés et ainsi offrir de meilleures performances. Ainsi, la gestion des berges avec une approche préventive est implémentée dans l'algorithme final de marche aléatoire.

Compilation des résultats

Tableau 5.1 Résultat de l'expérimentation avec l'algorithme MA pour le rorqual commun

	\bar{X}	$\overline{EMAN} \pm ic$	$\overline{KS} \pm ic$	EM
Vitesse	1.319 ± 0.001	0.003 ± 0.001	0.012 ± 0.000	
Durée	489.758 ± 0.277	0.025 ± 0.001	0.085 ± 0.000	
Distance	644.103 ± 0.975	0.116 ± 0.002	0.040 ± 0.000	
Angle de braquage	-0.021 ± 0.001	1.059 ± 0.002	0.266 ± 0.000	
Groupe	1.268 ± 0.003	0.809 ± 0.015	0.709 ± 0.016	
Répartition observée				0.178
Répartition uniforme				0.903

Tableau 5.2 Résultat de l'expérimentation avec l'algorithme MA pour le béluga

	\bar{X}	$\overline{EMAN} \pm ic$	$\overline{KS} \pm ic$	EM
Vitesse	1.643 ± 0.001	0.001 ± 0.000	0.010 ± 0.000	
Durée	445.977 ± 0.516	0.148 ± 0.001	0.070 ± 0.000	
Distance	720.955 ± 0.930	0.095 ± 0.001	0.044 ± 0.000	
Angle de braquage	-0.023 ± 0.000	1.044 ± 0.001	0.378 ± 0.000	
Groupe	1.484 ± 0.002	0.812 ± 0.000	0.529 ± 0.001	
Répartition observée				0.270
Répartition uniforme				0.858

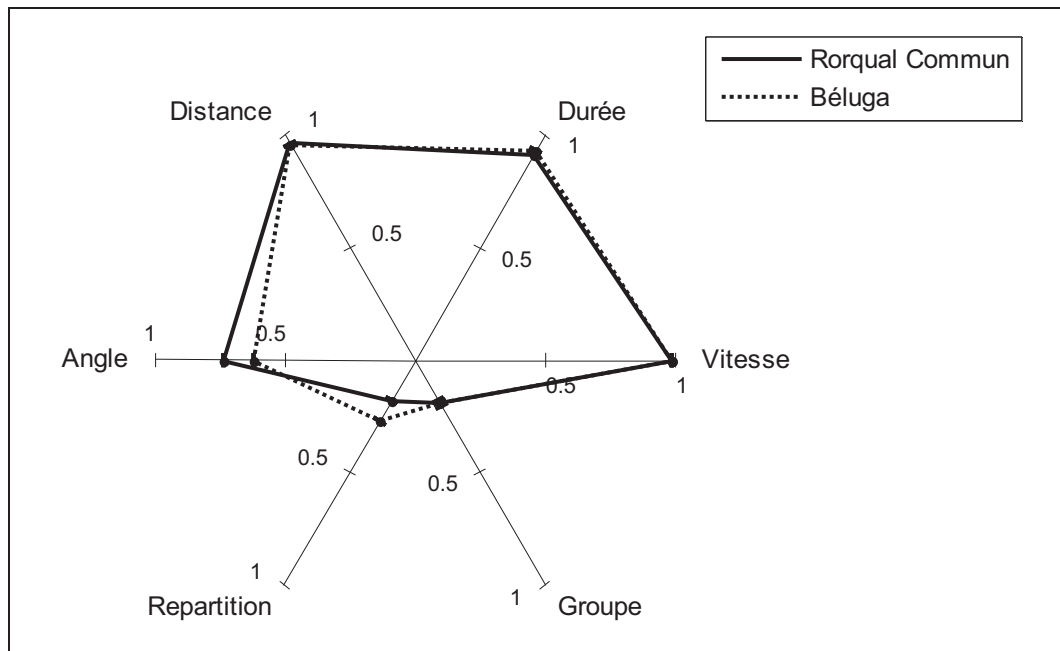


Figure 5.1 Résultat de l'expérimentation avec l'algorithme MA.

Discussion

L'algorithme de marche aléatoire génère certains patrons de trajectoire de manière très prédictible. Les patrons de la vitesse, la durée et la distance des mouvements simulés reproduisent fidèlement les patrons observés avec $m_{KS}, v_{KS}, \tau_{KS} < 0.10$. Ces excellents

résultats sont prévisibles dans la mesure où l'algorithme utilise la méthode de Monte-Carlo avec les fonctions de répartition des données observées τ et ν pour générer le prochain mouvement. L'algorithme génère également le patron de l'angle de braquage théorique de $\psi \cong 0$ typique à la marche aléatoire. Cependant, avec la statistique $\psi_{KS} > 0.25$ pour le rorqual commun et encore plus pour le béluga, il devient évident que l'algorithme de marche aléatoire ne reproduit pas adéquatement le patron de l'angle de braquage pour ces deux espèces.

Le patron de socialisation n'est pas explicitement pris en compte avec cet algorithme. Puisque les individus simulés n'ont pas conscience des autres individus environnants, les groupes sont générés uniquement de manière aléatoire lorsque deux individus se croisent. Dans la réalité, le rorqual commun et le béluga sont deux espèces grégaires. L'erreur importante entre les patrons de socialisation simulés et observés, $EMAN_{groupe} > 0.80$, indique clairement que la marche aléatoire ne peut reproduire ce patron de socialisation pour ces deux espèces.

La diffusion spatiale des individus est typiquement uniforme pour la marche aléatoire simulée sur une longue période de temps. Cependant, ce type de diffusion ne représente pas la répartition des mammifères marins dans le Saguenay. L'efficacité du modèle démontre que l'algorithme diffuse les individus de manière relativement uniforme dans l'environnement $EM_{uniforme} > 0.85$ mais ne reproduit pas la répartition spatiale des mammifères observés $EM_{observé} < 0.30$.

5.1.2 Marche aléatoire corrélée (MAC)

Évaluation des paramètres

Tout comme l'algorithme de marche aléatoire, il est nécessaire de gérer le cas d'exception où le mouvement déterminé par l'algorithme de marche aléatoire corrélée déplace l'agent en dehors de son habitat. Dans ce cas, deux stratégies sont analysées. La première stratégie, réactive, consiste à accepter une collision avec les berges, arrêter le mouvement en cours et

amorcer un nouveau mouvement. À l'opposé, la seconde stratégie préventive valide lorsqu'un mouvement est amorcé qu'il se termine à l'intérieur de son habitat. Si ce test s'avère faux, le mouvement est annulé et un nouveau mouvement est créé en pigeant les trois attributs d'un mouvement : la durée τ , la vitesse v et l'angle de braquage ψ .

Les stratégies réactive et préventive ont été simulées dans le modèle. Le protocole expérimental ainsi que les résultats sont disponibles dans l'annexe iv aux tableaux IV-3 et IV-4. Tout comme l'algorithme de marche aléatoire, la stratégie préventive obtient de meilleurs résultats puisqu'elle n'interrompt pas de mouvements en cours. Pour cette raison, cette stratégie de type préventive est aussi implémentée dans l'algorithme final de la marche aléatoire corrélée.

Compilation des résultats

Tableau 5.3 Résultat de l'expérimentation avec l'algorithme MAC pour le rorqual commun

	\bar{X}	$\overline{EMAN} \pm ic$	$\overline{KS} \pm ic$	EM
Vitesse	1.284 ± 0.001	0.030 ± 0.001	0.021 ± 0.001	
Durée	483.913 ± 0.313	0.037 ± 0.001	0.082 ± 0.000	
Distance	614.552 ± 0.912	0.065 ± 0.002	0.028 ± 0.000	
Angle de braquage	0.327 ± 0.001	0.071 ± 0.002	0.022 ± 0.000	
Groupe	1.247 ± 0.003	0.812 ± 0.015	0.717 ± 0.016	
Répartition observée				0.164
Répartition uniforme				0.921

Tableau 5.4 Résultat de l'expérimentation avec l'algorithme MAC pour le béluga

	\bar{X}	$\overline{EMAN} \pm ic$	$\overline{KS} \pm ic$	EM
Vitesse	1.607 ± 0.001	0.020 ± 0.001	0.021 ± 0.000	
Durée	430.063 ± 0.690	0.178 ± 0.001	0.067 ± 0.000	
Distance	666.275 ± 1.491	0.012 ± 0.002	0.039 ± 0.000	
Angle de braquage	0.478 ± 0.001	0.095 ± 0.001	0.050 ± 0.000	
Groupe	1.433 ± 0.002	0.819 ± 0.000	0.558 ± 0.001	
Répartition observée				0.272
Répartition uniforme				0.912

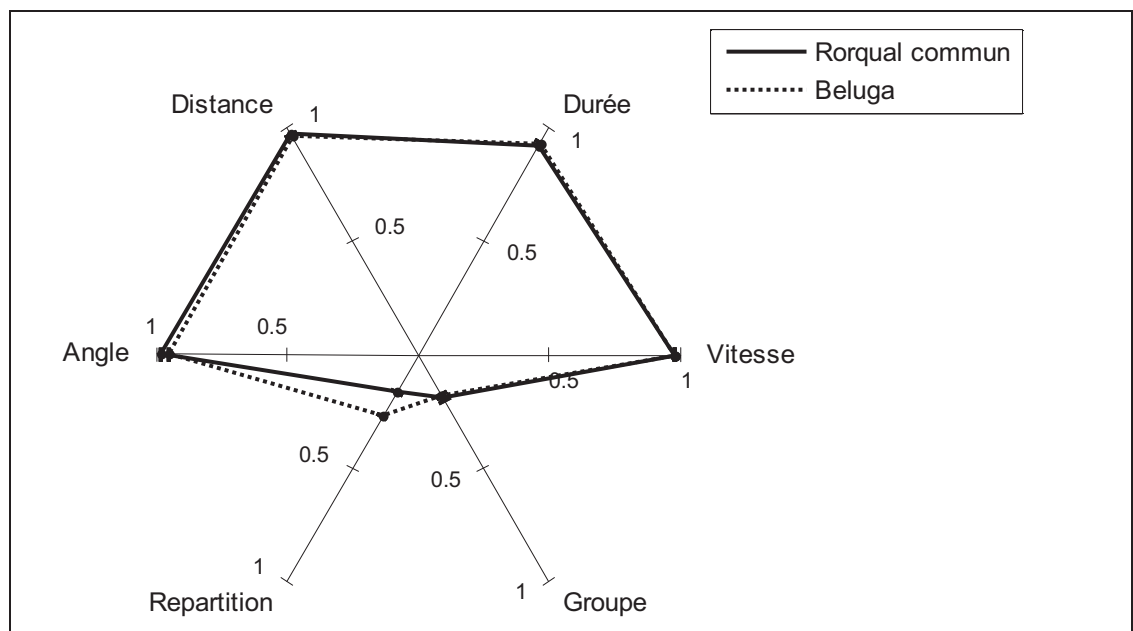


Figure 5.2 Résultat de l'expérimentation avec l'algorithme MAC.

Discussion

L'algorithme de marche aléatoire corrélée génère les patrons de trajectoire de manière aussi prédictible que l'algorithme précédent. Comme son prédécesseur, les patrons de vitesse, durée et distance reproduisent fidèlement les patrons observés avec des résultats $m_{KS}, v_{KS}, \tau_{KS} < 0.10$. Cependant, contrairement à son prédécesseur, la méthode de Monte-

Carlo est aussi utilisée pour générer le prochain angle de braquage. Cette méthode permet de simuler fidèlement l'angle de braquage, $\psi_{KS} < 0.10$, et ainsi reproduire correctement les quatre patrons qui forment la trajectoire des individus.

Cependant, les patrons simulés de socialisation et de répartition ne reproduisent pas les patrons observés pour les mêmes raisons que celles décrites pour la marche aléatoire. Les individus se diffusent uniformément et individuellement dans l'espace, ce qui est complètement à l'opposé des comportements du rorqual commun et du béluga comme le démontre la dissemblance de ces deux patrons $EMAN_{groupe} > 0.80$ et $EM_{observé} < 0.30$.

5.1.3 Marche aléatoire corrélée avec indice de résidence (MAC-R)

Évaluation des paramètres

Lors du calcul de l'indice de résidence maximal à la section 3.3, tous les calculs sont basés sur des pas de temps t_{step} et t_{limit} de 60 secondes qui limitent l'indice de résidence maximal ρ_{limit} selon l'équation 3.6. Cette contrainte permet de ne jamais générer un mouvement d'une durée inférieure à 60 secondes, même pour l'indice de résidence le plus élevé. Cependant, cette contrainte génère des mouvements de plusieurs jours pour l'indice de résidence minimal. Dans un environnement limité comme le Saint-Laurent, ces mouvements très longs sont généralement interrompus dû aux berges. Ces frontières empêchent de simuler librement la diffusion des individus et biaisent potentiellement le patron de répartition prédit par l'indice de résidence.

Pour comprendre l'effet des variables t_{step} et t_{limit} , six expérimentations préliminaires ont été effectuées. Le protocole expérimental ainsi que les résultats sont disponibles à l'annexe iv aux tableaux IV-5 et IV-6. Dans les trois premières expérimentations, le grain temporel de la simulation t_{step} est fixé à 60 secondes, mais la contrainte de durée minimale d'un mouvement t_{limit} est relaxée à 1, 10 et 60 secondes. Cette relaxation de la contrainte t_{limit} permet de diminuer la durée maximale des mouvements. Cependant, puisque le pas de temps de la

simulation est de 60 secondes, tous les mouvements dont la durée théorique est inférieure à 60 secondes sont simulés durant un pas de temps complet. Les trois dernières expérimentations préliminaires consistent à diminuer le grain temporel de simulation t_{step} à 10 secondes et de faire varier à nouveau la contrainte t_{limit} à 1, 10 et 60 secondes. Le fait de diminuer le grain temporel de la simulation permet de diminuer l'erreur d'arrondissement entre la durée théorique, calculée en fonction de l'indice de résidence, et la durée simulée, limitée par le grain temporel de la simulation.

Les résultats de ces expérimentations démontrent que les deux variables t_{step} et t_{limit} modifient considérablement les patrons simulés. Le fait de diminuer la contrainte de la durée de mouvement minimale t_{limit} à 10 secondes ou 1 seconde améliore grandement le patron de répartition. Cependant, c'est en diminuant simultanément t_{step} et t_{limit} à 10 secondes et 1 seconde respectivement qu'on obtient le meilleur patron de répartition avec un résultat $EM > 0.95$ pour le rorqual commun. Cependant, ces excellents résultats s'obtiennent au détriment des patrons de trajectoire. Pour équilibrer l'erreur entre tous les patrons et respecter le grain temporel du modèle, les paramètres utilisés pour l'expérimentation finale sont $t_{step} = 60$ secondes et $t_{limit} = 10$ secondes.

Compilation des résultats

Tableau 5.5 Résultat de l'expérimentation avec l'algorithme MAC-R pour le rorqual commun

	\bar{X}	$\overline{EMAN} \pm ic$	$\overline{KS} \pm ic$	EM
Vitesse	1.376 ± 0.001	0.040 ± 0.000	0.061 ± 0.000	
Durée	143.391 ± 3.638	0.715 ± 0.007	0.912 ± 0.002	
Distance	153.662 ± 2.577	0.734 ± 0.004	0.671 ± 0.002	
Angle de braquage	0.243 ± 0.001	0.309 ± 0.001	0.979 ± 0.000	
Groupe	2.392 ± 0.023	0.639 ± 0.029	0.467 ± 0.031	
Répartition observée				0.746
Répartition uniforme				-0.644

Tableau 5.6 Résultat de l'expérimentation avec l'algorithme MAC-R pour le béluga

	\bar{X}	$\overline{EMAN} \pm ic$	$\overline{KS} \pm ic$	EM
Vitesse	1.669 ± 0.001	0.017 ± 0.001	0.037 ± 0.001	
Durée	682.048 ± 10.862	0.303 ± 0.021	0.412 ± 0.007	
Distance	820.827 ± 11.465	0.246 ± 0.017	0.322 ± 0.007	
Angle de braquage	0.397 ± 0.001	0.249 ± 0.003	0.100 ± 0.001	
Groupe	2.041 ± 0.018	0.742 ± 0.002	0.329 ± 0.004	
Répartition observée				0.513
Répartition uniforme				0.741

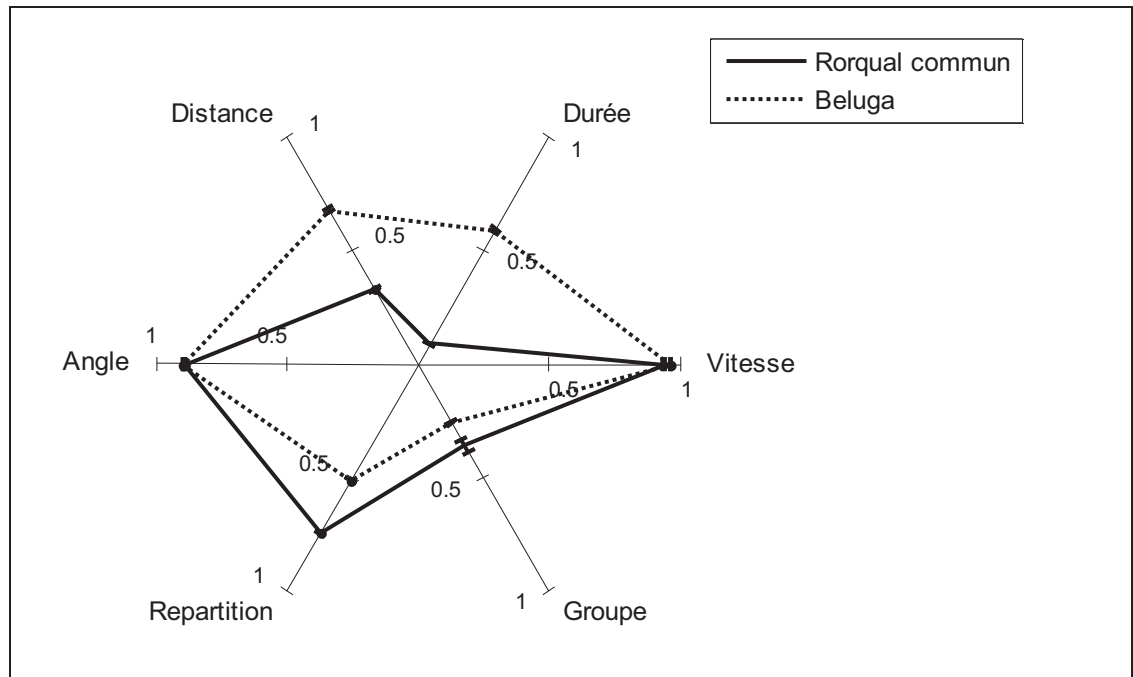


Figure 5.3 Résultat de l'expérimentation avec l'algorithme MAC-R.

Discussion

L'algorithme de marche aléatoire corrélée avec indice de résidence est basé sur l'algorithme de MAC et prend en compte la préférence des habitats. Cependant, certains patrons de trajectoire qui présentaient d'excellents résultats ont régressé alors que le patron de répartition s'est faiblement amélioré.

La méthode de Monte-Carlo est utilisée pour piger la vitesse et l'angle de braquage du prochain mouvement. Ainsi, cette méthode permet de simuler correctement le patron de la vitesse $v_{KS} < 0.10$ et le patron de l'angle de braquage $\psi_{KS} < 0.10$. Cependant, les patrons de la durée et de la distance simulés sont nettement différents des patrons observés avec $\tau_{KS}, m_{KS} > 0.20$. Il faut revenir à la structure de l'algorithme, décrite à la section 3.3, pour comprendre l'origine de l'erreur de ces deux patrons. La durée du déplacement est calculée en fonction de l'indice de résidence selon l'équation 3.3. Auparavant, cet indice de résidence a été calibré en fonction de l'indice de résidence maximal avec la relation 3.4. Cet indice de résidence maximal a été calculé à partir de la fonction objectif décrite à l'équation 3.7.

Cependant, cette équation prend pour acquis que la répartition spatiale des mammifères sera respectée et les mouvements seront ininterrompus lors du calcul de la distance moyenne des déplacements. Ainsi, l'erreur des patrons de durée et de distance des mouvements s'explique en partie par le non-respect du patron de répartition spatiale des mammifères avec une efficacité de modèle $EM_{rorqual} = 0.741$ et $EM_{béluga} = 0.513$. De plus, les mouvements sont à l'occasion interrompus puisque l'algorithme génère des mouvements trop longs selon l'indice de résidence. Ces mouvements trop longs sont inévitablement interrompus lors d'une collision avec une berge.

La répartition spatiale des individus est grandement tributaire de deux paramètres : le grain temporel de la simulation et la contrainte de la durée minimale d'un mouvement. Comme le démontre les expérimentations préliminaires au tableau-A IV-6 de l'annexe iv, il est possible d'atteindre un excellent résultat, $EM_{rorqual} > 0.95$, en utilisant un grain temporel plus petit que les spécifications du modèle $t_{step} = 10$ et en ignorant la contrainte de la durée minimale d'un mouvement $t_{limit} = 1$. Cependant, les résultats finaux sont plus décevants avec $EM_{rorqual} = 0.741$ et $EM_{béluga} = 0.513$ puisque l'on tente de respecter le grain temporel du modèle et de reproduire les patrons de mouvement en utilisant $t_{step} = 60$ et $t_{limit} = 10$.

Finalement, le patron de socialisation n'est pas reproduit dans cet algorithme avec un résultat très faible $g_{EMAN} > 0.60$. Cette erreur était prévisible puisque la notion de voisinage n'est pas implémentée dans cet algorithme. Comparativement à l'algorithme de marche aléatoire corrélée, le patron de socialisation s'améliore faiblement puisque la préférence des individus pour certaines zones spécifiques favorise indirectement la proximité entre eux.

5.1.4 Marche aléatoire corrélée avec influence sociale (MAC-S)

Évaluation des paramètres

L'algorithme de regroupement des individus, décrit à la section 3.4, nécessite de fixer quatre paramètres : le rayon d'évitement α , le rayon de socialisation ρ , la vitesse angulaire

maximale θ' et le facteur de pondération de l'influence sociale ω . Le tableau 5.7 résume les valeurs des paramètres utilisés.

Tableau 5.7 Paramètres utilisés pour l'algorithme MAC-S

	α (m)	β (m)	θ' (rad/s)
Rorqual commun	20	2000	0.0123
Béluga	10	2000	0.0177

Le rayon d'évitement α est fixé arbitrairement à partir d'observations des groupes d'individus dans la zone d'étude. Heureusement, ce paramètre n'a pas beaucoup d'influence sur le patron résultant. De manière beaucoup plus rigoureuse, le rayon de socialisation β est fixé à 2000 mètres afin d'être cohérent avec le patron de taille de groupe extrait préalablement. La vitesse angulaire maximale θ' utilise le 97.5^e percentile de la vitesse angulaire observée par espèce. Cet intervalle de confiance est calculé avec la méthode non paramétrique de *bootstrap* avec 1000 itérations. Finalement, le dernier paramètre, le facteur de socialisation ω , est paramétré de manière dynamique afin de reproduire le patron de taille de groupe. Deux méthodes sont proposées pour calculer le paramètre ω dynamiquement, soit par une régression ou par une boucle de contrôle avec rétroaction.

Paramétrage du facteur de socialisation ω par régression

À l'aide d'une régression multilinéaire, il est possible d'estimer la fonction du facteur d'influence social $\omega(n, \bar{g}_{int})$ en fonction de la taille de la population et de la taille de groupe désiré. La régression est basée sur des données extraites de plusieurs simulations du modèle de marche aléatoire corrélée avec influence sociale utilisant les paramètres par défaut, à l'exception de la taille de population initiale n et du facteur d'influence sociale ω qui sont variables. Afin d'éliminer le bruit dû au régime transitoire, les tailles de groupe sont enregistrées les deux derniers jours sur une simulation de 30 jours. Le protocole expérimental ainsi que les résultats sont disponibles à l'annexe iv aux tableaux IV-7 à IV-9.

La régression multilinéaire estime la fonction $\omega(n, \bar{g}_{\text{int}})$ de manière plus précise en appliquant une transformation logarithmique aux données en entrées et en sorties. Le tableau 5.8 décrit la fonction estimée $\omega(n, \bar{g}_{\text{int}})$ ainsi que la précision de la régression par les statistiques RMSE et R^2 .

Tableau 5.8 Équation et statistiques de la régression multilinéaire

	Rorqual commun	Béluga
$\omega(n, \bar{g}_{\text{int}})$	$10^{1.469 \cdot \log(n) - 1.346 \cdot \log(\bar{g}_{\text{int}}) - 0.859}$	$10^{1.277 \cdot \log(n) - 1.319 \cdot \log(\bar{g}_{\text{int}}) - 0.933}$
$RMSE$	0.069320	0.101406
R^2	0.969958	0.935729
\overline{R}^2	0.968623	0.932872

Paramétrage du facteur de socialisation ω par rétroaction

L'utilisation de ces deux équations de régression afin de déterminer le facteur de socialisation en fonction du nombre d'individus et de la taille de groupe observée ne tient pas compte d'une variable importante : le cycle rapide des marées. Ainsi, le facteur de socialisation obtenu par cette régression est valide uniquement en régime permanent. Théoriquement, une solution potentielle serait de complexifier la régression afin d'ajouter deux nouvelles variables en entrées : la taille de groupe lors de la marée précédente et la durée de la marée actuelle. En pratique, une telle démarche serait beaucoup trop coûteuse en temps de calcul.

Une solution alternative est de convertir le problème de régression sous la forme d'un problème d'asservissement. Dans ce cas, la stratégie consiste à fixer initialement les facteurs de socialisation à l'aide de la régression obtenue précédemment. Ensuite, le système est simulé sur une longue période de temps et compile en temps réel le patron de la taille de groupe des agents. Périodiquement, les paramètres de socialisation sont ajustés par un contrôleur en fonction de l'erreur $e(t)$ entre les patrons de socialisation observés g_{obs} et

simulés g_{sim} . Les contrôleurs de type proportionnel (P) et proportionnel intégral dérivé (PID) sont testés pour ce système.

Le contrôleur de type proportionnel est très simple à implémenter et couramment utilisé pour sa simplicité. Les concepteurs d'Écopath (Kavanagh et al., 2004) utilisent ce concept de rétroaction pour paramétrer leur modèle avec une descente de gradient. Ainsi, le paramètre ω_{t+1} est fixé par soustraction de ω_t par une fraction K_p de l'erreur $e(t)$. La figure 5.4 ainsi que les équations 5.1 et 5.2 décrivent le contrôleur de type proportionnel implémenté.

$$e(t) = \int_{t-1}^t (g_{obs} - g_{sim}) \partial t \quad (5.1)$$

$$\omega_{t+1} = \omega_t - K_p e(t) \quad (5.2)$$

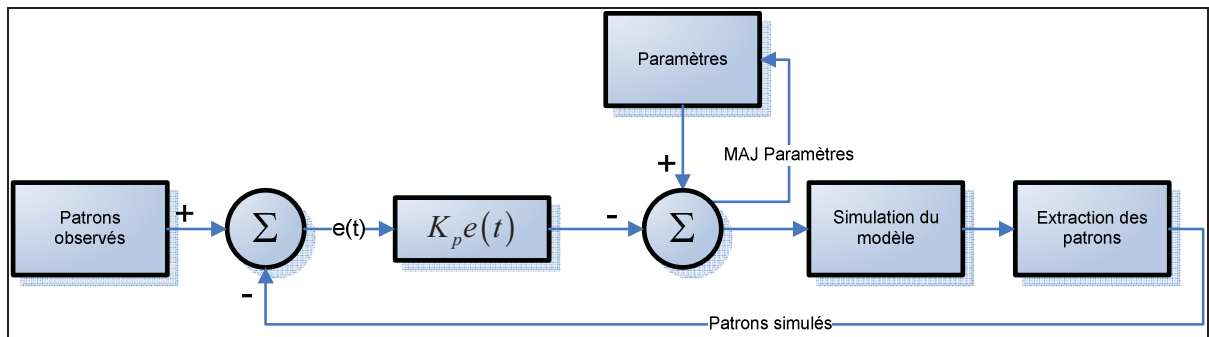


Figure 5.4 Diagramme d'un contrôleur de type proportionnel.

Le second contrôleur implémenté, de type proportionnel intégral dérivé, augmente en complexité. En plus d'utiliser l'erreur $e(t)$, le contrôleur intègre l'ensemble des erreurs depuis le début de la simulation ainsi que la dérivé de l'erreur au temps t .

$$\omega_{t+1} = \omega_t - \left(K_p e(t) + K_i \int_0^t e(x) \partial x + K_d \frac{\partial e(t)}{\partial t} \right) \quad (5.3)$$

Trois paramètres sont présents dans un contrôleur de type PID, soit le coefficient proportionnel K_p , le coefficient intégrateur K_i et le coefficient dérivatif K_d . La méthode de Ziegler–Nichols propose une approche non optimale mais couramment utilisée afin de fixer ces trois coefficients. Cette méthode consiste à déterminer le gain critique K_c à partir duquel un contrôleur proportionnel oscille et la période T_c de cette oscillation. À partir de ces deux informations, les trois coefficients sont calculés :

$$K_p = 0.6K_c \quad (5.4)$$

$$K_i = \frac{2K_p}{T_c} \quad (5.5)$$

$$K_d = \frac{K_p T_c}{8} \quad (5.6)$$

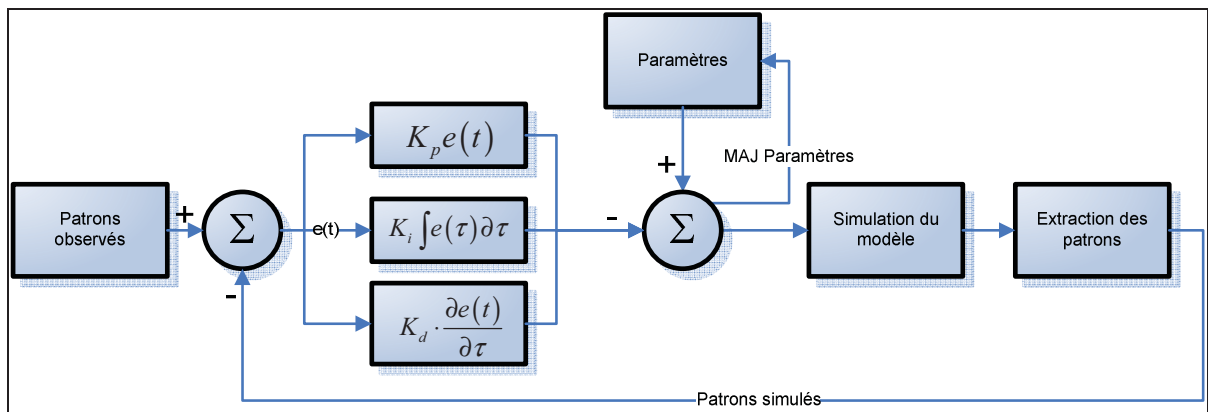


Figure 5.5 Diagramme d'un contrôleur proportionnel intégral dérivé.

Les deux contrôleurs P et PID ont été implémentés et plusieurs expérimentations préliminaires ont été exécutées pour déterminer leurs meilleurs paramètres. Le protocole expérimental de ces expérimentations préliminaires ainsi que les résultats sont disponibles aux tableaux IV-10 et IV-11 de l'annexe iv.

Par la suite, une seconde série d'expérimentation analyse en profondeur le meilleur contrôleur de type P et de type PID pour déterminer quel type de contrôleur à utiliser dans la

boucle de rétroaction. Les résultats de ces expérimentations sont disponibles à l'annexe iv aux tableaux IV-12 et IV-13. Puisque les deux contrôleurs donnent des résultats similaires, le contrôleur de type proportionnel avec un gain de 0.03 est sélectionné pour sa simplicité afin de contrôler par rétroaction le facteur de socialisation.

Comparaison des méthodes de paramétrage du facteur de socialisation ω

Deux méthodes de paramétrage du facteur de socialisation ω ont été étudiées, soit par une fonction de régression ou par une boucle de rétroaction. Les tableaux IV-14 et IV-15 de l'annexe iv présentent le protocole expérimental et les résultats utilisés pour comparer ces deux méthodes.

Dans le cas des bélugas, les deux techniques ont donné des résultats très similaires en fixant le paramètre ω entre 2.49 et 2.72. Cette similarité s'explique par la structure sociale des bélugas qui n'est pas influencée par les marées. Ainsi, le système n'est pas déstabilisé à chaque marée et peut atteindre son régime permanent où la moyenne des tailles de groupe se stabilise en fonction du paramètre ω .

Cependant, la taille des groupes de rorquals communs varie considérablement en fonction des marées. Le rythme des marées, diurnes dans le Saint-Laurent, est beaucoup trop rapide pour atteindre le régime permanent du système. Ainsi, le fait d'être constamment en régime transitoire explique la grande différence entre le paramètre ω fixé par régression et par rétroaction.

Le patron de la taille de groupe est beaucoup mieux respecté en utilisant la méthode de paramétrage avec un contrôleur proportionnel. Pour le rorqual commun, l'erreur moyenne de la taille de groupe est de 0.2485 comparativement à 0.3468 avec l'utilisation de la fonction de régression. À partir de ces résultats, la méthode de rétroaction par contrôleur proportionnel avec un gain de 0.03 est implémentée dans l'algorithme final de marche aléatoire corrélée avec influence sociale.

Compilation des résultats

Tableau 5.9 Résultat de l'expérimentation avec l'algorithme MAC-S pour le rorqual commun

	\bar{X}	$\overline{EMAN} \pm ic$	$\overline{KS} \pm ic$	EM
Vitesse	1.578 ± 0.003	0.193 ± 0.002	0.131 ± 0.001	
Durée	142.435 ± 1.992	0.717 ± 0.004	0.760 ± 0.002	
Distance	181.121 ± 2.487	0.686 ± 0.004	0.555 ± 0.002	
Angle de braquage	0.840 ± 0.007	1.385 ± 0.020	0.395 ± 0.007	
Groupe	7.128 ± 0.462	0.237 ± 0.041	0.155 ± 0.016	
Répartition observée				0.323
Répartition uniforme				0.674

Tableau 5.10 Résultat de l'expérimentation avec l'algorithme MAC-S pour le béluga

	\bar{X}	$\overline{EMAN} \pm ic$	$\overline{KS} \pm ic$	EM
Vitesse	1.794 ± 0.002	0.094 ± 0.001	0.083 ± 0.001	
Durée	130.568 ± 0.585	0.751 ± 0.001	0.657 ± 0.002	
Distance	203.695 ± 0.921	0.691 ± 0.001	0.511 ± 0.002	
Angle de braquage	0.795 ± 0.005	0.506 ± 0.009	0.248 ± 0.004	
Groupe	8.482 ± 0.281	0.145 ± 0.025	0.165 ± 0.008	
Répartition observée				0.335
Répartition uniforme				0.733

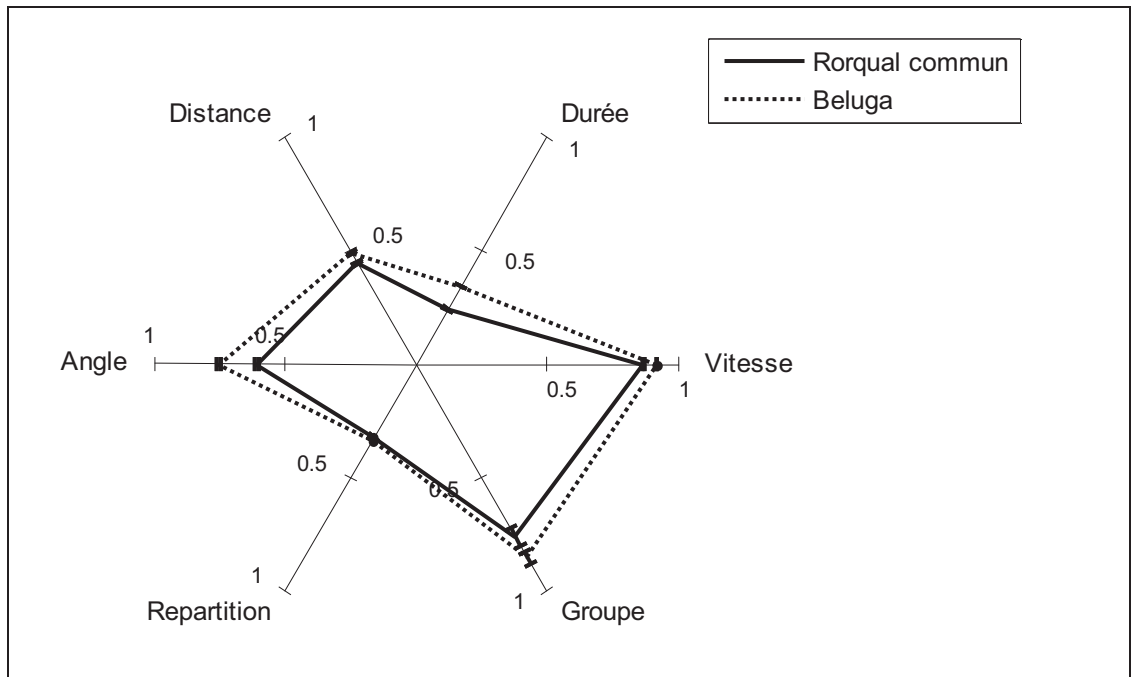


Figure 5.6 Résultat de l'expérimentation avec l'algorithme MAC-S.

Discussion

L'algorithme de marche aléatoire corrélée avec influence sociale MAC-S a permis d'améliorer considérablement le patron de taille de groupe. Cependant, ce gain s'est fait au détriment de certains patrons de mouvement qui ont régressé comparativement à la marche aléatoire corrélée simple.

La vitesse est le seul patron de mouvement qui soit correctement reproduit avec un résultat $v_{KS} < 0.15$. Cependant, les patrons de distance et de durée des mouvements sont très peu représentatifs des données réelles avec des résultats $\tau_{KS}, m_{KS} > 0.50$. Cette piètre performance, comparativement à la marche aléatoire corrélée simple, s'explique par les changements de direction continuels des individus pour s'aligner aux individus environnants. Lors de la simplification des trajectoires, expliquée à la section 4.2.1, chaque mouvement est subdivisé en plusieurs morceaux à cause de sa forme irrégulière. Ainsi, l'angle de braquage est modifié et les patrons de durée et de distance des mouvements deviennent trop courts par rapport aux patrons observés.

Cet algorithme proposé par Couzin s'est avéré très difficile à paramétrer afin de respecter le patron de taille de groupe. Malgré l'utilisation d'un contrôleur de type proportionnel, l'enchaînement des marées est trop rapide pour regrouper efficacement les rorquals communs dans un si court laps de temps. Ainsi, le patron de socialisation est peu satisfaisant pour le rorqual commun avec un résultat $g_{EMAN} = 0.237$. Pour le béluga, qui a une structure sociale stable au fil des marées, le patron de taille de groupe est plus intéressant avec $g_{EMAN} = 0.145$. Il est probable que la méthode de paramétrage par régression aurait donné des résultats similaires ou légèrement meilleurs pour le béluga comme le démontre les expérimentations préliminaires présentées au tableau-A IV-11 de l'annexe iv. Cependant, le contrôleur de type proportionnel a été sélectionné afin d'obtenir un algorithme générique fonctionnel pour toutes les espèces.

Enfin, le patron de distribution spatiale des mammifères n'est aucunement reproduit avec l'algorithme MAC-S avec un résultat $EM < 0.35$. Ce résultat était prévisible puisque, comme les algorithmes MA et MAC, cet algorithme n'utilise aucune information concernant la répartition spatiale.

5.1.5 Marche aléatoire corrélée avec indice de résidence et influence sociale (MAC-RS)

Évaluation des paramètres

L'algorithme MAC-RS, décrit à la section 3.5, utilise simultanément l'indice de résidence et le facteur de socialisation pour déterminer le prochain mouvement à effectuer. Ces deux méthodes ont été paramétrées indépendamment l'une de l'autre lors de l'implémentation des algorithmes MAC-R et MAC-S. Les paramètres trouvés précédemment sont réutilisés directement dans cet algorithme, ce qui permet d'éliminer complètement la phase d'évaluation des paramètres.

Compilation des résultats

Tableau 5.11 Résultat de l'expérimentation avec l'algorithme MAC-RS pour le rorqual commun

	\bar{X}	$\overline{EMAN} \pm ic$	$\overline{KS} \pm ic$	EM
Vitesse	1.108 ± 0.014	0.163 ± 0.011	0.135 ± 0.005	
Durée	184.633 ± 4.108	0.633 ± 0.008	0.757 ± 0.004	
Distance	159.200 ± 2.246	0.724 ± 0.004	0.636 ± 0.004	
Angle de braquage	0.785 ± 0.011	1.230 ± 0.031	0.346 ± 0.010	
Groupe	6.695 ± 0.316	0.303 ± 0.046	0.189 ± 0.019	
Répartition observée				0.306
Répartition uniforme				0.752

Tableau 5.12 Résultat de l'expérimentation avec l'algorithme MAC-RS pour le béluga

	\bar{X}	$\overline{EMAN} \pm ic$	$\overline{KS} \pm ic$	EM
Vitesse	1.347 ± 0.024	0.179 ± 0.014	0.139 ± 0.011	
Durée	156.620 ± 1.619	0.701 ± 0.003	0.664 ± 0.003	
Distance	178.441 ± 3.008	0.729 ± 0.005	0.611 ± 0.006	
Angle de braquage	0.718 ± 0.015	0.360 ± 0.029	0.167 ± 0.011	
Groupe	10.317 ± 0.826	0.324 ± 0.102	0.299 ± 0.011	
Répartition observée				0.428
Répartition uniforme				0.708

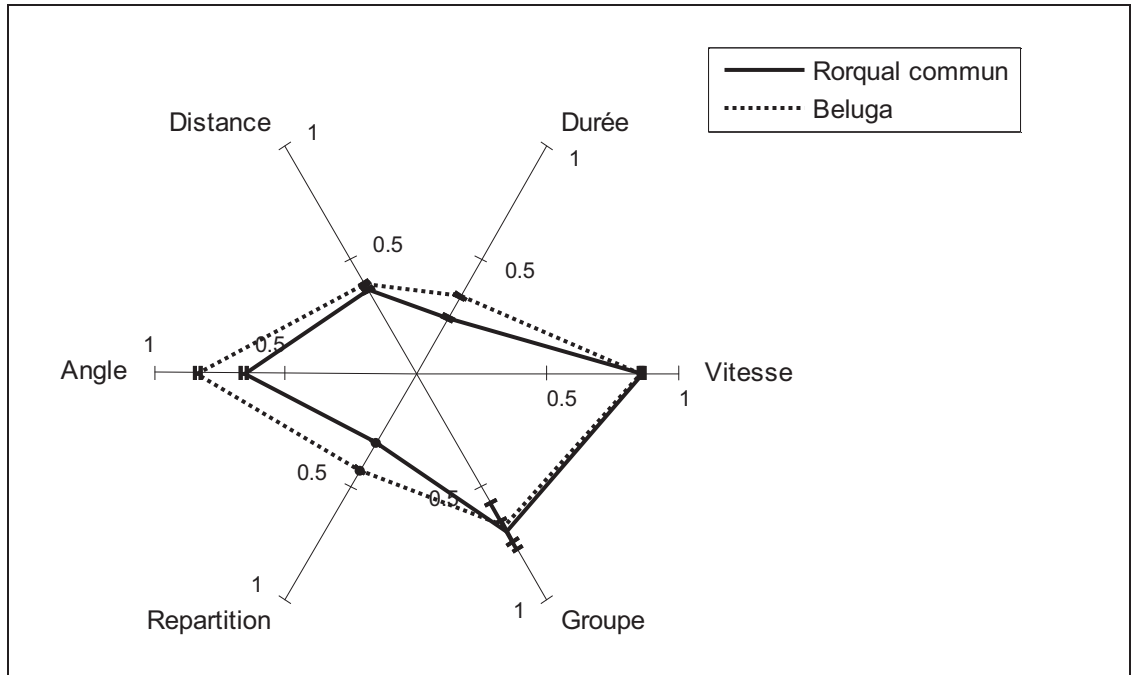


Figure 5.7 Résultat de l'expérimentation avec l'algorithme MAC-RS.

Discussion

L'algorithme MAC-RS utilise comme fondement la marche aléatoire corrélée qui reproduit correctement les patrons de trajectoire. À cet algorithme de base, nous ajoutons l'indice de résidence et le facteur de socialisation qui ont reproduit approximativement les patrons de répartition spatiale et de taille de groupe dans les expérimentations précédentes. Cependant, l'utilisation de tous ces mécanismes simultanément n'a pas donné les résultats escomptés.

Les patrons de trajectoire obtenus avec l'algorithme MAC-RS sont très similaires à ceux obtenus précédemment avec l'algorithme MAC-S. Expliquée précédemment dans la discussion de l'algorithme MAC-S à la section 5.1.4, la composante sociale rend la trajectoire très sinueuse et fausse l'analyse des patrons de trajectoire. Ainsi, il y a seulement le patron de vitesse qui est reproduit avec $v_{KS} < 0.15$ alors que les patrons de distance, de durée et d'angle de braquage ne sont pas respectés avec $\tau_{KS}, m_{KS} > 0.50$ et $\psi_{EMAN} > 0.35$.

De plus, le patron de taille de groupe n'est pas respecté avec des résultats de $g_{EMAN} = 0.303$ pour le rorqual commun et de $g_{EMAN} = 0.324$ pour le béluga. Pour le rorqual commun, la taille de groupe simulée est insuffisante comparativement aux données observées, ce qui laisse croire que l'algorithme MAC-RS diffuse trop les individus dans leur environnement pour que l'algorithme de Couzin puisse les regrouper. Pour les bélugas, c'est le cas inverse qui se produit où la taille de groupe simulée est trop grande, potentiellement causé par la très forte concentration de bélugas près des îles de Kamouraska.

Enfin, le patron de répartition n'est pas respecté avec des résultats $EM < 0.45$ pour les deux espèces, ce qui est encore plus faible qu'avec l'algorithme MAC-R. Cette baisse des résultats est explicable par le fait que l'algorithme MAC-RS ne répond plus à deux prémisses de la marche aléatoire corrélée avec indice de résidence : les mouvements ne sont plus linéaires et les individus ont des interactions entre eux. À partir de ce point, les équations de diffusion de cet algorithme ne sont plus valides et donnent ces résultats finaux mitigés.

5.1.6 Marche aléatoire corrélée avec minimisation de la moyenne des biais normalisés (MAC-MMBN)

Évaluation des paramètres

L'algorithme MAC-MMBN sélectionne l'angle de braquage qui minimise l'erreur des patrons de répartition, d'angle de braquage et de taille de groupe. La fonction objectif de ce problème d'optimisation utilise la somme pondérée des biais présentée à l'équation 5.7. Ainsi, il est nécessaire de paramétrer les trois poids w_u, w_ψ, w_g pour implémenter l'algorithme final de MAC-MMBN.

En utilisant un pas de 0.10 dans l'intervalle $[0,1]$ pour chacun des paramètres, il existe 66 possibilités de combinaisons des trois paramètres, en imposant que leur somme soit égale à 1. Les expérimentations préliminaires de ces 66 combinaisons ont permis d'éliminer certaines combinaisons inefficaces et de conserver les 25 meilleures. Par la suite, une seconde série d'expérimentations a permis d'analyser plus rigoureusement ces 25 combinaisons. Le

protocole expérimental ainsi que les résultats de cette seconde série d'expérimentations sont disponibles aux tableaux IV-16 et IV-17 de l'annexe iv. Les paramètres finaux qui minimisent l'erreur moyenne, décrite à l'équation 5.7, sont $w_u = 0.08$, $w_g = 0.60$ et $w_\psi = 0.32$.

$$\text{erreur} = \frac{\psi_{ks_{commun}} + g_{e_{commun}} + (1 - ME_{obs_{commun}}) + \psi_{ks_{beluga}} + g_{e_{beluga}} + (1 - ME_{obs_{beluga}})}{6} \quad (5.7)$$

Compilation des résultats

Tableau 5.13 Résultat de l'expérimentation avec l'algorithme MAC-MMBN pour le rorqual commun

	\bar{X}	$\overline{EMAN} \pm ic$	$\overline{KS} \pm ic$	EM
Vitesse	1.338 ± 0.001	0.011 ± 0.001	0.025 ± 0.000	
Durée	480.704 ± 1.156	0.043 ± 0.002	0.097 ± 0.002	
Distance	626.814 ± 1.672	0.086 ± 0.003	0.037 ± 0.001	
Angle de braquage	0.255 ± 0.001	0.274 ± 0.003	0.078 ± 0.001	
Groupe	6.605 ± 0.454	0.128 ± 0.015	0.280 ± 0.005	
Répartition observée				0.942
Répartition uniforme				-2.514

Tableau 5.14 Résultat de l'expérimentation avec l'algorithme MAC-MMBN pour le béluga

	\bar{X}	$\overline{EMAN} \pm ic$	$\overline{KS} \pm ic$	EM
Vitesse	1.666 ± 0.001	0.015 ± 0.001	0.027 ± 0.000	
Durée	394.975 ± 3.354	0.245 ± 0.006	0.147 ± 0.007	
Distance	625.557 ± 5.916	0.050 ± 0.009	0.095 ± 0.006	
Angle de braquage	0.377 ± 0.003	0.286 ± 0.005	0.117 ± 0.001	
Groupe	7.035 ± 0.020	0.110 ± 0.003	0.540 ± 0.004	
Répartition observée				0.831
Répartition uniforme				-0.164

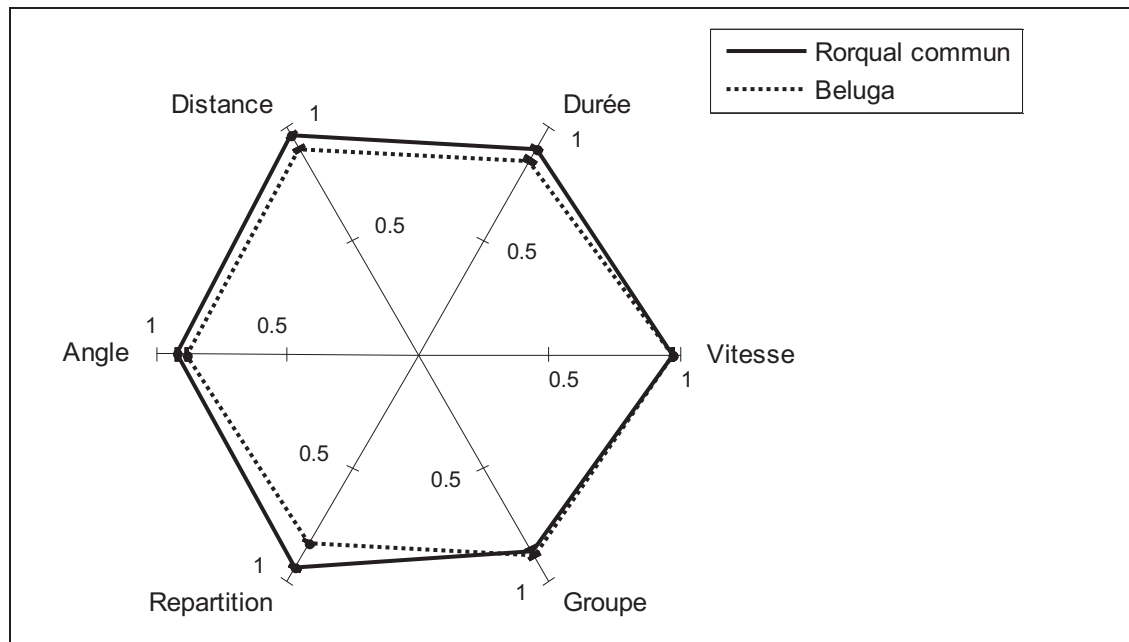


Figure 5.8 Résultat de l'expérimentation avec l'algorithme MAC-MMBN.

Discussion

L'algorithme de MAC-MMBN est le seul algorithme qui reproduit simultanément les six patrons de déplacement des mammifères marins. La marche aléatoire corrélée permet de recréer trois de ces patrons et la méthode de minimisation de la moyenne des biais normalisés permet de reproduire les trois patrons restants.

Lors du déplacement des mammifères marins, les vitesses et les durées des mouvements sont pigées avec la méthode de Monte-Carlo typique de la marche aléatoire. La distance des mouvements est un patron secondaire car elle n'est pas pigée explicitement, mais calculée à partir de ces deux premières variables. Cette méthode permet d'obtenir d'excellents résultats, avec $\tau_{KS}, v_{KS}, m_{KS} < 0.10$, qui démontrent clairement que le modèle reproduit correctement la trajectoire des individus.

L'angle de braquage, la répartition spatiale et la taille de groupe sont les trois patrons qui sont contrôlés par la méthode de minimisation des biais. Expliqué précédemment dans le chapitre 3.6, ces trois patrons sont souvent en contradiction lorsque vient le temps de déterminer la direction du prochain mouvement. En pondérant correctement l'importance de chacun des patrons lors de l'évaluation des paramètres, il est possible de répartir l'erreur équitablement entre eux. Les résultats obtenus pour ces trois patrons, $\psi_{KS} < 0.12$, $EM > 0.83$ et $g_{EMAN} < 0.13$, démontrent que l'algorithme reproduit adéquatement ces patrons pour les deux espèces.

5.2 Temps de simulation

La complexité de traitement des six algorithmes implémentés varie considérablement. Dans le but de décrire grossièrement cet écart, le temps nécessaire pour simuler 24 réplifications de 30 jours avec 135 individus est noté pour chaque algorithme. Cet exercice n'a pas la prétention de mesurer la complexité algorithmique des fonctions, mais uniquement de compiler un descripteur de performance relatif facilement interprétable. Pour ne pas biaiser les résultats, toutes les simulations ont été exécutées séquentiellement, sur la même machine de calcul et sans aucun autre programme ouvert en parallèle. Le tableau 5.15 compile le temps d'exécution des expérimentations en fonction des algorithmes ainsi que le ratio de temps d'exécution par rapport à la marche aléatoire simple. Le temps de simulation englobe le temps de chargement de 24 réplifications du modèle, l'exécution d'approximativement 140

millions de déplacements d'individus et l'écriture de 24 fichiers de sortie de 333 mégaoctets chacun.

Tableau 5.15 Temps d'exécution des expérimentations en fonction de l'algorithmes et ratio de temps d'exécution par rapport à la marche aléatoire simple

	MA	MAC	MAC-R	MAC-S	MAC-RS	MAC-MMBN
Temps (minutes)	189	189	194	276	278	460
Ratio	1.00	1.00	1.03	1.46	1.47	2.44

Le temps d'exécution de l'algorithme le plus simple, la marche aléatoire, est utilisé comme temps de référence. Il est possible de constater que les algorithmes MAC et MAC-R ne sont pas significativement plus longs à exécuter que le temps de référence, ce qui est compréhensible puisque ces algorithmes modifient uniquement la façon dont l'angle de braquage et la distance du déplacement sont pigés. Cependant, l'ajout de l'influence sociale dans les algorithmes MAC-S et MAC-RS augmente significativement le temps d'exécution avec un ratio de 1.46 et 1.47 respectivement. Ces algorithmes utilisent deux nouvelles fonctions, la recherche des individus dans le voisinage et la sommation des vecteurs des individus environnants, qui s'avèrent très couteuses en temps de traitement. Finalement, l'algorithme MMBN est le plus lent avec un ratio de 2.44. Ce long temps de traitement, 144% plus long que la marche aléatoire, s'explique par le fait que l'algorithme projette 20 mouvements en considérant les individus environnants avant de déterminer lequel de ces mouvements est le plus susceptible de reproduire tous les patrons.

Ainsi, les algorithmes MA, MAC et MAC-R sont les trois algorithmes les plus rapides et l'algorithme MMBN est définitivement le plus lent. Cependant, le temps de traitement n'est pas un point critique dans la sélection d'un modèle. La différence de quelques heures dans l'exécution d'une simulation est malgré tout négligeable et ne sera pas prise en compte lors de la sélection finale de l'algorithme.

5.3 Discussion générale des approches utilisées

Cette section a pour objectif d'analyser les résultats dans leur ensemble. À partir de cette analyse, un algorithme est recommandé pour l'implémenter dans le modèle final.

L'objectif initial de ce modèle était de modéliser avec une approche ascendante les déplacements spatio-temporels des rorquals communs et des bélugas dans une portion du Saint-Laurent. Ainsi, à partir des patrons identifiés précédemment, l'utilisation d'une approche ascendante implique de modéliser les mouvements locaux des individus et de laisser émerger les patrons globaux de la répartition spatiale et de la taille de groupe. Les cinq premiers algorithmes, MA, MAC, MAC-R, MAC-S et MAC-RS, utilisent tous cette approche. Malheureusement, pour diverses raisons expliquées dans les discussions précédentes de la section 5.1, aucun de ces cinq algorithmes n'a été en mesure de reproduire adéquatement tous les patrons simultanément comme le résume le tableau 5.16.

À partir de cette impasse, la modélisation des mammifères avec une approche descendante a été tentée. Ce type d'approche implique que les patrons globaux de répartition spatiale et de taille de groupe contrôlent les mouvements locaux des individus. Pour coupler ces patrons globaux aux mouvements locaux, la fonction pondérée proposée dans l'algorithme MAC-MMBN évalue plusieurs mouvements et détermine lequel d'entre eux génère les meilleurs patrons. Cette approche descendante a donné d'excellents résultats en reproduisant correctement l'ensemble des patrons malgré un temps de traitement plus long comparativement aux approches ascendantes.

Tableau 5.16 Tableau récapitulatif des patrons reproduits en fonction de l’algorithme de déplacement utilisé

	MA	MAC	MAC-R	MAC-S	MAC-RS	MAC-MMBN
Vitesse	✓	✓	✓	✓	✓	✓
Durée	✓	✓				✓
Distance	✓	✓				✓
Angle de braquage		✓				✓
Groupe				✓		✓
Répartition observée						✓

Enfin, l’algorithme MAC-MMBN est recommandé pour modéliser les déplacements des mammifères marins dans le modèle global final. En plus de reproduire les six patrons identifiés précédemment, la structure de l’algorithme permet de complexifier facilement le modèle en introduisant de nouveaux patrons. Pour ce faire, il suffit d’ajouter des termes supplémentaires dans la fonction objectif et de rééquilibrer les poids de la fonction, comparativement aux approches ascendantes qui sont difficilement modifiables sans altérer le comportement global du modèle. Ainsi, la structure simple et évolutive de l’algorithme implémenté permettra d’utiliser facilement le modèle dans des projets futurs.

CONCLUSION

Dans ce mémoire, nous avons tenté de répondre à la question : « *Comment représenter les déplacements spatio-temporels des rorquals communs et des bélugas dans le Parc Marin Saguenay-Saint-Laurent au cours de la saison touristique ?* ». Pour y parvenir, l'approche de modélisation basée sur individus a été utilisée. Le développement d'un modèle, spatialement explicite, a permis de simuler le déplacement de plusieurs individus dans le parc marin et ses environs avec une résolution de 100 mètres.

Pour structurer le processus de développement du modèle, l'approche de modélisation orientée par patrons a été utilisée. Cette approche a permis d'identifier les patrons importants dans le système qui doivent être reproduits dans le modèle. Par la suite, cette approche de modélisation nous a permis de valider quantitativement le modèle en mesurant la similarité des patrons simulés et observés.

Il y a trois types de patrons qui ont été jugés importants à inclure dans le modèle : les patrons de trajectoire, de taille de groupe et de répartition spatiale. Les patrons de trajectoire, comprenant la vitesse, la durée, la distance et l'angle de braquage des mouvements, ont été extraits à partir de 80 suivis d'individus qui ont été discrétisés et simplifiés. Le patron de taille de groupe, dépendant de la marée pour le rorqual commun selon une analyse préliminaire, a été extrait à partir des données AOM. Finalement, ces données ont aussi été utilisées pour créer une carte de distribution pour chaque espèce, en corrigeant préalablement certains biais d'effort et d'échantillonnage inhérents à ces données.

Pour comparer ces patrons observés aux patrons simulés par notre modèle, nous avons utilisé trois métriques. La statistique de Kolmogorov-Smirnov (KS) ainsi que l'erreur moyenne absolue normalisée (EMAN) ont permis de mesurer la similarité entre deux ensembles de données provenant des patrons de trajectoire et de taille de groupe. Cependant, pour comparer le patron de répartition spatiale, la mesure d'efficacité du modèle (EM) a été privilégiée afin de quantifier la ressemblance entre deux cartes.

Ces trois mesures de similarité varient à chaque répllication pour un même modèle, causé par les processus stochastiques qui sont utilisés lors de la simulation. Pour décrire cette variabilité inter-répllication avec un intervalle de confiance, le test de Student a été utilisé. Ainsi, nous avons effectué 24 répllications pour chacun de nos modèles, calculé les trois métriques de similarité à chaque répllication et enfin utilisé le test de Student pour déterminer l'intervalle de confiance de ces métriques. Cette démarche a permis de s'assurer que la variation inter-répllication des modèles n'était pas excessive et que le comportement des modèles était relativement prédictible.

Finalement, pour tenter de répondre à la question « *Comment représenter le déplacement des rorquals communs et des bélugas* » en respectant tous les patrons énumérés précédemment, nous avons implémenté six algorithmes de déplacement. En privilégiant une approche ascendante, nous avons tenté d'utiliser cinq algorithmes basés sur la théorie de la marche aléatoire. Cependant, aucun d'entre eux n'a été en mesure de reproduire simultanément les patrons globaux de taille de groupe et de répartition spatiale en contrôlant uniquement les mouvements locaux. Ainsi, nous avons proposé un algorithme avec une approche descendante, la marche aléatoire corrélée avec minimisation de la moyenne des biais normalisés, qui utilise une fonction objectif pondérée pour déterminer le meilleur mouvement à effectuer. Cet algorithme s'avère être une réponse adéquate à la question de modélisation puisqu'il reproduit convenablement l'ensemble des patrons observés. Ainsi, le modèle final développé dans le cadre de ce mémoire utilise l'algorithme MAC-MMBN et déplace correctement les rorquals communs et les bélugas dans le Parc Marin Saguenay-Saint-Laurent au cours de la saison touristique.

La contribution de ce mémoire au domaine de la modélisation de systèmes écologiques se situe à trois niveaux : structurel, algorithmique et applicatif. En premier lieu, ce mémoire présente une démarche complète de modélisation orientée par patrons structurée selon les règles de l'art. Il présente séquentiellement les techniques et les stratégies utilisées pour mener à terme chaque étape d'un projet concret de modélisation par patrons. Au niveau

algorithmique, ce mémoire contribue à explorer l'utilisation de différents algorithmes de déplacement dans un contexte de modélisation orientée par patrons. Nous proposons un algorithme simple, la marche aléatoire corrélée avec minimisation de la moyenne des biais normalisés, qui permet de déterminer les meilleurs mouvements à effectuer pour respecter tous les patrons, même ceux en contradiction les uns avec les autres. Finalement, la contribution applicative de ce mémoire sera d'intégrer notre modèle dans un modèle beaucoup plus large développé par notre groupe de recherche. Celui-ci tentera de modéliser le déplacement des mammifères marins et le trafic maritime dans l'estuaire du Saint-Laurent afin de simuler l'effet de différents scénarios de réglementation et de zonage dans le parc marin. Nous souhaitons que le fruit de ce travail, le modèle développé par notre groupe de recherche, fasse une différence dans la protection et la préservation des mammifères marins dans le Parc Marin Saguenay-Saint-Laurent.

RECOMMANDATIONS

Ce projet de modélisation comporte de nombreuses facettes que nous n'avons pas eu le temps d'explorer. Nous proposons ici trois fonctionnalités intéressantes qui n'ont malheureusement pas été implémentées par manque de temps ou de données.

Une première amélioration simple au modèle serait d'ajouter le patron de cycle respiratoire pour les mammifères marins. Ce patron, très important dans le processus d'interaction avec les bateaux, n'a pas été implémenté par manque de données. Celui-ci serait très simple à intégrer dans le modèle puisqu'il ne modifie pas les autres patrons utilisés actuellement. Une simple pige aléatoire dans la fonction de probabilité cumulative de la durée des intervalles respiratoires observés permettrait de remplacer les durées constantes actuellement utilisées par le modèle.

Une seconde amélioration potentielle au modèle serait d'associer des classes comportementales aux mammifères marins. Dans le cadre d'un autre projet, nous avons extrait les attributs de durée, vitesse, distance, angle de braquage, bathymétrie, gradient de la bathymétrie et marée pour chaque observation dans les données de suivis d'individus. À l'aide de l'algorithme de regroupement automatique KPPV à 3 classes, nous avons été en mesure de classer ces observations selon les trois comportements typiques des mammifères marins : alimentation, nage directionnelle et repos. Ce type d'analyse pourrait être réalisé à nouveau afin d'enrichir notre base de données géographiques en ajoutant l'attribut du comportement de l'individu. Par la suite, ces données enrichies permettraient d'extraire les fonctions de répartition conditionnelles à l'état comportemental de l'individu et d'utiliser ces fonctions conditionnelles directement dans le modèle.

Finalement, les mammifères ne sont pas réactifs face aux bateaux dans la version actuelle du modèle. Certains membres de notre groupe de recherche travaillent actuellement à caractériser cette réaction des mammifères pour éventuellement l'implémenter dans une version ultérieure du modèle. Les deux réactions investiguées sont le changement de

l'intervalle respiratoire et de la direction des mammifères en présence d'un bateau. Dans chacun des cas, la structure actuelle de l'application permettra d'implémenter rapidement ces réactions dans le modèle grâce à l'approche modulaire du sous-modèle respiratoire et à la flexibilité de l'algorithme MAC-MMBN. Cette dernière recommandation permettrait d'exploiter au maximum l'approche de modélisation basée sur individus afin d'étudier l'interaction mutuelle entre tous les individus, mammifères et bateaux, dans le modèle global.

ANNEXE I

DONNÉES GÉOGRAPHIQUES DISPONIBLES

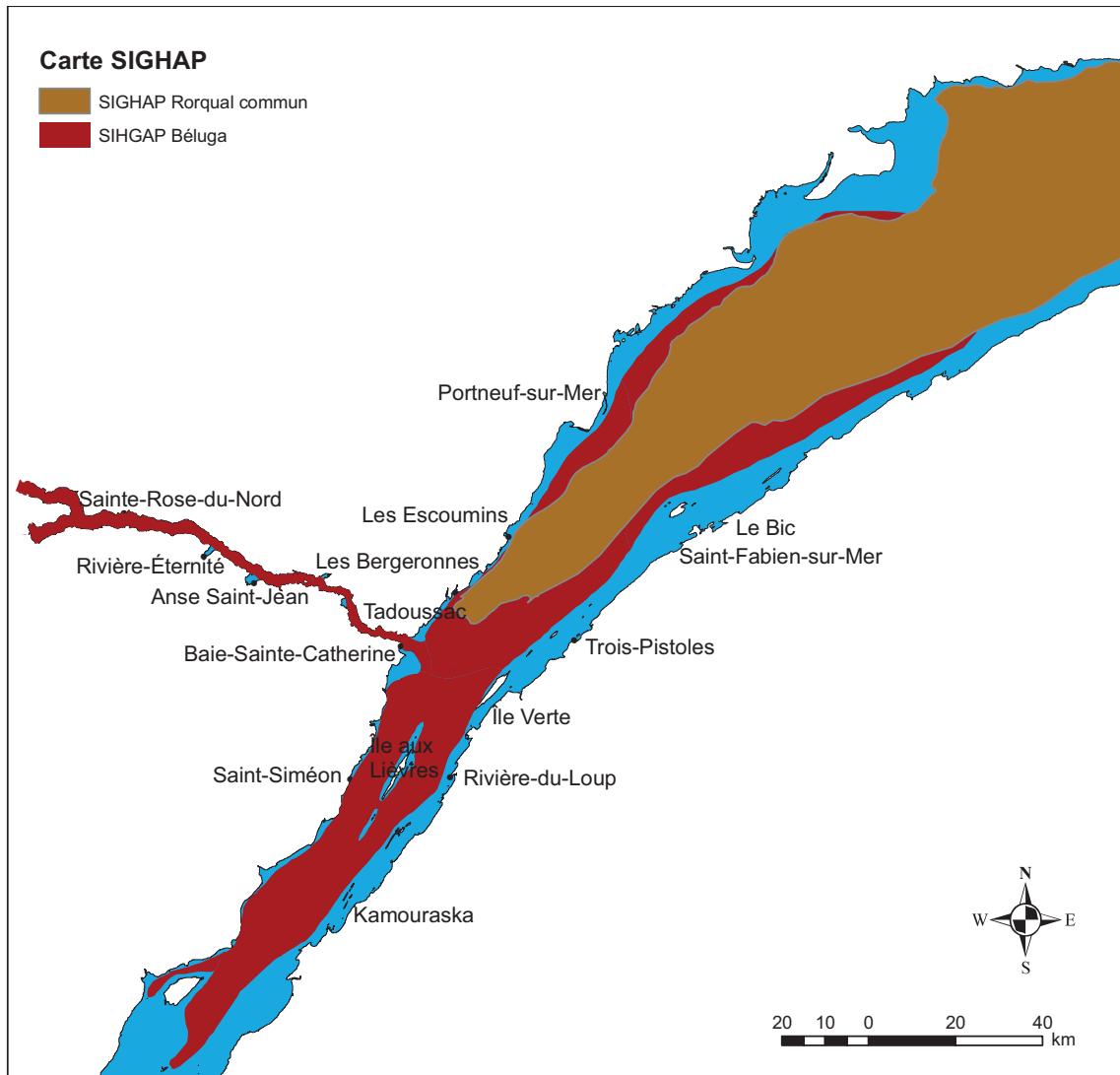


Figure-A I-1 Carte SIGHAP du rorqual commun et du béluga dans l'aire d'étude.

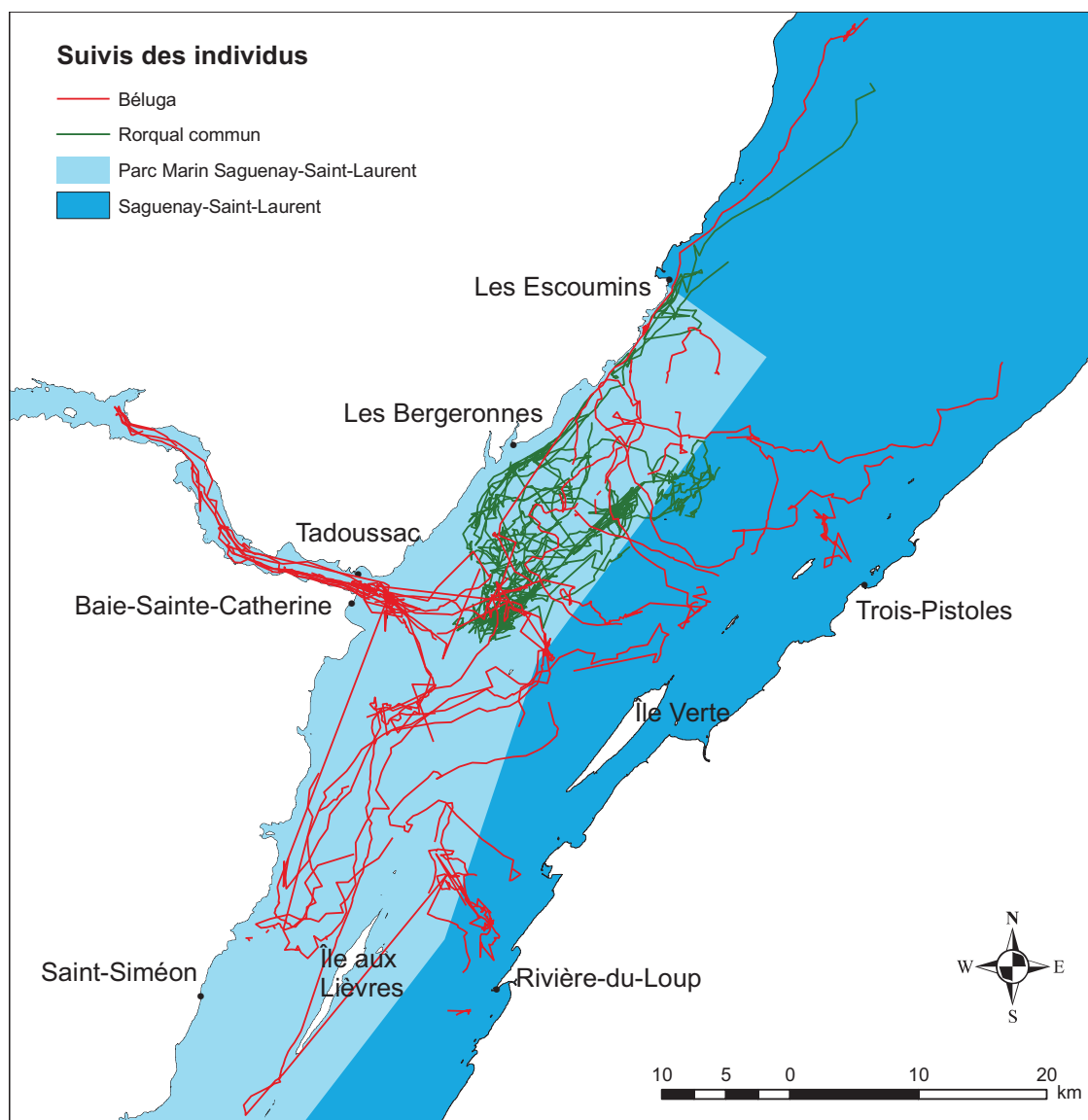


Figure-A I-2 Carte des suivis d'individus des rorquals communs et des bélugas.

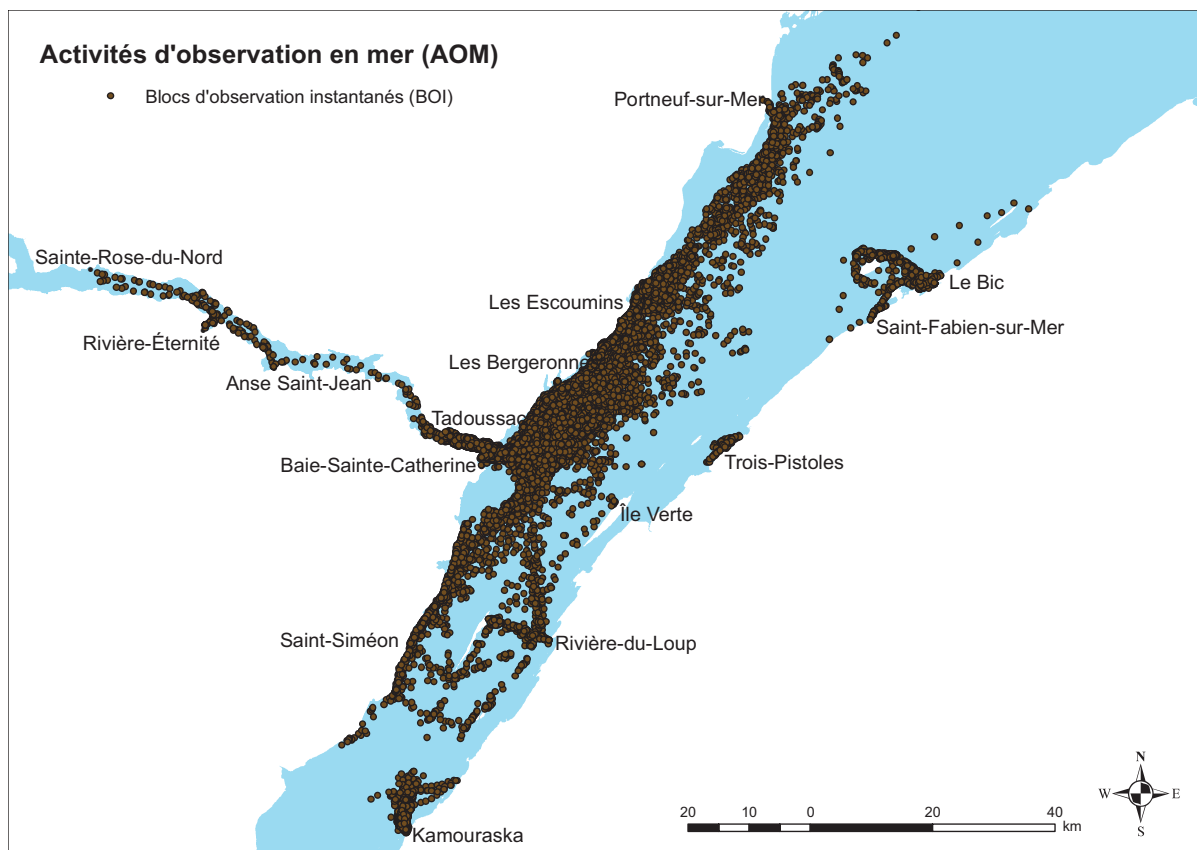


Figure-A I-3 Carte des observations AOM disponibles.

ANNEXE II

RÉPARTITION SPATIALE DES ESPÈCES EN FONCTION DES FILTRES DE PRÉTRAITEMENT

Tableau-A II-1 Répartition spatiale des rorquals communs en fonction des filtres de prétraitement

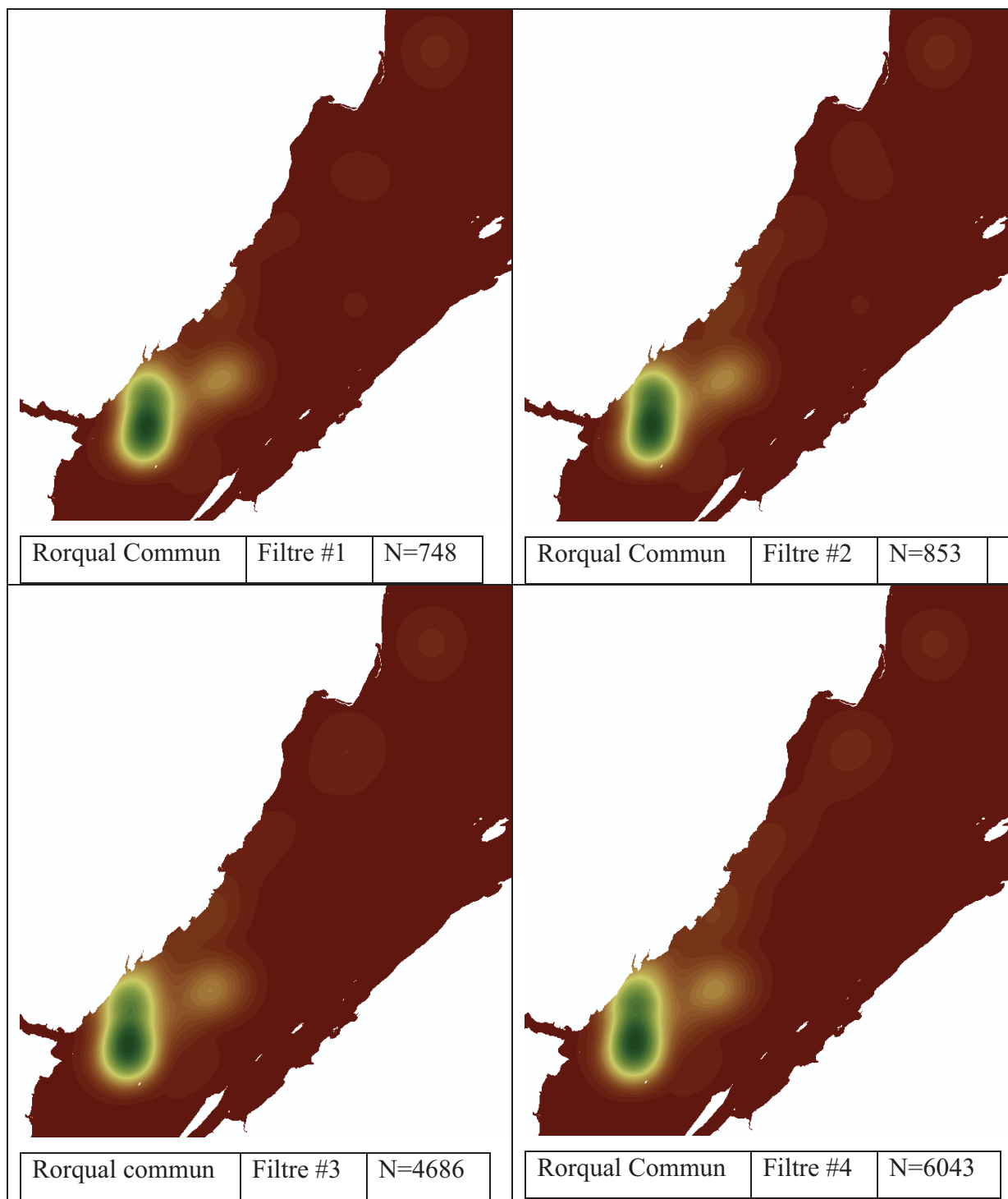
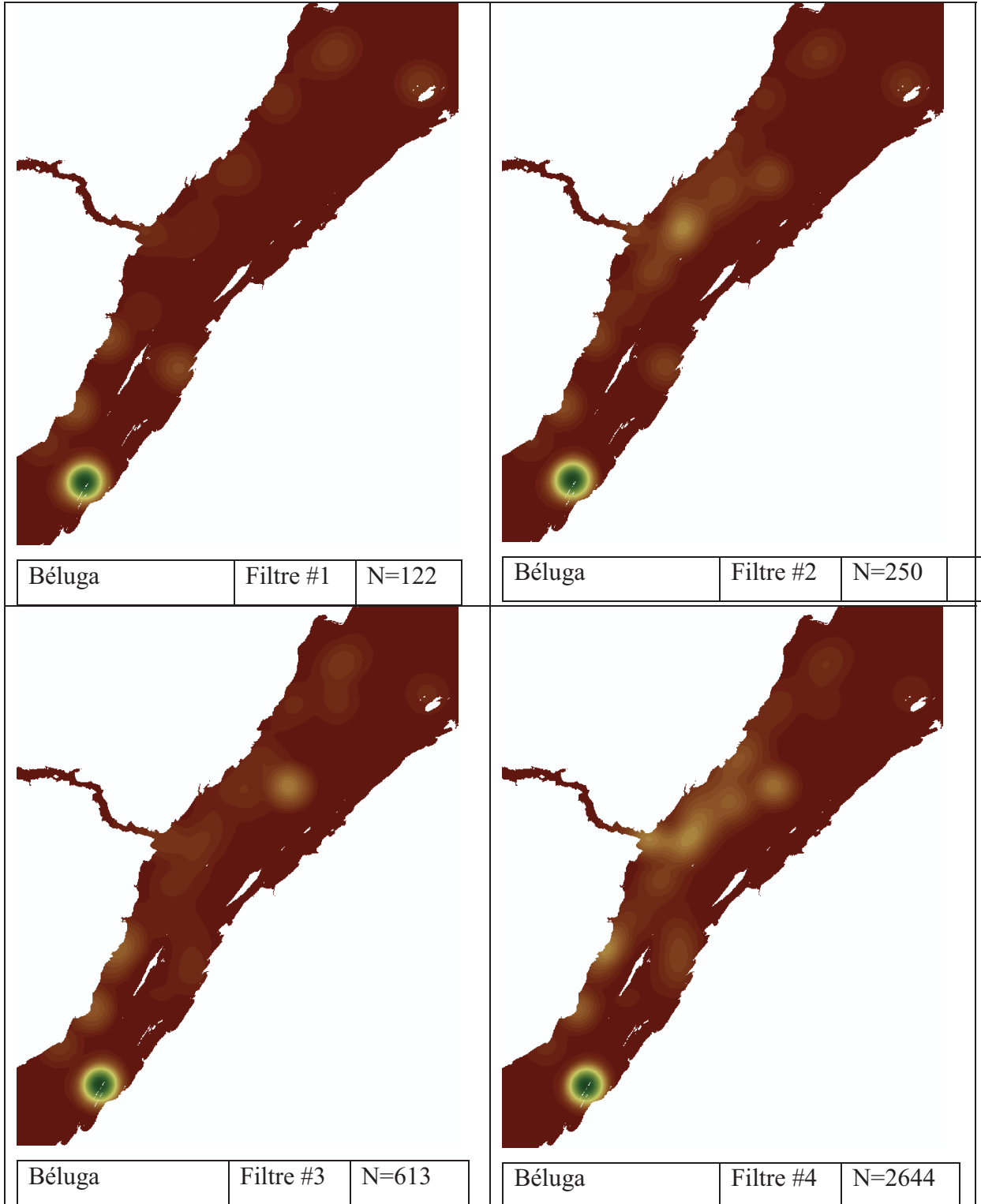


Tableau-A II-2 Répartition spatiale des bélugas en fonction des filtres de prétraitement



ANNEXE III

CARTE DE RÉPARTITION DES ESPÈCES

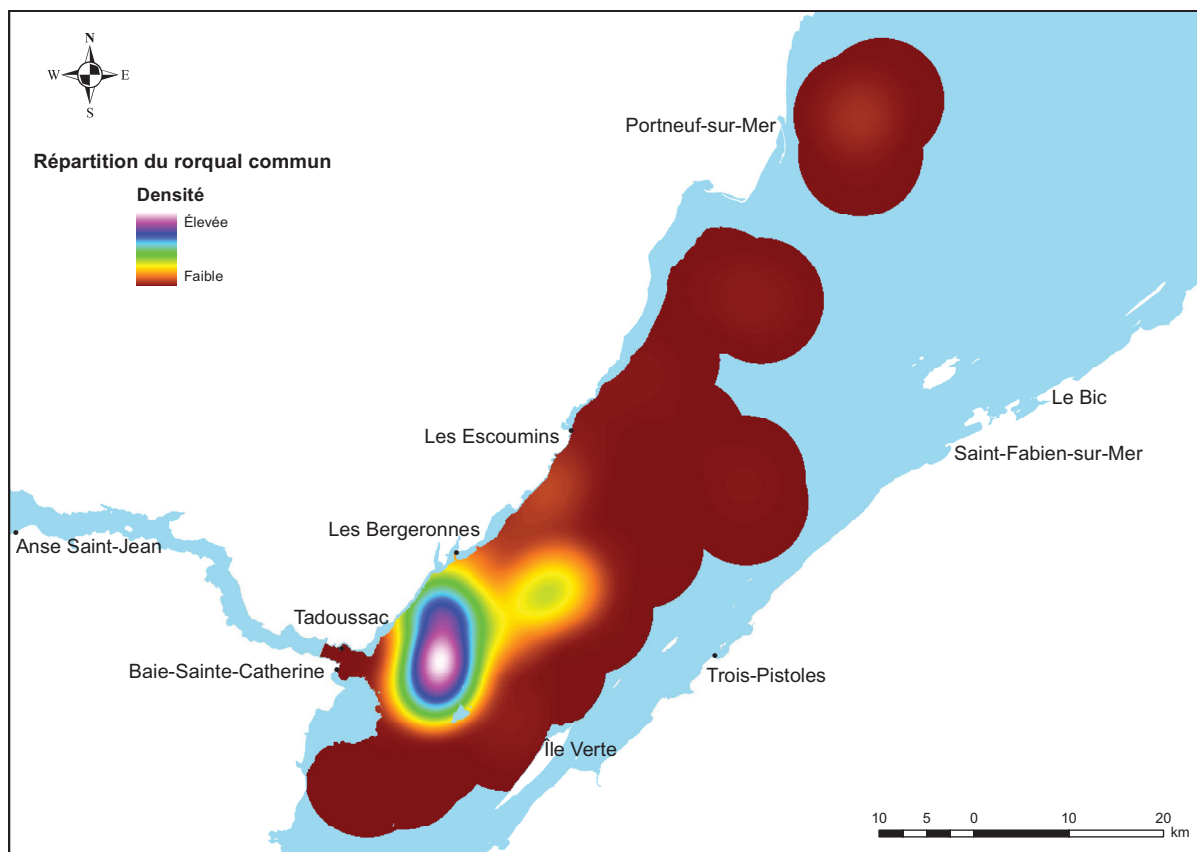


Figure-A III-1 Carte de répartition du rorqual commun.

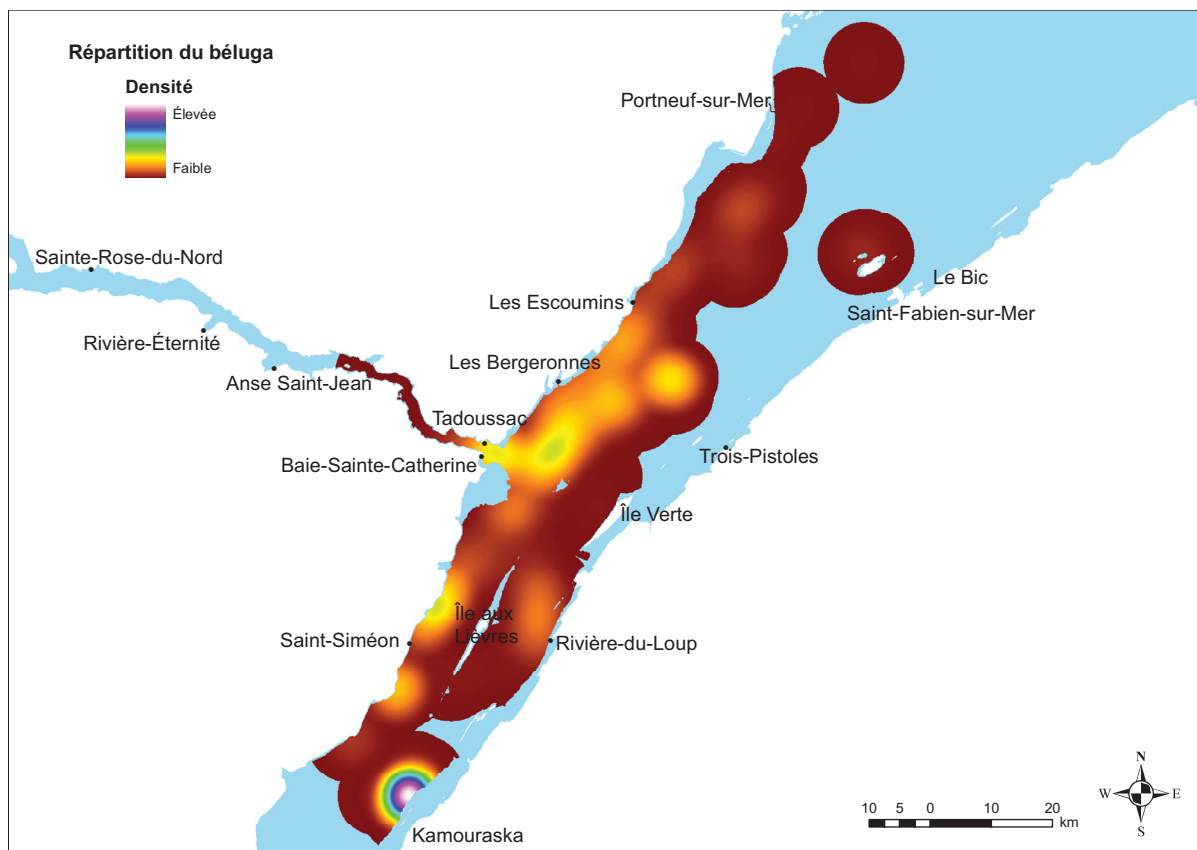


Figure-A III-2 Carte de répartition du béluga.

ANNEXE IV

ÉVALUATION DES PARAMÈTRES

Tableau-A IV-1 Paramètres des expérimentations avec l'algorithme MA

Durée de la simulation	30 jours
Pas de temps	60 secondes
Nombre de réplifications	10
Mammifères présents	35 rorquals communs 100 bélugas
Mode de déplacement	MA
Enregistrement des données	60 secondes

Tableau-A IV-2 Résultats des expérimentations avec l'algorithme MA

Stratégie	Espèce	v_{ks}	τ_{ks}	m_{ks}	ψ_{ks}	ME_{umi}
Réactive	Rorqual commun	0.0160	0.213	0.111	0.274	0.923
Préventive	Rorqual commun	0.0154	0.159	0.0699	0.269	0.918
Réactive	Béluga	0.0162	0.219	0.157	0.389	0.919
Préventive	Béluga	0.0149	0.142	0.099	0.380	0.900

Tableau-A IV-3 Paramètres des expérimentations avec l'algorithme MAC

Durée de la simulation	30 jours
Pas de temps	60 secondes
Nombre de réplifications	10
Mammifères présents	35 rorquals communs 100 bélugas
Mode de déplacement	MAC
Enregistrement des données	60 secondes

Tableau-A IV-4 Résultats des expérimentations avec l'algorithme MAC

Stratégie	Espèce	v_{ks}	τ_{ks}	m_{ks}	ψ_{ks}	ME_{umi}
Réactive	Rorqual commun	0.0222	0.199	0.0961	0.0379	0.920
Préventive	Rorqual commun	0.0186	0.153	0.0676	0.0253	0.916
Réactive	Béluga	0.0284	0.179	0.114	0.0692	0.918
Préventive	Béluga	0.0200	0.126	0.0803	0.0551	0.912

Tableau-A IV-5 Paramètres des expérimentations avec l'algorithme MAC-R

Durée de la simulation	30 jours
Pas de temps	60 secondes
Nombre de réplifications	2
Mammifères présents	35 rorquals communs 100 bélugas
Mode de déplacement	MAC-R
Enregistrement des données	60 secondes

Tableau-A IV-6 Résultats des expérimentations préliminaires avec l'algorithme MAC-R

	τ_{step}	τ_{limit}	v_{ks}	τ_{ks}	m_{ks}	ψ_{ks}	ME_{obs}	ME_{uni}
Rorqual commun	10	1	0.105	0.842	0.876	0.273	0.955	-4.16
	10	10	0.118	0.589	0.549	0.12	0.768	-0.619
	10	60	0.0419	0.443	0.315	0.208	0.47	0.512
	60	1	0.011	0.837	0.696	0.0918	0.822	-1.71
	60	10	0.0361	0.588	0.479	0.0675	0.731	-0.421
	60	60	0.0267	0.447	0.235	0.143	0.512	0.43
Béluga	10	1	0.0994	0.64	0.669	0.248	0.737	0.197
	10	10	0.0551	0.421	0.382	0.294	0.495	0.729
	10	60	0.0311	0.452	0.419	0.382	0.401	0.81
	60	1	0.0348	0.607	0.494	0.107	0.64	0.399
	60	10	0.0452	0.417	0.276	0.201	0.485	0.732
	60	60	0.039	0.443	0.302	0.273	0.39	0.812

Tableau-A IV-7 Paramètres des expérimentations avec l'algorithme MAC-S (Régression)

Durée de la simulation	30 jours
Pas de temps	60 secondes
Nombre de réplifications	25
Mammifères présents	15-30-60-90-120 et 150 rorquals communs 100-200-400-600-800 et 1000 bélugas
Mode de déplacement	MAC-S
Enregistrement des données	29 ^e et 30 ^e jour Intervalle 60 minutes

Tableau-A IV-8 Résultats des expérimentations avec l'algorithme MAC-S pour le rorqual commun (Régression)

$n \backslash \omega$	0.5	1	1.5	2	3	4	5	10
15	7.73	4.63	2.68	3.07	2.03	1.30	1.15	1.05
30	15.82	9.13	6.53	6.19	4.15	3.22	2.83	2.10
60	30.50	18.86	12.87	13.06	8.63	6.22	5.77	4.22
90	50.66	30.86	21.39	21.63	14.76	10.11	8.82	6.61
120	70.21	45.74	32.25	31.32	19.34	12.62	11.55	8.89
150	87.22	62.85	40.10	34.84	25.80	16.23	14.28	10.88

Tableau-A IV-9 Résultats des expérimentations avec l'algorithme MAC-S pour le béluga (Régression)

$n \backslash \omega$	0.5	1	1.5	2	3	4	5	10
100	33.66	16.88	10.61	9.84	7.87	5.24	4.35	3.59
200	64.40	34.67	18.60	17.38	15.08	11.15	9.13	7.52
400	108.63	68.03	35.55	34.79	29.08	21.54	18.74	15.06
600	171.31	118.81	50.89	47.78	42.62	32.91	28.12	22.46
800	232.39	161.44	63.52	63.67	60.78	42.57	37.48	29.94
1000	249.93	186.62	80.12	78.06	66.68	53.36	46.45	37.60

Tableau-A IV-10 Paramètres des expérimentations avec l'algorithme MAC-S (Rétroaction)

Durée de la simulation	90 jours
Pas de temps	60 secondes
Nombre de réplifications	5
Mammifères présents	35 rorquals communs 100 bélugas
Mode de déplacement	MAC-S
Enregistrement des données	Intervalle 60 minutes

Tableau-A IV-11 Résultats des expérimentations avec l'algorithme MAC-S (Rétroaction)

Batch	Contrôleur	k_c	P_C	$\overline{g_e}$	$\overline{g_{ks}}$	$\overline{g_e}$	$\overline{g_{ks}}$
				commun	commun	beluga	beluga
1	P	0.005		0.2806	0.1197	0.2178	0.2160
2	P	0.01		0.2320	0.1415	0.1353	0.1665
3	P	0.03		0.2297	0.1330	0.0548	0.1390
4	P	0.05		0.2415	0.1537	0.0546	0.1540
5	P	0.10		0.2485	0.1678	0.0433	0.1415
6	PID	0.005	5	0.2308	0.1388	0.3645	0.2245
7	PID	0.01	5	0.2458	0.1479	0.2643	0.1978
8	PID	0.03	5	0.2294	0.1236	0.1905	0.1625
9	PID	0.05	5	0.2342	0.1395	0.0590	0.1440
10	PID	0.10	5	0.2482	0.1623	0.0455	0.1588
11	PID	0.005	10	0.6689	0.2132	0.2158	0.2023
12	PID	0.01	10	0.5054	0.1652	0.3160	0.2055
13	PID	0.03	10	0.2735	0.1237	0.2960	0.2050
14	PID	0.05	10	0.2299	0.1311	0.1563	0.1570
15	PID	0.10	10	0.2575	0.1628	0.1058	0.1658
16	PID	0.005	20	0.8471	0.2795	0.3198	0.2065
17	PID	0.01	20	0.3677	0.1409	0.4618	0.2120
18	PID	0.03	20	0.2390	0.1396	0.1273	0.1888
19	PID	0.05	20	0.2254	0.1425	0.0843	0.1715
20	PID	0.10	20	0.2345	0.1530	0.0826	0.1485

Tableau-A IV-12 Paramètres des expérimentations avec l'algorithme MAC-S
(Rétroaction #2)

Durée de la simulation	365 jours
Pas de temps	60 secondes
Nombre de réplifications	10
Mammifères présents	35 rorquals communs 100 bélugas
Mode de déplacement	MAC-S
Enregistrement des données	Intervalle 60 minutes

Tableau-A IV-13 Résultats des expérimentations avec l'algorithme MAC-S
(Rétroaction #2)

Batch	Contrôleur	k_c	P_C	$\overline{g_e}$ commun	$\overline{g_{ks}}$ commun	$\overline{g_e}$ beluga	$\overline{g_{ks}}$ beluga
1	P	0.03		0.2485	0.152875	0.0635	0.1555
2	PID	0.05	5	0.246	0.152	0.06495	0.146

Tableau-A IV-14 Paramètres des expérimentations avec l'algorithme MAC-S
(Comparaison)

Durée de la simulation	365 jours
Pas de temps	60 secondes
Nombre de réplifications	10
Mammifères présents	35 rorquals communs, 100 bélugas
Mode de déplacement	MAC-S
Enregistrement des données	Intervalle 60 minutes

Tableau-A IV-15 Résultats des expérimentations avec l'algorithme MAC-S
(Comparaison)

	Marée	Taille de groupe	Régression			Rétroaction		
			$\bar{\omega}$	g_e	g_{ks}	$\bar{\omega}$	g_e	g_{ks}
Commun	Flot	8.68	1.4027	0.16	0.114	0.1149	0.201	0.114
	Hautes eaux	12.64	0.8457	0.348	0.189	0.0135	0.36	0.237
	Jusant	4.32	3.5887	0.298	0.136	2.9957	0.244	0.174
	Basses eaux	5.62	2.5184	0.581	0.207	0.5239	0.189	0.0865
Béluga	Flot	8.68	2.7243	0.00173	0.18	2.4906	0.0677	0.151
	Hautes eaux	12.64	2.7243	0.00407	0.181	2.5753	0.0733	0.154
	Jusant	4.32	2.7243	0.00686	0.181	2.7125	0.0594	0.159
	Basses eaux	5.62	2.7243	0.00394	0.18	2.6032	0.0536	0.158

Tableau-A IV-16 Paramètres des expérimentations avec l'algorithme MAC-MMBN

Durée de la simulation	30 jours
Pas de temps	60 secondes
Nombre de réplifications	2
Mammifères présents	35 rorquals communs, 100 bélugas
Mode de déplacement	MAC-MMBN
Enregistrement des données	Intervalle 60 minutes

La table des résultats se lit tel que :

		w_g	
w_u	ψ_{ks}	$\psi_{ks}(commun)$	$\psi_{ks}(beluga)$
	\bar{g}_e	$\bar{g}_e(commun)$	$\bar{g}_e(beluga)$
	ME_{obs}	$ME_{obs}(commun)$	$ME_{obs}(beluga)$
		Score	

$$w_\psi = 1 - w_u - w_g$$

Tableau-A IV-17 Résultats des expérimentations avec l'algorithme MAC-MMBN

$w_u \setminus w_g$		0.20		0.40		0.60		0.80		0.90	
0.02	ψ_{ks}	0.071	0.118	0.074	0.118	0.076	0.122	0.079	0.132	0.089	0.145
	$\overline{g_e}$	0.104	0.104	0.104	0.091	0.088	0.085	0.121	0.072	0.080	0.071
	ME_{obs}	0.944	0.784	0.940	0.731	0.923	0.710	0.908	0.720	0.931	0.737
		0.112		0.119		0.123		0.129		0.119	
0.04	ψ_{ks}	0.071	0.113	0.075	0.121	0.079	0.126	0.082	0.137	0.101	0.151
	$\overline{g_e}$	0.138	0.136	0.136	0.093	0.116	0.097	0.108	0.076	0.109	0.093
	ME_{obs}	0.952	0.788	0.941	0.780	0.942	0.772	0.916	0.803	0.902	0.773
		0.120		0.117		0.117		0.114		0.130	
0.06	ψ_{ks}	0.070	0.115	0.075	0.118	0.079	0.124	0.094	0.134	0.128	0.171
	$\overline{g_e}$	0.183	0.157	0.126	0.113	0.119	0.101	0.085	0.098	0.103	0.079
	ME_{obs}	0.960	0.842	0.950	0.797	0.937	0.793	0.939	0.808	0.918	0.829
		0.121		0.114		0.116		0.110		0.122	
0.08	ψ_{ks}	0.070	0.113	0.076	0.118	0.080	0.126	0.098	0.140	0.149	0.211
	$\overline{g_e}$	0.217	0.180	0.142	0.118	0.118	0.110	0.091	0.112	0.107	0.068
	ME_{obs}	0.956	0.850	0.940	0.851	0.938	0.844	0.943	0.777	0.912	0.806
		0.129		0.110		0.109		0.120		0.136	
0.10	ψ_{ks}	0.072	0.112	0.077	0.118	0.084	0.127	0.108	0.144	0.256	0.396
	$\overline{g_e}$	0.281	0.202	0.170	0.136	0.116	0.104	0.104	0.090	0.124	0.090
	ME_{obs}	0.960	0.843	0.950	0.852	0.934	0.816	0.945	0.725	0.928	0.815
		0.144		0.117		0.114		0.129		0.187	

ANNEXE V

SCRIPTS D'EXTRACTION DES PATRONS SIMULÉS

Script_path.sh

```
#!/bin/bash
cat $1 | sort.sh | simplify.awk | path.awk | species.awk
```

Sort.sh

```
sort -n -t, -k1,1n -k2,2n -k5,5n -k3,3n -S100048
```

simplify.awk

```
gawk '
BEGIN {
    distance_limit = 5;
    FS=",";

    #Columns index
    iID=5;
    iX=6;
    iY=7;

    #ID of the current path
    current_ID=-999;
}
{
    if (NR==1) {
        print $0;
    }
    else if (NR==2) {
        #Premiere Ligne
        print $0;
        current_ID=$iID;
        current_x=$iX;
        current_y=$iY;
    }
    else if ($iID!=current_ID) {
        print buffer[NR-1];

        delete buffer;
    }
}
```

```

delete xBuffer;
delete yBuffer;
for (i in buffer) {
    delete buffer[i];
    delete xBuffer[i];
    delete yBuffer[i];
}

print $0;
current_ID=$iID;
current_x=$iX;
current_y=$iY;
}

else {

    distance_max=0;
    for (i in buffer) {

        #http://www.allegro.cc/forums/thread/589720
        x=xBuffer[i];
        y=yBuffer[i];
        x1=current_x;
        y1=current_y;
        x2=$iX;
        y2=$iY;
        distance = DistanceFromLine2(x,y,x1,y1,x2,y2);
        #print distance
        if (distance > distance_max) {
            distance_max=distance;
        }

        #Petite simplification pour reduire le temps de traitement...
        if ( ( distance < 1 ) && ( i < NR-20 ) ) {
            #print "Point " i " deleted!"
            delete buffer[i];
            delete xBuffer[i];
            delete yBuffer[i];
        }
    }

    if (distance_max > distance_limit) {
        #print "Distance maximale" distance_max >> "simplify.csv";
        print buffer[NR-1];
        current_x=xBuffer[NR-1];
        current_y=yBuffer[NR-1];

        delete buffer;
        delete xBuffer;
        delete yBuffer;
        for (i in buffer) {
            delete buffer[i];
            delete xBuffer[i];
            delete yBuffer[i];
        }
    }
}

```

```

    }

    #Insertion du point comme nouveau point intermediaire
    buffer[NR]=$0;
    xBuffer[NR]=$iX;
    yBuffer[NR]=$iY;
}
}

#http://www.vb-helper.com/howto_distance_point_to_line.html
#Calculate the distance between the point and the segment.
function DistanceFromLine2(px, py, X1, Y1, X2, Y2, dx, dy, t)
{
    dx = X2 - X1;
    dy = Y2 - Y1;
    if ( ( dx == 0 ) && ( dy == 0 ) ) {
        # Its a point not a line segment.
        dx= px - X1;
        dy= py - Y1;
        return sqrt(dx * dx + dy * dy);
    }

    # Calculate the t that minimizes the distance.
    t = ((px - X1) * dx + (py - Y1) * dy) / (dx * dx + dy * dy);

    #See if this represents one of the segment
    # end points or a point in the middle.
    if (t < 0) {
        dx = px - X1;
        dy = py - Y1;
    }
    else if (t > 1) {
        dx = px - X2;
        dy = py - Y2;
    }
    else {
        dx = px - (X1 + t * dx);
        dy = py - (Y1 + t * dy);
    }

    return sqrt(dx * dx + dy * dy);
}

# http://www.codeguru.com/forum/showthread.php?t=194400
# Compute distance between point C and the line AB
function DistanceFromLine(cx, cy, ax, ay, bx, by, r_numerator,
r_denominator, r, px, py, s, distanceLine, xx, yy, distanceSegment, dist1,
dist2)
{
    r_numerator = (cx-ax)*(bx-ax) + (cy-ay)*(by-ay);
    r_denominator = (bx-ax)*(bx-ax) + (by-ay)*(by-ay);
}

```



```

r = r_numerator / r_denominator;

px = ax + r*(bx-ax);
py = ay + r*(by-ay);
s = ((ay-cy)*(bx-ax) - (ax-cx)*(by-ay)) / r_denominator;
if (s<0) {s=-s;}
distanceLine = s*sqrt(r_denominator);

# (xx,yy) is the point on the lineSegment closest to (cx,cy)
xx = px;
yy = py;

if ( ( r >= 0 ) && ( r <= 1 ) )
{
    distanceSegment = distanceLine;
}
else
{

    dist1 = (cx-ax)*(cx-ax) + (cy-ay)*(cy-ay);
    dist2 = (cx-bx)*(cx-bx) + (cy-by)*(cy-by);
    if (dist1 < dist2)
    {
        xx = ax;
        yy = ay;
        distanceSegment = sqrt(dist1);
    }
    else
    {
        xx = bx;
        yy = by;
        distanceSegment = sqrt(dist2);
    }

}

return distanceSegment;
}

END {}'
```

Path.awk

```

gawk '
BEGIN {
    FS=",";
    pi=3.141592653589793

    #TimeStep_Duration (second)
    timestep = 60

    #Columns index
    iTick=3
    iID=5;
    iX=6;
    iY=7;

    #ID of the current Agent
    current_ID=-999;

    #Previous Data...
    prev_2_x=-999;
    prev_2_y=-999;
    prev_1_x=-999;
    prev_1_y=-999;
    prev_1_t=-999;
}

{
    if (NR==1) {
        print $0 " ,\"Duration\", \"speed\", \"Length\", \"Angle\", \"Psi\"";
    }

    else if ($iID!=current_ID) {
        current_ID=$iID;
        prev_2_x=-999;
        prev_2_y=-999;
        prev_1_x=-999;
        prev_1_y=-999;
        prev_1_t=-999;
    }

    else if (prev_2_x!=-999) {
        duration_s = ($iTick-prev_1_t)*timestep; #seconds
        duration_h = duration_s/3600; #hours
        m1 = sqrt(($iX-prev_1_x)^2+($iY-prev_1_y)^2); #meter
        speed = m1/duration_s;

        b=sqrt((prev_2_x-prev_1_x)^2+(prev_2_y-prev_1_y)^2);
        c=m1;
        a=sqrt(($iX-prev_2_x)^2+($iY-prev_2_y)^2);
        psi = ( a^2 - b^2 - c^2) / (-2 * b * c);
        angle = pi-acos(psi); #radian
    }
}

```

```

    psi=cos(angle);

    #Print the result
    print $0 "," duration_s "," speed "," m1 "," angle "," psi;
}

prev_2_x = prev_1_x;
prev_2_y = prev_1_y;
prev_1_x = $iX;
prev_1_y = $iY;
prev_1_t = $iTick;
}

# http://www.gladir.com/CODER/AWK/acos.htm
function abs(myvar) {
    if(myvar<0) myvar=-myvar;
    return myvar;
}

function acos(myvar, pi, tmp) {
    pi=3.141592653589793
    if(abs(myvar)==1) {
        return (1-myvar)*pi/2
    } else {
        tmp=myvar*myvar;
        if (tmp>1) {tmp=1;}
        return atan2(-myvar,sqrt(1-tmp))+2*atan2(0.5,0.5)
    }
}
}'

```

Species.awk

```

gawk '
BEGIN {
    FS=",";

    #Index AgentType
    iReplication=2;
    iType=4;
    iGroupWhale=8;
    iTide=9;
    iDuration=10;
    iSpeed=11;
    iLength=12;
    iAngle=13;
    iPsi=14;

    #Make directory
    system("rm -rf path_102")
    system("rm -rf path_105")
    system("mkdir path_102")
    system("mkdir path_105")
}

```

```

}
{
  if (NR==1) {
    print $0 >> "path_102/path.csv";
    print $0 >> "path_105/path.csv";
  }

  else {
    path = "path_" $iType "/path.csv";
    path_groupwhale = "path_" $iType "/path_groupwhale.csv";
    path_tide = "path_" $iType "/path_groupwhale_tide_"
$iTide ".csv";
    path_tide_replication = "path_" $iType "/path_groupwhale_tide_"
$iTide "_replication.csv";
    path_duration = "path_" $iType "/path_duration.csv";
    path_speed = "path_" $iType "/path_speed.csv";
    path_length = "path_" $iType "/path_length.csv";
    path_angle = "path_" $iType "/path_angle.csv";
    path_psi = "path_" $iType "/path_psi.csv";
    path_replication = "path_" $iType "/path_replication.csv";

    print $0 >> path;
    print $iGroupWhale+1 >> path_groupwhale;
    print $iGroupWhale+1 >> path_tide;
    print $iDuration >> path_duration;
    print $iSpeed >> path_speed;
    print $iLength >> path_length;
    print $iAngle >> path_angle;
    print $iPsi >> path_psi;
    print $iReplication >> path_replication;
    print $iReplication >> path_tide_replication;
  }
}
END {}'

```

ANNEXE VI

SCRIPTS D'EXTRACTION DU PATRON DE RÉPARTITION

Script_ef.sh

```
#!/bin/bash
echo 'Set environment variables...'
export PATH=/opt/local/bin:/opt/local/sbin:$PATH
path_z='Z:$PWD'
beluga_obs='"C:\Documents and Settings\Philippe\My
Documents\Arcgis\Raster\repartitionAOM\beluga\beluga_norm"'
beluga_flat='"C:\Documents and Settings\Philippe\My
Documents\Arcgis\Raster\repartitionAOM\beluga\sighap_norm"'

commun_obs='"C:\Documents and Settings\Philippe\My
Documents\Arcgis\Raster\repartitionAOM\commun\commun_norm"'
commun_flat='"C:\Documents and Settings\Philippe\My
Documents\Arcgis\Raster\repartitionAOM\commun\sighap_norm"'

echo 'Extracting density map...'
cat $1 | density.awk

#Compute fin whale EF
echo 'Computing model efficiency for fin whale...'
compute_ef.sh $path_z/path_102/model_ef.txt"
$path_z/path_102/density.txt" "$commun_obs" "$commun_flat" 0
sleep 30

#Compute beluga EF
echo 'Computing model efficiency for beluga...'
compute_ef.sh $path_z/path_105/model_ef.txt"
$path_z/path_105/density.txt" "$beluga_obs" "$beluga_flat" 1
sleep 15

#Clear all terminal console
sleep 5
kill -TERM `ps -ef | grep 'Windows Batch' | grep -v grep | awk '0'`
```

density.awk

```

gawk '
BEGIN {
    FS=",";

    #Index AgentType
    iType=4;
    iX=6;
    iY=7;

    total_102=0;
    total_105=0;
    for(x=0;x<2205; x++) {
        for(y=0;y<1983; y++) {
            densite_102[x, y] = 0;
            densite_105[x, y] = 0;
        }
    }

}

{

    x=int(($iX-370582)/100);
    y=int(($iY-5237060)/100);

    if ($iType==102) {
        densite_102[x,y]=densite_102[x,y]+1;
        total_102=total_102+1;
    }
    else if ($iType==105) {
        densite_105[x,y]=densite_105[x,y]+1;
        total_105=total_105+1;
    }
}

END {

    if (total_102 > 0) {
        #total_102=total_102/100000.0;

        print "ncols          2205"      >> "path_102/density.txt";
        print "nrows          1983"      >> "path_102/density.txt";
        print "xllcorner      370582"    >> "path_102/density.txt";
        print "yllcorner      5237060"    >> "path_102/density.txt";
        print "cellsize       100"        >> "path_102/density.txt";
        print "NODATA_value  0.000000">> "path_102/density.txt";

        for(y=1982;y>=0; y--) {
            for(x=0;x<2205; x++) {
                printf("%f ",densite_102[x,y]/total_102) >>
"path_102/density.txt";
            }
            printf("\n") >> "path_102/density.txt";
        }
    }
}

```

```

    }
}

if (total_105 > 0) {
  #total_105=total_105/100000.0;

  print "ncols          2205"    >> "path_105/density.txt";
  print "nrows          1983"    >> "path_105/density.txt";
  print "xllcorner      370582"  >> "path_105/density.txt";
  print "yllcorner      5237060" >> "path_105/density.txt";
  print "cellsize       100"     >> "path_105/density.txt";
  print "NODATA_value   0.000000">> "path_105/density.txt";

  for(y=1982;y>=0; y--) {
    for(x=0;x<2205; x++) {
      printf("%f ",densite_105[x,y]/total_105) >>
"path_105/density.txt";
    }
    printf("\n") >> "path_105/density.txt";
  }
}
}'

```

Compute_ef.sh

```

#!/bin/bash
# Usage ./compute_ef.sh <output> <map_simulated_ascii> <map_obs_raster>
<unique_id>
commande='c:\Python24\python "C:\Documents and Settings\Philippe\My
Documents\Arcgis\ArcToolBox\calcul_ef.py"'
map_bool="'C:\Documents and Settings\Philippe\My
Documents\Arcgis\Raster\mnt_bin\mnt_bool"'

path_output=$1
map_sim=$2
map_obs=$3
map_flat=$4
unique_id=$5

rm /Users/philippe/Documents/Parallels/My\ Boot\ Camp\Windows\
Disks/C/batch_105.bat
echo $commande $path_output $map_sim $map_obs $map_flat $map_bool
$unique_id >> /Users/philippe/Documents/Parallels/My\ Boot\ Camp\Windows\
Disks/C/batch_105.bat
open -n /Users/philippe/Documents/Parallels/My\ Boot\ Camp\Windows\
Applications\Windows\ Batch_105.app/

```

calcul_ef.py (ESRI Python ArcScript)

```

# -----
# calcul_ef.py
# Created on: Wed Oct 22 2008 03:07:56 PM
#   (generated by ArcGIS/ModelBuilder)
# -----
# Import system modules
import sys, string, os, arcgisscripting

# Create the Geoprocessor object
gp = arcgisscripting.create()

# Check out any necessary licenses
gp.CheckOutExtension("spatial")

gp.OverwriteOutput = 1    # setup for overwrites

# Load required toolboxes...
gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Spatial
Analyst Tools.tbx")
gp.AddToolbox("C:/Documents and Settings/Philippe/My
Documents/Arcgis/ArcToolBox/Utilities.tbx")
gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Conversion
Tools.tbx")
gp.AddToolbox("C:/Documents and Settings/Philippe/My
Documents/Arcgis/ArcToolBox/Utilities.tbx")

# Script arguments...
file_output  = sys.argv[1]
raster_ascii = sys.argv[2]
density_obs  = sys.argv[3]
density_flat = sys.argv[4]
mnt_bool     = sys.argv[5]
unique_id    = sys.argv[6]

# Local variables...
string_id    = "_" + str(unique_id)
raster_brut  = "C:\\tmp_script\\txt2raster" + string_id
raster_big   = "C:\\tmp_script\\big_raster" + string_id
points_grid  = "C:\\tmp_script\\pointsGrid" + string_id + ".shp"
Kernel_raster= "C:\\tmp_script\\rasterkern" + string_id
kernel_clip  = "C:\\tmp_script\\kernelClip" + string_id
raster_norm  = "C:\\tmp_script\\rasterNorm" + string_id
zero = "0"
Population_field = "GRID_CODE"

# Process: ASCII to Raster...
gp.AddMessage("Executing ascii to raster conversion...")
gp.ASCIIToRaster_conversion(raster_ascii, raster_brut, "FLOAT")

# Process: convert decimal to int
gp.AddMessage("Executing raster normalisation...")
gp.Divide_sa(raster_brut, "0.00000001", raster_big)

```



```

# Process: Raster to Point...
gp.AddMessage("Executing conversion raster to points...")
gp.RasterToPoint_conversion(raster_big, points_grid)

# Process: Kernel Density...
gp.AddMessage("Executing kernel density...")
tempEnvironment0 = gp.extent
gp.extent = raster_brut
gp.KernelDensity_sa(points_grid, Population_field, Kernel_raster, "100",
"6606", "SQUARE_MAP_UNITS")
#gp.extent = tempEnvironment0

# Process: Con...
gp.AddMessage("Executing conditional clip...")
gp.Con_sa(mnt_bool, Kernel_raster, kernel_clip, zero, "")

# Process: Normalize...
gp.AddMessage("Executing raster normalisation...")
gp.Normalize(kernel_clip, raster_norm)

#-----
# BEGIN Process: EF versus observed...
#-----

gp.AddMessage("Computing model efficiency versus simulated data...")
observed = density_obs
expected = raster_norm

#(O-E)
moe = "C:\\tmp_script\\ef_temp1"+ string_id

#(O-E)^2
soe = "C:\\tmp_script\\ef_temp2"+ string_id

#(O-O)
moo = "C:\\tmp_script\\ef_temp3"+ string_id

#(O-O)^2
soo = "C:\\tmp_script\\ef_temp4"+ string_id

gp.Minus_sa(observed, expected, moe)
gp.Power(moe,2,soe)

observed_mean = gp.GetRasterProperties(observed, "MEAN")
gp.Minus_sa(observed, observed_mean, moo)
gp.Power(moo,2,soo)

soe_mean = gp.GetRasterProperties(soe, "MEAN")
soo_mean = gp.GetRasterProperties(soo, "MEAN")

if soo_mean==0:
    gp.AddMessage("Divides by zero, cannot compute model efficiency")
    ef_vs_obs=0
else:
    ef_vs_obs = 1 - (soe_mean / soo_mean)

```

```

gp.AddMessage("Model efficiency simulated = " + str(ef_vs_obs))

#-----
# END Process: EF versus observed...
#-----

#-----
# BEGIN Process: EF versus flat...
#-----

gp.AddMessage("Computing model efficiency versus uniform data...")
observed = density_flat
expected = raster_norm

#(O-E)
moe = "C:\\tmp_script\\ef_temp1"+ string_id

#(O-E)^2
soe = "C:\\tmp_script\\ef_temp2"+ string_id

#(O-O)
moo = "C:\\tmp_script\\ef_temp3"+ string_id

#(O-O)^2
soo = "C:\\tmp_script\\ef_temp4"+ string_id

gp.Minus_sa(observed, expected, moe)
gp.Power(moe,2,soe)

observed_mean = gp.GetRasterProperties(observed, "MEAN")
gp.Minus_sa(observed, observed_mean, moo)
gp.Power(moo,2,soo)

soe_mean = gp.GetRasterProperties(soe, "MEAN")
soo_mean = gp.GetRasterProperties(soo, "MEAN")

if soo_mean==0:
    gp.AddMessage("Divides by zero, cannot compute model efficiency")
    ef_vs_flat=0
else:
    ef_vs_flat = 1 - (soe_mean / soo_mean)
    gp.AddMessage("Model efficiency uniform = " + str(ef_vs_flat))

#-----
# END Process: EF versus observed...
#-----

gp.AddMessage("Writing model efficiency output...")
fileHandle = open(file_output,"a")
fileHandle.write("%10.6f \n" % ef_vs_obs)
fileHandle.write("%10.6f \n" % ef_vs_flat)
fileHandle.close()
gp.AddMessage("End of execution : Calcul_ef.py")

```

ANNEXE VII

SCRIPTS DE COMPILATION DES PATRONS SIMULÉS DANS R

Model_result.R

```
compute_ks <- function(table_replication, path_sim, path_obs, name) {

  #Load Table
  table_sim <- read.table(file= path_sim,head=FALSE) [,1];
  table_obs <- read.table(file= path_obs,head=FALSE) [,1];

  #compute statistics
  mean_replication = mean(table_sim)
  ks_replication   = ks.test(table_sim, table_obs)$statistic;
  ne_replication   =abs((mean(table_sim) -
mean(table_obs))/mean(table_obs));

  if(max(table_replication)-min(table_replication)>0) {

    mean_replication=c()
    ks_replication=c()
    ne_replication=c()

    #Compute mean for each replication
    mean_replication <- tapply(table_sim, table_replication, mean);
    mean_ttest <- t.test(mean_replication)
    mean_ic <- (mean_ttest$conf.int [2]-mean_ttest$conf.int [1])/2;

    #Compute KS for each replication
    ks_replication <- tapply(table_sim, table_replication,
function(z) ks.test(z, table_obs)$statistic);
    ks_ttest <- t.test(ks_replication)
    ks_ic <- (ks_ttest$conf.int [2]-ks_ttest$conf.int [1])/2;

    #Compute NE for each replication
    ne_replication <- tapply(table_sim, table_replication,
function(z) abs((mean(z)-mean(table_obs))/mean(table_obs)));
    ne_ttest <- t.test(ne_replication);
    ne_ic <- (ne_ttest$conf.int [2]-ne_ttest$conf.int [1])/2;
  } else {
    mean_ic=0;
    ks_ic=0;
    ne_ic=0;
  }

  #Create array output
  cat("\n", file= output_array, append=TRUE, sep="")
  cat(name, "\t ", file= output_array, append=TRUE, sep="")
}
```

```

    cat(format(mean(mean_replication), digits=10), "\t ", file=
output_array, append=TRUE, sep="")
    cat(format(mean(ne_replication), digits=10), "\t ", file=
output_array, append=TRUE, sep="")
    cat(format(mean(ks_replication), digits=10), file= output_array,
append=TRUE, sep="")
    #star <- if (table_ks["p.value"] < 0.50) "" else "*";
    #cat(star, "\t ", file= output_array, append=TRUE, sep="")
    cat("\t", file= output_array, append=TRUE, sep="")

    cat(" 0 \t", file= output_array, append=TRUE, sep="")
    cat(format(mean_ic, digits=10), "\t ", file= output_array,
append=TRUE, sep="")
    cat(format(ne_ic, digits=10), "\t ", file= output_array, append=TRUE,
sep="")
    cat(format(ks_ic, digits=10), "\t ", file= output_array, append=TRUE,
sep="")

    #Free memory and return result
    rm(table_sim);
    rm(table_obs);
    return(c(mean_replication, ks_replication, ne_replication));
}

compute_group_stat <- function(table_all, name) {

    dim(table_all)=c(length(table_all)/12,12)
    mean_table=unlist(c(table_all[,c(1,4,7,10)]));
    ks_table =unlist(c(table_all[,c(2,5,8,11)]));
    ne_table =unlist(c(table_all[,c(3,6,9,12)]));

    mean_mean <- mean(mean_table);
    mean_ttest <- t.test(mean_table);
    mean_ic <- (mean_ttest$conf.int[2]-mean_ttest$conf.int[1])/2;

    ks_mean <- mean(ks_table);
    ks_ttest <- t.test(ks_table);
    ks_ic <- (ks_ttest$conf.int[2]-ks_ttest$conf.int[1])/2;

    ne_mean <- mean(ne_table);
    ne_ttest <- t.test(ne_table);
    ne_ic <- (ne_ttest$conf.int[2]-ne_ttest$conf.int[1])/2;

    #Create array output
    cat("\n", file= output_array, append=TRUE, sep="")
    cat(name, "\t ", file= output_array, append=TRUE, sep="")
    cat(format(mean_mean, digits=10), "\t ", file= output_array,
append=TRUE, sep="")
    cat(format(ne_mean, digits=10), "\t ", file= output_array,
append=TRUE, sep="")
    cat(format(ks_mean, digits=10), file= output_array, append=TRUE,
sep="")
    cat("\t", file= output_array, append=TRUE, sep="")

```

```

        cat(" 0 \t", file= output_array, append=TRUE, sep="")
        cat(format(mean_ic, digits=10), "\t ", file= output_array,
append=TRUE, sep="")
        cat(format(ne_ic, digits=10), "\t ", file= output_array, append=TRUE,
sep="")
        cat(format(ks_ic, digits=10), "\t ", file= output_array, append=TRUE,
sep="")

        return(c())
}

#####
SPECIES=102
#####

#Write the output header
output_array<-"path_102/model_result.txt";
cat("\n\nVariable \t Moy. \t Err. \t KS. \t EF. \tMoy.ic\t Err.ic\t
KS.ic", file= output_array, append=TRUE, sep="")

#Compute path patterns
table_replication <-
read.table(file="path_102/path_replication.csv",head=FALSE) [,1];
speed_ks <-compute_ks(table_replication, "path_102/path_speed.csv",
"/Users/philippe/Documents/scripts/R/102/speed.txt", "Vitesse ")
duration_ks<-compute_ks(table_replication,
"path_102/path_duration.csv", "/Users/philippe/Documents/scripts/R/102/time
.txt", "Duree ")
length_ks <-compute_ks(table_replication, "path_102/path_length.csv",
"/Users/philippe/Documents/scripts/R/102/dist.txt", "Distance ")
psi_ks <-compute_ks(table_replication, "path_102/path_psi.csv",
"/Users/philippe/Documents/scripts/R/102/psi.txt", "Angle braquage")

#Compute group patterns
#group_ks <-compute_ks(table_replication, "path_102/path_groupwhale.csv",
"/Users/philippe/Documents/scripts/R/102/group_all.txt", "Groupe
")

table_replication <-
read.table(file="path_102/path_groupwhale_tide_0_replication.csv",head=FAL
SE) [,1];
group_0_ks<-compute_ks(table_replication,
"path_102/path_groupwhale_tide_0.csv", "/Users/philippe/Documents/scripts/R
/102/group_tide_0.txt", "Groupe_maree_0")

table_replication <-
read.table(file="path_102/path_groupwhale_tide_1_replication.csv",head=FAL
SE) [,1];
group_1_ks<-compute_ks(table_replication,
"path_102/path_groupwhale_tide_1.csv", "/Users/philippe/Documents/scripts/R
/102/group_tide_1.txt", "Groupe_maree_1")

```

```

table_replication <-
read.table(file="path_102/path_groupwhale_tide_2_replication.csv",head=FALSE) [,1];
group_2_ks<-compute_ks(table_replication,
"path_102/path_groupwhale_tide_2.csv","/Users/philippe/Documents/scripts/R
/102/group_tide_2.txt", "Groupe_maree_2")

table_replication <-
read.table(file="path_102/path_groupwhale_tide_3_replication.csv",head=FALSE) [,1];
group_3_ks<-compute_ks(table_replication,
"path_102/path_groupwhale_tide_3.csv","/Users/philippe/Documents/scripts/R
/102/group_tide_3.txt", "Groupe_maree_3")

compute_group_stat(c(group_0_ks, group_1_ks, group_2_ks,
group_3_ks), "Groupe          ");

#Get and write the repartition pattern
if(file.exists("path_102/model_ef.txt")) {
  table_density <-
read.table(file="path_102/model_ef.txt",head=FALSE) [,1];
  cat("\nObserved density \t\t\t", format(table_density[1], digits=10),
file=output_array, append=TRUE, sep="")
  cat(" \nUniform density \t\t\t", format(table_density[2], digits=10),
file=output_array, append=TRUE, sep="")
} else {
  cat("\nObserved density \t\t\t 0", file=output_array, append=TRUE,
sep="")
  cat(" \nUniform density \t\t\t 0", file=output_array, append=TRUE,
sep="")
}

#####
SPECIES=105
#####

#Write the output header
output_array<-"path_105/model_result.txt";
cat("\n\nVariable \t Moy. \t Err. \t KS. \t EF. \tMoy.ic\t Err.ic\t
KS.ic", file= output_array, append=TRUE, sep="")

#Compute path patterns
table_replication <-
read.table(file="path_105/path_replication.csv",head=FALSE) [,1];
speed_ks <-compute_ks(table_replication, "path_105/path_speed.csv",
"/Users/philippe/Documents/scripts/R/105/speed.txt", "Vitesse ")
duration_ks<-compute_ks(table_replication,
"path_105/path_duration.csv","/Users/philippe/Documents/scripts/R/105/time
.txt", "Duree ")
length_ks <-compute_ks(table_replication, "path_105/path_length.csv",
"/Users/philippe/Documents/scripts/R/105/dist.txt", "Distance")
psi_ks <-compute_ks(table_replication, "path_105/path_psi.csv",
"/Users/philippe/Documents/scripts/R/105/psi.txt", "Angle braquage")

```

```

#Compute group patterns
#group_ks <-compute_ks(table_replication, "path_105/path_groupwhale.csv",
"/Users/philippe/Documents/scripts/R/105/group_all.txt", "Groupe      ")

table_replication <-
read.table(file="path_105/path_groupwhale_tide_0_replication.csv",head=FALSE) [,1];
group_0_ks<-compute_ks(table_replication,
"path_105/path_groupwhale_tide_0.csv","/Users/philippe/Documents/scripts/R
/105/group_all.txt", "Groupe_maree_0")

table_replication <-
read.table(file="path_105/path_groupwhale_tide_1_replication.csv",head=FALSE) [,1];
group_1_ks<-compute_ks(table_replication,
"path_105/path_groupwhale_tide_1.csv","/Users/philippe/Documents/scripts/R
/105/group_all.txt", "Groupe_maree_1")

table_replication <-
read.table(file="path_105/path_groupwhale_tide_2_replication.csv",head=FALSE) [,1];
group_2_ks<-compute_ks(table_replication,
"path_105/path_groupwhale_tide_2.csv","/Users/philippe/Documents/scripts/R
/105/group_all.txt", "Groupe_maree_2")

table_replication <-
read.table(file="path_105/path_groupwhale_tide_3_replication.csv",head=FALSE) [,1];
group_3_ks<-compute_ks(table_replication,
"path_105/path_groupwhale_tide_3.csv","/Users/philippe/Documents/scripts/R
/105/group_all.txt", "Groupe_maree_3")

compute_group_stat(c(group_0_ks, group_1_ks, group_2_ks,
group_3_ks),"Groupe      ");

#Get and write the repartition pattern
if(file.exists("path_105/model_ef.txt")) {
  table_density <-
read.table(file="path_105/model_ef.txt",head=FALSE) [,1];
  cat("\nObserved density \t\t\t", format(table_density[1], digits=10),
file=output_array, append=TRUE, sep="")
  cat(" \nUniform density \t\t\t", format(table_density[2], digits=10),
file=output_array, append=TRUE, sep="")
} else {
  cat("\nObserved density \t\t\t 0", file=output_array, append=TRUE,
sep="")
  cat(" \nUniform density \t\t\t 0", file=output_array, append=TRUE,
sep="")
}

```

BIBLIOGRAPHIE

- Alderman, J., et S. A. Hinsley. 2007. « Modelling the third dimension: Incorporating topography into the movement rules of an individual-based spatially explicit population model ». *Ecological Complexity*, vol. 4, n° 4, p. 169-181.
- Anwar, S. M., C. A. Jeanneret, L. Parrott et D. J. Marceau. 2007. « Conceptualization and implementation of a multi-agent model to simulate whale-watching tours in the St. Lawrence Estuary in Quebec, Canada ». *Environmental Modelling and Software*, vol. 22, n° 12, p. 1775-1787.
- Bennett, D. A., et W. Tang. 2006. « Modelling adaptive, spatially aware, and mobile agents: Elk migration in Yellowstone ». *International Journal of Geographical Information Science*, vol. 20, n° 9, p. 1039-1066.
- Bergman, C. M., J. A. Schaefer et S. N. Luttich. 2000. « Caribou movement as a correlated random walk ». *Oecologia*, vol. 123, n° 3, p. 364-374.
- Christensen, V, et CJ Walters. 2004. « Ecopath with Ecosim: methods, capabilities and limitations ». *Ecological Modelling*, vol. 172, n° 2-4, p. 109-139.
- Codling, EA, MJ Plank et S Benhamou. 2008. « Random walk models in biology ». *Journal of The Royal Society Interface*.
- Coquillard, Patrick, et David R. C. Hill. 1997. *Modélisation et simulation d'écosystèmes : des modèles déterministes aux simulations à évènements discrets*. Coll. « Ecologie ». Paris: Masson, xx, 273 p.
- Couzin, I. 2007. « Connections Collective minds ». *Nature*, vol. 445, p. 715.
- Couzin, I. D., J. Krause, N. R. Franks et S. A. Levin. 2005. « Effective leadership and decision-making in animal groups on the move ». *Nature*, vol. 433, n° 7025, p. 513-6.
- Couzin, I. D., J. Krause, R. James, G. D. Ruxton et N. R. Franks. 2002. « Collective Memory and Spatial Sorting in Animal Groups ». *Journal of Theoretical Biology*, vol. 218, n° 1, p. 1-11.
- DeAngelis, Donald L. 2001. « Individual-Based Models: Tracking Variability Among Individuals ». *Modeling in Natural Resource Management: Development, Interpretation, and Application*.
- Efron, B, et R Tibshirani. 1986. « Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy ». *Statistical science*, p. 54-75.

- Forester, J. D., A. R. Ives, M. G. Turner, D. P. Anderson, D. Fortin, H. L. Beyer, D. W. Smith et M. S. Boyce. 2007. « STATE–SPACE MODELS LINK ELK MOVEMENT PATTERNS TO LANDSCAPE CHARACTERISTICS IN YELLOWSTONE NATIONAL PARK ». *Ecological Monographs*, vol. 77, n° 2, p. 285-299.
- Gardner, M. 1970. « Mathematical games: The fantastic combinations of John Conway's new solitaire game 'Life' ». *Scientific American*, vol. 223, n° 4, p. 120-123.
- Gimblett, H. R., M. T. Richards et R. M. Itami. 1997. « A complex systems approach to simulating human behaviour using synthetic landscapes ». *Complexity International*, vol. 6.
- Grimm, V, et SF Railsback. 2005. *Individual-based modelling and ecology*. Princeton University Press.
- Grimm, Volker, Eloy Revilla, Uta Berger, Florian Jeltsch, Wolf M. Mooij, Steven F. Railsback, Hans-Hermann Thulke, Jacob Weiner, Thorsten Wiegand et Donald L. DeAngelis. 2005. « Pattern-Oriented Modeling of Agent-Based Complex Systems: Lessons from Ecology ». *Science*, vol. 310, n° 5750 (November 11, 2005), p. 987-991.
- Herrera, F., M. Lozano et J. L. Verdegay. 1998. « Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis ». *Artif. Intell. Rev.*, vol. 12, n° 4, p. 265-319.
- Huse, Geir, et Jarl Giske. 1998. « Ecology in Mare Pentium: an individual-based spatio-temporal model for fish with adapted behaviour ». *Fisheries Research*, vol. 37, n° 1-3, p. 163-178.
- Judson, Olivia P. 1994. « The rise of the individual-based model in ecology ». *Trends in ecology & evolution*, vol. 9, n° 1, p. 9-14.
- Kavanagh, P, N Newlands, V Christensen et D Pauly. 2004. « Automated parameter optimization for Ecopath ecosystem models ». *Ecological Modelling*, vol. 172, n° 2-4, p. 141-149.
- Le Larousse des noms communs*. 2008. Paris: Larousse, 1510 p. p.
- Little, L. R., S. Kuikka, A. E. Punt, F. Pantus, C. R. Davies et B. D. Mapstone. 2004. « Information flow among fishing vessels modelled using a Bayesian network ». *Environmental Modelling and Software*, vol. 19, n° 1, p. 27-34.
- Michaud, Robert. « Suivis VHF des bélugas dans l'estuaire du Saint-Laurent ». Non publié.

- Michaud, Robert, Véronik de la Chenelière, Marie-Hélène D'Arcy et Michel Moisan. 2008. *Les activités d'observation en mer (AOM) dans l'estuaire du Saint-Laurent : Zone de protection marine estuaire du Saint-Laurent et Parc Marin Saguenay-Saint-Laurent (Rapport annuel)*.
- Michaud, Robert, et J. Girard. 1997a. « Les rorquals communs et les activités d'observation en mer dans l'estuaire du Saint-Laurent entre 1994 et 1996 : 1) Étude de l'utilisation du territoire et évaluation de l'exposition aux activités d'observation à l'aide de la télémétrie VHF, Projet réalisé dans le cadre d'une entente d'entreprise conjointe (Groupe de recherche et d'éducation sur le milieu marin (GREMM), ministère de l'Environnement et Faune du Québec, ministère des Pêches et Océans, ministère du Patrimoine canadien, Parcs Canada). ». ministère de l'Environnement et Faune du Québec, ministère des Pêches et Océans, ministère du Patrimoine canadien, Parcs Canada, 30 p. + annexes.
- Michaud, Robert, et J. Girard. 1997b. « Les rorquals communs et les activités d'observation en mer dans l'estuaire du Saint-Laurent entre 1994 et 1996 : 2. Évaluation de l'impact des activités d'observation en mer sur le comportement des rorquals commun, Projet réalisé dans le cadre d'une entente d'entreprise conjointe (Groupe de recherche et d'éducation sur le milieu marin (GREMM), ministère de l'Environnement et Faune du Québec, ministère des Pêches et Océans, ministère du Patrimoine canadien, Parcs Canada). ». ministère de l'Environnement et Faune du Québec, ministère des Pêches et Océans, ministère du Patrimoine canadien, Parcs Canada, 24 p. + annexes.
- Ministères des Pêches et des Océans. 2004. « Système d'information pour la gestion de l'habitat de poisson (SIGHAP) ». Ministère des Pêches et des Océans, Institut Maurice-Lamontagne.
- Morales, J. M., D. Fortin, J. L. Frair et E. H. Merrill. 2005. « Adaptive models for large herbivore movements in heterogeneous landscapes ». *Landscape Ecology*, vol. 20, n° 3, p. 301-316.
- Nelson, E. 1967. *Dynamical theories of Brownian motion*. Princeton University Press Princeton, NJ.
- Nyberg, J. B., B. G. Marcot et R. Sulyma. 2006. « Using Bayesian belief networks in adaptive management ». *Can. J. For. Res.*, vol. 36.
- Parker, D. C., S. M. Manson, M. A. Janssen, M. J. Hoffmann et P. Deadman. 2003. « Multi-Agent Systems for the Simulation of Land-Use and Land-Cover Change: A Review ». *Annals of the Association of American Geographers*, vol. 93, p. 314-337.

- Parrott, L., et H. Lange. 2004. « Use of interactive forest growth simulation to characterise spatial stand structure ». *Forest Ecology and Management*, vol. 194, n° 1-3, p. 29-47.
- Parrott, Lael, et Robert Kok. 2000. « Incorporating Complexity in Ecosystem Modelling ». *Complexity International*, vol. 7.
- Pearl, J. 1996. « Decision making under uncertainty ». *ACM Computing Surveys (CSUR)*, vol. 28, n° 1, p. 89-92.
- Pearl, J., et S. Russell. 2003. « Bayesian Networks ». *The Handbook of Brain Theory and Neural Networks*.
- Railsback, S. F. 2001. « Getting “results”: the pattern-oriented approach to analyzing natural systems with individual-based models ». *Natural Resource Modeling*, vol. 14, n° 3, p. 465–475.
- Railsback, SF, SL Lytinen et SK Jackson. 2006. « Agent-based simulation platforms: Review and development recommendations ». *Simulation*, vol. 82, n° 9, p. 609.
- Schank, Jeffrey C. . 2001. « Beyond reductionism: Refocusing on the individual with individual-based modeling ». *Complexity*, vol. 6, n° 3, p. 33-40.
- Schlesinger, Stewart. 1979. « Terminology for model credibility ». *Simulation*, vol. 32, n° 3, p. 103-104.
- Torrens, P., et I. Benenson. 2005. « Geographic Automata Systems ». *International Journal of Geographical Information Science*, vol. 19, n° 4, p. 385-412.
- Torrens, P. M., et A. Nara. 2007. « Modeling gentrification dynamics: A hybrid approach ». *Computers, Environment and Urban Systems*, vol. 31, n° 3, p. 337-361.
- Turchin, P. 1998. *Quantitative analysis of movement: measuring and modeling population redistribution in animals and plants*. Sinauer Associates.
- Turchin, P. 1991. « Translating Foraging Movements in Heterogeneous Environments into the Spatial Distribution of Foragers ». *Ecology*, vol. 72, n° 4, p. 1253-1266.
- Wiegand, T., F. Jeltsch, I. Hanski et V. Grimm. 2003. « Using pattern-oriented modeling for revealing hidden information: a key for reconciling ecological theory and application ». *Oikos*, vol. 100, n° 2, p. 209-222.
- Wolfram, S. 1984. « Universality and complexity in cellular automata ». *Physica D: Nonlinear Phenomena*, vol. 10, n° 1-2, p. 1-35.

- Wootton, J. T. 2001. « Local interactions predict large-scale pattern in empirically derived cellular automata ». *Nature*, vol. 413, n° 6858, p. 841-4.
- Wuensche, A. 1994. « The Ghost in the Machine: Basins of Attraction of Random Boolean Networks ». *Artificial Life III*, p. 465-501.